

MAXIMALES
ERFOLG

Microsoft®

Egbert Jeschke, Eckehard Pfeifer,
Helmut Reinke, Sara Unverhau,
Bodo Fienitz

Excel

Formeln & Funktionen

Einführung in die Nutzung von
Formeln und Funktionen

Komplette Referenz
aller Funktionen
von **Excel 2007**
bis **2013**



Hunderte von
Anwendungsbeispielen
zum Download

MAXIMALER ERFOLG
MINIMALER PREIS: € 19,90 [D]

3., aktualisierte und erweiterte Auflage

Microsoft
Press

Egbert Jeschke, Eckehard Pfeifer, Helmut Reinke, Sara Unverhau, Bodo Fienitz

Microsoft Excel – Formeln & Funktionen Das Maxibuch

3. Auflage

The logo for Microsoft Press, featuring the word "Microsoft" in white and "Press" in blue, both in a sans-serif font, set against a black square background.

Microsoft[®]
Press

Egbert Jeschke, Eckehard Pfeifer, Helmut Reinke, Sara Unverhau, Bodo Fienitz:
Microsoft Excel – Formeln & Funktionen – Das Maxibuch, 3. Auflage
Copyright © 2013 by O'Reilly GmbH & Co. KG

Das in diesem Buch enthaltene Programmmaterial ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Autor, Übersetzer und der Verlag übernehmen folglich keine Verantwortung und werden keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programmmaterials oder Teilen davon entsteht.

Das Werk einschließlich aller Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die in den Beispielen verwendeten Namen von Firmen, Organisationen, Produkten, Domänen, Personen, Orten, Ereignissen sowie E-Mail-Adressen und Logos sind frei erfunden, soweit nichts anderes angegeben ist. Jede Ähnlichkeit mit tatsächlichen Firmen, Organisationen, Produkten, Domänen, Personen, Orten, Ereignissen, E-Mail-Adressen und Logos ist rein zufällig.

Kommentare und Fragen können Sie gerne an uns richten:

Microsoft Press Deutschland
Konrad-Zuse-Straße 1
85716 Unterschleißheim
E-Mail: mspressde@oreilly.de

15 14 13 12 11 10 9 8 7 6 5 4 3 2 1
15 14 13

ISBN 978-3-86645-235-0 PDF-ISBN 978-3-8483-3047-8
EPUB-ISBN 978-3-8483-0182-9 MOBI-ISBN 978-3-8483-1183-5

© 2013 O'Reilly Verlag GmbH & Co. KG
Balthasarstr. 81, 50670 Köln
Alle Rechte vorbehalten

Fachlektorat: Georg Weiherer, Münzenberg
Korrektorat: Dorothee Klein, Judith Klein, Karin Baeyens, Siegen
Layout und Satz: Cordula Winkler, mediaService, Siegen (www.mediaservice.tv)
Umschlaggestaltung: Hommer Design GmbH, Haar (www.HommerDesign.com)
Gesamtherstellung: Kösel, Krugzell (www.KoeselBuch.de)

Übersicht

Vorwort	21
Teil A Einstieg und Grundlagen – Nicht aller Anfang ist schwer	27
1 Der Einstieg in Excel 2013	29
2 Von der Zahl zur Formel	81
3 Formeln und Funktionen	115
Teil B Aufbauwissen – Excel setzt keine Grenzen	135
4 Funktionen im besonderen Einsatz	137
5 Benutzerdefinierte Funktionen	169
Teil C Referenz – Alle Funktionen im Detail	195
6 Datums- und Zeitfunktionen	197
7 Text- und Datenfunktionen	229
8 Logische Funktionen	261
9 Nachschlage- und Verweisfunktionen	277
10 Informationsfunktionen	311
11 Statistische Funktionen	337
12 Datenbankfunktionen	567
13 Cubefunktionen	601
14 Finanzmathematische Funktionen	617
15 Mathematische und trigonometrische Funktionen	709
16 Konstruktionsfunktionen	803
17 Webfunktionen	861

Teil D Anhang	875
A Funktionslisten alphabetisch	877
B Funktionen nach Kategorie	937
C Die Beispieldateien zum Buch	957
Beispielverzeichnis	961
Stichwortverzeichnis	965

Inhaltsverzeichnis

Vorwort	21
Teil A Einstieg und Grundlagen – Nicht aller Anfang ist schwer	27
1 Der Einstieg in Excel 2013	29
Szenario & Ziele	30
Das Gesamtpaket macht's – Look und Inhalt	30
Nun geht's los ... Die Neuheiten in Excel 2013	32
ASAP – Die Schnellanalyse	34
Daten ergänzen – oder lieber ergänzen lassen	37
Der Tippgeber – Empfohlene Diagramme und ihre Bearbeitung	38
Ganz schön »schnittig« – Filtern von Tabellendaten mit Datenschnitten	43
Kompatibel oder nicht? Excel und die Dateiformate	44
»Alle wissen Bescheid statt ich weiß Bescheid« – Daten online zur Verfügung stellen	47
Fazit	51
Aus Daten werden Erkenntnisse – Arbeiten mit der PivotTable	51
Neue Funktionen in der Kurzübersicht	52
Szenario und Beispiel	53
Wer mit wem – Tabellen verknüpfen, Beziehungen erstellen	55
Auf den Punkt – Arbeiten mit der Timeline	65
Die Frage nach dem »Wie?« – PivotTable-Empfehlungen	67
Interaktiv und visuell – Microsoft Power View	69
Volle Kraft voraus: Das ist Power View	69
Neuigkeiten im Überblick	70
Erstellen eines Power View-Blatts	72
Fazit	79
2 Von der Zahl zur Formel	81
Eingeben von Formeln	82
Operatoren	82
Arithmetische Operatoren	83
Vergleichsoperatoren für die Logik	84
Der Textoperator &	85
Bezugsoperatoren	86

Bezüge in Formeln	89
Relative Bezüge	90
Absolute Bezüge	90
Gemischte Bezüge	92
Was ist ein Zirkelbezug?	94
Matrixformeln	96
Tipps & Tricks	97
Formelansicht ein/aus	97
Formeln in mehrere Zellen eingeben	98
Formeln in mehrere Blätter eingeben	98
Formelzellen schnell markieren	99
Auf welche Zellen bezieht sich die Formel?	99
Formeln kopieren/verschieben	100
Formelerggebnisse in Festwerte umwandeln	107
Schnelle Umrechnung vorhandener Werte	108
Formeln schützen	108
Formeln verstecken	108
Art der Formelberechnung bestimmen	109
Formeln analysieren	110
3 Formeln und Funktionen	115
Was kennzeichnet Tabellenfunktionen?	117
Syntax einer Funktion	117
Argumente als Rechendaten einer Funktion	118
Eingabe einer Funktion	121
Den Funktions-Assistenten nutzen	122
Bearbeitung von Formeln	130
Tipps & Tricks	130
Argumente anzeigen	130
Eine komplexe Formel in allen Mappen verfügbar machen	131
Teilberechnungen in Formeln	132
Dokumentation von Formeln in Excel 2013	133
Teil B Aufbauwissen – Excel setzt keine Grenzen	135
4 Funktionen im besonderen Einsatz	137
Funktionen in Namen	138
Aktuelle Informationen abrufen	138
Zahlungsziele als »Textbausteine«	141
Dynamische Bereichsnamen	142
Funktionen in der bedingten Formatierung	143
Die Wochenenden farbig hervorheben	145
Wochenenden und Feiertage hervorheben	146
Identische Werte hervorheben	149

Listendrucklayout	154
Top 3 hervorheben	155
Zellen mit Leerzeichen hervorheben	156
Hilfslinien für die Navigation in Arbeitsblättern	157
Datengruppen absetzen	160
Teilergebnisse formatieren	161
Tipps zur bedingten Formatierung	162
Funktionen bei der Datenprüfung	163
Zellschutz mithilfe der Datenprüfung	163
Variable Listenbereiche	164
Die Eingabe mit Formeln einschränken	165
5 Benutzerdefinierte Funktionen	169
Eine benutzerdefinierte Tabellenfunktion erstellen	171
Die Funktion <i>Kreisfläche()</i>	172
Die Funktion <i>Viereckfläche()</i>	179
Funktionen mit mehreren Argumenten	179
Funktionen mit optionalen Argumenten	180
Verzweigungen mit logischen Bedingungen	181
Die Funktion <i>FlächeKrSeg()</i>	186
Die Funktion <i>Quersumme()</i>	187
Die Funktion <i>Kreisfläche1()</i>	190
Die Funktionen im Funktions-Assistenten	193
Funktionen in einem Add-In speichern	194
Teil C Referenz – Alle Funktionen im Detail	195
6 Datums- und Zeitfunktionen	197
ARBEITSTAG()	201
ARBEITSTAG.INTL()	202
BRTEILJAHRE()	204
DATEDIF()	205
DATUM()	207
DATWERT()	208
EDATUM()	209
HEUTE()	210
ISOKALENDERWOCHE()	211
JAHR()	212
JETZT()	213
KALENDERWOCHE()	214
MINUTE()	215
MONAT()	216
MONATSENDE()	217

NETTOARBEITSTAGE()	218
NETTOARBEITSTAGE.INTL()	219
SEKUNDE()	220
STUNDE()	221
TAG()	222
TAGE()	222
TAGE360()	223
WOCHENTAG()	225
ZEIT()	227
ZEITWERT()	228
7 Text- und Datenfunktionen	229
ASC()	231
BAHTTEXT()	232
CODE()	232
DM()	234
ERSETZEN() / ERSETZENB()	235
FEST()	236
FINDEN() / FINDENB()	237
GLÄTTEN()	238
GROSS()	239
GROSS2()	240
IDENTISCH()	241
KLEIN()	242
LÄNGE() / LÄNGEB()	243
LINKS() / LINKSB()	244
PHONETIC()	245
RECHTS() / RECHTSB()	246
SÄUBERN()	247
SUCHEN() / SUCHENB()	248
T()	250
TEIL() / TEILB()	250
TEXT()	252
UNICODE()	253
UNIZEICHEN()	253
VERKETTEN()	254
WECHSELN()	255
WERT()	256
WIEDERHOLEN()	257
ZAHLENWERT()	259
ZEICHEN()	260

8	Logische Funktionen	261
	FALSCH()	262
	NICHT()	264
	ODER()	265
	UND()	267
	WAHR()	268
	WENN()	269
	WENNFEHLER()	272
	WENNV()	273
	XODER()	274
9	Nachschlage- und Verweisfunktionen	277
	ADRESSE()	279
	BEREICH.VERSCHIEBEN()	281
	BEREICHE()	284
	FORMELTEXT	285
	HYPERLINK()	286
	INDEX()	287
	INDIREKT()	292
	MTRANS()	293
	PIVOTDATENZUORDNEN()	295
	RTD()	295
	SPALTE()	297
	SPALTEN()	298
	SVERWEIS()	299
	VERGLEICH()	301
	VERWEIS()	304
	WAHL()	306
	WVERWEIS()	307
	ZEILE()	309
	ZEILEN()	310
10	Informationsfunktionen	311
	BLATT()	313
	BLÄTTER()	313
	FEHLER.TYP()	314
	INFO()	316
	ISTBEZUG()	317
	ISTFEHL()	318
	ISTFEHLER()	319
	ISTFORMEL()	320
	ISTGERADE()	321
	ISTKTEXT()	321

ISTLEER()	322
ISTLOG()	324
ISTNV()	325
ISTTEXT()	325
ISTUNGERADE()	327
ISTZAHL()	327
N()	328
NV()	329
TYP()	330
ZELLE()	333
11 Statistische Funktionen	337
Darstellung der Funktionen in diesem Kapitel	346
ACHSENABSCHNITT()	347
ANZAHL()	349
ANZAHL2()	351
ANZAHLLEEREZELLEN()	352
BESTIMMTHEITSMASS()	352
BETA.INV() / BETAINV()	355
BETA.VERT() / BETAVERT()	358
BINOM.INV() / KRITBINOM()	360
BINOM.VERT() / BINOMVERT()	362
BINOM.VERT.BEREICH()	365
CHIQU.INV.RE() / CHIINV()	367
CHIQU.INV()	369
CHIQU.TEST() / CHITEST()	370
CHIQU.VERT.RE() / CHIVERT()	374
CHIQU.VERT()	376
EXPON.VERT() / EXPONVERT()	376
F.INV.RE() / FINV()	379
F.INV()	382
FISHER()	383
FISHERINV()	387
F.TEST() / FTEST()	387
F.VERT.RE() / FVERT()	389
F.VERT()	392
GAMMA	393
GAMMA.INV() / GAMMAINV()	394
GAMMALN()	396
GAMMALN.GENAU()	397
GAMMA.VERT() / GAMMVERT()	398
GAUSS()	401

GEOMITTEL()	403
GESTUTZTMITTEL()	405
G.TEST() / GTEST()	406
HARMITTEL()	408
HÄUFIGKEIT()	411
HYPGEOM.VERT() / HYPGEOMVERT()	415
KGRÖSSTE()	418
KKLEINSTE()	420
KONFIDENZ.NORM() / KONFIDENZ()	421
KONFIDENZ.T()	425
KORREL()	426
KOVAR()	428
KOVARIANZ.P()	431
KOVARIANZ.S()	432
KURT()	433
LOGNORM.INV() / LOGINV()	436
LOGNORM.VERT() / LOGNORMVERT()	437
MAX()	439
MAXA()	440
MEDIAN()	441
MIN()	443
MINA()	444
MITTELABW()	445
MITTELWERT()	447
MITTELWERTA()	449
MITTELWERTWENN()	451
MITTELWERTWENNS()	454
MODUS.EINF() / MODALWERT()	457
MODUS.VIELF()	459
NEGBINOM.VERT() / NEGBINOMVERT()	460
NORM.INV() / NORMINV()	462
NORM.S.INV() / STANDNORMINV()	464
NORM.S.VERT() / STANDNORMVERT()	466
NORM.VERT() / NORMVERT()	468
PHI()	472
PEARSON()	472
POISSON.VERT() / POISSON()	476
QUANTIL()	479
QUANTIL.EXKL()	481
QUANTIL.INKL()	481
QUANTILSRANG()	482
QUARTILSRANG.EXKL()	483

QUARTILSRANG.INKL()	484
QUARTILE()	485
QUARTILE.EXKL()	488
QUARTILE.INKL()	488
RANG()	489
RANG.GLEICH()	490
RANG.MITTELW()	491
RGP()	492
RKP()	497
SCHÄTZER()	500
SCHIEFE()	503
SCHIEFE.P()	506
STABW.S() / STABW()	508
STABWA()	511
STABW.N() / STABWN()	513
STABWNA()	514
STANDARDISIERUNG()	516
STEIGUNG()	519
STFEHLERYX()	521
SUMQUADABW()	524
T.INV.2S() / TINV()	526
T.INV()	528
TREND()	528
T.TEST() / TTEST()	533
T.VERT.2S() / TVERT()	539
T.VERT()	541
T.VERT.RE()	541
VAR.S() / VARIANZ()	542
VARIANZA()	544
VAR.P() / VARIANZEN()	546
VARIANZENA()	547
VARIATION()	549
VARIATIONEN()	553
VARIATIONEN2()	555
WAHRSCBEREICH()	557
WEIBULL.VERT() / WEIBULL()	559
ZÄHLENWENN()	562
ZÄHLENWENNS()	564
12 Datenbankfunktionen	567
Argumente in Datenbankfunktionen	569
Arbeiten mit Datenbanken und -sätzen	570
Verwendung von Steuerelementen	572

Die Datenbankfunktionen im Überblick	572
Darstellung der Funktionen in diesem Kapitel	573
DBANZAHL()	573
DBANZAHL2()	575
DBAUSZUG()	577
DBMAX()	578
DBMIN()	580
DBMITTELWERT()	581
DBPRODUKT()	582
DBSTDABW()	584
DBSTDABWN()	586
DBSUMME()	587
DBVARIANZ()	590
DBVARIANZEN()	591
PIVOTDATENZUORDNEN()	593
13 Cubefunktionen	601
CUBEELEMENT()	604
CUBEELEMENTEIGENSCHAFT()	607
CUBEKPIELEMENT()	608
CUBEMENGE()	610
CUBEMENGENANZAHL()	612
CUBERANGELEMENT()	612
CUBEWERT()	614
14 Finanzmathematische Funktionen	617
AMORDEGRK()	623
AMORLINEARK()	625
AUFGELZINS()	627
AUFGELZINSF()	629
AUSZAHLUNG()	630
BW()	631
DIA()	634
DISAGIO()	635
DURATION()	636
EFFEKTIV()	638
GDA()	640
GDA2()	641
IKV()	642
ISPMT()	644
KAPZ()	645
KUMKAPITAL()	646
KUMZINSZ()	647

KURS()	649
KURSDISAGIO()	653
KURSFÄLLIG()	654
LIA()	656
MDURATION()	657
NBW()	658
NOMINAL()	660
NOTIERUNGBRU()	660
NOTIERUNGDEZ()	661
PDURATION()	662
QIKV()	663
RENDITE()	665
RENDITEDIS()	669
RENDITEFÄLL()	671
RMZ()	672
TBILLÄQUIV()	674
TBILLKURS()	675
TBILLRENDITE()	676
UNREGER.KURS()	677
UNREGER.REND()	680
UNREGLE.KURS()	681
UNREGLE.REND()	683
VDB()	684
XINTZINSFUSS()	686
XKAPITALWERT()	688
ZINS()	690
ZINSSATZ()	692
ZINSTERMNZ()	694
ZINSTERMTAGE()	695
ZINSTERMTAGNZ()	696
ZINSTERMTAGVA()	697
ZINSTERMVZ()	698
ZINSTERMZAHL()	699
ZINSZ()	700
ZSATZINVEST()	701
ZW()	702
ZW2()	704
ZZR()	705
15 Mathematische und trigonometrische Funktionen	709
ABRUNDEN()	714
ABS()	715
AGGREGAT()	716

ARABISCH()	718
ARCCOS()	719
ARCCOSHYP()	721
ARCCOT()	722
ARCCOTHYP()	724
ARCSIN()	725
ARCSINHYP()	726
ARCTAN()	727
ARCTAN2()	729
ARCTANHYP()	730
AUFRUNDEN()	731
BASIS()	732
BOGENMASS()	733
COS()	734
COSEC()	736
COSECHYP()	737
COSHYP()	738
COT()	739
COTHYP()	740
DEZIMAL()	742
EXP()	743
FAKULTÄT()	744
GANZZAHL()	745
GERADE()	745
GGT()	746
GRAD()	747
ISO.OBERGRENZE()	748
KGV()	749
KOMBINATIONEN()	750
KOMBINATIONEN2()	751
KÜRZEN()	752
LN()	752
LOG()	753
LOG10()	754
MDET()	755
MEINHEIT()	756
MINV()	757
MMULT()	759
OBERGRENZE()	760
OBERGRENZE.GENAU()	760
OBERGRENZE.MATHEMATIK()	761
PI()	763

POLYNOMIAL()	763
POTENZ()	764
POTENZREIHE()	765
PRODUKT()	767
QUADRATESUMME()	767
QUOTIENT()	768
REST()	769
RÖMISCH()	771
RUNDEN()	772
SEC()	773
SECHYP()	775
SIN()	776
SINHYP()	778
SUMME()	779
SUMMENPRODUKT()	780
SUMMEWENN()	781
SUMMEWENNS()	783
SUMMEX2MY2()	785
SUMMEX2PY2()	786
SUMMEXMY2()	787
TAN()	788
TANHYP()	790
TEILERGEBNIS()	792
UNGERADE()	794
UNTERGRENZE()	794
UNTERGRENZE.GENAU()	794
UNTERGRENZE.MATHEMATIK()	795
VORZEICHEN()	797
VRUNDEN()	798
WURZEL()	799
WURZELPI()	800
ZUFALLSBEREICH()	800
ZUFALLSZAHL()	801
ZWEIFAKULTÄT()	802
16 Konstruktionsfunktionen	803
Umwandlungsfunktionen	806
BININDEZ()	809
BININHEX()	810
BININOKT()	811
DEZINBIN()	812
DEZINHEX()	813
DEZINOKT()	814

HEXINBIN()	815
HEXINDEZ()	816
HEXINOKT()	817
OKTINBIN()	818
OKTINDEZ()	819
OKTINHEX()	819
UMWANDELN()	820
Funktionen für Bitoperationen	824
BITLVERSCHIEB()	824
BITODER()	825
BITUND()	826
BITRVERSCHIEB()	826
BITXODER()	827
Funktionen für komplexe Zahlen	828
IMABS()	830
IMAGINÄRTEIL()	831
IMAPOTENZ()	832
IMARGUMENT()	832
IMCOS()	833
IMCOSHYP()	834
IMCOSEC()	834
IMCOSECHYP()	835
IMCOT()	835
IMDIV()	836
IMEXP()	836
IMKONJUGIERTE()	837
IMLN()	838
IMLOG10()	838
IMLOG2()	839
IMPRODUKT()	839
IMREALTEIL()	840
IMSEC()	840
IMSECHYP()	841
IMSIN()	841
IMSINHYP()	842
IMSUB()	842
IMSUMME()	843
IMTAN()	844
IMWURZEL()	844
KOMPLEXE()	845
Funktionen der höheren Mathematik	846
BESSELI()	847
BESSELJ()	849
BESSELK()	850
BESSELY()	852

GAUSSE.GENAU() bzw. GAUSSFEHLER()	853
GAUSSFKOMPL.GENAU() bzw. GAUSSFKOMPL()	855
Sprungfunktionen	856
DELTA()	856
GGANZZAHL()	858
17 Webfunktionen	861
URLCODIEREN()	862
WEBDIENST()	863
XMLFILTERN()	865
Was ist XML?	867
Was sind Namespaces?	869
Was ist XPath?	871
Teil D Anhang	875
A Funktionslisten alphabetisch	877
Funktionsliste Deutsch-Englisch	878
Funktionsliste Englisch-Deutsch	907
B Funktionen nach Kategorie	937
Datums- und Zeitfunktionen	938
Text- und Datenfunktionen	939
Logische Funktionen	940
Nachschlage- und Verweisfunktionen	940
Informationsfunktionen	941
Statistische Funktionen	942
Datenbankfunktionen	947
Cubefunktionen	948
Finanzmathematische Funktionen	948
Mathematische und trigonometrische Funktionen	951
Konstruktionsfunktionen	954
Webfunktionen	956
C Die Beispieldateien zum Buch	957
Beispielverzeichnis	961
Stichwortverzeichnis	965

Vorwort

Wir freuen uns, Ihnen die dritte Neuauflage unseres Maxi-Funktionsbuchs vorlegen zu können. Seit wir dieses Buch geschrieben haben, sind für Excel-Verhältnisse zwei kleine Revolutionen passiert: Vor wenigen Jahren kam Excel 2007 mit vielen neuen Eigenschaften, neuer Oberfläche und Möglichkeiten auf den Markt. Nicht lange danach erschien Excel 2010 mit einer Vielzahl neuer Funktionen, einer noch tieferen Integration in das Microsoft SharePoint System und gleichzeitig erschien mit dem Microsoft PowerPivot-Add-In ein mächtiges Werkzeug zur Datenanalyse mit Excel. Im Zusammenspiel mit den PowerPivot- und den Excel-Services im SharePoint wird Excel mehr und mehr zum professionellen Berichtsdesigner für Webanwendungen.

Seit wenigen Monaten steht uns Excel 2013 zur Verfügung, das sich abermals mit Veränderungen in der Oberfläche und Bedienung, neuen Funktionen und neuen Möglichkeiten präsentiert. Was die Datenanalysten freut: Microsoft stellt neben dem PowerPivot-Add-In das Power View-Add-In für Excel 2013 zur Verfügung. Wir werden darauf eingehen.

Es war also erneut notwendig, das erste komplette deutsche Kompendium zu den Excel-Arbeitsblattfunktionen zu überarbeiten. Wir hoffen, dass Sie, liebe Leserinnen und Leser, das neue Konzept begrüßen, und Ihnen dieses Nachschlagewerk viele Anregungen und Hilfestellungen geben wird.

Auch in diesem Vorwort möchten wir uns bei unseren Lektoren Thomas Braun-Wiesholler, Sylvia Hasselbach und Georg Weiherer für ihre Inspiration, Geduld und Mühe bedanken. Ausdrücklich bedanken möchten wir uns bei Jens Bock, der an den früheren Auflagen mitwirkte und seinen Beitrag im Kapitel über die mathematischen Funktionen leistete.

Wir haben versucht, in allen Kapiteln fundierte Beschreibungen, praxisnahe Beispiele und Lösungen zu bringen und die Materialfülle fehlerfrei darzubieten. Ob es uns gelungen ist, diesen Ansprüchen gerecht zu werden, entscheiden Sie. Wir sind Realisten und wissen, dass man ein Buch wie dieses trotz aller Bemühungen immer noch besser machen kann.

Wir danken deshalb an dieser Stelle allen Leserinnen und Lesern, die uns bisher mit Fragen und Hinweisen bei der Verbesserung dieses Buchs geholfen haben, und freuen uns auf Ihre Kritiken, Vorschläge und Hinweise zu diesem Buch.

Wer war an diesem Buch beteiligt?

Wir – das sind Excel-Spezialisten aus verschiedenen Berufen mit sehr unterschiedlichen Erfahrungen. In diesem Handbuch finden Sie unsere gesammelten Erfahrungen aus der täglichen Arbeit mit Excel im Controlling, im Softwaretraining, in der Statistik, in Datenbankprojekten oder in der Programmierung. Sie werden sich fragen, wer sich dahinter verbirgt:

- ▶ **Sara Unverhau** ist internationale Diplom-Betriebswirtin mit den Spezialgebieten Marketing, Controlling und Sprachen. Sie ist im Bereich Marketing und Kommunikation tätig. Sara verfasste bereits zahlreiche Artikel und Buchkapitel zu betriebswirtschaftlichen Themen bei verschiedenen Verlagen.
- ▶ **Dr. Bodo Fienitz** ist promovierter Chemiker und seit 1991 hauptberuflich Dozent in der Erwachsenenbildung. Er unterrichtet neben naturwissenschaftlichen Fächern (Chemie, Physik und Biologie) regelmäßig EDV-bezogene Themen in Theorie und Anwendung. Die Spannweite geht von EDV-Grundlagen über Betriebssysteme bis zu kaufmännischen und

technischen Branchenlösungen. Den ersten Kontakt zur EDV hatte er im Jahre 1971 während seines Studiums der Elektrotechnik an der TU Berlin. Für seine Promotionsarbeit in Chemie an der FU Berlin während der 80er-Jahre setzte er rechnergesteuerte Prozesssteuerung, Messdatenerfassung und Verfahren zur numerischen Messdatenauswertung ein. Im Zusammenhang mit der notwendigen Visualisierung und Weiterverarbeitung der Daten wurde Dr. Fienitz zum Excel-Experten.

- ▶ **Egbert Jeschke** ist seit Jahren als Trainer und Anwendungsberater und -entwickler tätig. Auch bzw. gerade als jemand, der sich den relationalen und multidimensionalen Datenbanken verschrieben hat, weiß er die Stärken von Excel zu schätzen und zu nutzen. Fundierte Kenntnisse in SQL, VBA, SharePoint und den anderen Office-Anwendungen machen ihn zum Allrounder. Er ist Autor bzw. Mitautor verschiedener Bücher zum Einsatz von Office-Software bei Microsoft Press und anderen Verlagen.
- ▶ **Dr. Eckehard Pfeifer** aus Dresden ist habilitierter Mathematiker und als freiberuflicher Berater, Entwickler und Trainer tätig. Er ist Microsoft Certified Application Developer .NET und hat sich im Wesentlichen auf das Office-Umfeld spezialisiert. Als Autor schreibt er für verschiedene Fachmagazine und Lose-Blatt-Sammlungen. Er ist Mitautor verschiedener Bücher bei Microsoft Press (Excel-Handbuch, PowerPoint-Handbuch, Excel-Programmierrezepte, Office 2007-Programmierung). Sie erreichen ihn über seine Website www.dr-e-pfeifer.net.
- ▶ **Helmut Reinke**, der Spiritus Rector dieses Teams, steht für zahlreiche Veröffentlichungen zu den Office-Anwendungen, zu betriebswirtschaftlichen Lösungen, Projektmanagement und zur Unternehmensführung. Er ist Organisationsprogrammierer mit den Schwerpunkten Erfolgs- und Unternehmensplanung und ein Excel-Guru der ersten Stunde. Zu seiner langjährigen Tätigkeit in Vertrieb und Handel kommt eine fundierte didaktische Erfahrung, die er als Dozent und Schulungsleiter gewonnen hat.

Genug der Vorrede. Viele für Sie hoffentlich nützliche Seiten liegen vor Ihnen. Machen Sie etwas daraus, und wenn Ihnen das Buch gefällt – bitte behalten Sie es nicht für sich. Schreiben Sie uns, was Ihnen gefällt und vor allem, was wir besser machen müssen.

Wir versprechen Ihnen, dass wir bei Fragen und Schwierigkeiten für Sie da sind. Schreiben Sie uns an diese Adresse: info@mindbusiness.de.

Mehr Informationen darüber, womit wir uns sonst noch beschäftigen, finden Sie in Internet unter <http://www.mindbusiness.de>.

Wie ist dieses Buch aufgebaut?

Mit dieser Auflage machen wir einen Schnitt: Grundsätzlich haben wir die Tests zu den beschriebenen Funktionen in den Versionen 2007 bis 2013 von Excel vorgenommen. Wir haben beschlossen, ab dieser Auflage die Excel-Versionen bis 2003, von wenigen Ausnahmen abgesehen, nicht mehr zu behandeln. Einerseits ist diese Version in den Büros und bei den Privatanwendern seltener geworden. Andererseits haben wir alle Funktionen von Excel 2003 und früheren Versionen in den bisherigen Auflagen besprochen.

Die meisten Beschreibungen, mit Ausnahme 2007/2010/2013-spezifischer Merkmale, können jedoch auch von den Lesern genutzt werden, die mit Microsoft Excel 97-2003 arbeiten. Wo es notwendig (und uns bekannt) war, haben wir auf Besonderheiten hinsichtlich der Versionen hingewiesen.

Bestimmte Textstellen haben wir besonders gekennzeichnet, damit Ihre Aufmerksamkeit auf die wichtigen Aussagen gerichtet wird. Dabei gibt es diese Kennzeichnungen:

Kategorie	Bedeutung
Hinweis	Wissenswerte Zusatzhinweise zum Thema
Wichtig	Macht Sie auf Fakten aufmerksam, die Sie unbedingt wissen und beachten sollten
Tipp	Verrät Tipps und Tricks im gerade behandelten Kontext

Das Buch ist in drei Abschnitte unterteilt. Dem Abschnitt »Einstieg und Grundlagen – Nicht aller Anfang ist schwer« folgt der Abschnitt »Aufbauwissen – Excel setzt keine Grenzen«. Der dritte Abschnitt »Referenz – Alle Funktionen im Detail« mit den Funktionsbeschreibungen macht naturgemäß den Hauptteil des Buchs aus. Die Anhänge beschließen das Buch.

Abschnitt »Einstieg und Grundlagen – Nicht aller Anfang ist schwer«

In diesem Abschnitt finden Sie die drei einführenden Kapitel zur Arbeit mit Excel, besonders der Version 2013, im Allgemeinen und mit Formeln und Funktionen im Besonderen. Nach den Neuigkeiten in Excel 2013 haben wir hier für die Einsteiger, aber auch für die Profis unter Ihnen einen Abriss mit allen wichtigen Hinweisen, Anleitungen, Tipps und Tricks zur Arbeit mit Formeln und zum Einsatz der Tabellenfunktionen zusammengestellt.

Abschnitt »Aufbauwissen – Excel setzt keine Grenzen«

Die beiden anschließenden Kapitel zeigen Ihnen, dass Sie die meisten Funktionen nicht nur im Tabellenblatt einsetzen können und wie Sie Ihre eigenen Funktionen programmieren und einsetzen können.

Das Kapitel 4 »Funktionen im besonderen Einsatz« zeigt Beispiele für die Anwendung der Funktionen in Namen, in der bedingten Formatierung und bei der Gültigkeits- bzw. Datenprüfung. Wir denken, dass Sie hier viele Anregungen und Hinweise für Ihre Excel-Lösungen bekommen.

Das Kapitel 5 »Benutzerdefinierte Funktionen« führt Sie in die Programmierwelt unter Excel ein. Alle Tabellenfunktionen, die Sie noch in Excel vermissen, können Sie selbst erstellen und dann wie die eingebauten Funktionen verwenden. Wie Sie dabei vorgehen, erfahren Sie in diesem Kapitel.

Abschnitt »Referenz – Alle Funktionen im Detail«

Dies ist der Hauptabschnitt und -inhalt dieses Buchs. Die Kapitel 6 bis 17 enthalten die nach Kategorien unterteilten Funktionsbeschreibungen. Die Funktionen sind innerhalb der Kapitel alphabetisch aufgeführt. Hier die Inhalte der Kapitel:

Kapitel 6	Datums- und Zeitfunktionen
Kapitel 7	Text- und Datenfunktionen
Kapitel 8	Logische Funktionen
Kapitel 9	Nachschlage- und Verweisfunktionen
Kapitel 10	Informationsfunktionen
Kapitel 11	Statistische Funktionen
Kapitel 12	Datenbankfunktionen
Kapitel 13	Cubefunktionen (Neu ab Excel 2007)
Kapitel 14	Finanzmathematische Funktionen
Kapitel 15	Mathematische und trigonometrische Funktionen
Kapitel 16	Konstruktionsfunktionen
Kapitel 17	Webfunktionen (Neu ab Excel 2013)

Anhänge

Wir haben diesem Buch drei Anhänge hinzugefügt. Im Anhang A finden Sie die alphabetisch sortierten Funktionslisten – »Deutsch-Englisch« und »Englisch-Deutsch«. Damit sollten Sie in der Lage sein, jede Funktion auch in der jeweils anderen Sprache schnell benennen zu können. Der Anhang B enthält die deutsche Funktionsliste nach Kategorien sortiert und im Anhang C finden Sie eine Übersicht der Beispieldateien zu diesem Buch.

Beispieldateien zum Buch

Zum besseren Verständnis finden Sie innerhalb der Beispieldateien zu diesem Buch für jede Excel-Funktion ein kurzes oder auch umfangreicheres Beispiel. Gekennzeichnet sind die Beispiele im Buch durch das nebenstehend dargestellte Symbol. Sie können eine ZIP-Datei mit allen Beispieldateien im Internet von einer der beiden folgenden Adressen herunterladen:



www.microsoft-press.de/support/9783866452350

oder

<http://msp.oreilly.de/support/2314/773>.

Eine Übersicht zum Inhalt der ZIP-Datei finden Sie im Anhang C.

Und nun wünschen wir Ihnen viel Erfolg bei der Arbeit mit Microsoft Excel und hoffen, dass dieses Buch Ihnen dabei hilft und Sie inspiriert!

Teil A

Einstieg und Grundlagen – Nicht aller Anfang ist schwer

Kapitel 1	Der Einstieg in Excel 2013	29
Kapitel 2	Von der Zahl zur Formel	81
Kapitel 3	Formeln und Funktionen	115

Kapitel 1

Der Einstieg in Excel 2013

Szenario & Ziele	30
Das Gesamtpaket macht's – Look und Inhalt	30
Nun geht's los ... Die Neuheiten in Excel 2013	32
Kompatibel oder nicht? Excel und die Dateiformate	44
Aus Daten werden Erkenntnisse – Arbeiten mit der PivotTable	51
Interaktiv und visuell – Microsoft Power View	69

In diesem Kapitel finden Sie den Einstieg in die Arbeit mit Excel 2013 und seinen neuen Funktionen. Anhand eines praktischen Anwendungsbeispiels möchten wir Sie neugierig machen auf die Möglichkeiten der neuen Version. Daher ist der nachfolgende Rundflug und Querschnitt durch die Excel-Praxis sicher für alle Leser informativ.

Szenario & Ziele

Für unser erstes Beispiel greifen wir auf Bevölkerungs- und Beschäftigungsdaten des statistischen Bundesamts zu. Damit sollen folgende Funktionen dargestellt werden:

- ▶ Schnellanalyse
- ▶ Sparklines
- ▶ Berechnungsergebnisse
- ▶ Datenvervollständigung
- ▶ Empfohlene Diagramme
- ▶ Datenschnitte
- ▶ Kompatibilität
- ▶ Daten veröffentlichen und freigeben



Die folgenden Praxisbeispiele für Excel 2013 finden Sie fertig im Ordner `\Ms5-235\Kap01` in der Arbeitsmappe `Excel_Uebungsbeispiel.xlsx`. In der Arbeitsmappe `Excel_Uebungsbeispiel_leer.xlsx` liegen nur die Daten vor und Sie können hier alle Beispiele selbst erstellen.

Das Gesamtpaket macht's – Look und Inhalt

Eines fällt sofort nach dem Start von Excel 2013 auf: Das Programm präsentiert sich in einem neuen Gewand – angepasst an das Betriebssystem Windows 8. Aufgeräumt und puristisch. Aber nicht nur das Aussehen, sondern natürlich auch der Funktionsumfang hat sich geändert und erweitert. So finden Sie viele neue Features, die Sie dabei unterstützen, noch schneller perfekte Ergebnisse zu erzielen – sei es in der Bearbeitung oder auch in der Darstellung von Daten.

Tipp Standardmäßig ist das Menüband in Excel 2013 ausgeblendet. Dies hat den Vorteil, dass Sie mehr Platz im Arbeitsbereich haben. Klicken Sie auf einen der Registernamen *START*, *EINFÜGEN*, *SEITEN-LAYOUT* usw., wird das Menüband mit all seinen Funktionen eingeblendet.

Möchten Sie jederzeit Zugriff auf die Funktionen im Menüband haben, können Sie es »anheften« – also fixieren. Damit ist es dauerhaft eingeblendet. Öffnen Sie dazu das Menüband und klicken Sie ganz rechts auf das Symbol mit dem Pinn.

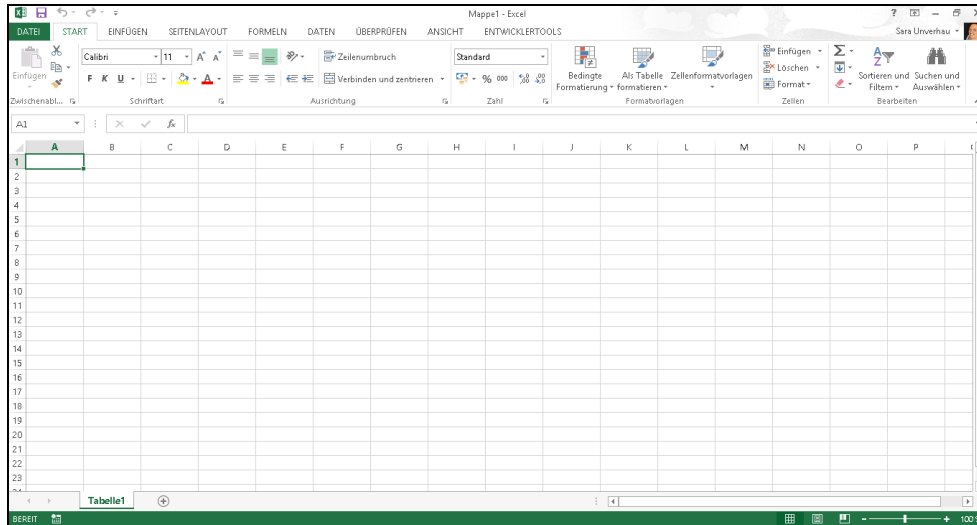


Abbildung 1.1: Der neue Look von Excel 2013



Abbildung 1.2: Sie möchten die Funktionen des Menübands immer im direkten Zugriff? Kein Problem.

Die zahlreichen Funktionen der Kategorien *Math. & Trigonom.*, *Finanzmathematik*, *Statistik*, *Technisch*, *Datum & Zeit*, *Informationen*, *Matrix*, *Datenbank*, *Logik* und *Text* helfen Ihnen, Zahlenkolonnen noch einfacher auszuwerten. Gänzlich neu im Funktionsumfang von Excel 2013 sind die *Web-Funktionen*, mit deren Hilfe auf vorhandene REST-kompatible (Representational State Transfer¹) Webdienste verwiesen werden kann, um von dort Daten zu empfangen.

Eine alphabetische Übersicht aller Funktionen, in der auch die neu in Excel 2013 eingeführten Funktionen enthalten sind, finden Sie im Anhang A.

Hinweis

Die Beschreibung der neuen Web-Funktionen von Excel 2013 finden Sie im Kapitel 17.

¹ REST beschreibt eine mögliche Architektur von Webservices und ihren Funktionen. Einen Einstieg in dieses Thema finden Sie im Internet u.a. hier:

- http://de.wikipedia.org/wiki/Representational_State_Transfer
- <http://it-republik.de/jaxenter/artikel/REST---Der-bessere-Web-Service-2158.html>
- <http://www.oio.de/public/xml/rest-webservices.htm>

Nun geht's los ... Die Neuheiten in Excel 2013

Learning by doing – legen wir los und starten wir einen praxisnahen Rundflug durch Excel 2013.

1. Öffnen Sie Excel 2013.

Schon hier – wenn das Arbeitsblatt noch nicht geöffnet ist – sehen Sie einige Änderungen. So können Sie neben einer neuen, leeren Arbeitsmappe zwischen zahlreichen Vorlagen aus den Bereichen Budget, Formulare, Berichte u.v.m. wählen, in denen die unterschiedlichsten Anforderungen aus dem Arbeitsalltag aufgenommen und bereits umgesetzt wurden. Das heißt, Ihnen ist damit sehr viel Konfigurations- und Entwurfsarbeit abgenommen. Entscheiden Sie sich für eine der Vorlagen, können Sie sich direkt auf Ihre zu verarbeitenden Daten konzentrieren.

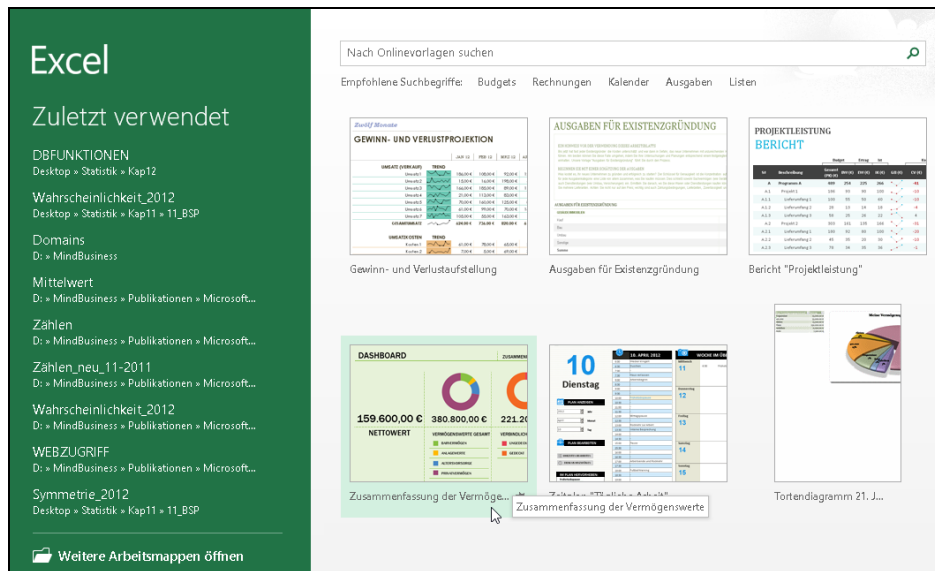


Abbildung 1.3: Das Vorlagenmanagement in Excel 2013 wurde erheblich erweitert

2. Führen Sie einen Doppelklick auf die von Ihnen gewünschte Vorlage aus. Um eine neue, leere Arbeitsmappe zu öffnen, klicken Sie auf den entsprechenden Eintrag.



Da wir uns im ersten Schritt auf die neue Datenbearbeitung in Excel 2013 – und nicht auf das Design – konzentrieren wollen, öffnen Sie bitte die Datei *Excel_Uebungsbeispiel_leer.xlsx* im Ordner *\Ms5-235\Kap01*.



Abbildung 1.4: Nicht nur auf die Vorlagen kann zugegriffen werden – auch auf die zuletzt verwendeten Dateien

Für das nachfolgende Beispiel haben wir uns verschiedener Daten aus dem statistischen Jahrbuch (Deutschland) bedient. Diese sollen nun Schritt für Schritt aufbereitet werden.

Sollten Sie jetzt zwei Arbeitsmappen, die neue, leere sowie das Übungsbeispiel geöffnet haben, stellen Sie fest, dass unterschiedliche Arbeitsmappen in 2013 in jeweils einem separaten Fenster geöffnet und angezeigt werden.

Hinweis

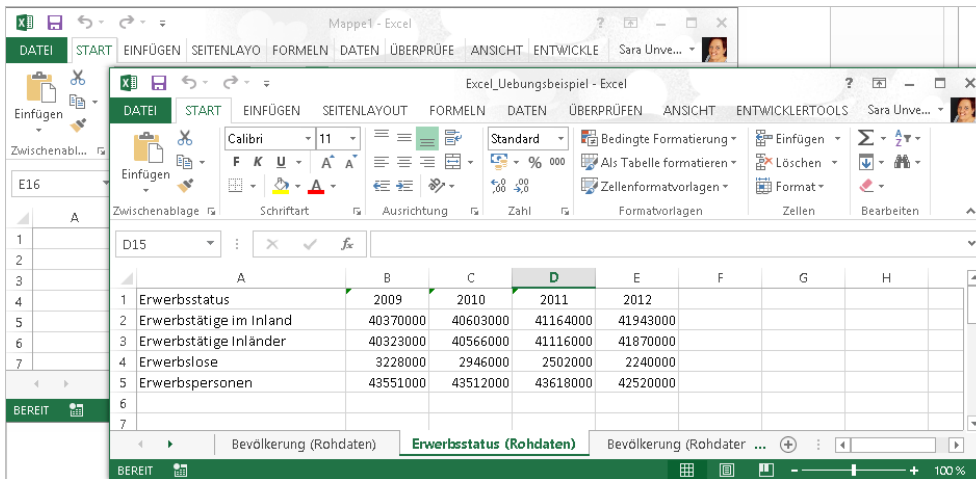


Abbildung 1.5: Behalten Sie den Überblick – mit den separaten Arbeitsmappen kein Problem

Die Aufteilung in mehrere Fenster macht es einfacher, an zwei Arbeitsmappen – und auch an mehreren Monitoren – gleichzeitig zu arbeiten.

3. Öffnen Sie in der Arbeitsmappe *Excel_Uebungsbeispiel_leer.xlsx* das Tabellenblatt *Bevölkerung (Rohdaten)*.
4. Um die dargestellten Bevölkerungszahlen zunächst in Tausenderwerten, und damit übersichtlicher, darzustellen, gehen Sie wie folgt vor:
 - ▶ Markieren Sie den Zellbereich *B2:D10* und klicken Sie auf das kleine Pfeilsymbol unten rechts in der Gruppe *Zahl* der Registerkarte *START*
 - ▶ In dem sich öffnenden Dialogfeld *Zahlen formatieren* klicken Sie in der Kategorie *Benutzerdefiniert* auf den Eintrag *###0*

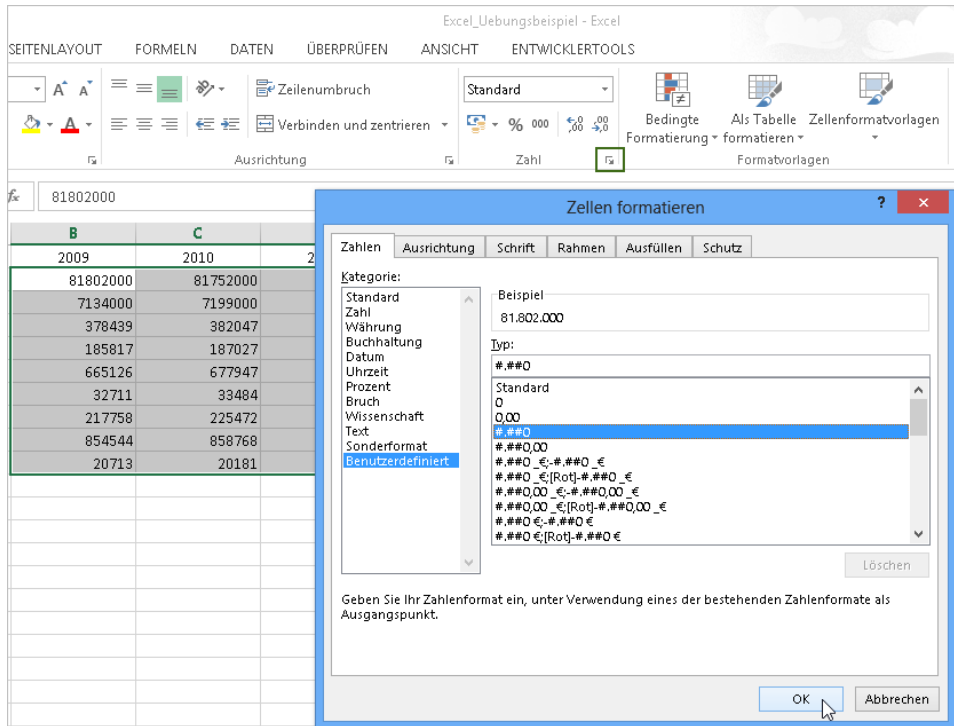


Abbildung 1.6: In der Zellenformatierung stehen Ihnen zahlreiche Optionen zur Verfügung

ASAP – Die Schnellanalyse

As soon as possible – oftmals schon ein Standardspruch unter den E-Mails der Kollegen. Manchmal nervig – aber manchmal reicht ein schnelles Diagramm oder eine erste fixe Analyse aus, um sich über Zahlenreihen einen Eindruck zu verschaffen. Excel 2013 bietet eine Funktion, mit der Sie Daten per Schnellanalyse auswerten können. Die Schnellanalyse ermöglicht es, in wenigen Schritten Ergebnisse wie Summen oder Mittelwerte zu berechnen, Formatierungen in Form von Symbolen vorzunehmen, Diagramme zu erstellen oder auch wichtige Tabelleninhalte über Sparklines zu visualisieren.

Sparklines gibt es seit der Version 2010 in Excel. Die sogenannten »Wortgrafiken« werden eingesetzt, um Zahlen in einem Linien-, Säulen- oder Gewinn- & Verlustdiagramm auf platz sparende Weise grafisch zu erklären. Das heißt, Sparklines stellen in einem stark miniaturisierten Diagramm die Entwicklung eines numerischen Werts dar, wodurch sich dieser einfacher interpretieren lässt.

Arbeiten mit Sparklines

Um die Entwicklung unserer Bevölkerungszahlen nun grafisch anzuzeigen, gehen Sie so vor:

1. Markieren Sie die Inhalte der Bevölkerungstabelle.
2. Nach dem Markieren der Daten erscheint sofort das Symbol für die *Schnellanalyse*.

81.802.000	81.752.000	81.844.000	
7.134.000	7.199.000	7.410.000	
378.439	382.047	377.816	
185.817	187.027	187.640	
665.126	677.947	662.685	
32.711	33.484	31.940	
217.758	225.472	224.744	
854.544	858.768	852.328	
20.713	20.181	20.373	

Abbildung 1.7: Die Schnellanalyse macht ihrem Namen alle Ehre

3. Klicken Sie auf das Symbol, öffnen sich die dahinter liegenden Analysefunktionen der Bereiche *FORMATIERUNG*, *DIAGRAMME*, *ERGEBNISSE*, *TABELLEN* und *SPARKLINES*.

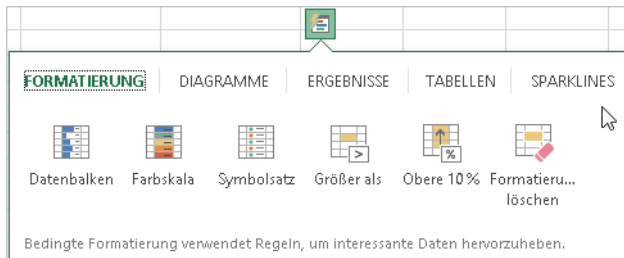


Abbildung 1.8: Die unterschiedlichen Kategorien der Schnellanalyse lassen keine Wünsche offen

4. Um die Daten in Form von Sparklines anzuzeigen, öffnen Sie die Registerkarte *SPARKLINES* und zeigen Sie mit der Maus auf eine der vorgeschlagenen Darstellungsformen an. Durch die Vorschaufunktion wird das Ergebnis sofort angezeigt.

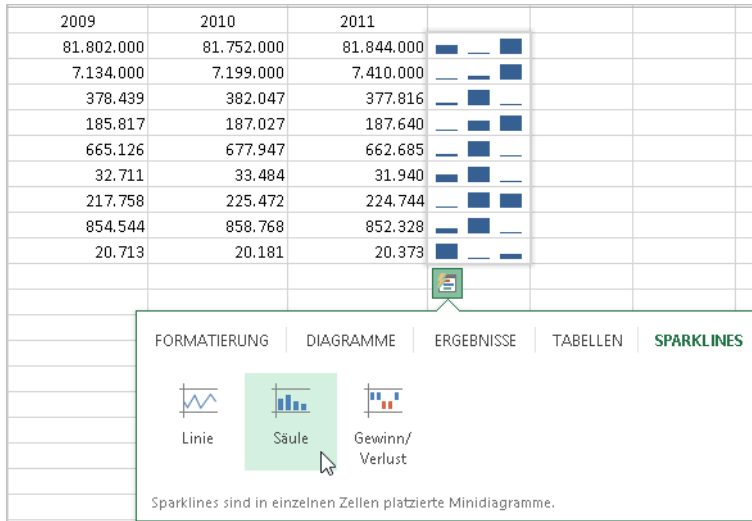


Abbildung 1.9: Entscheiden Sie selbst, welche Variante der Schnellanalyse Ihnen am besten gefällt

- Sagt Ihnen die dargestellte Form zu, klicken Sie auf Ihre Auswahl, um die Sparklines abschließend neben der Tabelle einzufügen.

Ergebnisse darstellen

Nachfolgend noch ein Beispiel für das Arbeiten mit der Schnellanalyse. Sie möchten für das Jahr 2012 den prozentualen Anteil erwerbstätiger Inländer sowie der Erwerbslosen im Verhältnis zur Anzahl aller Erwerbstätigen sehen.

- Öffnen Sie das Tabellenblatt *Erwerbsstatus (Rohdaten)* in der Excel-Datei *Excel_Uebungsbeispiel_leer.xlsx*.
- Markieren Sie in der Tabelle die Zellen *E3* und *E4*.
- Klicken Sie auf das Symbol für die Schnellanalyse und öffnen Sie die Registerkarte *ERGEBNISSE*.
- Wählen Sie den Eintrag *% Gesamt* (für die Spalte).

Hinweis

Klicken Sie im Fenster der Schnellanalyse nicht gleich auf das erste Symbol *% Gesamt*, sondern blättern Sie zunächst nach rechts. Beim ersten Symbol würde lediglich der Wert »100,00%« in der Zelle *E5* angegeben.

A	B	C	D	E	F	G	H
Erwerbsstatus	2009	2010	2011	2012			
Erwerbstätige im Inland	40.370.000	40.603.000	41.164.000	41.943.000			
Erwerbstätige Inländer	40.323.000	40.566.000	41.116.000	41.870.000	94,92%		
Erwerbslose	3.228.000	2.946.000	2.502.000	2.240.000	5,08%		
Erwerbstätige gesamt	43.551.000	43.512.000	43.618.000	44.110.000			

Abbildung 1.10: Erstellen Sie prozentuale Zusammenhänge – auf Knopfdruck

- Die entsprechenden Werte werden sofort angezeigt.

Daten ergänzen – oder lieber ergänzen lassen

Ein weiteres Highlight in puncto neue Funktionen ist sicherlich die Blitzvorschau – ein Assistent, der die Arbeit für Sie erledigt. Die Anwendung sucht bei Dateneingabe nach möglichen Erfassungsmustern. Sobald das Feature erkannt hat, welchen Vorgang Sie ausführen möchten, gibt es die restlichen Daten sofort komplett ein bzw. schlägt Ihnen die Vervollständigung nach dem erkannten Muster vor.

Achten Sie darauf, dass sich die aktive Zelle in dem Bereich befindet, der automatisch ausgefüllt werden soll.

Hinweis

Lassen Sie uns diese geniale Funktion in der Praxis verdeutlichen. Da es nicht zu den statistischen Bevölkerungszahlen passt, ein anderes kleines Beispiel.


Sie haben eine Liste mit Vor- und Nachnamen, die beide in einer Zelle erfasst sind. Den Personen in der Liste möchten Sie einen Brief schreiben, der mit einer personalisierten Anrede wie beispielsweise »Sehr geehrte Frau Unverhau« beginnt. Sie benötigen also lediglich den Nachnamen der gelisteten Personen.

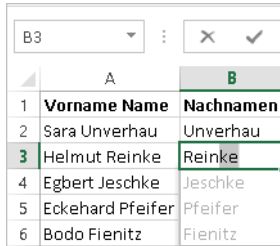
- Öffnen Sie in der Datei *Excel_Uebungsbeispiel_leer.xlsx* das Tabellenblatt *Serienbrief*. Sie sehen die Liste mit den bereits erfassten Namen Ihrer Adressaten.

	A	B
1	Vorname Name	Nachnamen
2	Sara Unverhau	
3	Helmut Reinke	
4	Egbert Jeschke	
5	Eckehard Pfeifer	
6	Bodo Fienitz	

Abbildung 1.11: Listen für die weitere Verarbeitung vorbereiten


- Klicken Sie in die Zelle B2 und geben Sie den Nachnamen der ersten Person, also in diesem Fall »Unverhau« ein.

3. Bestätigen Sie mit der -Taste und geben Sie in die Zelle B3 den Nachnamen »Reinke« ein.
4. Bereits jetzt hat die Blitzvorschau die Logik hinter Ihrer Eingabe erkannt und schlägt Ihnen die passende Lösung zum Ausfüllen der noch fehlenden Zellen vor.

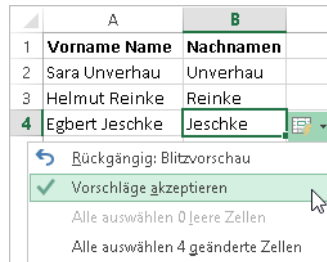


	A	B
1	Vorname Name	Nachnamen
2	Sara Unverhau	Unverhau
3	Helmut Reinke	Reinke
4	Egbert Jeschke	Jeschke
5	Eckehard Pfeifer	Pfeifer
6	Bodo Fienitz	Fienitz

Abbildung 1.12: Nutzen Sie die virtuelle Intelligenz, um Ihre Arbeit vervollständigen zu lassen

5. Wenn Sie mit dem von der Blitzvorschau vorgeschlagenem Muster einverstanden sind, bestätigen Sie durch Drücken der -Taste.

Tipp Haben Sie die Vervollständigung Ihrer Daten durch die Blitzvorschau bestätigt, erscheint das Symbol *Blitzvorschauoptionen*. Über dieses haben Sie die Möglichkeit, die Blitzvorschau rückgängig zu machen, die Vorschläge zu akzeptieren oder auch die geänderten Zellen zu markieren.



	A	B
1	Vorname Name	Nachnamen
2	Sara Unverhau	Unverhau
3	Helmut Reinke	Reinke
4	Egbert Jeschke	Jeschke

Abbildung 1.13: Nutzen Sie die Bearbeitungsoptionen der Blitzvorschau für Änderungen u.Ä.

Die Blitzvorschau ist definitiv ein toller Helfer in der täglichen Arbeit mit Microsoft Excel 2013.

Der Tipgeber – Empfohlene Diagramme und ihre Bearbeitung

Wer kennt das nicht: Man hat eine Tabelle mit qualitativ hochwertigen Daten und möchte deren Inhalt mithilfe eines Diagramms visualisieren. Nur mit welchem? Die Auswahl ist groß und wirklich schlecht ist keines – aber letztlich sucht man natürlich immer nach dem Optimum.

Hier kommt eine weitere neue Funktion in Excel 2013 zum Zug: »Empfohlene Diagramme«. Basis für die Diagrammvorschläge sind die von Ihnen markierten Tabellendaten. Über ein Vorschaufenster werden zusätzlich noch die Vorteile und Eigenschaften der jeweiligen Diagramme beschrieben.

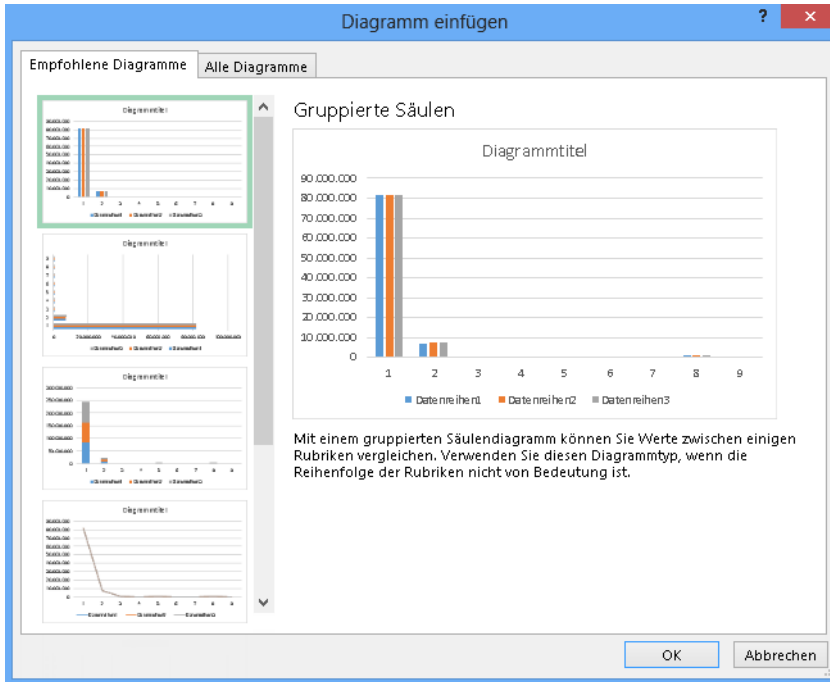


Abbildung 1.14: Eine erste Idee hilft oft, um das gewünschte Ziel zu erreichen

Ist nichts Passendes dabei oder haben Sie eine gänzlich andere Vorstellung, können Sie, wie in Abbildung 1.14, direkt über das Dialogfeld *Diagramm einfügen* die Registerkarte *Alle Diagramme* öffnen und dort aus der Liste aller vorhandenen Diagrammformen das für Sie passende auswählen.

Nachfolgend testen wir die Funktion in der Praxis:

1. Öffnen Sie in der Datei *Excel_Uebungsbeispiel_leer.xlsx* das Tabellenblatt *Erwerbsstatus (Rohdaten)*.
2. Markieren Sie die gesamte Tabelle.

	A	B	C	D	E
1		2009	2010	2011	2012
2	Erwerbstätige im Inland	40.370.000	40.603.000	41.164.000	41.943.000
3	Erwerbstätige Inländer	40.323.000	40.566.000	41.116.000	41.870.000
4	Erwerbslose	3.228.000	2.946.000	2.502.000	2.240.000
5	Erwerbstätige gesamt	43.551.000	43.512.000	43.618.000	44.110.000

Abbildung 1.15: Geben Sie vor, welche Werte Sie visualisieren möchten

3. Klicken Sie auf die Registerkarte *EINFÜGEN* und wählen Sie in der Gruppe *Diagramme* die Schaltfläche *Empfohlene Diagramme*.

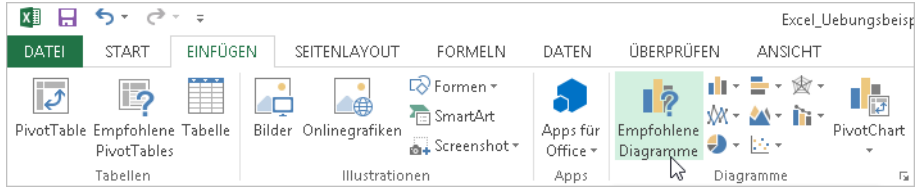


Abbildung 1.16: Welches Diagramm passt am besten aufgrund der markierten Daten?

4. Betrachten Sie nun im Dialogfeld *Empfohlene Diagramme* die vorgeschlagenen Optionen und entscheiden Sie sich für die passende Variante. In unserem Beispiel nehmen wir das Diagramm *Gruppierter Säulen/Linien*.

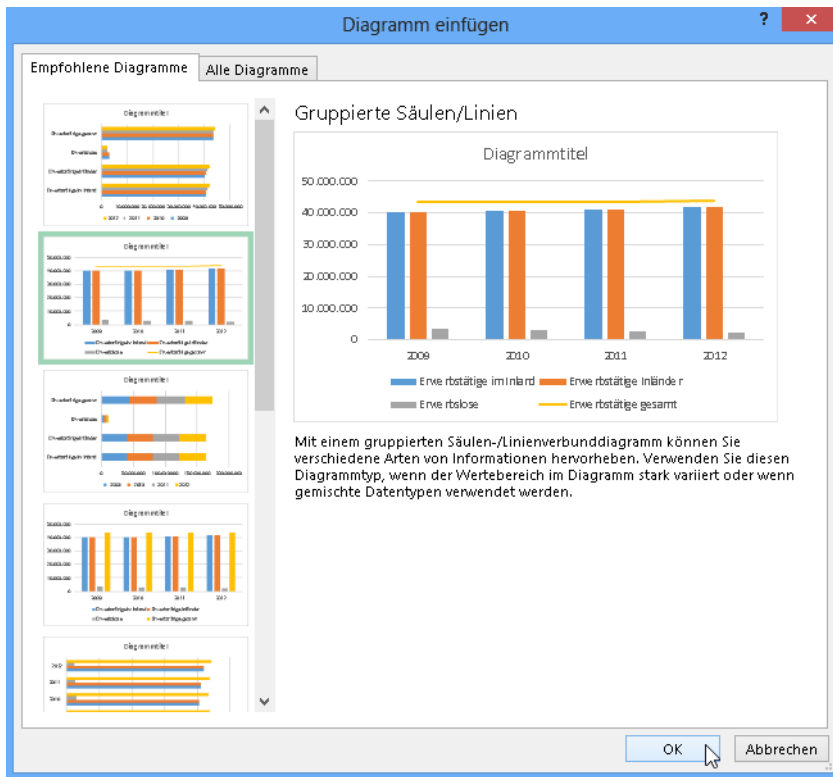


Abbildung 1.17: Die Vorschau der vorgeschlagenen Diagramme macht das mögliche Ergebnis sichtbar

5. Bestätigen Sie Ihre Wahl mit OK.

Das Diagramm wurde nun eingefügt. Auch hier sehen Sie ein Excel 2013-Novum.

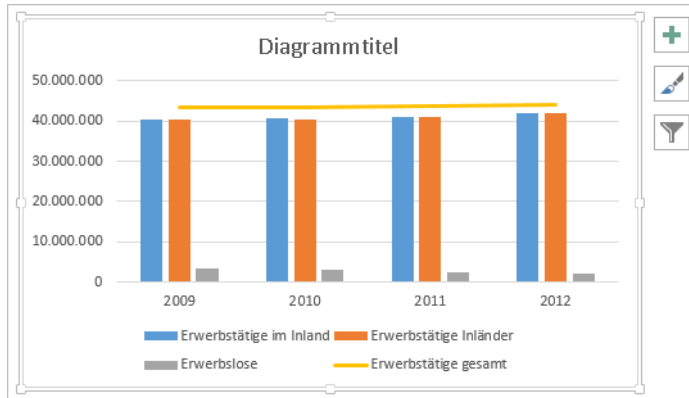


Abbildung 1.18: Das Diagramm kann sofort weiter angepasst werden – dank der Bearbeitungssymbole

Über die in Abbildung 1.18 dargestellten Symbole stehen Ihnen sofort verschiedene Formatierungsoptionen zu nachfolgenden Kategorien zur Verfügung:

- ▶ Diagrammelemente
- ▶ Diagrammformatvorlagen
- ▶ Diagrammfilter

Sie können so in Minutenschnelle Elemente wie den Achsentitel oder die Datenbeschriftung anpassen sowie das Aussehen des Diagramms oder auch die Daten verändern.

Möchten Sie umfassendere Änderungen an Ihrem Diagramm vornehmen, stehen Ihnen ab Excel 2013 auch die sogenannten Diagrammtools – bestehend aus den Registerkarten *ENTWURF* und *FORMAT* – zur Verfügung. Diese sind aktiv, wenn Ihr Diagramm ausgewählt bzw. selektiert ist.

Tip

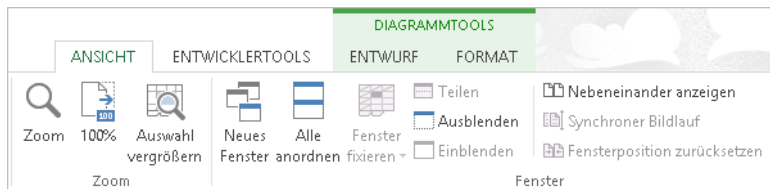


Abbildung 1.19: Die Diagrammtools unterstützen Sie dabei, Diagramme optimal zu visualisieren

Über die Diagrammtools haben Sie alle Möglichkeiten der Diagrammbearbeitung und -beschriftung.

Bedienung mit Touch

Übrigens – viele der bis hierhin genannten Bearbeitungsmöglichkeiten können Sie auch per Touchbedienung durchführen. Immer mehr Menschen verfügen über Tablet-PCs – ob nun zum Lesen oder auch zum effektiven Arbeiten. Mit Excel 2013 ist man in der Lage, viele Einstellungen auch mit dem Finger oder Stift vorzunehmen und vor allem Inhalte perfekt zu präsentieren.

Passen Sie zunächst die Symbolleiste für den Schnellzugriff an und aktivieren Sie im Drop-downmenü der *Symbolleiste für den Schnellzugriff* den Eintrag *Fingereingabe-/Mausmodus*. Anschließend können Sie schnell zwischen den Eingabemodi wechseln.

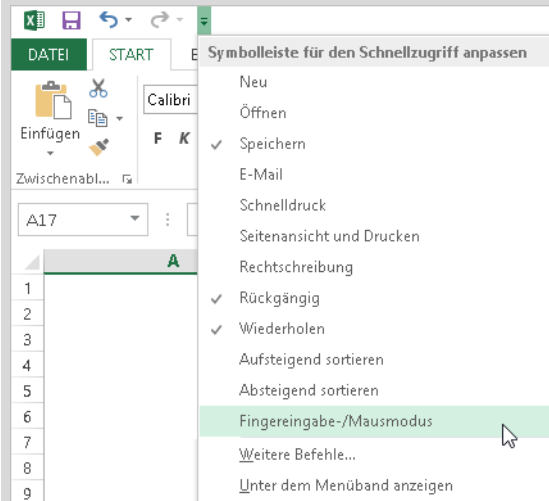


Abbildung 1.20: Excel in der mobilen Anwendung – heute ein Muss

Um im Fingereingabemodus schnell arbeiten zu können, ist es wichtig, die passenden Gesten zu kennen.

Tabelle 1.1
Wichtige
Touchgesten im
Überblick

Zweck	Geste
Zur Vollbildansicht wechseln	Tippen Sie in der Titelleiste auf die Schaltfläche <i>Menüband-Anzeigeoptionen</i> (rechts neben dem Fragezeichensymbol) und dann auf <i>Menüband automatisch ausblenden</i>
Zur Standardansicht wechseln	Tippen Sie auf die Schaltfläche <i>Menüband-Anzeigeoptionen</i> und dann auf <i>Registerkarten und Befehle anzeigen</i>
Die Symbolleiste für den Schnellzugriff anpassen	Drücken und halten Sie eine beliebige Schaltfläche auf der Symbolleiste für den Schnellzugriff gedrückt und lassen Sie sie dann los
Bildlauf durchführen	Tippen Sie auf das Dokument und ziehen Sie es nach oben und unten
Vergrößern	Spreizen Sie zwei Finger auseinander
Verkleinern	Drücken Sie zwei Finger zusammen
Die Einfügemarke platzieren	Tippen Sie in die Datei
Text markieren	Tippen Sie in den Text, und ziehen Sie dann den Auswahlziehpunkt ▶

Zweck	Geste
Text formatieren	Tippen Sie auf den markierten Text und tippen Sie dann in der Minisymbolleiste auf eine Formatierungsoption
Objekte verschieben	Tippen Sie auf das Objekt und ziehen Sie es
Objektgröße ändern	Tippen Sie auf das Objekt, dann auf den Ziehpunkt zur Größenänderung des Objekts und ziehen Sie diesen
Objekte drehen	Tippen Sie auf das Objekt, dann auf den Drehpunkt und ziehen Sie diesen
Eine Zelle bearbeiten	Doppeltippen Sie in die Zelle
Einen Datenbereich markieren	Tippen Sie auf eine Zelle und ziehen Sie dann den Auswahlziehpunkt
Den Inhalt einer Zelle löschen	Tippen Sie auf eine Zelle, um sie zu markieren, tippen Sie erneut darauf, um die Minisymbolleiste aufzurufen, und tippen Sie dann auf <i>Löschen</i>

Ganz schön »schnittig« – Filtern von Tabellendaten mit Datenschnitten

Wenn Sie eine Tabelle in Excel erstellen, werden automatisch Filtersteuerelemente zu den Tabellenüberschriften hinzugefügt. Soweit nichts Neues. Ebenso die Datenschnitte als solches, die es bereits in der Excel-Version 2010 gab, um Daten in einer PivotTable interaktiv zu filtern.

Datenschnitte können Sie ab Excel 2013 auch zum Filtern von Tabellendaten erstellen. Vorteil: Die einfach zu verwendenden Datenschnitte zeigen den jeweils aktuellen Filter, sodass Sie genau wissen, welche Daten Sie betrachten.

Lassen Sie uns die Funktion an einem Beispiel verdeutlichen:

1. Öffnen Sie in der Arbeitsmappe *Excel_Uebungsbeispiel_leer.xlsx* das Tabellenblatt *Bevölkerungsdaten* Tabelle.
2. Klicken Sie auf eine beliebige Stelle in der Tabelle, um im Menüband die kontextbezogene Registerkarte *TABELLENTOOLS* zu aktivieren. Öffnen Sie hier die Registerkarte *ENTWURF*.
3. Klicken Sie in der Gruppe *Tools* auf *Datenschnitt einfügen*.

Position	2009	2010	2011
Bevölkerungsstand	81.802.000	81.752.000	81.844.000
davon Ausländer	7.134.000	7.199.000	7.410.000
Eheschließungen	378.439	382.047	377.816
Ehescheidungen	185.817	187.027	187.640
Lebendgeborene	665.126	677.947	662.685
davon Ausländer	32.711	33.484	31.940
Nichtehelich Lebendgeborene	217.758	225.472	224.744
Gestorbene	854.544	858.768	852.328
dar.: Ausländer/-innen	20.713	20.181	20.378

Abbildung 1.21: Eine echte Bereicherung – der Datenschnitt funktioniert nicht mehr nur für die PivotTable

- Aktivieren Sie im Dialogfeld *Datenschnitt auswählen* das Kontrollkästchen für die Elemente, für die Datenschnitte erstellt werden sollen. In unserem Beispiel aktivieren Sie das Kontrollkästchen *Position*.

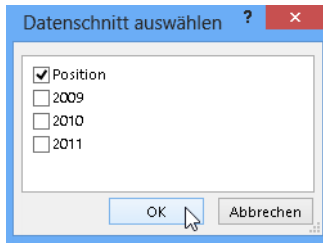


Abbildung 1.22: Nach welchen Kriterien Sie filtern möchten, bestimmen Sie

- Bestätigen Sie mit *OK*. Es wird nun für jede aktivierte Tabellenüberschrift ein Datenschnitt angezeigt, über den Sie jetzt beliebig filtern können.
- Klicken Sie im Datenschnitt auf das gewünschte Element, nach der die Tabelle gefiltert werden soll.

	A	B	C	D	E	F	G
1	Position	2009	2010	2011	Position Bevölkerungszustand dar.: Ausländer/-innen davon Ausländer Ehescheidungen Eheschließungen Gestorbene Lebendgeborene Nichtehelich Lebendge...		
2	Bevölkerungsstand	81.802.000	81.752.000	81.844.000			
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							

Abbildung 1.23: Einfacher geht's nicht – Datenkolonnen auf Klick filtern

Tipp Möchten Sie mehrere Elemente des Datenschnitts markieren, um die Tabelle danach zu filtern, halten Sie während des Markiervorgangs die **[Strg]**-Taste gedrückt.

Kompatibel oder nicht? Excel und die Dateiformate

Einige Neuerungen gibt es auch im Bereich der Dateiformate. Gleich vorab: In Excel 2013 werden Arbeitsmappen weiterhin im *.xlsx*-Format gespeichert – und in der Regel werden Sie dies wohl auch so tun. Dennoch kommt es immer wieder vor, dass Sie Ihre Excel-Dateien in anderen Dateiformaten speichern und zur Verfügung stellen möchten. So zum Beispiel in einem Dateiformat früherer Excel-Versionen (*.xls*) oder als PDF- oder XML-Datei.

All das ist natürlich kein Problem und über die sogenannte Backstage-Ansicht (Registerkarte *DATEI*) realisierbar.

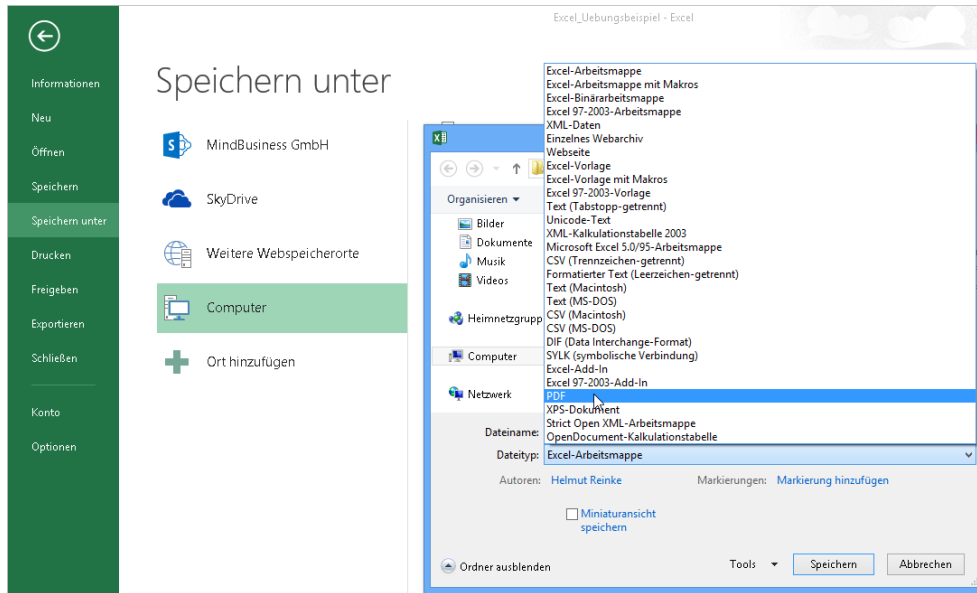


Abbildung 1.24: Unproblematisch: das Speichern von Arbeitsmappen in anderen Dateiformaten

Bitte beachten Sie: Haben Sie eine Excel-Datei in 2013 erstellt, kann es durchaus vorkommen, dass einzelne Inhalte der Arbeitsmappe nicht in dem von Ihnen gewählten neuen Dateiformat abbildbar sind und somit nicht gespeichert werden.

Achtung

Zurück in die Zukunft: 2013 in 2003

Möchten Sie eine Excel 2013-Arbeitsmappe als Excel 97-2003-Arbeitsmappe speichern, um sie beispielsweise einem Kunden zu schicken, der sie in diesem Format benötigt, können Sie die Datei entsprechend konvertieren.

Wir empfehlen Ihnen die folgende Vorgehensweise:

- ▶ Prüfen Sie zunächst die Kompatibilität
- ▶ Speichern Sie das Dokument dann als *.xls*-Datei

So gehen Sie dazu vor:

1. Die Arbeitsmappe, die Sie als *.xls*-Datei speichern möchten, ist geöffnet.
2. Klicken Sie auf *DATEI*. Sie befinden sich automatisch im Bereich *Informationen* der Backstage-Ansicht.
3. Wählen Sie im Dropdownmenü der Schaltfläche *Auf Probleme prüfen* den Eintrag *Kompatibilität prüfen*.

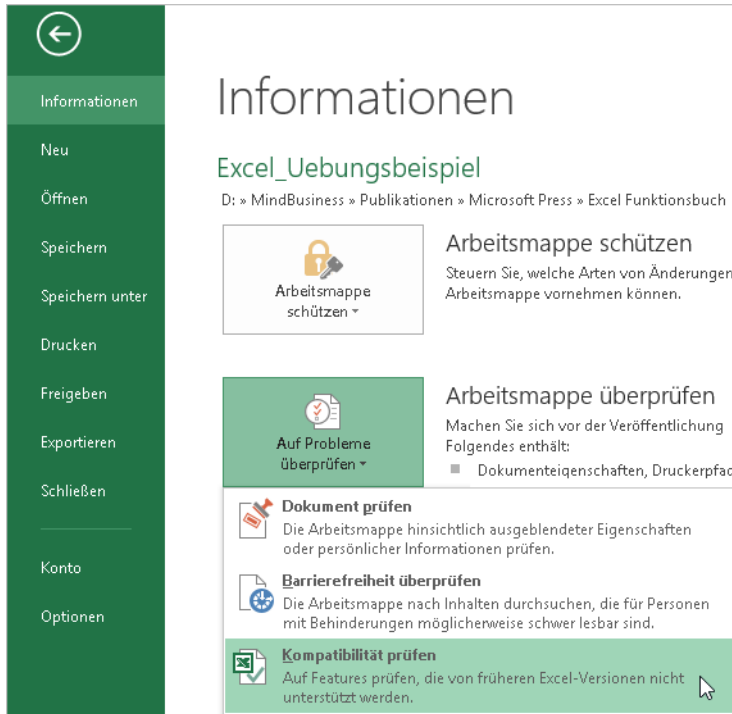


Abbildung 1.25: Gibt es Probleme oder nicht? Die Kompatibilitätsprüfung bringt Licht ins Dunkel.

4. Werden Komponenten in der Datei entdeckt, die nicht zu früheren Excel-Versionen kompatibel sind, werden diese über das Dialogfeld *Microsoft Excel - Kompatibilitätsprüfung* angezeigt.

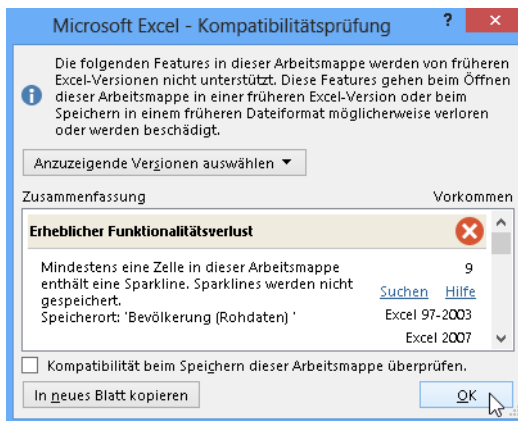


Abbildung 1.26: Fehlerquellen schnell und einfach beheben

5. Bereinigen Sie die dargestellten Probleme und prüfen Sie die Arbeitsmappe erneut. Werden Ihnen keinen Fehler mehr angezeigt, können Sie das Dokument konvertieren.

6. Klicken Sie auf *Speichern unter*, wählen Sie den gewünschten Speicherort und anschließend im Dialogfeld *Speichern unter* im Dropdownmenü der Liste *Dateityp* den Eintrag *Excel 97-2003-Arbeitsmappe* (Abbildung 1.24).

Die Arbeitsmappe kann nun vom Empfänger geöffnet werden.

Alternativ können Anwender der 97-2003er Excel-Versionen ein sogenanntes Kompatibilitätspack herunterladen. Mit dem Download werden alle notwendigen Updates installiert, sodass anschließend das neue Dateiformat in der alten Version geöffnet werden kann.

Hinweis

Den Download finden Sie unter <http://support.microsoft.com/kb/924074/de>.

Up to date: 2003 in 2013

Wenn Sie von der Excel-Version 97-2003 zu Excel 2013 wechseln, können Sie Ihre Excel-Dateien im alten *.xls*-Format ganz einfach in das neue Dateiformat *.xlsx* konvertieren.

Gehen Sie dazu wie folgt vor:

1. Öffnen Sie Ihre Excel 97-2003-Arbeitsmappe mit Excel 2013.
2. Klicken Sie auf *DATEI*. Sie befinden sich automatisch im Bereich *Informationen* der Backstage-Ansicht.
3. Klicken Sie auf die Schaltfläche *Konvertieren*. Excel 2013 hat automatisch erkannt, dass ein Arbeitsblatt einer älteren Version geöffnet wurde.

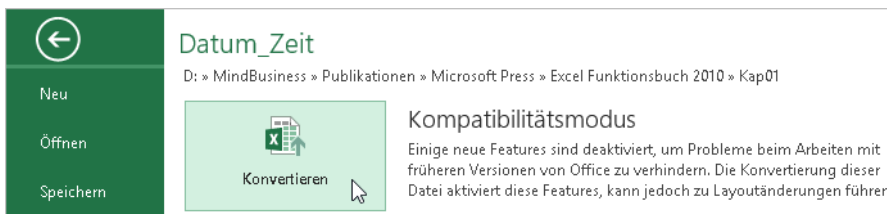


Abbildung 1.27: Aus alt mach neu: alte Arbeitsmappen in das neue Dateiformat konvertieren

4. Es öffnet sich das Dialogfeld *Speichern unter*. Als *Dateityp* ist bereits *Excel-Arbeitsmappe* (entspricht dem *.xlsx*-Format) eingestellt. Speichern Sie Ihr Dokument am gewünschten Speicherort.

»Alle wissen Bescheid statt ich weiß Bescheid« – Daten online zur Verfügung stellen

In einer zunehmend vernetzten Welt, und nicht nur unternehmens-, sondern weltweit verteilten Teams, wird es immer wichtiger, Informationen online zur Verfügung zu stellen. Nur so stehen die Daten 24 Stunden am Tag und 365 Tage im Jahr zur Verfügung.

Excel 2013 bietet zahlreiche Funktionalitäten zur Informationsverarbeitung und unterstützt Anwender bei der Bereitstellung und Verteilung von Wissen. Einen kurzen Überblick über die Möglichkeiten geben wir Ihnen nachfolgend.

Über die Backstage-Ansicht der Anwendung können Sie Dateien öffnen und speichern, aber auch freigeben und exportieren. So speichern Sie Arbeitsmappen ab jetzt auch bei Onlinediensten wie Office 365 oder SkyDrive und können so von überall und zu jeder Zeit auf Ihre Informationen zugreifen.

Tip Bei Onlinediensten veröffentlichte Dokumente werden nicht nur von Ihnen bearbeitet. Freigegebene Arbeitsblätter stehen, unabhängig davon, welches Gerät die Person nutzt oder wo sie sich befindet, immer in der neuesten Version zur Verfügung. Ebenso ist es möglich, mit mehreren Beteiligten in Echtzeit an einem Dokument zusammenzuarbeiten.

Wie Sie eine Excel-Arbeitsmappe in SkyDrive zur Verfügung stellen und anschließend einer Person freigeben, zeigen wir Ihnen nachfolgend.

Nachdem alle Daten Ihrer Excel-Arbeitsmappe bearbeitet und optimal visualisiert sind, wollen Sie die Welt – oder vielleicht im ersten Schritt auch erst mal nur Ihre Kollegen – an Ihren Erkenntnissen teilhaben lassen.

Achtung Um dieses Beispiel selber auszuprobieren, benötigen Sie ein Microsoft Online-Konto. Mit diesen können Sie Ihre Dateien online, beispielsweise in SkyDrive oder Office 365, speichern.

Gehen Sie wie folgt vor:

1. Ihre Arbeitsmappe, in unserem Beispiel die Datei *Excel_Uebungsbeispiel_leer.xlsx*, ist geöffnet. Klicken Sie auf die Registerkarte *DATEI*.
2. Öffnen Sie den Bereich *Speichern unter* und wählen Sie den Eintrag *SkyDrive* (oder einen anderen Onlinedienst).

Hinweis In unserem Beispiel bin ich bereits mit meinem User »su@mindbusiness.de« bei meinem SkyDrive-Konto angemeldet.

3. Klicken Sie auf die Schaltfläche *Durchsuchen*, um den passenden SkyDrive-Ordner für Ihre Datei anzugeben. Die Vorgehensweise ist dabei die gleiche wie in Ihrer lokalen Ordnerstruktur.

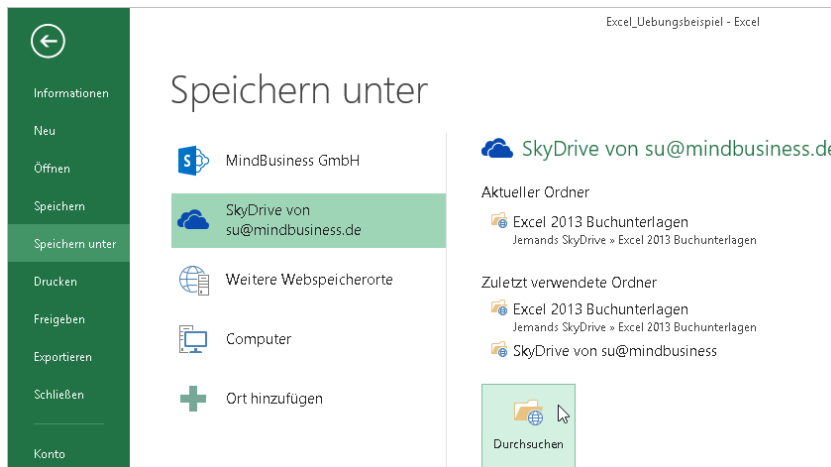


Abbildung 1.28: Der Trend: Daten online verwalten

4. Geben Sie alle wichtigen Informationen an und klicken Sie auf die Schaltfläche *Speichern*. Der Upload auf SkyDrive wird nun durchgeführt und die Arbeitsmappe liegt online bereit für die weitere Bearbeitung.

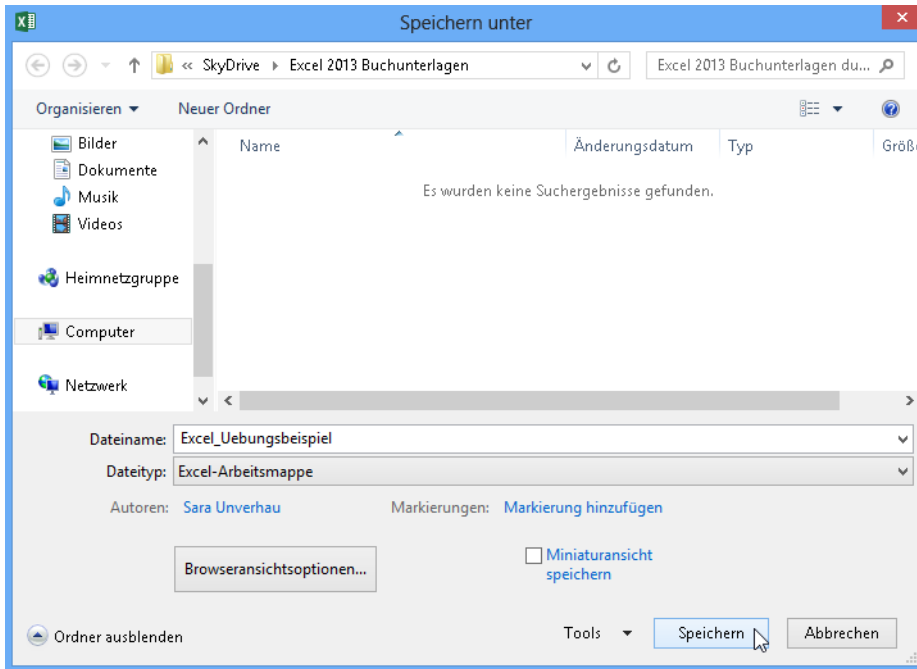


Abbildung 1.29: Speichern Sie Ihre Dateien wie gewohnt – nur eben online

5. Um das Dokument einem Kollegen freizugeben und zur Verfügung zu stellen, klicken Sie erneut auf *DATEI* und anschließend auf den Bereich *Freigeben*.
6. Wählen Sie den Eintrag *Person einladen*. Geben Sie jetzt die E-Mail-Adresse des Empfängers ein, ergänzen Sie einen passenden Text, legen Sie die gewünschten Berechtigungen fest und klicken Sie abschließend auf die Schaltfläche *Freigeben*.
7. Die eingeladene Person erhält nun eine E-Mail mit dem Link zum Excel-Arbeitsblatt und kann so Ihre Excel-Datei bearbeiten und wieder speichern.

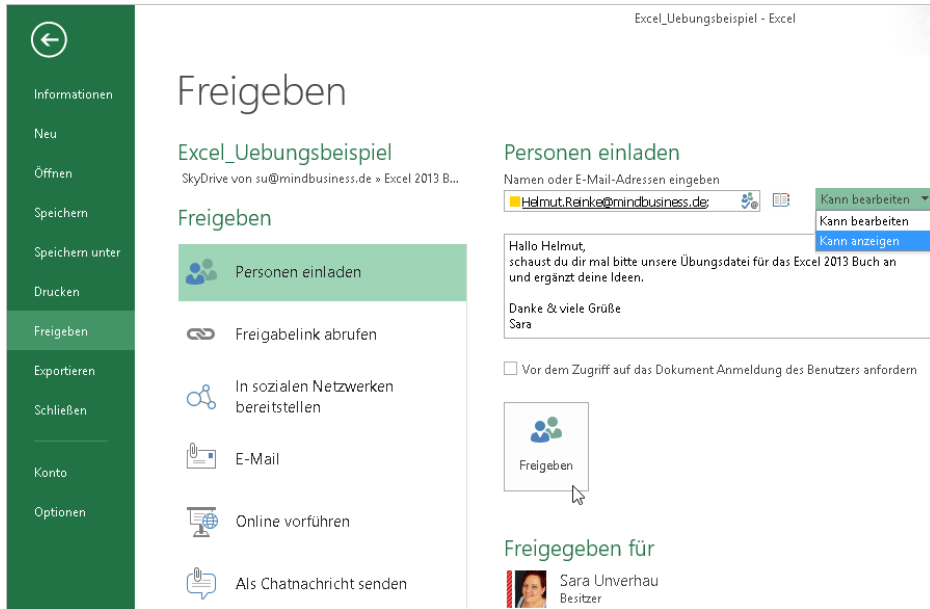


Abbildung 1.30: Ein Ende der E-Mail-Flut – Dokumente mit Kollegen online teilen

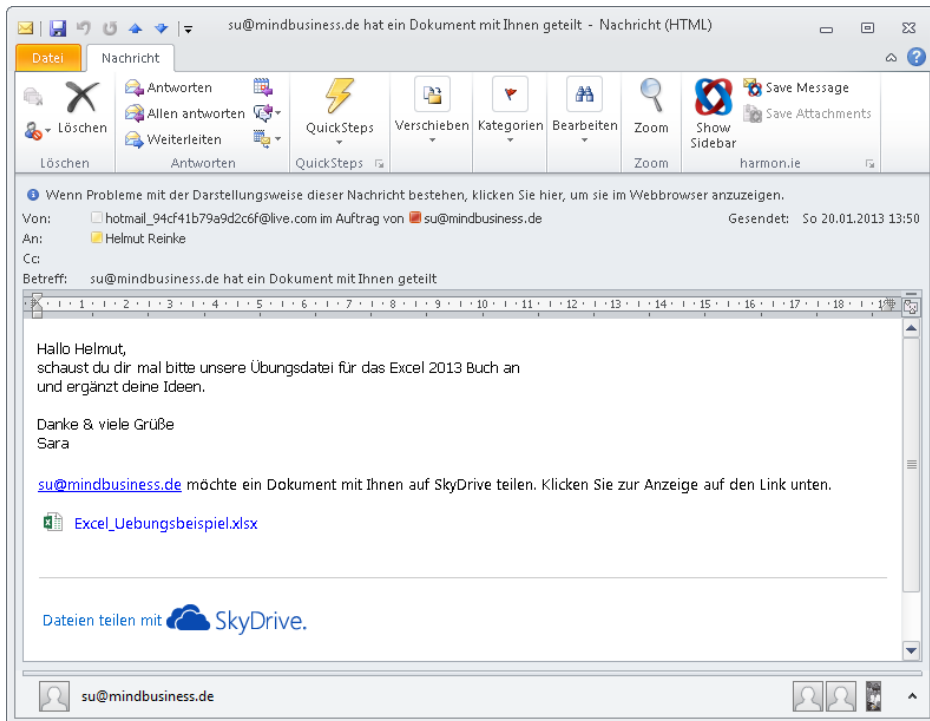


Abbildung 1.31: In wenigen Schritten geben Sie bestimmte Dokumente für Kollegen frei

Fazit

Sie haben nun einen Überblick über die neuen Basisfunktionen von Excel 2013 erhalten. Das Update auf Excel 2013 lohnt sich vor allem für Einsteiger, denen die Tabellenkalkulation bislang zu komplex war. Auch wenn Excel natürlich weiterhin den gewohnt Umfang an Funktionen und Tools anbietet, um fundierte Entscheidungen im Analysebereich zu ermöglichen, wurde in Excel 2013 doch erheblich an der Benutzerfreundlichkeit »geschraubt«. Einfache Aufgaben lassen sich dank der Vorlagen und Schnellhilfen auch ohne tiefer gehendes Know-how umsetzen.

Mit der Möglichkeit, Informationen über Onlinedienste wie SkyDrive und Office 365 zu teilen, wurde ein weiterer Schritt in Richtung Zukunft gemacht.

Freuen Sie sich nun auf die nachfolgenden Abschnitte zu den Themen PowerPivot und Power View.

Aus Daten werden Erkenntnisse – Arbeiten mit der PivotTable

Aus Daten Erkenntnisse zu gewinnen ist – gerade in Anbetracht der Informationsflut, die uns tagtäglich ereilt – nicht immer ganz einfach. Oder zumindest leichter gesagt als getan.

Es gibt drei wichtige Schritte der Informationsaufbereitung:

- ▶ **Daten sammeln und verwalten** Bedeutet, dass Daten aus allen relevanten Quellen gespeichert und in entsprechender Qualität zur Verfügung gestellt werden müssen
- ▶ **Daten aufbereiten und anreichern** Heißt, Daten verständlich, eindeutig und schnell konsumierbar anzubieten und Datenschutz zu gewährleisten
- ▶ **Einblicke und Analysen** Meint das Erstellen von Dashboards und Berichten, die die Benutzer in ihrer jeweiligen Arbeitsweise unterstützen – vom Standard-Bericht bis zum Self-Service BI (Business Intelligence)

Nun gibt es unzählige Systeme und unterschiedliche Anforderungen, die unter einen Hut gebracht werden müssen. Schon wenn man die Herausforderungen von Anwendern und IT-Spezialisten vergleicht, wird klar, dass die Beteiligten innerhalb dieses Informationsprozesses unterschiedliche Ziele haben.

- ▶ Anwender wollen agil sein und ...
 - ▶ ... müssen Daten suchen, analysieren und aufbereiten,
 - ▶ ... wollen Ergebnisse teilen und gemeinsam daran arbeiten,
 - ▶ ... benötigen jederzeitigen Zugriff auf Ihre Daten – von überall
- ▶ Im Fokus der IT-Spezialisten hingegen steht die Qualität. Sie sind interessiert an ...
 - ▶ ... der Konsistenz und Integrität der Daten,
 - ▶ ... der Sicherheit und Verfügbarkeit aller Informationen,
 - ▶ ... der Verteilung und Skalierbarkeit

Es muss also die richtige Balance zwischen Flexibilität und Qualität gefunden werden.

Mit dem Thema Self-BI in Excel wird den Anwendern ein Werkzeug an die Hand gegeben, mit dem sie qualitativ hochwertige Auswertungen erstellen können und ihrer Anforderung nach Agilität gerecht werden.

Excel 2013 unterstützt seine Anwender in vielerlei Hinsicht bei der Analyse von Daten. Im Bereich PivotTable jedoch gibt es drei neue Funktionen, mit deren Hilfe Sie Auswertungen praxisorientierter gestalten und noch übersichtlicher und einfacher realisieren können.

Neue Funktionen in der Kurzübersicht

Hierzu zählen unter anderem die Funktionen:

- ▶ Tabellen verbinden
- ▶ Timeline
- ▶ Empfohlene PivotTable

Tabellen verbinden

Mit dieser Funktion können zwei oder mehrere Datenbereiche in eine PivotTable integriert werden. Bis Excel 2010 war diese Funktion nur in der Erweiterung »PowerPivot« verfügbar. Haben Sie mehrere Tabellen in Ihr Datenmodell integriert, lassen sich die Felder aller Tabellen in der Pivot-Auswertung verwenden. Die Spaltenüberschriften oder Felder müssen dabei nicht identisch sein, da zwischen den einzelnen Feldern und Tabellen Beziehungen hergestellt werden können.

Das heißt: Ab Excel 2013 können Sie, ohne alle wichtigen Daten zuvor in einer einzigen Datentabelle zusammenfassen zu müssen, Daten noch effizienter auswerten und haben dabei Zugriff auf alle leistungsfähigen Analysefeatures.

Timeline

Eine weitere spannende Funktion im Bereich PivotTable ist die Timeline-Funktion, mit deren Hilfe Sie einen Filter über beliebige Zeitschienen setzen können. Das heißt, mit der Timeline-Funktion haben Sie ein Werkzeug, das es ermöglicht, einfach und schnell Daten auf Basis eines vorgegebenen Zeitfensters zu filtern.

Die Timeline bringt ein eigenes Register mit und kann ähnlich den Datenschnitten formatiert werden.

Hinweis Weitere Informationen zum Thema Datenschnitt lesen Sie im Abschnitt »Ganz schön »schnittig« – Filtern von Tabellendaten mit Datenschnitten« ab Seite 43.

Ein weiterer Vorteil: Anhand einer Zeitachse lassen sich PivotTable- oder PivotChart-Daten über unterschiedliche Zeiträume einfacher vergleichen.

Empfohlene PivotTable

Das Erstellen einer PivotTable aus einer Datentabelle ist grundsätzlich recht einfach. Oft stellen sich jedoch die Fragen: »Wie werte ich die Daten am besten aus? Wo positioniere ich meine Felder in der Pivot-Auswertung, um mir die Daten sinnvoll darstellen zu lassen?« Das Auswählen der richtigen Felder, um Daten in einem PivotTable-Bericht zusammenzufassen, kann eine anspruchsvolle Aufgabe sein.

Excel 2013 unterstützt Sie hierbei zukünftig, wenn Sie das möchten. Wenn Sie eine PivotTable erstellen, empfiehlt Excel mehrere Möglichkeiten, wie die Daten zusammengefasst werden können. Außerdem erhalten Sie eine kurze Vorschau des Feldlayouts, sodass Sie ganz einfach das für Ihre Anforderungen passende auswählen können.

Szenario und Beispiel

Alle drei soeben skizzierten Funktionen möchten wir Ihnen nachfolgend an einem Beispiel näher bringen. Stellen Sie sich folgendes Szenario vor: Sie haben drei Tabellen *Mitarbeiter*, *Projekte* und *Projektstunden*, die in unterschiedlichen Tabellenblättern verwaltet werden. Sie wollen darstellen, welcher Mitarbeiter wie viele Stunden an welchem Projekt gearbeitet hat:

- ▶ **Mitarbeiter** Beschreibt, wie die Mitarbeiter heißen, ob sie intern oder extern arbeiten, welches Kürzel sie haben und welche Mitarbeiternummer ihnen zugeordnet ist
- ▶ **Projekte** Ist eine Auflistung aller laufenden Projekte mit Projektname, Projektnummer, Zeitraum, Planstunden und Plankosten
- ▶ **Projektstunden** Beinhaltet alle tatsächlich geleisteten Stunden nach Projekt und Mitarbeiter und zeigt außerdem die angefallenen Istkosten

Um diese Daten unterschiedlicher Tabellen miteinander zu verbinden und anschließend auszuwerten, musste man früher die Tabellen recht komplex, z.B. per Funktion SVERWEIS miteinander verknüpfen. Das geht jetzt erheblich einfacher.

PowerPivot-Add-In aktivieren

Um den vollen Funktionsumfang von PowerPivot in Excel 2013 auszuschöpfen und das nachfolgende Beispiel nachzuvollziehen, aktivieren Sie das Add-In für PowerPivot. Gehen Sie dazu wie folgt vor:

1. Öffnen Sie Microsoft Excel 2013. Klicken Sie auf *DATEI/Optionen*.
2. Klicken Sie im Dialogfeld *Excel-Optionen* auf die Kategorie *Add-Ins*.
3. Wählen Sie im Dropdownmenü von *Verwalten* den Eintrag *COM-Add-Ins* und klicken Sie auf die Schaltfläche *Gehe zu*. ▶



Abbildung 1.32: Über die Backstage-Ansicht erreichen Sie die Excel-Optionen

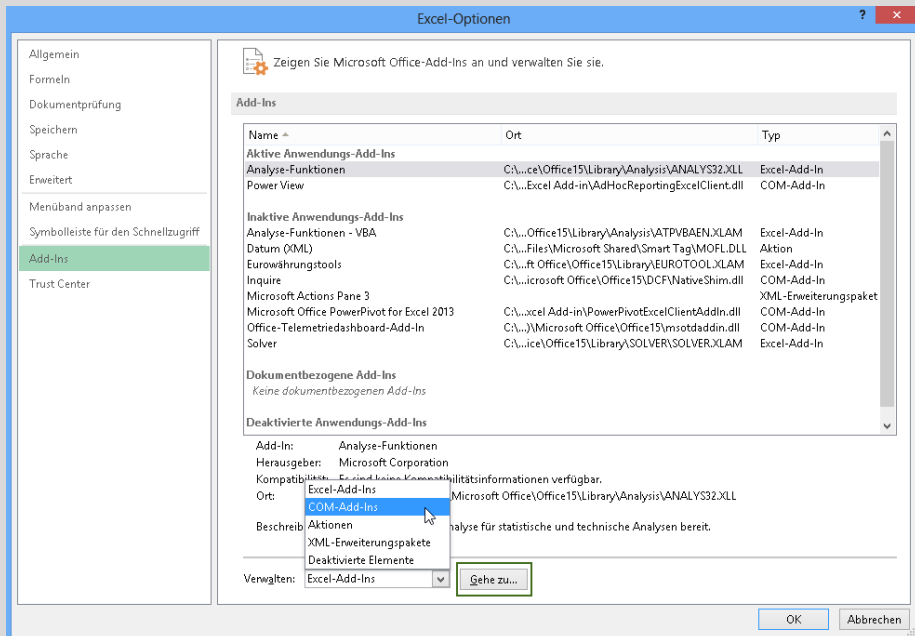


Abbildung 1.33: Die Verwaltung der Add-Ins können Sie individuell bearbeiten ►

4. Es öffnet sich das Dialogfeld *COM-Add-Ins*. Aktivieren Sie unter *Verfügbare Add-Ins* die Kontrollkästchen *Microsoft Office PowerPivot for Excel 2013* und *Power View*.

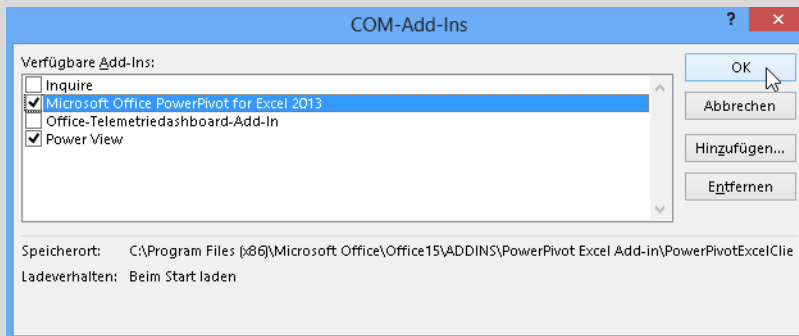


Abbildung 1.34: Holen Sie sich die volle Power in Excel

5. Nach dem Bestätigen mit *OK* wird das PowerPivot-Add-In nun als neue Registerkarte mit der Bezeichnung *POWERPIVOT* neben der Registerkarte *ANSICHT* eingeblendet.



Abbildung 1.35: Nutzen Sie den vollen Funktionsumfang über die neue Registerkarte

Wer mit wem – Tabellen verknüpfen, Beziehungen erstellen

Die folgenden Praxisbeispiele in Excel 2013 finden Sie im Ordner `\Ms5-235\Kap01` in der Arbeitsmappe *PowerPivot_Beispiel_bearbeitet.xlsx*. Möchten Sie die Übung Schritt für Schritt nachvollziehen und mitmachen, öffnen Sie die Datei *PowerPivot_Beispiel_leer.xlsx*.



So gehen Sie vor, um die Tabellen und deren Inhalte miteinander zu verknüpfen:

1. Öffnen Sie die Excel Datei *PowerPivot_Beispiel_leer.xlsx*.
2. Sie sehen die drei oben beschriebenen Tabellenblätter *Mitarbeiter*, *Projekte* und *Projektstunden*. Öffnen Sie das erste Tabellenblatt *Mitarbeiter* und klicken Sie in die Tabelle.
3. Öffnen Sie die Registerkarte *POWERPIVOT* und klicken Sie auf die Schaltfläche *Zu Datenmodell hinzufügen*.

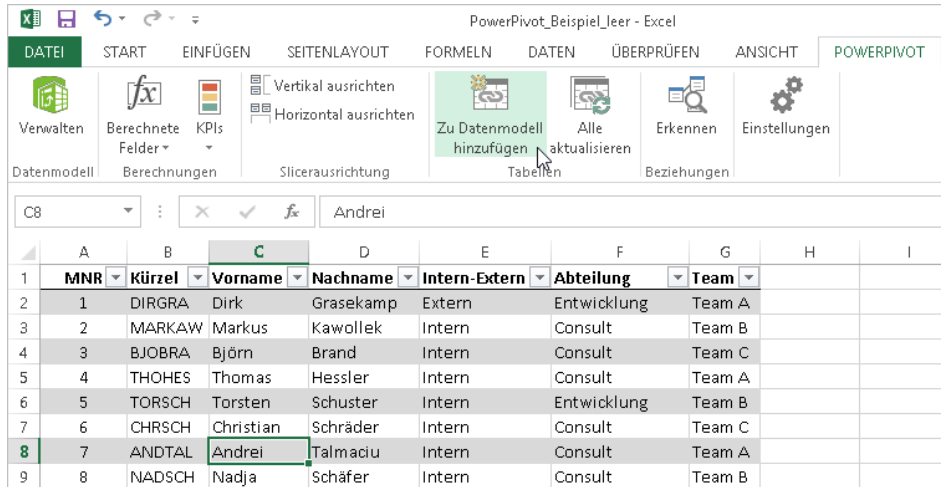


Abbildung 1.36: Bauen Sie das Datenmodell auf, um umfangreiche Auswertungen zu erstellen

- Es öffnet sich das Dialogfeld *PowerPivot für Excel*, über das die Datenmodelle verwaltet werden. Sie sehen hier die Darstellung der Mitarbeiterdaten im Tabellenblatt *xls_Mitarbeiter*. Schließen Sie das Dialogfeld über das Symbol *Schließen* rechts oben in der Ecke des Dialogfelds.

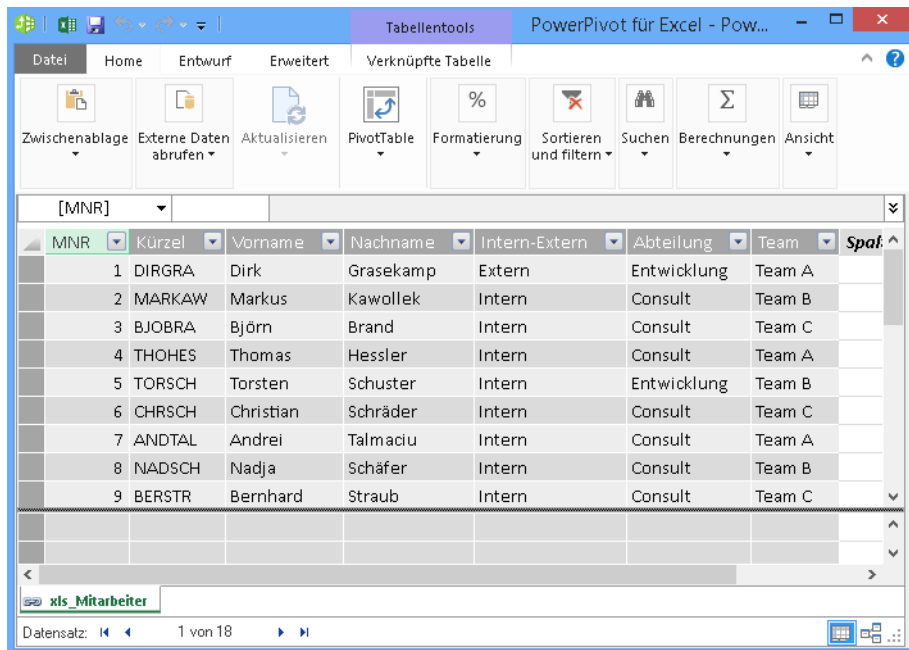


Abbildung 1.37: Die Daten können auch im Datenmodell bearbeitet werden

5. Wiederholen Sie den Vorgang für die Tabellen *Projekte* und *Projektstunden*, sodass anschließend alle drei Tabellen im Datenmodell verfügbar sind. Wie in Abbildung 1.38 dargestellt, werden die Tabellen als Datenmodelle *xls_Mitarbeiter*, *xls_Projekte* und *xls_Projektstunden* beschrieben.

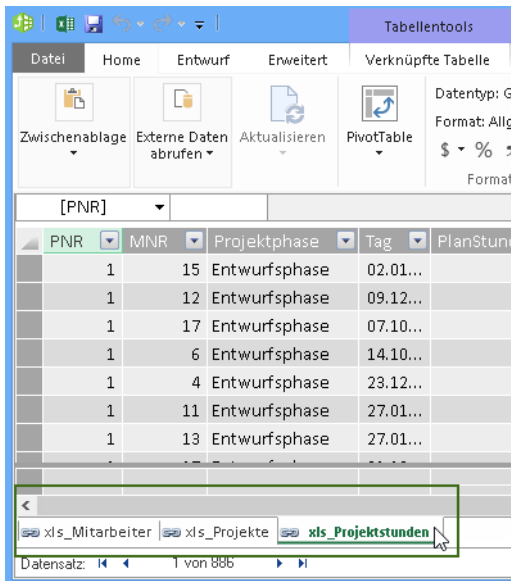


Abbildung 1.38: Der erste Schritt ist getan – das Datenmodell wurde definiert

Lassen Sie das Add-In *PowerPivot für Excel* geöffnet. Wir werden gleich damit weiter arbeiten.

Jetzt müssen die Beziehungen zwischen den Tabellen hergestellt werden, damit PowerPivot weiß, welche Bereiche womit verknüpft werden müssen. Um das Beispiel einfach und verständlich zu halten, gibt es in den drei Tabellen eindeutige Schlüssel, die die Auswertung erleichtern. Dazu zählen:

- ▶ Im Tabellenblatt *Mitarbeiter* die Mitarbeiternummer *MNR*
- ▶ Im Tabellenblatt *Projekte* die Projektnummer *PNR*
- ▶ Im Tabellenblatt *Projektstunden* die Mitarbeiternummer *MNR* und die Projektnummer *PNR*

Es gibt im PowerPivot-Add-In die Möglichkeit, in eine Diagrammansicht zu wechseln, die die aktuellen Beziehungen zwischen den Datentabellen darstellt.

- ▶ Klicken Sie dazu in der Gruppe *Ansicht* auf *Diagrammansicht*
- ▶ Die Darstellung der tabellarischen Modelldaten ändert sich in ein Diagramm

Wie Sie in Abbildung 1.39 erkennen, bestehen momentan noch keine Beziehungen zwischen den Tabellendaten. Wenn wir am Ende des Beispiels noch einmal die Diagrammansicht aufrufen, sehen Sie die Verknüpfungen zwischen den Daten.

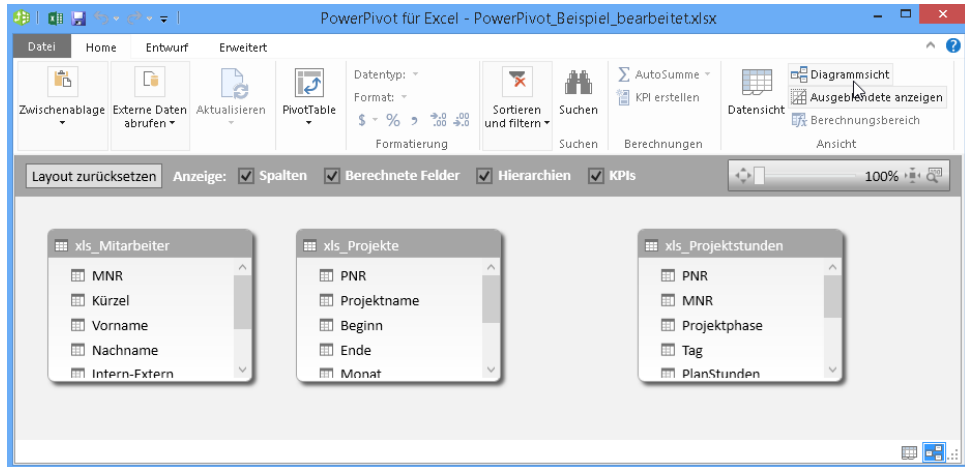


Abbildung 1.39: Visuelle Darstellung der Beziehungen innerhalb des Datenmodells

1. Das Add-In *PowerPivot für Excel* ist noch geöffnet und Sie sehen die verschiedenen Datenmodelle. Öffnen Sie das Dropdownmenü der Schaltfläche *PivotTable* und wählen Sie den gleichnamigen Eintrag.

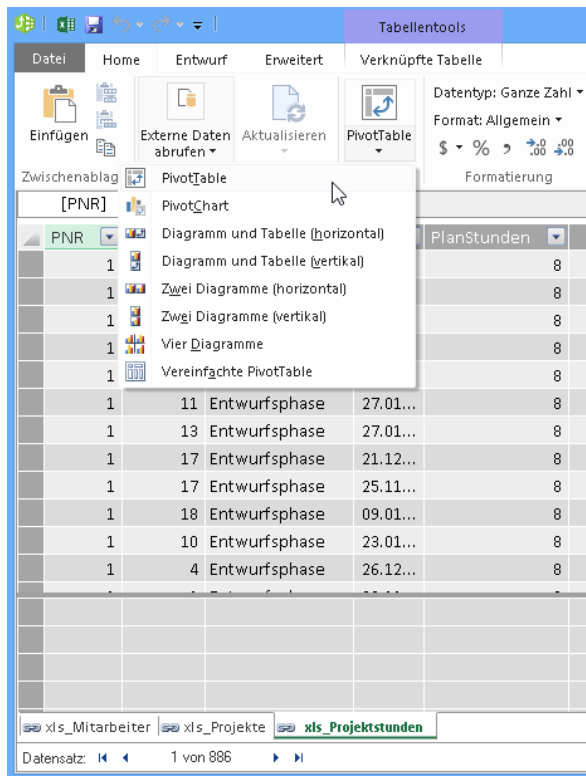


Abbildung 1.40: Es folgt Schritt 2 – die Erstellung der PivotTable

- Bestätigen Sie das sich nun geöffnete Dialogfeld *Pivot einfügen* mit einem Klick auf *OK*, um ein neues Arbeitsblatt anzulegen.



Abbildung 1.41: Legen Sie fest, ob Sie die PivotTable in ein neues oder das gleiche Arbeitsblatt einfügen möchten

- Die PivotTable wurde auf einem neuen Tabellenblatt eingefügt. Die PivotTable-Felder werden ebenfalls direkt eingeblendet.

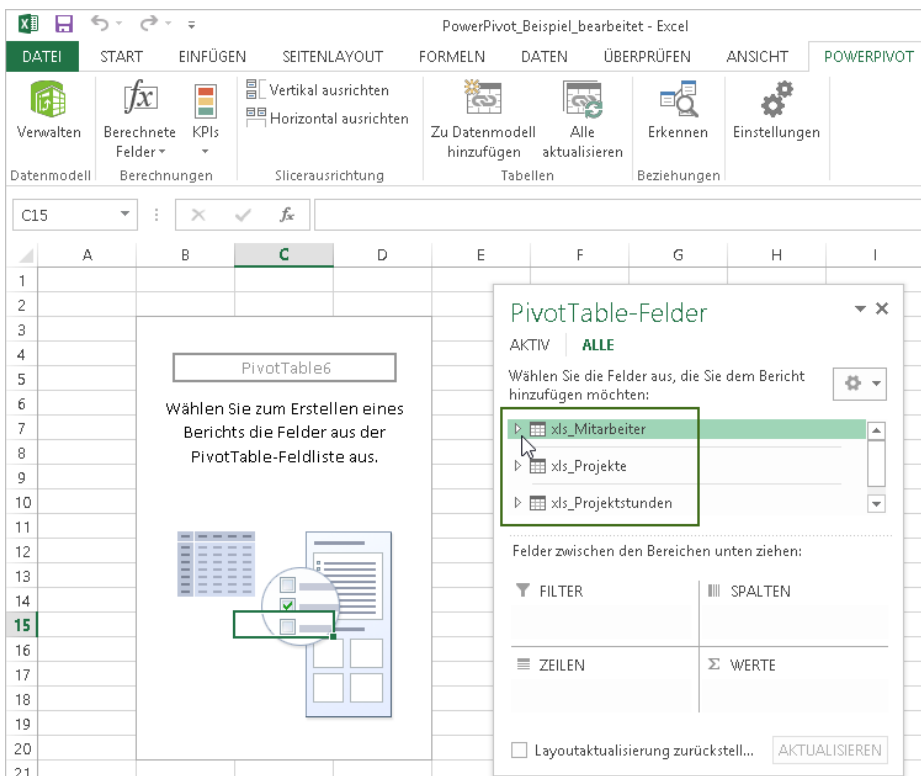


Abbildung 1.42: Eine neue Dimension – die Verbindung unterschiedlicher Tabellen in einer PivotTable

Wie die Abbildung 1.42 zeigt, sind alle drei zuvor eingebundenen Tabellen im Datenmodell vorhanden. Auf dieser Basis können nun die Beziehungen erstellt werden.

Hinweis

Klicken Sie innerhalb des Aufgabenbereichs *PivotTable-Felder* auf den kleinen Pfeil vor einer der Tabellen *xls_Mitarbeiter*, *xls_Projekt* oder *xls_Projektstunden*, können Sie sich alle Felder der Tabellen anzeigen lassen.

- Öffnen Sie die Liste unter *xls_Mitarbeiter* durch Klicken auf den Pfeil und aktivieren Sie das Kontrollkästchen *Nachname*. Das Feld wird automatisch in der Pivot-Feldliste im Bereich *ZEILEN* eingefügt.

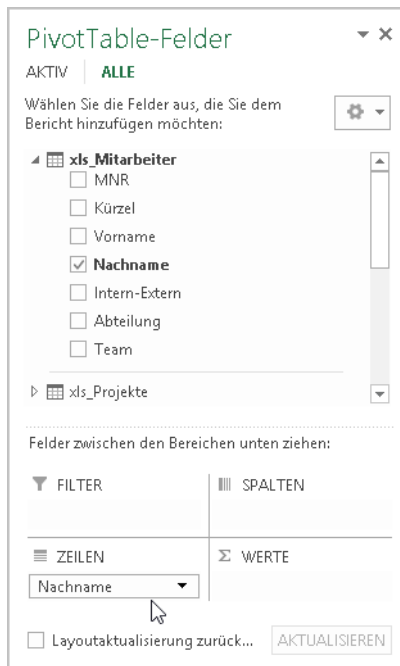


Abbildung 1.43: Starten Sie die Zusammenstellung der Pivot-Feldliste

- Öffnen Sie die Datenquelle *xls_Projektstunden* und aktivieren Sie das Kontrollkästchen des Felds *Stunden*. Es wird in der Pivot-Matrix unter Σ *WERTE* eingefügt.

Sie haben nun zwei Tabellen – Mitarbeiter und Projektstunden – miteinander verbunden. Die Abbildung 1.44 zeigt, dass PowerPivot sofort einen Hinweis einblendet, dass das Erstellen von Beziehungen erforderlich ist.

- Klicken Sie auf die Schaltfläche *Erstellen* neben dem Hinweis.

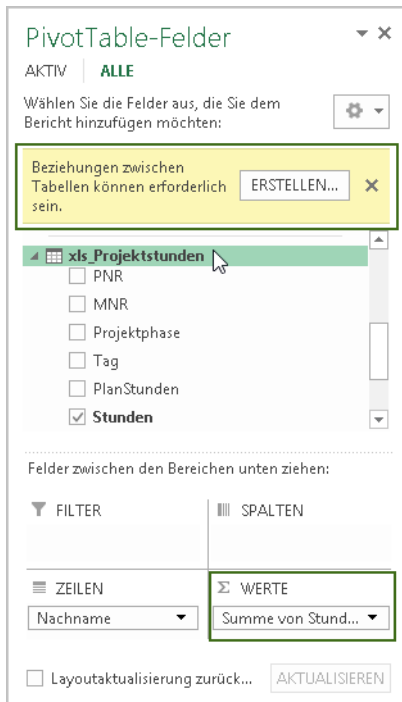


Abbildung 1.44: Die Auswertung nimmt Formen an

Das Ziel des nächsten Arbeitsschritts:

Hinweis

- ▶ Das Feld *MNR* (Mitarbeiternummer) der Tabelle »Projektstunden« muss mit dem Feld *MNR* der Tabelle »Mitarbeiter« verknüpft werden
- ▶ Das Feld *PNR* (Produktnummer) der Tabelle »Projektstunden« muss mit dem Feld *PNR* der Tabelle »Produkte« verknüpft werden

Erst wenn die Beziehungen zwischen den Tabellen und deren Felder erstellt sind, ist es möglich, übergreifende Auswertungen und Pivot-Analysen zu erstellen. In unserem Beispiel haben wir der Einfachheit halber die Felder *Mitarbeiternummer* und *Projektnummer* in allen Tabellen gleich genannt. Grundsätzlich könnten die Felder natürlich aber auch anders, z.B. »MNR« und »Mitarbeiter-Nr.« heißen. Durch das Erstellen der Beziehungen spielt der Name der Felder keine Rolle mehr.

7. Nehmen Sie die Einstellungen im Dialogfeld *Beziehung erstellen* wie nachfolgend dargestellt vor. In unserem Beispiel erstellen wir zunächst die Beziehung zwischen den Feldern *MNR* (Mitarbeiternummer) der Tabellen *Projektstunden* und *Mitarbeiter*. Öffnen Sie die Listenfelder, um die jeweiligen Einträge entsprechend der Abbildung 1.45 auszuwählen, und bestätigen Sie mit *OK*.

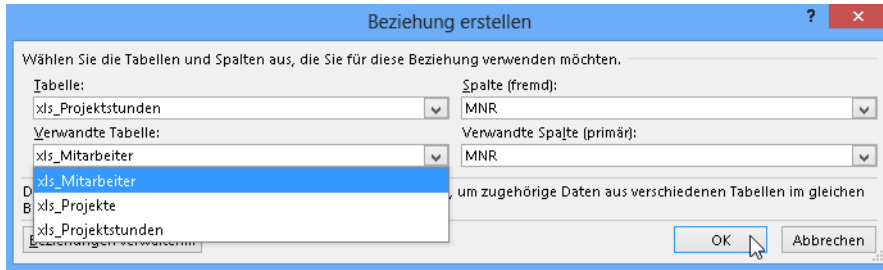


Abbildung 1.45: Achten Sie darauf, die Beziehungen zwischen den Tabellen richtig aufzubauen

8. Sie sehen nun wieder Ihre PivotTable sowie die Übersicht der PivotTable-Felder. Aktivieren Sie nun das Kontrollkästchen *Projektname* im Bereich *xls_Projekte*. Das Feld ordnet sich in der Pivot-Matrix automatisch unter *ZEILEN* an. Es erscheint erneut der Hinweis, dass das Erstellen von Beziehungen notwendig ist.

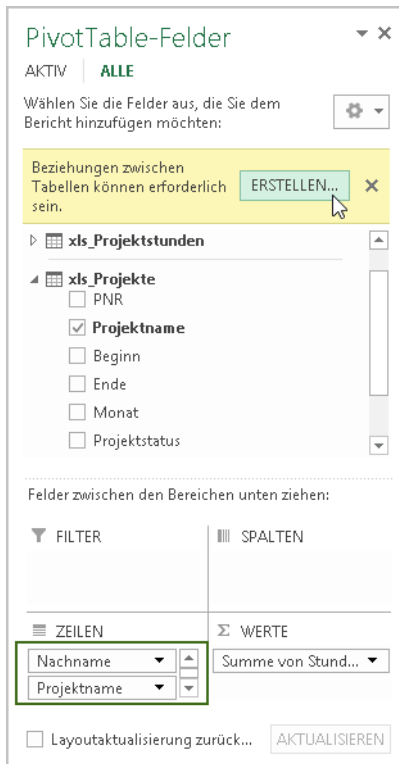


Abbildung 1.46: Je nachdem, welche Daten Sie aus welchen Tabellen einbinden, müssen mehrere Beziehungen aufgebaut werden

9. Klicken Sie auf *Erstellen* und wählen Sie die Beziehungen wie in nachfolgender Abbildung 1.47 dargestellt. Es wird jetzt die Beziehung zwischen den Feldern *PNR* (Projektnummer) der Tabellen *Projektstunden* und *Projekte* hergestellt.

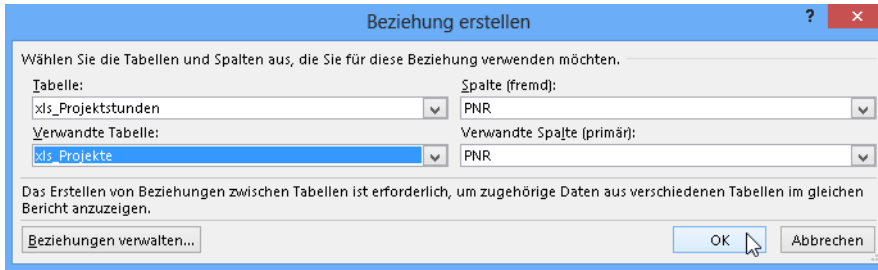


Abbildung 1.47: Nun ist es geschafft: Die Tabellen wurden miteinander verknüpft

10. Bestätigen Sie Ihre Angaben durch Klicken auf *OK*.

Sie haben jetzt alle Voraussetzungen für die Analyse Ihrer Tabellen geschaffen und können diese nun darstellen. Um zu zeigen, dass das Erstellen der Beziehungen zwischen den Tabellen erfolgreich war, bauen wir nun folgende Auswertung auf:

- ▶ Welcher Mitarbeiter war
- ▶ wie viele Stunden
- ▶ an welchem Projekt beschäftigt?

Damit sind alle drei Tabellen in die zu erstellende Analyse involviert. Gehen Sie so vor:

1. Klicken Sie in Ihre PivotTable. Ziehen Sie das Feld *Projektname* in den Bereich *SPALTEN*. Die Felder müssen nun so in den Bereichen angeordnet sein (Abbildung 1.48):
 - ▶ Aus der »xls_Mitarbeiter« das Feld *Nachname* unter *ZEILEN*
 - ▶ Aus der »xls_Projekte« das Feld *Projektname* unter *SPALTEN*
 - ▶ Aus der »xls_Projektstunden« das Feld *Stunden* unter Σ *WERTE*

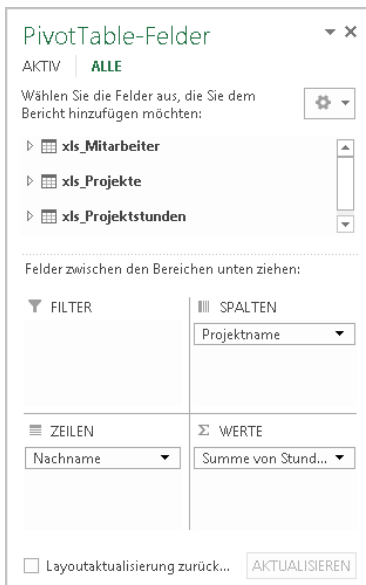


Abbildung 1.48: Das Finale – fügen Sie die Felder unterschiedlicher Tabellen in der PivotTable zusammen

2. Die Darstellung der Auswertung sehen Sie auszugsweise in Abbildung 1.49.

Summe von Stunden	Spaltenbes	Cloud4Gantt	Cloud4Plan	Cloud4SWOT	OneBoard	OneFinance	OneMind
Brand			31	6	48	50	27
Diestelhorst	30		91	9	69	100	53
Grasekamp	13		89	34	82	112	47
Herzog	14		39	16	33	38	29
Hessler	18		54	3	26	35	28
Jeschke	8		50	1	50	46	30
Kauffmann	7		64		40	33	32
Kawollek	3		46	15	25	56	24
Müller	3		30	20	26	56	25
Reinke	7		51	11	31	52	19
Schäfer			9	13	25	37	10
Schräder	6		38	18	32	21	21
Schuster			38	11	24	33	35
Schwarz	13		40	13	28	55	45
Straub	17		42	40	52	37	22
Talmaci	9		31	2	32	62	11
Gesamtergebnis	148	743	212	623	823	458	

Abbildung 1.49: Ihre Auswertung – tabellenübergreifend – auf einen Blick

Tipp

Sie wollen sich die Beziehungen zwischen den Tabellen in Diagrammform anzeigen lassen? Vor Beginn der Übung waren hier noch keine Beziehungen erkennbar. Rufen Sie die Diagrammansicht in PowerPivot für Excel erneut auf, sehen Sie Ihre Einstellungen visualisiert (Abbildung 1.50).

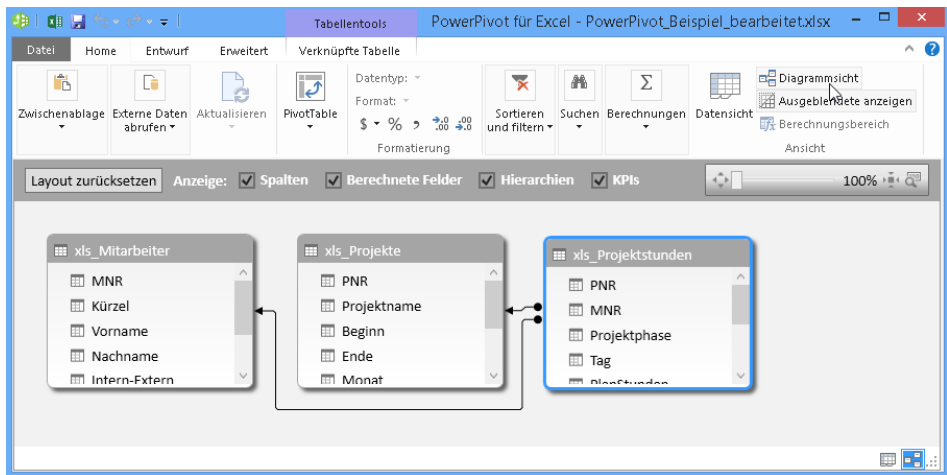


Abbildung 1.50: Visuelle Darstellung von Beziehungen innerhalb des Datenmodells

Auf den Punkt – Arbeiten mit der Timeline

Im nächsten Schritt möchten wir Ihnen das Arbeiten mit der Timeline an einem kurzen einfachen Beispiel verdeutlichen.

Das folgende Praxisbeispiel in Excel 2013 finden Sie im Ordner `\\Ms5-235\Kap01` in der Arbeitsmappe `PowerPivot_Beispiel_bearbeitet.xlsx`. Möchten Sie die Übung Schritt für Schritt nachvollziehen, öffnen Sie die Datei `PowerPivot_Beispiel_Leer.xlsx`.



Sie haben die Tabelle *Projektstunden* und möchten sehen, wie viele Stunden in welchem Zeitraum für welches Projekt aufgewendet wurden. Um die einzelnen Zeiträume zu analysieren, verwenden Sie die Timeline.

Informationen zum Thema finden Sie im Abschnitt »Timeline« ab Seite 52.

Hinweis

So gehen Sie vor:

1. Öffnen Sie die Excel-Datei und klicken Sie in die Tabelle des Arbeitsblatts *Projektstunden*.
2. Klicken Sie auf *EINFÜGEN/PivotTable* (Abbildung 1.51). Bestätigen Sie die Voreinstellungen (*Tabelle/Bereich: xls_Projektstunden* und *Neues Arbeitsblatt*) des Dialogfelds *PivotTable erstellen* mit *OK*.

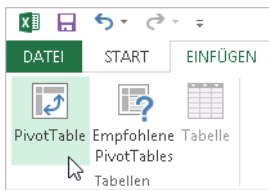


Abbildung 1.51: Erstellen Sie eine weitere PivotTable

3. Aktivieren Sie in der PivotTable-Feldliste die Kontrollkästchen der Felder *Projektphase* und *Stunden* (Abbildung 1.52).

	A	B	C	D	E	F
1						
2						
3	Zeilenbeschriftungen	Summe von Stunden				
4	Design	697				
5	Entwicklung	711				
6	Entwurfsphase	716				
7	Implementierung	1087				
8	Nacharbeit	135				
9	Testphase	522				
10	Gesamtergebnis	3868				
11						
12						
13						
14						
15						
16						
17						
18						
19						

PivotTable-Felder	
Wählen Sie die Felder aus, die Sie dem Bericht hinzufügen möchten:	
<input checked="" type="checkbox"/> Projektphase	
<input type="checkbox"/> Tag	
<input type="checkbox"/> PlanStunden	
<input checked="" type="checkbox"/> Stunden	
Felder zwischen den Bereichen unten ziehen:	
FILTER	SPALTEN
ZEILEN	WERTE
Projektphase	Summe von Stu...
<input type="checkbox"/> Layoutaktualisierung zurück... <input type="button" value="AKTUALISIEREN"/>	

Abbildung 1.52: Geben Sie die Felder an, die Sie auswerten möchten

4. Klicken Sie auf eine beliebige Zelle in der PivotTable und wählen Sie unter *EINFÜGEN* in der Gruppe *Filter* den Eintrag *Zeitachse*.

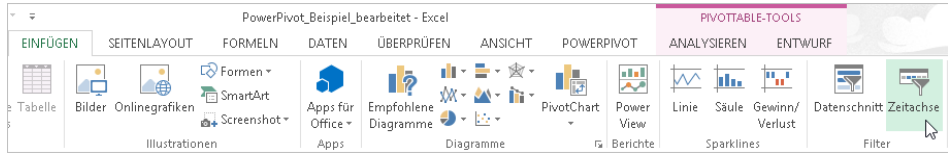


Abbildung 1.53: Mit der Zeitachse können Sie schnell und einfach zeitliche Filter setzen

5. Es öffnet sich das Dialogfeld *Zeitachse einfügen*, in dem alle zur Verfügung stehenden Zeitachsen aufgeführt werden. In unserem Beispiel gibt es nur eine mit der Bezeichnung *Tag*. Falls es mehrere gibt, treffen Sie die gewünschte Auswahl durch Aktivieren der Kontrollkästchen.

Aktivieren Sie in diesem Fall das Kontrollkästchen des Eintrags *Tag* und bestätigen Sie mit *OK* (Abbildung 1.54).

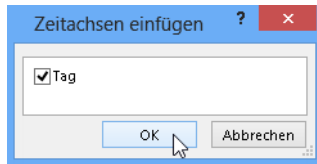


Abbildung 1.54: Legen Sie fest, wonach Sie filtern möchten

6. Die Timeline öffnet sich – und mit ihr eine neue Registerkarte *ZEITACHSENTOOLS*, über die Sie die Timeline noch beliebig anpassen können.

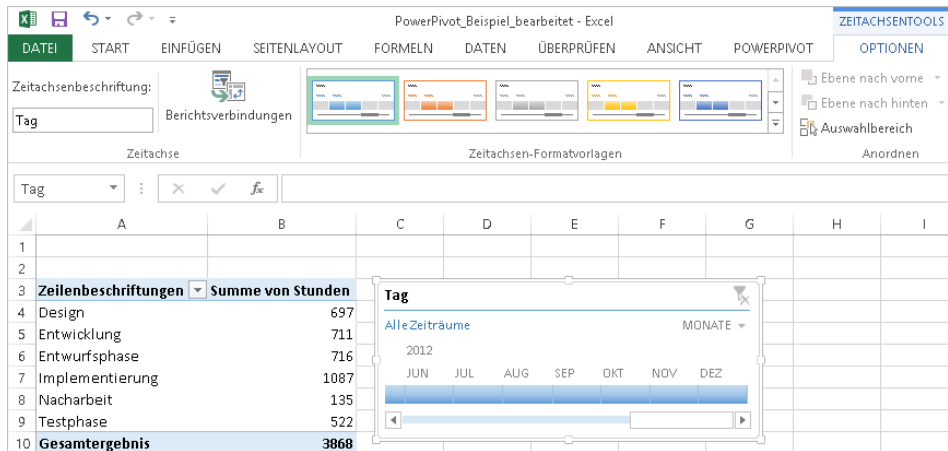


Abbildung 1.55: Die Timeline ist dank des Schiebereglers einfach zu bedienen

- Wenn Sie jetzt beispielsweise wissen möchten, wie viele Stunden für die einzelnen Projekte im ersten Halbjahr 2012 aufgebracht wurden, verschieben Sie ganz einfach den Anfangs- und Endpunkt der Zeitachse über den Schieberegler (Abbildung 1.56) mit der Maus auf den entsprechenden Monat.

Bitte beachten Sie: Die Bildlaufleiste am unteren Rand des Zeitachsentools dient nur dazu, einen gewünschten Zeitraum ins Bild zu ziehen. Sie bewirkt noch keine Filterung auf der Zeitachse. Diese erreichen Sie erst durch die Markierung des gewünschten Zeitraums auf der Zeitachse mit Hilfe der Maus.

Hinweis

- Die Daten in der PivotTable werden auf den von Ihnen angegebenen Zeitraum angepasst – das heißt: Es werden nur die Stunden im ersten Halbjahr 2012 angezeigt (Abbildung 1.56).

3	Zeilenbeschriftungen	Summe von Stunden	
4	Design	201	
5	Entwicklung	205	
6	Entwurfsphase	185	
7	Implementierung	250	
8	Nacharbeit	34	
9	Testphase	147	
10	Gesamtergebnis	1022	

Abbildung 1.56: Filtern von PivotTable-Informationen über eine Timeline: ab Excel 2013 kein Problem mehr

Die Frage nach dem »Wie?« – PivotTable-Empfehlungen

Zum Abschluss dieses Abschnitts zum Thema PivotTable möchten wir Ihnen noch die Funktion »Empfohlene PivotTables« vorstellen, da sie sehr hilfreich sein kann.

Informationen zum Thema finden Sie im Abschnitt »Empfohlene PivotTable« ab Seite 53.

Hinweis

Sie wissen nun grundsätzlich, wie Sie sämtliche Daten Ihrer Tabellen analysieren können, und sind so in der Lage, bessere Unternehmensentscheidungen zu treffen. Manchmal ist es jedoch schwer, einen Anfang zu finden – gerade dann, wenn besonders viele Daten vorliegen.

Das folgende Praxisbeispiel in Excel 2013 finden Sie im Ordner `Ms5-235\Kap01` in der Arbeitsmappe `PowerPivot_Beispielerarbeitet.xlsx`. Möchten Sie die Übung Schritt für Schritt nachvollziehen und mitmachen, öffnen Sie die Datei `PowerPivot_Beispielerleer.xlsx`.



So unterstützt Sie Excel dabei:

- Öffnen Sie die Excel-Datei und hier das Tabellenblatt *Projektstunden*. Die Tabelle enthält knapp 900 Datensätze, die ausgewertet werden sollen. Nur wie?
- Nutzen Sie den »Ideenlieferant« Excel. Markieren Sie eine Zelle innerhalb der Tabelle. Öffnen Sie die Registerkarte *EINFÜGEN* und klicken Sie in der Gruppe *Tabellen* auf die Schaltfläche *Empfohlene PivotTables* (Abbildung 1.57).

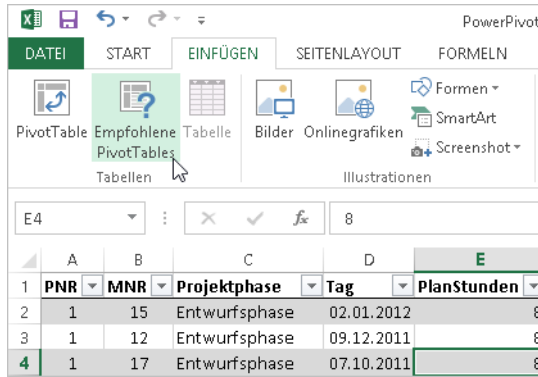


Abbildung 1.57: Empfohlene PivotTables – eine hilfreiche Funktion für erste Schritte in Analyseprozessen

3. Es öffnet sich das Dialogfeld *Empfohlene PivotTables* (Abbildung 1.58). Darin aufgelistet sehen Sie verschiedene Pivot-Auswertungsmöglichkeiten. Im rechten Bereich des Dialogfelds sehen Sie eine kleine Vorschau des möglichen Ergebnisses.
4. Markieren Sie die gewünschte PivotTable und bestätigen Sie Ihre Wahl mit **OK**.

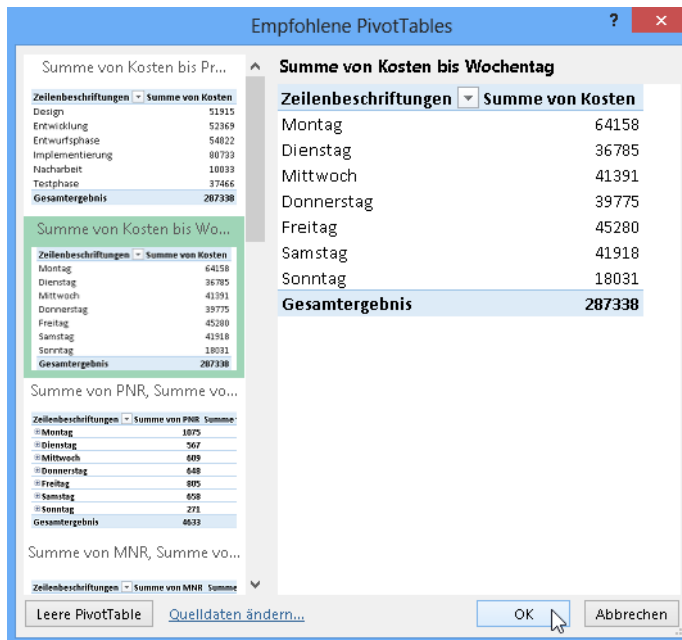


Abbildung 1.58: Es gibt verschiedene Empfehlungen auf Basis Ihrer Daten, zwischen denen Sie wählen können

Excel erstellt die PivotTable in einem neuen Arbeitsblatt und zeigt die Feldliste an, sodass Sie die PivotTable-Daten jederzeit neu anordnen können. Ist keine der angezeigten Auswertungen passend, können Sie aus dem Dialogfeld *Empfohlene PivotTables* heraus auch eine leere PivotTable erstellen. Klicken Sie dazu auf die gleichnamige Schaltfläche wie in Abbildung 1.58 unten links dargestellt.

Interaktiv und visuell – Microsoft Power View

Die Begriffe »Self-BI« und »Power View« sind in aller Munde – und mit Excel 2013 rutscht Power View noch mehr in den Vordergrund. Aber was ist das überhaupt? Wie funktioniert es und was bringt es? Das Marktforschungsunternehmen Gartner schätzt, dass die Menge der weltweit vorhandenen Daten jährlich um 59 Prozent steigt – dabei entfallen 70 bis 85 Prozent auf unstrukturierte Daten. Diese wiederum sollen mit Business Intelligence-Anwendungen wie Microsoft Excel, PowerPivot für Excel 2013 und Power View bearbeitet, strukturiert und analysiert werden. Dies ist aufgrund des großen Funktionsumfangs zwar grundsätzlich möglich, jedoch dennoch eine große Herausforderung.

Bitte beachten Sie: Power View erfordert Microsoft Silverlight. Wenn Sie bei der erstmaligen Nutzung nicht über Silverlight verfügen, installieren Sie bitte Silverlight und starten Sie Excel neu.

Achtung

Volle Kraft voraus: Das ist Power View

Power View ermöglicht die interaktive Durchsuchung, Visualisierung und Darstellung von Daten und eine intuitive Berichterstellung für Zahlenjongleure und Entscheider. Power View ist sowohl in SharePoint als auch – jetzt in noch größerem Funktionsumfang – in Microsoft Excel 2013 verfügbar. Viele der jetzt in Power View in Excel 2013 verfügbaren Funktionen waren bereits in Power View in SharePoint 2010 enthalten und wurden um weitere neue Funktionen ergänzt.

Auf der Registerkarte *POWER VIEW* finden Sie die interaktiven und leistungsfähigen Datenauswertungs-, Datendarstellungs- und Datenpräsentationsfeatures, die sich einfach anwenden lassen. Power View ermöglicht es Ihnen, Diagramme, Datenschnitte und weitere Datenvisualisierungen auf einem einzelnen Blatt zu verwalten.

Damit können Anwender agil und intuitiv aussagekräftige Berichte erstellen. Um die Daten zum Leben zu erwecken, werden einfach Tabellen, Karten und eine Vielzahl von Diagrammen in die interaktive Sicht gezogen.

Mithilfe von Power View können Sie mit Daten interagieren ...

- ▶ ... die sich in derselben Excel-Arbeitsmappe wie das Power View-Blatt befinden,
- ▶ ... die sich in Datenmodellen in Excel-Arbeitsmappen befinden, die in einem PowerPivot-Katalog veröffentlicht wurden,
- ▶ ... die sich in tabellarischen Modellen befinden, die auf SQL Server 2012-Analysis Services-Instanzen (SSAS) bereitgestellt wurden

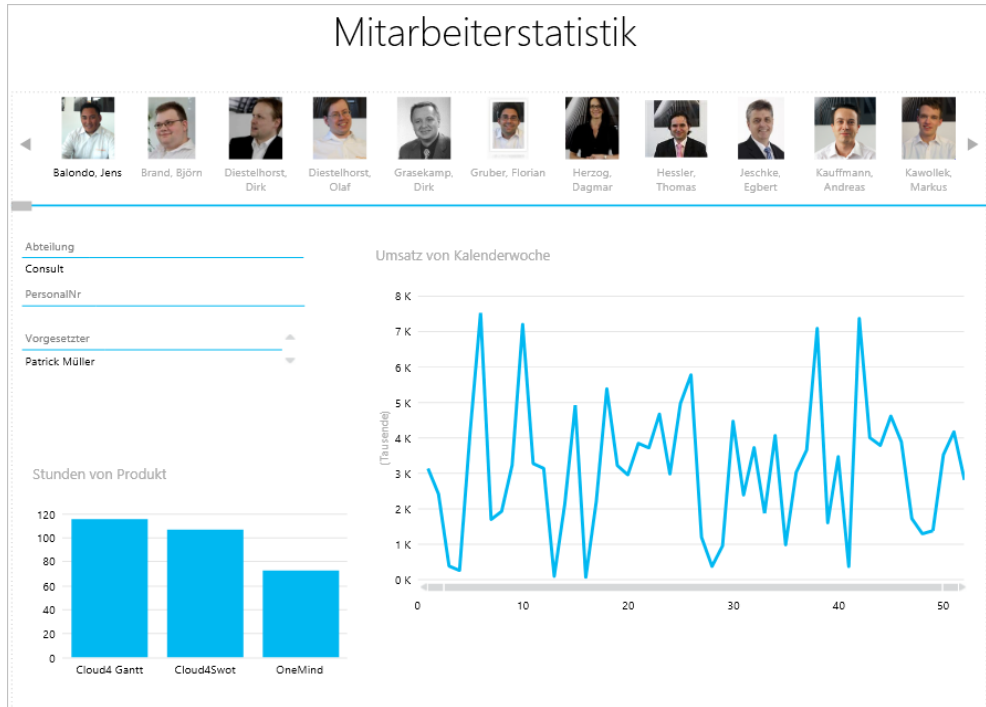


Abbildung 1.59: Darstellung einer Power View-Mitarbeiterstatistik in SharePoint

Ein Power View-Bericht ist jederzeit bereit, präsentiert zu werden. Dabei werden die Daten dynamisch und nicht statisch wiedergegeben – dabei stehen alle Informationen sofort auf einen Blick zur Verfügung. Power View ist mit einem Lese- und Vollbildpräsentationsmodus ausgestattet, in denen das Menüband und andere Entwurfstools ausgeblendet werden, um mehr Raum für Visualisierungen bereitzustellen. Der Export einer interaktiven Version des Power View-Berichts nach PowerPoint lässt sich schnell und einfach durchführen. Wissenswert ist dabei, dass für jede Sicht in Power View eine separate PowerPoint-Folie generiert wird. Die Interaktion mit nach PowerPoint exportierten Power View-Berichten entspricht in etwa der Interaktion mit Power View-Sichten im Lese- und Vollbildmodus von Power View.

Neuigkeiten im Überblick

Power View – vor allem auch in Kombination mit der PivotTable – ist ein unglaublich mächtiges Berichtswerkzeug. Da wir jedoch ein Buch über Excel-Formeln und -Funktionen schreiben, reicht der Platz an dieser Stelle leider nicht aus, Ihnen alle Funktionen und Vorteile von Power View darzustellen und zu beschreiben.

Da wir das Thema jedoch sehr spannend finden, ist uns wichtig, Ihnen einen Überblick über die Möglichkeiten und die grundsätzliche Funktionsweise zu geben, vor allem auch durch das Beispiel im Verlauf dieses Kapitels.

Zunächst wollen wir jedoch in der Tabelle 1.2 stichpunktartig die Neuerung in Power View in Excel 2013 auflisten.

Funktion	Beschreibung
Präsentationsbereit	<ul style="list-style-type: none"> – Daten können jederzeit mit den Echtzeiten präsentiert werden – Berichte werden direkt angezeigt – ohne Vorschau – Bei der Präsentation der Berichte in SharePoint sind aus Platzgründen das Menüband und andere Tools ausgeblendet – Aktionen wie Filtern und Hervorheben sind Standard
Basierend auf einem Datenmodell	<ul style="list-style-type: none"> – Excel 2013-Daten können direkt als Grundlage für Power View in Excel verwendet werden – Beim Hinzufügen von Tabellen und Definieren von Beziehungen wird ein Datenmodell im Hintergrund erstellt – Power View-Berichte können auch auf Basis eines tabellarischen Modells erstellt werden – Tabellarische Modelle und Datenmodelle fungieren als Brücke zwischen den Komplexitäten von Back-End-Datenquellen und der Perspektive der Daten
Erstellen von Diagrammen und anderen Visualisierungen	<ul style="list-style-type: none"> – Visualisierungen in Form von Tabellen und Matrizen über Kreis-, Balken- und Blasendiagramme bis hin zu Gruppen mehrerer Diagramme lassen sich in Minutenschnelle erstellen – Tabellen können jederzeit in andere Visualisierungen konvertiert werden
Filterung und Hervorhebung von Daten	<ul style="list-style-type: none"> – Power View erkennt Beziehungen in Datenmodellen eigenständig. So können Sie eine Visualisierung verwenden, um alle Visualisierungen in einem Blatt bzw. einer Sicht zu filtern und hervorzuheben. – Visualisierungen eines Blatts oder einer Sicht können gefiltert werden
Slicer	<ul style="list-style-type: none"> – Ähnlich wie in Excel gibt es auch in Power View Slicer (Datenschnitte) – Slicer bieten die Option, Visualisierungen zu filtern
Sortierung	<ul style="list-style-type: none"> – Tabellen, Matrizen, Balken- und Säulendiagramme u.v.m. kann in Power View sortiert werden – Sie können entweder nach Attributen, z.B. Produktname, oder nach numerischen Werten, z.B. Gesamtumsatz, aufsteigend oder absteigend sortieren
Power View in Excel Services und Excel Web App, und in Office 365	<ul style="list-style-type: none"> – Wenn Sie Power View-Blätter in Excel erstellen, können Sie sie lokal in Excel Services und in Office 365 anzeigen lassen und interaktiv nutzen – Power View-Blätter in Excel 2013 können nur auf einem Clientcomputer bearbeitet werden – Power View wird lokal nicht unter Excel Web App unterstützt – In Office 365 zeigen Sie Power View-Blätter mit Excel Web App Data Center, einer Komponente von SharePoint Online, anstatt mit Excel Services, an
Kreisdiagramme	<ul style="list-style-type: none"> – Kreisdiagramme sind in Power View einfach oder anspruchsvoll gestaltet – die Funktionalitäten wurden erheblich erweitert
Karten	<ul style="list-style-type: none"> – Karten in Power View verwenden Bing-Kartenkacheln – Zoomen und schwenken ist damit problemlos möglich – Orte und Felder sind auf der Karte als Punkte dargestellt: Je höher der Wert ist, desto größer ist der Punkt ▶

Tabelle 1.2
Power View in Excel 2013 – und auch SharePoint 2013 – hält viele spannende, neue Funktionen bereit

Funktion	Beschreibung
Key Performance Indicators (KPIs)	– Fügen Sie dem Power View-Bericht KPIs hinzu, um Ziele noch deutlicher zu visualisieren
Hierarchien	– In Datenmodellen vorhandene Hierarchien können auch in Power View-Berichten abgebildet werden
Drillup und Drilldown	– Mit der Drillup-/Drilldownfunktion können Sie Ebenen in Berichten anzeigen oder ausblenden
Berichtsstile, Designs und Anpassung der Textgröße	– Power View verfügt über viele neue Berichtsdesigns – Ein geändertes Design gilt für alle Power View-Sichten im Bericht bzw. Blätter in der Arbeitsmappe
Hintergründe	– Das Anpassen von Hintergrundfarben und auch das Einfügen und Bearbeiten von Bildern ist um ein Vielfaches verbessert worden
Links	– Links können ganz einfach in Power View-Sichten eingefügt und bearbeitet werden – Es können Adressen jeglicher Art dahinter gelegt werden
Drucken	– Das Ausdrucken von Power View-Blättern in Excel und Sichten in SharePoint funktioniert problemlos – Es wird immer das ausgedruckt, was auf dem Blatt oder der Sicht gerade sichtbar ist
Sprachen und Leserichtung	– Power View unterstützt sowohl in Excel als auch in SharePoint Sprachen mit Leserichtung von rechts nach links
Abwärts- und Aufwärtskompatibilität in Power View	– Power View 2010 Dateien können in 2013 geöffnet und bearbeitet werden – umgekehrt funktioniert dies noch nicht
Power View und Excel Services	– Power View und Excel Services sind abwärts-, aber nicht aufwärtskompatibel

Erstellen eines Power View-Blatts

Genug der Theorie. Nachfolgend werden wir einen einfachen Power View-Bericht in Excel 2013 erstellen, um Ihnen die Komplexität dieses Tools zu demonstrieren. Die Erstellung eines Power View-Blatts in Microsoft Excel 2013 ist denkbar einfach. Wir bleiben bei dem im Abschnitt »Aus Daten werden Erkenntnisse – Arbeiten mit der PivotTable« ab Seite 51 erstellten Beispiel.



Die folgenden Praxisbeispiele in Excel 2013 finden Sie im Ordner `Ms5-235\Kap01` in der Arbeitsmappe `PowerView_Beispiel_bearbeitet.xlsx`. Möchten Sie die Übung Schritt für Schritt nachvollziehen, öffnen Sie die Datei `PowerView_Beispiel_leer.xlsx`.

Ziel ist es, darzustellen, inwieweit die geplanten Stunden für ein Projekt von den tatsächlich geleisteten abweichen.

1. Öffnen Sie in der Excel-Datei das Tabellenblatt *Auswertung MA-Projekt-Stunden*.
2. Klicken Sie auf *EINFÜGEN* und in der Gruppe *Berichte* auf *Power View*.

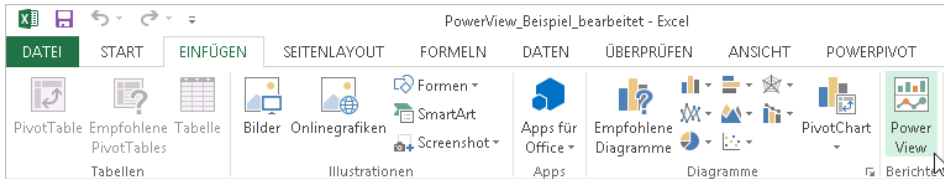


Abbildung 1.60: Power View lässt Ihre Zuschauer staunen und Ihre Auswertungen strahlen

3. Power View wird nun geladen. Abschließend sehen Sie das leere Power View-Blatt, das nun mit den Daten gefüllt werden kann.

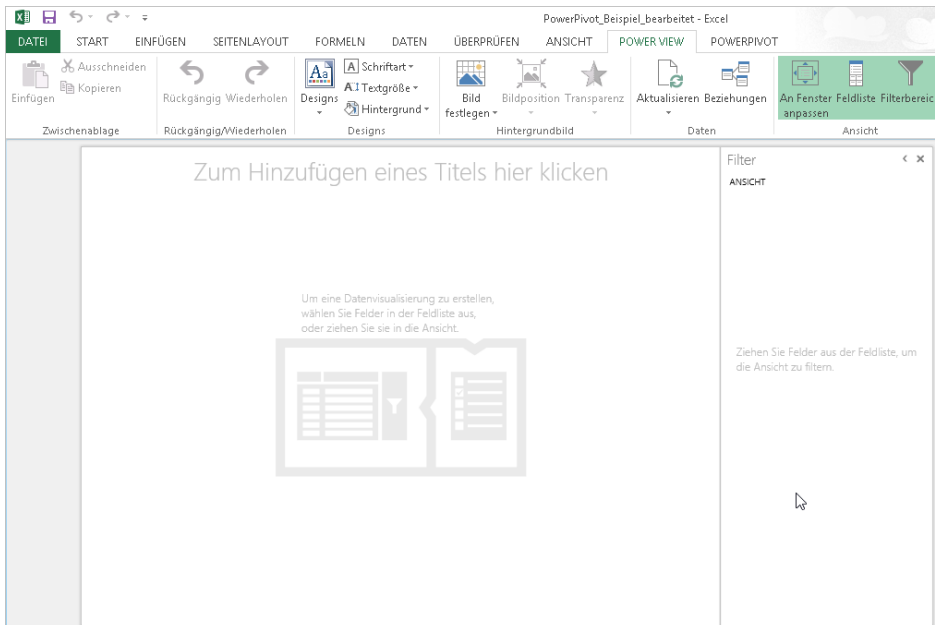


Abbildung 1.61: Los geht's – es stehen Ihnen alle Möglichkeiten offen

Wie die Abbildung 1.61 zeigt, werden über eine weitere Registerkarte *POWER VIEW* zahlreiche Funktionen zur Bearbeitung des Power View-Blatts zur Verfügung gestellt.

Bitte beachten Sie: Die Daten der Excel-Tabelle beruhen auf den vorangegangenen Beispielen, in denen wir ein Datenmodell und Beziehungen zwischen den Daten erstellt haben. Wäre dies nicht der Fall gewesen, müssten Sie das an dieser Stelle nachholen, da nur so die Auswertung tabellenübergreifend funktioniert.

Hinweis

4. Zunächst soll ein Diagramm erstellt werden, das die Plan- und Ist-Stunden für die einzelnen Projektphasen darstellt. Öffnen Sie in der Power View-Feldliste den Bereich *xls_Projektstunden* und aktivieren Sie die Kontrollkästchen der Einträge Σ *PlanStunden*, Σ *Stunden* und *Projektphase*.

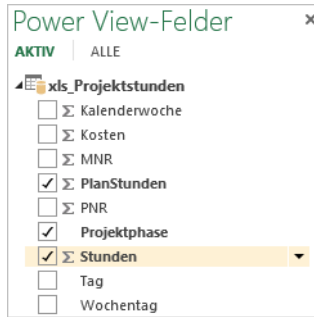


Abbildung 1.62: Aktivieren Sie nur das, was Sie sehen wollen

- Die aktivierten Daten stehen nun im Power View-Blatt. Klicken Sie auf *ENTWURF* und in der Gruppe *Visualisierung wechseln* auf die Schaltfläche *Anderes Diagramm*. Wählen Sie den Eintrag *Linie* (Abbildung 1.63).

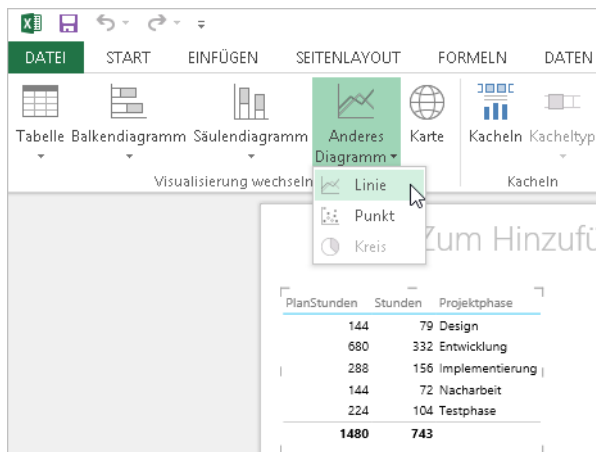


Abbildung 1.63: Bestimmen Sie die Darstellung der Informationen selbst

- Die Datenliste wird in ein Liniendiagramm umgewandelt. Verbreitern Sie die Diagrammansicht etwas, damit die Darstellung der Inhalte klarer wird (Abbildung 1.64).

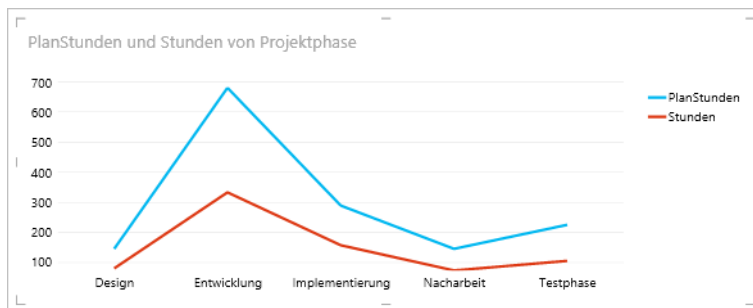


Abbildung 1.64: Das erste Diagramm steht

Sie möchten das Diagramm nun nach Projekten filtern, um zu sehen, für welches Projekt welche Stunden angefallen sind.

7. Klicken Sie in eine leere Stelle des Power View-Blatts (außerhalb des Diagramms). Aktivieren Sie das Kontrollkästchen *Projektname* unter *xls_Projekte* sowie die Kontrollkästchen Σ *PlanStunden*, Σ *Stunden* unter *xls_Projektstunden* (Abbildung 1.65).

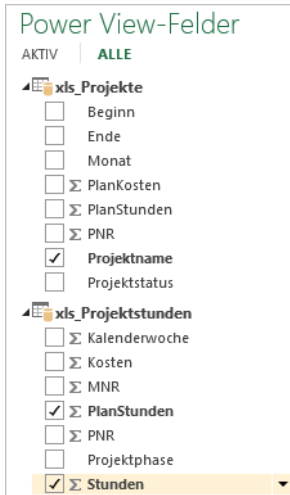


Abbildung 1.65: Erweitern Sie Ihre Analysedaten

8. Klicken Sie auf *ENTWURF* und in der Gruppe *Visualisierung wechseln* auf die Schaltfläche *Balkendiagramm*. Wählen Sie den Eintrag *Gruppiertes Balkendiagramm* (Abbildung 1.66).

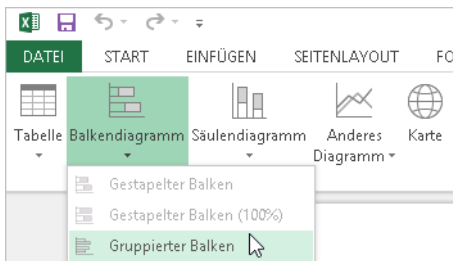


Abbildung 1.66: Gruppierete Balken sind ideal für die Darstellung zweier Werte im Vergleich

9. Die Daten werden in einem Balkendiagramm dargestellt. Verbreitern Sie die Diagrammansicht etwas.



Abbildung 1.67: Soll und Ist im Balkendiagramm – durch unterschiedliche Farben schön visualisiert

10. Zu guter Letzt fügen wir noch eine Liste der Mitarbeiter mit den tatsächlich benötigten Ist-Stunden ein.
11. Klicken Sie in eine leere Stelle des Power View-Blatts (außerhalb des Diagramms). Aktivieren Sie das Kontrollkästchen *Nachname* unter *xls_Mitarbeiter* sowie das Kontrollkästchen Σ *Stunden* unter *xls_Projektstunden*.
12. Um dem Power View-Blatt einen Titel hinzuzufügen, klicken Sie in die Titelliste und tragen den gewünschten Text ein (Abbildung 1.68).



Abbildung 1.68: Geben Sie Ihrer Analyse eine aussagekräftige Überschrift

Das Power View-Blatt ist nun fertig und sollte etwa so wie in Abbildung 1.69 aussehen.

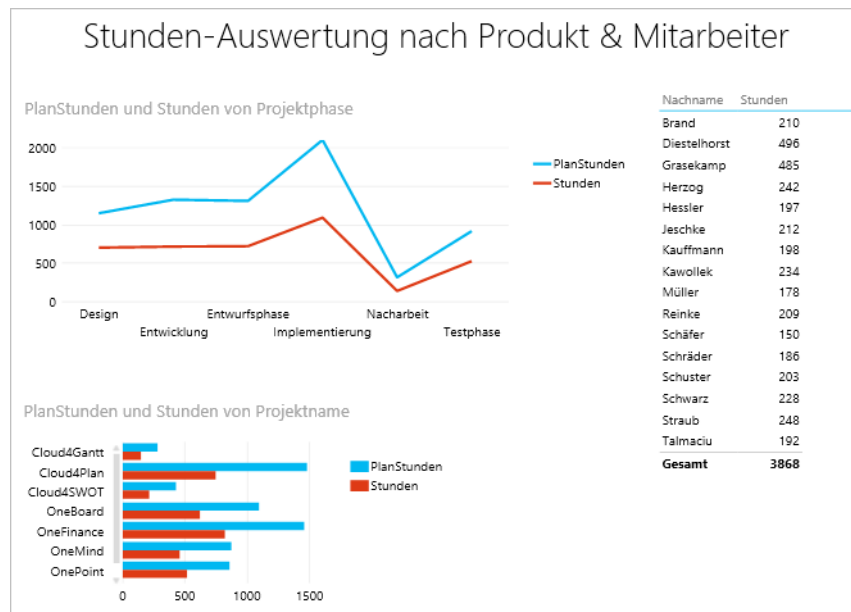


Abbildung 1.69: Überblick in wenigen Schritten – Power View macht's möglich

Speichern Sie es und aktualisieren Sie die Inhalte über die Registerkarte *POWER VIEW* in der Gruppe *Daten*. Klicken Sie auf die Schaltfläche *Aktualisieren* und wählen Sie den Eintrag *Alle aktualisieren* (Abbildung 1.70).

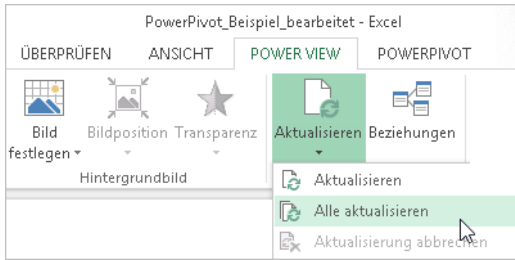


Abbildung 1.70: Damit Ihre Auswertung richtig angezeigt wird, regelmäßig aktualisieren

Testen Sie nun Ihre Auswertungen: Klicken Sie auf eines der Projekte, werden im Diagramm automatisch die verbrauchten Soll- und Ist-Stunden pro Projektphase sowie bei den Mitarbeitern die jeweils tatsächlich benötigten Stunden. Egal, welches Projekt Sie auswählen – die Auswertung aktualisiert sich automatisch.

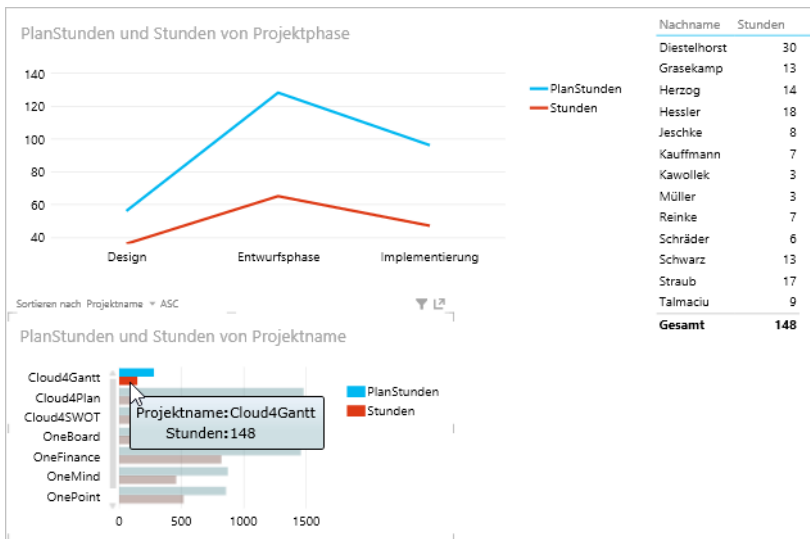


Abbildung 1.71: Auswertungen auf Knopfdruck

Herzlichen Glückwunsch! Sie haben Ihr erstes eigenes Power View-Analyseblatt erstellt. Ein einfaches Beispiel, das noch viel Raum für weitere Auswertungen lässt.

Kennen Sie Robert Bruckner? Wenn nicht, dann sollten Sie ihn und sein Wissen über Power View dringend kennenlernen. Wie bereits erwähnt, haben wir in diesem Buch nicht den Platz, ausführlich auf das Thema »Power View« einzugehen. Daher möchten wir es nicht versäumen, Sie auf ein geniales und eingängiges Power View Beispiel aufmerksam zu machen:

Tipp

Robert M. Bruckner, Principal Software Architect bei Microsoft, hat dieses Beispiel in einem Vortrag auf <http://www.katapult.tv> – dem Webcast-Kanal von Microsoft – gezeigt und uns und Ihnen freundlicherweise zur Verfügung gestellt.

In diesem Beispiel können Sie sich alle Haifischangriffe nach Land seit 1950 anzeigen lassen.

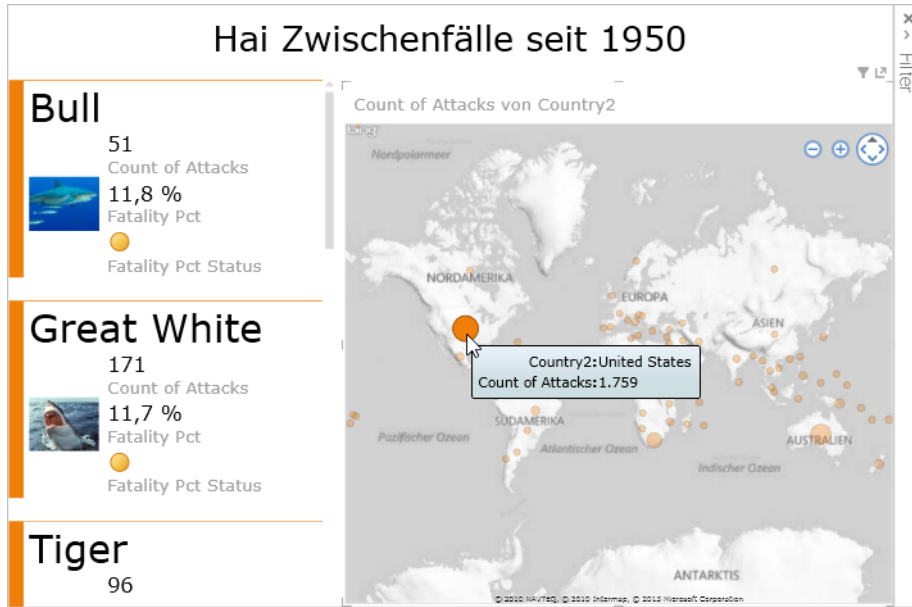


Abbildung 1.72: Haifischalarm – Wo hat wann ein Hai Menschen attackiert?

Oder alle Olympischen Medaillen – ebenfalls nach Land.

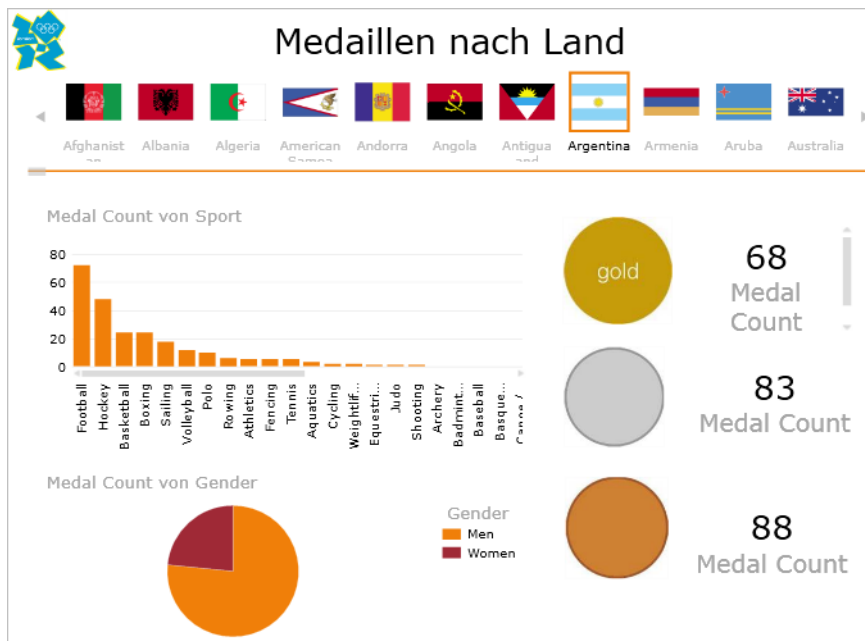


Abbildung 1.73: Der Klick auf eine Fahne reicht, um die Anzahl der gewonnenen Medaillen anzuzeigen

Eine sehr gelungene Darstellung der Mächtigkeit von Power View in Excel 2013.

Sie finden dieses Power View-Praxisbeispiel im Ordner `\Ms5-235\Kap01` in der Arbeitsmappe *Olympische Spiele und Haiangriffe.xlsx*.



Fazit

Sie haben nun in den letzten beiden Abschnitten zu den Themen PowerPivot und Power View gesehen, wie viele Möglichkeiten in den beiden Anwendungen stecken. Auch wenn es nur einfache Beispiele waren, sollten diese doch ausreichen, um einen Eindruck davon zu bekommen, was tatsächlich möglich ist. Mit diesen beiden Tools sind Sie bezüglich Ihrer Analysen auf der sicheren Seite – und es gibt nahezu keine Grenzen mehr.

Durch die Möglichkeit des Veröffentlichens der Daten ist zusätzlich ein weiterer Schritt in Richtung Zukunft getan – mit Microsoft Excel 2013. Wir wünschen Ihnen nun viel Erfolg beim Erforschen des umfangreichen Funktionsschatzes von Excel in den Versionen bis 2013.

Kapitel 2

Von der Zahl zur Formel

Eingeben von Formeln	82
Operatoren	82
Bezüge in Formeln	89
Matrixformeln	96
Tipps & Tricks	97

Wenn Excel für Sie rechnen soll, müssen Sie Formeln eingeben, aus denen das Programm die angestrebten Ergebnisse bilden kann. Die Eingabe von Formeln läuft in einem anderen Modus ab als die übrige Dateneingabe. Sie als Anwender werden diesen Unterschied jedoch fast nicht bemerken.

Im Prinzip ist die Vorgehensweise in den Excel-Arbeitsblättern mit der auf einem Rechenblatt aus Papier vergleichbar. Der Unterschied zwischen dem papiernen Rechenblatt und Excel ist der, dass Sie beim handschriftlichen Rechnen die Rechenregeln im Kopf haben und wissen müssen, wie man damit zu einem brauchbaren Ergebnis kommt. Die Ausgangswerte, die Sie zuerst in das Rechenblatt eintragen, werden in Excel als Eingabewerte bezeichnet. Die von Ihnen aufgestellten Rechenregeln, durch die diese Eingabewerte zu einem Ergebnis geführt werden, legen Sie in einer Formel fest.

Der wesentliche Vorteil des elektronischen Rechenblatts, also der Tabellenkalkulation, ist der, dass Sie das einmal erstellte Rechenmodell immer wieder verwenden können. Sie geben lediglich neue Eingabewerte ein und die Formeln mit der darin enthaltenen Rechenregel führen nach der stets gleichen Methode zum Ergebnis. Kurz: Wenn Sie einmal eine Tabelle korrekt erstellt haben, können Sie beliebig oft Werte nach diesen Regeln berechnen lassen.

Es ist auch wichtig zu wissen, dass Sie in Excel keine algebraischen Buchstabenformeln berechnen können. Sämtliche Berechnungen müssen auf natürlichen Zahlenwerten beruhen, welche real in eine Tabelle eingetragen wurden. Ebenso ist es bei Diagrammen nicht möglich, einen Graphen direkt durch eine Formel berechnen und zeichnen zu lassen: Alle Werte, die im Diagramm dargestellt werden sollen, müssen sich real in einer Tabelle befinden bzw. dort berechnet worden sein.

Eingeben von Formeln

In Excel gilt: Wenn das erste Zeichen einer Eingabe das Gleichheitszeichen (=) ist, wird die nachfolgende Eingabe als Berechnungsformel ausgewertet. Sie können eine Formeleingabe auch mit dem Plus- oder Minuszeichen einleiten. Excel setzt jedoch später noch das Gleichheitszeichen davor.

Eine Formel wird immer aus Operanden und Operatoren gebildet. Dabei sind die Operanden die Werte, mit denen gerechnet werden soll, und die Operatoren stellen die Rechenvorschriften dar, die zur Berechnung angewendet werden (Abbildung 2.1).

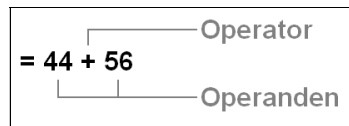


Abbildung 2.1: Operanden und Operatoren bilden die Grundbestandteile einer Formel

Operatoren

Bei den Operatoren, welche Sie in Formeln verwenden können, herrscht eine größere Vielfalt als gemeinhin bekannt. Deshalb wollen wir hier einmal alle Operatoren systematisch betrachten. Beginnen wir mit den am häufigsten verwendeten Operatoren in Ihren Excel-Formeln.

Arithmetische Operatoren

Mit den arithmetischen Operatoren führen Sie elementare Rechenoperationen durch und erzeugen als Ergebnis numerische Werte. Die Tabelle 2.1 fasst die arithmetischen Operatoren zusammen, die in Excel-Formeln eingesetzt werden können.

Operator	Operation	Beispielformel	Ergebnis
+	Addition	=15+13	28
-	Subtraktion ^a	=104-74	30
/	Division	=42/3	14
*	Multiplikation	=5*8	40
^	Potenzierung	=2^8	256
%	Division durch Hundert (Prozent) ^b	=140%	1,4

^a Steht das Minuszeichen vor einem Zahlenwert (Negation), wird es als Vorzugsoperator ausgewertet.

^b Hier handelt es sich wie bei Vorzeichen um einen »unären«, d.h. einseitigen Operator. Er benötigt nur einen Operanden.

Diese Beispiele finden Sie im Arbeitsblatt *Arithmetische Operatoren* in der Arbeitsmappe *Kap02.xlsx*. Sie finden diese im Ordner `\Ms5-235\Kap02`.



Mit diesen Operatoren können Sie in Excel-Tabellen Formeln erstellen, indem Sie die Formeln unmittelbar in Zellen eingeben. Nach der Eingabe wird in den jeweiligen Zellen das Ergebnis der Formel angezeigt, wobei in der Bearbeitungsleiste die Formel als Zelleninhalt zu sehen ist.

Die unmittelbare Eingabe von numerischen Werten in Formeln wird als die Eingabe von Konstanten bezeichnet. Die Verwendung von Konstanten ist eigentlich die Ausnahme in Formeln. Sie sollten sie möglichst vermeiden und durch Bezüge auf Eingabezellen mit den an der Formel beteiligten Werten ersetzen. Ansonsten nehmen Sie sich den oben erwähnten Vorteil von elektronischen Rechenblättern.

Hinweis

Die in der Tabelle dargestellten Formeln dienen lediglich dem Verständnis der Wirkungsweise der Operatoren.

Rechenreihenfolge durch Klammern bestimmen

Wenn Sie die Wertigkeit bzw. Rangfolge der Operationen bei der Berechnung ändern möchten oder aus arithmetischen Gründen ändern müssen, sind die Ausdrücke, die zuerst ausgewertet werden sollen, in runde Klammern einzuschließen. Testen Sie die Wirkungsweise von runden Klammern, indem Sie die beiden in der Tabelle 2.2 dargestellten Formeln in unterschiedliche Zellen eingeben und das Ergebnis vergleichen.

Formel	Ergebnis
=3+8*4	35
=(3+8)*4	44

Tabelle 2.2
Auswirkung von runden Klammern

Wichtig

In einer Formel muss die Anzahl der öffnenden runden Klammern immer gleich der Anzahl der schließenden runden Klammern sein. Excel meldet sonst einen Fehler und markiert die zu korrigierende Stelle in der Formel bzw. bietet die Korrektur der Formel an.



Die Tabelle 2.2 finden Sie in der Arbeitsmappe *Kap02.xlsx* auf dem Blatt *Rangfolge*. Die Arbeitsmappe befindet sich im Ordner `\\Ms5-235\Kap02`.

Damit Sie nicht die Übersicht verlieren und leichter kontrollieren können, ob die Anzahl der öffnenden Klammern gleich der Zahl der schließenden Klammern ist, stellt Excel Ihnen folgende Eingabehilfen zur Verfügung:

- ▶ Bei Eingabe einer schließenden Klammer wird in der Bearbeitungsleiste – bei direkter Zellbearbeitung auch in der Zelle selbst – die geöffnete Klammer kurzzeitig **fett** hervorgehoben
- ▶ Wenn Sie eine bestehende Formel bearbeiten, wird – je nach Einstellung und Bearbeitungsart – ein Klammerpaar kurzzeitig dann **fett** hervorgehoben, wenn Sie die Einfügemarke über eine der beiden Klammern bewegen

Rangfolge der arithmetischen Operatoren

Wenn ein Ausdruck mehrere Operatoren enthält, bestimmt die Rangfolge der Operatoren die Reihenfolge, in der die Elemente des Ausdrucks ausgewertet werden. Die Standardrangfolge kann durch das Setzen von runden Klammern innerhalb einer Elementgruppe geändert werden. Die Tabelle 2.3 zeigt die Rangfolge, in der die Operatoren in einer Excel-Formel ausgewertet werden.

Tabelle 2.3
Operatoren und
ihre Rangfolge

Rang	Operator	Beschreibung
1	–	Negation eines Werts (z.B. –34)
2	%	Division eines Werts durch Hundert (Prozent)
3	^	Potenzierung eines Werts
4	* und /	Multiplikation und Division
5	+ und –	Addition und Subtraktion

Sicher erinnern Sie sich noch an die alte arithmetische Weisheit »Punktrechnung geht vor Strichrechnung!«, welche Ihnen im Zweifelsfall weiterhilft.

Hinweis

Enthält eine Formel Operatoren, die die gleiche Priorität besitzen, wertet Excel die Operatoren – und somit die Formel – von links nach rechts aus.

Vergleichsoperatoren für die Logik

Mit den Vergleichsoperatoren können Sie Werte, Texte oder Zellinhalte vergleichen. Am häufigsten setzen Sie solche Ausdrücke in den Logik-Funktionen, z.B. WENN(), ein. Das Ergebnis ist immer ein Wahrheitswert (boolescher Wert). Die Tabelle 2.4 zeigt alle booleschen Operatoren und Anwendungsbeispiele zum Verständnis von Vergleichsausdrücken.

Operator	Operation (Vergleich)	Beispielformel	Ergebnis
=	ist gleich	=5=8	FALSCH
<	kleiner als	=5<8	WAHR
>	größer als	=5>8	FALSCH
<=	kleiner oder gleich	=5<=8	WAHR
>=	größer oder gleich	=5>=8	FALSCH
<>	ungleich (nicht)	=5<>8	WAHR

Tabelle 2.4
Boolesche
Vergleichs-
operatoren

Die Tabelle 2.4 finden Sie in der Arbeitsmappe *Kap02.xlsx* auf dem Blatt *Vergleichsoperatoren*. Die Arbeitsmappe befindet sich im Ordner *\Ms5-235\Kap02*.



Der Textoperator &

Manchmal ergibt sich die Notwendigkeit, die Ergebnisse einer oder mehrerer Formeln in einer Zelle zusammenzufassen. Hier kommt der Textoperator »&« (kaufmännisches Und-Zeichen) zum Einsatz. Wenn Sie zwei Werte – gleichgültig welchen Datentyps – mit dem Textoperator verbinden, entspricht das Ergebnis immer dem Datentyp »Text«.

Dies bedeutet, dass Zahlenwerte von Excel automatisch zu Text konvertiert werden. Sie können so mit dem Textoperator zwei Zahlenwerte als Text in einer Zelle unterbringen. Die sich daraus ergebende Textfolge kann zu weiteren Berechnungen nicht mehr unmittelbar herangezogen werden!

Sollte in der Formel statt eines Zellbezugs unmittelbar Text verwendet werden, muss dieser Text in Anführungszeichen gesetzt werden. Dagegen müssen Zahlen nicht in Anführungszeichen stehen. Wenn Sie Zellbezüge verwenden, entfallen die Anführungszeichen ebenfalls – gleichgültig, welcher Datentyp in den Zellen anzutreffen ist.

Hinweis

Angenommen, Sie möchten den Inhalt der Zellen *A1* und *A2* durch den Textoperator verbinden, so ergibt dies folgende Formel:

=A1&A2

Durch den Textoperator werden die Werte unmittelbar aneinander gefügt. Wenn Sie jedoch zwischen den Werten ein Leerzeichen wünschen, müssen Sie dies eigens eingeben, und zwar in Anführungszeichen. Angenommen, zwischen den Werten der Zellen *A1* und *A2* soll sich ein Leerzeichen befinden. In diesem Fall ist die obige Formel wie folgt abzuwandeln:

=A1&" "&A2

Schauen Sie dazu die Beispiele im Arbeitsblatt *Textoperator* in der Arbeitsmappe *Kap02.xlsx* an. Sie finden diese im Ordner *\Ms5-235\Kap02*. Die Abbildung 2.2 zeigt den Inhalt dieses Arbeitsblatts.



	A	B	C	D	E
1	Textoperator				
2				Berechnung	Formel
3	Deutsch	land		Deutschland	=A3&B3
4	weiße	Feder		Federweiße	=B4&A4
5					
6	Excel		2007	Excel 2007	=A6&" "&B6
7	Excel	Funktionen		Excel-Funktionen	=A7&"-"&B7
8		50000	Einwohner	50000 Einwohner	=A8&" "&B8
9					
10		47	11	4711	=A10&B10
11		6	49	6 aus 49	=A11&" aus "&B11
12		1	25000	1:25000	=A12&" ":"&B12
13					
14	Helmut	Reinke		Helmut Reinke	=A14&" "&B14
15	Sara	Unverhau		Unverhau, Sara	=B15&" "&A15
16					

Abbildung 2.2: Beispiele für den Einsatz des Textoperators

Bezugsoperatoren

Durch die Bezugsoperatoren können Sie in einer Formel oder Funktion bestimmte Zellen oder Zellbereiche zur Berechnung übergeben, d.h., sie verknüpfen Zellbereiche für die Durchführung von Berechnungen. Sie haben die folgenden Operatoren zur Verfügung:

- ▶ **Bereichsoperator** : (Doppelpunkt) Erstellt einen Bezug auf alle Zellen, die zwischen zwei Bezügen liegen, einschließlich der beiden Bezugszellen, z.B. B3:B20
- ▶ **Verbindungsoperator** ; (Semikolon) Ermöglicht die Verbindung mehrerer Zellen oder Bezüge in einem Ausdruck, z.B. SUMME (B3:B20;D3:D20)
- ▶ **Schnittmengenoperator** ' ' (Leerschritt) Erstellt einen Bezug zu Zellen, die für beide Bezüge gleich sind, d.h. die Schnittmenge beider Bereiche darstellen, z.B. B7:D7 C6:C8

Der Bereichsoperator

Unter einem Bereich versteht man einen Teil der Arbeitsblattmatrix, also neben- und untereinander liegende Zellen. Wenn zwei Zellbezüge durch einen Doppelpunkt verbunden werden, bilden sie mit den dazwischen liegenden Zellen einen Bereich. Die Größe und Form der Bereiche kann unterschiedlich sein. In Funktionen gilt ein Bereich – gleichgültig, wie groß er ist – als ein Argument.

Mögliche Bereichsbezüge

Obwohl die Angabe von Bereichsbezügen einfach ist, wollen wir hier die möglichen Varianten veranschaulichen.

The image shows an Excel spreadsheet grid with columns A through G and rows 1 through 14. Three ranges are highlighted with dashed lines and labels:

- A yellow rectangle labeled **B3:D10** covers cells from B3 to D10.
- A blue vertical rectangle labeled **F2:F9** covers cells from F2 to F9.
- An orange horizontal rectangle labeled **B13:F13** covers cells from B13 to F13.

Abbildung 2.3: Mögliche Bereichsformen und ihre Schreibweisen

Die Abbildung 2.3 zeigt folgende Regeln für die Bereichsbezug-Schreibweisen:

- ▶ Bei einem Bereich, der mehrere Spalten und Zeilen umfasst, wird die Zelle der linken oberen Ecke mit der Zelle der rechten unteren Ecke zu einem Bereich verbunden; z.B. *B3:D10*
- ▶ Bei einem Bereich, der in einer Zeile liegt, wird die linke Zelle mit der rechten Zelle zu einem Bereich verbunden; z.B. *B13:F13*
- ▶ Bei einem Bereich, der in einer Spalte liegt, wird die oberste Zelle mit der untersten Zelle zu einem Bereich verbunden; z.B. *F2:F9*

Bezüge auf ganze Spalten oder Zeilen

Wenn Sie sich auf Spalten oder Zeilen – in beiden Fällen von der ersten bis zur letzten Zelle – beziehen wollen, können Sie dies mit den Bezügen aus Tabelle 2.5 tun. Bezüge für andere Spalten oder Zeilen sind analog zu den Angaben in der Tabelle herzustellen.

Bezug auf ...	Eingabe
komplette Spalte E	<i>E:E</i>
komplette Zeile 7	<i>7:7</i>
alle Zeilen von Zeile 2 bis Zeile 5	<i>2:5</i>
die gesamte Tabelle	<i>A:XFD</i> oder <i>1:1048576</i>

Tabelle 2.5
Schreibweisen
für Bezüge auf
ganze Zeilen
oder Spalten

Letztgenannte Bezüge können erst ab Excel 2007 funktionieren, da hier die entsprechenden Spalten- und Zeilennummern zur Verfügung stehen.

Hinweis

Die Angaben für die gesamte Tabelle kommen bei Bezügen auf andere Blätter in Frage. In einer Formel auf demselben Blatt würden Sie durch Einbeziehung der eigenen Formelzelle einen Zirkelbezug erzeugen. Mehr dazu erfahren Sie im Abschnitt »Was ist ein Zirkelbezug?« ab Seite 94.

Der Verbindungsoperator

Durch den Verbindungs- oder Vereinigungsoperator können Sie Zellen, die nicht nebeneinander liegen, einer Funktion zur Berechnung übergeben. Der Verbindungsoperator ist das Semikolon (;). Wenn mehrere Zellen einer Funktion mit dem Verbindungsoperator übergeben werden, zählt jeder Zellbezug, der durch Semikolon eingegrenzt ist, als eigenständiges Argument.

Wären die drei Bereiche in Abbildung 2.3 zu summieren, müsste jede Zellengruppe in der SUMME()-Funktion einzeln angegeben werden. Die Funktion hätte drei Argumente, jeweils getrennt durch ein Semikolon:

```
=SUMME(B3:D10;F2:F9;B13:F13)
```

Der Schnittmengenoperator

Der Schnittmengenoperator wird selten benutzt, darf jedoch in der vollständigen Aufzählung der Bezugsoperatoren nicht fehlen. Durch den Schnittmengenoperator – das Leerzeichen – lässt sich ein Bezug auf die Zellen herstellen, die mehreren unterschiedlichen Bezügen gemeinsam sind. Anders ausgedrückt: Unter der Schnittmenge sind in Excel die Werte zu verstehen, die in der Fläche liegen, in der sich mehrere Bereiche überschneiden.

	A	B	C	D	E	F	G	H
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								

Abbildung 2.4: Einfaches Beispiel einer Schnittmenge

Die in Abbildung 2.4 dargestellte Schnittmenge wird aus den Bereichen B2:E11 und C8:G14 gebildet. In einer Formel oder Funktion würde diese Schnittmenge – mit dem Leerzeichen – wie folgt dargestellt werden: B2:E11 C8:G14

Das Ergebnis dieser Schnittmenge wird als explizite Schnittmenge bezeichnet. Wenn diese explizite Schnittmenge beispielsweise summiert werden soll, ergibt sich folgende Formel:

```
=SUMME(B2:E11 C8:G14)
```

Die Anwendung von Schnittmengen findet meist im Zusammenhang mit Bereichsnamen statt, wogegen die Anwendung mit reinen Zellbezügen seltener ist.

Bezüge in Formeln

Wie zu Beginn dieses Kapitels bereits erwähnt, entspricht die Eingabe von Konstanten (Zahlenwerten) in Formeln nicht dem Sinn und Zweck einer Tabellenkalkulation. Bei dieser Methode müssten Sie bei jeder Änderung eines Werts auch die Formel entsprechend ändern.

Den Inhalt einer Zelle adressiert man über deren Zellbezug. Durch die Bezüge erkennt Excel, aus welchen Zellen die in einer Formel verwendeten Werte zu entnehmen sind.

	A	B	C	D	E
1					
2		Tagespauschale	Tage	Gesamt	
3		66 €	5	330 €	
4					

Abbildung 2.5: Für die Multiplikation von Pauschale und Tage werden die Zellbezüge verwendet

Grundsätzlich können Sie Bezüge über die Tastatur eingeben. Dabei dürfen keine Leerzeichen in die Formel gelangen. Auf die Groß- oder Kleinschreibung kommt es nicht an. Allerdings besteht beim Schreiben immer eine gewisse Fehlergefahr, da Sie sich bei den Bezügen irren oder vertippen könnten.

Alle Bezüge in Kleinbuchstaben werden nach Abschluss der Formeleingabe durch Excel in Großbuchstaben umgewandelt, wenn die Formel syntaktisch richtig eingegeben wurde. Ist dies nicht der Fall, überprüfen Sie bitte die Eingabe auf Syntax- und Schreibfehler.

Hinweis

Wenn Sie Zellbezüge einfacher und bequemer eingeben und dabei noch Fehler vermeiden wollen, bedienen Sie sich der sogenannten Zeigemethode. Durch die Zeigemethode können Sie Zellbezüge schnell und nahezu fehlerfrei eingeben. Verfahren Sie bei der Formeleingabe wie folgt:

1. Geben Sie in einer Formel zuerst das Gleichheitszeichen ein.
2. Geben Sie den folgenden Operator, den Funktionsnamen, die Klammer oder andere einleitende Formelbestandteile ein.
3. Markieren Sie die Zelle oder den Zellbereich, deren bzw. dessen Zellbezug in die Formel eingehen soll. Die Markierung wird durch eine gestrichelte Linie, den sogenannten Laufrahmen, gekennzeichnet und der Zellbezug erscheint korrekt in der Formel.
4. Geben Sie den folgenden Operator, den Funktionsnamen, die Klammer oder andere folgende Formelbestandteile ein.
5. Zeigen Sie durch Markieren auf weitere, beteiligte Zellen oder Bereiche.
6. Fahren Sie mit den Schritten 2 bis 5 bis zum Ende der Formeleingabe fort.

Durch den Einsatz von Zellbezügen wird es Ihnen erst möglich, alle Vorteile einer Tabellenkalkulation auszuschöpfen. Sie sind in der Lage, beliebige Rechenmodelle zu erstellen, in denen nur noch die Eingabewerte verändert werden müssen, um zu neuen Ergebnissen zu kommen.

Für Ihre praktische Arbeit bedeutet dies, dass Sie ein Rechenmodell nur ein einziges Mal erstellen müssen, um es anschließend für gleichartige Aufgaben ständig benutzen zu können. In der Tabelle aus Abbildung 2.6 können Sie die Eingabewerte der beiden Spalten *B* und *C* ändern, während in der Spalte *D* sofort das Produkt gebildet wird.

	A	B	C	D	E
1					
2		Tagespauschale	Tage	Gesamt	
3		66 €	5	330 €	
4		55 €	2	110 €	
5		45 €	3	135 €	
6		75 €	4	300 €	
7		60 €	5	300 €	
8		50 €	6	300 €	
9		70 €	7	490 €	
10					

Abbildung 2.6: Die Formel mit Zellbezügen kann nach unten kopiert werden

Ein weiterer Vorteil besteht darin, dass die gezeigte Formel durch Ziehen am Ausfüllkästchen nach unten kopiert werden kann. Damit erhalten Sie schnell alle benötigten Berechnungen, was bei Verwendung von Konstanten nicht der Fall wäre.



Die Tabelle aus Abbildung 2.6 finden Sie im Arbeitsblatt *Relativer Bezug* in der Arbeitsmappe *Kap02.xlsx*. Sie finden diese im Ordner *MS5-235\Kap02*.

Relative Bezüge

Wenn Sie durch Ziehen am Ausfüllkästchen die Formel in Abbildung 2.6 nach unten kopieren, erhalten Sie das für jede Zeile korrekte Berechnungsergebnis. Der Grund: Excel passt beim Kopieren die Bezüge in der Formel der jeweiligen Zeile an, indem es die aktuelle Zeilennummer einsetzt. Dies funktioniert jedoch nur, weil die Zeilennummer in der Formel »relativ« ist.

Einen relativen Bezug können Sie als ein räumliches Verhältnis zwischen zwei Zellen ansehen. Die Betrachtung des räumlichen Verhältnisses erfolgt dabei immer von der Formelzelle aus. Dieses räumliche Verhältnis – und nicht die buchstabengetreue Schreibweise eines Bezugs – wird beim Kopieren übertragen.

Die Formel in der Zelle *D3* der in Abbildung 2.6 gezeigten Tabelle liest sich vom Standort *D3* aus betrachtet so: »Nimm den Wert aus der Zelle zwei Spalten weiter links, gleiche Zeile (=B3) und multipliziere (=B3*) mit dem Wert aus der Zelle eine Spalte weiter links, gleiche Zeile (=B3*C3)«.

Wichtig

Grundsätzlich kann zur Veränderung der relativen Zellbezüge beim Kopieren gesagt werden:

- ▶ Beim horizontalen Kopieren verändern sich die Spaltenbezüge im Verhältnis zum jeweiligen Standort
- ▶ Beim vertikalen Kopieren verändern sich die Zeilenbezüge im Verhältnis zum jeweiligen Standort

Absolute Bezüge

Hier haben wir das Gegenstück zum relativen Bezug. Unter einem absoluten Bezug wird ein Zellbezug verstanden, der sich beim Kopieren oder Ausfüllen nicht verändert. Sie benötigen derartige Bezüge, um stets auf eine ganz bestimmte Zelle zu verweisen und mit dem dort vorhandenen Wert zu rechnen.

Nehmen wir an, Sie möchten in einer Tabelle den Mehrwertsteuerbetrag zu einer Liste von Nettopreisen berechnen, wobei der anzuwendende Mehrwertsteuersatz zentral in einer Zelle gepflegt wird. Die einmal entwickelte Formel soll kopierbar sein.

	A	B	C	D
1		MWSt.-Satz	19%	
2				
3		Nettopreis	MWSt.-Betrag	Gesamt
4		229,00 €	=B4*C1	
5		45,65 €		
6		109,89 €		
7		299,00 €		
8		19,99 €		

Abbildung 2.7: Ausgangssituation mit Formel in Zelle C4

Die eingegebene Formel in der Zelle C4 berechnet einwandfrei die für den in B4 aufgeführten Nettopreis anzusetzende Mehrwertsteuer. Die Formel lautet:

=B4*C1

Wenn Sie diese Formel nun in der Spalte nach unten ausfüllen, also kopieren, erhalten Sie durch das Verändern der relativen Zellbezüge völlig unrealistische Ergebnisse. In der Zelle C6 erscheint gar die Fehlermeldung #WERT!. In der Zelle C3 ist der Text »MWSt.-Betrag« enthalten. Dies führt unweigerlich zur Fehlermeldung #WERT!.

Wenn Sie den Auftrag, der in der Formel der Zelle C4 steckt, analysieren, kommen Sie zu dem Schluss, dass der Bezug auf die Mehrwertsteuer unveränderbar sein muss. Diese Unveränderbarkeit wird erreicht, wenn Sie vor den Spalten- (C) und den Zeilenbezug (1) ein Dollarzeichen (\$) setzen. Ein derartiger Zellbezug wird beim Kopieren oder Ausfüllen nicht verändert und in Excel als absoluter Bezug bezeichnet.

In der Beispieltabelle der Abbildung 2.7 muss demnach die Formel in der Zelle C4 wie folgt lauten:

=B4*\$C\$1

Wenn Sie mit dieser Formel, ausgehend von der Zelle C4, den Bereich ausfüllen und diese Formel nach unten kopieren, erhalten Sie in der gesamten Tabelle jeweils das korrekte Ergebnis (Abbildung 2.8).

	A	B	C	D
1		MWSt.-Satz	19%	
2				
3		Nettopreis	MWSt.-Betrag	Gesamt
4		229,00 €	43,51 €	272,51 €
5		45,65 €	8,67 €	54,32 €
6		109,89 €	=B6*\$C\$1	130,77 €
7		299,00 €	56,81 €	355,81 €
8		19,99 €	3,80 €	23,79 €

Abbildung 2.8: Korrekte Ergebnisse durch den absoluten Zellbezug auf die Zelle \$C\$1

Das fertige Beispiel aus Abbildung 2.8 finden Sie im Arbeitsblatt *Absoluter Bezug* in der Arbeitsmappe *Kap02.xlsx*. Diese befindet sich im Ordner *MS5-235\Kap02*.



Gemischte Bezüge

Streng genommen hätte es keines absoluten Bezugs bedurft, um im Beispiel in Abbildung 2.8 die Formel korrekt zu kopieren. Da nur nach unten, d.h. in derselben Spalte kopiert wurde, hätte der Spaltenbezug nicht absolut sein müssen. Diese Überlegung führt uns zu den gemischten Bezügen, die – wie der Name es sagt – eine Mischform zwischen relativem und absolutem Bezug darstellt.

Angenommen, Sie wollen eine Tabelle erstellen, aus der die Werte für das Einmaleins abgelesen werden können. Die Rechenmethode ist im Prinzip klar. In der Bearbeitungsleiste der Tabelle in Abbildung 2.9 sehen Sie die Formel für die Zelle F6: =A6*F2. In B3 müsste sie entsprechend lauten: =A3*B2. Keine der Formeln ist jedoch sinnvoll kopierbar. Sicher ahnen Sie schon, dass es nicht nötig ist, die Formel für jede Zelle neu zu entwickeln.

	A	B	C	D	E	F	G	H	I	J	K
	Kleines Einmaleins										
1											
2		1	2	3	4	5	6	7	8	9	
3	1										
4	2										
5	3										
6	4					20					
7	5										
8	6										
9	7										
10	8										
11	9										
12											

Abbildung 2.9: Tabelle zur Darstellung des kleinen Einmaleins

Wenn es Ihnen gelingen würde, wahlweise den Spalten- oder den Zeilenbezug zu sperren, könnten Sie in der Tabelle der Abbildung 2.9 eine einmal in B3 entwickelte Formel sowohl nach unten als auch nach rechts kopieren. Die folgenden Überlegungen führen zur Lösung:

- ▶ Sie müssen den Spaltenbezug für die Werte aus Spalte A sperren, um die Formel horizontal kopieren zu können und trotzdem die Werte immer aus der Spalte A einzulesen. Der Zeilenbezug für die Werte aus Spalte A muss relativ bleiben, damit er beim vertikalen Kopieren angepasst wird. Es ergibt sich der Zellbezug »\$A3«.
- ▶ Für den Bezug auf die Werte in Zeile 2 verhält es genau umgekehrt: Der Spaltenbezug muss relativ bleiben, während der Zeilenbezug durch das \$-Zeichen gesperrt werden muss. Damit bleibt der Bezug auf die Zeile beim vertikalen Kopieren unverändert, während beim horizontalen Kopieren die Spaltenbezüge angepasst werden. Aus dieser Überlegung ergibt sich die Schreibweise »B\$2«.

Somit ergibt sich bei der Beispieltabelle für die Zelle B3 folgende Formel:

= \$A3*B\$2

Probieren Sie dies in Ihrer Beispieltabelle aus. Sie werden immer die korrekten Ergebnisse erzielen.

	A	B	C	D	E	F	G	H	I	J	K
1	Kleines Einmaleins										
2		1	2	3	4	5	6	7	8	9	
3	1	1	2	3	4	5	6	7	8	9	
4	2	2	4	6	8	10	12	14	16	18	
5	3	3	6	9	12	15	18	21	24	27	
6	4	4	8	12	16	20	24	28	32	36	
7	5	5	10	15	20	25	30	35	40	45	
8	6	6	12	18	24	30	36	42	48	54	
9	7	7	14	21	28	35	42	49	56	63	
10	8	8	16	24	32	40	48	56	64	72	
11	9	9	18	27	36	45	54	63	72	81	
12											

Abbildung 2.10: Korrekt rechnende Tabelle mit gemischten Bezügen

Das fertige Beispiel aus Abbildung 2.10 finden Sie im Arbeitsblatt *Gemischter Bezug* in der Arbeitsmappe *Kap02.xlsx*. Diese befindet sich im Ordner *\Ms5-235\Kap02*.



Als elementaren Grundsatz leiten Sie ab: Bei einem gemischten Bezug ist der Bezugssteil (Spalten- oder Zeilenbezug), vor dem das Dollarzeichen steht, gesperrt und beim Kopieren unveränderbar. Vereinfacht können Sie sich merken:

- ▶ Möchten Sie immer Werte aus einer bestimmten Spalte übernehmen, muss der Spaltenbezug durch das Dollarzeichen gesperrt werden
- ▶ Möchten Sie immer Werte aus einer bestimmten Zeile übernehmen, muss der Zeilenbezug durch das Dollarzeichen gesperrt werden

Mit der Taste **[F4]** können Sie die Bezugsart ändern. Markieren Sie den Zellbezug in einer Formel und drücken Sie mehrmals die **[F4]**-Taste. Mit jedem Tastendruck ändert sich die Schreibweise (Relativ – Absolut – Gemischt – Relativ usw.).

Tip

Zuletzt soll hier noch auf ein Feature ab Excel 2007 hingewiesen werden: In als »Tabelle« gekennzeichneten Blattbereichen finden Sie eine neue Schreibweise für (relative) Bezüge. Beispiel:

Hinweis

`=Tabelle1[[#Diese Zeile];[Kapital]]+Tabelle1[[#Diese Zeile];[Gewinnanteil]]`

Im Beispiel werden je Zeile die Werte aus den Spalten mit den Überschriften »Kapital« und »Gewinnanteil« addiert. Bei Eingabe einer Formel schreibt Excel für die markierten Bezüge in der gezeigten Art und Weise mit und füllt selbständig die angrenzenden leeren Zellen mit der neuen Formel aus – ein sehr effektives Arbeiten.

Was ist ein Zirkelbezug?

Zirkelbezüge entstehen meist durch Eingabefehler. Man versteht darunter einen Bezug auf die Zelle, in der sich die Formel selbst befindet – quasi einen Bezug auf sich selbst. Formeln mit einem Zirkelbezug kann Excel standardmäßig nicht lösen. Sie erhalten stattdessen eine Meldung, wie in Abbildung 2.11.

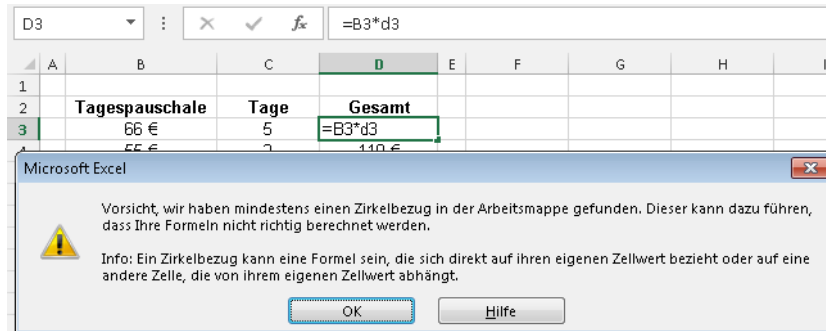


Abbildung 2.11: Fehlerhafte Eingabe in der Zelle D3 mit der Fehlermeldung von Excel 2013

Excel 2007 und 2010 warteten in der Fehlermeldung noch mit etwas umfangreicheren Erklärungen zum Zirkelbezug auf (Abbildung 2.12).

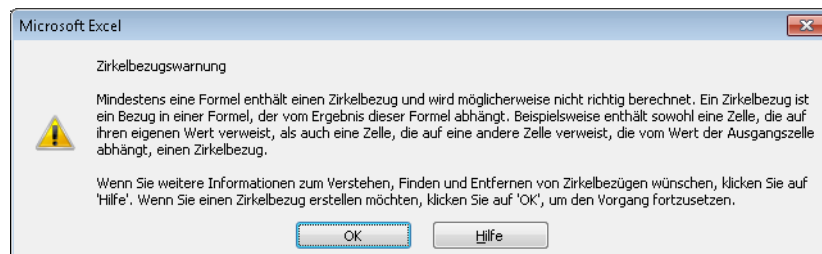


Abbildung 2.12: Die *Zirkelverweis*-Fehlermeldung von Excel 2010

Hinweis Wenn Sie in der Zirkelbezugsfehlermeldung auf die Schaltfläche *OK* klickten, wurde im Tabellenfenster von Excel 2003 die *Zirkelverweis*-Symbolleiste eingeblendet. Ab Excel 2007 können Sie das *Zirkelbezüge*-Symbol im Dropdownmenü *Fehlerüberprüfung* auf der Registerkarte *FORMELN* verwenden, um die Zellen innerhalb des Zirkelbezugs zu durchlaufen. Nutzen Sie dabei die Schaltflächen *Spur zum Nachfolger* und *Spur zum Vorgänger*, um die Gründe des aufgetretenen Problems durch Pfeile anzeigen zu lassen (Abbildung 2.14).

Seit Excel 2007 erscheint nach dem Klick auf *Hilfe* in der Zirkelbezugswarnung (Abbildung 2.11) ein Hilfenfenster mit Anleitungen, wie mit Zirkelbezügen zu verfahren ist (Abbildung 2.13).

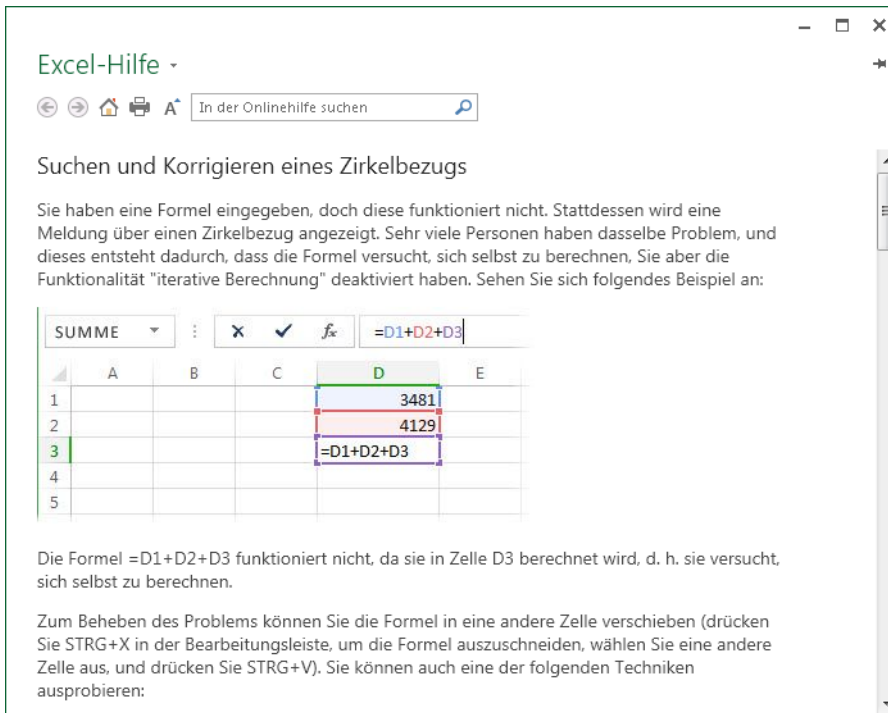


Abbildung 2.13: Seit Excel 2007 steht umfangreiche Hilfe bei Zirkelbezügen zur Verfügung; hier die Hilfe von Excel 2013

Wie schon im Hinweis oben erwähnt, ist das Suchen und Entfernen von Zirkelbezügen ab Excel 2007 auf der Registerkarte *FORMELN* zu finden. In der Gruppe *Formelüberwachung* klicken Sie auf die Schaltfläche *Fehlerüberprüfung*, dann auf *Zirkelbezüge* (Excel 2007/2010 *Zirkelverweise*) und dann auf die angezeigten Bezüge (Abbildung 2.14). Allerdings wird hier (sowie links unten in der Statusleiste) immer nur der letzte eingegebene Zirkelbezug angezeigt.

Wenn man die Warnung sowie das Hilfefenster schließt, ohne sofort den Zirkelbezug zu korrigieren, wird bei einer erneuten Eingabe einer Formel mit Zirkelbezug in dieselbe oder eine andere Zelle keine erneute Warnung sowie Hilfefenster angezeigt. Lediglich in der Statusleiste erfolgt rechts neben dem Status *BEREIT* ein leicht übersehbarer Hinweis auf den Zirkelbezug in der Form *Zirkelbezüge: D3*. Korrigieren Sie Zirkelbezüge also unbedingt sofort nach dem Warnhinweis, da Sie die Hinweise darauf später leicht vergessen und übersehen können.

Wichtig

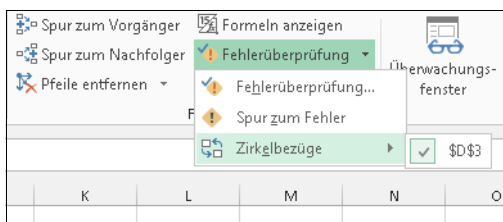


Abbildung 2.14: Zirkelbezüge auffinden in Excel 2013

Hinweis Excel kann nicht automatisch alle geöffneten Arbeitsmappen berechnen, wenn eine von ihnen einen Zirkelbezug enthält. Sie können einen Zirkelbezug entfernen oder jede Zelle, die in den Zirkelbezug einbezogen ist, einmal berechnen, indem Sie die Ergebnisse der vorherigen Iteration verwenden. Sofern Sie die Standardeinstellungen für Iterationen nicht ändern, beendet Excel die Berechnung nach 100 Iterationsschritten oder wenn sich alle Werte in dem Zirkelbezug zwischen zwei Iterationen um einen Betrag von weniger als 0,001 ändern, je nachdem, welcher Fall zuerst eintritt.

Matrixformeln

Um eine Matrixformel verstehen zu können, betrachten Sie zunächst die Arbeitsweise einer normalen Einzelwertformel. Eine Einzelwertformel erzeugt ein Ergebnis aus mehreren Operanden. Beispielsweise erzeugt die Formel $=B4*C4$ das Produkt der beiden angegebenen Zellen. Vergleichen Sie hierzu die Abbildung 2.15: Die Matrixformel $\{=F4:F8*G4:G8\}$ erzeugt eine Menge von insgesamt fünf Ergebnissen, in diesem Fall jedes Mal das Produkt der beiden nebeneinander stehenden Zellen in den links und rechts vom Multiplikationszeichen stehenden Bereichen.

	A	B	C	D	E	F	G
2	Einzelwertformeln				Matrixformeln		
3	Tagespauschale	Tage	Gesamt		Tagespauschale	Tage	Gesamt
4	55 €	5	=A4*B4		55 €	5	{=E4:E8*F4:F8}
5	50 €	5	=A5*B5		50 €	5	{=E4:E8*F4:F8}
6	45 €	4	=A6*B6		45 €	4	{=E4:E8*F4:F8}
7	60 €	3	=A7*B7		60 €	3	{=E4:E8*F4:F8}
8	63 €	7	=A8*B8		63 €	7	{=E4:E8*F4:F8}

Abbildung 2.15: Gegenüberstellung einer Einzelwert- und Matrixformel



Das fertige Beispiel aus Abbildung 2.15 finden Sie im Arbeitsblatt *Matrixformeln* in der Arbeitsmappe *Kap02.xlsx*. Diese befindet sich im Ordner `\Ms5-235\Kap02`.

Bei der Einzelwertformel werden die Formeln durch relative Bezüge gebildet, die sich durch Ausfüllen in der Spalte von Zeile zu Zeile verändern. Dagegen ist die Matrixformel im gesamten Bereich (Matrix) unverändert. Dennoch werden von Zeile zu Zeile unterschiedliche Ergebnisse gebildet.

Da eine Matrixformel mehrere Ergebnisse liefern kann, können Sie – bei wiederholter Eingabe gleicher Formeln – Zeit sparen. Generell ist jedoch festzuhalten, dass eine Matrixformel mehr Speicher als die entsprechenden Einzelwertformeln belegt.

Wichtig

Eine Matrixformel erkennen Sie an den geschweiften Klammern {...}, die die Formel einschließen. Diese geschweiften Klammern setzt Excel automatisch um Matrixformeln und können (bzw. dürfen) nicht manuell eingefügt werden. Um eine Formel als Matrixformel einzugeben, müssen Sie die Eingabe mit der Tastenkombination **[Strg] + [↕] + [↵]** beenden.

Zwischen der Bearbeitung einer Einzelwertformel und dem Bearbeiten von Matrixformeln gibt es gewichtige Unterschiede. Beim Bearbeiten einer Matrixformel ist immer die **gesamte** Matrixformel betroffen und damit auch der gesamte Bereich. Dies bedeutet, dass Sie nicht einen Teil oder gar nur eine Zelle ändern können. Folgende Operationen sind nicht möglich:

- ▶ Abändern des Inhalts einer einzigen Zelle
- ▶ Löschen oder Verschieben von Zellen, die Teil einer Matrix sind
- ▶ Einfügen von Zellen, Zeilen oder Spalten in eine bestehende Matrix

Sollten Sie dennoch versuchen, eine dieser Operationen durchzuführen, erhalten Sie eine Fehlermeldung angezeigt.

Wenn Sie nicht genau wissen, wo der Bereich einer Matrixformel beginnt bzw. endet, gehen Sie wie folgt vor:

Tip

1. Markieren Sie eine Zelle, die der Matrix angehört.
2. Rufen Sie das Dialogfeld *Gehe zu* mit der Taste **[F5]** auf.
3. Klicken Sie im Dialogfeld *Gehe zu* auf die Schaltfläche *Inhalte*.
4. Im daraufhin geöffneten Dialogfeld wählen Sie die Option *Aktuelles Array* aus und bestätigen mit *OK*. Hierdurch wird die Matrix markiert, in der sich die markierte Zelle befindet.

Tipps & Tricks

In diesem Abschnitt wollen wir Ihnen in kurzer Form unsere wichtigsten Tipps und Tricks bei der Arbeit mit Formeln vermitteln. Je nachdem, wie intensiv Sie mit Excel arbeiten, werden Sie die im Folgenden beschriebenen Verfahren mehr oder weniger häufig benötigen.

Formelansicht ein/aus

Wenn Sie sich in einer Tabelle, die Sie nicht selbst erstellt haben, eine Übersicht verschaffen wollen, z.B. weil die Tabelle sehr viele Formeln und Verknüpfungen enthält, erleichtern Sie sich die Arbeit, indem Sie anstelle der Formelergebnisse die Formeln und die Verknüpfungen selbst anzeigen. In Excel 2007 ist das wie bei den Vorgängerversionen schnell erledigt:

1. Aktivieren bzw. öffnen Sie das betreffende Arbeitsblatt.
2. Drücken Sie die Tastenkombination **[Strg] + [#]**.

Sie sehen, dass Excel nun nicht die Formelergebnisse, sondern die dahinter stehenden Formeln anzeigt. Da die Formeln mehr Platz beanspruchen, passt Excel automatisch die Spaltenbreite an.

Seit Excel 2010 funktioniert die Tastenkombination **[Strg] + [#]** leider nicht mehr. Sie müssen die Anzeige der Formeln auf der Registerkarte *FORMELN* in der Gruppe *Formelüberwachung* auf die Schaltfläche *Formeln anzeigen* umschalten.

Hinweis



Das Ein- und Ausschalten der Formelanzeige ändert keineswegs die Daten in Ihrer Tabelle. Der Befehl wirkt sich nur auf die Ansicht aus und kann durch erneutes Umschalten jederzeit wieder zurückgenommen werden.

Der Vollständigkeit halber sei noch erwähnt, dass das oben beschriebene Umschalten einer Änderung in den Excel-Optionen entspricht: In Excel 2007 klicken Sie auf die *Office*-Schaltfläche oben links, und anschließend auf die Schaltfläche *Excel-Optionen*; ab Excel 2010 klicken Sie im Menüband auf die

Registerkarte *DATEI* und dann auf die Schaltfläche *Optionen*. Im Dialogfeld *Excel-Optionen* finden Sie in der Kategorie *Erweitert* das entsprechende Kontrollkästchen zum Ein- bzw. Ausschalten der Formelanzzeige im Blatt (Abbildung 2.16).

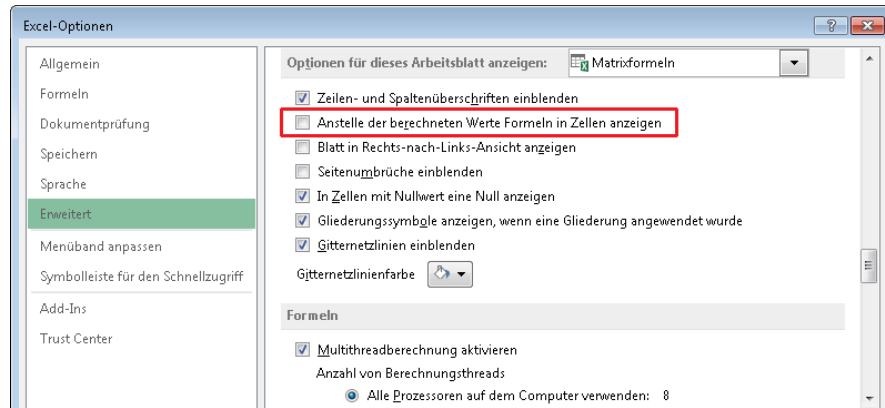


Abbildung 2.16: Die Arbeitsblatt-Optionen in Excel 2013

In allen Versionen gilt: Diese Einstellung gilt nur für das aktive Arbeitsblatt.

Formeln in mehrere Zellen eingeben

Wenn Sie bei der Formeleingabe bereits wissen, dass die Formel in mehrere Zellen einzugeben ist, können Sie die Formeleingabe in alle betreffenden Zellen in einem Arbeitsgang durchführen:

1. Klicken Sie auf die Zelle, in welche die Formel eingetragen werden soll, und erweitern Sie von hier aus die Markierung auf den Bereich, der die kopierte Formel enthalten soll. Es ist auch eine Mehrfachauswahl möglich.
2. Geben Sie die Formel in die aktive Zelle ein.
3. Schließen Sie die Formeleingabe mit der Tastenkombination **Strg** + **↵** ab.

Der zuvor markierte Bereich wird komplett mit der eingegebenen Formel ausgefüllt.

Formeln in mehrere Blätter eingeben

In Mappen mit gleichartig aufgebauten Arbeitsblättern können Sie Formeln (aber auch alle anderen Inhalte) gleichzeitig in mehrere Blätter eingeben. Sie müssen die betreffenden Blätter vorher nur markieren (gruppieren).

1. Markieren Sie alle betreffenden Blätter, indem Sie bei festgehaltener **Strg**-Taste die einzelnen Blattregister mit der Maus anklicken. Den Gruppierungsmodus erkennen Sie an der Anzeige *[Gruppe]* in der Titelleiste des Fensters.

Tipp

Handelt es sich um einen Block von aufeinander folgenden Blättern, können Sie nach dem Anklicken des ersten Blatts die **⇧**-Taste festhalten und mit der Maus das letzte Blatt dieses Blocks anklicken.

2. Geben Sie die Formel(n) in das aktive Blatt ein.
3. Heben Sie die Blattgruppierung auf, indem Sie ein einzelnes Register markieren oder im Kontextmenü (erreichbar über einen Klick mit der rechten Maustaste auf ein markiertes Register) für die Blattregister den Befehl *Gruppierung aufheben* wählen.

Die Formel(n) wurden quasi wie mit einem Durchschlag in alle markierten Blätter geschrieben. Wenn nötig, kombinieren Sie einfach die oben gezeigte Formeleingabe in mehrere Zellen mit der Eingabe in mehrere Blätter.

Formelzellen schnell markieren

Excel markiert alle Zellen, welche Formeln enthalten, wenn Sie diese Schritte befolgen:

1. Wählen Sie die Taste **[F5]** (oder **[Strg] + [G]**).
2. Klicken Sie auf die Schaltfläche *Inhalte*.
3. Wählen Sie im Dialogfeld *Inhalte auswählen* die Option *Formeln*. Sie können dabei noch anhand der Formelergebnisse unterscheiden, welche Formeln zu markieren sind.
4. Klicken Sie auf die Schaltfläche *OK*.

Wenn Sie jetzt alle markierten Formelzellen einzeln durcharbeiten müssen, benutzen Sie die **[↵]**-Taste, um von einer markierten Zelle zur nächsten zu wechseln. In der Bearbeitungsleiste oben können Sie die jeweilige Formel bearbeiten. Mit **[↔] + [↵]** bewegen Sie sich in die entgegengesetzte Richtung.

Wollen Sie die Markierung aufheben, aber trotzdem alle Formeln durcharbeiten, weisen Sie den markierten Zellen eine Signalfarbe zu, die Sie später wieder entfernen können.

Auf welche Zellen bezieht sich die Formel?

Wollen Sie schnell einmal sehen, auf welche Zellen sich eine markierte Formel bezieht, drücken Sie die **[F2]**-Taste. Sie öffnet die Zelle zum Bearbeiten¹ und markiert die beteiligten Formelinhalte, also auch die Zellbezüge verschiedenfarbig (Abbildung 2.17).

	A	B	C	D
2				
3		Eingaben		
4	Jährliche Verzinsung	6%		
5	Geplante Laufzeit des Sparplans in Jahren	10		Monatlich zu sparender Betrag für eine Zielsparsumme von 50.000 € nach 10 Jahren
6	anzusparender Betrag	50.000 €		=RMZ(B4/12;B5*12;0;B6)
7				

Abbildung 2.17: Das Öffnen der Zelle mit der **[F2]**-Taste markiert alle Bezüge verschiedenfarbig in Formel und Blatt

¹ Je nach Einstellung in den *Excel-Optionen*, Kategorie *Erweitert*, Abschnitt *Bearbeitungsoptionen*, Kontrollkästchen *Direkte Zellbearbeitung zulassen* wird entweder die Zelle selbst oder die Bearbeitungsleiste zum Ändern des Zellinhalts geöffnet.

Formeln kopieren/verschieben

Das Eingeben jeder einzelnen Formel führt bei größeren Tabellen zu einem sehr hohen Arbeitsaufwand. Besser ist es, wenn Sie sich Arbeitstechniken erschließen, bei denen Sie den Aufwand durch Kopieren der Formeln deutlich verringern können. Ebenso verhält es sich bei notwendigen Korrekturen, wo Sie durch Verschieben von Formeln und Werten Ihre Tabellen schnell neu gestalten können.

Grundsätzlich müssen zu kopierende oder zu verschiebende Bereiche vorher markiert werden. Im Folgenden erhalten Sie weitergehende Hinweise zum Markieren.

Wie in allen Windows-Programmen können Sie das Markieren mit zwei grundsätzlich unterschiedlichen Techniken durchführen:

- ▶ Markieren mit der Tastatur oder einem Befehl
- ▶ Markieren mit der Maus

Markieren mit Tasten

Die einfachste und bequemste Art des Markierens wird Ihnen durch die Maus angeboten. Dennoch kann es bisweilen sinnvoll sein, Markierungen mit den Tasten bzw. mit Tastenkombinationen durchzuführen. In der Tabelle 2.6 finden Sie nützliche Tastenkombinationen zum Markieren.

Tabelle 2.6
Markieren mit
der Tastatur

Markierung	Tastenkombination
Aktuelle Zeile	⇧ + Leertaste
Aktuelle Spalte	Strg + Leertaste
Aktueller Block eingegebener Daten	Strg + ⇧ + + (d.h. Strg + *)
Von aktiver Zelle in gewünschte Richtung	⇧ + Pfeiltasten
Von aktiver Zelle zum Ende des Datenblocks	⇧ + Strg + Ende
Von aktiver Zelle zum Anfang des Datenblocks	⇧ + Strg + Pos1
Gesamtes Arbeitsblatt	Strg + ⇧ + Leertaste oder Strg + A

Wenn Sie – von der aktiven Zelle ausgehend – die Markierung erweitern wollen, können Sie dies auch folgendermaßen erreichen: Drücken Sie die Taste **F8**. Dadurch gelangen Sie in den sogenannten »Erweiterungsmodus«, was in der Statusleiste durch das Hervorheben der Buchstaben *ERW* bzw. der Angabe *AUSWAHL ERWEITERN* angezeigt wird. Jetzt können Sie mit den Pfeiltasten die Markierung bequem in alle Richtungen erweitern. Um den Erweiterungsmodus abzuschalten, drücken Sie erneut die Taste **F8** oder die **Esc**-Taste.

Markieren mit der Maus

Mit der Maus können Sie Zellbereiche durch Ziehen markieren. Unter dem Ziehen ist das Bewegen der Maus bei gedrückter linker Maustaste zu verstehen. Achten Sie darauf, dass Sie nebenstehendes Maussymbol zum Markieren haben. Die Tabelle 2.7 beschreibt nützliche Markiertechniken mithilfe der Maus.

Markierung	Aktion
Eine einzelne Zelle	Klicken Sie auf diese Zelle
Eine Spalte	Klicken Sie auf den entsprechenden Spaltenkopf
Mehrere Spalten	Ziehen Sie den Mauszeiger mit gedrückter linker Maustaste über die betreffenden Spaltenköpfe
Eine Zeile	Klicken Sie auf den entsprechenden Zeilenkopf
Mehrere Zeilen	Ziehen Sie den Mauszeiger mit gedrückter linker Maustaste über die betreffenden Zeilenköpfe
Einen zusammenhängenden Zellbereich	Ziehen Sie den Mauszeiger mit gedrückter linker Maustaste von der ersten Zelle diagonal zur letzten Zelle
Nicht zusammenhängende Einzelzellen oder Zellbereiche (Mehrfachauswahl)	Halten Sie die [Strg] -Taste gedrückt, während Sie auf weitere zu markierende Zellen klicken oder weitere Zellbereiche/Spalten/Zeilen markieren
Alle Zellen in einem Arbeitsblatt ^a	Klicken Sie auf die Fläche, die sich im Schnittpunkt der Spalten- und Zeilenköpfe befindet (linke obere Ecke des Arbeitsblattfensters)

Tabelle 2.7
Markieren mit der Maus

^a Das Markieren der gesamten Tabelle benötigen Sie, um ein Format generell in der Tabelle zu ändern (z.B. die Schriftart).

Innerhalb eines markierten Zellbereichs können Sie sich nach unten oder nach rechts durch Drücken der -Taste bewegen. Wenn Sie sich in umgekehrter Richtung bewegen wollen, nach oben oder nach links, drücken Sie die Tastenkombination + .

Hinweis

Formelzellen verschieben

Unter dem Verschieben ist das Ausschneiden, d.h. Löschen der Formel in der Ursprungszelle und das anschließende Einfügen in eine andere Zelle zu verstehen. Dazu stehen Ihnen zwei Techniken zur Verfügung:

- ▶ Die Schaltflächen in der Gruppe *Zwischenablage* auf der Registerkarte *START* bzw. im Kontextmenü nach einem Rechtsklick auf den markierten Zellbereich
- ▶ Das direkte Verschieben mithilfe der Maus

Verschieben mithilfe der Zwischenablage

So können Sie Zellen, die Formeln beinhalten, innerhalb einer Tabelle an einen anderen Platz verschieben:

1. Markieren Sie die Zelle oder die Zellen, die verschoben werden sollen.
2. Klicken Sie auf der Registerkarte *START* in der Gruppe *Zwischenablage* auf die Schaltfläche *Ausschneiden* oder drücken Sie die Tastenkombination **[Strg] + [X]**.

3. Klicken Sie auf die Zelle, in welche die ausgeschnittene Zelle verschoben werden soll. Wenn Sie einen Zellbereich verschieben möchten, stellt diese Zelle die linke obere Zelle des neuen Bereichs dar.
4. Klicken Sie auf der Registerkarte *START* in der Gruppe *Zwischenablage* auf die Schaltfläche *Einfügen* oder drücken Sie die Tastenkombination **[Strg] + [V]**.

Wenn Sie die Zelle verschoben haben, untersuchen Sie, ob sich an der Formel oder dem Ergebnis etwas verändert hat. Das Ergebnis der Formel wird in der Tabelle angezeigt und ist mit dem zuvor erzielten Ergebnis identisch. Zur Kontrolle der Formel klicken Sie auf die verschobene Zelle und lesen den Zelleninhalt in der Bearbeitungsleiste ab. Auch hier ist alles gleich geblieben und entspricht dem Original.

Wichtig Daraus können Sie folgenden Grundsatz ableiten: Beim Verschieben behält der Zelleninhalt und somit auch die Formel den Originalzustand bei.

Verschieben mit der Maus

Sie möchten Formelzellen ausschließlich mit der Maus verschieben. Die Zwischenablage soll nicht beansprucht werden. Mit der Maus können Sie schnell Zellen an einen beliebigen Ort innerhalb der Tabelle verschieben. Gehen Sie so vor:

1. Markieren Sie die Zelle(n), die verschoben werden soll(en).
2. Bewegen Sie den Mauszeiger auf der Grenze der Markierung. Das Symbol wird zu einem weißen Pfeil mit schwarzem Pfeilkreuz an der Spitze (Abbildung 2.18).
3. Ziehen Sie die Zelle(n) mit gedrückter linker Maustaste an den gewünschten neuen Standort und lassen dort die Maustaste los.

Preis	Menge	Gesamt
22,55 €	4	90,20 €
33,50 €	3	100,50 €
17,56 €	5	87,80 €

Abbildung 2.18: Zum Verschieben muss der Mauszeiger als Pfeilkreuz dargestellt werden

Während des Ziehens wird der zu verschiebende Bereich als grauer Schattenrahmen dargestellt. In der Nähe dieses Rahmens wird durch ein QuickInfo-Feld der Zellbezug bzw. der Zellbereich angezeigt, in den der zu verschiebende Bereich abgelegt wird, wenn die Maustaste losgelassen wird.

Verschieben Sie die Zellen auf einen bereits mit Daten oder Formeln gefüllten Bereich, fragt Excel, ob die Inhalte des Zielbereiches überschrieben werden dürfen. Bei aufmerksamer Arbeit können also keine Daten durch ein versehentliches Verschieben verloren gehen.

Formelzellen kopieren

Beim Kopieren verbleiben die Daten an ihrem Ursprungsort und eine Kopie dieser Daten wird an einem anderen Ort eingetragen. Im Falle von kopierten Formeln gibt es jedoch einiges zu beachten, denn hier wird wichtig, ob die Formelbezüge relativ, absolut oder gemischt sind (siehe den Abschnitt »Bezüge in Formeln« ab Seite 89).

Auch beim Kopieren gibt es mehrere Methoden. Folgende Möglichkeiten bieten sich an:

- ▶ Die Befehle *Kopieren* und anschließend *Einfügen* auf der Registerkarte *START* in der Gruppe *Zwischenablage*
- ▶ Durch das direkte Kopieren mithilfe der Maus
- ▶ Durch das Ziehen am Ausfüllkästchen der Formelzelle. Das Ergebnis ist mit dem der ersten beiden Methoden identisch.

Kopieren über die Zwischenablage

Zum Kopieren mit den Schaltflächen auf der Registerkarte *START* verfahren Sie wie folgt:

1. Markieren Sie die Zelle(n), die kopiert werden soll(en).
2. Klicken Sie auf der Registerkarte *START* in der Gruppe *Zwischenablage* auf die Schaltfläche *Kopieren* oder drücken Sie die Tastenkombination **[Strg] + [C]**.
3. Klicken Sie auf die Zelle(n), in die die Daten eingefügt werden sollen. Wenn Sie einen Zellbereich kopieren, ist diese Zelle die obere linke Ecke des Bereichs.
4. Klicken Sie auf der Registerkarte *START* in der Gruppe *Zwischenablage* auf die Schaltfläche *Einfügen* oder drücken Sie die Tastenkombination **[Strg] + [V]**.

Um wiederholt die kopierten Zellen einzufügen, müssen Sie den Arbeitsschritt 4 wiederholen. Dies gilt auch bei Benutzung des Befehls *Ausschneiden* in der Gruppe *Zwischenablage* auf der Registerkarte *START* zum Verschieben.

Hinweis

Sie können das Einfügen aus der Zwischenablage beim Verschieben oder Kopieren auch mit der **[↵]**-Taste abschließen. In diesem Fall jedoch leert Excel die Zwischenablage, d.h., das Kopieren oder Verschieben ist damit abgeschlossen.

Kopieren mit der Maus

Nehmen wir an, Sie möchten Zellen ausschließlich unter Verwendung der Maus kopieren. Im Prinzip ähnelt das Vorgehen dem Verschieben mit der Maus – bis auf den Einsatz der **[Strg]**-Taste. Und so geht es:

1. Markieren Sie die Zelle(n), welche kopiert werden sollen.
2. Bewegen Sie den Mauszeiger auf der Grenze der Markierung. Das Symbol wird zu einem weißen Pfeil mit schwarzem Pfeilkreuz an der Spitze. Drücken Sie zusätzlich die **[Strg]**-Taste, erscheint ein Pluszeichen statt des Pfeilkreuzes (Abbildung 2.19).
3. Halten Sie die **[Strg]**-Taste und ziehen Sie die Zelle(n) mit der linken Maustaste an den gewünschten neuen Standort. Lassen Sie dort zuerst die linke Maustaste und danach die **[Strg]**-Taste los.

Achten Sie darauf, dass Sie tatsächlich zuerst die Maustaste und erst danach die **[Strg]**-Taste loslassen. Wenn Sie die umgekehrte Reihenfolge verwenden, heben Sie den Kopierbefehl auf und verschieben dann die Zellinhalte.

Wichtig

Preis	Menge	Gesamt
22,55 €	4	90,20 €
33,50 €	3	100,50 €
17,56 €	5	87,80 €

Abbildung 2.19: Zum Kopieren muss der Mauszeiger ein Pfeil mit einem Pluszeichen sein

Untersuchen Sie, was sich an den Zellinhalten geändert hat, indem Sie die kopierten Zellen anklicken und in der Bearbeitungsleiste deren Inhalt überprüfen. Lassen Sie uns kurz das Thema des Abschnitts »Relative Bezüge« (siehe Seite 90) wiederholen:

Vielleicht erzielen Sie den besten Aha-Effekt, wenn Sie zuerst einmal versuchen, sich selbst die Veränderungen zu erklären. Fragen Sie sich, welcher Rechenauftrag sich in räumlicher Beziehung in der Ursprungszelle (in Abbildung 2.20 die Zelle D4) befindet. Welche Zellen sollen – von der Zelle D4 aus gesehen – multipliziert werden? Wie lautet der Rechenauftrag, der sich nach dem Kopieren in den Zellen befindet?

	A	B	C	D	E	F
3		Tagespauschale	Tage	Gesamt	Formel	
4		55 €	5	275 €	=B4*C4	
5		50 €	5			
6		45 €	4			
7		60 €	3			- €
8		63 €	7			=D7*E7
9		70 €	2	140,00 €		
10				=B9*C9		
11						
12						- €
13						=D12*E12
14						
15		#BEZUG!				- €
16		=#BEZUG!*A15				=C15*D15

Abbildung 2.20: Formelveränderungen nach dem Kopieren

Wenn Sie den Rechenauftrag in der Zelle D4 verbal ausdrücken, könnte er lauten: »Multipliziere die zwei linken Nachbarzellen«. Genau dieser Rechenauftrag wurde auch in alle anderen Zellen kopiert.

- ▶ In den Zellen E15, F7 und F12 ergibt die Berechnung den Wert 0, da die beiden linken Nachbarzellen leer sind
- ▶ In der Zelle B15 kommt es für den ersten Zellbezug zu der Fehlermeldung #BEZUG!, den die Formel als Endwert zurückgibt
- ▶ Nur in der Zelle D9 ergibt sich ein sinnvolles Ergebnis, denn hier müssen tatsächlich die beiden Nachbarzellwerte multipliziert werden

Zusammenfassend können Sie sagen: Die Zellbezüge haben sich beim Kopieren immer im Verhältnis zu ihrem jeweiligen Standort so verändert, dass der ursprüngliche Auftrag – der in der Formel verankert ist – an allen Standorten in der gleichen Art ausgeführt wird.

Wie Sie bereits wissen, wird ein Zellbezug, der diese Eigenschaften besitzt, als relativer Bezug bezeichnet (lat.: relatio = Verhältnis). Wollen Sie ein anderes Verhalten beim Kopieren erreichen, müssen absolute oder gemischte Zellbezüge einsetzen.

Kopieren mit der Ausfüllen-Funktion

Eine weitere Möglichkeit, mit der Sie schnell und exakt Ihre Formeln kopieren können, eröffnet Ihnen das sogenannte Ausfüllen, das in Excel in einigen Variationen vorhanden ist.

Um an das Ausfüllen nach unten zu gelangen, öffnen Sie einfach das Dropdownmenü *Füllbereich* in der *Bearbeiten*-Gruppe der Registerkarte *START* (Abbildung 2.21) und wählen direkt den gewünschten Befehl aus.

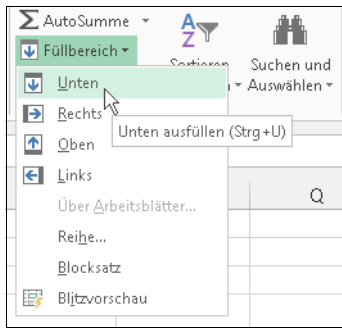


Abbildung 2.21: Das Dropdownmenü *Füllbereich* in der *Bearbeiten*-Gruppe der Registerkarte *START* von Excel 2013

Für die beiden häufigsten Ausfüllrichtungen können Sie auch Tastenkombinationen benutzen: Sie füllen mit der Tastenkombination **[Strg] + [U]** nach unten und mit der Tastenkombination **[Strg] + [R]** nach rechts aus.

Tip

Mit der Maus ausfüllen

Das Ausfüllen mit der Maus ist die einfachste und sehr wahrscheinlich auch die am weitesten verbreitete Form des Ausfüllens. In der rechten unteren Ecke der aktiven Zelle oder eines markierten Bereichs finden Sie ein kleines Kästchen, das sogenannte Ausfüllkästchen. Wenn Sie den Mauszeiger exakt auf dieses Ausfüllkästchen bewegen, wird dieser kurz vor dem Ausfüllkästchen zu einem fettschwarzen Pluszeichen und signalisiert dadurch die Bereitschaft zum Ausfüllen. Ziehen Sie mit gedrückter linker Maustaste über den Bereich, in den die Formel kopiert werden soll.

Sollten Sie beim Ausfüllen über die Zelle oder Zellen fahren, die die Formel enthält, wird diese grau dargestellt. Wenn Sie in diesem Augenblick die linke Maustaste loslassen, wird der Inhalt der Zelle oder Zellen, die grau abgeblendet sind, gelöscht.

Hinweis

Sollte Ihnen dieses Missgeschick widerfahren, klicken Sie in der *Symbolleiste für den Schnellzugriff* oben in der Titelleiste auf die Schaltfläche *Rückgängig* oder drücken Sie die Tastenkombination **[Strg] + [Z]**.

Probieren Sie die oben Mausaktion einmal mit gedrückter **rechter** Maustaste aus! Nach dem Loslassen erscheint ein Kontextmenü, das Ihnen die schnelle Auswahl zwischen den nun möglichen Aktionen ermöglicht (Abbildung 2.22).

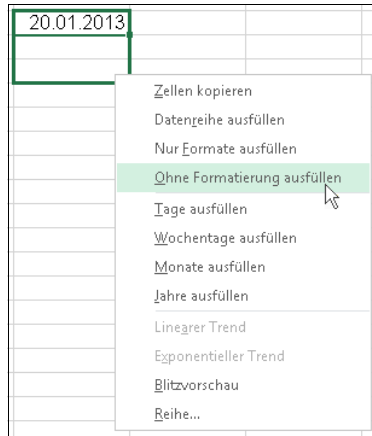


Abbildung 2.22: Das Kontextmenü nach dem Ziehen am Ausfüllkästchen mit der **rechten** Maustaste in Excel 2013

Ausfüllen mit einem Doppelklick

Die pfiffigste Art des Ausfüllens ist jene mit einem Doppelklick. Dabei müssen Sie folgende Beschränkung beachten:

- ▶ Es kann nur vertikal von oben nach unten ausgefüllt werden
- ▶ Sollten sich in der gleichen Spalte unmittelbar unterhalb der Zelle (deren Inhalt durch Ausfüllen kopiert werden soll) Zellen mit Formeln befinden, werden diese kommentarlos überschrieben

Der Ausfüllbereich endet nach folgenden Maßgaben, wenn Excel in den unmittelbar angrenzenden Zellen keine Eingaben findet:

- ▶ Es wird so lange ausgefüllt, wie Excel in der gleichen Spalte, unmittelbar unterhalb der Zelle, weitere Zellen mit Formeln findet. Diese Eingaben werden überschrieben.
- ▶ Wenn sich in der unmittelbar links angrenzenden Spalte Daten befinden, richtet sich der Ausfüllbereich in Excel 2007 nach den in der linken Spalte befindlichen Daten. Interessant: Eine ausgeblendete linke Nachbarspalte wird nicht beachtet, sondern die nächste, sichtbare Spalte. Die unmittelbar angrenzende rechte Spalte bleibt außer Betracht.
- ▶ Gute Nachrichten zur Orientierung an der linken Nachbarspalte beim Ausfüllen mit Doppelklick: Ab Excel 2010 gilt die vorige Beschreibung nicht mehr – Excel erkennt auch bei lückenhafter linker Nachbarspalte das Ende der Liste und kopiert in der richtigen Länge nach unten.
- ▶ Befinden sich ausschließlich in der angrenzenden rechten Spalte Daten, richtet sich der Ausfüllbereich nach den Daten in der rechten Spalte

Um angrenzende Zellen durch einen Doppelklick auszufüllen, verfahren Sie bitte wie folgt:

1. Setzen Sie die Markierung auf die Zelle, deren Formel in den unteren Bereich ausgefüllt werden soll.
2. Zeigen Sie auf das Ausfüllkästchen, bis der Mauszeiger als Pluszeichen dargestellt wird. Wenn Sie jetzt einen Doppelklick ausführen, wird nach den weiter oben geschilderten Methoden der Bereich mit der Formel ausgefüllt.

Formelerggebnisse in Festwerte umwandeln

In bestimmten Situationen ist es notwendig, Formeln durch ihr Ergebnis zu ersetzen, z.B. wenn Sie einen Snapshot der Werte für eine Archivierung benötigen und verhindern müssen, dass eine Neuberechnung mit geänderten Ausgangswerten erfolgt.

Hier setzen Sie eine Spezialform des Kopierens mit der Maus ein, indem Sie die rechte Maustaste benutzen:

1. Markieren Sie die Zellen, die Formeln oder Verknüpfungen enthalten.
2. Klicken Sie mit der **rechten** Maustaste auf den Rand der markierten Zellen und ziehen Sie die Auswahl bei gedrückter Maustaste eine Spalte nach rechts und sofort wieder zurück.
3. Aus dem sich öffnenden Kontextmenü wählen Sie den Befehl *Hierhin nur als Werte kopieren*.

Das Verfahren können Sie auch (allerdings etwas aufwendiger) über die Zwischenablage absolvieren, indem Sie eine Spezialform des Kopierens über die Zwischenablage einsetzen:

1. Markieren Sie die Zellen, die Formeln oder Verknüpfungen enthalten.
2. Wählen Sie den Befehl *START/Kopieren* oder die Tastenkombination **[Strg] + [C]**.
3. Wählen Sie dann den Befehl *Einfügen/Inhalte einfügen* in der Gruppe *Zwischenablage* auf der Registerkarte *START*.
4. Im Dialogfeld *Inhalte einfügen* markieren Sie die Option *Werte* und schließen mit *OK* ab.

Alternativ können Sie direkt den Befehl *Werte einfügen* im Dropdownmenü des *Einfügen*-Symbols wählen. Ab Version 2010 bietet Excel im Abschnitt *Werte einfügen* sogar Varianten an (Abbildung 2.23).

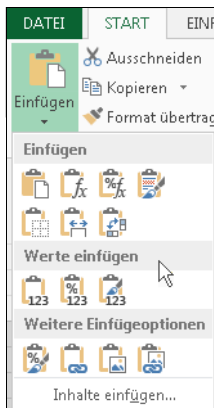





Abbildung 2.23: Der Abschnitt *Werte einfügen* in Excel 2013

Schauen Sie sich das Dialogfeld *Inhalte einfügen* genauer an und probieren Sie auch die anderen Optionen aus. Eine dieser Optionen beschreibt der folgende Abschnitt. Sie haben hier eine Fülle von Arbeitserleichterungen – nutzen Sie sie!

Tip

Schnelle Umrechnung vorhandener Werte

Nehmen wir an, Sie haben noch alte Tabellen mit vielen DM-Beträgen. Sie möchten alle DM-Werte in den Zellen schnell in Euro umrechnen. Dabei möchten Sie den kleinen Nachteil der Euro-Umrechnungsfunktion von Excel umgehen, die nicht auf demselben Bereich umrechnet, sondern das Ergebnis immer in einen anderen Bereich schreibt.

1. In eine beliebige freie Zelle schreiben Sie den Umrechnungskurs 1,95583 und bestätigen mit der -Taste.
2. Markieren Sie jetzt die soeben definierte Zelle mit dem Umrechnungskurs.
3. Drücken Sie die Tastenkombination +.
4. Markieren Sie den Bereich mit Ihren DM-Werten.
5. Wählen Sie im Menüband auf der Registerkarte *START* den Befehl *Inhalte einfügen* im Dropdownmenü des *Einfügen*-Symbols.
6. Aktivieren Sie die Option *Dividieren* und bestätigen Sie mit *OK*.

Formeln schützen

Wenn Sie vermeiden wollen, dass Ihre Formeln durch andere Bearbeiter verändert werden, haben Sie in Excel die Möglichkeit, bestimmte Zellbereiche, z.B. Ihre Formelzellen, zu schützen. Befolgen Sie dazu diese Arbeitsschritte:

1. Markieren Sie den Zellbereich mit den Formeln, die Sie schützen möchten.
2. Klicken Sie mit der rechten Maustaste auf diesen markierten Bereich und wählen Sie aus dem Kontextmenü den Befehl *Zellen formatieren*.
3. Auf der Registerkarte *Schutz* kontrollieren Sie das Kontrollkästchen *Gesperrt* und aktivieren es, sofern es nicht aktiviert ist, und bestätigen mit *OK*.
4. Klicken Sie auf das Symbol *Blatt schützen* der Registerkarte *ÜBERPRÜFEN* im Menüband.
5. Vergeben Sie ggf. ein Kennwort, welches Ihnen erlaubt, später Änderungen am Arbeitsblatt durchzuführen.
6. Klicken Sie auf *OK*, um den Blattschutz zu aktivieren.

Bei jedem Versuch, schreibend auf eine Formelzelle zuzugreifen, meldet Excel, dass Veränderungen in diesem Bereich nicht möglich sind.

Formeln verstecken

Sie können noch einen Schritt weiter gehen und verhindern, dass die in Ihren Formelzellen enthaltenen Formeln in der Bearbeitungsleiste angezeigt werden. Befolgen Sie dazu diese Arbeitsschritte:

1. Markieren Sie den Zellbereich mit den Formeln, den Sie schützen möchten.
2. Klicken Sie mit der rechten Maustaste auf diesen markierten Bereich und wählen Sie im Kontextmenü den Befehl *Zellen formatieren* aus.

3. Auf der Registerkarte *Schutz* aktivieren Sie das Kontrollkästchen *Ausgeblendet* und bestätigen mit *OK*.
4. Klicken Sie auf das Symbol *Blatt schützen* der Registerkarte *ÜBERPRÜFEN* im Menüband.
5. Vergeben Sie ggf. ein Kennwort, welches Ihnen erlaubt, später Änderungen am Arbeitsblatt durchzuführen.
6. Klicken Sie auf *OK*, um den Blattschutz zu aktivieren.

Unabhängig davon, ob die Zelle gesperrt wurde, erfolgt jetzt keine Anzeige der Formel in der Bearbeitungsleiste oben. Dieses Feature ergibt sicherlich nur in Kombination mit der Sperrung der Zelle(n) Sinn.

Art der Formelberechnung bestimmen

Sie können festlegen, ob Excel sofort nach einer Eingabe neu rechnen soll oder nicht. Die entsprechenden Optionen dafür finden Sie in den *Excel-Optionen* in der Kategorie *Formeln*. Die Auswahl anderer *Berechnungsoptionen* wirkt sich auf alle geöffneten Arbeitsblätter in der aktuellen Sitzung aus. Die Standardeinstellung für die Berechnung lautet *Automatisch*. Die Optionen haben folgende Bedeutungen:

- ▶ **Automatisch** Berechnet alle abhängigen Formeln bei jeder Änderung an einem Wert, einer Formel oder einem Namen. Dies ist die Standardeinstellung für die Berechnung.
- ▶ **Automatisch außer bei Datentabellen** Berechnet alle abhängigen Formeln außer Datentabellen. Für die Berechnung von Datentabellen klicken Sie im Menüband auf der Registerkarte *FORMELN* im Abschnitt *Berechnung* auf die Schaltfläche *Neu berechnen (F9)*.
- ▶ **Manuell** Berechnet geöffnete Arbeitsmappen nur, wenn Sie im Arbeitsblatt die Taste F9 drücken oder auf der Registerkarte *FORMELN* im Abschnitt *Berechnung* auf die Schaltfläche *Neu berechnen (F9)* klicken. Durch Klicken auf die Option *Manuell* aktiviert Excel automatisch das Kontrollkästchen *Vor dem Speichern die Arbeitsmappe neu berechnen*.
- ▶ **Vor dem Speichern die Arbeitsmappe neu berechnen** Berechnet Daten vor dem Speichern der Arbeitsmappe neu. Dauert das Speichern einer Arbeitsmappe zu lange, kann die Wartezeit evtl. durch das Deaktivieren des Kontrollkästchens verkürzt werden.

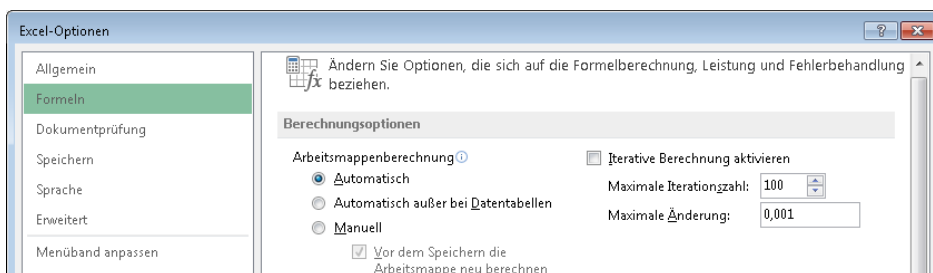


Abbildung 2.24: Die Berechnungsoptionen in Excel 2013

Die Aktionen *Neu berechnen* (F9) sowie *Blatt berechnen* (Umschalt)+(F9) sind direkt in der Gruppe *Berechnung* auf der Registerkarte *FORMELN* aufzurufen (Abbildung 2.25). Auch die Berechnungsoptionen *Automatisch* (beide Varianten) und *Manuell* können Sie nach einem Klick auf das Symbol *Berechnungsoptionen* umschalten.



Abbildung 2.25: Die drei Symbole in der Gruppe *Berechnung* auf der Registerkarte *FORMELN* von Excel 2013

Formeln analysieren

Beim Bearbeiten von Formeln werden im sogenannten Bearbeitungsmodus zu Ihrer Erleichterung alle Zellen und Bereiche, auf die sich die Formel bezieht, farblich angezeigt und von einem gleichfarbigen Rahmen umgeben. Dadurch können Sie den jeweiligen Bezug sehr einfach den dazugehörigen Zellen zuordnen.

In den Bearbeitungsmodus gelangen Sie entweder, indem Sie auf die Zelle, die eine Formel enthält, einen Doppelklick ausführen, oder indem Sie die Zelle, welche die Formel enthält, markieren und dann die **F2**-Taste drücken.

Die Formelüberwachung

Das Hervorheben der Beziehungen zwischen den Zellen wird durch die *Formelüberwachung* besonders gut dargestellt. Wenn Sie überprüfen müssen, aus welchen Zellen die Formel in der Abbildung 2.26 den neuen Seitenumfang berechnet, aktivieren Sie die Formelüberwachung. Mit ihrer Hilfe schrumpft diese Arbeit auf wenige Mausklicks zusammen und wird grafisch einwandfrei gelöst – wie in Abbildung 2.26 zu sehen ist. Markieren Sie die betreffende Formelzelle und wählen Sie den Befehl *Spur zum Vorgänger* im Abschnitt *Formelüberwachung* auf der Registerkarte *FORMELN*.

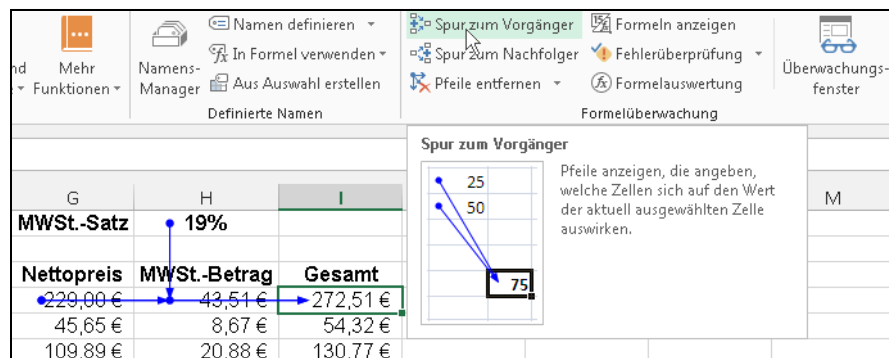


Abbildung 2.26: Die Spur zum Vorgänger zeigt die Bezüge der Formelzelle an

Durch Spurpfeile wird der Fluss von Werten und Formelergebnissen in einem Arbeitsblatt angezeigt. So können Sie sogenannte *Vorgänger* (Zellen, auf die sich der Bezug einer Formel bezieht) oder *Nachfolger* (Zellen, die einen Bezug zu anderen Zellen besitzen) aufspüren und anzeigen lassen.

Um die Spurpfeile der Formelüberwachung anzuzeigen, müssen in den *Excel-Optionen* folgende Einstellungen beachtet werden: In der Kategorie *Erweitert* im Abschnitt mit den Arbeitsblattoptionen muss unter *Objekte anzeigen als* die Option *Alle* eingestellt sein. Ist das Optionsfeld *Nichts (Objekte ausblenden)* aktiviert, werden die Spurpfeile der Formelüberwachung nicht angezeigt!

Hinweis

Den Daten- und Formelfluss verfolgen

Angenommen, Sie wollen herausfinden, aus welchen Zellen bzw. Zellergebnissen sich die Formel einer bestimmten Zelle zusammensetzt. Mit der Formelüberwachung können Sie den Fluss von Formeln und Daten verfolgen. Gehen Sie wie folgt vor:

1. Markieren Sie die Zelle, von deren Position aus die Überprüfung erfolgen soll. In dieser Zelle kann eine Formel enthalten sein bzw. eine Formel kann auf diese Zelle Bezug nehmen oder eine Fehlermeldung beinhalten.
2. Aktivieren Sie die Registerkarte *FORMELN*, um auf die Gruppe *Formelüberwachung* (Abbildung 2.26) zuzugreifen.
 - ▶ **Spur zum Vorgänger** Wenn Sie den Befehl zum ersten Mal aufrufen, werden Spuren zu allen Zellen gelegt, die unmittelbar in den Bezügen der Formel verwendet werden. Wählen Sie den Befehl erneut, wenn zusätzlich die nächste Vorgängerebene angezeigt werden soll.
 - ▶ **Spur zum Nachfolger** Wenn Sie Spuren zu den Zellen legen möchten, die entweder von dem Wert oder vom Ergebnis dieser Zelle abhängig sind, wählen Sie diesen Befehl aus. Wenn Sie die nächste Nachfolgerebene sehen möchten, müssen Sie den Befehl erneut auswählen.
 - ▶ **Fehlerüberprüfung/Spur zum Fehler** Wenn die markierte Zelle eine Fehlermeldung enthält, können Sie ggf. eine Spur zu jener Zelle legen, die die Ursache für den Fehler enthält. Spuren zu Fehlerzellen werden standardmäßig in Rot gelegt. Sollte die markierte Zelle für diesen Befehl nicht geeignet sein, erhalten Sie einen entsprechenden Hinweis durch das Programm.
 - ▶ **Pfeile entfernen** Durch diesen Befehl entfernen Sie alle zuvor gelegten Spuren

	A	B	C
10	Berechnen von Gewinnanteilen		
11	Gewinn:	185.000,00 €	
12	Person	Kapital	Gewinnanteil
13	A	124.000,00 €	58.304,74 €
14	B	85.700,00 €	40.296,10 €
15	C	183.750,00 €	86.399,16 €
16	Summen	393.450,00 €	185.000,00 €
17			

Abbildung 2.27: Spuren zu den Vorgängerebenen auf zwei Ebenen

Zeigt der Spurpfeil auf eine Formel, geschieht dies mit einer durchgehend blauen Linie (Abbildung 2.27).

Bei einer Fehlerspur wird die Linie durchgehend rot dargestellt. Sollten sich mehrere Fehler in einer Spur befinden, hält die Formelüberwachung an und lässt Sie die weitere Vorgehensweise festlegen.

Hinweis Die Excel-Versionen ab 2002 kennzeichnen jede Fehlerstelle im Arbeitsblatt durch ein grünes Dreieck in der linken oberen Zellecke, sofern Sie dies in den Optionen nicht abgeschaltet haben. Markieren Sie eine solche Zelle, erscheint das Achtung-Symbol davor. Dessen QuickInfo enthält die Fehlermeldung und Sie können im Dropdownmenü festlegen, mit welcher Aktion Sie auf den gemeldeten Fehler reagieren wollen.

Wenn eine Spur zu einem externen Bezug (z.B. eine andere Tabelle in gleicher Arbeitsmappe) angezeigt wird, ist deren Linie schwarz und zeigt auf ein Tabellensymbol. Ein Doppelklick auf den schwarzen Pfeil bringt Sie zum Ziel.

Der Abschnitt *Formelüberwachung*

Weil sie sehr nützlich sind, wollen wir in diesem Abschnitt die Symbole aus dem Abschnitt *Formelüberwachung* zusammenfassend erklären.

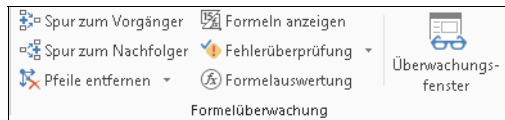













Abbildung 2.28: Der Abschnitt *Formelüberwachung* auf der Registerkarte *FORMELN*

Die Schaltflächen der Gruppe *Formelüberwachung* werden in Tabelle 2.8 anhand ihrer jeweiligen Beschriftung besprochen.

Tabelle 2.8
Die Schaltflächen des *Formelüberwachung*-Abschnitts auf der Registerkarte *FORMELN*

QuickInfo	Auswirkung
<p><i>Spur zum Vorgänger</i></p> 	Die Spuren von der markierten Zelle zu den unmittelbaren Vorgängerkellen werden gekennzeichnet. Um weitere Ebenen von Vorgängern anzuzeigen, müssen Sie erneut auf die Schaltfläche klicken.
<p><i>Spur zum Nachfolger</i></p> 	Zu Formeln, die unmittelbar auf die markierte Zelle Bezug nehmen, werden Spurpfeile gezeichnet. Durch einen erneuten Klick werden Spurpfeile – falls vorhanden – zur nächsten Ebene gezeichnet.
<p><i>Pfeile entfernen</i></p> 	Im aktiven Blatt werden alle zuvor gezeichneten Spurpfeile gelöscht. Das Dropdownmenü enthält auch die beiden folgenden Befehle.
<p><i>Spur zum Vorgänger entfernen</i></p> 	Im Dropdownmenü <i>Pfeile entfernen</i> : Die Spurpfeile zu einer Vorgängerebene werden gelöscht. Waren mehrere Vorgängerebenen angezeigt, müssen Sie für jede Ebene erneut klicken.
<p><i>Spur zum Nachfolger entfernen</i></p> 	Die Spurpfeile zur Nachfolgerebene werden gelöscht. Waren mehrere Nachfolgerebenen angezeigt, müssen Sie für jede Ebene erneut klicken.



QuickInfo	Auswirkung
Formeln anzeigen 	Schaltet die Formelanzeige ein/aus (siehe den Abschnitt »Formelansicht ein/aus« ab Seite 97)
Fehlerüberprüfung 	Im gleichnamigen Dropdownmenü: Excel startet die Fehlerüberprüfung, die alle Fehler anzeigt und die Ursachen beschreibt (Abbildung 2.29)
Spur zum Fehler 	Durch einen Klick auf diese Schaltfläche können Spurpfeile zu einer Fehlerquelle angezeigt werden
Zirkelbezüge 	Zeigt alle in der Arbeitsmappe vorhandenen Zirkelbezüge an. Mit Klick auf einen Bezug erfolgt der Sprung zur gewählten Zelle.
Formelbewertung 	Mithilfe dieser Funktion lösen Sie schrittweise eine Formel auf. In diesem Analyseprozess sind auch Fehler schnell erkannt.
Überwachungsfenster 	Bei der Fehlersuche oder in anderen Situationen können Sie ständig die Inhalte ausgewählter Zellen beobachten (Abbildung 2.30)

Fehlersuche 1

Ein Beispiel: In einem Arbeitsblatt erscheinen nicht die erwarteten Berechnungsergebnisse. Sie wollen feststellen, ob bei den Daten oder Formeln Fehler vorliegen. Starten Sie die Fehlerüberprüfung, indem Sie den Befehl *Fehlerüberprüfung* aufrufen. Es erscheint das Dialogfeld für den ersten gefundenen Fehler (Abbildung 2.29). Sie entscheiden über die nächsten Schritte.

Alternativ können Sie auch auf Zellen mit dem grünen Dreieck in der linken oberen Ecke und dann auf das daneben erscheinende Achtung-Symbol klicken. Im dann aufgehenden Menü wählen Sie die weiteren Schritte.

	A	B	C	D
10	Berechnen von Gewinnanteilen			
11	Gewinn:	185.000,00 €		
12	Person	Kapital	Gewinnanteil	
13	A	124.000,00 €	58.304,74 €	
14	B	85.700,00 €	#WERT!	
15	C	183.750,00 €	86.399,16 €	
16	Summen	393.450,00 €	#WERT!	

Fehlerüberprüfung

Fehler in Zelle C14
 =B14/\$B\$16*\$A\$11

Fehler in Wert
 Ein in der Formel verwendeter Wert ist vom falschen Datentyp.

Hilfe für diesen Fehler anzeigen

Berechnungs-Schritte anzeigen...

Fehler ignorieren

In Bearbeitungsleiste bearbeiten

Optionen...

Zurück Weiter

Abbildung 2.29: Im Fehlerüberprüfungsmodus werden alle fehlerhaften Formeln und Daten untersucht

Fehlersuche 2

Ein weiteres Beispiel: Am Ende einer Reihe von aufeinander aufbauenden Berechnungsschritten erscheint ein nicht erwartetes Ergebnis. Sie wollen während der Manipulation der Ausgangsdaten alle Zwischenberechnungen beobachten. Gehen Sie wie folgt vor:

1. Wählen Sie den Befehl *Überwachungsfenster anzeigen* im Abschnitt *Formelüberwachung* der Registerkarte *FORMELN*. Es erscheint das Dialogfeld *Überwachungsfenster* (Abbildung 2.30).
2. Klicken Sie auf die Schaltfläche *Überwachung hinzufügen* und wählen Sie die erste Zelle mit einer Zwischenberechnung aus. Klicken Sie dann auf *Hinzufügen*.
3. Verfahren Sie so mit allen weiteren Zwischenberechnungszellen.
4. Manipulieren Sie nun die Ausgangswerte. Im Überwachungsfenster (Abbildung 2.30) sehen Sie, wie sich die Zwischenberechnungen verhalten.

Wird das Fenster nicht benötigt, schließen Sie es. Die zu überwachenden Zellen bleiben Ihnen erhalten. Sie sehen dies, wenn Sie später das Fenster wieder einschalten.

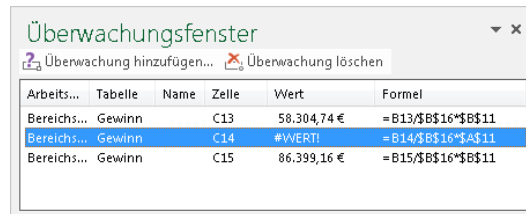


Abbildung 2.30: Im Überwachungsfenster beobachten Sie den Inhalt ausgewählter Zellen

Gerade bei der Fehlersuche ist die FormelAuswertung besonders interessant. Excel lässt Sie schrittweise an der Auflösung der Formel teilhaben.



Markieren Sie eine Formelzelle und wählen dann den Befehl *FormelAuswertung*. Im nun geöffneten Dialogfeld (Abbildung 2.31) lösen Sie mithilfe der Schaltflächen *EinzelSchritt* bzw. *Prozedurschritt* die Formel auf. Auf diese Art und Weise finden Sie auch schnell heraus, ob in Teilen der Formel Denkfehler stecken, die von Excel nicht als Fehler gefunden werden können, aber trotzdem zu falschen Ergebnissen führen.

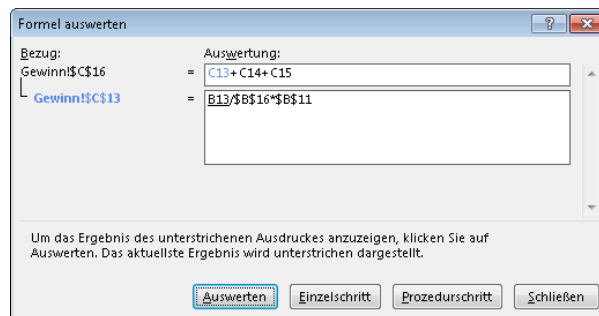


Abbildung 2.31: In der FormelAuswertung sehen Sie, wie Excel eine Formel auflöst

Wichtig Das gesamte Verhalten von Excel bezüglich der Fehler- und Formelüberwachung steuern Sie in den *Excel-Optionen* in der Kategorie *Formeln*. Beachten Sie hier die Abschnitte *Fehlerüberprüfung* und *Regeln für die Fehlerüberprüfung*.

Kapitel 3

Formeln und Funktionen

Was kennzeichnet Tabellenfunktionen?	117
Eingabe einer Funktion	121
Tipps & Tricks	130

In Kapitel 2 haben Sie Grundlegendes über die Eingabe von Formeln erfahren. Sie werden beim Aufbau Ihrer Kalkulationen sehr schnell einen Punkt erreichen, wo die Formeln umständlich und je nach Rechenweg auch kompliziert werden. Nehmen Sie die Summenbildung in Abbildung 3.1. Haben Sie Lust, eine so lange Formel einzugeben?

	A	B	C	D	E
1					
2		Filiale 1	Filiale 2	Filiale 3	
3	Januar	20.066,00 €	10.792,00 €	17.934,00 €	
4	Februar	20.421,00 €	10.445,00 €	16.989,00 €	
5	März	18.849,00 €	11.130,00 €	16.767,00 €	
6	April	18.882,00 €	10.865,00 €	15.318,00 €	
7	Mai	19.427,00 €	10.181,00 €	15.777,00 €	
8	Juni	20.155,00 €	10.032,00 €	17.754,00 €	
9	Juli	18.939,00 €	10.290,00 €	15.871,00 €	
10	August	19.080,00 €	10.162,00 €	17.445,00 €	
11	September	18.490,00 €	11.111,00 €	15.593,00 €	
12	Oktober	18.218,00 €	11.130,00 €	15.193,00 €	
13	November	19.955,00 €	10.912,00 €	15.658,00 €	
14	Dezember	20.416,00 €	11.663,00 €	15.404,00 €	
15	Gesamt	=B3+B4+B5+B6+B7+B8+B9+B10+B11+B12+B13+B14			
16					

Abbildung 3.1: Wer will schon solche Formeln zur Summenbildung eintippen?

Was, wenn nicht 12, sondern 100 oder noch mehr Werte zu addieren sind? Spätestens hier sollten Sie sich mit dem Thema »Tabellenfunktionen« beschäftigen. Im konkreten Fall verkürzt und erleichtert die Funktion SUMME() die Formeleingabe. Die genaue Beschreibung dieser Funktion finden Sie in Kapitel 15.

	A	B	C	D
1				
2		Filiale 1	Filiale 2	Filiale 3
3	Januar	20.066,00 €	10.792,00 €	17.934,00 €
4	Februar	20.421,00 €	10.445,00 €	16.989,00 €
5	März	18.849,00 €	11.130,00 €	16.767,00 €
6	April	18.882,00 €	10.865,00 €	15.318,00 €
7	Mai	19.427,00 €	10.181,00 €	15.777,00 €
8	Juni	20.155,00 €	10.032,00 €	17.754,00 €
9	Juli	18.939,00 €	10.290,00 €	15.871,00 €
10	August	19.080,00 €	10.162,00 €	17.445,00 €
11	September	18.490,00 €	11.111,00 €	15.593,00 €
12	Oktober	18.218,00 €	11.130,00 €	15.193,00 €
13	November	19.955,00 €	10.912,00 €	15.658,00 €
14	Dezember	20.416,00 €	11.663,00 €	15.404,00 €
15	Gesamt	=SUMME(B3:B14)		
16		SUMME(Zahl1; [Zahl2]; ...)		
17				

Abbildung 3.2: Die Summenbildung mithilfe einer Excel-Funktion

Was kennzeichnet Tabellenfunktionen?

Stellen Sie sich eine Funktion als ein kleines Rechenprogramm vor, das in der Lage ist, bestimmte Rechenoperationen durchzuführen. Wie jeder Mensch haben auch die Funktionen einen Namen, der sie identifiziert. Im Namen wird meist auch die Rechenaufgabe, welche die Funktion löst, in Kurzform beschrieben. Die Tabellenfunktionen dienen zur Ausführung komplexerer mathematischer, statistischer, technischer oder logischer Berechnungen, der Suche nach bestimmten Informationen sowie der Manipulation von Texten.

Am häufigsten werden Funktionen in Formeln eingesetzt, obwohl ihre Verwendung in Namen und Makros (mehr zu Namen in Kapitel 4, zu Makros in Kapitel 5) ebenso möglich ist. Innerhalb von Formeln können Sie Funktionen mehrfach oder auch verschachtelt einsetzen.

Eine Formel konnte in Excel 2003 in bis zu sieben Ebenen verschachtelte Funktionen enthalten. Ab Excel 2007 sind bis zu 64 Ebenen möglich. Dabei ist es Sache des Anwenders – und nicht der Funktion –, darauf zu achten, dass die arithmetische Korrektheit der Formel gewahrt bleibt.

Durch Funktionen wird das Erstellen von einwandfrei funktionierenden Formeln nicht nur bequemer, sondern vor allem auch deutlich schneller. Sie sollten stets versuchen, eher eine Funktion einzusetzen, als Ihre eigenen Formeln zu schreiben. Funktionen sind

- ▶ schnell in der Berechnung,
- ▶ benötigen wenig Platz in der Formel und
- ▶ verringern das Risiko eines Schreibfehlers.

Um rechnen zu können, benötigen Funktionen Informationen (Daten), die in Excel als Argumente oder Parameter bezeichnet werden. Welcher Art diese Argumente sein müssen, wie viele Argumente benötigt werden und ob man auf bestimmte Argumente ggf. verzichten kann, können Sie in der Excel-Hilfe zur entsprechenden Funktion nachlesen.

Syntax einer Funktion

Unter der Syntax der Funktion ist in diesem Zusammenhang die Eingaberegeln – oder Eingabefolge – einer Funktion zu verstehen. Wenn Sie die Syntax nicht einhalten, erhalten Sie von Excel eine Fehlermeldung angezeigt.

Die Syntax für eine Funktion ist immer folgendermaßen aufgebaut (Abbildung 3.3):

Wichtig

- ▶ Die Eingabe einer Funktion beginnt mit dem Funktionsnamen. Steht sie am Beginn einer Formel, beginnt die Eingabe immer mit dem Gleichheitszeichen.
- ▶ Nach dem Namen folgt eine öffnende runde Klammer. Durch diese Klammer wird Excel mitgeteilt, wo die Argumente beginnen. Vor und hinter einer Klammer sind keine Leerzeichen zulässig.
- ▶ Nach der öffnenden Klammer werden die Argumente eingetragen. Die Argumente müssen der von der Funktion geforderten Datenart entsprechen. Einige Funktionen besitzen sogenannte optionale Argumente, die zur Ausführung der Funktion nicht unbedingt erforderlich sind. Nähere Auskunft erhalten Sie in der Excel-Hilfe.
- ▶ Argumente können Konstanten oder Formeln sein, die der geforderten Datenart entsprechen oder diese durch Berechnung liefern.

- ▶ Die Argumente werden jeweils durch ein Semikolon¹ (;) voneinander getrennt
- ▶ Ist für ein Argument die Datenart *Text* erforderlich, muss dieser Text in Anführungszeichen (") gesetzt werden. Sollte ein Argument eine Berechnung sein, kann diese Berechnung wiederum selbst Funktionen enthalten.
- ▶ Wird für ein Argument einer Funktion selbst eine Funktion eingegeben, wird diese Funktion als »verschachtelt« bezeichnet. Sie können in Excel bis zu sieben Funktionsebenen in einer Formel verschachteln.
- ▶ Nachdem alle Argumente eingegeben sind, ist eine schließende runde Klammer einzugeben. Achten Sie darauf, dass Klammern grundsätzlich als Paar zu verwenden sind. Das bedeutet, dass jeder öffnenden Klammer eine schließende Klammer gegenüber stehen muss.

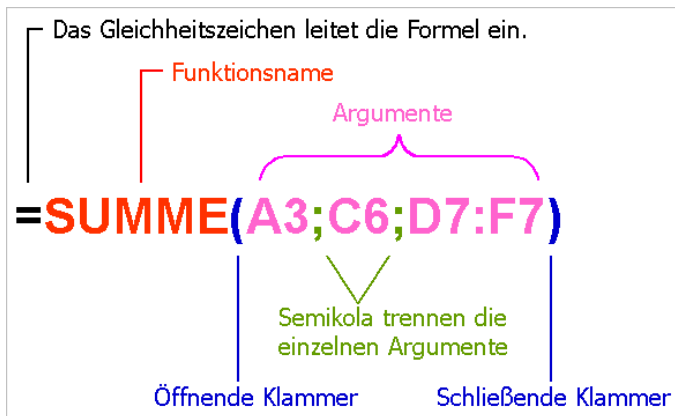


Abbildung 3.3: Allgemeine Syntax am Beispiel der *SUMME()*-Funktion

Argumente als Rechen­daten einer Funktion

Die an eine Funktion zur Verarbeitung übergebenen Daten heißen Argumente. In der Beschreibung der Funktion, die Sie in diesem Buch, in der Excel-Hilfe oder in anderen Nachschlagewerken erhalten, finden Sie die Anzahl der geforderten Argumente sowie deren Datentyp:

- ▶ Als Argument können Sie jeden beliebigen Ausdruck (Konstante, Formel usw.) verwenden, solange dieser Ausdruck (oder dessen Ergebnis) dem verlangten Datentyp entspricht
- ▶ Im später noch zu besprechenden Funktions-Assistenten werden Argumente, die erforderlich sind, **fett** angezeigt. Hingegen werden Argumente, deren Angabe optional ist, in Normalschrift dargestellt.
- ▶ Mehrere Argumente werden durch Semikola (;) getrennt
- ▶ Leerzeichen innerhalb von Funktionen sollten Sie vermeiden, denn sie führen meist zu Fehlermeldungen. Dies gilt generell beim Schreiben von Formeln.

Bei der Übergabe von Argumenten ist der geforderte Datentyp zu beachten. Die Tabelle 3.1 zeigt Ihnen, welche Datentypen in Excel benutzt werden:

¹ In der deutschen Lokalisierung trennt das Semikolon die Argumente, in der englischen das Komma.

Datentyp	Beispielfunktion	Eingabeart
Zahl	=SUMME(Zahl1;Zahl2;...)	Eine Zahl (Konstante); ein Zellenbezug; eine Formel, deren Ergebnis eine Zahl ergibt
Text	=VERKETTEN(Text1;Text2;...)	Text in Anführungszeichen; Zellenbezug
Zellenbezug	=ZEILE(Bezug)	Zellenbezug
Datums-/ Zeitwert	=MONAT(Zahl)	Fortlaufende Zahl; Datum in Anführungszeichen; Zellenbezug
Logisch	=UND(Wahrheitswert1;Wahrheitswert2;...)	Vergleichsausdruck;Wahr; Falsch; Zellenbezug

Tabelle 3.1
Diese Datentypen kennt Excel

Neben dem korrekten Datentyp sollten Sie auch auf eine möglichst rationelle Übergabe der Daten achten. Je kürzer die Datenübergabe ist, desto schneller wird auch die betreffende Funktion arbeiten. Dazu beachten Sie bitte das folgende Beispiel.

Varianten der Argumentübergabe

Welche Möglichkeiten gibt es, die vier Zahlenwerte aus der Abbildung 3.4 als Argumente der Funktion SUMME() zur Bildung einer Summe zu übergeben?

	A	B	C
1			
2		27,5	
3		20,0	
4		12,5	
5		30,0	
6	Σ	?	
7			

Abbildung 3.4: Diese Werte sind der Funktion SUMME() zu übergeben

Variante 1: Werte als Konstanten

Sie übergeben die vier Werte als Konstanten. Beispiel:

=SUMME(27,5;20,0;12,5;30,0)

Die Funktion ermittelt zwar das richtige Ergebnis, jedoch geht sie eindeutig am Sinn einer Tabellenkalkulation vorbei. Bei jeder Änderung eines der Werte in der Tabelle muss auch die Funktion entsprechend geändert werden.

Variante 2: Werte als Zellbezüge

Sie übergeben jeden einzelnen Wert als Zellbezug. Beispiel:

=SUMME(B2;B3;B4;B5)

Auch die so geschriebene Funktion wird das richtige Ergebnis ausgeben. Die Übergabe ist im technischen Sinn korrekt. In diesem Fall ist die Übergabe jedoch umständlich, da eine kürzere Schreibweise für die vier Zellen möglich ist.

Variante 3: Werte in einer Addition

Sie übergeben die drei Eingabewerte als eigene Additionsformel. Beispiel:

```
=SUMME(B2+B3+B4+B5)
```

Die Funktion wird, da das Ergebnis der Addition der vier Zellen ein Wert vom Typ *Zahl* ist (so wie von der Funktion gefordert), erneut das richtige Ergebnis berechnen. Insgesamt handelt es sich jedoch um eine völlig unsinnige Übergabe. Wenn Sie die Addition bereits in den Klammern durchführen, wozu benötigen Sie dann noch die Funktion SUMME()? Solche unsinnigen Argumente belasten bei häufiger Wiederholung die Rechengeschwindigkeit.

Trotzdem zeigt dieses Beispiel, dass Sie natürlich einen Rechenausdruck als Argument übergeben können.

Variante 4: Werte als Bereichsbezug

Sie übergeben die aneinander grenzenden Eingabewerte als Bereichsbezug. Beispiel:

```
=SUMME(B2:B5)
```

Die Funktion wird das richtige Ergebnis ausgeben. Die Übergabe der Werte ist in der kürzesten Form erfolgt, sodass die Funktion mit nur einem Argument (dem Bereich) arbeiten kann. Dies kann als die optimale Übergabeform bezeichnet werden und sollte Ihre Wahl sein.

Ein weiteres Beispiel

Wenn mehrere Zahlenwerte, die sich erst aus verschiedenen Berechnungen ergeben, in eine Funktion übernommen werden müssen, ist es nicht unbedingt notwendig, Zwischenberechnungen einzurichten. Die Berechnung dieser Teilergebnisse kann selbstverständlich in der Funktion stattfinden.

Das Problem können Sie sich recht einfach am Beispiel der Abbildung 3.5 verdeutlichen. In dieser Tabelle sind die mathematischen Aufgabenstellungen auf ein Mindestmaß reduziert, um das Wesentliche – die Übergabe der Argumente – deutlicher werden zu lassen. Sie könnten in den Zellen E2 bis E5 die Formeln für das jeweilige Zwischenergebnis bilden, müssen dies aber nicht.

	A	B	C	D	E
1					
2		40	+	9	?
3		100	-	51	?
4		7	*	7	?
5		245	/	5	?
6			Summe		?
7					

Abbildung 3.5: Die Ergebnisse der vier Berechnungen sollen addiert werden

Angenommen, Sie wollen mit der Funktion SUMME() die Ergebnisse der vier Berechnungen aus der Tabelle in Abbildung 3.5 bilden. Wie übergeben Sie die Argumente?

Aus jeder der vier Berechnungen bilden Sie eine Formel, die Sie der Funktion SUMME() als Argument übergeben. Bei der Auflösung der Formel wird das Programm zuerst die Werte für die Argumente berechnen. Aus diesen Teilergebnissen wird anschließend die geforderte Summe gebildet. Die Formel sieht in diesem Fall folgendermaßen aus:

```
=SUMME(B2+D2;B3-D3;B4*D4;B5/D5)
```

Die Reihenfolge, in der diese vier zu berechnenden Argumente eingegeben werden, ist im Falle der SUMME()-Funktion völlig gleichgültig. Bei der Bildung der Summe gilt auch hier die Regel, dass die Summanden, sprich: die Argumente der Funktion, beliebig angeordnet werden können.

Bitte beachten Sie: Eine Funktion wird immer von innen nach außen aufgelöst.

Wichtig

Natürlich wäre die Summenbildung mit der Formel =SUMME(E2:E5) viel kürzer und damit auch übersichtlicher, wenn die Zwischenergebnisse in der Spalte E zur Verfügung stünden. Wägen Sie also die Vor- und Nachteile immer im konkreten Anwendungsfall ab.

Eingabe einer Funktion

Sie können jede Funktion per Hand eingeben. Dazu müssen Sie den Namen der Funktion sowie die Syntax der Argumente einigermaßen kennen.

Im Gegensatz zu früheren Versionen sind seit Excel 2002 an Ihre Syntaxkenntnisse nicht mehr so hohe Anforderungen gestellt. Den VBA-Programmierern ist sie bereits geläufig: die kontextbezogene Syntax-Hilfe, die als Funktions-QuickInfo eingeblendet wird, sobald Sie den Funktionsnamen und die öffnende Klammer geschrieben haben (siehe Abbildung 3.6). Das jeweils zu notierende Argument ist fett hervorgehoben. Schreiben Sie das erste Semikolon, wird unten das zweite Argument hervorgehoben usw.

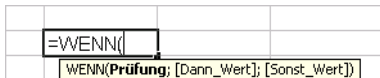


Abbildung 3.6: Die QuickInfo der Funktion WENN() erscheint seit Excel 2002 bei der manuellen Eingabe

Wenn Ihnen eine Funktion wirklich vertraut ist, sind Sie mit der manuellen Eingabe meist schneller als bei der Verwendung von Eingabehilfen.

Seit Excel 2007 wurde die Unterstützung bei der Formeleingabe weiter verbessert. Beginnen Sie mit dem Schreiben einer Formel, erscheint eine Vorschlagsliste (Abbildung 3.7).

Neu ab 2007

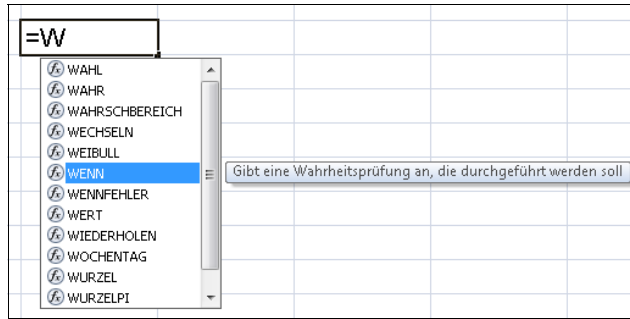


Abbildung 3.7: Excel schlägt eine Liste der mit »W« beginnenden Funktionen vor und beschreibt sie auch

Hinweis Voraussetzung für die Anzeige der QuickInfo ist, dass Sie in den *Excel-Optionen* in der Kategorie *Erweitert*, Abschnitt *Anzeige*, das Kontrollkästchen *QuickInfos für Funktionen anzeigen* aktivieren. Die in *Abbildung 3.7* gezeigte Funktionsliste erscheint, wenn in den *Optionen* von Excel das Kontrollkästchen *AutoVervollständigen für Formel*, zu finden in der Kategorie *Formeln*, Abschnitt *Arbeiten mit Formeln*, aktiviert ist. Standardmäßig ist dies der Fall.

Die Groß- und Kleinschreibung spielt bei der Eingabe einer Funktion keine Rolle. Excel wandelt nach Eingabe der Formel alle Buchstaben in den Funktionsnamen in Großbuchstaben um.

Den Funktions-Assistenten nutzen

Durch den Funktions-Assistenten erhalten Sie bei der Auswahl und Eingabe einer Funktion Unterstützung. Sie rufen den Funktions-Assistenten entweder über

- ▶ einen Klick auf das Symbol *Funktion einfügen* in der Registerkarte *FORMELN*,
- ▶ die Schaltfläche *Funktion einfügen* in der Bearbeitungsleiste (*Abbildung 3.8*) oder
- ▶ den Befehl *Weitere Funktionen* im Dropdownmenü des *AutoSumme*-Symbols auf der Registerkarte *START* auf.

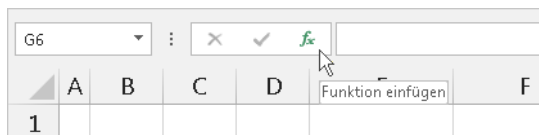


Abbildung 3.8: Funktions-Assistent in der Bearbeitungsleiste aufrufen

Der Funktions-Assistent führt Sie durch die gesamte Eingabe, wobei die Funktion und deren Argumente erläutert werden. Während der gesamten Eingabeprozedur kann die Funktionserstellung abgebrochen werden, indem Sie auf die Schaltfläche *Abbrechen* des Funktions-Assistenten klicken oder die `[Esc]`-Taste drücken.

Eine praktische Übung

Für einen Kredit von 100.000 € soll bei zehn Jahren Laufzeit und 7 % Zinssatz die monatliche Rate errechnet werden. Hierfür setzen Sie natürlich eine Funktion ein.

	A	B	C	D
1				
2		Kredit	100.000 €	
3		Zinsen	7,0%	
4		Laufzeit in Jahren	10	
5				
6		Monatliche Rate:	?	
7				

Abbildung 3.9: Die Aufgabenstellung zur Ratenberechnung

1. Markieren Sie die Tabellenzelle C6, in die Sie die Funktion einfügen wollen.
2. Klicken Sie auf die Schaltfläche *Funktion einfügen* in der Bearbeitungsleiste, um den Funktions-Assistenten zu öffnen.

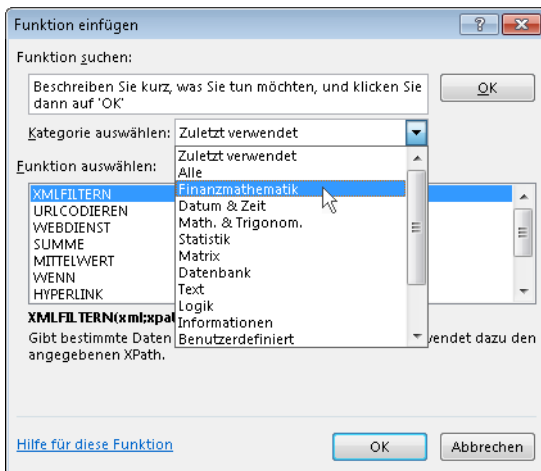


Abbildung 3.10: Das Dialogfeld des Funktions-Assistenten

3. Wählen Sie im Listenfeld *Kategorie auswählen* den von Ihnen benötigten Funktionstyp durch Anklicken aus, in diesem Fall *Finanzmathematik* (Abbildung 3.10). Durch die Auswahl der Kategorie wird die große Anzahl der verfügbaren Funktionen zu überschaubaren Auswahlmengen zusammengefasst.
4. Markieren Sie im Listenfeld *Funktion auswählen* die von Ihnen benötigte Funktion, in diesem Fall *RMZ*. Im unteren Teil des Dialogfelds werden dann die allgemeine Syntax der Funktion sowie eine Kurzbeschreibung angezeigt. Klicken Sie auf die Schaltfläche *OK*.

5. Im folgenden Dialogfeld steht Ihnen für jedes Funktionsargument ein Eingabefeld zur Verfügung. Dabei werden die obligatorischen Argumente **fett**, die optionalen Argumente in normalem Text angezeigt (Abbildung 3.11).

The screenshot shows the 'Funktionsargumente' dialog box for the RMZ function. The arguments are listed as follows:

Argument	Wert	Ergebnis
Zins	C3/12	= 0,005833333
Zr	C4*12	= 120
Bw	C2	= 100000
Zw		= Zahl
F		= Zahl
		= -1161,084792

Gibt die konstante Zahlung einer Annuität pro Periode zurück.

Bw ist der Barwert: Der Gesamtbetrag, den eine Reihe zukünftiger Zahlungen zum gegenwärtigen Zeitpunkt wert ist.

Formelergebnis = -1.161 €

[Hilfe für diese Funktion](#) [OK] [Abbrechen]

Abbildung 3.11: Dialogfeld zur Eingabe der Argumente

6. Füllen Sie die Eingabefelder entsprechend der Abbildung 3.11 aus.
7. Um die Funktion in die Tabelle einzufügen, klicken Sie auf die Schaltfläche **OK**.

Die Formel lautet =RMZ(C3/12;C4*12;C2). Sie ergibt eine monatliche Rate von 1.161,08 €. Das Ergebnis wird als negative Zahl angezeigt. Wollen Sie die Rate als positive Zahl darstellen, multiplizieren Sie das Ganze noch mit -1 .

Der *Zins* ist natürlich der Jahreszins und muss für die Berechnung des monatlichen Zinses durch 12 geteilt werden. Im Argument *Zr* müssen Sie die Laufzeit von 10 Jahren mit 12 multiplizieren, um auf die Anzahl der Monatsraten insgesamt zu kommen (120).

Tipp Geben Sie in die verschiedenen Eingabefelder die gewünschten Argumente in dem jeweils verlangten Datentyp ein. Anstelle der Eingabe von Konstanten empfiehlt es sich, den jeweiligen Zellenbezug zu der entsprechenden Zelle in der Tabelle einzugeben. Sie können eine Zelle durch Anklicken und einen Zellbereich durch Markieren eingeben.

Wenn Sie in ein Eingabefeld klicken, wird im unteren Teil des Dialogfelds eine Erläuterung zu dem jeweiligen Argument angezeigt.

Falls Sie bereits vorhandene Namen für Zellen oder Bereiche eingeben wollen, können Sie die Namen eintragen oder Sie drücken die **[F3]**-Taste, um die Namen aus dem Dialogfeld *Namen einfügen* zu übernehmen.

Die Suche nach Funktionen

Die Entwickler bei Microsoft haben sich etwas für jene Anwender ausgedacht, die noch keinen Überblick über die vorhandenen Funktionen in Excel haben. Sehr häufig taucht in Excel-Trainings nach der Vorstellung bestimmter Funktionen die Frage »Ja, aber wie komme

ich darauf, dass Excel für meine Problemstellung eine Funktion bereit stellt?« auf. Genau dies ist einer der Gründe, warum wir dieses Buch geschrieben haben. Allerdings kann seit Excel 2002 die Antwort auch lauten: »Geben Sie Ihre Problemstellung im Funktions-Assistenten ein und lassen Sie alle Funktionen zu diesem Thema suchen.«

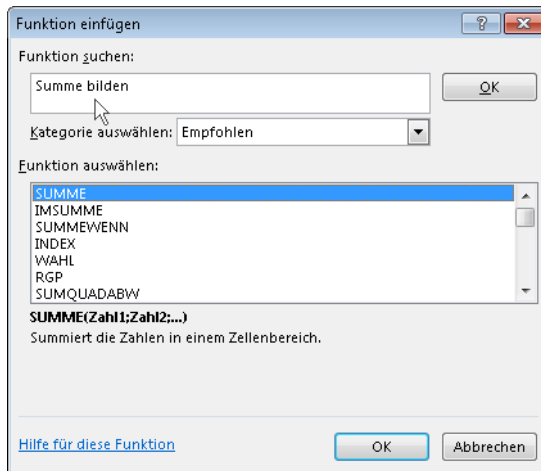


Abbildung 3.12: Seit Excel 2002 verfügt der Assistent über eine Suchfunktion

Rufen Sie einmal den Assistenten auf und tippen Ihre Frage oder Problemstellung ein, wie z.B. die in Abbildung 3.12. Wie in der Excel-Hilfe kommt es natürlich auch hier darauf an, die richtigen Stichwörter im Text anzugeben. Sicherlich sind manche Suchergebnisse nicht brauchbar. Oft erhalten Sie aber eine Auswahl von Funktionen quer durch die Kategorien, in der die benötigte Funktion enthalten ist (Abbildung 3.13).

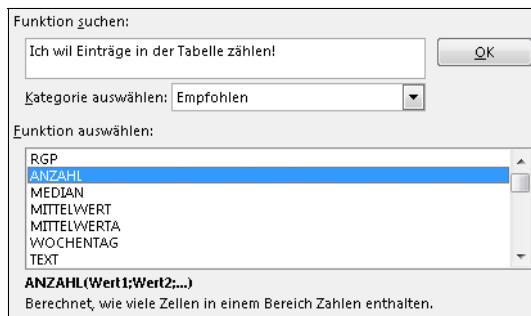


Abbildung 3.13: Die Ergebnisliste in Excel 2013 enthält Funktionen zum Zählen

Seltsamerweise hatte Excel 2007 diese Fähigkeit verloren. Die in Abbildung 3.14 gezeigte Ergebnisliste für dasselbe Ansinnen hat nur bei gutem Willen etwas mit der Aufgabenstellung zu tun. Seit Excel 2010 sieht das Ergebnis wieder besser aus, lässt aber noch die Frage offen, warum eine Funktion wie HÄUFIGKEIT() erst sehr weit unten in der empfohlenen Liste erscheint. Das konnte der Excel-Assistent schon mal besser.

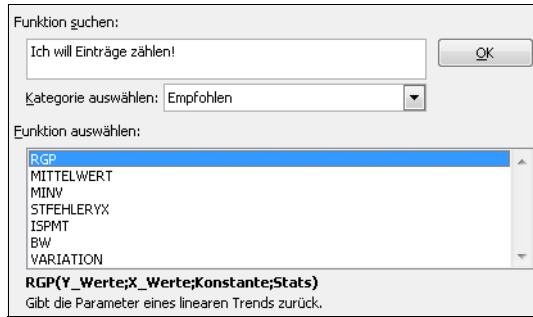


Abbildung 3.14: Die Ergebnisliste in Excel 2007 zeigt das Ergebnis eines verzweifelten Kampfs mit der Aufgabenstellung

Vereinfachte Eingabe der Zellbezüge

Auf der rechten Seite jedes Eingabefelds für Argumente finden Sie eine kleine Schaltfläche mit einem nach links oben gerichteten roten Pfeil.



Abbildung 3.15: Felder zur Eingabe von Argumenten mit Schaltfläche zum Markieren der Bezüge

Wenn Sie auf diese Schaltfläche klicken, wird das restliche Dialogfeld ausgeblendet, sodass Ihnen nur noch das Eingabefeld zur Verfügung steht und Sie eine freie Sicht auf die gesamte Tabelle haben. Darüber hinaus wird das Eingabefeld horizontal etwas vergrößert, um Ihnen mehr Raum für Ihre Eingaben zur Verfügung zu stellen.

Wenn Sie diesen Zustand hergestellt haben, können Sie Zellbezüge – auch Bezüge zu ganzen Bereichen – durch Markieren in der Tabelle eingeben. Hierbei gelten alle Ihnen bekannten Regeln zum Markieren. Wenn Sie einen Bereich markieren, wird automatisch der Bereichsoperator (:) und bei einzelnen Zellen (Sie markieren einzelne Zellen bei gedrückter **[Strg]**-Taste) der Vereinigungsoperator (;) eingetragen. Diese Methode ist nicht nur einfacher, sondern sie hilft auch, Fehler zu vermeiden.

Wenn Sie wieder zur vorherigen Ansicht zurückkehren möchten, klicken Sie erneut auf die Schaltfläche in der rechten Ecke des Eingabefelds.

Excel-Hilfe zu Funktionen

Über den Funktions-Assistenten haben Sie auch den schnellen Zugriff auf das Hilfesystem von Excel, speziell natürlich zu den Funktionen. In der Excel-Hilfe ist jede Funktion mit allen Argumenten und Beispielen ausführlich beschrieben.²

Innerhalb des Funktions-Assistenten erhalten Sie zu der ausgewählten Funktion Hinweise, wenn Sie im linken unteren Teil des Dialogfelds auf den Hyperlink *Hilfe für diese Funktion* klicken. Sie werden dann unmittelbar zum Hilfethema der ausgewählten Funktion geführt (siehe Abbildung 3.16).

² Hier hat sich seit Excel 2002 im Aufruf und der Bedienung der Offline-Hilfe einiges geändert. Wir beschreiben hier die Hilfe in der Version 2013. Grundsätzlich finden Sie die Funktionsbeschreibungen auch in den älteren Versionen schnell aus dem Assistenten heraus.

Sollten Sie dieses Buches lediglich ausgeliehen und nicht ständig verfügbar haben, empfiehlt es sich, die Hilfethemen der Funktionen, die Sie häufiger benutzen, auszudrucken. Auf diese Weise können Sie sich im Laufe der Zeit ein kleines – speziell auf Ihre Bedürfnisse zugeschnittenes – Funktionenbuch erstellen. Um ein Thema auszudrucken, klicken Sie auf die Schaltfläche *Drucken*.

Tipp

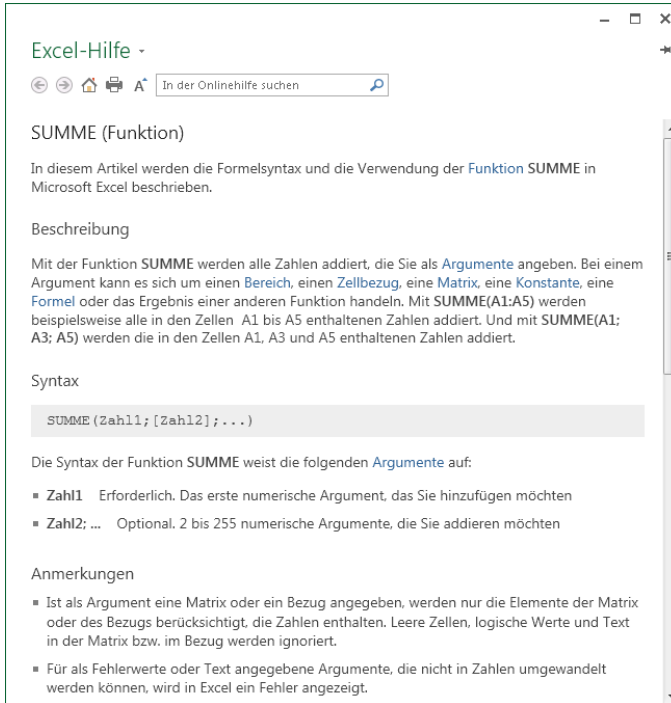


Abbildung 3.16: Der Hilfetext zur Funktion *SUMME()* in Excel 2013

Die Abbildung 3.16 zeigt die Onlinehilfe, d.h., wenn Sie die Hilfe anfordern, wird der Inhalt auf Ihrem PC kurz mit dem Microsoft Office-Bereich im Web abgeglichen und dargestellt. So haben Sie immer den aktuellen Stand der Hilfe von den Microsoft Office-Webseiten. Ist Ihr PC einmal nicht mit dem Internet verbunden, erscheint der lokal gespeicherte Stand mit dem entsprechenden Hinweis darauf (Abbildung 3.17).

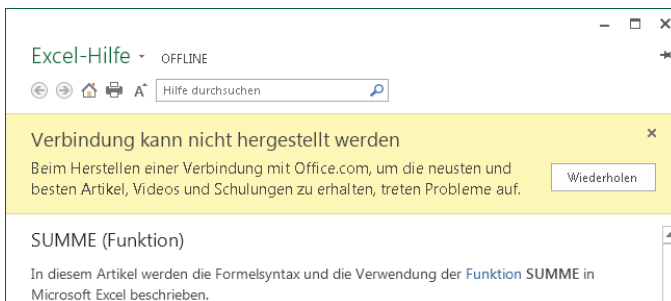


Abbildung 3.17: Der Hinweis, dass dies ggfs. nicht der aktuelle Hilfetext zur Funktion *SUMME()* ist

Verschachtelte Funktionen eingeben

Wenn Sie eine Funktion innerhalb einer anderen Funktion als Argument eingeben, spricht man von einer verschachtelten Funktion. Eine unmittelbare Schaltfläche zur Eingabe verschachtelter Funktionen gibt es nicht.

Beispiel: Sie möchten den Sinuswert für einen Winkel berechnen. Der Winkel steht in der Maßeinheit Grad in Ihrer Tabelle (Abbildung 3.18). Da die Sinus-Funktion von Excel den Winkel in der Maßeinheit Bogenmaß erwartet, müssen Sie die Funktion `BOGENMASS()` einsetzen. Durch den Einsatz beider Funktionen gelangen Sie zu einer verschachtelten Eingabe.

	A	B	C	D
1				
2		Winkel	150 °	
3				
4		Sinus	?	
5				

Abbildung 3.18: Die Gradangabe muss für die Sinus-Funktion erst in das Bogenmaß umgerechnet werden

Verfahren Sie zur Eingabe wie folgt:

1. Markieren Sie die Zelle, in welche die Formel eingegeben werden soll, hier ist es *C4*. Wählen Sie den Befehl *Funktion einfügen*.
2. Im Listenfeld *Kategorie auswählen* markieren Sie den Eintrag *Math. & Trigonom.* Anschließend markieren Sie in der Liste *Funktion auswählen* den Eintrag *SIN* und klicken auf *OK*.
3. Wenn Sie in ein Eingabefeld anstatt eines Arguments eine Funktion eingeben möchten, öffnen Sie das Kombinationsfeld der Funktionspalette. Die Funktionspalette befindet sich während der Funktionseingabe am linken Ende der Bearbeitungsleiste (an der Stelle, an der sich sonst das Namenfeld befindet). Hier wählen Sie die Funktion *BOGENMASS* (Abbildung 3.19), wodurch diese in die *SIN*-Funktionsklammern eingefügt wird.

Hinweis

Sollte die Funktion nicht in der Liste zu finden sein, erhalten Sie diese über *Weitere Funktionen* angezeigt. Die Liste zeigt immer nur die zuletzt verwendeten Funktionen, sodass Sie eventuell die Funktion *BOGENMASS* nicht direkt auswählen können.

4. Das Dialogfeld für den Assistenten zeigt nun die Funktion *BOGENMASS* an. Den Stand der Formel sehen Sie in der Bearbeitungsleiste (siehe dazu die Abbildung 3.20).

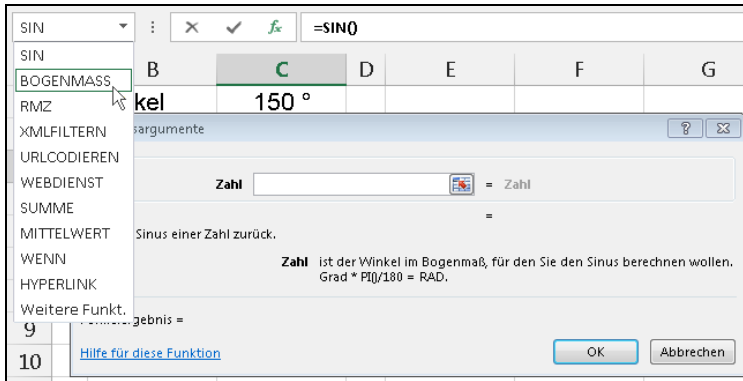


Abbildung 3.19: Das Argument ist wiederum eine Funktion

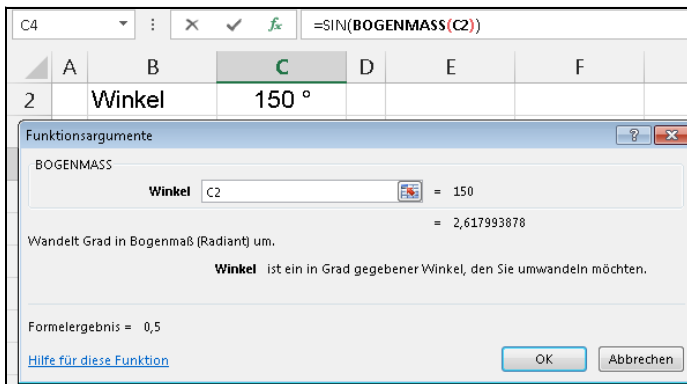


Abbildung 3.20: In der Bearbeitungsleiste ist die Verschachtelung zu sehen

5. Geben Sie nun für die Funktion BOGENMASS als Argument den Zellbezug C2 ein. Schließen Sie mit OK.

Die fertige Formel lautet



`=SIN(BOGENMASS(C2))`

Für einen Winkel von 150° ermittelt sie den Sinuswert 0,5.

Ein OK in der zweiten verschachtelten Funktion beendet die Formeleingabe! Wenn Sie bei verschachtelten Funktionen im Assistenten zur ersten Funktion zurückzukehren müssen, weil hier noch nicht alle Argumente ausgefüllt sind, klicken Sie in der Bearbeitungsleiste auf den entsprechenden Funktionsnamen. Im Dialogfeld bekommen Sie dann die Argumente für diese Funktion angezeigt und haben Gelegenheit, die fehlenden Argumente einzugeben.

Wichtig

Bearbeitung von Formeln

Sie können natürlich jede in eine Formel eingegebene Funktion manuell bearbeiten. Änderungen schließen Sie mit der -Taste ab. Zur unmittelbaren Bearbeitung in der Zelle führen Sie einen Doppelklick auf die entsprechende Zelle aus oder drücken die -Taste. Ebenso können Sie in die Bearbeitungsleiste klicken und dort die Formel bearbeiten.

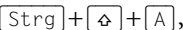
Zur Bearbeitung von Formeln mit Funktionen können Sie auch den Funktions-Assistenten verwenden. Dazu klicken Sie in der Bearbeitungsleiste auf die Schaltfläche *Funktion einfügen* oder rufen den gleichnamigen Befehl in der Registerkarte *FORMELN* auf. Nehmen Sie alle erforderlichen Änderungen in den Argumenten vor. Danach beenden Sie das Dialogfeld mit einem Klick auf die Schaltfläche *OK*.

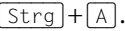
Beachten Sie auch die weitergehenden Hinweise zum Eingeben, Testen und Modifizieren von Formeln in Kapitel 2.


Tipps & Tricks

Zum Abschluss dieses Kapitels wollen wir Ihnen noch drei Tipps bzw. Tricks im Zusammenhang mit der Eingabe von Funktionen vermitteln. Wir nutzen die beschriebenen Verfahren seit Jahren und möchten sie Ihnen nicht vorenthalten.

Argumente anzeigen

Wenn Sie bei der Eingabe von Formeln die zugehörigen Argumente vergessen haben, können Sie diese in der Bearbeitungsleiste anzeigen lassen. Wenn Sie beispielsweise die Funktion *SVERWEIS()* verwenden wollen, tragen Sie das Gleichheitszeichen und den Text »*SVERWEIS*« sowie eine öffnende Klammer ein. Drücken Sie nun die Tastenkombination , werden die Argumente zu dieser Funktion in die Bearbeitungsleiste eingetragen.

Wollen Sie statt der Argumente den Funktions-Assistenten verwenden, drücken Sie die Tastenkombination . Daraufhin wird der Funktions-Assistent für die bereits eingetragene Funktion gestartet.

Hinweis Seit Excel 2002 werden die Argumente zur eingetragenen Funktion auch in einer QuickInfo angezeigt (siehe die Abbildung 3.6 auf Seite 121). Voraussetzung dafür ist, dass in den *Excel-Optionen* das Kontrollkästchen *QuickInfos für Funktionen anzeigen* aktiv ist. Ab Excel 2007 können Sie Funktionsnamen auch schnell aus einem Listenfeld auswählen. Markieren Sie die gewünschte Funktion und drücken dann die -Taste.

Eine komplexe Formel in allen Mappen verfügbar machen

Vielleicht haben Sie das schon erlebt: Sie benötigen in jeder neuen Mappe einen Satz von aufwendigen Formeln. Um sich Schreibarbeit zu sparen, halten Sie in solchen Fällen eine Beispielmappe offen, um aus dieser die benötigten Formeln zu kopieren.

Sie können aber auch die Formeln unter Verwendung der AutoKorrektur speichern. Das eigentliche Einsatzgebiet des *AutoKorrektur*-Features ist das Korrigieren häufiger Schreibfehler und dürfte Ihnen vermutlich eher aus Microsoft Word geläufig sein. Nutzen Sie die AutoKorrektur, um eine Zeichenfolge durch eine Formel ersetzen zu lassen. Das ist besonders dann sinnvoll, wenn es sich um eine Funktion handelt, die komplexer ist und die vom Aufbau her allgemein verwendet werden kann.

Für die Eingabe der häufig verwendeten Tabellenfunktion SUMME() stellt Excel ein eigenes Symbol (*AutoSumme*) bereit. Über die Tastenkombination **ALT** + **⇧** + **=** ist die Funktion auch für Tastaturfreunde schnell verfügbar. Excel trägt dabei auch gleich einen Zellbezug ein und geht dabei so vor: Befinden sich oberhalb der aktiven Zelle Zahlen, wird der Bezug in diese Richtung zeigen. Findet Excel nach oben keine Zahlen, werden die links von der aktiven Zelle liegenden Zellen durchsucht und der Bezug eingetragen. Wenn Sie häufig den gleichen Tabellenaufbau haben, suchen Sie vielleicht nach einer Möglichkeit, wie Sie die Summe für die unterhalb liegenden 500 Zellen schnell erstellen können, ohne lange über den Bezug nachdenken zu müssen.

Die folgende Formel berechnet die Summe der 500 unterhalb der aktiven Zelle liegenden Zellen:

```
=SUMME(INDIREKT(ADRESSE(ZEILE()+1;SPALTE());4;WAHR);WAHR):INDIREKT(ADRESSE(ZEILE()+501;SPALTE());4;WAHR);WAHR))
```

Um diese Formel global zur Verfügung zu stellen und auch schnell eintragen zu können, gehen Sie so vor:

1. Wählen Sie in den *Excel-Optionen* die Kategorie *Dokumentprüfung* und klicken auf die Schaltfläche *AutoKorrektur-Optionen*. Tragen Sie auf der Registerkarte *AutoKorrektur* im Eingabefeld *Ersetzen eine Zeichenfolge* ein.

Beachten Sie, dass es sich um eine eindeutige Zeichenfolge handeln sollte, da die AutoKorrektur-Liste von allen Office-Programmen verwendet wird. Wählen Sie z.B. eine Zeichenfolge mit einem Sonderzeichen, das selten verwendet wird. Eine solche Zeichenfolge könnte z.B. *_sum500* sein.

Wichtig

2. Im Eingabefeld *Durch* tragen Sie die oben gezeigte Formel ein. Achten Sie darauf, dass im Eingabefeld *Durch* ein Gleichheitszeichen vor die Zeichenfolge geschrieben werden muss!
3. Klicken Sie auf die Schaltfläche *Hinzufügen* und schließen Sie mit *OK* ab.

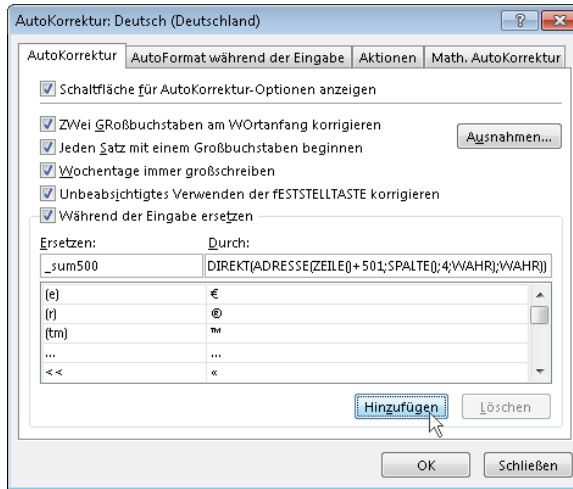


Abbildung 3.21: Eine Formel über die AutoKorrektur schnell verfügbar machen

Wurde der AutoKorrektur-Eintrag richtig festgelegt, müssen Sie nicht einmal mehr ein Gleichheitszeichen eingeben, die Eingabe der Zeichenfolge `_sum500` genügt.

Teilberechnungen in Formeln

Bei der Analyse komplexerer Formeln und bei der Fehlersuche können Sie einzelne Bestandteile der Formel, z.B. Funktionsargumente in Form von Bezügen oder geschachtelten Funktionsaufrufen, ausrechnen lassen. Anstelle des Bezugs oder des Funktionsaufrufs steht dann der Wert. Und das geht so:

1. Öffnen Sie die Formelzelle mit `[F2]` oder per Doppelklick zum Bearbeiten.
2. Markieren Sie den Teilbereich der Formel, dessen Ergebnis berechnet werden soll.
3. Drücken Sie nun die `[F9]`-Taste. Excel ersetzt den markierten Teil der Formel durch das Ergebnis.

Auf diese Art und Weise können Sie beispielsweise eine Formel schrittweise auflösen und anhand der Teilschritte die Formel analysieren.


Hinweis Natürlich berechnet Excel das Ergebnis nur dann, wenn der Inhalt der Markierung berechenbar ist. Ansonsten werden anstelle der Markierung die entsprechenden Fehlermeldungen notiert. Wenn Sie nichts markieren und nur die Einfügemarke in der geöffneten Formel blinkt, ersetzt Excel die gesamte Formel durch das Ergebnis.

Achtung Im Regelfall brechen Sie diese Änderungen in der Formel mit `[Esc]` ab, da Sie die Formel nicht verändern wollen. Sollen die Werte anstelle der Rechenausdrücke in der Formel stehen, müssen Sie mit der `[↵]`-Taste abschließen.

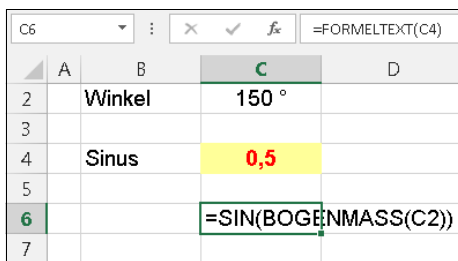
Dokumentation von Formeln in Excel 2013

Im Kapitel 2 haben Sie bereits gesehen, wie Sie von der Ergebnis- in die Formelansicht und zurück schalten können (Registerkarte *FORMELN/Formeln anzeigen*). Für die Dokumentation von Formeln ist das ggfs. ungeeignet. Deshalb haben wir in den Excel-Versionen bis 2010 Formeln oft auch aus der Bearbeitungsleiste in die Zwischenablage kopiert und in einer anderen Zelle mit voran gestelltem Hochkomma (') wieder eingefügt. Dadurch, dass die kopierte Formel nicht mehr mit einem Gleichheitszeichen beginnt, wird die kopierte Formel als Text angezeigt.

Seit Excel 2013 haben Sie ein Alternative – die neue Funktion FORMELTEXT(). Die Anwendung ist ganz einfach:

1. Markieren Sie eine Zelle, wo Sie eine Formel dokumentieren wollen.
2. Schreiben Sie =FORMELTEXT(und geben dann den Bezug der betreffenden Formelzelle an.
3. Tippen Sie die schließende Funktionsklammer ein und beenden Sie mit der -Taste.

In Abbildung 3.22 sehen Sie ein Beispiel. Die Formel =FORMELTEXT(C4) dokumentiert in der Zelle C6 die Formel aus C4. Selbstredend muss für längere Formeln im Bereich rechts von FORMELTEXT()-Berechnung freier Platz sein.



	A	B	C	D
2		Winkel	150 °	
3				
4		Sinus	0,5	
5				
6			=FORMELTEXT(C4)	
7				

Abbildung 3.22: Eine Formel mithilfe der neuen Excel 2013-Funktion FORMELTEXT() im Blatt dokumentieren

Das Ganze hat sogar den Vorteil, dass Sie im Idealfall mehrere Formeln durch Kopieren der Dokumentationsformel schnell darstellen können und geänderte Formeln immer automatisch auch mit der Änderung dokumentiert sind.

Mehr zur Funktion FORMELTEXT() erfahren Sie in Kapitel 9 zu den Nachschlage- und Verweisfunktionen.

Hinweis

Teil B

Aufbauwissen – Excel setzt keine Grenzen

Kapitel 4	Funktionen im besonderen Einsatz	137
Kapitel 5	Benutzerdefinierte Funktionen	169

Kapitel 4

Funktionen im besonderen Einsatz

Funktionen in Namen	138
Funktionen in der bedingten Formatierung	143
Funktionen bei der Datenprüfung	163

In diesem Kapitel wollen wir auf die Nutzung der Arbeitsblattfunktionen im Zusammenhang mit anderen Excel-Features eingehen. Im ihrem Zusammenwirken sind die Excel-Programmfunktion stark. Durch das Unterlegen mit Arbeitsblattfunktionen werden die Möglichkeiten so erweitert, dass kaum Wünsche offen bleiben.

Schauen Sie sich an, welche Rolle die Funktionen im Zusammenhang mit

- ▶ Namen,
- ▶ bedingten Formatierungen und
- ▶ Gültigkeitsprüfungen

spielen können.

Funktionen in Namen

Excel bietet nicht nur zahlreiche Funktionen für die Berechnung unterschiedlichster Sachverhalte, sondern stellt für die Angabe der Argumente in diesen Funktionen auch verschiedene Möglichkeiten zur Verfügung. In Kapitel 2 haben Sie die verschiedenen Schreibweisen von Bezügen kennen gelernt. Neben der Verwendung von Zahlen und relativen, gemischten und absoluten Bezügen ist hier die Verwendung von Namen für Zellen oder Zellbereiche die herausragende Möglichkeit.

Ein Name kann aber mehr sein als nur ein Zellbezug. Er kann für

- ▶ einen Bezug,
- ▶ eine Konstante oder
- ▶ eine Formel

verwendet werden.

Hinweis Wie Sie Namen festlegen und welche Vorteile die Verwendung von Namen bietet, können Sie ausführlich in »Microsoft Office Excel 2010 – Das Handbuch«, Microsoft Press, ISBN-13: 978-3-86645-142-1, oder in »Microsoft Office Excel 2013 – Das Handbuch, ISBN-13: 978-3-86645-158-2, nachlesen.

Aktuelle Informationen abrufen

Für gespeicherte Arbeitsmappen können Sie mit einer Informationsfunktion (siehe Kapitel 10) Pfad, Datei- und Blattname abrufen. Die Formel lautet:

```
=ZELLE("Dateiname")
```

Das Ergebnis enthält allerdings alle Bestandteile in einer einzigen Zeichenfolge, etwa so:

```
C:\MSPress\Excel-Funktionen\Kap04\[Kap04.xlsx]Funktion_Namen
```

Das Instrumentarium zum Ab- oder Ausschneiden der einzelnen Bestandteile liefern uns die Textfunktionen aus Kapitel 7. Dies könnte z.B. in Formularen nützlich sein, in denen Sie diese Information im Arbeitsblatt selbst und nicht in der Kopf- oder Fußzeile unterbringen wollen, wo Excel eigene Textbausteine dafür hat: `&[Pfad]`, `&[Datei]` und `&[Register]`.

Die gezeigten Berechnungen funktionieren nur in bereits gespeicherten Arbeitsmappen, weil die Berechnung `=Zelle("Dateiname")` bei ungespeicherten Mappen einen nicht weiter auswertbaren Leerstring ("") ergibt.

Pfad der aktuellen Mappe abrufen

Den Beginn der Zeichenfolge macht der Pfad der Arbeitsmappe. Er endet vor der öffnenden eckigen Klammer »[«. Also schneiden Sie den Pfad an dieser Stelle ab:

```
=LINKS(ZELLE("Dateiname");FINDEN("[";ZELLE("Dateiname"))-1)
```

Die Funktion `FINDEN()` ermittelt die Positionsnummer der öffnenden Klammer und `LINKS()` ermittelt die Zeichenfolge bis zur Position vor dieser Klammer.

1. Klicken Sie im Menüband auf der Registerkarte *FORMELN* auf das Symbol *Namen definieren* (Abbildung 4.2).
2. Tippen Sie im Textfeld *Name* des Dialogfelds den Text »Pfad« ein.
3. Im Feld *Bezieht sich auf* geben Sie die oben abgebildete Formel ein und klicken auf *OK*.

Die Abbildung 4.1 zeigt das Dialogfeld *Neuer Name* in Excel 2013.

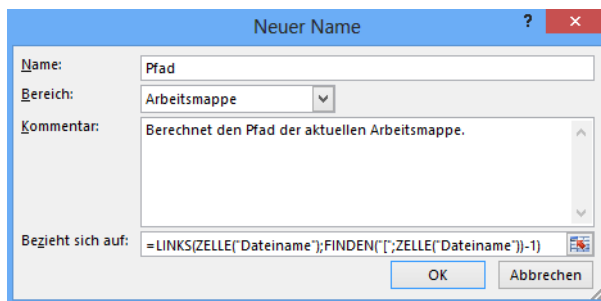


Abbildung 4.1: Für den Namen *Pfad* wird eine Berechnung hinterlegt

Ab sofort können Sie in dieser Arbeitsmappe den aktuellen Pfad der Datei mit der Formel `=Pfad` abrufen.

Dateiname der aktuellen Mappe abrufen

Den Dateinamen zu extrahieren erfordert etwas mehr Aufwand. Gefragt ist die Zeichenfolge zwischen den eckigen Klammern:

```
=TEIL(ZELLE("Dateiname");FINDEN("[";ZELLE("Dateiname"))+1;  
FINDEN("]";ZELLE("Dateiname"))-(FINDEN("[";ZELLE("Dateiname"))+1))
```

Mit der Funktion `TEIL()` schneiden Sie den Dateinamen heraus, wobei die Argumente *Erstes_Zeichen* und *Anzahl_Zeichen* durch Ermitteln der beiden Klammernpositionen wieder mit der Funktion `FINDEN()` erfolgt.

Rufen Sie erneut den Befehl *Namen definieren* auf. Tippen Sie in das Listenfeld *Name* den Text »Dateiname« ein. Ins Feld *Bezieht sich auf* tragen Sie bitte die oben abgebildete Formel ein und klicken auf *OK*.

Ab sofort können Sie in dieser Arbeitsmappe den Dateinamen mit der Formel =Dateiname abrufen.

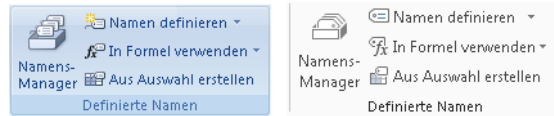


Abbildung 4.2: Die Gruppe *Definierte Namen* auf der Registerkarte *FORMELN* in Excel 2007 (links) und Excel 2013 (rechts)

Tip Die Funktion ZELLE() funktioniert auch in der deutschen Excel-Version mit dem Parameter "*File-name*" anstelle des deutschen Wortes "*Dateiname*". Dies könnte für Sie von Bedeutung sein, wenn Sie Ihre Arbeitsmappen an Personen, welche mit einem englisch eingestellten Excel arbeiten, weitergeben. Der deutsche Parameter würde bei diesen einen Fehler erzeugen.

Den aktuellen Blattnamen abrufen

Den aktuellen Blattnamen schneiden Sie entweder mit der Funktion RECHTS() oder ebenfalls mit der Funktion TEIL() hinten ab. Bei der Informationsfunktion ZELLE() müssen Sie allerdings eine kleine Erweiterung vornehmen, damit der Name *Blattname* auch wirklich auf jedem Arbeitsblatt die Registerbeschriftung ausgibt. Im zweiten, optionalen Argument muss ein Bezug auf eine beliebige Zelle des Blatts enthalten sein.

Damit Excel Bezüge wie »A1« nicht automatisch um den Blattnamen erweitert, etwa »Tabelle1!A1«, verwenden Sie die Funktion INDIREKT() für diesen Blattzellenbezug. Wir entscheiden uns hier für den Einsatz der TEIL()-Funktion, weil diese kürzer ausfällt als die Variante mit der RECHTS()-Funktion:

```
=TEIL(ZELLE("Dateiname";INDIREKT("A1"));FINDEN("]";ZELLE("Dateiname";INDIREKT("A1")))+1;255)
```

Mit der Funktion TEIL() schneiden Sie den Blattnamen heraus, wobei das Argument *Erstes_Zeichen* durch Ermitteln der Position der schließenden Klammer mit FINDEN() erfolgt. Für das Argument *Anzahl_Zeichen* setzen Sie einfach die Zahl 255 ein. Damit haben Sie eine maximale Länge abgedeckt.

Rufen Sie noch einmal den Befehl *Namen definieren* auf. Tippen Sie in das Listenfeld *Name* den Text »Blattname« ein. In das Feld *Bezieht sich auf* tragen Sie bitte die oben abgebildete Formel ein und klicken auf *OK*.

Ab sofort können Sie in dieser Arbeitsmappe den Blattnamen mit der Formel =Blattname abrufen.

Für die Erstellung, Bearbeitung und Verwaltung gibt es seit Excel 2007 ein neues Instrument – den Namens-Manager (Abbildung 4.3). Die Excel-Gemeinde war sich einig: Es war höchste Zeit, dass die über viele Versionen gleiche Namensverwaltung eine Renovierung bekam. Das Ergebnis kann sich sehen lassen.

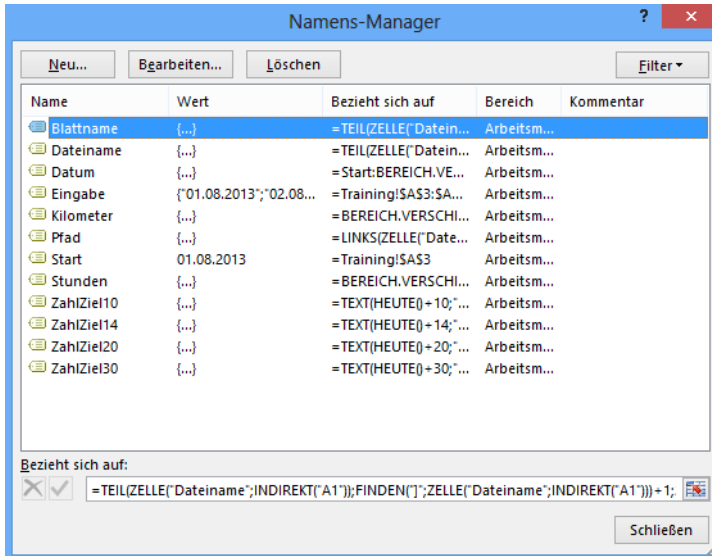


Abbildung 4.3: Der Namens-Manager stellt seit Excel 2007 wesentlich bessere Arbeitsmöglichkeiten mit Namen zur Verfügung

Zahlungsziele als »Textbausteine«

In Ihrer Firma gibt es für die Rechnungslegung verschiedene Zahlungsziele. Dies reicht von sofortiger Fälligkeit über Fristen von 10, 14, 20 bis 30 Tage. Bei der Angabe des jeweiligen Zahlungsziels wollen Sie einfach einen Begriff eingeben, der dann ähnlich einem Textbaustein funktioniert.

Rufen Sie auf der Registerkarte *FORMELN* den Befehl *Namen definieren* auf. Tippen Sie in das Listenfeld *Name* als ersten Text »ZahlZiel10« ein. In das Feld *Bezieht sich auf* tragen Sie die folgende Formel ein:

```
=TEXT(HEUTE()+10;"TT.MM.JJJJ")
```

Bestätigen Sie mit *OK*.

Wiederholen Sie die Prozedur und legen Sie den zweiten Namen »ZahlZiel14« an. Unten im Feld *Bezieht sich auf* notieren Sie die folgende Formel:

```
=TEXT(HEUTE()+14;"TT.MM.JJJJ")
```

Bestätigen Sie wiederum mit *OK*.

Erzeugen Sie dann die Namen »ZahlZiel20« und »ZahlZiel30« mit den Formeln

```
=TEXT(HEUTE()+20;"TT.MM.JJJJ")
```

und

```
=TEXT(HEUTE()+30;"TT.MM.JJJJ")
```

Sie können nun z.B. in einem Rechnungsformular mit der Formel
 ="Bitte überweisen Sie den Rechnungsbetrag bis zum "&ZahlZiel20&"."
 das jeweils gewünschte Zahlungsziel durch Angabe des Namens in der Formel festlegen.



Alle gezeigten Beispiele haben wir in der Beispieldatei *Kap04_Namen.xlsx* auf dem Arbeitsblatt *Fkt_Namen* für Sie zum Nachprüfen und Ausprobieren realisiert. Sie finden die Datei im Ordner *\Ms5-235\Kap04*.

Dynamische Bereichsnamen

Wie leistungsfähig Namen sein können, zeigt das folgende Beispiel: Angenommen, Sie haben eine Liste, in die z.B. jede Woche oder jeden Tag weitere Werte eingetragen werden. Bei den Auswertungen der Liste soll natürlich immer die Liste in voller Länge in die Berechnungen eingehen, der Bezug soll also gleichsam mitwachsen. Wie lässt sich das erreichen?

Nehmen wir an, Sie führen eine Trainingstabelle, in der Sie jeden Tag Trainingsdaten eintragen. In Spalte *A* steht das Datum, in Spalte *B* die Zeitdauer und in Spalte *C* die gelaufene Strecke. Jedes Kalenderjahr erhält ein eigenes Arbeitsblatt. Für ein Jahr liegt der maximale Bereich inklusive Überschrift und Berücksichtigung von Schaltjahren also bei 367 Zeilen. Legen Sie einen Namen fest, der immer nur auf den verwendeten Bereich zeigt.

Auch hier liegt die Lösung in der Vergabe von *Namen*:

1. Vergeben Sie für den Datenbereich *A3:A368* den Namen »Eingabe«.
2. Legen Sie dann für die Zelle *A3* den Namen »Start« fest.
3. Legen Sie den Namen »Datum« fest. Unter *Bezieht sich auf* tragen Sie die folgende Formel ein:
`=Start:BEREICH.VERSCHIEBEN(Start;MAX(0;ANZAHL(Eingabe)-1);0)`
4. Verwenden Sie für den Namen »Stunden« den Bezug
`=BEREICH.VERSCHIEBEN(Start;0;1):BEREICH.VERSCHIEBEN(Start;MAX(0;ANZAHL(Eingabe)-1);1)`
5. Verwenden Sie für den Namen »Kilometer« den Bezug
`=BEREICH.VERSCHIEBEN(Start;0;2):BEREICH.VERSCHIEBEN(Start;MAX(0;ANZAHL(Eingabe)-1);2)`

Um einen dynamischen Bereich zu ermitteln, können Sie in Excel die Funktion `BEREICH.VERSCHIEBEN()` verwenden. Diese Funktion liefert einen Bezug, der gegenüber dem angegebenen Bezug versetzt ist. Die Funktion `BEREICH.VERSCHIEBEN(Bezug;Zeilen;Spalten;Höhe;Breite)` ermittelt zunächst die Anzahl der (numerischen) Einträge im Eingabebereich und verwendet diese Zahl für die Rückgabe des neuen Bereichs.

Wichtig ist es, festzuhalten, dass die Namen *Stunden* und *Kilometer* auf Basis der Einträge in der Spalte *Datum* ermittelt werden. In diesem Beispiel müssen die Einträge also immer mit einem Datum versehen sein.

	A	B	C	D	E	F
1	Trainingsliste 2013					
2	Datum	Stunden	Km		Anz. Trainings	18
3	01.08.2013	1,8	9,0		Ges. Stunden	23,4
4	02.08.2013	1,4	7,0		Ges. Km	117,0
5	03.08.2013	1,2	6,0			
6	04.08.2013	1,8	9,0			
7	05.08.2013	0,5	2,5			
8	06.08.2013	1,4	7,0			
9	07.08.2013	1,0	5,0			
10	08.08.2013	0,7	3,5			
11	09.08.2013	1,2	6,0			
12	10.08.2013	1,4	7,0			
13	11.08.2013	1,1	5,5			
14	12.08.2013	1,4	7,0			
15	13.08.2013	0,7	3,5			
16	14.08.2013	1,8	9,0			
17	15.08.2013	1,4	7,0			
18	16.08.2013	1,8	9,0			
19	17.08.2013	1,4	7,0			
20	18.08.2013	1,4	7,0			
21						
22						

Abbildung 4.4: Die Namen zeigen auf den erweiterten Bereich, wenn weitere Werte eingetragen werden

Namen für konstante Werte oder Formeln werden im Namenfeld und im Dialogfeld *Gehe zu* nicht angezeigt. Gleichwohl können Sie Namen, die einen Bezug zurückgeben, hier eintragen und damit den Bezug überprüfen. Um die Namen zu testen, tragen Sie diese im Namenfeld ein oder rufen Sie über mit **[F5]** das *Gehe zu*-Dialogfeld auf und tragen den Bezug »Datum« oder »Kilometer« ein.

Hinweis

Fügen Sie zum Test neue Einträge hinzu. Rufen Sie mit der **[F5]**-Taste den Befehl *Gehe zu* auf und markieren den Bezug erneut, werden Sie feststellen, dass der Name nun auf den erweiterten Bereich zeigt. Sie sehen außerdem in den Berechnungen im Kopf der Liste immer die aktuellen Werte.

Alle gezeigten Beispiele haben wir in der Beispieldatei *Kap04_Namen.xlsx* auf dem Arbeitsblatt *Training* für Sie zum Nachprüfen und Ausprobieren realisiert. Sie finden die Datei im Ordner *(Ms5-235)\Kap04*.



Mit einem dynamischen Namen haben Sie auch eine ideale Grundlage für ein dynamisches Diagramm.

Tipp

Funktionen in der bedingten Formatierung

Sicher wissen Sie schon, dass man bei der Festlegung von bedingten Formatierungen auch Formeln verwenden kann. Um eine Formel als Formatierungskriterium zu verwenden, rufen Sie im Menüband auf der Registerkarte *START* den Befehl *Bedingte Formatierung/Neue Regel* auf und ändern im Dialogfeld *Neue Formatierungsregel* den Eintrag *Formel zur Ermittlung der zu formatierenden Zellen verwenden* (Abbildung 4.5). Das Ergebnis der Formel muss den Wahrheitswert *WAHR* oder *FALSCH* zurückgeben. Wird *WAHR* zurückgegeben, erfolgt die bedingte Formatierung gemäß Ihren Festlegungen.

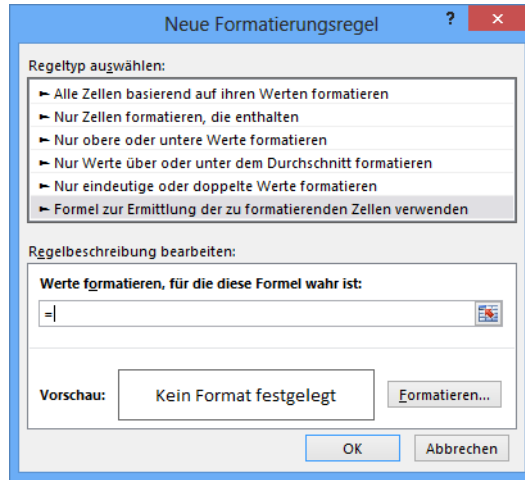


Abbildung 4.5: Das Dialogfeld *Neue Formatierungsregel* in Excel 2013

Mit der Möglichkeit, anstelle von konstanten Vergleichswerten oder Zellbezügen eine Formel zu verwenden, steht Ihnen auch bei der bedingten Formatierung nahezu die gesamte Welt der Excel-Funktionen zur Verfügung. Die erfreuliche Konsequenz: Die bedingte Formatierung wird zu einem schier unendlichen Testgebiet, wenn es darum geht, Zellen unter den unterschiedlichsten Bedingungen automatisch bestimmte, zuvor festgelegte Formate zuzuweisen. Die nachfolgenden Beispiele sollen Sie auf Ideen bringen und Ihnen ggf. sogar fertige Lösungen an die Hand geben.



Die bedingte Formatierung hat mit Excel 2007 einige Neuerungen erfahren. Hier wurde die Grenze von maximal drei Bedingungen aufgehoben. Sie können jetzt beliebig viele bedingte Formatierungen für einen Bereich festlegen. Die Grenze wird nur durch Ihren Arbeitsspeicher gesetzt.

Mit Datenbalken oder Farbskalen können Sie innerhalb einer Zelle bestimmte Zustände signalisieren. Mit den neuen Symbolen haben Sie ebenfalls neue, vielfältige Möglichkeiten, z.B. für Ampelfunktionen. Außerdem gibt es einen Manager zum Verwalten der bedingten Formatierungen.

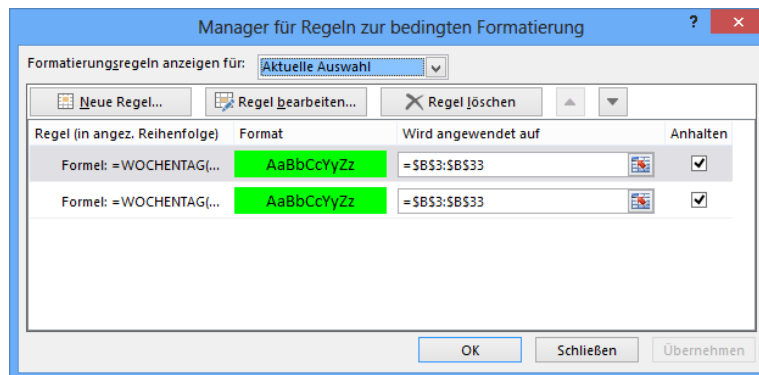


Abbildung 4.6: Der Regel-Manager für die bedingte Formatierung in Excel 2013

In der Beispielmappe *Kap04_BF.xlsx* haben wir beim weiter unten folgenden Temperaturbeispiel und auf dem Extrablatt *Neu ab 2007* die Möglichkeiten illustriert.

Eine umfassende Beschreibung der neuen bedingten Formatierung in Excel 2010 bietet Ihnen der Titel »Microsoft Excel 2010 – Das Handbuch«, Microsoft Press, ISBN-13: 978-3-86645-142-1.

Die Wochenenden farbig hervorheben

In einer Spalte mit fortlaufenden Datumsangaben sollen die auf Samstag und Sonntag fallenden Tage als Wochenendtage gekennzeichnet werden. Sie können sich hier die Excel-Funktion WOCHENTAG() zunutze machen, die zu einem Datumswert jeweils den Wochentag berechnet und in Kapitel 6 genauer erklärt wird. Erforderlich ist nur ein Argument, nämlich das Datum. Die Funktion WOCHENTAG() liefert als Ergebnis eine Zahl zwischen 1 und 7. Dabei entspricht *Sonntag* der Zahl 1 und *Samstag* der Zahl 7 (Montag würde also der Ziffer 2 entsprechen, Mittwoch der Nummer 4 usw.).

Mit diesem Wissen ausgerüstet, können Sie dafür sorgen, dass in der markierten Datumsreihe die Wochenenden einen hellgrünen Hintergrund bekommen. Nehmen wir an, Sie haben im Bereich von B3:B33 die Datumsangaben für einen Monat stehen. Markieren Sie diesen Datumsbereich und gehen dann so vor:

1. Rufen Sie für die markierte Datumsliste auf der Registerkarte *START* den Befehl *Bedingte Formatierung* auf und ändern Sie unter *Neue Regel* den Eintrag *Formel zur Ermittlung der zu formatierenden Zellen verwenden*.
2. Tragen Sie nun Eingabefeld folgende Formel ein: `=Wochentag(B3)=1`. Diese Formel prüft, ob der Wochentag der jeweils aktuellen Zelle – daher auch der relative Bezug – auf den Sonntag (also die 1) fällt. Trifft dies zu, wird auch die noch festzulegende Formatierung angewandt.
3. Klicken Sie auf die Schaltfläche *Formatieren* und wählen Sie auf der Registerkarte *Ausfüllen* beispielsweise die Farbe *Hellgrün* aus. Bestätigen Sie die Eingabe des Formats und die Definition der Bedingung jeweils mit *OK*.
4. Wiederholen Sie die beiden letzten Schritte, indem Sie erneut den Befehl *Neue Regel* wählen. Die Formel muss dieses Mal prüfen, ob das Datum der jeweils aktuellen Zelle auf einen Samstag fällt, der für Excel der Tag mit der Nummer 7 in der Woche ist. Daher muss die Formel so lauten: `=Wochentag(B3)=7`.

Sie sehen, aus einer Kombination von bedingter Formatierung und verschiedenen Excel-Funktionen lassen sich schnell und einfach gesuchte Informationen optisch hervorheben.

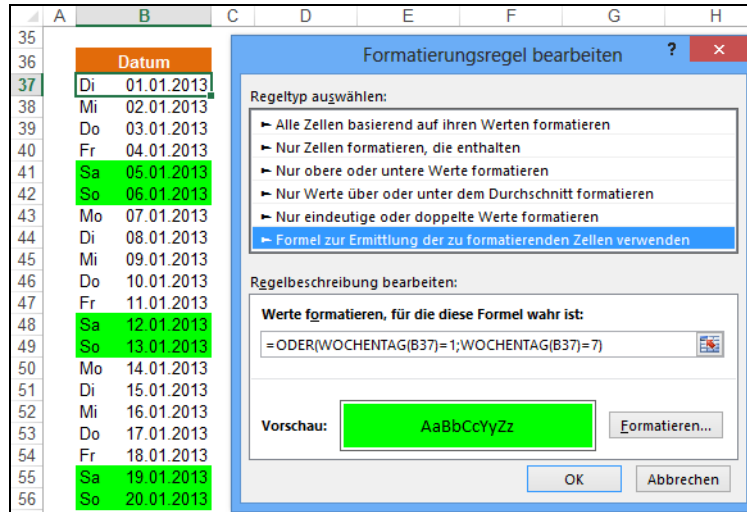


Abbildung 4.7: Verwendung der Funktion *WOCHENTAG()* in der bedingten Formatierung

Wenn Sie für beide Tage nur eine Farbe angeben wollen, ist es eigentlich nicht nötig, zwei (der drei möglichen) Bedingungen zu verwenden. Mit der bekannten Logik-Funktion *ODER()* (siehe Kapitel 8) benötigen Sie nur eine Bedingung:

=ODER(WOCHENTAG(B37)=1;WOCHENTAG(B37)=7)

Die Abbildung 4.7 zeigt das vollständig ausgefüllte Dialogfeld und das Resultat für die Datumsliste. Wenn wir einmal dabei sind, über alternative Möglichkeiten nachzudenken, hier gleich noch eine andere Variante, die sich die Möglichkeiten der Text-Funktion *TEXT()* (siehe Kapitel 7) zunutze macht:

=ODER(TEXT(B72;"TTT")="So";TEXT(B72;"TTT")="Sa")

Die Tatsache, dass die Formatanweisung »TTT« bei der Umwandlung des numerischen Datumsausdrucks in Text die zweibuchstabigen Wochentage ergibt, können Sie natürlich für den bei dieser Aufgabe notwendigen Vergleich nutzen. Vielleicht ist diese Variante sogar besser lesbar für Sie, da Sie bei der *WOCHENTAG()*-Funktion schon wissen müssen, wofür die Wochentagsnummern stehen.



Die gezeigten Beispiele haben wir in der Beispieldatei *Kap04_Bf.xlsx* auf dem Arbeitsblatt *Wochenende* für Sie zum Nachprüfen und Ausprobieren realisiert. Sie finden die Datei im Ordner `\Ms5-235\Kap04`.

Wochenenden und Feiertage hervorheben

Farbig hervorgehobene Wochenenden in Kalenderübersichten sind gut, aber viele Anwender werden sofort fragen: »Und was ist mit den Tagen, an denen ebenfalls nicht gearbeitet wird – den Feiertagen, den sogenannten Brückentagen im Umfeld von Feiertagen und Wochenenden oder bestimmten firmeninternen festgelegten arbeitsfreien Tagen?« Möglicherweise besteht

sogar darüber hinaus ein Bedarf, andere wichtige Tage automatisch hervorheben zu lassen, wenn diese auf ein bestimmtes Datum oder einen bestimmten Wochentag fallen – etwa Termine für bestimmte Überweisungen, Liefertermine, Wartungstermine, monatliche oder quartalsweise Berichtstermine usw.

In jedem Fall werden solche Termine in Listen erfasst und können damit von Excel berücksichtigt werden.

In unserem Beispiel sollen neben den Wochenenden zwei weitere Anforderungen abgearbeitet werden: Feiertage und andere arbeitsfreie Tage. Damit müssen insgesamt vier Zustände abgeprüft werden:

- ▶ Samstage/Sonntage
- ▶ Feiertage und sonstige freie Tage

Für diese Lösung benötigen Sie neben der Datums-Funktion WOCHENTAG() noch die Matrix-Funktionen VERGLEICH() sowie die Informations-Funktion ISTZAHL().

Mit Namen arbeiten

Bevor Sie die bedingte Formatierung festlegen, steht ein kleiner Zwischenschritt an: Die »Freie Tage-Liste« muss eingegeben und benannt werden. Die Abbildung 4.8 zeigt den Inhalt und die Position der Liste. Übernehmen Sie diese in Ihr Arbeitsblatt. Natürlich muss die Liste den Zeitraum abdecken, den Sie bedingt formatieren wollen.

	F	G	H
25			
26			
27		Freie Tage	
28		Sa 08.12.2012	
29		Di 25.12.2012	
30		Mi 26.12.2012	
31		Di 01.01.2013	
		So 06.01.2013	

Abbildung 4.8: Die Freie Tage-Liste hat den Namen *FreieTage*

Um nun diesen Bereich in Formeln leichter ansprechen zu können, empfiehlt es sich, ihn mit einem Namen zu versehen. Gehen Sie dazu wie folgt vor:

1. Markieren Sie den Bereich von G27:G31 und klicken Sie links oben in das *Namenfeld*, in dem momentan die Koordinate G27 für die erste markierte Zelle steht.
2. Tippen Sie für den markierten Zellbereich den Namen *FreieTage* ein.
3. Schließen Sie mit der -Taste ab.

Das Ergebnis sehen Sie in Abbildung 4.8. Der markierte Bereich heißt von nun an »Freie-Tage«.

Die Funktion VERGLEICH() einsetzen

Um nun die Datumsangaben daraufhin zu untersuchen, ob sie auf Tage fallen, die im Bereich *FreieTage* enthalten sind, ist die Funktion VERGLEICH() erforderlich. Sie hat die folgende Syntax:

=VERGLEICH(Suchkriterium;Suchmatrix;Vergleichstyp)

Das *Suchkriterium* steht in jeder Zelle der Datumsliste. Die *Suchmatrix* ist der Bereich *FreieTage*. Der *Vergleichstyp* gibt an, wie Excel die Werte in der Matrix mit dem Suchkriterium vergleichen soll. In diesem Beispiel ist der Typ 0 angebracht, der den Index des Werts in der *Suchmatrix* liefert, der **genau** mit dem *Suchkriterium* übereinstimmt. Die Daten im *Suchmatrix*-Bereich können beliebig eingetragen werden und müssen nicht unbedingt sortiert vorliegen.

Suchen und Finden

Die Funktion VERGLEICH() liefert als Ergebnis eine Zahl, nämlich die Position, die der jeweils gefundene Wert innerhalb der *Suchmatrix* einnimmt. Fällt der Vergleich negativ aus – ist also das Datum nicht in der *Freie Tage*-Liste vorhanden –, dann ist das Resultat keine Zahl, sondern der Fehlerwert #NV.

Für die Aufgabenstellung ist damit klar: Wird ein Datum aus der Datumsliste auch in dem Bereich *FreieTage* gefunden, liefert Excel eine Zahl. Bleibt also sicherzustellen, dass die bedingte Formatierung nur angewendet wird, wenn das Ergebnis der Funktion VERGLEICH() eine Zahl ist. Hierfür verwenden Sie die Informationsfunktion ISTZAHL(). Sie überprüft, ob der Wert einer Zelle oder eines Ergebnisses eine Zahl ist. Damit ergibt sich die verschachtelte Formel aus ISTZAHL() und VERGLEICH():

=ISTZAHL(VERGLEICH(B3;FreieTage;0))

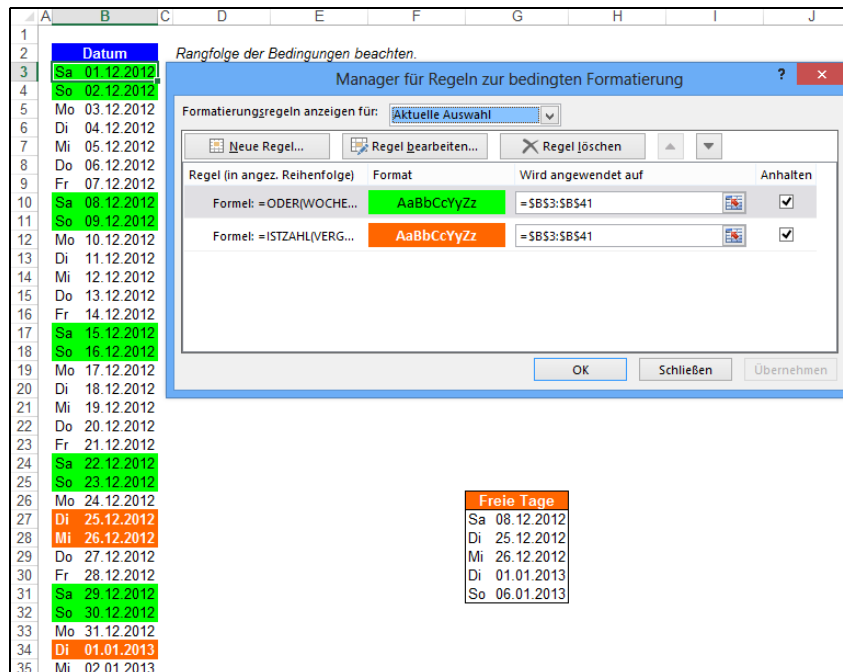


Abbildung 4.9: Zusätzlich zu den Wochenenden sind die freien Tage hervorgehoben

Weisen Sie den freien Tagen eine rote Hintergrundfarbe und als Kontrast eine fette, weiße Schriftfarbe zu. Die Abbildung 4.9 zeigt das fertige Ergebnis. Folgende Schritte sind dazu notwendig:

1. Markieren Sie den Datumsbereich und rufen Sie auf der Registerkarte *START* im Dropdownmenü *Bedingte Formatierung* den Befehl *Regeln verwalten* auf.
2. Lassen Sie die Einstellungen für die bereits vorhandene Regel (Wochenenden) unverändert und klicken Sie auf die Schaltfläche *Neue Regel*.
3. Im Dialogfeld *Neue Formatierungsregel* markieren Sie *Formel zur Ermittlung der zu formatierenden Zellen verwenden*. Im Eingabefeld *Werte formatieren, für die diese Formel wahr ist*: geben Sie die Formel `=ISTZAHL(VERGLEICH(B3;FreieTage;0))` ein.
4. Wechseln Sie anschließend über die Schaltfläche *Formatieren* auf die Registerkarten *Ausfüllen* und *Schrift* und legen Sie dort die Farben fest. Bestätigen Sie die Eingabe des Formats und die Definition der Bedingung jeweils mit *OK*.

Im Ergebnis der Formatierung müssten nun die Datumsangaben 25./26.12.2012 und 01.01.2013 in Rot/Weiß erscheinen.

Die Anweisungen über die Funktion *Bedingte Formatierung* wirken in diesem Beispiel so, dass Feiertage im Kalender nur dann in der Kombination *Rot/Weiß* hervorgehoben werden, wenn sie nicht auf ein Wochenende fallen.

Hinweis

Wollen Sie, dass alle Feiertage – auch, wenn sie auf ein Wochenende fallen – in Rot/Weiß erscheinen, müssen Sie bei den Festlegungen für die bedingte Formatierung auch mit der Formel für diese Tage beginnen. Die erste Bedingung ist dann für die Feiertage zuständig, die zweite Bedingung für die Wochenenden.

Das Beispiel einschließlich des Vergleichs zwischen verschiedenen Reihenfolgen der Bedingungen finden Sie in der Arbeitsmappe *Kap04_BF.xlsx* auf dem Arbeitsblatt *FreieTage*. Die Datei ist im Ordner `\Ms5-235\Kap04` gespeichert.



Am Beispiel der Kalenderübersichten wurde deutlich, dass die Möglichkeiten der Funktion *Bedingte Formatierung* sehr vielseitig sind, wenn Sie die verschiedenen Tabellenfunktionen von Excel nutzen. In dem Bestreben, wichtige Informationen in Arbeitsblättern optisch hervorzuheben, steht Ihrer Kreativität nichts im Wege.

Identische Werte hervorheben

Hin und wieder wäre es nützlich, wenn identische Inhalte signalisiert würden. Die Inhalte von Zellen können Sie mit verschiedenen Funktionen vergleichen. Um die Ergebnisse des Vergleichs von Zellen deutlich hervorzuheben, eignet sich die bedingte Formatierung hervorragend.

Wenn alle gleich sind

Sollen beispielsweise die Zellen im Bereich $B3:B7$ hellblau formatiert werden, wenn die Inhalte aller Zellen gleich sind (Abbildung 4.10), führen Sie die folgenden Schritte aus:

1. Markieren Sie den Bereich $B3:B7$. Wählen Sie unter *Bedingte Formatierung* den Befehl *Neue Regel* und markieren *Formel zur Ermittlung der zu formatierenden Zellen verwenden*.
2. Tragen Sie die Formel $=UND(IDENTISCH(B3; \$B\$3: \$B\$7))$ ein.
3. Legen Sie per Schaltfläche *Formatieren* das gewünschte Format fest und bestätigen Sie die Eingaben mit *OK*.

Die Zellen werden nur formatiert, wenn **alle** Inhalte gleich sind.

	A	B	C	D	E
1					
2		Vergleich aller Inhalte			
3		Computer sind auch nur Menschen.			
4		Computer sind auch nur Menschen.			
5		Computer sind auch nur Menschen.			
6		Computer sind auch nur Menschen.			
7		Computer sind auch nur Menschen.			
8					

Wenn die Inhalte aller Zellen gleich sind, werden alle formatiert.
Formel ist: $=UND(IDENTISCH(B3; \$B\$3: \$B\$7))$

Abbildung 4.10: Alle Inhalte müssen gleich sein, um formatiert zu werden

Die absoluten und relativen Bezüge sind hier und bei den folgenden Beispielen von großer Bedeutung. Wenn Sie festgestellt haben, dass eine Formel für die bedingte Formatierung nicht korrekt arbeitet, rufen Sie diese unter *Bedingte Formatierung/Regeln verwalten* erneut auf und ändern die Formel. Wenn Sie versuchen, mit den Pfeiltasten die Einfügemarke an eine bestimmte Stelle zu bewegen, werden statt der erwarteten Bewegung innerhalb des Eingabefelds andere Zellbezüge eingetragen. Sie können das verhindern, wenn Sie das Eingabefeld für die Formel aktivieren und die Taste $[F2]$ drücken. Sie haben vom »Zeigen-Modus« in den »Bearbeiten-Modus« umgeschaltet. Jetzt funktionieren die Pfeiltasten wie erwartet.

Hinweis In Excel 2007 gab es Probleme mit der bedingten Formatierung und der oben verwendeten Formel: Die bedingte Formatierung markiert nur die identischen Zellen, nicht wie in der Aufgabe beschrieben »alle oder keine«. Hier und bei den zwei folgenden Beispielen verhielt sich die Version 2007 anders als ihre Vorgänger. Sie werten die Formeln, welche als Matrixformeln zu berechnen sind, anders aus.

In den Versionen 2010 und 2013 haben wir wieder das erwartete Verhalten festgestellt. Deshalb finden Sie in der Beispieldatei alternative Berechnungen, welche die Aufgaben auch erfüllen:

$=ZÄHLENWENN(\$B\$3: \$B\$7; \$B\$3) = ANZAHL2(\$B\$3: \$B\$7)$

Wenn nur einer gleich ist

Schauen wir uns einen anderen Fall an: Hier sollen alle Zellen im Bereich *B10:B14* formatiert werden, wenn es im Bereich *B11:B14* eine Zelle gibt, die mit Zelle *B10* übereinstimmt. Führen Sie dazu die folgenden Schritte aus:

1. Markieren Sie den Bereich *B11:B15* und wählen den Befehl *Bedingte Formatierung/Neue Regel*. Wählen Sie *Formel zur Ermittlung der zu formatierenden Zellen verwenden* aus.
2. Tragen Sie die Formel `=ODER(IDENTISCH(B10;B11:B14))` ein.
3. Legen Sie über die Schaltfläche *Formatieren* das gewünschte Format fest.
4. Bestätigen Sie die Eingabe des Formats und die Definition der Bedingung jeweils mit *OK*.

Als Ergebnis werden alle Zellen formatiert, wenn es im Bereich *B11:B14* eine Zelle mit dem gleichen Inhalt wie in Zelle *B10* gibt (Abbildung 4.11).

	A	B	C	D	E
8					
9		Vergleich aller Inhalte mit einer Zelle			
10		Computer sind auch nur Menschen.			
11		Computer sind nur Menschen.			
12		Männer sind auch nur Menschen.			
13		Computer sind auch nur Menschen.			
14		Computer werden wie Menschen sein.			
15					

Wenn einer der anderen Inhalte mit Zelle B10 übereinstimmt, werden alle formatiert.
Formel ist: `=ODER(IDENTISCH(B10;B11:B14))`

Abbildung 4.11: Nur eine andere Zelle muss mit *B10* übereinstimmen

In der Version Excel 2007 gab es Probleme mit der bedingten Formatierung und der oben verwendeten Formel: Nach dem Öffnen war die Formatierung nicht zu sehen. Erst wenn unter *Regel bearbeiten* diese ohne Änderung noch einmal mit *OK* bestätigt wurde, war die Formatierung zu sehen. Nach dem Speichern und erneutem Öffnen der Mappe ging das Spiel von vorn los ...

Hinweis

Die Version 2007 verhält sich anders als ihre Vorgänger. Sie wertet die Formeln, welche als Matrixformeln zu berechnen sind, anders aus. Auch hier haben wir alternative Formeln verwendet, über welche die bedingte Formatierung dauerhaft wirkt:

`=ZÄHLENWENN(B10:B14;B10)>1`

Wenn zwei (oder mehr) gleich sind

Auch die folgende Lösung vergleicht die Zellen eines Bereichs. Hier werden diejenigen Zellen formatiert, für die es Duplikate im markierten Bereich gibt. Dabei ist der Inhalt selbst nicht von Bedeutung.

Im Bereich *B17:B21* sollen alle Zellen formatiert werden, wenn die Inhalte der Zellen gleich, also Duplikate vorhanden sind. Führen Sie die folgenden Schritte aus:

1. Markieren Sie den Bereich *B17:B21* und wählen auf der Registerkarte *START* den Befehl *Bedingte Formatierung/Neue Regel*. Wählen Sie *Formel zur Ermittlung der zu formatierenden Zellen verwenden* aus.
2. Tragen Sie die Formel `=SUMME(WENN(B17=B17:B21;1;0))>1` ein.

3. Legen Sie nach einem Klick auf die Schaltfläche *Formatieren* das gewünschte Format fest.
4. Bestätigen Sie die Formatierung und die Formel jeweils mit *OK*.

Die Zellen werden jetzt nur formatiert, wenn es Zellen mit gleichen Inhalten gibt. Für die Prüfung wird die Anzahl der Übereinstimmungen im markierten Bereich mit der aktiven Zelle gezählt. Da die aktive Zelle selbst mitgezählt wird, muss das Ergebnis größer als 1 sein, um zur Formatierung zu führen.

	A	B	C	D	E
15					
16		Vergleich aller Zellen auf Duplikate			
17		Computer sind auch nur Menschen.			
18		Computer sind vielleicht Menschen.			
19		Menschen wie Computer		Nur übereinstimmende Zellen werden farbig formatiert. Formel ist: =SUMME(WENN(B17=\$B\$17:\$B\$21;1;0))>1	
20		Computer sind nur Menschen.			
21		Menschen wie Computer			
22					

Abbildung 4.12: Die Duplikate werden hervorgehoben

Hinweis Die Version 2007 verhält sich anders als ihre Vorgänger. Sie wertet die Formeln, welche als Matrixformeln zu berechnen sind, anders aus. Deshalb haben wir in der Version 2007 einfach eine alternative Formel verwendet, über welche die bedingte Formatierung dauerhaft wirkt:

=ZÄHLENWENN(\$B\$17:\$B\$21;B17)>1



Alle drei Beispiele finden Sie in der Arbeitsmappe *Kap04_BF.xlsx* auf dem Arbeitsblatt *Vergleich*. Die Datei ist im Ordner `\Ms5-235\Kap04` gespeichert.

Wo sind die Unterschiede?

Manchmal kommt es vor, dass man zwei Versionen eines Arbeitsblatts vorliegen hat. Dann taucht das Problem auf, die Arbeitsblätter vergleichen zu müssen, um die Unterschiede herauszufinden. Auch hier kann die bedingte Formatierung helfen.

Gegeben sind zwei Versionen eines Arbeitsblatts mit Unfallstatistiken – *Unfall1* und *Unfall2*. Im Blatt *Unfall2* sollen diejenigen Zellen markiert werden, die einen anderen Inhalt haben als die entsprechende(n) Zelle(n) in *Unfall1*.

Eigentlich eine leichte Aufgabe, brauchen doch nur die Zellinhalte miteinander verglichen zu werden. Aber bis Excel 2007 gibt es eine kleine Schwierigkeit, die Sie sich zunächst ansehen sollten:

1. Markieren Sie den Datenbereich im Arbeitsblatt *Unfall2*.
2. Wählen Sie den Befehl *Bedingte Formatierung/Neue Regel* und stellen *Formel zur Ermittlung der zu formatierenden Zellen* verwenden ein.
3. Tragen Sie die Formel `=B4<>Unfall1!B4` ein.
4. Legen Sie das gewünschte Format fest (Schaltfläche *Formatieren*) und bestätigen jeweils mit *OK*.
5. Sie erhalten jetzt eine Fehlermeldung, die besagt, dass Bezüge auf andere Arbeitsblätter nicht zulässig seien.

Unverständlich, aber es ist so: Beim Festlegen der bedingten Formatierung mit Bezügen auf andere Arbeitsblätter oder Arbeitsmappen taucht ein Problem auf, denn mit den externen Bezügen kann Excel bis zur Version 2007 an dieser Stelle leider nichts anfangen.

Die Lösung liegt hier darin, eine Tabellenfunktion zu verwenden, welche den Bezug jeweils errechnet. Mit diesem »Trick« können Sie auch mit anderen Arbeitsblättern vergleichen.

Wählen Sie den Bereich für den Vergleich mittels der bedingten Formatierung ausreichend groß. Damit werden auch eventuell vorhandene neue bzw. gelöschte Datensätze als Unterschied markiert.

Hinweis

Um einen externen Blattbezug für die bedingte Formatierung zu verwenden, gehen Sie wie folgt vor:

1. Aktivieren Sie das Arbeitsblatt *Unfall2* und markieren Sie den Datenbereich.
2. Wählen Sie den Befehl *Bedingte Formatierung/Neue Regel* und anschließend den Regeltyp *Formel zur Ermittlung der zu formatierenden Zellen verwenden*.

3. Tragen Sie dann die folgende Formel ein:

```
=B4<>INDIREKT(ADRESSE(ZEILE(B4);SPALTE(B4);WAHR;WAHR;"Unfall1"))
```

4. Legen Sie über die Schaltfläche *Formatieren* das gewünschte Format fest.

5. Bestätigen Sie die Eingaben jeweils mit *OK*.

Jetzt werden die Unterschiede zum Arbeitsblatt *Unfall1* sichtbar (Abbildung 4.13). Die Funktionsweise der beteiligten Funktionen *INDIREKT()*, *ZEILE()* und *SPALTE()* können Sie im Kapitel 10 nachlesen.

Der Bezug auf das Arbeitsblatt wird als Zeichenfolge eingetragen. Das hat zur Folge, dass bei einer Namensänderung des Arbeitsblatts die Formel **nicht** angepasst wird. Sie dürfen in diesem Fall nicht vergessen, die Formel für die bedingte Formatierung anzupassen.

Wichtig

	A	B	C	D	E	F	G
1	Entwicklung der gemeldeten Unfallzahlen im Straßenverkehr						
2							
3	Land	2007	2008	2009	2010	2011	2012
4	NRW	847.919	802.800	485.202	188.577	152.628	232.426
5	BAY	710.654	664.802	587.084	780.685	618.273	188.409
6	HES	396.347	535.543	159.472	354.137	632.883	379.262
7	BRB	558.784	695.357	633.524	559.837	548.559	631.978
8	BWB	352.400	652.646	216.647	749.896	358.148	834.914
9	SHS	192.579	201.340	580.000	260.539	683.464	375.280
10	RPF	851.526	382.802	431.882	380.567	640.104	580.925
11	NSN	387.490	239.883	226.285	524.396	424.771	109.108
12	THÜ	135.889	189.133	318.886	146.005	747.328	471.613
13	HBG	329.809	386.825	865.007	340.281	453.471	428.973
14	BLN	117.222	160.589	198.056	879.868	710.976	265.116
15	SAN	123.608	843.000	214.100	259.151	213.288	325.060
16	MVP	552.060	274.995	425.132	841.377	676.841	309.595
17	SAX	610.516	279.053	829.739	743.316	567.779	763.007
18							
19	Diese Tabelle zeigt die Unterschiede zu Unfall1 mittels der Bedingten Formatierung.						

Abbildung 4.13: Die Unterschiede der Tabellen markiert die bedingte Formatierung

Die vorgeschlagene Lösung funktioniert generell auch mit Arbeitsblättern, die sich in verschiedenen Arbeitsmappen befinden – allerdings mit einer Einschränkung, dass beide Mappen geöffnet sein müssen. Dann können Sie für die bedingte Formatierung die folgende Formel einsetzen. Sie enthält in eckigen Klammern zusätzlich den Namen der externen Mappe:

=B4<>INDIREKT(ADRESSE(ZEILE(B4);SPALTE(B4);WAHR;WAHR;"[Unfa111.xlxs]Unfa111"))

Die gute Nachricht für alle Leser mit einer neueren Excel-Version: Ab der Version 2010 können Sie mit der einfachen Formel =B4<>Unfa111!B4 arbeiten.



Das Beispiel finden Sie in der Arbeitsmappe *Kap04_BF.xlsx* auf dem Arbeitsblatt *Unfall2*. Dazu gehört das Arbeitsblatt *Unfall1*. Die Datei ist im Ordner `\Ms5-235\Kap04` gespeichert.

Listendrucklayout

Für den Druck einer Liste macht sich zur Betonung der Zeilen die Unterlegung jeder zweiten Zeile mit einem Farbwert eventuell besser als Rahmenlinien. Bei großen Listen wäre es allerdings sehr mühsam, jede zweite Zeile manuell zu formatieren. Zudem würde ein einziger Sortiervorgang die ganze Arbeit zunichte machen. Auch hier hilft die bedingte Formatierung:

1. Markieren Sie den Listenbereich ohne die Überschriften.
2. Rufen Sie den Befehl *START/Bedingte Formatierung/Neue Regel* auf und stellen den Eintrag *Formel zur Ermittlung der zu formatierenden Zellen verwenden* ein.
3. Tragen Sie dann die Formel =REST(ZEILE();2)=1 für das Hervorheben der ungeraden Zeilen ein.
4. Legen Sie über die Schaltfläche *Formatieren* das gewünschte Format fest.
5. Bestätigen Sie die Eingaben jeweils mit *OK*.

	A	B	C	D
1	Funktion	Beschreibung	Kategorie	Add-In
2	AMORDEGRK	Liefert den für eine Abrechnungsperiode anzusetzenden Abschreibungsbe	Finanzmathematik	Analyse-Funktionen
3	AMORLINEARK	Liefert den für eine Abrechnungsperiode anzusetzenden Abschreibungsbe	Finanzmathematik	Analyse-Funktionen
4	AUFGELZINS	Liefert die aufgelaufenen Zinsen (Stückzinsen) eines Wertpapiers mit peri	Finanzmathematik	Analyse-Funktionen
5	AUFGELZINSF	Liefert die aufgelaufenen Zinsen (Stückzinsen) eines Wertpapiers, die bei	Finanzmathematik	Analyse-Funktionen
6	AUSZAHLUNG	Liefert den Auszahlungsbetrag eines voll investierten Wertpapiers am Fäll	Finanzmathematik	Analyse-Funktionen
7	BW	Liefert den Barwert einer Investition	Finanzmathematik	Standard
8	DIA	Liefert die arithmetisch-degressive Abschreibung eines Wirtschaftsguts fü	Finanzmathematik	Standard
9	DISAGIO	Liefert den in Prozent ausgedrückten Abschlag (Disagio) eines Wertpapie	Finanzmathematik	Analyse-Funktionen
10	DURATION	Liefert die jährliche Duration eines Wertpapiers mit periodischen Zinszahl	Finanzmathematik	Analyse-Funktionen
11	EFFEKTIV	Liefert die jährliche Effektivverzinsung	Finanzmathematik	Analyse-Funktionen
12	GDA	Liefert die degressive Doppelraten-Abschreibung eines Wirtschaftsguts fü	Finanzmathematik	Standard
13	GDA2	Liefert die geometrisch-degressive Abschreibung eines Wirtschaftsguts fü	Finanzmathematik	Standard
14	IKV	Liefert den internen Zinsfuß einer Investition ohne Finanzierungskosten oc	Finanzmathematik	Standard
15	ISPMT	Gibt den Zinssatz für gleichgroße Rückzahlungsraten zurück	Finanzmathematik	Standard
16	KAPZ	Liefert die Kapitalrückzahlung einer Investition für die angegebene Periode	Finanzmathematik	Standard
17	KUMKAPITAL	Berechnet die aufgelaufene Tilgung eines Darlehens, die zwischen zwei F	Finanzmathematik	Analyse-Funktionen

Abbildung 4.14: Die ungeraden Zeilen sind hellgrau hinterlegt

Diese Formatierung wird durch keine Sortierung gestört und ergibt ein gutes Drucklayout. Die Wirkungsweise der Funktion REST() finden Sie in Kapitel 15 beschrieben.

Wenn Sie ein dynamisch wachsendes Arbeitsblatt haben und die bedingte Formatierung »auf Vorrat« für die maximale Länge der Liste einrichten, würden die Zeilenmarkierungen unterhalb des Listenendes stören. Dazu wird die Formel um eine weitere Prüfung erweitert. Beide Prüfungen werden mit der UND()-Funktion verknüpft. Mit der Formel

=UND(REST(ZEILE();2)=1;\$A2<>"")

für die bedingte Formatierung wird zusätzlich die erste Spalte der Liste geprüft. Enthält diese einen Wert und erfüllt die Zeilennummer die gewünschte Bedingung, wird die Leselinie angezeigt. Wenn Sie diese Formel einsetzen wollen, können Sie vor der Eingabe der Bedingung einen größeren Bereich markieren. Die Formatierung wird erst durchgeführt, wenn der Bereich dann auch mit Daten gefüllt wurde.

Das Beispiel haben wir in der Arbeitsmappe *Kap04_BF.xlsx* auf dem Arbeitsblatt *Drucklayout* eingerichtet. Die Datei ist im Ordner `\Ms5-235\Kap04` gespeichert.



Top 3 hervorheben

Die Kennzeichnung von Top-Elementen sollte mit der bedingten Formatierung kein Problem sein. Auch hier schöpfen Sie aus dem Funktionen-Fundus von Excel. Nehmen wir an, Sie haben für verschiedene Orte über den Monat gemessene Temperaturreihen. Es sollen die drei wärmsten und die drei kältesten Temperaturen signalisiert werden.

	A	B	C	D	E
1	Temperaturmessungen im September				
2					
3		Berlin	Köln	Hamburg	München
4	01.09.2013	9°C	27°C	22°C	30°C
5	02.09.2013	24°C	19°C	12°C	25°C
6	03.09.2013	27°C	22°C	19°C	19°C
7	04.09.2013	28°C	10°C	22°C	11°C
8	05.09.2013	15°C	19°C	19°C	12°C
9	06.09.2013	27°C	24°C	11°C	23°C
10	07.09.2013	24°C	14°C	26°C	22°C
11	08.09.2013	12°C	27°C	12°C	14°C
12	09.09.2013	18°C	14°C	23°C	25°C
13	10.09.2013	23°C	26°C	13°C	26°C
14	11.09.2013	20°C	13°C	11°C	16°C
15	12.09.2013	26°C	30°C	23°C	12°C
16	13.09.2013	26°C	16°C	22°C	16°C
17	14.09.2013	27°C	12°C	17°C	12°C
18	15.09.2013	24°C	11°C	8°C	11°C
19	16.09.2013	18°C	13°C	20°C	10°C

Abbildung 4.15: Rot für die heißesten, blau für die kältesten Tage des Monats

Um diese Aufgabe zu lösen, gehen Sie so vor:

1. Markieren Sie die Spalten mit den Temperaturwerten einschließlich der Überschriften (im Beispiel ist das der Bereich `B3:E33`). Mit der Tastenkombination `[Strg] + [⇩] + [F3]` rufen Sie ohne Umwege das Dialogfeld *Namen aus Auswahl erstellen* auf.
2. Bestätigen Sie den Vorschlag, die Überschriften aus der obersten Zeile (der Markierung) als Namen für die Spalten zu verwenden, mit *OK*.

Sie haben nun die Datenspalten mit Bereichsnamen, nämlich den Orten aus der Überschrift, versehen. Diese werden in den Formeln für die bedingte Formatierung benötigt.

3. Markieren Sie den Bereich mit den Temperaturwerten (im Beispiel ist das der Bereich *B4:E33*) und rufen Sie den Befehl *START/Bedingte Formatierung/Neue Regel* auf.
4. Im Dialogfeld *Neue Formatierungsregel* wählen Sie den Regeltyp *Formel zur Ermittlung der zu formatierenden Zellen verwenden* und tragen die folgende Formel ein:

```
=ODER(B4=MAX(INDIREKT(B$3));B4=KGRÖSSTE(INDIREKT(B$3);2);
B4=KGRÖSSTE(INDIREKT(B$3);3))
```

5. Klicken Sie auf die Schaltfläche *Formatieren* und wählen Sie auf der Registerkarte *Schrift* eine Formatierung aus (z.B. fett und rot). Bestätigen Sie mit *OK*.
 6. Klicken Sie auf die Schaltfläche *Neue Regel*. Für die zweite Regel tragen Sie die folgende Formel ein:
- ```
=ODER(B4=MIN(INDIREKT(B$3));B4=KKLEINSTE(INDIREKT(B$3);2);
B4=KKLEINSTE(INDIREKT(B$3);3))
```
7. Klicken Sie auf die Schaltfläche *Formatieren* und wählen auf der Registerkarte *Schrift* eine Formatierung aus (z.B. fett und dunkelblau). Bestätigen Sie mit *OK*.

Die Funktion *INDIREKT()* erzeugt den Bereichsnamen aus den Überschriften. So können Sie die bedingten Formate in einem Schritt für alle Spalten erzeugen. Mehr über die statistischen Funktionen *MIN()*, *MAX()*, *KKLEINSTE()* und *KGRÖSSTE()* erfahren Sie in Kapitel 11. Sie sehen, dass die interdisziplinäre Zusammenarbeit der Funktionen aus verschiedenen Kategorien eine gewisse Mächtigkeit entwickelt.

**Tipp** Ab der Version 2007 können Sie im *AutoFilter* (Symbol *Filtern* auf der Registerkarte *DATEN*) auch nach Formatierungen filtern, z.B. nach Schrift- oder Hintergrundfarbe!



Das Beispiel haben wir in der Arbeitsmappe *Kap04\_BF.xlsx* auf dem Arbeitsblatt *Top3* eingerichtet. Die Datei ist im Ordner *\Ms5-235\Kap04* gespeichert.

## Zellen mit Leerzeichen hervorheben

Wenn Sie Daten aus einer Textdatei importiert haben, kennen Sie vielleicht auch das folgende Problem: In einigen Zellen sind Leerzeichen enthalten, die Ihnen bei der weiteren Bearbeitung Probleme bereiten. Über den Befehl *Suchen* (*[Strg]+[F]*) können Sie zwar nach dem Leerzeichen suchen, aber da vielleicht nicht alle Leerzeichen ersetzt werden können, ist diese Methode doch sehr umständlich. Die Funktion *GLÄTTEN()* könnte helfen, aber Sie legen Wert auf eine Einzelfallprüfung. Sie suchen also nach einer Möglichkeit, die Zellen, die ein Leerzeichen enthalten, hervorzuheben. Lassen Sie sich dabei von der bedingten Formatierung helfen.

Häufig stören nach dem Datenimport Leerzeichen, insbesondere am Ende des Texts, weil sie hier nur schwer zu finden sind. Auf den ersten Blick können Sie nicht feststellen, ob der Zellinhalt mit einem Leerzeichen endet. Erst, wenn Sie mit der Taste *[F2]* zum letzten Zeichen der Bearbeitungsleiste wechseln, wird dies deutlich. Sie können Zellen mit bestimmten Inhalten mit der bedingten Formatierung finden – das gilt auch für Leerzeichen. Sie benötigen in der bedingten Formatierung diese Formeln:

```
=LINKS(C5;1)=ZEICHEN(32)
```

ermittelt ein führendes Leerzeichen. *C5* ist dabei die linke obere Ecke der Markierung.

=RECHTS(D5;1)=ZEICHEN(32)

ermittelt ein Leerzeichen am Ende. D5 ist dabei die linke obere Ecke der Markierung.

=UND(LINKS(E5;1)=ZEICHEN(32);RECHTS(E5;1)=ZEICHEN(32))

ermittelt, ob ein Leerzeichen zu Beginn und am Ende steht. E5 ist dabei die linke obere Ecke der Markierung.

=FINDEN(ZEICHEN(32);F5)>0

ermittelt ein beliebig vorkommendes Leerzeichen. F5 ist dabei die linke obere Ecke der Markierung.

Die genannten Beispiele finden Sie in der Arbeitsmappe *Kap04\_BF.xlsx* auf dem Arbeitsblatt *Leer*. Die Datei ist im Ordner *\Ms5-235\Kap04* gespeichert.



## Hilfslinien für die Navigation in Arbeitsblättern

Ein häufiges Problem ist das Ablesen von Werten aus Tabellen. Es wird ein Wert benötigt, der im Schnittpunkt einer bestimmten Zeile und einer bestimmten Spalte steht. Wie in Abbildung 4.16 gezeigt, soll in diesem Beispiel für einen Betrag und die Anzahl der Teilzahlungen der jeweilige Ratenbetrag schnell in der Tabelle zu finden sein. Auf einem Blatt Papier kann man ein Lineal oder einen Stift anlegen, aber auf dem Bildschirm ... nun, das wäre weniger sinnvoll. Mit der bedingten Formatierung geht es auch anders.

|    | A | B                    | C          | D          | E                           | F          | G           | H               | I        | J        |
|----|---|----------------------|------------|------------|-----------------------------|------------|-------------|-----------------|----------|----------|
| 1  |   |                      |            |            |                             |            |             |                 |          |          |
| 2  |   | <b>Kreditberater</b> |            |            |                             |            |             |                 |          |          |
| 3  |   | <b>Betrag</b>        | 15.000 €   |            |                             |            |             |                 |          |          |
| 4  |   | <b>Laufzeit</b>      | 36         |            | <b>Monatliche Rate</b>      |            | 451,25 €    |                 |          |          |
| 5  |   | <b>Zinssatz</b>      | 5,25%      |            | <b>Effektiver Kaufpreis</b> |            | 16.244,97 € |                 |          |          |
| 6  |   |                      |            |            |                             |            |             |                 |          |          |
| 7  |   |                      |            |            |                             |            |             |                 |          |          |
| 8  |   | <b>Betrag</b>        |            |            |                             |            |             | <b>Laufzeit</b> |          |          |
| 9  |   | 1.000 €              | 252,74 €   | 127,47 €   | 85,72 €                     | 57,89 €    | 43,98 €     | 30,08 €         | 23,14 €  | 18,99 €  |
| 10 |   | 2.000 €              | 505,48 €   | 254,95 €   | 171,44 €                    | 115,79 €   | 87,97 €     | 60,17 €         | 46,29 €  | 37,97 €  |
| 11 |   | 3.000 €              | 758,22 €   | 382,42 €   | 257,17 €                    | 173,68 €   | 131,95 €    | 90,25 €         | 69,43 €  | 56,96 €  |
| 12 |   | 4.000 €              | 1.010,96 € | 509,89 €   | 342,89 €                    | 231,57 €   | 175,93 €    | 120,33 €        | 92,57 €  | 75,94 €  |
| 13 |   | 5.000 €              | 1.263,70 € | 637,37 €   | 428,61 €                    | 289,47 €   | 219,92 €    | 150,42 €        | 115,71 € | 94,93 €  |
| 14 |   | 6.000 €              | 1.516,44 € | 764,84 €   | 514,33 €                    | 347,36 €   | 263,90 €    | 180,50 €        | 138,86 € | 113,92 € |
| 15 |   | 7.000 €              | 1.769,18 € | 892,31 €   | 600,05 €                    | 405,25 €   | 307,88 €    | 210,58 €        | 162,00 € | 132,90 € |
| 16 |   | 8.000 €              | 2.021,92 € | 1.019,79 € | 685,78 €                    | 463,15 €   | 351,87 €    | 240,67 €        | 185,14 € | 151,89 € |
| 17 |   | 9.000 €              | 2.274,66 € | 1.147,26 € | 771,50 €                    | 521,04 €   | 395,85 €    | 270,75 €        | 208,28 € | 170,87 € |
| 18 |   | 10.000 €             | 2.527,40 € | 1.274,73 € | 857,22 €                    | 578,93 €   | 439,83 €    | 300,83 €        | 231,43 € | 189,86 € |
| 19 |   | 11.000 €             | 2.780,14 € | 1.402,21 € | 942,94 €                    | 636,82 €   | 483,82 €    | 330,92 €        | 254,57 € | 208,85 € |
| 20 |   | 12.000 €             | 3.032,88 € | 1.529,68 € | 1.028,67 €                  | 694,72 €   | 527,80 €    | 361,00 €        | 277,71 € | 227,83 € |
| 21 |   | 13.000 €             | 3.285,62 € | 1.657,16 € | 1.114,39 €                  | 752,61 €   | 571,78 €    | 391,08 €        | 300,86 € | 246,82 € |
| 22 |   | 14.000 €             | 3.538,36 € | 1.784,63 € | 1.200,11 €                  | 810,50 €   | 615,77 €    | 421,17 €        | 324,00 € | 265,80 € |
| 23 |   | 15.000 €             | 3.791,11 € | 1.912,10 € | 1.285,83 €                  | 868,40 €   | 659,75 €    | 451,25 €        | 347,14 € | 284,79 € |
| 24 |   | 16.000 €             | 4.043,85 € | 2.039,58 € | 1.371,55 €                  | 926,29 €   | 703,73 €    | 481,33 €        | 370,28 € | 303,78 € |
| 25 |   | 17.000 €             | 4.296,59 € | 2.167,05 € | 1.457,28 €                  | 984,18 €   | 747,72 €    | 511,42 €        | 393,43 € | 322,76 € |
| 26 |   | 18.000 €             | 4.549,33 € | 2.294,52 € | 1.543,00 €                  | 1.042,08 € | 791,70 €    | 541,50 €        | 416,57 € | 341,75 € |
| 27 |   | 19.000 €             | 4.802,07 € | 2.422,00 € | 1.628,72 €                  | 1.099,97 € | 835,69 €    | 571,58 €        | 439,71 € | 360,73 € |
| 28 |   | 20.000 €             | 5.054,81 € | 2.549,47 € | 1.714,44 €                  | 1.157,86 € | 879,67 €    | 601,67 €        | 462,85 € | 379,72 € |

**Abbildung 4.16:** Die farbliche Führung hebt die Auswahl in der Raten-Matrix hervor

In der Zelle C9 ermittelt die Tabellenfunktion RMZ() den Teilzahlungsbetrag über die Formel

=RMZ(\$C\$5/12;C\$8;-\$B9)

Die Mischung aus absoluten (Zins) und relativen Bezügen (Zahlungszeiträume und Barwert) erlaubt das Kopieren dieser Formel über den Bereich C9:J28. Mehr zu finanzmathematischen Funktionen, speziell zu RMZ(), finden Sie in Kapitel 14.

Durch die Verwendung der Datenprüfung können Sie in den Zellen C3 den Betrag und in C4 die Laufzeit komfortabel aus einer Liste von Werten auswählen. Für den Betrag werden die Werte aus dem Bereich *Betrag* (dieser Name steht für den Zellbereich B9:B28) verwendet, für die Laufzeit die Werte aus dem Bereich *Monate* (dieser Name steht für den Zellbereich C8:J8). Weitere Beispiele zu Daten- bzw. Gültigkeitsprüfungen finden Sie im Abschnitt »Funktionen bei der Datenprüfung« ab Seite 163.

### Markierung des gewählten Betrags bzw. der Laufzeit

Zunächst muss der gesuchte Betrag in der Vorspalte der Liste markiert werden. Dazu gehen Sie wie folgt vor:

1. Markieren Sie den Bereich B9:B28. Wählen Sie auf der Registerkarte *START* den Befehl *Bedingte Formatierung/Neue Regel* auf.
2. Im jetzt geöffneten Dialogfeld wählen Sie wieder den Regeltyp *Formel zur Ermittlung der zu formatierenden Zellen verwenden* aus.
3. Tragen Sie dann die Formel =B9=\$C\$3 ein.
4. Legen Sie das gewünschte Format fest und schließen Sie die beiden Dialogfelder jeweils mit *OK*.

Nun gilt es, die in Zelle C4 eingestellte Laufzeit in der Überschrift des Arbeitsblatts zu markieren. Für diese Aufgabe sind die folgenden Schritte nötig:

1. Markieren Sie den Bereich C8:J8.
2. Wählen Sie erneut den Befehl *Bedingte Formatierung/Neue Regel*. Im jetzt geöffneten Dialogfeld wählen Sie den Regeltyp *Formel zur Ermittlung der zu formatierenden Zellen verwenden*.
3. Tragen Sie die Formel =C8=\$C\$4 ein.
4. Legen Sie das gewünschte Format fest und schließen Sie die beiden Dialogfelder wieder mit *OK*.

Damit ist die Grundlage der Markierung ermittelt.

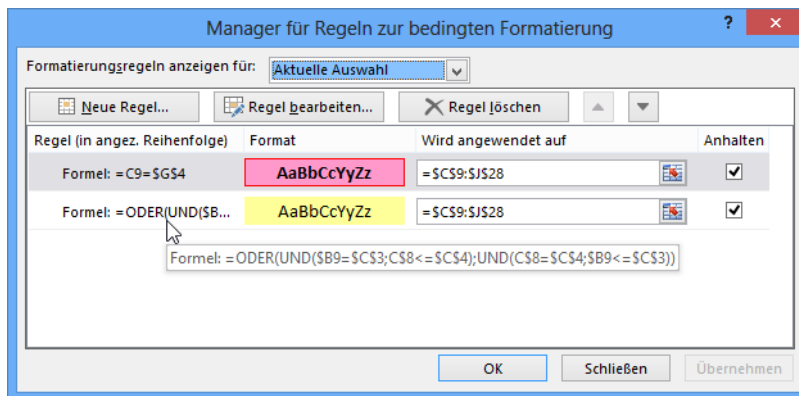
### Markierung der Ergebniszelle und der Führung dorthin

Etwas schwieriger gestaltet sich allerdings das Anzeigen als Hilfslinie, weil dabei mehrere Bedingungen zu prüfen sind. Es sollen ja alle Zellen eine Formatierung erhalten, die sich in der gefundenen Zeile befinden und außerdem auch alle Zellen, die sich in der gefundenen Spalte befinden. Diese Bedingungen werden durch die ODER()-Funktion eingestellt. Beide Bedingungen haben die Einschränkung, dass jeweils nur bis zu einer bestimmten Spalte bzw. Zeile die Formatierung ausgeführt werden soll. Es gelten also zwei Bedingungen für jeden Fall; diese werden über die UND()-Funktion verknüpft.

Gehen Sie wie folgt vor, um die Ergebniszelle hervorzuheben und den Datenbereich mit den Hilfslinien zu versehen:

1. Markieren Sie den Datenbereich *C9:J28* und rufen Sie auf der Registerkarte *START* den Befehl *Bedingte Formatierung/Neue Regel* auf.
2. Wählen Sie *Formel zur Ermittlung der zu formatierenden Zellen verwenden* und notieren die Formel  $=C9=\$G\$4$ .
3. Legen Sie das gewünschte Format für die Ergebniszelle fest und klicken dann auf die Schaltfläche *Neue Regel*.
4. Wählen Sie wieder *Formel zur Ermittlung der zu formatierenden Zellen verwenden* und tragen folgende Formel ein:  
 $=\text{ODER}(\text{UND}(\$B9=\$C\$3;C\$8<=\$C\$4); \text{UND}(C\$8=\$C\$4; \$B9<=\$C\$3))$
5. Legen Sie das gewünschte Format für die Markierung fest und bestätigen Sie abschließend jeweils mit *OK*.

Die Abbildung 4.17 zeigt die Formeln für die Bedingungen im Dialogfeld *Manager für Regeln zur bedingten Formatierung*. Wichtig ist die Reihenfolge der Bedingungen. Wenn die Zelle auch dem Berechnungsergebnis oben im Kopfteil entspricht, soll sie als Ergebniszelle besonders hervorgehoben werden. Da sie auch die zweite Bedingung erfüllt, würde die an zweiter Stelle festgelegte Formatierung nie gezeigt werden.



**Abbildung 4.17:** Zunächst die erste Bedingung für die Ergebniszelle, dann die Bedingung für die Führungslinien

Beachten Sie auch die gemischten Bezüge bei den Vergleichen. Nur so erreichen Sie das gewünschte Ziel. Das Ergebnis ist überzeugend. Man kann die Zusammenhänge in der Raten-Tabelle schneller erkennen und das Ganze hat damit eine bessere Systematik.

Das gezeigte Beispiel finden Sie im Ordner *\Ms5-235\Kap04* in der Arbeitsmappe *Kap04\_BF.xlsx* auf dem Arbeitsblatt *Kredit*.



## Datengruppen absetzen

Ein weiteres Beispiel: In einer umfangreichen Liste, die nach Firmen sortiert ist, soll nach Datensätzen mit der gleichen Firma eine Trennlinie eingefügt werden. So können Sie zusammengehörige Gruppen schnell erkennen (Abbildung 4.18). Manuell ist das kein Problem, aber geht es auch dynamisch?

|    | A                          | B           | C        | D              | E             | F          |
|----|----------------------------|-------------|----------|----------------|---------------|------------|
| 1  | Firma                      | Ort         | Datu     | Land           | Artikelname   | Umsa       |
| 2  | Alfreds Futterkiste        | Berlin      | 01.05.12 | Deutschland    | Flotemysost   | 1.961,00 € |
| 3  | Alfreds Futterkiste        | Berlin      | 02.09.11 | Deutschland    | Mozzarella di | 1.729,00 € |
| 4  | Alfreds Futterkiste        | Berlin      | 31.08.11 | Deutschland    | Röd Kaviar    | 1.226,00 € |
| 5  | Alfreds Futterkiste        | Berlin      | 03.10.11 | Deutschland    | Longlife Tofu | 1.012,00 € |
| 6  | Alfreds Futterkiste        | Berlin      | 10.02.12 | Deutschland    | Rhönbräu      | 1.950,00 € |
| 7  | Alfreds Futterkiste        | Berlin      | 02.09.11 | Deutschland    | Lakkalikööri  | 1.733,00 € |
| 8  | Alfreds Futterkiste        | Berlin      | 02.11.11 | Deutschland    | Original      | 1.871,00 € |
| 9  | Ana Trujillo Emparedados y | México D.F. | 11.04.12 | Mexiko         | Chai          | kein       |
| 10 | Ana Trujillo Emparedados y | México D.F. | 05.05.12 | Mexiko         | Chang         | 1.615,00 € |
| 11 | Ana Trujillo Emparedados y | México D.F. | 18.03.12 | Mexiko         | Aniseed Syrup | 1.771,00 € |
| 12 | Ana Trujillo Emparedados y | México D.F. | 03.02.12 | Mexiko         | Chef Anton's  | 1.218,00 € |
| 13 | Antonio Moreno Taquería    | México D.F. | 03.07.12 | Mexiko         | Röd Kaviar    | 1.580,00 € |
| 14 | Antonio Moreno Taquería    | México D.F. | 15.11.11 | Mexiko         | Longlife Tofu | 1.371,00 € |
| 15 | Antonio Moreno Taquería    | México D.F. | 08.11.11 | Mexiko         | Rhönbräu      | 1.637,00 € |
| 16 | Antonio Moreno Taquería    | México D.F. | 11.12.11 | Mexiko         | Lakkalikööri  | 1.604,00 € |
| 17 | Antonio Moreno Taquería    | México D.F. | 17.11.11 | Mexiko         | Original      | 1.369,00 € |
| 18 | Around the Horn            | London      | 09.04.12 | Großbritannien | Chai          | kein       |

Abbildung 4.18: Bei jedem Firmenwechsel wird ein Trennstrich eingezogen

Mithilfe der bedingten Formatierung ist diese Aufgabe auch dynamisch realisierbar, und zwar so:

1. Markieren Sie den Datenbereich (im Beispiel sind das die Zellen A2:F86).
2. Sortieren Sie die Daten über den Befehl *DATEN/Sortieren* nach der Firma.
3. Wählen Sie den Befehl *START/Bedingte Formatierung/Neue Regel*.
4. Im geöffneten Dialogfeld wählen Sie den Regeltyp *Formel zur Ermittlung der zu formatierenden Zellen verwenden* aus und tragen in das Formelfeld die Formel  $=A2<>A3$  ein.
5. Über die Schaltfläche *Formatieren* wählen Sie auf der Registerkarte *Rahmen* den unteren Rahmen aus.
6. Schließen Sie die Dialogfelder jeweils mit *OK*.

Die genannte Formel prüft, ob der Wert der nachfolgenden Zeile ungleich dem der aktuellen Zeile ist. Keine bedingte Formatierung wird vorgenommen, wenn die Zellen den gleichen Inhalt haben. Wenn die Bedingung *WAHR* ist, wird das Format zugewiesen. Damit haben Sie eine Trennlinie zwischen die Gruppen gezeichnet.

**Hinweis** Auch wenn Sie die Daten absteigend sortieren, wird die Trennlinie korrekt nach dem Gruppenwechsel gezeichnet.



Das gezeigte Beispiel finden Sie im Ordner `\Ms5-235\Kap04` in der Arbeitsmappe `Kap04_BF.xlsx` auf dem Arbeitsblatt *Gruppen*.



## Teilergebnisse formatieren

Wenn Sie eine Liste über *DATEN/Teilergebnis* unter Verwendung der Funktion *SUMME()* gruppiert haben, wollen Sie vielleicht die Teilergebnisse optisch hervorheben. Natürlich möchten Sie das nicht jedes Mal »von Hand« erledigen. Auch bei diesem Problem kann Ihnen die bedingte Formatierung helfen.

|    | A                          | B            | C                |
|----|----------------------------|--------------|------------------|
| 1  | <b>Vertreter</b>           | <b>Datum</b> | <b>Umsätze</b>   |
| 2  | Bärwald                    | 27.09.2005   | 22.611 €         |
| 3  | Bärwald                    | 04.10.2005   | 20.710 €         |
| 4  | Bärwald                    | 16.10.2005   | 20.063 €         |
| 5  | Bärwald                    | 30.10.2005   | 25.567 €         |
| 6  | <b>Bärwald Ergebnis</b>    |              | <b>88.951 €</b>  |
| 7  | Friedrichs                 | 19.09.2005   | 23.853 €         |
| 8  | Friedrichs                 | 12.10.2005   | 17.592 €         |
| 9  | Friedrichs                 | 30.10.2005   | 18.827 €         |
| 10 | <b>Friedrichs Ergebnis</b> |              | <b>60.272 €</b>  |
| 11 | Jansen                     | 05.09.2005   | 20.839 €         |
| 12 | Jansen                     | 21.10.2005   | 10.382 €         |
| 13 | Jansen                     | 09.11.2005   | 18.027 €         |
| 14 | <b>Jansen Ergebnis</b>     |              | <b>49.248 €</b>  |
| 15 | Maier                      | 09.09.2005   | 24.352 €         |
| 16 | Maier                      | 04.10.2005   | 14.817 €         |
| 17 | Maier                      | 17.10.2005   | 18.285 €         |
| 18 | <b>Maier Ergebnis</b>      |              | <b>57.434 €</b>  |
| 19 | Müller                     | 11.10.2005   | 17.930 €         |
| 20 | Müller                     | 21.12.2005   | 13.415 €         |
| 21 | <b>Müller Ergebnis</b>     |              | <b>31.345 €</b>  |
| 22 | Schmidt                    | 30.10.2005   | 27.933 €         |
| 23 | Schmidt                    | 03.11.2005   | 21.284 €         |
| 24 | Schmidt                    | 15.12.2005   | 13.287 €         |
| 25 | <b>Schmidt Ergebnis</b>    |              | <b>62.504 €</b>  |
| 26 | Schulze                    | 17.09.2005   | 24.059 €         |
| 27 | Schulze                    | 02.10.2005   | 21.214 €         |
| 28 | Schulze                    | 10.11.2005   | 12.519 €         |
| 29 | <b>Schulze Ergebnis</b>    |              | <b>57.792 €</b>  |
| 30 | <b>Gesamtergebnis</b>      |              | <b>407.546 €</b> |

**Abbildung 4.19:** Die Teilergebnisse sind mit Fettschrift, grauem Hintergrund und mit Rahmen abgesetzt

Führen Sie die folgenden Schritte aus, um eine solche Formatierung zu erreichen:

1. Sortieren Sie die Daten über *DATEN/Sortieren* nach dem Vertreter.
2. Wählen Sie *DATEN/Teilergebnis* und gruppieren Sie die Daten nach der Spalte *Vertreter*. Bilden Sie die Summe vom *Umsatz* und schließen Sie mit *OK* ab.
3. Markieren Sie die Daten im Bereich A2:C30 und wählen Sie die *Bedingte Formatierung/Neue Regel*. Markieren Sie den Regeltyp *Formel zur Ermittlung der zu formatierenden Zellen verwenden*.
4. Als Formel tragen Sie ein: `=ISTZAHL(FINDEN("rgebnis";$A2))` ein. Beachten Sie den gemischten Bezug für die Zelle \$A2!
5. Legen Sie nun noch das gewünschte Format fest und bestätigen Sie abschließend die Eingaben jeweils mit Klick auf *OK*.

Das gezeigte Beispiel finden Sie im Ordner `Ms5-235\Kap04` in der Arbeitsmappe `Kap04_BF.xlsx` auf dem Arbeitsblatt *Teilergebnisse*.



Teilergebnis



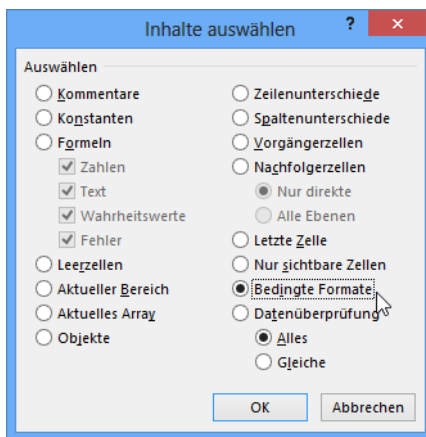
## Tipps zur bedingten Formatierung

Seit Excel 2007 ist mit dem Manager für Regeln zur bedingten Formatierung eine intuitive Bedienung und komfortable Verwaltung der bedingten Formate möglich. Das Ändern, Löschen und Hinzufügen von bedingten Formaten und deren Regeln muss nicht mehr groß erläutert werden. Trotzdem hier noch einige Tipps.

## Zellen mit bedingter Formatierung finden

Mitunter weiß man nicht mehr genau, für welche Zellbereiche eines Blatts oder einer Mappe bedingte Formatierungen festgelegt wurden. In diesem Fall hilft Ihnen Excel mit einer wenig bekannten Möglichkeit:

1. Rufen Sie mit der Taste **[F5]** das Dialogfeld *Gehe zu* auf (alternativ wählen Sie die Tastenkombination **[Strg]+[G]**).
2. Klicken Sie dort auf die Schaltfläche *Inhalte*.



**Abbildung 4.20:** Suche nach Zellen, die bedingte Formate enthalten

3. Markieren Sie den Eintrag *Bedingte Formate* und unter *Datenprüfung* die Option *Alles* (Abbildung 4.20). Starten Sie den Vorgang mit Klick auf die Schaltfläche *OK*.

Auf diese Weise können Sie beispielsweise für alle gefundenen Zellen mit bedingter Formatierung die zuvor festgelegten Formate schnell wieder löschen oder ändern.

Wollen Sie nicht alle Zellen mit bedingter Formatierung, sondern nur diejenigen finden, die identische bedingte Formate haben, dann klicken Sie unter *Datenprüfung* auf *Gleiche*.

## Troubleshooting zur bedingten Formatierung

Für den Fall, dass eine bedingte Formatierung nicht korrekt angezeigt wird, können mehrere Ursachen vorliegen:

- ▶ Es kann sein, dass mehrere Bedingungen festgelegt wurden. Wenn mehrere Bedingungen auf eine Zelle oder einen Zellbereich zutreffen, wendet Excel nur die Formatierung für die erste zutreffende Bedingung an.
- ▶ Wenn Bedingungen festgelegt wurden, die sich überschneiden, weist Excel ebenfalls nur das Format der ersten zutreffenden Bedingung zu
- ▶ Haben Sie beim Festlegen einer bedingten Formatierung keine Konstanten, sondern Formeln mit Zellbezügen verwendet, kann es sein, dass sich inzwischen die Zellbezüge geändert haben

## Funktionen bei der Datenprüfung

Die *Datenprüfung* auf der Registerkarte *DATEN* stellt für Eingabekontrollen einiges bereit. So können Sie u.a. die Eingabe auf bestimmte Datentypen und Inhalte einschränken. Richtig spannend wird es, wenn Sie für die Gültigkeitsprüfung auch Funktionen verwenden, also berechnete Gültigkeitswerte haben.

### Zellschutz mithilfe der Datenprüfung

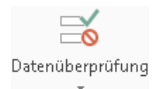
Die Datenprüfung bzw. Gültigkeitsprüfung kann auch für einen einfachen Zellschutz ohne Passwort verwendet werden. Sollen etwa Formatierungsoptionen verfügbar bleiben, ist der Standardzellschutz ungeeignet, weil dieser auch das Formatieren von Zellen verhindert. Über den Menübandbefehl *ÜBERPRÜFEN/Benutzer dürfen Bereiche bearbeiten* geht das zwar, aber eine interessante, weil einfache Möglichkeit bietet die Datenprüfung dennoch.

Nachdem Sie ein Arbeitsblatt mit Texten, Werten und Formeln gefüllt haben, gehen Sie so vor:

1. Markieren Sie das gesamte Blatt oder den verwendeten Bereich.
2. Wählen Sie dann den Menübandbefehl *DATEN/Datenprüfung*.
3. Stellen Sie das Listenfeld *Zulassen* auf *Benutzerdefiniert* und tragen Sie diese Formel ein:

```
=LÄNGE(A1)<1
```

Damit wäre eine Eingabe nur dann gültig, wenn sie weniger als ein Zeichen lang ist. Die Zelle ist damit vor dem direkten Überschreiben, wie es versehentlich vorkommen kann, geschützt. Das ist selbstverständlich kein Schutz, der unberechtigte Benutzer abhalten kann, Daten zu zerstören. Zumal dies mit der Entf-Taste immer noch funktioniert. Aber eine schnelle Hilfe für das Unterbinden eigener Unzulänglichkeiten ist es durchaus.



## Variable Listenbereiche

Vielleicht wollen Sie für die Datenprüfung auch verschiedene Listen festlegen, die in Abhängigkeit einer anderen Zelle verwendet werden. Dann sollten Sie das folgende Beispiel genauer ansehen.

Für die Auswahl einzelner Entgeltgruppen soll eine Gültigkeitsprüfung festgelegt werden. Die Auswahl der Entgeltgruppe soll dabei so eingeschränkt werden, dass die für die jeweilige Entgeltmethode gültige Liste verwendet wird. Die Tabelle 4.1 beschreibt die Regeln.

| Entgeltmethode         | Entgeltgruppe aus der Liste |
|------------------------|-----------------------------|
| A (Akkord)             | Lohn                        |
| P (Prämie)             | Lohn                        |
| Z (Zeitlohn)           | Lohn                        |
| G (Gehalt)             | Gehalt                      |
| GP (Gehalt mit Prämie) | Gehalt                      |

**Tabelle 4.1:** Für Akkord-, Prämien- u. Zeitlöhner wird die Auswahl auf die Lohn-Entgeltgruppen und für Gehaltsempfänger wird die Auswahl auf die Gehalts-Entgeltgruppen eingeschränkt

Die Lösung verwendet Namen, d.h., auf dem Blatt *Entgelttabelle* trägt der Listenbereich mit den Lohngruppen den Namen *Lohn* und der Bereich mit den Gehaltsgruppen trägt den Namen *Gehalt*.

| Abhängige Gültigkeitsliste                                                                          |                     |
|-----------------------------------------------------------------------------------------------------|---------------------|
| Entgeltmethode:<br>A = Akkord<br>P = Prämie<br>Z = Zeitlohn<br>G = Gehalt<br>GP = Gehalt mit Prämie | Lohn-/Gehaltsgruppe |
| G                                                                                                   | M4                  |
| A                                                                                                   | LG3                 |
| A                                                                                                   | LG4n20              |
| P                                                                                                   | LG3n19              |
| P                                                                                                   | LG3n19              |
| G                                                                                                   | LG3n20              |
| A                                                                                                   | LG3                 |
| A                                                                                                   | LG4n19              |
| A                                                                                                   | LG4n20              |
| GP                                                                                                  | LG4                 |
| Z                                                                                                   | LG5n19              |
| Z                                                                                                   | LG5n20              |
| Z                                                                                                   | LG9                 |
| Z                                                                                                   | LG3n19              |

**Abbildung 4.21:** Die Auswahlliste ist von der gewählten Entgeltmethode abhängig

Die Gültigkeitsprüfung für die Entgeltmethode im Bereich *B4:B15* ist überschaubar. Hier reicht ein fester Eintrag. Unter *DATEN/Datenüberprüfung* wurde unter *Zulassen* der Eintrag *Liste* eingestellt und unter *Quelle* wurde diese Liste eingetragen:

A;P;Z;G;GP

Die Gültigkeitsprüfung für die Entgeltgruppe soll nun den Wert der Spalte *B* berücksichtigen. Dazu stellen Sie das Listenfeld *Zulassen* auf *Liste* und tragen als *Quelle* die folgende Formel ein:

```
=WENN(ODER($B4="G";$B4="GP");Gehalt;Lohn)
```

Diese Formel setzt die in Tabelle 4.1 gezeigten Regeln um und zeigt abhängig von der Auswahl der Entgeltmethode die zutreffende Liste.

Das gezeigte Beispiel finden Sie im Ordner `\Ms5-235\Kap04` in der Arbeitsmappe *Kap04\_DP.xlsx* auf dem Arbeitsblatt *Abhängige Liste*. Dazu gehört das Arbeitsblatt *Entgelttabelle*.



## Die Eingabe mit Formeln einschränken

Wenn die bisher vorgestellten Möglichkeiten für die Definition von Gültigkeitsregeln noch nicht ausreichen, können Sie auch Formeln für die Prüfung verwenden. Wenn Sie im Dialogfeld *Datenüberprüfung* im Listenfeld *Zulassen* den Eintrag *Benutzerdefiniert* wählen, können Sie im Eingabefeld *Formel* eine Formel eintragen.

Hier lassen sich Formeln verwenden, die als Rückgabewert einen der Wahrheitswerte *WAHR* oder *FALSCH* zurückgeben. Gibt die Formel *WAHR* zurück, sind die Daten gültig, ist das Ergebnis der Formel der Wahrheitswert *FALSCH*, sind die Daten ungültig und die Fehlermeldung wird angezeigt.

## Doppelte Einträge verhindern

Ein häufiges Problem bei der Pflege von Listen ist die Vermeidung doppelter Einträge. In einer Kundenliste sollen z.B. doppelte Kundennummern verhindert werden, oder in einer Liste mit Aktienkursen soll jedes Datum nur einmal eingetragen werden. Für diese Problemstellung können Sie eine Formel in der Gültigkeitsprüfung verwenden.

Sie wollen im Bereich *B2:B20* sicherstellen, dass jeder Eintrag nur einmal vorkommen kann. Wenn versucht wird, einen Wert mehrfach einzutragen, soll eine Fehlermeldung darauf hinweisen.

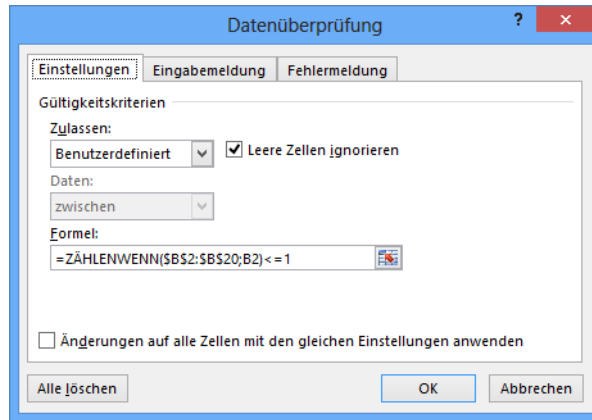
Um die Prüfung so festzulegen, dass jeder Eintrag nur ein einziges Mal verwendet werden kann, gehen Sie wie folgt vor:

1. Markieren Sie zunächst den Prüfbereich *B2:B20*.
2. Wählen den Menübandbefehl *DATEN/Datenüberprüfung*. Auf der Registerkarte *Einstellungen* wählen Sie im Listenfeld *Zulassen* den Eintrag *Benutzerdefiniert*.
3. Im Listenfeld *Formel* fügen Sie den Ausdruck ein, der die Daten, die zugelassen sind, beschreibt. Um doppelte Einträge zu verhindern, verwenden Sie die Formel

```
=ZÄHLENWENN(B2:B20;B2)<=1
```

4. Wechseln Sie zur Registerkarte *Fehlermeldung* und tragen Sie die Fehlermeldung ein.
5. Beenden Sie die Eingabe mit *OK*.

**Wichtig** Achten Sie hier unbedingt darauf, dass der zu durchsuchende Bereich  $\$B\$2:\$B\$20$  als absoluter Bezug eingegeben werden muss, also mit den Dollarzeichen. Dadurch ist dieser Bereich für alle Zellen mit dieser Gültigkeitsprüfung gleich. Das zweite Argument der Funktion ZÄHLENWENN(*Bereich*; *Suchkriterien*) wird mit einem relativen Bezug angegeben. Dieser Bezug soll angepasst werden und auf eine einzelne Eingabezelle zeigen.



**Abbildung 4.22:** Mit dieser Formel wird geprüft, wie oft ein Eintrag im Eingabebereich vorhanden ist

Prüfen Sie doch einmal nach, wie Excel die Formel für die Gültigkeit im Bereich  $B2:B20$  eingetragen hat. In Zelle  $B2$  ist die Formel

=ZÄHLENWENN( $\$B\$2:\$B\$20$ ;B2)<=1

eingetragen. In Zelle  $B3$  dagegen lautet die Formel

=ZÄHLENWENN( $\$B\$2:\$B\$20$ ;B3)<=1

und in Zelle  $B4$

=ZÄHLENWENN( $\$B\$2:\$B\$20$ ;B4)<=1

Der durchsuchte Bereich ist also immer der Bereich  $B2:B20$ , wohingegen das Suchkriterium auf die jeweilige Zelle zeigt. Ob es sich bei dem Eintrag um Text oder Zahlen handelt, spielt bei dieser Gültigkeitsprüfung keine Rolle. Jeder Eintrag wird unabhängig von seinem Datentyp geprüft.



Das gezeigte Beispiel finden Sie im Ordner  $\backslash Ms5-235\Kap04$  in der Arbeitsmappe  $Kap04\_DP.xlsx$  auf dem Arbeitsblatt *Doppel*.

## Hinweis auf vollständige Eingabe zeigen

Die Verwendung von Formeln bietet zahlreiche Möglichkeiten. So können Sie damit auch einen Bereich überwachen und eine Meldung anzeigen, wenn alle Daten eingetragen sind.

Sie wollen eine Meldung anzeigen, wenn alle Felder des Bereichs *B2:B15* gefüllt sind. Um den Hinweis auf vollständige Erfassung der Daten anzuzeigen, gehen Sie wie folgt vor:

1. Markieren Sie den Bereich *B2:B15* und wählen den Menübandbefehl *DATEN/Datenüberprüfung*.
2. Wählen Sie auf der Registerkarte *Einstellungen* im Listenfeld *Zulassen* den Eintrag *Benutzerdefiniert* und tragen im Feld *Formel* diese Formel ein:  
`=ANZAHLEEREZELLEN($B$2:$B$15)>0`
3. Beachten Sie dabei bitte die absoluten Bezüge. Wechseln Sie dann zur Registerkarte *Fehlermeldung* und stellen den Stil *Informationen* ein.
4. Tragen Sie einen Titel, z.B. »Vollständigkeit«, ein und geben Sie im Feld *Fehlermeldung* den gewünschten Text ein.
5. Beenden Sie die Eingabe mit *OK*.

Ist das letzte Feld gefüllt, wird ein Hinweis ausgegeben, der anzeigt, dass die Daten vollständig sind. Sie können dann weitere Aktionen veranlassen oder mit der Auswertung der Daten beginnen.

Das gezeigte Beispiel finden Sie im Ordner `\Ms5-235\Kap04` in der Arbeitsmappe *Kap04\_DP.xlsx* auf dem Arbeitsblatt *Vollständig*.







# Kapitel 5

## Benutzerdefinierte Funktionen

|                                                    |            |
|----------------------------------------------------|------------|
| Eine benutzerdefinierte Tabellenfunktion erstellen | <b>171</b> |
| Die Funktion <i>Kreisfläche()</i>                  | <b>172</b> |
| Die Funktion <i>Viereckfläche()</i>                | <b>179</b> |
| Die Funktion <i>FlächeKrSeg()</i>                  | <b>186</b> |
| Die Funktion <i>Quersumme()</i>                    | <b>187</b> |
| Die Funktion <i>Kreisfläche1()</i>                 | <b>190</b> |
| Die Funktionen im Funktions-Assistenten            | <b>193</b> |
| Funktionen in einem Add-In speichern               | <b>194</b> |

Zahlreiche eingebaute mathematische Funktionen erlauben es Ihnen, mit Excel selbst komplizierte Berechnungen durchzuführen. Was aber, wenn genau die gesuchte Funktion nicht enthalten ist oder wenn Sie statt eines komplizierten Formelkonstrukts lieber eine einfachere Schreibweise vorziehen? Hier setzt dieses Kapitel an und soll Ihnen eine Lösung zeigen.

Wenn man die Nutzung von Excel mit dem Fahren in einem Kraftfahrzeug vergleicht, haben Sie bisher komfortabel am Steuer gesessen und die von den Konstrukteuren eingebauten Regler und Anzeigen verwendet. Viele Bedienungselemente lagen offen im Armaturenbrett, einige mehr verborgen, andere wiederum konnte man erst nach Öffnen einer Abdeckung betätigen. Aber alles, was Sie nutzten, war vom Werk aus vorgesehen.

Wollen Sie etwas benutzen, was mit dem Fahrzeug nicht mitgeliefert wurde, müssen Sie es selbst einbauen. Dazu steigen Sie aus, gehen um das Fahrzeug herum und öffnen den Motorraum. Dort bauen Sie mit dem passenden Werkzeug an geeigneter Stelle das Zusatzaggregat ein. Wenn Sie alles richtig gemacht haben, ist Ihr Fahrzeug um ein Detail reicher.

### Was ist ein Makro?

Makro ist das Kurzwort für »Makrobefehl«, weshalb es im Grunde »der Makro« heißen muss. Es hat sich aber eingebürgert, »das Makro« zu sagen.

Ein Makrobefehl ist eine Befehlsfolge, die schriftlich definiert oder aufgezeichnet wird und dann unter einem Namen oder Kürzel gespeichert werden kann, um häufig wiederkehrende Abläufe mit festen Befehlssequenzen zu vereinfachen. Wenn diese Befehlsfolge erneut ablaufen soll, braucht man lediglich das Makro über dessen Namen oder einer zugewiesenen Tastenkombination aufzurufen, und die Befehle werden in der vorgegebenen festen Reihenfolge ausgeführt.

Anfänglich wurden diese Befehle stur abgearbeitet. Mit der Weiterentwicklung der EDV kamen jedoch Verfeinerungen hinzu, sodass das Makro auch auf Bedingungen, Systemzustände und Benutzereingaben flexibel reagieren konnte. Auf diese Weise sind komplette Makrosprachen entwickelt worden. Allerdings hatte jede Anwendung – auch aus den früheren Office-Paketen – seine eigene Makrosprache. Daraus ist dann Mitte der 1990er Jahre die Programmiersprache VBA hervorgegangen, die diese Makrosprachen im gesamten Office-Paket vereinheitlichen soll. Die anwendungsspezifischen Unterschiede bleiben natürlich weiterhin bestehen. Was soll beispielsweise Word mit der bedingten Formatierung von Excel anfangen?

Bleiben wir bei diesem Bild: Das Cockpit Ihres Excel-Programms ist die Tabellen- bzw. Diagrammansicht und als Armaturenbrett erkennen Sie sicher die Menü- und Symbolleisten; das Firmenemblem befindet sich in der Ecke ganz links oben. – Aber wo ist der Motorraum? Der Motorraum ist (und hier hinkt der Vergleich) werksseitig verschlossen worden, Sie dürfen nur in einen Vorraum hinein. Aber hier können Sie auf (fast) alle eingebauten Details zugreifen und sie für eigene Zwecke nutzen. Dieser Vorraum ist die VBA<sup>1</sup>-Entwicklungsumgebung, auch Visual Basic-Editor genannt: ein Modul, das aus den Makrosprachen früherer Office-Anwendungen und der Programmiersprache Basic zu einem mächtigen Werkzeug entwickelt wurde, mit dem Sie alle nur erdenklichen Erweiterungen für die verschiedenen

<sup>1</sup> VBA: »Visual Basic for Applications« heißt die Programmiersprache der in das Office-Paket integrierten Entwicklungsumgebung (in Excel seit der Version 97).

Office-Produkte, also auch für Excel, programmieren können. Wie viel Sie über VBA wissen müssen, hängt von dem Schwierigkeitsgrad Ihres Projekts ab. Gerade die Lösung mathematischer Probleme verlangt da nicht allzu viel von Ihnen.

## Eine benutzerdefinierte Tabellenfunktion erstellen

Mit VBA können Sie nicht nur Tabellen- oder Diagrammarbeiten von Excel automatisch erledigen lassen, sondern auch benutzerdefinierte Funktionen schreiben. Realisiert werden diese Funktionen über eine spezielle Form von VBA-Prozeduren, eben eine »Funktion« (engl.: *function*).

Die wesentlichen Merkmale einer Funktion kennen Sie aus den Formeln in den Tabellenblättern. Eine Funktion besitzt immer einen eindeutigen Namen, immer ein rundes Klammernpaar und meistens ein oder mehrere Argumente, die jeweils durch ein Semikolon getrennt werden; gewöhnlich gibt eine Funktion genau einen Wert als Ergebnis zurück. Allgemein ausgedrückt bedeutet das: *Funktionsname(Arg1; Arg2; ...)*. Die Anzahl der Argumente hängt von der Aufgabe und dem Einsatz der jeweiligen Funktion ab.

Zum Erstellen einer Funktion benötigen Sie:

- ▶ einen Funktionsnamen
- ▶ die Anzahl der Argumente und ihre Benennung
- ▶ den Rechenweg (Algorithmus), der zum Ergebnis führt

Die ersten beiden Punkte sind für einen Anwender in der Tabelle sichtbar, während ihm der letzte Punkt verborgen bleibt. Was auch niemanden stört, solange die Funktion korrekt arbeitet.

Der prinzipielle Aufbau einer Prozedur in VBA zur Erstellung einer Funktion gestaltet sich so: Eine Funktion wird durch die Anweisung *Function* gefolgt von dem Namen der Funktion, den eventuellen Argumenten und der Typenweisung festgelegt; das ist die sogenannte Funktionsdeklaration. Das Ergebnis von Berechnungen oder Prüfungen wird dem Namen der Funktion über den Operator »=*» zugewiesen. Das Ende einer Funktion wird über die Anweisung *End Function* angezeigt. Die Einheit vom einleitenden Schlüsselwort *Function* bis zum abschließenden *End Function* wird Funktionsblock genannt. Mit diesem groben Abriss können Sie sicher noch nichts anfangen. Lassen Sie uns lieber ein paar Beispiele durchgehen.*

Das Beispiel *Kreisfläche()* soll Ihnen zeigen,

- ▶ wie man die integrierte Entwicklungsumgebung zur Erstellung benutzerdefinierter Funktionen nutzt,
- ▶ wie Parameter als Argumente an die Funktion übergeben werden,
- ▶ wie Berechnungen mit dem Argument und Konstanten durchgeführt werden und
- ▶ wie das Ergebnis der Berechnung in das Arbeitsblatt gelangt.

Mithilfe des Beispiels *Viereckfläche()* erlernen Sie

- ▶ den Umgang mit mehreren Argumenten und
- ▶ den Umgang mit optionalen Argumenten und
- ▶ das Auswerten von Bedingungen mit *If*-Abfragen.

Die Funktion *FlächeKrSeg()* ist ein Beispiel für

- ▶ das Bereithalten von Standardwerten für fehlende optionale Argumente und
- ▶ das Auswerten von Bedingungen mit *If*-Abfragen.

Beim Beispiel *Quersumme()* lernen Sie

- ▶ den Einsatz von Schleifen
- ▶ und VBA-Funktionen
- ▶ und den Umgang mit Variablen

kennen.

Das Beispiel *Kreisfläche1()* führt Ihnen

- ▶ den Einsatz der eingebauten Tabellenfunktionen in benutzerdefinierten Funktionen

vor. Schließlich werden Ihnen die Möglichkeiten gezeigt, mithilfe des Funktions-Assistenten und von Add-Ins Excel in Ihren Arbeitsmappen Ihre eigenen Funktionen zur Verfügung zu stellen.

**Hinweis** Für die folgenden Erläuterungen sollten Sie die Registerkarte *ENTWICKLERTOOLS* einblenden.

Bei Excel 2013 und 2010 klicken Sie dazu auf der Registerkarte *DATEI* auf *Optionen*, dann auf *Menüband anpassen*. Rechts unter dem Text *Menüband anpassen* wählen Sie die Listeneinträge *Alle Registerkarten* oder *Hauptregisterkarten* aus. In der Liste darunter versehen Sie das Kontrollkästchen *Entwicklertools* mit einem Häkchen.

Bei Excel 2007 klicken Sie auf die *Office*-Schaltfläche, dann auf die Schaltfläche *Excel-Optionen* und achten darauf, dass die Kategorie *Häufig verwendet* ausgewählt ist. Dann aktivieren Sie das Kontrollkästchen *Entwicklerregisterkarte in der Multifunktionsleiste anzeigen*. Nach der Bestätigung mit *OK* wird die Registerkarte *Entwicklertools* rechts in der Multifunktionsleiste angezeigt.

## Die Funktion *Kreisfläche()*

In den folgenden sieben Punkten finden Sie eine ausführliche Anleitung zum Erstellen der benutzerdefinierten Funktion *Kreisfläche(Radius)*<sup>2</sup>, über deren Aufruf Sie in einer Excel-Tabelle die Fläche eines Kreises berechnen lassen können, wenn Sie als Argument den Radius in Form einer Zahl, eines berechneten Ausdrucks oder eines Zellbezugs übergeben.

Hier sehen Sie nun alle Schritte im Detail:

1. Mit der Tastenkombination Alt + F11 rufen Sie in allen Excel-Versionen den Visual Basic-Editor auf.

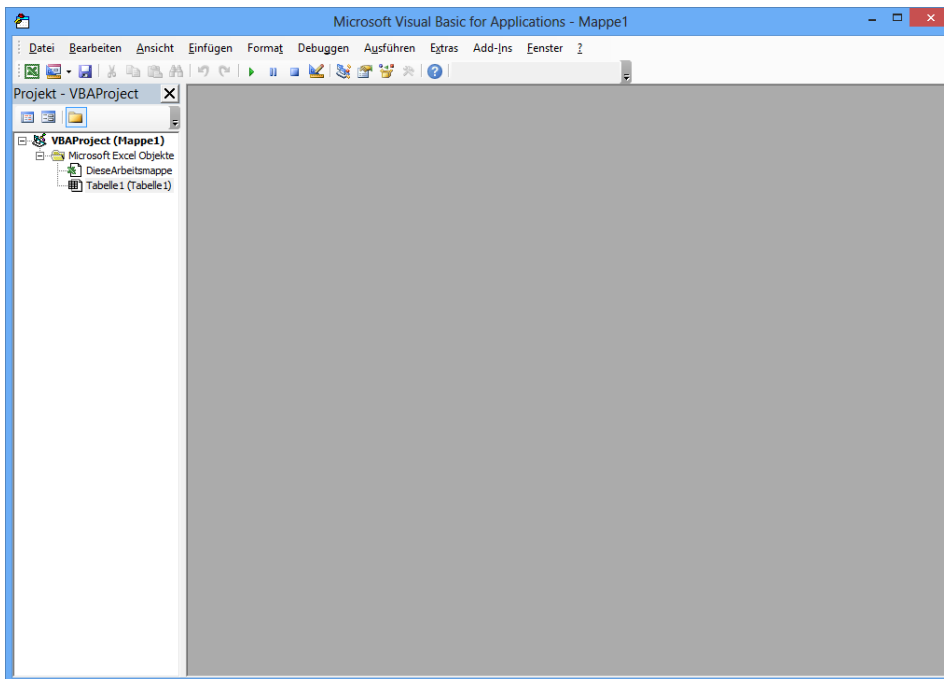
<sup>2</sup> Die Schreibweise der Argumente, normal oder fett, entspricht der späteren Darstellung im Funktions-Assistenten.

Sie erreichen den Visual Basic-Editor auf der Registerkarte *ENTWICKLERTOOLS* ganz links mit der Schaltfläche *Visual Basic*.

Sie können, wie bei Windows-Fenstern üblich, oben die Titelleiste, die Menüleiste und die Symbolleisten sehen. Am linken Rand befinden sich der Projekt-Explorer und das Eigenschaftsfenster untereinander. Sollte das Eigenschaftsfenster fehlen, können Sie es leicht mit der Funktionstaste **F4** öffnen. Die große graue Fläche ist der Arbeitsbereich. Er ist im Augenblick noch leer (Abbildung 5.1).

Im Projekt-Explorer sehen Sie (angelehnt an den Stil der Ordner-Ansicht des Windows-Explorers) Ihre Excel-VBA-Projekte: Zu jeder geöffneten Excel-Arbeitsmappe gibt es einen Ordner mit dem Namen *VBAProject*, gefolgt von dem in Klammern gesetzten Namen der jeweiligen Arbeitsmappe, z.B. *Geometrie.xlsx*, oder, wenn die Arbeitsmappe noch nicht gespeichert ist, *Mappe1*. Zusätzlich gibt es standardmäßig den Unterordner *Microsoft Excel Objekte* mit den Objekten *DieseArbeitsmappe* und *Tabelle1*. Die Anzahl und Namen dieser Unterordner können je nach Ihrer Arbeit im Visual Basic-Editor variieren. Im folgenden Schritt fügen Sie einen neuen Unterordner mit dem Namen *Module* mit dem Objekt *Modul1* hinzu.

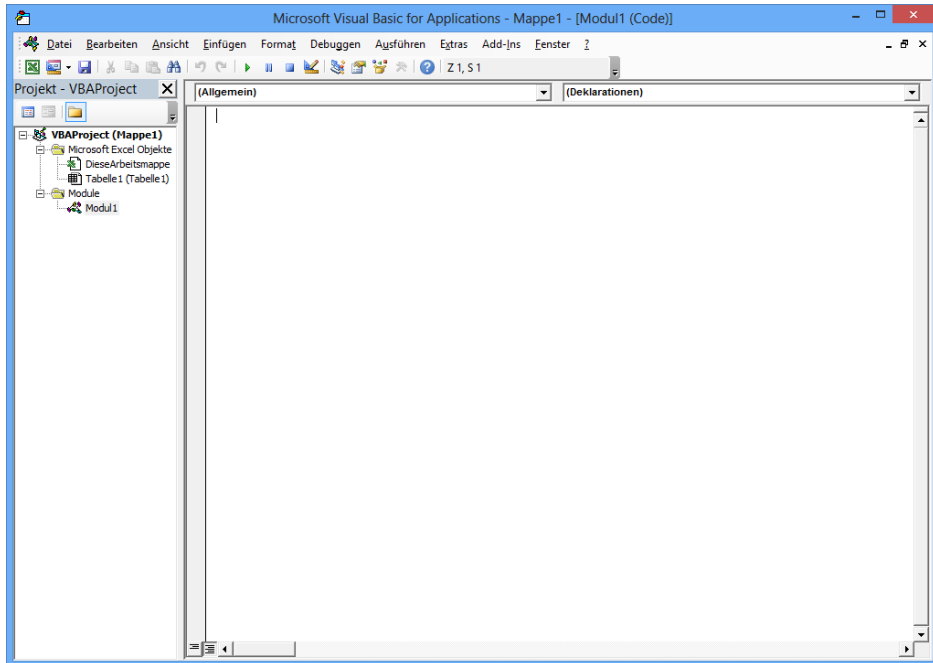
Das Eigenschaftsfenster zeigt Ihnen für das jeweils ausgewählte Excel-Objekt dessen Eigenschaften und ihre aktuellen Werte an. Zum Nacharbeiten des Folgenden ist das Verständnis des Projekt-Explorers und des Eigenschaftsfensters nicht erforderlich.



**Abbildung 5.1:** Der Visual Basic-Editor nach seinem ersten Aufruf

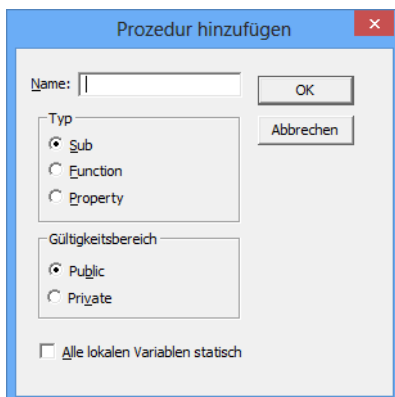
- Über *Einfügen/Modul* bereiten Sie die Arbeitsfläche zur Aufnahme des Befehlscodes vor (Abbildung 5.2)

Mit VBA vertraute Anwender wissen, dass es noch andere Möglichkeiten zur Aufnahme von VBA-Prozeduren gibt. Diese sind zur Erstellung von Tabellenfunktion jedoch nicht geeignet.



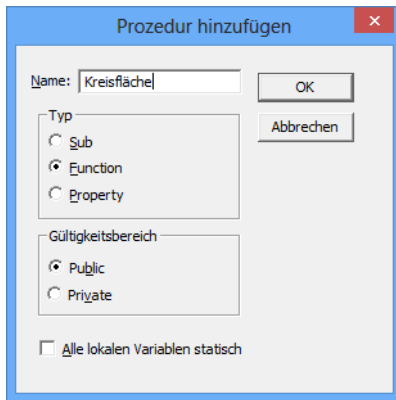
**Abbildung 5.2:** Der Visual Basic-Editor mit einem neuen, aber noch leeren Modul

- Mit *Einfügen/Prozedur* öffnen Sie dann das Dialogfeld aus Abbildung 5.3, in dem Sie die Vorbereitungen für die neue Funktion treffen können.



**Abbildung 5.3:** Das Dialogfeld *Prozedur hinzufügen* mit Standardvorgaben

4. Tippen Sie im Dialogfeld *Prozedur hinzufügen* in das Feld *Name* den Text »Kreisfläche« ein und wählen Sie die Option *Function* aus. Weitere Angaben sind für unseren Fall nicht erforderlich, und mit manchen der angebotenen Optionen würden wir uns sogar Probleme einhandeln (Abbildung 5.4).



**Abbildung 5.4:** Das Dialogfeld *Prozedur hinzufügen* mit den notwendigen Angaben zur Deklaration einer Funktion

Wenn Sie Namen für Ihre eigenen Funktionen vergeben, müssen Sie ein paar Konventionen beachten, denn nicht alle Zeichen, die auf der Tastatur verfügbar sind, eignen sich für Funktionsnamen, weil viele (Sonder-)Zeichen sowohl in Excel-Tabellen als auch innerhalb des Visual Basic-Editors feste Bedeutungen haben wie beispielsweise die Rechenzeichen »+«, »-«, »\*« und »/«.

#### Hinweis

In der deutschen Version von Microsoft Excel können Sie

- ▶ die Buchstaben a...z und A...Z,
- ▶ die Ziffern 0...9,
- ▶ die deutschen Umlaute,
- ▶ das »ß« und
- ▶ den Unterstrich »\_«

verwenden, wobei der Funktionsname mit einem Buchstaben beginnen muss. Groß- und Kleinschreibung können Sie nach Belieben einsetzen. Die Schreibweise wird aus der Funktionsdeklaration automatisch in den Code übernommen, auch wenn Sie dort beim Eingeben eine andere Schreibweise verwenden. Die VBA-Entwicklungsumgebung lässt keine Namen für verschiedene Funktionen zu, die sich nur in der Groß- und Kleinschreibung unterscheiden (das gilt auch entsprechend bei Bereichs- und Variablennamen).

Auch wenn Sie keine Argumente für die Funktion benötigen, ist es innerhalb des Codes unbedingt erforderlich, das Klammersymbol mit anzugeben. An diesen Klammern kann Excel den Funktionsnamen von einem anderen Namen unterscheiden.

5. Wenn Sie die Schaltfläche *OK* betätigen, sehen Sie jetzt im Arbeitsbereich zwei Zeilen entsprechend dem Listing 5.1, zwischen denen sich die Schreibmarke in einer leeren Zeile befindet. Das Schlüsselwort *Public* bewirkt, dass unsere neue Funktion auch von anderen VBA-Modulen verwendet werden darf. Zur ausschließlichen Verwendung der Funktion als Tabellenfunktion ist das Schlüsselwort *Public* nicht erforderlich. Der Einfachheit halber ändern wir nichts an der Vorgabe des Prozedur-Assistenten.

```
Public Function Kreisfläche()
|
End Function
```




**Listing 5.1:** Leerer Prozedurblock nach Angabe des Funktionsnamens

6. Nun müssen Sie das Argument Ihrer Funktion deklarieren und in den Prozedurkopf so einarbeiten wie in Listing 5.2. Als Erstes ist der Name des Arguments anzugeben. Dieser Name wird später vom Funktions-Assistenten angezeigt, wenn Sie mit seiner Hilfe diese Funktion in einer Zelle Ihrer Tabelle eingeben wollen. Diesen Namen können Sie frei wählen; er sollte aber deutlich auf den Zweck des Arguments hinweisen. Mit dem Schlüsselwort *As* gefolgt von *Double* legen Sie den Datentyp unseres Arguments auf »Double« fest, was dem Standardzahlenformat der Excel-Tabellen entspricht. Der Datentyp »Double« heißt auch »Gleitkommazahl mit doppelter Genauigkeit« und wird mit 8 Byte im Bereich von  $-1,79769313486231 \cdot 10^{308}$  bis  $-4,94065645841247 \cdot 10^{-324}$  für negative Werte und im Bereich von  $4,94065645841247 \cdot 10^{-324}$  bis  $1,79769313486231 \cdot 10^{308}$  für positive Werte gespeichert. Etwas weiter hinten in diesem Kapitel wird Ihnen noch der Datentyp »Variant« begegnen, der für optionale Argumente benötigt wird. Es existieren noch weitere Datentypen, die in diesem Zusammenhang allerdings nicht weiter erwähnt werden sollen, weil sie für unsere Zwecke nicht erforderlich sind.

```
Public Function Kreisfläche(Radius As Double)
|
End Function
```

**Listing 5.2:** Prozedurkopf mit der Deklaration des Arguments *Radius*

### Tipp


Wenn Sie beim Eingeben des Programmcodes bestimmte Bezeichner oder Schlüsselwörter benutzen, kommt es häufig vor, dass der Visual Basic-Editor Ihnen mit einem Listenfeld zu Hilfe kommt. Sie können, wenn Sie damit unsicher sind, einfach weiter schreiben. Das Listenfeld verschwindet dann wieder. Sie können aber auch den passenden Begriff aus der Liste in gewohnter Weise auswählen und mit der -Taste die Übernahme in den Code abschließen. Sie gelangen daraufhin mit der Einfügemarke in die nächste Zeile. Wollen Sie allerdings in derselben Zeile bleiben und dort weiter schreiben, müssen Sie die Übernahme des Begriffs mit  +  beenden.

7. Guter Programmierstil ist es auch, einen Datentyp für den Rückgabewert, also für das Ergebnis, zu vergeben, weil der Visual Basic-Editor ansonsten den Typ *Variant* einsetzt. Der Datentyp »Variant« kann jeden anderen Datentyp ersetzen, sodass man flexibler im Umgang mit den Datentypen sein kann; nachteilig wirken sich allerdings dessen höherer Speicherplatzbedarf und langsamere Ausführungsgeschwindigkeit aus. Da wir an dieser Stelle nicht mit anderen Datentypen als dem Excel-Standardformat rechnen müssen, verwenden wir am besten den Datentyp *Double*. Schreiben Sie daher *As Double* zusätzlich hinter die geschlossene Klammer (Listing 5.3).



```
Public Function Kreisfläche(Radius As Double) As Double
|
End Function
```

**Listing 5.3:** Prozedurkopf mit der Deklaration des Arguments *Radius* und des Rückgabetyps

8. Nun geben Sie zwischen diesen beiden Zeilen die eigentliche Prozedur, also den ausführbaren Code, ein (Listing 5.4). Eine Einrückung mit der -Taste am Anfang der Zeile hält den Code übersichtlich.

```
Public Function Kreisfläche(Radius As Double) As Double
 Kreisfläche = 3.14159265358979 * Radius ^ 2
End Function
```

**Listing 5.4:** Komplette Prozedur für die Funktion *Kreisfläche()*

Damit haben Sie die benutzerdefinierte Funktion *Kreisfläche()* mit dem Argument *Radius* vollständig deklariert. Schauen wir uns das Bisherige noch einmal an:

- ▶ Am Ende von Schritt 4 hatten Sie das Gerüst des Funktionsblocks erstellt
- ▶ Mit den Schritten 5 und 6 haben Sie die Funktionsdeklaration vervollständigt
- ▶ Den eigentlichen Programmcode haben Sie in Schritt 7 der Funktion hinzugefügt. Über das Argument *Radius* gelangt der Zahlenwert aus der Tabelle in unseren ausführbaren Code, wo er als gleichnamige Variable verwendet wird. Er wird mit zwei Konstanten in einer Rechenoperation zum Flächenwert verarbeitet und mithilfe des Gleichheitszeichens wird dieses Ergebnis dem Namen der Funktion zugewiesen. Durch die Verwendung dieses Namens in der Tabelle wird dieser Ergebniswert aus dem Programmcode in die Excel-Zelle geholt und steht dort für weitere Schritte zur Verfügung.
- ▶ Die Kreiszahl Pi haben Sie als feste Zahl eingetragen. Mit einer anderen Möglichkeit werden wir uns etwas später in diesem Kapitel beschäftigen.

Hierbei ist noch anzumerken, dass es kein Versehen war, die Konstante für Pi mit einem Punkt als Dezimalzeichen statt wie in Deutschland üblich mit einem Komma zu schreiben. Der Visual Basic-Editor erfordert in den grundlegenden Techniken angloamerikanische Konventionen und englische Sprache, während auf dem Arbeitsblatt weiterhin deutsche Schreibweisen notwendig sind. Deswegen müssen die Argumente der Arbeitsblatt-Funktionen mit einem Semikolon, die Argumente der Funktionen im Visual Basic-Editor mit einem Komma getrennt werden. Die übrigen Regeln für die Schreibweise entsprechen denen für die Eingabe von Formeln in Zellen des Arbeitsblatts (Abbildung 5.5).

An jeder Stelle einer Zeile können Sie im Programmtext durch ein Apostroph (Hochkomma) einen Text einleiten, der vom Visual Basic-Editor zwar angeführt, aber ansonsten ignoriert wird. Er dient zur Kommentierung des Programms. Hier können Sie Hinweise und Erläuterungen zum Programm unterbringen, die zum Verständnis an dieser Stelle beitragen sollen.

**Tipp**

Beachten Sie bitte, dass von der Stelle des Apostrophs bis zum Ende der Anweisung (auch in der nächsten Zeile!) keine ausführbaren Ausdrücke mehr stehen können!

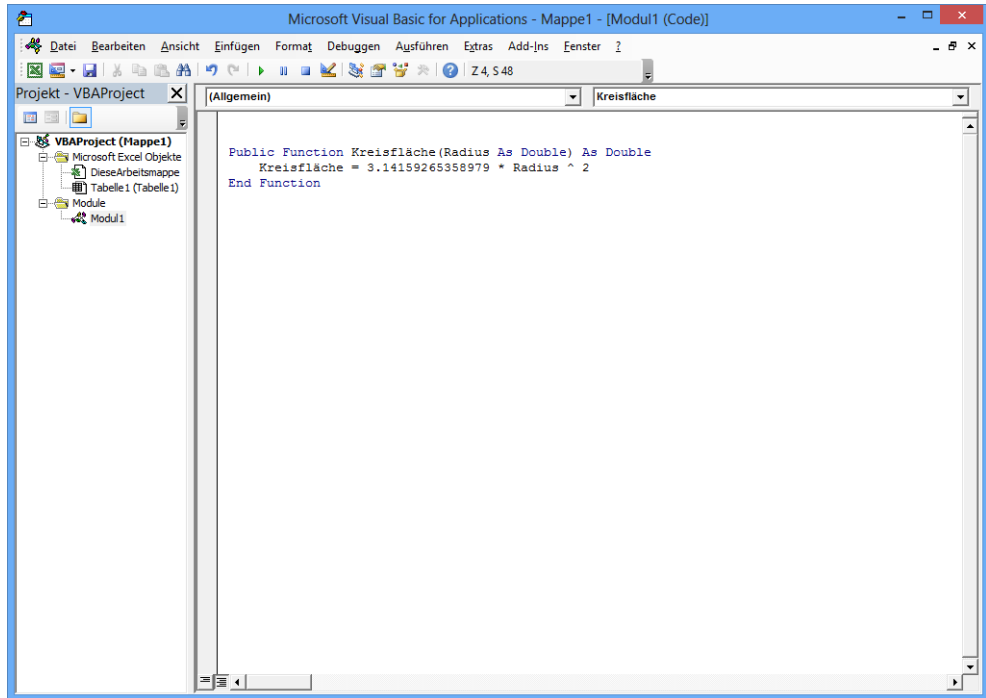


Abbildung 5.5: Der Visual Basic-Editor mit der fertigen Prozedur *Kreisfläche*()

Sicherlich sind Sie neugierig, ob diese Funktion auch arbeitsfähig ist. Um das zu prüfen, schalten Sie zur Tabellenansicht von Excel zurück – entweder über die Taskleiste oder mit der Tastenkombination **Alt + F11**. Sie dürfen den Visual Basic-Editor auch schließen – sogar ohne zu speichern; das VBA-Modul ist Teil Ihrer Arbeitsmappe und wird zusammen mit dieser gespeichert.

Dort tragen Sie bitte in einer Zelle einen Zahlenwert als Radius ein (z.B. die Zahl 2 in der Zelle A1) und in einer anderen Zelle die Formel

=Kreisfläche(A1)

Sollten Sie für den Radius eine andere Zelle als A1 gewählt haben, müssen Sie den Zellbezug in der Kreisflächenfunktion entsprechend einsetzen.

Nach Abschluss der Formeleingabe erhalten Sie sofort das Ergebnis, den Wert für die Fläche des Kreises mit dem Radius, der in Zelle A1 angegeben ist (oder welche Sie gewählt haben): hier also 12,5663706.

Jedes Mal, wenn Sie den Wert in der Zelle, auf die sich die Funktion *Kreisfläche* bezieht, ändern, berechnet Excel das Arbeitsblatt und damit diese selbst definierte Funktion neu. Also verhält sich die benutzerdefinierte Funktion exakt wie eine fest installierte Tabellenfunktion.

Die benutzerdefinierten Funktionen werden Ihnen auch unter

- ▶ Versionen 2007-2013: Registerkarte *FORMELN*, Schaltfläche *Funktion einfügen* – Symbol:  $f_x$
- ▶ Versionen 2002/2003: bei *AutoSumme/Weitere Funktionen*
- ▶ Versionen bis 2000: im Funktions-Assistenten – Symbol:  $f_x$

in der Kategorie *Benutzerdefiniert* in alphabetischer Reihenfolge unter Angabe der Argumente angezeigt.

Dieses Beispiel finden Sie Ordner `\Ms5-235\Kap05` in der Arbeitsmappe *Geometrie.xlsm* auf dem Arbeitsblatt *Kreis*. In *Modul1* dieser Arbeitsmappe können Sie mithilfe des Visual Basic-Editors den Code dazu einsehen.



## Die Funktion *Viereckfläche()*

Mit der Funktion *Viereckfläche(Kante1;Kante2;Höhe)* soll die Fläche eines beliebigen Vierecks mit mindestens zwei parallelen Kanten (Quadrat, Rechteck, Parallelogramm oder Trapez) berechnet werden. Hierbei geht es weniger darum, Ihnen die Formeln der ebenen Geometrie bei ihrer Anwendung zu erleichtern, als vielmehr um die Vorstellung neuer Techniken.

Anhand dieser Funktion soll Ihnen gezeigt werden, wie Sie innerhalb benutzerdefinierter Funktionen

- ▶ mehrere Argumente,
- ▶ optionale Argumente und
- ▶ logische Bedingungen

handhaben können.

## Funktionen mit mehreren Argumenten

Wenn auch zahlreiche Funktionen wie z.B. die trigonometrischen Funktionen und unser vorhergehendes Beispiel *Kreisfläche()* mit einem Argument auskommen, gibt es viele Funktionen, die zur Berechnung ihres Rückgabewerts zwei oder mehr Argumente benötigen. Für die Berechnung der Fläche eines Trapezes sind drei Werte notwendig, die als Argumente unserer Funktion *Viereckfläche()* übergeben werden müssen.

Für die Deklaration mehrerer Argumente ist auch die erste Zeile des Funktionsblocks vorgesehen, wo Sie die Argumente als mit Kommas getrennte Aufzählung zwischen den Klammern eingeben können. Anders als beim Aufruf einer Funktion mit mehreren Argumenten innerhalb eines Arbeitsblatts müssen im Visual Basic-Editor die Argumente jeweils mit einem Komma getrennt werden, da hier die angloamerikanischen Konventionen gelten. Abgesehen von einer gleich zu erwähnenden Ausnahme ist Ihnen die Reihenfolge der Argumente frei gestellt. Wenn nun die neue Funktion *Viereckfläche()* heißen und mit den drei Argumenten *Kante1*, *Kante2* und *Höhe* ausgestattet werden soll, kann die Deklaration der Funktion so aussehen:

```
Public Function Viereckfläche(Kante1, Kante2, Höhe)
```



Wenn Sie mit den Formeln für die Flächenberechnungen etwas unsicher sind, sehen Sie doch einfach im Ordner `\Ms5-235\Kap05` in der Arbeitsmappe *Geometrie.xlsm* auf dem Arbeitsblatt *Grundlagen* nach. Dort sind Flächenberechnungen erläutert.

Allerdings brauchen wir für unsere Funktion nicht in jedem Fall alle drei Argumente.

- ▶ Ein Quadrat hat nur Seiten von gleicher Länge, d.h. nur eine Seitenlänge: Ein Argument genügt hier.
- ▶ Für das Rechteck und das Parallelogramm müssen wir zwei Seiten bzw. die Länge eines Paares der Parallelen und ihren Abstand, also jeweils zwei Argumente übergeben
- ▶ Nur beim Trapez müssen alle drei Argumente angegeben werden

Damit kommen wir zum nächsten Punkt.

## Funktionen mit optionalen Argumenten

Diese Überschrift nennt den Fachbegriff für die Argumente, die nicht bei jedem Funktionsaufruf benötigt werden. Weil beim Berechnen der Funktion im Arbeitsblatt nicht auf den Programmcode Einfluss genommen werden kann, muss man schon beim Erstellen der Funktion im Visual Basic-Editor die Funktion darauf vorbereiten, dass es vorkommen kann, dass bestimmten Argumenten zur Laufzeit keine Werte übergeben werden. Diese Argumente werden durch das Schlüsselwort *Optional*, das in der Funktionsdeklaration jedem Namen eines optionalen Arguments vorangestellt wird, gekennzeichnet. Generell muss zuerst die Deklaration der erforderlichen Argumente abgeschlossen sein, ehe optionale Argumente deklariert werden dürfen. Diese Reihenfolge ist zwingend.

Die Deklaration einer hypothetischen Funktion *Eigenbau* mit den zwei erforderlichen Argumenten *Immer1* und *Immer2* und den zwei optionalen Argumenten *Manchmal1* und *Manchmal2* sähe dann so aus:

```
Public Function Eigenbau(Immer1, Immer2, Optional Manchmal1, Optional Manchmal2)
```

Wie Sie hier sehen können, ist für jedes optionale Argument das Schlüsselwort *Optional* anzugeben. Übertragen auf unsere Funktion *Viereckfläche* erhalten wir unter Einbeziehung der Typdeklaration *Variant* (s.u.) für die Argumente und für die Funktion

```
Public Function Viereckfläche(Kante1 As Double, Optional Kante2 As Variant, _
Optional Höhe As Variant) As Double
```

Nach dieser Funktionsdeklaration »weiß« Excel, dass von nun an eine benutzerdefinierte Funktion namens *Viereckfläche* mit den drei Argumenten *Kante1*, *Kante2* und *Höhe*, von denen die beiden letzten auch fehlen dürfen, existieren soll, und dass sowohl die Funktion als auch das erste Argument das Excel-Standardzahlenformat verwenden sollen.

**Hinweis** Eine Besonderheit in diesem Beispiel für eine Deklaration ist noch zu erwähnen. Die Grundregel ist es, eine Anweisung in eine Zeile zu schreiben. Dieses Gebot kann nicht immer eingehalten werden und oft ist es auch übersichtlicher, eine Anweisung in die folgende Zeile umzubrechen. Dazu schreibt man die Zeichenfolge `[Leertaste]+[ ]` an das Ende der ersten Zeile dieser Anweisung. Der Visual Basic-Editor interpretiert dann die beiden mit dieser Zeichenfolge getrennten Zeilen als eine Anweisung. Auf diese Weise können auch mehr als zwei aufeinander folgende Zeilen verbunden werden. Die oben aufgeführte Funktionsdeklaration ist daher **eine** Anweisung, obwohl sie aus **zwei** Textzeilen besteht.

Was Excel jedoch noch nicht »weiß«, ist, wie es mit fehlenden Übergabewerten bei optionalen Argumenten umgehen soll. Wir müssen unsere Rechenvorschrift so abfassen, dass alle möglichen Kombinationen von genutzten und ungenutzten Argumenten im Programmcode berücksichtigt werden. Dazu brauchen wir die Möglichkeit, feststellen zu können, welches Argument mit einem Übergabewert versehen wurde und welches nicht.

Zu diesem Zweck gibt es die VBA-Funktion<sup>3</sup> *IsMissing()*, die ein spezielles Attribut eines optionalen Arguments abfragen kann, um bei fehlendem Argument den Wahrheitswert *True* zurückzugeben, ansonsten *False*. Um was für ein spezielles Attribut handelt es sich dabei? Dem Wert, der mithilfe eines Arguments einer Funktion bekannt gemacht wird, ist es nicht anzusehen, ob es ein korrekt übermittelter oder ein »weggelassener« Wert ist. Jeder Wert eines Arguments kann beabsichtigt sein; man kann keinen besonderen Wert reservieren, der als Hinweis auf ein nicht angegebenes Argument dienen könnte. Diese Information muss zusätzlich zum Inhalt des Arguments übermittelt werden. Mit »reinen« Datentypen geht das jedoch nicht; sondern nur mit Daten vom Typ »Variant«. Unter anderem haben sie deswegen einen höheren Platzbedarf. Damit wären wir bei dem Pferdefuß der optionalen Argumente: Sie erfordern immer den Datentyp »Variant«. Die Deklaration von optionalen Argumenten als *Double* ist also nicht möglich. Weil *Variant* der Standarddatentyp ist, kann man auf eine explizite Deklaration verzichten, was im Folgenden auch zur besseren Übersicht gemacht wird.

Allerdings sind wir damit noch nicht viel weiter gekommen: Wie kommen wir zum passenden Rechenweg für eine Quadrat-, Rechteck-, Parallelogramm- oder Trapezfläche?

Die Antwort steckt in der speziell für dieses Problem entwickelten VBA-Funktion *IsMissing()*. Je nach Sachlage nimmt ihr Ergebnis *False* oder *True* an. Wenn diese Funktion nun den Wert *False* zurückgibt, bedeutet dies, dass das optionale Argument korrekt mit einem Wert versehen ist und in der folgenden Rechnung verwendet werden darf. Erhalten wir dagegen das Ergebnis *True*, darf das Argument bzw. die daraus abgeleitete Variable in der folgenden Rechnung nicht verwertet werden, weil kein gültiger Wert dafür vorhanden ist. Derartige Fallunterscheidungen werden in der Programmiertechnik in Verzweigungen mit logischen Bedingungen durchgeführt, von denen es mehrere Varianten gibt, die zum großen Teil auch in VBA eingesetzt werden können.

## Verzweigungen mit logischen Bedingungen

Zur Lösung unseres Problems genügt die einfachste Form der logischen Bedingungen. Sie lehnt sich sehr nahe an unsere sprachliche Ausdrucksweise an – allerdings in Englisch. Zur sprachlichen Formulierung von Bedingungen dient das Wort »wenn« (oder das Synonym »falls«), für die Einleitung der Alternative stehen uns »sonst« und »andernfalls« zur Verfügung. Demnach könnten wir die Aufgabe dem Programm so mitteilen: »Wenn das Argument Höhe fehlt, berechne die Fläche für Quadrat oder Rechteck, sonst berechne die Trapez- oder Parallelogrammfläche.« Ins Englische übersetzt und unter Berücksichtigung der VBA-Funktion *IsMissing()* lautet unsere logische Bedingung dann so: *If IsMissing(Höhe) Then*, wonach die Anweisungen für die erste Berechnung folgen müssten. Daraufhin folgt das Schlüsselwort

<sup>3</sup> Eine Funktion, die nur im VBA-Modul und nicht innerhalb des Arbeitsblatts verwendet werden kann.

Else, das die zweite Berechnungsalternative einleitet. Mit `End If` wird dieser sogenannte *If*-Block abgeschlossen. Allgemein ausgedrückt sähe dann ein kompletter *If*-Block entsprechend dem Listing 5.5) aus.

```
If Bedingung Then
 Anweisungsfolge1
Else
 Anweisungsfolge2
End If
```

**Listing 5.5:** Ein *If*-Block in allgemeiner Darstellung

Die »Bedingung« ist ein logischer Ausdruck, der den logischen Wert (den Wahrheitswert) *True* oder *False* liefert. Das kann ein logischer Vergleich mit einem Vergleichsoperator wie » $X > 2$ « oder »Alter  $\leq 50$ « sein oder eine Funktion, die einen Wahrheitswert zurückgibt, wie *IsMissing()*. »Anweisungsfolge1« und »Anweisungsfolge2« sind die beiden Rechenwege, die alternativ zum Zuge kommen, je nachdem, welchen Wahrheitswert die Auswertung der logischen Bedingung ergibt.

Die Einrückungen sind nicht Pflicht, erhöhen aber die Lesbarkeit und damit die Verständlichkeit des Programms. Eine Zusammenfassung der verschiedenen Möglichkeiten, Verzweigungen in benutzerdefinierte Funktionen einzubauen, gibt der folgende Überblick über die *If*-Anweisungen in VBA.

### Überblick über die *If*-Anweisungen in VBA

Die *If*-Anweisung lässt sich in mehreren Varianten verwenden:

► Die einfachste Variante

```
If Alter >= 50 Then Merkmal = 1
```

»Alter« und »Merkmal« sind zwei Variablen, die Zahlenwerte beinhalten. Wenn die Variable »Alter« eine Zahl enthält, die größer oder gleich 50 ist, wird der Variablen »Merkmal« der Wert 1 zugewiesen. Hat die Variable »Alter« einen kleineren Wert als 50, wird die Anweisung hinter *Then* übergangen und es geht einfach hinter der *If*-Anweisung mit der nächsten Zeile weiter. Hinter dem Schlüsselwort *Then* darf nur eine Anweisung stehen, weil wir nur eine Zeile zur Verfügung haben. Deshalb ist auch keine *End If*-Anweisung nötig (und möglich).

► Die ausführliche Variante

```
If Alter >= 50 Then
 Merkmal = 1
End If
```

Programmtechnisch ist dies dasselbe wie im vorherigen Punkt. Nur die Darstellung ist übersichtlicher, allerdings auch nicht so Platz sparend. Die einzeilige Anweisung ist in mehrere Zeilen aufgeteilt worden, sodass mehrere Anweisungen hinter dem Schlüsselwort *Then* möglich sind. Daher muss das Ende der von der Bedingung abhängenden Anweisungen mit einer *End If*-Anweisung gekennzeichnet werden. ►

► Der vollständige *If*-Block

```
If Alter >= 50 Then
 Merkmal = 1
Else
 Merkmal = 2
End If
```

Hier werden beide Möglichkeiten der Bedingung verwertet. Ist die Bedingung »Alter >= 50« wahr, erhält »Merkmal« den Wert 1. Wenn die Bedingung nicht wahr ist, wird ihr der Wert 2 zugewiesen. Sowohl zwischen *Then* und *Else* als auch zwischen *Else* und *End If* können mehrere Anweisungen stehen:

```
If Alter >= 50 Then
 Merkmal = 1
 Kennung = "K17"
Else
 Merkmal = 1
 Kennung = ""
End If
```

*Kennung* ist eine Textvariable, der im ersten Fall der String (eine Zeichenfolge) *K17*, im zweiten Fall dagegen ein »Leerstring«, sozusagen nichts zugewiesen wird.

► Die »hässliche« Variante

```
If Alter >= 50 Then Merkmal = 1 : Kennung = "K17" Else Merkmal = 1 : Kennung = ""
```

Dies stellt keine Empfehlung dar, sondern soll nur der Vollständigkeit halber erwähnt sein. Sie ist in ihrer Wirkung äquivalent zum vorherigen Punkt. Sicherlich die kürzeste Variante, aber das Programm wird dadurch schwer lesbar. Der Doppelpunkt erlaubt, mehrere Anweisungen in eine Zeile zu schreiben, die ansonsten in getrennte Zeilen gehörten. Er übernimmt innerhalb einer Zeile die Funktion der Zeilenendeschaltung: das Kennzeichnen eines Anweisungsendes.

Damit sind wir in der Lage, den von der *IsMissing()*-Funktion zurückgegebenen Wahrheitswert auszuwerten und die Entscheidung, was beim Fehlen eines Arguments geschehen soll, in den Programmteil unserer Funktion einzuarbeiten.

Allerdings ist die Sache noch etwas komplizierter. Wenn wir auf diese Weise herausbekommen, dass das Argument *Kante2* fehlt, haben wir immer noch keine eindeutige Situation. Denn sowohl das Quadrat als auch das Parallelogramm kommen zur Flächenberechnung ohne das Argument *Kante2* aus. Daher entscheidet das Vorhandensein des Arguments *Höhe*, ob wir die Fläche eines Quadrats berechnen wollen und deswegen nur ein Argument brauchen oder ob wir für das Parallelogramm noch den Wert *Höhe* übergeben müssen.

Auch wenn das Argument *Kante2* angegeben wird, haben wir immer noch zwei Alternativen: Rechteck und Trapez. Erst das Argument *Höhe* entscheidet, welches der beiden Vierecke in diesem Fall zur Flächenberechnung herangezogen wird.

Es müssen also zwei *If*-Blöcke in einen übergeordneten *If*-Block geschachtelt werden. Die Tabelle 5.1 soll die Möglichkeiten einander gegenüberstellen.

| Argument »Kante2« | Argument »Höhe« | Viereck        |
|-------------------|-----------------|----------------|
| Nicht vorhanden   | Nicht vorhanden | Quadrat        |
| Nicht vorhanden   | Vorhanden       | Parallelogramm |
| Vorhanden         | Nicht vorhanden | Rechteck       |
| Vorhanden         | Vorhanden       | Trapez         |

**Tabelle 5.1:** Mit optionalen Argumenten kann die Art des zu berechnenden Vierecks festgelegt werden

Der erste *If*-Block (Fehler! Verweisquelle konnte nicht gefunden werden.) wertet aus, ob das dritte Argument *Höhe* vorhanden ist oder nicht und verzweigt in die passenden Anweisungsblöcke.

```
If IsMissing(Kante2) Then
 ... 'hier stehen die Anweisungen für Quadrat oder Parallelogramm
Else
 ... ' hier stehen die Anweisungen für Rechteck oder Trapez!
End If
```

**Listing 5.6:** Test auf Anwesenheit des Arguments *Kante2*

Die inneren *If*-Blöcke (Listing 5.7 und Listing 5.8) entscheiden je nach Anwesenheit des Arguments *Höhe* über die Berechnung von Quadrat oder Parallelogramm bzw. über die Berechnung von Rechteck oder Trapez.

```
If IsMissing(Höhe) Then
 Viereckfläche = Kante1 ^ 2 'Quadrat
Else
 Viereckfläche = Kante1 * Höhe 'Parallelogramm
End If
```

**Listing 5.7:** Prüfen, ob das Argument *Höhe* angegeben wurde, wenn *Kante2* fehlt

Dieser *If*-Block soll die Position der drei Punkte im ersten *If*-Block einnehmen und der folgende *If*-Block nimmt die Position der drei Punkte im zweiten *If*-Block ein.

```
If IsMissing(Höhe) Then
 Viereckfläche = Kante1 * Kante2 'Rechteck
Else
 Viereckfläche = (Kante1 + Kante2) / 2 * Höhe 'Trapez
End If
```

**Listing 5.8:** Prüfen, ob das Argument *Höhe* angegeben wurde, wenn *Kante2* vorhanden ist



Damit sind alle notwendigen Teile zum Aufbau einer eigenen Funktion *Viereckfläche()* besprochen. Der Reihe nach besprochen gestaltet sich die Funktion so:

- ▶ In der Funktionsdeklaration haben wir den Namen *Viereckfläche* festgelegt und bestimmt, dass diese Funktion drei Argumente (*Kante1*, *Kante2* und *Höhe*) besitzen soll, von denen die letzten beiden optional sein sollen. Die Funktion soll einen Wert im Excel-Standardzahlenformat zurückgeben.
- ▶ Mit der ersten *If*-Abfrage behandeln wir das optionale Argument *Kante2*
- ▶ Das zweite optionale Argument *Höhe* wird im ersten inneren *If*-Block zum ersten Mal ausgewertet. Diese Verzweigung führt zu den passenden Berechnungen von Quadrat und Parallelogramm.
  - ▶ Die darauf folgende Anweisung berechnet die Fläche des Quadrats für den Fall, dass *Höhe* fehlt
  - ▶ Das Schlüsselwort *Else* leitet die Alternative ein
  - ▶ Die darauf folgende Anweisung berechnet die Fläche des Parallelogramms, wenn *Höhe* angegeben wurde
  - ▶ Der erste innere *If*-Block wird in der nächsten Zeile mit der *End If*-Anweisung beendet
- ▶ Das zweite Schlüsselwort *Else* leitet die Alternative ein, wenn *Kante2* angegeben wurde
- ▶ Das zweite optionale Argument *Höhe* wird im zweiten inneren *If*-Block zum zweiten Mal überprüft. Diese Verzweigung führt zu den passenden Berechnungen von Rechteck und Trapez.
  - ▶ Die darauf folgende Anweisung berechnet die Fläche des Rechtecks für den Fall, dass *Höhe* fehlt
  - ▶ Das Schlüsselwort *Else* leitet die Alternative ein
  - ▶ Die darauf folgende Anweisung berechnet die Fläche des Parallelogramms, wenn *Höhe* angegeben wurde
  - ▶ Der zweite innere *If*-Block wird in der nächsten Zeile mit der *End If*-Anweisung beendet
- ▶ Nun kann mit der nächsten *End If*-Anweisung auch der erste *If*-Block abgeschlossen werden
- ▶ Nun ist alles für die eigene Funktion angegeben worden und dies wird dem Visual Basic-Editor mit der Anweisung *End Function* mitgeteilt

Den kompletten Programmcode finden Sie unten in Listing 5.9.

```
Public Function Viereckfläche(Kante1 As Double, Optional Kante2, Optional Höhe) As Double
If IsMissing(Kante2) Then
 If IsMissing(Höhe) Then
 Viereckfläche = Kante1 ^ 2 'Quadrat
 Else
 Viereckfläche = Kante1 * Höhe 'Parallelogramm
 End If
Else
 If IsMissing(Höhe) Then
 Viereckfläche = Kante1 * Kante2 'Rechteck
 Else
 Viereckfläche = (Kante1 + Kante2) / 2 * Höhe 'Trapez
End Function
```

```

End If
End If
End Function

```

**Listing 5.9:** Die benutzerdefinierte Funktion *Viereckfläche()* mit den beiden optionalen Argumenten *Kante2* und *Höhe*

Falls das Argument *Kante1* fehlt, gibt diese selbst definierte Funktion den Fehler *#WERT!* zurück. Damit ist sichergestellt, dass keine fehlerhaften Werte in das Arbeitsblatt übernommen werden, wenn das notwendige Argument *Kante1* nicht angegeben wurde.



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap05` in der Arbeitsmappe *Geometrie.xlsm* auf dem Arbeitsblatt *Viereck*. In *Modul1* dieser Arbeitsmappe können Sie mithilfe des Visual Basic-Editors den Code dazu einsehen.

## Die Funktion *FlächeKrSeg()*

Bei zahlreichen Tabellenfunktionen gibt es Argumente, die optional sind. Aber häufig betrifft dies Argumente, auf die nicht grundsätzlich verzichtet werden soll, sondern die fast immer einen bestimmten Wert besitzen und nur in speziellen Fällen einen davon abweichenden Wert annehmen sollen. Deswegen soll in dem Beispiel *FlächeKrSeg()*

- ▶ das Bereithalten von Standardwerten für fehlende optionale Argumente und
- ▶ das Auswerten von Bedingungen mit *If*-Abfragen

demonstriert werden.

So besitzt die finanzmathematische Funktion zur Berechnung von Annuitäten *RMZ()* fünf Argumente, von denen die letzten beiden optional sind: der Zeitwert *Zw* und die Fälligkeit *F*. Für die Berechnung einer Annuität kann auf diese Argumente allerdings nicht verzichtet werden. Der Anwender ist jedoch von der Pflicht zur Angabe dieser Argumente entbunden. Weil üblicherweise Kredite vollständig zurückgezahlt werden (Zeitwert *Zw* = 0) und die Zinszahlung am Ende der jeweiligen Zahlungsperiode (Fälligkeit *F* = 0), also nachschüssig gezahlt wird, sind diese Standardwerte in der Tabellenfunktion *RMZ()* vorgegeben. Unterlässt der Anwender die Angabe eines dieser Argumente, so wird sein Standardwert in der Berechnung verwendet.

Dieses Verfahren kann man auch bei selbst definierten Funktionen einsetzen.

### Optionale Argumente mit Standardwerten

Die Syntax zum Festlegen von Standardwerten bei optionalen Argumenten folgt diesem Schema:

```
Public Function Eigenbau(Immer1, Immer2, Optional Manchmal, Optional So_oder_so = Wert)
```

Dieses Beispiel zeigt zwei obligatorische und zwei optionale Argumente, von denen das letzte den Standardwert »Wert« zugewiesen bekommt. Dafür werden zu der Deklaration der optionalen Variablen *So\_oder\_so* lediglich ein Gleichheitszeichen und »Wert« als Standardwert hinzugefügt. Zur Ausführungszeit ist die Variable *So\_oder\_so* innerhalb des Programmcodes

in jedem Fall gültig definiert: Entweder wurde das Argument *So\_oder\_so* in der Formel des Arbeitsblatts mit einem Wert versehen oder es wurde mit dem Standardwert aus der Funktionsdeklaration belegt.

Dies soll das Beispiel *FlächeKrSeg(Radius;Phi;WMAß)* illustrieren. Der Zweck dieser Funktion soll die Berechnung der Fläche eines Kreissegments aus dem Radius des Kreises und des Winkels des Segments sein. In der Schule ist es üblich, Winkelangaben in Grad zu machen, während in Wissenschaft und Technik das Bogenmaß (*Radian* oder *rad*) bevorzugt wird. Die Funktion *FlächeKrSeg()* soll nun mit beiden Alternativen umgehen können. Welches von den beiden Winkelmaßen eingesetzt wird, soll über das optionale Argument *WMAß* gesteuert werden. Wird beim Funktionsaufruf das Argument *WMAß* weggelassen, kommt die Standardvorgabe in der Funktionsdeklaration zum Zuge und das als Variable im ausführbaren Code eingesetzte Argument *WMAß* hat den Wert eins. In diesem Fall verzweigt die *If*-Abfrage in die Flächenberechnung, wo das Gradmaß verwendet wird.

Bei jedem anderen Wert von *WMAß* führt die *If*-Abfrage zur Berechnung der Fläche mit dem Winkel *Phi* im Bogenmaß (Listing 5.10). Dies wird durch Einsetzen eines von eins verschiedenen Übergabewerts für das Argument *WMAß* in der Tabellenformel erreicht.

```
Public Function FlächeKrSeg(Radius As Double, Phi As Double, Optional WMAß = 1)
If WMAß = 1 Then
 FlächeKrSeg = 3.14159265358979 * Radius ^ 2 * Phi / 360
Else
 FlächeKrSeg = Radius ^ 2 * Phi / 2
End If
End Function
```

**Listing 5.10:** Die Funktion *FlächeKrSeg()* verwendet das optionale Argument *WMAß* mit einem Standardwert

Sie finden im Ordner *Ms5-235\Kap05* in der Arbeitsmappe *Geometrie.xlsm* auf dem Arbeitsblatt *Kreissegment* dieses Beispiel und auf dem Arbeitsblatt *Grundlagen* einige Erläuterungen dazu. In *Modul1* dieser Arbeitsmappe können Sie mithilfe des Visual Basic-Editors den Code einsehen.



## Die Funktion *Quersumme()*

Mit dem Beispiel *Quersumme(Zahl)* verlassen wir kurz die Geometrie und wenden uns der Arithmetik zu, – obwohl Sie bemerken werden, dass wir innerhalb des VBA-Codes die Arithmetik auch nicht allein anwenden werden. Doch dazu später mehr ...

In diesem Beispiel geht es um

- ▶ den Einsatz von Schleifen
- ▶ und VBA-Funktionen
- ▶ und um den Umgang mit Variablen.

Zur Berechnung der Quersumme einer Zahl nimmt man nacheinander die einzelnen Ziffern dieser Zahl und addiert sie auf. Vielleicht stoßen Sie sich an dem Begriff »nacheinander«. Denn diese Formulierung – wenn sie auch nicht ganz glücklich ist – soll zum eigentlichen Thema überleiten.

Noch einmal ganz langsam: Um eine Quersumme zu berechnen,

1. wird die erste Ziffer aus der Zahl genommen und zu einer Zwischensumme, die augenblicklich noch null ist, hinzuaddiert. Dann wird ...
2. die zweite Ziffer aus der Zahl genommen und zu der Zwischensumme hinzuaddiert. Dann wird ...
3. die dritte Ziffer aus der Zahl genommen und zu der Zwischensumme hinzuaddiert. Dann wird ...
4. die vierte Ziffer aus der Zahl genommen und zu der Zwischensumme hinzuaddiert. Dann wird ...
5. usw.

Das sind wiederkehrende gleichartige Arbeiten. Für derartige Tätigkeiten ist der Rechner geschaffen: Er kann im Gegensatz zum Menschen ermüdungs- und damit fehlerfrei gleichförmige Schritte auch in sehr großer Zahl ausführen. Deswegen gibt es in jeder Programmiersprache Elemente, die es erlauben, dieselben Programmschritte wiederholt auszuführen. Der allgemeine Begriff dafür ist »Schleife« (engl.: loop).

## Programmierung von Schleifen

VBA kennt mehrere Arten von Schleifen, die sich darin unterscheiden, wie die Anzahl der Durchläufe, der Wiederholungen, geregelt ist:

- ▶ *For...Next*
- ▶ *Do While...Loop / Do...Loop While*
- ▶ *Do Until...Loop / Do...Loop Until*<sup>4</sup>

Jede dieser Schleifen besteht aus drei Teilen:

1. Die erste Anweisung, die den Eintritt in die Schleife regelt: der »Schleifenkopf« (in der o.a. Aufzählung vor den drei Punkten verkürzt gezeigt)
2. Die Folge von Anweisungen, die während der Schleifendurchläufe abgearbeitet wird: der »Schleifenrumpf«<sup>5</sup> (durch die drei Punkte dargestellt)
3. Die letzte Anweisung, die das Ende der Schleife bestimmt: der »Schleifenfuß« (nach den drei Punkten verkürzt angeführt)

Bei der *For...Next*-Schleife wird die Anzahl der Durchläufe von einem Zähler bestimmt. Dagegen wird die *While*-Schleife so lange ausgeführt, wie eine Bedingung (hinter *While* angegeben) WAHR ist. Die *Until*-Schleife wiederum wird beendet, wenn die hinter *Until* angegebene Bedingung WAHR wird.

Für die Berechnung der Quersumme kann man die *For...Next*-Schleife gut einsetzen, weil die Zahl der Durchläufe durch die Anzahl der Ziffern vorgegeben ist. Die Bedingung zum Verlassen der Schleife wird nicht erst während der Laufzeit der Schleife ermittelt, sondern steht schon von vornherein fest.

<sup>4</sup> Bei den beiden Varianten der *While*- bzw. *Until*-Schleifen geht es darum, ob die Schleife mindestens einmal (*Do...Loop While*) durchlaufen wird oder nicht (*Do While...Loop*).

<sup>5</sup> Auch »Schleifenkörper«.

Wie die Grundform der *For...Next*-Schleife aussieht, ist in Listing 5.11 gezeigt.

```
Zwischensumme = 0
For I = 1 To 10
 Zwischensumme = Zwischensumme + i
Next I
```

**Listing 5.11:** Eine einfache *For...Next*-Schleife

Mit der ersten Anweisung wird die Variable namens *Zwischensumme* auf null gesetzt. Eine Variable repräsentiert eine Zahl, deren Wert während der Laufzeit des Programms geändert werden kann. Derartige Variablen werden innerhalb des Funktionsblocks zur Aufnahme von Zwischensummen und Hilfswerten eingesetzt. Sie treten nach außen hin nicht in Erscheinung. Für die Formeln des Arbeitsblatts sind Variablen unsichtbar. Es ist nicht nur guter Programmierstil, eine Variable am Anfang des Funktionsblocks zu deklarieren, der Programmierer kann so auch den Datentyp kontrollieren. Die dazu passende Anweisung

```
Dim Zwischensumme As Long
```

würde den Datentyp der Variablen als »Long«, also als Ganzzahl im Bereich von  $-2.147.483.647$  bis  $+2.147.483.647$  bestimmen. Sie ist der Übersichtlichkeit halber in Listing 5.11 weggelassen worden.

Der Schleifenkopf besteht aus der Anweisung

```
For I = 1 To 10
```

Dabei ist *I* der Zähler, der die Zahl der Durchläufe mitzählt. Diese Schleife soll zehnmal durchlaufen werden, wobei der Zähler *I* nacheinander die Werte 1 bis 10 annimmt.

Im Schleifenrumpf kommt dem Zähler hier noch eine weitere Aufgabe zu: Er wird jeweils mit seinem aktuellen Wert zur Zwischensumme hinzuaddiert.

```
Zwischensumme = Zwischensumme + i
```

Der Schleifenfuß mit dem Schlüsselwort *Next* und der erneuten Angabe des Zählers erhöht den Zähler um eins. Wenn dadurch der im Schleifenkopf angegebene Endwert noch nicht überschritten ist, setzt ein neuer Durchlauf ein. Anderenfalls wird mit der nächsten Anweisung hinter der Schleife fortgefahren.

Damit sind alle wichtigen Elemente, die für die Funktion *Quersumme()* benötigt werden, erwähnt worden – mit einer Ausnahme, die aber besser an der fertigen Funktion in Listing 5.12 erklärt werden soll.

```
Public Function Quersumme(Zahl As Double) As Long
```

```
 Dim I As Integer
```

```
 Dim S As String
```

```
 Quersumme = 0
```

```
 S = Str(Zahl)
```

```
 For I = 1 To Len(S)
```

```
 Quersumme = Quersumme + Val(Mid(S, I, 1))
```

```
 Next I
```

```
 End Function
```

**Listing 5.12:** Berechnung der Quersumme einer Zahl

Nach der Deklaration der Funktion *Quersumme()* als Funktion vom Datentyp *Long* mit dem Argument *Zahl* im Excel-Standardzahlenformat enthalten die nächsten beiden Anweisungen Variablendeklarationen. *I* ist die Zählvariable für die *For...Next*-Schleife, deren größter Wert der Stellenanzahl von *Zahl* entspricht.

Neues bietet dagegen die zweite Variable: *S* wird als *String*, als Textvariable bekannt gemacht. Ihr Sinn wird erst beim Betrachten des Schleifenrumpfs klar. Es existiert nämlich keine Funktion in Excel, weder als Tabellenfunktion noch als VBA-Funktion, mit der man die Anzahl der Stellen einer Zahl feststellen kann oder mit der sich einzelne Ziffern aus einer Zahl extrahieren lassen. (Ersteres könnte man noch mit einer Logarithmus-Funktion hinbekommen, Letzteres gelänge dagegen nur sehr umständlich.)

Deshalb wird hier *Zahl* mittels der VBA-Funktion *Str()* in die String-Variable *S* umgewandelt. Bei dieser kann man dann mit der VBA-Funktion *Len()* (von engl.: »length«) die Textlänge ermitteln, was der Ziffernanzahl von *Zahl* und damit der Anzahl der Schleifendurchläufe entspricht.

Innerhalb der Schleife gibt die VBA-Funktion *Mid()* nacheinander die einzelnen Ziffern von *Zahl* als Zeichen heraus, indem sie aus der String-Variablen *S* ab der Stelle *I* jeweils ein Zeichen herauskopiert. (Die Anzahl der zu kopierenden Zeichen werden durch das dritte Argument, hier die Eins, bestimmt.) Zeichen können nicht zu Zahlen hinzuaddiert werden. Deshalb wandelt die VBA-Funktion *Val()* (von engl.: »value«) die als Zeichen dargestellten einzelnen Ziffern von *Zahl* wieder in Zahlen zurück, die zu *Quersumme* hinzugezählt werden können.

Wenn alle Stellen von *Zahl* abgearbeitet worden sind, wird die Schleife verlassen und der Wert der Variablen *Quersumme* wird von der Tabellenfunktion *Quersumme()* der Formel im Arbeitsblatt übermittelt.

In diesem Beispiel findet der Anfänger sicherlich den Umweg über die String-Variable und den Einsatz der dazu gehörenden String-Funktionen *Str()*, *Len()*, *Mid()* und *Val()* sehr verwirrend. Es ist aber dennoch die am wenigsten anspruchsvolle Lösung. Der Einsatz von Schleifen, Variablen und VBA-Funktionen ließ sich anhand weniger Zeilen zeigen.



Sie finden dieses Beispiel im Ordner `Ms5-235\Kap05` in der Arbeitsmappe *Quersumme.xlsm* auf dem Arbeitsblatt *Quersumme*. In *Modul1* dieser Arbeitsmappe können Sie mithilfe des Visual Basic-Editors den Code einsehen. Auf dem Arbeitsblatt *Spezial* befinden sich zwei weitere Lösungen für die Berechnung der Quersumme einer Zahl. Eine Lösung kommt dabei ohne VBA-Funktionen aus.

## Die Funktion *Kreisfläche1()*

Diese Funktion soll dasselbe leisten wie die am Anfang dieses Kapitels beschriebene Funktion *Kreisfläche()*. Sie soll hier aber eine Verbesserung erfahren und Ihnen den Einsatz der integrierten Tabellenfunktionen vorführen.

Die Funktion *Kreisfläche()* benutzte zur Berechnung der Kreisfläche die Kreiszahl Pi als Konstante mit dem Wert 3,14159265358979 (unter Ausnutzung der in Excel maximal möglichen Stellenzahl). Doch nicht immer hat man die Kreiszahl Pi mit allen Nachkommastellen in der notwendigen Genauigkeit zur Hand, sodass man auf zusätzliche Tabellenwerke angewiesen scheint.

Der geübte Excel-Anwender könnte (und sollte) nun dagegenhalten, dass Excel über eine Tabellen-Funktion  $Pi()$  verfügt, die mit hinreichender Genauigkeit (und Geschwindigkeit) den Wert von Pi angibt. Mit den schon beschriebenen benutzerdefinierten Funktionen wurde gezeigt, dass diese beiden Welten – das Excel-Arbeitsblatt und die VBA-Entwicklungs-umgebung – nicht unvereinbar sind. Zahlen können aus Zellen in einem Arbeitsblatt einer VBA-Funktion übergeben werden und die Rechenergebnisse dieser Funktion gelangen wieder in die Zelle zurück. So ist es nicht verwunderlich, dass man in VBA die Möglichkeit hat, eingebaute Tabellenfunktionen zu verwenden.

## Tabellenfunktionen in benutzerdefinierten Funktionen verwenden

In Excel sind über 300 Tabellenfunktionen integriert. Da liegt es nahe, diese auch in eigenen Prozeduren (Funktionen) zu verwenden und damit Programmierarbeit und -zeit einzusparen. Die Vorgehensweise, um innerhalb eines eigenen VBA-Projekts Tabellenfunktionen zu nutzen, erscheint dem Neuling jedoch umständlich und undurchsichtig, obwohl sich für den Anwender im Grunde nichts Kompliziertes dahinter verbirgt. Die umständliche Schreibweise ist lediglich eine Folge der objektorientierten Programmierung, die man zum Einsetzen der eingebauten Tabellenfunktionen aber nicht beherrschen muss.

### Objektorientierte Programmierung

Zu dem Thema »Objektorientierte Programmierung« soll nur so viel gesagt werden:

Um der Fülle an Details in großen Anwendungen während der Programmierung Herr zu werden, führte man eine Hierarchie dieser Details, die man Objekte nennt, ein. Will man auf ein untergeordnetes Objekt Bezug nehmen, muss der gesamte Weg durch die Hierarchie angegeben werden.

Wenn Sie mit Betriebssystemen vertraut sind, kennen Sie das schon: Die korrekte Bezeichnung einer Datei erfordert die Angabe von Laufwerk, Ordner, eventuelle Unterordner und Dateiname – oder wie wäre es vielleicht mit »Laufwerksobjekt«, »Ordnerobjekt« und »Dateiobjekt«?

Nur wird in der Programmierung nicht der Rückwärtsschrägstrich (engl.: back slash) zur Trennung der Objekte eingesetzt, sondern ein Punkt.

Die Schreibweise zur Nutzung der Tabellenfunktion  $Pi()$  in unserer VBA-Funktion lautet<sup>6</sup>:

```
Application.WorksheetFunction.Pi
```

**Listing 5.13:** Nutzung der Tabellenfunktion  $Pi()$  in VBA

Damit weisen wir den Visual Basic-Editor an, innerhalb der Anwendung (»application«; also Excel) aus den eingebauten Tabellenfunktionen (»WorksheetFunction«) die Funktion  $Pi$  zu verwenden. Im Gegensatz zu den Formeln in dem Arbeitsblatt können wir hier auf die Angabe der Klammern verzichten, da sich schon aus dem Begriff »WorksheetFunction« ergibt, dass es sich um einen Funktionsnamen handeln muss.

<sup>6</sup> Sie werden sich erinnern: Der Visual Basic-Editor »versteht« nur Englisch.

Ein Vergleich mit einem alltäglichen Beispiel mag dies verdeutlichen:

Auto.Motor.PS

So könnte es aussehen, wenn Sie auf die gleiche Denk- und Schreibweise die Leistung eines Kraftfahrzeugs mitgeteilt bekommen wollen.

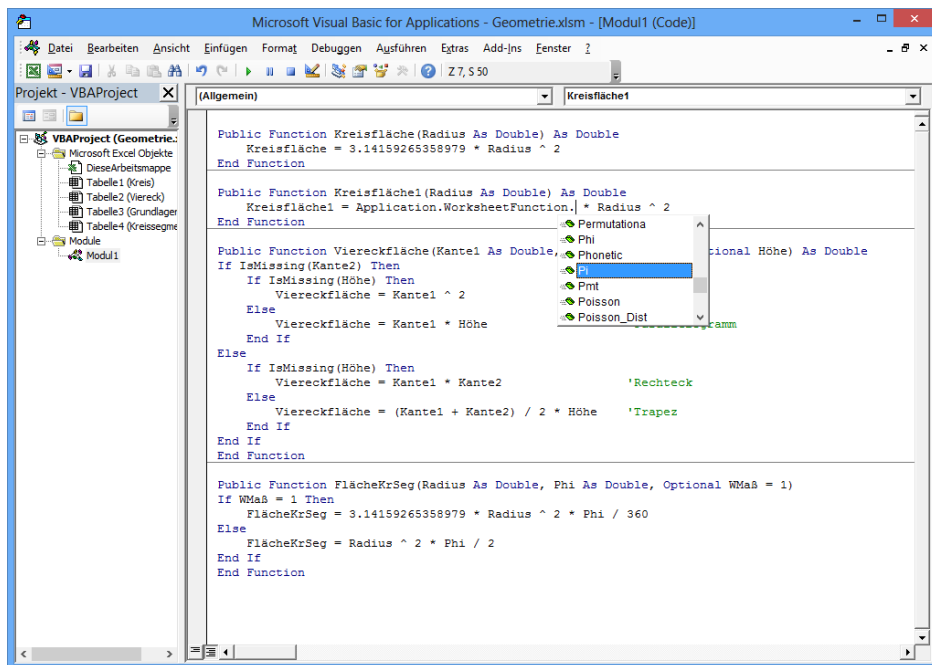
Jetzt können wir unsere verbesserte Kreisflächenfunktion fertig stellen. Statt der konstanten Zahl für die Kreiszahl Pi setzen wir den Ausdruck aus Listing 5.13 ein:

```
Public Function Kreisfläche1(Radius As Double) As Double
 Kreisfläche1 = Application.WorksheetFunction.Pi * Radius ^ 2
End Function
```

**Listing 5.14:** Tabellenfunktion  $Pi()$  in einer benutzerdefinierten Funktion

Es muss eingeräumt werden, dass das Beispiel mit der Tabellenfunktion  $Pi()$  recht einfach ist. Im Regelfall muss man dagegen drei Besonderheiten beachten:

1. Auch wenn eine deutschsprachige Excel-Version vorliegt, werden innerhalb des Visual Basic-Editors die englischen Namen für die Tabellenfunktionen benutzt.
2. Nicht alle Tabellenfunktionen können in VBA eingesetzt werden, weil ihre Aufgaben schon über reine VBA-Funktionen verwirklicht wurden.
3. Einige Tabellenfunktionen haben gleichnamige Entsprechungen in VBA, die allerdings andere Ergebnisse liefern (z.B.  $\log$ ), weil sie dort anders definiert sind.



**Abbildung 5.6:** Listenfeld im Visual Basic-Editor zur leichteren Eingabe von Objekten und Bezeichnern



Zumindest für die beiden ersten Punkte bietet der Visual Basic-Editor mit seinen Listenfeldern Hilfe an: Wenn Sie den Programmcode eingeben, können Sie den geeigneten Begriff aus Dropdownlisten auswählen, auch wenn Ihnen die exakte Schreibweise unbekannt ist (Abbildung 5.6).

Suchen Sie mit dem Windows-Explorer nach der Datei *Vbaliste.xls*. Sie befindet sich gewöhnlich in einem Ordner der Microsoft Office-Installation. Diese Datei enthält zu den deutschen Funktionen die englischen Übersetzungen der Funktionen auf dem Arbeitsblatt *Tabellenfunktionen*. Die ab der Excel-Version 2007 hinzugekommenen Funktionen fehlen allerdings noch.

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap05` in der Arbeitsmappe *Geometrie.xlsm* auf dem Arbeitsblatt *Kreis*. In *Modul1* dieser Arbeitsmappe können Sie mithilfe des Visual Basic-Editors den Code dazu einsehen.

**Tipp**

## Die Funktionen im Funktions-Assistenten

Mit VBA erstellte Funktionen werden im Funktions-Assistenten platziert.

### Eigene Funktionen verwenden

Alle korrekt definierten Funktionen können in einem Arbeitsblatt verwendet werden. Steht in der Zelle *A1* eine positive Zahl für den Radius eines Kreises, können Sie in der Zelle *B1* die Kreisfläche dazu über die Formel `=Kreisfläche(A1)` berechnen. Voraussetzung hierfür ist, dass sich das Modul und das Arbeitsblatt in derselben Arbeitsmappe befinden. Sonst erhalten sie den Fehlerwert `#NAME?`.

Aber auch in einer anderen Arbeitsmappe kann die Funktion verwendet werden. Bei geöffneter Arbeitsmappe erweitern Sie den Namen der Funktion um den Hinweis auf den externen Bezug:

```
=Geometrie.XLS!Kreisfläche(A1)
```

Ist die Arbeitsmappe mit der Funktion nicht geöffnet, muss zusätzlich noch der Pfad angegeben werden, z.B.

```
=C:\Daten\Geometrie.XLS!Kreisfläche(A1)
```

Wenn die Arbeitsmappe mit der benutzerdefinierten Funktion geladen ist, lassen sich die Tabellenfunktionen auch über den Funktions-Assistenten eintragen. Wählen Sie in der Kategorie *Benutzerdefiniert* des Dialogfelds *Funktion einfügen* (Registerkarte *FORMELN*, Gruppe *Funktionsbibliothek*, Schaltfläche *Funktion einfügen*) die gewünschte Funktion aus. Daraufhin bietet Ihnen der Funktions-Assistent das Dialogfeld zur Eingabe des Arguments bzw. der Argumente an.

**Tipp**

Beachten Sie bei der Weitergabe von Dateien, die benutzerdefinierte Funktionen verwenden, dass diese Funktionen in der Mappe gespeichert sein müssen. Alternativ können Sie die Datei mit den Funktionen ebenfalls zur Verfügung stellen. Wenn Sie dies nicht tun, kann die Berechnung nicht durchgeführt werden und die Zellen, die benutzerdefinierte Funktionen verwenden, zeigen den Fehlerwert `#NAME?` an.

**Wichtig**

## Funktionen in einem Add-In speichern

Wenn Sie benutzerdefinierte Funktionen erstellt haben und diese weitergeben wollen, stellt sich die Frage, ob der Benutzer die Funktionen auch einsehen soll oder ob diese besser in einer ausgeblendeten Datei abgelegt werden. Excel bietet hierfür eine spezielle Speicherform an, ein sogenanntes Add-In.

Um eigene Add-Ins zu erstellen, gehen Sie wie folgt vor:

1. Erstellen Sie die Module, die das Add-In enthalten soll.
2. Kompilieren Sie zunächst das Projekt über den Menübefehl *Debuggen/Kompilieren von VBAProject*. Sie stellen damit sicher, dass keine Syntaxfehler enthalten sind.
3. Wechseln Sie in das Excel-Fenster.
4. Wählen Sie *DATEI/Speichern unter* (Excel 2007: Nach einem Klick auf die *Office*-Schaltfläche den Befehl *Speichern unter/Andere Formate*).
5. Wählen Sie das Dateiformat *Excel-Add-In (\*.xlam)* und vergeben Sie einen Namen.
6. Wechseln Sie wieder in den VBA-Editor.
7. Markieren Sie im Projekt-Explorer den Eintrag *Diese Arbeitsmappe*.
8. Wechseln Sie in das Fenster *Eigenschaften* (eventuell noch mit der Funktionstaste F4 oder über *Ansicht/Eigenschaftenfenster* öffnen) und stellen Sie die Eigenschaft *IsAddin* auf *True*. Damit sind die enthaltenen Tabellenblätter im Excel-Fenster ausgeblendet.
9. Speichern Sie die Änderungen über den VBA-Editor.

**Wichtig** Wenn Sie Excel jetzt beenden, können Sie noch eine Sicherungskopie anlegen. Excel bietet nochmals das Standardformat an. Sie sollten dies auf jeden Fall tun.

Das Add-In können Sie nun über den Add-Ins-Manager einbinden. Wenn die Datei geladen ist, wird diese im Projekt-Explorer des VBA-Editors angezeigt. Sie können die Datei hier markieren und bearbeiten.

Die Arbeitsblätter dieser Datei sind zwar ausgeblendet, die enthaltenen Funktionen werden aber im Funktions-Assistenten in der Kategorie *Benutzerdefiniert* angezeigt.

# Teil C

## Referenz – Alle Funktionen im Detail

|            |                                               |     |
|------------|-----------------------------------------------|-----|
| Kapitel 6  | Datums- und Zeitfunktionen                    | 197 |
| Kapitel 7  | Text- und Datenfunktionen                     | 229 |
| Kapitel 8  | Logische Funktionen                           | 261 |
| Kapitel 9  | Nachschlage- und Verweisfunktionen            | 277 |
| Kapitel 10 | Informationsfunktionen                        | 311 |
| Kapitel 11 | Statistische Funktionen                       | 337 |
| Kapitel 12 | Datenbankfunktionen                           | 567 |
| Kapitel 13 | Cubefunktionen                                | 601 |
| Kapitel 14 | Finanzmathematische Funktionen                | 617 |
| Kapitel 15 | Mathematische und trigonometrische Funktionen | 709 |
| Kapitel 16 | Konstruktionsfunktionen                       | 803 |
| Kapitel 17 | Webfunktionen                                 | 861 |



# Kapitel 6

## Datums- und Zeitfunktionen

|                    |     |                         |     |
|--------------------|-----|-------------------------|-----|
| ARBEITSTAG()       | 201 | MONAT()                 | 216 |
| ARBEITSTAG.INTL()  | 202 | MONATSENDE()            | 217 |
| BRTEILJAHRE()      | 204 | NETTOARBEITSTAGE()      | 218 |
| DATEDIF()          | 205 | NETTOARBEITSTAGE.INTL() | 219 |
| DATUM()            | 207 | SEKUNDE()               | 220 |
| DATWERT()          | 208 | STUNDE()                | 221 |
| EDATUM()           | 209 | TAG()                   | 222 |
| HEUTE()            | 210 | TAGE()                  | 222 |
| ISOKALENDERWOCHE() | 211 | TAGE360()               | 223 |
| JAHR()             | 212 | WOCHENTAG()             | 225 |
| JETZT()            | 213 | ZEIT()                  | 227 |
| KALENDERWOCHE()    | 214 | ZEITWERT()              | 228 |
| MINUTE()           | 215 |                         |     |



In diesem Kapitel beschreiben wir Funktionen, die es Ihnen erlauben, mit Datums- und Zeitwerten zu rechnen. Des Weiteren können Sie durch Datums- und Zeitfunktionen mit der sogenannten Systemzeit Ihres Computers arbeiten. Dazu sollte die Systemzeit Ihres PC korrekt eingestellt sein.

## Tipps & Tricks und ein wenig Theorie

### Eingabe von Datumswerten

Wenn Sie bei der Datumseingabe mit dem numerischen Block Ihrer Tastatur arbeiten, ist es meist hinderlich, nach der Tag- und Monatszahl einen Punkt einzugeben.

Diesen Punkt müssen Sie nicht unbedingt eingeben, da Excel auch den Schrägstrich (/) und das Minuszeichen (–) als Trennzeichen bei Datumseingaben erkennt und verarbeitet, zum Beispiel *13/8/12* für den 13.08.2012 oder *26-4-13* für den 26.04.2013. Diese beiden Zeichen finden Sie auch auf dem numerischen Block und Sie können damit komfortabel arbeiten.

### Anzeige von Datumswerten

Wenn Excel eine Eingabe als Datumswert erkennt, wird die entsprechende Zelle automatisch als Datum formatiert. Geben Sie beispielsweise *7-10* ein, wird dies als der 7. Oktober des aktuellen Jahrs gedeutet und die Zelle entsprechend formatiert. Alle weiteren Eingaben in diese Zelle werden somit als Datumswerte dargestellt.

Um eine möglicherweise ungewollte Ausgabe wieder auf den Standard zurückzusetzen, verfahren Sie wie folgt:

1. Markieren Sie die entsprechende Zelle.
2. Wählen Sie auf der Registerkarte *START* in der Gruppe *Bearbeiten* das Symbol *Löschen/ Formate löschen*. Dadurch wird der Zelle wieder das *Standard*-Format zugewiesen. Alternativ können Sie das *Standard*-Zahlenformat auch mit den Tasten **[Strg] + [⇩] + [6]** zuweisen.

### Das Datumssystem von Excel

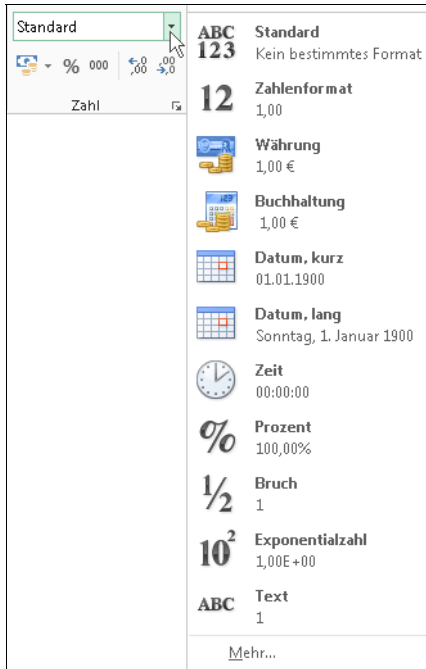
Microsoft Excel speichert Datumsangaben als fortlaufende Zahlen, damit sie für Berechnungen verwendet werden können. Standardmäßig ermittelt Excel ein Datum nach der fortlaufenden Zahl, die mit der Zahl *1* am 1. Januar 1900 beginnt und am 31. Dezember 9999 mit der Zahl *2.958.465* endet. Das bedeutet, dass Excel nur mit Datumswerten, die zwischen diesen beiden Terminen liegen, rechnen kann.

### Wichtig

Microsoft Excel für Apple Macintosh verwendet ein anderes Datumssystem als Standard. Hier beginnt die Zeitrechnung am 01.01.1904. Aus Kompatibilitätsgründen bieten die Windows-Versionen von Excel eine Option, mit dem 1904er Datumssystem zu arbeiten. Sie sollten diese Option nur wählen, wenn Sie Arbeitsmapen zwischen den Excel-Versionen auf Windows- und Apple Macintosh-Rechnern austauschen müssen. Achtung: Diese Einstellung gilt für die aktive Mappe, alle bereits eingetragenen Datumsangaben werden angepasst!

Wenn Sie bei einem Datums- und Zeitwert die fortlaufende Zahl angezeigt haben möchten, müssen Sie nur diese Zelle in das *Standard*-Zahlenformat zurückformatieren. Hierzu reicht ein Klick auf die *Zahlenformatauswahl* auf der Registerkarte *START* (Abbildung 6.1).

Für die fortlaufende Zahl gilt Folgendes: Durch die Zahl, die links vom Komma steht, wird das Datum dargestellt. So bedeutet die Zahl *41275*, dass diese Anzahl an Tagen seit dem 01.01.1900 vergangen ist. Das Datumsergebnis wäre hier der 1. Januar 2013.



**Abbildung 6.1:** Zahlenformatauswahl über die Registerkarte *START* u.a. mit den Datumsformaten

Die Zahlen, die bei der fortlaufenden Zahl rechts vom Komma stehen, geben die Uhrzeit an. Dabei gilt, dass ein Tag den Wert *1* besitzt. Wenn Sie diesen Wert durch die Anzahl der Stunden, die ein Tag besitzt, dividieren, erhalten Sie den Dezimalbruch für eine Stunde; also  $1/24 = 0,04166667$ . Die Angabe *0,5* bedeutet nichts anderes, als dass exakt die Hälfte eines Tags vergangen und es genau 12:00 Uhr ist. Bei der Angabe *0,25* ist es 6:00 Uhr früh, bei *0,75* ist es 18:00 Uhr am späten Nachmittag usw.

Mithilfe der in Tabelle 6.1 gezeigten Formatanweisungen können Sie das Datum und die Uhrzeit nach Ihren Anforderungen anzeigen lassen. In den Beispielmappen zu diesem Kapitel finden Sie zahlreiche Formatierungsvarianten.

**Zahlenformate für Datum und Zeit**

| Formatanweisung | Bedeutung/Ergebnis                                                                   |
|-----------------|--------------------------------------------------------------------------------------|
| T               | Darstellung Tagzahl im Datum ohne die führende Null                                  |
| TT              | Darstellung Tagzahl im Datum mit führender Null bei einstelligen Tagzahlen           |
| TTT             | Darstellung des Tags als abgekürzter Wochentag (Mo, Di, Mi, Do, Fr, Sa, So)          |
| TTTT            | Darstellung des Tags als Wochentagname (Montag, Dienstag usw.)                       |
| M               | Darstellung der Monatszahl im Datum ohne die führende Null                           |
| MM              | Darstellung Monatszahl im Datum mit führender Null bei den einstelligen Monatszahlen |
| MMM             | Darstellung des abgekürzten Monats als Wort (Jan, Feb, Mrz usw.)                     |
| MMMM            | Darstellung des Monatsnamens (Januar, Februar, März usw.)                            |
| J oder JJ       | Darstellung der zweistelligen Jahreszahl aus dem Datum                               |

**Tabelle 6.1**  
Mit Zahlenformaten erhalten Sie die gewünschte Darstellung eines Datums bzw. der Uhrzeit

| Formatanweisung | Bedeutung/Ergebnis                                                                                                                                    |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|
| JJJ oder JJJJ   | Darstellung der vierstelligen Jahreszahl aus dem Datum                                                                                                |
| h               | Darstellung der Stunde ohne führende Null                                                                                                             |
| hh              | Darstellung der Stunde mit führender Null                                                                                                             |
| m               | Darstellung der Minute ohne führende Null                                                                                                             |
| mm              | Darstellung der Minute mit führender Null                                                                                                             |
| [h] oder [hh]   | Darstellung der Stunden auch mit mehr als 24 Stunden                                                                                                  |
| [m] oder [mm]   | Darstellung der Minuten auch mit mehr als 60 Minuten                                                                                                  |
| [s] oder [ss]   | Darstellung der Sekunden auch mit mehr als 60 Sekunden                                                                                                |
| hh:mm AM        | Darstellung der Zeit in der 12-Stundendarstellung, AM: Ante Meridiem (lat. für »Vor dem Mittag«, d.h. die Stunden von Mitternacht bis mittags um 12)  |
| hh:mm PM        | Darstellung der Zeit in der 12-Stundendarstellung, PM: Post Meridiem (lat. für »Nach dem Mittag«, d.h. die Stunden von mittags um 12 bis Mitternacht) |

**Berücksichtigung der Schaltjahre**

In der Datumsberechnung müssen auch die Schaltjahre korrekt berechnet werden. Im Gregorianischen Kalender – welcher hierzulande verwendet wird – ist alle vier Jahre ein Schaltjahr vorgesehen, in dem der Februar 29 Tage hat. Als Sonderfall gilt, dass dann kein Schaltjahr ist, wenn die Jahreszahl ohne Rest durch 100 teilbar ist, mit der Ausnahme: Ist die Jahreszahl ohne Rest durch 400 teilbar, handelt es sich trotzdem um ein Schaltjahr. Ist diese letzte Regel bei der Programmierung nicht berücksichtigt, wird es statt eines 29. Februar 2000 weitere Fehler geben.

Innerhalb der oben angegebenen Datumsgrenzen rechnet Excel alle Datumsdifferenzen korrekt aus, auch über Monats- und Jahresgrenzen, einschließlich der Berücksichtigung von Schaltjahren.

Nach diesem kurzen Abriss, der beim Grundverständnis der Datums- und Zeitfunktionen von Excel helfen soll, kommen wir nun zu den konkreten Funktionen und ihren Anwendungsmöglichkeiten.

**Tabelle 6.2**

Die Datums- und Zeitfunktionen in der Übersicht

| Funktion           | Beschreibung                                                                                                                     |
|--------------------|----------------------------------------------------------------------------------------------------------------------------------|
| ARBEITSTAG()       | Berechnet das Datum vor oder nach einer Anzahl von Arbeitstagen                                                                  |
| ARBEITSTAG.INTL()  | Berechnet das Datum vor oder nach einer Anzahl von Arbeitstagen unter Beachtung der Festlegung für das Wochenende (Version 2010) |
| BRTEILJAHRE()      | Berechnet die Differenz zwischen einem Start- und Enddatum in Jahren                                                             |
| DATEDIF()          | Liefert die Differenz zwischen einem Start- und Enddatum in Jahren, Monaten oder Tagen                                           |
| DATUM()            | Berechnet einen Datumswert aus den angegebenen Jahres-, Monats- und Tagzahlen                                                    |
| DATWERT()          | Wandelt ein als Text vorliegendes Datum in einen Datumswert                                                                      |
| EDATUM()           | Errechnet den Datumswert, der n Monate vor bzw. nach einem Ausgangsdatum liegt                                                   |
| HEUTE()            | Liefert das aktuelle Datum (Systemdatum des PCs)                                                                                 |
| ISOKALENDERWOCHE() | Berechnet die Kalenderwochennummer nach ISO-Norm für einen Datumswert (Version 2013)                                             |





| Funktion                | Beschreibung                                                                                                                      |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| JAHHR()                 | Extrahiert die Jahreszahl aus einem Datumswert                                                                                    |
| JETZT()                 | Liefert das aktuelle Datum und Uhrzeit (Systemdatum des PCs)                                                                      |
| KALENDERWOCHE()         | Berechnet die Kalenderwochennummer für einen Datumswert                                                                           |
| MINUTE()                | Extrahiert die Minutenzahl aus einem Zeitwert                                                                                     |
| MONAT()                 | Extrahiert die Monatszahl aus einem Datumswert                                                                                    |
| MONATSENDE()            | Liefert das letzte Datum des Monats, der n Monate vor oder nach einem Ausgangsdatum liegt                                         |
| NETTOARBEITSTAGE()      | Berechnet die Anzahl der Arbeitstage zwischen zwei Datumsangaben                                                                  |
| NETTOARBEITSTAGE.INTL() | Berechnet die Anzahl der Arbeitstage zwischen zwei Datumsangaben unter Beachtung der Festlegung für das Wochenende (Version 2010) |
| SEKUNDE()               | Extrahiert die Sekundenzahl aus einem Zeitwert                                                                                    |
| STUNDE()                | Extrahiert die Stundenzahl aus einem Zeitwert                                                                                     |
| TAG()                   | Extrahiert die Tagzahl aus einem Datumswert                                                                                       |
| TAGE()                  | Berechnet die Anzahl der Tage zwischen zwei Datumsangaben (Version 2013)                                                          |
| TAGE360()               | Berechnet die Anzahl der Tage zwischen zwei Datumsangaben, basierend auf einem Jahr mit 360 Tagen                                 |
| WOCHENTAG()             | Wandelt einen Datumswert in die Wochentagsnummer von 1 bis 7 um                                                                   |
| ZEIT()                  | Berechnet den Uhrzeitwert aus den angegebenen Stunden-, Minuten- und Sekundenzahlen                                               |
| ZEITWERT()              | Wandelt eine als Text vorliegende Uhrzeit in einen Zeitwert                                                                       |

## ARBEITSTAG() WORKDAY()

ARBEITSTAG(*Ausgangsdatum*; *Tage*; *Freie\_Tage*)

Die Funktion gibt das Datum vor oder nach einer bestimmten Anzahl von Arbeitstagen (*Tage*), gemessen von *Ausgangsdatum*, zurück.

*Ausgangsdatum* (erforderlich) ist ein Datum, das den Start der Berechnung festlegt.

*Tage* (erforderlich) ist die vor oder hinter *Ausgangsdatum* liegende Anzahl von Tagen, die nicht auf ein Wochenende fallen oder zu den freien Tagen gehören – also Arbeitstage. Ein positiver Wert für *Tage* errechnet ein zukünftiges, ein negativer Wert ein zurückliegendes Datum.

*Freie\_Tage* (optional) ist eine Liste mit einer oder mehreren Datumsangaben, die alle Arten von arbeitsfreien Tagen repräsentieren kann, die aus dem Arbeitskalender ausgeschlossen werden sollen, beispielsweise staatliche oder regionale Feiertage.

Wenn Sie Zahlungsziele, zu erwartende Lieferzeiten oder die Anzahl bereits verstrichener Arbeitstage berechnen wollen, können Sie es einfach mit der Formel *Datum plus Tage* bzw. *Enddatum minus Startdatum* berechnen. Soll das Zahlungsziel jedoch nicht auf das Wochenende oder einen Feiertag fallen, ist die Funktion ARBEITSTAG() für Sie von Interesse.

**Syntax**

**Definition**

**Argumente**

**Hintergrund**

Die Funktion zählt bei der Berechnung Wochenenden sowie die Tage, die in *Freie\_Tage* angegeben sind, nicht zu den Arbeitstagen. Bei der Liste *Freie\_Tage* kann es sich um einen Zellbereich, der die Datumsangaben enthält, oder um eine Matrixkonstante der fortlaufenden Zahlen handeln, welche die Datumsangaben darstellen.

Beachten Sie, dass die Funktion den Tag, den *Ausgangsdatum* darstellt, bei der Zählung nicht berücksichtigt. Wenn Sie also das Ergebnis mithilfe der Funktion NETTOARBEITSTAGE() gegenrechnen, wird Ihnen NETTOARBEITSTAGE() immer einen Tag mehr an Differenz ausrechnen.

Beachten Sie auch diese Hinweise bei der Arbeit mit der Funktion:

- ▶ Ist eines der Argumente kein zulässiges Datum, gibt ARBEITSTAG() den Fehlerwert #WERT! zurück
- ▶ Ergibt die Addition *Ausgangsdatum* + *Tage* ein ungültiges Datum, gibt ARBEITSTAG() den Fehlerwert #ZAHL! zurück
- ▶ Ist *Tage* keine ganze Zahl, werden deren Nachkommastellen abgeschnitten

**Praxiseinsatz** In einem Rechnungsformular soll das Zahlungsziel mit 14 Tagen ab dem aktuellen Datum berechnet werden, darf aber nicht auf ein Wochenende fallen. Die Formel

```
=ARBEITSTAG(HEUTE();14)
```

ergab am 27.01.2013 das Datum 14.02.2013, der 14. Arbeitstag nach dem 27.01.2013.

Bei einem Zahlungsziel von 10 Tagen ergibt die Berechnung

```
=ARBEITSTAG("12.12.2012";10)
```

für das Ausgangsdatum 12.12.2012 das Zieldatum 26.12.2012. Unter Berücksichtigung der Weihnachtsfeiertage ergibt die Berechnung

```
=ARBEITSTAG("12.12.2012";10;{"25.12.2012";"26.12.2012"})
```

das Zieldatum 28.12.2012.

Beachten Sie bitte in der obigen Formel, dass die Liste der freien Tage von geschweiften Klammern {} und nicht von runden Klammern umgeben sein muss!

**Siehe auch** ARBEITSTAG.INTL(), EDATUM(), HEUTE(), JETZT(), MONATSENDE(), NETTOARBEITSTAGE(), NETTOARBEITSTAGE.INTL(), TAGE(), TAGE360()



Die gezeigten Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* (Excel 2007/2013) auf dem Arbeitsblatt *ARBEITSTAG*.

Neu in Excel  
2010

## ARBEITSTAG.INTL()



## WORKDAY.INTL()

**Syntax** ARBEITSTAG.INTL(*Ausgangsdatum*; *Tage*; *Wochenende*; *Freie\_Tage*)

**Definition** Die Funktion gibt das Datum vor oder nach einer bestimmten Anzahl von Arbeitstagen (*Tage*), gemessen von *Ausgangsdatum*, zurück.

**Ausgangsdatum** (erforderlich) ist ein Datum, das den Start der Berechnung festlegt.

**Tage** (erforderlich) ist die vor oder hinter *Ausgangsdatum* liegende Anzahl von Tagen, die nicht auf ein Wochenende fallen oder zu den freien Tagen gehören – also Arbeitstage. Ein positiver Wert für *Tage* errechnet ein zukünftiges, ein negativer Wert ein zurückliegendes Datum.

**Wochenende** (optional) ist eine Zahl oder Zeichenfolge, die angibt, welche Tage als Wochenende und nicht als Arbeitstage bewertet werden.

**Freie\_Tage** (optional) ist eine Liste mit einer oder mehreren Datumsangaben, die alle Arten von arbeitsfreien Tagen repräsentieren kann, die aus dem Arbeitskalender ausgeschlossen werden sollen, beispielsweise staatliche oder regionale Feiertage.

Siehe *Hintergrund* zur Funktion ARBEITSTAG(). Zusätzlich haben Sie die Möglichkeit, festzulegen, welche Wochentage als Wochenende bewertet und damit nicht mitgezählt werden. Wenn Sie die Wochenenttage mit einer Zahl oder einer Zeichenfolge angeben, tun Sie dies mit diesen Werten:

| Zahl                  | Wochentage           | Entspricht der Zeichenfolge |
|-----------------------|----------------------|-----------------------------|
| 1 (oder keine Angabe) | Samstag, Sonntag     | 0000011                     |
| 2                     | Sonntag, Montag      | 1000001                     |
| 3                     | Montag, Dienstag     | 1100000                     |
| 4                     | Dienstag, Mittwoch   | 0110000                     |
| 5                     | Mittwoch, Donnerstag | 0011000                     |
| 6                     | Donnerstag, Freitag  | 0001100                     |
| 7                     | Freitag, Samstag     | 0000110                     |
| 11                    | Nur Sonntag          | 0000001                     |
| 12                    | Nur Montag           | 1000000                     |
| 13                    | Nur Dienstag         | 0100000                     |
| 14                    | Nur Mittwoch         | 0010000                     |
| 15                    | Nur Donnerstag       | 0001000                     |
| 16                    | Nur Freitag          | 0000100                     |
| 17                    | Nur Samstag          | 0000010                     |

Außer der Zeichenfolge 1111111 sind alle weiteren Kombinationen möglich. Die 0 steht für einen Arbeitstag, die 1 für einen Feiertag.

In einem Rechnungsformular soll das Zahlungsziel mit 14 Tagen ab dem aktuellen Datum berechnet werden, darf aber nicht auf ein Wochenende fallen. Die Formel

```
=ARBEITSTAG.INTL(HEUTE();14)
```

ergab am 27.01.2013 das Datum 14.02.2013, der 14. Arbeitstag nach dem 27.01.2013.

## Argumente

## Hintergrund

## Praxiseinsatz

Bei einem Zahlungsziel von 10 Tagen ergibt die Berechnung

```
=ARBEITSTAG.INTL("12.12.2012";10)
```

für das Ausgangsdatum *12.12.2012* das Zieldatum *26.12.2012*. Unter Berücksichtigung der Weihnachtsfeiertage ergibt die Berechnung

```
=ARBEITSTAG.INTL("12.12.2012";10;1;{"25.12.2012";"26.12.2012"})
```

das Zieldatum *28.12.2012*.

Beachten Sie bitte in der obigen Formel, dass die Liste der freien Tage von geschweiften Klammern {} und nicht von runden Klammern umgeben sein muss!

**Siehe auch** ARBEITSTAG(), EDATUM(), HEUTE(), JETZT(), MONATSENDE(), NETTOARBEITSTAGE(), NETTOARBEITSTAGE.INTL(), TAGE(), TAGE360()



Die gezeigten Beispiele finden Sie im Ordner `\Ms5-235\Kap06` in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *ARBEITSTAG.INTL*.

## BRTEILJAHRE() YEARFRAC()

**Syntax** BRTEILJAHRE(*Ausgangsdatum*; *Enddatum*; *Basis*)

**Definition** Die Funktion wandelt die ganzzahlige Differenz (in Tagen) zwischen *Ausgangsdatum* und *Enddatum* in Bruchteile von Jahren um, d.h., Sie erhalten das Ergebnis als Dezimalzahl (in Jahren).

**Argumente** *Ausgangsdatum* (erforderlich) ist ein Datum, das dem Anfangsdatum der Berechnung entspricht.

*Enddatum* (erforderlich) ist ein Datum, das das Enddatum der Berechnung angibt.

*Basis* (optional) gibt an, auf welcher Basis die Zinstage gezählt werden. Ihnen stehen folgende Optionen zur Verfügung:

- ▶ 0 oder nicht angegeben USA (NASD) 30/360
- ▶ 1 Taggenau/Taggenau
- ▶ 2 Taggenau/360
- ▶ 3 Taggenau/365
- ▶ 4 Europa 30/360

**Hintergrund** Durch die Verwendung der Funktion BRTEILJAHRE() können z.B. Laufzeiten von Forderungen oder Verbindlichkeiten besser miteinander verglichen werden. Diese Funktion hat also einen engen Bezug zu den finanzmathematischen Funktionen (siehe Kapitel 14).

Beachten Sie auch diese Hinweise beim Einsatz der Funktion:

- ▶ Alle Argumente werden durch Abschneiden der Nachkommastellen zu ganzen Zahlen gekürzt
- ▶ Ist *Ausgangsdatum* oder *Enddatum* kein zulässiges Datum, gibt BRTEILJAHRE() den Fehlerwert #WERT! zurück
- ▶ Ist *Basis* kleiner 0 bzw. größer 4, gibt BRTEILJAHRE() den Fehlerwert #ZAHL! zurück

Für die Laufzeitangabe einer Finanzierung vom 1.1.2012 bis zum 10.10.2013 ermittelt die Formel

=BRTEILJAHRE("01.01.2012";"10.10.2013")

eine Laufzeit von 1,775 Jahren, basierend auf der US-Norm mit einem 30-Tage-Monat und einem 360-Tage-Jahr. Hier weitere Beispiele (siehe auch Abbildung 6.2):

- ▶ =BRTEILJAHRE("01.01.2012";"10.10.2013";1) ergibt den Wert 1,77291
- ▶ =BRTEILJAHRE("01.01.2012";"10.10.2013";2) ergibt den Wert 1,80000
- ▶ =BRTEILJAHRE("01.01.2012";"10.10.2013";3) ergibt den Wert 1,77534
- ▶ =BRTEILJAHRE("01.01.2012";"10.10.2013";4) ergibt den Wert 1,77500

| E4 |                      |                 |              |                   |
|----|----------------------|-----------------|--------------|-------------------|
| A  | B                    | C               | D            | E                 |
| 1  | <b>BRTEILJAHRE</b>   |                 |              |                   |
| 2  |                      |                 |              |                   |
| 3  | <i>Ausgangsdatum</i> | <i>Enddatum</i> | <i>Basis</i> | <i>Berechnung</i> |
| 4  | 01.01.2012           | 10.10.2013      |              | 1,77500           |
| 5  | 01.01.2012           | 10.10.2013      | 1            | 1,77291           |
| 6  | 01.01.2012           | 10.10.2013      | 2            | 1,80000           |
| 7  | 01.01.2012           | 10.10.2013      | 3            | 1,77534           |
| 8  | 01.01.2012           | 10.10.2013      | 4            | 1,77500           |

**Abbildung 6.2:** Die Laufzeit einer Finanzierung mit BRTEILJAHRE() berechnet

ARBEITSTAG(), ARBEITSTAG.INTL(), EDATUM(), HEUTE(), JETZT(), MONATS-ENDE(), NETTOARBEITSTAGE(), NETTOARBEITSTAGE.INTL(), TAGE(), TAGE360()

Die gezeigten Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *BRTEILJAHRE*.



**Siehe auch**

## DATEDIF() DATEDIF()

DATEDIF(*Ausgangsdatum*; *Enddatum*; *Einheit*)

Diese Funktion ermittelt die Differenz zwischen *Anfangs-* und *Enddatum* (Zeitraum) in Jahren, Monaten oder Tagen.

*Ausgangsdatum* (erforderlich) ist das Startdatum des zu berechnenden Zeitraums.

*Enddatum* (erforderlich) ist ein Datum, das das letzte Datum oder Enddatum des Zeitraums angibt.

*Einheit* (erforderlich) ist der Informationstyp, der zurückgegeben werden soll. Die folgenden Einheiten stehen Ihnen dabei zur Verfügung:

- ▶ »Y« liefert die Anzahl der vollständigen Jahre im Zeitraum
- ▶ »M« liefert die Anzahl der vollständigen Monate im Zeitraum
- ▶ »D« liefert die Anzahl der Tage im Zeitraum

**Syntax**

**Definition**

**Argumente**

- ▶ »MD« liefert die Differenz zwischen den Tagen im *Anfangsdatum* und *Enddatum*. Die Monate und Jahre der Datumsangaben werden ignoriert.
- ▶ »YM« liefert die Differenz zwischen den Monaten in *Anfangsdatum* und *Enddatum*. Die Tage und Jahre der Datumsangaben werden ignoriert.
- ▶ »YD« liefert die Differenz zwischen den Tagen im *Anfangsdatum* und *Enddatum*. Die Jahre der Datumsangaben werden ignoriert.

**Hintergrund** Diese Funktion zum Errechnen von Datumsdifferenzen in verschiedenen Einheiten und mit verschiedenen Rechenmethoden hält Microsoft seit Jahren in Excel versteckt. Sie ist seit Längem in Excel vorhanden, jedoch nicht dokumentiert, d.h., im Funktions-Assistenten und in der Excel-Hilfe suchen Sie die Funktion vergebens.

Alle Datumsangaben können als Textzeichenfolgen (in Anführungszeichen), als fortlaufende Zahlen oder als Ergebnis anderer Formeln oder Funktionen eingegeben werden. Die Beispiele zeigen die Wirkungsweise der Funktion.

**Praxiseinsatz** Für die Berechnung der Beschäftigungszeiten von Mitarbeitern zu einem Stichtatum verwenden Sie die Funktion DATEDIF(). So ergibt die Berechnung

=DATEDIF("01.01.2008";"13.11.2013";"Y")

für einen Mitarbeiter, welcher am 1.1.2008 eingestellt wurde, dass er zum Stichtag 13.11.2013 5 Jahre beschäftigt ist. Für die Rückstellung von Weihnachtsgeld, das abhängig von den Beschäftigungsmonaten berechnet wird, ermitteln Sie mit der Formel

=DATEDIF("01.01.2008";"30.11.2013";"M")

Zum Stichtag 30.11.2013 den Wert 70 Monate für einen Mitarbeiter, der seit dem 1.1.2008 in der Firma ist (vgl. Zeile 5 in Abbildung 6.3).

| E6 |                      |                 |              |                   |
|----|----------------------|-----------------|--------------|-------------------|
| A  | B                    | C               | D            | E                 |
| 1  | <b>DATEDIF</b>       |                 |              |                   |
| 2  |                      |                 |              |                   |
| 3  | <i>Ausgangsdatum</i> | <i>Enddatum</i> | <i>Basis</i> | <i>Berechnung</i> |
| 4  | 01.01.2008           | 11.11.2013      | D            | 2141              |
| 5  | 01.01.2008           | 30.11.2013      | M            | 70                |
| 6  | 01.01.2008           | 13.11.2013      | Y            | 5                 |
| 7  | 01.01.2008           | 14.11.2013      | MD           | 13                |
| 8  | 01.01.2008           | 15.11.2013      | YM           | 10                |
| 9  | 01.01.2008           | 16.11.2013      | YD           | 320               |

**Abbildung 6.3:** Beschäftigungszeiten ermitteln

**Siehe auch** ARBEITSTAG(), ARBEITSTAG.INTL(), EDATUM(), MONATSENDE(), NETTOARBEITSTAGE(), NETTOARBEITSTAGE.INTL(), TAGE(), TAGE360()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *DATEDIF*.

# DATUM()



DATUM(*Jahr;Monat;Tag*)

Die Funktion gibt die fortlaufende Zahl zurück, die ein durch die Argumente *Jahr*, *Monat* und *Tag* bestimmtes Datum darstellt.

*Jahr* (erforderlich) ist eine Zahl, welche ein bis vier Stellen umfassen kann.

*Monat* (erforderlich) ist eine Zahl, die den Monat des Jahrs darstellt.

*Tag* (erforderlich) ist eine Zahl, die den Tag des Monats darstellt.

Bei Datumsberechnungen gibt es häufig den Anlass, ein Datum zu zerlegen, um z.B. ganze Monate zu addieren. Das Berechnungsergebnis muss aber wieder in ein reguläres Datum einfließen. Hier setzt die Funktion DATUM() an: Aus den angegebenen Einzelbestandteilen eines Datums wird wieder ein (numerischer) Datumsausdruck zusammengesetzt, welchen Sie in weiteren Datumsberechnungen einsetzen können. Die DATUM()-Funktion eignet sich am besten für Formeln, in denen *Jahr*, *Monat* und *Tag* Formeln und keine Konstanten sind.

Excel interpretiert das Argument *Jahr* entsprechend des Datumssystems, das Sie verwenden. Standardmäßig verwendet Excel für Windows das 1900-Datumssystem; Excel für den Macintosh verwendet das 1904-Datumssystem.

Für das 1900-Datumssystem gilt Folgendes:

- ▶ Wenn *Jahr* zwischen 0 (Null) und 1899 (einschließlich) liegt, addiert Excel den Wert zu 1900, um das Jahr zu berechnen
- ▶ Wenn *Jahr* zwischen 1900 und 9999 (einschließlich) liegt, verwendet Excel diesen Wert als Jahresangabe
- ▶ Wenn *Jahr* kleiner 0 bzw. größer oder gleich 10000 ist, gibt Excel den Fehlerwert #ZAHL! zurück

Für das 1904-Datumssystem gilt Folgendes:

- ▶ Wenn *Jahr* zwischen 4 und 1899 (einschließlich) liegt, addiert Excel den Wert zu 1900, um das Jahr zu berechnen
- ▶ Wenn *Jahr* zwischen 1904 und 9999 (einschließlich) liegt, verwendet Excel diesen Wert als Jahresangabe
- ▶ Wenn *Jahr* kleiner 4 bzw. größer oder gleich 10000 ist oder wenn *Jahr* zwischen 1900 und 1903 (einschließlich) liegt, gibt Excel den Fehlerwert #ZAHL! zurück

Für beide Zeitsysteme gilt:

- ▶ Wenn *Monat* größer 12 ist, wird die Anzahl der Monate zum ersten Monat des angegebenen Jahrs addiert
- ▶ Wenn *Tag* größer als die Anzahl der Tage des angegebenen Monats ist, wird diese Anzahl zum ersten Tag des Monats addiert

**Syntax**

**Definition**

**Argumente**

**Hintergrund**

**Praxiseinsatz** In einer Buchhaltungsanwendung muss zu einem gegebenen Datum der Monatsletzte und -erste berechnet werden. Die Formel

```
=DATUM(JAHR("23.07.2012");MONAT("23.07.2012");1)
```

ergibt den Monatsersten *01.07.2012*. Sie setzt die Jahreszahl und die Monatszahl aus einem gegebenen Datum zusammen mit der Tagzahl 1 zum Monatsersten für dieses Datum zusammen. Die Formel

```
=DATUM(JAHR("23.07.2012");MONAT("23.07.2012")+1;1)-1
```

ergibt den Monatsletzten *31.07.2012*. Sie setzt die Jahreszahl und die um eins erhöhte Monatszahl (also Folgemonat) aus einem gegebenen Datum zusammen mit der Tagzahl 1 zum Monatsersten des Folgemonats zusammen. Zieht man davon einen Tag ab, erhält man den Monatsletzten.

Weitere Beispiele:

- ▶ =DATUM(112;1;2) ergibt den *02.01.2012* (1900+112)
- ▶ =DATUM(2012;1;2) ergibt den *02.01.2012*
- ▶ =DATUM(2012;14;2) ergibt den *02.02.2013*
- ▶ =DATUM(2012;1;35) ergibt den *04.02.2012*

**Siehe auch** ZEIT(), DATWERT(), HEUTE(), JAHR(), MONAT(), TAG(), ZEITWERT()



Diese Beispiele finden Sie im Ordner `Ms5-235\Kap06` in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *DATUM*.

## DATWERT()



## DATEVALUE()

**Syntax** DATWERT(*Datumstext*)

**Definition** Die Funktion wandelt ein als Text vorliegendes Datum in eine fortlaufende Zahl entsprechend des Excel-Datumssystems um.

**Argumente** *Datumstext* (erforderlich) ist Text, der ein Tagesdatum in einem der Microsoft Excel-Datumsformate darstellt.

**Hintergrund** In allen Fällen, wo Ihnen eine Datumsangabe im Textformat vorliegt (bspw. aus Importen), verwenden Sie DATWERT(), um das vorliegende Datum in eine fortlaufende Zahl, sprich Datum, umzuwandeln. Mit dem Ergebnis können Sie dann weitere Berechnungen anstellen.

Die meisten Funktionen wandeln zwar Datumstextwerte automatisch in fortlaufende Zahlen um, aber in bestimmten Fällen kann diese Regel einmal nicht gelten, z.B. bei importierten Daten oder bei Arbeitsblatffunktionen aus Drittanbieter-Add Ins. Wenn Sie solche Fälle ausschließen und sicher gehen wollen, wandeln Sie mit DATWERT() um.

Angaben wie *"12.12.2012"* oder *"11/9/2013"* sind Textzeichenfolgen in Anführungszeichen, die gültige Datumsangaben darstellen. Wird mit dem standardmäßigen Datumssystem von Excel für Windows gearbeitet, muss *Datumstext* ein Datum darstellen, das zwischen dem 1. Januar 1900 und dem 31. Dezember 9999 liegt, beim standardmäßigen Datumssystem von



Excel für den Macintosh muss *Datumstext* ein Datum darstellen, das zwischen dem 1. Januar 1904 und dem 31. Dezember 9999 liegt. Entspricht Datumstext einem Datum, das außerhalb dieses Bereichs liegt, gibt DATWERT() den Fehlerwert #WERT! zurück.

Ist in *Datumstext* kein Jahr angegeben, verwendet DATWERT() die Jahresangabe, die von der Systemuhr Ihres PCs zur Verfügung gestellt wird. In *Datumstext* angegebene Uhrzeiten werden ignoriert.

Nach einem Import erscheinen die Werte einer Datumsspalte als Text. Diese sollen für weitere Berechnungen in echte Datumswerte, also numerische Werte umgewandelt werden. Die Formel

```
=DATWERT("12.12.2012")
```

ergibt den Datumswert *12.12.2012*, welcher eine fortlaufende Datumszahl im Excel-Datumsystem darstellt. Weitere Beispiele:

- ▶ =DATWERT("11.11") ergibt im Jahr 2013 den Datumswert *11.11.2013*
- ▶ =DATWERT("11/9/2012") ergibt den Datumswert *11.09.2012*<sup>1</sup>
- ▶ =DATWERT("9-11-2013") ergibt den Datumswert *09.11.2013*<sup>2</sup>
- ▶ =DATWERT("August 2013") ergibt den Datumswert *01.08.2013*
- ▶ =DATWERT("12.2012") ergibt den Datumswert *01.12.2012*
- ▶ =DATWERT("7/1999") ergibt den Datumswert *01.07.1999*
- ▶ =DATWERT("2012-11-22") ergibt den Datumswert *22.11.2012*

DATUM(), HEUTE(), JETZT(), ZEITWERT()

Die Beispiele finden Sie im Ordner `\Ms5-235\Kap06` in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *DATWERT*.

**Praxiseinsatz**

**Siehe auch**



## EDATUM() EDATE()

EDATUM(*Ausgangsdatum*; *Monate*)

Die Funktion gibt die fortlaufende Zahl des Datums zurück, das eine bestimmte Anzahl von Monaten vor bzw. nach dem in *Ausgangsdatum* angegebenen Datum liegt.

*Ausgangsdatum* (erforderlich) ist ein Datum, das dem Anfangsdatum entspricht.

*Monate* (erforderlich) gibt an, wie viele Monate vor oder nach dem *Ausgangsdatum* liegen sollen. Ein positiver Wert für *Monate* ergibt ein in der Zukunft, ein negativer Wert ein in der Vergangenheit liegendes Datum.

Mit der Funktion EDATE() können Sie Rückzahlungs- oder Fälligkeitstermine berechnen, die auf denselben Tag eines Monats fallen wie der jeweilige Emissionstermin. Eine Alternative zu dieser Funktion bestände darin, mit den Funktionen JAHR(), MONAT() und TAG()

**Syntax**

**Definition**

**Argumente**

**Hintergrund**

<sup>1</sup> Je nach Ländereinstellung kann das Datum "11/9/2012" auch als 09.11.2012 interpretiert werden.

<sup>2</sup> Je nach Ländereinstellung kann das Datum "9-11-2013" auch als 11.09.2013 interpretiert werden.

das *Ausgangsdatum* in seine Bestandteile zu zerlegen, die ermittelte Monatszahl um den Wert *Monate* zu erhöhen und alle drei Bestandteile anschließend wieder mithilfe der Funktion DATUM() zu einem regulären Datum (fortlaufende Zahl) zusammensetzen:

=DATUM(JAHR(Ausgangsdatum);MONAT(Ausgangsdatum)+Monate;TAG(Ausgangsdatum))

Es könnten Probleme auftreten, wenn Sie Datumsangaben als Text eingeben. Ist *Ausgangsdatum* kein zulässiges Datum, gibt EDATUM() den Fehlerwert #WERT! zurück. Ist *Monate* keine ganze Zahl, werden die Nachkommastellen abgeschnitten.

**Praxiseinsatz** Für einen 18-monatigen Baustelleneinsatz soll das Ende berechnet werden. Die Formel =EDATUM("02.01.2012";18) ergibt das Datum 02.07.2013 (Abbildung 6.4). Weitere Beispiele:

- ▶ =EDATUM("02.01.2012";24) ergibt das Datum 02.01.2014
- ▶ =EDATUM("03.01.2012";0) ergibt das Datum 03.01.2012
- ▶ =EDATUM("04.01.2012";-5) ergibt das Datum 04.08.2011

|   | A             | B                    | C             | D                 |
|---|---------------|----------------------|---------------|-------------------|
| 1 | <b>EDATUM</b> |                      |               |                   |
| 2 |               |                      |               |                   |
| 3 |               | <i>Ausgangsdatum</i> | <i>Monate</i> | <i>Berechnung</i> |
| 4 |               | 02.01.2012           | 18            | 02.07.2013        |
| 5 |               | 02.01.2012           | 24            | 02.01.2014        |
| 6 |               | 03.01.2012           |               | 03.01.2012        |
| 7 |               | 04.01.2012           | -5            | 04.08.2011        |
| 8 |               | 05.01.2012           | -9            | 05.04.2011        |
| 9 |               | 31.12.2012           | 2             | 28.02.2013        |

**Abbildung 6.4:** Endedatum für Projekteinsätze berechnen

**Siehe auch** ARBEITSTAG(), ARBEITSTAG.INTL(), DATEDIF(), DATUM(), MONATSENDE(), NETTOARBEITSTAGE(), NETTOARBEITSTAGE.INTL(), TAGE(), TAGE360()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *EDATUM*.

## HEUTE() TODAY()

**Syntax** HEUTE()

**Definition** Die Funktion HEUTE() gibt die fortlaufende Zahl des aktuellen Datums zurück.

**Argumente** Keine

**Hintergrund** Bei der Eingabe der HEUTE()-Funktion erhalten Sie als Ergebnis das aktuelle Datum ohne die Uhrzeit. Wenn für das Zellenformat vor der Eingabe der Funktion die Option *Standard* eingestellt war, wird das Ergebnis als Datum formatiert.

Die verwandte Funktion JETZT() liefert ebenfalls das aktuelle Datum, aber einschließlich der aktuellen Uhrzeit. Der Unterschied zwischen HEUTE() und JETZT() besteht also darin, dass HEUTE() nur das aktuelle Datum und die Funktion JETZT() zum Datum auch die aktuelle Systemzeit abrufen.

Bevorzugen Sie bei der Bildung von Datumsdifferenzen, bei denen die Differenz in Tagen wesentlich ist, die Funktion HEUTE(). Für Berechnungen wie die Tage, die seit Rechnungslegung vergangen sind, setzen Sie die Funktion HEUTE() ein. Anhand der festgestellten Differenz können Sie dann entscheiden, ob Fristen eingehalten oder überschritten wurden.

Die Ergebnisse beider Funktionen werden nicht ständig aktualisiert. Es finden nur dann Aktualisierungen statt, wenn die Tabelle geöffnet oder die zugehörige Tabelle berechnet wird. Die Neuberechnung einer Tabelle können Sie durch Drücken der Taste F9 erzwingen.

Natürlich ist die Richtigkeit bzw. Genauigkeit beider Funktionen davon abhängig, dass Ihre PC-interne Systemuhr die korrekte Systemzeit führt.

In einem Rechnungsformular soll automatisch immer das aktuelle Tagesdatum stehen. Die Formel =HEUTE() setzt das aktuelle Datum in das Formular. Weitere Einsatzmöglichkeiten:

- ▶ =HEUTE()+14 ergibt das Datum heute in 14 Tagen
- ▶ =HEUTE()+200 ergibt das Datum heute in 200 Tagen
- ▶ =HEUTE()-100 ergibt das Datum heute vor 100 Tagen
- ▶ =HEUTE()-"14.11.1959" ergibt für den Geburtstag 14.11.1959 das aktuelle Alter in Tagen

DATUM(), DATWERT(), JETZT(), TAG()

Diese Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt HEUTE.

#### Praxiseinsatz

#### Siehe auch



## ISOKALENDERWOCHE()



## ISOWEEKNUM()

ISOKALENDERWOCHE(*Datum*)

Die Funktion gibt eine ganze Zahl zurück, welche die ISO-Wochennummer des angegebenen Datums im Jahr darstellt.

*Datum* (erforderlich) ist das Datum, deren KW ermittelt werden soll.

Das Jahr wird in Kalenderwochen (KW) eingeteilt, die durchnummeriert sind. Ein Jahr hat 52 oder 53 Kalenderwochen. Seit der Festlegung der DIN-Norm 1355 im Jahre 1974 ist die erste Kalenderwoche die Woche eines Jahrs, in der mindestens vier der ersten sieben Januartage liegen (also die Woche, in der der 4. Januar liegt). Inhaltlich gleich regelt dies die Internationale Norm ISO 8601 von 1988, die von der Europäischen Union als EN 28601 im Jahre 1992 übernommen und in Deutschland als DIN EN 28601 (1993) umgesetzt wurde.

Demzufolge haben die meisten Jahre 52 Wochen. Der jährliche überzählige Tag sowie die Schaltjahre führen aber dazu, dass es alle fünf bis sechs Jahre ein Jahr mit 53 Wochen gibt, so die Jahre 1992, 1998, 2004, 2009 oder 2015. Da es keine »halben« Wochen gibt, gehören manche Tage ihrer KW noch zu einem anderen Jahr, als es der üblichen Datumsangabe entspricht (z.B. gehörte der 2.01.2000 noch zur 53. KW 1999).

Neu in Excel  
2013

Syntax

Definition

Argumente

Hintergrund

**Praxiseinsatz** In einem Excel-basierten Einsatzplan soll zu jedem Datum die Kalenderwoche gezeigt werden. Verwenden Sie hierzu die Formel =ISOKALENDERWOCHE(Datumswert).

- ▶ =ISOKALENDERWOCHE("01.01.2005") ergibt die KW 53
- ▶ =ISOKALENDERWOCHE("03.01.2005") ergibt die KW 1
- ▶ =ISOKALENDERWOCHE("20.10.2005") ergibt die KW 42
- ▶ =ISOKALENDERWOCHE("31.12.2012") ergibt die KW 1
- ▶ =ISOKALENDERWOCHE("02.01.2013") ergibt die KW 1

**Siehe auch** TAGE(), TAGE360(), WOCHENTAG(), JAHR(), MONAT(), TAG()



Die gezeigten Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *ISOKALENDERWOCHE* (nur für Excel 2013).

## JAHR() YEAR()

**Syntax** JAHR(*Zahl*)

**Definition** Die Funktion extrahiert aus einem Datum die darin enthaltene vierstellige Jahreszahl. Das Ergebnis von JAHR() ist eine ganze Zahl, die einen Wert von 1900 bis 9999 annehmen kann.

**Argumente** *Zahl* (erforderlich) ist das Datum des Jahrs, das Sie ermitteln möchten.

**Hintergrund** Mithilfe dieser Funktion sowie der Funktionen MONAT() und TAG() können Sie ein (gültiges) Datum in seine Bestandteile zerlegen, um damit z.B. weitere Berechnungen anzustellen. Diese Funktion können Sie immer dann einsetzen, wenn in Tabellen Werte nach Jahren zusammengefasst werden sollen und es nicht mehr auf das Tagesdatum ankommt.

Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden. Die von den Funktionen JAHR(), MONAT() und TAG() zurückgegebenen Werte sind gregorianische Werte (entsprechend des Gregorianischen Kalenders).

**Praxiseinsatz** Um die Filterung von Jahren mit dem Excel-*AutoFilter* bzw. die Sortierung nach Jahren zur Verfügung zu stellen, sollen in einer Hilfsspalte die Jahreswerte aus einer Datumsspalte ermittelt werden. Verwenden Sie dazu die Berechnung =JAHR(Datumswert). Hier einige Anwendungen und Hinweise:

- ▶ =JAHR(HEUTE()) ergibt im Jahr 2013 den Wert 2013
- ▶ =JAHR("14.11.1959") ergibt den Wert 1959
- ▶ =JAHR("01.01.1900") ergibt den Wert 1900
- ▶ =JAHR("31.12.1899") ergibt den Fehlerwert #WERT!
- ▶ =JAHR("31.12.9999") ergibt den Wert 9999
- ▶ =JAHR("01.01.10000") ergibt den Fehlerwert #WERT!

**Siehe auch** DATUM(), HEUTE(), JETZT(), MONAT(), TAG(), WOCHENTAG(), SEKUNDE(), MINUTE(), STUNDE()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *JAHR*.

# JETZT()



JETZT()

Die Funktion liefert die fortlaufende Zahl des aktuellen Datums und der aktuellen Uhrzeit.

Keine

Bei Eingabe der JETZT()-Funktion erhalten Sie das aktuelle Datum sowie die aktuelle Uhrzeit als Ergebnis. Insbesondere, wenn Sie die aktuelle Uhrzeit für Berechnungen oder zur Anzeige benötigen, ist die Verwendung der Funktion angezeigt. Wenn für das Zellenformat vor der Eingabe der Funktion die Option *Standard* eingestellt war, wird das Ergebnis als Datum formatiert.

Die verwandte Funktion HEUTE() liefert ebenfalls das aktuelle Datum, nicht aber die aktuelle Uhrzeit. Sollten Sie die Funktion JETZT() zu Berechnungen heranziehen – beispielsweise zur Berechnung von Zinstagen – beachten Sie bitte, dass sich die gleichzeitige Ausgabe der Uhrzeit unter Umständen störend auswirken kann. Bevorzugen Sie bei der Bildung von Datumsdifferenzen, bei denen die Differenz in Tagen wesentlich ist, deshalb die Funktion HEUTE().

Die Ergebnisse beider Funktionen werden nicht ständig aktualisiert. Es finden nur dann Aktualisierungen statt, wenn die Tabelle geöffnet oder die zugehörige Tabelle berechnet wird. Die Neuberechnung einer Tabelle können Sie durch Drücken der Taste **F9** erzwingen.

Natürlich ist die Richtigkeit bzw. Genauigkeit beider Funktionen davon abhängig, dass Ihre PC-interne Systemuhr die korrekte Systemzeit führt.

In einem Formular sollen aktuelles Datum und Uhrzeit bei jedem Ausdruck erscheinen. Verwenden Sie dazu die Funktion JETZT() in einer Zelle des Formulars: =JETZT() ergibt das aktuelle Datum und die aktuelle Uhrzeit entsprechend der Systemzeit des PCs. Hier weitere Anwendungsbeispiele:

- ▶ =JETZT()+1/24 ergibt Datum/Uhrzeit in einer Stunde
- ▶ =JETZT()+12/24 ergibt Datum/Uhrzeit in 12 Stunden
- ▶ =JETZT()-1/24 ergibt die aktuelle Zeit in Lissabon (nach Sommerzeit)
- ▶ =JETZT()-6/24 ergibt die aktuelle Zeit in New York (nach Sommerzeit)
- ▶ =JETZT()-9/24 ergibt die aktuelle Zeit in Los Angeles (nach Sommerzeit)
- ▶ =JETZT()+8/24 ergibt die aktuelle Zeit in Sydney (nach Sommerzeit)
- ▶ =JETZT()+7/24 ergibt die aktuelle Zeit in Tokio (nach Sommerzeit)
- ▶ =JETZT()+2/24 ergibt die aktuelle Zeit in Moskau (nach Sommerzeit)

**Syntax**

**Definition**

**Argumente**

**Hintergrund**

**Praxiseinsatz**

| C4 |                                              |                  |                    |
|----|----------------------------------------------|------------------|--------------------|
| A  | B                                            | C                | D                  |
| 3  |                                              | Berechnung       | Uhrzeit (1-24 Uhr) |
| 4  | Jetzt                                        | 27.01.2013 15:11 |                    |
| 5  | In einer Stunde                              | 27.01.2013 16:11 |                    |
| 6  | In 12 Stunden                                | 28.01.2013 03:11 |                    |
| 7  |                                              |                  |                    |
| 8  | Datum/Uhrzeit anderer Zeitzonen (Sommerzeit) |                  |                    |
| 9  | Lissabon                                     | 27.01.2013 14:11 |                    |
| 10 | New York                                     | 27.01.2013 09:11 |                    |
| 11 | Los Angeles                                  | 27.01.2013 06:11 |                    |
| 12 | Sydney                                       | 27.01.2013 23:11 |                    |
| 13 | Tokio                                        | 27.01.2013 22:11 |                    |
| 14 | Moskau                                       | 27.01.2013 17:11 |                    |
| 15 |                                              |                  |                    |

Abbildung 6.5: Ab Excel 2007 können Sie Uhrzeiten durch bedingte Formate auch grafisch darstellen

**Siehe auch** DATUM(), JAHR(), MONAT(), TAG(), STUNDE(), MINUTE(), SEKUNDE(), WOCHENTAG()



Die gezeigten Beispiele finden Sie im Ordner `\\Ms5-235\Kap06` in der Arbeitsmappe `Datum_Zeit.xlsx` auf dem Arbeitsblatt `JETZT`.

## KALENDERWOCHE() WEEKNUM()

**Syntax** KALENDERWOCHE(*Datum*;Rückgabe)

**Definition** Die Funktion gibt eine ganze Zahl zurück, welche die Wochennummer des angegebenen Datums im Jahr darstellt.

**Argumente** *Datum* (erforderlich) ist das Datum, deren KW ermittelt werden soll.

*Zahl\_Typ* (optional) ist eine Zahl, durch die festgelegt wird, mit welchem Tag eine Woche beginnt.

- ▶ 1 (Standardeinstellung): Die Woche beginnt am Sonntag. Die Wochentage sind, bei Sonntag beginnend, von 1 bis 7 nummeriert.
- ▶ 2: Die Woche beginnt am Montag. Die Wochentage sind, bei Montag beginnend, von 1 bis 7 nummeriert.

**Neu in Excel 2010** Seit Excel 2010 kennt die Funktion zwei Systeme und einen erweiterten *Zahl\_Typ*. Die *Zahl\_Typen* 1 bis 17 stehen für das System 1, welches als erste KW diejenige zählt, die den 1. Januar enthält. Der *Zahl\_Typ* 21 steht für das System 2, welches als erste KW diejenige zählt, die den ersten Donnerstag des Jahres enthält (ISO-gemäß).

- ▶ 11: Die Woche beginnt am Montag
- ▶ 12: Die Woche beginnt am Dienstag
- ▶ 13: Die Woche beginnt am Mittwoch
- ▶ 14: Die Woche beginnt am Donnerstag

- ▶ 15: Die Woche beginnt am Freitag
- ▶ 16: Die Woche beginnt am Samstag
- ▶ 17: Die Woche beginnt am Sonntag
- ▶ 21: Die Woche beginnt am Montag

Siehe auch die Funktion ISOKALENDERWOCHE(). Die Berechnung der KW mit dieser Funktion erfolgt standardmäßig nach amerikanischen Regeln (System 1) und damit nicht nach der DIN-Norm. In Excel 2007 und früher gibt es daher immer wieder Jahre, in denen die Berechnung mit der Funktion KALENDERWOCHE() nicht zum gewünschten Ergebnis führt.

#### Hintergrund

Sollten Sie also die KW nach den in Deutschland geltenden Normen benötigen, wählen Sie ab Excel 2010 den *Zahl\_Typ 21* oder in den älteren Excel-Versionen andere Berechnungswege. Im Internet finden Sie verschiedene Lösungen. An dieser Stelle möchten wir zwei Lösungen nennen:

1. Auf der Internetseite <http://www.excelformeln.de/formeln.html?welcher=7> finden Sie Arbeitsblattformeln und Erläuterungen dazu, die nach unseren Prüfungen korrekt arbeiten.
2. Eine in VBA programmierte benutzerdefinierte Funktion finden Sie in den Kapiteln 31 in »Microsoft Office Excel 2007 – Das Handbuch«, ISBN-13: 978-3-86645-103-2, erschienen bei Microsoft Press. Mehr zu benutzerdefinierten Funktionen erfahren Sie in Kapitel 5.

In einem Excel-basierten Einsatzplan soll zu jedem Datum die Kalenderwoche nach US-Norm gezeigt werden. Verwenden Sie hierzu die Formel =KALENDERWOCHE(Datumswert).

#### Praxiseinsatz

Hier noch Hinweise auf problematische Fälle in Excel bis zur Version 2007 hinsichtlich der in Deutschland geltenden Berechnung für die Kalenderwoche:

- ▶ =KALENDERWOCHE("01.01.2005") ergibt mit *Zahl\_Typ* 1 und 2 die KW 1; richtig wäre 53
- ▶ =KALENDERWOCHE("03.01.2005") ergibt mit *Zahl\_Typ* 1 und 2 die KW 2; richtig wäre 1
- ▶ =KALENDERWOCHE("20.10.2005") ergibt mit *Zahl\_Typ* 1 und 2 die KW 43; richtig wäre 42
- ▶ =KALENDERWOCHE("31.12.2012") ergibt mit *Zahl\_Typ* 1 die KW 53 und mit *Zahl\_Typ* 2 die KW 54; richtig wäre 1
- ▶ =KALENDERWOCHE("02.01.2013") ergibt mit *Zahl\_Typ* 1 und 2 die richtige KW 1

ISOKALENDERWOCHE(), JAHR(), MONAT(), TAG(), TAGE(), TAGE360(), WOCHENTAG()

#### Siehe auch

Die gezeigten Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt KALENDERWOCHE.



## MINUTE()



MINUTE(*Zahl*)

#### Syntax

Die Funktion extrahiert aus einer fortlaufenden Zahl (Uhrzeit mit oder ohne Datum) die Minutenzahl. Die Minute wird als ganze Zahl ausgegeben, die einen Wert von 0 bis 59 annehmen kann.

#### Definition

*Zahl* (erforderlich) ist eine gültige Uhrzeit (und ggf. Datumsangabe).

#### Argumente

**Hintergrund** Mithilfe dieser Funktion sowie der Funktionen STUNDE() und SEKUNDE() können Sie eine (gültige) Zeitangabe in ihre Bestandteile zerlegen, um damit z.B. weitere Berechnungen anzustellen.

Zeitangaben können als Textzeichenfolgen in Anführungszeichen (z.B. "06:43"), als Dezimalzahlen (z.B. 0,27986111 – dieser Wert stellt 06:43 Uhr dar) oder als Ergebnis anderer Formeln oder Funktionen eingegeben werden.

**Praxiseinsatz** In einer Anwendung sind die über den vollen Stunden liegenden Minutenwerte zu ermitteln. Die Formel =MINUTE("06:43") ergibt den Wert 43 Minuten.

=MINUTE("24.12.2010 18:12") ergibt den Wert 12.

=MINUTE(JETZT()) ergibt die aktuelle Minute.

**Siehe auch** STUNDE(), SEKUNDE(), JAHR(), MONAT(), TAG(), WOCHENTAG()



Die gezeigten Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe Datum\_Zeit.xlsx auf dem Arbeitsblatt MINUTE.

## MONAT() MONTH()

**Syntax** MONAT(*Zahl*)

**Definition** Die Funktion extrahiert aus einem Datum die darin enthaltene Monatszahl. Die errechneten Monatszahlen liegen im Bereich von 1 (Januar) bis 12 (Dezember).

**Argumente** *Zahl* (erforderlich) ist das Datum, dessen Monatszahl Sie ermitteln wollen.

**Hintergrund** Mithilfe dieser Funktion sowie der Funktionen JAHR() und TAG() können Sie ein (gültiges) Datum in seine Bestandteile zerlegen, um damit z.B. weitere Berechnungen anzustellen. Diese Funktion können Sie immer dann einsetzen, wenn in Tabellen Werte nach Monaten zusammengefasst werden sollen und es nicht mehr auf das Tagesdatum ankommt.

Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden. Die von den Funktionen JAHR(), MONAT() und TAG() zurückgegebenen Werte sind gregorianische Werte (entsprechend des Gregorianischen Kalenders).

**Praxiseinsatz** Um die Filterung von Monaten mit dem Excel-AutoFilter bzw. die Sortierung nach Monaten zur Verfügung zu stellen, sollen in einer Hilfsspalte die Monatswerte aus einer Datumsspalte ermittelt werden. Verwenden Sie dazu die Berechnung =MONAT(Datumswert). Hier weitere Anwendungen und Hinweise:

- ▶ =MONAT(HEUTE()) ergibt im September den Wert 9
- ▶ =MONAT("14.11.1959") ergibt den Wert 11
- ▶ =MONAT("01.01.1900") ergibt den Wert 1
- ▶ =MONAT("31.12.1899") ergibt den Fehlerwert #WERT!
- ▶ =MONAT("31.12.9999") ergibt den Wert 12
- ▶ =MONAT("01.01.10000") ergibt den Fehlerwert #WERT!



JAHR(), TAG(), WOCHENTAG(), ISOKALENDERWOCHE(), KALENDERWOCHE(), STUNDE(), MINUTE(), SEKUNDE()

Siehe auch

Die Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *MONAT*.



## MONATSENDE() EOMONTH()

MONATSENDE(*Ausgangsdatum*; *Monate*)

Syntax

Diese Funktion gibt die Zahl des letzten Tags in dem Monat zurück, der die im Argument *Monate* angegebene Anzahl von Monaten vor bzw. nach dem *Ausgangsdatum* liegt.

Definition

*Ausgangsdatum* (erforderlich) ist ein Datum, das den Starttag der Berechnungen markiert.

Argumente

*Monate* (erforderlich) gibt an, wie viele Monate vor oder nach dem *Ausgangsdatum* liegen sollen. Ein positiver Wert für *Monate* ergibt ein in der Zukunft, ein negativer Wert ein in der Vergangenheit liegendes Datum.

Die Funktion hat ihr Haupteinsatzgebiet im Bereich kaufmännische Berechnungen und im Bankwesen. Mit MONATSENDE() können Sie Rückzahlungs- oder Fälligkeitstermine berechnen, die auf den letzten Tag eines Monats fallen.

Hintergrund

Eine Alternative zu dieser Funktion bestünde darin, mit den Funktionen JAHR() und MONAT() das *Ausgangsdatum* in seine Bestandteile zu zerlegen, die ermittelte Monatszahl um den Wert *Monate+1* zu erhöhen, für den Tag eine *1* zu setzen, alle drei Bestandteile anschließend wieder mithilfe der Funktion DATUM() zu einem regulären Datum (fortlaufende Zahl) zusammensetzen und von diesem Datum einen Tag abzuziehen:

=DATUM(JAHR(Ausgangsdatum);MONAT(Ausgangsdatum)+Monate+1;1)-1

Probleme können auftreten, wenn Sie Datumsangaben als Text eingeben. Ist *Monate* keine ganze Zahl, werden die Nachkommastellen abgeschnitten. Ist *Ausgangsdatum* kein zulässiges Datum, gibt MONATSENDE() den Fehlerwert #ZAHL! zurück. Ergibt die Berechnung *Ausgangsdatum plus Monate* ein ungültiges Datum, gibt die Funktion den Fehlerwert #ZAHL! zurück.

Für eine am 1.01.2012 beginnende Finanzierung mit 18 Monaten Laufzeit soll als Fälligkeitstermin der Monatsletzte nach 18 Monaten ausgewiesen werden. Die Formel

Praxiseinsatz

=MONATSENDE("01.01.2012";18)

ergibt als Fälligkeitsdatum den 31.07.2013 (siehe Abbildung 6.6).

| D4 |                      |               |                   |
|----|----------------------|---------------|-------------------|
| A  | B                    | C             | D                 |
| 1  | <b>MONATSENDE</b>    |               |                   |
| 2  |                      |               |                   |
| 3  | <i>Ausgangsdatum</i> | <i>Monate</i> | <i>Berechnung</i> |
| 4  | 01.01.2012           | 18            | 31.07.2013        |
| 5  | 27.01.2013           | 6             | 31.07.2013        |
| 6  | 01.12.2013           | 2             | 28.02.2014        |

**Abbildung 6.6:** Das Fälligkeitsdatum einer Finanzierung berechnen

**Siehe auch** ARBEITSTAG(), ARBEITSTAG.INTL(), DATEDIF(), DATUM(), EDATUM(), NETTOARBEITSTAGE(), NETTOARBEITSTAGE.INTL(), TAGE(), TAGE360()



Die dargestellten Beispiele finden Sie im Ordner `\Ms5-235\Kap06` in der Arbeitsmappe `Datum_Zeit.xlsx` auf dem Arbeitsblatt `MONATSENDE`.

## NETTOARBEITSTAGE() NETWORKDAYS()

**Syntax** NETTOARBEITSTAGE(*Ausgangsdatum*; *Enddatum*; *Freie\_Tage*)

**Definition** Die Funktion gibt die Anzahl der Arbeitstage für das angegebene Zeitintervall zurück.

**Argumente** *Ausgangsdatum* (erforderlich) ist ein Datum, das den Beginn des Zeitintervalls angibt.  
*Enddatum* (erforderlich) ist ein Datum, das das Ende des Zeitintervalls angibt.

*Freie\_Tage* (optional) ist ein Bereich mit einer oder mehreren Datumsangaben, der alle Arten von arbeitsfreien Tagen repräsentieren kann, die aus dem Arbeitskalender ausgeschlossen werden sollen, z.B. staatliche oder regionale Feiertage. Bei der Liste kann es sich entweder um einen Zellbereich, der die Datumsangaben enthält, oder eine Matrixkonstante der fortlaufenden Zahlen handeln, die die Datumsangaben darstellen.

**Hintergrund** Unter Zuhilfenahme der Funktion NETTOARBEITSTAGE() können Sie die für Arbeitnehmer zu zahlenden Leistungen berechnen, die auf der zu einem bestimmten Zeitraum gehörenden Anzahl an Arbeitstagen basieren.

Standardmäßig lässt die Funktion bei der Berechnung der Zeitdifferenz in Tagen die im Intervall enthaltenen Wochenenden, also die Samstag und Sonntage, weg. Zusätzlich können Sie Ihre Feiertagsliste in einen Zellbereich schreiben und diesen dann als *Freie\_Tage*-Liste angeben, um diese Datumsangaben ebenfalls aus der Berechnung herauszunehmen.

Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden. Ist eines der Argumente kein zulässiges Datum, gibt NETTOARBEITSTAGE() den Fehlerwert `#WERT!` zurück.

Beachten Sie, dass die Funktion den Tag, den *Ausgangsdatum* darstellt, bei der Zählung mit berücksichtigt. Wenn Sie also das Ergebnis mithilfe der Funktion ARBEITSTAG() gegenrechnen, müssen Sie ARBEITSTAG() immer einen Tag weniger im Argument *Tage* übergeben.

**Praxiseinsatz** Ein Projekteinsatz wurde vom 12.12.2012 bis zum 6.6.2013 festgelegt. Es soll die Anzahl der abrechenbaren Arbeitstage in diesem Zeitraum unter Berücksichtigung der Feiertage ermittelt werden. Die Formel

```
=NETTOARBEITSTAGE("12.12.2012";"06.06.2013";{"25.12.2012";"26.12.2012";"01.01.2013";
"01.04.2013";"01.05.2013";"09.05.2013";"20.05.2013"})
```

ermittelt 120 abrechenbare Arbeitstage für den festgelegten Projektzeitraum.

**Hinweis** Beachten Sie bitte in der obigen Formel, dass die Liste der freien Tage von geschweiften Klammern {} und nicht von runden Klammern umgeben sein muss!

Die Abbildung 6.7 zeigt die Berechnung unter Verwendung von Zellbezügen auf Start, Ende und Feiertagsliste.

| =NETTOARBEITSTAGE(B8;C8;D8:D14) |                         |                     |                 |                   |                   |
|---------------------------------|-------------------------|---------------------|-----------------|-------------------|-------------------|
|                                 | A                       | B                   | C               | D                 | E                 |
| 1                               | <b>NETTOARBEITSTAGE</b> |                     |                 |                   |                   |
| 2                               |                         |                     |                 |                   |                   |
| 3                               |                         | <i>Anfangsdatum</i> | <i>Enddatum</i> | <i>Freie_Tage</i> | <i>Berechnung</i> |
| 8                               |                         | 12.12.2012          | 06.06.2013      | 25.12.2012        | 120               |
| 9                               |                         |                     |                 | 26.12.2012        |                   |
| 10                              |                         |                     |                 | 01.01.2013        |                   |
| 11                              |                         |                     |                 | 01.04.2013        |                   |
| 12                              |                         |                     |                 | 01.05.2013        |                   |
| 13                              |                         |                     |                 | 09.05.2013        |                   |
| 14                              |                         |                     |                 | 20.05.2013        |                   |

**Abbildung 6.7:** Die abrechenbaren Arbeitstage unter Berücksichtigung der Wochenenden und Feiertage ermitteln

ARBEITSTAG(), ARBEITSTAG.INTL(), EDATUM(), MONATSENDE(), NETTOARBEITSTAGE.INTL(), DATEDIF(), TAGE(), TAGE360()

Das gezeigte und weitere Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *NETTOARBEITSTAGE*.

**Siehe auch**



## NETTOARBEITSTAGE.INTL()



## NETWORKDAYS.INTL()

NETTOARBEITSTAGE.INTL(*Ausgangsdatum*; *Enddatum*; *Wochenende*; *Freie\_Tage*)

Die Funktion gibt die Anzahl der Arbeitstage für das angegebene Zeitintervall unter Beachtung der Wochenenddefinition zurück.

*Ausgangsdatum* (erforderlich) ist ein Datum, das den Beginn des Zeitintervalls angibt.

*Enddatum* (erforderlich) ist ein Datum, das das Ende des Zeitintervalls angibt.

*Wochenende* (optional) ist eine Zahl oder Zeichenfolge, die angibt, welche Tage als Wochenende und nicht als Arbeitstage bewertet werden.

*Freie\_Tage* (optional) ist ein Bereich mit einer oder mehreren Datumsangaben, der alle Arten von arbeitsfreien Tagen repräsentieren kann, die aus dem Arbeitskalender ausgeschlossen werden sollen, z.B. staatliche oder regionale Feiertage. Bei der Liste kann es sich entweder um einen Zellbereich, der die Datumsangaben enthält, oder eine Matrixkonstante der fortlaufenden Zahlen handeln, die die Datumsangaben darstellen.

Siehe die Funktion NETTOARBEITSTAGE(). Zusätzlich haben Sie die Möglichkeit, festzulegen, welche Wochentage als Wochenende bewertet und damit nicht mitgezählt werden. Wenn Sie die Wochenendtage mit einer Zahl oder einer Zeichenfolge angeben, tun Sie dies mit diesen Werten:

**Neu in Excel 2010**

**Syntax**

**Definition**

**Argumente**

**Hintergrund**

| Zahl                  | Wochentage           | Entspricht der Zeichenfolge |
|-----------------------|----------------------|-----------------------------|
| 1 (oder keine Angabe) | Samstag, Sonntag     | 0000011                     |
| 2                     | Sonntag, Montag      | 1000001                     |
| 3                     | Montag, Dienstag     | 1100000                     |
| 4                     | Dienstag, Mittwoch   | 0110000                     |
| 5                     | Mittwoch, Donnerstag | 0011000                     |
| 6                     | Donnerstag, Freitag  | 0001100                     |
| 7                     | Freitag, Samstag     | 0000110                     |
| 11                    | Nur Sonntag          | 0000001                     |
| 12                    | Nur Montag           | 1000000                     |
| 13                    | Nur Dienstag         | 0100000                     |
| 14                    | Nur Mittwoch         | 0010000                     |
| 15                    | Nur Donnerstag       | 0001000                     |
| 16                    | Nur Freitag          | 0000100                     |
| 17                    | Nur Samstag          | 0000010                     |

Außer der Zeichenfolge 111111 sind alle weiteren Kombinationen möglich. Die 0 steht für einen Arbeits-, die 1 für einen Feiertag.

**Praxiseinsatz** Ein Projekteinsatz wurde vom 12.12.2012 bis zum 06.06.2013 festgelegt. Es soll die Anzahl der abrechenbaren Arbeitstage in diesem Zeitraum unter Berücksichtigung der Feiertage ermittelt werden. Die Formel

```
=NETTOARBEITSTAGE.INTL("12.12.2012";"06.06.2013";1;{"25.12.2012";"26.12.2012";
"01.01.2013";"01.04.2013";"01.05.2013";"09.05.2013";"20.05.2013"})
```

ermittelt 120 abrechenbare Arbeitstage für den festgelegten Projektzeitraum.

**Hinweis** Beachten Sie bitte in der obigen Formel, dass die Liste der freien Tage von geschweiften Klammern {} und nicht von runden Klammern umgeben sein muss!

**Siehe auch** ARBEITSTAG(), ARBEITSTAG.INTL(), EDATUM(), MONATSENDE(), NETTOARBEITSTAGE(), DATEDIF(), TAGE(), TAGE360()



Das gezeigte und weitere Beispiele finden Sie im Ordner `Ms5-235\Kap06` in der Arbeitsmappe `Datum_Zeit.xlsx` (Excel 2010) auf dem Arbeitsblatt `NETTOARBEITSTAGE.INTL`.

## SEKUNDE() SECOND()

**Syntax** SEKUNDE(*Zahl*)

**Definition** Die Funktion extrahiert aus einer fortlaufenden Zahl (Uhrzeit mit oder ohne Datum) die Sekundenzahl. Die Sekunde wird als ganze Zahl ausgegeben, die einen Wert von 0 bis 59 annehmen kann.

**Argumente** *Zahl* (erforderlich) ist die eine gültige Uhrzeit (und ggf. Datum).

Mithilfe dieser Funktion sowie der Funktionen STUNDE() und MINUTE() können Sie eine (gültige) Zeitangabe in ihre Bestandteile zerlegen, um damit z.B. weitere Berechnungen anzustellen.

Zeitangaben können als Textzeichenfolgen in Anführungszeichen (z.B. "06:43"), als Dezimalzahlen (z.B. 0,27986111 – dieser Wert stellt 06:43 Uhr dar) oder als Ergebnis anderer Formeln oder Funktionen eingegeben werden.

- ▶ =SEKUNDE("13.07.2008 20:48:31") ergibt den Wert 31
- ▶ =SEKUNDE("06:43:12") ergibt den Wert 12
- ▶ =SEKUNDE(JETZT()) ergibt die aktuelle Sekunde

STUNDE(), MINUTE(), JAHR(), MONAT(), TAG(), JETZT(), ZEIT()

Die Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *SEKUNDE*.

## STUNDE() HOUR()

STUNDE(*Zahl*)

Die Funktion extrahiert aus einer fortlaufenden Zahl (Uhrzeit mit oder ohne Datum) die Stundenzahl. Die Stunde wird als ganze Zahl ausgegeben, die einen Wert von 0 bis 23 annehmen kann.

*Zahl* (erforderlich) ist die eine gültige Uhrzeit (und ggf. Datum).

Mithilfe dieser Funktion sowie der Funktionen MINUTE() und SEKUNDE() können Sie eine (gültige) Zeitangabe in ihre Bestandteile zerlegen, um damit z.B. weitere Berechnungen anzustellen.

Zeitangaben können als Textzeichenfolgen in Anführungszeichen (z.B. "06:43"), als Dezimalzahlen (z.B. 0,27986111 – dieser Wert stellt 06:43 Uhr dar) oder als Ergebnis anderer Formeln oder Funktionen eingegeben werden.

In einer Zeiterfassung sollen nur die vollen Stunden aus gegebenen Uhrzeiten weiterberechnet werden. Die Formel =STUNDE("06:43") ergibt den Wert von 6 Stunden. Weitere Beispiele:

- ▶ =STUNDE("27.01.2013 17:56") ergibt den Wert 17
- ▶ =STUNDE(JETZT()) ergibt die aktuelle Stunde

MINUTE(), SEKUNDE(), JAHR(), MONAT(), TAG(), JETZT(), ZEIT()

Die gezeigten Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *STUNDE*.

### Hintergrund

### Praxiseinsatz

### Siehe auch



### Syntax

### Definition

### Argumente

### Hintergrund

### Praxiseinsatz

### Siehe auch



## TAG() DAY()

**Syntax** TAG(*Zahl*)

**Definition** Die Funktion extrahiert aus einem Datum die darin enthaltene Tagzahl. Das Ergebnis von TAG() ist eine ganze Zahl, die einen Wert von 1 bis 31 annehmen kann.

**Argumente** *Zahl* (erforderlich) ist das Datum, dessen Tagzahl Sie ermitteln wollen.

**Hintergrund** Mithilfe dieser Funktion sowie der Funktionen JAHR() und MONAT() können Sie ein (gültiges) Datum in seine Bestandteile zerlegen, um damit z.B. weitere Berechnungen anzustellen. Diese Funktion können Sie immer dann einsetzen, wenn in Tabellen Werte nach Tagen zusammengefasst werden sollen und es nicht mehr auf das Gesamtdatum ankommt.

Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden. Die von den Funktionen JAHR(), MONAT() und TAG() zurückgegebenen Werte sind gregorianische Werte (entsprechend des Gregorianischen Kalenders).

**Praxiseinsatz** Um in einer Geburtstagsliste die Filterung von Tagen mit dem Excel-*AutoFilter* bzw. die Sortierung nach Tagen zur Verfügung zu stellen, sollen in einer Hilfsspalte die Tagzahlen aus einer Datumsspalte ermittelt werden. Verwenden Sie dazu die Berechnung =TAG(Datumswert). Hier weitere Anwendungen und Hinweise:

- ▶ =TAG("13.07.2013") ergibt den Wert 13
- ▶ =TAG("14.11.1959") ergibt den Wert 14
- ▶ =TAG("01.01.1900") ergibt den Wert 1
- ▶ =TAG("31.12.1899") ergibt den Fehlerwert #WERT!
- ▶ =TAG("31.12.9999") ergibt den Wert 31
- ▶ =TAG("01.01.10000") ergibt den Fehlerwert #WERT!

**Siehe auch** DATUM(), JAHR(), MINUTE(), MONAT(), STUNDE(), SEKUNDE(), WOCHENTAG(), ZEIT()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *TAG*.

## Neu in Excel 2013 TAGE() DAYS()

**Syntax** TAGE(*Zieldatum*; *Ausgangsdatum*)

**Definition** Diese Funktion berechnet die Anzahl der zwischen *Ausgangs-* und *Zieldatum* liegenden Tage.

**Argumente** *Ausgangsdatum* (erforderlich) und *Zieldatum* (erforderlich) sind die beiden Tagesdaten, für die Sie die dazwischenliegenden Tage berechnen möchten.

**Hintergrund** Diese neue Funktion komplettiert die Datumsberechnungsfunktionen. Im Grunde erfolgt die Berechnung immer, indem das Ausgangsdatum vom Zieldatum abgezogen wird – eine Berechnung, die Sie sicher auch bisher schon durch einfaches Subtrahieren des einen Datums vom anderen durchgeführt haben.

Sind ein oder beide Parameter als Datumstext angegeben, bildet Excel den entsprechenden Datumswert und berechnet damit die korrekte Tagesdifferenz. Übergeben Sie in einem oder beiden Parametern Zeichenfolgen, die nicht als Datumstext erkannt werden, wird der Fehlerwert **#WERT!** ausgegeben.

Enthalten beide oder ein Parameter einen numerischen Wert, der keine gültige Datumszahl darstellt, wird der Fehlerwert **#ZAHL!** zurückgegeben.

Im Mahnwesen muss geprüft werden, wie viele Tage seit Rechnungslegung vergangen sind. Überschreitet die Differenz die vereinbarten Zahlungsziele, wird die Mahnung ausgelöst. Zur Berechnung der Zeitdifferenz hier einige Beispiele und Hinweise:

- ▶ =TAGE("31.12.2013";"31.12.2012") ergibt den Wert 365
- ▶ =TAGE("01.01.2013";"31.12.2013") ergibt den Wert -364 (negativer Wert, weil das *Zieldatum* kleiner als das *Ausgangsdatum* ist)
- ▶ =TAGE("31.12.2012";"01.01.2012") ergibt den Wert 365
- ▶ =TAGE("01.01.2013";"01.01.2012") ergibt den Wert 366 (weil 2012 ein Schaltjahr war).
- ▶ =TAGE("29.02.2013";"01.01.2012") ergibt den Fehler **#WERT!** (den 29.02.2013 gibt bzw. gab es nicht)
- ▶ =TAGE(2958466;"01.01.2014") ergibt den Fehler **#ZAHL!** (der Datumswert 2.958.466 stellt kein gültiges Datum dar)

DATEDIF(), EDATUM(), MONATSENDE(), NETTOARBEITSTAGE(), NETTOARBEITSTAGE.INTL(), TAGE360()

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap06` in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *TAGE*.

**Praxiseinsatz**

**Siehe auch**



## TAGE360() DAYS360()

TAGE360(*Ausgangsdatum*; *Enddatum*; *Methode*)

Diese Funktion berechnet, ausgehend von einem Jahr, das 360 Tage umfasst, die Anzahl der zwischen *Ausgangs-* und *Enddatum* liegenden Tage.

*Ausgangsdatum* (erforderlich) und *Enddatum* (erforderlich) sind die beiden Tagesdaten, für die Sie die dazwischenliegenden Tage berechnen möchten.

*Methode* (optional) ist ein Wahrheitswert, der angibt, ob in der Berechnung die amerikanische oder die europäische Methode verwendet werden soll. Hier die Bedeutung der Wahrheitswerte:

- ▶ **FALSCH** (oder nicht angegeben): US-Methode (NASD)  
Ist das *Ausgangsdatum* der 31. eines Monats, wird dieses Datum zum 30. desselben Monats. Ist das *Enddatum* der 31. eines Monats und das *Ausgangsdatum* ein Datum vor dem 30. eines Monats, wird das *Enddatum* zum 1. des darauffolgenden Monats. In allen anderen Fällen wird das *Enddatum* zum 30. desselben Monats.
- ▶ **WAHR**: Europäische Methode  
Jedes Ausgangs- und Enddatum, das auf den 31. eines Monats fällt, wird zum 30. desselben Monats.

**Syntax**

**Definition**

**Argumente**

**Hintergrund** Sie können diese Funktion als Hilfe für die Berechnung von taggenauen Zinsen verwenden, wenn die Zinsen nach den in Deutschland üblichen Gepflogenheiten ermittelt werden sollen. Dabei hat das Jahr 360 Tage und jeder Monat wird mit 30 Tagen berechnet.

Steht das *Ausgangsdatum* nach dem *Enddatum*, liefert TAGE360() eine negative Zahl, was in Zinsberechnungen unweigerlich zu einem Fehlergebnis führt. Achten Sie bei Zinsberechnungen auch darauf, dass der Zinsfuß in der Excel-Prozentformatierung angegeben ist. Bei der Berechnung gehen Sie wie folgt vor:

Zuerst berechnen Sie durch einfache Multiplikation des Kapitals mit dem Zinsfuß, wie viel Zinsen in einem Jahr anfallen würden. Diesen Wert dividieren Sie anschließend durch 360, um festzustellen, wie viel Zinsen an einem Tag anfallen. Diese multiplizieren Sie schließlich mit der Anzahl der Tage, die Sie mit der Funktion TAGE360() ermitteln. Wenn Sie diese Aussage in eine Formel auf einen Bruchstrich schreiben, ergibt sich folgende Formel:

$$\text{Zinsen} = \frac{\text{Kapital} \times \text{Zinssatz} \times \text{TAGE360}(\text{Anfangsdatum}; \text{Enddatum}; \text{WAHR})}{360}$$

Falls notwendig, können Sie das Ergebnis dieser Berechnung noch mit der Funktion RUNDEN(), siehe auch Kapitel 15, auf die zweite Dezimalstelle runden.

**Praxiseinsatz** Angenommen, Sie möchten für ein Kapital von 250.000 € (in Zelle C13 erfasst) bei einem Zinssatz von 5,25 % (in E13) die taggenauen Zinsen vom 22.11.2012 (in C14) bis zum 31.05.2013 (in E14) berechnen. Bei der Eingabe des Zinssatzes haben Sie darauf geachtet, dass der Prozentsatz in der Excel-Formatierung eingegeben wurde. Zur Berechnung der Zinstage setzen Sie die Funktion TAGE360() ein:

=C13\*E13\*TAGE360(C14;E14;WAHR)/360

Das Ergebnis beträgt 6.854,17 €. Die Teilberechnung TAGE360(C14;E14;WAHR) ermittelt 188 Zinstage (siehe Abbildung 6.8).

|    | A                                                     | B            | C          | D        | E          |
|----|-------------------------------------------------------|--------------|------------|----------|------------|
| 1  | <b>TAGE360</b>                                        |              |            |          |            |
| 2  |                                                       |              |            |          |            |
| 11 | <b>Taggenaue Zinsen berechnen (mit 360-Tage-Jahr)</b> |              |            |          |            |
| 12 |                                                       |              |            |          |            |
| 13 |                                                       | Kapital      | 250.000 €  | Zinssatz | 5,25%      |
| 14 |                                                       | Anfangsdatum | 22.11.2012 | Enddatum | 31.05.2013 |
| 15 |                                                       |              |            |          |            |
| 16 |                                                       | Zinstage     | 188        | Zinsen   | 6.854,17 € |
| 17 |                                                       |              |            |          |            |

**Abbildung 6.8:** Mithilfe der TAGE360()-Funktion können Sie taggenaue Zinsen errechnen

**Siehe auch** DATEDIF(), EDATUM(), MONATSENDE(), NETTOARBEITSTAGE(), NETTOARBEITSTAGE.INTL(), TAGE()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *TAGE360*.



# WOCHENTAG()



WOCHENTAG(*Zahl*;Typ)

Die Funktion wandelt ein Datum (fortlaufende Zahl) in eine Wochentagsnummer um. Der Wochentag wird als ganze Zahl ausgegeben, die einen Wert von 1 (Sonntag) bis 7 (Samstag) annehmen kann.

*Zahl* (erforderlich) ist ein Datum, dessen Wochentagsnummer Sie ermitteln wollen.

*Typ* (optional) ist eine Zahl (1, 2, oder 3), die den Typ des Rückgabewerts bestimmt. Die Typangaben haben folgende Bedeutung für die Ergebniszahl:

- ▶ 1 (oder nicht angegeben): Zahl 1 steht für Sonntag, 2 für Montag, ..., 7 für Samstag
- ▶ 2: Zahl 1 steht für Montag, 2 für Dienstag, ..., 7 für Sonntag
- ▶ 3: Zahl 0 steht für Montag, 1 für Dienstag, ..., 6 für Sonntag

Seit Excel 2010 kennt die Funktion erweiterte *Typ*-Angaben 11 bis 17:

- ▶ 11: Die Woche beginnt am Montag (1) und endet am Sonntag (7)
- ▶ 12: Die Woche beginnt am Dienstag (1) und endet am Montag (7)
- ▶ 13: Die Woche beginnt am Mittwoch (1) und endet am Dienstag (7)
- ▶ 14: Die Woche beginnt am Donnerstag (1) und endet am Mittwoch (7)
- ▶ 15: Die Woche beginnt am Freitag (1) und endet am Donnerstag (7)
- ▶ 16: Die Woche beginnt am Samstag (1) und endet am Freitag (7)
- ▶ 17: Die Woche beginnt am Sonntag (1) und endet am Samstag (7)

Diese Funktion kann Ihnen immer dann nützlich sein, wenn es darum geht, aus einer Datumsreihe bestimmte Wochentage herauszufiltern. Sie leistet z.B. gute Dienste, wenn Sie Wochenendtage hervorheben wollen.

Denken Sie daran, dass Sie bei der Berechnung eines Wochentags als Wort nicht unbedingt den Umweg über die Funktion WOCHENTAG() gehen müssen. Der Einsatz der Funktion TEXT(), siehe auch Kapitel 7, hilft hier auch weiter:

```
=TEXT(HEUTE());"TTTT")
```

In einer Datumsspalte sollen die Sonntage durch fett rote Schrift hervorgehoben werden. Setzen Sie hierzu die *Bedingte Formatierung* ein. Markieren Sie zuerst die Datumsreihe. Klicken Sie im Menüband auf der Registerkarte *START* in der Gruppe *Formatvorlagen* auf das Symbol *Bedingte Formatierung* und im Dropdownmenü auf den Befehl *Neue Regel*. Als Regeltyp wählen Sie den Eintrag *Formel zur Ermittlung der zu formatierenden Zellen verwenden* (siehe Abbildung 6.9).

## Syntax

## Definition

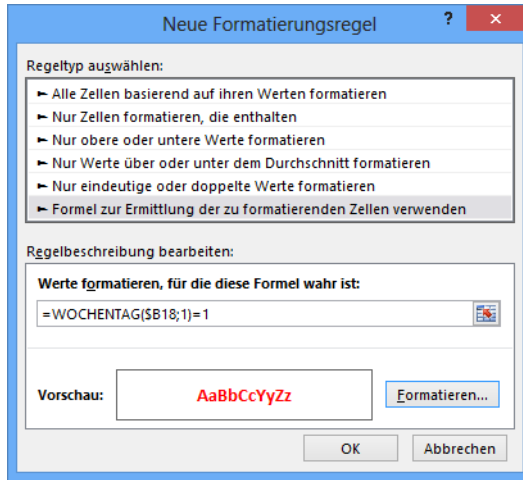
## Argumente

## Neu in Excel 2010

## Hintergrund

## Praxiseinsatz





**Abbildung 6.9:** Die Hervorhebung aller Sonntage durch die bedingte Formatierung in Excel 2013

Als Formel geben Sie `=WOCHENTAG($B18;1)=1` ein und richten über einen Klick auf die Schaltfläche *Formatieren* die Schrifthervorhebung ein. Ein mögliches Ergebnis zeigt die Abbildung 6.10.

|           | A | B                                | C |
|-----------|---|----------------------------------|---|
| 16        |   | <b>Alle Sonntage hervorheben</b> |   |
| 17        |   |                                  |   |
| 18        |   | <b>27.01.2013</b>                |   |
| 19        |   | 28.01.2013                       |   |
| 20        |   | 29.01.2013                       |   |
| 21        |   | 30.01.2013                       |   |
| 22        |   | 31.01.2013                       |   |
| 23        |   | 01.02.2013                       |   |
| 24        |   | 02.02.2013                       |   |
| <b>25</b> |   | <b>03.02.2013</b>                |   |
| 26        |   | 04.02.2013                       |   |
| 27        |   | 05.02.2013                       |   |
| 28        |   | 06.02.2013                       |   |
| 29        |   | 07.02.2013                       |   |
| 30        |   | 08.02.2013                       |   |
| 31        |   | 09.02.2013                       |   |
| 32        |   | <b>10.02.2013</b>                |   |
| 33        |   | 11.02.2013                       |   |

**Abbildung 6.10:** Die Sonntage in der Datumsspalte sind durch eine bedingte Formatierung automatisch fett und rot markiert

Hier noch die unterschiedlichen Wirkungen des Parameters *Typ*:

- ▶ `=WOCHENTAG("03.08.2013";1)` ergibt den Wert 7 (Samstag)
- ▶ `=WOCHENTAG("03.08.2013";2)` ergibt den Wert 6
- ▶ `=WOCHENTAG("03.08.2013";3)` ergibt den Wert 5

**Siehe auch** JAHR(), MONAT(), TAG(), STUNDE(), MINUTE(), SEKUNDE(), HEUTE(), ISOKALENDERWOCHE(), KALENDERWOCHE()



Die Beispiele finden Sie im Ordner `\Ms5-235\Kap06` in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *WOCHENTAG*.

# ZEIT()



## TIME()

ZEIT(*Stunde;Minute;Sekunde*)

Die Funktion gibt die fortlaufende Dezimalzahl zurück, die eine durch die Argumente *Stunde*, *Minute* und *Sekunde* bestimmte Uhrzeit darstellt.

**Stunde** (erforderlich) ist eine Zahl von 0 (Null) bis 32.767, die die Stunde angibt. Jeder Wert, der größer als 23 ist, wird durch 24 geteilt und der Rest als Wert für die Stunde angenommen. Beispiel: ZEIT(28;0;0) entspricht ZEIT(4;0;0) – 04:00 Uhr oder 4/24 oder 0,16667.

**Minute** (erforderlich) ist eine Zahl von 0 bis 32.767, die die Minute angibt. Jeder Wert, der größer als 59 ist, wird durch 24\*60 geteilt und in Stunden und Minuten umgerechnet. Beispiel: ZEIT(0;150;0) entspricht ZEIT(2;30;0) – 02:30 Uhr oder 0,1041667.

**Sekunde** (erforderlich) ist eine Zahl von 0 bis 32.767, die die Sekunde angibt. Jeder Wert, der größer als 59 ist, wird durch 24\*60\*60 geteilt und in Stunden, Minuten und Sekunden umgerechnet. Beispiel: ZEIT(0;0;12011) entspricht ZEIT(3;20;11) – 03:20:11 Uhr oder 0,139016204.

Bei Uhrzeitberechnungen gibt es häufig den Anlass, eine Uhrzeit zu zerlegen, um z.B. Minuten zu addieren. Das Berechnungsergebnis muss aber wieder in einen regulären Zeitwert einfließen. Hier setzt die Funktion ZEIT() an: Aus den angegebenen Einzelbestandteilen einer Uhrzeit wird wieder ein (numerischer) Zeitwert gebildet, welchen Sie in weiteren Zeitberechnungen einsetzen können. Die ZEIT()-Funktion eignet sich am besten für Formeln, in denen *Stunde*, *Minute* und *Sekunde* Formeln und keine Konstanten sind.

Eine von der Funktion zurückgegebene Dezimalzahl ist ein Wert im Bereich von 0 bis 0,99999999 und entspricht einer Uhrzeit von 00:00:00 (24:00:00) bis 23:59:59 Uhr. Wenn für das Zellenformat vor der Eingabe der Funktion die Option *Standard* eingestellt war, wird das Ergebnis als Uhrzeit formatiert.

In einer Anwendung liegen Normzeiten für technische Abläufe in Stunden, Minuten und Sekunden zerlegt vor. Die drei Normen sollen zu einer Zeitangabe zusammengefügt werden, welche als Gesamtdauer mit nicht mehr als 24 Stunden ausgegeben wird. Verwenden Sie hierzu die Formel =ZEIT(Normstunden;Normminuten;Normsekunden). Die folgenden Beispiele zeigen mögliche Ergebnisse und Besonderheiten bei Überschreitung der 24-Stunden-, der 60-Minuten- bzw. der 60-Sekundengrenze:

- ▶ =ZEIT(5;6;11) ergibt den Zeitwert 05:06:11
- ▶ =ZEIT(13;10;0) ergibt den Zeitwert 13:10:00
- ▶ =ZEIT(23;45;30) ergibt den Zeitwert 23:45:30
- ▶ =ZEIT(24;15;30) ergibt den Zeitwert 00:15:30
- ▶ =ZEIT(26;30;30) ergibt den Zeitwert 02:30:30
- ▶ =ZEIT(12;80;10) ergibt den Zeitwert 13:20:10
- ▶ =ZEIT(12;59;120) ergibt den Zeitwert 13:01:00

DATUM(), STUNDE(), MINUTE(), SEKUNDE(), JAHR(), MONAT(), TAG(), ZEITWERT()

Die gezeigten Beispiele finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *ZEIT*.

**Syntax**

**Definition**

**Argumente**

**Hintergrund**

**Praxiseinsatz**

**Siehe auch**



## ZEITWERT() TIMEVALUE()

**Syntax** ZEITWERT(*Zeit*)

**Definition** Die Funktion konvertiert eine als Text vorliegende Zeitangabe in einen Zeitwert (fortlaufende Zahl). Diese fortlaufende Zahl ist ein Wert im Bereich von 0 (Null) bis 0,99999999 und entspricht einer Uhrzeit von 00:00:00 (24:00:00) bis 23:59:59 Uhr.

**Argumente** *Zeit* (erforderlich) ist die Zeichenfolge einer Zeitangabe in einem beliebigen Microsoft Excel-Zeitformat. Datumsangaben im Argument *Zeit* werden bei der Berechnung ignoriert.

**Hintergrund** In allen Fällen, wo Ihnen eine Uhrzeitangabe im Textformat vorliegt (bspw. aus Importen), verwenden Sie ZEITWERT(), um die vorliegende Uhrzeit in eine fortlaufende Zahl, sprich Zeitwert, umzuwandeln. Mit dem Ergebnis können Sie dann weitere Berechnungen anstellen.

Die meisten Funktionen wandeln zwar Uhrzeittextwerte automatisch in fortlaufende Zahlen um, aber in bestimmten Fällen kann diese Regel einmal nicht gelten, z.B. bei importierten Daten oder bei Arbeitsblatffunktionen aus Drittanbieter-Add Ins. Wenn Sie solche Fälle ausschließen und sichergehen wollen, wandeln Sie die Werte mit ZEITWERT() um.

**Praxiseinsatz** Nach einem Import erscheinen die Werte einer Uhrzeitspalte als Text. Diese sollen für weitere Berechnungen in echte Zeit-, also numerische Werte umgewandelt werden. Die Formel  
=ZEITWERT("06:00:00")

ergibt den Zeitwert 0,25 oder mit dem Zahlenformat *hh:mm* formatiert 06:00, welcher einen Zeitwert im Excel-Uhrzeitsystem darstellt. Hier weitere Beispiele:

- ▶ =ZEITWERT("06:00 PM") ergibt den Wert 0,75
- ▶ =ZEITWERT("06:45:16") ergibt den Wert 0,281435185
- ▶ =ZEITWERT("12:00:00") ergibt den Wert 0,5

**Siehe auch** DATWERT(), ZEIT(), STUNDE(), MINUTE(), SEKUNDE(), JETZT()



Die Beispiele zu dieser Funktion finden Sie im Ordner \Ms5-235\Kap06 in der Arbeitsmappe *Datum\_Zeit.xlsx* auf dem Arbeitsblatt *ZEITWERT*.

# Kapitel 7

## Text- und Datenfunktionen

|                          |     |                      |     |
|--------------------------|-----|----------------------|-----|
| ASC()                    | 231 | RECHTS() / RECHTSB() | 246 |
| BAHTTEXT()               | 232 | SÄUBERN()            | 247 |
| CODE()                   | 232 | SUCHEN() / SUCHENB() | 248 |
| DM()                     | 234 | T()                  | 250 |
| ERSETZEN() / ERSETZENB() | 235 | TEIL() / TEILB()     | 250 |
| FEST()                   | 236 | TEXT()               | 252 |
| FINDEN() / FINDENB()     | 237 | UNICODE()            | 253 |
| GLÄTTEN()                | 238 | UNIZEICHEN()         | 253 |
| GROSS()                  | 239 | VERKETTEN()          | 254 |
| GROSS2()                 | 240 | WECHSELN()           | 255 |
| IDENTISCH()              | 241 | WERT()               | 256 |
| KLEIN()                  | 242 | WIEDERHOLEN()        | 257 |
| LÄNGE() / LÄNGEB()       | 243 | ZAHLENWERT()         | 259 |
| LINKS() / LINKSB()       | 244 | ZEICHEN()            | 260 |
| PHONETIC()               | 245 |                      |     |



Text

In diesem Kapitel sind alle Text- und Datenfunktionen beschrieben. Sie finden sämtliche Werkzeuge, um Texte oder Zahlen umzuwandeln, Zeichenketten zu manipulieren und auszuwerten. So benötigen Sie in importierten Listen oft die Möglichkeit, Texte aufzuteilen, miteinander zu kombinieren oder anderweitig zu untersuchen.

**Tabelle 7.1**  
Die Text- und  
Datenfunk-  
tionen in der  
Übersicht

| <b>Funktion</b>          | <b>Beschreibung</b>                                                                               |
|--------------------------|---------------------------------------------------------------------------------------------------|
| ASC()                    | Konvertiert Double-Byte-Text in Single-Byte-Text                                                  |
| BAHTTEXT()               | Konvertiert eine Zahl in Thai-Text mit dem Suffix »Baht«                                          |
| CODE()                   | Gibt die Codezahl des ersten Zeichens eines Texts zurück                                          |
| DM()                     | Konvertiert eine Zahl in Währungstext                                                             |
| ERSETZEN() / ERSETZENB() | Ersetzt eine bestimmte Anzahl Zeichen im Text ab einer bestimmten Position                        |
| FEST()                   | Konvertiert eine Zahl in Text mit einer festen Anzahl Dezimalstellen                              |
| FINDEN() / FINDENB()     | Sucht eine Zeichenfolge innerhalb eines Texts                                                     |
| GLÄTTEN()                | Löscht führende und nachfolgende Leerzeichen in einem Text                                        |
| GROSS()                  | Wandelt einen Text in Großbuchstaben um                                                           |
| GROSS2()                 | Wandelt die ersten Buchstaben aller Wörter in Großbuchstaben um                                   |
| IDENTISCH()              | Prüft, ob zwei Texte identisch sind                                                               |
| KLEIN()                  | Wandelt einen Text in Kleinbuchstaben um                                                          |
| LÄNGE() / LÄNGEB()       | Gibt die Anzahl der Zeichen eines Texts zurück                                                    |
| LINKS() / LINKSB()       | Gibt eine bestimmte Anzahl Zeichen vom Beginn eines Texts zurück                                  |
| PHONETIC()               | Extrahiert die phonetischen (Furigana-) Zeichen aus einer Textzeichenfolge                        |
| RECHTS() / RECHTSB()     | Gibt eine bestimmte Anzahl Zeichen vom Ende eines Texts zurück                                    |
| SÄUBERN()                | Löscht alle nicht druckbaren Zeichen in einem Text                                                |
| SUCHEN() / SUCHENB()     | Sucht eine Zeichenfolge innerhalb eines Texts ohne Berücksichtigung von Groß- und Kleinschreibung |
| T()                      | Wandelt Werte nach bestimmten Regeln in Text um                                                   |
| TEIL() / TEILB()         | Gibt eine bestimmte Anzahl Zeichen ab einem Startzeichen aus einem Text zurück                    |
| TEXT()                   | Konvertiert numerische Werte in Text                                                              |
| UNICODE()                | Gibt die Unicodezahl des ersten Zeichens eines Texts zurück (Version 2013)                        |
| UNIZEICHEN()             | Gibt das Unicodezeichen für eine Codezahl zurück (Version 2013)                                   |
| VERKETTEN()              | Verknüpft mehrere Texte zu einem Text                                                             |
| WECHSELN()               | Ersetzt eine bestimmte Anzahl Zeichen im Text                                                     |
| WERT()                   | Konvertiert einen (Zahl-)Text in einen numerischen Wert                                           |
| WIEDERHOLEN()            | Wiederholt eine Zeichenfolge entsprechend der Vorgabe                                             |
| ZAHLENWERT()             | Konvertiert unabhängig vom Gebietschema des PCs einen (Zahl-)Text in einen numerischen Wert       |
| ZEICHEN()                | Gibt das Zeichen für eine Codezahl zurück                                                         |

# ASC()



ASC()

ASC(*Text*)

Die Funktion wandelt bei Sprachen mit einem Double-Byte-Zeichensatz (DBCS) Zeichen voller Breite (Double-Byte-Zeichen) in Zeichen halber Breite (Single-Byte-Zeichen, SBCS) um.

*Text* (erforderlich) ist der Text oder der Bezug auf eine Zelle, die den Text enthält, den Sie ändern möchten. Enthält dieser Text keinen Buchstaben voller Breite, wird er nicht geändert.

Mit dieser Funktion können Sie Zeichen mit voller Breite (Double-Byte) in Zeichen mit halber Breite (Single-Byte) umwandeln. Da Excel im Unicodemodus arbeitet und auch unter Windows inzwischen die Codepages durch Unicode abgelöst worden sind, hat diese Umwandlungsfunktion eine geringe praktische Relevanz.

Zum Speichern, Lesen und Übertragen von Textdaten verwenden Computer Codepages (CP), in denen jedem Textzeichen ein Code (ein numerischer Wert) zugeordnet ist. Single-Byte-CP enthalten nur 256 verschiedene Zeichen. Da hier nicht die Zeichen aller Sprachen Platz finden, hat jede Sprache ihre eigene CP.

Da sich Single-Byte-CPs nicht für asiatische Sprachen eignen, wurden Double-Byte-CPs für diese Sprachen entwickelt, in denen jedes Zeichen durch Doppelbyte-Werte dargestellt wird. In den ersten 128 Byte der Double-Byte-CPs finden Sie immer die Zeichen der lateinischen Schrift. Ein Nachteil von verschiedenen CPs, egal ob single oder double, ist, dass Sie für die korrekte Wiedergabe wissen müssen, aus welcher CP ein darzustellendes Zeichen kommt. Das macht den Datenaustausch zwischen Computern nicht eben leichter.

Deshalb wurde »Unicode« als Standard für die Zeichencodierung entwickelt. Hier liegt nun ein universeller Zeichensatz für alle bekannten Schriften der Welt vor. Jedes einzelne Zeichen dieser Schriften hat hier einen eindeutigen Code. Da Windows und Office heute Unicode benutzen und nur noch selten unter bestimmten CPs laufen, ist das Umwandlungsthema nach außen verlagert worden, d.h., im Datenaustausch mit Fremdsystemen kann es um die Umwandlung von Unicode in eine CP oder umgekehrt gehen. Ist das Zeichen dann in einer Excel-Arbeitsmappe gelandet, hat die Konvertierung in Unicode bereits stattgefunden.

Weiter gilt:

- ▶ Diese Funktion wird nicht im Funktions-Assistenten von Excel angeboten, nach manueller Eingabe aber unterstützt
- ▶ Wird in *Text* ein leerer Wert übergeben, gibt ASC() auch einen leeren Wert zurück
- ▶ Numerische Werte werden von ASC() als Text zurückgegeben
- ▶ Wird eine Matrix in *Text* übergeben, gibt ASC() den ersten Text aus der Matrix zurück

Aus oben genannten Gründen finden Sie ein Beispiel zu dieser Funktion ausschließlich innerhalb der Beispieldateien zum Buch.

CODE(), ZEICHEN(), UNICODE(), UNIZEICHEN()

Die Funktion finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt ASC.

**Syntax**

**Definition**

**Argumente**

**Hintergrund**

**Praxiseinsatz**

**Siehe auch**



## BAHTTEXT()



BAHTTEXT()

**Syntax** BAHTTEXT(*Zahl*)

**Definition** Diese Funktion wandelt numerische Werte, also Zahlen, in das Zahlwort um, und zwar in Thai-Text. Diesem wird das Suffix »Baht« angefügt.

**Argumente** *Zahl* (erforderlich) ist eine Zahl, die in Text konvertiert werden soll, oder ein Bezug auf eine Zelle, die eine Zahl enthält, oder eine Formel, die zu einer Zahl ausgewertet wird.

**Hintergrund** Das Umwandeln von numerischen Werten in die entsprechenden Zahlwörter wird auf Scheckvordrucken, in Verträgen und an anderen Stellen benötigt, um Zahlenausdrücke abzusichern, Fälschungen zu erschweren oder Missverständnisse zu vermeiden.

In der deutschen Lokalisierung können Sie zwar Zahlenwerte in Thai-Text ausgeben, die Umwandlung in deutsche Zahlwörter suchen Sie in Excel jedoch vergebens. Benötigen Sie auch eher Thai-Text als deutschen Text? Was das Office-Schwesterprogramm Word in Feldausdrücken mit dem Schalter \\* *DOLLARTEXT* anbietet, dürfen Sie vom Zahlenspezialist Excel nicht verlangen. Benötigen Sie diese Funktion, müssen Sie sie programmieren bzw. eine programmierte Lösung im Internet suchen. Mehr zu selbst programmierten Funktionen lesen Sie in Kapitel 5.

**Praxiseinsatz** Als Thailand-Besucher möchten Sie überprüfen, ob die textliche Wertangabe auf Ihrer Hotelgutschrift auch dem angegebenen Zahlenwert entspricht. Wenn Sie den Wert 12 in das Zahlwort in Thai-Text umwandeln möchten, nutzen Sie die Formel

```
=BAHTTEXT(12)
```

Sie ergibt die Textausgabe: ????????????

**Siehe auch** TEXT(), RÖMISCH() (Mathematik & Trigonometrie)



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *BAHTTEXT*.

## CODE()



CODE()

**Syntax** CODE(*Text*)

**Definition** Die Funktion gibt die Codezahl des ersten Zeichens aus *Text* zurück. Die ausgegebene Codezahl entspricht dem Zeichensatz, mit dem Ihr PC arbeitet.

**Argumente** *Text* (erforderlich) ist der Text, für den Sie die Codezahl des ersten Zeichens bestimmen möchten.

**Hintergrund** Für verschiedene Problemstellungen ist es nützlich, den Codewert eines Zeichens ermitteln zu können, z.B. für Sonderzeichen. CODE() ist das Gegenstück zur ZEICHEN()-Funktion, welche für den angegebenen Code das Textzeichen zurückgibt.

Da die Funktion immer nur den Codewert für das erste Zeichen in *Text* zurückgibt, sind eigentlich nur einbuchstabile Argumente sinnvoll. CODE("Excel") gibt ebenso wie CODE("Ergonomie") oder CODE("E") den Wert 69 für das große »E« zurück.



Um den Code für den zweiten oder dritten Buchstaben auszugeben, verwenden Sie die, ebenfalls in diesem Kapitel beschriebene, Funktion TEIL():

=CODE(TEIL("Excel";2;1)) ergibt den Code für das kleine »x«: 120.

Mit Excel 2013 erfährt die Funktion ihre Erweiterung bzw. Ersetzung auf den international standardisierten Unicode, der alle bekannten Sprachzeichen der Welt enthält, durch die Funktion UNICODE().

In einer Liste sollen in einer extra Spalte den Nachnamen Kategorienummern zugeordnet werden. Alle mit »A« beginnenden Nachnamen erhalten die Kategorienummer 1, alle mit »B« die 2 usw. Dabei soll keine Unterscheidung zwischen Groß- und Kleinbuchstaben stattfinden, d.h., »a« wird wie »A« behandelt. Die Formel verwendet eine andere Text-Funktion: GROSS().

=CODE(GROSS("torsten"))-64

|    | H                      | I           | J                | K | L |
|----|------------------------|-------------|------------------|---|---|
| 18 | <b>Kategorienummer</b> |             |                  |   |   |
| 19 |                        |             |                  |   |   |
| 20 | <b>Nr</b>              | <b>Name</b> | <b>Kategorie</b> |   |   |
| 21 | 1                      | torsten     | 20               |   |   |
| 22 | 2                      | Andreas     | 1                |   |   |
| 23 | 3                      | maria       | 13               |   |   |
| 24 | 4                      | Evelyn      | 5                |   |   |
| 25 | 5                      | Dagmar      | 4                |   |   |
| 26 | 6                      | Sara        | 19               |   |   |
| 27 | 7                      | Knut        | 11               |   |   |
| 28 | 8                      | bernd       | 2                |   |   |

**Abbildung 7.1:** Die Berechnung der Kategorienummer für den Anfangsbuchstaben

Die gezeigte Formel ermittelt den Wert 20 (Abbildung 7.1). Das ist die Position des »T« im Alphabet. Die Funktion GROSS() hat aus dem Argument torsten das Argument Torsten gemacht. Da es mit dem Code 65 für das große »A« losgeht, muss der Wert 64 abgezogen werden, um die Positionsnummer im Alphabet zu erhalten. Da Umlaute sehr viel höhere Codewerte haben, erhalten Sie hier höhere Werte, die nicht an die Kategorienummer 26 (Z) anschließen.

ASC(), ZEICHEN(), UNICODE(), UNIZEICHEN()

Die oben gezeigten Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *CODE*. Eine Codetabelle finden Sie außerdem auf dem Arbeitsblatt *ZEICHEN*.

**Praxiseinsatz**

**Siehe auch**



## DM() DOLLAR()

**Syntax** DM(*Zahl*; *Dezimalstellen*)

**Definition** Diese Funktion mit dem inzwischen historischen Namen konvertiert *Zahl* in Text und ordnet ein Währungssymbol zu. Der Name der Funktion und das zugeordnete Währungssymbol hängen von den Ländereinstellungen ab. Bei der Umwandlung werden die Dezimalstellen entsprechend der angegebenen *Dezimalstellen* gerundet. Als Standardformat wird  $\text{\#.\#}0,00 \text{ €}$ ;  $\text{\#.\#}0,00 \text{ €}$  (bei Einstellung des €-Symbols in Ihren Ländereinstellungen) verwendet.

**Argumente** *Zahl* (erforderlich) ist ein numerischer Wert, ein Bezug auf eine Zelle, die eine Zahl enthält, oder eine Formel, deren Ergebnis eine Zahl ist.

*Dezimalstellen* (optional) ist die Anzahl der Ziffern nach dem Dezimalkomma.

- ▶ Ist *Dezimalstellen* negativ, wird *Zahl* links vom Dezimalkomma gerundet
- ▶ Fehlt das Argument *Dezimalstellen*, wird es als 2 angenommen

**Hintergrund** Der Unterschied zwischen dem Formatieren einer Zelle über die Registerkarte **ZAHLEN** im Dialogfeld *Zellen formatieren* und dem direkten Formatieren einer Zahl mit der DM()-Funktion besteht darin, dass DM() den numerischen Wert in Text umwandelt. Eine über das Dialogfeld *Zellen formatieren* formatierte Zahl dagegen ist nach wie vor ein numerischer Wert.

Ungeachtet dieses Unterschieds können Sie die mit der Funktion DM() konvertierten Zahlen in weiteren Formeln verwenden, da Excel Zahlen, die als Textwerte erscheinen, bei einer Berechnung implizit in numerische Werte umwandelt.

Der Unterschied, ob Sie einen Währungsbetrag in Text- oder numerischer Form vorliegen haben, wird z.B. dann wichtig, wenn Sie ein Formular gestalten und der Betrag in einer engen Spalte darzustellen ist, deren Nachbarzelle frei bleibt. Wenn die Zahl sehr groß ist, erscheint bei einer über das Dialogfeld *Zellen formatieren* formatierten Zahl eine #-Reihe, anstelle des Betrags als Zeichen. Dies weist darauf hin, dass die Spaltenbreite für die Darstellung der Zahl nicht ausreicht. Mit DM() konvertierte Beträge würden jedoch in die freie Nachbarzelle hineinragen. Es könnte sein, dass Sie damit im Formulargitter etwas flexibler hinsichtlich der darzustellenden Zahlgrößen sind.

**Praxiseinsatz** In einer Liste, welche einem Word-Seriendruck als Datenquelle dient, soll sichergestellt werden, dass die Preisspalte unverändert in Word ankommt. Deshalb werden die Werte der Preisspalte in einer berechneten Spalte mithilfe der Funktion DM() in den entsprechenden Währungstext umgewandelt. Im Word-Seriendruck wird dann die berechnete Spalte verwendet.

Hier noch einige Beispiele, welche Varianten des Einsatzes der Funktion zeigen:

- ▶ =DM(12,56) ergibt 12,56 €. =DM(38612,60;-1) ergibt 38.610 €
- ▶ =DM(12,56;0) ergibt 13 €. =DM(38612,60;-2) ergibt 38.600 €
- ▶ =DM(12,46) ergibt 12 €. =DM(38612,60;-3) ergibt 39.000 €
- ▶ =DM(PI();3) ergibt 3,142 €. =DM(38612,60;-4) ergibt 40.000 €

**Siehe auch** FEST(), T(), TEXT(), WERT(), ZAHLENWERT()



Die gezeigten und weitere Beispiele finden Sie im Ordner `\Ms5-235\Kap07` in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *DM*.

# ERSETZEN() / ERSETZENB()



## REPLACE(), REPLACEB()

ERSETZEN(*Alter\_Text*; *Erstes\_Zeichen*; *Anzahl\_Zeichen*; *Neuer\_Text*)

**Syntax**

ERSETZENB(*Alter\_Text*; *Erstes\_Zeichen*; *Anzahl\_Bytes*; *Neuer\_Text*)

Die Funktion ersetzt einen Teil der Textzeichenfolge *Alter\_Text*, beginnend an der Position *Erstes\_Zeichen*, mit der Länge *Anzahl\_Zeichen*, durch die neue, in *Neuer\_Text* angegebene, Textzeichenfolge.

**Definition**

ERSETZENB() funktioniert analog für Double-Byte-Zeichen.

*Alter\_Text* (erforderlich) ist die Zeichenfolge, in der Sie einige Zeichen ersetzen möchten.

**Argumente**

*Erstes\_Zeichen* (erforderlich) ist die Position, an der mit dem Ersetzen durch *Neuer\_Text* begonnen werden soll.

*Anzahl\_Zeichen/Anzahl\_Bytes* (erforderlich) gibt an, wie viele der zu *Alter\_Text* gehörenden Zeichen/Bytes ERSETZEN()/ERSETZENB() durch *Neuer\_Text* ersetzen soll.

*Neuer\_Text* (erforderlich) ist die Zeichenfolge, die die durch *Erstes\_Zeichen* und *Anzahl\_Zeichen* bezeichnete Zeichenfolge ersetzen soll.

Diese Tabellenfunktion benötigen Sie zur Manipulation von Texten, und zwar immer dann, wenn Sie in einer Zeichenfolge

**Hintergrund**

- ▶ eine bestimmte,
- ▶ an einer bekannten Stelle beginnende Zeichenfolge mit
- ▶ fester Länge

ersetzen müssen. Ein Anwendungsfall sind z.B. Listen mit Platzhaltern, die Sie durch feste oder errechnete Ausdrücke ersetzen müssen.

Im Gegensatz zur Funktion WECHSELN(), welche ebenfalls innerhalb eines Texts eine bestimmte Zeichenfolge austauscht, verwenden Sie ERSETZEN() immer dann, wenn Sie innerhalb eines Texts eine **an einer bestimmten Position** beginnende Zeichenfolge ersetzen möchten.

1. In einer Liste soll das Wort »Paragraph« durch das Zeichen »§« ersetzt werden. In B25 steht der Text »Paragraph 3, Abs. 4«. Die Formel

**Praxiseinsatz**

=ERSETZEN(B25;SUCHEN("Paragraph";B25);LÄNGE("Paragraph");"§")

ergibt den Text »§ 3, Abs. 4« (siehe Abbildung 7.2).

|    | A | B                               | C | D | E                | F            |
|----|---|---------------------------------|---|---|------------------|--------------|
| 22 |   |                                 |   |   |                  |              |
| 23 |   | 'Paragraph' durch '§' ersetzen: |   |   |                  |              |
| 24 |   | Der Paragraph 14 BGB            | 5 | 9 | §                | Der § 14 BGB |
| 25 |   | Paragraph 3, Abs. 4             | 1 | 9 | §                | § 3, Abs. 4  |
| 26 |   |                                 |   |   | In einer Formel: | § 3, Abs. 4  |
| 27 |   |                                 |   |   |                  |              |

**Abbildung 7.2:** Die Ersetzung des Wortes *Paragraph* durch das Zeichen

2. In einer Liste soll das Wort »Datum« durch das aktuelle Datum ersetzt werden. In A2 steht der Text »Berlin, Datum«. Die Formel

```
=ERSETZEN(A2;SUCHEN("Datum";A2);LÄNGE("Datum");TEXT(HEUTE());"TT.MM.JJJJ")
```

ergibt am 24.12.2013 den Text »Berlin, 24.12.2013«.

**Siehe auch** GLÄTTEN(), SUCHEN(), TEIL(), WECHSELN()



Die Beispiele und weitere Berechnungen finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *ERSETZEN*.

## FEST() FIXED()

**Syntax** FEST(*Zahl*; *Dezimalstellen*; *Keine\_Punkte*)

**Definition** Die Funktion konvertiert den in *Zahl* angegebenen numerischen Wert als Text mit einer festen Anzahl von Nachkommastellen.

**Argumente** *Zahl* (erforderlich) ist der numerische Wert, den Sie in Text umwandeln möchten.

*Dezimalstellen* (optional) ist die anzuzeigende Anzahl der Ziffern rechts vom Dezimal komma.

*Keine\_Punkte* (optional) ist ein Wahrheitswert. Ist *Keine\_Punkte* WAHR, fügt FEST() in einen als Ergebnis gelieferten Text keine Tausendertrennzeichen (Punkte) ein.

**Hintergrund** Wie die Funktion DM() rundet FEST() den angegebenen Wert *Zahl* und wandelt diesen in Text um. Durch die Angabe der *Dezimalstellen* bestimmen Sie, ob gerundet werden soll bzw. wie viele Nachkommastellen der textformatierte Wert haben soll.

In Excel können Zahlen maximal 15 gültige Ziffernstellen haben. Der größte zulässige Wert für *Dezimalstellen* ist 127. Geben Sie im Argument *Dezimalstellen* einen negativen Wert an, werden die Stellen von *Zahl* links vom Dezimalkomma gerundet.

Fehlt das Argument *Dezimalstellen*, wird es als 2 (Nachkommastellen) angenommen. Geben Sie im Argument *Keine\_Punkte* den Wahrheitswert *FALSCH* oder nichts an, enthält der zurückgegebene Text die üblicherweise als Tausendertrennzeichen verwendeten Punkte.

Wie bei DM() liegt der Unterschied zwischen dem Formatieren einer Zahl über das Dialogfeld *Zellen formatieren*, Registerkarte *Zahlen* und dem direkten Formatieren einer Zahl mit der Funktion FEST() darin, dass bei FEST() das Ergebnis als Text vorliegt. Eine über das Dialogfeld *Zellen formatieren* formatierte Zahl dagegen ist nach wie vor ein numerischer Wert.

Ungeachtet dieses Unterschieds können Sie die mit der Funktion FEST() konvertierten Zahlen in weiteren Formeln verwenden, da Excel Zahlen, die als Textwerte erscheinen, bei einer Berechnung implizit in numerische Werte umwandelt.

**Praxiseinsatz** In einer Liste, welche einem Word-Seriendruck als Datenquelle dient, soll sichergestellt werden, dass eine Spalte mit Zahlwerten unverändert in Word ankommt. Deshalb werden die Werte dieser Spalte in einer berechneten Spalte mithilfe der Funktion FEST() in den entsprechenden Zahltext umgewandelt. Im Word-Seriendruck wird dann die berechnete Spalte verwendet.

Hier noch einige Beispiele, welche Varianten des Einsatzes der Funktion zeigen:

- ▶ =FEST(12,56) ergibt 12,56.=FEST(1234,56;-1;1) ergibt 1230
- ▶ =FEST(12,56;0) ergibt 13.=FEST(1234,56;-2;WAHR) ergibt 1200
- ▶ =FEST(12,46) ergibt 12.=FEST(1234,56;-3;0) ergibt 1.000
- ▶ =FEST(PI();3) ergibt 3,142.=FEST(1234,56;-4;FALSCH) ergibt 0

DM(), RUNDEN(), TEXT(), WERT(), ZAHLENWERT()

Die Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *FEST*.

Siehe auch



## FINDEN() / FINDENB()



FINDEN(*Suchtext*; *Text*; *Erstes\_Zeichen*)

FINDENB(*Suchtext*; *Text*; *Erstes\_Byte*)

Die Funktion FINDEN() sucht die in *Suchtext* angegebene Zeichenfolge innerhalb der in *Text* angegebenen Zeichenfolge und gibt die Anfangsposition von *Suchtext* ab dem ersten Zeichen von *Text* zurück, es sei denn, Sie bestimmen mit dem dritten Argument, dass die Suche nicht beim ersten Zeichen beginnen soll.

FINDENB() funktioniert analog für Double-Byte-Zeichen.

*Suchtext* (erforderlich) ist der Text, den Sie finden möchten.

*Text* (erforderlich) ist der Text, in dem gesucht werden soll.

*Erstes\_Zeichen/Erstes\_Byte* (optional) gibt an, bei welchem Zeichen die Suche begonnen werden soll. Das erste zu *Text* gehörende Zeichen/Byte hat die Nummer 1. Fehlt das Argument, wird 1 angenommen.

Bei der Manipulation von Texten ist es häufig notwendig, die Position eines bestimmten Zeichens oder einer Zeichenfolge innerhalb des Texts zu bestimmen, um diesen beispielsweise herauszulösen, zu ersetzen usw. Zu diesem Zweck können Sie die Funktionen FINDEN() oder SUCHEN() einsetzen.

Im Gegensatz zu SUCHEN()/SUCHENB() berücksichtigt die FINDEN()-/FINDENB()-Funktion die Schreibweise (Groß-/Kleinbuchstaben) der beteiligten Zeichenfolgen und lässt keine Platzhalterzeichen zu.

Verwenden Sie *Erstes\_Zeichen*, damit ein Suchvorgang nicht vom ersten Zeichen an beginnt, sondern eine bestimmte Anzahl von Zeichen überspringt. Wenn Sie z.B. in der Zeichenfolge »XLS2003\_FormatZellenSchutz« die Position des ersten »e« im hinteren Teil der Zeichenfolge bestimmen wollen, müssen Sie *Erstes\_Zeichen* mit 9 belegen, sodass der erste Teil der Zeichenfolge nicht durchsucht wird. FINDEN() beginnt mit dem neunten Zeichen und stellt fest, dass das 16. Zeichen mit dem Suchtext übereinstimmt.

Syntax

Definition

Argumente

Hintergrund

Wichtig

Beachten Sie auch die folgenden Regeln bei der Verwendung der Funktion:

- ▶ Ist *Suchtext* eine leere Zeichenfolge (""), ermittelt FINDEN() den Wert 1 bzw. den in *Erstes\_Zeichen* angegebenen Wert
- ▶ *Suchtext* darf **keine Platzhalter** enthalten. Wird *Suchtext* nicht in *Text* gefunden, gibt FINDEN() den Fehlerwert #WERT! zurück.
- ▶ Ist *Erstes\_Zeichen* kleiner oder gleich 0, gibt FINDEN() den Fehlerwert #WERT! zurück
- ▶ Ist *Erstes\_Zeichen* größer als die Zeichenlänge von *Text*, gibt FINDEN() den Fehlerwert #WERT! zurück

**Praxiseinsatz** Für eine Analyse soll in einer Zeichenfolge, die durch einen Unterstrich zweigeteilt ist, der zweite Teil nach einem (großen) »S« durchsucht und die Position seines erstmaligen Vorkommens in der Gesamtzeichenfolge bestimmt werden. Da die Länge des ersten Teils schwanken kann, muss auch die Position des Trennzeichens bestimmt werden, um nicht den ersten Teil zu durchsuchen. Angenommen, die Zeichenfolge »XLS2003\_FormatZellenSchutz« steht in der Zelle C19, dann ermittelt die Formel

```
=FINDEN("S";C19;FINDEN("_";C19)+1)
```

den Wert 21, denn das »S« im zweiten Teil der Zeichenfolge steht an 21. Position des Gesamttexts. Die Formel ohne das dritte Argument

```
=FINDEN("S";C19)
```

würde die Positionsnummer 3 ermitteln, denn das »S« kommt bereits im ersten Teil der Zeichenfolge vor.

| F19 |          |                            |                |
|-----|----------|----------------------------|----------------|
| C   | D        | E                          | F              |
| 3   | Suchtext | Text                       | Erstes_Zeichen |
| 19  | S        | XLS2003_FormatZellenSchutz | 9              |
|     |          |                            | 21             |

**Abbildung 7.3:** Die Zelle D19 enthält die Formel `=FINDEN("_";D19)+1`, ihr Ergebnis wird ins zweite Argument übernommen

**Siehe auch** ERSETZEN(), IDENTISCH(), LÄNGE(), SUCHEN(), TEIL(), WECHSELN



Diese und weitere Beispiele finden Sie im Ordner `\Ms5-235\Kap07` in der Arbeitsmappe `Text_Daten.xlsx` auf dem Arbeitsblatt `FINDEN`.

## GLÄTTEN() TRIM()

**Syntax** GLÄTTEN(*Text*)

**Definition** Diese Funktion löscht alle Leerzeichen in *Text*, die nicht als jeweils einzelne, zwischen Wörtern stehende Trennzeichen dienen.

**Argumente** *Text* (erforderlich) ist der Text, aus dem Sie die führenden und nachfolgenden Leerzeichen entfernen wollen.

Die Funktion GLÄTTEN() können Sie für Texte verwenden, die Sie aus anderen Anwendungsprogrammen übernommen haben und die eventuell unerwünschte Leerzeichen enthalten. Dies ist der typische Anwendungsfall in der Praxis, denn überflüssige Leerzeichen am Beginn oder Ende Ihrer Textdaten stören oft die Weiterverarbeitung, z.B. bei der Verwendung im Seriendruck oder bei der Suche nach dieser Zeichenfolge in einer Liste.

Enthält die Zeichenfolge *Text* weder führende noch nachgestellte Leerzeichen, erhalten Sie keine Fehlermeldung, sondern die identische Zeichenfolge als Ergebnis.

Nach einem Datenimport stellen Sie fest, dass die Daten aus dem Vorsystem eine Anzahl nachfolgender Leerzeichen enthalten. Für die Weiterverarbeitung müssen diese entfernt werden. Der Einsatz der *Ersetzen*-Funktion mit `[Strg]+[H]` ist nicht möglich, da Sie die Leerschritte zwischen den Wörtern nicht entfernen dürfen. Also setzen Sie die Funktion GLÄTTEN() ein, um die überflüssigen Leerzeichen zu entfernen. Für die Weiterverarbeitung benutzen Sie das Ergebnis der Berechnung.

Hier noch einige Beispiele zur Wirkung der Funktion:

| Formel                          | Ergibt ...                                                                                              |
|---------------------------------|---------------------------------------------------------------------------------------------------------|
| =GLÄTTEN("August")              | ... den unveränderten Text <i>August</i>                                                                |
| =GLÄTTEN("August ")             | ... den um das nachfolgende Leerzeichen gekürzten Text <i>August</i>                                    |
| =GLÄTTEN(" August")             | ... den um die zwei vorangestellten Leerzeichen gekürzten Text <i>August</i>                            |
| =GLÄTTEN(" August ")            | ... den um die zwei vorangestellten und die drei nachfolgenden Leerzeichen gekürzten Text <i>August</i> |
| =GLÄTTEN(" August der Starke ") | ... den um das vorangestellte und das nachfolgende Leerzeichen gekürzten Text <i>August der Starke</i>  |

ERSETZEN(), SÄUBERN(), TEIL(), WECHSELN()

Die Beispiele finden Sie im Ordner `\Ms5-235\Kap07` in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *GLÄTTEN*.

## GROSS() UPPER()

GROSS(*Text*)

Die Funktion GROSS() wandelt *Text* komplett in Großbuchstaben um.

*Text* (erforderlich) steht für den Text, der in Großbuchstaben umgewandelt werden soll. *Text* kann sowohl ein Bezug als auch eine Zeichenfolge sein.

Diese Funktion bietet Ihnen eine schnelle und bequeme Möglichkeit, ganze Spalteninhalte in Listen bei Bedarf in kompletter Großbuchstabendarstellung zu erhalten.

Eine andere Anwendung besteht darin, den Suchtext bzw. den zu durchsuchenden Text bei Zeichenvergleichen vor dem Vergleich in Großbuchstaben umzuwandeln, damit der Vergleich unabhängig von einer evtl. variierenden Groß- und Kleinschreibung ist.

Hintergrund

Praxiseinsatz

Siehe auch



Syntax

Definition

Argumente

Hintergrund

Übergeben Sie im Argument *Text* einen numerischen Wert, wird dieser in unformatierten Text (Zahlenformat *Standard*) umgewandelt. Beziehen Sie sich auf diesen Text in numerischen Berechnungen, wandelt Excel diesen implizit in den Zahlenwert um, d.h., Sie können trotz der Textumwandlung mit dem Ergebnis weitere Berechnungen anstellen.

**Praxiseinsatz** Für einen Textvergleich, bei dem es auf die Groß- und Kleinschreibung nicht ankommt, bietet es sich an, den zu überprüfenden Text in Großbuchstaben umzuwandeln. Der Vergleichstext liegt ebenfalls in Großschreibweise vor. Nun ist es egal, in welcher Schreibweise der zu überprüfende Text vorliegt. Sowohl der Vergleich

- ▶ =GROSS("Buchstaben")="BUCHSTABEN" als auch
- ▶ =GROSS("buchstaben")="BUCHSTABEN"

geben das Resultat *WAHR* zurück. Die Abbildung 7.4 zeigt die Anwendung.

|    | B                   | C                 | D                |
|----|---------------------|-------------------|------------------|
| 21 | <i>Text</i>         | <i>Prüftext</i>   | <i>Identisch</i> |
| 22 | Buchstaben          | BUCHSTABEN        | WAHR             |
| 23 | buchstaben          | BUCHSTABEN        | WAHR             |
| 24 | kleinE buchStaben   | KLEINE BUCHSTABEN | WAHR             |
| 25 | Kleinere Buchstaben | KLEINE BUCHSTABEN | FALSCH           |

**Abbildung 7.4:** Der Vergleich ist unabhängig von der Groß-/Kleinschreibung

Hier weitere Beispiele:

- ▶ =GROSS("Excel") ergibt den Text *EXCEL*
- ▶ =GROSS("eXCEL") ergibt den Text *EXCEL*
- ▶ =GROSS("1.232,56") ergibt den Text *1232,56*
- ▶ =GROSS(HEUTE()) ergibt am 24.12.2013 den Text *41632*

**Siehe auch** GROSS2(), KLEIN()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *GROSS*.

## GROSS2() PROPER()

**Syntax** GROSS2(*Text*)

**Definition** GROSS2() wandelt den **ersten** Buchstaben aller Wörter einer Zeichenfolge *Text* in Großbuchstaben um. Alle anderen Buchstaben werden in Kleinbuchstaben umgewandelt.

**Argumente** *Text* (erforderlich) ist in Anführungszeichen eingeschlossener Text, eine Formel, die Text zurückgibt, oder ein Bezug auf eine Zelle, die den Text enthält, den Sie teilweise groß schreiben möchten.

**Hintergrund** Auch heute noch erhalten Sie von Großrechnersystemen beim Datenimport komplett großgeschriebene Listen. In diesen Fällen bietet die Tabellenfunktion GROSS2() die Möglichkeit der Umwandlung in normal geschriebene Textdaten.



Mit dieser Funktion können Sie auch Namen, die einen Bindestrich enthalten, behandeln. Bei der Übergabe von numerischen Werten verhält sich die Funktion GROSS2() genauso wie GROSS().

In einer importierten Liste sollen die komplett groß geschriebenen Wörter in die übliche Schreibweise, d.h. mit einem Großbuchstaben beginnend und ansonsten in Kleinbuchstaben umgewandelt werden. Die folgenden Beispiele zeigen die Wirkung der Funktion.

- ▶ =GROSS2("willy wichtig") ergibt den Text *Willy Wichtig*
- ▶ =GROSS2("Excel") ergibt den unveränderten Text *Excel*
- ▶ =GROSS2("eXCEL") ergibt den Text *Excel*
- ▶ =GROSS2("KNUT-RENÉ BIEDERMANN") ergibt den Text *Knut-René Biedermann*

GROSS(), KLEIN()

Diese Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *GROSS2*.

**Praxiseinsatz**

**Siehe auch**



## IDENTISCH() EXACT()

IDENTISCH(*Text1*; *Text2*)

Mit dieser Funktion können Sie prüfen, ob zwei Zeichenfolgen identisch sind. Als Ergebnis wird einer der Wahrheitswerte *WAHR* oder *FALSCH* ausgegeben.

*Text1* (erforderlich) ist die erste zu vergleichende Zeichenfolge.

*Text2* (erforderlich) ist die zweite zu vergleichende Zeichenfolge.

Als Matrixfunktion eingesetzt, können Sie mit der Funktion IDENTISCH() unter anderem feststellen, ob sich bestimmte Inhalte in einer Liste befinden. Auch können Sie IDENTISCH() verwenden, um Text zu prüfen, der in ein Formular eingegeben wird.

Die Funktion beachtet die Groß- und Kleinschreibung, ignoriert aber Formatierungsunterschiede.

Nach Eingabe einer Zeichenfolge soll ermittelt werden, ob diese in einer Liste vorhanden ist oder nicht. Die Liste erstreckt sich im Bereich *B23:B48*, die Vergleichszeichenfolge steht in der Zelle *D22*. Geben Sie in eine Zelle diese Formel ein:

```
=ODER(IDENTISCH(D22;B23:B48))
```

und schließen Sie die Eingabe mit der Tastenkombination Strg + ↕ + ← ab. Die Formel sieht dann so aus (Abbildung 7.5):

```
{=ODER(IDENTISCH(D22;B23:B51))}
```

Als Matrixformel eingesetzt, bedarf es noch der Funktion ODER(), damit ein Treffer des Einzelwerts in der Liste ermittelt werden kann.

**Syntax**

**Definition**

**Argumente**

**Hintergrund**

**Praxiseinsatz**

|    | B                                          | C         | D      |
|----|--------------------------------------------|-----------|--------|
| 21 | <b>Ist der gesuchte Text in der Liste?</b> |           |        |
| 22 | Liste                                      | Suchtext: | Finden |
| 23 | ASC                                        | Ergebnis: | FALSCH |
| 24 | BAHTTEXT                                   |           |        |
| 25 | CODE                                       | Suchtext: | FINDEN |
| 26 | DM                                         | Ergebnis: | WAHR   |
| 27 | ERSETZEN                                   |           |        |
| 28 | FEST                                       |           |        |
| 29 | FINDEN                                     |           |        |
| 30 | GLÄTTEN                                    |           |        |
| 31 | GROSS                                      |           |        |
| 32 | GROSS2                                     |           |        |
| 33 | IDENTISCH                                  |           |        |
| 34 | KLEIN                                      |           |        |
| 35 | LÄNGE                                      |           |        |
| 36 | LINKS                                      |           |        |
| 37 | PHONETIC                                   |           |        |
| 38 | RECHTS                                     |           |        |
| 39 | SÄUBERN                                    |           |        |
| 40 | SUCHEN                                     |           |        |
| 41 | T                                          |           |        |
| 42 | TEIL                                       |           |        |
| 43 | TEXT                                       |           |        |
| 44 | UNICODE                                    |           |        |
| 45 | UNIZEICHEN                                 |           |        |
| 46 | VERKETTEN                                  |           |        |
| 47 | WECHSELN                                   |           |        |
| 48 | WERT                                       |           |        |
| 49 | WIEDERHOLEN                                |           |        |
| 50 | ZAHLENWERT                                 |           |        |
| 51 | ZEICHEN                                    |           |        |

**Abbildung 7.5:** Im Zusammenspiel mit einer Logikfunktion können Sie bestimmen, ob der Suchtext in der Spalte vorkommt

Weitere Beispiele für den Einsatz der Funktion:

- ▶ =IDENTISCH("Microsoft Excel";"Microsoft excel") ergibt FALSCH
- ▶ =IDENTISCH("Donaudampfschiffahrt";"Donaudampfschiffahrt") ergibt FALSCH
- ▶ =IDENTISCH("Gazelle";"Gazelle") ergibt WAHR
- ▶ =IDENTISCH("Willy Wichtig";"Willy Wichtich") ergibt FALSCH

**Siehe auch** DELTA(), LÄNGE(), SUCHEN()



Die gezeigten Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *IDENTISCH*.

## KLEIN() LOWER()

**Syntax** KLEIN(*Text*)

**Definition** Die Funktion wandelt die in *Text* übergebene Zeichenfolge komplett in Kleinbuchstaben um.

**Argumente** *Text* (erforderlich) ist der Text, den Sie in Kleinbuchstaben umwandeln möchten.

**Hintergrund** Die Funktion KLEIN() ist das Gegenstück zur Funktion GROSS(), welche eine vollständige Großbuchstabenweise erzeugt. Alle Großbuchstaben werden in Kleinbuchstaben umgewandelt, wobei an Zeichen, die keine Buchstaben sind, keine Änderungen vorgenommen werden.

Eine mögliche Anwendung besteht darin, den Suchtext bzw. den zu durchsuchenden Text bei Zeichenvergleichen vor dem Vergleich in Kleinbuchstaben umzuwandeln, damit der Vergleich unabhängig von einer evtl. variierenden Groß- und Kleinschreibung ist.

Übergeben Sie im Argument *Text* einen numerischen Wert, wird dieser in unformatierten Text (Zahlenformat *Standard*) umgewandelt. Beziehen Sie sich auf diesen Text in numerischen Berechnungen, wandelt Excel diesen implizit in den Zahlenwert um, d.h. Sie können trotz der Textumwandlung mit dem Ergebnis weitere Berechnungen anstellen.

Für einen Textvergleich, bei dem es auf die Groß- und Kleinschreibung nicht ankommt, bietet es sich an, den zu überprüfenden Text in Kleinbuchstaben umzuwandeln. Der Vergleichstext liegt ebenfalls in Kleinschreibweise vor. Nun ist es egal, in welcher Schreibweise der zu überprüfende Text vorliegt. Sowohl der Vergleich

- ▶ =KLEIN("Buchstaben")="buchstaben" als auch
- ▶ =KLEIN("BUCHSTABEN")="buchstaben"

geben das Resultat WAHR zurück (siehe Abbildung 7.6).

|    | B                  | C                | D                |
|----|--------------------|------------------|------------------|
| 21 | <i>Text</i>        | <i>Prüftext</i>  | <i>Identisch</i> |
| 22 | Buchstaben         | buchstaben       | WAHR             |
| 23 | BUCHSTABEN         | buchstaben       | WAHR             |
| 24 | großE BuchStaben   | große buchstaben | WAHR             |
| 25 | größere Buchstaben | große buchstaben | FALSCH           |

**Abbildung 7.6:** Der Vergleich ist von Groß-/Kleinschreibweise unabhängig

Hier weitere Beispiele:

- ▶ =KLEIN("Willy Wichtig") ergibt den Text *willy wichtig*
- ▶ =KLEIN("Excel") ergibt den Text *excel*
- ▶ =KLEIN("eXCEL") ergibt den Text *excel*
- ▶ =KLEIN(HEUTE()) ergibt am 24.12.2008 den Text 39806

GROSS(), GROSS2()

Diese und weitere Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *KLEIN*.

**Praxiseinsatz**

**Siehe auch**



## LÄNGE() / LÄNGEB() LEN(), LENB()

LÄNGE(*Text*)

LÄNGEB(*Text*)

LÄNGE() gibt die Anzahl der Zeichen der Zeichenfolge *Text* zurück. LÄNGEB() gibt die Anzahl der Bytes zurück, die die Zeichen in *Text* darstellen und wird für Double-Byte-Zeichen verwendet.

*Text* (erforderlich) ist der Text, dessen Länge Sie ermitteln möchten.

Die Anwendungsfälle dieser Funktion sind vielfältig. Sie benötigen sie im Zusammenhang mit Funktionen wie z.B. TEIL(), LINKS() oder RECHTS(), wenn die Teilzeichenfolge berechnet werden muss. Entsprechende Beispiele finden Sie unter den genannten Funktionen.

**Syntax**

**Definition**

**Argumente**

**Hintergrund**

Verwenden Sie die Funktion auch zur Prüfung, ob eine Eingabe eine bestimmte Länge über- oder unterschritten hat. Weiter können Sie vor einem Datentransport schnell ermitteln, ob und welche Inhalte von Textspalten eine bestimmte Länge überschreiten.

Leerzeichen zählen bei der Längenermittlung als Zeichen. Übergeben Sie numerische Werte im Argument *Text*, wird die Zeichenzahl des unformatierten Zahltexts ermittelt.

**Praxiseinsatz** Bevor eine Excel-Liste zur Weiterverarbeitung exportiert werden soll, kommen Zweifel auf, ob die Spalte mit den Beschreibungen auch die in der Schnittstellenbeschreibung vorgesehene Begrenzung auf maximal 10 Zeichen einhält. Zur Überprüfung wird die Funktion LÄNGE() herangezogen. In der Ergebnisspalte können Überschreitungen mithilfe des Auto-Filters sehr schnell herausgefiltert und korrigiert werden. Hier einige Beispiele:

- ▶ =LÄNGE("CD") ergibt den Wert 2
- ▶ =LÄNGE("Excel 2007") ergibt den Wert 10
- ▶ =LÄNGE("Microsoft") ergibt den Wert 9
- ▶ =LÄNGE("Keine Panik!") ergibt den Wert 12
- ▶ =LÄNGE("") ergibt den Wert 0
- ▶ =LÄNGE(" ") ergibt den Wert 1
- ▶ =LÄNGE("1,345 \$") ergibt den Wert 7
- ▶ =LÄNGE(HEUTE()) ergibt den Wert 5

**Siehe auch** IDENTISCH(), FINDEN(), SUCHEN()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *LÄNGE*.

## LINKS() / LINKSB()



**Syntax** LINKS(*Text*; *Anzahl\_Zeichen*)  
LINKSB(*Text*; *Anzahl\_Bytes*)

**Definition** LINKS() gibt die durch *Anzahl\_Zeichen* bestimmten ersten Zeichen in einer Textzeichenfolge zurück. LINKSB() funktioniert analog, wird für Double-Byte-Zeichen verwendet und gibt die ersten Bytes zurück.

**Argumente** *Text* (erforderlich) ist die Zeichenfolge mit den Zeichen, die Sie von links abschneiden möchten. *Anzahl\_Zeichen* (optional) gibt die Anzahl der Zeichen an, die LINKS() extrahieren soll.

**Hintergrund** Um den ersten Teil mit bestimmter Länge aus einer Zeichenfolge zu ermitteln, steht Ihnen die Funktion LINKS() zur Verfügung. Im Argument *Text* können Sie Texte oder Zahlen angeben. Insbesondere, wenn Zeichenfolgen nach einem bestimmten Schema aufgebaut sind (PLZ Ort, ISBN usw.), kommen die Funktionen LINKS(), RECHTS() und TEIL() zum Extrahieren bestimmter Zeichenfolgen infrage.

Das Argument *Anzahl\_Zeichen* muss größer oder gleich 0 (Null) sein. Ist *Anzahl\_Zeichen* größer als die Gesamtlänge von *Text*, gibt LINKS() den gesamten Text aus. Fehlt das Argument, wird es als 1 angenommen.

In einer Tabellenspalte sind Namen nach dem Schema »Vorname Nachname« enthalten. Aus dieser sollen nun die Vornamen extrahiert werden. Der erste Name steht in der Zelle *H4*. Sie können mit dieser Formel den Vornamen extrahieren:

=LINKS(H4;SUCHEN(" ";H4)-1)

Unter Zuhilfenahme der Funktion SUCHEN() stellen Sie die Position des trennenden Leerzeichens fest. Ziehen Sie davon eine Stelle ab, erhalten Sie die Position des letzten Zeichens des Vornamens und damit die Anzahl der Zeichen, welche von links ermittelt werden sollen.

|   | G | H                  | I          | J |
|---|---|--------------------|------------|---|
| 2 |   | Nur der Vorname    |            |   |
| 3 |   | Text               | Berechnung |   |
| 4 |   | Willy Wichtig      | Willy      |   |
| 5 |   | Roland Fröhlich    | Roland     |   |
| 6 |   | Martin Lorenz      | Martin     |   |
| 7 |   | Walter Kirchberger | Walter     |   |

**Abbildung 7.7:** Durch die Kombination mit SUCHEN() wird die Lösung flexibel

Hier weitere Beispiele zur Veranschaulichung:

- ▶ =LINKS("Donaudampfschiffahrt";5) ergibt den Text *Donau*
- ▶ =LINKS("Gazelle";4) ergibt den Text *Gaze*
- ▶ =LINKS("Willy Wichtig";5) ergibt den Text *Willy*
- ▶ =LINKS("Excel";1) ergibt den Text *E*
- ▶ =LINKS("Excel";2) ergibt den Text *Ex*

RECHTS(), TEIL()

Diese und weitere Beispiele finden Sie im Ordner `Ms5-235\Kap07` in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *LINKS*.

Siehe auch



## PHONETIC()



PHONETIC(*Bezug*)

Die Funktion extrahiert die phonetischen (Furigana-) Zeichen aus einer Textzeichenfolge.

*Bezug* (erforderlich) ist eine Textzeichenfolge oder ein Verweis auf eine einzelne Zelle bzw. einen Zellbereich, die bzw. der eine Furigana-Textzeichenfolge enthält. Ist *Bezug* ein Zellbereich, wird die Furigana-Textzeichenfolge in der linken oberen Zelle des Bereichs zurückgegeben. Ist *Bezug* ein Bereich nicht angrenzender Zellen, wird der Fehler #NV zurückgegeben.

Das moderne japanische Schriftsystem besteht aus

- ▶ den »Kanji«-Zeichen (stammen aus der chinesischen Schrift und bilden als Logogramme meist den Wortstamm),
- ▶ den Silbenschriften »Hiragana« (oft für grammatikalische Formen) und »Katakana« (hauptsächlich für Fremdwörter) und
- ▶ unserem lateinischen Alphabet, das in Japan als »Rōmaji« bezeichnet wird.

Syntax

Definition

Argumente

Hintergrund

Diese Schriftarten haben unterschiedliche spezifische Funktionen und werden in Alltagstexten parallel verwendet.

Bei der Lautschrift »Furigana« handelt es sich um kleine »Hiragana«-Zeichen, die neben einem »Kanji« geschrieben werden, um dessen Aussprache anzugeben. Sie stehen vor allem neben selteneren Kanji, die der Leser vielleicht nicht kennt oder vergessen haben könnte.

Besagte Funktion extrahiert also diese Furigana-Zeichen aus einem japanischen Text. Voraussetzung für die Anwendung dieser Funktion ist die Installation der ostasiatischen Sprachen auf Ihrem PC. Die Funktion hat für den normalen Excel-Anwender in Deutschland keine praktische Bedeutung.

**Praxiseinsatz** Aus oben genannten Gründen wird auf ein Beispiel für diese Funktion verzichtet.

**Siehe auch** Keine

## RECHTS() / RECHTSB()



**Syntax** RECHTS(*Text*; *Anzahl\_Zeichen*)  
RECHTSB(*Text*; *Anzahl\_Bytes*)

**Definition** RECHTS() gibt die letzten, durch *Anzahl\_Zeichen* bestimmten Zeichen eines Texts zurück. RECHTSB() funktioniert unter Angabe von *Anzahl\_Bytes* für Texte, welche aus Double-Byte-Zeichen bestehen.

**Argumente** *Text* (erforderlich) ist die Zeichenfolge mit den Zeichen, die Sie extrahieren möchten.

*Anzahl\_Zeichen* (optional) gibt die Anzahl der Zeichen an, die RECHTS() extrahieren soll.

**Hintergrund** Um den letzten Teil mit bestimmter Länge aus einer Zeichenfolge zu ermitteln, steht Ihnen die Funktion RECHTS() zur Verfügung. Im Argument *Text* können Sie Texte oder Zahlen angeben. Insbesondere, wenn Zeichenfolgen nach einem bestimmten Schema aufgebaut sind (PLZ Ort, ISBN usw.), kommen die Funktionen RECHTS(), LINKS() und TEIL() zum Extrahieren bestimmter Zeichenfolgen infrage.

*Anzahl\_Zeichen* muss größer oder gleich 0 (Null) sein. Gibt das Argument mehr Zeichen an, als der Text lang ist, gibt RECHTS() den gesamten Text zurück. Fehlt das Argument, wird es als 1 angenommen.

**Praxiseinsatz** In einer Tabellenspalte sind Namen nach dem Schema »Vorname Nachname« enthalten. Aus dieser sollen nun die Nachnamen extrahiert werden. Der erste Name steht in der Zelle H4. Sie können mit dieser Formel den Nachnamen extrahieren:

```
=RECHTS(H4;LÄNGE(H4)-SUCHEN(" ";H4))
```

Unter Zuhilfenahme der Funktionen LÄNGE() und SUCHEN() stellen Sie die Anzahl der von rechts zu lesenden Zeichen fest (Abbildung 7.8). Von der mit LÄNGE() ermittelten Gesamtlänge des Texts ziehen Sie die mit SUCHEN() bestimmte Position des trennenden Leerzeichens ab.

|   | G | H                  | I           | J | K |
|---|---|--------------------|-------------|---|---|
| 2 |   | Nur der Nachname   |             |   |   |
| 3 |   | Text               | Berechnung  |   |   |
| 4 |   | Willy Wichtig      | Wichtig     |   |   |
| 5 |   | Roland Fröhlich    | Fröhlich    |   |   |
| 6 |   | Martin Lorenz      | Lorenz      |   |   |
| 7 |   | Walter Kirchberger | Kirchberger |   |   |

**Abbildung 7.8:** Die Zerlegung von Texten ist kein Hexenwerk, sondern einfache Anwendung von Textfunktionen

LINKS(), LINKSB(), TEIL()

Die oben gezeigten Beispiele finden Sie im Ordner `\Ms5-235\Kap07` in der Arbeitsmappe `Text_Daten.xlsx` auf dem Arbeitsblatt `RECHTS`.

## SÄUBERN()



SÄUBERN(*Text*)

Die Funktion löscht alle nicht druckbaren Zeichen aus dem übergebenen Text.

*Text* (erforderlich) ist eine beliebige Tabellenblattinformation, aus der Sie die nicht druckbaren Zeichen entfernen möchten.

Verwenden Sie SÄUBERN() für Texte, die aus anderen Anwendungen importiert wurden und eventuell Zeichen enthalten, die das von Ihnen verwendete Betriebssystem nicht drucken kann. Beispielsweise können Sie SÄUBERN() verwenden, um maschinennahen Code zu entfernen, der sich häufig am Anfang und Ende einer Datendatei befindet und nicht gedruckt werden kann.

Außerdem können Sie mit dieser Funktion störende Zeichen wie z.B. Zeilenumbrüche entfernen und ersparen sich ggf. mehrere Gänge mit dem *Ersetzen*-Befehl.

In der Hauptsache sind es die Zeichen mit einem Code von 1 bis 31, siehe Funktionen CODE() und ZEICHEN(), die gesäubert werden.

Hin und wieder kommt es vor, dass Sie Zeilenumbrüche aus Datenlisten entfernen müssen. Dies ist z.B. zwingend, wenn Sie eine Liste in ein Textformat exportieren müssen. Hier würden die Umbrüche die Liste zerstören, denn sie bedeuten in den Textformaten einen Wechsel zum nächsten Datensatz.

Nehmen Sie an, in der Zelle A2 steht der zweizeilige Text

»August ¶  
der Starke«

=SÄUBERN(A2) ergibt den einzeiligen Text *August der Starke*, welcher ohne Probleme exportiert werden kann.

Ein weiteres Beispiel: Während die Formel

=SÄUBERN("Donau"&ZEICHEN(13)&"dampfschiff"&ZEICHEN(32)&"fahrt")

den Zeilenumbruch, d.h. das ZEICHEN(13), entfernt, bleibt der Leerschritt, d.h. das ZEICHEN(32), im Ergebnistext erhalten: *Donaudampfschiff fährt*.

Siehe auch



Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

**Siehe auch** GLÄTTEN(), CODE(), UNICODE(), UNIZEICHEN(), ZEICHEN()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *SÄUBERN*.

## SUCHEN() / SUCHENB() SEARCH(), SEARCHB()

**Syntax** SUCHEN(*Suchtext*; *Text*; *Erstes\_Zeichen*)  
SUCHENB(*Suchtext*; *Text*; *Erstes\_Zeichen*)

**Definition** Die Funktion SUCHEN() gibt die Positionsnummer des Zeichens zurück, an der der *Suchtext* erstmals gefunden wurde. Die Suche beginnt an der mit *Erstes\_Zeichen* festgelegten Position.

Bei SUCHENB() basiert das Ergebnis auf der Anzahl der Bytes, die die einzelnen Zeichen in *Text* verwenden. Diese Funktion wird für Double-Byte-Zeichenfolgen verwendet.

**Argumente** *Suchtext* (erforderlich) ist die Zeichenfolge, nach der Sie suchen möchten.

*Text* (erforderlich) ist der Text, in dem Sie nach *Suchtext* suchen möchten.

*Erstes\_Zeichen* (optional) ist die Nummer des Zeichens/Bytes in *Text*, ab dem Sie mit der Suche beginnen möchten.

**Hintergrund** Bei der Manipulation von Texten ist es häufig notwendig, die Position eines bestimmten Zeichens oder einer Zeichenfolge innerhalb des Texts zu bestimmen, um diesen beispielsweise herauszulösen, zu ersetzen usw. Zu diesem Zweck können Sie die Funktionen SUCHEN() oder FINDEN() einsetzen.

**Wichtig** Im Gegensatz zu FINDEN()-/FINDENB() berücksichtigt die SUCHEN()/SUCHENB()-Funktion die Schreibweise (Groß-/Kleinbuchstaben) der beteiligten Zeichenfolgen **nicht**, lässt aber die Platzhalterzeichen »\*« und »?« zu.

SUCHEN() gibt immer die Anzahl der Zeichen ab dem Anfang von *Text* zurück. Dabei werden auch die übersprungenen Zeichen gezählt, wenn *Erstes\_Zeichen* größer 1 ist.

Eine als *Suchtext* angegebene Zeichenfolge kann die Platzhalterzeichen »?« und »\*« enthalten. Ein Fragezeichen ersetzt ein Zeichen, ein Sternchen ersetzt eine beliebige Zeichenfolge. Suchen Sie nach einem Fragezeichen oder Sternchen, müssen Sie vor dem zu suchenden Zeichen eine Tilde (~) eingeben.

Verwenden Sie *Erstes\_Zeichen*, damit ein Suchvorgang nicht vom ersten Zeichen an beginnt, sondern eine bestimmte Anzahl von Zeichen überspringt. Wenn Sie z.B. in der Zeichenfolge »XLS2003\_FormatZellenSchutz« die Position des ersten »e« im hinteren Teil der Zeichenfolge bestimmen wollen, müssen Sie *Erstes\_Zeichen* mit 9 belegen, sodass der erste Teil der Zeichenfolge nicht durchsucht wird. SUCHEN() beginnt mit dem neunten Zeichen und stellt fest, dass das 16. Zeichen mit dem Suchtext übereinstimmt.



Beachten Sie auch die folgenden Regeln bei der Verwendung der Funktion:

- ▶ Ist *Suchtext* eine leere Zeichenfolge (""), ermittelt SUCHEN() den Wert 1 bzw. den in *Erstes\_Zeichen* angegebenen Wert
- ▶ Wird *Suchtext* nicht in *Text* gefunden, gibt SUCHEN() den Fehlerwert #WERT! zurück
- ▶ Fehlt das Argument *Erstes\_Zeichen*, wird es als 1 angenommen
- ▶ Ist *Erstes\_Zeichen* kleiner oder gleich 0, gibt SUCHEN() den Fehlerwert #WERT! zurück
- ▶ Ist *Erstes\_Zeichen* größer als die Zeichenlänge von *Text*, gibt SUCHEN() den Fehlerwert #WERT! zurück

Für die Textspalte einer Liste soll untersucht werden, ob der jeweilige Text in eckige Klammern eingerahmte Wörter oder Zahlen enthält. Hier hilft eine berechnete Spalte mit einer Mustersuche, wie in Abbildung 7.9 gezeigt wird. Die dort gezeigte Formel

**Praxiseinsatz**

=SUCHEN(C25;D25)

ergibt 5, denn an Position fünf beginnt das gesuchte Muster [\*] im Text aus D25.

| F25              |          |                                            |                |            |
|------------------|----------|--------------------------------------------|----------------|------------|
| =SUCHEN(C25;D25) |          |                                            |                |            |
|                  | C        | D                                          | E              | F          |
| 3                | Suchtext | Text                                       | Erstes_Zeichen | Berechnung |
| 25               | [*]      | Der [kleine] Hafen lag in der Abendsonne.  |                | 5          |
| 26               | [*]      | Die PLZ-Schreibweise ist [D-]12345.        |                | 26         |
| 27               | [*]      | Hier kommt nur eine öffnende [Klammer vor. |                | #WERT!     |
| 28               | [*]      | Hier kommt gar keine Klammer vor.          |                | #WERT!     |

**Abbildung 7.9:** Die Mustersuche findet das Muster oder liefert einen Fehler

Weitere Beispiele:

- ▶ =SUCHEN("u";"Donaudampfschiffahrt";1) ergibt den Wert 5
- ▶ =SUCHEN("U";"Donaudampfschiffahrt";1) ergibt den Wert 5
- ▶ =SUCHEN("f";"Donaudampfschiffahrt";1) ergibt den Wert 10
- ▶ =SUCHEN("D";"Donaudampfschiffahrt";1) ergibt den Wert 1
- ▶ =SUCHEN("d";"Donaudampfschiffahrt";1) ergibt den Wert 1
- ▶ =SUCHEN("d";"Donaudampfschiffahrt";2) ergibt den Wert 6

ERSETZEN(), ERSETZENB(), FINDEN(), FINDENB(), TEIL(), TEILB(), WECHSELN()

**Siehe auch**

Diese und weitere Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *SUCHEN*.



## T() T0

**Syntax** =T(*Wert*)

**Definition** Diese Funktion T() wandelt das Argument *Wert* nach bestimmten Regeln in Text um.

**Argumente** *Wert* (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), den Sie in Text umwandeln möchten.

**Hintergrund** Die Funktion T() befindet sich aus Kompatibilitätsgründen mit anderen Tabellenkalkulationsprogrammen in der Liste der Excel-Funktionen. In der Regel ist es nicht notwendig, Argumente in Text umzuwandeln, da dies automatisch geschieht.

Wollen Sie die Funktion benutzen, beachten Sie, dass aus Zahlen nicht etwa der Text der Ziffern entsteht, sondern eine leere Zeichenfolge. Dasselbe geschieht mit Wahrheitswerten. Fehlerwerte werden in Fehlerwerte überführt, nicht in deren Text-Form.

**Praxiseinsatz** Aus den gegebenen Informationen heraus soll auf ein Beispiel verzichtet werden.

**Siehe auch** FEST(), DM(), TEXT(), N() Informationsfunktionen, WERT(), ZELLE()



Einige Berechnungsbeispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *T*.

## TEIL() / TEILB() MID(), MIDB()

**Syntax** TEIL(*Text*; *Erstes\_Zeichen*; *Anzahl\_Zeichen*)  
TEILB(*Text*; *Erstes\_Zeichen*; *Anzahl\_Byte*)

**Definition** Die Funktion TEIL() gibt eine Teilzeichenfolge von *Text* zurück. Diese wird durch die in *Anzahl\_Zeichen* angegebene Anzahl von Zeichen und die in *Erstes\_Zeichen* angegebene Startposition bestimmt. TEILB() funktioniert analog für Double-Byte-Zeichenfolgen.

**Argumente** *Text* (erforderlich) ist die Zeichenfolge mit den Zeichen, die Sie extrahieren möchten.

*Erstes\_Zeichen* (erforderlich) ist die Position des ersten Zeichens, ab dem Sie Text extrahieren möchten. Für das erste Zeichen von *Text* gilt, dass *Erstes\_Zeichen* den Wert 1 hat usw.

*Anzahl\_Zeichen/Anzahl\_Byte* (erforderlich) gibt die Anzahl der Zeichen/Bytes an, die TEIL()/TEILB() aus *Text* zurückgeben soll.

**Hintergrund** Um einen Teil mit bestimmter Länge aus einer Zeichenfolge zu ermitteln, steht Ihnen die Funktion TEIL() zur Verfügung. Im Argument *Text* können Sie Texte oder Zahlen angeben. Insbesondere, wenn Zeichenfolgen nach einem bestimmten Schema aufgebaut sind (PLZ Ort, ISBN usw.), kommen die Funktionen TEIL(), LINKS() und RECHTS() zum Extrahieren bestimmter Zeichenfolgen infrage.

Beachten Sie diese Hinweise für die Anwendung der Funktion:

- ▶ Ist *Erstes\_Zeichen* größer als die Gesamtlänge von *Text*, gibt TEIL() die Zeichenfolge "" (leere Zeichenfolge) zurück
- ▶ Ist *Erstes\_Zeichen* kleiner als die Gesamtlänge von *Text*, doch überschreiten *Erstes\_Zeichen* plus *Anzahl\_Zeichen* die Gesamtlänge von *Text*, gibt TEIL() alle bis zum Ende von *Text* gehörenden Zeichen zurück
- ▶ Ist *Erstes\_Zeichen* kleiner 1, gibt TEIL() den Fehlerwert #WERT! zurück
- ▶ Ist *Anzahl\_Zeichen* negativ, gibt TEIL() den Fehlerwert #WERT! zurück

Aus einer Spalte mit Artikelnummern, welche sich aus drei Segmenten wie folgt zusammensetzt, wird das mittlere Segment benötigt, also die Produktgruppe benötigt.

- ▶ 23 – Produkthauptgruppe, zweistellig
- ▶ 015 – Produktgruppe, dreistellig
- ▶ 11290 – Produktnummer, fünfstellig

Zusammengesetzt lautet die Artikelnummer: 2301511290. Da bekannt ist, dass die gesuchte Nummer vom dritten bis fünften Zeichen reicht, werden diese Angaben in die TEIL()-Funktion eingesetzt, um die Produktgruppennummer zu ermitteln (Abbildung 7.10). Die Formel =TEIL("2301511290";3;3) ergibt die Produktgruppennummer 015.

|   | B                    | C              | D              | E          |
|---|----------------------|----------------|----------------|------------|
|   | Text                 | Erstes_Zeichen | Anzahl_Zeichen | Berechnung |
| 3 |                      |                |                |            |
| 4 | 2301511290           | 3              | 3              | 015        |
| 5 | 0811700678           | 3              | 3              | 117        |
| 6 | 1200301940           | 3              | 3              | 003        |
| 7 | Donaudampfschiffahrt | 6              | 5              | dampf      |
| 8 | Gazelle              | 1              | 4              | Gaze       |
| 9 | Gazelle              | 4              | 4              | elle       |

**Abbildung 7.10:** Die Ermittlung eines Teils der Artikelnummer

Zur Veranschaulichung hier noch einige Beispiele:

- ▶ =TEIL("Donaudampfschiffahrt";6;5) ergibt den Text *dampf*
- ▶ =TEIL("Gazelle";1;4) ergibt den Text *Gaze*
- ▶ =TEIL("Gazelle";4;4) ergibt den Text *elle*
- ▶ =TEIL("Louis";2;3) ergibt den Text *oui*
- ▶ =TEIL("Excel";1;2) ergibt den Text *Ex*
- ▶ =TEIL("Excel";2;3) ergibt den Text *xce*

CODE(), FINDEN(), LINKS(), RECHTS(), SUCHEN(), UNICODE()

Die Beispiele zur Funktion TEIL() finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *TEIL*.

**Praxiseinsatz**

**Siehe auch**



## TEXT() TEXT()

**Syntax** TEXT(*Wert*; *Textformat*)

**Definition** Die Funktion TEXT() formatiert die im Argument *Wert* übergebene Zahl entsprechend der im Argument *Textformat* übergebenen Formatierungsanweisung und wandelt sie in Text um.

**Argumente** *Wert* (erforderlich) ist ein numerischer Wert, eine Formel, die einen numerischen Wert liefert, oder ein Bezug auf eine Zelle, die einen numerischen Wert enthält.

*Textformat* (erforderlich) ist ein als Zeichenfolge angegebenes Zahlenformat im Dialogfeld *Zellen formatieren* auf der Registerkarte *Zahlen* im Feld *Kategorie*.

**Hintergrund** Im Zusammenhang mit dem Verketteten von Text, wo statische Texte mit Berechnung verknüpft werden, um dynamische Formulierungen zu erhalten, und in anderen Fällen benötigen Sie die Möglichkeit, numerische Werte in Text zu konvertieren. Die Funktion TEXT() bietet Ihnen diese Möglichkeit einschließlich der Formatierung mit den aus dem Dialogfeld *Zellen formatieren* (*Format/Zellen*) bekannten Zahlenformatanweisungen.

Bei den Formatanweisungen im Argument *Textformat* können Sie auch benutzdefinierte Formate übergeben. Es gibt allerdings einige Einschränkungen:

- ▶ Die Formatanweisung darf kein Sternchen (\*) enthalten
- ▶ Das Zahlenformat *Standard* darf nicht angegeben werden
- ▶ Die Farbanweisungen, z.B. »[Rot]« für negative Zahlen, haben keine Wirkung

Wie bei den Funktionen DM() und FEST() liegt der Unterschied zwischen dem Formatieren einer Zahl über das Dialogfeld *Zellen formatieren*, Registerkarte *ZAHLEN* und dem direkten Formatieren einer Zahl mit der Funktion TEXT() darin, dass bei TEXT() das Ergebnis als Text vorliegt. Eine über über das Dialogfeld *Zellen formatieren* formatierte Zahl dagegen ist nach wie vor ein numerischer Wert.

Ungeachtet dieses Unterschieds können Sie die mit der Funktion TEXT() konvertierten Zahlen in weiteren Formeln verwenden, da Excel Zahlen, die als Textwerte erscheinen, bei einer Berechnung implizit in numerische Werte umwandelt.

**Praxiseinsatz** In einem Rechnungsformular soll der Satz mit dem Zahlungsziel dynamisch sein, d.h. auf das aktuelle Datum sind 14 Tage zu addieren und als Zahlungsziel auszuweisen. Die Formel

=VERKETTEN("Zahlen Sie den Betrag bitte bis zum ";TEXT(HEUTE()+14;"TT.MM.JJJJ");".")

ergibt am 1.12.2013 den Text *Zahlen Sie den Betrag bitte bis zum 15.12.2013*. Ohne die TEXT()-Funktion würde die Textverkettung

=VERKETTEN("Zahlen Sie den Betrag bitte bis zum ";HEUTE()+14;".")

auch funktionieren, allerdings könnte Ihr Kunde mit dem Zahlungsziel *Zahlen Sie den Betrag bitte bis zum 41623*. vermutlich nichts anfangen.

In einem anderen Fall soll das aktuelle Datum als formulierter Satz erscheinen. Die Formel

="Heute ist "&TEXT(HEUTE());"TTTT"&",& der "&TEXT(HEUTE());"T. MMM JJJJ"&"."

ergibt am 1.12.2013 den Text *Heute ist Sonntag, der 1. Dezember 2013*.

ASC(), DM(), FEST(), T(), WERT(), ZAHLENWERT()

Die gezeigten Beispiele und weitere Berechnungen finden Sie im Ordner `\Ms5-235\Kap07` in der Arbeitsmappe `Text_Daten.xlsx` auf dem Arbeitsblatt `TEXT`.

Siehe auch



## UNICODE()



UNICODE()

UNICODE(*Text*)

Die Funktion gibt den Unicodewert (Codepoint) des ersten Zeichens aus *Text* zurück. Der ausgegebene Unicodewert entspricht dem internationalen Standard (ISO 10646 Universal Character Set).

*Text* (erforderlich) ist der Text, für den Sie den Unicodewert des ersten Zeichens bestimmen möchten.

Diese Funktion komplettiert bzw. erweitert die bisher schon vorhandene Funktion CODE(). Im Gegensatz zur CODE()-Funktion, die nur die ersten 255 Zeichen (mit dem ASCII-Code von 1 bis 255) kennt, umfasst UNICODE() alle multilingualen Zeichensätze. Für verschiedene Problemstellungen ist es nützlich, den Unicodewert eines Zeichens ermitteln zu können, z.B. für Sonderzeichen. UNICODE() ist das Gegenstück zur UNIZEICHEN()-Funktion, welche für den angegebenen Unicode das entsprechende Textzeichen zurückgibt.

Da die Funktion immer nur den Codewert für das erste Zeichen in *Text* zurückgibt, sind eigentlich nur einbuchstabile Argumente sinnvoll. UNICODE("Excel") gibt ebenso wie UNICODE("Ergonomie") oder UNICODE("E") den Wert 69 für das große »E« zurück.

=UNICODE("?") ergibt den Code für das (kleine) »L« aus dem georgischen Alphabet: 4314.

Zum Thema »Unicode« beachten Sie bitte auch die Hintergrundbeschreibung zur Funktion ASC() zu Beginn dieses Kapitels.

Siehe CODE()-Funktion. Die dort gezeigten Beispiele können auf alle bekannten Zeichensätze für die Sprachen dieser Erde erweitert werden.

ASC(), CODE(), ZEICHEN(), UNIZEICHEN()

Die oben gezeigten Beispiele finden Sie im Ordner `\Ms5-235\Kap07` in der Arbeitsmappe `Text_Daten.xlsx` auf dem Arbeitsblatt `UNICODE`. Eine Unicodetabelle finden Sie außerdem auf dem Arbeitsblatt `UNIZEICHEN`.

Neu in Excel  
2013

Syntax

Definition

Argumente

Hintergrund

Hinweis

Praxiseinsatz

Siehe auch



## UNIZEICHEN()



UNICHAR()

UNIZEICHEN(*Zahl*)

Die Funktion UNIZEICHEN() gibt das Unicodezeichen zurück, dessen Wert der angegebenen *Zahl* entspricht.

*Zahl* (erforderlich) ist eine Zahl (Codepoint) entsprechend dem internationalen Standard »Unicode«, die das von Ihnen gewünschte Zeichen angibt.

Neu in Excel  
2013

Syntax

Definition

Argumente

**Hintergrund** Verwenden Sie UNIZEICHEN(), um ggf. vorhandene Codierungen, die Sie aus Dateien erhalten, die auf Computern anderen Typs erstellt wurden, in Zeichen umzuwandeln. Ebenso können Sie die Funktion verwenden, um nicht druckbare Zeichen wie z.B. den Zeilenumbruch in berechnete Texte einzusetzen. Nicht zuletzt können Sie auch in Ihrer Standardschriftart fremdsprachige Begriffe erzeugen.

Die Funktion ist sozusagen das Gegenstück zur Funktion UNICODE(), welche den Unicodewert (Codepoint) für ein Zeichen ermittelt. Hier sind alle bekannten Zeichen der Sprachen dieser Erde eingeschlossen.

**Hinweis** Zum Thema »Unicode« beachten Sie bitte auch die Hintergrundbeschreibung zur Funktion ASC() zu Beginn dieses Kapitels.

**Praxiseinsatz** Siehe Funktion ZEICHEN(). Die dort erläuterten Beispiele funktionieren grundsätzlich auch mit UNIZEICHEN(). Mit der folgenden Formel können Sie den Namen der Hauptstadt Georgiens in Originalsprache und -schrift im Arbeitsblatt darstellen (Tbilisi oder ??????):

```
=VERKETTEN(UNIZEICHEN(4311);UNIZEICHEN(4305);UNIZEICHEN(4312);UNIZEICHEN(4314);UNIZEICHEN(4312);UNIZEICHEN(4321);UNIZEICHEN(4312))
```

**Siehe auch** CODE(), UNICODE(), ZEICHEN()



Die obigen Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *UNIZEICHEN*.

## VERKETTEN() CONCATENATE()

**Syntax** VERKETTEN(*Text1*; *Text2*; ...)

**Definition** Die Funktion VERKETTEN() verknüpft mehrere Textzeichenfolgen zu einer Gesamtzeichenfolge.

**Argumente** *Text1*; *Text2*; ... sind 1 bis 30 Argumente, die Zeichenfolgen (Texte) angeben, welche zu einer Gesamtzeichenfolge verbunden werden sollen. Als Argumente dürfen Zeichenfolgen (Texte), Zahlen und Bezüge auf einzelne Zellen verwendet werden.

**Hintergrund** In den Beispielen zur Funktion TEXT() finden Sie bereits einen Anwendungsfall für diese Funktion. Mit ihrer Hilfe können Sie dynamische Formulierungen, die sich aus statischen und berechneten Textteilen zusammensetzen, bilden.

Beachten Sie bitte, dass die Argumente ohne Trennzeichen (wie in Sätzen üblicherweise Leerzeichen oder Komma) zusammengefügt werden. Schließen Sie diese Trennzeichen in die Argumente ein oder setzen Sie sie als eigenständiges Argument ein. Im Falle des Leerzeichens notieren Sie entweder einen Leerschritt zwischen den Anführungsstrichen ("" ) oder Sie nutzen die Funktion ZEICHEN(32). Mit dem Codewert 32 gibt die ZEICHEN()-Funktion ein Leerzeichen zurück.

Anstelle der Funktion VERKETTEN() können Sie auch den Operator »&« verwenden, um als Zeichenfolgen vorliegende Elemente miteinander zu verbinden.

Wenn in *B22* der Name »Maier«, in *C22* der Vorname »Ruprecht« und in *D22* das Geburtsdatum »6.12.1974« steht, bildet die Formel

```
=VERKETTEN(C22;" ";B22;" wurde am ";TEXT(D22;"TT.MM.JJJJ");" geboren. Das war ein ";TEXT(D22;"TTTT");". ")
```

den Satz

*Ruprecht Maier wurde am 06.12.1974 geboren. Das war ein Freitag.*

|    | B                                                                | C              | D                   | E | F |
|----|------------------------------------------------------------------|----------------|---------------------|---|---|
| 20 |                                                                  |                |                     |   |   |
| 21 | <b>Name</b>                                                      | <b>Vorname</b> | <b>Geburtsdatum</b> |   |   |
| 22 | Maier                                                            | Ruprecht       | 06.12.1974          |   |   |
| 23 |                                                                  |                |                     |   |   |
| 24 | Ruprecht Maier wurde am 06.12.1974 geboren. Das war ein Freitag. |                |                     |   |   |
| 25 |                                                                  |                |                     |   |   |

**Abbildung 7.11:** Die Verkettung von Texten und Berechnungen ergibt einen Satz

ERSETZEN(), WECHSELN(), WIEDERHOLEN()

Dieses Beispiel und weitere finden Sie im Ordner `Ms5-235\Kap07` in der Arbeitsmappe `Text_Daten.xlsx` auf dem Arbeitsblatt `VERKETTEN`.

Siehe auch



## WECHSELN() SUBSTITUTE()

WECHSELN(*Text*;*Alter\_Text*;*Neuer\_Text*;*Ntes\_Auftreten*)

Die Funktion WECHSELN() ersetzt bestimmte Zeichen oder Zeichenfolgen in einem Text durch neue Zeichen oder Zeichenfolgen.

*Text* (erforderlich) ist der in Anführungszeichen gesetzte Text oder Bezug auf eine Zelle, die den Text enthält, in dem Zeichen ausgetauscht werden sollen.

*Alter\_Text* (erforderlich) ist die Zeichenfolge, die Sie ersetzen möchten.

*Neuer\_Text* (erforderlich) ist die Zeichenfolge, mit der Sie *Alter\_Text* ersetzen möchten.

*Ntes\_Auftreten* (optional) gibt an, an welchen Stellen *Alter\_Text* durch *Neuer\_Text* ersetzt werden soll. Wenn Sie *Ntes\_Auftreten* angeben, wird nur dieses Vorkommen von *Alter\_Text* ersetzt; andernfalls wird *Alter\_Text* an jeder Stelle, an der er in *Text* vorkommt, durch *Neuer\_Text* ersetzt.

Diese Tabellenfunktion benötigen Sie zur Manipulation von Texten, und zwar immer dann, wenn Sie in einem Text

- ▶ eine bestimmte Zeichenfolge mit
- ▶ fester Länge

einmalig oder mehrfach ersetzen müssen. Ein Anwendungsfall sind z.B. Listen mit Platzhaltern, die Sie durch feste oder errechnete Ausdrücke ersetzen müssen.

Syntax

Definition

Argumente

Hintergrund

WECHSELN() können Sie immer dann verwenden, wenn Sie innerhalb eines Texts eine bestimmte Zeichenfolge austauschen möchten. ERSETZEN() sollten Sie immer dann verwenden, wenn Sie innerhalb eines Texts eine an einer bestimmten Position beginnende Zeichenfolge ersetzen möchten.

**Wichtig** WECHSELN() berücksichtigt bei der Suche nach *Alter\_Text* die Groß- und Kleinschreibung.

**Praxeiseinsatz** In einer Liste wollen Sie die deutschen Umlaute ersetzen, d.h. anstatt »ä« die Zeichen »ae«, anstatt »ö« die Zeichen »oe« und anstatt »ü« die Zeichen »ue«. Dies können Sie erreichen, indem Sie die Funktion WECHSELN() dreimal verschachteln:

```
=WECHSELN(WECHSELN(WECHSELN("Dönerverkäuferprüfung";"ä";"ae");"ö";"oe");"ü";"ue")
```

ergibt die *Doenerverkaeuferpruefung*. Hier weitere Beispiele:

- ▶ =WECHSELN("Donaudampfschiffahrt";"Donau";"Elbe") ergibt *Elbedampfschiffahrt*
- ▶ =WECHSELN("Zelle";"l";"t";2) ergibt den Text *Zelte*
- ▶ =WECHSELN("Retter";"t";"ntn";1) ergibt den Text *Rentner*
- ▶ =WECHSELN(WECHSELN("Reckenhose";"R";"H");"h";"r") ergibt *Heckenrose*
- ▶ =WECHSELN(WECHSELN("Wiesenrand";"W";"R");"r";"w") ergibt *Riesenwand*

**Siehe auch** ERSETZEN(), ERSETZENB(), FINDEN(), GLÄTTEN(), SUCHEN()



Diese und weitere Beispiele für den Einsatz und die Wirkung der Funktion finden Sie im Ordner `Ms5-235\Kap07` in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt *WECHSELN*.

## WERT() VALUE()

**Syntax** WERT(*Text*)

**Definition** Die Funktion WERT() wandelt den in *Text* angegebenen Inhalt in eine Zahl um.

**Argumente** *Text* (erforderlich) gibt den in Anführungszeichen eingeschlossenen Text oder einen Bezug auf eine Zelle an, die den Text enthält, den Sie umwandeln möchten.

**Hintergrund** Solange Sie ausschließlich in und mit Excel unterwegs sind, können Sie Zahlen, die aus irgendwelchen Gründen als Text vorliegen, in weiteren Formeln verwenden, da Excel Zahlen, die als Textwerte erscheinen, bei einer Berechnung implizit in numerische Werte umwandelt. Sobald Sie aber aus Drittanwendungen importierte Daten vorliegen haben oder mit Excel-Add-Ins von Drittanbietern arbeiten, die eigene Tabellenblattformeln stellen, kann es Ihnen passieren, dass Zahlen im Textformat nicht wie sonst in Excel verarbeitet werden und Fehler auslösen.

In der Regel ist es also nicht erforderlich, die Funktion WERT() innerhalb einer Formel zu verwenden; die genannten Ausnahmen bestätigen diese Regel. Für die Ausnahmen ist es gut, dass die Funktion aus Gründen der Kompatibilität zu anderen Tabellenkalkulationsprogrammen zur Verfügung steht.



Sie können eine als Text importierte Liste von Werten schnell so umwandeln, dass Excel sie als Werte erkennt und anzeigt (im Standardfall rechtsbündig):

**Tip**

1. Tippen Sie in eine freie Zelle den Wert *1* ein.
2. Kopieren Sie die Zelle mit den Tasten `[Strg] + [C]` in die Zwischenablage.
3. Markieren Sie den Bereich mit den »Text«-Zahlen und wählen Sie auf der Registerkarte *START* im Dropdownmenü zur Schaltfläche *Einfügen* den Befehl *Inhalte einfügen* aus.
4. Im Dialogfeld *Inhalte einfügen* markieren Sie im Abschnitt *Einfügen* die Option *Werte* und im Abschnitt *Vorgang* die Option *Multiplizieren* und klicken dann auf *OK*.

Nun sollten alle Werte im markierten Bereich als Zahlen und nicht mehr als (linksbündiger) Text angezeigt werden. Die Zelle mit der Zahl 1 wird nicht mehr benötigt. Diese können Sie anschließend löschen.

Der Inhalt des Arguments *Text* kann in jedem der Formate vorliegen, die Excel bezüglich konstanter Zahlen sowie Tages- und Uhrzeitangaben unterstützt. Liegt *Text* in keinem dieser Formate vor, gibt die Funktion `WERT()` den Fehlerwert `#WERT!` zurück.

Für die Weiterverarbeitung von importierten Zahlen aus Textdateien muss sichergestellt werden, dass die importierte Zahl tatsächlich als numerischer Wert übergeben wird. Hierzu setzen Sie die `WERT()`-Funktion ein:

**Praxiseinsatz**

- ▶ `=WERT("1.234")` ergibt den Wert *1234*
- ▶ `=WERT(1234)` ergibt den Wert *1234*
- ▶ `=WERT("09.09.2008")` ergibt den Wert *39700*
- ▶ `=WERT(WAHR)` hingegen ergibt den Fehlerwert `#WERT!`

`DM()`, `FEST()`, `TEXT()`, `N()` Informationsfunktionen, `ZAHLENWERT()`

**Siehe auch**

Die gezeigten und weitere Beispiele finden Sie im Ordner `\Ms5-235\Kap07` in der Arbeitsmappe `Text_Daten.xlsx` auf dem Arbeitsblatt `WERT`.



## WIEDERHOLEN() REPT()

`WIEDERHOLEN(Text;Multiplikator)`

**Syntax**

Die Funktion wiederholt den *Text* so oft wie in *Multiplikator* angegeben.

**Definition**

*Text* (erforderlich) ist das Zeichen bzw. die Zeichenfolge, das bzw. die Sie wiederholen möchten.

**Argumente**

*Multiplikator* (erforderlich) ist eine positive Zahl, die angibt, wie oft *Text* wiederholt werden soll.

**Hintergrund**

Verwenden Sie `WIEDERHOLEN()`, um eine Zeichenfolge in einer bestimmten Häufigkeit in einer Zelle anzugeben. Zusammen mit Funktionen wie `VERKETTEN()` können Sie beliebige Ausdrücke errechnen. Sie können die Funktion auch für diagrammartige Darstellungen im Tabellenblatt »missbrauchen«.

Beachten Sie auch folgende Hinweise im Umgang mit der Funktion:

- ▶ Ist *Multiplikator* mit 0 (Null) belegt, gibt `WIEDERHOLEN()` eine leere Zeichenfolge ("" ) zurück
- ▶ Ist *Multiplikator* keine ganze Zahl, werden deren Nachkommastellen abgeschnitten

- ▶ Ist *Multiplikator* eine negative Zahl, gibt WIEDERHOLEN() den Fehler #WERT! zurück
- ▶ Das Ergebnis der Funktion darf nicht mehr als 32.767 Zeichen enthalten. Andernfalls gibt WIEDERHOLEN() den Fehler #WERT! zurück

**Praxiseinsatz**

In einer Liste stehen im Bereich F6:F16 ein Name und daneben, in G6:G16, steht eine Punktezahl. In der Spalte H6:H16 sollen in einer Zelle der Name linksbündig und die Punkte rechtsbündig stehen. Die Breite soll sich an der größten notwendigen Gesamtbreite orientieren.

Zunächst schaffen Sie eine Hilfsspalte (I), um dort die größte Gesamtlänge für Name und Punkte zu ermitteln. Markieren Sie den Bereich I6:I16 und geben Sie diese Formel ein:

=MAX(LÄNGE(F6:F16)+LÄNGE(G6:G16))+1

Schließen Sie die Eingabe mit der Tastenkombination **Strg** + **↕** + **↵** ab. Die eingegebene Matrixformel (Ihre Formel wird in geschweifte Klammern {...} eingeschlossen) bestimmt in jeder Zeile die maximal benötigte Länge für einen Text aus Namen und Punktezahl. Geben Sie nun die Formel

=F6&WIEDERHOLEN(" ";I6-LÄNGE(F6)-LÄNGE(G6))&G6

in die Zelle H6 ein (Abbildung 7.12) und kopieren diese nach unten. Mithilfe der maximalen Gesamtlänge und der aktuellen Gesamtlänge in der Zeile können Sie den nötigen Zwischenraum an Leerschritten berechnen und mit WIEDERHOLEN() realisieren. Achtung: Das Beispiel ist nur bei Verwendung von nicht proportionalen Schriften (z.B. *Courier*), wo jedes Zeichen dieselbe Breite hat, sinnvoll.

|    | F                                                                           | G             | H            | I    | J      | K |
|----|-----------------------------------------------------------------------------|---------------|--------------|------|--------|---|
| 3  | In einer Spalte soll linksbündig der Name und rechtsbündig die Zahl stehen. |               |              |      |        |   |
| 4  | Die Breite soll sich an der größten notwendigen Gesamtbreite orientieren.   |               |              |      |        |   |
| 5  | <b>Name</b>                                                                 | <b>Punkte</b> | <b>Liste</b> |      | (Hilf) |   |
| 6  | Hansen                                                                      | 37            | Hansen       | 37   | 14     |   |
| 7  | Franke                                                                      | 123           | Franke       | 123  | 14     |   |
| 8  | Heukrodt                                                                    | 3             | Heukrodt     | 3    | 14     |   |
| 9  | Land                                                                        | 32            | Land         | 32   | 14     |   |
| 10 | van Gogh                                                                    | 2345          | van Gogh     | 2345 | 14     |   |
| 11 | Schulz                                                                      | 99            | Schulz       | 99   | 14     |   |
| 12 | Leutheuser                                                                  | 134           | Leutheuser   | 134  | 14     |   |
| 13 | Jarosch                                                                     | 54            | Jarosch      | 54   | 14     |   |
| 14 | Kuhn                                                                        | 4             | Kuhn         | 4    | 14     |   |
| 15 | Bal                                                                         | 1204          | Bal          | 1204 | 14     |   |
| 16 | Kern                                                                        | 22            | Kern         | 22   | 14     |   |

**Abbildung 7.12:** Die flexible Ausfüllung des Zwischenraums mit Leerzeichen

**Siehe auch** VERKETTEN(), ERSETZEN(), WECHSELN()



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap07` in der Arbeitsmappe `Text_Daten.xlsx` auf dem Blatt `WIEDERHOLEN`.

# ZAHLENWERT()



## NUMBERVALUE()

ZAHLENWERT(*Text*; *Dezimaltrennzeichen*; *Gruppentrennzeichen*)

Die Funktion ZAHLENWERT() wandelt den in *Text* angegebenen Inhalt in einen numerischen Wert um, und zwar unabhängig vom eingestellten Gebietsschema des PCs.

*Text* (erforderlich) gibt den in Anführungszeichen eingeschlossenen Text oder einen Bezug auf eine Zelle an, die den Text enthält, den Sie umwandeln möchten.

*Dezimaltrennzeichen* (optional) gibt das Zeichen an, welches in *Text* als Dezimaltrennzeichen interpretiert werden soll.

*Gruppentrennzeichen* (optional) gibt das Zeichen an, welches in *Text* als Zahlengruppentrennzeichen interpretiert werden soll – z.B. Tausendertrennzeichen.

Siehe auch Hintergrundbeschreibung zur Funktion WERT().

Der Inhalt des Arguments *Text* kann in jedem der Formate vorliegen, die Excel bezüglich konstanter Zahlen sowie Tages- und Uhrzeitangaben unterstützt. Liegt *Text* in keinem dieser Formate vor, gibt die Funktion ZAHLENWERT() den Fehlerwert #WERT! zurück. Dieser Fehlerwert wird auch zurückgegeben, wenn die beiden optionalen Argumente *Dezimaltrennzeichen* und *Gruppentrennzeichen* aus ungültigen Zeichen bestehen.

Enthält das Argument *Text* eine leere Zeichenfolge (""), dann wird als Ergebnis eine 0 (Null) zurückgegeben. Alle Leerzeichen im Argument *Text* werden bei der Berechnung ignoriert.

Endet das Argument *Text* mit einem oder mehreren Prozentzeichen (%), gehen diese – wie bei der Verwendung in Formeln – in die Berechnung ein. Mehrere Prozentzeichen wirken additiv. So gibt die Formel =ZAHLENWERT("99%") den Wert 0,0099 zurück, genau wie die Formel =99%.

Fehlen die beiden Argumente *Dezimaltrennzeichen* und *Gruppentrennzeichen*, dann werden bei der Umwandlung von Text die im aktuellen Gebietsschema eingestellten Trennzeichen verwendet, d.h., die Funktion verhält sich wie die Funktion WERT().

Bestehen die Argumente *Dezimaltrennzeichen* und *Gruppentrennzeichen* aus mehreren Zeichen, wird nur das erste Zeichen für die Berechnung verwendet. Weiter gilt für die beiden optionalen Argumente:

- ▶ Das als *Dezimaltrennzeichen* angegebene Zeichen darf nicht mehrmals im *Text* vorkommen. Ansonsten erscheint der Fehlerwert #WERT! als Ergebnis.
- ▶ Das als *Gruppentrennzeichen* angegebene Zeichen darf im *Text* nicht **nach** dem Dezimaltrennzeichen erscheinen. Ansonsten gibt ZAHLENWERT() den Fehlerwert #WERT! zurück.

Für die Weiterverarbeitung von importierten Zahlen aus Textdateien muss sichergestellt werden, dass die importierte Zahl tatsächlich als numerischer Wert übergeben wird. Hierzu setzen Sie die ZAHLENWERT()-Funktion ein:

- ▶ =ZAHLENWERT("1.234") ergibt den Wert 1234
- ▶ =ZAHLENWERT(1234) ergibt den Wert 1234
- ▶ =ZAHLENWERT("1.234"; ".") ergibt den Wert 1,234
- ▶ =ZAHLENWERT("12-6") ergibt im Jahr 2013 den Wert 41437 (das Datum 12.06.2013)
- ▶ =ZAHLENWERT("99%") ergibt den Wert 0,99

Neu in Excel  
2013

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

- ▶ =ZAHLENWERT("99%") ergibt den Wert 0,0099
- ▶ =ZAHLENWERT(WAHR) hingegen ergibt den Fehlerwert #WERT!

**Siehe auch** DM(), FEST(), TEXT(), N() Informationsfunktionen, WERT()



Die gezeigten und weitere Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt ZAHLENWERT.

## ZEICHEN() CHAR()

**Syntax** ZEICHEN(*Zahl*)

**Definition** Die Funktion ZEICHEN() gibt das der in *Zahl* angegebenen Codezahl entsprechende Zeichen zurück.

**Argumente** *Zahl* (erforderlich) ist eine Zahl zwischen 1 und 255, die das von Ihnen gewünschte Zeichen angibt.

**Hintergrund** Verwenden Sie ZEICHEN(), um ggf. vorhandene Codierungen, die Sie aus Dateien erhalten, die auf Computern anderen Typs erstellt wurden, in Zeichen umzuwandeln. Ebenso können Sie die Funktion verwenden, um nicht druckbare Zeichen wie z.B. den Zeilenumbruch in berechnete Texte einzusetzen.

Die Funktion ist quasi das Gegenstück zur Funktion CODE(), welche den Codewert für ein Zeichen ermittelt. Das jeweils für einen bestimmten Code erzeugte Zeichen richtet sich nach dem Zeichensatz, der auf Ihrem Computer verwendet wird.

Mit Excel 2013 erfährt die Funktion ihre Erweiterung bzw. Ersetzung auf den international standardisierten Unicode, der alle bekannten Sprachzeichen der Welt enthält, durch die Funktion UNIZEICHEN().

**Praxiseinsatz** Bei der Verwendung bestimmter Zeichen wie das Anführungszeichen (") in Formeln kann es etwas schwierig bzw. umständlich werden. Wollen Sie z.B. ein Hochkomma (') durch ein Anführungszeichen per Formel ersetzen, sieht die Formel so aus:

```
=WECHSELN(E2;"'";"''")
```

In der Zelle E2 steht der Text *Das 'Hochkomma'*. Die Formel ergibt *Das "Hochkomma"*, d.h., die einfachen Striche wurden korrekt durch die Doppelstriche ersetzt. Während es beim Hochkomma noch ausreichte, es im zweiten Argument einfach in Aufzählungszeichen zu setzen, müssen im dritten Argument vier Gänsefüßchen notiert werden und nicht, wie vielleicht vermutet, drei.

Hier wirkt nachteilig, dass das Anführungszeichen ja eine Funktion als Textkennzeichner hat, nun aber selber den Text stellt. Die Alternative kann so aussehen:

```
=WECHSELN(E2;ZEICHEN(39);ZEICHEN(34))
```

Entscheiden Sie selbst, welche der beiden Formeln für Sie besser einzusetzen ist.

**Siehe auch** ASC(), CODE(), UNICODE(), UNIZEICHEN()



Die obigen Beispiele finden Sie im Ordner \Ms5-235\Kap07 in der Arbeitsmappe *Text\_Daten.xlsx* auf dem Arbeitsblatt ZEICHEN.

# Kapitel 8

## Logische Funktionen

|              |     |
|--------------|-----|
| FALSCH()     | 262 |
| NICHT()      | 264 |
| ODER()       | 265 |
| UND()        | 267 |
| WAHR()       | 268 |
| WENN()       | 269 |
| WENNFEHLER() | 272 |
| WENNV()      | 273 |
| XODER()      | 274 |



In diesem Kapitel lernen Sie die Funktionen der Logik kennen. Im engeren Sinne gehören hierzu NICHT(), UND() und ODER(). Aus verschiedenen Gründen existieren aber auch die kaum verwendeten WAHR() und FALSCH(). Allen gemeinsam ist die Erzeugung und/oder Kombination von Wahrheitswerten. Die WENN()-Funktion ist in der Lage, solche Wahrheitswerte auszuwerten und entsprechende »Verzweigungen« umzusetzen. Dort, wo es zur Auswertung von Argumenten kommt, spielen vor allem Vergleichsoperatoren (=, <, >, <=, >=, <>) eine wichtige Rolle. Unter Excel 2013 sind die Funktionen WENNNV() und XODER() hinzugekommen.

**Tabelle 8.1**  
Die Logik-  
funktionen in  
der Übersicht

| Funktion     | Beschreibung                                                                                                                   |
|--------------|--------------------------------------------------------------------------------------------------------------------------------|
| FALSCH()     | Gibt den Wahrheitswert <i>FALSCH</i> zurück                                                                                    |
| NICHT()      | Kehrt den Wahrheitswert ihres Arguments um                                                                                     |
| ODER()       | Gibt den Wahrheitswert <i>WAHR</i> zurück, wenn mindestens <b>eines</b> der Argumente den Wahrheitswert <i>WAHR</i> hat        |
| UND()        | Gibt den Wahrheitswert <i>WAHR</i> nur dann zurück, wenn <b>alle</b> Argumente vom Wahrheitswert <i>WAHR</i> sind              |
| WAHR()       | Gibt den Wahrheitswert <i>WAHR</i> zurück                                                                                      |
| WENN()       | Gibt eine Wahrheitsprüfung an, die durchgeführt werden soll                                                                    |
| WENNFehler() | Berechnet eine definierte Ausgabe im Fall, dass ein Ausdruck fehlerhaft ist, und gibt ansonsten das Ergebnis des Ausdrucks aus |
| WENNNV()     | Berechnet eine definierte Ausgabe im Fall, dass ein Ausdruck zum Fehler #NV führt                                              |
| XODER()      | Gibt den Wahrheitswert <i>WAHR</i> nur dann zurück, wenn <b>genau eines</b> der Argumente <i>WAHR</i> ist                      |

## FALSCH() FALSE()

**Syntax** FALSCH()

**Definition** Gibt den Wahrheitswert, auch »boolescher Wert« genannt, *FALSCH* zurück.

**Argumente** Diese Funktion verwendet keine Argumente.

**Hintergrund** Der Einsatz dieser Funktion wird eher selten erfolgen, da die Verwendung des Worts »FALSCH« die gleiche Wirkung hat (siehe den folgenden Hinweis): Excel interpretiert das Ergebnis der Funktion als einen Wahrheitswert – hier *FALSCH*. Dabei können Sie noch etwas tiefer eindringen: Die durch numerische Operationen erzwungene Interpretation ist der Zahlenwert 0 (Null), wie Sie am Ergebnis der Formel =3+FALSCH in einer beliebigen Zelle erkennen können.

Jedoch sollten Sie sich diesen Umstand nicht in jedem Falle zunutze machen wollen, da etwa der direkte Vergleich zweier Zellen, von denen eine den Wert *FALSCH* und die andere den Wert 0 (Null) enthält, das Ergebnis *FALSCH* liefert.

Sie können auch das Wort »FALSCH« (die Großschreibung ist zunächst unwichtig; »Falsch« oder »falsch« führen zum gleichen Ergebnis) als Text direkt in ein Arbeitsblatt oder eine Formel eingeben. Excel interpretiert dieses Wort dann als den Wahrheitswert *FALSCH* und auch die Formatierung der Zelle wird im ersten Fall diesem Umstand gerecht. Möchten Sie dies vermeiden,

- ▶ formatieren Sie die Zelle vor der Eingabe als Text (Registerkarte *START/Format/Zellen formatieren* oder kürzer *START/Zahl* und dort *Text* auswählen)
- ▶ oder schreiben ein Leerzeichen vor das Wort »falsch«
- ▶ oder beginnen den Text mit einem Apostroph (').

Beginnen Sie die Eingabe mit einem Gleichheitszeichen, interpretiert Excel die Eingabe von »FALSCH« als Wahrheitswert, auch wenn Sie die runden Klammern weglassen. Das ist anders als etwa bei der Funktion HEUTE(), wo das Weglassen der Klammern zu einem Fehler führt.

Mithilfe der Wahrheitswerte lässt sich sehr schön die Wirkung der beiden logischen Verknüpfungen »Und« bzw. »Oder« erkennen:

- ▶ Die »Oder«-Verknüpfung zweier Wahrheitswerte ist immer *WAHR*, es sei denn, beide Wahrheitswerte sind *FALSCH*
- ▶ Die »Und«-Verknüpfung zweier Wahrheitswerte ist immer *FALSCH*, es sei denn, beide Wahrheitswerte sind *WAHR*

Bei bedingten Formatierungen einer oder mehrerer Zellen können die Bedingungen relativ komplex sein. Zur besseren Übersichtlichkeit empfiehlt sich u.U. der explizite Einsatz von Wahrheitswerten.

Sie haben mithilfe der Funktion HEUTE() in die Zelle B13 Ihres Arbeitsblatts das jeweils aktuelle Datum beim Betrachten der Tabelle eingetragen. Sie möchten nun, dass der Hintergrund der Zelle in der letzten Dekade der durch drei teilbaren Monate (Quartalsende) rot und zu allen anderen Zeitpunkten grün erscheint.

Dazu vergleichen Sie das Ergebnis der Berechnung

`UND(REST(MONAT(B13);3)=0;TAG(B13)>20`

mit dem Wahrheitswert *WAHR*. In dieser Formel liefert

`REST(MONAT(B13);3)=0`

den Wert *WAHR*, wenn es sich um einen Monat am Quartalsende handelt (die Teilung durch den Wert 3 geht auf) und

`TAG(B13)>20`

ist *WAHR*, wenn die Tageszahl größer als 20 ausfällt.

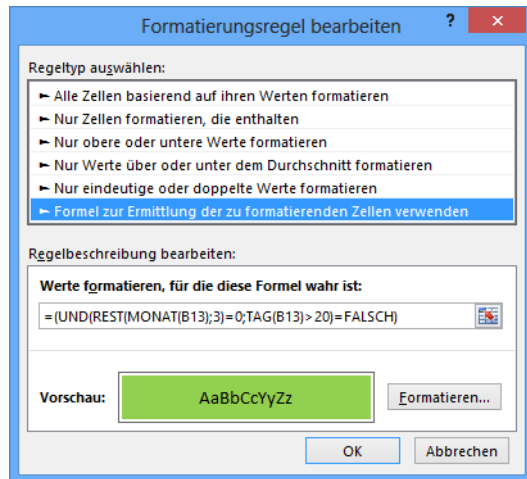
Zur optischen Abgrenzung können Sie nun alle anderen Tage des Jahrs mithilfe des Gegenteils, also durch

`=(UND(REST(MONAT(B13);3)=0;TAG(B13)>20)=FALSCH)`

herausfiltern.

**Hinweis****Praxiseinsatz****Logische Verknüpfungen****Bedingte Formatierungen**

Die gefundenen Formeln tragen Sie durch die Schaltfläche *Neue Regel* unter *START/Formatvorlagen/Bedingte Formatierung* in der Kategorie *Formel zur Ermittlung der zu formatierenden Zellen verwenden* ein. Das dort erscheinende Dialogfeld gleicht dem aus Abbildung 8.1, welches das Ändern bestehender Regeln erlaubt.



**Abbildung 8.1:** Das Dialogfeld zur Bearbeitung bedingter Formatierungen

**Siehe auch** WAHR(), NICHT(), UND(), ODER()



Die Beispiele finden Sie im Ordner `Ms5-235\Kap08` in der Arbeitsmappe *Wahrheitswerte.xlsx* auf dem Arbeitsblatt *Beispiele*. Das Arbeitsblatt *Interpretation und Verhalten* spiegelt die Hintergrund-erläuterungen wider.

## NICHT() NOT()

**Syntax** NICHT(*Wahrheitswert*)

**Definition** Diese Funktion kehrt den Wahrheitswert ihres Arguments um. NICHT() verwenden Sie immer dann, wenn Sie möchten, dass das logische Gegenteil eines Ausdrucks in die weiteren Berechnungen einfließen soll.

**Argumente** *Wahrheitswert* (erforderlich) ist ein Wert oder Ausdruck, der einen der Werte *WAHR* oder *FALSCH* annehmen kann.

**Hintergrund** Wenn das Argument *Wahrheitswert* den Wert *FALSCH* hat, gibt die Funktion NICHT() den Wert *WAHR* zurück. Hat *Wahrheitswert* den Wert *WAHR*, gibt NICHT() den Wert *FALSCH* zurück.

**Praxiseinsatz** Im zweiten Beispiel zur Funktion *FALSCH()* wurde die Bedingung, ob ein Tag in die letzte Dekade eines Monats am Quartalsende fällt oder nicht, mithilfe der Formeln

`= (UND(REST(MONAT(B13);3)=0;TAG(B13)>20)=WAHR)`

bzw.

`= (UND(REST(MONAT(B13);3)=0;TAG(B13)>20)=FALSCH)`



geprüft. Beide Bedingungen lassen sich etwas kürzer formulieren:

```
=(UND(REST(MONAT(B16);3)=0;TAG(B16)>20))
```

bzw.

```
=NICHT(UND(REST(MONAT(B16);3)=0;TAG(B16)>20)).
```

Beachten Sie die Verwendung von NICHT() zum Abfragen des logischen Gegenstücks. Die Kurzformulierungen machen sich zunutze, dass  $=(\text{Ausdruck}=\text{WAHR})$  dasselbe logische Ergebnis wie  $=(\text{Ausdruck})$  liefert. Die letzte der oben verwendeten Kurzformulierungen ließe sich etwas länger auch als

```
=(NICHT(UND(REST(MONAT(B16);3)=0;TAG(B16)>20))=WAHR)
```

schreiben.

ODER(), UND()

Die Beispiele finden Sie im Ordner `\Ms5-235\Kap08` in der Arbeitsmappe *Wahrheitswerte.xlsx* auf dem Arbeitsblatt *Beispiele*.

Siehe auch



## ODER() OR()

ODER(*Wahrheitswert1*; *Wahrheitswert2*; ...)

Diese Funktion gibt den Wahrheitswert *WAHR* zurück, wenn mindestens eines der Argumente den Wahrheitswert *WAHR* hat. Sie gibt nur dann den Wahrheitswert *FALSCH* zurück, wenn alle Argumente gleichzeitig den Wahrheitswert *FALSCH* haben.

Die Argumente *Wahrheitswert1* (erforderlich); *Wahrheitswert2* (optional); ... (es können bis zu 255 sein) enthalten Wahrheitswerte oder Bedingungen, die überprüft und zu Wahrheitswerten ausgewertet werden sollen, sind also jeweils entweder *WAHR* oder *FALSCH*.

Verwenden Sie Argumente, die nicht aus Bezügen stammen und Text sind, gibt ODER() den Fehlerwert *#WERT!* zurück. Alle Zahlen außer 0 (Null) werden als *WAHR* interpretiert. Ein weggelassenes Argument (etwa in  $=\text{ODER}(\text{WAHR};)$ ) wird als *FALSCH* interpretiert.

Verwenden Sie Argumente, die aus Bezügen oder Matrizen stammen, sollten diese in der Regel, wie eben erwähnt, als Wahrheitswerte, also *WAHR* oder *FALSCH*, auswertbar sein.

Es gibt Ausnahmen: Ist mindestens ein Argument (Ausdruck oder Bezug) auswertbar (enthält also selbst einen Wahrheitswert oder liefert diesen nach Auswertung eines Zellbezugs), werden alle anderen Argumente,

- ▶ die Text enthalten oder diesen aus Bezügen liefern,
- ▶ die Bezüge auf leere Zellen enthalten,

ignoriert. Enthalten Argumente Fehlerwerte, weist auch das Ergebnis einen Fehler auf.

Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

Dieses Beispiel soll die Hintergrundaufstellungen unterlegen. Schreiben Sie in die Zellen *B22* und *B23* das Wort »Text«. Dann liefert ODER(*B22*; *B23*) den Fehlerwert *#WERT!*, jedoch ODER(*B22*; *B23*; *FALSCH*) bzw. ODER(*B22*; *B23*; 0) den Wert *FALSCH*. Der Wert *WAHR* wird geliefert, wenn Sie *FALSCH* durch *WAHR* und 0 (Null) durch irgendeine andere Zahl ersetzen.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Wissenswertes  
Details

Zu den gleichen Ergebnissen gelangen Sie, wenn Sie in die Zellen *B25* und *B27* das Wort »Text« eintragen, in *B26* jedoch einen Wahrheitswert oder eine Zahl. Bei der Auswertung von `ODER(B25:B27)` werden die Textzellen (sie könnten auch leer sein) ignoriert.

**Noch ein wenig  
Logik-Grund-  
wissen**

Im Beispiel zur Funktion `NICHT()` wurde die Bedingung, ob ein Tag in die letzte Dekade eines Monats am Quartalsende fällt oder nicht, mithilfe der Formeln

`=(UND(REST(MONAT(B16);3)=0;TAG(B16)>20))`

bzw.

`=NICHT(UND(REST(MONAT(B16);3)=0;TAG(B16)>20)).`

getestet. Unter Beachtung der »Rechenregeln«

- ▶ die logische Umkehrung einer »Oder«-Verknüpfung von Aussagen ist das gleiche wie die »Und«-Verknüpfung der logischen Umkehrungen der Einzelaussagen und
- ▶ die logische Umkehrung einer »Und«-Verknüpfung von Aussagen ist das gleiche wie die »Oder«-Verknüpfung der logischen Gegenteile der Einzelaussagen

lässt sich die letzte der beiden obigen Formeln auch als

`=ODER(NICHT(REST(MONAT(B16);3)=0);NICHT(TAG(B16)>20))`

bzw. kurz

`=ODER(REST(MONAT(B19);3)>0;TAG(B19)<=20)`

schreiben. Die *ODER*-Funktion hat also hier erfolgreich die *UND*-Funktion ersetzt.

**Ergänzung  
zur Hilfe**

Der in der Excel-Hilfe gegebene Hinweis

»Sie können unter Verwendung einer *ODER*-Matrixformel prüfen, ob ein bestimmter Wert innerhalb einer Matrix vorhanden ist. Drücken Sie zum Eingeben einer Matrixformel `[Strg] + [⇧] + [↵]`.«

ist wie folgt zu interpretieren:

Schreiben Sie in die Zelle *B34* den Wert 2 und in die Zellen *C34* bis *C36* die Werte 1, 2 und 3. Gehen Sie nun zu Zelle *D34* und schreiben dort

`=ODER(B34=C34:C36)`

Schließen diese Eingabe allerdings nicht mit der `[↵]`-Taste ab, sondern mit der Tastenkombination `[Strg] + [⇧] + [↵]`, erscheint im Ergebnis *WAHR*. Ändern Sie den Wert von *B34* zu 4, ändert sich das Ergebnis zu *FALSCH*.

Die Matrixformel wird von Excel als

`=ODER(B34=C34;B34=C35;B34=C36)`

interpretiert, damit ist das Ergebnis *WAHR*, wenn eines der Argumente *WAHR* liefert, also der Inhalt von *B34* mit einer der Zahlen in der Spalte *C34:C36* übereinstimmt.

**Tipp**

Die Nutzung der Matrixformel anstelle der ausführlichen Formel, die eben als Erläuterung diente, hebt die Beschränkung der Argumente auf 255 in gewissem Sinne auf. Sie können den Vergleich von *B34* mit (nahezu) beliebig vielen Zahlen erzwingen, da nur `ODER(B34=Bezugszellen)` stehen bleibt, also ein einziges Argument.

## UND(), NICHT()

Die Beispiele finden Sie im Ordner `\Ms5-235\Kap08` in der Arbeitsmappe *Wahrheitswerte.xlsx* auf dem Arbeitsblatt *Beispiele*.

**Siehe auch**



## UND() AND()

UND(*Wahrheitswert1*; *Wahrheitswert2*; ...)

Die Funktion gibt den Wahrheitswert *WAHR* nur dann zurück, wenn alle Argumente vom Wahrheitswert *WAHR* sind. Ist der Wert wenigstens eines Arguments *FALSCH*, gibt diese Funktion den Wert *FALSCH* zurück.

Die Argumente *Wahrheitswert1* (erforderlich); *Wahrheitswert2* (optional); ... entstehen aus bis zu 255 Werten, die direkt eingegeben werden oder aus Bedingungen resultieren, die überprüft werden sollen, und jeweils *WAHR* oder *FALSCH* sein können.

Die Argumente sollten Wahrheitswerte (*WAHR* bzw. *FALSCH*), Zahlen oder aus Matrizen bzw. Bezügen resultierende Werte, die Wahrheitswerte enthalten bzw. aus denen sich Wahrheitswerte ermitteln lassen, sein.

Enthält ein als Matrix oder Bezug angegebenes Argument Text oder leere Zellen, werden diese Werte ignoriert, sofern wenigstens eines der Argumente eine Auswertung zulässt.

Gibt es unter den Argumenten nicht mindestens eins,

- ▶ welches selbst Wahrheitswert ist bzw.
- ▶ als solcher aus einem Bezug resultiert oder
- ▶ durch Auswertung von logischen Ausdrücken entsteht,

gibt UND() den Fehlerwert *#WERT!* zurück.

Enthalten Argumente Fehlerwerte, weist auch das Ergebnis einen Fehlerwert auf.

Dieses Beispiel soll die Hintergrundausführungen illustrieren. Schreiben Sie in die Zellen *H22* und *H23* das Wort »Text«. Dann liefert UND(*H22*; *H23*) den Fehlerwert *#WERT!*, jedoch UND(*H22*; *H23*; *FALSCH*) bzw. UND(*H22*; *H23*; 0) den Wert *FALSCH*. Der Wert *WAHR* wird geliefert, wenn Sie *FALSCH* durch *WAHR* und 0 (Null) durch irgendeine andere Zahl ersetzen.

Zu den gleichen Ergebnissen gelangen Sie, wenn Sie in die Zellen *H25* und *H27* das Wort »Text« eintragen, in *H26* jedoch einen Wahrheitswert oder eine Zahl. Bei der Auswertung von UND(*H25*; *H27*) werden die Textzellen (sie könnten auch leer sein) so lange ignoriert, wie wenigstens eine der Zellen einen Wahrheitswert oder eine Zahl liefert.

Das zweite Beispiel zur Funktion ODER() zeigt einen Zusammenhang zwischen UND(), ODER() und NICHT().

ODER(), NICHT()

Die Beispiele finden Sie im Ordner `\Ms5-235\Kap08` in der Arbeitsmappe *Wahrheitswerte.xlsx* auf dem Arbeitsblatt *Beispiele*.

**Syntax**

**Definition**

**Argumente**

**Hintergrund**

**Praxiseinsatz**

**Wissenswerte  
Details**

**Ein weiteres  
Beispiel**

**Siehe auch**



# WAHR()



# TRUE()

**Syntax** WAHR()

**Definition** Diese Funktion gibt den Wahrheitswert *WAHR* zurück.

**Argumente** Diese Funktion verwendet keine Argumente.

**Hintergrund** Die Verwendung dieser Funktion wird eher selten erfolgen, da die Verwendung des Worts »WAHR« die gleiche Wirkung hat (siehe den folgenden Hinweis): Excel interpretiert das Ergebnis der Funktion als einen Wahrheitswert – hier *WAHR*. Dabei können Sie noch etwas tiefer eindringen: Die durch numerische Operationen erzwungene Interpretation ist der Zahlenwert *1*, wie Sie am Ergebnis der Formel =3+WAHR in einer beliebigen Zelle erkennen können.

Jedoch sollten Sie sich diesen Umstand nicht in jedem Falle zu nutzen machen wollen, da etwa der direkte Vergleich zweier Zellen, von denen eine den Wert *WAHR* und die andere den Wert *1* enthält, das Ergebnis *FALSCH* liefert.

**Hinweis** Sie können auch das Wort »WAHR« (die Großschreibung ist zunächst unwichtig, »Wahr« oder »wahr« führen zum gleichen Ergebnis) als Text direkt in ein Arbeitsblatt oder in eine Formel eingeben. Excel interpretiert dieses Wort dann als den Wahrheitswert *WAHR* und auch die Formatierung der Zelle wird im ersten Fall diesem Umstand gerecht. Möchten Sie dies vermeiden,

- ▶ formatieren Sie die Zelle vor der Eingabe als Text (Registerkarte *START/Format/Zellen formatieren*) oder
- ▶ schreiben ein Leerzeichen vor das Wort »wahr« oder
- ▶ beginnen den Text mit einem Apostroph.

Beginnen Sie die Eingabe mit einem Gleichheitszeichen (=), interpretiert Excel die Eingabe von »WAHR« als Wahrheitswert, auch wenn Sie die runden Klammern weglassen. Das ist anders als etwa bei der Funktion *HEUTE()*, wo das Weglassen der Klammern zu einem Fehler führt.

**Praxiseinsatz** Mithilfe der Wahrheitswerte lässt sich sehr schön die Wirkung der beiden logischen Verknüpfungen »Und« bzw. »Oder« erkennen:

**Etwas Logik-Grundwissen**

- ▶ Die »Oder«-Verknüpfung zweier Wahrheitswerte ist immer *WAHR*, es sei denn, beide Wahrheitswerte sind *FALSCH*
- ▶ Die »Und«-Verknüpfung zweier Wahrheitswerte ist immer *FALSCH*, es sei denn, beide Wahrheitswerte sind *WAHR*

**Ein weiteres Beispiel** Das Beispiel 2 zur Funktion *FALSCH()* gibt einen Einblick in die Verwendung von Wahrheitswerten, wobei Sie statt der Funktionen auch deren Wahrheitswert verwenden können.

**Siehe auch** *FALSCH()*, *NICHT()*, *UND()*, *ODER()*



Die Beispiele finden Sie im Ordner *\Ms5-235\Kap08* in der Arbeitsmappe *Wahrheitswerte.xlsx* auf dem Arbeitsblatt *Beispiele*. Das Arbeitsblatt *Interpretation und Verhalten* spiegelt die Hintergrund-erläuterungen wider.

# WENN()



WENN(*Prüfung*; *Dann\_Wert*; *Sonst\_Wert*)

Gibt durch das zweite beziehungsweise dritte Argument an, wie im Anschluss an eine Wahrheitsprüfung, die auf dem ersten Argument beruht, weiter verfahren (gerechnet, berechnet, angezeigt) werden soll.

*Prüfung* (erforderlich) ist ein beliebiger Wert oder Ausdruck, der einen Wahrheitswert (*WAHR* oder *FALSCH*) liefert.

Dieses Argument kann einen beliebigen Vergleichsrechnungsoperator verwenden:

- ▶ Gleichheitszeichen: Sie können den direkten Vergleich eines Zellinhalts mit einer Zahl ( $A1=7$ ) oder zweier Zellinhalte ( $A1=B1$ ) durchführen
- ▶ Größer als, kleiner als: Auch hier ist der direkte Vergleich eines Zellinhalts mit einer Zahl ( $A1>7$  oder  $B1<8$ ) bzw. zweier Zellinhalte ( $A1>B1$  oder  $C1<D1$ ) möglich
- ▶ Nicht kleiner als, nicht größer als: Hier schwächen sie die Ungleichungen in den vorigen Beispielen ab ( $\geq$  statt  $>$  und  $\leq$  statt  $<$ )
- ▶ Ungleich: Das Ungleichheitszeichen  $<>$  wird wie die anderen Vergleichsoperatoren verwendet

Wie Sie sehen, liefert die Auswertung mittels Vergleichsoperator einen Wahrheitswert. Deshalb können für das Argument *Prüfung* auch Werte Verwendung finden, die durch die logischen Funktionen UND(), ODER() und NICHT() geliefert werden.

*Dann\_Wert* (erforderlich) ist der Wert, der durch die Funktion zurückgegeben wird, wenn das Argument *Prüfung* den Wahrheitswert *WAHR* liefert. Wenn *Prüfung* *WAHR* und *Dann\_Wert* leer ist, gibt die Funktion 0 (Null) zurück. Um den Wahrheitswert *WAHR* zu bekommen, verwenden Sie für dieses Argument den Wahrheitswert *WAHR* oder irgendeinen Ausdruck, der *WAHR* liefert. Der *Dann\_Wert* kann natürlich ein aus anderen Funktionen zusammengesetzter Ausdruck sein.

*Sonst\_Wert* (optional) ist derjenige Wert, der zurückgegeben wird, wenn das Argument *Prüfung* die Auswertung *FALSCH* ergibt. Wenn *Prüfung* *FALSCH* liefert und *Sonst\_Wert* nicht angegeben wird, wird der Wahrheitswert *FALSCH* zurückgegeben. Bleibt der *Sonst\_Wert* leer, erfolgt in diesem Falle die Rückgabe des Werts 0 (Null). Natürlich kann der *Sonst\_Wert* auch ein aus anderen Funktionen zusammengesetzter Ausdruck sein.

Verwenden Sie die WENN()-Funktion, um Tests, die auf Bedingungen beruhen, an Werten und Formeln auszuführen und entsprechend des Testergebnisses weiter zu verfahren.

Die WENN()-Funktion kann mit einleitendem Gleichheitszeichen allein in einer Zellformel stehen oder aber als Argument einer anderen Funktion verwendet werden.

Kompliziertere Bedingungen sind auf jeweils bis zu 64 WENN()-Funktionen als geschachtelte *Dann\_Wert*- bzw. *Sonst\_Wert*-Argumente beschränkt.

Sobald die beiden Argumente *Dann\_Wert* und *Sonst\_Wert* ausgewertet sind, gibt WENN() den Wert zurück, der durch diese Anweisungen berechnet wurde und der durch das Argument *Prüfung* bestimmt wird. Es werden stets beide Argumente ausgewertet, unabhängig davon, ob das durch den Wahrheitswert von *Prüfung* erforderlich ist oder nicht.

**Syntax**

**Definition**

**Argumente**

**Hintergrund**

**Praxiseinsatz** Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

**Ergänzung zur Hilfe** Der in der Hilfe angegebene Hinweis

»Ist eines der an eine WENN-Funktion übergebenen Argumente ein Array, wird bei der Ausführung dieser WENN-Anweisung jedes Element des Arrays ausgewertet.«

erspart Schreibarbeit und lässt sich in wenigstens drei Richtungen interpretieren:

- ▶ Das Argument *Prüfung* besteht aus einem Vergleich mit einer Matrix.

Tragen Sie in die Zellen *B2* bis *B4* Zahlen Ihrer Wahl ein. Nun markieren Sie die benachbarten Zellen *C2* bis *C4*, schreiben

```
=WENN(B2:B4>=0;"positiv";"negativ")
```

und schließen diese Eingabe mit der Tastenkombination `[Strg]+[⇧]+[↵]` ab. Im Ergebnis erhalten Sie die Informationen über das Vorzeichen der links stehenden Zahlen.

- ▶ Die *Dann*- und *Sonst*-Argumente enthalten Bezüge zu Matrixbereichen.

Tragen Sie hierzu in die Zellen *C6* bis *C8* die Worte »rot, grün, blau« und in die benachbarten *D6* bis *D8* die Wörter »schwarz, rot, gold« ein. In die Zelle *B10* schreiben Sie eine Zahl Ihrer Wahl. Nun soll in *C10* bis *C12* etwas stehen, was durch das Vorzeichen der Zahl in *B10* bestimmt ist. Markieren Sie dazu die drei Zellen und schreiben

```
=WENN(B10>0;C6:C8;D6:D8)
```

Beenden Sie die Eingabe mit der Tastenkombination `[Strg]+[⇧]+[↵]`. Durch Änderung von *B10* ändern sich nun gleichzeitig *C10* bis *C12*.

- ▶ Sowohl das Argument *Prüfung* als auch eines der *Dann*- bzw. *Sonst*-Argumente enthält einen Bezug zu einem oder mehreren Matrixbereichen.

Tragen Sie in die Zellen *G2* bis *G5* Zahlen Ihrer Wahl ein und schließen Sie in Zelle *G6* die Eingabe

```
=SUMME(WENN(G2:G5>0;G2:G5;0))
```

mit der Tastenkombination `[Strg]+[⇧]+[↵]` ab. Es werden nur die Zahlen summiert, die größer sind als 0 (Null).

**Hinweis** Hier besteht eine enge Beziehung zur Funktion `SUMMEWENN()`, deren vollständige Argumente im vorliegenden Fall `G2:G5;">0";G2:G5` lauten.

Eine ebensolche Verwandtschaft besteht zwischen `ZÄHLENWENN()` auf der einen und `ANZAHL()` und `WENN()` auf der anderen Seite.

In beiden Fällen liefert die Lösung mit der `WENN()`-Funktion mehr Spielraum hinsichtlich der zu prüfenden Bedingungen, da diese durch Einsatz von `UND()` und `ODER()` beliebig ausgebaut werden können.

**Auf bestimmte Werte reagieren** Dieses Beispiel soll zeigen, wie Sie die `WENN()`-Funktion zur optischen Auswertung von Berechnungen verwenden können. Nehmen Sie dazu die fiktiven Zahlen, die bei einer Investitionsrechnung wie in Abbildung 8.2 zugrunde gelegt werden sollen. Zwei Anschaffungsobjekte in den Varianten 1 und 2 fordern Anschaffungskosten in Höhe von 80 TEURO bzw. 90 TEURO und bringen in den Folgejahren einen Überschuss entsprechend der Tabelle. Für beide Varianten berechnet die Funktion `IKV()` die interne Kapitalverzinsung.

|    | A                    | B             | C             | D |
|----|----------------------|---------------|---------------|---|
| 13 |                      |               |               |   |
| 14 | Investitionsrechnung |               |               |   |
| 15 |                      |               |               |   |
| 16 |                      | Variante 1    | Variante 2    |   |
| 17 | Jahr 0               | - 80.000,00 € | - 90.000,00 € |   |
| 18 | Jahr 1               | 5.500,00 €    | 16.500,00 €   |   |
| 19 | Jahr 2               | 22.000,00 €   | 22.000,00 €   |   |
| 20 | Jahr 3               | 27.500,00 €   | 27.500,00 €   |   |
| 21 | Jahr 4               | 27.500,00 €   | 27.500,00 €   |   |
| 22 | Jahr 5               | 5.500,00 €    | 5.500,00 €    |   |
| 23 | IKV                  | 3,18%         | 3,45%         |   |
| 24 | Bewertung            |               | besser        |   |

Abbildung 8.2: Investitionsrechnung – ein Beispiel

Schreiben Sie nun in Zelle C24 die Formel

```
=WENN(C23>D23;"besser";"
```

und in Zelle D24 die Formel

```
=WENN(D23>C23;"besser";"
```

Dann können Sie mit den Zahlen »spielen« und bekommen optische Unterstützung in der Bewertungszeile.

Sie können die IKV-Zeile ausblenden, falls beim Betrachter kein Bedarf an den Grundlagen der Bewertung besteht.

**Tip**

Sie können die bedingte Formatierung der IKV-Zeile als Alternative zur Bewertungszeile nutzen.

Ein Großwarenhändler bietet für ein bestimmtes Produkt zum Grundpreis von 1,50 € pro Stück Rabatte nach folgender Staffel: 5 % für Stückzahlen ab 10, 10 % für Stückzahlen ab 100 und 20 % für Stückzahlen ab 1.000. Dabei soll der jeweilige Rabatt den gesamten Posten und nicht nur die über den Grenzen liegenden Stückzahlen betreffen (wie das etwa bei der Staffelfinanzierung von gewissen Festgeldanlagen oder Sparguthaben ist).

**Rabatt-  
Rechnung**

Eine mögliche Excel-Lösung zur Berechnung eines Gesamtpreises für eine beliebige Stückzahl nutzt die WENN()-Funktion in einer Formel (die Zelle C29 enthält die Stückzahl):

```
=(1-WENN(C29>=1000;20%;WENN(C29>=100;10%;WENN(C29>=10;5%;0))))*C29*1,5
```

Eine solche Formel ist aber wenig flexibel. Sie sollten bei der Umsetzung unbedingt eine kleine Tabelle mit den Grenzstückzahlen und den Rabatten anlegen und eine Zelle für den Preis pro Stück reservieren.

|    | A               | B    | C       |
|----|-----------------|------|---------|
| 26 |                 |      |         |
| 27 | Rabattstaffel   |      |         |
| 28 |                 |      |         |
| 29 | Stückzahl       |      | 12      |
| 30 | Preis pro Stck. |      | 1,50 €  |
| 31 |                 |      |         |
| 32 |                 | 10   | 5%      |
| 33 |                 | 100  | 10%     |
| 34 |                 | 1000 | 20%     |
| 35 |                 |      |         |
| 36 | Gesamtpreis     |      | 17,10 € |

Abbildung 8.3: Flexibilität durch klare Trennung von Ein- und Ausgabegrößen

Die dann mögliche Formel (siehe Beispieldatei)

$$=(1-WENN(C29>=B34;C34;WENN(C29>=B33;C33;WENN(C29>=B32;C32;0)))*C29*C30$$

ist zwar nicht mehr so schnell einleuchtend, der Blattaufbau erlaubt aber ein flexibles Rechnen für verschiedene Stückzahlen, Preise und Rabattstaffeln.

**Tipp** Sie erkennen unschwer, dass die Schachtelung vieler WENN()-Bedingungen schnell unübersichtlich und deshalb fehleranfällig wird. Im vorliegenden Fall bietet sich auch eine Lösung mittels SVERWEIS() an:

$$=(1-SVERWEIS(C29;B32:C34;2;WAHR))*C29*C30.$$

Dabei wird davon ausgegangen, dass in *B32* bis *B34* die Grenzstückzahlen und in *C32* bis *C34* die Rabatt-Prozente stehen. Damit ist *B32:C34* die in SVERWEIS() verlangte Matrix und 2 die Nummer der Spalte, die die Rückgabewerte enthält. Gesucht wird nach dem Inhalt von *C29* in der ersten Spalte der Matrix. Mit dem Wahrheitswert *WAHR* wird angegeben, dass es nicht auf eine genaue Übereinstimmung bei der Suche ankommt. Somit wird der Wert als gefunden angesehen, der dem gesuchten von allen, die kleiner als der gesuchte sind, am nächsten kommt. Das ist genau der in der Aufgabe verlangte Wert.

**Siehe auch** FALSCH(), NICHT(), ODER(), SVERWEIS(), SUMMEWENN(), UND(), VERWEIS(), WAHR(), WVERWEIS(), ZÄHLENWENN()



Die Beispiele finden Sie im Ordner *\Ms5-235\Kap08* in der Arbeitsmappe *Wahrheitswerte.xlsx* auf dem Arbeitsblatt *WENN*. Das Arbeitsblatt *Interpretation und Verhalten* spiegelt einige der Argumenterläuterungen wider.

Neu in 2007

## WENNFEHLER()



IFERROR ()

**Syntax** WENNFEHLER(*Wert*; *Wert\_falls\_Fehler*)

**Definition** Gibt durch das zweite Argument an, was im Anschluss an eine Fehlerprüfung, die auf dem ersten Argument beruht, angezeigt werden soll.

**Argumente** *Wert* (erforderlich) ist ein beliebiger Wert oder Ausdruck.

*Wert\_falls\_Fehler* (erforderlich) ist der Wert, der durch die Funktion zurückgegeben wird, wenn das Argument *Wert* einen Fehler liefert. Fehler sind hierbei durch ihre Typen #NV, #WERT!, #BEZUG!, #DIV/0!, #ZAHL!, #NAME? oder #NULL! bestimmt. Gibt es keinen Fehler, ist der Wert der Funktion gleich dem des ersten Arguments.

**Hintergrund** Verwenden Sie die WENNFEHLER()-Funktion, um Tests, die auf Bedingungen beruhen, an Werten und Formeln auszuführen und entsprechend des Testergebnisses weiter zu verfahren.

Wenn sich *Wert* oder *Wert\_falls\_Fehler* auf eine leere Zelle bezieht, reagiert die Funktion so, als ob diese die leere Zeichenkette enthält ("").

Wenn *Wert* eine Matrixformel darstellt, gibt WENNFEHLER() für jede Zelle im Bereich von *Wert* eine Matrix von Ergebnissen zurück.

**Praxiseinsatz** Diese Funktion kann als eine Kombination der WENN-Funktion mit bestimmten Fehlerfunktionen interpretiert werden und dadurch zu kürzeren und übersichtlichen Formeln führen.



Sie haben eine Liste mit den Geburtstagen von Personen (oder Artikel- oder Bestellnummern, Adressdetails, Telefonnummern usw.) angefertigt und möchten mithilfe von SVVERWEIS() darauf zugreifen. Die Abbildung 8.4 zeigt ein »Mini-Beispiel«.

|    | A | B              | C | D        | E          | F |
|----|---|----------------|---|----------|------------|---|
| 43 |   |                |   |          |            |   |
| 44 |   | Johann         |   | Hans     | 01.01.1955 |   |
| 45 |   | #NV            |   | Georg    | 10.10.1978 |   |
| 46 |   | nicht gefunden |   | Albrecht | 31.12.1964 |   |
| 47 |   | nicht gefunden |   |          |            |   |

**Abbildung 8.4:** Fehler bei der Suche in Listen erklären

Schreiben Sie in die Zelle *B45*

```
=SVERWEIS(B44;D44:E46;2;FALSCH)
```

(achten Sie auf den letzten Parameter *FALSCH*, der nach einer genauen Übereinstimmung des gesuchten Begriffs forscht), erhalten Sie den Fehlerwert *#NV*, der vor allem beim Ausdrucken der Tabelle störend wirkt<sup>1</sup>.

Im vorliegenden Fall können Sie sich mit der WENN()-Funktion in *B46* statt des einfacheren Eintrags in *B45* behelfen:

```
=WENN(ISTFEHLER(SVERWEIS(B44;D44:E46;2;FALSCH));"nicht gefunden";
SVERWEIS(B44;D44:E46;2;FALSCH))
```

Diese Formel ist etwas umständlich, vor allem auch wegen der eigentlichen Redundanz von SVVERWEIS(B44;D44:E46;2;FALSCH). Kürzer geht es da mit

```
=WENNFEHLER(SVERWEIS(B44;D44:E46;2;FALSCH);"nicht gefunden")
```

FEHLER.TYP(), ISTFEHLER(), ISTFEHL(), ISTNV(), ISTZAHL(), ISTTEXT(), WENNNV()

**Siehe auch**

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap08` in der Arbeitsmappe *Wahrheitswerte.xlsx* auf dem Arbeitsblatt *Beispiele*.



## WENNNV()



WENNNV(*Wert*; *Wert\_bei\_NV*)

Gibt durch das zweite Argument an, was im Anschluss an eine Fehlerprüfung, die auf dem ersten Argument beruht und zum Fehler *#NV* führt, angezeigt werden soll.

*Wert* (erforderlich) ist ein beliebiger Wert oder Ausdruck.

*Wert\_bei\_NV* (erforderlich) ist der Wert, der durch die Funktion zurückgegeben wird, wenn das Argument *Wert* den Fehler *#NV* liefert. Gibt es diesen Fehler nicht, ist der Wert der Funktion gleich dem des ersten Arguments.

**Neu in Excel 2013**

**Syntax**

**Definition**

**Argumente**

<sup>1</sup> In anderen Situationen wollen Sie vielleicht mit den gefundenen Zwischenwerten weiterrechnen und dabei Fehler ignorieren.

- Hintergrund** Diese Funktion verkörpert letztlich einen Spezialfall der WENNFehler()-Funktion.
- Praxiseinsatz** Unter den Annahmen des Beispiels zur WENNFehler()-Funktion liefert im Falle des durch den SVERWEIS() nicht gefundenen Namens die Formel
- ```
=WENNV(SVERWEIS(B44;D44:E46;2;FALSCH);"nicht gefunden")
```
- die gleichen Resultate wie im genannten Beispiel.
- Siehe auch** FEHLER.TYP(), ISTFEHLER(), ISTFEHL(), ISTNV(), ISTZAHL(), ISTTEXT(), WENNFehler()



Das Beispiel finden Sie im Ordner `Ms5-235\Kap08` in der Arbeitsmappe *Wahrheitswerte.xlsx* auf dem Arbeitsblatt *Beispiele*.

Neu in Excel
2013

XODER() XOR()

Syntax XODER(*Wahrheitswert1*; *Wahrheitswert2*; ...)

Definition Die Funktion gibt den Wahrheitswert *WAHR* nur dann zurück, wenn genau eines der Argumente vom Wahrheitswert *WAHR* ist. Anderenfalls, gibt diese Funktion den Wert *FALSCH* zurück.

Argumente Die Argumente *Wahrheitswert1* (erforderlich); *Wahrheitswert2* (optional); ... bestehen aus bis zu 254 Werten, die direkt eingegeben werden oder aus Bedingungen resultieren, die überprüft werden sollen und jeweils *WAHR* oder *FALSCH* sein können.

Hintergrund Die Argumente sollten Wahrheitswerte (*WAHR* bzw. *FALSCH*), Zahlen oder aus Matrizen bzw. Bezügen resultierende Werte, die Wahrheitswerte enthalten bzw. aus denen sich Wahrheitswerte ermitteln lassen, sein.

Enthält ein als Matrix oder Bezug angegebenes Argument Text oder leere Zellen, werden diese Werte ignoriert, sofern wenigstens eines der Argumente eine Auswertung zulässt.

Gibt es unter den Argumenten nicht mindestens ein Argument,

- ▶ welches selbst Wahrheitswert ist bzw.
- ▶ als solcher aus einem Bezug resultiert oder
- ▶ durch Auswertung von logischen Ausdrücken entsteht,

gibt XODER() den Fehlerwert *#WERT!* zurück.

Enthalten Argumente Fehlerwerte, weist auch das Ergebnis einen Fehlerwert auf.

Praxiseinsatz Die logische Interpretation von XODER() liegt unserem Sprachempfinden eigentlich näher als die der Funktion ODER(). Der Satz »Tini oder Tim holen die Oma vom Bahnhof ab.« wird wohl eher als wahr empfunden, wenn nur eines der Kinder hingeht. Gehen beide hin, werden viele die Aussage als falsch empfinden und folgen so der Interpretation von XODER() (ausschließendes Oder).

Nichtsdestotrotz findet die Funktion ODER() in der Mathematik (und damit in vielen Bereichen von Excel) häufiger Anwendung als ihr Gegenstück XODER(). Diese ist eher Spezialanwendungsgebieten wie der Schaltalgebra, der Prüfsummenbildung oder der Verschlüsselung von Nachrichten vorbehalten.

Innerhalb der Begleitdateien zum Buch finden Sie ein Beispiel, in welchem das Wort »Excel« mit einem (einfachen) Schlüssel verschlüsselt und mit diesem auch wieder entschlüsselt wird (symmetrische Verschlüsselung durch One-Time-Pad-Verfahren). Dazu werden die einzelnen Buchstaben in einzelne Zellen geschrieben und mithilfe der Funktion CODE() in ihren Zahlencode gebracht. Dieser wird durch die Funktion HEXINBIN() in eine Zeichenkette aus Nullen und Einsen umgewandelt, aus der wiederum mithilfe der Funktionen WERT() und TEIL() die Bestandteile als Ziffern gewonnen werden.

Diese Ziffern werden nun durch XODER() mit den entsprechenden Positionen des Schlüssels (der im Beispiel kurz gehalten und wiederholt angewendet wird) verbunden. Da das Ergebnis ein Wahrheitswert ist, der wieder zu einer Null oder Eins umgeformt werden muss, wird die XODER()-Funktion mit 1 multipliziert.

Wendet man nun den Schlüssel nochmals auf die Ziffern der verschlüsselten Nachricht an (Verknüpfung durch XODER() und Multiplikation mit 1), so entstehen Ziffern, die entsprechend ihrer Gruppe wieder zu Zeichenketten aus Nullen und Einsen zusammengefasst werden. Eine Umwandlung dieser Zeichenketten mittels BININDEZ liefert einen Zahlencode, der mithilfe von ZEICHEN() ausgewertet werden muss. Es entsteht wieder das Wort »Excel«.

FALSCH(); WAHR(); ODER(); UND()

Das Beispiel finden Sie im Ordner \Ms5-235\Kap08 in der Arbeitsmappe *Wahrheitswerte.xlsx* auf dem Arbeitsblatt *XODER*.

Siehe auch



Kapitel 9

Nachschlage- und Verweisfunktionen

ADRESSE()	279
BEREICH.VERSCHIEBEN()	281
BEREICHE()	284
FORMELTEXT	285
HYPERLINK()	286
INDEX()	287
INDIREKT()	292
MTRANS()	293
PIVOTDATENZUORDNEN()	295
RTD()	295
SPALTE()	297
SPALTEN()	298
SVERWEIS()	299
VERGLEICH()	301
VERWEIS()	304
WAHL()	306
WVERWEIS()	307
ZEILE()	309
ZEILEN()	310



Nachschlagen
und Verweisen ▾

Nachschlage- und Verweisfunktionen bzw. Matrixfunktionen sind in gewissem Sinne eines von vielen Sahnehäubchen von Excel. Vor allem der Einsatz der Verweis-Funktionen (SVVERWEIS(), VERWEIS() und WVERWEIS()), aber auch INDEX() und VERGLEICH()), die es erlauben, gezielt Informationen aus Teilen eines Arbeitsblatts herauszufinden, bringt Erstaunliches zu Wege. Oft ist es auch die Kombination der Funktionen, die zum gewünschten Ziel führt.

Einige Funktionen halten »globale« Informationen zu den betroffenen Zellen bereit (SPALTE(), ZEILE(), ADRESSE()), andere tun das im »Lokalen« (BEREICH.VERSCHIEBEN(), SPALTEN(), ZEILEN()).

Mag am Anfang die Vielfalt der Namen und die Zahl der Einsatzmöglichkeiten verwirren – auch hier macht Übung den Meister.

Tabelle 9.1

Die Nachschlage- und Verweisfunktionen in der Übersicht

Funktion	Beschreibung
ADRESSE()	Liefert einen Bezug auf eine Zelle einer Tabelle als Text
BEREICH.VERSCHIEBEN()	Gibt einen Bezug zurück, der gegenüber dem angegebenen Bezug versetzt ist
BEREICHE()	Gibt die Anzahl der innerhalb eines Bezugs aufgeführten Bereiche zurück
FORMELTEXT()	Liefert die Formel, die im gegebenen Bereich eingetragen wurde, als Text
HYPERLINK()	Erstellt eine Verknüpfung, über die ein auf einem Netzwerkservers, in einem Intranet oder im Internet gespeichertes Dokument geöffnet wird
INDEX()	Liefert aus einem Bezug oder einer Matrix den Wert mit dem gegebenen Index
INDIREKT()	Gibt den Bezug zum gegebenen Textwert zurück
MTRANS()	Gibt die transponierte Matrix der angegebenen Matrix zurück
PIVOTDATENZUORDNEN()	Gibt Daten aus einem PivotTable-Bericht zurück
RTD()	Ruft Echtzeitdaten von einem Programm ab, das die COM-Automatisierung unterstützt
SPALTE()	Gibt die Spaltennummer eines Bezugs zurück
SPALTEN()	Gibt die Anzahl der Spalten einer Matrix oder eines Bezugs zurück
SVVERWEIS()	Sucht in der ersten Spalte einer Tabelle nach einem Wert und gibt den Wert aus der Trefferzeile mit dem angegebenen Spaltenindex zurück
VERGLEICH()	Sucht Werte innerhalb eines Bezugs oder einer Matrix und liefert den Trefferindex zurück
VERWEIS()	Durchsucht die Werte eines Vektors oder einer Matrix
WAHL()	Verwendet Index, um einen Wert aus der Liste der Werteargumente zurückzugeben
WVERWEIS()	Sucht in der obersten Zeile einer Matrix nach Werten und gibt dann den Wert aus der Trefferspalte mit dem gegebenen Zeilenindex zurück
ZEILE()	Liefert die Zeilennummer eines Bezugs
ZEILEN()	Gibt die Anzahl der Zeilen in einem Bezug oder einer Matrix zurück

ADRESSE()



ADDRESS()

ADRESSE(*Zeile;Spalte;Abs;A1;Tabellenname*)

Diese Funktion wandelt ihre Argumente in eine Zeichenkette in Form eines Zellbezugs um.

Zeile und *Spalte* (erforderlich) legen die »Koordinaten« der erzeugten Adresse fest. Diese Argumente können beliebige Ausdrücke sein, die eine numerische Auswertung derart gestatten, dass eine Zeichenkette mit gültigen Bezugsangaben entsteht (also Werte zwischen 1 und maximal 1.048.476 für *Zeile* und 16.384 für *Spalte*).

Abs (optional) entscheidet über absolute bzw. relative Bezugsangaben. Die notwendige »Übersetzung« finden Sie in Tabelle 9.2.

Bezugsart	Argument <i>Abs</i>
Zeile und Spalte absolut	1
Zeile absolut, Spalte relativ	2
Zeile relativ, Spalte absolut	3
Zeile und Spalte relativ	4

A1 (optional) entscheidet über die verwendete Bezugsart (*A1* – also Buchstaben für die Spalten, Zahlen für die Zeilen – oder *Z1S1* – Zahlen für beide). Gefordert wird ein Wahrheitswert: *WAHR* entspricht *A1*, *FALSCH* entspricht *Z1S1*. Lassen Sie das Argument weg, wird *A1* als Bezugsart verwendet.

Tabellenname (optional) setzt der Bezugszeichenkette einen Tabellenblattnamen gefolgt von einem Ausrufezeichen voran. Es wird ein Ausdruck verlangt, der in Text umgewandelt werden kann. Verzichten Sie auf dieses Argument, wird nur eine »einfache« Bezugsangabe erzeugt.

Entstehen nach der Auswertung für die Argumente *Zeile* oder *Spalte* gebrochene positive Zahlen größer 1, wird in der weiteren Verwendung dieser Wert auf den größten ganzzahligen Anteil reduziert, die Nachkommastellen also abgeschnitten.

Es besteht kein Zusammenhang zwischen der Anzeige der Adress-Zeichenkette in *A1*- oder *Z1S1*-Schreibweise und der in den Optionen vorgenommenen Anzeigart von Bezügen in der Arbeitsmappe. Der Wert von *Adresse* richtet sich nach dem dritten Argument.

Es wird nicht geprüft, ob der angegebene *Tabellenname* tatsächlich existiert.

Die sofortige Verwendung des Ergebnisses der Funktion ADRESSE() als Bezug ist nicht möglich, da im Ergebnis eine Zeichenkette und kein Bezug entsteht. Sie können das u.a. mithilfe der Funktion ISTBEZUG() prüfen oder aber versuchen, die Zeichenkette als Bezug in der Formel einer anderen Zelle einzusetzen. Einen Ansatz zur Lösung dieses Problems finden Sie im zweiten Beispiel.

Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

Sie möchten für einen Tabellenausschnitt eine automatische Beschriftung mit den Buchstaben A, B usw. erzeugen. Dazu schreiben Sie in die linke obere Zelle des Ausschnitts (etwa *C14*) die Formel

```
=LINKS(ADRESSE(1;SPALTE()-SPALTE($C$14)+1;4);1)
```

und kopieren diese soweit Sie es wünschen nach rechts.

Syntax

Definition

Argumente

Tabelle 9.2

»Übersetzung«
der Bezugsart
in Zahlen

Hintergrund

Wichtig

Praxiseinsatz

**Automatische
Beschriftungen**

Die Funktion SPALTE() ermittelt die Spaltennummer der Zelle mit der Formel (Achtung: bei Angabe des Arguments $\$C\14 in der Zelle C14 entsteht kein Zirkelbezug). Die Subtraktion der Spaltenwerte und die Addition von 1 sorgen dafür, dass immer bei A angefangen wird. Addieren Sie 2, so beginnt es bei B usw. Die Funktion LINKS() mit dem zweiten Argument 1 reduziert die Ausgabe auf genau ein Zeichen.

Indirekte Adressierung

Die Verwendung der Formel =ADRESSE(6;2) liefert die Zeichenkette \$B\$6. Wollen Sie den Inhalt der Zelle B6 in weiteren Berechnungen verwenden, verwenden Sie die Funktion INDIRECT(), die die Zeichenkette als Argument nimmt und automatisch in einen gültigen Bezug umwandelt. Im Ergebnis von

=INDIREKT(ADRESSE(6;2))

entsteht nämlich der Inhalt der Zelle B6.

Letzte Zelle im Bereich finden

In verschiedenen Aufgabenstellungen ist es wichtig, den Inhalt der letzten Zelle rechts unten in einer Liste (oder eines allgemeinen Bereichs) zu benutzen, ohne im Vorhinein zu wissen, wie lang diese Liste ist.

Angenommen, Sie haben eine Liste mit der Notierung von Ein- und Auszahlungen wie in Abbildung 9.1.

	A	B	C	D
1	Datum	Vorgang	Betrag	Summe
2	01.08.2010	A	12,00 €	12,00 €
3	02.08.2010	B	13,00 €	25,00 €
4	03.08.2010	C	- 17,00 €	8,00 €
5	04.08.2010	D	29,75 €	37,75 €

Abbildung 9.1: Eine kleine »Buchführung«

Um nun an anderer Stelle im Arbeitsblatt (etwa rechts von der Liste) oder auf einem anderen Arbeitsblatt den aktuellen Kontostand (hier 37,75 €) weiter verwenden zu können, nutzen Sie eine Formel nach dem Muster

=INDIREKT(ADRESSE(ANZAHL(A:A)+1;4))

bzw.

=INDIREKT(ADRESSE(ANZAHL(Zahlungen!A:A)+1;4;;;"Zahlungen"))

Dabei ermittelt ANZAHL() die Anzahl numerischer Werte in der Spalte A (Sie sollten nur Datumswerte der Liste in dieser Spalte halten). Durch die Addition von 1 berücksichtigen Sie die Überschriftenzeile und die Funktion INDIRECT() erledigt den Rest wie im vorigen Beispiel beschrieben.

Die zweite Formel geht davon aus, dass sich Ihre Liste im Tabellenblatt *Zahlungen* befindet, diesen Parameter geben Sie der Funktion ADRESSE() mit.

Sie haben die Möglichkeit, der Tabelle der Zahlungen einen Namen zu geben (etwa *Buchführung*) und diesen in die Formel zur Ermittlung der letzten Zelle einfließen zu lassen:

=INDIREKT(ADRESSE(ANZAHL(Buchführung[Datum])+1;4))

Das in eckige Klammern geschriebene Argument zeigt auf die Datumsspalte.

Tipp

Diese Aufgabe lässt sich auch elegant mit der Vergabe dynamischer Namen durch die Funktion BEREICH.VERSCHIEBEN() lösen (siehe die dortigen Beispiele).

SPALTE(), ZEILE(), BEREICH.VERSCHIEBEN()

Die Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap09 in der Arbeitsmappe *Adresse.xlsx*.

Siehe auch



BEREICH.VERSCHIEBEN() OFFSET()

BEREICH.VERSCHIEBEN(*Bezug;Zeilen;Spalten;Höhe;Breite*)

Diese Funktion gibt einen Zellbezug zurück, der gegenüber einem angegebenen Bezug versetzt und/oder erweitert bzw. reduziert ist.

Bezug (erforderlich) ist ein Bezug zu einer einzelnen oder auch mehreren (insgesamt rechteckförmig aneinandergrenzenden) Zellen, die den Startpunkt des »Verschiebevorgangs« bilden. Ein Argument, welches keinen Bezug in diesem Sinne darstellt, führt zum Fehlerwert #WERT!.

Verwenden Sie einen benutzerdefinierten Namen für das Argument, müssen Sie die Eingabe der Formel auch bei Verwendung in nur einer Zelle mit + + (Matrixformel) abschließen. Andernfalls wird ein Fehler erzeugt.

Zeilen (erforderlich) beschreibt als ganze Zahl die Anzahl der Zeilen, um die Sie die obere linke Eckzelle des Bereichs nach oben oder nach unten verschieben möchten.

Spalten (erforderlich) lässt auch Ausdrücke zu, die sich als ganze Zahlen auswerten lassen. Angegeben wird mit diesem Argument, um wieviele Spalten Sie den Bereich nach links oder nach rechts verschieben möchten.

Höhe (optional) gibt die Höhe des neuen Bezugs in Zeilen an. Wenn Sie dieses Argument angeben, muss es als positive ganze Zahl auswertbar sein.

Breite (optional) arbeitet analog zu **Höhe**, nur dass hiermit eine Anzahl von Spalten beschrieben wird.

Beachten Sie, dass durch die Funktion nicht etwa Zellen physisch auf einem Arbeitsblatt verschoben werden. Verschoben wird nur der Bezug hinsichtlich eines ursprünglichen definierten Bereichs. Wird dabei für die Argumente *Zeile* oder *Spalte* ein Wert angegeben, der über den aktuellen »Blattrand« hinaus verweist, gibt BEREICH.VERSCHIEBEN() den Fehlerwert #BEZUG! zurück.

Die Funktion erwartet in den vier letzten Argumenten ganze Zahlen, von denen die beiden letzten positiv sein müssen. Können von Ihnen verwendete Ausdrücke in diesen Argumenten zu gebrochenen Zahlen ausgewertet werden, werden die Nachkommastellen abgeschnitten. Ein Fehler tritt nicht auf.

Verzichten Sie auf das Argument **Höhe** oder **Breite**, geht Excel davon aus, dass der neue Bezug dieselbe Höhe oder Breite wie der ursprüngliche Bezug hat.

Ist das Argument **Höhe** kleiner als die Höhe des Zielbereichs (also des Bereichs, in welchem Sie die Funktion verwenden), erscheint (außer im Falle von **Höhe** gleich 1) in den überzähligen Zellen der Fehlerwert #NV. Ein gleiches Verhalten zeigt das Argument **Breite**. Im Falle des Werts von 1 werden die entsprechenden Zeilen oder Spalten in den überzähligen Zellen wiederholt. Die Abbildung 9.2 zeigt ein Beispiel.

Syntax

Definition

Argumente

Wichtig

Hintergrund

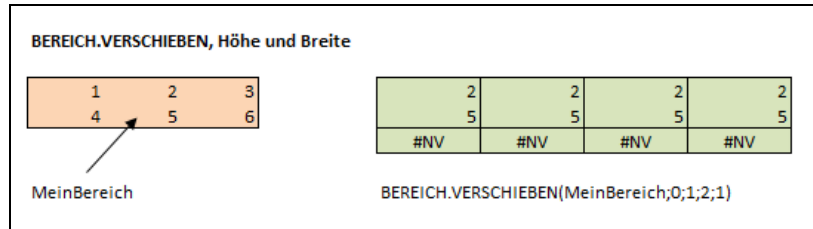


Abbildung 9.2: Die Wirkung von Höhe und Breite

Der Bezug, der den Namen *MeinBereich* trägt, hat die »Abmessungen« drei mal zwei. Der verschobene Bezug wird mit der Formel

=BEREICH.VERSCHIEBEN(MeinBereich;0;1;2;1)

umgesetzt, dabei wird nur auf den Teil ab Spalte 2 abgezielt (*Spalten* ist gleich 1, *Zeilen* gleich 0 – das bedeutet, es beginnt ab zweiter Spalte). Der Zielbereich hat die »Maße« vier mal drei. Der angegebene Wert für *Höhe* ist mit 2 für den ursprünglichen Bezug gerade richtig, für den Zielbereich aber zu klein. Der angegebene Wert für *Breite* ist 1 und damit passt die eine Spalte (Spalte 2) viermal in den Zielbereich.

Praxeinsatz Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

Ansprechen einzelner Zellen Mithilfe der Funktion lassen sich einzelne Zellen eines ursprünglichen benannten Bereichs ansprechen. In diesem Fall geht es also nicht um die gedachte Verschiebung des ganzen Bereichs, sondern nur um die linke obere Ecke des Bereichs. Mit der Angabe von *Höhe* und *Breite* gleich 1 erhalten Sie eine einzelne Zelle. Bei anderen Angaben beachten Sie bitte die Anmerkungen zu Abbildung 9.2.

Angenommen, der von Ihnen betrachtete Bereich hat den Namen *MeinBereich* und befindet sich in den Zellen B5 bis D6. Die Zelle D6 ist die »letzte« im Bereich und kann durch die Matrixformel

{=BEREICH.VERSCHIEBEN(MeinBereich;1;2)}

angesprochen werden. Um die notwendigen Schritte nach rechts bzw. unten zu gehen, orientieren Sie sich an der linken oberen Ecke des Bereichs: zwei nach rechts und einen nach unten.

Besonders eindrucksvoll gestaltet sich diese Art der Zellermittlung bei dynamischen Bereichen, deren Ausdehnung Ihnen zur Zeit des Tabellenblattentwurfs noch nicht vorliegt und die sich im Laufe der Zeit auch ändern kann (dynamische Listen, per Hand eingegeben oder aus einer Datenbank aktualisiert). Hier können Ihnen Funktionen wie ANZAHL() oder ZÄHLENWENN() bei der Angabe der Positionen behilflich sein können. Details bringen die folgenden Beispiele.

Listen variabler Länge Angenommen, Sie haben eine Liste wie in Abbildung 9.3 (unterer Teil) vor sich. Sie möchten durch den Einsatz von Datenbank-Funktionen Informationen der Liste herausfiltern. Die Liste ändert sich laufend, da manche Datensätze hinzukommen, andere als erledigt entfernt wurden.

	A	B	C	D	E
1			Vorgänge	Kategorie	Betrag
2			2	Deutsch	48,97 €
3			Vorgänge	Kategorie	Betrag
4			3	Zeichnen	54,37 €
5					
6		Nr.	Vorgang	Kategorie	Betrag
7		1	Hefte	Deutsch	12,99 €
8		2	Bücher	Deutsch	35,98 €
9		3	Stifte	Zeichnen	9,99 €
10		4	Farben	Zeichnen	26,49 €
11		5	Pinself	Zeichnen	17,89 €

Abbildung 9.3: Auswertung von Listen mittels dynamischer Namen

Bereiten Sie zuerst die Liste sowie die darüber liegenden Kriterienbereiche in einem Tabellenblatt namens *Rechnungen* vor. Die Überschriften der Kriterienbereiche müssen nicht unbedingt den Überschriften der Liste gleich lautend sein, weist »Vorgänge« statt »Vorgang« auf die Ermittlung der Anzahl hin.

Markieren Sie zunächst die Überschriften der Liste. Öffnen Sie anschließend im Menüband die Registerkarte *FORMELN* und klicken Sie in der Gruppe *Definierte Namen* auf *Namen definieren*. Vergeben Sie den Namen *Liste* für den durch die Formel

```
=BEREICH.VERSCHIEBEN(Rechnungen!$B$6:$E$6;0;0;ANZAHL(Rechnungen!$B:$B)+1)
```

definierten Bereich. Hier wird entsprechend der Anzahl numerischer Einträge in der *B*-Spalte (diese ist damit in gewissem Sinne »reserviert«) die linke obere Ecke des Bereichs der Überschriften (*\$B\$6:\$E\$6*) dynamisch durch das sich anpassende Argument *Höhe* erweitert, +1 ist notwendig, um die Überschriften mit in die Liste einzubeziehen.

Um beispielsweise die Einträge in der Kategorie »Deutsch« zu summieren, verwenden Sie eine Formel nach dem Muster

```
=DBSUMME(Liste;E6;D1:D2)
```

DBSUMME() verlangt als erstes Argument den Listenbereich, als zweites einen Hinweis auf die zu summierende Spalte durch Angabe des Namens bzw. eines Bezugs mit dem Namen und als drittes die zu berücksichtigenden Kriterienfelder. Analog lauten die anderen Einträge zur Auswertung, also etwa

```
=DBANZAHL(Liste;B6;D1:D2)
```

zur Ermittlung der Anzahl, wobei hier das zweite Argument auch leer bleiben kann.

Sie haben auch die Möglichkeit, auf die Vergabe des Bereichsnamens zu verzichten, indem Sie die Liste als »Tabelle« formatieren. Hat diese den (als Standard zunächst vergebenen) Namen *Tabelle1*, lautet die letzte Formel

```
=DBANZAHL(Tabelle1[#A11e];B6;D1:D2)
```

Eine ähnliche Vorgehensweise wie im vorigen Beispiel können Sie nutzen, wenn Diagramme dynamische Quellen haben sollen, deren »Länge« der Datenreihen variieren kann. Definieren Sie einfach einen dynamischen Bereichsnamen oder eine »intelligente« Tabelle (*EINFÜGEN/Tabelle*) und nutzen Sie den dadurch benannten Bereich zur Erstellung des Diagramms. Beachten Sie, dass in den Einstellungen zur Datenquelle von Diagrammen bzw. in der Funktion *DATENREIHE()* in der Eingabezeile zur Definition der Quelle die Benutzung der Funktion *BEREICH.VERSCHIEBEN()* nicht erlaubt wird.

**Dynamische
Diagramme**

Noch einmal Adressen Im dritten Beispiel zur Funktion ADRESSE() wurde die Information zum Inhalt einer bestimmten Zelle durch die Funktionen ADRESSE() und INDIREKT() ermittelt. Ebenso elegant kann die Ermittlung durch Kombination der beiden ersten Beispiele dieses Abschnitts geschehen.

Um also im Beispiel aus Abbildung 9.1 auf Seite 280 die aktuelle Summe in einer anderen Zelle verwenden zu können, vergeben Sie den Namen *Zahlungsreihe* für den dynamischen Bereich

```
=BEREICH.VERSCHIEBEN(Zahlungen!$A$1:$D$1;0;0;ANZAHL(Zahlungen!$A:$A)+1)
```

auf dem Tabellenblatt *Zahlungen*. In Abhängigkeit von Spalte A und ihren Einträgen passt sich der Bereich (inklusive Überschriften) an.

Den Bezug auf die rechte unterste Ecke des Bereichs, die den aktuellen Gesamtstand enthält, bekommen Sie nun über die Matrixformel

```
{=BEREICH.VERSCHIEBEN(Zahlungsreihe;ZEILEN(Zahlungsreihe)-1;3)}
```

in welcher Sie beachten, dass die Angabe der Zeilen- und Spaltennummern bei Null beginnt.

Mit der Formatierung als Tabelle (Name *Tabelle2*) statt des dynamischen Namens entsteht

```
{=BEREICH.VERSCHIEBEN(Tabelle2[#A11e];ZEILEN(Tabelle2[#A11e])-1;3)}
```

Siehe auch ADRESSE(), INDIREKT(), WAHL()



Die Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap09 in der Arbeitsmappe *Bereich.verschieben.xlsx*.

BEREICHE() AREAS()

Syntax BEREICHE(*Bezug*)

Definition Diese Funktion gibt die Anzahl der innerhalb des Arguments *Bezug* aufgeführten Bereiche zurück.

Argumente *Bezug* (erforderlich) muss sich als Bezug auf einen oder mehrere Zellbereiche auswerten lassen. Andernfalls resultiert daraus ein Fehler (Sie können die Formel nicht eingeben) oder ein Fehlerwert.

Hintergrund Besteht Ihr Argument aus mehreren Bezügen, die durch Semikola getrennt sind, muss die Formeleingabe mit einem zusätzlichen Klammerpaar erfolgen, beispielsweise

```
=BEREICHE((A1;A2))
```

oder

```
=BEREICHE((A1:A2;B3))
```

(andernfalls erfolgt eine Interpretation des Semikolons als Listentrennzeichen und die Eingabe endet mit einer Fehlermeldung).

Versuchen Sie die Anzahl der Bereiche einer leeren Menge zu ermitteln, erhalten Sie den Fehlerwert *#NULL!*, etwa bei der Verwendung der Schnittmenge aus A1 und A2 in

```
=BEREICHE(A1 A2)
```

Die Anzahl von Bereichen zu zählen, die vorher irgendwie angegeben werden müssen, gehört sicher nicht zu den täglichen Problemen des Excel-Anwenders.

Denkbar ist allerdings folgende Situation. Sie haben einer Liste, die sich dynamisch in der Länge verändert, mithilfe der Funktion `BEREICH.VERSCHIEBEN()` einen Namen zugewiesen (siehe die Beispiele zu dieser Funktion) oder nutzen die Formatierung der Liste als »Tabelle«. Sie haben die Absicht, diese Liste stets »von oben«, also durch Hinzufügen von Zeilen gleich nach der Überschrift zu füllen. Sie möchten, dass die Liste nicht länger als 100 Einträge wird. Wird Zeile 100 erreicht, wollen Sie durch eine sich einfärbende Überschriftenzeile darauf aufmerksam gemacht werden.

Der Gedanke ist nun, die Bereiche in der Schnittmenge zwischen Ihrer Liste (die den Namen *Liste* erhalten hat) und der Zelle *A101* durch

```
=BEREICHE(Liste Sonstige!$A$101)
```

zu zählen (angenommen, Ihre Liste beginnt bei *A1*). Wird der Wert 1 ermittelt, soll über eine bedingte Formatierung die Überschriftenzeile eingefärbt werden. Leider schlägt die Umsetzung dieses Gedankens fehl, da das Dialogfeld zur bedingten Formatierung keine Schnittmengenoperationen (Angabe von Zellbezügen mit Leerzeichen dazwischen) erlaubt.

Sie greifen deshalb zu einem Trick und geben der obigen Formel einen Namen (Registerkarte *FORMELN* in der Gruppe *Definierte Namen* den Befehl *Namen definieren* aufrufen und dann den Namen – etwa *Formel* – eintragen und als Bezug die obige Formel eingeben). Nunmehr schreiben Sie in das *Formel*-Feld des Dialogfelds zur bedingten Formatierung der Überschriftenzeile die Formel

```
=(Formel=1)
```

und werden somit bei entsprechender Vergabe einer Farbe für das Muster der Zellen über das Erreichen von 100 Zeilen informiert.

`ADRESSE()`, `INDEX()`, `SPALTE()`, `SPALTEN()`, `ZEILE()`, `ZEILEN()`, `ZELLE()`

Das Beispiel (in leicht abgewandelter Form) sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner `\Ms5-235\Kap09` in der Arbeitsmappe *Bezugsfunktionen.xlsx* auf dem Arbeitsblatt *Sonstige*.

Praxiseinsatz

Siehe auch



FORMELTEXT FORMULATEXT

`FORMELTEXT`(*Bezug*)

Die Funktion liefert die Formel, die in dem durch das Argument definierten Bereich eingetragen wurde (das ist der Text der Bearbeitungsleiste).

Bezug (erforderlich) erwartet die Angabe eines Ausdrucks, der auf eine einzelne Zelle verweist (Ausnahme: Einsatz als Matrixformel).

Fehler entstehen, wenn

- ▶ die angesprochene Zelle keine Formel enthält (*#NV*),
- ▶ der Bezug ungültig ist (*#NV* bei nicht vorhandenen Bezügen oder *#NAME* bei nicht definierten Namen),

Neu in Excel 2013

Syntax

Definition

Argumente

Hintergrund

- ▶ im Bezug eine Formel mit Bezugsfehler steht (*#BEZUG*),
- ▶ die Formel zu lang – mehr als 8.192 Zeichen – ist (*#NV*),
- ▶ der Bezug zu einer nicht geöffneten Mappe verweist (*#NV*) oder
- ▶ der Bezug auf eine geschützte Zelle verweist (*#NV*).

Wird ein Bereich angesprochen, der mehr als eine Zelle enthält, steht die Formel der linken oberen Ecke als Stellvertreter im Ergebnis der Funktion.

Praxeinsatz Anstatt der Formelanzeige für das gesamte Tabellenblatt lassen sich gezielt die Formeln einzelner Zellen zur Verdeutlichung anzeigen. Das sollte vor allem während des Entwurfs eines komplexen Arbeitsblatts hilfreich sein.

Siehe auch Keine



Zum Testen finden Sie im Ordner `\Ms5-235\Kap09` in der Arbeitsmappe *Bezugsfunktionen.xlsx* das Arbeitsblatt *Sonstige*.

HYPERLINK() HYPERLINK()

Syntax `HYPERLINK(Hyperlink_Adresse;Freundlicher_Name)`

Definition Diese Funktion erstellt einen Hyperlink zu einem Dokument auf dem lokalen Rechner, einem Netzwerkserver oder einem Server im Intranet oder Internet.

Argumente *Hyperlink_Adresse* (erforderlich) erwartet die Angabe eines Ausdrucks, der sich als Zeichenkette auswerten lässt und den Namen bzw. vollständigen Pfad zum aufgerufenen Dokument angibt.

Freundlicher_Name (optional) gibt den in der Zelle angezeigten Wert an. Es kommt damit ein Ausdruck infrage, der sich als Text, Zahl, Wahrheits- oder Fehlerwert auswerten lässt. Fehlt dieses Argument, wird die Auswertung von *Hyperlink_Adresse* angezeigt.

Hintergrund Anstatt mit `[Strg]+[K]` statische Hyperlinkeinträge zu setzen, können Sie mit dieser Funktion die Hyperlinks auch berechnen, also dynamisch machen. Beachten Sie:

- ▶ Ist das erste Argument als gültige Adresse mit eventueller Textmarke auswertbar¹ und
- ▶ ist das zum Typ der aufgerufenen Datei passende Programm (etwa Microsoft Word oder der Microsoft Internet Explorer) ordnungsgemäß installiert,

startet dieses Programm und öffnet die gewünschte Datei, sobald auf die Zelle mit der eingetragenen Funktion geklickt wird.

Wollen Sie die Zelle bearbeiten, halten Sie die gedrückte Maustaste so lange über der Zelle, bis sich der Mauszeiger in ein Kreuz verwandelt.

Wird keine gültige Adresse im ersten Argument angegeben, erzeugt der Klick auf die Zelle eine Fehlermeldung.

Praxeinsatz Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

¹ Dateiname im selben Verzeichnis wie die Arbeitsmappe, vollständiger Pfad inkl. Name zu einer Datei auf dem lokalen Rechner, vollständiger Pfad und Name zu einer Datei im dateibasierten Netzwerk, Angabe einer Adresse zu einer Domäne oder einer Datei im Internet oder Intranet unter Verwendung des Protokolls wie `http://www.microsoft.com`.

Der folgende Eintrag öffnet den Internet Explorer mit der deutschen Office-Website von Microsoft:

```
=HYPERLINK("http://office.microsoft.com/de-de/default.aspx "; "Internet-Test")
```

Angezeigt wird in der Zelle der Link »Internet-Test«.

Mit

```
=HYPERLINK("dokument#textmarke"; "Word-Test")
```

öffnen Sie das genannte Word-Dokument, welches sich im gleichen Ordner wie die Arbeitsmappe befinden muss, und der Text der Textmarke »textmarke« wird markiert. In der Zelle angezeigt wird der Link »Word-Test«.

Keine

Die Beispiele finden Sie im Ordner `Ms5-235\Kap09` in der Arbeitsmappe *Bezugsfunktionen.xlsx* auf dem Arbeitsblatt *Sonstige*. Im gleichen Verzeichnis befindet sich eine Word-Datei zum Testen.

Link zu Office Online

Gezielter Zugriff auf Word-Dokumente

Siehe auch



INDEX() INDEX()

Syntax 1 (Matrixversion):

```
INDEX(Matrix; Zeile; Spalte)
```

Syntax 2 (Bezugsversion):

```
INDEX(Bezug; Zeile; Spalte; Bereich)
```

Diese Funktion verwendet einen Zeilen- und/oder einen Spalten-Index, um aus einer Matrix (mehrzelliger Verbund von Zellen in rechteckiger Form oder eine in geschweifte Klammern gesetzte, durch Semikolon getrennte Folge von Ausdrücken) oder einem Bezug, der auch aus mehreren rechteckigen Bezügen bestehen kann, einen oder mehrere Werte entsprechend der Indexangaben auszuwählen.

Syntax 1 (Matrixversion):

Matrix (erforderlich) wird als ein Zellbereich oder eine Matrixkonstante erwartet.

Zeile (optional) kann weggelassen werden, wenn das Argument *Matrix* nur einem einzeiligen Gebilde entspricht, muss als nicht negative ganze Zahl ausgewertet werden können und gibt die Zeilennummer der auszuwählenden Werte an.

Spalte (optional) ist ein Ausdruck, der sich in eine nicht negative ganze Zahl umwandeln lassen muss. Dieses Argument gibt die Spaltennummer der auszuwählenden Werte an. Es kann weggelassen werden, wenn der Bezug *Matrix* nur aus einer Spalte besteht.

Syntax 2 (Bezugsversion):

Bezug (erforderlich) muss als ein Bezug auf einen oder mehrere rechteckige Zellbereiche auswertbar sein.

Zeile und **Spalte** (optional) haben die gleiche Bedeutung wie in der Matrixversion. Sie bestimmen die Zeilen- bzw. Spaltennummern der auszuwählenden Werte und müssen deshalb als nicht negative ganze Zahlen auswertbar sein.

Syntax

Definition

Argumente

Besteht das Argument *Bezug* aus mehreren Teilen, werden diese in der Reihenfolge ihrer Notierung nummeriert.

Bereich (optional) erwartet eine positive ganze Zahl. Hierdurch können Sie auf den Teilbereich mit der entsprechenden Nummer zugreifen.

Hinweis Besteht das Argument *Bezug* nur aus einem einzigen Teilbereich, können Sie das Argument *Bereich* weglassen. Die Bezugsversion stimmt dann mit der Matrixversion überein.

Hintergrund Die Angabe der Argumente *Zeile* und *Spalte* in der Matrixversion erfolgt in der Regel zwischen 1 und der Anzahl der Zeilen bzw. Spalten im rechteckigen Bezug oder in einer speziellen Liste von Werten. Besteht das Rechteck nur aus einer einzigen Zeile (Spalte), können Sie das Argument *Zeile* (*Spalte*) weglassen. Lassen Sie *Zeile* weg, ist das durch zwei aufeinander folgende Semikola zu markieren. Lassen Sie *Zeile* im Falle mehrerer Zeilen weg, können Sie die durch *Spalte* angegebene Spalte mithilfe einer Matrixformel in untereinander liegenden Zellen referenzieren. Analoges erreichen Sie zeilenweise, indem Sie das Argument *Spalte* bei einer mehrspaltigen Matrix weglassen.

Verlassen Sie mit Ihren Argumenten *Zeile* bzw. *Spalte* die »Abmessungen« des gedachten Rechtecks, erhalten Sie den Fehlerwert *#BEZUG!* angezeigt.

So erhalten Sie mit

```
=INDEX(B4:C6;3;2)
```

das Element, welches sich in *C6* befindet (dritte Zeile, zweite Spalte des Bereichs *B4:C6*).

Analog ist die Situation im Fall von

```
=INDEX({2;4;6;8};2;1)
```

(das ist keine Matrixformel, die durch + + entsteht, die geschweiften Klammern tippen Sie selbst ein). Die Folge {2;4;6;8} wird als Spalte interpretiert und die Formel liefert den Wert 4 (zweite Zeile in der ersten und einzigen Spalte).

Eine Besonderheit ist die Angabe von mehrspaltigen (sogenannten) Matrixkonstanten. Hier erfolgt die Trennung von Zeilen und Spalten durch die Trennzeichen Semikolon und Punkt. {11.12.13;21.22.23} wird als zweizeiliges Gebilde (die Zeilen lauten 11, 12, 13 und 21, 22, 23) mit drei Spalten erkannt (die Spalten lauten entsprechend 11, 21 und 12, 22 sowie 13, 23). Es liefert also

```
=INDEX({11.12.13;21.22.23};2;3)
```

den Wert 23 (zweite Zeile, dritte Spalte).

Die oben angedeutete Besonderheit beim Einsatz von Matrixformeln liegt in Folgendem: Sie lassen eines der Argumente *Zeile* oder *Spalte* weg oder setzen es gleich 0 (Null). Ihre Eingabe in mehrere Zellen einer Spalte bzw. mehrere Zellen einer Zeile schließen Sie wie immer mit + + ab. Schreiben Sie beispielsweise

```
{=INDEX({11.12.13;21.22.23};0;3)}
```

oder

```
{=INDEX({11.12.13;21.22.23};;3)}
```

in zwei untereinander liegende Zellen als Matrixformel, erhalten Sie das Wertepaar 13, 23.

Schreiben Sie

```
{=INDEX({11.12.13;21.22.23};2;0)}
```

oder

```
{=INDEX({11.12.13;21.22.23};2)}
```

in drei nebeneinander liegende Zellen als Matrixformel, entsteht der Dreier 21, 22, 23.

Analog gehen Sie vor, wenn das Argument *Matrix* einen Zellbereich betrifft.

Ist Ihr »Zielbereich« mit der Matrixformel von der Ausdehnung her größer als die zur Verfügung stehende »Quelle«, entsteht in den überzähligen Zellen der Fehlerwert #NV.

In der Bezugsversion ist das erste Argument zwingend ein Bezug. Wollen Sie auf mehrere Bereiche abstellen, müssen diese nicht zusammenhängen, aber natürlich jeder für sich einen rechteckigen Zellverbund betreffen. Der dann einfachste Fall ist der einzellige Verbund. Das Argument *Bezug* ist im Fall von mehreren Bezügen, die durch Semikola voneinander getrennt werden, selbst in runde Klammern zu schreiben, damit Excel bei der Zuordnung der Argumente der Funktion nicht durcheinander kommt.

Die Reihenfolge der Bezüge im Argument bestimmt eine ganzzahlige Nummer beginnend ab 1, die es gestattet, den gemeinten Zellbereich im Argument *Bereich* zu identifizieren. Die Argumente *Zeile* und *Spalte* arbeiten dann so wie in der Matrixversion.

Verlassen Sie mit einem der Argumente *Zeile*, *Spalte* oder *Bereich* die durch *Bezug* gesetzten Grenzen, liefert die Funktion den Fehlerwert #BEZUG!

Durch eine Funktionsangabe wie

```
=INDEX((B18:C20;E18:G19);3;2;1)
```

erhalten Sie das Element (oder genauer einen Bezug auf dieses) der dritten Zeile in der zweiten Spalte des ersten Bereichs, also einen Bezug auf C20.

INDEX() arbeitet im Argument *Bezug* natürlich auch mit benannten Bereichen. Geben Sie etwa den Zellen B18:C20 den Namen *erster* (Registerkarte FORMELN, Gruppe *Definierte Namen/Namen definieren*) und dem Bereich von E18 bis G19 den Namen *zweiter*. In diesem Fall erhalten Sie mit

```
=INDEX((erster;zweiter);2;1;2)
```

einen Bezug auf die Zelle in der zweiten Zeile der ersten Spalte des Bereichs mit Namen *zweiter*, also E19. Sie können auch allen Zellen B18:C20;E18:G19 (genau in dieser Reihenfolge) den Namen *beide* geben. Dann liefert Ihnen

```
=INDEX(beide;2;3;2)
```

die Informationen aus der dritten Spalte der zweiten Zeile des zweiten Teilbereichs, also G19.

Sie können eines der Argumente *Zeile/Spalte* oder auch beide weglassen (das heißt, Sie lassen den Platz zwischen den Semikola leer) und damit Spalten oder Zeilen, im zweiten Fall einen ganzen Bereich referenzieren. In jedem dieser Fälle ist die Formel als Matrixformel einzusetzen (Abschluss mit **Strg**+**↵**+**↵**), andernfalls erhalten Sie den Fehlerwert #WERT! angezeigt. Eine Ausnahme gibt es: Der angesprochene Bereich besteht nur aus einer Zeile oder Spalte.

Hinweis Für die Argumente *Zeile*, *Spalte* und *Bereich* werden ganze Zahlen erwartet. Übergeben Sie gebrochene Zahlen, werden diese bei der Auswertung nicht gerundet, sondern die Nachkommastellen abgeschnitten.

Praxiseinsatz Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

In Listen suchen und finden Dieses Beispiel betrifft die Syntax-Version 1. Angenommen, Sie haben eine Liste mit Produkten, in der unter anderem in Spalte 1 der Name des Produkts und in Spalte 2 dessen Preis steht. Diese Liste pflegen Sie auf einem besonderen Tabellenblatt (oder lassen sich die Informationen als Abfrage aus einer Datenbank in dieses Arbeitsblatt importieren). Auf einem anderen Arbeitsblatt möchten Sie durch ein Kombinationsfeld (Formularsteuerelement) Zugriff auf den Namen der Produkte haben, eines auswählen können und dazugehörige Informationen (hier den Preis) in eine Zelle schreiben lassen.

Vergeben Sie für den Datenteil Ihrer Liste den Namen *Preisliste* und für die erste Spalte mit den Produktnamen den Namen *Produkte*.

Ziehen Sie als Nächstes ein Kombinationsfeld auf und vereinbaren als Eingabebereich den Wert *Produkte* und als Zellverknüpfung *\$B\$28*.

Tipp Sie können Ihr Kombinationsfeld über der obigen Zelle platzieren und machen so den kleinen Helfer unsichtbar.

In eine Zelle Ihrer Wahl schreiben Sie nun

```
=INDEX(Preisliste;B28;1)
```

um den gewählten Produktnamen (die Zeile wird durch die Wahl im Kombinationsfeld manipuliert, die Spaltennummer ist 1) und in eine andere Zelle

```
=INDEX(Preisliste;B28;2)
```

um den zugehörigen Preis anzeigen zu lassen. Die Abbildung 9.4 zeigt ein entsprechendes Beispiel.

	A	B	C
37			
38		3	Produkt A
39			
40		Name	Produkt A
41		Preis	56,00 €

Abbildung 9.4: Das Kombinationsfeld sollte später die verknüpfte Hilfszelle überdecken

Hinweis Sie können diese Aufgabe auch mit einem Kombinationsfeld ActiveX-Steuerelemente und unter Einsatz von *SVERWEIS()* lösen. Im Gegenteil zum Formularsteuerelement, welches den Index des gewählten Eintrags an die verknüpfte Zelle liefert, gibt das Kombinationsfeld der Toolbox den Text des gewählten Eintrags an die verknüpfte Zelle (*LinkedCell*) zurück.

Details finden Dieses Beispiel betrifft die Syntax-Version 2. Sie haben in der Vorbereitung einer Fortbildungsveranstaltung drei Teile dieser Veranstaltung voneinander abgegrenzt und bieten Einzel- und Komplettbuchung an. Außerdem gibt es einen Rabatt für Frühbucher, die vor einem Stichtag buchen. Die Details sehen wie in Abbildung 9.5 aus.

46				
	Veranstaltung	Nr.	Frühbucher	Normal
47	PreCon	1	249,00 €	274,00 €
48	MainCon	2	899,00 €	989,00 €
49	PostCon	3	159,00 €	175,00 €
50	Komplett	4	1.175,00 €	1.295,00 €
51				
52	Stichtag	01.06.2013		
53	Konferenz	PreCon		
54	Buchungsdatum	01.05.2013		
55	Preis	249,00 €		

Abbildung 9.5: Ermittlung von Details mittels *SVERWEIS()*, *INDEX()* und *WENN()*

Sie möchten aus der gebuchten Variante und dem Buchungsdatum den Preis ermitteln. Dazu können Sie folgende Formel einsetzen:

```
=INDEX((D47:D50;E47:E50);SVERWEIS(C53;B47:C50;2;FALSCH);;WENN(C54<C52;1;2))
```

Sie teilen den Bereich der Preise in zwei Teile – vergessen Sie nicht die runden Klammern bei der Angabe des Bezugs –, ermitteln mit *SVERWEIS()* die Veranstaltungsnummer und nutzen diese als Zeilenindex. Einen Spaltenindex benötigen Sie nicht, da Sie in den Teilbereichen nur eine Spalte haben. In welchem Teilbereich nachzuschauen ist, entscheidet die *WENN()*-Funktion durch Vergleich des Buchungsdatums mit dem Stichtag.

Bei der vorliegenden Anordnung der Daten können Sie auch ohne die *INDEX()*-Funktion zum Ziel kommen, da der *SVERWEIS()* mit

```
=SVERWEIS(C53;B47:E50;WENN(C54<C52;3;4);FALSCH)
```

das gleiche Ergebnis liefert. Der *SVERWEIS()* ist aber auf die vorliegende Anordnung der Daten fixiert und versagt für den Fall, dass sich etwa Normalpreise und Rabatte (oder andere »gesplittete« Informationen) verteilt über das Arbeitsblatt oder gar in zwei verschiedenen Arbeitsblättern befinden. Das kann eventuell bei langen und umfangreicheren Datenlisten, die vielleicht noch aus Datenquellen per Abfrage gezogen werden, durchaus der Fall sein.

Mithilfe von *INDEX()* lassen sich spezielle Zellen in benannten Bereichen ansprechen. Dies ist vor allem dann wirkungsvoll, wenn die Bereiche von der Größe her dynamisch aufgebaut werden. Beispiele hierzu finden Sie im Abschnitt zur Funktion *BEREICH.VERSCHIEBEN()*.

Nehmen Sie an, Sie möchten die jeweils letzten Zellen rechts unten von zwei Bereichen addieren. Diese Bereiche sollen den Namen *NummerEins* und *NummerZwei* tragen und von ihrer Ausdehnung her beim Eintragen der Formel noch nicht bekannt sein. Dann kommen Sie sowohl mit der Bezugsversion

```
=INDEX((NummerEins;NummerZwei);ZEILEN(NummerEins);SPALTEN(NummerEins);1)
+INDEX((NummerEins;NummerZwei);ZEILEN(NummerZwei);SPALTEN(NummerZwei);2)
```

als auch mit der Matrixversion

```
=INDEX(NummerEins;ZEILEN(NummerEins);SPALTEN(NummerEins))+INDEX(NummerZwei;
ZEILEN(NummerZwei);SPALTEN(NummerZwei))
```

zum Zug.

Hinweis

Zellen aus benannten Bereichen

Tipp Holen Sie Daten aus externen Quellen, stehen diese automatisch in einem benannten Bereich. Die Einsatzmöglichkeiten von INDEX(), ZEILE(), SPALTE() und SVVERWEIS() sind dann sehr vielfältig und folgen den eben beschriebenen Mustern, unabhängig wie groß der Datenbereich nach jedem Aktualisieren der Abfrage ausfällt.

Siehe auch SVVERWEIS(), VERGLEICH(), VERWEIS(), WAHL(), WVERWEIS()



Die Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap09 in der Arbeitsmappe *Bezugsfunktionen.xlsx* auf dem Arbeitsblatt *Index*.

INDIREKT()



INDIRECT()

Syntax INDIREKT(*Bezug*;A1)

Definition Diese Funktion wandelt eine Zeichenkette in einen Bezug um.

Argumente *Bezug* (erforderlich) verlangt einen Ausdruck, der als Zeichenkette interpretiert werden kann. Die Funktion versucht, diese Zeichenkette in einen gültigen Bezug (Zellbezug oder Name eines benannten Bereichs) umzuwandeln.

A1 (optional) muss als Wahrheitswert auswertbar sein. Hier kann angegeben werden, ob das erste Argument in A1-Schreibweise identifiziert werden soll (*WAHR* oder weggelassen) oder in Z1S1-Schreibweise (*FALSCH*).

Hintergrund Kann das Argument *Bezug* nicht zu einem gültigen Bezug oder Namen ausgewertet werden, wird der Fehlerwert #BEZUG! zurückgegeben.

Die Verwendung externer Bezüge zu anderen Arbeitsmappen setzt voraus, dass diese geöffnet sind.

Praxiseinsatz Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

Zelladressen einsetzen Im zweiten Beispiel zur Funktion ADRESSE() ist beschrieben, wie die Zeichenkette, die durch eine Adress-Angabe in Form von ADRESSE(*Zeile*;Spalte) entsteht, durch INDIREKT() in einen Bezug umgewandelt und genutzt werden kann.

Hinweis In den Ausführungen zur Funktion INDEX() wurde deutlich, dass beispielsweise durch

```
INDEX(A:Z;27;2)
```

das Gleiche wie durch

```
INDIREKT(ADRESSE(27;2))
```

erreicht werden kann.

Investitionen beurteilen In der Investitionsanalyse kann durch Diversifikation (Streuung) verschiedener Investitionen zu bestimmten Risikostufen eine als optimal einzustufende Gesamtstrategie verfolgt werden. Dabei gilt: Je höher das geschätzte Risiko der Einzelinvestition, desto höher muss deren erwarteter Erfolg sein. In Abbildung 9.6 sehen Sie einen stark vereinfachten ersten Ansatz zum Vorgehen.

	A	B	C
26			
27		Risiko	Prozent
28		hoch	30%
29		mittel	20%
30		klein	10%
31			
32		Einsatz	20.000,00 €
33		Risiko	hoch
34		Rückkehr	26.000,00 €

Abbildung 9.6: Ansatz zur Investitionsanalyse

Der Anwender soll beim Benutzen der Tabelle nur die Wörter »hoch«, »mittel«, »klein« verwenden, um aus dem eingesetzten Kapital den erwarteten Rückfluss abzulesen. Das bereiten Sie durch die Vergabe der Namen *hoch* für Zelle C28 (das ist die erste der Zellen mit den zugeordneten prozentualen Renditekennziffern), *mittel* für C29 und *klein* für C30. Bei gegebenem Einsatz in C32 erhalten Sie durch

$=C32*(1+INDIREKT(C33))$

in C34 den erwarteten Rückfluss. Das funktioniert, weil INDIREKT() die durch den Anwender eingegebene Zeichenkette in C33 in einen Namen umwandelt, der in diesem Falle ein gültiger Bezug ist.

Mit einer Gültigkeitsregel (Registerkarte DATEN, Gruppe *Datentools/Datenüberprüfung*) lässt sich die Auswahl in C34 durch eine Auswahlliste (Dropdownfeld) noch komfortabler einrichten.

INDEX(), BEREICH.VERSCHIEBEN()

Die Beispiele finden Sie im Ordner `\Ms5-235\Kap09` in der Arbeitsmappe *Bezugsfunktionen.xlsx* auf dem Arbeitsblatt *Sonstige*.

Tip

Siehe auch



MTRANS()



MTRANS(*Matrix*)

Diese Funktion überführt eine gegebene Matrix (rechteckiger Bereich oder Matrixkonstante) in ihre transponierte Matrix.

Matrix (erforderlich) ist der Bezug auf einen rechteckigen Zellbereich oder eine Matrixkonstante.

MTRANS() sollte als Matrixformel (außer im Falle nur einer Zelle) in einen Bereich, der über genau so viele Zeilen bzw. Spalten verfügt, wie die Ausgangsmatrix Spalten bzw. Zeilen hat, eingesetzt werden (Abschluss mit `[Strg]+[↕]+[↶]`). Es erfolgt ja der Austausch von Zeilen und Spalten: die Zeilen der »alten« werden zu den Spalten der »neuen« Matrix. Dabei wird die erste Zeile zur ersten Spalte, die zweite Zeile wird zur zweiten Spalte usw.

Der Bereich, der die Ursprungsmatrix enthält, darf keine leeren Zellen enthalten. Andernfalls entsteht im Ergebnis der Fehlerwert #WERT!.

Syntax

Definition

Argumente

Hintergrund

Wichtig

Verwenden Sie im Zielbereich weniger Zeilen oder Spalten als notwendig, werden fehlende Inhalte einfach abgeschnitten. Verwenden Sie jedoch mehr, werden die überzähligen Zellen mit dem Fehlerwert #NV gefüllt.

Bei der Verwendung von Matrixkonstanten (das sind in geschweifte Klammern eingeschlossene Folgen von Zahlen oder Text), sollten Sie beachten, dass in einem Ausdruck wie

```
=MTRANS({11.12.13;21.22.23})
```

das Argument als zweizeilige Matrix mit drei Spalten (der Punkt ist deren Trennzeichen) interpretiert wird und das Ergebnis in eine Matrix mit drei Zeilen und zwei Spalten passt. Im Ausdruck

```
=MTRANS({1;2;3;4})
```

stellt das Argument jedoch eine einzelne Spalte dar, die in eine Zeile umgewandelt wird.

Hinweis Im Dialogfeld, welches nach dem Füllen der Zwischenablage mit Zellinhalten (Kopieren) und dem Aufruf von *START/Einfügen/Inhalte einfügen* erscheint, haben Sie ebenfalls die Möglichkeit, eine Matrix zu transponieren. Das Ergebnis wird allerdings vom Original getrennt, die Verwendung von MTRANS() hält die Verbindung zwischen Quelle und Ziel »am Leben«. Das Gleiche geschieht mit der entsprechenden EinfügetOption ab Excel 2010.

Praxiseinsatz Für verschiedene Aufgabenstellungen der linearen Algebra, der linearen Optimierung und der Entscheidungstheorie wird das Rechnen mit Matrizen und Vektoren benötigt und verwendet. Dabei sind Vektoren immer als Spaltenvektoren zu interpretieren. Das Skalarprodukt zweier Vektoren ist in diesem Fall die Matrixmultiplikation des transponierten ersten Vektors mit dem zweiten. Excel stellt hierfür neben MTRANS() auch MMULT() für die Matrixmultiplikation zur Verfügung.

Die Norm einer n-dimensionalen quadratischen Matrix A kann aus folgender Aufgabe definiert und ermittelt werden: Bestimme die Zahl, die sich aus dem Maximum aller Skalarprodukte zwischen Ax und x ergibt, wenn x alle Vektoren durchläuft, deren Norm (das ist die Wurzel aus dem Skalarprodukt von x und sich selbst) gleich 1 ist.

Eine solche Aufgabe ist eine Angelegenheit für den Solver (liegt als Add-In zu Excel vor und muss als solches aktiviert werden). Angenommen, die infrage kommende dreidimensionale Matrix steht in den Zellen von B16 bis D18. Einen Platzhalter für alle Vektoren x halten Sie in E16 bis E18 (veränderbare Zellen) bereit. In G16 notieren Sie das Skalarprodukt von x mit sich selbst (eine Nebenbedingung):

```
=MMULT(MTRANS(E16:E18);E16:E18)
```

In G18 schreiben Sie die Formel, die das Skalarprodukt von Ax mit x bestimmt (Zielzelle):

```
=MMULT(MTRANS(MMULT(B16:D18;E16:E18));E16:E18)
```

Rufen Sie nun bei markierter Zelle G18 den Solver auf (*DATEN/Analyse/Solver*) auf.

Im Dialogfeld wie in Abbildung 9.7 haben Sie durch Hinzufügen einer Nebenbedingung alle Vorbereitungen zum Lösen getroffen. Wird eine Lösung durch den Solver gefunden, ist die Wurzel aus dieser gleich der gesuchten Norm.

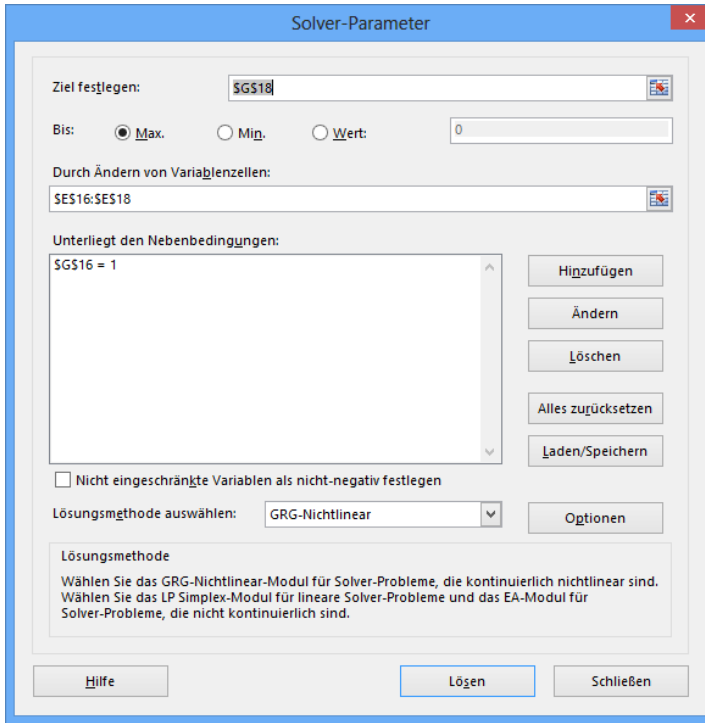


Abbildung 9.7: Normbestimmung von Matrizen mit dem Solver (der Solver präsentiert sich ab Excel 2010 mit einer neuer Oberfläche)

MDET(), MINV(), MMULT()

Die Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner `Ms5-235\Kap09` in der Arbeitsmappe *Bezugsfunktionen.xlsx* auf dem Arbeitsblatt *MTRANS*.

Siehe auch



PIVOTDATENZUORDNEN()

Diese Funktion wird in Kapitel 12 besprochen.

RTD() RTD()

RTD(*ProgID*; *Server*; *Thema1*; [*Thema2*]; ...)

Diese Funktion ruft Echtzeitdaten von einem Programm ab, das derartige Daten liefert.

ProgID (erforderlich) ist eine Zeichenkette, die den Klassennamen (*ProgID*) der für die Echtzeitdaten verantwortlichen Klasse in einer registrierten DLL-Datei angibt.

Server (optional) bezeichnet den Rechner, auf welchem die DLL-Datei installiert wurde. Das kann der lokale Rechner, aber auch ein Rechner im Netz sein.

Syntax

Definition

Argumente

Thema1, *Thema2*, ... sind Parameter, die die angeforderten Echtzeitdaten in Abhängigkeit vom konkreten Programm charakterisieren. Nur der erste ist nicht optional.

Hinweis Zeichenketten werden in doppelten Anführungszeichen erwartet, sofern es sich nicht um Zellbezüge handelt, in denen Text steht.

Hintergrund Versuchen Sie diese Funktion zu verwenden, ohne dass ein gültiger Echtzeitdaten-Server gefunden werden kann, wird der Fehlerwert #NV zurückgegeben.

Befindet sich der Server auf dem lokalen Rechner, kann das Argument weggelassen (die Stelle zwischen den Semikola bleibt leer) oder eine leere Zeichenkette angegeben werden. Verwenden Sie diese Funktion in VBA, ist die Angabe der leeren Zeichenkette (auch *vbNullString*) Pflicht.

Hinweis Ist der Server im Netz installiert, muss der Benutzer der Arbeitsmappe entsprechende Zugriffsrechte auf COM+-Komponenten des sendenden Rechners haben. Auch in diesem Fall ist die DLL-Datei lokal zu registrieren.

Die Entwicklung ist nur mit Programmen möglich, die DLL-Dateien auf COM-Basis erstellen können. Stellvertretend sei hier die Microsoft Visual Studio-Reihe genannt.

Praxiseinsatz Innerhalb der Beispieldateien zum Buch finden Sie eine Datei namens *exampleRTD.dll*, die Sie mithilfe der Befehlszeile *regsvr32.exe "Laufwerk:\Vollständiger Pfad\Dateiname"* registrieren können² (der zusätzliche Parameter */u* hebt die Registrierung wieder auf). Diese Datei liefert die Echtzeitdaten der jeweiligen Stunde, Minute und Sekunde.

Einsetzen können Sie den Server nun durch die Formeln

```
=RTD("RealTime.c1sRTDS";;"Stunde")
```

```
=RTD("RealTime.c1sRTDS";;"Minute")
```

bzw.

```
=RTD("RealTime.c1sRTDS";;"Sekunde")
```

Wichtig Nach der Installation der DLL-Datei werden mögliche Echtzeitdaten alle zwei Sekunden abgerufen. Wollen Sie das ändern, ist das nur mit VBA möglich. Führen Sie dazu irgendwann die folgende Prozedur aus:

```
Sub ChangeThrottleInterval ()
    Application.RTD.ThrottleInterval = 1000
End Sub
```

Im konkreten Fall wird dadurch die Taktzeit auf eine Sekunde verringert – ab jetzt und in der Zukunft.

Erfahrene Nutzer können auch nach dem Eintrag *RTDThrottleInterval* in der Windows-Registrierung suchen und dort den Wert anpassen.

Tipp Mithilfe der Funktion ZEIT() haben Sie eine ständig laufende Uhr auf Ihrem Arbeitsblatt:

```
=ZEIT(RTD("RealTime.c1sRTDS";;"Stunde");RTD("RealTime.c1sRTDS";;"Minute");RTD("RealTime.c1sRTDS";;"Sekunde"))
```

² Sie benötigen hierzu Administratorenrechte. Dazu öffnen Sie eine Eingabeaufforderung als Administrator und schreiben dort die genannte Befehlszeile hinein.

Anders als laufende Makros behindert der Echtzeitserver nicht das weitere Bearbeiten des aktuellen Arbeitsblatts.

Keine

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap09` in der Arbeitsmappe `exampleRTD.xlsm`. In diesem Ordner befindet sich auch die Laufzeitbibliothek `exampleRTD.dll`.

Siehe auch



SPALTE() COLUMN()

SPALTE(*Bezug*)

Diese Funktion liefert die (globale) Spaltennummer eines Bezugs.

Bezug (optional) muss als Bezug auf eine einzelne Zelle oder einen rechteckigen Zellbereich auswertbar sein.

Lassen Sie das Argument *Bezug* weg, wird die Spaltennummer geliefert, die der Zelle entspricht, in der die Funktion steht.

Ist *Bezug* ein Zellbereich (der auch durch einen Namen angegeben werden kann), kann die Funktion in Matrixformeln Verwendung finden (Abschluss mit `[Strg]+[↕]+[↔]`). Ist die spaltenweise Ausdehnung des Zielbereichs dabei kleiner als die spaltenweise Ausdehnung des Arguments, werden die Informationen der fehlenden Zellen am Ende abgeschnitten. Ist der Zielbereich in dieser Ausdehnung größer, entsteht in den überzähligen Zellen der Fehlerwert `#NV`.

Sie möchten in einer Zeile die Zahlen addieren, deren Spaltennummern gerade (ungerade) sind.

Leider schlägt der Versuch, hier die Funktion `ISTGERADE()` (bzw. `ISTUNGERADE()`) in gleicher wie der folgenden, aber kürzerer Form einzusetzen, fehl, da diese Funktionen keine mehrzelligen Bezüge akzeptieren (Fehlerwert `#WERT!`).

Eine Zahl ist gerade, wenn die Formel

$$(2 * \text{GANZZAHL}(A1/2) - A1) = 0$$

den Wahrheitswert `WAHR` liefert. Im Fall von

$$((2 * \text{GANZZAHL}(A1/2) - A1) <> 0) = \text{WAHR}$$

ist die Zahl nicht gerade.

Das machen Sie sich zunutze. Angenommen, Ihre Zahlenreihe steht in der Zeile `B21` bis `E21`. Dann liefert die Matrixformel

$$\{= \text{SUMME}((\text{GANZZAHL}(\text{SPALTE}(\text{B21}: \text{E21})/2) * 2 - \text{SPALTE}(\text{B21}: \text{E21}) = 0) * \text{B21}: \text{E21})\}$$

die Summe über alle Spalten mit geraden Spaltennummern und

$$\{= \text{SUMME}((\text{GANZZAHL}(\text{SPALTE}(\text{B21}: \text{E21})/2) * 2 - \text{SPALTE}(\text{B21}: \text{E21}) <> 0) * \text{B21}: \text{E21})\}$$

die Summe über alle mit ungeraden Spaltennummern. Das funktioniert deshalb, weil die Wahrheitswerte `WAHR` als 1 und `FALSCH` als 0 interpretiert werden.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Hinweis

Siehe auch SPALTEN(), ZEILE()



Die Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap09 in der Arbeitsmappe *Bezugsfunktionen.xlsx* auf dem Arbeitsblatt *Spalten-Zeilen*.

SPALTEN() COLUMNS()

Syntax SPALTEN(*Matrix*)

Definition Diese Funktion gibt die Anzahl der Spalten einer Matrixkonstanten oder eines rechteckigen Bezugs zurück.

Argumente *Matrix* (erforderlich) erwartet eine Matrixkonstante oder einen Bezug auf einen rechteckigen Zellbereich.

Hintergrund Versuchen Sie, einen nicht rechteckigen (mehnteiligen) Bereich als Argument zu verwenden, erhalten Sie die Fehlermeldung »zu viele Argumente«. Schließen Sie diese Argumente in zusätzliche runde Klammern ein, entsteht der Fehlerwert #BEZUG!. Wird der Bereich durch Schnittmengen – diese werden durch ein Leerzeichen zwischen Zellbezügen wie in

=SPALTEN(B2:D4 E2:E4)

gebildet – definiert und ist diese Schnittmenge leer, ergibt sich der Fehlerwert #NULL!.

Matrixkonstanten sind Folgen von Zahlen oder Text, die Sie selbst in geschweifte Klammern einschließen müssen. Dabei wird ein Punkt als Trennzeichen zwischen Zeilen und Spalten verwendet. Der einfache Ausdruck {1;2;3;4} wird als eine einzelne Spalte interpretiert, wie das Ergebnis von

=SPALTEN({1;2;3;4})

zeigt. Dem gegenüber ist {11.12.13;21.22.23} dreispaltig, was durch das Ergebnis von

=SPALTEN({11.12.13;21.22.23})

dokumentiert werden kann.

Praxisbezug Mithilfe dieser Funktion sowie der verwandten Funktion ZEILEN() können Sie einfach auf ausgesuchte Zellen eines benannten Bereichs zugreifen. Besonders wirkungsvoll ist das dann, wenn der Bereich dynamischen Veränderungen unterliegt. Beispiele hierzu finden Sie im Abschnitt zur Funktion BEREICH.VERSCHIEBEN().

Angenommen, der infrage kommende Bereich trägt den Namen *MeinBereich*. Dann liefert

=INDEX(MeinBereich;ZEILEN(MeinBereich);SPALTEN(MeinBereich))

einen Bezug zur letzten unteren rechten Zelle des Bereichs.

Siehe auch SPALTE(), ZEILEN()



Das Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap09 in der Arbeitsmappe *Bezugsfunktionen.xlsx* auf dem Arbeitsblatt *Spalten-Zeilen*.

SVERWEIS()



VLOOKUP()

SVERWEIS(*Suchkriterium*; *Matrix*; *Spaltenindex*; *Bereich_Verweis*)

Diese Funktion sucht in der am weitesten links gelegenen Spalte eines rechteckigen Bezugs oder einer Matrixkonstanten nach einem Wert. Entsprechend der übergebenen Argumente wird beim Finden dieses oder eines ihn vertretenden Werts in der durch den Bezug definierten Tabelle (Liste) eine entsprechende Anzahl von Schritten nach rechts gegangen und der dortige Eintrag zurückgegeben.

Suchkriterium (erforderlich) muss als Text, Zahl oder Wahrheitswert auswertbar sein. Es ist der Wert, nach welchem in der ersten Spalte von links der *Matrix* gesucht wird.

Bei der Suche nach Text wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Matrix (erforderlich) ist ein Bezug auf einen rechteckigen Zellbereich (dieser kann auch durch einen Namen benannt sein) bzw. eine Matrixkonstante (das ist eine in geschweifte Klammern geschriebene Folge von Zahlen oder Text).

Spaltenindex (erforderlich) muss als positive ganze Zahl auswertbar sein, die nicht größer als die Anzahl der Spalten im Argument *Matrix* ist. Es gibt die Spaltennummer der Spalte an, aus der die durch die Funktion gelieferten Informationen stammen.

Bereich_Verweis (optional) verlangt einen Wahrheitswert. Dieser bestimmt, ob in der ersten Spalte nach einer genauen Übereinstimmung mit dem *Suchkriterium* gesucht werden soll (Argument ist gleich *FALSCH*) oder nicht (Argument ist gleich *WAHR* oder weggelassen).

Prinzipiell kann von zwei Situationen ausgegangen werden:

Die erste Situation: Sie suchen in der linken Spalte nach einem bestimmten Begriff oder einer bestimmten Zahl und akzeptieren nur diese. Dann ist es nicht von Bedeutung, ob die Tabelle nach der ersten Spalte sortiert ist oder nicht. Wichtig ist nur, dass Sie in einer nicht sortierten Tabelle unbedingt den Parameter *Bereich_Verweis* als *FALSCH* deklarieren müssen. Tun Sie das nicht oder lassen den Parameter weg, wird Ihr Suchbegriff in aller Regel nicht gefunden, selbst wenn er in der linken Spalte tatsächlich vorhanden ist (Ausnahmen sind zufällige »günstige« Situationen). Die Ursache liegt darin, dass Excel die Suche von unten beginnt. Stößt es dabei auf einen Eintrag, der in lexikalischer oder numerischer Ordnung vor dem gesuchten steht, wird das so interpretiert, als ob »am gesuchten bereits vorbeigegangen worden sein muss«. Da er nicht gefunden wurde, ist er also auch nicht da. In diesen Situationen wird dann der Wert als Stellvertreter als gefunden genommen, der in der genannten Ordnung – in Abwärtsrichtung gesehen – als nächster folgt. Diese Interpretation durch Excel ist in der Regel nicht sachgemäß und kann durch die Angabe von *FALSCH* als viertem Parameter vermieden werden – steht der Wert nicht in der ersten Spalte, erscheint der Fehlerwert #NV als Ergebnis.

Die zweite Situation: Sie sind nicht an einer genauen Übereinstimmung des gefundenen Werts in der ersten Spalte interessiert. Das wird bei Texten sicher nur in seltenen Fällen Ihr Wunsch sein. Aber bei Zahlen, genauer gesagt Intervall-Grenzen, wie sie bei Rabattstaffeln, Zinsstaffeln, Einkommensteuerbeträgen und anderen »Einordnungsproblemen« vorkommen, ist die Lage ganz anders. Wichtig ist nun, dass Ihre Liste (Tabelle) tatsächlich nach der ersten Spalte geordnet sein muss, die Intervallgrenzen also in aufsteigender (numerischer) Ordnung vorliegen. Wollen Sie in der Tat eine solche Grenze finden, verwenden Sie als vier-

Syntax

Definition

Argumente

Hinweis

Hintergrund

ten Parameter den Wahrheitswert *FALSCH*. Setzen Sie diesen Parameter als *WAHR* an oder lassen ihn weg, wird Ihnen Excel (und das ist dann sachgemäß und gewünscht) die Intervall-Grenze liefern, die in absteigender Richtung dem gesuchten Wert am nächsten kommt.

Hinweis Sie können das Argument *Matrix* auch als sogenannte Matrixkonstante übergeben. Dabei handelt es sich um in geschweifte Klammern eingeschlossene Zeichen- oder Zahlenfolgen. Zeilen werden durch ein Semikolon getrennt, Spalten durch einen Punkt. So ist {1;2;3;4} ein einspaltiges Gebilde, bestehend aus vier Zeilen. Bei {11.12.13;21.22.23} handelt es sich um eine dreispaltige Matrix, bestehend aus zwei Zeilen. Auf Letztere kann durch SVERWEIS() etwa mittels

=SVERWEIS(B27;{11.12.13;21.22.23};3)

zugriffen werden (steht in B27 der Wert 21, wird 23 zurückgegeben).

Diese Art der Verwendung von SVERWEIS() dürfte jedoch kaum Anwendung finden, da das Eingeben der Matrizen aufwendig, fehleranfällig und sehr unflexibel ist.

Praxeinsatz Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

Kombination mit anderen Funktionen Im zweiten Beispiel zur INDEX()-Funktion wurde die Funktion SVERWEIS() eingesetzt, um eine genaue Übereinstimmung des gesuchten Werts zu finden. Weitere Beispiele finden Sie im Abschnitt zu den Funktionen ISTFEHLER() und ISTNV().

Ein Beispiel zur Verwendung von Intervallen bei Rabatt-Staffeln finden Sie im Tipp zum dritten Beispiel der WENN()-Funktion.

SVERWEIS() findet leider nur Werte rechts von der Suchspalte. Wollen Sie Werte links davon ermitteln, können Sie dies im Falle einer geforderten genauen Übereinstimmung von *Suchkriterium* und gefundenem Wert durch eine Kombination der Funktionen INDEX() und VERGLEICH() erledigen.

Angenommen, Sie haben eine Liste mit einer Zuordnung von Ziffern zu Text wie in Abbildung 9.8.

	A	B	C	D	E	F
30						
31		Ziffer	Text			
32		-1	klein		gesuchter Text	klein
33		0	mittel		ermittelte Ziffer	-1
34		1	groß			

Abbildung 9.8: Mithilfe von INDEX() und VERGLEICH() den SVERWEIS() simulieren

Der Bereich von B32 bis B34 hat den Namen *Bereich* erhalten. Sie möchten die zum Suchtext »klein« gehörende Ziffer ermitteln. Umgekehrt ist es mit SVERWEIS() kein Problem:

=SVERWEIS(-1;Bereich;2;FALSCH)

Die Originalaufgabe ist ein wenig komplizierter. Eine mögliche Lösung zeigt die folgende Formel:

=INDEX(Bereich;VERGLEICH(F32;BEREICH.VERSCHIEBEN(Bereich;0;1;;1);0);1)

Durch `BEREICH.VERSCHIEBEN()` definieren Sie die zweite Spalte von *Bereich* als die Spalte, in der gesucht wird. Der `VERGLEICH()` liefert dann die (lokale) Zeile in *Bereich*, in der der Wert aus *F32* gefunden wurde. `INDEX()` nimmt diese Zeilennummer und Sie geben noch die Spaltennummer 1 hinzu.

`INDEX()`, `VERGLEICH()`, `VERWEIS()`, `WVERWEIS()`

Einige Tests zu den Hintergrundinformationen und das Beispiel finden Sie im Ordner `Ms5-235\Kap09` in der Arbeitsmappe *Verweise.xlsx* im Arbeitsblatt *SVERWEIS*.

Siehe auch



VERGLEICH() MATCH()

`VERGLEICH(Suchkriterium;Suchmatrix;Vergleichstyp)`

Diese Funktion sucht Text, Zahlen oder Wahrheitswerte innerhalb eines Bezugs oder einer Matrix und gibt deren Position zurück.

Suchkriterium (erforderlich) ist der gesuchte Wert und kann eine Zahl, eine Zeichenfolge oder ein Wahrheitswert bzw. ein Bezug auf eine Zelle sein, die einen solchen Wert enthält.

Suchmatrix (erforderlich) ist der Bezug auf einen einzeiligen oder einspaltigen Zellbereich bzw. eine Matrixkonstante (in geschweifte Klammern geschriebene Folge von Werten) und bestimmt den Ort, wo nach dem im ersten Argument angegebenen Wert gesucht werden soll.

Vergleichstyp (optional) muss sich zu einer Zahl, die die Werte -1 , 0 oder 1 hat, auswerten lassen. Das Argument gibt an, unter welchen Voraussetzungen und wie nach **Suchkriterium** in der **Suchmatrix** gesucht wird. Wird es weggelassen, arbeitet Excel so, als ob es gleich 1 gesetzt wurde.

Das Argument **Vergleichstyp** muss passend zur Ausgangssituation gewählt werden und dabei gleichzeitig den Erwartungen an die Funktion entsprechen:

- ▶ Ist **Vergleichstyp** gleich -1 , gibt `VERGLEICH()` die Position des kleinsten Werts zurück, der größer oder gleich **Suchkriterium** ist. Die Elemente der **Suchmatrix** sollten dazu sinnvollerweise in absteigender Reihenfolge angeordnet sein: Zuerst kommen die Wahrheitswerte *WAHR* und *FALSCH*, dann die Buchstaben in umgekehrter alphabetischer Reihenfolge und schließlich die Zahlen in absteigender Anordnung.
- ▶ Ist **Vergleichstyp** gleich 0 , gibt `VERGLEICH()` die Position des ersten Werts zurück, der gleich **Suchkriterium** ist. Die Elemente der **Suchmatrix** dürfen dann in beliebiger Reihenfolge angeordnet sein.
- ▶ Ist **Vergleichstyp** gleich 1 , gibt `VERGLEICH()` die Position des größten Werts aus der **Suchmatrix** zurück, der kleiner oder gleich **Suchkriterium** ist. Dazu sollten die Elemente der **Suchmatrix** sinnvollerweise in aufsteigender Reihenfolge angeordnet sein. Zahlen kommen vor den Buchstaben (in alphabetischer Reihenfolge), danach die Wahrheitswerte *FALSCH* und *WAHR*.

`VERGLEICH()` unterscheidet sich also von den Funktionen mit dem Wort »Verweis« im Namen (wie `SVERWEIS()`) dadurch, dass Positionen bestimmt werden und nicht die Werte selbst. Die Ausgangssituation wie im Falle **Vergleichstyp** = -1 ist in `SVERWEIS()` leider gar nicht vorgesehen.

Syntax

Definition

Argumente

Hintergrund

Wird kein übereinstimmender Wert gefunden, gibt die Funktion den Fehlerwert #NV zurück. Dieser Fehlerwert wird auch geliefert, wenn das Argument *Suchkriterium* nicht der Forderung nach einer Zeile bzw. einer Spalte entspricht. In den Situationen, in denen die Wahl des Parameters *Vergleichstyp* nicht der Ordnung der *Suchmatrix* entsprechend erfolgt, kann es ebenfalls zu diesem Fehlerwert kommen, obgleich der gesuchte Wert in der Liste vorhanden ist.

In Listen mit Zeichenketten wird es sicher die Regel sein, dass *Vergleichstyp* = 0 gewählt wird. Andernfalls kann es zu kuriosen Rückgabewerten kommen, falls ein Wert nicht gefunden wird. In Listen mit Zahlen, die Intervallgrenzen angeben (Rabattstaffeln, Zinsstaffeln, Punkteinteilungen und andere Einteilungsaufgaben), ist eine steigende bzw. fallende Anordnung von *Suchmatrix* aus der Aufgabenstellung heraus garantiert.

Eine Besonderheit ist im Falle von *Vergleichstyp* = 0 besonders wirkungsvoll: Wird für Suchkriterium eine Zeichenfolge angegeben, darf diese die Platzhalter * (Sternchen) für eine beliebige Zeichenfolge und ? (Fragezeichen) für ein einzelnes Zeichen enthalten.

VERGLEICH() unterscheidet beim Abgleichen von Zeichenfolgen (Texten) nicht zwischen Groß- und Kleinbuchstaben.

Praxiseinsatz

Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

Suchen in mehreren Spalten

Mit wenigen Handgriffen lässt sich mithilfe dieser Funktion eine mehrspaltige Suche realisieren. Angenommen, Sie haben eine (hier stark verkürzte) Preisliste für Oberbekleidung wie in Abbildung 9.9.

	A	B	C	D	E	F
43						
44			Artikel	Farbe	Preis	
45			Hemd	blau	44,95 €	
46			Hemd	gelb	49,95 €	
47			Hose	rot	89,95 €	
48			Hose	schwarz	84,95 €	
49						
50		gesucht:	Hose	schwarz		
51		Position:	4		Preis:	84,95 €
52						
53		gesucht:	Hose, rot			
54		Position:	3		Preis:	89,95 €

Abbildung 9.9: Mehrspaltiges Suchen in einer »symbolischen« Preisliste

Zuerst ermitteln Sie die Position der gesuchten »schwarzen Hose« durch

```
{=VERGLEICH(C50 & D50;C45:C48 & D45:D48;0)}
```

(das ist eine Matrixformel und deshalb mit der Tastenkombination **Strg** + **↕** + **↶** abzuschließen). Mit dem &-Zeichen gelingt es Ihnen, sowohl das *Suchkriterium* zu verknüpfen als auch aus den zwei Suchspalten eine einzige zu machen. Die INDEX()-Funktion ermittelt Ihnen aus der gefundenen Position den gesuchten Wert:

```
=INDEX(E45:E48;C51)
```

Natürlich könnten Sie beide Formeln zu einer einzigen für den gesuchten Preis vereinen:

```
{=INDEX(E45:E48;VERGLEICH(C50 & D50;C45:C48 & D45:D48))}
```

Wollen Sie das Suchkriterium in einer einzigen Zelle halten (C53 in Abbildung 9.9), kann die Formel zur Positionsbestimmung so aussehen:

```
{=VERGLEICH(C53;C45:C48 & " " & D45:D48;0)}
```

(auch das ist eine Matrixformel). Sie haben hier die Suchspalten entsprechend des verwendeten Musters in C53 miteinander kombiniert.

Durch die Verwendung von Platzhaltern im gesuchten Wert gelingt es durch entsprechende Formeln auch Artikel zu finden, von denen man nur einen Teil des Namens bzw. der Farbe kennt. Schreiben Sie in C58 »Ho« und in D58 »ro«, wird durch die Matrixformel

```
{=VERGLEICH("*" & C58 & "*" & D58 & "*";C45:C48 & D45:D48;0)}
```

auch die Zeile gefunden, in der die »rote Hose« steht.

Da die Formeln durchaus kompliziert werden können, sollten Sie bei solchen Aufgabenstellungen auch die Alternative des Filterns mit dem Spezialfilter ins Kalkül ziehen. Beim Kopieren der gefundenen Listenausschnitte (anstelle des Filterns am gleichen Ort) beachten Sie jedoch, dass eine solche Kopie vom Original losgelöst existiert und sich nicht mit diesem aktualisiert.

Hinweis

Kreuztabellen (das ist die einfachste Form einer PivotTable) begegnen uns im Leben sehr oft: Fahrpläne an Bus- oder Straßenbahnhaltstellen, Entfernungangaben zwischen Städten in Almanachen und Taschenkalendern, Logarithmentabellen und vieles mehr.

**Kreuztabellen
in der Praxis**

Angenommen, Sie haben einen »Telefon-Navigator« in Tabellenform vorliegen, der Sie über den günstigsten Anbieter zu verschiedenen Tageszeiten informiert. Die Abbildung 9.10 zeigt einen fiktiven Ausschnitt.

	A	B	C	D	E
73					
74			Anbieter 1	Anbieter 2	Anbieter 3
75		00:00	0,08 €	0,27 €	0,18 €
76		06:00	0,18 €	0,07 €	0,14 €
77		09:00	0,27 €	0,09 €	0,18 €
78		18:00	0,18 €	0,10 €	0,14 €
79		21:00	0,11 €	0,27 €	0,18 €
80					
81		gesucht:	07:00		
82		gefunden:	2		
83		Minimum:	0,07 €		
84		Anbieter:	Anbieter 2		

Abbildung 9.10: Navigieren im Tarif-Dschungel

Das Finden einer Uhrzeit in der linken Spalte sieht aus wie eine Aufgabe für SVERWEIS(). Nur, in welcher Spalte steht das Ergebnis? Mithilfe von VERGLEICH() ermitteln Sie die gesuchte Zeile (für 7 Uhr ist das hier die Zeile 2), das Minimum findet die Funktion MIN() im Zusammenspiel mit BEREICH.VERSCHIEBEN():

```
=VERGLEICH(C81;B75:B79;1)
```

und

```
=MIN(BEREICH.VERSCHIEBEN(C75:E79;C82-1;0;1;3))
```

Wiederum könnten Sie beide Formeln zu einer zusammenfassen.

Den zu einem minimalen Tarif gehörenden Anbieter ermitteln Sie nun aus

```
=INDEX(C74:E74;1;VERGLEICH(C83;BEREICH.VERSCHIEBEN(C75:E79;C82-1;0;1;3);0))
```

BEREICH.VERSCHIEBEN() bringt Sie dabei in die richtige Zeile (beachten Sie, dass deren Nummerierung bei 0 beginnt), C83 enthält das ermittelte Minimum, nach welchem Sie nun durch VERGLEICH() suchen, und INDEX() nimmt die Informationen, um in der Kopfzeile nachzuschauen.

Tipp Die Markierung der Zelle mit dem günstigsten Tarif erledigen Sie durch bedingte Formatierung. Nehmen Sie diese in C75 durch die Formel

```
=UND(C74=MIN(BEREICH.VERSCHIEBEN($C$74:$E$78;VERGLEICH($C$80;$B$74:$B$78;1)-1;0;1;3));(ZEILE(C74)-ZEILE($C$74)+1)=$C$81)
```

vor (es wird geprüft, ob in der Zelle das Minimum steht und ob die Zeile die richtige ist), wählen eine entsprechende Füllfarbe und kopieren anschließend die Formatierung mithilfe des Symbols *Format übertragen*.

Siehe auch INDEX(), SVERWEIS(), WVERWEIS()



Die Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap09 in der Arbeitsmappe *Verweise.xlsx* auf dem Arbeitsblatt *VERGLEICH*.

VERWEIS() LOOKUP()

Syntax Syntax 1 (Vektorversion):

```
VERWEIS(Suchkriterium; Suchvektor; Ergebnisvektor)
```

Syntax 2 (Matrixversion):

```
VERWEIS(Suchkriterium; Matrix)
```

Definition Diese Funktion durchsucht die Werte einer Matrix (Bezug auf einen rechteckigen Bereich bzw. eine Matrixkonstante). Besteht die Matrix aus nur einer Spalte bzw. nur einer Zeile, kommt die Vektorversion zum Ansatz.

Die Vektorversion sucht in einem Vektor nach einem bestimmten Wert und gibt einen Wert aus derselben Position in einem zweiten Vektor zurück.

Die Matrixversion sucht in der ersten Zeile oder Spalte einer Matrix nach dem angegebenen Wert und gibt einen Wert aus derselben Position in der letzten Zeile oder Spalte derselben Matrix zurück.

Argumente Syntax 1 (Vektorversion):

Suchkriterium (erforderlich) muss sich als eine Zahl, eine Zeichenkette oder ein Wahrheitswert auswerten lassen. Es kann also auch ein Bezug auf eine Zelle mit einem solchen Wert sein.

Suchvektor (erforderlich) ist der Bezug auf einen Zellbereich, der nur eine Zeile oder Spalte enthält. Zulässige Elemente von Suchvektor sind Zeichenfolgen (Texte), Zahlen oder Wahrheitswerte.

Ergebnisvektor (erforderlich) ist analog zu *Suchvektor* der Bezug auf einen Zellbereich, der nur eine Zeile oder Spalte enthält. Dieses Argument muss genau so viele Elemente umfassen wie das Argument *Suchvektor*. Lassen Sie dieses Argument weg, arbeiten Sie automatisch mit der Matrixversion.

Syntax 2 (Matrixversion):

Suchkriterium (erforderlich) muss sich auch in dieser Syntaxversion als ein Wert (Zahl, Text oder Wahrheitswert) auswerten lassen.

Matrix (erforderlich) ist der Bezug auf einen rechteckigen Zellbereich, der entweder Text, Zahlen oder Wahrheitswerte enthält.

Statt der Angabe von Zellbezügen können Sie auch Matrixkonstanten verwenden. Das sind in geschweifte Klammern eingeschlossene Folgen von Werten. Das Semikolon trennt hierbei Zeilen voneinander, ein Punkt trennt die Spalten. So ist $\{1;2;3;4\}$ ein einspaltiger Vektor, bestehend aus vier Zeilen. Bei $\{11.12.13;21.22.23\}$ handelt es sich um eine dreispaltige Matrix, bestehend aus zwei Zeilen.

Hinweis

In beiden Syntaxversionen ist es wichtig, dass die Werte in den Bereichen, in denen gesucht wird, in aufsteigender Reihenfolge angeordnet sind: zuerst die Zahlen, dann die Buchstaben in alphabetischer Reihenfolge, dann die Wahrheitswerte *FALSCH* und *WAHR*. Ist das nicht der Fall, gibt VERWEIS() möglicherweise einen unerwarteten und sogar falschen Wert zurück. Anders als bei den Funktionen SVERWEIS(), WVERWEIS() und VERGLEICH() haben Sie keine Möglichkeit, die Art der Suche weiter zu spezifizieren.

Hintergrund

Bei Zeichenfolgen (Texten) wird nicht zwischen Groß- und Kleinbuchstaben unterschieden.

Kann die VERWEIS()-Funktion in der Vektorversion keinen Wert finden, der mit dem jeweiligen Suchkriterium übereinstimmt, verwendet sie den größten Wert in *Suchvektor*, der kleiner oder gleich *Suchkriterium* ist. Ist *Suchkriterium* kleiner als der kleinste Wert in *Suchvektor*, gibt VERWEIS() den Fehlerwert #NV zurück.

In der Matrixversion bestimmt die Ausdehnung der Matrix, wo gesucht wird. Hat die Matrix mehr Spalten als Zeilen, wird *Suchkriterium* in der ersten Zeile gesucht. Andernfalls wird *Suchkriterium* in der ersten Spalte gesucht. Kann kein übereinstimmender Wert gefunden werden, wird der größte aller Werte genommen, der kleiner oder gleich *Suchkriterium* ist. Ist *Suchkriterium* kleiner als alle Vergleichswerte, meldet die Funktion den Fehlerwert #NV.

Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

Praxiseinsatz

Die Matrixversion von VERWEIS() ist den Funktionen WVERWEIS() und SVERWEIS() sehr ähnlich. Der Unterschied besteht darin, dass WVERWEIS() die erste Zeile, SVERWEIS() die erste Spalte und VERWEIS() diejenige Spalte oder Zeile von *Matrix* mit dem *Suchkriterium* abgleicht, die sich entsprechend den Dimensionen von *Matrix* ergibt. Mit WVERWEIS() und SVERWEIS() können Sie durch die Angabe eines Index in eine bestimmte Zeile oder Spalte springen und sind dadurch flexibler als bei der Verwendung von VERWEIS() in der Matrixversion, welche immer den Wert, der in der letzten Zeile bzw. Spalte steht, wählt.

Damit lassen sich denkbare Beispiele für die Matrixversion ausnahmslos durch die anderen Funktionen umsetzen.

Für den Fall variabler Listen (etwa die Länge ändert sich bei der Aktualisierung im Falle einer externen Datenabfrage) wird in aller Regel die Spaltenzahl kleiner als die Zeilenzahl sein. Somit gelingt das Suchen in den Überschriften nicht, um beispielsweise Informationen der letzten Zeile zu erhalten. Auch hier stehen dann mit WVERWEIS() in Kombination mit ZEILEN() oder mit VERGLEICH() in Kombination mit INDEX() zielgerichtete Alternativen bereit.

Excel sollte im Allgemeinen nicht als Ersatz für ein Datenbank-Managementsystem (wie Access oder gar der Microsoft SQL Server) verwendet werden. Bewahren Sie in »Ausnahmefällen« dennoch Datenlisten in Arbeitsblättern auf, kann die Vektorversion anders als SVERWEIS() eine Suche über mehrere nicht zusammenhängende Bereiche, die sich vielleicht sogar auf verschiedenen Arbeitsblättern befinden, umsetzen. Einschränkende Voraussetzung dazu ist, dass die Suchvektoren geordnet sein müssen und die Ergebnisvektoren von den Positionen her passend zu den Suchvektoren aufgebaut werden.

Siehe auch INDEX(), SVERWEIS(), VERGLEICH(), WVERWEIS()



Ein Beispiel zu den alternativen Verwendungsmöglichkeiten finden Sie im Ordner \Ms5-235\Kap09 in der Arbeitsmappe *Verweise.xlsx* auf dem Arbeitsblatt *VERWEIS*.

WAHL() CHOOSE()

Syntax WAHL(*Index*; *Wert1*; *Wert2*; ...)

Definition Diese Funktion verwendet einen Index, um einen Wert aus der Liste der Werteargumente zurückzugeben.

Argumente *Index* (erforderlich) gibt an, welcher Wert aus den folgenden Wertargumenten ausgewählt werden soll.

Wert1; *Wert2*; ... (das erste Wertargument ist erforderlich) sind durch Semikola getrennte Werteargumente. Dies können Zahlen, Zellbezüge, definierte Namen, Formeln, Funktionen oder Text sein. Die Anzahl ist beschränkt auf 254 Argumente.

Hintergrund *Index* muss sich als eine ganze Zahl zwischen 1 und 254 auswerten lassen. Damit kommen also auch Formeln oder ein Bezug auf eine Zelle infrage, die eine solche Zahl liefern.

Ist *Index* kleiner als 1 oder größer als die Anzahl der Werteargumente in der Liste, gibt WAHL() den Fehlerwert #WERT! zurück.

Ist *Index* eine gebrochene Zahl, werden die Nachkommastellen abgeschnitten, bevor die Auswahl aus der Liste vorgenommen wird.

Sie können WAHL() in einer Matrixformel einsetzen (Abschluss mit Strg + ↵ + ←), wenn Sie *Index* als Matrix angeben. Jedoch ist hier Vorsicht geboten, um nicht unerwartete Fehler zu erhalten. So liefert der Eintrag von

```
{=WAHL({1;2};SUMME(E41:G41);SUMME(E42:G42))}
```

in zwei untereinander stehenden Zellen in der ersten die Summe von *E41* bis *G41* und in der zweiten die Summe von *E42* bis *G42*. Schreiben Sie jedoch

```
{=SUMME(WAHL({1;2};E41:G41;E42:G42))}
```

wird in beide Zellen die Gesamtsumme von *E41* bis *G42* eingetragen. Verwenden Sie aber

=SUMME(WAHL(1;E41:G41;E42:G42))

und

=SUMME(WAHL(2;E41:G41;E42:G42))

getrennt (keine Matrixformeln), erhalten Sie die korrekten Ergebnisse.

Sie haben in den Zellen von B46 bis B52 die Namen der Wochentage, beginnend ab Sonntag, eingetragen. Dann liefert Ihnen

= "Heute ist " & WAHL(WOCHENTAG(HEUTE());B46;B47;B48;B49;B50;B51;B52) & ". "

in einer beliebigen Zelle den Text »Heute ist [*Wochentagsname*].«

INDEX()

Das Beispiel sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap09 in der Arbeitsmappe *Bezugsfunktionen.xlsx* auf dem Arbeitsblatt *Sonstige*.

Praxiseinsatz

Siehe auch



WVERWEIS() HLOOKUP()

WVERWEIS(*Suchkriterium*; *Matrix*; *Zeilenindex*; *Bereich_Verweis*)

Diese Funktion sucht in der oberen Zeile eines rechteckigen Bezugs oder einer Matrixkonstanten nach einem Wert. Entsprechend der übergebenen Argumente wird beim Suchen dieses oder eines ihn vertretenden Werts in der durch den Bezug definierten Tabelle eine entsprechende Anzahl von Schritten nach unten gegangen und der dortige Eintrag zurückgegeben.

Suchkriterium (erforderlich) muss als Text, Zahl oder Wahrheitswert auswertbar sein. Dieses Argument ist der Wert, nach welchem in der obersten Zeile der *Matrix* gesucht wird.

Bei der Suche nach Text wird nicht zwischen Groß- und Kleinschreibung unterschieden.

Matrix (erforderlich) ist ein Bezug auf einen rechteckigen Zellbereich (dieser kann auch durch einen Namen benannt sein) bzw. eine Matrixkonstante (das ist eine in geschweifte Klammern geschriebene Folge von Zahlen oder Text).

Zeilenindex (erforderlich) muss als positive ganze Zahl auswertbar sein, die nicht größer als die Anzahl der Zeilen im Argument *Matrix* ist. Das Argument gibt die Zeilennummer der Zeile an, aus der die durch die Funktion gelieferten Informationen stammen.

Bereich_Verweis (optional) verlangt einen Wahrheitswert. Dieser bestimmt, ob in der ersten Zeile nach einer genauen Übereinstimmung mit dem *Suchkriterium* gesucht werden soll (Argument ist gleich *FALSCH*) oder nicht (Argument ist gleich *WAHR* oder weggelassen).

Prinzipiell kann von zwei Situationen ausgegangen werden:

Die erste Situation: Sie suchen in der obersten Zeile nach einem bestimmten Begriff oder einer bestimmten Zahl und akzeptieren nur diese. Dann ist es nicht von Bedeutung, ob die Tabelle nach der ersten Zeile sortiert ist oder nicht. Wichtig ist nur, dass Sie in einer nicht sortierten Tabelle unbedingt den Parameter *Bereich_Verweis* als *FALSCH* deklarieren müssen. Tun Sie das nicht oder lassen den Parameter weg, wird Ihr Suchbegriff in aller Regel nicht gefunden, selbst wenn er in der obersten Zeile tatsächlich vorhanden ist (Ausnahmen sind zufällige »günstige« Situationen). Die Ursache liegt darin, dass Excel die Suche von hin-

Syntax

Definition

Argumente

Hinweis

Hintergrund

ten (ganz rechts) beginnt. Stößt es dabei auf einen Eintrag, der in lexikalischer oder numerischer Ordnung vor dem gesuchten Eintrag steht, wird das so interpretiert, als ob »am gesuchten bereits vorbeigegangen worden sein muss«. Da er nicht gefunden wurde, ist er also auch nicht da. In diesen Situationen wird dann der Wert als Stellvertreter als gefunden genommen, der in der genannten Ordnung – in Abwärtsrichtung gesehen – als nächster folgt. Diese Interpretation durch Excel ist in der Regel nicht sachgemäß und kann durch die Angabe von *FALSCH* als viertem Parameter vermieden werden. Steht der Wert nicht in der ersten Zeile, erscheint der Fehlerwert *#NV* als Ergebnis.

Die zweite Situation: Sie sind nicht an einer genauen Übereinstimmung des gefundenen Werts in der ersten Zeile interessiert. Das wird bei Texten sicher nur in seltenen Fällen Ihr Wunsch sein. Aber bei Zahlen, genauer gesagt Intervallgrenzen, wie sie bei Rabattstaffeln, Zinsstaffeln, Einkommensteuerbeträgen und anderen »Einordnungsproblemen« vorkommen, ist die Lage ganz anders. Wichtig ist nun, dass Ihre Liste (Tabelle) tatsächlich nach der ersten Zeile geordnet sein muss, die Intervallgrenzen also in aufsteigender (numerischer) Ordnung vorliegen. Wollen Sie tatsächlich eine solche Grenze finden, verwenden Sie als vierten Parameter den Wahrheitswert *FALSCH*. Setzen Sie diesen Parameter als *WAHR* an oder lassen ihn weg, wird Ihnen Excel (und das ist dann sachgemäß und gewünscht) die Intervallgrenze liefern, die in absteigender Richtung dem gesuchten Wert am nächsten kommt.

Hinweis Sie können das Argument *Matrix* auch als sogenannte Matrixkonstante übergeben. Dabei handelt es sich um in geschweifte Klammern eingeschlossene Zeichen- oder Zahlenfolgen. Zeilen werden durch ein Semikolon getrennt, Spalten durch einen Punkt. So ist *{1;2;3;4}* ein einspaltiges Gebilde, bestehend aus vier Zeilen. Bei *{11.12.13;21.22.23}* handelt es sich um eine dreispaltige Matrix, bestehend aus zwei Zeilen. Auf letztere kann durch *WVERWEIS()* etwa mittels

`=WVERWEIS(B27;{11.12.13;21.22.23};2)`

zugegriffen werden (steht in *B27* der Wert 13, wird 23 zurückgegeben).

Diese Art der Verwendung von *WVERWEIS()* dürfte jedoch kaum Anwendung finden, da die Eingabe der Matrizen aufwendig, fehleranfällig und sehr unflexibel ist.

Praxiseinsatz Eine typische Situation beim Finden von Werten in Kreuztabellen ist die Suche eines Kriteriums in der linken Spalte bei gleichzeitiger Berücksichtigung von zu beachtenden Werten in der Kopfzeile. Die Abbildung 9.11 zeigt das Beispiel eines fiktiven Fahrplans. Ermittelt werden sollen die möglichen Minuten bei gegebener Haltestelle und Uhrzeit.

	A	B	C	D	E	F	G
31							
32			00:00	06:00	10:00	16:00	21:00
33		Schlossallee	3, 33	3, 18, 33, 48	3, 23, 43	3, 18, 33, 48	3, 33
34		Parkstraße	8, 38	8, 23, 38, 53	8, 28, 48	8, 23, 38, 53	8, 38
35		Hauptbahnhof	12, 42	12, 27, 42, 57	12, 32, 52	12, 27, 42, 57	12, 42
36		Rathausplatz	13, 43	13, 28, 43, 58	13, 33, 53	13, 28, 43, 58	13, 43
37		Neue Straße	15, 45	0, 15, 30, 45	15, 35, 55	0, 15, 30, 45	15, 45
38		Los	18, 48	3, 18, 33, 48	18, 38, 58	3, 18, 33, 48	18, 48
39							
40		Haltestelle	Parkstraße				
41		Uhrzeit	07:15				
42		Minuten	8, 23, 38, 53				

Abbildung 9.11: Abfahrtszeiten in einem Fahrplan ermitteln

WVERWEIS() schafft diese Aufgabe nicht allein, da zusätzlich die Zeile angegeben werden muss, in welcher sich die Haltestelle befindet. Mithilfe der kombinierten Formel

```
=WVERWEIS(C41;C32:G38;VERGLEICH(C40;B33:B38;0)+1)
```

ist die Aufgabe jedoch gelöst. VERGLEICH() sucht in der ersten Spalte nach der Haltestelle aus C40, die Addition von 1 geschieht wegen der Einbeziehung der Kopfzeile in die Suchmatrix.

Die Auswahlliste der Haltestellen (Dropdownfeld) erhalten Sie durch Vergabe einer Gültigkeitsregel für C40, die nur Werte aus einer Liste zulässt.

INDEX(), SVERWEIS(), VERGLEICH(), VERWEIS()

Das Beispiel sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap09 in der Arbeitsmappe *Verweise.xlsx* auf dem Arbeitsblatt *WVERWEIS*.

Tipp

Siehe auch



ZEILE()



ZEILE(*Bezug*)

Diese Funktion liefert die (globale) Zeilennummer eines Bezugs.

Bezug (optional) muss als Bezug auf eine einzelne Zelle oder einen rechteckigen Zellbereich auswertbar sein.

Lassen Sie das Argument *Bezug* weg, wird die Zeilennummer geliefert, die der Zelle entspricht, in der die Funktion steht.

Ist *Bezug* ein Zellbereich (der auch durch einen Namen angegeben werden kann), kann die Funktion in Matrixformeln Verwendung finden (Abschluss mit $\boxed{\text{Strg}} + \boxed{\uparrow} + \boxed{\leftarrow}$). Ist die zeilenweise Ausdehnung des Zielbereichs dabei kleiner als die zeilenweise Ausdehnung des Arguments, werden die Informationen der fehlenden Zellen am Ende abgeschnitten. Ist der Zielbereich in dieser Ausdehnung größer, entsteht in den überzähligen Zellen der Fehlerwert #NV.

Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

Möchten Sie, beginnend ab Zelle A15, eine in den Zellen nach unten fortlaufende Nummerierung erzeugen, schreiben Sie in A15 die Formel

```
=ZEILE()-ZEILE($A$15)+1 & ". "
```

und kopieren diese (mit dem Ausfüllkästchen) so weit, wie die Nummerierung gehen soll.

Möchten Sie die erste Spalte eines benannten Bereichs (Name *Bereich*) mit einer fortlaufenden Nummerierung versehen, können Sie dies durch die Matrixformel

```
{=ZEILE(Bereich)-ZEILE(BEREICH.VERSCHIEBEN(Bereich;0;0;1))+1 & ". "}
```

erledigen. Beachten Sie, dass diese Formel vollkommen ohne eine Position des Bereichs auskommt und auch keinen Zirkelbezug erzeugt.

SPALTE(), ZEILEN()

Die Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap09 in der Arbeitsmappe *Bezugsfunktionen.xlsx* auf dem Arbeitsblatt *Spalten-Zeilen*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Fortlaufende Nummerierung

Fortlaufende Nummerierung im Bereich

Siehe auch



ZEILEN() ROWS()

Syntax ZEILEN(*Matrix*)

Definition Diese Funktion gibt die Anzahl der Zeilen einer Matrixkonstanten oder eines rechteckigen Bezugs zurück.

Argumente *Matrix* (erforderlich) ist eine Matrixkonstante oder ein Bezug auf einen rechteckigen Zellbereich.

Hintergrund Versuchen Sie einen nicht rechteckigen (mehrteiligen) Bereich als Argument zu verwenden, erhalten Sie die Fehlermeldung »zu viele Argumente«. Schließen Sie diese Argumente in zusätzliche runde Klammern ein, entsteht der Fehlerwert #BEZUG!. Wird der Bereich durch Schnittmengen – diese werden durch ein Leerzeichen zwischen Zellbezügen wie in =ZEILEN(B2:D4 E2:E4) gebildet – definiert und ist diese Schnittmenge leer, ergibt sich der Fehlerwert #NULL!.

Matrixkonstanten sind Folgen von Zahlen oder Text, die Sie selbst in geschweifte Klammern einschließen müssen. Dabei wird ein Punkt als Trennzeichen zwischen Zeilen und Spalten verwendet. Der einfache Ausdruck {1;2;3;4} wird als Matrix mit vier Zeilen interpretiert, wie das Ergebnis von

=ZEILEN({1;2;3;4})

zeigt. Dem gegenüber ist {11.12.13;21.22.23} zweizeilig, was durch das Ergebnis von

=ZEILEN({11.12.13;21.22.23})

dokumentiert werden kann.

Praxiseinsatz Um auf die letzte Zelle rechts unten eines benannten Bereichs zuzugreifen, können Sie die folgende Formel einsetzen:

=INDEX(MeinBereich;ZEILEN(MeinBereich);SPALTEN(MeinBereich))

Dabei ist *MeinBereich* der Name des infrage kommenden Bereichs.

Siehe auch SPALTEN(), ZEILE()



Das Beispiel sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap09 in der Arbeitsmappe *Bezugsfunktionen.xlsx* auf dem Arbeitsblatt *Spalten-Zeilen*.

Kapitel 10

Informationsfunktionen

BLATT()	313
BLÄTTER()	313
FEHLER.TYP()	314
INFO()	316
ISTBEZUG()	317
ISTFEHL()	318
ISTFEHLER()	319
ISTFORMEL()	320
ISTGERADE()	321
ISTKTEXT()	321
ISTLEER()	322
ISTLOG()	324
ISTNV()	325
ISTTEXT()	325
ISTUNGERADE()	327
ISTZAHL()	327
N()	328
NV()	329
TYP()	330
ZELLE()	333



Dieses Kapitel macht Sie mit den sogenannten Informationsfunktionen bekannt. Dabei handelt es sich vor allem um solche Funktionen, die über Inhalt (Text, Werte, Fehler) und Zustand (Formatierungen) von Zellen informieren. Sie werden begeistert von den IST-Funktionen sein, die es Ihnen erlauben, in Abhängigkeit von Zellinhalten durch den Einsatz der WENN()-Funktion Ihre Rechenwege bzw. Datenvisualisierungen zu verzweigen und anzupassen. Doch nicht nur die WENN()-Funktion erlaubt ein angepasstes Verhalten, auch bedingte Formatierungen und Gültigkeitsregeln können von den gezogenen Informationen profitieren. Insofern sind diese Funktionen in die Gruppe derjenigen nützlichen Helfer einzuordnen, die bei Praxislösungen eher im Hintergrund bleiben.

Tabelle 10.1

Die Nachschlage- und Verweisfunktionen in der Übersicht

Funktion	Beschreibung
BLATT()	Gibt die Nummer des Blatts zurück, welches durch den Bezug <i>Bereich</i> angesprochen wird
BLÄTTER()	Gibt die Anzahl der Blätter zurück, welche durch den Bezug <i>Bereich</i> angesprochen werden
FEHLER.TYP()	Gibt eine Zahl zurück, die einem der Fehlerwerte in Excel entspricht
INFO()	Gibt Informationen zur Betriebssystemumgebung sowie Excel selbst zurück
ISTBEZUG()	Gibt <i>WAHR</i> zurück, wenn das Argument <i>Wert</i> einen gültigen Zellbezug darstellt
ISTFEHL()	Gibt <i>WAHR</i> zurück, wenn das Argument <i>Wert</i> einen Fehlerwert darstellt oder durch Berechnung ergibt
ISTFEHLER()	Gibt <i>WAHR</i> zurück, wenn das Argument <i>Wert</i> einen Fehlerwert darstellt oder durch Berechnung ergibt
ISTFORMEL()	Gibt <i>WAHR</i> zurück, wenn ein Bezug auf eine existierende Zelle verweist und sich dort eine Formel befindet
ISTGERADE()	Gibt <i>WAHR</i> zurück, wenn das Argument <i>Zahl</i> eine gerade Zahl ist
ISTKTEXT()	Gibt <i>WAHR</i> zurück, wenn das Argument <i>Wert</i> keine Zeichenkette (Text) ist
ISTLEER()	Gibt <i>WAHR</i> zurück, wenn das Argument <i>Wert</i> sich auf eine leere Zelle bezieht
ISTLOG()	Gibt <i>WAHR</i> zurück, wenn das Argument <i>Wert</i> als Wahrheitswert ausgewertet wird
ISTNV()	Gibt <i>WAHR</i> zurück, wenn die Auswertung des Arguments <i>Wert</i> den Fehlerwert <i>#NV</i> liefert
ISTTEXT()	Gibt <i>WAHR</i> zurück, wenn das Argument <i>Wert</i> eine Zeichenkette (Text) ist
ISTUNGERADE()	Gibt <i>WAHR</i> zurück, wenn das Argument <i>Zahl</i> eine ungerade Zahl ist
ISTZAHL()	Gibt <i>WAHR</i> zurück, wenn sich das Argument <i>Wert</i> als Zahl (numerischer Wert) auswerten lässt
N()	Gibt den in eine Zahl umgewandelten Wert des Arguments <i>Wert</i> zurück
NV()	Gibt den Fehlerwert <i>#NV</i> zurück
TYP()	Gibt eine Zahl zurück, die den Datentyp des angegebenen Arguments charakterisiert
ZELLE()	Liefert Informationen zu Formatierung, Position oder Inhalt der angegebenen Zelle

BLATT() SHEET()

BLATT(*Bereich*)

Diese Funktion gibt die Nummer des Blatts zurück, welches durch den Bezug *Bereich* angesprochen wird. Wird *Bereich* nicht angegeben, liefert die Funktion die Nummer des Blatts, in welchem sie selbst steht.

Bereich (optional) ist ein Bezug auf einen Bereich (der auch benannt sein kann) bzw. der Name eines Blatts.

Ist das durch seinen Namen gesuchte Blatt nicht vorhanden bzw. zeigt der Zellbezug auf ein nicht vorhandenes Blatt, entsteht der Fehler #NV. Im zweiten Fall öffnete die Arbeitsmappe mit der Aufforderung, nicht korrekte Verknüpfungen zu korrigieren. Wird die Verknüpfung entfernt, entsteht statt des Funktionseintrags der Eintrag #NV.

Wurde ein Zellbereich nach Verwendung der Funktion entfernt, entsteht der Fehler #BEZUG!.

Die Verwendung eines nicht gültigen Namens für den Zellbereich führt zu #NAME?.

Man kann die Funktion mit einem Bereichsargument verwenden, um einen erklärenden Text anzufertigen, der auf ein bestimmtes Blatt verweist, unabhängig davon, an welcher Stelle es sich in der Mappe einordnet und wie es konkret heißt.

BLÄTTER(), SPALTE(), ZEILE(), ZELLE()

Einige Experimente finden Sie im Ordner \Ms5-235\Kap10 in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *Sonstige*.

Neu in Excel
2013

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



BLÄTTER() SHEETS()

BLÄTTER(*Bereich*)

Diese Funktion gibt die Anzahl der Blätter zurück, welche durch den Bezug *Bereich* angesprochen werden. Wird *Bereich* nicht angegeben, liefert die Funktion die Anzahl der Blätter in der benutzten Mappe.

Bereich (optional) ist ein Bezug auf einen (dreidimensionalen) Bereich, der auch benannt sein kann.

Zeigt der Zellbezug auf ein nicht vorhandenes Blatt, entsteht der Fehler #NV.

Ist der Zellbezug nicht auswertbar, entsteht der Fehler #WERT!. Das ist etwa bei Verwendung einer Zeichenkette oder eines Wahrheitswerts der Fall.

Wurde ein Zellbereich nach Verwendung der Funktion entfernt, entsteht der Fehler #BEZUG!

Die Verwendung eines nicht gültigen Namens für den Zellbereich führt zu #NAME?.

Angenommen, Sie benutzen 3D-Bezüge, um über mehrere Blätter hinweg in der Form *Tabelle1:Tabelle7!A1* oder durch einen 3D-Namen (das ist ein Name, der für hintereinander liegende Zellen auf mehreren Blättern vergeben wurde) auf mehrere gleichartige Zellen zuzugreifen. Dann haben Sie mit der Funktion die Möglichkeit, die Anzahl der Blätter in diesen

Neu in Excel
2013

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Bezügen zu prüfen. So können Sie feststellen (etwa durch bedingte Formatierung), ob wirklich vier Quartale oder auch zwölf Monate an einer Auswertung beteiligt sind.

Siehe auch BLATT(), SPALTEN(), ZEILEN()



Einige Experimente finden Sie im Ordner \Ms5-235\Kap10 in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *Sonstige*.

FEHLER.TYP() ERROR.TYPE()

Syntax FEHLER.TYP(*Fehlerwert*)

Definition Diese Funktion gibt eine Zahl zurück, die einem der Fehlerwerte in Excel entspricht. Ist kein Fehler in der Zelle oder der Berechnung vorhanden, liefert die Funktion selbst den Fehlerwert #NV.

Argumente *Fehlerwert* (erforderlich) ist der Wert eines Fehlers (angezeigter Fehlerwert in einer Zelle bzw. Ergebnis einer fehlerhaften Berechnung), dessen Kennnummer Sie finden möchten.

Hintergrund Die Funktion kann u.U. in einer WENN()-Funktion verwendet werden, um einen möglichen erzeugten Fehlerwert durch eine Zeichenfolge, die dem Anwender einen erläuternden Text liefert, zu ersetzen. Dabei ist die Kenntnis des Zusammenhangs zwischen Fehlerwert und Rückgabewert wie in Tabelle 10.2 notwendig. Der Fehler mit der Nummer 8 tritt nur auf, solange Daten aus anderen Quellen geholt werden (etwa bei der Verwendung von Cube-Funktionen), je nach Ergebnis wird dann in der Zelle ein Wert oder ein anderer Fehler angezeigt.

Tabelle 10.2
Zusammenhang zwischen Fehlerwerten und Ergebnissen der Funktion FEHLER.TYP()

Fehlerwert	Rückgabewert
#NULL!	1
#DIV/0!	2
#WERT!	3
#BEZUG!	4
#NAME?	5
#ZAHL!	6
#NV	7
#DATEN_ABRUFEN	8
Kein Fehler	#NV

Tipp Für den Fall, dass Sie häufig den Fehlerwert in einer anderen Darstellung anzeigen lassen möchten, empfiehlt sich der Einsatz einer benutzerdefinierten Funktion, die per VBA programmiert wird.

Praxiseinsatz Bedingte Formatierung Sie beabsichtigen, eine bedingte Formatierung von Zellen so vorzunehmen, dass Fehlerwerte durch einen farbigen Zellhintergrund hervorgehoben werden. Dazu können Sie wie in Abbildung 10.1 vorgehen, wo allgemeine Fehler durch die Funktion ISTFEHLER() erkannt und die Zellen gelb markiert werden. Der schwerwiegende Fehler der Division durch Null (die Funktion FEHLER.TYP() liefert dann den Wert 2) hingegen wird rot dargestellt.

Achten Sie auf die Reihenfolge der Bedingungen: Würden Sie zuerst auf den Wahrheitswert von ISTFEHLER() abstellen, käme die zweite Bedingung mit dem Fehlertyp nicht mehr zum Zuge.

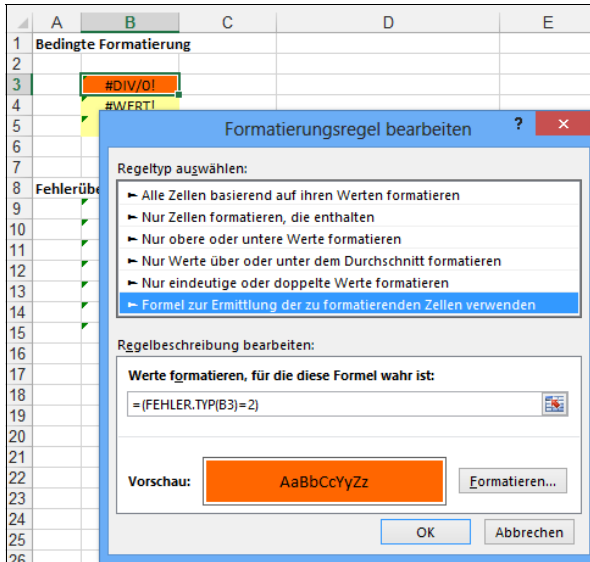


Abbildung 10.1: Bedingte Formatierung unter Nutzung von Fehlerinformationen

Das wiederholte Herstellen des Zusammenhangs zwischen Fehlertyp (also den Zahlen von 1 bis 7) und dem zugrunde liegenden Fehlerwert (*#DIV/0!* usw.) kann durch eine benutzerdefinierte Funktion, die eventuell in der persönlichen Makroarbeitsmappe aufbewahrt wird, verkürzt und vereinfacht werden. Das Listing 10.1 zeigt einen möglichen Vorschlag, der in Abbildung 10.2 zum Einsatz kommt.

Eine benutzerdefinierte Funktion

```
Function FehlerInWorten(Bereich As Range)
    If WorksheetFunction.IsError(Bereich.Value) Then
        Select Case CStr(Bereich.Value)
            Case "Fehler 2000"
                FehlerInWorten = "Bereiche schneiden sich nicht"
            Case "Fehler 2007"
                FehlerInWorten = "Division durch Null"
            Case "Fehler 2015"
                FehlerInWorten = "Ausdruck nicht berechenbar"
            Case "Fehler 2023"
                FehlerInWorten = "Zellbezug verloren"
            Case "Fehler 2029"
                FehlerInWorten = "Name nicht definiert"
            Case "Fehler 2036"
                FehlerInWorten = "Zahlenwert nicht darstellbar"
            Case "Fehler 2042"
                FehlerInWorten = "Wert nicht vorhanden"
        End Select
    Else
        FehlerInWorten = "kein Fehler"
    End If
End Function
```

Listing 10.1: Zuordnung von Text zu Fehlerwerten per VBA – eine Alternative zu *FEHLER.TYP*

8	Fehlerübersicht		
9	✓	#NULL!	1 Bereiche schneiden sich nicht
10	✓	#DIV/0!	2 Division durch Null
11	✓	#WERT!	3 Ausdruck nicht berechenbar
12	✓	#BEZUG!	4 Zellbezug verloren
13	✓	#NAME?	5 Name nicht definiert
14	✓	#ZAH!	6 Zahlenwert nicht darstellbar
15	✓	#NV	7 Wert nicht vorhanden
16		2 ✓ #NV	kein Fehler

Abbildung 10.2: Links die Anzeige des Fehlers, rechts die Auswertung mittels benutzerdefinierter Funktion

Hinweis Ein Hinweis für VBA-Profis: Zellwerte in Zellen sind vom Typ *Variant*. Fehlerwerte haben den »Untertyp« *Error*, weshalb die Umwandlung mithilfe von *CStr* notwendig wird.

Siehe auch ISTFEHL(), ISTFEHLER(), ISTNV()



Das Beispiel finden Sie im Ordner *Ms5-235\Kap10* in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *Fehler*. Der Quellcode für die benutzerdefinierte Funktion befindet sich im Modul *modFunction* dieser Arbeitsmappe.

INFO()



Syntax INFO(*Typ*)

Definition Gibt einige Informationen zur aktuellen Betriebssystemumgebung sowie Excel selbst zurück.

Argumente *Typ* (erforderlich) ist eine Zeichenkette (Text in Anführungszeichen), mit deren Hilfe Sie bestimmen, welche Art von Informationen Sie erhalten möchten.

Hintergrund Die Tabelle 10.3 zeigt den Zusammenhang zwischen dem als Argument *Typ* anzugebenden Text und der gelieferten Information.

Tabelle 10.3

Die Zeichenketten-Argumente und deren Bedeutung

Argument	Rückgabewert
<i>Dateienzahl</i>	Anzahl von Arbeitsblättern in allen geöffneten Arbeitsmappen (auch ausgeblendeten) und geladenen Add-Ins
<i>Berechne</i> ^a	Anzeige des aktuellen Neuberechnungsmodus
<i>System</i>	Der Name des Betriebssystems, für Windows wird <i>pcdos</i> , für Macintosh <i>mac</i> ausgegeben (Text)
<i>Sysversion</i>	Version des aktuellen Betriebssystems, der Zellinhalt ist als Text formatiert
<i>Ursprung</i>	Gibt den absoluten Bezug der ersten oberen, linken Zelle im aktuellen Fensterbereich zurück, die momentan sichtbar ist. Vorangestellt wird ein Präfix in der Form \$A: (Text).
<i>Version</i>	Die Version von Excel als Text
<i>Verzeichnis</i>	Der Pfad des aktuellen Verzeichnisses oder Ordners

^a In den Versionen vor Excel 2010 heißt dieses Argument *Rechenmodus*.

Wichtig Die Ergebnisse der eingetragenen Funktion aktualisieren sich zum Teil nicht von selbst. Das ist klar, wenn Sie zur Anzeige des »Rechenmodus« diesen zwischenzeitlich in »Manuell« gewandelt haben. Dann hilft nur die Taste **[F9]** zur Neuberechnung.

Nicht ganz so offensichtlich ist das Verhalten im Falle von *Verzeichnis*. Rufen Sie beispielsweise das Dialogfeld zum Öffnen einer Datei auf, ändert sich das aktuelle Verzeichnis. Öffnen Sie aber keine Datei mithilfe des Dialogfelds (Sie entscheiden sich für *Abbrechen*), wird diese Änderung erst nach einer Neuberechnung in der Zelle sichtbar.

Der Begriff *Dateianzahl* stammt sicher aus den Zeiten, in denen es noch keine Arbeitsmappen gab und ist eher irreführend.

Tip

Mithilfe der Funktion INFO() lässt sich automatisch ein Hyperlink zum aktuellen Verzeichnis erstellen. Die Funktion HYPERLINK() leistet dazu die weiteren notwendigen Dienste:

Praxiseinsatz

```
=HYPERLINK(INFO("Verzeichnis"))
```

Beachten Sie das oben beschriebene Aktualisierungsproblem. Das funktioniert so bis Windows XP. Ab Vista müssen Sie noch den letzten Backslash entfernen, wenn in *B11* die gewünschte Information steht:

```
=HYPERLINK(LINKS(B11;LÄNGE(B11)-1))
```

Eine direkte Angabe von

```
=HYPERLINK(LINKS(INFO("Verzeichnis");LÄNGE(INFO("Verzeichnis"))-1))
```

führt wohl nicht zum Erfolg.

ZELLE()

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap10` in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *Info*.

Siehe auch

ISTBEZUG()



ISTBEZUG(*Wert*)

Syntax

Gibt den Wahrheitswert *WAHR* zurück, wenn das Argument *Wert* einen gültigen Zellbezug (Adresse oder Name) darstellt. Andernfalls wird der Wahrheitswert *FALSCH* zurückgegeben.

Definition

Wert (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), den Sie prüfen möchten.

Argumente

Diese Funktion ist eine der IST-Funktionen, die in Abhängigkeit des Arguments einen Wahrheitswert über dieses Argument zurückgeben. Das Argument der IST-Funktionen wird vor der Auswertung nicht umgewandelt. Das heißt u.a., eine Zeichenkette, die wie eine Zahl aussieht, bleibt eine Zeichenkette (Text) und wird nicht als Zahl interpretiert. IST-Funktionen finden häufig ihre Anwendung im Zusammenspiel mit der WENN()-Funktion, um in Abhängigkeit der aufgetretenen Situation in Anzeige und Berechnung zu verzweigen. Auch bedingte Formatierungen und Gültigkeitsregeln können auf den Ergebnissen der IST-Funktionen aufbauen.

Hintergrund

ISTBEZUG() prüft nur, ob das Argument als Zellbezug verstanden werden kann oder nicht. Damit liefern Argumente wie *B1* oder *Tabelle1!\$A\$1* den Wahrheitswert *WAHR*. Im zweiten Fall auch dann, wenn es die *Tabelle1* in der Mappe gar nicht gibt (allerdings versucht Excel nach Eingabe der Formel diese Tabelle zunächst zu finden). Argumente wie *-B1*, *12* oder *A1* ergeben den Wahrheitswert *FALSCH*.

Hinweis Haben Sie die Option *Z1S1-Bezugsart* eingestellt, werden die entsprechenden Adressen als Bezüge erkannt (also etwa *Z2S3* oder *Z(-1)S(-2)*).

**Praxiseinsatz
Modifizierte
Fehleranzeigen** Sie haben in einer Arbeitsmappe für einen Bereich den Namen »ABC« vergeben und nutzen diesen für Berechnungen in einer Zelle außerhalb dieses Bereichs. Sie möchten dem (unbeabsichtigten) Löschen dieses Namens etwas entgegensetzen, was auf die nun nicht mehr mögliche Berechnung hinweist. Die einfache Formel

=MITTELWERT(ABC)

könnte dann zu

=WENN(ISTBEZUG(ABC);MITTELWERT(ABC);"Name ABC wurde gelöscht")

erweitert werden. Damit wird Ihr Text anstelle des sonst entstehenden Fehlerwerts #NAME? angezeigt.

**Vorsicht ist
angebracht** Der Gebrauch von ISTBEZUG() liefert manchmal auch Überraschungen. So ist

=ISTBEZUG(A1)

vom Wahrheitswert *WAHR*, jedoch

=ISTBEZUG(ADRESSE(1;1))

vom Wahrheitswert *FALSCH*. Die Ursache liegt darin, dass ADRESSE() einen Text und keinen Bezug liefert, der Text lässt sich offenbar aber nicht auswerten.

Siehe auch ISTLEER(), ISTFEHL(), ISTFEHLER(), ISTFORMEL(), ISTLOG(), ISTNV(), ISTKTEXT(), ISTZAHL(), ISTTEXT()



Die Beispiele sowie einige Test zu den Hintergrundinformationen finden Sie im Ordner *Ms5-235\Kap10* in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *IST-Funktionen*.

ISTFEHL() ISERR()

Syntax ISTFEHL(*Wert*)

Definition Die Funktion gibt den Wahrheitswert *WAHR* zurück, wenn *Wert* einen Fehlerwert darstellt oder durch Berechnung ergibt. Anderenfalls wird der Wahrheitswert *FALSCH* zurück gegeben.

Eine Ausnahme bildet der Fehlerwert #NV – hier gibt die Funktion den Wahrheitswert *FALSCH* zurück (das ist der Unterschied zur Funktion ISTFEHLER()).

Argumente *Wert* (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), den Sie prüfen möchten.

Hintergrund Diese Funktion ist eine der IST-Funktionen, die in Abhängigkeit des Arguments einen Wahrheitswert über dieses Argument zurückgeben. Das Argument der IST-Funktionen wird vor der Auswertung nicht umgewandelt. Das heißt u.a., eine Zeichenkette, die wie eine Zahl aussieht, bleibt eine Zeichenkette (Text) und wird nicht als Zahl interpretiert. IST-Funktionen finden häufig ihre Anwendung im Zusammenspiel mit der WENN()-Funktion, um in Abhängigkeit der aufgetretenen Situation in Anzeige und Berechnung zu verzweigen. Auch bedingte Formatierungen und Gültigkeitsregeln können auf den Ergebnissen der IST-Funktionen aufbauen.

Vergleichen Sie hier auch die Hinweise und Beispiele zur Funktion FEHLER.TYP(), die eine detaillierte Untersuchung eines möglichen Fehlerwerts erlaubt.

Sie möchten den Mittelwert über die Zahlen eines Bereichs (hier B26 bis B28) berechnen und dabei den Fehler in der Anzeige vermeiden, der eintritt, wenn nicht wenigstens in einer der Zellen des Bereichs eine Zahl steht, die Mittelwertbildung also mit einem Fehler endet:

```
=WENN(ISTFEHL(MITTELWERT(B26:B28));"Prüfen Sie die Eingabewerte.";MITTELWERT(B26:B28))
```

FEHLER.TYP(), ISTLEER(), ISTFEHLER(), ISTFORMEL(), ISTLOG(), ISTNV(), ISTKTEXT(), ISTZAHL(), ISTBEZUG(), ISTTEXT()

Das Beispiel sowie einige Tests zur Definition finden Sie im Ordner \Ms5-235\Kap10 in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *IST-Funktionen*.

Praxiseinsatz

Siehe auch



ISTFEHLER()



ISERROR()

ISTFEHLER(*Wert*)

Die Funktion gibt den Wahrheitswert *WAHR* zurück, wenn *Wert* einen Fehlerwert darstellt oder durch Berechnung ergibt. Andernfalls wird der Wahrheitswert *FALSCH* zurückgegeben.

Der Unterschied zur Funktion ISTFEHL() besteht im Verhalten für den Argumentwert *#NV*.

Wert (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), den Sie prüfen möchten.

Diese Funktion ist eine der IST-Funktionen, die in Abhängigkeit des Arguments einen Wahrheitswert über dieses Argument zurückgeben. Das Argument der IST-Funktionen wird vor der Auswertung nicht umgewandelt. Das heißt u.a., eine Zeichenkette, die wie eine Zahl aussieht, bleibt eine Zeichenkette (Text) und wird nicht als Zahl interpretiert. IST-Funktionen finden häufig ihre Anwendung im Zusammenspiel mit der WENN()-Funktion, um in Abhängigkeit der aufgetretenen Situation in Anzeige und Berechnung zu verzweigen. Auch bedingte Formatierungen und Gültigkeitsregeln können auf den Ergebnissen der IST-Funktionen aufbauen.

Vergleichen Sie hier auch die Hinweise und Beispiele zur Funktion FEHLER.TYP(), die eine detaillierte Untersuchung eines möglichen Fehlerwerts erlaubt.

Sie habe eine Liste mit den Geburtstagen von Personen (Artikel- oder Bestellnummern, Adressdetails, Telefonnummern usw.) angefertigt und möchten mithilfe von SVERWEIS() darauf zugreifen. Die Abbildung 10.3 zeigt ein »Mini-Beispiel«.

Syntax

Definition

Hinweis

Argumente

Hintergrund

Praxiseinsatz

	A	B	C	D	E
35					
36		Johann		Hans	01.01.1955
37		#NV		Georg	10.10.1978
38		nicht gefunden		Albrecht	31.12.1964
39					

Abbildung 10.3: Fehler bei der Suche in Listen erklären

Schreiben Sie in die Zelle *B37*

```
=SVERWEIS(B36;D36:E38;2;FALSCH)
```

(achten Sie auf den letzten Parameter *FALSCH*, der nach einer genauen Übereinstimmung des gesuchten Begriffs forscht), erhalten Sie den Fehlerwert *#NV*, der vor allem beim Ausdrucken der Tabelle störend wirkt. In anderen Situationen wollen Sie vielleicht mit den gefundenen Zwischenwerten weiterrechnen und dabei Fehler ignorieren.

Im vorliegenden Fall helfen Sie sich mit der *WENN()*-Funktion in *B38* statt des einfacheren Eintrags in *B37*:¹

```
=WENN(ISTFEHLER(SVERWEIS(B36;D36:E38;2;FALSCH));"nicht gefunden";  
SVERWEIS(B36;D36:E38;2;FALSCH))
```

Siehe auch FEHLER.TYP(), ISTLEER(), ISTFEHL(), ISTLOG(), ISTNV(), ISTKTEXT(), ISTZAHL(), ISTBEZUG(), ISTTEXT(), WENNNV()



Das Beispiel sowie einige Tests zur Definition finden Sie im Ordner *\Ms5-235\Kap10* in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *IST-Funktionen*.

Neu in Excel
2013

ISTFORMEL()



ISFORMULA()

Syntax

ISTFORMEL(*Bezug*)

Definition

Die Funktion gibt den Wahrheitswert *WAHR* zurück, wenn das Argument *Bezug* auf eine existierende Zelle verweist und sich dort eine Formel befindet. Anderenfalls wird der Wahrheitswert *FALSCH* zurückgegeben. Besteht der Bezug aus mehreren Zellen, ist die linke obere Ecke der Vertreter.

Argumente

Bezug (erforderlich) muss in Form eines Zellbezugs oder eines definierten Namens angegeben werden.

Hintergrund

Anders als etwa die Funktion *ISTZAHL()* verlangt diese Funktion unbedingt einen Bezug. Existiert dieser nicht, erscheint *#NAME?*. Alle anderen Eingaben scheitern an einer Fehlermeldung von Excel.

Praxiseinsatz

Obwohl es verschiedene Möglichkeiten des Blattschutzes gibt (und das Ausblenden von Formeln gehört dazu), kann es sein, dass Sie weniger restriktiv handeln wollen. Es reicht vielleicht, Zellen mit Formeln so zu markieren, dass ihr Inhalt nur mit Vorsicht verändert werden sollte. Dabei hilft eine bedingte Formatierung, die auf der Funktion aufsetzt und im Falle der Rückgabe von *WAHR* die Zelle entsprechend einfärbt.

Siehe auch

ISTLEER(), ISTFEHL(), ISTFEHLER(), ISTLOG(), ISTNV(), ISTKTEXT(), ISTBEZUG(), ISTTEXT(), ISTZAHL()



Die Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner *\Ms5-235\Kap10* in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *IST-Funktionen*.

¹ Nutzer von Excel 2013 werden sich hier über die Funktion *WENNNV()* freuen, die die Angelegenheit verkürzt.

ISTGERADE()



ISTGERADE(*Zahl*)

Die Funktion gibt den Wahrheitswert *WAHR* zurück, wenn das Argument *Zahl* eine gerade Zahl ist. Ist das Argument eine ungerade Zahl, wird der Wahrheitswert *FALSCH* zurückgegeben.

Zahl (erforderlich) ist der zu prüfende Ausdruck.

Sie können der Funktion beliebige Ausdrücke als Argumente übergeben. Handelt es sich um Zahlen, werden ganze Zahlen wie erwartet behandelt. Bei gebrochenen Zahlen werden die Nachkommastellen abgeschnitten und der so entstehende Wert ausgewertet. Dadurch ist das Ergebnis von =ISTGERADE(2,4) gleich *WAHR* und =ISTGERADE(-1,6) liefert den Wahrheitswert *FALSCH*.

Ist das Argument numerisch nicht auswertbar, wird für Zeichenketten und Wahrheitswerte der Fehlerwert #*WERT!* ermittelt, für Fehlerwerte ist das Ergebnis gleich dem Argument.

Die Funktion ISTGERADE() ist der Gegenpart zur Funktion ISTUNGERADE():

ISTGERADE(Zahl)=NICHT(ISTUNGERADE(Zahl))

Angenommen, Sie wollen jeder zweiten Zeile eines Arbeitsblatts zur optischen Unterscheidung eine andere Hintergrundfarbe zuweisen und die vorhandenen Tabellenformatierungen erscheinen Ihnen hier als nicht genügend. Es liegt nahe, die *Bedingte Formatierung* zu verwenden und den Rückgabewert der Funktion ZEILE() darauf zu prüfen, ob er gerade ist. Diese Prüfung könnte mithilfe der Formel

=ISTGERADE(ZEILE(Zelle))

erfolgen.

Nachdem Sie den Eintrag der Formel für die Zelle *A1* vorgenommen haben, können Sie das Format durch das Symbol *Format übertragen* auf alle anderen gewünschten Zellen übertragen.

Denkbar sind auch weitere Prüfungen, die etwa jeweils zwei gruppierte Zeilen betreffen. Dazu ist nicht ZEILE(), sondern etwa (ZEILE()+1)/2 zu untersuchen:

=(ISTGERADE((ZEILE(Zelle)+1)/2))

ISTUNGERADE(), GERADE(), UNGERADE()

Das Beispiel sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner *Ms5-235\Kap10* in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *IstGerade*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



ISTKTEXT()



ISTKTEXT(*Wert*)

Die Funktion gibt den Wahrheitswert *WAHR* zurück, wenn das Argument *Wert keine* Zeichenkette (Text) ist. Besteht jedoch das Argument aus Text, wird *FALSCH* zurückgegeben.

Wert (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), den Sie prüfen möchten.

Syntax

Definition

Argumente

Hintergrund Diese Funktion ist eine der IST-Funktionen, die in Abhängigkeit des Arguments einen Wahrheitswert über dieses Argument zurückgeben. Das Argument der IST-Funktionen wird vor der Auswertung nicht umgewandelt. Das heißt u.a., eine Zeichenkette, die wie eine Zahl aussieht, bleibt eine Zeichenkette (Text) und wird nicht als Zahl interpretiert. IST-Funktionen finden häufig ihre Anwendung im Zusammenspiel mit der WENN()-Funktion, um in Abhängigkeit der aufgetretenen Situation in Anzeige und Berechnung zu verzweigen. Auch bedingte Formatierungen und Gültigkeitsregeln können auf den Ergebnissen der IST-Funktionen aufbauen.

Der Rückgabewert der Funktion ist stets *WAHR*, es sei denn, das Argument ist nicht leerer Text. Es wird also auch *WAHR* ermittelt, wenn das Argument ein Bezug auf eine leere Zelle ist. Das ist konsequent, denn ist der Bezug auf eine Zelle gerichtet, die wiederum einen Bezug auf eine leere Zelle enthält, ist es von den eingestellten Optionen abhängig, ob in diesem Fall in der Zelle mit dem Bezug eine 0 (Null) angezeigt werden soll oder nicht.

Der Gegenpart dieser Funktion ist die Funktion ISTTEXT():

ISTKTEXT(Wert)=NICHT(ISTTEXT(Wert))

Praxiseinsatz Zeichenketten, die für die eingestellte Zellbreite zu lang sind, werden nur dann über den Rand hinaus angezeigt, wenn die rechte Nachbarzelle leer ist. Das kann dann zu Fehlanzeigen führen, wenn die Zeichenketten dynamisch als Auswertung von Berechnungen entstehen.

Sie können nun der Zelle rechts neben einer zu erwartenden »langen« Zeichenkette eine Gültigkeitsregel zuweisen, die das Einfügen von Informationen in diese Zelle verbietet, ohne zu einem starken Mittel wie dem Blattschutz zu greifen.

Markieren Sie die Zelle, die Sie sperren möchten (etwa *H42*). Rufen Sie auf der Registerkarte *DATEN* in der Gruppe *Datentools* den Befehl *Datenüberprüfung/Datenüberprüfung* auf und wählen Sie auf der Registerkarte *Einstellungen* im Listenfeld *Zulassen* den Eintrag *Benutzerdefiniert* aus. In das *Formel*-Feld schreiben Sie

=ISTKTEXT(G42)

wobei *G42* der Bezug zur Zelle mit dem zu erwartenden Text ist.

Siehe auch ISTLEER(), ISTFEHL(), ISTFEHLER(), ISTFORMEL(), ISTLOG(), ISTNV(), ISTZAHL(), ISTBEZUG(), ISTTEXT()



Das Beispiel finden Sie im Ordner *Ms5-235\Kap10* in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *IST-Funktionen*.

ISTLEER() ISBLANK()

Syntax ISTLEER(*Wert*)

Definition Die Funktion gibt den Wahrheitswert *WAHR* zurück, wenn das Argument *Wert* sich auf eine leere Zelle bezieht. In allen anderen Fällen wird der Wahrheitswert *FALSCH* ermittelt.

Argumente *Wert* (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), den Sie prüfen möchten.

Hintergrund

Diese Funktion ist eine der IST-Funktionen, die in Abhängigkeit des Arguments einen Wahrheitswert über dieses Argument zurückgeben. Das Argument der IST-Funktionen wird vor der Auswertung nicht umgewandelt. Das heißt u.a., eine Zeichenkette, die wie eine Zahl aussieht, bleibt eine Zeichenkette (Text) und wird nicht als Zahl interpretiert. IST-Funktionen finden häufig ihre Anwendung im Zusammenspiel mit der WENN()-Funktion, um in Abhängigkeit der aufgetretenen Situation in Anzeige und Berechnung zu verzweigen. Auch bedingte Formatierungen und Gültigkeitsregeln können auf den Ergebnissen der IST-Funktionen aufbauen.

Der durch die Funktion zurückgegebene Wahrheitswert ist nicht nur bei Bezügen auf nicht leere Zellen *FALSCH*, sondern auch bei Argumenten, die keine Bezüge oder keine gültigen Bezüge (nicht existierende Namen) sind, also auch bei der Angabe von Text, Zahlen, Wahrheitswerten und Fehlern.

Durch die Übernahme von Zellwerten durch Zellbezüge kann es gelegentlich zu unerwarteten Ergebnissen kommen. Dies soll ein stark vereinfachtes Beispiel demonstrieren.

Praxiseinsatz

In Abbildung 10.4 wird in der *H*-Spalte ein Bezug auf die Zellen der *G*-Spalte hergestellt. Dies wird in den Zeilen 50 bis 52 durch einfache Übernahme (=G50 usw.) erreicht. Durch dieses Vorgehen ergibt sich ein Mittelwert in *I50* aus drei Zahlen (0, 1, 2) zu 1. Das ist aber nicht korrekt, denn die Zahl 0 (Null) gibt es in der *G*-Spalte nicht.

	F	G	H	I
49				
50		1	1	1
51			0	
52		2	2	
53				
54		1	1	1,5
55				
56		2	2	
57				

Abbildung 10.4: Probleme bei der Mittelwertbildung in Zellen mit Bezügen

Die korrekte Lösung geschieht in den Zeilen 54 bis 56. Dort wird die Übernahme in die *H*-Spalte durch

```
=WENN(ISTLEER(G54);"";G54)
```

(ähnlich für die Zeilen 55 und 56) vorgenommen, im Ergebnis entsteht in *I54* der Mittelwert aus zwei Zahlen (1 und 2).

Vor allem bei komplizierten Werteübernahmen und weiteren Berechnungen ist es empfehlenswert, sich die Wirkungen an möglichst vereinfachten Beispielen zu verdeutlichen.

Tip

ISTFEHL(), ISTFEHLER(), ISTFORMEL(), ISTLOG(), ISTNV(), ISTKTEXT(), ISTZAHL(), ISTBEZUG(), ISTTEXT()

Siehe auch

Das Beispiel sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner `\Ms5-235\Kap10` in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *IST-Funktionen*.



ISTLOG()



ISLOGICAL()

Syntax ISTLOG(*Wert*)

Definition Diese Funktion gibt den Wahrheitswert *WAHR* zurück, wenn das Argument *Wert* als Wahrheitswert ausgewertet wird. Andernfalls ist der Rückgabewert der Wahrheitswert *FALSCH*.

Argumente *Wert* (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), den Sie prüfen möchten.

Hintergrund Diese Funktion ist eine der IST-Funktionen, die in Abhängigkeit des Arguments einen Wahrheitswert über dieses Argument zurückgeben. Das Argument der IST-Funktionen wird vor der Auswertung nicht umgewandelt. Das heißt u.a., eine Zeichenkette, die wie eine Zahl aussieht, bleibt eine Zeichenkette (Text) und wird nicht als Zahl interpretiert. IST-Funktionen finden häufig ihre Anwendung im Zusammenspiel mit der WENN()-Funktion, um in Abhängigkeit der aufgetretenen Situation in Anzeige und Berechnung zu verzweigen. Auch bedingte Formatierungen und Gültigkeitsregeln können auf den Ergebnissen der IST-Funktionen aufbauen.

Praxeinsatz Die folgenden Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

Umgangssprachliche Wahrheitswerte Mithilfe der WENN()-Funktion lassen sich die Wahrheitswerte in Umgangssprache umwandeln, etwa

```
=WENN(ISTLOG(H60);WENN(H60=WAHR;"korrekt";"leider daneben");H60)
```

Die Zelle *H60* (Spalte oder Zeile) kann dann u.U. auch ausgeblendet werden. ISTLOG() wird hier eingesetzt, um zu testen, ob eine sprachliche Anpassung überhaupt sinnvoll ist. Die zweite WENN()-Funktion prüft im Bedarfsfall den Wahrheitswert und übersetzt ihn.

Wahrheitswerte visualisieren Sie können die *Bedingte Formatierung* nutzen, um Zellen mit Wahrheitswerten entsprechend dem jeweiligen Wahrheitswert etwa grün (*WAHR*) und rot (*FALSCH*) einzufärben und dabei die Schriftfarbe dem Hintergrund anzupassen.

Falls Sie im Vorhinein noch nicht wissen, in welchen Zellen eine solche Gestaltung notwendig wird, können Sie diese mit zwei Handgriffen vorbereiten. Markieren Sie eine der in Frage kommenden Zellen und geben dort die bedingte Formatierung an. Übertragen Sie dann mit dem Symbol *Format übertragen* diese Formatierung auf alle restlichen Zellen (des Bereichs, Mehrfachbereichs, Blatts). Die notwendigen Formeln lauten:

```
=UND(ISTLOG(G63);G63)
```

für den »grünen« und

```
=UND(ISTLOG(G63);NICHT(G63))
```

für den »roten« Teil, falls *G63* Ihre Startzelle ist. Im ersten Fall ist die Gesamtbedingung *WAHR*, wenn sowohl der Zelleninhalt ein Wahrheitswert und dieser gleich *WAHR* ist. Im zweiten Fall wird ebenfalls auf das Vorhandensein eines Wahrheitswerts, der allerdings *FALSCH* sein muss, getestet. In allen anderen Situationen bleibt die Einfärbung aus.

Siehe auch ISTLEER(), ISTFEHL(), ISTFEHLER(), ISTFORMEL(), ISTNV(), ISTKTEXT(), ISTZAHL(), ISTBEZUG(), ISTTEXT()



Die Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap10 in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *IST-Funktionen*.

ISTNV() ISNA()

ISTNV(*Wert*)

Diese Funktion gibt den Wahrheitswert *WAHR* zurück, wenn die Auswertung des Arguments *Wert* den Fehlerwert *#NV* liefert. In allen anderen Situationen wird der Wahrheitswert *FALSCH* zurück gegeben.

Wert (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), den Sie prüfen möchten.

Diese Funktion ist eine der IST-Funktionen, die in Abhängigkeit des Arguments einen Wahrheitswert über dieses Argument zurückgeben. Das Argument der IST-Funktionen wird vor der Auswertung nicht umgewandelt. Das heißt u.a., eine Zeichenkette, die wie eine Zahl aussieht, bleibt eine Zeichenkette (Text) und wird nicht als Zahl interpretiert. IST-Funktionen finden häufig ihre Anwendung in Zusammenspiel mit der WENN()-Funktion, um in Abhängigkeit der aufgetretenen Situation in Anzeige und Berechnung zu verzweigen. Auch bedingte Formatierungen und Gültigkeitsregeln können auf den Ergebnissen der IST-Funktionen aufbauen.

Beim Einsatz der Funktion SVERWEIS() (oder WVERWEIS()) kann es oft zu der Situation kommen, dass ein bestimmter Suchbegriff in der ersten Spalte (Zeile) der Matrix nicht gefunden wird. Damit sind aber alle weiteren Berechnungen durch den Fehlerwert *#NV* bestimmt. Wenn Sie dies unterbinden wollen, können Sie wie im Beispiel zur Funktion ISTFEHLER() vorgehen, die Formel jedoch durch

```
=WENN(ISTNV(SVERWEIS(B36;D36:E38;2;FALSCH));"nicht gefunden";
SVERWEIS(B36;D36:E38;2;FALSCH))
```

ersetzen². Nur der »Nicht vorhanden«-Fehler wird durch einen Kommentar ausgewertet, andere mögliche Fehler bleiben als solche sichtbar.

ISTLEER(), ISTFEHL(), ISTFEHLER(), ISTFORMEL(), ISTLOG(), ISTKTEXT(), ISTZAHL(), ISTBEZUG(), ISTTEXT(), WENNNV()

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap10` in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *IST-Funktionen*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



ISTTEXT() ISTEKT()

ISTTEXT(*Wert*)

Die Funktion gibt den Wahrheitswert *WAHR* zurück, wenn das Argument *Wert* eine nicht leere Zeichenkette (Text) ist. Andernfalls wird der Wahrheitswert *FALSCH* zurückgegeben.

Wert (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), den Sie prüfen möchten.

Diese Funktion ist eine der IST-Funktionen, die in Abhängigkeit des Arguments einen Wahrheitswert über dieses Argument zurückgeben. Das Argument der IST-Funktionen wird vor der Auswertung nicht umgewandelt. Das heißt u.a., eine Zeichenkette, die wie eine Zahl aus-

Syntax

Definition

Argumente

Hintergrund

² Anwender von Excel 2013 können die Formel durch Verwendung der Funktion WENNNV() verkürzen.

sieht, bleibt eine Zeichenkette (Text) und wird nicht als Zahl interpretiert. IST-Funktionen finden häufig ihre Anwendung im Zusammenspiel mit der WENN()-Funktion, um in Abhängigkeit der aufgetretenen Situation in Anzeige und Berechnung zu verzweigen. Auch bedingte Formatierungen und Gültigkeitsregeln können auf den Ergebnissen der IST-Funktionen aufbauen.

Der Rückgabewert der Funktion ist stets *FALSCH*, es sei denn, das Argument ist nicht leerer Text. Es wird also auch *FALSCH* ermittelt, wenn das Argument ein Bezug auf eine leere Zelle ist. Das ist konsequent, denn ist der Bezug auf eine Zelle gerichtet, die wiederum einen Bezug auf eine leere Zelle enthält, ist es von den eingestellten Optionen abhängig, ob in diesem Fall in der Zelle mit dem Bezug eine 0 (Null) angezeigt werden soll oder nicht.

Der Gegenpart zu dieser Funktion ist die Funktion ISTKTEXT():

ISTTEXT(Wert)=NICHT(ISTKTEXT(Wert))

Praxiseinsatz Bei der Berechnung von Brutto- aus Nettobeträgen können die Ergebnisse dadurch unterschiedlich ausfallen, dass 19 %, 7 % oder keine Mehrwertsteuer erhoben werden.

Sie können diese drei Fälle durch eine Auswahlliste erfassen und im Falle von »keine« die Zahl 0 % eintragen. Sie können aber auch »keine« stehen lassen und in der zu verwendenden Formel auf Text testen.

Angenommen, Sie haben Ihre Berechnungen wie in Abbildung 10.5 vorbereitet.

Artikel	Netto	MwSt-Satz	MwSt	Brutto
A-Produkt	231,00 €	keine	- €	231,00 €
B-Produkt	125,00 €	19%	23,37 €	146,37 €

Abbildung 10.5: Einfache Mehrwertsteuerberechnung

Die Prozentsätze der K-Spalte wurden hier Platzgründen nahe der Tabelle untergebracht, können sich aber außerhalb des Sichtfensters bzw. auf einem anderen Arbeitsblatt befinden. Wichtig ist, dass die Liste im Auswahldialog zu Gültigkeitsregeln für die M-Spalte verwendet wird.

Die Berechnungen der N-Spalte erfolgen nun alle nach der gleichen Formel (die Sie, wie auch die Gültigkeitsregel, einfach bei der Listenerweiterung kopieren können):

=WENN(ISTTEXT(M42);;L42*M42)

Siehe auch ISTLEER(), ISTFEHL(), ISTFEHLER(), ISTFORMEL(), ISTLOG(), ISTNV(), ISTKTEXT(), ISTZAHL(), ISTBEZUG()



Das Beispiel sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap10 in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *IST-Funktionen*.

ISTUNGERADE()



ISTUNGERADE(*Zahl*)

Die Funktion gibt den Wahrheitswert *WAHR* zurück, wenn das Argument *Zahl* eine ungerade Zahl ist. Ist das Argument eine gerade Zahl, wird der Wahrheitswert *FALSCH* zurückgegeben.

Zahl (erforderlich) ist der zu prüfende Ausdruck.

Sie können der Funktion beliebige Ausdrücke als Argumente übergeben. Handelt es sich um Zahlen, werden ganze Zahlen wie erwartet behandelt. Bei gebrochenen Zahlen werden die Nachkommastellen abgeschnitten und der so entstehende Wert ausgewertet. Dadurch ist das Ergebnis von =ISTUNGERADE(2,4) gleich *FALSCH* und =ISTUNGERADE(-1,6) liefert den Wahrheitswert *WAHR*.

Ist das Argument numerisch nicht auswertbar, wird für Zeichenketten und Wahrheitswerte der Fehlerwert #*WERT!* ermittelt, für Fehlerwerte ist das Ergebnis gleich dem Argument.

Der Gegenpart zu dieser Funktion ist die Funktion ISTGERADE():

ISTUNGERADE(*Zahl*)=NICHT(ISTGERADE(*Zahl*))

Als Beispiel kann jenes zur Funktion ISTGERADE() dienen, wobei Sie nunmehr alle Zeilen mit ungeraden Zeilennummern einfärben möchten.

ISTGERADE(), GERADE(), UNGERADE()

Das Beispiel sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap10 in der Arbeitsmappe *Informationen.xlsx* auf dem Arbeitsblatt *IstUngerade*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



ISTZAHL()



ISTZAHL(*Wert*)

Diese Funktion gibt den Wahrheitswert *WAHR* zurück, wenn sich das Argument *Wert* ohne Umwandlung zu einer Zahl (numerischer Wert) auswerten lässt. In allen anderen Fällen ist der Rückgabewert der Wahrheitswert *FALSCH*.

Wert (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), den Sie prüfen möchten.

Diese Funktion ist eine der IST-Funktionen, die in Abhängigkeit des Arguments einen Wahrheitswert über dieses Argument zurückgeben. Das Argument der IST-Funktionen wird vor der Auswertung nicht umgewandelt. Das heißt u.a., eine Zeichenkette, die wie eine Zahl aussieht, bleibt eine Zeichenkette (Text) und wird nicht als Zahl interpretiert. IST-Funktionen finden häufig ihre Anwendung im Zusammenspiel mit der WENN()-Funktion, um in Abhängigkeit der aufgetretenen Situation in Anzeige und Berechnung zu verzweigen. Auch bedingte Formatierungen und Gültigkeitsregeln können auf den Ergebnissen der IST-Funktionen aufbauen.

Syntax

Definition

Argumente

Hintergrund

Beachten Sie, dass diese Funktion *FALSCH* zurückgibt, wenn das Argument *Wert* einen Bezug auf eine leere Zelle darstellt. Das ist konsequent im Sinne der »Informatik«, aber nicht innerhalb von Excel. Haben Sie einen Bezug zu einer leeren Zelle hergestellt, erscheint nämlich in der Zelle mit dem Bezug der Wert 0 (Null), was Sie durch die entsprechende Option (*Excel-Optionen/Erweitert/In Zellen mit Nullwert eine Null anzeigen*) unterbinden können. Gerechnet wird dann aber trotzdem mit 0 (Null).

Praxiseinsatz Im Beispiel zur Funktion ISTTEXT() wurde die Formel zur Berechnung der Mehrwertsteuer in der *N*-Spalte von Abbildung 10.5 durch

```
=WENN(ISTTEXT(M42);;L42*M42)
```

angegeben. Das hat zur Folge, dass im Falle »keine« Mehrwertsteuer dort die Zahl 0 (Null) bzw. nach Formatierung »– €« erscheint. Wollen Sie das nicht, sondern soll eine leere Zelle angezeigt werden, lautet die angepasste Formel

```
=WENN(ISTTEXT(M42);"";L42*M42)
```

Damit wird aber durch

```
=L42+N42
```

zur Berechnung des Bruttobetrags ein Fehler erzeugt. Diesen können Sie durch Einsatz der Funktion ISTZAHL() im Zusammenspiel mit der WENN()-Funktion vermeiden:

```
=WENN(ISTZAHL(N42);L42+N42;L42)
```

Siehe auch ISTLEER(), ISTFEHL(), ISTFEHLER(), ISTFORMEL(), ISTLOG(), ISTNV(), ISTKTEXT(), ISTBEZUG(), ISTTEXT()



Das Beispiel sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap10 in der Arbeitsmappe *Informationen.xlsx* auf dem Arbeitsblatt *IST-Funktionen*.

N() N()

Syntax N(*Wert*)

Definition Diese Funktion gibt den in eine Zahl umgewandelten Wert des Arguments *Wert* zurück.

Argumente *Wert* (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), den Sie in eine Zahl umwandeln möchten.

Hintergrund Im Allgemeinen ist es nicht erforderlich, die Funktion N() anzuwenden, da Excel in den Fällen, wo eine Zahl erwartet werden kann und eine Umwandlung möglich ist, Werte bei Bedarf automatisch umwandelt. Eine Ausnahme bilden die IST-Funktionen in diesem Kapitel. Wohl aus Kompatibilitätsgründen zu anderen Tabellenkalkulations-Programmen steht diese Funktion dennoch zur Verfügung.

Die Funktion N() wandelt Ausdrücke gemäß der folgenden Tabelle 10.4 um.

Argument (Ausdruck)	Rückgabewert
Zahl	Die Zahl selbst
Datum im eingebauten Format	Fortlaufende Zahl des Datums
Zeitangabe	Gebrochene Zahl der Uhrzeit
WAHR	1
FALSCH	0
Fehlerwert	Den Fehlerwert selbst
Text, leere Zelle	0

Tabelle 10.4
Ergebnisse der
Umwandlung
durch die
Funktion *N()*

Datumsangaben in Excel werden als fortlaufende Zahlen gespeichert. In der Standardeinstellung bekommt der 1. Januar 1900 den Wert 1. Genauer: der 31. Dezember 1899, 24 Uhr, bekommt den Wert 1, genauso 1. Januar 1900, 0 Uhr. Bruchteile eines Tags werden in Bruchteilen von 1 erfasst, 12 Uhr hat den Wert 0,5, 18 Uhr den Wert 0,75 und 3:46 Uhr den Wert 0,15694444. Der 1. Januar 1900, 12 Uhr hat damit den Wert 1,5, der 2. Januar 1900, 18 Uhr den Wert 2,75 usw. Bruchteile ohne ganzen Anteil können als Datum formatiert werden, erhalten dann aber den nicht existenten Null-ten Januar 1900 statt des 31. Dezember 1899 zugeordnet.

Hinweis

Auf dieser Basis ist das Rechnen mit Datums- und Zeitangaben möglich (Addition, Subtraktion).

Die Standardeinstellung des ersten Tags kann über *Excel-Optionen/Erweitert/1904-Datumswerte verwenden* geändert werden. Dann bekommt der 1. Januar 1904, 0 Uhr den Wert 0, der 2. Januar 1904, 0 Uhr (= 1. Januar 1904, 24 Uhr) den Wert 1. Bei dieser Umstellung werden alle Datumswerte umbewertet, es ist also Vorsicht geboten.

Auf ein weiteres Beispiel kann hier unter Hinweis auf die eben gemachten Überlegungen verzichtet werden.

Praxiseinsatz

T()

Siehe auch

Einige Tests zu den Umwandlungen und Datumsüberlegungen aus den Hintergrundinformationen finden Sie im Ordner *Ms5-235\Kap10* in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *Sonstige*.



NV()



NA()

NV()

Syntax

Diese Funktion gibt den Fehlerwert *#NV*, der das Nichtvorhandensein eines Werts signalisiert, zurück.

Definition

Die Funktion verlangt keine Argumente.

Argumente

Obwohl die Funktion keine Argumente verlangt, ist dennoch die Angabe des Klammerpaars notwendig. Andernfalls interpretiert Excel Ihre Eingabe als Text.

Hintergrund

Ähnlich wie bei den Funktionen *WAHR()* und *FALSCH()*, deren Ergebnisse auch durch die Angabe der entsprechenden Wahrheitswerte erzeugt werden können, gelangen Sie zum Ergebnis der Funktion *NV()* auch durch die einfache Eingabe des Fehlerwerts *#NV* selbst. Die *NV()*-Funktion steht wohl aus Gründen der Kompatibilität mit anderen Tabellenkalkulationsprogrammen zur Verfügung.

Sie können diese Funktion bzw. ihr Ergebnis nutzen, um explizit leere Zellen als solche zu kennzeichnen. Damit verhindern Sie, dass diese Zellen (in der Regel unbeabsichtigt) in weitere Berechnungen eingehen. Allerdings liefern Formeln, die sich auf solche Zellen beziehen, auch den gleichen Fehlerwert. Damit ist die Funktion eher als Vorsichtsmaßnahme denn als Mittel zur Automatisierung bzw. Verschönerung, wie etwa die IST-Funktionen dieses Kapitels, zu sehen.

Praxeinsatz Die letzte Bemerkung wird am Beispiel der Funktion ISTLEER() deutlich. Dort hatten Sie die Absicht, die Übertragung leerer Zellen zu vermeiden, durch den Einsatz eines Tests auf leere Zellen umgesetzt. Würden Sie stattdessen die Funktion NV() verwenden (oder deren Fehlerwert), kämen Sie Ihrer ursprünglichen Absicht nicht näher, wie Abbildung 10.6 zeigt.

	A	B	C	D	E	F
1						
2		1	1	#NV		
3		#NV	#NV			
4		2	2			
5						
6		1	1	1,5		
7		#NV				
8		2	2			
9						

Abbildung 10.6: Eine Alternative zu Abbildung 10.4

In der C-Spalte stehen zur Illustration des Geschehens die »einfachen« Bezüge zur B-Spalte, aus den Werten soll in D2 der Mittelwert bestimmt werden. Würde in C3 eine 0 (Null) stehen, bildet das nicht die Situation der B-Spalte ab, deshalb haben Sie die leere Zelle der B-Spalte bewusst als leer (=NV() oder #NV) gekennzeichnet. Leider wird dieses #NV nun durch alle Berechnungen durchgetragen.

Der Ausweg besteht in der Verwendung der Funktion ISTNV() in den Zeilen 6 bis 8. Das korrekte Ergebnis ist dann aber mit dem gleichen Aufwand wie unter Einsatz von ISTLEER() erreicht.

Siehe auch ISTLEER(), ISTFEHL(), ISTFEHLER(), ISTLOG(), ISTNV(), ISTKTEXT(), ISTZAHL(), ISTBEZUG(), ISTTEXT(), WENNNV()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap10 in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *NV*.

TYP() TYPE()

Syntax TYP(*Wert*)

Definition Diese Funktion gibt eine Zahl zurück, die den Datentyp des angegebenen Arguments charakterisiert.

Argumente *Wert* (erforderlich) ist ein beliebiger Ausdruck (Zahl, Text, Formel ohne Gleichheitszeichen, Wahrheitswert, Fehlerwert, Bezug oder Name), dessen Typ Sie prüfen möchten.

Diese Funktion steht in engem Zusammenhang zu den sogenannten IST-Funktionen, die in Abhängigkeit des Arguments einen Wahrheitswert über dieses Argument zurückgeben. Sie findet häufig ihre Anwendung im Zusammenspiel mit der WENN()-Funktion, um in Abhängigkeit der aufgetretenen Situation in Anzeige und Berechnung zu verzweigen. Auch bedingte Formatierungen und Gültigkeitsregeln können auf dem Rückgabewert aufbauen.

Um die Funktion einzusetzen, ist die Kenntnis der Zuordnung aus Tabelle 10.5 notwendig.

Argument	Rückgabewert
Zahl	1
Text	2
Wahrheitswert	4
Fehlerwert	16
Matrix	64

Hintergrund

Tabelle 10.5

Die Übersetzung der Zahlen in die ihnen entsprechenden Typen

Mit Ausnahme des letzten Typs können Sie die anderen Ergebnisse durch Einsatz der Funktionen ISTZAHL(), ISTTEXT(), ISTKTEXT(), ISTLOG(), ISTFEHLER() nachbilden. So haben

ISTZAHL(B12)

und

TYP(B12) = 1

den gleichen Wahrheitswert.

Es gibt aber einige IST-Funktionen, die mit TYP() nicht erfasst werden (ISTLEER(), ISTNV(), ISTBEZUG()). Da ISTFEHL() im Falle von #NV keinen Fehler erkennt, gehört auch diese Funktion in diese Gruppe.

Sie stehen vor der Wahl, für eine Verzweigung (WENN()-Funktion, bedingte Formatierung) die Funktion TYP() zu nehmen oder aber eine der IST-Funktionen. Ein Rat kann nicht allgemein gültig sein. Im Falle der IST-Funktionen müssen Sie in der WENN()-Funktion nach den IST-Funktionen »schachteln«, im Falle der TYP()-Funktion »schachteln« Sie nach den Ziffern für den Typ. So entsteht in der gewünschten Zelle nach

```
=WENN(ISTFEHLER(F26);"Fehler";WENN(ISTZAHL(F26);"Zahl";WENN(ISTTEXT(F26);"Text")))
```

die gleiche Information wie nach

```
=WENN(TYP(F26)=16;"Fehler";WENN(TYP(F26)=1;"Zahl";WENN(TYP(F26)=2;"Text")))
```

Sicher ist die Lesbarkeit im ersten Fall besser.

Leider können Sie TYP() nicht verwenden, um festzustellen, ob eine Zelle eine Formel enthält. Steht in der Zelle, auf die Sie Bezug nehmen, eine Formel, gibt TYP() den Typ des Ergebnisses der Formel zurück.

Den Wert 64 erhalten Sie nicht, wenn in der Zelle, auf die Sie Bezug nehmen, eine Matrixformel steht, sondern wenn Sie einen Bereich als Argument von TYP() angeben und Ihre Eingabe mit der Tastenkombination **Strg** + **Alt** + **↵** abschließen (Matrixformel mit TYP() als Bestandteil).

Tipp

**Praxiseinsatz
IST-Funktionen nachbilden**

Sie können die Beispiele zu den IST-Funktionen in diesem Kapitel auch mithilfe der TYP()-Funktion formulieren. So lautet die zweite Bedingung =ISTFEHLER(B3) im Beispiel aus Abbildung 10.1 (siehe Seite 315) nunmehr

```
=(TYP(B3)=16)
```

Auch das Beispiel aus Abbildung 10.5 (siehe Seite 326) lässt sich mit

```
=WENN(TYP(M42)=2;;L42*M42)
```

anstelle von =WENN(ISTTEXT(M42);;L42*M42) lösen.

Und auch die farbige Kennzeichnung von Zellen mit Wahrheitswerten im Beispiel 2 der IST-LOG()-Funktion kann für die »roten« Zellen durch

```
=UND(TYP(G63)=4;G63)
```

und für die »grünen« durch

```
=UND(TYP(G63)=4;NICHT(G63))
```

erfolgen.

**Benutzerdefinierte
Funktion einrichten**

Wollen Sie häufig Informationen, die sich aus dem Typ eines Werts ergeben, für textliche Darstellungen nutzen, ist der Griff zu einer benutzerdefinierten Funktion zu empfehlen.

```
Function TypInWorten(Bereich As Range) As String
    Dim var As Variant
    var = Bereich.Value
    If Not IsEmpty(var) Then
        If IsError(var) Then
            TypInWorten = "Fehler"
        Else
            If var = True Or var = False And Not IsEmpty(var) Then
                TypInWorten = "Wahrheitswert"
            Else
                If IsNumeric(var) Then
                    TypInWorten = "Zahl"
                Else
                    TypInWorten = "Text"
                End If
            End If
        End If
    End If
    TypInWorten = ""
End Function
```

Listing 10.2: Diese Funktion erspart Ihnen den wiederholten Einsatz der WENN()-Funktion

Das Listing 10.2 prüft zunächst, ob die Zelle einen Fehler enthält. Ist das nicht der Fall, wird festgestellt, ob der Zellwert WAHR oder FALSCH ist. Ist auch das nicht der Fall, muss nur noch auf numerische Verwertbarkeit geprüft werden.

Für VBA-Profis: Leider gibt es nicht für alle Arbeitsblattfunktionen ein entsprechendes *Worksheet-Function*-Pendant im Objektmodell. So muss gelegentlich (wie im Beispiel) selbst Hand angelegt werden.

ISTLEER(), ISTFEHL(), ISTFEHLER(), ISTFORMEL(), ISTLOG(), ISTNV(), ISTKTEXT(), ISTZAHL(), ISTBEZUG(), ISTTEXT()

Die Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner `\Ms5-235\Kap10` in der Arbeitsmappe *Informationen.xlsm* auf dem Arbeitsblatt *Sonstige*. Der Code für das zweite Beispiel befindet sich im Modul *modFunction* der gleichen Mappe.



Tip

Siehe auch

ZELLE() CELL()

ZELLE(*Infotyp*;Bezug)

Diese Funktion liefert (textliche) Informationen zu Formatierung, Position oder Inhalt der Zelle, die sich ganz links oben in dem Bereich befindet, der durch das Argument *Bezug* angegeben wurde.

Mithilfe des zweiten Arguments *Bezug* (optional) bestimmen Sie die Zelle, über die Sie die Auskunft, die Sie mit dem ersten Argument *Infotyp* (erforderlich) festgelegt haben, erhalten wollen. Lassen Sie das zweite Argument weg, erhalten Sie die gewünschten Informationen über die jeweils zuletzt geänderte Zelle Ihrer Mappe.

Um die Funktion erfolgreich einzusetzen, ist es notwendig, die möglichen Informationstypen und deren Übergabeform (als Zeichenkette) zu kennen. Die Tabelle 10.6 hält die notwendigen Informationen bereit.

Syntax

Definition

Argumente

Hintergrund

Argument (Text)	Rückgabe
<i>Adresse</i>	Der absolute Bezug zur linken oberen Ecke als Text, ggf. mit Tabellen- und Mappennamen (die Mappe muss geöffnet sein, ansonsten erscheint der Fehlerwert #NV)
<i>Breite</i>	Auf eine ganze Zahl gerundete Spaltenbreite der betreffenden Zelle. Die Einheit ist dabei die Breite eines Zeichens der in der Standard-Formatvorlage festgelegten Schriftart, sofern dies eine Schrift ist, in der alle Zeichen gleich breit sind.
<i>Dateiname</i>	Vollständiger Pfad zur Arbeitsmappe inkl. Angabe der Tabelle, in der sich die betreffende Zelle befindet (leere Zeichenkette, falls diese Mappe noch nie gespeichert wurde)
<i>Farbe</i>	Sind negative Zahlen in der betreffende Zelle farbig formatiert, wird 1, im gegenteiligen Fall 0 (Null) geliefert
<i>Format</i>	Es wird eine textliche Information über das Zahlenformat der betreffenden Zelle ausgegeben, wobei Sie die »Übersetzung« in Tabelle 10.7 finden. Ein »-« am Ende des Texts bedeutet, dass die Zelle für negative Werte farbig formatiert ist. Ein Paar runder Klammern am Ende des Texts weist darauf hin, dass die Zelle zumindest für positive (!) Werte mit Klammern formatiert ist.
<i>Inhalt</i>	Liefert den Wert (nicht die Formel) der betreffenden Zelle
<i>Klammern</i>	Ergibt den Wert 1, wenn wenigstens das Format positiver Zahlen als Zahlen mit Klammern definiert wurde, ansonsten 0 (Null) ▶

Tabelle 10.6

Diese Argumente werden als Zeichenketten an ZELLE() übergeben

Argument (Text)	Rückgabe
<i>Präfix</i>	Informiert über die Ausrichtung von Text in der betreffenden Zelle (' = linksbündig, " = rechtsbündig, ^ = zentriert, \ = wiederholend ausgefüllt), steht kein Text in der Zelle, wird eine leere Zeichenkette zurückgegeben
<i>Schutz</i>	Liefert 1, wenn die betreffende Zelle im Falle des Blattschutzes gesperrt ist, sonst 0 (Null)
<i>Spalte</i>	Diese Information betrifft die Spaltennummer der betreffenden Zelle
<i>Typ</i>	Gibt Auskunft über den Datentyp in der betreffenden Zelle: l = Text, b = leer, w = sonst
<i>Zeile</i>	Diese Information betrifft die Zeilennummer der betreffenden Zelle

Hinweis Befinden sich Mappennamen, Tabellenbezeichnungen und/oder Zelladressen in den zurückgegebenen Zeichenketten, ist der Aufbau immer gleich:

Ordnerpfad\[Mappename]Arbeitsblatt!absolute Adresse

also etwa

F:\Kapitel 10\Ms5-235\Kap10\[Informationen.xls]Zelle

(Pfad bis zum Arbeitsblatt) oder auch

[Mappe2.xls]Tabelle1!\$A\$1

(Bezug zur Zelle).

Die Funktion liefert einige »kryptische« Informationen, die es zu übersetzen gilt. Die Tabelle 10.7 hilft bei diesem Vorhaben.

Tabelle 10.7
Zahlenformate –
codiert und im
Klartext

Das liefert ZELLE()	Und das bedeutet es
S	Standard
F0	0
0,0	#,##0
F2	0,00
,2	#,##0,00
W0	Währungsangaben ohne Nachkommastellen
W0-	Währungsangaben ohne Nachkommastellen, negative Werte rot
W2	Währungsangaben mit zwei Nachkommastellen
W2-	Währungsangaben mit zwei Nachkommastellen, negative Werte rot
P0	0%
P2	0,00%
E2	0,00E+00
S	# ??/?? oder # ??/??
D1	Alle Datumsangaben nach dem Muster TT.MM.JJ, wobei die Anzahl von T, M und J variieren kann
D2	Alle Datumsangaben nach dem Muster TT.MM., wobei die Anzahl von T und M variieren kann



Das liefert ZELLE()	Und das bedeutet es
D3	Alle Datumsangaben nach dem Muster MM.JJ, wobei die Anzahl von M und J variieren kann
D4	Alle Datumsangaben, in denen die Reihenfolge von Tag und Monat »vertauscht« wurde, etwa MM/TT/JJ
D5	Alle Datumsangaben ohne Jahreszahl, in denen die Reihenfolge von Tag und Monat »vertauscht« wurde, etwa MM/TT
U1	h:mm:ss AM/PM
U2	h:mm AM/PM
U3	h:mm:ss
U4	h:mm

Die Excel-Hilfe weicht von der Tabelle 10.7 fehlerhaft in wesentlichen Punkten ab, da u.a. auf keinerlei Sprachspezifika eingegangen wurde³. Außerdem erzeugen gewisse »eingebaute« Formate (etwa für Datum und Uhrzeit) die Anzeige von *S*, und erst ein benutzerdefiniertes Format nach den Mustern der Tabelle bringt die erwartete Information.

Hinweis

Die ZELLE()-Funktion wird zwar aus Kompatibilitätsgründen mit anderen Tabellenkalkulationsprogrammen bereitgestellt. Dennoch bieten sich einige interessante Einsatzmöglichkeiten.

Die Anzeige der Rückgabewerte muss nicht immer aktuell sein. Speichern Sie etwa eine Arbeitsmappe, wird der Dateiname nicht automatisch aktualisiert. Ändern Sie das Format negativer Zahlen hinsichtlich der Farbe, wird dies nicht sofort durch die Anzeige der Zahl 1 oder 0 quittiert. Die Aktualisierung erfolgt erst nach einer Neuberechnung, die Sie natürlich auch mit der F9-Taste manuell erzwingen können.

Wichtig

Die Beispiele zeigen die Anwendung der Funktion und was dabei zu beachten ist.

Sie können versuchen, Zelländerungen, die der Nutzer vornimmt, durch visuelle Effekte zu verdeutlichen und somit zu warnen. So zeigt etwa

```
=WENN(ZELLE("Adresse")="$C$63";"Achtung";"OK")
```

in C64 eine Änderung der Nachbarzelle C63 an. Wollen Sie keine weitere Zelle opfern, lasse sich über eine bedingte Formatierung der Zelle C63 durch die Bedingung

```
=(ZELLE("Adresse")="$C$63")
```

eine farbliche Hervorhebung erzeugen. Verantwortlich für das Funktionieren ist der fehlende Bezug in der Funktion ZELLE(), da somit immer die aktuell verarbeitete Zelle protokolliert wird.

Einen Zellschutz stellt das Verfahren natürlich nicht dar. So setzt die Bearbeitung einer anderen Zelle alles wieder zurück, ohne dass C63 selbst in den Urzustand gebracht wurde. Auch bringt ein sofortiges Reagieren des Nutzers auf den visuellen Hinweis mit *Bearbeiten/Rückgängig* die Zelle in den Urzustand, jedoch verschwindet der Warnhinweis erst nach der Bearbeitung einer anderen Zelle.

Praxiseinsatz**Änderungen
visualisieren**

³ Etwa G für General statt S für Standard, C für Currency statt W für Währung.

Eingabe der Funktion vereinfachen

Die Funktion ZELLE() verlangt als erstes Argument eine Zeichenkette. Wollen Sie die Funktion mehrfach nutzen, lassen sich die notwendigen Begriffe in einer kleinen Liste auf einem Arbeitsblatt bereithalten und statt der Formel =ZELLE("Typ",B52) können Sie nun =ZELLE(\$B\$51;B52) verwenden, falls in B51 das Wort »Typ« steht.

Gesperrte Zellen visualisieren

Die Absicht, gesperrte Zellen in einem geschützten Arbeitsblatt zu ändern, endet mit einer (manchmal doch störenden) Fehlermeldung. Wollen Sie dem Anwender gesperrte Zellen visuell näher bringen, können Sie diese einfärben. Andersherum kann es gelegentlich ebenso sinnvoll sein, bearbeitbare Zellen einzufärben. Der Aufwand für Sie kann dadurch reduziert werden, dass Sie im gesamten Arbeitsblatt oder seinen definierten Teilen eine bedingte Formatierung angeben (Sie geben diese nur für eine Zelle an, den Rest erledigt *Format übertragen* vor dem endgültigen Sperren/Entsperren), die auf die Sperrung anspricht. Die Abbildung 10.7 zeigt eine mögliche Variante.

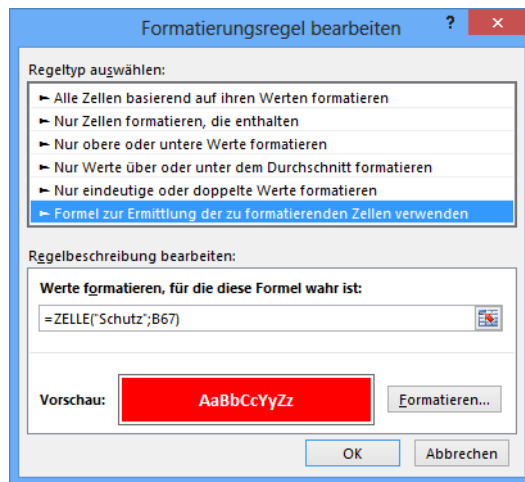


Abbildung 10.7: Bedingte Formatierung zur Visualisierung gesperrter Zellen

Dass die Eingabe =ZELLE("Schutz";B67) anstelle der ausführlichen Anweisung =(ZELLE("Schutz";B67)=1) funktioniert, liegt daran, dass im Falle der Sperrung die Zahl 1 geliefert wird, die Excel als WAHR interpretiert (0 würde als FALSCH durchgehen).

Fortlaufende Nummerierungen

Mithilfe von

=ZELLE("Zeile";C72)-71 & ". "

lässt sich auf einfache Weise beginnend ab Zelle C72 abwärts eine ab 1 beginnende fortlaufende Nummerierung (mit Punkt) erzeugen. Der Eintrag in die Zelle selbst führt nicht zu einem Zirkelbezug!

Siehe auch

Alle Informationsfunktionen



Die Beispiele sowie einige Tests zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap10 in der Arbeitsmappe *Informationen.xlsx* auf dem Arbeitsblatt *Zelle*.

Kapitel 11

Statistische Funktionen

Darstellung der Funktionen in diesem Kapitel	346	F.VERT.RE() / FVERT()	389
ACHSENABSCHNITT()	347	F.VERT()	392
ANZAHL()	349	GAMMA	393
ANZAHL2()	351	GAMMA.INV() / GAMMAINV()	394
ANZAHLLEEREZELLEN()	352	GAMMALN()	396
BESTIMMTHEITSMASS()	352	GAMMALN.GENAU()	397
BETA.INV() / BETAINV()	355	GAMMA.VERT() / GAMMVERT()	398
BETA.VERT() / BETAVERT()	358	GAUSS()	401
BINOM.INV() / KRITBINOM()	360	GEOMITTEL()	403
BINOM.VERT() / BINOMVERT()	362	GESTUTZTMITTEL()	405
BINOM.VERT.BEREICH()	365	G.TEST() / GTEST()	406
CHIQU.INV.RE() / CHIINV()	367	HARMITTEL()	408
CHIQU.INV()	369	HÄUFIGKEIT()	411
CHIQU.TEST() / CHITEST()	370	HYPGEOM.VERT() / HYPGEOMVERT()	415
CHIQU.VERT.RE() / CHIVERT()	374	KGRÖSSTE()	418
CHIQU.VERT()	376	KKLEINSTE()	420
EXPON.VERT() / EXPONVERT()	376	KONFIDENZ.NORM() / KONFIDENZ()	421
F.INV.RE() / FINV()	379	KONFIDENZ.T()	425
F.INV()	382	KORREL()	426
FISHER()	383	KOVAR()	428
FISHERINV()	387	KOVARIANZ.P()	431
F.TEST() / FTEST()	387	KOVARIANZ.S()	432
		KURT()	433

LOGNORM.INV() / LOGINV()	436	RANG.GLEICH()	490
LOGNORM.VERT() / LOGNORMVERT()	437	RANG.MITTELW()	491
MAX()	439	RGP()	492
MAXA()	440	RKP()	497
MEDIAN()	441	SCHÄTZER()	500
MIN()	443	SCHIEFE()	503
MINA()	444	SCHIEFE.P()	506
MITTELABW()	445	STABW.S() / STABW()	508
MITTELWERT()	447	STABWA()	511
MITTELWERTA()	449	STABW.N() / STABWN()	513
MITTELWERTWENN()	451	STABWNA()	514
MITTELWERTWENNS()	454	STANDARDISIERUNG()	516
MODUS.EINF() / MODALWERT()	457	STEIGUNG()	519
MODUS.VIELF()	459	STFEHLERYX()	521
NEGBINOM.VERT() / NEGBINOMVERT()	460	SUMQUADABW()	524
NORM.INV() / NORMINV()	462	T.INV.2S() / TINV()	526
NORM.S.INV() / STANDNORMINV()	464	T.INV()	528
NORM.S.VERT() / STANDNORMVERT()	466	TREND()	528
NORM.VERT() / NORMVERT()	468	T.TEST() / TTEST()	533
PHI()	472	T.VERT.2S() / TVERT()	539
PEARSON()	472	T.VERT()	541
POISSON.VERT() / POISSON()	476	T.VERT.RE()	541
QUANTIL()	479	VAR.S() / VARIANZ()	542
QUANTIL.EXKL()	481	VARIANZA()	544
QUANTIL.INKL()	481	VAR.P() / VARIANZEN()	546
QUANTILSRANG()	482	VARIANZENA()	547
QUARTILSRANG.EXKL()	483	VARIATION()	549
QUARTILSRANG.INKL()	484	VARIATIONEN()	553
QUARTILE()	485	VARIATIONEN2()	555
QUARTILE.EXKL()	488	WAHRSCBEREICH()	557
QUARTILE.INKL()	488	WEIBULL.VERT() / WEIBULL()	559
RANG()	489	ZÄHLENWENN()	562
		ZÄHLENWENNS()	564



Frage- stellungen

In diesem Kapitel versuchen wir, Ihnen so plastisch und einfach wie möglich die statistischen Funktionen in Microsoft Excel darzustellen und näher zu bringen. Dies bedeutet jedoch nicht, dass ein Ungeübter die Statistik auch sofort versteht und umsetzen kann. Grundlagen der Mathematik sind auf jeden Fall notwendig, um die Aussagen der Statistik und der statistischen Funktionen deuten zu können.

Die Statistik befasst sich vor allem mit folgenden Fragen:

- ▶ Wie soll man Daten beschreiben?
- ▶ Welche Schlüsse/Ergebnisse kann man aus Daten ziehen?

Durch statistische Auswertungen sind wir in der Lage, umfangreiche und komplizierte Datensätze möglichst anschaulich darzustellen. Außerdem leistet die Statistik, wie nachfolgend aufgelistet, Folgendes:

- ▶ Sie zeigt, wie man große Datenmengen durch einige wenige Zahlen und eventuelle Zusatzangaben mit minimalem Informationsverlust zusammenfassen kann
- ▶ Sie beschreibt einfache und wichtige Modelle zur Erfassung der Natur (z.B. linearer oder quadratischer Zusammenhang zweier Größen oder ihrer Logarithmen; additives oder multiplikatives Zusammenwirken mehrerer Faktoren)
- ▶ Sie bietet häufig benutzbare Modelle für die zufälligen Schwankungen und zufälligen Fehler, die in Daten beobachtet werden können (z.B. Binomial- und Poisson-Verteilung für die Anzahl der Fälle, in denen ein bestimmtes Ereignis zufällig eintritt; Normalverteilung – Gaußsche Glockenkurve – für die Größe von Messfehlern)
- ▶ Sie untersucht und vergleicht verschiedene Versuchspläne gleichen Umfangs zur Messung mehrerer Effekte oder zur Prüfung einer Hypothese, oder zur schrittweisen Suche eines Optimums oder zum Ziehen einer Stichprobe aus einer strukturierten Grundgesamtheit. Dadurch kann, bei gleicher Genauigkeit, der Versuchsaufwand oft stark reduziert werden.
- ▶ Sie prüft, inwieweit beobachtete Abweichungen von einem Modell dem Zufall zugeschrieben werden können, also ob Daten und Modelle oder Hypothese im Rahmen der zufälligen Fehler miteinander vereinbar sind oder nicht (Tests)
- ▶ Sie liefert eine möglichst gute Anpassung der unbekanntenen Konstanten (Parameter) eines Modells an die Daten, unter Berücksichtigung des Vorhandenseins von zufälligen (eventuell auch groben) Fehlern (Schätzungen – genauer Punktschätzungen). Gleichzeitig gibt sie die ungefähre Genauigkeit dieser Anpassung an (Standardfehler, Vertrauensbereiche).

Funktion 2010-2013 / 2003-2007	Beschreibung
ACHSENABSCHNITT()	Gibt den Schnittpunkt der linearen Regressionsgeraden zurück
ANZAHL()	Zählt die Anzahl der Zahlen in der Liste mit Argumenten
ANZAHL2()	Zählt die Anzahl der Werte in der Liste mit Argumenten
ANZAHLLEEREZELLEN()	Gibt die Anzahl der leeren Zellen in einem Bereich an
BESTIMMTHEITSMASS()	Gibt das Quadrat des Pearsonschen Korrelationskoeffizienten zurück
BETA.INV() (2010) / BETAINV()	Gibt das Quantil einer angegebenen Betaverteilung zurück
BETA.VERT() (2010) / BETAVERT()	Gibt die Werte der kumulierten Betaverteilungsfunktion zurück ▶

Tabelle 11.1
Die statistischen
Funktionen im
Überblick

Funktion 2010-2013 / 2003-2007	Beschreibung
BINOM.VERT() (2010) / BINOMVERT()	Gibt Wahrscheinlichkeiten einer binomialverteilten Zufallsvariablen zurück
BINOM.VERT.BEREICH() (2013)	Gibt die Erfolgswahrscheinlichkeit eines Versuchsergebnisses als Binomialverteilung zurück
QHIQU.INV.RE() (2010) / CHIINV()	Gibt Quantile der (rechtsseitigen) Verteilungsfunktion (1-Alpha) der Chi-Quadrat-Verteilung zurück
QHIQU.INV() (2010)	Gibt Quantile der linksseitigen Verteilungsfunktion der Chi-Quadrat-Verteilung zurück
CHI.TEST() (2010) / CHITEST()	Gibt die Teststatistik eines Unabhängigkeitstests zurück
CHI.VERT.RE() (2010) / CHIVERT()	Gibt Werte der (rechtsseitigen) Verteilungsfunktion (1-Alpha) einer Chi-Quadrat-verteilten Zufallsgröße zurück
CHI.VERT() (2010)	Gibt Werte der linksseitigen Verteilungsfunktion (1-Alpha) einer Chi-Quadrat-verteilten Zufallsgröße zurück
EXPON.VERT() (2010) / EXPONVERT()	Gibt Wahrscheinlichkeiten einer exponentialverteilten Zufallsvariablen zurück
F.INV.RE() (2010) / FINV()	Gibt Quantile der rechtsseitigen F-Verteilung zurück
F.INV() (2010)	Gibt Quantile der linksseitigen F-Verteilung zurück
FISHER()	Gibt die Fisher-Transformation zurück
FISHERINV()	Gibt die Umkehrung der Fisher-Transformation zurück
F.TEST() (2010) / FTEST()	Gibt die Teststatistik eines F-Tests zurück
F.VERT.RE() (2010) / FVERT()	Gibt Werte der Verteilungsfunktion (1-Alpha) einer F-verteilten Zufallsvariablen zurück
F.VERT() (2010)	Gibt Werte der Verteilungsfunktion (1-Alpha) einer (linksseitigen) F-verteilten Zufallsvariablen zurück
GAMMA() (2013)	Gibt den Wert der Gammafunktion für eine bestimmte Zahl zurück
GAMMA.INV() (2010)/ GAMMAINV()	Gibt Quantile der Gammaverteilung zurück
GAMMALN()	Gibt den natürlichen Logarithmus der Gammafunktion zurück
GAMMALN.GENAU() (2010)	Gibt den natürlichen Logarithmus der Gammafunktion auf 15 Stellen zurück
GAMMA.VERT() (2010) / GAMMVERT()	Gibt die Wahrscheinlichkeiten einer gammaverteilten Zufallsvariablen zurück
GAUSS() (2013)	Gibt die Wahrscheinlichkeit, mit der eine standardnormal-verteilte Zufallsvariable zwischen dem Mittelwert der Grundgesamtheit und x Standardabweichungen vom Mittelwert liegt, zurück
GEOMITTEL()	Gibt das geometrische Mittel zurück
GESTUTZTMITTEL()	Gibt den Mittelwert einer Datengruppe zurück, ohne die Randwerte zu berücksichtigen
GTEST()	Gibt den einseitigen Wahrscheinlichkeitswert für einen Gaußtest (Normalverteilung) zurück



Funktion 2010-2013 / 2003-2007	Beschreibung
HARMITTEL()	Gibt das harmonische Mittel zurück
HÄUFIGKEIT()	Gibt eine Häufigkeitsverteilung als einspaltige Matrix zurück
HYPGEOM.VERT() (2010) / HYPGEOMVERT()	Gibt Wahrscheinlichkeiten einer hypergeometrisch-verteilten Zufallsvariablen zurück
KGRÖSSTE()	Gibt den k-größten Wert einer Datengruppe zurück
KKLEINSTE()	Gibt den k-kleinsten Wert einer Datengruppe zurück
KONFIDENZ.NORM() (2010) / KONFIDENZ()	Gibt das 1-Alpha Konfidenzintervall für den Erwartungswert einer Zufallsvariablen zurück
KONFIDENZ.T() (2010)	Gibt das Konfidenzintervall für den Erwartungswert einer (Student) t-verteilten Zufallsvariablen zurück
KORREL()	Gibt den Korrelationskoeffizienten zweier Reihen von Merkmalsausprägungen zurück
KOVARIANZ.P() / KOVAR()	Gibt die Kovarianz einer Grundgesamtheit, den Mittelwert der für alle Datenpunktpaare gebildeten Produkte der Abweichungen
KOVARIANZ.T() (2010)	Gibt die Kovarianz einer Stichprobe, den Mittelwert für alle Datenpunktpaare gebildeten Produkte der Abweichungen zurück
BINOM.INV() (2010) / KRITBINOM()	Gibt den kleinsten Wert zurück, für den die kumulierten Wahrscheinlichkeiten der Binomialverteilung kleiner oder gleich einer Grenzwahrscheinlichkeit sind
KURT()	Gibt die Kurtosis (Exzess) einer Datengruppe zurück
LOGINV()	Gibt Perzentile der Lognormalverteilung zurück
LOGNORM.VERT() (2010) / LOGNORMVERT()	Gibt Werte der Verteilungsfunktion einer lognormalverteilten Zufallsvariablen zurück
MAX()	Gibt den größten Wert in einer Liste mit Argumenten zurück
MAXA()	Gibt den größten Wert in einer Liste mit Argumenten zurück. Dazu zählen Zahlen, Text und Wahrheitswerte.
MEDIAN()	Gibt den Median der angegebenen Zahlen zurück
MIN()	Gibt den kleinsten Wert in einer Liste mit Argumenten zurück
MINA()	Gibt den kleinsten Wert in einer Liste mit Argumenten zurück. Dazu zählen Zahlen, Text und Wahrheitswerte.
MITTELABW()	Gibt die durchschnittliche absolute Abweichung von Datenpunkten von ihrem Mittelwert zurück
MITTELWERT()	Gibt den Mittelwert der zugehörigen Argumente zurück
MITTELWERTA()	Gibt den Mittelwert der zugehörigen Argumente zurück. Dazu gehören Zahlen, Text und Wahrheitswerte.
MITTELWERTWENN()	Der Durchschnittswert (arithmetisches Mittel) für alle Zellen in einem Bereich, die einem angegebenen Kriterium entsprechen, wird zurückgegeben
MITTELWERTWENNS()	Gibt den Durchschnittswert (arithmetisches Mittel) aller Zellen zurück, die mehreren Kriterien entsprechen ►

Funktion 2010-2013 / 2003-2007	Beschreibung
MODUS.EINF() (2010) / MODALWERT()	Gibt den am häufigsten vorkommenden Wert in einer Datengruppe zurück
MODUS.VIELF() (2010)	Gibt ein vertikales Array der am häufigsten vorkommenden oder wiederholten Werte in einem Array oder Datenbereich zurück
NEGBINOMVERT()	Gibt Wahrscheinlichkeiten einer negativbinomialverteilten Zufallsvariablen zurück
NORMINV() (2010) / NORMINV()	Gibt Quantile der Normalverteilung zurück
NORM.S.INV() (2010) / STANDNORMINV()	Gibt Quantile der Standardnormalverteilung zurück
NORM.S.VERT() (2010) / STANDNORMVERT()	Gibt Wahrscheinlichkeiten einer standardnormalverteilten Zufallsvariablen zurück
NORM.VERT() (2010)/ NORMVERT()	Gibt Wahrscheinlichkeiten einer normalverteilten Zufallsvariablen zurück
PEARSON()	Gibt den Pearsonschen Korrelationskoeffizienten zurück
POISSON.VERT() (2010) / POISSON()	Gibt Wahrscheinlichkeiten einer poissonverteilten Zufallsvariablen zurück
QUANTIL()	Gibt das Alpha-Quantil einer Gruppe von Daten zurück
QUANTILSRANG()	Gibt den prozentualen Rang (Alpha) eines Werts in einer Gruppe von Daten zurück
QUANTILSRANG.EXKL() (2010)	Gibt den prozentualen Rang (Alpha) eines Werts in einem Dataset als Prozentsatz des Datasets (0..1 ausschließlich) zurück
QUANTILSRANG.INKL() (2010)	Gibt den prozentualen Rang (Alpha) eines Werts in einem Dataset als Prozentsatz des Datasets (0..1 einschließlich) zurück
QUARTILE()	Gibt die Quartile einer Datengruppe zurück
QUARTILE.EXKL() (2010)	Gibt die Quartile eines Datasets zurück, basierend auf Perzentilwerten von 0..1 ausschließlich
QUARTILE.INKL() (2010)	Gibt die Quartile eines Datasets zurück, basierend auf Perzentilwerten von 0..1 einschließlich
PHI() (2013)	Gibt den Wert der Dichtefunktion für eine Standardnormalverteilung zurück
RANG()	Gibt den Rang zurück, den eine Zahl innerhalb einer Liste von Zahlen einnimmt
RANG.GLEICH() (2010)	Gibt den Rang, den eine Zahl innerhalb einer Liste von Zahlen einnimmt zurück: die Größe relativ zu anderen Werten in der Liste; wenn mehrere Werte die gleiche Rangzahl aufweisen, wird der oberste Rang dieser Gruppe von Werten zurückgegeben
RANG.MITTELW() (2010)	Gibt den Rang, den eine Zahl innerhalb einer Liste von Zahlen einnimmt zurück: die Größe relativ zu anderen Werten in der Liste; wenn mehrere Werte die gleiche Rangzahl aufweisen, wird die durchschnittliche Rangzahl zurückgegeben
RGP()	Gibt die Parameter eines linearen Trends zurück
RKP()	Gibt die Parameter eines exponentiellen Trends zurück



Funktion 2010-2013 / 2003-2007	Beschreibung
SCHÄTZER()	Gibt den Schätzwert für einen linearen Trend zurück
SCHIEFE()	Gibt die Schiefe einer Verteilung zurück
SCHIEFE.P() (2013)	Gibt die Schiefe einer Verteilung auf Basis einer Grundgesamtheit zurück
STABW.S() (2010) / STABW()	Schätzt die Standardabweichung ausgehend von einer Stichprobe
STABWA()	Schätzt die Standardabweichung ausgehend von einer Stichprobe, die Zahlen, Text und Wahrheitswerte enthält
STABW.N() (2010) / STABWN()	Berechnet die Standardabweichung auf der Grundlage der Grundgesamtheit
STABWNA()	Berechnet die Standardabweichung auf der Grundlage der Grundgesamtheit, die Zahlen, Text und Wahrheitswerte enthält
STANDARDISIERUNG()	Gibt einen standardisierten Wert zurück
NORM.S.INV() (2010) / STANDNORMINV()	Gibt Quantile der Standardnormalverteilung zurück
NORM.S.VERT() (2010) / STANDNORMVERT()	Gibt Wahrscheinlichkeiten einer standardnormalverteilten Zufallsvariablen zurück
STEIGUNG()	Gibt die Steigung der linearen Regressionsgeraden zurück
STFEHLERYX()	Gibt den Standardfehler der geschätzten y-Werte für alle x-Werte der Regression zurück
SUMQUADABW()	Gibt die Summe von quadrierten Abweichungen zurück
T.INV.2S() (2010) / TINV()	Gibt (zweiseitige) Quantile der (Student) t-Verteilung zurück
T.INV() (2010)	Gibt linksseitige Quantile der (Student) t-Verteilung zurück
TREND()	Gibt Werte zurück, die sich aus einem linearen Trend ergeben
T.TEST() (2010) / TTEST()	Gibt die Teststatistik eines Studentschen t-Tests zurück
T.VERT.2S() (2010) / TVERT()	Liefert die (Student) t-Verteilung für zwei Endflächen zurück
T.VERT() (2010)	Liefert die (Student) t-Verteilung der linken Endfläche zurück
T.VERT.RE() (2010)	Liefert die (Student) t-Verteilung der rechten Endfläche zurück
VAR.S() (2010) / VARIANZ()	Schätzt die Varianz auf der Grundlage einer Stichprobe
VAR.P() (2010) / VARIANZEN()	Berechnet die Varianz auf der Grundlage der Grundgesamtheit
VARIANZA()	Schätzt die Varianz ausgehend von einer Stichprobe, die Zahlen, Text und Wahrheitswerte enthält
VARIANZENA()	Berechnet die Varianz auf der Grundlage einer Grundgesamtheit, die Zahlen, Text und Wahrheitswerte enthält
VARIATION()	Gibt Werte zurück, die sich aus einem exponentiellen Trend ergeben
VARIATIONEN()	Gibt die Anzahl der Permutationen für eine bestimmte Anzahl von Objekten zurück
VARIATIONEN2() (2013)	Gibt die Anzahl der Möglichkeiten – mit Wiederholungen – für eine angegebene Menge von Objekten zurück
WAHRSCHBEREICH()	Gibt die Wahrscheinlichkeit für ein von zwei Werten eingeschlossenes Intervall zurück

Funktion 2010-2013 / 2003-2007	Beschreibung
WEIBULL.VERT() (2010) / WEIBULL()	Gibt Wahrscheinlichkeiten einer weibullverteilten Zufallsvariablen zurück
ZÄHLENWENN()	Die Zellen eines Bereichs, deren Inhalte mit den Suchkriterien übereinstimmen, werden gezählt
ZÄHLENWENNS()	Die Zellen eines Bereichs, deren Inhalte mit mehreren Kriterien übereinstimmen, werden gezählt

Neu in Excel 2007 Die nachfolgend aufgelisteten Funktionen stehen Ihnen erst ab Excel 2007 zur Verfügung:

- ▶ MITTELWERTWENN()
- ▶ MITTELWERTWENNS()
- ▶ ZÄHLENWENNS()

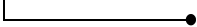
Achtung Zahlreiche statistische Funktionen sind für Excel 2010 überarbeitet worden: Einerseits wurden ihre Genauigkeit und Leistungsfähigkeit erhöht, andererseits wurden Benutzerwünsche eingearbeitet, was sich in der Anzahl und Art der Parameter und auch in der Anwendung der Funktion niederschlug. Darüber hinaus wurden einige Funktionsnamen an Namenskonventionen angepasst, ohne dass an der Funktion selbst etwas geändert wurde.

Um vorhandene Excel-Tabellen mit den ursprünglichen Funktionen nicht hinfällig werden zu lassen, wurden die vertrauten Funktionsvarianten beibehalten (und in die neue Kategorie *Kompatibilitätsfunktionen* eingeordnet) und die überarbeiteten Varianten mit einem leicht geänderten Namen (zu erkennen an einem Punkt im Namen) zusätzlich aufgenommen. Leider erkennt man am Namen aber nicht, zu welcher Gruppe der geänderten Funktionen die jeweilige gehört: Wurde nur der Name geändert oder der Algorithmus oder ist die Funktion neu? Dabei sollen Ihnen die folgenden Listen helfen.

In diesem Buch soll, um den Umfang zu begrenzen, der Schwerpunkt auf die Funktionen gelegt werden, die für Excel 2010-2013 charakteristisch sind. Daher werden die althergebrachten Varianten nicht ausführlich vorgestellt, sondern nur mit einem Hinweis auf die Neuerungen versehen.

Neu in Excel 2010 Folgende der aufgelisteten Funktionen stehen Ihnen erst ab Excel 2010 zur Verfügung: (nur neue Funktionen, ohne Funktionen, bei denen sich der Algorithmus oder lediglich der Name geändert hat)

- ▶ CHIQU.INV()
- ▶ CHIQU.VERT()
- ▶ F.INV()
- ▶ F.VERT
- ▶ GAMMALN.GENAU()
- ▶ KONFIDENZ.T()
- ▶ KOVARIANZ.P()
- ▶ KOVARIANZ.S()
- ▶ MODUS.VIELF()

- 
- ▶ QUANTIL.EXKL()
 - ▶ QUANTIL.INKL()
 - ▶ QUANTILSRANG.EXKL()
 - ▶ QUANTILSRANG.INKL()
 - ▶ QUARTILE.EXKL()
 - ▶ QUARTILE.INKL()
 - ▶ RANG.GLEICH()
 - ▶ RANG.MITTELW()
 - ▶ T.INV()
 - ▶ T.VERT()
 - ▶ T.VERT.RE()

In der folgenden Auflistung finden Sie die Funktionen, bei denen die Genauigkeit und die Ausführungsgeschwindigkeit verbessert wurden.

- ▶ BETA.INV()
- ▶ BETA.VERT()
- ▶ BINOM.INV()
- ▶ BINOM.VERT()
- ▶ CHIQU.INV.RE()
- ▶ CHIQU.TEST()
- ▶ F.INV.RE()
- ▶ F.VERT.RE()
- ▶ GAMMA.INV()
- ▶ GAMMA.VERT()
- ▶ GAMMALN()
- ▶ GEOMITTEL()
- ▶ HYPGEOM.VERT()
- ▶ LOGNORM.INV()
- ▶ LOGNORM.VERT()
- ▶ NEGBINOM.VERT()
- ▶ NORM.INV()
- ▶ NORM.S.INV()
- ▶ NORM.S.VERT()
- ▶ NORM.VERT()
- ▶ POISSON.VERT()
- ▶ RGP()

- ▶ STABW.S()
- ▶ T.INV.2S()
- ▶ T.VERT.2S()
- ▶ T.VERT.RE()
- ▶ VAR.S()

Hinweis Die Funktionen, bei denen sich lediglich der Name geändert hat, entnehmen Sie bitte der Tabelle 11.1 auf Seite 339.

Neu in Excel 2013 Abschließend stellen wir Ihnen natürlich noch die in Excel 2013 ergänzten Funktionen vor. Sie sind in Tabelle 11.1 mit dem Hinweis »(2013)« versehen. Dazu zählen:

- ▶ BINOM.VERT.BEREICH()
- ▶ GAMMA()
- ▶ GAUSS()
- ▶ PHI()
- ▶ SCHIEFE.P()
- ▶ VARIATIONEN2

Darstellung der Funktionen in diesem Kapitel

Die Mehrzahl der in diesem Kapitel genannten und dargestellten Funktionen haben wir anhand eines einheitlichen Beispiels verdeutlicht und versuchen Ihnen auf diese Weise aufzuzeigen, welche Fragestellung mit welcher Funktion beantwortet werden kann.

Das Szenario Ein kleines Unternehmen, das sich auf die Herstellung von Software spezialisiert hat, vermarktet seine Produkte sowohl über das Internet bzw. über seine Webseiten als auch über Außendienstmitarbeiter.

Regelmäßig werden Auswertungen erstellt, die beispielsweise die Umsätze der über das Web verkauften Produkte wiedergeben oder aber auch die Anzahl der Webzugriffe und Klicks darstellt.

Da Webseiten immer mehrere Zwecke gleichzeitig erfüllen, kann das Unternehmen neben seiner eigenen Präsentation und der Kommunikation nach Außen vor allem auch wertvolle Informationen über die Besucher an sich, aber auch über deren Interessegebiete generieren.

Es liegen folgende Daten vor:

- ▶ Zugriffe auf die Webseite pro Tag
- ▶ Zugriff pro Unterseite (Produkte, Wissen, Events usw.)
- ▶ Gesamtumsatz pro Tag und zu jedem Produkt
- ▶ Anzahl der Klicks beim Versand von Newslettern

- ▶ Kosten und Zeit der Webseitenpflege
- ▶ Besuche der Außendienstmitarbeiter pro Tag
- ▶ Umsätze der Außendienstmitarbeiter pro Monat

Aus didaktischen Gründen haben wir für einige Funktionen andere Beispiele gewählt, um die ohnehin schon für »Nichtstatistiker« schwer verdauliche Kost etwas verständlicher zu machen.

Hinweis

Dies betrifft vor allem den Bereich »Wahrscheinlichkeit«.

Die oben genannten und dem Unternehmen bereits vorliegenden Daten der Webauswertung finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe `Webzugriff.xlsx` auf dem Arbeitsblatt `Webzugriff`.



ACHSENABSCHNITT() INTERCEPT()

ACHSENABSCHNITT(*Y_Werte*; *X_Werte*)

Syntax

Die Funktion ACHSENABSCHNITT() gibt den Schnittpunkt der Regressionsgeraden zurück. Vergleichen Sie die Abbildung 11.133 auf Seite 521 unter der Funktion STEIGUNG().

Definition

Das heißt, die Funktion berechnet den Punkt, an dem eine Gerade die Y-Achse unter Verwendung vorhandener x-Werte und y-Werte schneidet. Der Schnittpunkt basiert auf einer optimal angepassten Regressionsgeraden, die durch die x-Werte und y-Werte gezeichnet wird.

Y_Werte (erforderlich) ist die Gruppe der abhängigen Messwerte oder Daten.

Argumente

X_Werte (erforderlich) ist die Gruppe der unabhängigen Messwerte oder Daten.

Als Argumente sollten Zahlen, Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten. Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Hinweis

Umfassen *Y_Werte* und *X_Werte* unterschiedlich viele oder keine Datenpunkte, gibt ACHSENABSCHNITT() den Fehlerwert #NV zurück.

Die Regressionsanalyse ist das am häufigsten angewandte multivariate Analyseverfahren. Bei der Einfachregression wird die Beziehung zwischen einer abhängigen Variablen und einer unabhängigen Variablen untersucht. Das heißt, es geht letztendlich darum, den gefundenen Zusammenhang durch eine lineare Funktion zu erfassen. Hierbei wird die Gerade gesucht, bei der die Summe der quadrierten Abweichungen minimiert ist.

Hintergrund

Die Regressionsgerade bzw. -ebene ist demnach nur eine Schätzgröße und gibt Auskunft über die statistische Beziehung zwischen den Variablen und dem Proportionalitätsfaktor. Anders ausgedrückt: Die Regressionsgerade gibt die Richtung der Beziehung zwischen der abhängigen und der/den unabhängigen Variablen an sowie die Höhe der proportionalen Veränderung in der abhängigen Variablen, wenn die unabhängige um eine Einheit erhöht oder vermindert wird.

Die Funktion `ACHSENABSCHNITT()` berechnet den y-Achsenabschnitt der Regressionsgeraden beim x-Wert 0, also den Wert b der Geradengleichung:

$$y = mx + b$$

Für den Punkt b , an dem die Regressionsgerade die Y-Achse schneidet, ergibt sich daraus folgende Gleichung:

$$b = y - mx$$

Hierbei wird die Steigung m wie folgt berechnet:

$$m = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2}$$

Für diese Gleichung sind x und y die Stichprobenmittelwerte `MITTELWERT(X_Werte)` und `MITTELWERT(Y_Werte)`.

Hinweis Detaillierte Informationen zur Regressionsanalyse finden Sie unter `RGP()` auf Seite 492.

Praxiseinsatz Die Funktion `ACHSENABSCHNITT()` ist sehr einfach zu handhaben.

Ein Softwarehersteller möchte seine Webseite näher analysieren. Dazu hat die Marketingabteilung die Zugriffe auf die Webseite von Januar 2007 bis Juni 2008 aus dem firmeneigenen System generiert und in einer Excel-Tabelle zusammengefasst. Zusätzlich wurden alle in dieser Zeit eingegangenen Onlinebestellungen zahlenmäßig erfasst und den Webseitenzugriffen gegenüber gestellt.

Ziel soll es sein, eine Grafik zu erstellen, die zeigt, wie sich die Bestellungen in Abhängigkeit von den Webseitenzugriffen entwickelt haben. Hierbei wird auch der Achsenabschnitt errechnet – also der Schnittpunkt der Regressionsgeraden mit der *Bestellungen*-Achse (siehe Abbildung 11.1).

Monat	Webseitenzugriff	Bestellungen
Januar 2007	236	6
Februar 2007	11593	17
März 2007	18491	456
April 2007	11743	544
Mai 2007	11452	349
Juni 2007	26651	854
Juli 2007	16287	427
August 2007	17750	337
September 2007	19985	899
Oktober 2007	17285	1011
November 2007	30369	720
Dezember 2007	19674	1069
Januar 2008	28464	1070
Februar 2008	25000	498
März 2008	24574	1401
April 2008	23141	1076
Mai 2008	17700	1563
Juni 2008	3702	1790
Achsenabschnitt		524,0479792

Abbildung 11.1: Die Berechnung des Achsenabschnitts für die Darstellung der Regressionsgeraden

Die Bestellungen (abhängige Y-Werte), die Webseitenzugriffe (unabhängige X-Werte) sowie der Achsenabschnitt b stellen sich wie in Abbildung 11.2 dar:

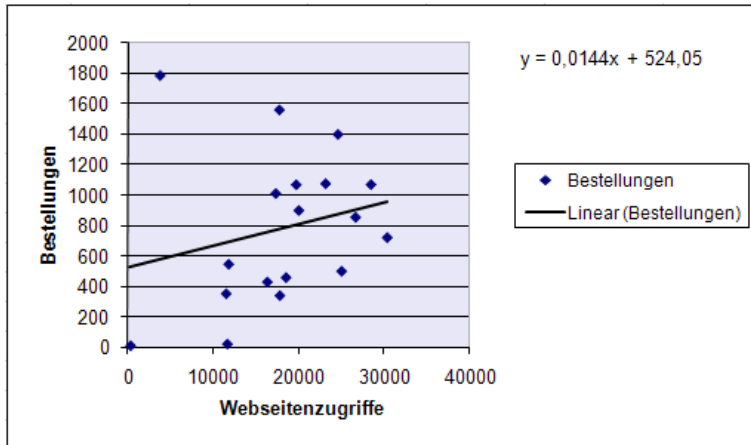


Abbildung 11.2: Die Darstellung der Regressionsgeraden mit dem errechneten Achsenabschnitt von 524,05

Lässt man sich für das Diagramm zunächst die Bestellungen in Abhängigkeit von den Webseitenzugriffen als Punktdiagramm anzeigen und fügt anschließend für die Datenpunkte der Bestellungen eine lineare Trendlinie ein, erhält man automatisch den Achsenabschnitt mit dem zuvor über die Funktion `ACHSENABSCHNITT()` errechneten Wert 524,05.

Die Gleichung für die Trendlinie kann man sich durch Aktivieren des Kontrollkästchens *Gleichung im Diagramm darstellen* anzeigen lassen. Klicken Sie hierzu im Kontextmenü der Trendlinie auf *Trendlinie formatieren* und aktivieren Sie auf der Registerkarte *Optionen* des Dialogfelds *Trendlinie formatieren* den entsprechenden Eintrag.

Tip

`BESTIMMTHEITSMASS()`, `PEARSON()`, `RGP()`, `RKP()`, `SCHÄTZER()`, `STEIGUNG()`, `STFEHLERYX()`, `TREND()`, `VARIATION()`

Siehe auch

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Regression.xlsx* auf dem Arbeitsblatt *Achsenabschnitt*.



ANZAHL() COUNT()

`ANZAHL(Wert1;Wert2;...)`

Mit der Funktion `ANZAHL()` wird berechnet, wie viele Zahlen eine bestimmte Liste von Argumenten enthält. Die Funktion `ANZAHL()` wird verwendet, um zu ermitteln, aus wie vielen Einträgen ein Zahlenfeld besteht, das in einem bestimmten Bereich oder in einer Matrix gespeichert ist.

Wert1 (erforderlich); *Wert2* (optional); ... sind 1 bis 255 Argumente (30 bis Excel 2003). Diese Argumente können zwar Daten unterschiedlichen Typs enthalten oder sich auf Daten unterschiedlichen Typs beziehen, für die Zählung jedoch werden nur Zahlenwerte berücksichtigt.

Syntax

Definition

Argumente

Hintergrund Die Funktion ANZAHL() übernimmt eine einfache, jedoch oftmals sehr arbeitszeitsparende Aufgabe. Vor allem bei großen, mit vielen Zahlenwerten versehenen Tabellen wäre ein Abzählen von Hand extrem zeitaufwendig. In solch einem Fall kann einem diese Funktion zum Errechnen der Anzahl eingegebener Werte in einer Tabelle viel Arbeit abnehmen.

Hinweis Bei einer Zählung werden alle Argumente berücksichtigt, die entweder Zahlen, Null, Wahrheitswerte, Datumsangaben oder Zahlen in Textform sind. Argumente, die mit einem Fehlerwert oder mit einem Text belegt sind, der nicht in eine Zahl umgewandelt werden kann, werden hierbei ignoriert.

Ist ein Argument eine Matrix oder ein Bezug, werden bei einer Zählung nur die in dieser Matrix oder in diesem Bezug abgelegten Zahlen berücksichtigt. Alle zu der Matrix oder dem Bezug gehörenden leeren Zellen, Wahrheitswerte, Text oder Fehlerwerte werden ignoriert. Wenn Sie Wahrheitswerte, Text oder Fehlerwerte mit einbeziehen möchten, verwenden Sie die ANZAHL2()-Funktion.

Praxiseinsatz Als Beispiel ist in Abbildung 11.3 eine Tabelle vorgegeben, in die der Außendienstmitarbeiter des Softwareherstellers die Umsätze von Januar 2007 bis November 2008 hätten erfassen sollen.

	A	B	C
1	Fragestellung: In wieviel Monaten wurde ein Umsatz erfasst?		
2	Monat		Umsatz
3	Januar 07		17.500,00 €
4	Februar 07		17.867,00 €
5	März 07		10.966,00 €
6	April 07		
7	Mai 07		12.838,00 €
8	Juni 07		14.888,00 €
9	Juli 07		14.245,00 €
10	August 07		16.292,00 €
11	September 07		11.689,00 €
12	Oktober 07		
13	November 07		18.560,00 €
14	Dezember 07		15.697,00 €
15	Januar 08		16.022,00 €
16	Februar 08		geschlossen
17	März 08		12.556,00 €
18	April 08		18.681,00 €
19	Mai 08		14.643,00 €
20	Juni 08		
21	Juli 08		18.172,00 €
22	August 08		19.710,00 €
23	September 08		17.182,00 €
24	Oktober 08		11.045,00 €
25	November 08		18.494,00 €
27	=ANZAHL(C3:C25)		
28	Ergebnis:		19

Abbildung 11.3: In dieser Tabelle wurden innerhalb des Zellbereichs C3:C25 Umsätze innerhalb eines bestimmten Zeitraums erfasst

Der Vorgesetzte des Außendienstmitarbeiters möchte nun wissen, wie viele Zahlenargumente sich in einem bestimmten Bereich der Tabelle befinden und bedient sich deshalb der Funktion ANZAHL(). Durch die Eingabe des zu untersuchenden Bereichs C3:C25, ergibt sich der Wert 19 als Ergebnis aus der Formel ANZAHL(C3:C25).

Wenn die Daten markiert sind, können Sie die Anzahl der Zahlenargumente auch in der Statusleiste ablesen. Dazu klicken Sie mit der rechten Maustaste die Statusleiste an und wählen Sie im Kontextmenü den Eintrag *Numerische Zahl* wie in Abbildung 11.4 dargestellt.

Tipp



Abbildung 11.4: Ausschnitt aus dem Kontextmenü in der Statusleiste von Excel

Die Information über die aktuelle Markierung wird dann in der Statusleiste angezeigt (siehe Abbildung 11.5).

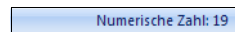


Abbildung 11.5: Die Anzahl der errechneten Zahlenargumente

ANZAHL2(), DBANZAHL(), DBANZAHL2(), MITTELWERT(), SUMME()

Siehe auch

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *Anzahl*.



ANZAHL2() COUNTA()

ANZAHL2(*Wert1*; *Wert2*; ...)

Mit der ANZAHL2() wird berechnet, wie viele Werte eine Liste von Argumenten enthält. Verwenden Sie ANZAHL2(), wenn Sie wissen möchten, wie viele zu einem Bereich oder einer Matrix gehörende Zellen Daten enthalten.

Wert1 (erforderlich); *Wert2* (optional); ... sind 1 bis 255 Argumente (30 bis Excel 2003), die die Werte angeben, die Sie in die Zählung einbeziehen möchten. Ein Wert ist in diesem Fall jede beliebige Art von Information, auch leerer Text (""), nicht jedoch leere Zellen. Ist ein Argument eine Matrix oder ein Bezug, werden alle zu der Matrix oder dem Bezug gehörenden leeren Zellen ignoriert. Wenn Sie Wahrheitswerte, Text oder Fehlerwerte von der Zählung ausnehmen möchten, verwenden Sie die ANZAHL()-Funktion.

Entsprechend der Funktion ANZAHL() können Sie auch mit dieser Funktion sehr viel Zeit für die Berechnung von Werten in einer Liste mit Argumenten sparen. Der einzige Unterschied besteht lediglich darin, dass ANZAHL2() neben Zahlen auch Text sowie Wahrheitswerte und Fehlerwerte berücksichtigt und Sie somit beliebige Einträge in einem Zellbereich zählen können.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz Entsprechend Abbildung 11.3 auf Seite 350 und der gleichen Zellenbereichsangabe erhalten Sie bei Einsatz der Funktion ANZAHL2() hier das Ergebnis »20«, da das Wort »geschlossen« als Eintrag im Zellbereich mitgerechnet wird.

Tipp Auch hier können Sie sich, wie bei der Funktion ANZAHL(), die errechnete Anzahl der Einträge im Zellbereich anzeigen lassen. Markieren Sie dazu die entsprechenden Daten und wählen Sie durch Klicken mit der rechten Maustaste in der Statusleiste den Eintrag *Anzahl* in Excel 2007-2013 (siehe Abbildung 11.4).

Siehe auch ANZAHL(), DBANZAHL(), DBANZAHL2(), MITTELWERT(), PRODUKT(), SUMME()



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *Anzahl2*.

ANZAHLLEEREZELLEN() COUNTBLANK()

Syntax ANZAHLLEEREZELLEN(*Bereich*)

Definition Die Funktion zählt die leeren Zellen in einem Zellbereich.

Argumente *Bereich* (erforderlich) ist der Bereich, von dem Sie wissen möchten, wie viele Zellen darin leer sind.

Hintergrund Wie bei ANZAHL() und ANZAHL2() macht auch diese Formel vor allem bei großen Tabellen Sinn und bringt auch erst dann die gewünschte Zeitersparnis.

Hinweis Es werden auch Zellen gezählt, in denen Formeln stehen, die leere Zeichenfolgen ("") zurückgeben. Zellen, die Nullwerte enthalten, werden nicht gezählt.

Praxiseinsatz Wenden Sie die Formel ANZAHLLEEREZELLEN() auf das Beispiel aus ANZAHL() an, ergibt sich als Ergebnis für den Zellbereich C3:C25 die Zahl »3«. Wie Sie in Abbildung 11.3 auf Seite 350 vergleichen können, enthält die Tabelle drei Zellen, die nicht mit Zahlen oder Text gefüllt sind.

Siehe auch ZÄHLENWENN()



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *Anzahlleerezellen*.

BESTIMMTHEITSMASS() RSQ()

Syntax BESTIMMTHEITSMASS(*Y_Werte*; *X_Werte*)

Definition Die Funktion BESTIMMTHEITSMASS() gibt das Quadrat des Pearsonschen Korrelationskoeffizienten zurück, entsprechend den in *Y_Werte* und *X_Werte* abgelegten Datenpunkten. Ein r^2 -Wert kann als der Anteil der Varianz von Y, der durch die Varianz von X erklärt wird, interpretiert werden.

Hinweis Detaillierte Informationen zum Pearsonschen Korrelationskoeffizienten und zur Funktion PEARSON() können Sie auf Seite 472 dieses Buchs nachlesen.

Y_Werte (erforderlich) ist eine Matrix oder ein Bereich von Datenpunkten.

X_Werte (erforderlich) ist eine Matrix oder ein Bereich von Datenpunkten.

Als Argumente sollten entweder Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten. Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Enthalten **Y_Werte** und **X_Werte** keine oder unterschiedlich viele Datenpunkte, gibt BESTIMMTHEITSMASS() den Fehlerwert #NV zurück.

Soll der Einfluss der unabhängigen Variablen auf die abhängige Variable bestimmt werden, setzt man die erklärte Streuung zur Gesamtstreuung ins Verhältnis. Dies wird als Bestimmtheitsmaß bezeichnet und ergibt sich rein rechnerisch durch die Quadrierung des einfachen Korrelationskoeffizienten.

Als Ergebnis der Berechnung des Bestimmtheitsmaßes erhält man also die Einfluss-Stärke in Prozent.

Der Korrelationskoeffizient ist eine Maßzahl für den linearen Zusammenhang zwischen zwei quantitativen Merkmalen. Er liegt zwischen -1 und $+1$ und ist positiv, wenn den hohen (bzw. niedrigen) Werten eines Merkmals jeweils hohe (bzw. niedrige) Werte des anderen Merkmals entsprechen. Er ist negativ im umgekehrten Falle. Der Wert liegt umso näher bei ± 1 , je stärker die Beziehung ist. Ein Wert bei 0 lässt auf das Fehlen einer linearen Beziehung schließen.

Das Bestimmtheitsmaß liegt zwischen 0 und +1. Liegt das Bestimmtheitsmaß bei 1, bedeutet dies, dass die abhängige Variable sich alleine durch die unabhängige Variable erklären lässt. Ist das Bestimmtheitsmaß beispielsweise $r^2 = 0,0354$, wird hiermit ausgesagt, dass sich nur 3,54 % der abhängigen Variablen von der unabhängigen Variablen erklären lassen. Der Rest entfällt auf andere Einflussgrößen. Die Differenz zwischen den beobachteten und geschätzten Werten der Funktion nennt man Störgrößen oder auch Residuen.

Bei völliger Linearität ist das Bestimmtheitsmaß = 1. In diesem Fall liegen alle Punkte auf der Regressionsgrade. Die nicht erklärte Varianz beträgt dann 0. Je niedriger das Bestimmtheitsmaß, desto geringer ist die Aussagekraft der Regressionsgraden.

Die Gleichung für den Pearsonschen Korrelationskoeffizienten r lautet:

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}}$$

Dabei sind x und y die Stichprobenmittelwerte MITTELWERT(**X_Werte**) und MITTELWERT(**Y_Werte**).

BESTIMMTHEITSMASS() gibt r^2 zurück, also das Quadrat des Korrelationskoeffizienten.

Der Softwarehersteller ist immer noch mit der Auswertung seiner Webseite beschäftigt. Die Marketingabteilung stellt sich nun die Frage, inwieweit die Onlinebestellungen vom Zugriff auf die Webseite abhängen.

Um also die Einflussstärke in Prozent herauszubekommen, wollen die Mitarbeiter das Bestimmtheitsmaß dieser beiden voneinander abhängigen Variablen berechnen.

Argumente

Hinweis

Hintergrund

Hinweis

Praxiseinsatz

Monat	Webseitenzugriff	Bestellungen
Januar 2007	236	6
Februar 2007	11593	17
März 2007	18491	456
April 2007	11743	544
Mai 2007	11452	349
Juni 2007	26651	854
Juli 2007	16287	427
August 2007	17750	337
September 2007	19985	899
Oktober 2007	17285	1011
November 2007	30369	720
Dezember 2007	19674	1069
Januar 2008	28464	1070
Februar 2008	25000	498
März 2008	24574	1401
April 2008	23141	1076
Mai 2008	17700	1563
Juni 2008	3702	1790
Bestimmtheitsmaß	0,053481332	

Abbildung 11.6: Die Berechnung des Bestimmtheitsmaßes, das die Abhängigkeit der Onlinebestellungen vom Webseitenzugriff in Prozent ausdrückt

Wie Sie der Abbildung 11.6 entnehmen können, wurden die Onlinebestellungen (abhängige Variable) den Webseitenzugriffen (unabhängige Variable) gegenübergestellt.

Anschließend wurde mit der Funktion BESTIMMTHEITSMASS() eine prozentuale Abhängigkeit der Onlinebestellungen zu den Webseitenzugriffen von 5,35 % errechnet.

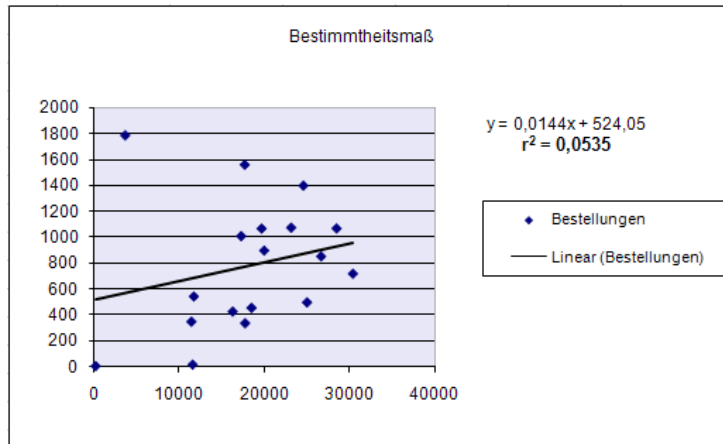


Abbildung 11.7: Die grafische Darstellung der Abhängigkeit zwischen Onlinebestellungen und Webseitenzugriffen

Auch die Darstellung der Onlinebestellungen in Abhängigkeit von den Webseitenzugriffen in Diagrammform liefert das gleiche Ergebnis für r^2 .

Das Bestimmtheitsmaß r^2 kann man sich durch Aktivieren des Kontrollkästchens *Bestimmtheitsmaß im Diagramm darstellen* anzeigen lassen. Klicken Sie hierzu im Kontextmenü der Trendlinie auf *Trendlinie formatieren* und aktivieren Sie auf der Registerkarte *Optionen* des Dialogfelds *Trendlinie formatieren* den entsprechenden Eintrag.

Tip

Die Berechnung des Bestimmtheitsmaßes und das Ergebnis von $r^2 = 0,0535$ sagt aus, dass lediglich 5,35 % der eingegangenen Onlinebestellungen auf die Webseitenzugriffe zurückzuführen sind.

ACHSENABSCHNITT(), KORREL(), KOVAR(), PEARSON(), RGP(), RKP(), STEIGUNG(), STFEHLERYX(), TREND()

Siehe auch

Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Regression.xlsx* auf dem Arbeitsblatt *Bestimmtheitsmaß*.



BETA.INV() / BETAINV()



BETA.INV(), BETAINV()

BETA.INV(*Wahrscheinlichkeit*;Alpha;Beta;A;B)

Syntax

Die Funktion BETA.INV() gibt das Quantil der angegebenen Beta-Verteilung zurück. Dies bedeutet, wenn *Wahrscheinlichkeit* = BETA.VERT(x ;...) ist, dann ist *BETA.INV(Wahrscheinlichkeit*;...) = x . BETA.INV() ist daher die Umkehrfunktion von BETA.VERT().

Definition

Die Beta-Verteilung kann für eine Projektplanung verwendet werden, um ausgehend von einem erwarteten Endtermin und der Streuung den wahrscheinlichen Endtermin zu modellieren.

Wahrscheinlichkeit (erforderlich) ist die zur Beta-Verteilung gehörende Wahrscheinlichkeit.

Argumente

Alpha (erforderlich) ist ein Parameter der Verteilung.

Beta (erforderlich) ist ein Parameter der Verteilung.

A (optional) ist die untere Grenze des Intervalls für X .

B (optional) ist die obere Grenze des Intervalls für X .

Ist eines der Argumente nicht numerisch, gibt die Funktion BETA.INV() den Fehlerwert #WERT! zurück.

Hinweis

Ist *Alpha* kleiner oder gleich 0 bzw. *Beta* kleiner oder gleich 0, gibt BETA.INV() den Fehlerwert #ZAHL! zurück.

Ist *Wahrscheinlichkeit* kleiner oder gleich 0 bzw. *Wahrscheinlichkeit* größer 1, gibt BETA.INV() den Fehlerwert #ZAHL! zurück.

Wird für *A* und *B* kein Wert angegeben, verwendet die Funktion BETA.INV() die Standardverteilung, d.h. $A = 0$ und $B = 1$.

Bei einem Wert für *Wahrscheinlichkeit* sucht BETA.INV() den Wert X so, dass *BETA.VERT(x;Alpha;Beta;A;B)* = *Wahrscheinlichkeit* gilt. Daher hängt die Genauigkeit von BETA.INV() von der Genauigkeit von BETA.VERT() ab. Die Funktion BETA.INV() geht bei der Suche iterativ vor. Hat die Suche nach 100 Iterationsschritten noch nicht konvergiert, gibt die Funktion den Fehlerwert #NV zurück.

Hintergrund Die Beta-Verteilung ist eine kontinuierliche Wahrscheinlichkeitsverteilung, also eine Verteilung, die aussagt, mit welcher Wahrscheinlichkeit eine Zufallsvariable x einen bestimmten Wert annimmt, über dem Intervall $[0,1]$.

Sie ist definiert durch die Wahrscheinlichkeitsdichte

$$f(x) = \frac{1}{B(p; q)} x^{p-1} (1-x)^{q-1}$$

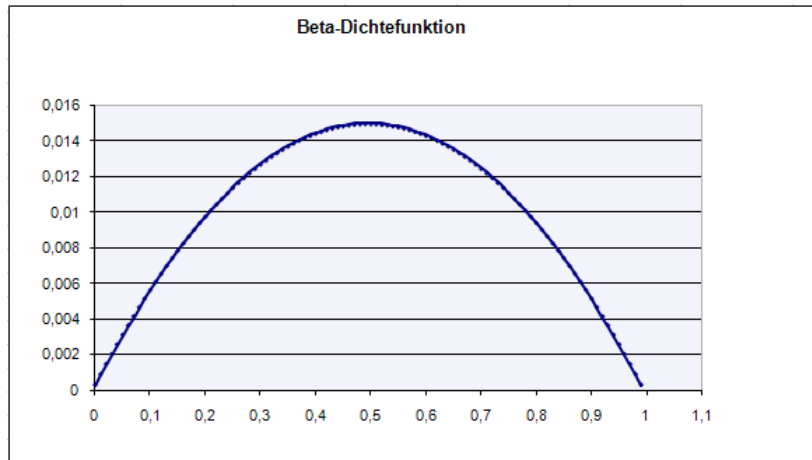


Abbildung 11.8: Die grafische Darstellung der Beta-Dichtefunktion im Intervall $[0,1]$ mit p und $q = 1$

Außerhalb des Intervalls $[0,1]$ wird sie durch $f(x)=0$ fortgesetzt. Sie besitzt die Parameter p und q . Um die Normierbarkeit zu garantieren, wird p, q größer 0 gefordert.

Der Vorfaktor $1/B(p; q)$ dient der korrekten Normierung, also der Normalisierung auf einen Wertebereich zwischen 0 und 1.

Der Ausdruck

$$B(p; q) = \frac{\Gamma(p)\Gamma(q)}{\Gamma(p+q)} = \int_0^1 u^{p-1} (1-u)^{q-1} du$$

steht für die Betafunktion, nach der die Verteilung auch benannt ist. $\Gamma(p)$ steht für die Gammafunktion.

Die Betafunktion (auch Eulersche Beta-Funktion) ist eine mathematische Funktion zweier positiver reeller oder zweier komplexer Zahlen x und y mit positivem Realteil, die durch folgende Formel definiert ist:

$$\beta(x, y) = \int_0^1 t^{x-1} (1-t)^{y-1} dt$$

Sie wird häufig auch mit einem großen Beta bezeichnet: $B(x, y)$.

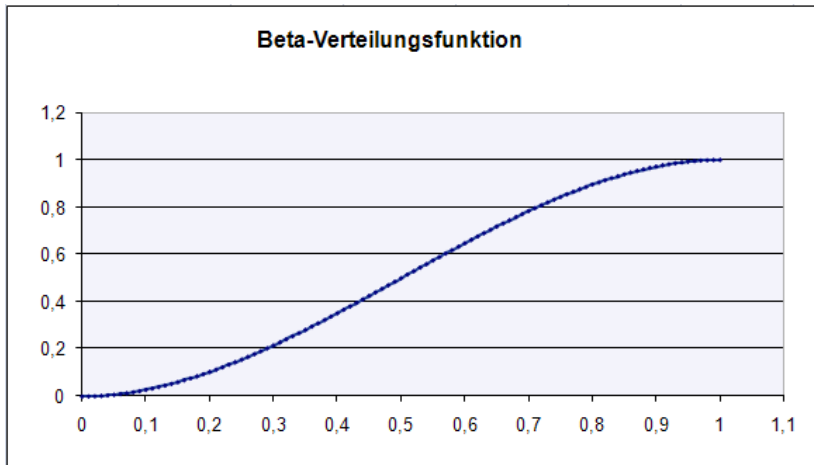


Abbildung 11.9: Die grafische Darstellung der Beta-Verteilungsfunktion im Intervall $[0,1]$ mit p und $q = 1$

Erwartungswert und Varianz der Betaverteilung sind

$$E(X) = \frac{p}{p+q} \text{ und } V(X) = \frac{pq}{(p+q+1)(p+q)^2}$$

Die Funktion BETA.INV() liefert Quantile für die soeben beschriebene Betaverteilung. Gehen Sie folgendermaßen zur Berechnung von BETA.INV() vor:

Praxiseinsatz

Gegeben sind die Werte:

- ▶ 0,685470581 = die zur Betaverteilung gehörende Wahrscheinlichkeit
- ▶ 8 = Parameter der Verteilung
- ▶ 10 = Parameter der Verteilung
- ▶ 1 = Untere Grenze
- ▶ 3 = Obere Grenze

Die Berechnung von BETA.INV() sehen Sie in Abbildung 11.10.

	B	C	D	E	F
7	Berechnung von BETA.INV()				
8	Bedeutung			Parameter	
9	Die zur Betaverteilung gehörende Wahrscheinlichkeit			0,685470581	
10	Parameter der Verteilung			8	
11	Parameter der Verteilung			10	
12	Untere Grenze			1	
13	Obere Grenze			3	
14	BETA.INV()			2	
15				=BETA.INV(E9;E10;E11;E12;E13)	

Abbildung 11.10: Berechnung von BETA.INV()

Unter Angabe der in Abbildung 11.10 dargestellten Parameter gibt die Funktion `BETA.INV()` das Quantil 2 der angegebenen Betaverteilung zurück.

Siehe auch `BETA.VERT()`, `BETAVERT()`



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Beta.inv*.

Ebenso finden Sie hier eine Tabelle, bei der Sie die Möglichkeit haben, selbst die Parameter *Alpha* und *Beta* zu bestimmen. Die Veränderung der Dichte- und Verteilungsfunktion wird durch die Eingabe der Parameter aufgezeigt.

BETA.VERT() / BETAVERT()



BETA.DIST(), BETADIST()

Syntax `BETA.VERT(X;Alpha;Beta;Kumuliert;A;B)`

Definition Die Funktion `BETA.VERT()` gibt die Werte der kumulierten Betaverteilungsfunktion zurück. Die Betaverteilung wird in der Regel verwendet, um die Streuung bei mehreren Stichproben zu bestimmten Vorgängen zu untersuchen. Beispielsweise kann prozentual ermittelt werden, wie viel Zeit am Tag Personen in Ihrer Freizeit vor dem Computer verbringen.

Argumente `X` (erforderlich) ist der Wert, an dem die Funktion im Intervall zwischen `A` und `B` ausgewertet werden soll.

Alpha (erforderlich) ist ein Parameter der Verteilung.

Beta (erforderlich) ist ein Parameter der Verteilung.

`Kumuliert` (erforderlich) ist der Wahrheitswert, der den Typ der Funktion bestimmt.

`A` (optional) ist die untere Grenze des Intervalls für `X`.

`B` (optional) ist die obere Grenze des Intervalls für `X`.

Hinweis Ist eines der Argumente nicht numerisch, gibt die Funktion `BETA.VERT()` den Fehlerwert `#WERT!` zurück.

Ist *Alpha* kleiner oder gleich 0 bzw. *Beta* kleiner oder gleich 0, gibt `BETA.VERT()` den Fehlerwert `#ZAH!` zurück.

Ist `X` kleiner `A`, `X` größer `B` oder `A` gleich `B`, gibt `BETA.VERT()` den Fehlerwert `#ZAH!` zurück.

Wird für `A` und `B` kein Wert angegeben, verwendet die Funktion `BETA.VERT()` die Standardverteilung, das heißt, `A = 0` und `B = 1`.

Hintergrund Durch die Berechnung von `BETA.VERT()` erhalten Sie den `Y`-Wert auf der Beta-Verteilungskurve. Das heißt, berechnen Sie `BETA.VERT()` auf Basis eines bestimmten `X`-Werts der Verteilung sowie den zwei Parametern der Verteilung *Alpha* und *Beta*, so ermitteln Sie den `Y`-Wert auf der Kurve (siehe Abbildung 11.11).

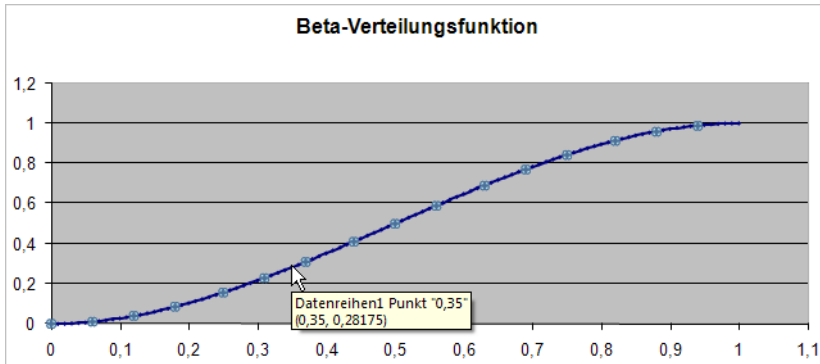


Abbildung 11.11: Mithilfe der Funktion *BETA.VERT()* ermitteln Sie den Y-Wert der Verteilungsfunktion

Mehr Informationen zur Beta-Verteilung finden Sie unter *BETA.INV()* / *BETAINV()* auf Seite 355.

Hinweis

Gehen Sie folgendermaßen zur Berechnung von *BETA.VERT()* vor.

Praxiseinsatz

Gegeben sind die Werte:

- ▶ 2 = Wert, für den die Funktion ausgewertet werden soll
- ▶ 8 = Parameter der Verteilung
- ▶ 10 = Parameter der Verteilung
- ▶ WAHR = kumuliert, spricht der Wahrheitswert der Funktion
- ▶ 1 = Untere Grenze
- ▶ 3 = Obere Grenze

Die Berechnung von *BETA.VERT()* sehen Sie in Abbildung 11.12.

E15		fx		=BETA.VERT(E9;E10;E11;WAHR;E13;E14)	
	B	C	D	E	F
7	Berechnung von BETA.VERT()				
8	Bedeutung			Parameter	
9	Wert, für den die Funktion ausgewertet werden soll			2	
10	Parameter der Verteilung			8	
11	Parameter der Verteilung			10	
12	Wahrheitswert (kumuliert)			WAHR	
13	Untere Grenze			1	
14	Obere Grenze			3	
15	BETA.VERT()			0,685470581	

Abbildung 11.12: Berechnung von *BETA.VERT()*

Unter Angabe der in Abbildung 11.12 dargestellten Parameter gibt die Funktion *BETA.VERT()* die Verteilungsfunktion (integrierte Dichtefunktion) einer Betaverteilung zurück.

Siehe auch BETA.INV(), BETAINV()



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Beta.vert*.

Ebenso finden Sie hier eine Tabelle, bei der Sie die Möglichkeit haben, selbst die Parameter *Alpha* und *Beta* zu bestimmen. Die Veränderung der Dichte- und Verteilungsfunktion wird durch die Eingabe der Parameter aufgezeigt.

BINOM.INV() / KRITBINOM()

BINOM.INV() / CRITBINOM()

Syntax BINOM.INV(*Versuche*; *ErfolgsWahrscheinlichkeit*; *Alpha*)

Definition Die Funktion BINOM.INV() (vor 2010 unter dem Namen KRITBINOM() bekannt) gibt den kleinsten Wert zurück, für den die kumulierten Wahrscheinlichkeiten der Binomialverteilung größer oder gleich einer Grenzwahrscheinlichkeit sind. Mit dieser Funktion können Sie Aufgaben erledigen, die im Bereich Qualitätssicherung anfallen. Mithilfe der BINOM.INV()-Funktion lässt sich beispielsweise ermitteln, wie viele defekte Teile höchstens an einem Fließband Ausschuss sein dürfen, ohne dass das gesamte Fertigungslos zurückgewiesen werden muss.

Argumente *Versuche* (erforderlich) ist die Anzahl der Bernoulli-Experimente.

ErfolgsWahrscheinlichkeit (erforderlich) ist die Wahrscheinlichkeit eines Erfolgs für jeden Versuch.

Alpha (erforderlich) ist die Grenzwahrscheinlichkeit.

Hinweis Ist eines der Argumente nicht numerisch, gibt die Funktion BINOM.INV() den Fehlerwert #WERT! zurück.

Ist *Versuche* keine ganze Zahl, werden die Nachkommastellen abgeschnitten.

Ist *Versuche* kleiner 0, gibt BINOM.INV() den Fehlerwert #ZAHL! zurück.

Ist *ErfolgsWahrscheinlichkeit* kleiner 0 oder *ErfolgsWahrscheinlichkeit* größer 1, gibt BINOM.INV() den Fehlerwert #ZAHL! zurück.

Ist *Alpha* kleiner 0 oder *Alpha* größer 1, gibt die Funktion BINOM.INV() den Fehlerwert #ZAHL! zurück.

Hintergrund Die Funktion BINOM.INV() liefert den kleinsten Wert für eine binomialverteilte Zufallsvariable, bei dem die angegebene Irrtumswahrscheinlichkeit oder Grenzwahrscheinlichkeit *Alpha* nicht überschritten wird. Die Grenzwahrscheinlichkeit entspricht mathematisch gesehen dem Alpha-Risiko.

Das Alpha-Risiko ist als Risiko definiert, mit dem der statistische Hypothesentest auf einen Sachverhalt hindeutet, der in Wahrheit gar nicht vorhanden ist, und spielt zum Beispiel in der Produktionstechnik eine Rolle, wenn man sich für eine neue Maschine entscheidet, die in Wahrheit nicht besser ist.

Im Grunde genommen ist das Alpha-Risiko, was meistens den Wert 10 %, 5 % oder kleiner annimmt, die Wahrscheinlichkeit dafür, dass das Datenmaterial eine entsprechend »extreme« oder »noch extremere« Ausprägung aus reinem Zufall annimmt.

Mit BINOM.INV() wird daher berechnet, wie oft ein bestimmtes Ereignis mit der Wahrscheinlichkeit p beim Durchführen einer Stichprobe mit n Wiederholungen höchstens eintreten darf, bevor seine kumulierten Wahrscheinlichkeiten einen Wert größer oder gleich der Irrtumswahrscheinlichkeit *Alpha* annehmen.

Diese Funktion ist nur bei Binomialverteilungen anwendbar. Es muss sich also um unabhängige Ziehungen handeln, bei denen nur zwei Ergebnisse möglich sind – entweder ein Ereignis tritt ein oder nicht.

Die Funktion BINOM.INV() ist die Umkehrfunktion von BINOM.VERT().

Mehr Informationen zur Binomialverteilung und der Funktion BINOM.VERT() / BINOMVERT() finden Sie auf Seite 362.

Hinweis

Um Ihnen die Funktion BINOM.INV() zu verdeutlichen, nehmen wir das Beispiel von der Funktion BINOM.VERT() zur Hand. Hier war die Situation gegeben, dass Sie in einer fremden Stadt 100 Personen (n Versuche) nach dem Weg zu einem bestimmten Ort gefragt haben. Auf diese Frage gibt es nur zwei Antwortmöglichkeiten: »Ja« oder »Nein«. Das heißt, die Wahrscheinlichkeit, dass Sie ein »Ja« zu hören bekommen, beträgt 50 %. Daraus ergibt sich ein p von 0,5.

Praxiseinsatz

Mithilfe der Funktion BINOM.VERT() wollten Sie nun wissen, wie hoch die Wahrscheinlichkeit ist, dass von 100 befragten Personen 66 Leute – also ca. $\frac{2}{3}$ der Befragten – mit »Ja« antworten und Sie somit einen Treffer landen.

Jetzt wollen Sie mit BINOM.INV() berechnen, wie oft die Antwort »Ja« mit der Wahrscheinlichkeit $p = 0,5$ beim Durchführen einer Stichprobe mit $n = 100$ Wiederholungen höchstens eintreten darf, bevor die kumulierte Wahrscheinlichkeit einen Wert größer oder gleich der Irrtumswahrscheinlichkeit *Alpha* annimmt.

Das Ergebnis zeigt die nachfolgende Abbildung 11.13.

	B	C	D	E	F	G	H	I
10	Stichprobengröße n		100	100	100	100	100	100
11	Erfolgswahrscheinlichkeit p		50%	50%	50%	50%	50%	50%
12	Grenzwahrscheinlichkeit (~Alpha Risiko)		0,1%	0,2%	0,4%	0,6%	0,8%	1,0%
13	BINOM.INV		35	36	37	37	38	38
14			=BINOM.INV(D10;D11;D12)					

Abbildung 11.13: Die Berechnung durch die Funktion *BINOM.INV()*

Bezogen auf eine Grenzwahrscheinlichkeit von 0,1 % dürfen höchstens 35 »Ja«-Antworten gegeben werden, bevor die kumulierte Wahrscheinlichkeit einen Wert größer oder gleich der Irrtumswahrscheinlichkeit *Alpha* annimmt.

Aussagen

Mit der Funktion BINOM.INV() können Sie auf diese Weise zu einer gegebenen Grenzwahrscheinlichkeit *Alpha*, die maximale Anzahl Merkmalsträger innerhalb der Stichprobe ermitteln.

Siehe auch BINOM.VERT(), BINOM.VERT.BEREICH(), FAKULTÄT(), HYPGEOM.VERT(), KOMBINATIONEN(), NEGBINOM.VERT(), VARIATIONEN(), VARIATIONEN2(), WAHRSCHEBEREICH()



Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Binom.inv*.

BINOM.VERT() / BINOMVERT()

BINOM.DIST(), BINOMDIST()

Syntax BINOM.VERT(*AnzahlErfolge*; *Versuche*; *ErfolgsWahrscheinlichkeit*; *kumuliert*)

Definition Die Funktion BINOM.VERT() gibt Wahrscheinlichkeiten einer binomialverteilten Zufallsvariablen zurück. Sie können die Funktion BINOM.VERT() bei Problemen mit einer festgelegten Anzahl von Tests oder Versuchen verwenden, wenn das Ergebnis jedes einzelnen Versuchs entweder *Erfolg* oder *Misserfolg* ist, die einzelnen Versuche voneinander unabhängig sind und die Wahrscheinlichkeit des Erfolgs für alle Versuche konstant ist. Mit der Funktion BINOM.VERT() lässt sich beispielsweise die Wahrscheinlichkeit ermitteln, mit der 50 von 100 Restaurantgästen für ein Rauchverbot sind.

Argumente *AnzahlErfolge* (erforderlich) ist die Anzahl der Erfolge in einer Versuchsreihe.
Versuche (erforderlich) ist die Anzahl der voneinander unabhängigen Versuche.
ErfolgsWahrscheinlichkeit (erforderlich) ist die Wahrscheinlichkeit eines Erfolgs für jeden Versuch.

kumuliert (erforderlich) ist ein Wahrheitswert, der den Typ der Funktion bestimmt. Ist *kumuliert* WAHR, gibt die Funktion BINOM.VERT() die Verteilungsfunktion zurück, also die Wahrscheinlichkeit dafür, dass es höchstens *AnzahlErfolge* an Erfolgen gibt. Ist *kumuliert* FALSCH, gibt BINOM.VERT() die Wahrscheinlichkeitsfunktion zurück, also die Wahrscheinlichkeit, dass es genau *AnzahlErfolge* an Erfolgen gibt.

Hinweis *AnzahlErfolge* und *Versuche* werden auf ganze Zahlen abgerundet.

Ist *AnzahlErfolge*, *Versuche* oder *ErfolgsWahrscheinlichkeit* nicht numerisch, gibt die Funktion BINOM.VERT() den Fehlerwert #WERT! zurück.

Ist *AnzahlErfolge* kleiner 0 oder *AnzahlErfolge* größer *Versuche*, gibt die Funktion BINOM.VERT() den Fehlerwert #ZAHL! zurück.

Ist *ErfolgsWahrscheinlichkeit* kleiner 0 oder *ErfolgsWahrscheinlichkeit* größer 1, gibt die Funktion BINOM.VERT() den Fehlerwert #ZAHL! zurück.

Hintergrund Der Begriff »Wahrscheinlichkeit« kann ganz allgemein als Maß für das Eintreffen eines bestimmten Ergebnisses bei einem Zufallsexperiment aus einer Menge von Elementen definiert werden, wobei keines der möglichen Ereignisse bevorzugt sein soll.

Wie hoch ist z.B. die Wahrscheinlichkeit, dass es bei der Produktion von 2.000 Tabletten zu 30 fehlerhaften Verpackungen kommt, wenn Sie bei der durchschnittlichen Produktion von 2 % fehlerhaften Verpackungen ausgehen?

Ebenso könnte man in die Betrachtung solche Sachverhalte einbeziehen, bei denen jedes Element nur zwei mögliche Zustände haben kann, wie z.B. die Eigenschaft *Kopf* oder *Zahl*, die nur den Zustand *WAHR* oder *FALSCH* annehmen kann. In Zahlen ausgedrückt kann jedes dieser Elemente entweder 1 oder 0 sein.

Zufallsexperimente, bei denen eine Zufallsvariable in zwei Kategorien eingeteilt werden kann, sind im Allgemeinen unter der Bezeichnung »Bernoulli-Experiment« bekannt, nach dem aus der Schweiz stammenden Mathematiker Jakob Bernoulli (1654 bis 1705).

Die Zufallsvariable (Bernoulli-Variable) X nimmt bei einer Wahrscheinlichkeit p den Wert $X=1$ (Erfolg) und für die Wahrscheinlichkeit $q=1-p$ den Wert $X=0$ (Misserfolg) an. Die Wahrscheinlichkeit p wird auch Erfolgsparameter genannt.

In der Praxis werden Sie häufig das Problem haben, dass viele Elemente in zwei Kategorien eingeteilt werden. Es liegt also eine Folge von n Bernoulli-Experimenten vor. Die Wahrscheinlichkeit, dass die Zufallsvariable $X=1$ in genau k Fällen vorkommt, wird nach der folgenden Formel von Bernoulli berechnet:

$$P(X = k) = \binom{n}{k} p^k q^{n-k}$$

Die Verteilung

$$F_B : k \leq P(X \leq k) = F_B(k; n, p) = \sum_{i=0}^k f_b(i; n, p)$$

der Zufallsvariablen heißt *Binomialverteilung* mit den Binomialkoeffizienten $\binom{n}{k}$ – sprich n über k .

Die Binomialverteilung ist eine der wichtigsten Wahrscheinlichkeitsverteilungen. Sie ist ein Spezialfall der Multinomialverteilung und beschreibt Ergebnisse von Bernoulli-Prozessen, die wiederum als Abfolge mehrerer, unter gleich bleibenden Bedingungen durchgeführter Bernoulli-Versuche definiert sind – wie z.B. der Münzwurf.

Sind *Erfolg* und *Misserfolg* beim Experiment Münzwurf gleich wahrscheinlich, gibt es bei einer Anzahl von zwölf Würfeln insgesamt 2^{12} mögliche Ergebnisse. Zum Beispiel bedeutet sieben Mal das Ergebnis *Zahl*, dass sich unter den zwölf Ergebnissen genau sieben Mal das Einzelergebnis *Zahl* befindet. Dies ist auf so viele Arten möglich, wie man sieben Objekte aus zwölf Objekten auswählen kann, also $\binom{12}{7}$ Mal. Damit ist die Wahrscheinlichkeit für sieben Mal *Zahl* gleich

$$\frac{\binom{12}{7}}{2^{12}}.$$

Für die Berechnung der Wahrscheinlichkeit einer binomialverteilten Zufallsvariable steht Ihnen in Excel die Funktion BINOM.VERT() zur Verfügung. Die Binomialverteilung kann sich wie in Abbildung 11.14 darstellen.

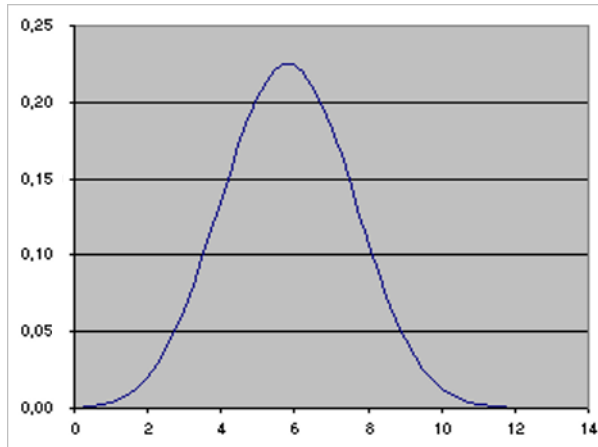


Abbildung 11.14: Binomialverteilung für $p=0,5$ und $n=12$

Die Dichtefunktion der Binomialverteilung lautet:

$$b(x; n, p) = \binom{n}{x} p^x (1-p)^{n-x}$$

wobei:

$$\binom{n}{x}$$

KOMBINATIONEN($n;x$) ist.

Die Verteilungsfunktion der Binomialverteilung lautet:

$$B(x, n, p) = \sum_{y=0}^x b(y; n, p)$$

Praxiseinsatz 1

Wer kennt das nicht: Man befindet sich in einer fremden Stadt, sucht nach dem Weg zu einem bestimmten Ort und fragt einen zufällig ausgewählten Passanten danach. Die Frage »Wissen Sie, wie ich dort hinkomme?« kann nur zwei Antworten zur Folge haben: Ja oder nein. Das heißt, die Wahrscheinlichkeit, dass Sie ein »Ja« zu hören bekommen, beträgt 50 %. Daraus ergibt sich ein p von 0,5.

Sie möchten nun wissen wie hoch die Wahrscheinlichkeit ist, dass von 100 befragten Personen 66 Leute – also ca. $\frac{2}{3}$ der Befragten – mit »Ja« antworten und Sie somit einen Treffer landen.

Argumente für die Funktion	Beispiel	Formel	
AnzahlErfolge	66		
Versuche	100		
Erfolgswahrscheinlichkeit	0,5		
Kumuliert Ja	WAHR	0,9996	=BINOM.VERT(C11;C12;C13;C14)
Kumuliert Nein	FALSCH	0,0005	=BINOM.VERT(C11;C12;C13;C15)

Abbildung 11.15: Die Berechnung der Wahrscheinlichkeit für eine binomialverteilte Zufallsvariable

Die Berechnung der Wahrscheinlichkeit für die binomialverteilte Zufallsvariable 66 stellt sich wie in Abbildung 11.15 aufgezeigt dar.

- ▶ Die Wahrscheinlichkeit, dass Sie bei 100 befragten Personen bis zu bzw. höchstens 66 »Ja-Antworten« erhalten, beträgt nahezu 100 %
- ▶ Die Wahrscheinlichkeit, dass Sie bei 100 befragten Personen genau 66 »Ja-Antworten« erhalten, beträgt 0,05 %

Aussagen

Auf diese Weise können Sie nun eine Vielzahl unterschiedlicher Wahrscheinlichkeiten berechnen.

Nehmen wir das Beispiel aus dem Abschnitt *Hintergrund* zur Funktion BINOM.VERT() / BINOMVERT() auf Seite 362. Hier wurde die Frage gestellt, wie hoch die Wahrscheinlichkeit ist, dass es bei der Produktion von 2.000 Tabletten, die einzeln verpackt werden, zu 30 fehlerhaften Verpackungen kommt, wenn Sie bei der durchschnittlichen Produktion von 2 % fehlerhaften Verpackungen ausgehen?

Praxiseinsatz 2

Das Ergebnis sehen Sie in Abbildung 11.16.

Argumente für die Funktion	Beispiel	Formel	
AnzahlErfolge	30		
Versuche	2000		
Erfolgswahrscheinlichkeit	0,02		
Kumuliert Ja	WAHR	0,0598	=BINOM.VERT(C25;C26;C27;C28)
Kumuliert Nein	FALSCH	0,0181	=BINOM.VERT(C25;C26;C27;C29)

Abbildung 11.16: BINOM.VERT() liefert die Wahrscheinlichkeit für eine binomial verteilte Zufallsvariable

- ▶ Die Wahrscheinlichkeit, dass Sie bei 2.000 verpackten Tabletten genau 30 fehlerhafte Verpackungen erhalten, beträgt 1,81 %
- ▶ Die Wahrscheinlichkeit, dass Sie bei 2.000 verpackten Tabletten 30 bis 40 fehlerhafte Verpackungen erhalten, beträgt 5,98 %

Aussagen

BINOM.INV(), BINOM.VERT.BEREICH(), FAKULTÄT(), HYPGEOM.VERT(), KOMBINATIONEN(), KRITBINOM(), NEGBINOMVERT(), VARIATIONEN(), VARIATIONEN2(), WAHRSCHEBEREICH()

Siehe auch

Diese Beispiele finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* (Excel 2007/2010) auf dem Arbeitsblatt *Binom.vert.*



BINOM.VERT.BEREICH()



BINOM.DIST.RANGE()

Neu in Excel
2013

BINOM.VERT.BEREICH(*Versuche;Erfolgswahrscheinlichkeit;Zahl_Erfolge;*
[Zahl2_Erfolge])

Syntax

Mit der Funktion BINOM.VERT.BEREICH() errechnen Sie die Erfolgswahrscheinlichkeit eines Versuchsergebnisses als Binomialverteilung. Das heißt, Sie können mit beispielsweise 48 angenommenen Erfolgen in 60 Versuchen und einer möglichen Erfolgswahrscheinlichkeit von 75 % die Binomialverteilung berechnen.

Definition

Argumente *Versuche* (erforderlich) ist die Anzahl voneinander unabhängiger Versuche, die durchgeführt werden. Die Zahl muss größer gleich 0 sein.

Erfolgswahrscheinlichkeit (erforderlich) ist die Wahrscheinlichkeit eines Erfolgs für jeden Versuch. Sie muss größer gleich 0 und kleiner gleich 1 sein.

Zahl_Erfolge (erforderlich) ist die Anzahl von Erfolgen in Bezug auf die Anzahl der Versuche. Die Zahl muss größer gleich 0 und kleiner gleich *Versuche* sein.

Zahl2_Erfolge (optional) gibt die Wahrscheinlichkeit zurück, mit der die Anzahl von erfolgreichen Versuchen zwischen *Zahl_Erfolge* und *Zahl2_Erfolge* liegt. Sie muss größer gleich *Zahl_Erfolge* und kleiner gleich *Versuche* sein.

Hinweis Ist der Wert eines Arguments außerhalb seines Wertebereichs, wird für BINOM.VERT.BEREICH() der Fehlerwert #ZAHL! zurückgegeben.

Weist mindestens eines der Argumente einen nicht numerischen Wert auf, wird für BINOM.VERT.BEREICH() der Fehlerwert #WERT! zurückgegeben.

Folgende Gleichung wird beim Einsatz der Funktion verwendet:

$$\sum_{k=S}^{S2} \binom{n}{k} p^k (1-p)^{n-k}$$

In dieser Gleichung ist

- ▶ n = Versuche
- ▶ p = Erfolgswahrscheinlichkeit
- ▶ S = Zahl_Erfolge
- ▶ S2 = Zahl2_Erfolge
- ▶ k = Iterationsvariable

Numerische Argumente werden durch Abschneiden der Nachkommastellen in ganze Zahlen umgewandelt.

Hintergrund Die Binomialverteilung ist eine der wichtigsten Wahrscheinlichkeitsverteilungen. Sie ist ein Spezialfall der Multinomialverteilung und beschreibt Ergebnisse von Bernoulli-Prozessen, die wiederum als Abfolge mehrerer, unter gleich bleibenden Bedingungen durchgeführter Bernoulli-Versuche definiert sind – wie z.B. der Münzwurf.

Hinweis Detaillierte Informationen zur Binomialverteilung sowie den Themen Wahrscheinlichkeit und Zufallsvariablen lesen Sie im Abschnitt BINOM.VERT().

Für die Berechnung der Erfolgswahrscheinlichkeit eines binomialverteilten Versuchsergebnisses steht Ihnen in Excel die Funktion BINOM.VERT.BEREICH() zur Verfügung.

Entsprechend der verwandten Funktion BINOM.VERT(), bei der Sie unter Verwendung des Wahrheitswerts *FALSCH* die Wahrscheinlichkeit berechnen, dass beispielsweise bei 2.000 verpackten Tabletten genau 30 fehlerhafte Verpackungen dabei sind (und in einer durchschnittlichen Produktion 2 % fehlerhafte produziert werden), errechnen Sie mit BINOM.VERT.BEREICH() die gleiche Wahrscheinlichkeit – allerdings nur dann, wenn Sie lediglich das erforderliche Argument *Zahl_Erfolge* verwenden.

Setzen Sie zusätzlich das optionale Argument *Zahl2_Erfolge* ein, rechnen Sie mit einem »Erfolgsbereich«. Es wird dann die Binomialverteilung basierend auf der Wahrscheinlichkeit von 30 bis beispielsweise 40 fehlerhaften Verpackungen bei 2.000 verpackten Tabletten errechnet – dabei wird ebenfalls davon ausgegangen, dass bei einer durchschnittlichen Produktion 2 % fehlerhafte Verpackungen produziert werden.

Bleiben wir bei dem gerade genannten Tablettenbeispiel. Hier wird die Frage gestellt, wie hoch die Wahrscheinlichkeit ist, dass es bei der Produktion von 2.000 Tabletten, die einzeln verpackt werden, zu 30 bzw. 30 bis 40 fehlerhaften Verpackungen kommt, wenn Sie bei der durchschnittlichen Produktion von 2 % fehlerhaften Verpackungen ausgehen?

Praxiseinsatz

Das Ergebnis sehen Sie in Abbildung 11.17.

Argumente für die Funktion	Beispiel	Formel	
Versuche	2000		
Erfolgswahrscheinlichkeit	0,02		
Zahl_Erfolge	30	0,0181	=BINOM.VERT.BEREICH(C11;C12;C13)
Zahl2_Erfolge	40	0,5002	=BINOM.VERT.BEREICH(C11;C12;C13;C14)

Abbildung 11.17: Es wird die Wahrscheinlichkeit für eine binomial verteilte Zufallsvariable berechnet

- ▶ Die Wahrscheinlichkeit, dass Sie bei 2.000 verpackten Tabletten genau 30 fehlerhafte Verpackungen erhalten, beträgt 1,81 %
- ▶ Die Wahrscheinlichkeit, dass Sie bei 2.000 verpackten Tabletten 30 bis (einschließlich) 40 fehlerhafte Verpackungen erhalten, beträgt 50,02 %

Aussagen

BINOM.VERT(), BINOM.INV(), FAKULTÄT(), HYPGEOM.VERT(), KOMBINATIONEN(), KRITBINOM(), NEGBINOMVERT(), VARIATIONEN(), WAHRSCHEBEREICH()

Siehe auch

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Binom.vert.bereich*.



CHIQU.INV.RE() / CHIINV()



CHISQ.INV.RT(), CHIINV()

Die Funktion CHIINV() wird seit Microsoft Excel 2010 durch die Funktion CHIQU.INV.RE() ersetzt und zusätzlich durch die Funktion CHIQU.INV() ergänzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht. Um die Abwärtskompatibilität von CHIQU.INV.RE() zu sichern, ist die Funktion CHIINV() weiter unter ihrem alten Namen verfügbar.

Hinweis

CHIQU.INV.RE(*Wahrscheinlichkeit;FreiheitsGrade*)

Syntax

Die Funktion CHIQU.INV.RE() gibt Perzentile der rechtsseitigen γ^2 -Verteilung zurück. Ist *Wahrscheinlichkeit* = *CHIQU.VERT.RE(x;...)* gegeben, gilt *CHIQU.INV.RE() (Wahrscheinlichkeit;...)* = *x*. Mithilfe dieser Funktion lassen sich zum Zweck der Validierung von Hypothesen beobachtete und erwartete Ergebnisse miteinander vergleichen.

Definition

Wahrscheinlichkeit (erforderlich) ist die zur γ^2 -Verteilung gehörende Wahrscheinlichkeit.

Argumente

FreiheitsGrade (erforderlich) gibt die Anzahl der Freiheitsgrade an.

Hinweis Ist eines der Argumente nicht numerisch, gibt CHIQU.INV.RE() den Fehlerwert #WERT! zurück.

Ist *Wahrscheinlichkeit* kleiner 0 oder *Wahrscheinlichkeit* größer 1, gibt CHIQU.INV.RE() den Fehlerwert #ZAH! zurück.

Ist *FreiheitsGrade* keine ganze Zahl, werden die Nachkommastellen abgeschnitten.

Ist *FreiheitsGrade* kleiner 1 bzw. *FreiheitsGrade* größer oder gleich 10^{10} , gibt CHIQU.INV.RE() den Fehlerwert #ZAH! zurück.

Bei gegebenem Wert für *Wahrscheinlichkeit* sucht CHIQU.INV.RE() den Wert x so, dass $\text{CHIQU.VERT.RE}(x; \text{FreiheitsGrade}) = \text{Wahrscheinlichkeit}$ gilt. Daher hängt die Genauigkeit von CHIQU.INV.RE() von der Genauigkeit von CHIQU.VERT.RE() ab. CHIQU.INV.RE() geht bei der Suche iterativ vor. Hat die Suche nach 100 Iterationsschritten noch nicht konvergiert, gibt die Funktion den Fehlerwert #NV zurück.

Hintergrund Das Ergebnis der Funktion CHIQU.INV.RE() ist die Prüfgröße c eines Vertrauensbereichs einer Chi-Quadrat verteilten Zufallsgröße. Diese Prüfgröße c wird auch kritischer Wert genannt.

Hinweis Mehr zum Thema Chi-Quadrat-Verteilung können Sie unter der Funktion CHIQU.TEST() / CHITEST() auf Seite 370 nachlesen.

Der Vorteil, Messresultate mithilfe von Vertrauensbereichen anzugeben, liegt darin, dass die Verlässlichkeit der Resultate quantifiziert werden kann, da aus Stichproben ermittelte Kennzahlen grundsätzlich mit Fehlern behaftet sind und man davon ausgehen kann, dass sie sich von den wahren Kennzahlen der Grundgesamtheit unterscheiden.

Um ganz genau zu sein, sollte also immer die Kenngröße mit ihrem entsprechenden Vertrauensintervall zu einer definierten *Wahrscheinlichkeit* (z.B. 60 %) angegeben werden. Diese *Wahrscheinlichkeit* wird auch statistische Sicherheit genannt.

Das Argument *Wahrscheinlichkeit* beschreibt das Signifikanzniveau, das mit der Funktion CHIQU.TEST() errechnet werden kann.

Das Argument *FreiheitsGrade* berechnet sich im c^2 -Anpassungstest nach der Anzahl der Summanden, vermindert um 1. Bei statistischen Tests ist die Findung der Anzahl von Freiheitsgraden, also die Anzahl voneinander unabhängiger Daten, von grundlegender Bedeutung. Erst ab Vorliegen von mindestens einem Freiheitsgrad ist ein statistisches Testen überhaupt möglich.

Die Umkehrfunktion zu CHIQU.INV.RE() ist die Funktion CHIQU.VERT.RE().

Praxiseinsatz Sie sind Hersteller von Vitaminpräparaten und wollen beweisen, dass durch die regelmäßige Einnahme von Vitamin-C-Tabletten Erkältungen vorgebeugt werden kann. Um diese Aussage zu prüfen, haben Sie zwei Stichproben aus der gleichen Grundgesamtheit gezogen, wobei 22 der insgesamt 936 Versuchspersonen eine Erkältung hatten.

Die erwarteten Werte sind die aus der ersten, die beobachteten Werte sind die aus der zweiten Stichprobe. Ihr Ziel ist es nun, zunächst aufgrund einer statistischen Berechnung zu beweisen, dass Ihre Annahme, dass Vitamin C vor Erkältungen schützt (Nullhypothese), stimmt.

Hierzu wollen Sie den kritischen Wert für die Zufallsgrößen mit einer vorgegebenen *Wahrscheinlichkeit* von 2,5 % berechnen, wie in Abbildung 11.18 dargestellt.

	A	B	C	D	E
21		Ermittlung der Prüfgröße c / Kritischer Wert			
22		Wahrscheinlichkeit = Signifikanzniveau α		0,025	= vorgegeben
23		FreiheitsGrade		1	=ANZAHL(C11:C12)-1
24		Kritischer Wert		5,0239	=CHIQU.INV.RE(D22;D23)

Abbildung 11.18: Die Berechnung des kritischen Werts über die Funktion *CHIQU.INV.RE()*

Der mit der Funktion *CHIQU.INV.RE()* berechnete kritische Wert liegt mit einem Signifikanzniveau von 2,5 % und einem Freiheitsgrad von 1 bei 5,0239.

Liegt v , als Maß für die Gesamtabweichung, nun unterhalb dieser Prüfgröße, kann die Nullhypothese angenommen werden und Ihre Vermutung, dass Vitamin C vor Erkältungen schützt, wird bestätigt.

Um v zu errechnen, werden die Differenzen der beobachteten Häufigkeiten und der erwarteten Häufigkeiten quadriert und wieder durch die erwartete Häufigkeit dividiert (Abbildung 11.19).

	A	B	C	D
27		Ermittlung der Prüfgröße v / Maß für Gesamtabweichung		
28		= $(C11-C15)^2/C15$		1,4705882
29		= $(C12-C16)^2/C16$		5
30		v=SUMME(D28:D29)		6,4705882

Abbildung 11.19: Die Prüfsumme v wird errechnet, um den Wert mit dem kritischen Wert zu vergleichen

Da v in diesem Fall über dem kritischen Wert c liegt, kann die Nullhypothese nicht angenommen werden. Das heißt, Ihre Vermutung, dass die regelmäßige Einnahme von Vitamin C vor Erkältungen schützt, kann nicht bestätigt werden.

CHIQU.INV(), *CHIQU.INV.RE()*, *CHIQU.TEST()*

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *ChiQu.Inv.Re*.

Siehe auch



CHIQU.INV() CHISQ.INV()

CHIQU.INV(Wahrscheinlichkeit;FreiheitsGrade)

Die Funktion *CHIQU.INV()* gibt Perzentile der linksseitigen χ^2 -Verteilung zurück. Ist *Wahrscheinlichkeit* = *CHIQU.VERT(x;...)* gegeben, gilt *CHIQU.INV(Wahrscheinlichkeit;...)* = x . Mithilfe dieser Funktion lassen sich zum Zweck der Validierung von Hypothesen beobachtete und erwartete Ergebnisse miteinander vergleichen.

Wahrscheinlichkeit (erforderlich) ist die zur χ^2 -Verteilung gehörende Wahrscheinlichkeit.

FreiheitsGrade (erforderlich) gibt die Anzahl der Freiheitsgrade an.

Ist eines der Argumente nicht numerisch, gibt *CHIQU.INV()* den Fehlerwert *#WERT!* zurück.

Ist *Wahrscheinlichkeit* kleiner 0 oder *Wahrscheinlichkeit* größer 1, gibt *CHIQU.INV()* den Fehlerwert *#ZAHL!* zurück.

Ist *FreiheitsGrade* keine ganze Zahl, werden die Nachkommastellen abgeschnitten.

Ist *FreiheitsGrade* kleiner 1 bzw. *FreiheitsGrade* größer oder gleich 10^{10} , gibt *CHIQU.INV()* den Fehlerwert *#ZAHL!* zurück.

Syntax

Definition

Argumente

Hinweis

Bei gegebenem Wert für *Wahrscheinlichkeit* sucht CHIQU.INV() den Wert x so, dass $CHIQU.VERT(x; FreiheitsGrade) = Wahrscheinlichkeit$ gilt. Daher hängt die Genauigkeit von CHIQU.INV() von der Genauigkeit von CHIQU.VERT() ab. CHIQU.INV() geht bei der Suche iterativ vor. Hat die Suche nach 100 Iterationsschritten noch nicht konvergiert, gibt die Funktion den Fehlerwert #NV zurück.

Hintergrund Betrachtet die zuvor beschriebene Funktion CHIQU.INV.RE() die Perzentile der rechtsseitigen χ^2 -Verteilung, beschreibt die Funktion CHIQU.INV() die linksseitige χ^2 -Verteilung.

Hinweis Mehr zum Thema Chi-Quadrat-Verteilung können Sie unter der Funktion CHIQU.TEST() / CHITEST() auf Seite 370 nachlesen.

Die Umkehrfunktion zu CHIQU.INV() ist die Funktion CHIQU.VERT().

Praxiseinsatz Vergleichen Sie hierzu das Beispiel der Funktion CHIQU.INV.RE() auf Seite 365.

Siehe auch CHIQU.INV(), CHIQU.INV.RE(), CHIQU.TEST()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *ChiQu.Inv.*

CHIQU.TEST() / CHITEST() CHISQ.TEST(), CHITEST()

Syntax CHIQU.TEST(*BeobachteteWerte*; *ErwarteteWerte*)

Definition Die Funktion CHIQU.TEST() gibt die Teststatistik eines χ^2 -Unabhängigkeitstests zurück. Das heißt, CHIQU.TEST() liefert den Wert der chi-quadranten (c^2) Verteilung für die Teststatistik (Prüfgröße) mit den entsprechenden Freiheitsgraden. Mithilfe von c^2 -Tests können Sie feststellen, ob in Experimenten die Ergebnisse bestätigt werden, die aufgrund von Hypothesen erwartet wurden.

Argumente *BeobachteteWerte* (erforderlich) ist der Bereich beobachteter Daten, mit dem Sie die erwarteten Werte testen möchten.

ErwarteteWerte (erforderlich) ist der Bereich erwarteter Beobachtungen, die sich aus der Division der miteinander multiplizierten Rangsummen und der Gesamtsumme berechnen.

Hinweis Enthalten *BeobachteteWerte* und *ErwarteteWerte* nicht dieselbe Anzahl von Datenpunkten, gibt CHIQU.TEST() den Fehlerwert #NV zurück.

Der c^2 -Test berechnet zunächst eine c^2 -Statistik, wozu er folgende Formel verwendet:

$$x^2 = \sum_{j=1}^i \sum_{j=1}^c \frac{(A_{ij} - E_{ij})^2}{E_{ij}}$$

wobei:

A_{ij} = tatsächliche Häufigkeit in der i -ten Zeile, j -ten Spalte

E_{ij} = erwartete Häufigkeit in der i -ten Zeile, j -ten Spalte

Z = Zeilenanzahl

S = Spaltenanzahl

Ein niedriger Wert von c^2 ist ein Kennzeichen für Unabhängigkeit. Wie anhand der Formel zu sehen ist, ist c^2 immer positiv oder gleich 0, aber 0 nur dann, wenn $A_{ij} = E_{ij}$ für jedes ij .

CHIQU.TEST() gibt die Wahrscheinlichkeit zurück, dass ein Wert der c^2 -Verteilung mit mindestens dem Wert, der das Ergebnis der obigen Formel war, zufällig passiert sein könnte, wenn *Unabhängigkeit* angenommen wird.

Für das Berechnen dieser Wahrscheinlichkeit verwendet die Funktion CHIQU.TEST() die c^2 -Verteilung mit einer geeigneten Anzahl von *Freiheitsgraden* (df).

- ▶ Bei r größer 1 und c größer 1 ist $df = (r - 1)(c - 1)$
- ▶ Bei r gleich 1 und c größer 1 ist $df = c - 1$
- ▶ Bei r größer 1 und c gleich 1 ist $df = r - 1$

$r = c = 1$ ist nicht zulässig, sodass in diesem Fall der Fehlerwert #NV zurückgegeben wird.

Damit CHIQU.TEST() brauchbar verwendet werden kann, sollte die Zahl der E_{ij} nicht zu klein sein. Einige Statistiker schlagen vor, dass jedes E_{ij} größer oder gleich 5 sein sollte.

Für die Analyse statistischer Zusammenhänge werden sogenannte Tests durchgeführt. Für den untersuchten Sachverhalt wird eine Nullhypothese formuliert. Diese Nullhypothese gibt an, welche Ergebnisse erwartet werden, wenn der statistische Zusammenhang exakt einen bestimmten Wert erreicht. Die Gegenhypothese, auch Alternativhypothese, ist häufig weniger scharf abgegrenzt und zeigt die Spannweite für die untersuchten Methoden.

Hintergrund

So nimmt beispielsweise bei der Ziehung der Lottozahlen die Nullhypothese an, dass kein Unterschied in der Wahrscheinlichkeit der Ziehung einer Kugel besteht, d.h. keine Zahl öfter gezogen wird als eine andere.

Die Gegenhypothese vermutet dies aber. Dabei wird ein sogenanntes Signifikanzniveau (a) zugrunde gelegt, das ein Maß für den erlaubten prozentualen Anteil an Fehlern darstellt, um eine Stichprobe noch als zufällig zu bezeichnen, d.h. die Nullhypothese zu bestätigen.

Beträgt das Signifikanzniveau beispielsweise $a = 0,03$, kann in drei von hundert Ziehungen eine Zahl öfter vorkommen als eine andere, um zu behaupten, dass trotzdem eine gleiche Chance für alle Kugeln besteht, d.h. die Nullhypothese angenommen wird. Beträgt der Fehler bis zu 3 %, ist das Ergebnis »signifikant«.

Einer der Tests, um statistische Zusammenhänge zu analysieren, ist der sogenannte Chi-Quadrat-Test. Beim Chi-Quadrat-Test wird die Abweichung zwischen beobachteten und erwarteten Häufigkeiten quadriert und die Summe der daraus resultierenden Dichtefunktion zur Annahme oder Ablehnung der Nullhypothese genutzt

Der c^2 -Test ist ein Verfahren der schließenden Statistik. Es gibt zum einen den c^2 -Anpassungstest, bei dem untersucht wird, ob die Grundgesamtheit einer erwarteten Verteilung entspricht. So kann z.B. getestet werden, ob bei der Ziehung der Lottozahlen jede Kugel die gleiche Chance hat, gezogen zu werden.

Der c^2 -Unabhängigkeitstest stellt ein Verfahren für nichtmessbare (qualitative) Merkmale dar. Hierbei wird beispielsweise überprüft, ob die in einer Stichprobe gemachten Beobachtungen bezüglich des Zusammenhangs zwischen Vitamin C und der Vorbeugung vor Erkältungen von anderen Stichproben bestätigt wurden.

Zur Berechnung verschiedener Komponenten bzw. Ergebnisse für einen c^2 -Test stehen Ihnen in Excel verschiedene Funktionen zur Verfügung. Dazu gehören:

- ▶ CHIUQ.TEST() – Berechnet das Signifikanzniveau beim c^2 -Test
- ▶ CHIUQ.VERT.RE() – Berechnet die Überschreitungswahrscheinlichkeit für die c^2 -Verteilung mit Freiheitsgraden
- ▶ CHIUQ.INV.RE() – Berechnet den kritischen Wert

Diese Funktionen liefern verschiedene Prüfgrößen, bei deren Vergleich man Rückschlüsse auf die aufgestellten Hypothesen ziehen kann. Die folgende Tabelle zeigt, wann welche Hypothese anzunehmen ist.

Tabelle 11.2
Wann wird die Nullhypothese abgelehnt bzw. angenommen?

Vergleich der Ergebnisse von	Annahme von
$\nu > c$	Gegenhypothese
$\nu < c$	Nullhypothese
Chiquetest $> a$	Gegenhypothese
Chiquetest $< a$	Nullhypothese
Chiquverte für $\nu > a$	Gegenhypothese
Chiquverte für $\nu < a$	Nullhypothese

Wie bereits erwähnt, berechnet die Funktion CHIUQ.TEST() das *Signifikanzniveau* dafür, dass die Werte einer Wertereihe mit den erwarteten Werten übereinstimmen. Hier wird also jeder Einzelwert der einen Datenreihe mit dem korrespondierenden Wert der zweiten Reihe verglichen.

Das errechnete Signifikanzniveau, das durch einen Zahlenwert zwischen 0 und 100 % dargestellt wird, drückt die Testsicherheit bei statistischen Hypothesentests aus. Erhalten Sie beispielsweise als Ergebnis für das Signifikanzniveau 90 % bedeutet dies, dass das vorliegende Zahlenmaterial, sofern es zufällig zustande gekommen ist, in nur 100 minus 90 = 10 % aller Fälle eine vergleichbare Ausprägung annehmen würde.

90 % Signifikanz bedeutet demnach also nicht, ...

- ▶ ... dass das vorliegende Datenmaterial mit nur 10%iger Wahrscheinlichkeit zufällig ist
- ▶ ... dass man mit 90%iger Wahrscheinlichkeit auf einen bestimmten Sachverhalt schließen kann

Das Signifikanzniveau ist also ein Maß für den erlaubten prozentualen Anteil an Fehlern, um eine Stichprobe als zufällig zu bezeichnen.

Praxiseinsatz

Sie sind Hersteller von Vitaminpräparaten und wollen, um Ihr Marketing mithilfe statistischer geprüfter Aussagen zu unterstützen, testen, ob durch die regelmäßige Einnahme von Vitamin-C-Tabletten Erkältungen vorgebeugt werden kann.

Um diese Aussage zu prüfen, haben Sie zwei Stichproben aus der gleichen Grundgesamtheit gezogen, wobei 22 der insgesamt 936 Versuchspersonen eine Erkältung hatten.

Die erwarteten Werte sind die aus der ersten, die beobachteten Werte sind die aus der zweiten Stichprobe.

Zunächst wollen Sie beweisen oder feststellen, wie hoch das Signifikanzniveau der Werte ist, um eine Aussage darüber treffen zu können, ob die Stichproben als zufällig zu bezeichnen sind und die Werte übereinstimmen.

Hierfür verwenden Sie die Funktion CHIQU.TEST(), wie in Abbildung 11.20 dargestellt.

	B	C	D	E	F
9	Werte	Erkältung	Keine Erkältung	Summe	
10	Beobachtete Werte				
11	Vitamin C	12	704	716	=SUMME(C11:D11)
12	Kein Vitamin C	10	210	220	=SUMME(C11:D11)
13	Summe	22	914	936	=SUMME(E11:E11)
14	Erwartete Werte				
15	Vitamin C	17	699	716	=SUMME(C15:D15)
16	Kein Vitamin C	5	215	220	=SUMME(C16:D16)
17	Summe	22	914	936	=SUMME(E15:E16)
18	CHIQU.TEST	0,0109674			
19		=CHIQU.TEST(C11:C12;C15:C16)			

Abbildung 11.20: Die Berechnung von *CHIQU.TEST()* mit den beobachteten und erwarteten Werten

Das errechnete Signifikanzniveau von 1 % sagt aus, dass das vorliegende Zahlenmaterial (Stichproben) in 100 minus 1, also in 99 % aller Fälle eine vergleichbare Ausprägung annimmt – somit können die Stichproben als zufällig bezeichnet werden.

Der durch CHIQU.TEST() errechnete Wert kann zusätzlich zur bisher dargestellten Aussagekraft auch die Wahrscheinlichkeit zu einem berechneten ν darstellen. Die Variable ν ist eine Prüfgröße, die ein Maß für die Gesamtabweichung darstellt und mit folgender Formel berechnet wird:

$$CHI - Formel r = 1 - \frac{6 \sum_{i=1}^n D_i^2}{n(n^2 - 1)}$$

Wenn die Prüfgröße ν einen sehr großen Wert annimmt, wird die Nullhypothese abgelehnt, was bedeutet, dass keine gleiche Wahrscheinlichkeit für alle Tests angenommen wird.

Ist die Wahrscheinlichkeit für ν kleiner als das diesem Wert zugeordnete Signifikanzniveau, kann der Unterschied als signifikant bezeichnet werden – dies bedeutet, dass die Gegenhypothese angenommen wird. Ist die Wahrscheinlichkeit für ν größer, ist der Unterschied somit nicht signifikant und die Nullhypothese wird vermutet.

CHIQU.INV.RE(), CHIQU.INV(), CHIQU.VERT(), CHIQU.VERT.RE()

Siehe auch

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *ChiQu.Test*.



CHIQU.VERT.RE() / CHIVERT()

CHISQ.VERT.RT(), CHIDIST()

Hinweis Die Funktion CHIVERT() wird seit Microsoft Excel 2010 durch die Funktion CHIQU.VERT.RE() ersetzt und zusätzlich durch die Funktion CHIQU.VERT() ergänzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht. Um die Abwärtskompatibilität von CHIQU.VERT.RE() zu sichern, ist die Funktion CHIVERT() weiter unter ihrem alten Namen verfügbar.

Syntax CHIQU.VERT.RE(*x*; *FreiheitsGrade*)

Definition Die Funktion CHIQU.VERT() gibt Werte der rechtsseitigen Verteilungsfunktion (1-Alpha oder Signifikanzrisiko) einer Chi-Quadrat-verteilten Zufallsgröße zurück. Die c^2 -Verteilung wird bei einem c^2 -Test benötigt. Mit diesem Test lassen sich beobachtete und erwartete Werte miteinander vergleichen. Beispielsweise wird in einem Test die Hypothese aufgestellt, dass durch die regelmäßige Einnahme von Vitaminpräparaten Erkältungen vorgebeugt werden kann. Durch Vergleichen der beobachteten mit den erwarteten Ergebnissen lässt sich die Hypothese validieren.

Argumente *x* (erforderlich) ist der Wert (Quantil), dessen Wahrscheinlichkeit (1-Alpha) Sie berechnen möchten.

FreiheitsGrade (erforderlich) gibt den Grad der Freiheit an.

Hinweis Ist eines der Argumente nicht numerisch, gibt CHIQU.VERT.RE() den Fehlerwert #WERT! zurück.

Ist *x* negativ, gibt CHIQU.VERT.RE() den Fehlerwert #ZAHL! zurück.

Ist *FreiheitsGrade* keine ganze Zahl, werden deren Nachkommastellen abgeschnitten.

Ist *FreiheitsGrade* kleiner 1 bzw. *FreiheitsGrade* größer oder gleich 10^{10} , gibt CHIQU.VERT.RE() den Fehlerwert #ZAHL! zurück.

CHIQU.VERT.RE() wird berechnet als $CHIQU.VERT.RE = P(X > x)$, wobei *X* eine Zufallsvariable mit der Verteilung c^2 ist.

Hintergrund Wie bereits oben im Abschnitt *Definition* erwähnt, berechnet die Funktion CHIQU.VERT.RE() Werte der Verteilungsfunktion einer Chi-Quadrat verteilten Zufallsgröße durch die Berechnung von 1-Alpha bzw. 1-Signifikanzniveau. Diese Werte werden auch »Alpha-Risiko« genannt und drücken in Form einer Zahl ein Risiko aus, mit dem darauf hingewiesen wird, dass eine statistische Hypothese möglicherweise gar nicht vorhanden ist oder nicht bestätigt werden kann.

In anderen Worten ausgedrückt kann man deshalb auch sagen, dass das Ergebnis von CHIQU.VERT.RE() eine Aussage über eine Irrtumswahrscheinlichkeit liefert.

Hinweis Mehr Informationen zum Thema Chi-Test und dem Signifikanzniveau erfahren Sie unter der Funktion CHIQU.TEST() auf Seite 370.

Die Umkehrfunktion von CHIQU.VERT.RE() ist die Funktion CHIQU.INV.RE().

Sie sind Hersteller von Vitaminpräparaten und wollen beweisen, dass durch die regelmäßige Einnahme von Vitamin-C-Tabletten Erkältungen vorgebeugt werden kann.

Praxiseinsatz

Um diese Aussage zu prüfen, haben Sie zwei Stichproben aus der gleichen Grundgesamtheit gezogen, wobei 22 der insgesamt 936 Versuchspersonen eine Erkältung hatten. Die erwarteten Werte sind die aus der ersten, die beobachteten Werte sind die aus der zweiten Stichprobe.

Ihr Ziel ist es nun, zunächst aufgrund einer statistischen Berechnung zu beweisen, dass Ihre Annahme, dass Vitamin C vor Erkältungen schützt (Nullhypothese), stimmt.

Den kritischen Wert c mit einer vorgegebenen Wahrscheinlichkeit (Signifikanzniveau a) von 2,5 % sowie die Freiheitsgrade und ν als Maß für die Gesamtabweichung haben Sie bereits berechnet.

Wie Sie die soeben genannten Komponenten berechnen, können Sie unter CHIQU.TEST() auf Seite 370 und unter CHIQU.INV.RE() auf Seite 365 nachlesen.

Hinweis

Bis jetzt mussten Sie, aufgrund der Ergebnisse Ihrer bisherigen statistischen Berechnungen, die Nullhypothese – also die Annahme, dass Vitamin C vor Erkältungen schützt – immer wieder verwerfen.

Deshalb möchten Sie nun einen letzten Versuch starten, diese Nullhypothese doch zu bestätigen. Sie wollen mit der Funktion CHIQU.VERT.RE() die Wahrscheinlichkeit für ν (Maß für Gesamtabweichung) berechnen, um anschließend das Signifikanzniveau a und den errechneten Wahrscheinlichkeitswert zu vergleichen.

- ▶ Ist CHIQU.VERT.RE für ν größer a , wird die Nullhypothese verworfen
- ▶ Ist CHIQU.VERT.RE für ν kleiner a , wird die Nullhypothese angenommen

Aussagen

Die Berechnung der Wahrscheinlichkeit für ν erfolgt wie in Abbildung 11.21 dargestellt.

	B	C	D	E
20	Ermittlung der Prüfgröße c / Kritischer Wert			
21	Wahrscheinlichkeit = Signifikanzniveau a		0,025	= vorgegeben
22	FreiheitsGrade		1	=ANZAHL(C11:C12)-1
23	Kritischer Wert		5,0239	=CHIQU.INV.RE(D21;D22)
24				
25				
26	Ermittlung der "Irrtumswahrscheinlichkeit" für ν			
27	= $(C11-C15)^2/C15$		1,4705882	
28	= $(C12-C16)^2/C16$		5	
29	ν =SUMME(D27:D28)		6,4705882	
30	CHIQU.VERT.RE		0,0109674	=CHIQU.VERT.RE(D29;D22)

Abbildung 11.21: CHIQU.VERT.RE() liefert die Irrtumswahrscheinlichkeit für den Wert ν

Vergleicht man nun den Chiquvertre für ν , sieht man, dass ν kleiner dem Signifikanzniveau a ist. Daraus folgt, dass die Nullhypothese angenommen werden kann.

CHIQU.TEST(), CHIQU.INV(), CHIQU.INV.RE()

Siehe auch

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Chiqu.Vert.Re*.



CHIQU.VERT()



CHISQ.DIST(), CHIDIST()

Syntax CHIQU.VERT(*x*; *FreiheitsGrade*; *kumuliert*)

Definition Die Funktion CHIQU.VERT() gibt die linksseitigen Werte der Verteilungsfunktion einer χ^2 -verteilten Zufallsvariablen zurück. Die Betaverteilung wird i.d.R. verwendet, um die Streuung bei mehreren Stichproben zu bestimmten Vorgängen zu untersuchen. Beispielsweise kann prozentual ermittelt werden, wie viel Zeit am Tag Personen vor dem Fernsehgerät verbringen.

Argumente *x* (erforderlich) ist der Wert (Quantil), dessen Wahrscheinlichkeit (1-Alpha) Sie berechnen möchten.

FreiheitsGrade (erforderlich) gibt den Grad der Freiheit an.

kumuliert (erforderlich) definiert den Wahrheitswert, der den Typ der Funktion bestimmt. Ist *kumuliert* mit WAHR belegt, gibt CHIQU.VERT() den Wert der Verteilungsfunktion zurück. Ist *kumuliert* mit FALSCH belegt, gibt CHIQU.VERT() den Wert der Dichtefunktion zurück.

Hinweis Ist eines der Argumente nicht numerisch, gibt CHIQU.VERT den Fehlerwert #WERT! zurück.

Ist *x* negativ, gibt CHIQU.VERT() den Fehlerwert #ZAHL! zurück.

Ist *FreiheitsGrade* keine ganze Zahl, werden deren Nachkommastellen abgeschnitten.

Ist *FreiheitsGrade* kleiner 1 bzw. *FreiheitsGrade* größer oder gleich 10^{10} , gibt CHIQU.VERT() den Fehlerwert #ZAHL! zurück.

Hintergrund Betrachtet die zuvor beschriebene Funktion CHIQU.VERT.RE() die Werte der rechtsseitigen Verteilungsfunktion (1-Alpha oder Signifikanzrisiko) einer Chi-Quadrat-verteilten Zufallsgröße, beschreibt die Funktion CHIQU.VERT() die linksseitige Verteilungsfunktion einer Chi-Quadrat-verteilten Zufallsgröße.

Hinweis Mehr Informationen zum Thema Chi-Test und dem Signifikanzniveau erfahren Sie unter der Funktion CHIQU.TEST() auf Seite 370.

Die Umkehrfunktion von CHIQU.VERT() ist die Funktion CHIQU.INV().

Praxiseinsatz Vergleichen Sie hierzu das Beispiel der Funktion CHIQU.VERT.RE() auf Seite 374.

Siehe auch CHIQU.TEST(), CHIQU.VERT(), CHIQU.VERT.RE()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *ChiQu.Vert.*

EXPON.VERT() / EXPONVERT()



EXPON.DIST() / EXPONDIST()

Syntax EXPON.VERT(*x*; *Lambda*; *kumuliert*)

Definition Die Funktion EXPON.VERT() gibt Wahrscheinlichkeiten einer exponential verteilten Zufallsvariablen zurück. Mithilfe der EXPON.VERT()-Funktion lassen sich Zeiträume zwischen Ereignissen modellieren. Eine Frage könnte z.B. sein, wie hoch die Wahrscheinlichkeit ist, dass in einem Call-Center bereits nach zwei Minuten ein Anruf eingeht, obwohl der Durchschnitt pro Stunde bei drei Minuten liegt.

x (erforderlich) ist der Wert für die Funktion.

λ (erforderlich) ist der übergebene Wert.

kumuliert (erforderlich) ist ein Wahrheitswert, der den Typ der Funktion bestimmt. Ist *kumuliert* mit *WAHR* belegt, gibt EXPON.VERT() den Wert der Verteilungsfunktion zurück. Ist *kumuliert* mit *FALSCH* belegt, gibt EXPON.VERT() den Wert der Dichtefunktion zurück.

Ist x oder λ nicht numerisch, gibt EXPON.VERT() den Fehlerwert #WERT! zurück.

Ist x kleiner 0, gibt EXPON.VERT() den Fehlerwert #ZAHL! zurück.

Ist λ kleiner oder gleich 0, gibt EXPON.VERT() den Fehlerwert #ZAHL! zurück.

Eine Exponentialfunktion liegt vor, wenn der Exponent einer Potenz als Variable betrachtet wird. Derartige Funktionen besitzen eine besondere Eigenschaft: In gleich großen Intervallen ändert sich ihr Funktionswert um den gleichen Faktor. Sie eignen sich daher hervorragend dazu, Wachstums- oder Zerfallsprozesse zu beschreiben, für die sich die betrachtete Größe in gleich langen Zeitintervallen um den gleichen Faktor ändert. Ihre Umkehrfunktionen heißen Logarithmen – ihr Zweck besteht darin, aus der Kenntnis einer Potenz und ihrer Basis den Exponenten zu gewinnen.

Als die Exponentialfunktion im engeren Sinne wird die Exponentialfunktion $x = e^x$ mit der Eulerschen Zahl e als Basis bezeichnet.

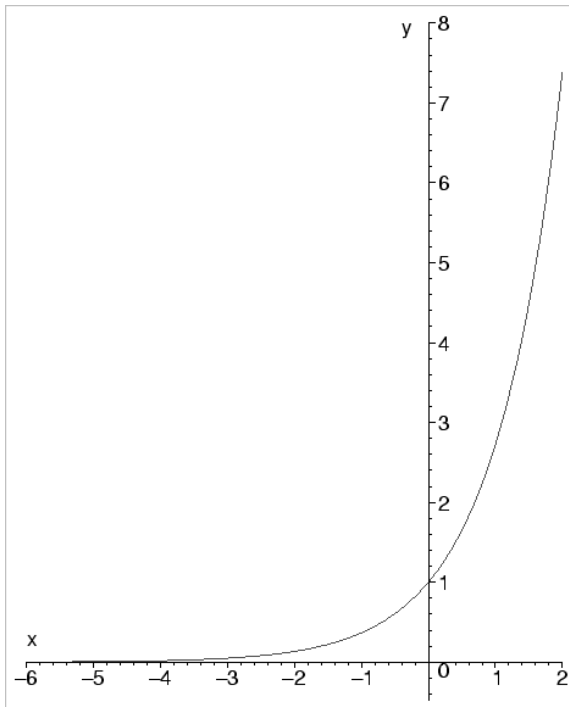


Abbildung 11.22: Die grafische Darstellung der Exponentialfunktion $y = e^x$

Argumente

Hinweis

Hintergrund

Wie bereits oben im Abschnitt *Definition* erwähnt, liefert die Funktion EXPON.VERT() die Wahrscheinlichkeiten für unabhängige Ereignisse, wie bei dem Beispiel »Ausgabezeit an einem Geldautomat« oder anderen Wartezeiten.

Ein anderes Beispiel wäre:

Wenn bekannt ist, dass in einem Support-Callcenter alle drei Minuten ein Anruf eingeht, können Sie mit dieser Funktion die Wahrscheinlichkeit berechnen, dass das Telefon schon nach spätestens einer Minute klingelt.

Da die Ereignisse »Telefonklingeln« und »Zeit« unabhängig voneinander sind, handelt es sich hierbei um exponentialverteilte Zufallsvariablen. Die Funktion EXPON.VERT() liefert die Wahrscheinlichkeiten dazu.

Die Gleichung für die Wahrscheinlichkeitsdichte lautet:

$$f(x; \lambda) = \lambda e^{-\lambda x}$$

Die Gleichung für die Verteilungsfunktion lautet:

$$F(x; \lambda) = 1 - e^{-\lambda x}$$

Praxiseinsatz

Bleiben wir bei dem oben genannten Beispiel mit dem Callcenter. Sie betreiben ein Callcenter für einen Druckerhersteller. Das Callcenter ist 24 Stunden am Tag besetzt. Sie möchten das Anrufverhalten der Kunden näher analysieren und haben deshalb einen Tag lang die eingehenden Anrufe pro Stunde gezählt. Das heißt, das Zeitintervall beträgt 60 Minuten.

Ihre Aufzeichnungen eingegangener Anrufe hat folgende Statistik ergeben:

Zeitintervall = 1 Stunde	Call Center Anrufe
Stunde 1	8
Stunde 2	9
Stunde 3	10
Stunde 4	20
Stunde 5	18
Stunde 6	22
Stunde 7	34
Stunde 8	32
Stunde 9	42
Stunde 10	37
Stunde 11	25
Stunde 12	20
Stunde 13	23
Stunde 14	26
Stunde 15	31
Stunde 16	21
Stunde 17	28
Stunde 18	20
Stunde 19	21
Stunde 20	38
Stunde 21	13
Stunde 22	2
Stunde 23	4
Stunde 24	1
Anzahl eingegangener Anrufe	505
Mittelwert	21,04

Abbildung 11.23: Die eingegangenen Anrufe eines Tags

Durch die Berechnung des Mittelwerts aller eingegangenen Anrufe können Sie nun folgende Aussagen treffen:

- ▶ Jede Stunde, also alle 60 Minuten, gehen im Schnitt 21 Anrufe ein
- ▶ Daraus folgt, dass knapp alle drei Minuten ein Kunde anruft

Nun möchten Sie wissen, wie hoch die Wahrscheinlichkeit ist, dass bereits nach zwei Minuten eine Person mit einer Frage zu einem Drucker anruft. Hierzu nutzen Sie die Funktion EXPON.VERT().

Welche Angaben machen Sie für die einzelnen Argumente der Funktion?

- ▶ $x = 2$, da die Wahrscheinlichkeit für einen Anruf nach zwei Minuten berechnet werden soll
- ▶ $\text{Lambda} = 3$, da dies die mittlere Anzahl an Ereignissen pro Zeitintervall darstellt und somit der übergebene Wert ist
- ▶ *kumuliert* = WAHR, da in unserem Beispiel die Verteilungsfunktion geliefert werden soll

Die Berechnung der Wahrscheinlichkeiten über die Funktion EXPON.VERT() liefert die in Abbildung 11.24 dargestellten Ergebnisse.

	E	F	G	H	I
21	Berechnung				
22	x (Minuten)	2	1	0,5	0,2
23	Lambda	3	3	3	3
24	Kumuliert	WAHR	WAHR	WAHR	WAHR
25	EXPON.VERT()	0,9975	0,9502	0,7769	0,4512
26		=EXPON.VERT(F22;F23;F24)			

Abbildung 11.24: EXPON.VERT() liefert die Wahrscheinlichkeit für einen gesuchten Wert

Die Wahrscheinlichkeit, dass bereits nach zwei Minuten ein Anruf im Callcenter eingeht, beträgt, laut der Berechnung über die Funktion EXPON.VERT(), 99 %.

Wie Sie des Weiteren der Abbildung 11.24 entnehmen können, nimmt die Wahrscheinlichkeit immer weiter ab, je kleiner die Einträge für das Argument x werden. Das heißt, die Wahrscheinlichkeit, dass nach 0,2 Minuten, also nach 12 Sekunden, bereits jemand anruft, beträgt lediglich 45 %.

GAMMA.VERT(), POISSON()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Expon.vert.*

Aussagen

Siehe auch



F.INV.RE() / FINV()



F.DIST.RT() / FINV()

Die Funktion FINV() wird seit Microsoft Excel 2010 durch die Funktion F.INV.RE() ersetzt und zusätzlich durch die Funktion F.INV() ergänzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht.

Um die Abwärtskompatibilität von F.INV.RE() zu sichern, ist die Funktion FINV() weiter unter ihrem alten Namen verfügbar.

Hinweis

Syntax F.INV.RE(*Wahrsch*; *Freiheitsgrade1*; *Freiheitsgrade2*)

Definition Die Funktion F.INV.RE() gibt Quantile der rechtsseitigen F-Verteilung zurück. Ist $p = F.VERT.RE(x, \dots)$, dann ist $F.INV.RE(p, \dots) = x$.

Die F-Verteilung kann in F-Tests verwendet werden, bei denen die Streuungen zweier Datenmengen ins Verhältnis gesetzt werden. Zum Beispiel können Sie die Verteilung der in Deutschland und Österreich erzielten Einkommen daraufhin analysieren, ob in den beiden Ländern ähnliche Einkommensverteilungen vorliegen.

Argumente *Wahrsch* (erforderlich) ist die zur F-Verteilung gehörige Wahrscheinlichkeit.

Freiheitsgrade1 (erforderlich) ist die Anzahl der Freiheitsgrade im Zähler.

Freiheitsgrade2 (erforderlich) ist die Anzahl der Freiheitsgrade im Nenner.

Hinweis Ist eines der Argumente nicht numerisch, gibt F.INV.RE() den Fehlerwert #WERT! zurück.

Ist *Wahrsch* kleiner 0 bzw. *Wahrsch* größer 1, gibt F.INV.RE() den Fehlerwert #ZAHL! zurück.

Ist *Freiheitsgrade1* oder *Freiheitsgrade2* keine ganze Zahl, wird der Dezimalanteil abgeschnitten.

Ist *Freiheitsgrade1* kleiner 1 bzw. *Freiheitsgrade2* größer oder gleich 10^{10} , gibt F.INV.RE() den Fehlerwert #ZAHL! zurück.

Ist *Freiheitsgrade2* kleiner 1 bzw. *Freiheitsgrade2* größer oder gleich 10^{10} , gibt F.INV.RE() den Fehlerwert #ZAHL! zurück.

Die Funktion F.INV.RE() kann dazu verwendet werden, kritische Werte der F-Verteilung zu berechnen. Zum Beispiel umfasst die Ausgabe einer ANOVA-Berechnung (Analysis of Variance) häufig Daten für die F-Verteilung (F-Statistik), F-Wahrscheinlichkeit sowie den kritischen F-Wert beim Signifikanzniveau 0,05. Wenn Sie den kritischen Wert von F ermitteln möchten, müssen Sie der Funktion F.INV.RE() das Signifikanzniveau als Argument *Wahrsch* übergeben.

Bei gegebenem Wert für *Wahrsch* sucht F.INV.RE() den Wert x so, dass $F.VERT.RE(x; \text{Freiheitsgrade1}; \text{Freiheitsgrade2}) = \text{Wahrsch}$ gilt. Daher hängt die Genauigkeit von F.INV.RE() von der Genauigkeit von F.VERT.RE() ab. F.INV.RE() geht bei der Suche iterativ vor. Hat die Suche nach 100 Iterationsschritten noch nicht konvergiert, gibt die Funktion den Fehlerwert #NV zurück.

Hintergrund Wie oben im Abschnitt *Argumente* angegeben, umfasst die Ausgabe einer ANOVA-Berechnung (Analysis of Variance) häufig Daten für die F-Verteilung (F-Statistik), die F-Wahrscheinlichkeit sowie den kritischen F-Wert beim Signifikanzniveau 0,05.

Mit dieser Funktion wird die einfache Varianzanalyse durchgeführt, um die Hypothese zu überprüfen, ob die Mittelwerte zweier oder mehrerer (aus Grundgesamtheiten mit demselben Mittelwert gezogenen) Stichproben identisch sind.

Die einfaktorielle Varianzanalyse wird dann angewendet, wenn es sich bei der Untersuchung um drei oder mehr Gruppen handelt, von denen mindestens eine Gruppe unabhängig von den anderen ist.

Sie testet gleichzeitig die Signifikanz der Differenz unter den arithmetischen Mitteln dieser Gruppen. Ihr Hauptzweck ist, herauszufinden, ob die Varianz zwischen Gruppenmittelwerten zufallsbedingt ist oder nicht.

Die Funktion F.INV.RE() berechnet den kritischen Wert einer Verteilung. Um diesen zu ermitteln, müssen Sie der Funktion F.INV.RE() das Signifikanzniveau als Argument *Wahrsch* übergeben.

Durch Berechnung von F.INV.RE() kann eine Aussage über die Nullhypothese getroffen werden. Die Argumente der Funktion F.INV.RE() sind die Wahrscheinlichkeit, d.h. das Signifikanzniveau α , und die Anzahl der Freiheitsgrade 1 und 2.

Sie sind Betriebspsychologe und möchten untersuchen, wie sich die Mitarbeiter mit dem Unternehmen identifizieren. Dazu haben Sie 15 Mitarbeiter zufällig ausgewählt. Jeder Mitarbeiter hat zehn Fragen aus anderen Fachgebieten beantwortet. Es waren dabei drei Antworten zur Auswahl vorgegeben, die angekreuzt werden konnten.

Die Antworten haben Sie bereits zusammengefasst und eine Zusammenfassung allgemeiner Ergebnisse ist erstellt (siehe Abbildung 11.25).

Praxiseinsatz

	B	C	D	E	F
8	Gruppe I	Gruppe II	Gruppe III		
9	4	2	2		
10	2	1	1		
11	1	4	6		
12	5	7	4		
13	2	5	7		
14					
15	ZUSAMMENFASSUNG				
16	Gruppen	Anzahl	Summe	Mittelwert	Varianz
17	Gruppe I	5	14	2,8	2,7
18	Gruppe II	5	19	3,8	5,7
19	Gruppe III	5	20	4,0	6,5

Abbildung 11.25: Die Ergebnisse der Befragung

Die Nullhypothese besagt, dass es keinen Unterschied zwischen den drei Gruppen gibt. Die Alternativhypothese vermutet einen Unterschied.

Das Signifikanzniveau sei 0,05.

Das Ergebnis der einfaktoriellen Varianzanalyse sehen Sie in Abbildung 11.26.

	B	C	D	E	F	G	H
22	Streuungsursache	Quadratsummen (SS)	Freiheitsgrade (df)	Mittlere Quadratsumme (MS)	Prüfgröße (F)	P-Wert	kritischer F-Wert
23	Unterschiede						
24	- zwischen den Gruppen	4,13	2	2,07	0,416107	0,67	3,89
25	- innerhalb der Gruppen	59,60	12	4,97			=F.INV.RE(0,05;D24;D25)
26	Gesamt	63,73	14				

Abbildung 11.26: Die Berechnung der einfaktoriellen Varianzanalyse und dem kritischen F-Wert

Das Ergebnis unterscheidet zwischen »Unterschiede zwischen den Gruppen« und »Unterschiede innerhalb der Gruppen«, denn die drei Gruppen unterscheiden sich zum einen untereinander, zum anderen unterscheiden sich aber auch die Ergebnisse der einzelnen Mitarbeiter »innerhalb der Gruppen«.

Die »Unterschiede zwischen den Gruppen« entsprechen dem zu untersuchenden Unterschied, die Unterschiede »innerhalb der Gruppen« entsprechen einem Zufallseinfluss.

Die Anzahl der Freiheitsgrade 1, das sind die Freiheitsgrade innerhalb der Gruppen, berechnet sich nach dem Umfang der drei Gruppen, jeweils um 1 vermindert, also $5-1+5-1+5-1=12$. Die Anzahl der Freiheitsgrade 2, also die Freiheitsgrade zwischen den Gruppen, ergibt sich aus der Anzahl der Gruppen minus 1, also $3-1=2$.

Für die Prüfgröße F erhält man einen Wert von 0,42 (Zelle F24). Vergleicht man diesen mit dem kritischen Wert F_{krit} , ermittelt durch die Funktion F.INV.RE(), kann man eine Aussage zur Nullhypothese aufstellen.

Ist der Wert der berechneten Prüfgröße F größer oder gleich F_{krit} , wird die Nullhypothese abgelehnt. In unserem Beispiel ist dies nicht der Fall, das heißt, die Nullhypothese wird angenommen und man geht davon aus, dass kein signifikanter Unterschied zwischen den drei Gruppen besteht.

Siehe auch F.TEST(), F.INV()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *F.inv.re*.

F.INV()



F.INV()

Syntax F.INV(*Wahrsch*; *Freiheitsgrade1*; *Freiheitsgrade2*)

Definition Die Funktion F.INV() gibt Quantile der linksseitigen F-Verteilung zurück.

Argumente *Wahrsch* (erforderlich) ist die zur F-Verteilung gehörige Wahrscheinlichkeit.

Freiheitsgrade1 (erforderlich) ist die Anzahl der Freiheitsgrade im Zähler.

Freiheitsgrade2 (erforderlich) ist die Anzahl der Freiheitsgrade im Nenner.

Hinweis Ist eines der Argumente kein numerischer Ausdruck, gibt F.INV() den Fehlerwert #WERT! zurück.

Ist *Wahrsch* < 0 oder *Wahrsch* > 1, gibt F.INV() den Fehlerwert #ZAH! zurück.

Ist *Freiheitsgrade1* oder *Freiheitsgrade2* keine ganze Zahl, wird der Dezimalanteil abgeschnitten.

Ist *Freiheitsgrade1* < 1 oder *Freiheitsgrade2* < 1, gibt F.INV() den Fehlerwert #ZAH! zurück.

Hintergrund Betrachtet die zuvor beschriebene Funktion F.INV.RE() die Quantile der rechtsseitigen F-Verteilung, beschreibt die Funktion F.INV() die linksseitigen Quantile. Mehr Informationen zur ANOVA-Analyse und der F-Verteilung lesen Sie unter F.INV.RE() / FINV().

Praxiseinsatz Vergleichen Sie hierzu das Beispiel der Funktion F.INV.RE() auf Seite 379.

Siehe auch F.TEST(), F.INV.RE()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *F.inv*.

FISHER()



FISHER()

FISHER(x)

Die Funktion FISHER() gibt die Fisher-Transformation für x zurück. Diese Transformation erzeugt eine Funktion, die normalverteilt ist und somit eine Schiefe von ungefähr Null besitzt. Mit dieser Funktion können Sie eine Hypothese bezüglich des Korrelationskoeffizienten prüfen. x (erforderlich) ist ein numerischer Wert, für den Sie die Transformation durchführen möchten.

Ist x nicht numerisch, gibt die Funktion FISHER() den Fehlerwert *#WERT!* zurück.

Ist x kleiner oder gleich -1 bzw. x größer oder gleich 1 , gibt FISHER() den Fehlerwert *#ZAHL!* zurück.

Während die Regression die Art und die Kovarianz die Stärke eines Zusammenhangs zwischen zwei Merkmalen beschreibt, ist die Korrelation ein Ausdruck für deren Enge.

Syntax**Definition****Argumente****Hinweis****Hintergrund**

Kurze Zusammenfassung der Produkt-Moment-Korrelation (Pearson)

- ▶ Sie beschreibt die Enge des linearen Zusammenhangs zweier Variablen
- ▶ Sie ist die standardisierte Kovarianz oder die Kovarianz
- ▶ Sie ändert sich nicht bei linearer Transformation der Werte
- ▶ Der Korrelationskoeffizient r nimmt Werte zwischen -1 und $+1$ an. Er ist also zweiseitig begrenzt.
- ▶ Er wird berechnet für intervallskalierte Variablen
- ▶ Seine Höhe wird beeinflusst durch die Varianz der beiden Merkmale
- ▶ Aus einer Korrelation lässt sich kein kausaler Zusammenhang ableiten

Die Fisher-Transformation, auch Fisher-Z-Transformation genannt, wurde von Sir Ronald Aylmer Fisher (1890 bis 1962), einem der bedeutendsten Genetiker, Evolutionstheoretiker und Statistiker des 20. Jahrhunderts, entwickelt.

Diese Art der Transformation entstand aus dem Gedanken heraus, Korrelationen zu mitteln.

Da der Korrelationskoeffizient jedoch nicht intervallskaliert ist, d.h. der Abstand zwischen $r = .20$ und $r = .40$ nicht der gleiche wie zwischen $r = .40$ und $r = .60$ ist, darf aus verschiedenen Korrelationen kein Mittelwert berechnet werden.

Die Lösung für dieses Problem liefert die Fisher-Transformation, die Korrelationen in intervallskalierte Variable umwandelt.

Zur Berechnung des Mittelwerts aus mehreren Korrelationen geht man folgendermaßen vor:

1. Umwandlung aller zu mittelnden Korrelationskoeffizienten in Fisher-Werte
2. Mittelung dieser z -Werte
3. Rücktransformation des gemittelten z -Werts in einen Korrelationskoeffizienten

Den transformierten Wert kann man entweder mit der Funktion FISHER() berechnen, ihn in einer Tabelle ablesen oder durch folgende Formel generieren:

$$z = \frac{1}{2} \ln \left(\frac{1+x}{1-x} \right)$$

Die z -Transformation bringt den Korrelationskoeffizienten in seinem gesamten Wertebereich (von -1 bis $+1$) »annähernd« auf Normalverteilungsform, das heißt, der Korrelationskoeffizient wird durch die Fisher-Transformation asymptotisch normalverteilt, sodass normalverteilungsbasierte Testmethoden angewandt werden können. Hierzu gehören auch Signifikanzberechnungen.

Die Formeln für den Erwartungswert und die Varianz des z -transformierten Korrelationskoeffizienten lauten:

$$E(x) = \frac{1}{2} \ln \left(\frac{1+p}{1-p} \right) \text{ und } \text{Varianz} = \frac{1}{(n-3)}$$

Hinweis Mehr Hintergrundinformationen finden Sie auf folgenden Seiten: Korrelation auf Seite 426, Kovarianz auf Seite 431 und 432, Bestimmtheitsmaß auf Seite 352, Regressionsanalyse auf Seite 492.

Praxiseinsatz Ein in der Softwarebranche tätiges Unternehmen verkauft auf der firmeneigenen Website alle vom Unternehmen angebotenen Produkte. Regelmäßig verschickt das Unternehmen Newsletter, um über Neuheiten zu informieren und bereits vorhandene, aber auch potenzielle Neukunden auf die Website und damit auf das Unternehmen aufmerksam zu machen.

Das Unternehmen besteht seit guten vier Jahren; Gründungsjahr war das Jahr 2005.

Die Marketingabteilung hat es sich nun zur Aufgabe gemacht, die Webseitenzugriffe sowie die Onlinebestellungen der vergangenen vier Jahre zu analysieren, um herauszufinden, inwieweit die Onlinebestellungen von den Webseitenzugriffen abhängen. Diese Aufgabe haben Sie bereits erledigt. Es wurde für jedes Jahr der Korrelationskoeffizient errechnet, um auf den Zusammenhang zwischen den beiden Variablen zu schließen.

Sehen Sie sich dazu die Abbildung 11.27 an.

Nun wollen die Mitarbeiter die durchschnittliche Korrelation der vergangenen vier Jahre berechnen. Da der Korrelationskoeffizient jedoch nicht intervallskaliert ist, darf aus verschiedenen Korrelationen kein Mittelwert berechnet werden – jedenfalls nicht so.

Hierzu ist die Funktion FISHER() notwendig, die den Korrelationskoeffizienten in einen intervallskalierten Wert umwandelt und auf diese Weise die Werte auf Normalverteilung bringt.

Die Marketingabteilung wandelt daher alle Korrelationskoeffizienten in Fisher-Werte um. Für das Argument x der Funktion FISHER() wird der Korrelationskoeffizient eingesetzt.

Das Ergebnis der Berechnung über die Funktion FISHER() sehen Sie in Abbildung 11.28.

	B	C	D	E	F	G	H
9		Unabhängig = x-Wert	Abhängig = y-Wert			Unabhängig = x-Wert	Abhängig = y-Wert
10		Webzugriffe	Bestellungen		Monat	Webzugriffe	Bestellungen
11	Januar 2005	89	-		Januar 2006	236	6
12	Februar 2005	65	4		Februar 2006	1.221	17
13	März 2005	198	65		März 2006	1.563	456
14	April 2005	358	38		April 2006	2.682	544
15	Mai 2005	287	48		Mai 2006	4.569	349
16	Juni 2005	896	25		Juni 2006	6.848	854
17	Juli 2005	965	89		Juli 2006	8.463	427
18	August 2005	735	198		August 2006	10.157	337
19	September 2005	1.398	376		September 2006	11.837	899
20	Oktober 2005	653	234		Oktober 2006	12.987	1.011
21	November 2005	498	76		November 2006	13.739	720
22	Dezember 2005	1.673	456		Dezember 2006	14.376	1.069
23	Korrelationskoeffizient		0,856291674		Korrelationskoeffizient		0,786671098
26							
27		Unabhängig = x-Wert	Abhängig = y-Wert			Unabhängig = x-Wert	Abhängig = y-Wert
28		Webzugriffe	Bestellungen			Webzugriffe	Bestellungen
29	Januar 2007	15.739	1.070		Januar 2008	22.786	1.421
30	Februar 2007	16.123	967		Februar 2008	23.784	1.508
31	März 2007	16.548	1.401		März 2008	24.574	1.876
32	April 2007	17.352	1.076		April 2008	25.111	1.948
33	Mai 2007	17.986	1.563		Mai 2008	25.789	2.094
34	Juni 2007	18.234	1.485		Juni 2008	26.948	2.134
35	Juli 2007	18.769	1.367		Juli 2008	15.635	1.673
36	August 2007	19.736	1.138		August 2008	10.157	337
37	September 2007	20.333	1.352		September 2008	11.837	899
38	Oktober 2007	20.987	1.343		Oktober 2008	12.987	1.011
39	November 2007	21.323	1.430		November 2008	13.739	720
40	Dezember 2007	21.999	1.375		Dezember 2008	14.376	1.069
41	Korrelationskoeffizient		0,44987926		Korrelationskoeffizient		0,90305595

Abbildung 11.27: Mit dem Korrelationskoeffizienten werden die Zusammenhänge aufgezeigt

	B	C	D	E	F	G	H
9		Unabhängig = x-Wert	Abhängig = y-Wert			Unabhängig = x-Wert	Abhängig = y-Wert
10		Webzugriffe	Bestellungen		Monat	Webzugriffe	Bestellungen
11	Januar 2005	89	-		Januar 2006	236	6
12	Februar 2005	65	4		Februar 2006	1.221	17
13	März 2005	198	65		März 2006	1.563	456
14	April 2005	358	38		April 2006	2.682	544
15	Mai 2005	287	48		Mai 2006	4.569	349
16	Juni 2005	896	25		Juni 2006	6.848	854
17	Juli 2005	965	89		Juli 2006	8.463	427
18	August 2005	735	198		August 2006	10.157	337
19	September 2005	1.398	376		September 2006	11.837	899
20	Oktober 2005	653	234		Oktober 2006	12.987	1.011
21	November 2005	498	76		November 2006	13.739	720
22	Dezember 2005	1.673	456		Dezember 2006	14.376	1.069
23	Korrelationskoeffizient		0,856291674		Korrelationskoeffizient		0,786671098
24	z-transformierter Wert		1,2793		z-transformierter Wert		1,0626
25	=FISHER(C23)				=FISHER(G23)		
26							
27		Unabhängig = x-Wert	Abhängig = y-Wert			Unabhängig = x-Wert	Abhängig = y-Wert
28		Webzugriffe	Bestellungen			Webzugriffe	Bestellungen
29	Januar 2007	15.739	1.070		Januar 2008	22.786	1.421
30	Februar 2007	16.123	967		Februar 2008	23.784	1.508
31	März 2007	16.548	1.401		März 2008	24.574	1.876
32	April 2007	17.352	1.076		April 2008	25.111	1.948
33	Mai 2007	17.986	1.563		Mai 2008	25.789	2.094
34	Juni 2007	18.234	1.485		Juni 2008	26.948	2.134
35	Juli 2007	18.769	1.367		Juli 2008	15.635	1.673
36	August 2007	19.736	1.138		August 2008	10.157	337
37	September 2007	20.333	1.352		September 2008	11.837	899
38	Oktober 2007	20.987	1.343		Oktober 2008	12.987	1.011
39	November 2007	21.323	1.430		November 2008	13.739	720
40	Dezember 2007	21.999	1.375		Dezember 2008	14.376	1.069
41	Korrelationskoeffizient		0,44987926		Korrelationskoeffizient		0,90305595
42	z-transformierter Wert		0,4845		z-transformierter Wert		1,4885
43	=FISHER(C41)				=FISHER(G41)		

Abbildung 11.28: Die Berechnung der transformierten Korrelationskoeffizienten mithilfe der Funktion FISHER()

Aus den über die Funktion FISHER() errechneten z-Werten kann nun der Mittelwert gebildet werden.

Mittelwert aus den einzelnen Z-Werten	
Z-Wert 1 aus 2005	1,2793
Z-Wert 1 aus 2006	1,0626
Z-Wert 1 aus 2007	0,4845
Z-Wert 1 aus 2008	1,4885
Mittelwert	1,0788

Abbildung 11.29: Die Berechnung des Mittelwerts aus den Z-Werten

Um den in Abbildung 11.29 dargestellten errechneten Mittelwert aus den z-Werten nun wieder in Form eines Korrelationskoeffizienten auszudrücken, verwenden Sie die Umkehrfunktion der Funktion FISHER() – die Funktion FISHERINV().

Denn erst wenn Sie den soeben errechneten Mittelwert wieder in einen Korrelationskoeffizienten umwandeln, können Sie eine Aussage über den mittleren Zusammenhang zwischen den Webseitenzugriffen und den Onlinebestellungen in den vergangenen vier Jahren treffen.

Mittelwert aus den einzelnen Z-Werten	
Z-Wert 1 aus 2005	1,2793
Z-Wert 1 aus 2006	1,0626
Z-Wert 1 aus 2007	0,4845
Z-Wert 1 aus 2008	1,4885
Mittelwert	1,0788
Korrelationskoeffizient	0,7927
=FISHERINV(C51)	

Abbildung 11.30: Die Umwandlung des Mittelwerts (aus den Z-Werten) in einen Korrelationskoeffizienten

Wie Sie der Abbildung 11.30 entnehmen können, ergibt sich aus dem errechneten Mittelwert ein Korrelationskoeffizient von 0,7927.

Aussagen Der mittlere Korrelationskoeffizient in Höhe von 0,7927 steht für eine positive Korrelation. Denn je näher der Korrelationskoeffizient gegen 1 strebt, desto perfekter der Zusammenhang. Das heißt, dass die Datenreihen *Webseitenzugriffe* und *Bestellungen* linear voneinander abhängig sind.

Anders ausgedrückt bedeutet der Korrelationskoeffizient von 0,7927: Wenn sich die Anzahl der Webseitenzugriffe beispielsweise durch verschiedene Marketingaktionen erhöht, steigt auch die Anzahl der Bestellungen über das Web.

Insgesamt betrachtet kann die Aussage getroffen werden, dass in den vergangenen vier Jahren die Onlinebestellungen immer von den Webseitenzugriffen abhängig waren – Grund genug, um auch weiterhin die Kunden auf die Unternehmens-Webseite zu locken.

Siehe auch FISHERINV(), KORREL(), KOVAR()



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Fisher*.

FISHERINV()



FISHERINV()

FISHERINV(*y*)

Die Funktion FISHERINV() gibt die Umkehrung der Fisher-Transformation zurück. Mithilfe dieser Transformation können Sie die Korrelation zwischen Datenbereichen oder Matrizen untersuchen.

Ist $y = \text{FISHER}(x)$, dann ist $\text{FISHERINV}(y) = x$.

y (erforderlich) ist der Wert, dessen Transformation Sie umkehren möchten.

Ist *y* nicht numerisch, gibt die Funktion FISHERINV() den Fehlerwert #WERT! zurück.

Die Funktion FISHERINV() ist die Umkehrfunktion zu FISHER(). Das heißt, sie liefert die umgekehrte Fisher-Transformation für eine übergebene Zahl.

Mit FISHERINV() berechnen Sie den ursprünglichen Wert *x* des Fisher-transformierten Werts *z*.

Mithilfe der Transformationsmöglichkeit über die Funktion FISHER() können Sie die Korrelation zwischen Datenbereichen oder -arrays untersuchen.

Die Gleichung für die Umkehrung der Fisher-Transformation lautet:

$$x = \frac{e^{2y} - 1}{e^{2y} + 1}$$

Mehr Informationen zur Fisher-Transformation finden Sie auf Seite 383 unter der Funktion FISHER().

Die Funktionsweise der Funktion FISHERINV() entnehmen Sie bitte dem Beispiel unter FISHER() auf Seite 383.

FISHER(), KORREL(), KOVAR()

Das gleiche Beispiel wie unter der Funktion FISHER() finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Fisherinv*.

Syntax

Definition

Argumente

Hinweis

Hintergrund

Hinweis

Praxiseinsatz

Siehe auch



F.TEST() / FTEST()



F.TEST()

F.TEST(*Matrix1*; *Matrix2*)

Die Funktion F.TEST() gibt die Teststatistik eines F-Tests zurück. Ein F-Test berechnet die einseitige Wahrscheinlichkeit, dass sich die Varianzen von *Matrix1* und *Matrix2* nicht signifikant unterscheiden.

Matrix1 (erforderlich) ist die erste Matrix oder der erste Wertebereich.

Matrix2 (erforderlich) ist die zweite Matrix oder der zweite Wertebereich.

Als Argumente müssen entweder Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Enthält eines der Argumente *Matrix1* oder *Matrix2* weniger als zwei Datenpunkte oder ist die Varianz von *Matrix1* oder *Matrix2* gleich 0, gibt F.TEST() den Fehlerwert #DIV/0! zurück.

Syntax

Definition

Argumente

Hinweis

Hintergrund Mit dieser Funktion können Sie feststellen, ob zwei Stichproben unterschiedliche Varianzen haben. Sind beispielsweise die Prüfungsergebnisse öffentlicher und privater Schulen bekannt, können Sie ermitteln, ob bei diesen Schultypen ein unterschiedlich breites Notenspektrum üblich ist. Untersucht wird der Unterschied in der Streuung (oder Vielfalt) der Prüfungsergebnisse.

F.TEST() berechnet also das Signifikanzniveau dafür, dass die Varianzen zweier Stichproben gleich sind – sich nicht signifikant unterscheiden. Die Fragestellung, die mit F.TEST() beantwortet werden soll, lautet demnach: Sind die beiden Varianzen der vorliegenden Stichproben als gleich zu betrachten oder nicht?

Zur Berechnung des Signifikanzniveaus wird der komplette Wertebereich der Stichproben verwendet. Die Werte der ersten Stichprobe ergeben *Matrix1* und die Werte der zweiten Stichprobe *Matrix2*. Anhand von *Matrix1* und *Matrix2* berechnet F.TEST() das Signifikanzniveau, ohne vorher die Stichprobenvarianzen berechnen zu müssen.

Das errechnete Signifikanzniveau stellt einen Zahlenwert, ausgedrückt in Prozent, zwischen 0 und 100 dar. Dieser Wert drückt die »Testsicherheit« bei statistischen Hypothesentests aus. Oftmals liegt die Testsicherheit bei 90 oder 95 %.

Was sagt das Ergebnis von F.TEST() aus?

90 % Signifikanz bei einem Testergebnis würde bedeuten, dass das vorliegende, zufällig zustande gekommene, Zahlenmaterial in nur 100 minus 90 = 10 % aller Fälle eine vergleichbare Ausprägung annehmen würde. Das heißt, man nimmt mit nur 10%iger Wahrscheinlichkeit an, dass das zufällige Datenmaterial eine 90%ige signifikante Ausprägung hat.

Folglich bedeuten 90 % Signifikanz nicht, dass das vorliegende Datenmaterial mit nur 10%iger Wahrscheinlichkeit zufällig ist und dass man mit 90%iger Wahrscheinlichkeit auf einen bestimmten Sachverhalt schließen kann.

Praxiseinsatz Im Rahmen einer medizinischen Studie wurde die Verträglichkeit eines Medikaments untersucht. Dazu liegen Ihnen die Testergebnisse sowie einige Erläuterungen vor.

Eine Gruppe nahm die normale Tagesdosis, die andere Gruppe nahm einmalig zu Beginn eine erhöhte Dosis ein. Das übergeordnete Ziel sollte es sein, zu ermitteln, ob die erhöhte Dosis den Heilungsprozess beschleunigen kann. Dazu wurde die Behandlungsdauer in Tagen ermittelt.

Die Nullhypothese besagt nun, dass es *keinen* Unterschied in beiden Gruppen bezüglich des Behandlungserfolgs gibt. Die Gegenhypothese besagt, dass die Gruppe 2 schneller wieder gesund ist, da die Behandlungsmethode erfolgreicher als die herkömmliche ist.

Ihnen wurde nun die Aufgabe zugetragen, die Testergebnisse zu analysieren und aufgrund Ihrer Auswertungen eine Aussage darüber zu treffen, ob die Nullhypothese angenommen werden kann oder verworfen werden muss.

Da Sie während der Testphase nicht anwesend waren und Ihnen noch einige Hintergrundinformationen fehlen, wollen Sie zunächst mit der Funktion F.TEST() herausfinden, mit wie vielprozentiger Wahrscheinlichkeit die Varianzen der beiden Stichproben gleich sind.

Das heißt, Sie möchten den Unterschied in der Streuung der Testergebnisse untersuchen.

	B	C
9	Gruppe 1 x_1	Gruppe 2 x_2
10	7	8
11	8	4
12	8	8
13	9	5
14	6	6
15	8	9
16	4	5
17	7	6
18	6	3
19	3	8
20		
21	Ergebnis für F.TEST	
22	0,890719801	=F.TEST(B10:B19;C10:C19)

Abbildung 11.31: *F.TEST()* liefert die Signifikanz für die Stichprobenvarianz

Was sagt das in Abbildung 11.31 dargestellte Ergebnis für F.TEST() von 0,89 aus?

Das Ergebnis von 89 % Signifikanz bedeutet, dass das vorliegende, zufällig zustande gekommene Zahlenmaterial in nur 100 minus 90 = 10 % aller Fälle eine vergleichbare Ausprägung annehmen würde. Das heißt, man nimmt mit nur 10%iger Wahrscheinlichkeit an, dass das zufällige Datenmaterial eine 90%ige signifikante Ausprägung hat.

In anderen Worten heißt das, die Berechnung der Signifikanz mithilfe von F.TEST() lässt darauf schließen, dass sich die Varianzen der beiden Stichproben mit 89%iger Sicherheit nicht unterscheiden.

Berechnet man die jeweilige Varianz der beiden Stichproben, wird dieses Ergebnis untermauert.

	E	F	G	H
9	Berechnete/gegebene Werte	Wert		
10	Beob. Varianz X_1^2 =	3,60	=VARIANZ(B10:B19)	
11	Beob. Varianz X_2^2 =	3,96	=VARIANZ(C10:C19)	

Abbildung 11.32: Die berechnete Varianz für *Matrix1* und *Matrix2*

Wie Sie der Abbildung 11.32 entnehmen können, unterscheiden sich die Varianzen der beiden Stichproben nur geringfügig. Damit ist das von F.TEST() errechnete Ergebnis mit einer Signifikanz von 89 % bestätigt.

F.VERT.RE(), F.INV.RE()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *F.test*.

Siehe auch



F.VERT.RE() / FVERT()



F.DIST.RT() / FDIST()

Die Funktion FVERT() wird seit Microsoft Excel 2010 durch die Funktion F.VERT.RE() ersetzt und zusätzlich durch die Funktion F.VERT() ergänzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht. Um die Abwärtskompatibilität von F.VERT.RE() zu sichern, ist die Funktion FVERT() weiter unter ihrem alten Namen verfügbar.

Hinweis

F.VERT.RE(*x*;Freiheitsgrade1;Freiheitsgrade2)

Syntax

Definition Die Funktion F.VERT.RE() gibt Werte der Verteilungsfunktion (1-Alpha) einer rechtsseitig F-verteilten Zufallsvariablen zurück. Mit dieser Funktion können Sie feststellen, ob zwei Datenmengen unterschiedlichen Streuungen unterliegen. Beispielsweise können Sie die Umfrageergebnisse dreier gleich großer Mitarbeitergruppen untersuchen und ermitteln, ob sich die Streuung bzw. die Ergebnisse zwischen den einzelnen Gruppen unterscheiden.

Argumente x (erforderlich) ist der Wert, für den die Funktion ausgewertet werden soll.

Freiheitsgrade1 (erforderlich) ist die Anzahl der Freiheitsgrade im Zähler.

Freiheitsgrade2 (erforderlich) ist die Anzahl der Freiheitsgrade im Nenner.

Hinweis Ist eines der Argumente nicht numerisch, gibt F.VERT.RE() den Fehlerwert #WERT! zurück.

Ist x negativ, gibt F.VERT.RE() den Fehlerwert #ZAHL! zurück.

Ist *Freiheitsgrade1* oder *Freiheitsgrade2* keine ganze Zahl, wird der Dezimalanteil abgeschnitten.

Ist *Freiheitsgrade1* kleiner 1 bzw. *Freiheitsgrade2* größer oder gleich 10^{10} , gibt F.VERT.RE() den Fehlerwert #ZAHL! zurück.

Ist *Freiheitsgrade2* kleiner 1 bzw. *Freiheitsgrade2* größer oder gleich 10^{10} , gibt F.VERT.RE() den Fehlerwert #ZAHL! zurück.

F.VERT.RE() wird durch $F.VERT.RE = P(F > x)$ berechnet, wobei F eine F-verteilte Zufallsvariable ist und die Verteilung die Freiheitsgrade *Freiheitsgrade1* und *Freiheitsgrade2* hat.

Hintergrund Die Funktion F.VERT.RE() liefert das einem Wert zugrunde liegende Signifikanzniveau. Bei der Funktion F.INV.RE() haben wir gesagt, dass man auf Basis einer Wahrscheinlichkeit und der Freiheitsgrade für verschiedene Datenmengen den kritischen Wert F berechnen kann. Das Argument *Wahrsch* wurde hier als gegeben angenommen.

Hinweis Mehr Hintergrundinformationen zur Funktion F.INV.RE() finden Sie auf Seite 379.

Mit der Funktion F.VERT.RE() können Sie die Wahrscheinlichkeit, also das Signifikanzniveau für den durch F.INV.RE() errechneten kritischen Wert generieren.

Praxiseinsatz Bleiben wir bei dem unter F.INV.RE() angewandten Beispiel.

Hier wurde eine Umfrage unter 15 zufällig ausgewählten Mitarbeitern durchgeführt, die jeweils zehn Fragen mit drei Antwortmöglichkeiten beantworten mussten.

Gruppe I	Gruppe II	Gruppe III		
4	2	2		
2	1	1		
1	4	6		
5	7	4		
2	5	7		
ZUSAMMENFASSUNG				
Gruppen	Anzahl	Summe	Mittelwert	Varianz
Gruppe I	5	14	2,8	2,7
Gruppe II	5	19	3,8	5,7
Gruppe III	5	20	4,0	6,5

Abbildung 11.33: Die Zusammenfassung der Umfrageergebnisse

Die Nullhypothese besagt, dass es keinen Unterschied zwischen den drei Gruppen gibt. Die Alternativhypothese vermutet einen Unterschied. Die einfache Varianzanalyse lieferte folgende Ergebnisse:

ANOVA	Quadratsummen (SS)	Freiheitsgrade (df)	Mittlere Quadratsumme (MS)	Prüfgröße (F)	P-Wert	kritischer F-Wert
Unterschiede						
- zwischen den Gruppen	4,13	2	2,07	0,416107	0,67	3,89 =F.INV.RE(0,05,D24,D25)
- innerhalb der Gruppen	59,60	12	4,97			
Gesamt	63,73	14				

Abbildung 11.34: Die Ergebnisse der einfaktoriellen Varianzanalyse

Mithilfe der Funktion F.INV.RE() und einem zugrunde gelegten Signifikanzniveau (Wahrscheinlichkeit) von 0,05 wurde ein kritischer Wert F_{krit} von 3,89 berechnet (siehe Abbildung 11.34).

Die Funktion F.VERT.RE() liefert Ihnen nun das Signifikanzniveau, das dem kritischen Wert zugrunde liegt – also die unter der Funktion F.INV.RE() angegebene Wahrscheinlichkeit von 0,05.

Folgende Größen sind in unserem Beispiel zur Berechnung von F.VERT.RE() gegeben:

GESUCHT = Signifikanzniveau	?
Freiheitsgrade 1 (zwischen den Gruppen)	2
Freiheitsgrade 2 (innerhalb der Gruppen)	12
F_{krit}	3,89

Abbildung 11.35: Die gegebenen Größen zur Berechnung des Signifikanzniveaus

Die Berechnung des Signifikanzniveaus für F_{krit} ergibt das in Abbildung 11.36 dargestellte Ergebnis.

	F	G	H	I	J
8	GESUCHT = Signifikanzniveau		?		
9	Freiheitsgrade 1 (zwischen den Gruppen)		2		
10	Freiheitsgrade 2 (innerhalb der Gruppen)		12		
11	F_{krit}		3,89		
12	Prüfgröße F		0,41		
13	FVERT für F_{krit} (Signifikanzniveau)		0,05	=F.VERT.RE(H11;H9;H10)	
14	FVERT für Prüfgröße F		0,67	=F.VERT.RE(H12;H9;H10)	

Abbildung 11.36: Die Berechnung des Signifikanzniveaus mithilfe von F.VERT.RE()

Entsprechend liefert die Funktion F.VERT.RE() für die Prüfgröße $F = 0,4161$ (Abbildung 11.37) eine Wahrscheinlichkeit von 0,67, also 67 %.

	F	G	H
8	GESUCHT = Signifikanzniveau		?
9	Freiheitsgrade 1 (zwischen den Gruppen)		2
10	Freiheitsgrade 2 (innerhalb der Gruppen)		12
11	F_{krit}		3,89
12	Prüfgröße F		0,41
13	FVERT für F_{krit} (Signifikanzniveau)		0,05
14	FVERT für Prüfgröße F		0,67

Abbildung 11.37: F.VERT.RE() liefert das Signifikanzniveau für die Prüfgröße F

Da der Signifikanzwert a größer ist als die Prüfgröße F , kann die Nullhypothese bestätigt werden. Das heißt, man kann davon ausgehen, dass es keinen signifikanten Unterschied zwischen den Gruppen gibt.

Siehe auch F.TEST(), F.INV.RE()



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* (Excel 2007/2010) auf dem Arbeitsblatt *F.vert.re*.

F.VERT() F.DIST()

Syntax F.VERT(x ;Freiheitsgrade1;Freiheitsgrade2;kumuliert)

Definition F.VERT() gibt Werte der Verteilungsfunktion (1-Alpha) einer (linkssseitigen) F-verteilter Zufallsvariablen zurück. Mit dieser Funktion können Sie feststellen, ob zwei Datenmengen unterschiedlichen Streuungen unterliegen.

Beispielsweise können Sie die Punktzahlen untersuchen, die Männer und Frauen bei einem Einstellungstest erzielt haben, und ermitteln, ob sich die für die Frauen gefundene Streuung von derjenigen der Männer unterscheidet.

Argumente x (erforderlich) ist der Wert, für den die Funktion ausgewertet werden soll.

Freiheitsgrade1 (erforderlich) ist die Anzahl der Freiheitsgrade im Zähler.

Freiheitsgrade2 (erforderlich) ist die Anzahl der Freiheitsgrade im Nenner.

kumuliert (erforderlich) gibt den Wahrheitswert zurück, der den Typ der Funktion bestimmt. Ist *kumuliert* mit WAHR belegt, gibt F.VERT() den Wert der Verteilungsfunktion zurück. Ist *kumuliert* mit FALSCH belegt, gibt F.VERT() den Wert der Dichtefunktion zurück.

Hinweis Ist eines der Argumente kein numerischer Ausdruck, gibt F.VERT() den Fehlerwert #WERT! zurück.

Ist x negativ, gibt F.VERT() den Fehlerwert #ZAHL! zurück.

Ist *Freiheitsgrade1* oder *Freiheitsgrade2* keine ganze Zahl, wird der Dezimalanteil abgeschnitten.

Ist *Freiheitsgrade1* < 1 , gibt F.VERT() den Fehlerwert #ZAHL! zurück.

Ist *Freiheitsgrade2* < 1 , gibt F.VERT() den Fehlerwert #ZAHL! zurück.

Hintergrund Betrachtet die zuvor beschriebene Funktion F.VERT.RE() die Quantile der rechtsseitigen F-Verteilung, beschreibt die Funktion F.VERT() die linksseitigen Quantile.

Praxiseinsatz Vergleichen Sie hierzu das Beispiel der Funktion F.VERT.RE() auf Seite 389.

Siehe auch F.TEST(), F.VERT.RE()



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* (Excel 2007/2010) auf dem Arbeitsblatt *F.vert*.

GAMMA GAMMA()

GAMMA(Zahl)

Mit der Funktion GAMMA() ermitteln Sie den Wert der Gammafunktion für eine bestimmte Zahl.

Zahl (erforderlich) definiert den Wert, für den Sie die Gammafunktion berechnen möchten.

Ist *Zahl* eine negative ganze Zahl oder 0, gibt die Funktion GAMMA() den Fehlerwert #ZAH! zurück.

Enthält *Zahl* ungültige Zeichen, gibt GAMMA() den Fehlerwert #WERT! zurück.

Die Gammaverteilung ist eine kontinuierliche Wahrscheinlichkeitsverteilung über der Menge der positiven reellen Zahlen. Sie ist definiert durch die Wahrscheinlichkeitsdichte bei x größer 0.

Carl Friedrich Gauß war es, der die Darstellung der Gammafunktion um negative Zahlen erweiterte. Er wies nach, dass ein Grenzwert für alle komplexen Zahlen außer $-1, -2, -3$ usw. existiert und die Funktionalgleichung $\Gamma(x) = (x-1)!$ erfüllt.

Die Gaußsche Definition der Gammafunktion lautet also:

$$\Gamma(x) = \lim_{n \rightarrow \infty} \frac{n! n^x}{x(x+1)\dots(x+n)}$$

Wobei Folgendes gilt: $x \in \mathbb{R} \setminus \{0, -1, -2, \dots\}$

Die Einschränkung der Definitionsmenge ergibt sich, da der Nenner des Bruchs bei negativen ganzzahligen x -Werten 0 ergeben würde. Die Gammafunktion hat an diesen Stellen also Polstellen.

Die Gammafunktion, die nach Gauß nun also auch negative Werte einschließt, stellt sich wie folgt dar:

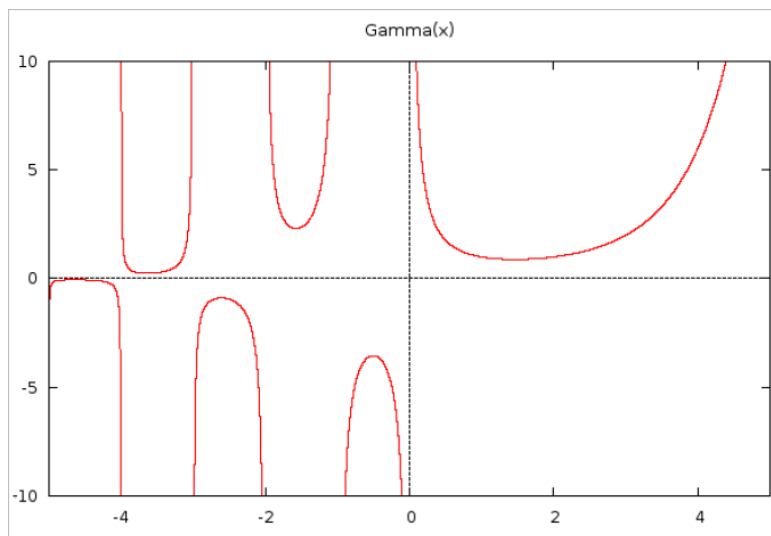


Abbildung 11.38: Die Gammafunktion für positive und negative, nicht ganzzahlige Werte

Neu in Excel
2013

Syntax

Definition

Argumente

Hinweis

Hintergrund

Mit der Funktion `GAMMA()` können Sie beliebige Werte der Gammafunktion schnell und einfach berechnen – ohne umfangreiche Formeln.

Die Funktion `GAMMA()` ist eng mit den Funktionen `GAMMA.INV()` und deren Umkehrfunktion `GAMMA.VERT()` verknüpft, die zur Kontrolle einer Gammaverteilung verwendet werden.

Hinweis Mehr Informationen zur Gammaverteilung und den damit verbundenen Funktionen `GAMMA.INV()` und `GAMMA.VERT()` finden Sie in den gleichnamigen Abschnitten.

Praxiseinsatz Der Berechnung von Werten der Gammafunktion über die Funktion `GAMMA()` ist sehr einfach. Gehen Sie folgendermaßen vor:

Gegeben sind verschiedene positive und negative Werte für das Argument *Zahl*:

- ▶ Positive Werte: 1 bis 5
- ▶ Negative Werte: –1,5 bis –3,75 (Achtung: Keine ganzen Zahlen zulässig)

Die Berechnung von `GAMMA()` sehen Sie in Abbildung 11.39.

x für G	G-Wert
1	1,0000
1,5	0,8862
2	1,0000
2,5	1,3293
3	2,0000
3,5	3,3234
4	6,0000
4,5	11,6317
5	24,0000
-1,5	2,3633
-2,5	-0,9453
-3,5	0,2701
-3,75	0,2679

Abbildung 11.39: Berechnung von `GAMMA()`

Unter Angabe der in Abbildung 11.39 dargestellten Daten für das Argument *Zahl* gibt die Funktion `GAMMA()` die verschiedenen Werte für die Gammaverteilung zurück.

Siehe auch `GAMMA.INV()`; `GAMMA.VERT()`



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Gamma*.

GAMMA.INV() / GAMMAINV()

`GAMMA.INV()` / `GAMMAINV()`

Syntax `GAMMA.INV(Wahrsch;Alpha;Beta)`

Definition Die Funktion `GAMMA.INV()` gibt Quantile der Gammaverteilung zurück. Gilt $p = \text{GAMMA.VERT}(x; \dots)$, dann gilt $\text{GAMMA.INV}(p; \dots) = x$.

Mit dieser Funktion können Sie eine Variable untersuchen, deren Verteilung eventuell schief ist.

Wahrsch (erforderlich) ist die zur Gammaverteilung gehörige Wahrscheinlichkeit

Argumente

Alpha (erforderlich) ist ein Parameter der Verteilung

Beta (erforderlich) ist ein Parameter der Verteilung. Wenn $Beta = 1$, gibt GAMMA.INV() die Standard-Gammaverteilung zurück.

Ist eines der Argumente kein numerischer Ausdruck, gibt die Funktion GAMMA.INV() den Fehlerwert #WERT! zurück.

Hinweis

Ist *Wahrsch* kleiner 0 bzw. *Wahrsch* größer 1, gibt GAMMA.INV() den Fehlerwert #ZAH! zurück.

Ist *Alpha* kleiner oder gleich 0 oder *Beta* kleiner oder gleich 0, gibt GAMMA.INV() den Fehlerwert #ZAH! zurück.

Bei einem Wert für *Wahrsch* sucht GAMMA.INV() den Wert x so, dass $GAMMA.VERT(x;Alpha;Beta;WAHR) = Wahrsch$ gilt.

Daher hängt die Genauigkeit von GAMMA.INV() von der Genauigkeit von der Funktion GAMMA.VERT() ab. GAMMA.INV() geht bei der Suche iterativ vor. Hat die Suche nach 100 Iterationsschritten noch nicht konvergiert, gibt die Funktion den Fehlerwert #NV zurück.

Die Funktion GAMMA.INV() ist die Umkehrfunktion von GAMMA.VERT() und kann zur Kontrolle einer Gammaverteilung verwendet werden. Die Argumente *Alpha* und *Beta* entsprechen den Werten aus der Funktion GAMMA.VERT().

Hintergrund

Die Wahrscheinlichkeit ist ein beliebiger Wert zwischen 0 und 100 %.

GAMMA.INV() berechnet die Stelle x auf der horizontalen Achse, die dem kumulierten Flächenanteil der Gammaverteilung entspricht.

Mehr Informationen zur Gammaverteilung und der Funktion GAMMA.VERT() finden Sie auf Seite 398 unter GAMMA.VERT().

Hinweis

Gehen Sie folgendermaßen zur Berechnung von GAMMA.INV() vor:

Praxiseinsatz

Gegeben sind die Werte:

- ▶ 0,068094 = die zur Gammaverteilung gehörende Wahrscheinlichkeit
- ▶ 9 = Parameter *Alpha* für die Verteilung
- ▶ 2 = Parameter *Beta* für die Verteilung

Die Berechnung von GAMMA.INV() sehen Sie in Abbildung 11.40:

	B	C	D	E	F
8	Bedeutung			Parameter	
9	Die zur Gammaverteilung gehörende Wahrscheinlichkeit			0,068094	
10	Parameter Alpha für die Verteilung			9	
11	Parameter Beta für die Verteilung			2	
12	GAMMAINV()			10	
13				=GAMMA.INV(E9;E10;E11)	

Abbildung 11.40: Berechnung von GAMMA.INV()

Unter Angabe der in Abbildung 11.40 dargestellten Parameter gibt die Funktion GAMMA.INV() das Quantil 10 der angegebenen Gammaverteilung zurück.

Siehe auch GAMMA(), GAMMA.VERT()



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Gamma.inv*.

Ebenso finden Sie hier eine Tabelle, bei der Sie die Möglichkeit haben, die Parameter *Alpha* und *Beta* selbst zu bestimmen. Die Veränderung der Dichte- und Verteilungsfunktion wird durch die Eingabe der Parameter aufgezeigt.

GAMMALN() GAMMALN()

Syntax GAMMALN(*x*)

Definition Die Funktion GAMMALN() gibt den natürlichen Logarithmus der Gammafunktion $\Gamma(x)$ zurück.

Argumente *x* (erforderlich) ist der Wert, für den GAMMALN() berechnet werden soll.

Hinweis Ist *x* kein numerischer Ausdruck, gibt die Funktion GAMMALN() den Fehlerwert #WERT! zurück.

Ist *x* kleiner oder gleich 0, gibt GAMMALN() den Fehlerwert #ZAHL! zurück.

Ein Potenzieren der Zahl *e* mit GAMMALN(*i*), wobei *i* eine ganze Zahl ist, führt zu demselben Ergebnis wie $(i - 1)!$.

Hintergrund Wie bereits in der Definition erwähnt, gibt die Funktion GAMMALN() den natürlichen Logarithmus der Gammafunktion zurück.

Hinweis Mehr Informationen zur Gammaverteilung finden Sie auf Seite 397 unter GAMMA.VERT().

Definition von Logarithmus

Beim Begriff Logarithmus handelt es sich um eine Zahl *x*, wie oft eine Zahl *b* (die Basis) potenziert werden muss, um eine Zahl *a* zu erhalten. Man unterscheidet folgende Logarithmen:

- ▶ Natürlicher Logarithmus auf der Basis *e* (Eulersche Zahl = 2,72)
- ▶ Zehner-Logarithmus mit Basis 10
- ▶ Zweier-Logarithmus mit Basis 2

Der Logarithmus (zur Basis *b*) einer Zahl *y* ist also diejenige Zahl *x*, mit der man die Basis *b* potenzieren muss, um die Zahl *y* zu erhalten. Der Logarithmus ist also ein Exponent. Deshalb ist die Logarithmusfunktion auch als Umkehrfunktion der Exponentialfunktion definiert.

GAMMALN() wird wie folgt berechnet:

$$\text{GAMMALN} = \text{LN}(\Gamma(x))$$

wobei

$$\Gamma(x) = \int_0^{\infty} e^{-u} u^{x-1} du$$

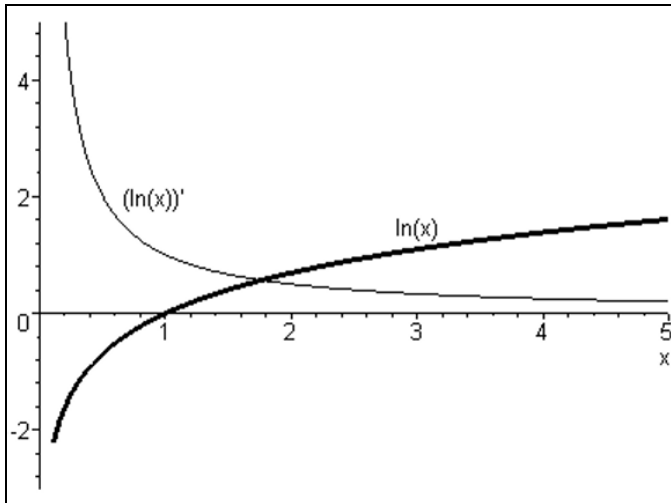


Abbildung 11.41: Der Graph der Logarithmusfunktion

Gehen Sie folgendermaßen zur Berechnung von GAMMALN() vor:
Gegeben ist der Wert 4, für den GAMMALN() berechnet werden soll.
Die Berechnung von GAMMALN() sehen Sie in Abbildung 11.42:

Praxiseinsatz

	B	C	D	E	F
8	Berechnung von GAMMALN()				
9	Bedeutung			Parameter	
10	Wert, für den GAMMALN berechnet werden soll			4	
11	GAMMALN()			1,7918	
12	=GAMMALN(E10)				

Abbildung 11.42: Berechnung von GAMMALN()

Unter Angabe des in Abbildung 11.42 dargestellten Parameters gibt die Funktion GAMMALN() den natürlichen Logarithmus – 1,7918 – der Gammafunktion zurück.

FAKULTÄT()

Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Gammaln*.

Siehe auch



GAMMALN.GENAU()



GAMMALN.PRECISE()

GAMMALN.GENAU(*x*)

Die Funktion GAMMALN.GENAU() gibt den natürlichen Logarithmus der Gammafunktion $\Gamma(x)$ zurück – berechnet auf 15 Stellen.

x (erforderlich) ist der Wert, für den GAMMALN.GENAU() berechnet werden soll.

Syntax

Definition

Argumente

Hinweis Ist x kein numerischer Ausdruck, gibt GAMMALN.GENAU() den Fehlerwert #WERT! zurück.

Ist $x \leq 0$, gibt GAMMALN.GENAU() den Fehlerwert #ZAHL! zurück.

Ein Potenzieren der Zahl e mit GAMMALN.GENAU(i), wobei i eine ganze Zahl ist, führt zu demselben Ergebnis wie $(i - 1)!$.

Hintergrund Lesen Sie hierzu die Hintergrundinformationen der Funktion GAMMALN() auf Seite 396.

Hinweis Mehr Informationen zur Gammaverteilung finden Sie auf Seite 397 unter GAMMA.VERT().

Praxiseinsatz Mehr Informationen zum Praxiseinsatz der Funktion GAMMA.GENAU() finden Sie analog zum Beispiel der Funktion GAMMALN() auf Seite 396.

Siehe auch FAKULTÄT(); GAMMA.VERT(); GAMMALN()



Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Gammaln.genau*.

GAMMA.VERT() /GAMMVERT()

GAMMA.DIST() / GAMMADIST()

Syntax GAMMA.VERT(x ;Alpha;Beta;kumuliert)

Definition Die Funktion GAMMA.VERT() gibt Wahrscheinlichkeiten einer gammaverteilten Zufallsvariablen zurück. Mit dieser Funktion können Sie Variablen untersuchen, die eine schiefe Verteilung besitzen. Die Gammaverteilung wird häufig bei Warteschlangenanalysen verwendet.

Argumente x (erforderlich) ist der Wert (Quantil), dessen Wahrscheinlichkeit $(1-\text{Alpha})$ Sie berechnen wollen.

Alpha (erforderlich) ist ein Parameter der Verteilung.

Beta (erforderlich) ist ein Parameter der Verteilung. Wenn $Beta = 1$, gibt GAMMA.VERT() die Standard-Gammaverteilung zurück.

kumuliert (erforderlich) ist der Wahrheitswert, der den Typ der Funktion bestimmt. Ist *kumuliert* mit WAHR belegt, berechnet GAMMA.VERT() den Wert der Verteilungsfunktion, also die Wahrscheinlichkeit, dass die Anzahl zufällig auftretender Ereignisse zwischen 0 und einschließlich x liegt. Ist *kumuliert* mit FALSCH belegt, liefert GAMMA.VERT() den Wert der Dichtefunktion.

Hinweis Ist x , *Alpha* oder *Beta* kein numerischer Ausdruck, gibt die Funktion GAMMA.VERT() den Fehlerwert #WERT! zurück.

Ist x kleiner 0, gibt GAMMA.VERT() den Fehlerwert #ZAHL! zurück.

Ist *Alpha* kleiner oder gleich 0 bzw. *Beta* kleiner oder gleich 0, gibt GAMMA.VERT() den Fehlerwert #ZAHL! zurück.

Die Dichtefunktion der standardisierten Gammaverteilung lautet:

$$f(x, \alpha) = \frac{x^{\alpha-1}e^{-x}}{\Gamma(\alpha)}$$

Ist $Alpha = 1$, gibt GAMMA.VERT() die Exponentialverteilung mit Folgendem zurück:

Ist n eine positive ganze Zahl, gibt GAMMA.VERT() unter der Voraussetzung, dass $Alpha = n/2$, $Beta = 2$ und $kumuliert = WAHR$ ist, dasselbe Ergebnis wie $(1 - CHIVERT(x))$ mit n Freiheitsgraden zurück.

Ist $Alpha$ eine positive ganze Zahl, wird GAMMA.VERT() auch als »Erlang-Verteilung« bezeichnet.

Die Gammaverteilung ist eine kontinuierliche Wahrscheinlichkeitsverteilung über der Menge der positiven reellen Zahlen. Sie ist definiert durch die Wahrscheinlichkeitsdichte bei x größer 0 .

Hintergrund

$$\Gamma(x) = \int_0^{\infty} e^{-u} u^{x-1} du$$

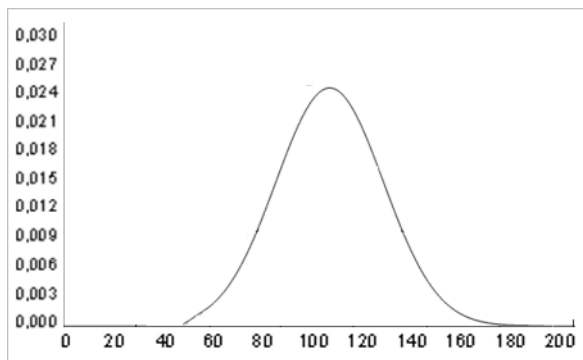


Abbildung 11.43: Die Darstellung der Gamma-Dichtefunktion (25,4)

Für andere Werte von x wird sie durch $f(x)=0$ fortgesetzt. Sie besitzt die Parameter p und q ($= Alpha$ und $Beta$). Um ihre Normierbarkeit zu garantieren, wird $p, q > 0$ gefordert.

Der Vorfaktor $bp/\Gamma(p)$ dient der korrekten Normierung. Der Ausdruck $\Gamma(p)$ steht für die Gammafunktion, nach der die Verteilung auch benannt ist.

Erwartungswert und Varianz der Gammaverteilung sind

$$E(x) = \frac{p}{b} \text{ und } V(x) = \frac{p}{b^2}$$

Die Gammaverteilung ist reproduktiv.

Die Reproduktivität einer Wahrscheinlichkeitsverteilung besagt, dass die Summe von unabhängigen Zufallsvariablen eines bestimmten Verteilungstyps wieder nach diesem Typ verteilt ist.

Tip

Als reproduktiv gelten zum Beispiel die Normalverteilung, die Poisson-Verteilung, die Gammaverteilung und die Cauchy-Verteilung.

Die Summe aus den stochastisch unabhängigen gammaverteilten Zufallsvariablen X und Y , die beide gammaverteilt sind mit den Parametern b und p_x bzw. p_y , ist wiederum gammaverteilt mit den Parametern b und $p_x + p_y$.

Die Gammaverteilung bildet eine sogenannte Familie für einige theoretische Verteilungsfunktionen:

- ▶ Die Chi-Quadrat-Verteilung mit k Freiheitsgraden ist eine Gammaverteilung mit den Parametern $p = k/2$ und $b = 1/2$
- ▶ Die Exponentialverteilung mit dem Parameter λ ist eine Gammaverteilung mit den Parametern $p = 1$ und $b = \lambda$. Dementsprechend entspricht die Erlang-Verteilung mit dem Parameter λ und n Freiheitsgraden einer Gammaverteilung mit den Parametern $p = n$ und $b = \lambda$.
- ▶ Der Quotient $X/(X + Y)$ aus den stochastisch unabhängigen Zufallsvariablen X und Y , die beide gammaverteilt sind mit den Parametern b und p_x bzw. p_y ist betaverteilt mit den Parametern p_x und p_y .

Alternativ zur obigen, im deutschsprachigen Raum üblichen Parametrisierung mit p und b findet man auch häufig die Folgende:

$$\alpha = a, \beta = \frac{1}{b}$$

Dichte und Momente ändern sich dabei dementsprechend (der Erwartungswert wäre hier beispielsweise $\alpha\beta$). Da diese Parametrisierung vor allem im angelsächsischen Raum vorherrscht, wird sie besonders häufig in der Fachliteratur verwendet. Um Missverständnissen vorzubeugen, wird empfohlen, die Momente explizit anzugeben, also beispielsweise von einer Gammaverteilung mit Erwartungswert ab und Varianz ab^2 zu sprechen. Hieraus sind dann Parametrisierung und die entsprechenden Parameterwerte eindeutig rekonstruierbar.

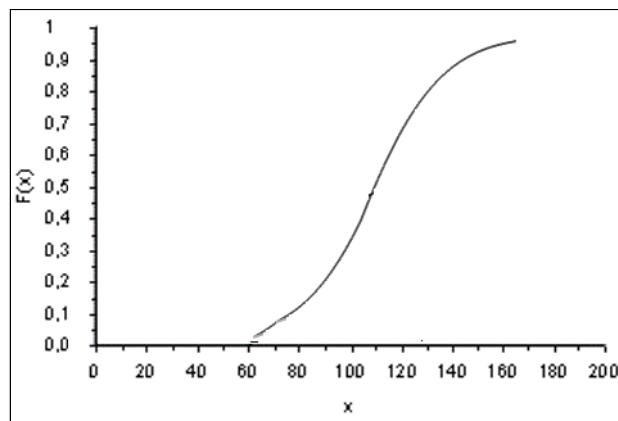


Abbildung 11.44: Die grafische Darstellung der Gamma-Verteilungsfunktion (25,4)

Die Funktion `GAMMA.VERT()` ist eine mathematische Verteilungsfunktion mit zwei Parametern (Formparameter *Alpha* und Skalenparameter *Beta*), die auf der Gamma-Funktion aufbaut.

`GAMMA.VERT()` ist die Umkehrfunktion von `GAMMA.INV`.

Gehen Sie folgendermaßen zur Berechnung von `GAMMA.VERT()` vor:

Gegeben sind die Werte:

- ▶ 10 = Wert (Quantil), dessen Wahrscheinlichkeit berechnet werden soll
- ▶ 2,70 = *Alpha*-Parameter der Verteilung
- ▶ 2 = *Beta*-Parameter der Verteilung

Die Berechnung von `GAMMA.VERT()` sehen Sie in Abbildung 11.45.

	B	C	D	E	F
7	Berechnung von GAMMA.VERT()				
8	Bedeutung			Parameter	
9	x= Wert (Quantil), dessen Wahrscheinlichkeit berechnen werden soll			10	
10	Parameter Alpha für die Verteilung			2,70	
11	Parameter Beta für die Verteilung			2	
12	Kumuliert = Wahr			WAHR	
13	Kumuliert = Falsch			FALSCH	
14	GAMMA.VERT() (für kumuliert = WAHR)			0,90679	
15				=GAMMA.VERT(E9;E10;E11;E12)	
16	GAMMA.VERT() (für kumuliert = FALSCH)			0,03364	
17				=GAMMA.VERT(E9;E10;E11;E13)	

Abbildung 11.45: Berechnung von `GAMMA.VERT()`

Unter Angabe der in Abbildung 11.45 dargestellten Parameter liefert die Funktion `GAMMA.VERT()` folgende Ergebnisse:

- ▶ 0,90679 = Wahrscheinlichkeit der Gammaverteilung bei den oben genannten Daten und dem Parameter *kumuliert* = *WAHR*
- ▶ 0,03364 = Wahrscheinlichkeit der Gammaverteilung bei den oben genannten Daten und dem Parameter *kumuliert* = *FALSCH*

`GAMMA()`, `CHIQU.VERT.RE()`, `EXPON.VERT()`, `GAMMA.INV()`

Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Gamma.vert*.

Ebenso finden Sie hier eine Tabelle, bei der Sie die Möglichkeit haben, selber die Parameter *Alpha* und *Beta* zu bestimmen. Die Veränderung der Dichte- und Verteilungsfunktion wird durch die Eingabe der Parameter aufgezeigt.

Siehe auch



GAUSS()



`GAUSS(z)`

Die Funktion `GAUSS()` berechnet die Wahrscheinlichkeit, mit der eine standardnormalverteilte Zufallsvariable zwischen dem Mittelwert der Grundgesamtheit und z Standardabweichungen vom Mittelwert liegt.

z (erforderlich) definiert die Anzahl der Standardabweichungen, mit der sie vom Mittelwert entfernt liegen.

Neu in Excel
2013

Syntax

Definition

Argumente

Hinweis Ist z keine gültige Zahl, gibt die Funktion `GAUSS()` den Fehlerwert `#ZAHLE!` zurück.

Ist z kein gültiger Datentyp, gibt `GAUSS()` den Fehlerwert `#WERT!` zurück.

Weil `NORM.S.VERT(0;Wahr)` immer den Wert 0,5 zurückgibt, ist `GAUSS(z)` immer um 0,5 kleiner als `NORM.S.VERT(z;Wahr)`.

Hintergrund Die Normalverteilung (nach Carl Friedrich Gauß) ist der wichtigste Typ kontinuierlicher Wahrscheinlichkeitsverteilungen, die eine Aussage darüber liefert, mit welcher Wahrscheinlichkeit eine Zufallsvariable X einen bestimmten Wert annimmt. Ihre Wahrscheinlichkeitsdichte wird auch Gauß-Funktion, Gauß-Kurve, Gauß-Glocke oder Glockenkurve genannt.

Da die Gaußsche Normalverteilung die Schiefe Null hat und somit immer symmetrisch zum Mittelwert ist, ist die Schiefe ein geeignetes Werkzeug, um eine beliebige Verteilung mit der Normalverteilung zu vergleichen.

Gauß arbeitete damit eine Theorie der Beobachtungsfehler aus, die aufs Engste mit der Normalverteilung, der Streuung und der Methode der kleinsten Quadrate verknüpft ist. Er konnte feststellen, dass unzählige Einzeleinflüsse dazu beitragen, mehr oder minder große Abweichungen vom Durchschnitt hervorzurufen.

Hinweis Mehr Informationen zur Gaußschen Normalverteilung und deren charakteristische Eigenschaften lesen Sie in den Abschnitten `G.TEST()` auf Seite 406, `NORM.VERT()` auf Seite 468 und `NORM.S.VERT()` auf Seite 466.

Praxeinsatz In der Praxis heißt das nun, dass Sie bei Eingabe des Werts für eine bestimmte Anzahl an Abweichungen vom Mittelwert die Wahrscheinlichkeit der Verteilung erhalten.

Um Ihnen die Funktion `GAUSS()` zu verdeutlichen, bleiben wir bei dem Beispiel des Glühbirnenherstellers, das wir bereits bei verschiedenen Funktionen – wie auch der Funktion `NORM.S.VERT()` – aufgeführt haben.

Sie sind Hersteller von Glühbirnen, die Sie europaweit vertreiben, und sind dabei, die Leistung der Glühbirnen zu analysieren. Eine Aufstellung der Messungen haben Sie bereits in Excel gemacht.

Auch die mittlere Lebenserwartung sowie die dazugehörige Standardabweichung haben Sie bereits errechnet. Die standardisierten Werte der Verteilung haben Sie mithilfe der Funktion `STANDARDISIERUNG()` errechnet.

Nun möchten Sie die Wahrscheinlichkeit, mit der eine standardnormalverteilte Variable zwischen dem Mittelwert und z Standardabweichungen vom Mittelwert liegt, zurückgeben. Dies erledigen Sie mithilfe der Funktion `GAUSS()`.

Gegeben ist

► $z = 2$

Somit müssen die Fragestellung in unserem Glühbirnenbeispiel sowie das Ergebnis wie in Abbildung 11.46 dargestellt lauten.

	B	C	D	E
7	Wie hoch ist die Wahrscheinlichkeit, dass eine Glühbirne zwischen 2000 Stunden (Mittelwert) und 2 Standardabweichungen vom Mittelwert entfernt brennt?			
8				
9	Informationen für die Berechnung	Beispiel	Ergebnis	
10	Der zu prüfende Wert z	2	0,477249868	=GAUSS(C10)
11	Mittlere Brenndauer einer Glühbirne in h	2000	0,977249868	=NORM.S.VERT(C10;WAHR)
12	Wert der Standardabweichung in h	579		Info: Weil NORM.S.VERT(0;WAHR) immer den Wert 0,5 zurück gibt, ist GAUSS(z) immer um 0,5 kleiner
13	Doppelte Standardabweichung in h	1158		

Abbildung 11.46: Berechnen Sie die Wahrscheinlichkeit mithilfe von GAUSS()

Welche Aussage kann nun getroffen werden?

- ▶ Die Wahrscheinlichkeit, dass eine Glühbirne zwischen 2.000 und 3.158 Stunden brennt, beträgt 48 %

G.TEST(), NORM.INV(), NORM.VERT(), NORM.S.INV(), NORM.S.VERT(), STANDARDISIERUNG()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Gauss*.

Aussagen

Siehe auch



GEOMITTEL() GEOMEAN()

GEOMITTEL(*Zahl1*; *Zahl2*; ...)

Die Funktion GEOMITTEL() gibt das geometrische Mittel einer Menge positiver Zahlen zurück. Zum Beispiel können Sie mit GEOMITTEL() eine mittlere Wachstumsrate berechnen, wenn für einen Zinsszins variable Zinssätze gegeben sind.

Das geometrische Mittel wird errechnet als n-te Wurzel aus dem Produkt aller Werte, wobei n die Anzahl der Werte ist.

Zahl1 (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 Argumente (30 bis Excel 2003), deren geometrisches Mittel berechnet werden soll. Anstelle der durch Semikola voneinander getrennten Argumente können Sie auch eine einzelne Matrix oder einen Bezug auf eine Matrix angeben.

Als Argumente müssen entweder Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Ist eine der Zahlen kleiner oder gleich 0, gibt GEOMITTEL den Fehlerwert #ZÄHL! zurück.

Die Gleichung für das geometrische Mittel lautet:

$$GM_y = \sqrt[n]{y_1 y_2 y_3 \dots y_n}$$

Das geometrische Mittel ist für positive Zahlen anwendbar und findet in erster Linie dort Anwendung, wo Wachstums- oder Minderungsprozesse betrachtet werden, bei denen der Zuwachs (oder die Minderung) nicht um einen festen Betrag stattfindet, sondern um einen Prozentsatz, der natürlich schwanken kann.

Syntax

Definition

Argumente

Hinweis

Hintergrund

So können Sie beispielsweise aus der Umsatztabelle die Umsatzentwicklung über einen Zeitraum betrachten, werden aber mit der Verwendung des arithmetischen Mittels nur ungenügende Ergebnisse erzielen. Hier ist das geometrische Mittel die bessere Wahl.

Praxiseinsatz Da der Leiter der Controllingabteilung des Softwareunternehmens betriebswirtschaftlich immer auf dem Laufenden ist, will er sehen, ob die Umsätze überhaupt steigen. Und wenn ja, will er wissen, um wie viel Prozent im Durchschnitt.

Hierzu verwendet er die Formel GEOMITTEL(). Würde er die Formel für das arithmetische Mittel einsetzen, erhielte er garantiert ein falsches Ergebnis.

Umsatz im Jahr 2008		Veränderung zum jeweiligen Vormonat	
Monat	Betrag		
Januar 08	13.332.315 €	X	
Februar 08	11.038.333 €	0,827938209	
März 08	7.031.695 €	0,637025083	
April 08	12.335.681 €	1,754296937	
Mai 08	4.476.739 €	0,362909757	
Juni 08	9.609.594 €	2,146561146	
Juli 08	7.136.415 €	0,742634392	
August 08	12.626.557 €	1,769313724	
September 08	15.515.165 €	1,228772420	
Oktober 08	8.466.178 €	0,545671155	
November 08	15.991.765 €	1,888900163	=GEOMITTEL(D13:D23) 0,967157
Dezember 08	9.233.586 €	0,577396304	=MITTELWERT(D13:D23) 1,134674

Abbildung 11.47: Die Errechnung des geometrischen Mittels aus den Umsatzdaten

Der Controllingleiter dividiert zunächst die jeweiligen Monatsergebnisse durch die Vormonats-ergebnisse und erhält damit einen Wachstumsfaktor, der definiert, wie groß der Monatsumsatz im Verhältnis zum Vormonat ist.

Anschließend berechnet er aus diesen Ergebnissen das geometrische Mittel und erhält damit einen rechnerischen Mittelwert, der aussagt, wie viel Prozent des Vormonatsumsatzes, über den gesamten Zeitraum betrachtet, erreicht wurden. Er bekommt also die mittlere Wachstumsrate bzw. die mittlere Veränderung des Umsatzes als Ergebnis.

Rechnen Sie hier das arithmetische Mittel und das geometrische Mittel aus, wie in Abbildung 11.47 dargestellt, kommen Sie zu zwei Werten, die auf den ersten Blick gar nicht so verschieden sind. Der Unterschied wird erst dann deutlich, wenn Sie versuchen, aus diesen Werten eine allgemeine Aussage abzuleiten.

Laut geometrischem Mittel wurden seit Januar 2008 durchschnittlich 97 % des Vormonatsumsatzes erreicht, laut arithmetischem Mittel um 113 %.

Das bedeutet, dass im Mittel kein Wachstum erreicht wurde, sondern ein Rückgang um 1 minus 0,967157 stattgefunden hat.

Siehe auch GESTUZTMITTEL(), HARMITTEL(), MEDIAN(), MITTELWERT(), MODALWERT()



Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Geomittel*.

GESTUTZTMITTEL()



TRIMMEAN()

GESTUTZTMITTEL(*Matrix*; *Prozent*)

Die Funktion GESTUTZTMITTEL() gibt den Mittelwert einer Datengruppe zurück, ohne die Randwerte zu berücksichtigen. GESTUTZTMITTEL() berechnet den Mittelwert einer Teilmenge der Datenpunkte, die darauf basiert, dass entsprechend des jeweils angegebenen Prozentsatzes die kleinsten und größten Werte der ursprünglichen Datenpunkte ausgeschlossen werden. Diese Funktion können Sie immer dann verwenden, wenn bei der Auswertung keine Daten berücksichtigt werden sollen, die als Ausreißer anzusehen sind.

Matrix (erforderlich) ist eine Matrix oder Gruppe von Werten, die ohne ihre Ausreißer gemittelt wird.

Prozent (erforderlich) ist der Prozentsatz der Datenpunkte, die nicht in die Bewertung eingehen sollen. Ist beispielsweise **Prozent** = 0,2, wird eine Datenmenge von 20 Punkten um 4 Punkte (20 x 0,2) verringert, und zwar um die zwei größten sowie die zwei kleinsten Werte der Datenmenge.

Ist **Prozent** kleiner 0 bzw. ist **Prozent** größer 1, gibt GESTUTZTMITTEL() den Fehlerwert #ZAHL! zurück.

Die Funktion GESTUTZTMITTEL() rundet die Anzahl der nicht berücksichtigten Datenpunkte auf das kleinste Vielfache von 2 ab. Ist **Prozent** = 0,1 (10 Prozent), so ergeben sich bei 30 Datenpunkten drei auszuschließende Punkte. Aus Symmetriegründen lässt GESTUTZTMITTEL() in diesem Fall den kleinsten und den größten Wert der Datenmenge (also nur zwei Werte) unberücksichtigt.

Üblicherweise werden für die Berechnung eines Mittelwerts alle Werte einer Datenreihe herangezogen. Es ist allerdings auch denkbar, dass Sie die Randbereiche ausblenden wollen. Etwa um ein Mittel ohne Ausreißer berechnen zu können. Sie erhalten damit ein gestutztes Mittel.

Kann man also davon ausgehen, dass die zu untersuchenden Daten Ausreißer, d.h. einige wenige zu hohe oder zu niedrige Werte enthalten, sortiert man die Beobachtungswerte nach aufsteigender Größe, schneidet eine gleiche Anzahl von Werten am Anfang und am Ende der Folge ab und berechnet von den übrig gebliebenen Werten den Mittelwert.

Ein 10%iges gestutztes Mittel erhält man, wenn man 5 % der Gesamtzahl aller Werte am unteren und 5 % am oberen Ende auslässt.

Der Leiter der Controllingabteilung des Softwareunternehmens hat die Umsätze der letzten 17 Monate, von Januar 2007 bis Mai 2008, zusammengestellt. Er möchte den durchschnittlichen Umsatz berechnen, jedoch Ausreißer nach oben oder unten nicht in die Berechnung mit einbeziehen. Deshalb sortiert er zunächst die Umsatzwerte der Größe nach.

Würde er in diesem Fall die Funktion MITTELWERT() benutzen, erhielte er ein nicht aussagefähiges Ergebnis, da die Ausreißer mit in die Berechnung einfließen und das Ergebnis negativ beeinflussen würden. Die Funktion MEDIAN() wäre hier zwar schon besser, aber ein 100%iges Ergebnis würde er auch hiermit nicht bekommen.

Deshalb bedient er sich der Funktion GESTUTZTMITTEL().

Syntax

Definition

Argumente

Hinweis

Hintergrund

Praxiseinsatz

	B	C	D	E	F
9	Umsatz nach Monaten sortiert			Umsatz nach Größe sortiert	
10	Januar 2007	171.487 €		Februar 2008	102.345 €
11	Februar 2007	190.806 €		Juni 2007	107.712 €
12	März 2007	126.931 €		April 2007	111.853 €
13	April 2007	111.853 €		März 2007	126.931 €
14	Mai 2007	133.793 €		Mai 2007	133.793 €
15	Juni 2007	107.712 €		Januar 2008	143.497 €
16	Juli 2007	165.805 €		April 2008	147.718 €
17	August 2007	187.659 €		März 2008	152.849 €
18	September 2007	177.890 €		Mai 2008	155.177 €
19	Oktober 2007	178.205 €		Juli 2007	165.805 €
20	November 2007	167.984 €		November 2007	167.984 €
21	Dezember 2007	192.030 €		Januar 2007	171.487 €
22	Januar 2008	143.497 €		September 2007	177.890 €
23	Februar 2008	102.345 €		Oktober 2007	178.205 €
24	März 2008	152.849 €		August 2007	187.659 €
25	April 2008	147.718 €		Februar 2007	190.806 €
26	Mai 2008	155.177 €		Dezember 2007	192.030 €
27					
28	=GESTUTZTMITTEL(F10:F26;0,10)			154.624,40 €	
29	=MITTELWERT(F12:F24)			155.449,85 €	

Abbildung 11.48: Die Berechnung des gestutzten Mittels aus den Umsätzen der vergangenen 17 Monate

Für das Argument *Prozent* gibt er, wie in Abbildung 11.48 dargestellt, 0,1 an. Das heißt, er möchte sowohl von oben als auch von unten 5 % der Werte aus den Urdaten weglassen bzw. nicht mit in die Berechnung mit einbeziehen. Für das gestutzte Mittel ergibt sich auf diese Weise ein durchschnittlicher Umsatz von 154.624,40 €.

Wie Sie in Abbildung 11.48 sehen können, ergibt die Berechnung des durchschnittlichen Umsatzes einen höheren Wert. Diese Tatsache ist bedingt durch die Ausreißer.

Siehe auch GEOMITTEL(), HARMITTEL(), MEDIAN(), MITTELWERT(), MODALWERT()



Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Mittelwert.xlsx* (Excel 2007/2010) auf dem Arbeitsblatt *Gestutztmittel*.

G.TEST() / GTEST()

Z.TEST() / ZTEST()

Syntax G.TEST(Array;μ0;Sigma)

Definition Die Funktion G.TEST() gibt den einseitigen Wahrscheinlichkeitswert für einen Gaußtest (Normalverteilung) zurück. Für einen Erwartungswert einer Zufallsvariablen, μ_0 , gibt G.TEST() die Wahrscheinlichkeit zurück, mit der der Stichprobenmittelwert größer als der Durchschnitt der für diese Datenmenge (Array) durchgeführten Beobachtungen – also dem beobachteten Stichprobenmittel – ist.

Argumente *Array* (erforderlich) ist das Array oder der Datenbereich, gegen das/den Sie μ_0 testen möchten. μ_0 (erforderlich) ist der zu testende Wert.

Sigma (optional) ist die bekannte Standardabweichung der Grundgesamtheit. Fehlt dieses Argument, wird mit der Standardabweichung der jeweiligen Stichprobe gearbeitet.

Ist Matrix leer, gibt die Funktion G.TEST() den Fehlerwert #NV zurück.

Mit Angabe von *Sigma* wird G.TEST() wie folgt berechnet:

$$G.TEST(Matrix, x) = 1 - NORM.VERT\left(\frac{\mu - x}{\sigma \div \sqrt{n}}\right)$$

Oder ohne Angabe von *Sigma*:

$$G.TEST(Array, \mu_0) = 1 - NORM.S.VERT\left(\left(\bar{x} - \mu_0\right) \div \left(s / \sqrt{n}\right)\right)$$

Wobei Folgendes gilt:

- ▶ *x* ist der Stichprobenmittelwert MITTELWERT(*Array*)
- ▶ *s* ist die Standardabweichung der Stichprobe STABW.S(*Array*)
- ▶ *n* ist die Anzahl der Beobachtungen für die Stichprobe ANZAHL(*Array*)

Die Funktion G.TEST() gibt die Wahrscheinlichkeit an, mit der der Stichprobenmittelwert größer als der beobachtete Wert MITTELWERT(*Array*) ist, bei zugrunde liegendem Erwartungswert einer Zufallsvariablen von μ_0 .

Aufgrund der Symmetrie der Normalverteilung gilt, falls MITTELWERT(*Array*) kleiner μ_0 , gibt G.TEST() einen Wert größer als 0,5 zurück.

Die folgende Excel-Formel kann zur Berechnung der *zweiseitigen* Wahrscheinlichkeit verwendet werden, mit der der Stichprobenmittelwert weiter von μ_0 entfernt liegt (in beide Richtungen) als MITTELWERT(*Array*), wobei μ_0 der zugrunde liegende Erwartungswert einer Zufallsvariablen ist:

$$=2 * MIN(GTEST(Array,\mu_0,Sigma), 1 - G.TEST(Array,\mu_0,Sigma))$$

Der Gaußtest, benannt nach dem Braunschweiger Mathematiker Carl Friedrich Gauß (1777 bis 1855), ist ein statistischer Test, der auf die Standardnormalverteilung zurückgreift und zur Untersuchung der Signifikanz eines Werts aus einer normalverteilten Grundgesamtheit genutzt wird, bei der Erwartungswert und Standardabweichung bekannt sein müssen.

Nähere Informationen zur Normalverteilung und der Funktion NORM.VERT() finden Sie auf Seite 468 dieses Buchs.

Führen Sie einen Gaußtest durch, beantworten Sie die Frage »Mit welcher Wahrscheinlichkeit stammt eine Stichprobe aus einer bestimmten Grundgesamtheit?«

Die Funktion G.TEST() berechnet daher die relative Lage eines Werts *x* im Vergleich zu einer als normalverteilt angenommenen Wertereihe.

Erhalten Sie als Ergebnis für die Funktion G.TEST() einen Wert von 1,5, bedeutet dies, dass *x* das 1,5-fache einer Standardabweichung vom Mittelwert der Wertereihe entfernt liegt.

Unter Verwendung dieser Funktion können Sie das Maß der Wahrscheinlichkeit ermitteln, mit dem eine bestimmte Beobachtung aus einer bestimmten Grundgesamtheit stammt.

Das Argument *Sigma* der Funktion ist die Standardabweichung der Grundgesamtheit und kann optional verwendet werden. Geben Sie *Sigma* nicht an, wird es aus dem Array, also aus dem Datenbereich, der mindestens aus 30 Werten bestehen sollte, berechnet.

Hinweis

Hintergrund

Hinweis

Praxiseinsatz Gehen Sie folgendermaßen zur Berechnung von G.TEST() vor:
Gegeben sind die Werte:

- ▶ Daten = Datenbereich, gegen den Sie μ_0 testen möchten
- ▶ 4 = Zufallsvariable μ_0 ist der zu testende Wert
- ▶ 6 = Zufallsvariable μ_0 ist der zu testende Wert

Die Berechnung von G.TEST() sehen Sie in Abbildung 11.49.

	B	C	D	E
8	Daten	Zufallsvariable	Ergebnisse für GTEST()	Angabe in Prozent
9	3	4	0,090574197	9,06%
10	6		=G.TEST(B9:B18;C9)	
11	7	6	0,863043389	86,30%
12	8		=G.TEST(B9:B18;C11)	
13	6			
14	5			
15	4			
16	2			
17	1			
18	9			

Abbildung 11.49: Die Berechnung von G.TEST()

Unter Angabe der in Abbildung 11.49 dargestellten Parameter gibt die Funktion G.TEST() den einseitigen Wahrscheinlichkeitswert für einen Gaußtest zurück. Folgende Ergebnisse wurden errechnet:

- ▶ Für einen angenommenen Erwartungswert einer Zufallsvariablen von 4 ergibt sich ein einseitiger Wahrscheinlichkeitswert von 0,09057 = 9,06 %
- ▶ Für einen angenommenen Erwartungswert einer Zufallsvariablen von 6 ergibt sich ein einseitiger Wahrscheinlichkeitswert von 0,86304 = 86,30 %

Siehe auch GAUSS(), KONFIDENZ(), NORM.INV(), NORM.VERT(), STANDARDISIERUNG(), NORM.S.INV(), NORM.S.VERT()



Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *G.test*.

HARMITTEL() HARMEAN()

Syntax HARMITTEL(*Zahl1*; *Zahl2*; ...)

Definition Die Funktion HARMITTEL() gibt das harmonische Mittel einer Datenmenge zurück. Ein harmonisches Mittel ist der Kehrwert eines aus Kehrwerten berechneten arithmetischen Mittels.

Argumente *Zahl1* (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 Argumente (30 bis Excel 2003), deren harmonisches Mittel berechnet werden soll. Anstelle der durch Semikola voneinander getrennten Argumente können Sie auch eine einzelne Matrix oder einen Bezug auf eine Matrix angeben.

Als Argumente müssen entweder Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Hinweis

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Ist eine der Zahlen kleiner oder gleich 0, gibt HARMITTEL() den Fehlerwert #ZAHL! zurück.

In der Statistik existieren verschiedene Mittelwerte, zu denen, neben dem geometrischen, dem arithmetischen und dem quadratischen Mittel auch das harmonische Mittel gehört. An dieser Stelle betrachten wir das harmonische Mittel näher.

Hintergrund

Das harmonische Mittel findet vor allem dann Anwendung, wenn Mittelwerte von Quotienten gebildet werden müssen, und es dient als Lagemaß, wenn die Beobachtungswerte zum einen Verhältniszahlen und zum anderen durch einen Bezug auf eine Einheit definiert sind.

Ein typisches Beispiel ist die Mittelung von Geschwindigkeiten, das heißt die Errechnung des Quotienten von Weg/Zeit, unter der Voraussetzung, dass die Wegstrecken bekannt sind.

Die Gleichung für das harmonische Mittel lautet:

$$\frac{1}{H_y} = \frac{1}{n} \sum \frac{1}{y_i}$$

Das harmonische Mittel einer Zahlenmenge ist immer kleiner als deren geometrisches Mittel, das wiederum immer kleiner ist als das zugehörige arithmetische Mittel.

Wichtig

Um Ihnen die Errechnung und Bedeutung des harmonischen Mittels aufzuzeigen, halten wir uns in diesem Beispiel an die oben bereits erwähnte, typische Darstellung des harmonischen Mittels unter Verwendung der Komponenten Geschwindigkeit und Zeit.

Praxiseinsatz

Ein Radfahrer macht eine Tour durch die Alpen und fährt eine 300 Kilometer lange Strecke. Die Strecke gliedert er in fünf Teilstrecken, in denen er jeweils die Geschwindigkeiten des Fahrrades misst.

Nun möchte der Radfahrer aus den fünf Teildurchschnittsgeschwindigkeiten die Gesamtdurchschnittsgeschwindigkeit errechnen. Mit dem Ergebnis möchte er sich die Frage beantworten, mit welcher konstanten Geschwindigkeit er die Strecke in der gleichen Zeit geschafft hätte.

Teilstrecke Nr.	1	2	3	4	5
Länge in km	60	60	60	60	60
Geschwindigkeit in km/h	20	15	25	23	30
Arithmetisches Mittel in km/h	22,60				
Geometrisches Mittel in km/h	22,02				
Harmonisches Mittel in km/h	21,42				

Abbildung 11.50: Die Berechnung der Gesamtdurchschnittsgeschwindigkeit über die Funktion *HARMITTEL()*

Für den besseren Überblick hat er gleichzeitig auch die ihm bereits bekannten Mittelwerte *Arithmetisches Mittel* und *Geometrisches Mittel* errechnet.

Um herauszubekommen, welche der Mittelwertberechnungen das für ihn sinnvollste Ergebnis liefert, wandelt er die Ergebnisse des arithmetischen, geometrischen und harmonischen Mittels auf Meter/Sekunde um und errechnet anschließend, wie lange er mit dieser Durchschnittsgeschwindigkeit für seine bereits zurückgelegten 300 km gebraucht hätte.

Arithmetisches Mittel in m/s		Geometrisches Mittel in m/s		Harmonisches Mittel in m/s	
22,6 km	1 h	22,02 km	1h	21,42 km	1 h
22600 m	3600 s	22020 m	3600 s	21420 m	3600 s
6,27 m	1 s	6,12 m	1 s	5,95 m	1 s

Arithmetisches Mittel in m/s umgerechnet auf die gefahrenen 300km (in Sekunden)		Geometrisches Mittel in m/s umgerechnet auf die gefahrenen 300km (in Sekunden)		Harmonisches Mittel in m/s umgerechnet auf die gefahrenen 300km (in Sekunden)	
6,27 m	1 s	6,12 m	1 s	5,95 m	1 s
0,00627 km	1 s	0,00612 km	1 s	0,00595 km	1 s
300 km	47846 s	300 km	49019 s	300 km	50420 s

Abbildung 11.51: Die Zeit in Sekunden für 300 km für das arithmetische, geometrische und harmonische Mittel

Mit dieser Berechnung ist auch die Aussage bestätigt, dass das geometrische Mittel kleiner ist als das harmonische Mittel und das arithmetische Mittel kleiner als das geometrische.

Um zu zeigen, dass hier tatsächlich das harmonische Mittel das sinnvollste Ergebnis liefert, treten wir den Beweis an.

Hierfür muss man zunächst die Geschwindigkeit v in m/s für die **tatsächlich möglichen** Kilometer bei geleisteter Geschwindigkeit v für die einzelnen gefahrenen Streckenabschnitte **in einer Stunde** berechnen. Für den ersten Streckenabschnitt würde dies bedeuten (wie Sie der Abbildung 11.52 entnehmen können), dass bei gleich bleibender Geschwindigkeit 20 Kilometer pro Stunde hätten gefahren werden können. Teilt man diese 20 Kilometer nun durch 3.600 Sekunden, erhält man die Geschwindigkeit v .

Die Formel hierfür lautet:

$$v = \frac{s}{t}$$

	Tatsächlich mögliche Strecke in Meter bei gleichbleibender Geschwindigkeit für Teilstrecke 1	Tatsächlich mögliche Strecke in Meter bei gleichbleibender Geschwindigkeit für Teilstrecke 2	Tatsächlich mögliche Strecke in Meter bei gleichbleibender Geschwindigkeit für Teilstrecke 3	Tatsächlich mögliche Strecke in Meter bei gleichbleibender Geschwindigkeit für Teilstrecke 4	Tatsächlich mögliche Strecke in Meter bei gleichbleibender Geschwindigkeit für Teilstrecke 5
	20000	15000	25000	23000	30000
Zeit (1 Stunde) in Sekunden (s)	3600	3600	3600	3600	3600
Geschwindigkeit v in m/s	5,56	4,17	6,94	6,39	8,33

Abbildung 11.52: Berechnung der Geschwindigkeit v in m/s ausgehend von der ursprünglich möglichen Strecke

Anschließend nimmt man das Ergebnis für die Geschwindigkeit in m/s und errechnet mit gleicher Formel, aufgelöst nach t (Zeit), den entsprechenden Zeitaufwand für die einzelnen geplanten Streckenabschnitte in Sekunden. Die oben genannte Formel, aufgelöst nach t , lautet dann:

$$t = \frac{s}{v}$$

Das Ergebnis dieser Berechnung können Sie der Abbildung 11.53 entnehmen.

	Länge in m von Teilstrecke 1	Länge in m von Teilstrecke 2	Länge in m von Teilstrecke 3	Länge in m von Teilstrecke 4	Länge in m von Teilstrecke 5
	60000	60000	60000	60000	60000
Geschwindigkeit v in m/s aus oben aufgestellter Berechnung	5,56	4,17	6,94	6,39	8,33
Zeit t in Sekunden	10791,37	14388,49	8645,53	9389,67	7202,88
Summe aus Zeit t	50417,94				

Abbildung 11.53: Zeitaufwand für die Streckenabschnitte in Sekunden und als Summe

Die Summe der Zeiten in Sekunden für die einzelnen Streckenabschnitte zeigt, dass der Wert annähernd dem des harmonischen Mittels entspricht. Die Differenz von drei Sekunden ergibt sich aus den gerundeten Werten.

Vergleicht man das tatsächliche Ergebnis von 50.417,94 Sekunden mit den oben errechneten Ergebnissen der verschiedenen Mittelwerte, so erkennt man, dass in diesem Beispiel das harmonische Mittel am besten zu gebrauchen ist.

GEOMITTEL(), GESTUTZMITTEL(), MEDIAN(), MITTELWERT(), MODALWERT()

Siehe auch

Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Mittelwert.xlsx* (Excel 2007/2010) auf dem Arbeitsblatt *Harmittel*.



HÄUFIGKEIT() FREQUENCY()

HÄUFIGKEIT(*Daten;Klassen*)

Syntax

Die Formel HÄUFIGKEIT() gibt eine Häufigkeitsverteilung als einspaltige Matrix zurück. Verwenden Sie diese Funktion beispielsweise, um die Umsätze innerhalb eines bestimmten Euro-Bereiches zu zählen. Da HÄUFIGKEIT() eine Matrix zurückgibt, muss die Formel als Matrixformel eingegeben werden.

Definition

Daten (erforderlich) entspricht einer Matrix von einer Wertemenge oder einem Bezug darauf, deren Häufigkeiten Sie zählen möchten. Enthält *Daten* keine Werte (Zahlen), gibt HÄUFIGKEIT() eine mit Nullen belegte Matrix zurück.

Argumente

Klassen (erforderlich) sind die als Matrix oder Bezug auf einen Zellbereich eingegebenen Intervallgrenzen, nach denen Sie die in *Daten* enthaltenen Werte einordnen möchten. Falls *Klassen* keine Werte enthält, gibt HÄUFIGKEIT() die Anzahl der zu *Daten* gehörenden Elemente zurück.

Hintergrund Um eine Reduktion von quantitativen Daten zu erreichen, werden vorhandene Daten in Klassen zusammengefasst, für die dann die Häufigkeit ermittelt wird. Jeder einzelne Fall wird immer in genau einer Klasse gezählt. Hierbei sollten Sie Folgendes bedenken:

- ▶ Zu viele Klassen bringen einen Informationsgewinn, aber wenig Übersicht, bei zu wenigen ist es genau umgekehrt
- ▶ Die Klassenbreite, also der Abstand zwischen den einzelnen Klassengrenzen, muss zwar nicht zwingend gleich groß sein, kann jedoch zum schnelleren Verständnis bezüglich des Sachverhaltes beitragen

Die Funktion HÄUFIGKEIT() führt eine Auszählung numerischer Daten für angegebene Intervalle durch. Das Besondere an der Funktion HÄUFIGKEIT() ist, dass sie im Gegensatz zu ZÄHLENWENN() nicht in jede der Ergebniszellen einzeln einzutragen ist, sondern in alle auf einmal. HÄUFIGKEIT() ist eine sogenannte Matrixformel und ihr Ergebnis von daher eine Matrix. Diese wird in unserem Fall aus vier Zahlen bestehen.

Hinweis Sobald Sie einen Bereich zusammenhängender Zellen markiert haben, in dem die als Ergebnis zurückgegebenen Häufigkeiten angezeigt werden sollen, ist HÄUFIGKEIT() als Matrixformel einzugeben.

HÄUFIGKEIT() ignoriert sowohl leere Zellen als auch Text.

Praxiseinsatz Nehmen wir wieder ein Umsatzbeispiel des Softwareherstellers zur Hand. Ein Außendienstmitarbeiter hat über einen Zeitraum von zwei Jahren jeden Monat seine Umsätze in einer Excel-Tabelle notiert. Sein Vorgesetzter möchte nun wissen, wie viele der Umsätze über 15.000 €, wie viele über 17.000 € und 19.000 € liegen und welche über 19.000 € hinausgehen. Es gibt also vier Klassen.

Wie in Abbildung 11.54 dargestellt, beinhalten die Zellen C3:C26 die zu klassifizierenden Daten. In der Spalte »Klassen« wurden die vier Gruppen angegeben, die die Basis für die Errechnung der Häufigkeit der Daten innerhalb dieser Klassen darstellen.

Wichtig Beachten Sie, dass der Bereich für das Ergebnis der Häufigkeitsverteilung eine Zelle mehr umfassen muss als die Anzahl der Klassen. Die Werte für die Klassen stellen die oberen Grenzwerte für die Auszählung dar, eine Klasse enthält also alle Werte bis einschließlich zur Klassenobergrenze. In der letzten Zeile werden die Werte oberhalb der höchsten Klassengrenze dargestellt. In unserem Beispiel beinhaltet die vierte Zeile also die Klasse über 19.000 €, die in der Datenmenge drei Mal vertreten ist.

Da es sich bei der HÄUFIGKEIT() wie bereits erwähnt um eine Matrixfunktion handelt, werden in der Spalte »Ergebnis« alle vier Ergebniszellen (von F3:F6) auf einmal markiert, um das Ergebnis auch in einer Matrix angezeigt zu bekommen.

Da die HÄUFIGKEIT() auch zu den Arrayfunktionen gehört, muss die Eingabe der Daten mit `[Strg]+[↵]+[↵]` beendet werden.

Tipp In Excel 2007 bis 2013 erreichen Sie die Analyse-Funktionen über die Registerkarte *DATEN* in der Gruppe *Analyse*. Sollten die Funktionen nicht zur Verfügung stehen, müssen Sie zunächst aktiviert werden. Klicken Sie dazu auf *DATEI* (Excel 2010/2013) bzw. auf die *Office*-Schaltfläche (Excel 2007) und wählen Sie den Eintrag *Optionen* (Excel 2010/2013) bzw. *Excel-Optionen* (Excel 2007).

Analyse-Funktionen in 2007-2013

Fragestellung:
Wie häufig ging der Umsatz bis 15000 €, wie oft bis 17000€, bis 19.000€ und wie oft war er größer als 19.000€?

Monat	Umsatz (Daten)
Januar 07	17.500 €
Februar 07	17.867 €
März 07	10.966 €
April 07	25.948 €
Mai 07	12.838 €
Juni 07	14.888 €
Juli 07	14.245 €
August 07	16.292 €
September 07	11.689 €
Oktober 07	12.587 €
November 07	18.560 €
Dezember 07	15.697 €
Januar 08	16.022 €
Februar 08	16.589 €
März 08	12.556 €
April 08	18.681 €
Mai 08	14.643 €
Juni 08	15.698 €
Juli 08	18.172 €
August 08	19.710 €
September 08	17.182 €
Oktober 08	11.075 €
November 08	18.494 €
Dezember 08	21.534 €

Klassen	Ergebnis
15000	9
17000	5
19000	7
	3

=HÄUFIGKEIT(\$C\$3:\$C\$26;E3:E5)

Achtung:
Die Funktion HÄUFIGKEIT() ist eine Arrayfunktion und muss deshalb mit der Tastenkombination Strg+Umschalt+Eingabe abgeschlossen werden.

Abbildung 11.54: Die Datenmenge (Umsätze) wird in Klassen eingeteilt, für die wiederum die Häufigkeit ermittelt werden soll

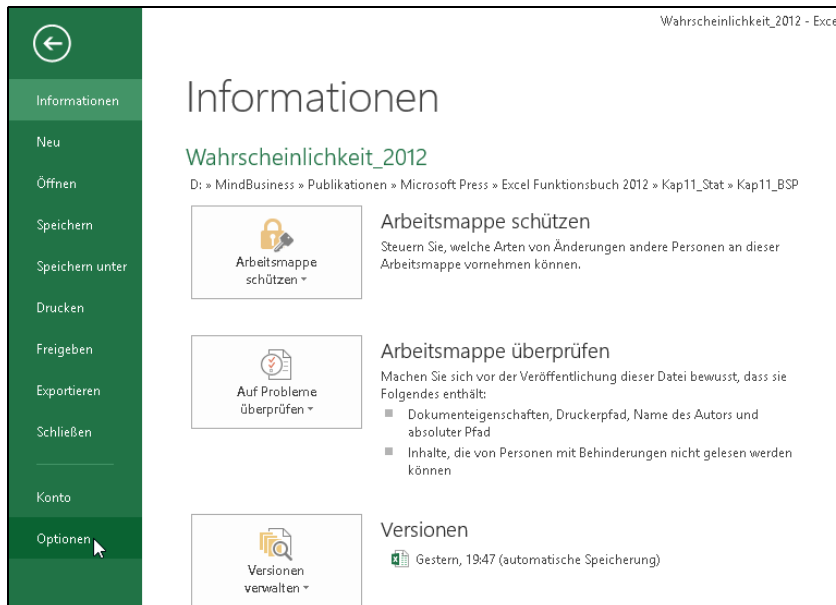


Abbildung 11.55: Der neue Weg zu den Add-Ins

Klicken Sie danach im Dialogfeld *Excel-Optionen* in der Kategorienliste auf *Add-Ins*, wählen Sie im Listenfeld von *Verwalten* den Eintrag *Excel-Add-Ins* und klicken Sie anschließend auf die Schaltfläche *Gehe zu*.

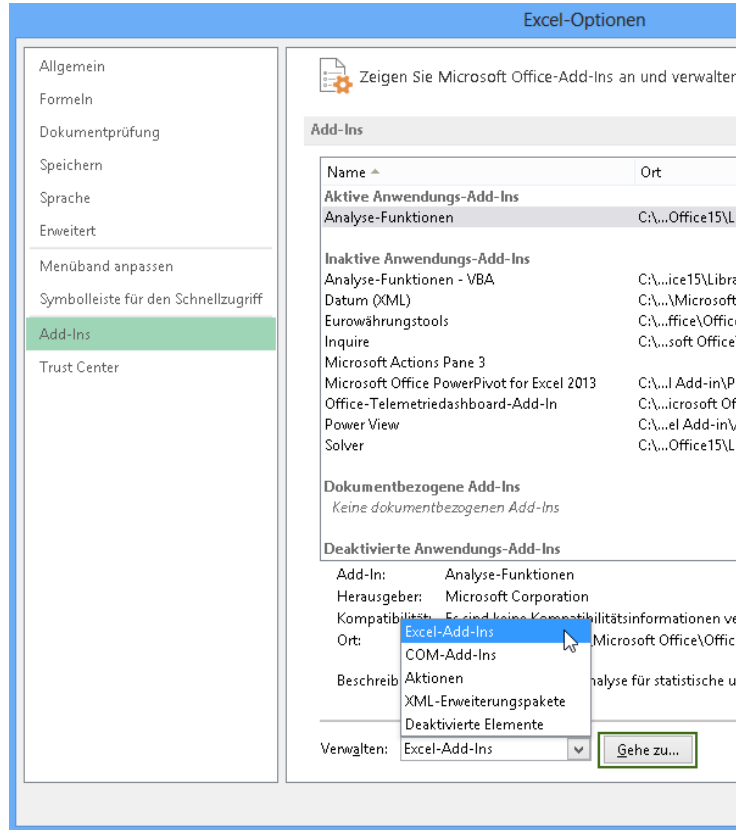


Abbildung 11.56: Der Weg zu den Excel-Add-Ins in 2010/2013

Aktivieren Sie abschließend im Dialogfeld *Add-Ins* das Kontrollkästchen *Analyse-Funktionen* und klicken Sie auf *OK*.

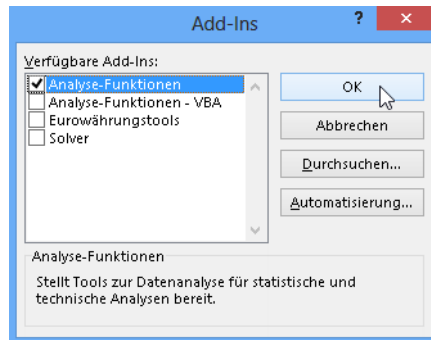


Abbildung 11.57: Aktivieren Sie die Analyse-Funktionen

Die Analyse-Funktionen werden nun installiert und stehen anschließend auf der Registerkarte *DATEN* bereit.

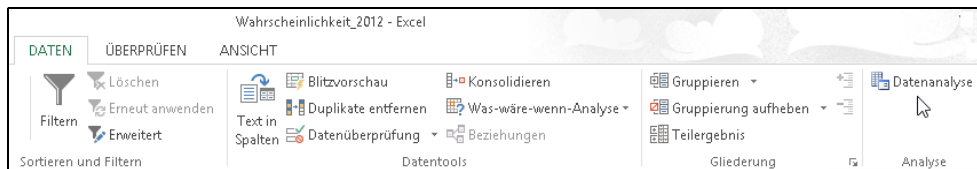


Abbildung 11.58: Die Bereitstellung der Analyse-Funktionen auf der Registerkarte DATEN

ANZAHL(), DBANZAHL()

Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *Häufigkeit*.

Siehe auch



HYPGEOM.VERT() / HYPGEOMVERT()

HYPGEOM.DIST() / HYPGEOMDIST()

HYPGEOM.VERT(*Erfolge_S*; *Umfang_S*; *Erfolge_G*; *Umfang_G*; *Kumuliert*)

Die Funktion HYPGEOM.VERT() gibt Wahrscheinlichkeiten einer hypergeometrisch-verteilten Zufallsvariablen zurück. HYPGEOM.VERT() berechnet die Wahrscheinlichkeit, in einer Stichprobe eine bestimmte Anzahl von Beobachtungen zu erhalten.

Dafür sind die folgenden Angaben erforderlich:

- ▶ Anzahl der in der Stichprobe erzielten Erfolge
- ▶ Umfang der Stichprobe
- ▶ Anzahl der in der Grundgesamtheit möglichen Erfolge
- ▶ Umfang der Grundgesamtheit
- ▶ Wahrheitswert zur Bestimmung der Funktion

Erfolge_S (erforderlich) ist die Anzahl der in der Stichprobe erzielten Erfolge.

Umfang_S (erforderlich) ist der Umfang (Größe) der Stichprobe.

Erfolge_G (erforderlich) ist die Anzahl der in der Grundgesamtheit möglichen Erfolge.

Umfang_G (erforderlich) ist der Umfang (Größe) der Grundgesamtheit.

Kumuliert (erforderlich) ist ein Wahrheitswert, der den Typ der Funktion bestimmt.

Alle Argumente werden durch Abschneiden der Nachkommastellen zu ganzen Zahlen gekürzt.

Ist eines der Argumente kein numerischer Ausdruck, gibt die Funktion HYPGEOM.VERT() den Fehlerwert #WERT! zurück.

Ist *Erfolge_S* kleiner 0 oder *Erfolge_S* größer als der kleinere der Werte von *Umfang_S* bzw. *Erfolge_G*, gibt HYPGEOM.VERT() den Fehlerwert #ZAHL! zurück.

Syntax

Definition

Argumente

Hinweis

Ist $Erfolge_S$ kleiner als der größere Wert von 0 beziehungsweise ($Umfang_S$ minus $Umfang_G$ plus $Erfolge_G$), gibt HYPGEOM.VERT() den Fehlerwert #ZAHL! zurück.

Ist $Umfang_S$ kleiner 0 bzw. $Umfang_S$ größer $Umfang_G$, gibt die Funktion HYPGEOM.VERT() den Fehlerwert #ZAHL! zurück.

Ist $Erfolge_G$ kleiner 0 bzw. $Erfolge_G$ größer $Umfang_G$, gibt die Funktion HYPGEOM.VERT() den Fehlerwert #ZAHL! zurück.

Ist $Umfang_G$ kleiner 0, gibt die Funktion HYPGEOM.VERT() den Fehlerwert #ZAHL! zurück.

Die Funktion HYPGEOM.VERT() wird verwendet, wenn einer endlichen Grundgesamtheit Probestücke entnommen werden, ohne dass diese Probestücke ersetzt werden.

Hintergrund Mithilfe der hypergeometrischen Verteilung können Sie die Frage »Wie groß ist die Wahrscheinlichkeit, in einer Stichprobe genau x Merkmalsträger vorzufinden?« beantworten.

Da aus dem gesamten Umfang, der Grundgesamtheit, eine zufällig gezogene Stichprobe, die nicht mehr zurückgelegt wird, genommen wird, kann hier **nicht** die Binomialverteilung angewendet werden.

Die Gleichung für eine hypergeometrische Verteilung lautet:

$$P(X = x) = h(x, n, M, N) = \frac{\binom{M}{x} \binom{N-M}{n-x}}{\binom{N}{n}}$$

Hierbei gilt:

$x = Erfolge_S$

$n = Umfang_S$

$M = Erfolge_G$

$N = Umfang_G$

Die Funktion HYPGEOM.VERT() können Sie für Problemstellungen einsetzen, bei denen eine begrenzte Grundgesamtheit vorliegt und jede Beobachtung entweder ein Erfolg oder ein Misserfolg sein kann und bei denen jede Teilmenge eines bestimmten Umfangs mit gleicher Wahrscheinlichkeit gewählt wird.

Praxiseinsatz Ein einfaches Beispiel, um Ihnen die Funktionsweise von HYPGEOM.VERT() darzustellen, ist das Spiel mit dem Glück. Denn mithilfe von HYPGEOM.VERT() können Sie die Gewinnchancen beim Lotto »6 aus 49« auf einfache Art und Weise berechnen.

Welche Argumente sind für das Lottobeispiel mit welchen Werten belegt?

- ▶ $Erfolge_S$ ist die Anzahl der in der Stichprobe erzielten Erfolge, das heißt, ein Spiel kann 1, 2, 3, 4, 5 oder 6 Gewinnzahlen aufweisen. Somit ist $Erfolge_S = 6$.
- ▶ $Umfang_S$ ist der Umfang der Stichprobe, also die Anzahl der Kugeln in einer Ziehung. Somit ist $Umfang_S = 6$.

- ▶ *Erfolge_G* ist die Anzahl der in der Grundgesamtheit möglichen Erfolge, das heißt die Anzahl der Gewinnkugeln. Somit ist *Erfolge_G* = 6.
- ▶ *Umfang_G* ist der Umfang der Grundgesamtheit und entspricht demnach 49 Kugeln. Somit ist *Umfang_G* = 49.
- ▶ *Kumuliert* ist falsch und bestimmt damit den Wahrheitswert der Funktion

Die Frage ist nun, wie hoch ist die Wahrscheinlichkeit, **genau** 6 Richtige zu haben? Die Antwort sehen Sie in Abbildung 11.59.

	B	C
9	Erfolge_S	6
10	Umfang_S	6
11	Erfolge_G	6
12	Umfang_G	49
13	Gewinnchance	0,000007%
14		=HYPGEOM.VERT(C9;C10;C11;C12;falsch)

Abbildung 11.59: *HYPGEOM.VERT()* liefert die Wahrscheinlichkeit dafür, genau sechs richtige Zahlen zu tippen

Sie sehen, wie schwindend gering die Wahrscheinlichkeit auf den großen Gewinn ist. Wer sich mit weniger richtigen Treffern und somit mit weniger Gewinnausschüttung begnügt, kann selbstverständlich mit der Funktion *HYPGEOM.VERT()* auch die Wahrscheinlichkeit dafür berechnen, aus den 6 aus 49 gezogenen Zahlen genau fünf, vier, drei, zwei, eine oder gar keine Zahl richtig zu haben.

Das Ergebnis ist in Abbildung 11.60 dargestellt.

	B	C	D	E	F	G	H	I
9	Erfolge_S	6	5	4	3	2	1	0
10	Umfang_S	6	6	6	6	6	6	6
11	Erfolge_G	6	6	6	6	6	6	6
12	Umfang_G	49	49	49	49	49	49	49
13	Gewinnchance	0,000007%	0,0018%	0,10%	1,77%	13,24%	41,30%	43,60%
14		=HYPGEOM.VERT(C9;C10;C11;C12;falsch)						

Abbildung 11.60: Berechnen Sie die Wahrscheinlichkeit für das Zutreffen von Merkmalen

BINOM.VERT(), *BINOM.VERT.BEREICH()*, *FAKULTÄT()*, *KOMBINATIONEN()*, *NEG-BINOMVERT()*, *VARIATIONEN()*; *VARIATIONEN2()*

Siehe auch

Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Hypgeom.vert*.



KGRÖSSTE()



Syntax KGRÖSSTE(*Matrix*;k)

Definition KGRÖSSTE() gibt den k-größten Wert einer Datengruppe zurück. Mit dieser Funktion können Sie eine Zahl auf Basis ihrer relativen Größe ermitteln. Beispielsweise können Sie mit KGRÖSSTE() den erst-, zweit- und dritthöchsten Umsatz einer entsprechenden Tabelle ermitteln.

Argumente *Matrix* (erforderlich) ist die Matrix oder der Datenbereich, deren k-größten Wert Sie bestimmen möchten.

k (erforderlich) ist der Rang des Elements einer Matrix oder eines Zellbereichs, dessen Wert zurückgegeben werden soll.

Hintergrund Aus dem Excel-Basiswissen sind die Formeln MIN() und MAX() bekannt, die aus einem Zellbereich die kleinste bzw. die größte Zahl herausuchen. Für Situationen, in denen man allerdings den zweigrößten oder drittkleinsten Wert benötigt, bedient man sich der Funktionen KGRÖSSTE() bzw. KKLEINSTE().

Auf diese Weise können Sie mit KGRÖSSTE() die Spitzenwerte und mit KKLEINSTE() die niedrigsten Werte eines Tabellenbereichs ermitteln.

Folgendes ist noch zu beachten:

- ▶ Ist *Matrix* leer, gibt KGRÖSSTE() den Fehlerwert #ZAHL! zurück
- ▶ Ist *k* kleiner oder gleich 0, oder ist *k* größer als die Anzahl der Zahlen, gibt KGRÖSSTE() den Fehlerwert #ZAHL! zurück
- ▶ Ist *n* die Anzahl der in einem Bereich abgelegten Zahlen (Datenpunkte), gibt KGRÖSSTE(*Matrix*;1) den größten und KGRÖSSTE(*Matrix*;n) den kleinsten Wert zurück

Praxiseinsatz 1 In einem theoretischen Beispiel zeigen wir Ihnen im Folgenden die Eingabemöglichkeiten in die Formel KGRÖSSTE(*Matrix*;k).

Wenn sich Ihre Werte z.B. im Bereich B1:B100 befinden, liefert KGRÖSSTE(B1:B100;1) durch *k*=1 den höchsten Wert der Matrix.

Hinweis Die Summe aus dem höchsten, dem zweithöchsten und dem dritthöchsten Wert erhalten Sie mit der Formel

=KGRÖSSTE(B1:B100;1)+KGRÖSSTE(B1:B100;2)+KGRÖSSTE(B1:B100;3).

Verwenden Sie eine Arrayformel, können Sie den Ausdruck folgendermaßen verkürzen:

=SUMME(KGRÖSSTE(B1:B100;{1;2;3}))

Wie bei den Arrayformeln üblich, müssen Sie die Eingabe mit der Tastenkombination **Strg**+**↵**+**↵** beenden. Die Formel wird dann automatisch mit einer geschweiften Klammer umgeben.

Obwohl diese Schreibweise kompakter ist als die zuvor genannte mit dem »+«-Operator, wird die Eingabe immer mühseliger, je mehr Spitzenwerte summiert werden sollen, da Sie die Parameter in der geschweiften Klammer selbst eingeben müssen.

Praxiseinsatz 2 Der Softwarehersteller unseres Beispiels möchte aus einer Tabelle, die die Umsätze der vergangenen zwei Jahre beinhaltet, die drei höchsten Umsätze herausfiltern, ohne die Daten vorher zu sortieren.

1	Fragestellung: Wie hoch war der größte, zweitgrößte, drittgrößte Umsatz?				
2	Monat	Umsatz	Berechnung der Spitzenwerte		
3	Januar 07	107.629,00 €	1	465.289 €	Größter Umsatz
4	Februar 07	185.385,00 €	2	400.000 €	Zweitgrößter Umsatz
5	März 07	180.807,00 €	3	354.865 €	Drittgrößter Umsatz
6	April 07	124.328,00 €	=KGRÖSSTE(Matrix;k)		
7	Mai 07	400.000,00 €			
8	Juni 07	185.675,00 €			
9	Juli 07	210.169,00 €			
10	August 07	221.729,00 €			
11	September 07	234.187,00 €			
12	Oktober 07	237.947,00 €			
13	November 07	210.088,00 €			
14	Dezember 07	207.791,00 €			
15	Januar 08	201.097,00 €			
16	Februar 08	222.460,00 €			
17	März 08	208.585,00 €			
18	April 08	206.387,00 €			
19	Mai 08	218.951,00 €			
20	Juni 08	159.658,00 €			
21	Juli 08	285.489,00 €			
22	August 08	465.289,00 €			
23	September 08	157.854,00 €			
24	Oktober 08	354.865,00 €			
25	November 08	311.001,00 €			
26	Dezember 08	146.895,00 €			

Abbildung 11.61: In der Datenreihe der Umsätze werden die drei höchsten Werte gesucht

Mit der Formel KGRÖSSTE() werden die drei höchsten Werte der Tabelle herausgesucht und wiedergegeben.

Nutzen Sie zusätzlich zu KGRÖSSTE() auch die bedingte Formatierung, können Sie, wie in **Abbildung 11.62** dargestellt, die entsprechenden Werte noch farblich hervorheben. **Tipp**

Dafür stehen Ihnen tolle Formatierungsmöglichkeiten zur Verfügung.

Öffnen Sie die Registerkarte *START* und wählen Sie im Dropdownmenü der Schaltfläche *Bedingte Formatierung* die gewünschte Formatierungsoption.

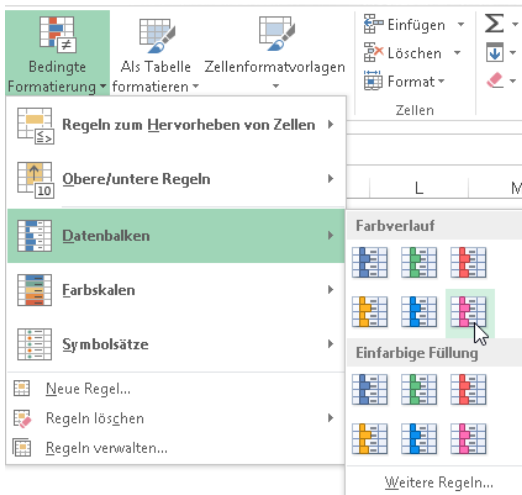


Abbildung 11.62: Wählen Sie zwischen einer Vielzahl an Formatierungsmöglichkeiten

Das Ergebnis zeigt eine eindeutige Formatierung im Bezug auf die Höhe der Werte.

Berechnung der Spitzenwerte		
1	465.289 €	Größter Umsatz
2	400.000 €	Zweitgrößter Umsatz
3	354.865 €	Drittgrößter Umsatz
=KGRÖSSTE(Matrix;k)		

Abbildung 11.63: Schnell und einfach formatieren ab Excel 2007

Siehe auch KKLEINSTE(), QUANTIL(), QUANTILSRANG(), QUARTILE()



Dieses Beispiel und das Beispiel aus dem Tipp finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *KGrösste*.

KKLEINSTE() SMALL()

Syntax KKLEINSTE(*Matrix*;K)

Definition KKLEINSTE() gibt den k-kleinsten Wert einer Datengruppe zurück. Mit dieser Funktion können Sie Werte ermitteln, die innerhalb einer Datenmenge eine bestimmte relative Größe haben.

Argumente *Matrix* (erforderlich) ist eine Matrix oder ein Bereich numerischer Daten, deren k-kleinsten Wert Sie bestimmen möchten.

K (erforderlich) ist der Rang des Elements einer Matrix oder eines Zellbereichs, dessen Wert zurückgegeben werden soll.

Hintergrund Die Formel KKLEINSTE() ist das Gegenstück zu KGRÖSSTE und liefert als Ergebnis den oder die kleinsten Werte eines Tabellenbereichs.

Bei KKLEINSTE() ist Folgendes zu beachten:

- ▶ Ist *Matrix* leer, gibt KKLEINSTE() den Fehlerwert #ZAHL! zurück
- ▶ Ist *K* kleiner oder gleich 0, oder ist *K* größer als die Anzahl der Zahlen (Datenpunkte), gibt KKLEINSTE() den Fehlerwert #ZAHL! zurück
- ▶ Ist *n* die Anzahl der in *Matrix* abgelegten Zahlen (Datenpunkte), gibt KKLEINSTE(*Matrix*;1) den kleinsten und KKLEINSTE(*Matrix*;n) den größten Wert zurück

Praxiseinsatz Der Deutsche Fußball-Bund hat die Besucher der Bundesligaspieltage in einer Excel-Datei erfasst und will nun ermitteln, an welchen Spieltagen die drei niedrigsten Besucherzahlen zu verbuchen sind, ohne die Daten vorher nach Größe zu sortieren und damit die Reihenfolge der Spieltage zu ändern.

Die Abbildung 11.64 zeigt das Ergebnis der Formel KKLEINSTE().

Tipp Wenn Sie nun noch wissen möchten, zu welchen der Spieltage die Ergebnisse von KKLEINSTE() gehören, können Sie entweder wie im Beispiel KGRÖSSTE auf die bedingte Formatierung zurückgreifen, oder mithilfe der Funktion VERGLEICH() die Nummer des Eintrags in der Liste ermitteln. Mit dieser Nummer und der Formel INDEX() generieren Sie anschließend den Spieltag.

	B	C	D	E	F
1	Fragestellung: Was war die kleinste, zweitkleinste, drittkleinste Besucherzahl an den Spieltagen 1-17?				
2	Spieltag	Besucherzahl	Berechnung der kleinsten Besucherzahl		
3	Bundesligaspieltag 1	107.629	1	107.629	
4	Bundesligaspieltag 2	185.385	2	124.328	
5	Bundesligaspieltag 3	180.807	3	146.215	
6	Bundesligaspieltag 4	124.328	=KKLEINSTE(Matrix;k)		
7	Bundesligaspieltag 5	146.215			
8	Bundesligaspieltag 6	185.675			
9	Bundesligaspieltag 7	210.169			
10	Bundesligaspieltag 8	221.729			
11	Bundesligaspieltag 9	234.187			
12	Bundesligaspieltag 10	237.947			
13	Bundesligaspieltag 11	210.088			
14	Bundesligaspieltag 12	207.791			
15	Bundesligaspieltag 13	201.097			
16	Bundesligaspieltag 14	222.460			
17	Bundesligaspieltag 15	208.585			
18	Bundesligaspieltag 16	206.387			
19	Bundesligaspieltag 17	218.951			

Abbildung 11.64: Aus den Besucherzahlen der einzelnen Spieltage werden die kleinsten Werte gefiltert

KGRÖSSTE(), QUANTIL.INKL(), QUANTIL.EXKL(), QUARTILE.INKL(), QUARTILE.EXKL(), QUANTILSRANG.INKL(), QUANTILSRANG.EXKL(); MAX(), MIN(), MEDIAN()

Siehe auch

Dieses Beispiel und das Beispiel aus dem Tipp finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *KKleinste*.



KONFIDENZ.NORM() / KONFIDENZ()

CONFIDENCE.NORM()

Die Funktion KONFIDENZ() wird seit Microsoft Excel 2010 durch die Funktionen KONFIDENZ.NORM() ersetzt und durch die Funktion und KONFIDENZ.T() ergänzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht. Um die Abwärtskompatibilität von KONFIDENZ.NORM() und KONFIDENZ.T() zu sichern, ist die Funktion KONFIDENZ() weiter unter ihrem alten Namen verfügbar.

Hinweis

KONFIDENZ(Alpha;StandardAbweichung;Umfang)

Die Funktion KONFIDENZ.NORM() gibt einen Wert zurück, den Sie zum Erstellen eines Konfidenzintervalls für den Erwartungswert einer Zufallsvariablen verwenden können. Das Konfidenzintervall bezeichnet einen Bereich mit Werten. Der Stichprobenmittelwert x liegt in der Mitte dieses Bereichs und der Bereich ist $x \pm \text{KONFIDENZ.NORM}()$.

Syntax

Definition

Wenn beispielsweise x das Stichprobenmittel für Lieferzeiten von per E-Mail bestellten Produkten ist, bezeichnet $x \pm \text{KONFIDENZ.NORM}()$ einen Bereich aus Erwartungswerten einer Zufallsvariablen. Bei jedem Erwartungswert einer Zufallsvariablen μ_0 in diesem Bereich ist die Wahrscheinlichkeit, ein Stichprobenmittel zu erhalten, das weiter von μ_0 entfernt ist als x , größer als Alpha . Bei jedem Erwartungswert einer Zufallsvariablen μ_0 außerhalb dieses

Bereichs, ist die Wahrscheinlichkeit, ein Stichprobenmittel zu erhalten, das weiter von μ_0 entfernt ist als x , niedriger als *Alpha*.

Angenommen x , *Standardabweichung* und *Umfang* werden zum Erstellen eines zweiseitigen Tests mit Signifikanzniveau *Alpha* für die Hypothese verwendet, dass der Erwartungswert einer Zufallsvariablen μ_0 lautet, dann kann in diesem Fall die Hypothese aufrecht erhalten werden, wenn μ_0 im Konfidenzintervall liegt, und muss verworfen werden, wenn μ_0 nicht im Konfidenzintervall liegt.

Das Konfidenzintervall erlaubt jedoch nicht die Schlussfolgerung, dass das nächste Paket mit einer Wahrscheinlichkeit von 1 minus *Alpha* innerhalb des dem Konfidenzintervall entsprechenden Zeitraums geliefert wird.

Argumente *Alpha* (erforderlich) ist die Irrtumswahrscheinlichkeit bei der Berechnung des Konfidenzintervalls. Das Konfidenzintervall ist gleich $100 \cdot (1 - \textit{Alpha})\%$, was bedeutet, dass ein Wert für *Alpha* von 0,05 einem Konfidenzniveau von 95 % entspricht.

Standardabweichung (erforderlich) ist die als bekannt angenommene Standardabweichung der Grundgesamtheit.

Umfang (erforderlich) ist der Umfang der Stichprobe.

Hinweis Ist eines der Argumente nicht numerisch, gibt KONFIDENZ.NORM() den Fehlerwert #WERT! zurück.

Ist *Alpha* kleiner oder gleich 0 bzw. *Alpha* größer oder gleich 1, gibt KONFIDENZ.NORM() den Fehlerwert #ZAHL! zurück.

Ist *Standardabweichung* kleiner oder gleich 0, gibt KONFIDENZ.NORM() den Fehlerwert #ZAHL! zurück.

Ist *Umfang* keine ganze Zahl, wird der Dezimalanteil abgeschnitten.

Ist *Umfang* kleiner 1, gibt KONFIDENZ.NORM() den Fehlerwert #ZAHL! zurück.

Ist *Alpha* gleich 0,05, muss die Fläche unter der Kurve der standardisierten Normalverteilung berechnet werden, die dem Wert $(1 - \textit{Alpha})$ bzw. 95 % entspricht. Dieser Wert ist $\pm 1,96$. Für das Konfidenzintervall gilt daher:

$$\bar{x} \pm 1,96 \left(\frac{\sigma}{\sqrt{n}} \right)$$

Hintergrund Konfidenzintervalle sind in der Statistik ein gebräuchliches Mittel, um Genauigkeiten von geschätzten Werten anzugeben. So gibt etwa das 95 %-Konfidenzintervall eines Mittelwerts den Bereich an, in dem der wahre Wert, d.h. der erwartete Wert der Grundgesamtheit, mit 95%iger Wahrscheinlichkeit liegt.

Je breiter ein Konfidenzintervall ist, umso ungenauer ist die Information, die es liefert. Ist das Konfidenzintervall dagegen sehr eng, »kennt« man den wahren Wert recht gut.

Aus diesem Grund ist es üblich, neben dem errechneten Durchschnitt (Mittelwert) entweder die Standardabweichung als Maß für die Streuung anzugeben oder das Konfidenzintervall.

Die Funktion KONFIDENZ.NORM() ermöglicht die Bestimmung des 1-*Alpha*-Konfidenzintervalls für den Erwartungswert (= Mittelwert) einer Wahrscheinlichkeitsverteilung.

Alpha ist die Irrtumswahrscheinlichkeit oder das sogenannte Alpha-Risiko. Zulässig sind nur Werte aus dem Intervall $[0..1]$, also von 0 bis 100 %.

Ist *Alpha* beispielsweise 0,05 (also 5 %), so ist die Wahrscheinlichkeit, dass der tatsächliche Mittelwert außerhalb des mithilfe der Funktion KONFIDENZ.NORM() ermittelten Intervalls liegt, 5 %. Anders ausgedrückt, der Mittelwert der Grundgesamtheit liegt zu 95 % in dem errechneten Intervall.

Man spricht hier vom 95 %-Konfidenzintervall.

Zur Berechnung des Konfidenzintervalls liegt eine Stichprobe aus einer normalverteilten Grundgesamtheit mit bekannter Standardabweichung und bekanntem Mittelwert vor. Als Ergebnis liefert die Funktion KONFIDENZ.NORM() die halbe Breite des Vertrauensbereichs des arithmetischen Mittelwerts der Stichprobe.

Das heißt, das 1-Alpha-Konfidenzintervall ist ein symmetrischer Bereich um den Mittelwert einer Stichprobe, das den Mittelwert der Grundgesamtheit mit einer Wahrscheinlichkeit von 1-Alpha enthält.

Wir möchten Ihnen die Funktion KONFIDENZ.NORM() anhand unseres Softwareunternehmens verdeutlichen. Das Unternehmen verkauft auf der firmeneigenen Website alle vom Unternehmen angebotenen Produkte.

Praxiseinsatz

	B	C	D
8		Unabhängig = x-Wert	Abhängig = y-Wert
9		Webzugriffe	Bestellungen
10	Januar 2005	89	keine
11	Februar 2005	65	4
12	März 2005	198	65
13	April 2005	358	38
14	Mai 2005	287	48
15	Juni 2005	896	25
16	Juli 2005	965	89
17	August 2005	735	198
18	September 2005	1.398	376
19	Oktober 2005	653	234
20	November 2005	498	76
21	Dezember 2005	1.673	456
22	Januar 2006	236	6
23	Februar 2006	1.221	17
24	März 2006	1.563	456
25	April 2006	2.682	544
26	Mai 2006	4.569	349
27	Juni 2006	6.848	854
28	Juli 2006	8.463	427
29	August 2006	10.157	337
30	September 2006	11.837	899
31	Oktober 2006	12.987	1.011
32	November 2006	13.739	720
33	Dezember 2006	14.376	1.069
34	Januar 2007	15.739	1.070
35	Februar 2007	16.123	967
36	März 2007	16.548	1.401
37	April 2007	17.352	1.076
38	Mai 2007	17.986	1.563
39	Juni 2007	18.234	1.485
40	Juli 2007	18.769	1.367
41	August 2007	19.736	1.138
42	September 2007	20.333	1.352
43	Oktober 2007	20.987	1.343
44	November 2007	21.323	1.430
45	Dezember 2007	21.999	1.375
46	Januar 2008	22.786	1.421
47	Februar 2008	23.784	1.508
48	März 2008	24.574	1.876
49	April 2008	25.111	1.948
50	Mai 2008	25.789	2.094
51	Juni 2008	26.948	2.134
52	Juli 2008	15.635	1.673

Abbildung 11.65: Die erfassten Zahlen für Webseitenzugriffe und Onlinebestellungen

Das Unternehmen besteht seit zehn Jahren. Obwohl die Webseite und auch die Möglichkeit der Onlinebestellungen bereits von Anfang an zur Verfügung stehen, werden erst seit acht Jahren die Webseitenzugriffe und auch die Anzahl der Onlinebestellungen zahlenmäßig festgehalten.

Der Marketingleiter des Unternehmens hat die Zahlen der letzten vier Jahre vorliegen und möchte nun beide Komponenten näher analysieren. Dafür hat er die Anzahl der Webseitenzugriffe pro Monat und die Onlinebestellungen pro Monat in Excel erfasst.

Auch den Stichprobenmittelwert und die Standardabweichung der Grundgesamtheit hat er bereits für beide Bereiche ermittelt. Da er nur mit einer Stichprobe arbeitet, möchte er nun ein Konfidenzintervall errechnen, in welchem zu 95%iger Sicherheit der Mittelwert der Grundgesamtheit liegt.

	F	G	H
9	Berechnungen/Angaben	Webzugriffe	Bestellungen
10	Mittelwert (SP)	11.308	870
11	Standardabweichung (GG)	9379,89	658,52
12	Irrtumswahrscheinlichkeit Alpha	0,05	0,05
13	Stichprobenumfang	43	42

Abbildung 11.66: Die Berechnung von Mittelwert, Standardabweichung und Stichprobenumfang

Da der Marketingleiter eine 95%ige Sicherheit möchte, ist $\alpha = 0,05$, also 5 %.

Die Berechnung des Konfidenzintervalls für den Mittelwert der Webseitenzugriffe und den Mittelwert der Onlinebestellungen können Sie der Abbildung 11.67 entnehmen.

	F	G	H
19		Webseitenzugriffe	Bestellungen
20	KONFIDENZ()	2803,57	199,16
21		=KONFIDENZ.NORM(G12;G11;G13)	=KONFIDENZ.NORM(H12;H11;H13)
22			
23		Untere Grenze	Obere Grenze
24		Mittelwert-KONFIDENZ	Mittelwert+KONFIDENZ
25	Intervall für Webseitenzugriff	8504,54	14111,69
26		=G10-G20	=G10+G20
27			
28			
29		Untere Grenze	Obere Grenze
30		Mittelwert-KONFIDENZ	Mittelwert+KONFIDENZ
31	Intervall für Bestellungen	670,34	1068,66
32		=H10-H20	=H10+H20

Abbildung 11.67: Mit `KONFIDENZ.NORM()` errechnen Sie das 1-Alpha-Konfidenzintervall

**Betrachtung
Webseiten-
zugriffe**

Das Ergebnis mithilfe der Funktion `KONFIDENZ.NORM()` lautet 2.803,57. Dieser Wert stellt die halbe Breite des Vertrauensbereichs des arithmetischen Mittelwerts der Stichprobe dar.

Um nun die untere und obere Grenze des Intervalls zu erhalten, wurde einmal der Stichprobenmittelwert plus `KONFIDENZ.NORM()` und einmal der Stichprobenmittelwert minus `KONFIDENZ.NORM()` errechnet.

Daraus ergibt sich für die untere Grenze ein Wert von 8504,54 und für die obere Grenze ein Wert von 14.111,69.

Da der Marketingleiter mit einem Alpha von 0,05 gerechnet hat, kann aufgrund dieses Ergebnisses **folgende Aussage** getroffen werden:

»Zu 95 % liegt der Mittelwert der Grundgesamtheit der Webseitenzugriffe im Konfidenzintervall von 8504,54 bis 14111,69.«

Da der Stichprobenumfang nur 43 ist, ergibt sich ein recht ungenaues Konfidenzintervall. Je größer die Stichprobe ist, desto genauer bzw. enger wird auch das Konfidenzintervall – das heißt, auch der wahre Mittelwert der Grundgesamtheit wird immer genauer.

G.TEST(), KONFIDENZ.T()

Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Konfidenz.norm*.

Aussagen

Siehe auch



KONFIDENZ.T() CONFIDENCE.T()

KONFIDENZ.T(*Alpha*; *Standardabweichung*; *Umfang*)

Die Funktion KONFIDENZ.T() gibt das Konfidenzintervall für den Erwartungswert einer Zufallsvariablen zurück, wobei der Studentsche T-Test verwendet wird.

Alpha (erforderlich) ist die Irrtumswahrscheinlichkeit bei der Berechnung des Konfidenzintervalls. Das Konfidenzintervall ist gleich $100 \cdot (1 - \text{Alpha})\%$, was bedeutet, dass ein Wert für **Alpha** von 0,05 einem Konfidenzniveau von 95 % entspricht.

Standardabweichung (erforderlich) ist die als bekannt angenommene Standardabweichung der Grundgesamtheit.

Umfang (erforderlich) ist der Umfang der Stichprobe.

Ist eines der Argumente nicht numerisch, gibt KONFIDENZ.T() den Fehlerwert #WERT! zurück.

Ist **Alpha** kleiner oder gleich 0 bzw. **Alpha** größer oder gleich 1, gibt KONFIDENZ.T() den Fehlerwert #ZAHL! zurück.

Ist **Standardabweichung** gleich 0, gibt KONFIDENZ.T() den Fehlerwert #ZAHL! zurück.

Ist **Umfang** keine ganze Zahl, wird der Dezimalanteil abgeschnitten.

Ist **Umfang** gleich 1, gibt KONFIDENZ.T() den Fehlerwert #DIV/0! zurück.

Hintergrundinformationen zu Konfidenzintervallen lesen Sie bei der Funktion KONFIDENZ.NORM() auf Seite 421.

Vergleichen Sie hierzu das Beispiel der Funktion KONFIDENZ.NORM() auf Seite 421.

G.TEST(), T.TEST(), KONFIDENZ(), KONFIDENZ.NORM()

Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* (Excel 2007/2010) auf dem Arbeitsblatt *Konfidenz.t*.

Syntax

Definition

Argumente

Hinweis

Hintergrund

Praxiseinsatz

Siehe auch



KORREL()



CORREL()

Syntax KORREL(*Matrix1*; *Matrix2*)

Definition Die Funktion KORREL() gibt den Korrelationskoeffizient einer zweidimensionalen Zufallsgröße zurück, deren Werte in den Zellbereichen *Matrix1* und *Matrix2* stehen. Mithilfe des Korrelationskoeffizienten lässt sich feststellen, ob es eine Beziehung zwischen zwei Eigenschaften gibt. Sie können beispielsweise die Beziehung zwischen den Zugriffen auf Ihre Webseite und den Onlinebestellungen untersuchen.

Argumente *Matrix1* (erforderlich) ist ein Zellbereich mit Werten.

Matrix2 (erforderlich) ist ein Zellbereich mit Werten.

Hinweis Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Enthalten *Matrix1* und *Matrix2* nicht dieselbe Anzahl von Datenpunkten, gibt KORREL() den Fehlerwert #NV zurück.

Ist eine der Matrizen *Matrix1* oder *Matrix2* leer, oder ist eine der zu deren Werten gehörenden Standardabweichungen s gleich 0, gibt KORREL() den Fehlerwert #DIV/0! zurück.

Hintergrund Gibt es einen Zusammenhang zwischen zwei Variablen? Eine solche Frage taucht häufig auf, wenn Daten, ob nun aus dem Web oder aus anderen Bereichen, analysiert und interpretiert werden. Hierzu können Sie die Korrelationsanalyse verwenden.

Mithilfe des Korrelationskoeffizienten lässt sich feststellen, ob es eine Beziehung zwischen zwei Merkmalen gibt. Das Ergebnis ist eine Zahl im Bereich von 1 (perfekter Zusammenhang) bis -1 (absolute Gegenläufigkeit). Das Vorzeichen gibt also die Richtung der Beziehung an.

Die Korrelationsanalyse ist eine der wichtigsten Methoden zur Bestimmung des linearen oder nichtlinearen Zusammenhangs zwischen zwei Variablen, in unserem Beispiel die Webseituzugriffe und die Bestellungen.

Die Gleichung wird nach folgender Formel berechnet:

$$p_{xy} = \frac{Cov(x, y)}{\sigma_x * \sigma_y}$$

mit

$$-1 \leq p_{xy} \leq 1$$

und mit

$$Cov(x, y) = \frac{1}{n} \sum_{j=1}^n (x_j - \mu_x)(y_j - \mu_y)$$

Der Wert des Korrelationskoeffizienten wird mit folgenden Aussagen belegt:

- ▶ Wenn der Wert kleiner als 0,3 ist, besteht ein geringer Zusammenhang zwischen den Websitebesuchen und den Bestellungen
- ▶ Wenn der Wert zwischen 0,3 und 0,5 liegt, besteht ein mäßiger Zusammenhang
- ▶ Wenn der Wert zwischen 0,5 und 0,7 liegt, besteht ein deutlicher Zusammenhang
- ▶ Wenn der Wert zwischen 0,7 und 0,9 liegt, besteht ein enger Zusammenhang
- ▶ Wenn der Wert größer als 0,9 ist, besteht ein sehr enger Zusammenhang

Ergebnis

Ein in der Softwarebranche tätiges Unternehmen verkauft auf der firmeneigenen Website alle vom Unternehmen angebotenen Produkte. Regelmäßig verschickt das Unternehmen Newsletter, um über Neuheiten zu informieren und bereits vorhandene, aber auch potenzielle Neukunden auf die Website und damit auf das Unternehmen aufmerksam zu machen.

Praxiseinsatz

Die Produktbestellungen auf der Website sind im vergangenen Jahr merklich gestiegen. Die Geschäftsführung möchte die Gründe hierfür erforschen. Hängt der Anstieg der Bestellungen mit dem aktiven Marketing und den straffen Werbemaßnahmen der letzten Zeit zusammen? Sind durch die vermehrten Websitezugriffe auch die Bestellungen gestiegen?

Es soll also aufgezeigt werden, ob ein Zusammenhang zwischen den Websitezugriffen und den Bestellungen besteht.

Erhalten Sie als Ergebnis einen Korrelationskoeffizienten nahe bei +1, zeigt dies einen positiven Zusammenhang zwischen den beiden Zahlenreihen auf. Das heißt, je größer die Werte der Variablen X (Bestellungen) werden, desto größer sind auch die Werte der Variablen Y (Webseitenzugriffe).

Hinweis

Bei einem Korrelationskoeffizient von 0 sind die beiden Variablen voneinander unabhängig.

Ist der Korrelationskoeffizient hingegen -1 , besteht ein vollständig negativer Zusammenhang zwischen X (Bestellungen) und Y (Webseitenzugriffe).

Monat	Webseitenzugriff	Bestellungen
Januar 2007	236	98
Februar 2007	11593	8000
März 2007	18491	6000
April 2007	11743	8587
Mai 2007	11452	7985
Juni 2007	26651	18968
Juli 2007	16287	9753
August 2007	17750	7857
September 2007	19985	13986
Oktober 2007	17285	6875
November 2007	30369	22765
Dezember 2007	19674	9465
Januar 2008	28464	19875
Februar 2008	25000	15987
März 2008	24574	9653
April 2008	23141	12986
Mai 2008	17700	8543
Juni 2008	3702	1654
Korrelationskoeffizient	0,89201155	

Abbildung 11.68: Die Berechnung des Korrelationskoeffizienten mit der Funktion *KORREL()*

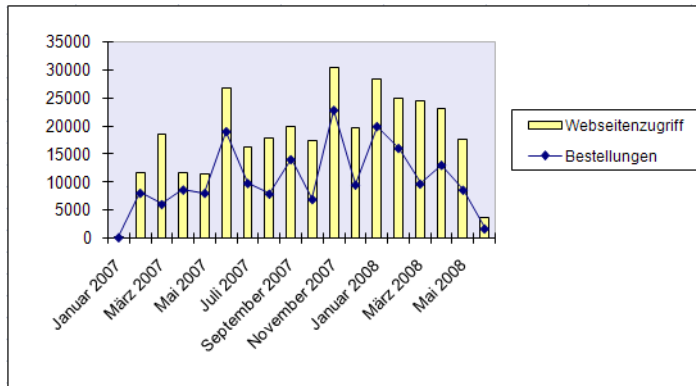


Abbildung 11.69: Die grafische Darstellung der Korrelation zwischen Webseitenzugriff und Bestellungen

Wie Sie der Abbildung 11.68 entnehmen können, lässt sich auch ohne Korrelationskoeffizienten bereits eine Abhängigkeit zwischen den Komponenten Webseitenzugriff und Bestellungen feststellen.

Die Berechnung über die Funktion `KORREL()` liefert den Beweis, wie in Abbildung 11.69 dargestellt.

Der Korrelationskoeffizient in Höhe von 0,89 zeigt einen positiven, *sehr engen* Zusammenhang zwischen den beiden Zahlenreihen auf. Anders ausgedrückt bedeutet dass, erhöht sich die Anzahl der Webseitenzugriffe, beispielsweise durch verschiedene Marketingaktionen, steigt auch die Anzahl der Bestellungen über das Web.

Siehe auch `FISHER()`, `FISHERINV()`, `KOVAR()`, `KOVARIANZ.P()`, `KOVARIANZ.S()`



Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Regression.xlsx* auf dem Arbeitsblatt *Korrel*.

KOVAR() COVAR()

Hinweis Die Funktion `KOVAR()` wird seit Microsoft Excel 2010 durch die Funktionen `KOVARIANZ.P()` und `KOVARIANZ.S()` ersetzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht. Um die Abwärtskompatibilität von `KOVARIANZ.P()` und `KOVARIANZ.S()` zu sichern, ist die Funktion `KOVAR()` weiter unter ihrem alten Namen verfügbar.

Syntax `KOVAR(Matrix1;Matrix2)`

Definition Die Funktion `KOVAR()` gibt die Kovarianz zweier Wertepaare zurück. Sie gibt Auskunft über den Zusammenhang zwischen zwei Datengruppen. So können Sie z.B. ermitteln, ob der vermehrte Eingang von Onlinebestellungen auf Ihrer Website mit der Anzahl der Webseitenzugriffe zusammenhängt. Zur Berechnung der Kovarianz wird die jeweilige Abweichung aller Wertpaare zwischen tatsächlichem Wert und dem Mittelwert miteinander multipliziert und anschließend daraus der Mittelwert gebildet.

Matrix1 (erforderlich) ist der erste Zellbereich, dessen Zellen mit ganzen Zahlen belegt sind.

Argumente

Matrix2 (erforderlich) ist der zweite Zellbereich, dessen Zellen mit ganzen Zahlen belegt sind.

Als Argumente sollten entweder Zahlen oder Namen, Matrizen bzw. Bezüge angegeben werden, die Zahlen enthalten.

Hinweis

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Verfügen *Matrix1* und *Matrix2* nicht über dieselbe Anzahl von Datenpunkten, gibt die Funktion KOVAR() den Fehlerwert #NV zurück.

Ist eine der beiden Matrizen *Matrix1* oder *Matrix2* leer, gibt KOVAR() den Fehlerwert #DIV/0! zurück.

Die Kovarianz beschreibt den Zusammenhang zwischen zwei statistischen Merkmalen x und y durch die Bezeichnungen *positiv* und *negativ*. Das heißt, es wird die Richtung der Abhängigkeit zwischen den beiden Merkmalen aufgezeigt. Die Kovarianz kann jeden reellen Wert annehmen.

Hintergrund

Welche Aussagen können nach Berechnung der Kovarianz getroffen werden?

- ▶ Ist die **Kovarianz positiv**, besitzen x und y tendenziell einen gleichsinnigen linearen Zusammenhang. Das bedeutet, liegen hohe Werte von x vor, liegen auch hohe Werte von y vor. Das gleiche gilt bei niedrigen Werten. In einem Schaubild liefe die Punktwolke von links unten nach rechts oben.
- ▶ Ist die **Kovarianz negativ**, weisen x und y einen gegensinnigen linearen Zusammenhang auf. Das bedeutet, dass hohe Werte der einen Zufallsvariablen mit niedrigen Werten der anderen Zufallsvariablen einhergehen. In einem Schaubild liefe die Punktwolke von links oben nach rechts unten.
- ▶ Ist das Ergebnis 0, besteht kein Zusammenhang oder eine U-förmige Beziehung zwischen den beiden Variablen x und y

Die Kovarianz gibt zwar die Richtung einer Beziehung zwischen zwei Variablen an, über die Stärke des Zusammenhangs jedoch wird keine Aussage getroffen. Ausschlaggebend dafür ist die Abhängigkeit der berechneten Kovarianz von den Maßeinheiten der beteiligten Variablen x und y . Ist die Kovarianz zweier Variablen mit der Maßeinheit »Meter 5,2« so ist die Kovarianz der gleichen Werte in der Maßeinheit Zentimeter »520«. Die Kovarianz ist also maßstabsabhängig.

Die Kovarianz ist als Maßzahl für den stochastischen Zusammenhang nur wenig anschaulich und auch schwer vergleichbar. Um einen Zusammenhang vergleichbar zu machen, kann man die Kovarianz standardisieren. Man erhält dann die Korrelation – sprich eine Beziehung zwischen zwei oder mehr quantitativen statistischen Variablen, deren Maßzahl sich zwischen +1 (perfekter linearer Zusammenhang), 0 (gar kein linearer Zusammenhang) und -1 (perfekter gegensätzlicher linearer Zusammenhang) bewegt.

Hinweis

Zwischen der Kovarianz und den beiden Standardabweichungen kann man folgende wichtige generelle Beziehung herleiten.

- ▶ Sind n Wertepaare $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ mit den Standardabweichungen s_x und s_y und der Kovarianz s_{xy} gegeben, dann gilt, dass die Kovarianz dem Betrag nach höchstens so groß wie das Produkt der Standardabweichungen ist
- ▶ Die obere Grenze wird genau dann erreicht, wenn zwischen x_i und y_i eine perfekte lineare Abhängigkeit besteht mit irgendwelchen festen Zahlen a und b

Die Kovarianz wird wie folgt berechnet:

$$KOVAR(X, Y) = \frac{1}{n} \sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)$$

Dabei sind x und y die Stichprobenmittelwerte $MITTELWERT(Array1)$ und $MITTELWERT(Array2)$, und n ist der Stichprobenumfang.

Praxiseinsatz

Blieben wir bei dem Beispiel des Softwareunternehmens, das auf der firmeneigenen Website alle vom Unternehmen angebotenen Produkte verkauft und, um den Verkauf anzukurbeln, deshalb regelmäßig Newsletter verschickt.

Die Produktbestellungen auf der Website sind im vergangenen Jahr merklich gestiegen. Um die Gründe hierfür zu erforschen, hat die Geschäftsführung mithilfe der Errechnung des Korrelationskoeffizienten bereits erste Zusammenhänge aufgezeigt.

Nun wollen Sie noch eine Aussage über die Richtung des Zusammenhangs zwischen Webseitenzugriff und Onlinebestellungen treffen und deshalb die Kovarianz berechnen.

Die Abbildung 11.70 zeigt das Ergebnis.

	B	C	D
10	Monat	Webseitenzugriff	Bestellungen
11	Januar 2007	236	98
12	Februar 2007	11593	8000
13	März 2007	18491	6000
14	April 2007	11743	8587
15	Mai 2007	11452	7985
16	Juni 2007	26651	18968
17	Juli 2007	16287	9753
18	August 2007	17750	7857
19	September 2007	19985	13986
20	Oktober 2007	17285	6875
21	November 2007	30369	22765
22	Dezember 2007	19674	9465
23	Januar 2008	28464	19875
24	Februar 2008	25000	15987
25	März 2008	24574	9653
26	April 2008	23141	12986
27	Mai 2008	17700	8543
28	Juni 2008	3702	1654
29	Korrelationskoeffizient		0,89201155
30	Mittelwert	18005,39	10502,06
31	Standardabweichung	8079,11	5978,92
32	Kovarianz		40694286,37
33			=KOVAR(D11:D28;C11:C28)

Abbildung 11.70: Mit der Funktion $KOVAR()$ wird die Richtung der Abhängigkeit zwischen zwei Wertereihen berechnet

Durch das positive Ergebnis der Berechnung der Kovarianz kann nun die Aussage getroffen werden, dass die x und y -Werte, also die Werte der Webseitenzugriffe und die der Bestellungen, tendenziell einen gleichsinnigen linearen Zusammenhang besitzen. Liegen hohe Werte von x vor, liegen auch hohe Werte von y vor, liegen niedrige x -Werte vor, liegen auch niedrige y -Werte vor.

Die positive Kovarianz wird durch den Korrelationskoeffizienten von 0,89 unterstrichen (siehe Abbildung 11.70), da dieser, je näher er an +1 liegt, einen immer größeren linearen Zusammenhang aufzeigt.

Wie man ohne die Funktion KOVAR(), also über die unter dem Abschnitt *Hintergrund* angegebene Formel, zum gleichen Ergebnis gekommen wäre, stellt Abbildung 11.71 dar.

	B	C	D
37	Webseitenzugriffe	Bestellungen	Produkt x*y
38	-17769,388889	-10404,055556	184873709,2
39	-6412,388889	-2502,055556	16044153,24
40	485,6111111	-4502,055556	-2186248,201
41	-6262,388889	-1915,055556	11992822,63
42	-6553,388889	-2517,055556	16495243,91
43	8645,6111111	8465,944444	73193263,35
44	-1718,388889	-749,0555556	1287168,744
45	-255,3888889	-2645,055556	675517,7994
46	1979,6111111	3483,944444	6896855,133
47	-720,3888889	-3627,055556	2612890,522
48	12363,611111	12262,94444	151614276,2
49	1668,6111111	-1037,055556	-1730442,423
50	10458,611111	9372,944444	98027980,91
51	6994,6111111	5484,944444	38365053,35
52	6568,6111111	-849,0555556	-5577115,756
53	5135,6111111	2483,944444	12756572,69
54	-305,3888889	-1959,055556	598273,7994
55	-14303,39	-8848,06	126557179,5
56	=C28-\$C\$30	=D28-\$D\$30	=B55*C55
	=Anzahl der Webseitenzugriffe im Juni (C28) abzüglich dem Mittelwert aus der Summe aller Webseitenzugriffe (C30)	=Anzahl der Bestellungen im Juni (D28) abzüglich dem Mittelwert aus der Summe aller Bestellungen (D30)	= Produkt aus den Ergebnissen der rot markierten Zellen unter Webseitenzugriffe und Bestellungen (F28*G28)
57			40694286,37
58			=MITTELWERT(D38:D55)
59			

Abbildung 11.71: Die Berechnung der Kovarianz über die mathematische Formel

FISHER(), FISHERINV(), KORREL(), KOVARIANZ.P(), KOVARIANZ.S()

Siehe auch

Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Regression.xlsx* auf dem Arbeitsblatt *Kovar*.



KOVARIANZ.P()



COVARIANCE.P()

KOVARIANZ.P(*Matrix1*; *Matrix2*)

Syntax

Die Funktion KOVARIANZ.P() gibt die Kovarianz einer Grundgesamtheit, den Mittelwert der für alle Datenpunktpaare gebildeten Produkte der Abweichungen zurück.

Definition

Die Kovarianz gibt Auskunft darüber, welcher Zusammenhang zwischen zwei Datengruppen besteht. Beispielsweise können Sie ermitteln, ob ein größeres Einkommen Folge des jeweiligen Ausbindungsgrads ist.

Matrix1 (erforderlich) ist der erste Zellbereich, dessen Zellen mit ganzen Zahlen belegt sind.

Argumente

Matrix2 (erforderlich) ist der zweite Zellbereich, dessen Zellen mit ganzen Zahlen belegt sind.

Hinweis Als Argumente sollten entweder Zahlen oder Namen, Matrizen bzw. Bezüge angegeben werden, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Verfügen *Matrix1* und *Matrix2* nicht über dieselbe Anzahl von Datenpunkten, gibt die Funktion KOVARIANZ.P() den Fehlerwert #NV zurück.

Ist eine der beiden Matrizen *Matrix1* oder *Matrix2* leer, gibt KOVARIANZ.P() den Fehlerwert #DIV/0! zurück.

Hinweis Mehr Informationen zum Thema Kovarianz lesen Sie im Abschnitt KOVAR() auf Seite 428.

Siehe auch FISHER(), FISHERINV(), KORREL(), KOVAR(), KOVARIANZ.S()



Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Regression.xlsx* (Excel 2007/2010) auf dem Arbeitsblatt *Kovarianz.p*.

KOVARIANZ.S() COVARIANCE.S()

Syntax KOVARIANZ.S(*Matrix1*; *Matrix2*)

Definition Die Funktion KOVARIANZ.S() gibt die Kovarianz einer Stichprobe zurück, d.h. den Mittelwert der für alle Datenpunktpaare gebildeten Produkte der Abweichungen.

Argumente *Matrix1* (erforderlich) ist der erste Zellbereich, dessen Zellen mit ganzen Zahlen belegt sind.
Matrix2 (erforderlich) ist der zweite Zellbereich, dessen Zellen mit ganzen Zahlen belegt sind.

Hinweis Als Argumente sollten entweder Zahlen oder Namen, Matrizen bzw. Bezüge angegeben werden, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Verfügen *Matrix1* und *Matrix2* nicht über dieselbe Anzahl von Datenpunkten, gibt die Funktion KOVARIANZ.S() den Fehlerwert #NV zurück.

Ist eine der beiden Matrizen *Matrix1* oder *Matrix2* leer, gibt KOVARIANZ.S() den Fehlerwert #DIV/0! zurück.

Mehr Informationen zum Thema Kovarianz lesen Sie im Abschnitt KOVAR().

Siehe auch FISHER(), FISHERINV(), KORREL(), KOVAR(), KOVARIANZ.P()



Dieses Beispiel finden Sie im Ordner *Buch\Kap11* in der Arbeitsmappe *Regression.xlsx* (Excel 2007/2010) auf dem Arbeitsblatt *Kovarianz.s*.

KURT()



KURT()

KURT(*Zahl1*; *Zahl2*; ...)

Die Funktion KURT() gibt die Kurtosis (Exzess) einer Datengruppe zurück. Die Kurtosis ist ein Maß für die Wölbung, das heißt, wie spitz oder flach eine Verteilung im Vergleich zu der Normalverteilung verläuft. Eine positive Kurtosis weist auf eine relativ schmale, spitze Verteilung und eine negative Kurtosis weist auf eine relativ flache Verteilung hin.

Zahl1 (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 Argumente (30 bis Excel 2003), für die Sie die Kurtosis berechnen möchten. An Stelle der durch Semikolons voneinander getrennten Argumente können Sie auch eine Matrix oder einen Bezug auf eine Matrix verwenden.

Als Argumente sollten entweder Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Sind weniger als vier Zahlen angegeben oder ist die Standardabweichung der Stichprobe gleich 0, gibt KURT() den Fehlerwert #DIV/0! zurück.

Die Kurtosis ist ein Maß dafür, wie spitz oder flach die Verteilung (Kurve) der beobachteten Werte (Daten) im Vergleich zu einer Normalverteilung ist.

Eine Normalverteilung hat eine Kurtosis von 0.

Nähere Informationen zur Normalverteilung und der Funktion NORM.VERT() / NORMVERT() finden Sie auf Seite 468 dieses Buchs.

Die Kurtosis, also die Steilheit der Verteilung, gibt an, ob die empirische Verteilung »steiler« ist als die Normalverteilung. Dies ist dann der Fall, wenn die Kurtosis positiv ist.

Eine Kurtosis ist wie folgt definiert:

$$\left\{ \frac{n(n+1)}{(n-1)(n-2)(n-3)} \sum \left(\frac{x_i - \bar{x}}{s} \right)^4 \right\} - \frac{3(n-1)^2}{(n-2)(n-3)}$$

Hierbei ist s die Standardabweichung der Stichprobe.

Die Kurtosis, ebenso wie die Schiefe einer Verteilung, charakterisieren die Form und den Grad der Symmetrie einer Verteilung. Beide Berechnungsmöglichkeiten haben eine wichtige Funktion beim Finden der »richtigen« Verteilung, mit deren Hilfe in angemessener Weise die Daten weiter ausgewertet werden können.

Mehr zum Thema *Schiefe* finden Sie auf Seite 503 dieses Buchs.

Die Kurtosis, auch Wölbung oder Exzess genannt, einer statistischen Verteilung ist definiert als:

$$\frac{m_4(\mu)}{\sigma^4} - 3$$

Syntax**Definition****Argumente****Hinweis****Hintergrund****Hinweis****Hinweis**

Hierbei ist $m_4(\mu)$ das vierte zentrale Moment und σ die Standardabweichung.

Wie bereits erwähnt, beschreibt die Wölbung die Abweichung des Verlaufs der gegebenen Wahrscheinlichkeitsverteilung zum Verlauf der Normalverteilung. Eine im Vergleich zur Normalverteilung flachere Wahrscheinlichkeitsverteilung wird »subgauß-förmig« genannt, eine im Vergleich spitzere Verteilung heißt »supergauß'sch«.

Praxiseinsatz

Um Ihnen die Funktionsweise der KURT() zu verdeutlichen, greifen wir auf das Webseitenbeispiel unseres Softwareherstellers zurück. Die Marketingabteilung möchte eine Auswertung des Download- und gesamten Webseitenbereichs machen und hierbei die Kurtosis, also die Wölbung »Verteilung der Webseitenklicks« berechnen.

Als Ergebnis dieser Berechnung erhalten die Mitarbeiter einen Wert, mit dessen Hilfe sie die Abweichung des Verlaufs der gegebenen Wahrscheinlichkeitsverteilung zum Verlauf der Normalverteilung beurteilen können.

	B	C	D	E	F
	DATUM	ANZAHL KLIKS IM BEREICH DOWNLOAD	MITTELWERT	ANZAHL KLIKS AUF DER GESAMTEN WEBSEITE	MITTELWERT
10					
11	Jan 07	200	684	11123	14424
12	Feb 07	260	684	12345	14424
13	Mrz 07	400	684	11414	14424
14	Apr 07	540	684	12564	14424
15	Mai 07	680	684	7432	14424
16	Jun 07	820	684	17836	14424
17	Jul 07	960	684	11468	14424
18	Aug 07	1100	684	12937	14424
19	Sep 07	1240	684	14714	14424
20	Okt 07	1240	684	11799	14424
21	Nov 07	1100	684	21732	14424
22	Dez 07	960	684	14107	14424
23	Jan 08	820	684	19539	14424
24	Feb 08	680	684	20397	14424
25	Mrz 08	540	684	17687	14424
26	Apr 08	400	684	17173	14424
27	Mai 08	260	684	13021	14424
28	Jun 08	120	684	12345	14424
29	Kurt	-1,27		-0,46	

Abbildung 11.72: Die Berechnung der Kurtosis sowohl für den Downloadbereich als auch für den gesamten Bereich der Webseite

Um eine Vorstellung der Wahrscheinlichkeitsverteilung im Vergleich zur Normalverteilung sowohl im Downloadbereich als auch für den gesamten Webseitenbereich zu erhalten, werden mit wenigen Mausklicks noch die dazugehörigen Diagramme erstellt. Diese stellen sich, wie in Abbildung 11.73 und in Abbildung 11.74 aufgezeigt, dar.

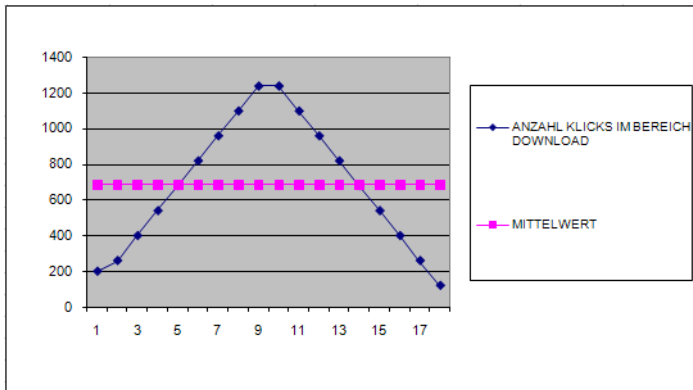


Abbildung 11.73: Die Darstellung der Kurtosis für den Downloadbereich

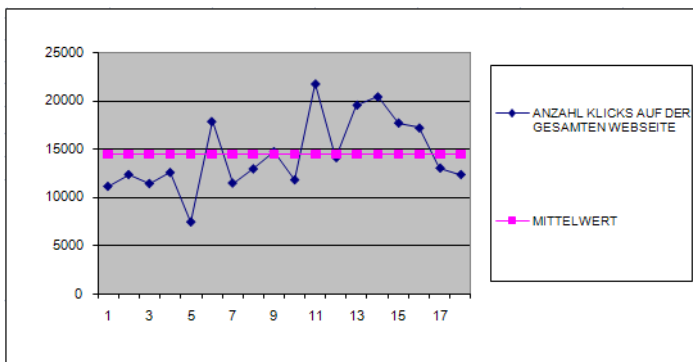


Abbildung 11.74: Die Darstellung der Kurtosis für den gesamten Webseitenbereich

1. Die negative Kurtosis für den Downloadbereich mit dem Wert $-1,27$ bedeutet, dass es sich hierbei um eine relativ flache Verteilung im Vergleich zur Normalverteilung handelt. Die Werte der empirischen Verteilung streuen weiter um den Mittelwert als bei der Normalverteilung.
2. Die negative Kurtosis für den gesamten Webseitenbereich mit dem Wert $-0,46$ bedeutet, dass es sich hierbei um eine relativ schmale, spitze Verteilung im Vergleich zur Normalverteilung, handelt. Die Werte der empirischen Verteilung streuen enger um den Mittelwert als bei der Normalverteilung.

Aussagen

SCHIEFE(), SCHIEFE.P(), STABW(), STDABWN(), VARIANZ(), VARIANZEN()

Siehe auch

Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Symmetrie.xlsx* auf dem Arbeitsblatt *Kurtosis*.



LOGNORM.INV() / LOGINV()

LOGNORM.INV() / LOGINV()

Syntax LOGNORM.INV(*Wahrsch*; *Mittelwert*; *Standabwn*)

Definition Die Funktion LOGNORM.INV() gibt Quantile der Lognormalverteilung von x zurück, wobei $\ln(x)$ mit den Parametern *Mittelwert* und *Standabwn* normal verteilt ist. Ist $p = \text{LOGNORM.VERT}(x, \dots)$, gilt $\text{LOGNORM.INV}(p, \dots) = x$. Bei gegebener Wahrscheinlichkeit p können Sie das entsprechende Quantil der Lognormalverteilung bestimmen.

Mit der Lognormalverteilung können Sie logarithmisch transformierte Daten analysieren.

Argumente *Wahrsch* (erforderlich) ist die zur Lognormalverteilung gehörige Wahrscheinlichkeit.

Mittelwert (erforderlich) ist der Mittelwert der Lognormalverteilung.

Standabwn (erforderlich) ist die Standardabweichung der Lognormalverteilung.

Hinweis Ist eines der Argumente kein numerischer Ausdruck, gibt die Funktion LOGNORM.INV() den Fehlerwert #WERT! zurück.

Ist *Wahrsch* kleiner 0 oder *Wahrsch* größer 1, gibt LOGNORM.INV() den Fehlerwert #ZAHL! zurück.

Ist *Standabwn* kleiner oder gleich 0, gibt LOGNORM.INV() den Fehlerwert #ZAHL! zurück.

Hintergrund Die Umkehrfunktion der Lognormalverteilung lautet:

$$\text{LOGNORM.INV}(p, \mu, \sigma) = e^{\left[\mu + \sigma x(\text{NORM.INV}(p))\right]}$$

Hinweis Mehr Informationen zur Lognormalverteilung und der Funktion LOGNORM.VERT() finden Sie auf Seite 437.

Praxiseinsatz Gehen Sie folgendermaßen zur Berechnung von LOGNORM.INV() vor:

Gegeben sind die Werte:

- ▶ 0,039084 = Die zur Lognormalverteilung gehörende Wahrscheinlichkeit (*Wahrsch*)
- ▶ 3,5 = Das Mittel von $\ln(x)$ (*Mittelwert*)
- ▶ 1,2 = Die Standardabweichung von $\ln(x)$ (*Standabwn*)

Die Berechnung von LOGNORM.INV() sehen Sie in Abbildung 11.75.

	B	C	D	E	F
8	Berechnung von LOGNORM.INV()				
9	Bedeutung			Parameter	
10	Die zur Lognormalverteilung gehörende Wahrscheinlichkeit			0,039084	
11	Mittel von $\ln(x)$			3,5	
12	Standardabweichung von $\ln(x)$			1,2	
13	LOGNORM.INV()			4,000025	
14				=LOGNORM.INV(E10;E11;E12)	

Abbildung 11.75: Die Berechnung von LOGNORM.INV()

Unter Angabe der in Abbildung 11.75 dargestellten Parameter gibt die Funktion LOGNORM.INV() die Quantile der Lognormalverteilung in Höhe von 4,000025 zurück.

EXP(), LN(), LOG(), LOG10(), LOGNORM.VERT()

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Lognorm.inv*.

Siehe auch



LOGNORM.VERT() / LOGNORMVERT()

LOGNORM.DIST() / LOGNORMDIST()

LOGNORM.VERT(*x*; *Mittelwert*; *Standabwn*; *Kumuliert*)

Die Funktion LOGNORM.VERT() gibt Werte der Verteilungsfunktion einer lognormalverteilten Zufallsvariablen zurück, wobei $\ln(x)$ normalverteilt ist mit den Parametern *Mittelwert* und *Standabwn*. Mit dieser Funktion können Sie Daten untersuchen, die logarithmisch transformiert wurden.

x (erforderlich) ist der Wert, für den die Funktion ausgewertet werden soll.

Mittelwert (erforderlich) ist der Mittelwert der Lognormalverteilung.

Standabwn (erforderlich) ist die Standardabweichung der Lognormalverteilung.

Kumuliert (erforderlich) ist der Wahrheitswert, der den Typ der Funktion bestimmt.

Ist eines der Argumente kein numerischer Ausdruck, gibt die Funktion LOGNORM.VERT() den Fehlerwert #WERT! zurück.

Ist *x* kleiner oder gleich 0, oder *Standabwn* kleiner oder gleich 0, gibt LOGNORM.VERT() den Fehlerwert #ZAHL! zurück.

Die Funktion LOGNORM.VERT() liefert die Wahrscheinlichkeiten einer logarithmisch normalverteilten Zufallsvariablen. Sie können mit dieser Funktion Wahrscheinlichkeitsverteilungen betrachten, bei denen nicht die Zufallsvariable selbst, sondern ihr natürlicher Logarithmus normalverteilt ist.

Die Lognormalverteilung ist formelmässig der Normalverteilung ähnlich, hat jedoch einen Logarithmus im Exponenten. Sie ist rechtsschief, das heißt, sie steigt stark an und fällt dann langsamer wieder ab (Abbildung 11.76).

Gilt also eine Zufallsvariable als lognormalverteilt, dann ist folglich ihr Logarithmus normalverteilt. Kennzeichnend für eine lognormalverteilte Variable ist, dass unendlich viele Einflüsse multiplikativ auf sie einwirken.

Häufig sind Einkommen lognormalverteilt. Einer der Gründe hierfür liegt in der normalerweise prozentualen Einkommenserhöhung. Große Einkommen werden auf diese Weise stark erhöht, kleinere Einkommen nur wenig. Im Laufe der Zeit verbleiben die vielen kleineren Einkommen unten, während die wenigen großen nach oben wegdriften. Deshalb entsteht die rechtsschiefe Verteilung. Das Logarithmieren überführt die multiplikative Struktur in eine additive. Die logarithmierten Werte sind dann normalverteilt.

Syntax

Definition

Argumente

Hinweis

Hintergrund

Ein weiterer Grund für die lognormalverteilte Einkommensstruktur ist, dass es viel weniger hochdotierte Positionen gibt. Die Masse der Arbeitsplätze sind mit mehr oder weniger geringem Einkommen belegt, wobei besonders niedrige Einkommen wieder seltener werden.

Diese Tatsache entspricht genau dem Verlauf der meisten Lognormalverteilungen.

Die Lognormalverteilung mit den Parametern μ und σ^2 ist auf den positiven reellen Zahlen durch die folgende Wahrscheinlichkeitsdichte erklärt.

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \frac{1}{x} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}$$

Die grafische Darstellung der Wahrscheinlichkeitsdichte der Lognormalverteilung sehen Sie in Abbildung 11.76.

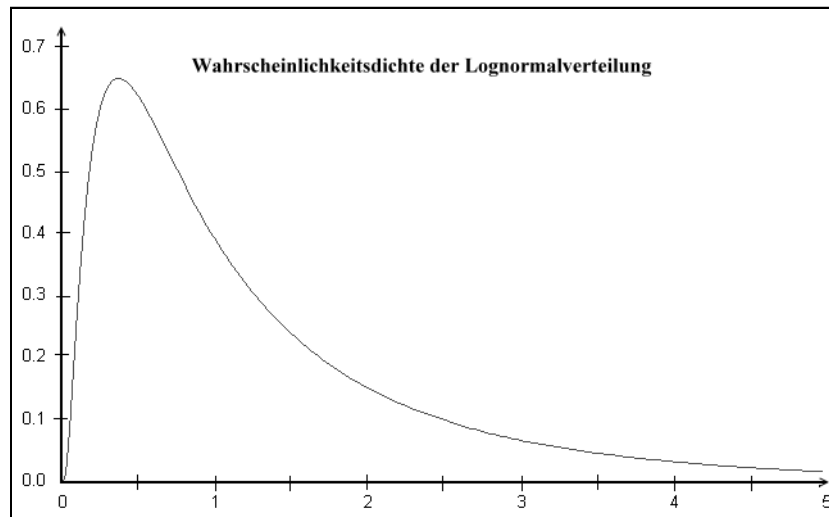


Abbildung 11.76: Wahrscheinlichkeitsdichte der Lognormalverteilung mit $\mu = 0$ und $\sigma = 1$

Die Gleichung für die Verteilungsfunktion einer logarithmischen Normalverteilung lautet:

$$\text{LOGNORM.VERT}(x, \mu, \sigma) = \text{NORM.VERT}\left(\frac{\ln(x) - \mu}{\sigma}\right)$$

Praxiseinsatz Gehen Sie folgendermaßen zur Berechnung von LOGNORM.VERT() vor:

Gegeben sind die Werte:

- ▶ 4 = Wert, für den die Funktion ausgewertet werden soll (x)
- ▶ 3,5 = Das Mittel von $\ln(x)$ (*Mittelwert*)
- ▶ 1,2 = Die Standardabweichung von $\ln(x)$ (*Standabwn*)
- ▶ Wahr = Der Wahrheitswert, der den Typ der Funktion bestimmt

Die Berechnung von LOGNORM.VERT() sehen Sie in Abbildung 11.77.

	B	C	D	E	F
8	Berechnung von LOGNORM.VERT()				
9	Bedeutung			Parameter	
10	Wert, für den die Funktion ausgewertet werden soll			4	
11	Mittel von ln(x)			3,5	
12	Standardabweichung von ln(x)			1,2	
13	Kumuliert			WAHR	
14	LOGNORM.VERT()			0,039083556	
15	=LOGNORM.VERT(E10;E11;E12;wahr)				

Abbildung 11.77: Die Berechnung von LOGNORM.VERT()

Unter Angabe der in Abbildung 11.77 dargestellten Parameter gibt die Funktion LOGNORM.VERT() die kumulierte Lognormalverteilung in Höhe von 0,039084 zurück.

EXP(), LN(), LOG(), LOG10(), LOGNORM.INV()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Lognorm.vert*.

Siehe auch



MAX()



MAX()

MAX(Zahl1;Zahl2;...)

MAX() gibt den größten Wert innerhalb einer Argumentliste zurück.

Zahl1 (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 Zahlen (30 bis Excel 2003), für die Sie den Maximalwert finden möchten.

Für die Ermittlung der kleinsten und größten Werte einer Datenmenge verwenden Sie die Funktionen MIN() und MAX().

Als Argumente können Sie Zahlen, aber auch leere Zellen, Wahrheitswerte und in Textform vorliegende Zahlen angeben. Als Fehlerwerte oder Text angegebene Argumente, die nicht in Zahlen umgewandelt werden können, führen zu Fehlern.

Ist als Argument eine Matrix oder ein Bezug angegeben, werden von deren oder dessen Elementen nur diejenigen berücksichtigt, die Zahlen sind. Zu der Matrix oder dem Bezug gehörende leere Zellen, Wahrheitswerte oder Text werden ignoriert. Sollen Wahrheitswerte und Text berücksichtigt werden, verwenden Sie stattdessen MAXA() bzw. MINA().

Enthalten die Argumente keine Zahlen, geben MAX() und MIN() den Wert 0 zurück.

Der Chef der Abteilung Rechnungswesen des Softwareherstellers will aus einer Tabelle, die die Umsätze der vergangenen zwei Jahre beinhaltet, den höchsten Wert angezeigt bekommen.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

	A	B	C	D	E
	Fragestellung: Wie hoch war der höchste Umsatz?				
1					
2		Monat	Umsatz		Der größte Umsatz
3		Januar 07	107.629 €		325.698 €
4		Februar 07	185.385 €		=MAX(C3:C26)
5		März 07	180.807 €		
6		April 07	124.328 €		
7		Mai 07	146.215 €		
8		Juni 07	185.675 €		
9		Juli 07	210.169 €		
10		August 07	221.729 €		
11		September 07	234.187 €		
12		Oktober 07	237.947 €		
13		November 07	210.088 €		
14		Dezember 07	207.791 €		
15		Januar 08	201.097 €		
16		Februar 08	222.460 €		
17		März 08	208.585 €		
18		April 08	206.387 €		
19		Mai 08	218.951 €		
20		Juni 08	325.698 €		
21		Juli 08	123.984 €		
22		August 08	325.146 €		
23		September 08	251.462 €		
24		Oktober 08	124.652 €		
25		November 08	123.568 €		
26		Dezember 08	301.245 €		

Abbildung 11.78: Mit der Formel `MAX()` wird der höchste Wert innerhalb der Tabelle ermittelt

Ohne die Daten vorher zu sortieren, erhält man mit `MAX()`, wie in Abbildung 11.78 dargestellt, den größten Wert einer Datenmenge.

Tip Um sich den dazugehörigen Monat anzeigen zu lassen, können Sie auf die bedingte Formatierung oder auf die unter `KKLEINSTE()` erläuterten Funktionen `VERGLEICH()` und `INDEX()` zurückgreifen.

Siehe auch `DBMAX()`, `MAXA()`, `MIN()`, `MINA`



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe `Zählen.xlsx` auf dem Arbeitsblatt `Max&Min`.

MAXA() MAXA()

Syntax `MAXA(Wert1;Wert2;...)`

Definition `MAXA()` gibt den größten Wert einer Liste von Argumenten zurück. Anders als bei `MAX()` werden auch Text und Wahrheitswerte wie `WAHR` und `FALSCH` wie Zahlen verglichen.

Argumente `Wert1` (erforderlich); `Wert2` (optional); ... sind 1 bis 255 Werte (30 bis Excel 2003), deren größter Wert ermittelt werden soll.

Hintergrund Haben Sie eine Tabelle, deren Inhalt bzw. Datenmenge nicht nur aus Zahlenwerten, sondern auch aus Text und Wahrheitswerten besteht, können Sie den größten oder kleinsten Wert dieser Tabelle mit der Funktion `MAXA()` bzw. `MINA()` ermitteln.

Bei `MAXA()` und `MINA()` zählen zu den zulässigen Argumenten Zahlen, leere Zellen, Wahrheitswerte oder Zahlen in Textdarstellung. Befinden sich unter den Argumenten falsche Werte, verursachen diese Fehler. Ist ein Argument als Matrix oder Bezug gegeben, werden in diesem Bereich nur Werte verwendet. Leere Zellen und Werte, die als Text gegeben sind, werden nicht berücksichtigt.

Wahrheitswerte werden folgendermaßen bewertet:

- ▶ Wahrheitswert *WAHR* = 1
- ▶ Wahrheitswert *FALSCH* = 0 (das gilt auch für Argumente die Text enthalten)

Enthalten die Argumente keine Werte, geben sowohl *MAXA()* als auch *MINA()* den Wert 0 zurück.

Sollen in der Berechnung Text und Wahrheitswerte nicht berücksichtigt werden, verwenden Sie stattdessen die Arbeitsblatffunktion *MAX()* bzw. *MIN()*.

Hinweis

Um Ihnen darzustellen, dass bei *MAXA()* beispielsweise auch Wahrheitswerte wie *WAHR* oder *FALSCH* berücksichtigt werden, und deren Wert, wie zuvor beschrieben bei 0 bzw. 1 liegt, haben wir für die anderen Werte der Beispieltabelle (Abbildung 11.79) Zahlen zwischen 0 und 1 gewählt.

Praxiseinsatz

	B	C	D	E
1	Fragestellung: Wie hoch ist der höchste Wert der Tabelle?			
2				
3				
4	Monate	Daten		MAXA()
5	Januar	WAHR		1
6	Februar	0,7	=MAXA(C5:C16)	
7	März	0,6		
8	April	0,4		
9	Mai	0,5		
10	Juni	0,2		
11	Juli	0,3		
12	August	0,2		
13	September	0,5		
14	Oktober	FALSCH		
15	November	0,7		
16	Dezember	0,3		

Abbildung 11.79: Zur Berechnung des höchsten Werts einer Datenmenge, die Text oder Wahrheitswerte beinhaltet, verwendet man *MAXA()*

Als Ergebnis von *MAXA()* erhalten Sie »1«, da der höchste Wert in diesem Fall durch den Wahrheitswert *WAHR* dargestellt wird.

DBMAX(), *MINA()*

Siehe auch

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *Maxa&Mina*.



MEDIAN()



MEDIAN(Zahl1;Zahl2;...)

Syntax

Der Median ist genau die Zahl, die in der Mitte einer Zahlenreihe liegt. Das heißt, dass die eine Hälfte der Zahlen Werte beinhaltet, die kleiner sind als der errechnete Median, und die andere Hälfte größere. Somit ist die Ordnungsnummer des Medians innerhalb der sortierten Datenreihe gleich der halben Anzahl der Elemente.

Definition

Zahl1 (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 Zahlen, deren Median Sie berechnen möchten.

Argumente

Hinweis Die Argumente müssen entweder Zahlen, Namen, Matrizen oder Bezüge sein, die Zahlen enthalten. Es werden alle Zahlen, die zu einem als Bezug oder Matrix angegebenen Argument gehören, geprüft.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Besteht eine Zahlenreihe aus einer geraden Anzahl von Zahlen, berechnet MEDIAN() den Mittelwert der beiden mittleren Zahlen.

Hintergrund Der Median, auch Zentralwert genannt, gibt den mittleren Wert aus einer Datenmenge an. Bei ungeraden Zahlen von Beobachtungswerten ist er einer der tatsächlichen beobachteten Werte. Gibt es keine eindeutige Mitte, was bei gerader Anzahl an Elementen der Fall ist, wird der Median über das arithmetische Mittel der beiden mittleren Werte berechnet.

Handelt es sich um klassifizierte Werte, liegt der Median in der Klasse, in der die Summenfunktion den Wert 0,5 übersteigt. In diesen Fällen muss er geschätzt bzw. durch lineare Interpolation genauer berechnet werden. Voraussetzung für die Berechnung des Medians ist Ordinalskalenniveau und dass die Beobachtungswerte nach ihrer Größe geordnet sind.

Auch wenn der Median im Prinzip leicht verständlich ist, ist er weniger bekannt als das arithmetische Mittel oder der Modalwert. Die Formel für den Median lautet:

$$\bar{x} = x_{j-1} + \frac{b_j}{n_j} \left(\frac{n}{2} - N_{j-1} \right)$$

Im Unterschied zum Mittelwert ist der Median nicht ausreißerabhängig. So bietet sich die Berechnung des Medians z.B. an, wenn man die mittlere Studiendauer errechnen möchte, ohne Langzeitstudenten und andere »Ausreißer« zu berücksichtigen.

Der Median ist also vor allem dann hilfreich, wenn die Werte an den Rändern von geringer Bedeutung sind oder wegen ihrer Ausreißereigenschaften zu Fehlinterpretationen führen könnten, wenn das arithmetische Mittel angewandt würde. Der Median liefert auch dann ein besseres Ergebnis als der Mittelwert, wenn die Zahl der untersuchten Fälle gering ist.

Eine weitere Eigenschaft des Median ist seine geringe Sensibilität. Bestimmte Werte einer vorgegebenen Zahlenmenge können durch andere Werte ersetzt werden, ohne dass dies eine Auswirkung auf den Median hat.

Der gemeinsame Median zweier statistischer Mengen kann nicht durch Zusammenfassen der entsprechenden Mediane beider Beobachtungsreihen ermittelt werden. Die Beobachtungsreihen müssen für eine solche Berechnung zusammengeführt, neu geordnet und anschließend erneut ermittelt werden.

Eine Verallgemeinerung des Medians stellen Quantile dar. Während der Median die Verteilung genau halbiert, teilen Quantile sie in mehrere gleiche Teile auf. Am gebräuchlichsten sind Quartile (vier Teile). Wie beim Median ist auch hier Ordinalskalenniveau erforderlich sowie eine Sortierung der Beobachtungswerte entsprechend ihrer Rangordnung.

Praxiseinsatz Die Marketingabteilung des Softwareherstellers hat eine Auswertung der Webseite für das vergangene Jahr vorgenommen. In der Auswertung wurden alle Klicks in den einzelnen Webseitenbereichen registriert. Nun möchte der Marketingleiter den Median, also den Zentralwert, errechnen, um den mittleren Wert aus dieser Datenmenge zu generieren und somit eine Aussage über den Webseitenzugriff in den letzten zwölf Monaten tätigen zu können.

	B	C	D	E	F	G	H	I
8	Fragestellung: Wie war der Zugriff auf die Webseite							
9	Umsätze	Events	Publikationen	Team	Training	Wissen	Gesamtergebnis	Mittelwert
10	Januar	22	2	19	19	43	105	21
11	Februar	1.170	356	505	1.319	3.000	6.350	1.270
12	März	1.545	756	1.307	2.233	6.116	11.957	2.391
13	April	1.168	518	884	1.903	2.860	7.333	1.467
14	Mai	1.139	684	835	1.589	3.126	7.373	1.475
15	Juni	5.083	1.678	820	2.428	6.682	16.691	3.338
16	Juli	3.609	844	276	1.559	3.311	9.599	1.920
17	August	4.810	622	365	1.502	3.317	10.616	2.123
18	September	5.581	893	309	1.628	3.542	11.953	2.391
19	Oktober	5.506	724	259	1.440	2.971	10.900	2.180
20	November	11.786	1.046	447	2.255	4.735	20.269	4.054
21	Dezember	6.227	339	234	1.583	2.652	11.035	2.207
22	Median	4.210	704	406	1.586	3.219	10.758	2.152
23	Mittelwert	3.971	705	522	1.622	3.530	1.622	2.070
24	Gesamt Median		1380	1609				

Abbildung 11.80: Für die Webseitenzugriffe werden sowohl der Median als auch der Mittelwert berechnet

In diesem Beispiel wurde der Median aus den Werten der beiden mittleren Werte jeder Beobachtungsreihe ermittelt, da es sich um eine gerade Anzahl an Elementen handelt.

Würde man die Beobachtungsreihe »Events« sortieren, wäre zu erkennen, dass es sich bei diesen beiden mittleren Werten um die Monate Juli (3.609) und August (4.810) handelt. Diese beiden Werte wurden mithilfe der Funktion MEDIAN() addiert und anschließend durch 2 geteilt. Als Ergebnis folgt ein Median von 4.210.

Wäre in diesem Beispiel ein 13. Monat angegeben, so würde als Median der Wert errechnet, der in einer nach dem Rang sortierten Datenreihe die siebte Stelle einnimmt.

Der Vergleich mit dem Mittelwert zeigt, dass beim Median die Ausreißer im Januar und November weniger ins Gewicht fallen.

Hinweis

ANZAHL(), ANZAHL2(), DBMITTELWERT(), MITTELWERT(), MITTELWERTWENN(), MITTELWERTWENNS(), MODUS.EINF(), SUMME()

Siehe auch

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Median*.



MIN() MIN()

MIN(*Zahl1*; *Zahl2*; ...)

Syntax

MIN() gibt den kleinsten Wert innerhalb einer Argumentliste zurück.

Definition

Zahl1 (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 Zahlen (30 bis Excel 2003), aus denen Sie die kleinste Zahl herausuchen möchten.

Argumente

Das Thema MIN() und MAX() sowie Hinweise zur Verwendung der Argumente haben wir bereits detailliert unter MAX() (siehe Seite 439) erläutert.

Hintergrund

Tipp Häufig soll bei der Ermittlung des kleinsten Werts auch der Wert 0 (Null) ausgeschlossen werden. Um den kleinsten Wert mit einer Bedingung zu ermitteln, verwenden Sie die Formel

`=MIN(Wenn(Bereich<>0;Bereich;""))`

die Sie als Matrixformel mit der Tastenkombination `[Strg] + [↕] + [↩]` eingeben.

Praxiseinsatz Aus einer Umsatztabelle des Softwareherstellers soll der kleinste Umsatz innerhalb des angegebenen Zeitraumes gesucht werden.

1	Fragestellung: Wie hoch war der niedrigste Umsatz?		
2	Monat	Umsatz	Der kleinste Umsatz
3	Januar 07	107.629 €	=MIN(C3:C26) 107.629 €
4	Februar 07	185.385 €	
5	März 07	180.807 €	
6	April 07	124.328 €	
7	Mai 07	146.215 €	
8	Juni 07	185.675 €	
9	Juli 07	210.169 €	
10	August 07	221.729 €	
11	September 07	234.187 €	
12	Oktober 07	237.947 €	
13	November 07	210.088 €	
14	Dezember 07	207.791 €	
15	Januar 08	201.097 €	
16	Februar 08	222.460 €	
17	März 08	208.585 €	
18	April 08	206.387 €	
19	Mai 08	218.951 €	
20	Juni 08	325.698 €	
21	Juli 08	123.984 €	
22	August 08	325.146 €	
23	September 08	251.462 €	
24	Oktober 08	124.652 €	
25	November 08	123.568 €	
26	Dezember 08	301.245 €	

Abbildung 11.81: Mit der Formel *MIN()* wird der niedrigste Wert innerhalb der Tabelle ermittelt

Ohne eine vorangegangene Datensortierung wird durch *MIN()* im Januar mit 107.629 € der kleinste Umsatz festgestellt.

Siehe auch *DBMIN()*, *MAX()*, *MAXA()*, *MINA()*



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Zählen.xlsx* (Excel 2007/2010) auf dem Arbeitsblatt *Max&Min*.

MINA() **MINA()**

Syntax *MINA(Wert1;Wert2;...)*

Definition *MINA()* gibt den kleinsten Wert einer Liste von Argumenten zurück. Text und Wahrheitswerte wie *WAHR* und *FALSCH* werden ebenso wie Zahlen verglichen.

Argumente *Wert1* (erforderlich); *Wert2* (optional); ... sind 1 bis 255 Werte (30 bis Excel 2003), deren kleinster Wert ermittelt werden soll.

Das Thema MAXA() und MINA() sowie Hinweise zur Verwendung der Argumente und deren Bedeutung haben wir bereits detailliert unter MAXA() (siehe Seite 440) erläutert.

Entsprechend unseres Beispiels unter MAXA() erhalten Sie für MINA(), wie in Abbildung 11.82 zu sehen, das Ergebnis 0, da der niedrigste Wert in diesem Fall durch den Wahrheitswert *FALSCH* dargestellt wird.

	B	C
2	Fragestellung: Wie hoch ist der niedrigste Wert der Tabelle?	
3		
4	Monate	Daten
5	Januar	WAHR
6	Februar	0,7
7	März	0,6
8	April	0,4
9	Mai	0,5
10	Juni	0,2
11	Juli	0,3
12	August	0,2
13	September	0,5
14	Oktober	FALSCH
15	November	0,7
16	Dezember	0,3

Abbildung 11.82: Zur Berechnung des niedrigsten Werts einer Datenmenge, die Text oder Wahrheitswerte beinhaltet, verwendet man *MINA()*

DBMIN(), MAXA()

Dieses Beispiel finden Sie im Ordner *\Ms5-235\Kap11* in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *Maxa&Mina*.

Siehe auch



MITTELABW()



MITTELABW(*Zahl1*; *Zahl2*; ...)

MITTELABW() gibt die durchschnittliche absolute Abweichung einer Reihe von Merkmalsausprägungen und ihrem Mittelwert zurück. Anders ausgedrückt: Die Funktion berechnet das arithmetische Mittel der Abweichungen einer Reihe von ihrem Mittelwert ohne Berücksichtigung des Vorzeichens. MITTELABW() ist ein Maß für die Streuung innerhalb einer Datengruppe.

Zahl1 (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 Argumente (30 bis Excel 2003), deren durchschnittliche absolute Abweichung berechnet werden soll. Anstelle der durch ein Semikolon voneinander getrennten Argumente können Sie eine Matrix oder einen Bezug angeben, der auf eine Matrix verweist.

Als Argumente sollten entweder Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten. Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Das Ergebnis der Berechnung von MITTELABW() wird von der Maßeinheit der eingegebenen Daten beeinflusst.

Syntax

Definition

Argumente

Hinweis

Hintergrund Um die Abweichung von beispielsweise Umsätzen oder, wie in unserem Fall, von monatlichen Zugriffen auf die Webseite in Bezug auf deren Mittelwert zu errechnen, steht Ihnen die Funktion MITTELABW() zur Verfügung. Mit dieser Funktion kann ein Maß für die Streuung innerhalb einer Datengruppe berechnet werden.

Streuungsmaße sind gewissermaßen Gütemaße für die Lageparameter. Sie geben an, wie gut bzw. wie schlecht eine Verteilung durch einen Lageparameter gekennzeichnet werden kann. Streuungsparameter beziehen sich auf Differenzen

- ▶ zwischen einzelnen Positionswerten (Spannweite, Quartils- bzw. Semiquartilsabstand)
- ▶ zwischen Einzelwerten und einem Mittelwert (durchschnittliche lineare Abweichung, Varianz, Standardabweichung)

Obwohl das Maß der mittleren Abweichung recht einfach zu berechnen ist, haben sich stattdessen die Standardabweichung und die Varianz durchgesetzt. Grund dafür waren theoretische Überlegungen, die letztlich in der überragenden Bedeutung der Normalverteilung für die schließende Statistik zu sehen ist.

Die Gleichung für die durchschnittliche Abweichung lautet:

$$\frac{1}{n} \sum |x - \bar{x}|$$

Praxiseinsatz Die Marketingabteilung des Softwareherstellers soll die Zugriffe der Kunden auf die Webseite analysieren. Es wurde bereits eine Excel-Tabelle erstellt, in der die Zugriffe auf die verschiedenen Webseitenbereiche für die vergangenen 18 Monate aufgelistet wurden.

	B	C	D	E	F	G	H	I
11	DATUM	ADVENT	DOWNLOAD	EVENT	TRAINING	WISSEN	PRODUKTE	GESAMT
12	Jan 07		6		19	43	22	90
13	Feb 07		498	38	1319	3000	1170	6025
14	Mrz 07		1401	119	2233	6116	1545	11414
15	Apr 07		1076	16	1903	2860	1168	7023
16	Mai 07		1563	15	1589	3126	1139	7432
17	Jun 07		1790	1853	2428	6682	5083	17836
18	Jul 07		1367	1622	1559	3311	3609	11468
19	Aug 07		1138	2170	1502	3317	4810	12937
20	Sep 07		1352	2611	1628	3542	5581	14714
21	Okt 07	1	1343	538	1440	2971	5506	11799
22	Nov 07	272	1430	1254	2255	4735	11786	21732
23	Dez 07	1874	1140	631	1583	2652	6227	14107
24	Jan 08	42	1421	1515	2224	4837	9500	19539
25	Feb 08	32	1508	1433	1418	4006	12000	20397
26	Mrz 08	10	2137	3114	1298	3128	8000	17687
27	Apr 08	10	1948	2347	1085	3575	8208	17173
28	Mai 08	9	1521	848	765	2139	7739	13021
29	Jun 08		426	440	123	768	778	2535
30	Mittelwert	281,25	1281,39	1209,65	1465,06	3378,22	5215,06	12607,17
31	Mittlere Abweichung	398,2	378,3	827,3	472,6	1093,9	3067,9	4785,7

Abbildung 11.83: Mittelwert und dessen mittlere Abweichung für die Webseitenzugriffe

Da sich die mittlere Abweichung auf den Mittelwert der einzelnen Datenreihen bezieht, errechnet die Marketingabteilung zunächst, wie in Abbildung 11.83 dargestellt, den Mittelwert für jeden Webseitenbereich mit der Funktion MITTELWERT(). Anschließend wird für jede der Datenreihen auch die mittlere Abweichung berechnet. Die Ergebnisse, also das arithmetische Mittel der Abweichungen der entsprechenden Reihe von ihrem Mittelwert, liefert die Funktion MITTELABW().

Übersichtlich dargestellt können nun die Mittelwerte und die mittlere Abweichung miteinander verglichen und analysiert werden.

Wie bereits erwähnt, kann mit der Funktion MITTELABW() ein Maß für die Streuung innerhalb einer Datengruppe berechnet werden, wobei sich die Streuungsparameter unter anderem auf Differenzen zwischen Einzelwerten und Mittelwert beziehen. Betrachten wir den Bereich *DOWNLOAD* in Abbildung 11.83, bedeutet das:

Die mittlere Abweichung beträgt im Bezug auf den Mittelwert für jeden einzelnen Monat des Downloadbereichs 378,3. Das heißt, es wurde im Vergleich zum errechneten Mittelwert im Schnitt pro Monat 378,3 mal mehr oder auch weniger auf den Bereich »DOWNLOAD« zugegriffen.

MITTELWERT(), MITTELWERTA(), STABW.S(), STDABWN(), SUMQUADABW(), VAR.S(), VAR.P()

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Varianz.xlsx* auf dem Arbeitsblatt *Mittelabw*.

Aussagen

Siehe auch



MITTELWERT() AVERAGE()

MITTELWERT(*Zahl1*; *Zahl2*; ...)

Der MITTELWERT() wird weitläufig auch als Durchschnitt oder arithmetisches Mittel bezeichnet. Er wird berechnet, indem man alle intervallskalierten Daten summiert und durch deren Anzahl dividiert.

Zahl1 (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 numerische Argumente (30 bis Excel 2003), deren Mittelwert Sie berechnen möchten.

Als Argumente sollten entweder Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Das arithmetische Mittel ist der bekannteste Mittelwert und hat damit auch die größte Akzeptanz unter den Nicht-Statistikern. Da der Mittelwert recht einfach zu berechnen ist und auch auf allen Werten beruht, spielt er in der schließenden Statistik eine wichtige Rolle.

Um den Mittelwert zu errechnen, werden die Einzelwerte eines Wertbereichs addiert und durch die Anzahl der Zahlenwerte dividiert. Die Berechnung des arithmetischen Mittels setzt Intervallskalenniveau voraus. Es ist immer eindeutig zu bestimmen.

Die Formel zur Berechnung des arithmetischen Mittels lautet:

$$\bar{x} = \frac{X_1 + X_2 + \dots + X_n}{n} = \frac{1}{n} \sum X_j$$

Aus dem arithmetischen Mittel zweier Beobachtungsreihen lässt sich ein gemeinsames gewogenes arithmetisches Mittel berechnen.

Der größte Nachteil des arithmetischen Mittels ist, dass durch die Einbeziehung aller Einzelwerte extreme Werte einer Verteilung großes Gewicht bekommen. Außerdem kann dieses Lagemaß auf einem Punkt liegen, für den nur wenige oder gar keine Beobachtungswerte vorliegen.

Syntax

Definition

Argumente

Hinweis

Hintergrund

Bei gruppierten Werten ist das arithmetische Mittel in jedem Fall nur ein Schätzwert, bei offenen Klassengrenzen, also stetigen Variablen, kann es gar nicht berechnet werden, wenn keine Zusatzinformationen für die Schätzung des Klassenmittelpunkts vorliegen.

Obwohl die Berechnung des arithmetischen Mittels metrisches Skalenniveau voraussetzt, ist es üblich, auch von ordinalskalierten Daten Mittelwerte zu berechnen. Dazu gehören beispielsweise Fragen zur Zufriedenheit mit bestimmten Leistungen. Dies ist insofern zulässig, als dass bei genügend großen Stichproben eine Normalverteilung der Daten angenommen werden darf (zentraler Grenzwertsatz) und damit ein Vertrauensintervall angegeben werden kann, in dem der wahre Mittelwert einer Verteilung liegt. Es müssen Mittelwert und Streuung einer genügend großen Stichprobe ($n > 30$) vorliegen.

Bei Daten, die eine Berechnung des arithmetischen Mittels zulassen, können auch Modus und Median bestimmt werden. Welches der drei Lagemaße herangezogen wird, hängt von der Fragestellung ab. Mittelwert, Modus und Median stellen unterschiedliche Informationen in den Vordergrund und nehmen normalerweise auch unterschiedliche Zahlenwerte an. Obwohl das arithmetische Mittel das gebräuchlichste Lagemaß ist, kann es beispielsweise aufgrund des niedrigen Skalenniveaus oder der Anfälligkeit gegenüber Ausreißern sinnvoll sein, mit Modus oder Median zu arbeiten.

Praxiseinsatz Der Softwarehersteller unseres Beispiels gibt der Marketingabteilung den Auftrag, den durchschnittlichen Zugriff pro Webseiten-Bereich im Jahr 2007 zu ermitteln, um auf Basis der Auswertung feststellen zu können, an welchen Bereichen das meiste Interesse besteht. Ferner hat das Unternehmen so die Möglichkeit, weniger gut besuchte Bereiche anzupassen bzw. zu überarbeiten und damit gleichzeitig interessanter zu gestalten.

Der Marketingleiter hat bereits vorgearbeitet und die Webzugriffe aus seinem System exportiert. Die exportierte Datei wiederum hat er in Excel importiert. Hier kann er nun mithilfe einer PivotTable die Zugriffe in den einzelnen Bereichen einfach und unkompliziert auswerten und analysieren.



Die Excel-Datei mit den Webzugriffen innerhalb einer PivotTable finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Webzugriff.xlsx* (Excel 2007-2013).

	B	C	D	E	F	G	H
8	Fragestellung: Wie war der durchschnittliche Zugriff auf die Homepage?						
9	ZUGRIFFE 2007	PRODUKTE	PUBLIKATIONEN	TEAM	TRAINING	WISSEN	GESAMTERGEBNIS
10	Januar	22	2	19	19	43	105
11	Februar	1.170	356	505	1.319	3.000	6.350
12	März	1.545	756	1.307	2.233	6.116	11.957
13	April	1.168	518	884	1.903	2.860	7.333
14	Mai	1.139	684	835	1.589	3.126	7.373
15	Juni	5.083	1.678	820	2.428	6.682	16.691
16	Juli	3.609	844	276	1.559	3.311	9.599
17	August	4.810	622	365	1.502	3.317	10.616
18	September	5.581	893	309	1.628	3.542	11.953
19	Oktober	5.506	724	259	1.440	2.971	10.900
20	November	11.786	1.046	447	2.255	4.735	20.269
21	Dezember	6.227	339	234	1.583	2.652	11.035
22	Mittelwert	3.971	705	522	1.622	3.530	10.348

Abbildung 11.84: Berechnung des durchschnittlichen Zugriffs pro Webseitenbereich mit *MITTELWERT()*

Den Marketingleiter interessieren die Bereiche »Produkte«, »Publikationen«, »Team«, »Training« und »Wissen« sowie das Gesamtergebnis. Er errechnet zunächst für jeden Bereich mithilfe der Funktion MITTELWERT() den durchschnittlichen Zugriff und kann damit schon Aussagen über die Beliebtheit der einzelnen Bereiche treffen. Denn wie Sie der Abbildung 11.84 entnehmen können, wurde auf den Bereich »Publikationen« im Durchschnitt entschieden seltener zugegriffen als auf den Bereich »Produkte«.

Würde man hier anstatt der Funktion MITTELWERT() die Funktion GESTUTZTMITTEL() verwenden, wäre das Ergebnis für den durchschnittlichen Zugriff im Produktbereich höher, da bei der Errechnung des gestutzten Mittels, je nach Vorgabe, ein bestimmter Prozentsatz an Werten nicht in die Berechnung des Mittels einer Datenreihe mit eingehen. Dieser Prozentsatz an Werten wird der sortierten Datenreihe von oben und unten abgezogen.

Hinweis

Zum Schluss kann der Marketingleiter noch den durchschnittlichen Webseitenzugriff für alle Bereiche innerhalb eines Jahres berechnen und erhält damit fundierten Überblick über die Geschehnisse auf der Webseite. Liegen ihm nun beispielsweise noch Vergleichswerte aus dem Vorjahr vor, können weitere Analysen vorgenommen werden.

Sollten Sie aus nicht zusammenhängenden Bereichen den Mittelwert bilden wollen, trennen Sie die Bereiche mit einem Semikolon. Zum Beispiel MITTELWERT(*Bereich 1; Bereich 2;...Bereich n*).

Hinweis

Beachten Sie aber, dass eine Tabellenfunktion maximal 255 Argumente aufnehmen kann. Um mehrere nicht zusammenhängende Bereiche zu markieren, können Sie auch die **[Strg]**-Taste gedrückt halten, während Sie die Zellen anklicken.

GEOMITTEL(), GESTUTZTMITTEL(), HAARMITTEL(), MEDIAN(), MITTELABW(), MITTELWERTA(), MITTELWERTWENN(), MITTELWERTWENNS(), MODUS.EINF()

Siehe auch

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Mittelwert*.



MITTELWERTA() AVERAGEA()

MITTELWERTA(*Wert1;Wert2;...*)

Syntax

Die Funktion MITTELWERTA() berechnet den Mittelwert der Werte in einer Liste mit Argumenten. Bei dieser Berechnung werden neben Zahlen auch Text und Wahrheitswerte wie *WAHR* und *FALSCH* berücksichtigt.

Definition

Wert1 (erforderlich); *Wert2* (optional); ... sind 1 bis 255 (30 bis Excel 2003) Zellen, Zellbereiche oder Werte, für die der Mittelwert berechnet werden soll.

Argumente

Zulässige Argumente sind Zahlen, Namen, Matrizen oder Zellbezüge.

Hinweis

Wenn Sie aus den Inhalten mehrerer Zellen einen Mittelwert bilden wollen, vergessen Sie nicht, dass leere Zellen anders behandelt werden als Zellen, die mit 0 belegt sind.

Tip

Dies gilt insbesondere dann, wenn Sie in Excel 2013 (und 2010) auf der Registerkarte *DATEI* den Befehl *Optionen* (bzw. in Excel 2007 auf die *Office-Schaltfläche* und anschließend auf *Excel-Optionen*) klicken und dann im Dialogfeld *Excel-Optionen* in der Kategorie *Erweitert* das Kontrollkästchen *In Zellen mit Nullwert eine Null anzeigen* aktivieren.

Im Gegensatz zu leeren Zellen werden mit 0 belegte Zellen berücksichtigt.

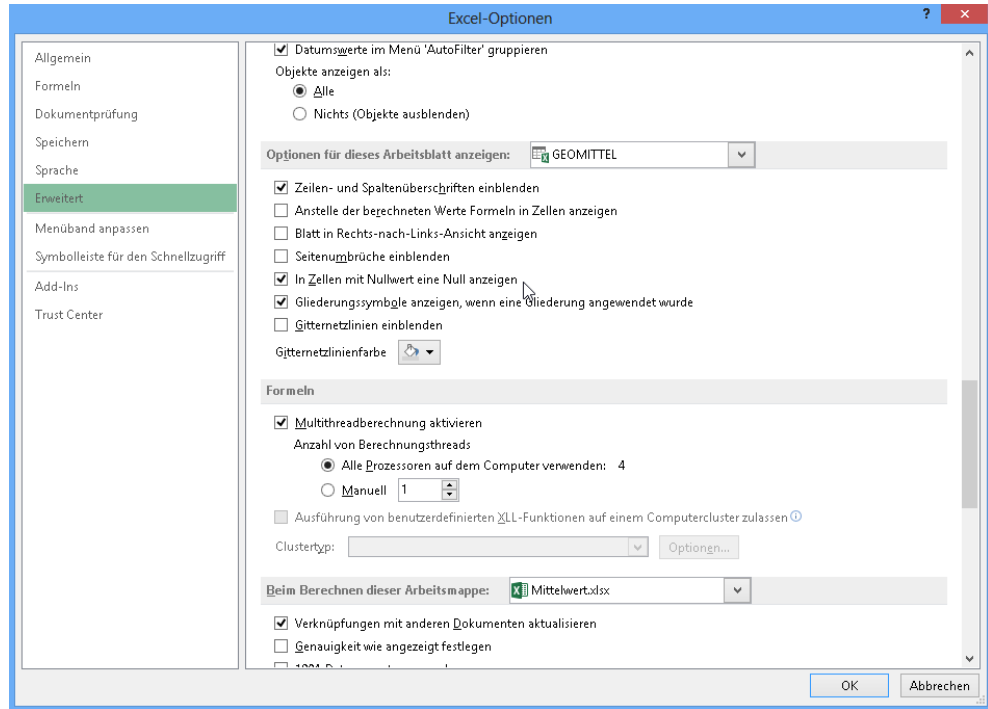


Abbildung 11.85: Zellen mit einer Null belegen

Hintergrund Detaillierte Hintergrundinformationen zum Mittelwert finden Sie unter MITTELWERT() auf Seite 447.

Für die Funktion MITTELWERTA() gilt zusätzlich: Sind in der Liste mit Argumenten auch Argumente, die Text enthalten, als Matrix oder Bezug angegeben, werden diese mit dem Wert 0 berechnet. Argumente, die WAHR enthalten, evaluieren zu 1, Argumente, die FALSCH enthalten, zu 0.

Sollen bei der Berechnung des Mittelwerts Textwerte nicht berücksichtigt werden, verwenden Sie die Funktion MITTELWERT().

Praxiseinsatz Ein Mitarbeiter aus der Controllingabteilung des Softwareherstellers hat eine Excel-Tabelle mit den Umsätzen der vergangenen zwölf Monate erfasst. Da in der Liste auch Text- und Wahrheitswerte erfasst sind, will er nun mithilfe der Funktion MITTELWERTA() den durchschnittlichen Umsatz errechnen.

Wie Sie in Abbildung 11.86 sehen können, hat der Controllingmitarbeiter im ersten Fall für den Februar den Text »Geschlossen« erfasst. Da die Funktion MITTELWERTA() Text automatisch in den Wert 0 umwandelt, werden alle zwölf Werte addiert und durch den Wert 12 geteilt. Das Ergebnis beträgt hier 916,67 €.

	B	C	D
7	UMSÄTZE	MIT TEXT	OHNE TEXT
8	Januar	1.000 €	1.000 €
9	Februar	Geschlossen	WAHR
10	März	1.000 €	1.000 €
11	April	1.000 €	1.000 €
12	Mai	1.000 €	1.000 €
13	Juni	1.000 €	1.000 €
14	Juli	1.000 €	1.000 €
15	August	1.000 €	1.000 €
16	September	1.000 €	1.000 €
17	Oktober	1.000 €	1.000 €
18	November	1.000 €	1.000 €
19	Dezember	1.000 €	1.000 €
20	MITTELWERT	1.000,00 €	1.000,00 €
21	MITTELWERTA	916,67 €	916,75 €

Abbildung 11.86: Bei der Berechnung von *MITTELWERTA()* werden auch Text- und Wahrheitswerte erfasst

In der zweiten Spalte erfasst der Mitarbeiter anstatt des Texts »Geschlossen« den Wert *WAHR*. Dieser Wahrheitswert wird bei der Berechnung von *MITTELWERTA()* zu 1 evaluiert. Somit werden auch hier alle zwölf Werte addiert und durch den Wert 12 geteilt. Das Ergebnis lautet dann 916,75 €.

Würde man die Funktion *MITTELWERT()*, die Textwerte nicht berücksichtigt, anwenden, würden nur die verbleibenden elf Einträge summiert und durch den Wert 11 geteilt. Wie Sie in *Abbildung 11.86* sehen können, beträgt das Ergebnis dann 1.000 €.

MITTELWERTABW(), *MITTELWERT()*

Dieses Beispiel finden Sie im Ordner *\Ms5-235\Kap11* in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *MittelwertA*.

Siehe auch



MITTELWERTWENN()



AVERAGEIF()

MITTELWERTWENN(Bereich; Kriterien; Mittelwert_Bereich)

MITTELWERTWENN() ist eine neue Funktion in Excel 2007!

Die Funktion *MITTELWERTWENN()* berechnet den Durchschnittswert für alle Zellen in einem Bereich, die einem angegebenen Kriterium entsprechen.

Bereich (erforderlich) entspricht mindestens einer Zelle, für die der Durchschnitt einschließlich der Zahlen oder Namen, Arrays oder Verweise mit Zahlen ermittelt werden soll.

Kriterien (erforderlich) entspricht den Kriterien in Form einer Zahl, eines Ausdrucks, Zellbezugs oder Texts, mit dem definiert wird, für welche Zellen der Durchschnitt ermittelt wird.

Mittelwert_Bereich (optional) entspricht den Zellen, für die tatsächlich der Durchschnitt ermittelt werden soll. Bei Auslassung wird der Bereich verwendet.

Neu in Excel 2007

Syntax

Hinweis

Definition

Argumente

Hinweis Zellen im *Bereich*, die WAHR oder FALSCH enthalten, werden ignoriert.

Wenn eine Zelle in *Mittelwert_Bereich* eine leere Zelle ist, wird sie von MITTELWERTWENN() ignoriert.

Wenn *Bereich* leer ist oder Textwerte enthält, wird von MITTELWERTWENN() der Fehlerwert #DIV/0! zurückgegeben.

Wenn eine Zelle in *Kriterien* leer ist, wird sie von MITTELWERTWENN() wie ein Nullwert behandelt.

Wenn keine Zellen im Bereich den *Kriterien* entsprechen, gibt MITTELWERTWENN() den Fehlerwert #DIV/0! zurück.

Die *Kriterien* dürfen die Platzhalter »Fragezeichen (?)« und »Sternchen (*)« enthalten. Ein Fragezeichen ersetzt bei Übereinstimmung ein einzelnes Zeichen und ein Sternchen eine beliebige Zeichenfolge. Wenn Sie nach einem tatsächlichen Fragezeichen oder Sternchen suchen möchten, geben Sie eine »Tilde (~)« vor dem zu suchenden Zeichen ein.

Mittelwert_Bereich muss nicht mit der Größe und der Form des Arguments *Bereich* übereinstimmen. Die tatsächlich zu mittelnden Zellen werden bestimmt, indem die obere linke Zelle in *Mittelwert_Bereich* als Anfangszelle verwendet wird und dann die Zellen aufgenommen werden, deren Größe und Form dem Argument *Bereich* entsprechen.

Hintergrund Detaillierte Hintergrundinformationen zum Mittelwert finden Sie unter MITTELWERT() auf Seite 447.

Mit der Funktion MITTELWERTWENN() wird die zentrale Tendenz gemessen. Dabei handelt es sich um die Stelle im Zentrum einer Zahlengruppe bei einer statistischen Verteilung.

Für die Funktion MITTELWERTWENN() gilt zusätzlich die Angabe eines bestimmten Kriteriums, das ausschlaggebend für die Berechnung des gesuchten Mittelwerts ist.

Folgende Maße der zentralen Tendenz sind am gebräuchlichsten:

- ▶ Mittelwert: arithmetisches Mittel der Verteilung (siehe Seite 447)
- ▶ Median: Wert in der Mitte einer Zahlengruppe (siehe Seite 441)
- ▶ Modalwert (MODUS.EINF() ab 2010): die am häufigsten vorkommende Zahl in einer Zahlengruppe (siehe Seite 457)

Bei einer symmetrischen Verteilung einer Zahlengruppe sind diese drei Maße der zentralen Tendenz identisch. Bei einer schiefen Verteilung einer Zahlengruppe können die Maße abweichen.

Praxiseinsatz Als Softwarehersteller müssen Sie Ihre Software auch vermarkten – hierfür gehen Sie verschiedene Wege. Unter anderem verschicken Sie regelmäßig Informationen über Ihre Produkte per E-Mail.

In der Regel bewerben Sie eine Software pro Newsletter, um zum einen nicht zu viel Inhalt zu haben und zum anderen besser auswerten zu können, wie sich die Klickrate auf Ihrer Webseite verändert, nachdem ein bestimmtes Mailing rausgegangen ist.

Es wurden die Daten der letzten 30 Monate gesammelt – das heißt, Ihnen liegt eine Liste mit den Klicks pro Webseite in einem bestimmten Monat sowie die Art des Mailings, dass in diesem Monat gemacht wurde, vor.

	B	C	D
7	Monat	Mailing-Aktion	Webzugriffe
8	Januar 2006	Software A	236
9	Februar 2006	Software B	1.221
10	März 2006	Software C	6.195
11	April 2006	Allgemein	2.682
12	Mai 2006	Software A	4.569
13	Juni 2006	Software B	12.229
14	Juli 2006	Software C	12.564
15	August 2006	Allgemein	10.157
16	September 2006	Software A	15.160
17	Oktober 2006	Software B	10.716

Abbildung 11.87: Auszug aus der auszuwertenden Klickliste nach Monat und Mailingaktion

Sie möchten nun wissen, wie hoch die durchschnittliche Klickrate auf Ihre Webseite nach den einzelnen bzw. unterschiedlichen Mailingaktionen war. Die Frage muss also lauten: Wie hoch ist der Mittelwert der Webseiten-Klickrate ausschließlich für die Mailingaktion »Software B«? Hierfür verwenden Sie die Funktion MITTELWERTWENN().

Die Formel hierfür entspricht Abbildung 11.88.

	B	C	D	E
7	Monat	Mailing-Aktion	Webzugriffe	
8	Januar 2006	Software A	236	
9	Februar 2006	Software B	1.221	
10	März 2006	Software C	6.195	
11	April 2006	Allgemein	2.682	
12	Mai 2006	Software A	4.569	
13	Juni 2006	Software B	12.229	
14	Juli 2006	Software C	12.564	
15	August 2006	Allgemein	10.157	
16	September 2006	Software A	15.160	
17	Oktober 2006	Software B	10.716	
18	November 2006	Software C	9.844	
19	Dezember 2006	Allgemein	11.975	
20	Januar 2007	Software A	24.695	
21	Februar 2007	Software B	11.593	
22	März 2007	Software C	18.491	
23	April 2007	Allgemein	11.743	
24	Mai 2007	Software A	11.452	
25	Juni 2007	Software B	26.651	
26	Juli 2007	Software C	16.287	
27	August 2007	Allgemein	17.750	
28	September 2007	Software A	19.985	
29	Oktober 2007	Software B	17.285	
30	November 2007	Software C	30.369	
31	Dezember 2007	Allgemein	19.674	
32	Januar 2008	Software A	28.464	
33	Februar 2008	Software B	25.000	
34	März 2008	Software C	24.574	
35	April 2008	Allgemein	23.141	
36	Mai 2008	Software A	17.700	
37	Juni 2008	Software B	22.456	
38			15.894	

Abbildung 11.88: Berechnung der durchschnittlichen Klickraten nach Versand des Mailings zu Software B

Wie Sie der Formel in Abbildung 11.88 entnehmen können, werden folgende Angaben für die Argumente der Formel verwendet:

- ▶ Kriterien: Zellen der Spalte C von C8:C37, da diese Spalte die Kriterien angibt, die generell als Basis für die Auswertung gelten sollen
- ▶ Bereich: Als Bereich wird der Ausdruck »Software B« angegeben, da Sie ausschließlich den Mittelwert für die Webseiten-Klickrate nach Versand eines Mailings zur *Software B* wissen möchten
- ▶ *Mittelwert_Bereich*: Hier geben Sie die Spalte D, also in unserem Beispiel die Zellen D8:D37 an, da aus diesen Zellen, entsprechend des angegebenen Kriteriums, der Mittelwert berechnet wird

Aussage Folgende Aussage kann nun getroffen werden:

Die durchschnittliche Klickrate nach Versand eines Mailings zu Software B lag in den letzten 30 Monaten bei 15.894 Klicks.

Berechnen Sie nun noch die durchschnittlichen Klickraten nach Versand der Mailings zu Software A, Software C und den allgemeinen Mailings und zusätzlich noch den normalen Mittelwert, sehen Sie, dass die durchschnittliche Klickrate nach der Mailingaktion zu Software A am höchsten ist. Somit ist diese Software möglicherweise am interessantesten für Ihre Kunden.

	C	D
38	Software B	15.894
39	Software A	17.432
40	Software C	16.903
41	Allgemein	13.875
42	Mittelwert gesamt	15.495

Abbildung 11.89: Berechnung aller Mittelwerte

Siehe auch MITTELWERT(), MITTELWERTWENNS(), MEDIAN(), MODUS.EINF(), ZÄHLENWENN()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Mittelwertwenn*.

Neu in Excel
2007

MITTELWERTWENNS()



AVERAGEIFS()

Syntax MITTELWERTWENNS(*Mittelwert_Bereich*; *Kriterien_Bereich1*; *Kriterien1*; *Kriterien_Bereich2*; *Kriterien2*...)

Hinweis MITTELWERTWENNS() ist eine neue Funktion ab Excel 2007!

Definition Die Funktion MITTELWERTWENNS() gibt den Durchschnittswert aller Zellen zurück, die mehreren Kriterien entsprechen.

Argumente *Mittelwert_Bereich* (erforderlich) entspricht mindestens einer Zelle, deren Durchschnitt einschließlich Zahlen oder Namen, Arrays oder Verweise mit Zahlen zu bestimmen ist.

Kriterien_Bereich1 (erforderlich) ist der Bereich, in dem die zugeordneten Kriterien zu bestimmen sind.

Kriterien1 (erforderlich) ist das Kriterium in Form einer Zahl, eines Ausdrucks, eines Zellenverweises oder Texts, das die Zellen definiert, deren Durchschnitt bestimmt werden soll.

Kriterien_Bereich2 (optional) sind die Bereiche 1 bis 127, in denen die zugeordneten Kriterien zu bestimmen sind.

Kriterien2 (optional) sind 1 bis 127 Kriterien in Form einer Zahl, eines Ausdrucks, eines Zellenverweises oder Texts, mit denen definiert wird, von welchen Zellen der Durchschnitt bestimmt werden soll.

Wenn *Mittelwert_Bereich* leer oder ein Textwert ist, wird von MITTELWERTWENNS() der Fehlerwert #DIV/0! zurückgegeben.

Hinweis

Wenn eine Zelle in einem *Kriterienbereich* leer ist, wird sie von MITTELWERTWENNS() wie ein Nullwert behandelt.

Zellen in einem Bereich, die den Wert WAHR enthalten, ergeben 1. Zellen im Bereich, die den Wert FALSCH enthalten, ergeben Null.

Die einzelnen Zellen in *Mittelwert_Bereich* werden nur in der Durchschnittsberechnung verwendet, wenn alle entsprechenden angegebenen Kriterien für diese Zelle zutreffen.

Im Gegensatz zu den Argumenten *Bereich* und *Kriterien* der Funktion MITTELWERTWENN() müssen bei MITTELWERTWENNS() alle *Kriterien_Bereich* über dieselbe Größe und Form wie *Summe_Bereich* verfügen.

Wenn Zellen in *Mittelwert_Bereich* nicht in Zahlen übersetzt werden können, wird von MITTELWERTWENNS() der Fehlerwert #DIV/0! zurückgegeben.

Wenn keine Zellen vorhanden sind, die allen Kriterien entsprechen, gibt MITTELWERTWENNS() den Fehlerwert #DIV/0! zurück.

Die Kriterien dürfen die Platzhalterzeichen »Fragezeichen (?)« und »Sternchen (*)« enthalten. Ein Fragezeichen ersetzt bei Übereinstimmung ein einzelnes Zeichen und ein Sternchen eine beliebige Zeichenfolge. Wenn Sie nach einem tatsächlichen Fragezeichen oder Sternchen suchen möchten, geben Sie eine Tilde (~) vor dem zu suchenden Zeichen ein.

Detaillierte Hintergrundinformationen zum Mittelwert finden Sie unter MITTELWERT() auf Seite 447. Wie bei der Funktion MITTELWERTWENN() wird auch mit der Funktion MITTELWERTWENNS() die zentrale Tendenz gemessen.

Hintergrund

Für die Funktion MITTELWERTWENNS() gilt zusätzlich die Angabe mehrerer bestimmter Kriterien, die ausschlaggebend für die Berechnung des gesuchten Mittelwerts sind. Weitere Informationen hierzu lesen Sie unter MITTELWERTWENN() auf Seite 451.

Wir bleiben bei dem Beispiel mit dem Mailingversand des Software-Herstellers. Wollte er bei der Funktion MITTELWERTWENN() noch wissen, wie hoch der durchschnittliche Zugriff auf die Webseite nach einer bestimmten Mailingaktion ist, möchte er nun ein weiteres Kriterium angeben, das besagt, dass ausschließlich die Klickraten in die Berechnung des Mittelwerts eingehen sollen, die größer 10.000 sind.

Praxiseinsatz

So kann er sicher sein, dass Ausreißer nach unten – möglicherweise bedingt durch Urlaubsmonate – sein Ergebnis nicht verfälschen.

		D39 fx =MITTELWERTWENNS(D9:D38;C9:C38;"Software A";D9:D38;">10000")				
	B	C	D	E	F	G
8	Monat	Mailing-Aktion	Webzugriffe			
9	Januar 2006	Software A	236			
10	Februar 2006	Software B	1.221			
11	März 2006	Software C	6.195			
12	April 2006	Allgemein	2.682			
13	Mai 2006	Software A	4.569			
14	Juni 2006	Software B	12.229			
15	Juli 2006	Software C	12.564			
16	August 2006	Allgemein	10.157			
17	September 2006	Software A	15.160			
18	Oktober 2006	Software B	10.716			
19	November 2006	Software C	9.844			
20	Dezember 2006	Allgemein	11.975			
21	Januar 2007	Software A	24.695			
22	Februar 2007	Software B	11.593			
23	März 2007	Software C	18.491			
24	April 2007	Allgemein	11.743			
25	Mai 2007	Software A	11.452			
26	Juni 2007	Software B	26.651			
27	Juli 2007	Software C	16.287			
28	August 2007	Allgemein	17.750			
29	September 2007	Software A	19.985			
30	Oktober 2007	Software B	17.285			
31	November 2007	Software C	30.369			
32	Dezember 2007	Allgemein	19.674			
33	Januar 2008	Software A	28.464			
34	Februar 2008	Software B	25.000			
35	März 2008	Software C	24.574			
36	April 2008	Allgemein	23.141			
37	Mai 2008	Software A	17.700			
38	Juni 2008	Software B	22.456			
39			19.576			

Abbildung 11.90: Berechnung des Mittelwerts durch MITTELWERTWENNS()

Wie Sie der Formel in Abbildung 11.90 entnehmen können, werden folgende Angaben für die Argumente der Formel verwendet:

- ▶ **Mittelwert_Bereich:** Zellen der Spalte D von D9:D38, da diese Spalte die Werte beinhaltet, aus denen der Mittelwert berechnet werden soll
- ▶ **Kriterien_Bereich1:** Definiert den ersten Kriterienbereich. Hier wird die Spalte C, also in unserem Beispiel die Zellen C9:C38, angegeben, da diese Zellen Kriterien beinhalten, nach denen gefiltert werden kann.
- ▶ **Kriterien1:** Definiert das erste Kriterium aus dem Kriterienbereich 1. Für Kriterien1 wird der Ausdruck »Software A« angegeben, da Sie ausschließlich den Mittelwert für die Webseiten-Klickrate nach Versand eines Mailings zur *Software A* wissen möchten.

- ▶ **Kriterien_Bereich2:** Definiert den zweiten Kriterienbereich. Hier wird die Spalte D, also in unserem Beispiel die Zellen D9:D38, angegeben, da diese Zellen Kriterien beinhaltet, nach denen gefiltert werden kann.
- ▶ **Kriterien2:** Definiert das zweite Kriterium aus dem Kriterienbereich 2. Für **Kriterien2** wird die Bedingung »>10000« angegeben, da Sie zusätzlich ausschließlich den Mittelwert für Webseiten-Klickraten, die größer als 10.000 sind, wissen möchten.

Folgende Aussage kann nun getroffen werden:

Die durchschnittliche Klickrate nach Versand eines Mailings zu Software A, nur unter Anbetracht der Klickraten größer als 10.000, lag in den letzten 30 Monaten bei 19.576 Klicks.

Berechnen Sie nun noch die durchschnittlichen Klickraten von Software B und Software C nach Versand des Mailings. Sie sehen nun, dass die durchschnittliche Klickrate nach der Mailingaktion zu Software C am höchsten ist.

	B	C	D
39	Software A, Zugriff > 10.000		19.576
40	Software B, Zugriff > 10.000		17.990
41	Software C, Zugriff > 10.000		20.457

Abbildung 11.91: Berechnung weiterer Mittelwerte unter Angabe verschiedener Kriterien

MITTELWERT(), MITTELWERTWENN(), MEDIAN(), MODUS.EINF(), ZÄHLENWENN()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Mittelwertwenns*.

Aussage

Siehe auch



MODUS.EINF() / MODALWERT()

MODE.SINGLE() / MODE()

MODUS.EINF(*Zahl1*; *Zahl2*; ...)

Die Funktion MODUS.EINF() gibt den häufigsten Wert einer Datengruppe zurück. Das Ergebnis der Funktion ist somit die Merkmalsausprägung, die in einer Datenmenge am häufigsten vertreten ist. Genauso wie MEDIAN() ist MODUS.EINF() ein Maß für die Lage der Werte.

Zahl1 (erforderlich); **Zahl2** (optional); ... sind 1 bis 255 Argumente (30 bis Excel 2003), für die Sie den Modalwert (Modus) berechnen möchten. An Stelle der durch Semikola getrennten Argumente können Sie auch eine Matrix oder einen Bezug auf eine Matrix verwenden.

Als Argumente sollten Zahlen, Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Enthält die jeweilige Datengruppe keine mehrfach vorkommenden Datenpunkte, gibt MODUS.EINF() den Fehlerwert #NV zurück.

In einer Wertemenge ist der Modalwert der am häufigsten vorkommende Wert, der Median ist der in der Mitte liegende Wert, und der Mittelwert ist der Durchschnitt. Kein einzelnes Maß der Zentraltendenz bietet ein vollständiges Bild der jeweiligen Daten.

Syntax

Definition

Argumente

Hinweis

Hintergrund Neben der Gesamtzahl aller Fälle interessiert auch, wie oft ein ganz bestimmter Wert in einem Bereich auftritt. Die Tabellenfunktion MODUS.EINF() liefert diese Merkmalsausprägung.

Angenommen, bestimmte Daten fallen hauptsächlich in zwei Bereiche, wobei sich die eine Hälfte der Daten um einen niedrigen Wert und die andere Hälfte um einen großen Wert gruppiert. Die Funktionen MITTELWERT() und MEDIAN() liefern einen Wert, der in der ziemlich leeren Mitte liegt, während MODUS.EINF() den häufigsten der Werte liefert.

Praxiseinsatz Der Leiter der Außendienstmitarbeiter des Softwareherstellers möchte die Anzahl der Besuche seiner Mitarbeiter in den Bundesländern Bayern, Hessen, Berlin, Bremen und der Pfalz auswerten und hat sich dazu bereits eine entsprechende Tabelle erstellt (siehe Abbildung 11.92).

	B	C	D	E	F	G
9	Anzahl der Außendienstbesuche					
10	Besuche	Bayern	Hessen	Berlin	Bremen	Pfalz
11	Montag	4	3	6	4	4
12	Dienstag	7	3	4	4	3
13	Mittwoch	2	2	4	6	7
14	Donnerstag	4	4	4	4	5
15	Freitag	1	1	4	7	8
16	Samstag	2	3	4	6	6
17						
18		4		=MODUS.EINF(C11:G16)		
19		12		=ZÄHLENWENN(C11:G16;C18)		

Abbildung 11.92: Die Berechnung des Modalwerts aus einer Datenliste

Er möchte nach der Auswertung eine Aussage darüber treffen können, wie viele Außendienstbesuche in der Regel erforderlich waren, bis der Vertragsabschluss unter Dach und Fach war.

Deshalb bedient er sich der Funktion MODUS.EINF().

Wie Sie in Abbildung 11.92 erkennen können, waren meistens vier Besuche notwendig, um einen Vertragsabschluss mit dem Kunden zu erreichen. Mit einer verschachtelten Funktion aus den Funktionen ZÄHLENWENN() und MODUS.EINF() können Sie zusätzlich noch zählen, wie oft der Modalwert im entsprechenden Wertebereich vorkommt.

Ebenso können Sie über die bedingte Formatierung die Zellen, die den Modalwert beinhalten, farblich hervorheben. Dazu markieren Sie den Datenbereich von C11:G16 und wählen in Excel 2007-2013 die Registerkarte *START*. Klicken Sie in der Gruppe *Formatvorlagen* auf die Schaltfläche *Bedingte Formatierung*.

Stellen Sie die Bedingung 1 auf den Eintrag *Formel ist* und geben Sie im Eingabefeld die Formel =MODUS.EINF(\$C\$11:\$G\$16)=C11 ein. Beachten Sie dabei die gemischten absoluten und relativen Bezüge. Wählen Sie die Schaltfläche *Format* und legen Sie auf der Registerkarte *Muster* eine Farbe fest. Schließen Sie die Eingabe jeweils mit *OK* ab. Die Werte des Datenbereichs, die dem Modalwert entsprechen, werden daraufhin im gewählten Muster dargestellt.

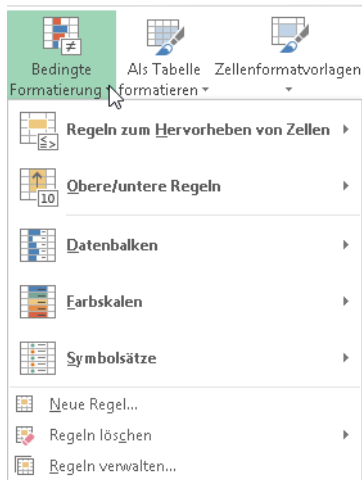


Abbildung 11.93: Bedingte Formatierung in Excel 2007 bis 2013

GEOMITTEL(), GESTUTZTMITTEL(), HARMITTEL(), MEDIAN(), MITTELWERT(), MITTELWERTWENN(), MITTELWERTWENNS()

Siehe auch

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Modus.einf*.



MODUS.VIELF()



MODE.MULT()

MODUS.VIELF(*Zahl1*; *Zahl2*; ...)

Syntax

Die ab der Version 2010 vorhandene, neue Funktion MODUS.VIELF() gibt ein vertikales Array der am häufigsten vorkommenden oder wiederholten Werte in einem Array oder Datenbereich zurück. Verwenden Sie für ein horizontales Array MTRANS (MODUS.VIELF(*Zahl1*, *Zahl2*, ...)).

Definition

Gibt es mehrere Modalwerte, werden mehrere Ergebnisse zurückgegeben. Da diese Funktion ein Array von Werten zurückgibt, muss die Formel als Arrayformel eingegeben werden.

Zahl1 (erforderlich) ist das erste numerische Argument, für das der Modalwert (Modus) berechnet werden soll.

Argumente

Zahl2 (optional); ... sind 2 bis 254 numerische Argumente, für die Sie den Modalwert (Modus) berechnen möchten. An Stelle der durch Semikolons getrennten Argumente können Sie auch ein Array oder einen Arraybezug verwenden.

Als Argumente können entweder Zahlen oder Namen, Arrays oder Bezüge angegeben werden, die Zahlen enthalten. Enthält ein als Array oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Hinweis

Als Fehlerwerte oder Text angegebene Argumente, die nicht in Zahlen umgewandelt werden können, führen zu Fehlern.

Enthält das jeweilige Dataset keine mehrfach vorkommenden Datenpunkte, gibt MODUS.VIELF() den Fehlerwert #NV zurück.

Hintergrund Mehr Informationen zum Modalwert (Modus) erfahren Sie im Abschnitt `MODUS.EINF()` / `MODALWERT()` auf Seite 457.

Praxiseinsatz Vergleichen Sie das Beispiel zur Funktion `MODUS.EINF()` auf Seite 457.

Siehe auch `GEOMITTEL()`, `GESTUTZTMITTEL()`, `HARMITTEL()`, `MEDIAN()`, `MITTELWERT()`, `MITTELWERTWENN()`, `MITTELWERTWENNS()`



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Modus.vielf*.

NEGBINOM.VERT() / NEGBINOMVERT()

NEGBINOM.DIST() / NEGBINOMDIST()

Syntax `NEGBINOM.VERT(Zahl_Misserfolge;Zahl_Erfolge;Erfolgswahrsch)`

Definition Die Funktion `NEGBINOM.VERT()` gibt Wahrscheinlichkeiten einer negativbinomialverteilten Zufallsvariablen zurück. `NEGBINOM.VERT()` berechnet, wie wahrscheinlich es ist, dass es genau *Zahl_Misserfolge* gibt, bevor der letzte positive Ausgang (*Zahl_Erfolge*) gezogen wird, wenn *Erfolgswahrsch* die gleichbleibende Wahrscheinlichkeit eines Erfolgs angibt.

Argumente *Zahl_Misserfolge* (erforderlich) ist die Zahl der ungünstigen Ereignisse.

Zahl_Erfolge (erforderlich) ist die Zahl der günstigen Ereignisse.

Erfolgswahrsch (erforderlich) ist die Wahrscheinlichkeit für den günstigen Ausgang des Experiments.

Hinweis *Zahl_Misserfolge* und *Zahl_Erfolge* werden durch Abschneiden ihrer Nachkommastellen in ganze Zahlen überführt.

Ist eines der Argumente kein numerischer Ausdruck, gibt die Funktion `NEGBINOM.VERT()` den Fehlerwert `#WERT!` zurück.

Ist *Erfolgswahrsch* kleiner 0 oder ist *Erfolgswahrsch* größer 1, gibt `NEGBINOM.VERT()` den Fehlerwert `#ZAHL!` zurück.

Ist *Zahl_Misserfolge* kleiner 0 bzw. *Zahl_Erfolge* kleiner 1, gibt `NEGBINOM.VERT()` den Fehlerwert `#ZAHL!` zurück.

Hintergrund Die Vorgehensweise dieser Funktion unterscheidet sich von der Binomialverteilung nur dadurch, dass die Anzahl der Erfolge feststeht und die Anzahl der Versuche variabel ist. Es handelt sich hierbei um eine negative Binomialverteilung. Analog zu einer Binomialverteilung wird vorausgesetzt, dass die jeweiligen Versuche voneinander unabhängig sind.

Hinweis Mehr Informationen zur Binomialverteilung finden Sie unter `BINOM.VERT()` auf Seite 362.

Die negative Binomialverteilung, auch Pascal-Verteilung genannt, liefert also die Wahrscheinlichkeit für das Erhalten von einer festen Anzahl von Misserfolgen vor dem X-ten Erfolg bei einem Zufallsexperiment mit voneinander unabhängigen Wiederholungen und nur zwei möglichen Ergebnissen (Erfolg oder Misserfolg).

Die Wahrscheinlichkeit für einen Erfolg pro Wiederholung entspricht der Erfolgswahrscheinlichkeit.

Die Gleichung einer negativen Binomialverteilung lautet:

$$nb(x, r, p) = \binom{x+r-1}{r-1} p(1-p)^x$$

Wobei:

- ▶ x ist gleich *Zahl_Misserfolge*
- ▶ r ist gleich *Zahl_Erfolge*
- ▶ p ist gleich *Erfolgswahrsch*

Sie sind im Urlaub in einer fremden Stadt, suchen den Weg zu einer bestimmten Sehenswürdigkeit und fragen zufällig ausgewählte Passanten nach der entsprechenden Richtung. Ihre Frage kann nur zwei Antworten zur Folge haben: Ja oder nein. Das heißt, die Wahrscheinlichkeit, dass Sie ein »Ja« zu hören bekommen, beträgt 50 %. Daraus ergibt sich ein p von 0,5.

Nachdem Sie einige Fußgänger befragt haben und keiner Ihnen richtig helfen konnte, kaufen Sie sich einen Stadtplan. Der Weg zur Sehenswürdigkeit ist nun klar.

Jetzt möchten Sie wissen, wie hoch die Wahrscheinlichkeit ist, dass Sie fünf Personen treffen, die Ihnen auf Anhieb den richtigen Weg sagen können bzw. wie viele »Es tut mir leid, ich weiß den Weg nicht« Sie zu hören bekommen, bevor Sie diese fünf Personen mit der richtigen Erklärung treffen.

Die gesuchte Wahrscheinlichkeit berechnen Sie mit der Funktion NEGBINOM.VERT().

	B	C	D	E	F	G	H	I
10	Argumente für die Funktion				Beispiele			
11	AnzahlMisserfolge	15	10	8	6	4	2	0
12	AnzahlErfolge	5	5	5	5	5	5	5
13	Erfolgswahrscheinlichkeit	0,5	0,5	0,5	0,5	0,5	0,5	0,5
14	NEGBINOM.VERT()	0,37%	3,05%	6,04%	10,25%	13,67%	11,72%	3,13%
15					=NEGBINOM.VERT(F11:F12:F13)			

Abbildung 11.94: Berechnen Sie die gesuchte Wahrscheinlichkeit mit der Funktion NEGBINOM.VERT()

Welche Aussagen kann für Abbildung 11.94 in Bezug auf *AnzahlMisserfolge* = 6 getroffen werden?

Die Wahrscheinlichkeit, dass Sie zunächst sechs Personen treffen, die Ihre Frage nach dem Weg verneinen müssen, bevor Sie fünf Personen treffen, die den Weg zu der von Ihnen gesuchten Sehenswürdigkeit kennen, beträgt 10,25 %.

BINOM.VERT(), BINOM.VERT.BEREICH(), FAKULTÄT(), HYPGEOM.VERT(), KOMBINATIONEN(), BINOM.INV(), VARIATIONEN(), VARIATIONEN2(), WAHRSCHEBEREICH()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Negbinom.vert.*

Praxiseinsatz

Aussagen

Siehe auch



NORM.INV() / NORMINV()**NORM.INV() / NORMINV()****Syntax** NORMINV(*Wahrsch*; *Mittelwert*; *Standabwn*)**Definition** Die Funktion NORM.INV() gibt Quantile der Normalverteilung zurück.**Argumente** *Wahrsch* (erforderlich) ist die zur Standardnormalverteilung gehörige Wahrscheinlichkeit.*Mittelwert* (erforderlich) ist das arithmetische Mittel der Verteilung.*Standabwn* (erforderlich) ist die Standardabweichung der Verteilung.**Hinweis** Ist eines der Argumente kein numerischer Ausdruck, gibt die Funktion NORM.INV() den Fehlerwert #WERT! zurück.Ist *Wahrsch* kleiner 0, oder ist *Wahrsch* größer 1, gibt NORM.INV() den Fehlerwert #ZAH! zurück.Ist *Standabwn* kleiner oder gleich 0, gibt NORM.INV() den Fehlerwert #ZAH! zurück.NORM.INV() verwendet die Standardnormalverteilung, wenn *Mittelwert* = 0 und *Standabwn* = 1 ist.

Vergleichen Sie hierzu auch die Funktion NORM.S.INV().

Hintergrund Die standardisierte Normalverteilung ist eine Normalverteilung mit dem Mittelwert = 0 und der Standardabweichung = 1. Jede Normalverteilung lässt sich in eine Standardnormalverteilung transformieren, indem von jedem Wert x_i der Mittelwert subtrahiert wird und dann die Differenz durch die Standardabweichung dividiert wird.Weil die Standardnormalverteilung normiert ist, lassen sich aus ihren Daten (bezeichnet als z -Werte) Flächen für jede Normalverteilung mithilfe folgender Formel berechnen:

$$z_i = \frac{x_i - \bar{x}}{s}$$

Aufgrund dieser Normierung liegen die Flächenanteile in Tabellenform vor und lassen sich für jeden z -Wert entweder aus Tafeln oder eben direkt per Excel-Funktionen entnehmen.

Die errechneten Flächenanteile werden immer von der linken Seite der Grafik (siehe Abbildung 11.95) aus gelesen.

Hinweis Mehr Informationen zur Normalverteilung finden Sie unter der Funktion NORM.VERT() auf Seite 468.

Wie Sie der Abbildung 11.95 entnehmen können, sind bei dieser Darstellung der Dichtefunktion der Standardnormalverteilung die Intervalle im Abstand 1, 2 und 3 Standardabweichungen vom Erwartungswert 0 angegeben. Diese Bereiche umfassen ca. 68 %, 95,5 % und 99,7 % der Fläche unter der Glockenkurve. Die gleichen Prozentsätze gelten für alle Normalverteilungen in Bezug auf die entsprechenden Erwartungswerte und Standardabweichungen.

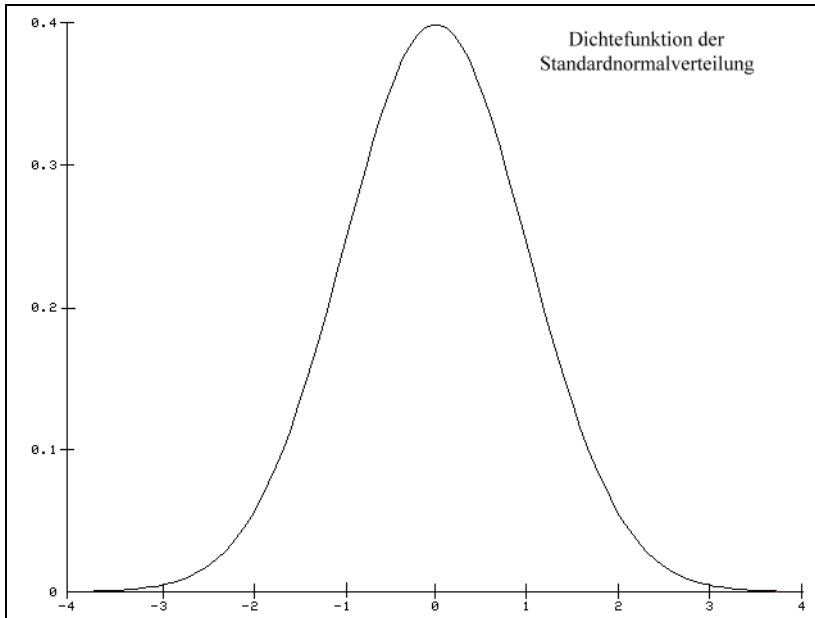


Abbildung 11.95: Dichtefunktion der Standardnormalverteilung

Sie sind Hersteller von Glühbirnen, die Sie europaweit vertreiben, und sind dabei, die Leistung der Glühbirnen zu analysieren. Die mittlere Lebenserwartung sowie die dazugehörige Standardabweichung haben Sie bereits errechnet. Nun möchten Sie wissen, wie hoch die Brennleistung von 85 % und 15 % Ihrer Glühbirnen unter Berücksichtigung des Mittelwerts und der Standardabweichung ist.

Praxiseinsatz

Unter Angabe der nachfolgenden Argumente nutzen Sie für diese Berechnung die Funktion NORM.INV().

- ▶ **Wahrsch** – 85 % (15 %), das heißt für 85 % (15 %) der Glühbirnen suchen Sie die dazugehörige Brennleistung
- ▶ **Mittelwert** – 2.000 Stunden, errechnet aus der Grundgesamtheit der Glühbirnen
- ▶ **Standabwn** – 579 Stunden, errechnet aus der Grundgesamtheit der Glühbirnen

Die Ergebnisse für beide Berechnungen können Sie Abbildung 11.96 entnehmen.

	B	C	D	E
7	Wie hoch ist die Brenngleistung von 85% der Glühbirnen bei einem Mittelwert von 2000 Stunden und einer Standardabweichung von 579?			
8				
9	Argumente für die Normalverteilung	Beispiel	Ergebnis	
10	Wahrscheinlichkeit	0,85		
11	Mittelwert	2000		
12	Standardabweichung	579		
13			2600	=NORM.INV(C10;C11;C12)
14				
15	Wie hoch ist die Brenngleistung von 15% der Glühbirnen bei einem Mittelwert von 2000 Stunden und einer Standardabweichung von 579?			
16				
17	Argumente für die Normalverteilung	Beispiel	Ergebnis	
18	Wahrscheinlichkeit	0,15		
19	Mittelwert	2000		
20	Standardabweichung	579		
21			1400	=NORM.INV(C18;C19;C20)

Abbildung 11.96: *NORM.INV()* liefert die Brenngleistung für 85 % bzw. 15 % der Glühbirnen

- Aussagen**
- ▶ 85 % der Glühbirnen haben eine Brenngleistung von bis zu 2.600 Stunden
 - ▶ 15 % der Glühbirnen haben eine Brenngleistung von nur 1.400 Stunden

Siehe auch *NORM.VERT()*, *GAUSS()*, *G.TEST()*, *PHI()*, *STANDARDISIERUNG()*, *NORM.S.VERT()*, *NORM.S.INV()*



Dieses Beispiel finden Sie im Ordner *Ms5-235\Kap11* in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Norm.inv.*

NORM.S.INV() / STANDNORMINV()

NORM.S.INV() / NORMSINV()

Syntax *NORM.S.INV(Wahrsch)*

Definition Die Funktion *NORM.S.INV()* gibt Quantile der Standardnormalverteilung zurück. Die Standardnormalverteilung hat einen Mittelwert von 0 und eine Standardabweichung von 1.

Argumente *Wahrsch* (erforderlich) ist die zur Standardnormalverteilung gehörige Wahrscheinlichkeit.

Hinweis Ist *Wahrsch* kein numerischer Ausdruck, gibt *NORM.S.INV()* den Fehlerwert *#WERT!* zurück.

Ist *Wahrsch* kleiner 0 oder ist *Wahrsch* größer 1, gibt *NORM.S.INV()* den Fehlerwert *#ZAHL!* zurück.

Bei gegebenem Wert für *Wahrsch* sucht *NORM.S.INV()* den Wert *z* so, dass *STANDNORMVERT(z) = Wahrsch* gilt.

Daher hängt die Genauigkeit von *NORM.S.INV()* von der Genauigkeit von *NORM.S.VERT()* ab. *NORM.S.INV()* geht bei der Suche iterativ vor. Hat die Suche nach 100 Iterationsschritten noch nicht konvergiert, gibt die Funktion den Fehlerwert *#NV* zurück.

Die Funktion NORM.S.INV() ist die Umkehrfunktion zur Funktion NORM.S.VERT() und liefert die zu einer bestimmten Wahrscheinlichkeit gehörende standardnormalverteilte Zufallsvariable bzw. den z -Wert einer Standardnormalverteilung.

Hintergrund

Mit der Funktion NORM.S.INV() wird also die Stelle z auf der horizontalen Achse berechnet, die dem kumulierten Flächenanteil der Standardnormalverteilung entspricht.

Mehr Informationen erhalten Sie auf nachfolgenden Seitenangaben: (Standard-) Normalverteilung auf Seite 468 unter der Funktion NORM.VERT(), Funktion STANDARDISIERUNG() auf Seite 516, Funktion NORM.S.VERT() auf Seite 466.

Hinweis

Wie schon unter den Funktion STANDARDISIERUNG() und NORM.S.VERT() möchten wir auch für die Funktion NORM.S.INV() auf das Beispiel des Glühbirnenherstellers zurückgreifen.

Praxiseinsatz

Sie sind Hersteller von Glühbirnen, die Sie europaweit vertreiben, und sind dabei, die Leistung der Glühbirnen zu analysieren. Eine Aufstellung der Messungen haben Sie bereits in Excel gemacht. Die mittlere Lebenserwartung sowie die dazugehörige Standardabweichung sind errechnet. Zusätzlich sind die Wahrscheinlichkeiten für die gemessenen Werte der Brennleistungen gegeben.

	B	C	D
	Testglühbirnen	Gemessene Brennleistung in Stunden (=X)	Wahrscheinlichkeiten der Verteilung
9			
10	1	1000	4,20%
11	2	1100	6,00%
12	3	1200	8,35%
13	4	1300	11,33%
14	5	1400	15,00%
15	6	1500	19,38%
16	7	1600	24,48%
17	8	1700	30,21%
18	9	1800	36,48%
19	10	1900	43,14%
20	11	2000	50,00%
21	12	2100	56,86%
22	13	2200	63,52%
23	14	2300	69,79%

Abbildung 11.97: Die Brennleistungen inklusive der jeweiligen Wahrscheinlichkeit

Nun möchten Sie, mithilfe der Funktion NORM.S.INV(), die standardnormalverteilten z -Werte (Variablen) der jeweiligen Wahrscheinlichkeit berechnen.

Welches Argument ist für die Funktion notwendig?

- ▶ **Wahrsch** = ist die zur Standardnormalverteilung gehörige Wahrscheinlichkeit (Spalte D)

Das Ergebnis können Sie der Abbildung 11.98 entnehmen.

E10 f _x =NORM.S.INV(D10)				
	B	C	D	E
	Testglühbirnen	Gemessene Brennleistung in Stunden (=X)	Wahrscheinlichkeiten der Verteilung	z-Werte
10	1	1000	4,20%	-1,7277
11	2	1100	6,00%	-1,5550
12	3	1200	8,35%	-1,3822
13	4	1300	11,33%	-1,2094
14	5	1400	15,00%	-1,0366
15	6	1500	19,38%	-0,8639
16	7	1600	24,48%	-0,6911
17	8	1700	30,21%	-0,5183
18	9	1800	36,48%	-0,3455
19	10	1900	43,14%	-0,1728
20	11	2000	50,00%	0,0000
21	12	2100	56,86%	0,1728
22	13	2200	63,52%	0,3455
23	14	2300	69,79%	0,5183
24	15	2400	75,52%	0,6911

Abbildung 11.98: Die Berechnung der standardnormalverteilten Variablen über NORM.S.INV()

Über die Funktion NORM.S.INV() und unter Angabe einer Wahrscheinlichkeit haben Sie alle standardnormalverteilten Werte der Verteilung generiert.

Hinweis Diese standardnormalverteilten Werte können Sie auch mithilfe der Funktion STANDARDISIERUNG() errechnen. Hierbei sind die Angabe der Argumente *x*, *Mittelwert* und *Standardabw*n notwendig. Vergleichen Sie die Funktion STANDARDISIERUNG() auf Seite 516.

Siehe auch GAUSS(), G.TEST(), NORM.INV(), NORM.VERT(), NORM.S.VERT(), PHI(), STANDARDISIERUNG()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Norm.s.inv*.

NORM.S.VERT() / STANDNORMVERT()

NORM.S.DIST() / NORMSDIST()

Syntax NORM.S.INV (*z;kumuliert*)

Definition Die Funktion NORM.S.VERT() gibt Werte bzw. Wahrscheinlichkeiten der Verteilungsfunktion einer standardnormalverteilten Zufallsvariablen zurück. Die Standardnormalverteilung hat einen Mittelwert von 0 und eine Standardabweichung von 1. Sie können diese Funktion anstelle einer Tabelle verwenden, in der Werte der Verteilungsfunktion der Standardnormalverteilung zusammengestellt sind.

z (erforderlich) ist der Wert der Verteilung (Quantil), dessen Wahrscheinlichkeit Sie berechnen möchten.

Argumente

kumuliert (erforderlich) ist ein Wahrheitswert, der den Typ der Funktion bestimmt. Ist *kumuliert* mit WAHR belegt, gibt NORM.S.VERT() den Wert der Verteilungsfunktion (kumulierte Dichtefunktion) zurück. Ist *kumuliert* mit FALSCH belegt, gibt NORM.S.VERT() den Wert der Dichtefunktion zurück.

Ist z kein numerischer Ausdruck, gibt die Funktion NORM.S.VERT() den Fehlerwert #WERT! zurück.

Die Funktion NORM.S.VERT() liefert die kumulierten Wahrscheinlichkeiten bzw. die Dichte einer standard-normalverteilten Zufallsvariablen.

Hintergrund

Die Standardnormalverteilung (auch »standardisierte Normalverteilung«) ist eine Normalverteilung, deren arithmetisches Mittel μ gleich Null und deren Standardabweichung σ gleich 1 ist.

Die Gleichung der Dichtefunktion der standardisierten Normalverteilung lautet:

$$f(z) = \frac{1}{\sqrt{2\pi}} e^{-\frac{z^2}{2}}$$

Die Funktion PHI() liefert wie die Funktion NORM.S.VERT() mit dem Parameter *kumuliert* = FALSCH den Wert der Dichtefunktion.

Hinweis

Mehr Informationen zur (Standard-) Normalverteilung und zur Funktion STANDARDISIERUNG() finden Sie auf Seite 464 unter der Funktion NORM.VERT() bzw. auf Seite 516 unter STANDARDISIERUNG().

Um Ihnen die Funktion NORM.S.VERT() in der Praxis zu verdeutlichen, bleiben wir bei dem Beispiel des Glühbirnenherstellers, das wir bereits unter der Funktion STANDARDISIERUNG() aufgeführt haben.

Praxiseinsatz

Sie sind Hersteller von Glühbirnen, die Sie europaweit vertreiben, und sind dabei, die Leistung der Glühbirnen zu analysieren. Eine Aufstellung der Messungen haben Sie bereits in Excel gemacht.

Auch die mittlere Lebenserwartung sowie die dazugehörige Standardabweichung haben Sie bereits errechnet. Die standardisierten Werte der Verteilung haben Sie mithilfe der Funktion STANDARDISIERUNG() errechnet.

Nun möchten Sie die Werte bzw. Wahrscheinlichkeiten aller standardnormalverteilten Variablen der Verteilung zurückgeben. Dies erledigen Sie mithilfe der Funktion NORM.S.VERT().

Welches Argument ist für die Funktion notwendig?

- ▶ z = ist der Wert der Verteilung (Quantil), dessen Wahrscheinlichkeit Sie berechnen möchten. In unserem Fall die Brennleistung in Form des standardnormalverteilten z -Werts.
- ▶ *kumuliert* = ist der Wahrheitswert, der den Typ der Funktion bestimmt

Die Ergebnisse können Sie beispielhaft Abbildung 11.99 entnehmen.

E10		fx =NORM.S.VERT(D10;WAHR)		
	B	C	D	E
	Testglühbirnen	Gemessene Brennleistung in Stunden (=X)	Standardisierung =z-Werte	Wahrscheinlichkeiten der Verteilung
9				
10	1	1000	-1,728	0,042
11	2	1100	-1,555	0,060
12	3	1200	-1,382	0,083
13	4	1300	-1,209	0,113
14	5	1400	-1,037	0,150
15	6	1500	-0,864	0,194
16	7	1600	-0,691	0,245
17	8	1700	-0,518	0,302
18	9	1800	-0,346	0,365
19	10	1900	-0,173	0,431
20	11	2000	0,000	0,500
21	12	2100	0,173	0,569
22	13	2200	0,346	0,635
23	14	2300	0,518	0,698
24	15	2400	0,691	0,755
25	16	2500	0,864	0,806

Abbildung 11.99: Berechnen Sie die Wahrscheinlichkeit standardnormalverteilter Variablen

Aussagen Welche Aussage kann nun getroffen werden?

Mit einer Wahrscheinlichkeit von 4,2 % (Zelle E10) tritt der standardnormalverteilte Wert – 1,728 (Zelle D10) auf.

Siehe auch GAUSS(), G.TEST(), NORM.INV(), NORM.VERT(), NORM.S.INV(), PHI(), STANDARDISIERUNG()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Norm.s.vert.*

NORM.VERT() / NORMVERT()

NORM.DIST() / NORMDIST()

Syntax NORM.VERT(*x*;Mittelwert;Standabwn;kumuliert)

Definition Die Funktion NORM.VERT() gibt die Normalverteilung für den angegebenen Mittelwert und die angegebene Standardabweichung zurück. Diese Funktion hat sehr viele Anwendungsgebiete innerhalb der Statistik – unter anderem auch beim Testen von Hypothesen.

Argumente *x* (erforderlich) ist der Wert der Verteilung (Quantil), dessen Wahrscheinlichkeit Sie berechnen möchten.

Mittelwert (erforderlich) ist das arithmetische Mittel der Verteilung.

Standabwn (erforderlich) ist die Standardabweichung der Verteilung.

kumuliert (erforderlich) ist der Wahrheitswert, der den Typ der Funktion bestimmt. Ist das Argument *kumuliert* mit *WAHR* belegt, gibt die Funktion NORM.VERT() den Wert der Verteilungsfunktion (kumulierte Dichtefunktion) zurück. Ist *kumuliert* mit *FALSCH* belegt, gibt NORM.VERT() den Wert der Dichtefunktion zurück.

Ist *Mittelwert* oder *Standabwn* kein numerischer Ausdruck, gibt die Funktion NORM.VERT() den Fehlerwert #WERT! zurück.

Hinweis

Ist *Standabwn* kleiner oder gleich 0, gibt NORM.VERT() den Fehlerwert #ZAHL! zurück.

Ist *Mittelwert* = 0, *Standabwn* = 1 und *kumuliert* = WAHR, gibt NORM.VERT() die Standardnormalverteilung (NORM.S.VERT()) zurück.

Um in Excel das Berechnen von Verteilungen und das Prüfen von Hypothesen vorzunehmen, stehen Ihnen verschiedene Funktionen zur Verfügung. So zum Beispiel die Funktion NORM.VERT(). Generell helfen Verteilungen bei typischen Fragen zur Wahrscheinlichkeit. Ein Beispiel wäre der Wurf einer Münze, bei der es nur zwei Ausprägungen gibt: Treffer oder Nicht-Treffer.

Hintergrund

Wie oben im Abschnitt »Definition« erwähnt, gibt die Funktion NORM.VERT() die Normalverteilung für bestimmte angegebene Werte zurück.

Die Normalverteilung (nach Carl Friedrich Gauß) ist der wichtigste Typ kontinuierlicher Wahrscheinlichkeitsverteilungen, die eine Aussage darüber liefert, mit welcher Wahrscheinlichkeit eine Zufallsvariable X einen bestimmten Wert annimmt. Ihre Wahrscheinlichkeitsdichte wird auch Gauß-Funktion, Gauß-Kurve, Gauß-Glocke oder Glockenkurve genannt.

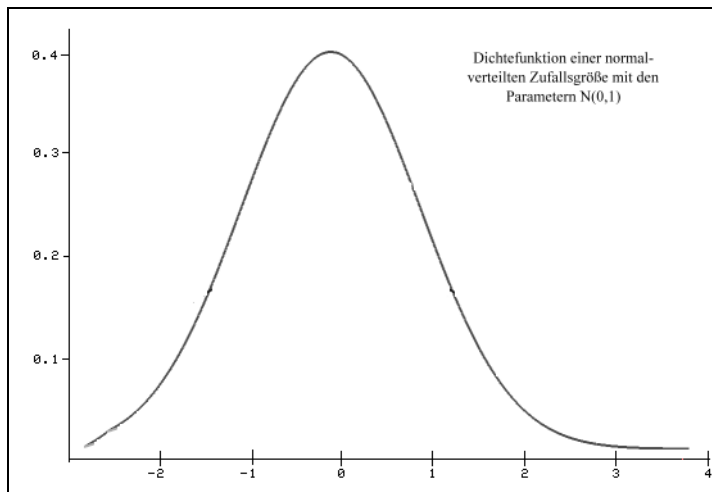


Abbildung 11.100: Verschiedene Dichtefunktionen von normalverteilten Zufallsgrößen grafisch dargestellt

Die besondere Bedeutung der Normalverteilung beruht unter anderem auf dem zentralen Grenzwertsatz, der besagt, dass eine Summe von n unabhängigen, identisch verteilten Zufallsvariablen in der Grenze normalverteilt ist.

Viele natur-, wirtschafts- und ingenieurwissenschaftliche Vorgänge lassen sich durch die Normalverteilung entweder exakt oder wenigstens in sehr guter Näherung beschreiben – vor allem Prozesse, die in mehreren Faktoren unabhängig voneinander in verschiedene Richtungen wirken.

Die Normalverteilung ist, im Gegensatz zur Binomialverteilung, symmetrisch aufgebaut (siehe Abbildung 11.100). Das heißt, sie gleicht einer Glockenkurve, bei der der niedrigste und höchste Wert die geringste und der Mittelwert die höchste Wahrscheinlichkeit hat.

Aussagen Was sind die Eigenschaften einer Normalverteilung?

- ▶ glockenförmig
- ▶ eingipfelig
- ▶ nähert sich asymptotisch der X-Achse
- ▶ und ist symmetrisch

Aufgrund dessen

- ▶ liegt das Maximum beim arithmetischen Mittel
- ▶ liegen je 50 % der Fläche beidseitig vom arithmetischen Mittel
- ▶ also sind auch arithmetisches Mittel und Median deckungsgleich
- ▶ fallen Modalwert, Median und arithmetisches Mittel zusammen
- ▶ liegen die Wendepunkte bei Mittelwert plus Standardabweichung und bei Mittelwert minus Standardabweichung

Für die meisten Verteilungen bietet Excel je zwei Funktionen an. Die Funktion zur Berechnung einer Verteilung, mit der Endung VERT, berechnet die Wahrscheinlichkeit, mit der ein bestimmter Wert eintritt. Die dazugehörige Umkehrfunktion, mit der Endung INV, berechnet den Wert, der mit einer bestimmten Wahrscheinlichkeit eintritt.

Hinweis Die Funktion NORM.INV() enthält als Argument eine Wahrscheinlichkeit, eine Standardabweichung und einen Mittelwert und liefert den Wert, der mit in der Funktion NORM.VERT() eingegeben wurde.

Die Gleichung der Dichtefunktion der Normalverteilung (kumuliert = *FALSCH*) lautet:

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\left(\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

Wenn kumuliert = *WAHR*, gibt die Formel das Integral einer gegebenen Formel von der negativen Unendlichkeit bis x zurück.

Praxiseinsatz Sie sind Hersteller von Glühbirnen, die Sie europaweit vertreiben, und sind dabei, die Leistung der Glühbirnen zu analysieren. Die mittlere Lebenserwartung sowie die dazugehörige Standardabweichung haben Sie bereits errechnet. Nun möchten Sie wissen, wie wahrscheinlich es ist, dass die Glühbirnen im alltäglichen Gebrauch länger oder weniger lang als die durchschnittliche Leistungsdauer halten.

Für diese Berechnung nutzen Sie die Funktion NORM.VERT().

Die Lebenserwartung Ihrer Glühbirnen sei normal verteilt mit

- ▶ einem Mittelwert von 2.000 »Brenn«-Stunden = Argument *Mittelwert*
- ▶ einer Standardabweichung von 579 Stunden = Argument *Standabwn*

Um den Wert der Verteilungsfunktion zu ermitteln, geben Sie für das Argument kumuliert den Wahrheitswert *WAHR* an. Wollen Sie den Wert der Dichtefunktion ermitteln, verwenden Sie den Wahrheitswert *FALSCH*.

Sie stellen sich folgende Frage:

Wie groß ist die Wahrscheinlichkeit, dass eine Glühbirne bis zu 2.600 Stunden oder nur bis zu 1.400 Stunden brennt? Und wie groß ist die Wahrscheinlichkeit, dass genau diese Stundenzahlen erreicht werden?

Die beiden Werte, 2.600 Stunden und 1.400 Stunden, stellen jeweils das Argument x dar – also Werte innerhalb der Verteilung, deren Wahrscheinlichkeit Sie berechnen möchten.

Die Ergebnisse für beide Berechnungen können Sie der Abbildung 11.101 entnehmen.

	B	C	D	E
8	Wie hoch ist die Wahrscheinlichkeit, dass die Glühbirne bis zu 2600 Stunden brennt?			
9				
10	Argumente für die Normalverteilung	Beispiel	Ergebnis	
11	Der zu prüfende Wert x	2600		
12	Mittelwert	2000		
13	Standardabweichung	579		
14	Kumuliert Ja	WAHR	0,85	=NORM.VERT(C11;C12;C13;C14)
15	Kumuliert Nein	FALSCH	0,0004	=NORM.VERT(C11;C12;C13;C15)
16				
22				
23	Argumente für die Normalverteilung	Beispiel	Ergebnis	
24	Der zu prüfende Wert x	1400		
25	Mittelwert	2000		
26	Standardabweichung	579		
27	Kumuliert Ja	WAHR	0,15	=NORM.VERT(C24;C25;C26;C27)
28	Kumuliert Nein	FALSCH	0,0004	=NORM.VERT(C24;C25;C26;C28)

Abbildung 11.101: Die Wahrscheinlichkeitsberechnung für verschiedene Brennleistungen

Welche Aussagen können aus den Ergebnissen getroffen werden?

Aussagen

1. Die Wahrscheinlichkeit, dass eine Glühbirne bis zu 2.600 Stunden brennt, beträgt 85 %.
2. Die Wahrscheinlichkeit, dass eine Glühbirne genau 2.600 Stunden brennt beträgt 0,04 %.
3. Die Wahrscheinlichkeit, dass eine Glühbirne nur bis zu 1.400 Stunden brennt beträgt 15 %.
4. Die Wahrscheinlichkeit, dass eine Glühbirne genau 1.400 Stunden brennt beträgt 0,04 %.

Auf diese Weise können Sie eine Vielzahl an Berechnungen anstellen, verschiedene Hypothesen prüfen und die Realisierungswahrscheinlichkeit angeben, mit der einzelne Merkmalsausprägungen innerhalb bestimmter Intervalle zu erwarten sind.

NORM.INV(), GAUSS(), G.TEST(), STANDARDISIERUNG(), NORM.S.INV(), NORM.S.VERT(), PHI()

Siehe auch

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Norm.vert.*



Neu in Excel
2013

PHI()



PHI()

Syntax PHI(*x*)

Definition Mit der Funktion PHI() können Sie den Wert der Dichtefunktion für eine Standardnormalverteilung berechnen.

Argumente *x* (erforderlich) ist die Zahl, für die Sie die Dichte der Standardnormalverteilung ermitteln möchten.

Hinweis Wenn *x* ein ungültiger numerischer Wert ist, gibt die Funktion PHI() den Fehlerwert #ZAHLE! zurück.

Wenn für das Argument *x* ein ungültiger Datentyp verwendet wird, z.B. ein nicht numerischer Wert, gibt die Funktion PHI() den Fehlerwert #WERT! zurück.

Hintergrund Die Normalverteilung (nach Carl Friedrich Gauß) ist der wichtigste Typ kontinuierlicher Wahrscheinlichkeitsverteilungen, die eine Aussage darüber liefert, mit welcher Wahrscheinlichkeit eine Zufallsvariable *X* einen bestimmten Wert annimmt. Ihre Wahrscheinlichkeitsdichte wird auch Gauß-Funktion, Gauß-Kurve, Gauß-Glocke oder Glockenkurve genannt.

Die besondere Bedeutung der Normalverteilung beruht unter anderem auf dem zentralen Grenzwertsatz, der besagt, dass eine Summe von *n* unabhängigen, identisch verteilten Zufallsvariablen in der Grenze normalverteilt ist.

Die Gleichung der Dichtefunktion der Standardnormalverteilung lautet:

$$f(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

Hinweis Die Funktion NORM.S.VERT() mit dem Parameter *kumuliert* = FALSCH liefert ebenfalls den Wert der Dichtefunktion.

Praxiseinsatz Alle wichtigen Informationen zur (Standard-)Normalverteilung und deren Praxiseinsatz finden Sie in den Abschnitten NORM.INV() auf Seite 462, NORM.VERT() auf Seite 468 sowie NORM.S.VERT()/STANDNORMVERT() auf Seite 466.

Siehe auch NORM.INV(), NORM.S.INV(), NORM.S.VERT(), NORM.VERT()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Phi*.

PEARSON()



PEARSON()

Syntax PEARSON(*Matrix1*; *Matrix2*)

Definition Die Funktion PEARSON() gibt den Pearsonschen Korrelationskoeffizienten *r* zurück. Dieser Koeffizient ist ein dimensionsloser Index mit dem Wertebereich –1,0 kleiner oder gleich 1,0 und ein Maß dafür, inwieweit zwischen zwei Datensätzen eine lineare Abhängigkeit besteht.

Argumente *Matrix1* (erforderlich) ist eine Reihe unabhängiger Werte.

Matrix2 (erforderlich) ist eine Reihe abhängiger Werte.

Als Argumente müssen entweder Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Hinweis

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Enthalten die Argumente *Matrix1* und *Matrix2* keine oder unterschiedlich viele Datenpunkte, gibt die Funktion PEARSON() den Fehlerwert #NV zurück.

Ein quantitatives Maß für die Stärke des Zusammenhangs bei metrischen Merkmalen bietet der Korrelationskoeffizient nach Bravais-Pearson (empirischer Korrelationskoeffizient).

Hintergrund

Dieser wird in Excel mit der Funktion PEARSON() berechnet. Als Eingabe fungieren zwei auszuwählende Datenreihen: *Matrix1* und *Matrix2*. Die erste Datenreihe stellt die Werte der unabhängigen Variable x dar, die zweite Datenreihe besteht aus den Werten der abhängigen Variable y .

Das Ergebnis der Berechnung für r über die Funktion PEARSON() liegt im Wertebereich zwischen -1 und $+1$. Doch was sagen die Ergebnisse innerhalb dieses Wertebereichs aus?

- ▶ Bei $r = -1$ liegt eine perfekte negative Korrelation vor – das heißt, die Datenreihen sind nicht voneinander abhängig
- ▶ Bei $r = +1$ liegt eine perfekte positive Korrelation vor – das heißt, die Datenreihen sind absolut voneinander abhängig
- ▶ Ein Wert in der Nähe von $r = 0$ bedeutet, dass keine lineare Korrelation vorliegt

Der Zusammenhang zwischen den beiden Merkmalen muss linear sein, da ein nichtlinearer Zusammenhang von der Funktion PEARSON() nicht erfasst wird. Deshalb sollte zusätzlich zur Berechnung des Korrelationskoeffizienten r immer auch das Streudiagramm betrachtet werden. Ansonsten könnte man fälschlicherweise behaupten, ein Merkmalspaar zeige, aufgrund des errechneten Werts für r , keinen Zusammenhang auf, obwohl ein nichtlinearer, also beispielsweise ein exponentieller, Zusammenhang vorliegt.

Der Wert eines Korrelationskoeffizienten, errechnet durch die Funktionen PEARSON() oder KORREL(), gibt keine Richtung der Beeinflussung zwischen zwei Variablen vor. Diese kann nur durch inhaltliche Überlegungen gefunden werden. Darüber hinaus muss bei der Interpretation einer statistisch gesicherten Korrelation von Fall zu Fall geklärt werden, ob es sich um einen echten oder nur um einen scheinbaren Zusammenhang handelt. Dazu ist jeweils Kenntnis über den Sachverhalt nötig.

Scheinkorrelationen, die inhaltlich nicht gerechtfertigt sind, findet man oft in Zeitreihen, wenn für die beiden Variablen der gleiche Trend vorherrscht.

Generell gilt: Man sollte eine hohe Korrelation nur als *Hinweis* auf einen bestehenden Zusammenhang ansehen. Auf kausale Zusammenhänge sollte aufgrund inhaltlicher Überlegungen geschlossen werden.

Die Formel für den Pearsonschen Korrelationskoeffizienten r lautet:

$$r = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}}$$

Dabei sind x und y die Stichprobenmittelwerte MITTELWERT(*Matrix1*) und MITTELWERT(*Matrix2*).

Hinweis Der Unterschied zwischen den Funktionen KORREL() und PEARSON() besteht darin, dass bei der Funktion PEARSON() ein linearabhängiger Zusammenhang zwischen zwei Komponenten vorausgesetzt wird. Als Argumente für die Funktion werden daher *Matrix1*, eine Reihe unabhängiger Werte, und *Matrix2*, eine Reihe abhängiger Werte, gefordert. Das heißt, über PEARSON() wird die Stärke einer bereits ersichtlichen Abhängigkeit berechnet, deren Ausmaß jedoch zunächst unbekannt ist.

Bei der Funktion KORREL() wird dieser lineare Zusammenhang nicht vorausgesetzt. Die Werte, für die eine Korrelation aufgezeigt werden soll, werden wie zwei unabhängige Variable betrachtet. Deshalb fordert die Funktion die Argumente *Matrix1*, ein Zellbereich mit Werten, und *Matrix2*, ebenfalls ein Zellbereich mit Werten. Das heißt, über KORREL() wird ein wahrscheinlicher Zusammenhang berechnet, der jedoch nicht zwingend sein muss.

Praxiseinsatz Ein in der Softwarebranche tätiges Unternehmen verkauft auf der firmeneigenen Website alle vom Unternehmen angebotenen Produkte. Regelmäßig versickt das Unternehmen Newsletter, um über Neuheiten zu informieren und bereits vorhandene, aber auch potenzielle Neukunden auf die Website und damit auf das Unternehmen aufmerksam zu machen.

Die Produktbestellungen auf der Website sind im vergangenen Jahr merklich gestiegen. Die Geschäftsführung möchte die Gründe hierfür erforschen. Hängt der Anstieg der Bestellungen mit dem aktiven Marketing und den straffen Werbemaßnahmen der letzten Zeit zusammen? Sind durch die vermehrten Websitezugriffe auch die Bestellungen gestiegen?

Es soll also aufgezeigt werden, inwieweit ein Zusammenhang zwischen den Websitezugriffen und den Bestellungen besteht. Durch die Erstellung eines Streudiagramms und mithilfe einer linearen Trendlinie (siehe Abbildung 11.102) wurde bereits aufgezeigt, dass ein linearer Zusammenhang zwischen den beiden Variablen x und y , also in diesem Fall den Webseitenzugriffen und den Bestellungen, besteht.

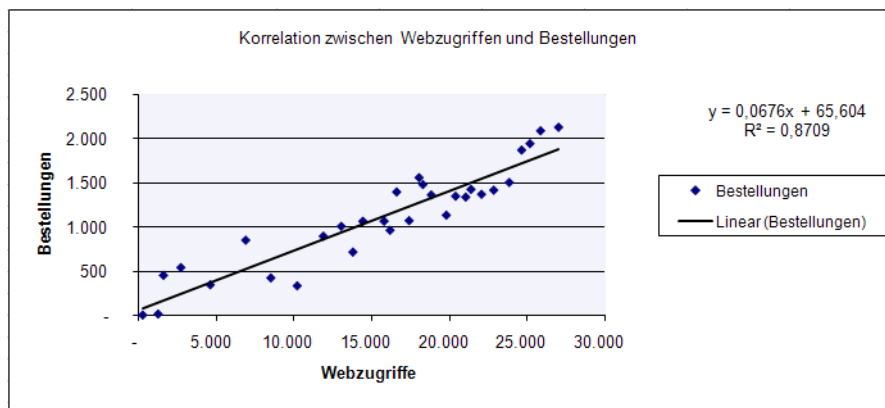


Abbildung 11.102: Das Streudiagramm zeigt bereits eine lineare Abhängigkeit zwischen den Variablen auf

Hinweis Liefert das Streudiagramm keinen eindeutigen linearen Zusammenhang zwischen den zwei zu betrachtenden Variablen, muss anstatt der Funktion PEARSON() die Funktion KORREL() eingesetzt werden.

Die Berechnung des Korrelationskoeffizienten r über die Funktion PEARSON() liefert den Beweis für eine lineare Abhängigkeit, wie in Abbildung 11.103 dargestellt.

	B	C	D
8		Unabhängig = x-Wert	Abhängig = y-Wert
9	Monat	Webzugriffe	Bestellungen
10	Januar 2006	236	6
11	Februar 2006	1.221	17
12	März 2006	1.563	456
13	April 2006	2.682	544
14	Mai 2006	4.569	349
15	Juni 2006	6.848	854
16	Juli 2006	8.463	427
17	August 2006	10.157	337
18	September 2006	11.837	899
19	Oktober 2006	12.987	1.011
20	November 2006	13.739	720
21	Dezember 2006	14.376	1.069
22	Januar 2007	15.739	1.070
23	Februar 2007	16.123	967
24	März 2007	16.548	1.401
25	April 2007	17.352	1.076
26	Mai 2007	17.986	1.563
27	Juni 2007	18.234	1.485
28	Juli 2007	18.769	1.367
29	August 2007	19.736	1.138
30	September 2007	20.333	1.352
31	Oktober 2007	20.987	1.343
32	November 2007	21.323	1.430
33	Dezember 2007	21.999	1.375
34	Januar 2008	22.786	1.421
35	Februar 2008	23.784	1.508
36	März 2008	24.574	1.876
37	April 2008	25.111	1.948
38	Mai 2008	25.789	2.094
39	Juni 2008	26.948	2.134
40	Korrelationskoeffizient nach Pearson		0,933231423
41	=PEARSON(C10:C39;D10:D39)		

Abbildung 11.103: Die Funktion PEARSON() liefert den Wert für den Korrelationskoeffizienten r

Der Korrelationskoeffizient in Höhe von 0,933 steht für eine nahezu perfekte positive Korrelation. Das heißt, da nichtlineare Zusammenhänge von der Funktion PEARSON() nicht erfasst werden, dass die Datenreihen *Webseitenzugriffe* und *Bestellungen* linear voneinander abhängig sind. Dies wurde bereits durch das Streudiagramm (siehe Abbildung 11.102) angenommen bzw. vorausgesetzt.

Anders ausgedrückt bedeutet der Korrelationskoeffizient von 0,933 auch, dass, wenn sich die Anzahl der Webseitenzugriffe beispielsweise durch verschiedene Marketingaktionen erhöht, auch die Anzahl der Bestellungen über das Web steigt.

KORREL(), ACHSENABSCHNITT(), BESTIMMTHEITSMASS(), RGP(), STEIGUNG(), STFEHLERYX()

Siehe auch

Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Regression.xlsx* auf dem Arbeitsblatt *Pearson*.



POISSON.VERT() / POISSON()

POISSON.DIST() / POISSON()

Syntax POISSON.DIST(*x*; *Mittelwert*; *kumuliert*)

Definition Die Funktion POISSON.VERT() gibt Wahrscheinlichkeiten einer poissonverteilten Zufallsvariablen zurück. Eine übliche Anwendung der Poissonverteilung ist die Modellierung der Anzahl der Ereignisse innerhalb eines bestimmten Zeitraums, beispielsweise die Anzahl der in einem Call-Center eingehenden Anrufe innerhalb einer Stunde, die einen sofortigen Vororteinsatz mit sich bringen.

Argumente *x* (erforderlich) ist die Zahl der Fälle.

Mittelwert (erforderliche) ist der erwartete Zahlenwert.

kumuliert (erforderlich) ist der Wahrheitswert, der den Typ der Funktion bestimmt. Ist *kumuliert* mit *WAHR* belegt, gibt POISSON.VERT() den Wert der Verteilungsfunktion der jeweiligen Poissonverteilung zurück, also die Wahrscheinlichkeit, dass die Anzahl zufällig eintretender Ereignisse zwischen 0 und einschließlich *x* liegt. Ist *kumuliert* mit *FALSCH* belegt, gibt POISSON.VERT() den Wert der Wahrscheinlichkeitsfunktion zurück, also die Wahrscheinlichkeit, dass die Anzahl der Ereignisse genau *x* sein wird.

Hinweis Ist *x* keine ganze Zahl, werden die Nachkommastellen abgeschnitten.

Ist *x* oder *Mittelwert* kein numerischer Ausdruck, gibt POISSON.VERT() den Fehlerwert #WERT! zurück.

Ist *x* kleiner 0, gibt POISSON.VERT() den Fehlerwert #ZAHL! zurück.

Ist *Mittelwert* kleiner oder gleich 0, gibt POISSON.VERT() den Fehlerwert #ZAHL! zurück.

Hintergrund Die Funktion POISSON.VERT(), benannt nach Denis Poisson (1781 bis 1840), liefert Wahrscheinlichkeiten einer poisson-verteilten Zufallsvariable. Die Poisson-Verteilung ist ein besonderes Modell für die Häufigkeit eines sogenannten »seltenen Ereignisses«. Sie beschreibt das Eintreffen voneinander unabhängiger, gleichartiger, seltener Ereignisse in einer großen Anzahl von Elementen.

Ein Beispiel wäre das Eintreffen von Kunden an einem Schalter oder das Eingehen von Telefonanrufen.

Die Poisson-Verteilung eignet sich besonders für Wahrscheinlichkeitsverteilungen, bei denen sehr viele Ergebnisse aus einer Stichprobe vorliegen und die Wahrscheinlichkeit, dass das zu untersuchende Ereignis eintritt, sehr klein ist.

Hier nähert sich die Poisson-Verteilung nämlich der Binomialverteilung an. Im Gegensatz zur Binomialverteilung benötigt die Poisson-Verteilung (abgesehen von *x*) jedoch nur einen Parameter – den Erwartungswert bzw. das Argument *Mittelwert*.

Für kleine Erfolgswahrscheinlichkeiten kann die Binomialverteilung also durch die Poisson-Verteilung angenähert werden. D.h., sie gilt, wenn die durchschnittliche Anzahl der Ereignisse das Ergebnis einer sehr großen Zahl von Ereignismöglichkeiten und einer sehr kleinen Ereigniswahrscheinlichkeit ist.

Als Beispiel hierzu wird oft der radioaktive Zerfall erwähnt. Aus einer sehr großen Anzahl von Atomen zerfällt in einer Zeiteinheit nur ein sehr kleiner Anteil der Atome. Dieser Zerfall ist rein zufällig und unabhängig von den schon zerfallenen Atomen. Dies ist eine wesentliche Voraussetzung für die Poisson-Verteilung.

Wie bereits erwähnt, geht das Poisson-Modell davon aus, dass das Ereignis sehr selten innerhalb eines bestimmten Zeitintervalls auftritt. Es ist dabei aber dennoch nur von der Länge des Intervalls und nicht von der Lage auf der Zeitachse abhängig.

Über die Poisson-Verteilung lässt sich also die Fehlerwahrscheinlichkeit bzw. ein Ereignis pro Einheit ermitteln.

Die Funktion POISSON.VERT() berechnet die folgenden Formeln.

Für kumuliert = *FALSCH*:

$$POISSON.VERT = \frac{e^{-\lambda} \lambda^x}{x!}$$

Für kumuliert = *WAHR*:

$$CUMPOISSON = \sum_{k=0}^x \frac{e^{-\lambda} \lambda^k}{k!}$$

Diese Formeln werden durch die Funktion POISSON.VERT() ersetzt mit *x* = Zahl der Fälle und Mittelwert = erwarteter Zahlenwert. Das Argument kumuliert bestimmt, ob genau der Mittelwert (kumuliert = *FALSCH*) oder höchstens der Mittelwert (kumuliert = *WAHR*) erreicht werden soll.

Stellen Sie sich vor, Sie sind Reifenhändler. Sie haben eine eigene »Hausmarke«, deren Qualität Sie über längere Zeit analysiert haben. Es hat sich herausgestellt, dass bei dem PKW-Reifentyp Ihrer Hausmarke pro 100.000 gefahrenen Kilometern durchschnittlich vier Reifenschäden auftreten.

Praxiseinsatz

	B	C
8		Reifenschaden auf 100.000 km
9	Testfahrzeug 1	5
10	Testfahrzeug 2	3
11	Testfahrzeug 3	6
12	Testfahrzeug 4	2
13	Testfahrzeug 5	3
14	Testfahrzeug 6	5
15	Testfahrzeug 7	4
16	Testfahrzeug 8	2
17	Testfahrzeug 9	3
18	Testfahrzeug 10	4
19	Testfahrzeug 11	5
20	Testfahrzeug 12	6
21	Testfahrzeug 13	3
22	Testfahrzeug 14	7
23	Testfahrzeug 15	2
24	Mittelwert	4

Abbildung 11.104: Die Analyse der Reifen ergibt eine Fehlerquote von 4 auf 100.000 Kilometer

Die vier Ereignisse, also die Reifenschäden, sind im Vergleich zum Intervall, den 100.000 Kilometern, ein recht seltenes Ereignis. Deshalb gilt hier die Poisson-Verteilung.

Sie stellen sich folgende Frage:

Wie groß ist die Wahrscheinlichkeit, dass nur genau drei Reifenschäden auf 100.000 km auftreten?

Sie haben folgende Werte für die Funktion POISSON.VERT() notwendigen Argumente:

- ▶ $x = 3$ (Zahl der Ereignisse, die Sie untersuchen möchten)
- ▶ *Mittelwert* = 4 (Zahl der erwarteten Ereignisse)
- ▶ *kumuliert* = *FALSCH*, da Sie die Wahrscheinlichkeit errechnen möchten, mit der **genau** die Anzahl der Ereignisse x eintritt.

Das Ergebnis stellt sich wie in Abbildung 11.105 dar.

	E	F	G	H
15	x = Zahl der "gewünschten" Ereignisse			3
16	Mittelwert = Zahl der erwarteten Ereignisse			4
17	Kumuliert			FALSCH
18	POISSON.VERT()			19,54%
19				=POISSON.VERT(H15;H16;H17)

Abbildung 11.105: Berechnen Sie die Wahrscheinlichkeit für genau drei Ereignisse

Mithilfe der Berechnung über die Funktion POISSON() ergibt sich eine Wahrscheinlichkeit von 19,54 %, dass genau drei Reifenschäden auf 100.000 Kilometer eintreten.

Möchten Sie wissen, wie groß die Wahrscheinlichkeit ist, dass die Anzahl der Reifenschäden auf 100.000 km zwischen 0 und einschließlich x , also 3, liegt, verwenden Sie für das Argument *kumuliert* den Wahrheitswert *WAHR*. Vergleichen Sie dazu Abbildung 11.106.

	E	F	G	H
26	x = Zahl der "gewünschten" Ereignisse			3
27	Mittelwert = Zahl der erwarteten Ereignisse			4
28	Kumuliert			WAHR
29	POISSON.VERT()			43,35%
30				=POISSON.VERT(H26;H27;H28)

Abbildung 11.106: Berechnen Sie die Wahrscheinlichkeit für 0 bis 3 Ereignisse

Die Berechnung über die Funktion POISSON.VERT() liefert eine Wahrscheinlichkeit von 43,35 %, dass 0 bis 3 Reifenschäden auf 100.000 Kilometer eintreten.

Siehe auch EXPON.VERT()



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Poisson.vert*.

QUANTIL()



PERCENTILE()

Die Funktion QUANTIL() wird ab Microsoft Excel 2010 durch die Funktionen QUANTIL.EXKL() und QUANTIL.INKL() ersetzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht. Um die Abwärtskompatibilität von QUANTIL.EXKL() und QUANTIL.INKL() zu sichern, ist die Funktion QUANTIL() weiter unter ihrem alten Namen verfügbar.

Hinweis

QUANTIL(*Matrix;Alpha*)

Syntax

Die Funktion QUANTIL() gibt das *Alpha*-Quantil einer Gruppe von Daten zurück. Mithilfe dieser Funktion können Sie einen Akzeptanzschwellenwert festlegen. Mithilfe dieses Schwellenwerts haben Sie beispielsweise die Möglichkeit, nur die Kunden zu einer Feier einzuladen, deren Umsätze oberhalb des 80 %-Quantils liegen.

Definition

In der Statistik bezeichnet man für eine Wahrscheinlichkeitsfunktion das *p*%-Quantil als jenen Wert des Ereignisraums, dem ein Wert von *p*% der Verteilungsfunktion zugeordnet ist. Es sind dann *p*% der Beobachtungen oder der Grundgesamtheit kleiner als das *p*%-Quantil.

Matrix (erforderlich) ist eine Matrix oder ein Datenbereich, die bzw. der die relative Lage der Daten beschreibt.

Argumente

Alpha (erforderlich) ist ein Wert aus dem geschlossenen Intervall von 0 bis 1.

Enthält Matrix keine oder mehr als 8.191 Datenpunkte, gibt die Funktion QUANTIL() den Fehlerwert #ZAH! zurück. Ist *Alpha* kein numerischer Ausdruck, gibt Quantil den Fehlerwert #WERT! zurück.

Hinweis

Ist *Alpha* kleiner 0 oder ist *Alpha* größer 1, gibt QUANTIL() den Fehlerwert #ZAH! zurück. Ist *Alpha* kein Vielfaches von $1/(n - 1)$, interpoliert QUANTIL(), um das *Alpha*-Quantil zu bestimmen.

Ein Quantil ist ein definierter Teil eines der Größe nach sortierten Datenbündels. Bei Quantilen handelt sich, mit Ausnahme des Medians, um Werte, die die Streuung eines Datenbündels charakterisieren sollen. Zu den wichtigsten Quantilen zählen die Dezile, die Quartile und der bereits erwähnte Median. Der Median entspricht dem 50 %-Quantil.

Hintergrund

Etwas formaler ausgedrückt kann man auch sagen, dass das *p*-Quantil einer Verteilung angibt, welcher Wert die unteren $p \cdot 100$ Prozent der Datenwerte von den oberen $100 - p \cdot 100$ Prozent trennt. Gibt man also z.B. das 0,65-Quantil einer Verteilung an, so besagt dieses, dass 65 Prozent aller Daten kleiner oder gleich dem betreffenden Datenwert sind und gleichzeitig 35 Prozent größer oder gleich diesem Wert.

Es kommt des Öfteren vor, dass es keinen Wert im Datenbündel gibt, der exakt dem betreffenden Quantil entspricht. Eine häufig verwendete Regel für die Bestimmung von Quantilen lautet (mit *n* = Anzahl der Datenwerte):

Wenn $n \cdot p$ keine ganze Zahl ist, so wird als Ordnungsziffer für den Quantilswert die auf $n \cdot p$ folgende ganze Zahl festgelegt. Ist $n \cdot p$ eine ganze Zahl, so liegt das Quantil zwischen der Ordnungszahl $n \cdot p$ und $n \cdot p + 1$; hier müssen also Interpolationsregeln eingesetzt werden, wie z.B. das arithmetische Mittel aus $n \cdot p$ und $n \cdot p + 1$.

Hat man als Beispiel 16 Datenwerte, ergibt sich für das 0,25-Quantil, was gleichzeitig dem ersten Quartil entspricht, die Ordnungsziffer 4 (aus $16 \cdot 0,25$). Das 0,25-Quantil oder 1. Quartil sollte aber möglichst zwischen den unteren 25 und den oberen 75 Prozent liegen; also ist es zweckmäßig, einen Wert zwischen dem 4. und dem 5. Wert des geordneten Datenbündels zu bestimmen.

Das erste Dezil wird bestimmt durch die Ordnungsziffer des beispielsweise auf $n \cdot p = 1,6$ folgenden Werts. In diesem Falle wäre das der **zweite** Wert im Datenbündel. Tatsächlich betrachtet sind dann mindestens 10 Prozent der Daten kleiner oder gleich und mindestens 90 Prozent größer oder gleich diesem Wert.

Die Feinheit der Abstufung der Quantile sollte natürlich passend zu den vorhandenen Daten gewählt werden.

Praxiseinsatz Nehmen wir wieder das Beispiel des Softwareherstellers zur Hand. Der Controllingleiter hat die Umsätze der verschiedenen Unternehmensbereiche für ein Jahr generiert und möchte die vorhandenen Daten unterteilen, um weitere Analysen machen zu können. Ziel ist es, nur die Umsätze für eine spätere Analyse zu generieren, die über dem 0,6-Quantil (60tes Perzentil) liegen. Mit Excel kann er jedes beliebige $p\%$ -Quantil ermitteln. Dabei muss für das Argument *Alpha* der Prozentsatz als Dezimalzahl eingegeben werden. Der Controllingleiter lässt sich die gewünschten Ergebnisse in 10 %-Schritten errechnen.

	B	C	D	E	F	G
9	Umsatz innerhalb verschiedener Unternehmensbereiche in einem Jahr					
10						
11	UMSÄTZE	EVENTS	PUBLIKATIONEN	TEAM	TRAINING	WISSEN
12	Januar	1.127 €	1.141,00 €	1.049,00 €	1.339,00 €	1.060,00 €
13	Februar	1.407 €	945,00 €	938,00 €	1.014,00 €	1.230,00 €
14	März	1.334 €	1.388,00 €	902,00 €	1.153,00 €	805,00 €
15	April	981 €	1.353,00 €	897,00 €	1.441,00 €	1.398,00 €
16	Mai	1.223 €	813,00 €	1.464,00 €	1.481,00 €	861,00 €
17	Juni	1.322 €	1.358,00 €	1.385,00 €	1.166,00 €	1.081,00 €
18	Juli	1.316 €	982,00 €	892,00 €	1.015,00 €	1.424,00 €
19	August	1.168 €	896,00 €	1.000,00 €	1.035,00 €	1.094,00 €
20	September	1.212 €	1.243,00 €	1.050,00 €	1.077,00 €	848,00 €
21	Oktober	1.218 €	1.088,00 €	998,00 €	1.446,00 €	1.108,00 €
22	November	1.311 €	945,00 €	1.222,00 €	828,00 €	1.465,00 €
23	Dezember	1.421 €	903,00 €	1.087,00 €	854,00 €	1.189,00 €
24	Quantil 0,1	1.131 €	897 €	898 €	870 €	849 €
25	Quantil 0,2	1.177 €	911 €	909 €	1.014 €	901 €
26	Quantil 0,3	1.214 €	945 €	956 €	1.021 €	1.066 €
27	Quantil 0,4	1.220 €	960 €	999 €	1.052 €	1.086 €
28	Quantil 0,5	1.267 €	1.035 €	1.025 €	1.115 €	1.101 €
29	Quantil 0,6	1.314 €	1.120 €	1.050 €	1.161 €	1.157 €
30	Quantil 0,7	1.320 €	1.212 €	1.076 €	1.287 €	1.218 €
31	Quantil 0,8	1.332 €	1.331 €	1.195 €	1.421 €	1.364 €
32	Quantil 0,9	1.400 €	1.358 €	1.369 €	1.446 €	1.421 €
33	Quantil 0,10	1.421 €	1.388 €	1.464 €	1.481 €	1.465 €

Abbildung 11.107: Die Quantile werden in 10 %-Schritten errechnet

Wie Sie der Abbildung 11.107 entnehmen können, stellen die Ergebnisse der einzeln berechneten Quantile gleichzeitig eine aufsteigende Rangfolge dar. Der Controllingleiter kann sich nun allen Umsätzen, die gleich oder größer dem 0,6-Quantil sind, widmen und die Zahlen weiter verarbeiten.

Über das 0,6-Quantil kann ausgesagt werden, dass 60 % der Werte innerhalb der Urdaten gleich oder kleiner und 40 % gleich oder größer dieses Werts sind.

KGRÖSSTE(), KKLEINSTE(), MAX(), MEDIAN(), MIN(), QUARTILSRANG(), QUARTILE()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Quantile*.

Siehe auch



QUANTIL.EXKL()



PERCENTILE.EXC()

QUANTIL.EXKL(*Array*; *k*)

Die Funktion QUANTIL.EXKL() gibt das *k*-Quantil von Werten in einem Bereich zurück, wobei *k* im Bereich von 0...1 ausschließlich liegt.

Array (erforderlich) ist ein Datenbereich, der die relative Lage der Daten beschreibt.

k (erforderlich) ist der Quantilwert aus dem geschlossenen Intervall von 0 bis ausschließlich 1.

Enthält Array keine Datenpunkte, gibt QUANTIL.EXKL() den Fehlerwert #ZÄHL! zurück.

Is *k* kein numerischer Ausdruck, gibt QUANTIL.EXKL() den Fehlerwert #WERT! zurück.

Bei $k \leq 0$ oder $k \geq 1$ gibt QUANTIL.EXKL() den Fehlerwert #ZÄHL! zurück.

Is *k* kein Vielfaches von $1/(n - 1)$, interpoliert QUANTIL.EXKL(), um das *k*-Quantil zu bestimmen.

QUANTIL.EXKL() interpoliert, wenn der Wert für das angegebene Quantil zwischen zwei Werten im Array liegt. Wenn keine Interpolation für das Quantil möglich und *k* angegeben ist, gibt Excel den Fehlerwert #ZÄHL! zurück.

Mehr Informationen zum Thema Quantile finden Sie im gleichnamigen Abschnitt auf Seite 479.

Mehr zum Einsatz der Funktion erfahren Sie mithilfe des Praxiseinsatzes der Funktion QUANTIL() auf Seite 479.

KGRÖSSTE(), KKLEINSTE(), MAX(), MEDIAN(), MIN(), QUARTILSRANG(), QUARTILE.EXKL(), QUARTILE.INKL()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Quantil.exkl*.

Syntax

Definition

Argumente

Hinweis

Hintergrund

Praxiseinsatz

Siehe auch



QUANTIL.INKL()



PERCENTILE.INC()

QUANTIL.INKL(*Array*; *k*)

Die Funktion QUANTIL.INKL() gibt das *k*-Quantil von Werten in einem Bereich zurück, wobei *k* im Bereich von 0...1 einschließlich liegt.

Mithilfe dieser Funktion können Sie einen Akzeptanzschwellenwert festlegen. So könnten Sie beispielsweise entscheiden, dass nur Kandidaten untersucht werden, deren Prüfungsergebnisse oberhalb des 90 %-Quantils liegen.

Array (erforderlich) ist ein Datenbereich, der die relative Lage der Daten beschreibt.

k (erforderlich) ist der Quantilwert aus dem geschlossenen Intervall von 0 bis 1.

Syntax

Definition

Argumente

Hinweis Enthält Array keine Datenpunkte, gibt QUANTIL.INKL() den Fehlerwert #ZAH! zurück.
Ist k kein numerischer Ausdruck, gibt QUANTIL.INKL() den Fehlerwert #WERT! zurück.
Ist $k < 0$, oder ist $k > 1$, gibt QUANTIL.INKL() den Fehlerwert #ZAH! zurück.
Ist k kein Vielfaches von $1/(n - 1)$, interpoliert QUANTIL.INKL(), um das k -Quantil zu bestimmen.

Hintergrund Mehr Informationen zum Thema Quantile finden Sie im gleichnamigen Abschnitt auf Seite 479.

Praxiseinsatz Mehr zum Einsatz der Funktion erfahren Sie mithilfe des Praxiseinsatzes der Funktion QUANTIL() auf Seite 479.

Siehe auch KGRÖSSTE(), KKLINSTE(), MAX(), MEDIAN(), MIN(), QUARTILSRANG(), QUARTILE.EXKL(), QUARTILE.INKNL()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Quantil.inkl*.

QUANTILSRANG() PERCENTRANK()

Hinweis Die Funktion QUANTILSRANG() wird ab Microsoft Excel 2010 durch die Funktionen QUANTILSRANG.EXKL() und QUANTILSRANG.INKL() ersetzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht.

Um die Abwärtskompatibilität von QUANTILSRANG.EXKL() und QUANTILSRANG.INKL() zu sichern, ist die Funktionen QUANTILSRANG() weiter unter ihrem alten Namen verfügbar.

Syntax QUANTILSRANG(*Matrix*; *x*; *Genauigkeit*)

Definition Die Funktion QUANTILSRANG() gibt den prozentualen Rang (Alpha) eines Werts zurück. Diese Funktion kann dazu verwendet werden, die relative Position zu ermitteln, die ein Wert innerhalb einer Datenmenge einnimmt. So können Sie beispielsweise mithilfe von QUANTILSRANG() ermitteln, welche relative Position das Ergebnis einer Eingangsuntersuchung innerhalb der Ergebnisse aller Untersuchungen einnimmt.

Argumente *Matrix* (erforderlich) ist die Matrix oder der Bereich numerischer Daten, die oder der die relative Lage der Daten beschreibt.

x (erforderlich) ist der Wert, dessen Rang Sie bestimmen möchten.

Genauigkeit (optional) ist ein Wert, der die Anzahl der Nachkommastellen des zurückgegebenen Quantilsrangs festlegt. Fehlt dieses Argument, verwendet QUANTILSRANG() drei Dezimalstellen (0,xxx).

Hinweis Ist die *Matrix* leer, liefert QUANTILSRANG() den Fehlerwert #ZAH!. Ist *Genauigkeit* kleiner 1, liefert QUANTILSRANG() den Fehlerwert #ZAH!.

Ist *x* nicht mit einem der in *Matrix* stehenden Werte identisch, interpoliert QUANTILSRANG(), um den entsprechenden Quantilsrang zu bestimmen.

Hintergrund Die Funktion QUANTILSRANG() ist die Umkehrfunktion von QUANTIL(). Mithilfe dieser Funktion können Sie die relative Position a des Werts x innerhalb einer Wertereihe ermitteln. Mehr Hintergrundinformationen zu den Quantilen finden Sie auf Seite 479.

Der Controllingleiter des Softwareherstellers möchte den Gesamtumsatz, also die Summe aller in verschiedenen Abteilungen getätigten Umsätze innerhalb eines Jahres, näher analysieren.

Sein Ziel ist es darzustellen, welche Position ein bestimmter Umsatz in einem bestimmten Monat aus der Menge aller Umsätze auf einer Skala von 1 bis 100 einnimmt. Allgemein ausgedrückt möchte er zeigen, wie die einzelnen Monatsumsätze im Vergleich zu der Gesamtheit aller Umsätze streuen. Hierzu bedient er sich der Funktion QUANTILSRANG().

	B	C	D	E	F	G	H
9	Umsatz innerhalb verschiedener Unternehmensbereiche in einem Jahr						
10							
11	UMSÄTZE	EVENTS	PUBLIKATIONEN	TEAM	TRAINING	GESAMT	QUANTILSRANG
12	Januar	1.127 €	1.141,00 €	1.049,00 €	1.339,00 €	4.656,00 €	0,55
13	Februar	1.407 €	945,00 €	938,00 €	1.014,00 €	4.304,00 €	0,27
14	März	1.334 €	1.388,00 €	902,00 €	1.153,00 €	4.777,00 €	0,82
15	April	981 €	1.353,00 €	897,00 €	1.441,00 €	4.672,00 €	0,64
16	Mai	1.223 €	813,00 €	1.464,00 €	1.481,00 €	4.981,00 €	0,91
17	Juni	1.322 €	1.358,00 €	1.385,00 €	1.166,00 €	5.231,00 €	1,00
18	Juli	1.316 €	982,00 €	892,00 €	1.015,00 €	4.205,00 €	0,09
19	August	1.168 €	896,00 €	1.000,00 €	1.035,00 €	4.099,00 €	-
20	September	1.212 €	1.243,00 €	1.050,00 €	1.077,00 €	4.582,00 €	0,45
21	Oktober	1.218 €	1.088,00 €	998,00 €	1.446,00 €	4.750,00 €	0,73
22	November	1.311 €	945,00 €	1.222,00 €	828,00 €	4.306,00 €	0,36
23	Dezember	1.421 €	903,00 €	1.087,00 €	854,00 €	4.265,00 €	0,18

Abbildung 11.108: Jeweiliger Quantilsrang für die Monatsumsätze verschiedener Bereiche

Nimmt man z.B. den Umsatz von 4.656,00 € im Monat Januar und den errechneten Quantilsrang von 0,55, so kann man folgende Aussage treffen: Der Umsatz von 4.656,00 € im Januar aus meinem Gesamtbereich aller Umsätze liegt auf einer Skala von 1 bis 100 bei 55. Das heißt, 55 % der Werte des Gesamtbereichs sind kleiner oder gleich bzw. 45 % der Werte sind größer oder gleich 4656,00 €.

KGRÖSSTE(), KKLINSTE(), MAX(), MEDIAN(), MIN(), QUANTIL(), QUANTIL.EXKL(), QUANTIL.INKL(), QUARTILE(), QUARTILE.EXKL(), QUARTILE.INKL()

Siehe auch

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Quantilsrang*.



QUANTILSRANG.EXKL()



QUANTILSRANG.EXKL(*Matrix*; *x*; *Genauigkeit*)

Syntax

Die Funktion QUANTILSRANG.EXKL() gibt den prozentualen (0..1 ausschließlich) Rang (Alpha) eines Werts in einem Dataset zurück.

Definition

Matrix (erforderlich) ist die Matrix oder der Bereich numerischer Daten, die oder der die relative Lage der Daten beschreibt.

Argumente

x (erforderlich) ist der Wert, dessen Rang Sie bestimmen möchten.

Genauigkeit (optional) ist ein Wert, der die Anzahl der Nachkommastellen des zurückgegebenen Quantilsrangs festlegt. Fehlt dieses Argument, verwendet QUANTILSRANG() drei Dezimalstellen (0,xxx).

Hinweis Ist Matrix leer, gibt QUANTILSRANG.EXKL() den Fehlerwert #ZAH! zurück.
Ist *Genauigkeit* < 1, gibt QUANTILSRANG.EXKL() den Fehlerwert #ZAH! zurück.
Ist x nicht mit einem der in Array stehenden Werte identisch, interpoliert QUANTILSRANG.EXKL(), um den entsprechenden Quantilsrang zu bestimmen.

Hintergrund Die Funktionen QUANTILSRANG.EXKL() und QUANTILSRANG.INKL() sind die Umkehrfunktionen von QUANTIL.EXKL() und QUANTIL.INKL(). Mithilfe dieser Funktion können Sie die relative Position *a* des Werts *x* innerhalb einer Wertereihe ermitteln.

Mehr Hintergrundinformationen zu den Quantilen finden Sie auf Seite 479.

Praxiseinsatz Mehr zum Einsatz der Funktion erfahren Sie mithilfe des Praxiseinsatzes der Funktion QUANTILSRANG() auf Seite 482.

Siehe auch KGRÖSSTE(), KLEINSTE(), MAX(), MEDIAN(), MIN(), QUANTIL.EXKL(), QUANTIL.INKL(), QUANTILE.EXKL(), QUANTILE.INKL()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Quantilsrang.exkl*.

QUANTILSRANG.INKL() PERCENTRANK.INC()

Syntax QUANTILSRANG.INKL(*Matrix*; *x*; *Genauigkeit*)

Definition Die Funktion QUANTILSRANG.INKL() gibt den prozentualen (0...1 einschließlich) Rang (Alpha) eines Werts in einem Dataset zurück.

Diese Funktion kann dazu verwendet werden, die relative Position zu ermitteln, die ein Wert innerhalb einer Datenmenge einnimmt. So können Sie beispielsweise mithilfe von QUANTILSRANG.INKL() ermitteln, welche relative Position das Ergebnis einer Eingangsuntersuchung innerhalb der Ergebnisse aller Untersuchungen einnimmt.

Argumente *Matrix* (erforderlich) ist die Matrix oder der Bereich numerischer Daten, die oder der die relative Lage der Daten beschreibt.

x (erforderlich) ist der Wert, dessen Rang Sie bestimmen möchten.

Genauigkeit (optional) ist ein Wert, der die Anzahl der Nachkommastellen des zurückgegebenen Quantilsrangs festlegt. Fehlt dieses Argument, verwendet QUANTILSRANG.INKL() drei Dezimalstellen (0,xxx).

Hinweis Ist Matrix leer, gibt QUANTILSRANG.INKL() den Fehlerwert #ZAH! zurück.

Ist *Genauigkeit* < 1, gibt QUANTILSRANG.INKL() den Fehlerwert #ZAH! zurück.

Ist x nicht mit einem der in Matrix stehenden Werte identisch, interpoliert QUANTILSRANG.INKL(), um den entsprechenden Quantilsrang zu bestimmen.

Hintergrund Die Funktionen QUANTILSRANG.EXKL() und QUANTILSRANG.INKL() sind die Umkehrfunktionen von QUANTIL.EXKL() und QUANTIL.INKL(). Mithilfe dieser Funktion können Sie die relative Position *a* des Werts *x* innerhalb einer Wertereihe ermitteln.

Mehr Hintergrundinformationen zu den Quantilen finden Sie auf Seite 479.

Mehr zum Einsatz der Funktion erfahren Sie mithilfe des Praxiseinsatzes der Funktion QUANTILSRANG() auf Seite 482.

Praxiseinsatz

KGRÖSSTE(), KKLINSTE(), MAX(), MEDIAN(), MIN(), QUANTILE.EXKL(), QUANTILE.INKL(), QUARTILE.EXKL(), QUARTILE.INKL()

Siehe auch

Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Quantilsrang.inkl*.



QUARTILE()



QUARTILE()

Die Funktion QUARTILE() wird ab Microsoft Excel 2010 durch die Funktionen QUARTILE.EXKL() und QUARTILE.INKL() ersetzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht.

Hinweis

Um die Abwärtskompatibilität von QUARTILE.EXKL() und QUARTILE.INKL() zu sichern, ist die Funktion QUARTILE() weiter unter ihrem alten Namen verfügbar.

QUARTILE(*Matrix*; *Quartil*)

Syntax

Die Funktion QUARTILE() gibt die Quartile einer Datengruppe zurück. Quartile werden häufig bei beispielsweise Verkaufs- oder Umfragedaten verwendet, um die Grundgesamtheiten in Gruppen einzuteilen und einen Überblick zu bieten. So können Sie als Beispiel mit QUARTILE() für eine Stichprobe erhobener Umsatzzahlen den Wert ermitteln, ab dessen Höhe ein Umsatz zu den oberen 25 Prozent der Umsatzkategorien gehört.

Definition

Über das Argument *Quartil* geben Sie an, welcher Wert ausgegeben werden soll. Die möglichen Werte für dieses Argument haben wir Ihnen in Tabelle 11.3 dargestellt.

Matrix (erforderlich) ist eine Matrix oder ein Zellbereich numerischer Werte, deren Quartile Sie bestimmen möchten.

Argumente

Quartil (erforderlich) gibt an, welcher Wert ausgegeben werden soll.

Ist die Matrix leer, gibt QUARTILE() den Fehlerwert #ZÄHL! zurück. Ist *Quartil* keine ganze Zahl, werden die Nachkommastellen abgeschnitten. Ist *Quartil* kleiner 0 oder ist *Quartil* größer 4, gibt QUARTILE() den Fehlerwert #ZÄHL! zurück.

Hinweis

MIN(), MEDIAN() bzw. MAX() geben denselben Wert wie QUARTILE() zurück, wenn *Quartil* mit 0, 2 beziehungsweise 4 belegt ist.

Neben dem Median kann man mit den Quantilen und Quartilen sehr gute Einteilungen der Daten berechnen. Während der Median die Verteilung genau halbiert, teilen Quantile sie in mehrere und Quartile in vier gleiche Teile auf. Das heißt, es lässt sich im Prinzip jeder beliebige Wert als Abschneidegrenze verwenden, um eine Einteilung der Daten vorzunehmen.

Hintergrund

Die Funktion QUARTILE() verlangt neben dem Argument *Matrix*, also dem zu untersuchenden Bereich, auch das *Quartil*, das zurückgegeben soll. Dies kann ein Wert zwischen 0 und 4 sein.

Die Bedeutung des Parameters Quartil finden Sie in nachfolgender Tabelle:

Tabelle 11.3	Wert ...	Liefert Ergebnis ...
Die Argumente für das Argument <i>Quartile</i>	0	den kleinsten Wert (Minimum)
	1	das untere Quartil (25 %-Quantil)
	2	den Median (50 %-Quantil)
	3	das obere Quartil (75 %-Quantil)
	4	den größten Wert (Maximum)

Wenn das Quartil 0 dem Minimum entspricht, das Quartil 4 dem Maximum und das Quartil 2 dem Median, ist es keine Kunst mehr, die Quartile 1 und 3 zu deuten.

Praxiseinsatz Ein Pharmaunternehmen, das unter anderem Kopfschmerztabletten produziert und diese im gesamten Bundesgebiet vertreibt, möchte den Umsatz der Tabletten in den Bundesländern Bayern, Hessen, Berlin, Bremen und Nordrhein-Westfalen pro 100.000 Einwohner je Bundesland auswerten.

Ziel ist es, eine Aussage darüber zu treffen, bis zu welchem Betrag 25 % des Umsatzes in einem bestimmten Bundesland innerhalb eines Jahrs getätigt werden.

Pro Bundesland liegen die Umsätze für ein ganzes Jahr, also zwölf Werte, vor. Bei der Berechnung der Quartile werden die Werte aufsteigend sortiert und anschließend, wie in Tabelle 11.3 dargestellt, errechnet.

	B	C	D	E	F	G
8	Umsatz Kopfschmerztabletten pro 100.000 Einwohner					
9						
10	Monat	Bayern	Hessen	Berlin	Bremen	NRW
11	Januar	800,00 €	236,00 €	189,00 €	987,00 €	1.563,00 €
12	Februar	1.190,00 €	356,00 €	505,00 €	1.319,00 €	3.000,00 €
13	März	1.545,00 €	756,00 €	1.307,00 €	2.233,00 €	6.116,00 €
14	April	1.168,00 €	518,00 €	884,00 €	1.903,00 €	2.860,00 €
15	Mai	1.139,00 €	684,00 €	835,00 €	1.589,00 €	3.126,00 €
16	Juni	5.083,00 €	1.678,00 €	820,00 €	2.428,00 €	6.682,00 €
17	Juli	3.609,00 €	844,00 €	276,00 €	1.559,00 €	3.311,00 €
18	August	4.810,00 €	622,00 €	365,00 €	1.502,00 €	3.317,00 €
19	September	5.581,00 €	893,00 €	309,00 €	1.628,00 €	3.542,00 €
20	Oktober	5.506,00 €	724,00 €	259,00 €	1.440,00 €	2.971,00 €
21	November	11.786,00 €	1.046,00 €	447,00 €	2.255,00 €	4.735,00 €
22	Dezember	6.227,00 €	339,00 €	234,00 €	1.583,00 €	2.652,00 €
23	Quartil 0 = Min	800	236	189	987	1.563
24	Quartil 1	1.185	478	272	1.487	2.943
25	Quartil 2 = Median	4.210	704	406	1.586	3.219
26	Quartil 3	5.525	856	824	1.986	3.840
27	Quartil 4 = Max	11.786	1.678	1.307	2.428	6.682

Abbildung 11.109: Die Jahresumsätze der einzelnen Bundesländer werden in Quartile unterteilt, um so eine bessere Übersicht zu erhalten

Betrachtet man, wie in Abbildung 11.109 dargestellt, die Umsätze für Bayern, so erkennt man, dass der kleinste Wert dem Minimum entspricht, der größte dem Maximum und das Quartil 2, also die 50 %-Grenze und gleichzeitig der Median, dem Wert 4.210.

Wie kommt man auf das 2. Quartil ohne Zuhilfenahme der Funktion QUARTILE()? Da Sie eine Zahlenreihe mit einer geraden Anzahl von Werten haben, muss die 50 %-Grenze irgendwo zwischen dem fünften und sechsten Wert liegen. Addieren Sie die beiden Werte und dividieren das Ergebnis durch zwei, erhalten Sie den Median. Gäbe es 13 Werte, läge der Median bzw. das 2. Quartil an siebter Stelle der sortierten Zahlenreihe.

Sie haben nun bereits folgende Quartile vorliegen:

- ▶ Quartil 0 = 800
- ▶ Quartil 2 = 4.210
- ▶ Quartil 4 = 11.786

Fehlen noch Quartil 1 und Quartil 3, die die 25 %- und 75 %-Grenze in der angegebenen Zahlenreihe definieren. Das heißt, es gilt, ausgehend von der sortierten Urdatenliste, die Werte an der Position 0,25 und 0,75 zu definieren.

Betrachten wir die Abbildung 11.109, müssen nun die Werte berechnet werden, von denen gesagt wird, dass 25 % der Beobachtungen kleiner oder gleich (1. Quartil) und 75 % der Beobachtungen größer oder gleich (3. Quartil) diesem Wert sind.

In unserem Beispiel haben wir zwölf Zahlenwerte (n). Multipliziert man n mit 0,25 sowie mit 0,75, kann man die Aussage treffen, dass das 1. Quartil zwischen dem dritten und vierten und das 3. Quartil zwischen dem neunten und zehnten Wert der sortierten Urdaten liegen muss.

Die Excelfunktion QUARTIL() liefert für das 1. und 3. Quartil die folgenden Ergebnisse:

- ▶ 1. Quartil = 1.185
- ▶ 3. Quartil = 5.525

Wie Sie sehen können, wird die zuvor getätigte Aussage durch die Ergebnisse der Excel-Berechnung untermauert. So liegt beispielsweise das 1. Quartil zwischen dem dritten und vierten Wert der Datenreihe, jedoch nicht in der Mitte, sondern eher in Richtung des vierten Werts.

Im Bezug auf Abbildung 11.109 können Sie nun z.B. folgende Aussagen treffen:

- ▶ In Bayern werden 25 % des Umsatzes bis 1.185,00 € erreicht
- ▶ In Nordrhein-Westfalen werden 75 % des Umsatzes bis 3.840,00 € erreicht

KGRÖSSTE(), KKLINSTE(), MAX(), MEDIAN(), MIN(), QUANTIL(), QUANTIL.EXKL(), QUANTIL.INKL(), QUANTILSRANG(), QUANTILSRANG.EXKL(), QUANTILSRANG.INKL()

Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe `Mittelwert.xlsx` (Excel 2007/2010) auf dem Arbeitsblatt `Quartile`.

Aussagen

Siehe auch



QUARTILE.EXKL() QUARTILE.EXC()

Syntax QUARTILE.EXKL(*Matrix*; *Quartile*)

Definition Die Funktion QUARTILE.EXKL() gibt die Quartile eines Datensets zurück, basierend auf Perzentilwerten von 0...1 ausschließlich.

Argumente *Matrix* (erforderlich) ist eine Matrix oder ein Zellbereich numerischer Werte, deren Quartile Sie bestimmen möchten.

Quartile (erforderlich) gibt an, welcher Wert ausgegeben werden soll.

Hinweis Ist Matrix leer, gibt QUARTILE.EXKL() den Fehlerwert #ZÄHL! zurück.

Ist Quartile keine ganze Zahl, werden die Nachkommastellen abgeschnitten.

Ist Quartile ≤ 0 oder ist Quartile ≥ 4 , gibt QUARTILE.EXKL() den Fehlerwert #ZÄHL! zurück.

MIN(), MEDIAN() bzw. MAX() geben denselben Wert wie QUARTILE.EXKL() zurück, wenn Quartile mit 0 (Null), 2 beziehungsweise 4 belegt ist.

Hintergrund Mehr Informationen zu den Quartilen lesen Sie im gleichnamigen Abschnitt auf Seite 485.

Praxiseinsatz Mehr zum Einsatz der Funktion erfahren Sie mithilfe des Praxiseinsatzes der Funktion QUARTILE() auf Seite 485.

Siehe auch KGRÖSSTE(), KKLLEINSTE(), MAX(), MEDIAN(), MIN(), QUANTIL(), QUANTIL.EXKL(), QUANTIL.INKL(), QUANTILSRANG(), QUANTILSRANG.EXKL(), QUANTILSRANG.INKL()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Quartile.exkl*.

QUARTILE.INKL() QUARTILE.INC()

Syntax QUARTILE.INKL(*Matrix*; *Quartile*)

Definition Gibt die Quartile eines Datensets zurück, basierend auf Perzentilwerten von 0...1 einschließlich.

Argumente *Matrix* (erforderlich) ist eine Matrix oder ein Zellbereich numerischer Werte, deren Quartile Sie bestimmen möchten.

Quartile (erforderlich) gibt an, welcher Wert ausgegeben werden soll.

Hinweis Ist Matrix leer, gibt QUARTILE.INKL() den Fehlerwert #ZÄHL! zurück.

Ist Quartil keine ganze Zahl, werden die Nachkommastellen abgeschnitten.

Ist Quartil < 0 , oder ist Quartil > 4 , gibt QUARTILE.INKL() den Fehlerwert #ZÄHL! zurück.

MIN(), MEDIAN() bzw. MAX() geben denselben Wert wie QUARTILE.INKL() zurück, wenn Quartil mit 0 (Null), 2 beziehungsweise 4 belegt ist.

Hintergrund Mehr Informationen zu den Quartilen lesen Sie im gleichnamigen Abschnitt auf Seite 485.

Praxiseinsatz Mehr zum Einsatz der Funktion erfahren Sie mithilfe des Praxiseinsatzes der Funktion QUARTILE() auf Seite 485.

KGRÖSSTE(), KKLEINSTE(), MAX(), MEDIAN(), MIN(), QUANTIL(), QUANTIL.EXKL(), QUANTIL.INKL(), QUANTILSRANG(), QUANTILSRANG.EXKL(), QUANTILSRANG.INKL()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Mittelwert.xlsx* auf dem Arbeitsblatt *Quartile.inkl*.



Siehe auch

RANG()



RANK()

Die Funktion RANG() wird seit Microsoft Excel 2010 durch die Funktionen RANG.GLEICH() und RANG.MITTELW() ersetzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht. Um die Abwärtskompatibilität von RANG.GLEICH() und RANG.MITTELW() zu sichern, ist die Funktion RANG() weiter unter ihrem alten Namen verfügbar.

Hinweis

RANG(*Zahl*; *Bezug*; *Reihenfolge*)

Syntax

RANG() gibt den Rang zurück, den eine Zahl innerhalb einer Liste von Zahlen einnimmt. Als Rang einer Zahl wird deren Größe, bezogen auf die anderen Werte der jeweiligen Liste, bezeichnet. Wenn Sie die Liste sortieren würden, würde die Rangzahl der einzelnen Werte deren Position angeben.

Definition

Zahl (erforderlich) ist die Zahl, deren Rangzahl Sie bestimmen möchten.

Argumente

Bezug (erforderlich) ist eine Matrix mit Zahlen oder ein Bezug auf eine Liste von Zahlen. Nicht numerische Werte im Bezug werden hierbei ignoriert.

Reihenfolge (optional) ist eine Zahl, die angibt, wie der Rang von Zahl bestimmt werden soll. Ist *Reihenfolge* mit Null belegt oder nicht angegeben, wird der Rang der angegebenen Zahl so bestimmt, als wäre Bezug eine in absteigender Reihenfolge sortierte Liste.

Ist *Reihenfolge* mit einem Wert ungleich Null belegt, wird der Rang von Zahl so bestimmt, als wäre Bezug eine in aufsteigender Reihenfolge sortierte Liste.

Diese Funktion ist vor allem bei großen Datenmengen sehr empfehlenswert und zeitsparend, da die Rangverteilung »per Hand« in solch einem Fall sehr schwer und unübersichtlich wird.

Hintergrund

Zu beachten ist, dass RANG() zwei gleichen Zahlen denselben Rang zuordnet. Der darauf folgende Rang wird dann übersprungen. Gelegentlich kann es jedoch auch wünschenswert sein, eine Rangdefinition zu verwenden, die gleiche Werte berücksichtigt. Dies lässt sich erreichen, indem ein Korrekturfaktor zu dem Wert addiert wird, den RANG() zurückgibt. Dieser Korrekturfaktor ist sowohl geeignet, wenn der Rang in absteigender Reihenfolge (*Reihenfolge* = 0 oder nicht angegeben) berechnet wird, als auch, wenn er in aufsteigender Reihenfolge (*Reihenfolge* = Wert ungleich 0) berechnet wird.

Der Korrekturfaktor für wertgleiche Ränge lautet:

$$[ANZAHL(Bezug) + 1 - RANG(Zahl; Bezug; 0) - RANG(Zahl; Bezug; 1)]/2$$

Wenn Sie nicht den Rang, sondern den Wert des zweitgrößten bzw. zweitkleinsten Werts ermitteln möchten, verwenden Sie dazu die Funktionen KGRÖSSTE() bzw. KKLEINSTE().

Hinweis

Praxiseinsatz Der Softwarehersteller hat eine Tabelle mit den Umsätzen der letzten zwei Jahre erstellt. Da es den Geschäftsführer interessiert, welcher Monat welchen Platz erreicht hat, wollen Sie nun eine Rangfolge unter den einzelnen Umsätzen erstellen, um anschließend eine entsprechende Aussage treffen zu können.

	B	C	D	E	F	G
	Fragestellung:					
1	Wie ist die Rangfolge des Umsatzes?					
2	Monat	Umsatz	Rang			
3	Januar 07	107.629 €	24		=RANG(C3;\$C\$3:\$C\$26)	
4	Februar 07	185.385 €	18			
5	März 07	180.807 €	19		Hinweis:	
6	April 07	124.328 €	22		Um einen absoluten Bezug zum Bereich zu	
7	Mai 07	146.215 €	21		bekommen, drücken Sie nach Angabe des	
8	Juni 07	185.675 €	17		entsprechenden Bereiches die Taste F4.	
9	Juli 07	210.169 €	11		Sie erkennen absolute Bezüge an den Dollar-	
10	August 07	221.729 €	9		Zeichen innerhalb der Klammer.	
11	September 07	234.187 €	7			
12	Oktober 07	237.947 €	6			
13	November 07	210.088 €	12			
14	Dezember 07	207.791 €	14			
15	Januar 08	201.097 €	16			
16	Februar 08	222.460 €	8			
17	März 08	208.585 €	13			
18	April 08	206.387 €	15			
19	Mai 08	218.951 €	10			
20	Juni 08	365.425 €	5			
21	Juli 08	542.896 €	2			
22	August 08	365.478 €	4			
23	September 08	165.845 €	20			
24	Oktober 08	642.598 €	1			
25	November 08	432.695 €	3			
26	Dezember 08	123.458 €	23			

Abbildung 11.110: Mit der Formel *RANG()* generieren Sie die Rangverteilung unter den Umsätzen

In der Spalte *Rang* werden die Ränge 1 bis 24 entsprechend der Umsätze in den einzelnen Monaten verteilt. Da alle Umsätze unterschiedlich hoch waren, wurden alle Ränge belegt.

Siehe auch Keine



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *Rang*.

RANG.GLEICH() RANK.EQ()

Syntax RANG.GLEICH(*Zahl*; *Bezug*; *Reihenfolge*)

Definition RANG.GLEICH() gibt den Rang zurück, den eine Zahl innerhalb einer Liste von Zahlen einnimmt. Als Rang einer Zahl wird deren Größe, bezogen auf die anderen Werte der jeweiligen Liste, bezeichnet. Wenn Sie die Liste sortieren würden, würde die Rangzahl der einzelnen Werte deren Position angeben.

Argumente *Zahl* (erforderlich) ist die Zahl, deren Rangzahl Sie bestimmen möchten.

Bezug (erforderlich) ist eine Matrix mit Zahlen oder ein Bezug auf eine Liste von Zahlen. Nicht numerische Werte im Bezug werden hierbei ignoriert.

Reihenfolge (optional) ist eine Zahl, die angibt, wie der Rang von Zahl bestimmt werden soll. Ist *Reihenfolge* mit Null belegt oder nicht angegeben, wird der Rang der angegebenen Zahl so bestimmt, als wäre Bezug eine in absteigender Reihenfolge sortierte Liste.

Ist *Reihenfolge* mit einem Wert ungleich Null belegt, wird der Rang von Zahl so bestimmt, als wäre Bezug eine in aufsteigender Reihenfolge sortierte Liste.

Mehr Informationen zum Rang lesen Sie im gleichnamigen Abschnitt auf Seite 489.

Mehr zum Einsatz der Funktion erfahren Sie mithilfe des Praxiseinsatzes der Funktion RANG() auf Seite 489.

Keine

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *Rang.gleich*.

Hintergrund

Praxiseinsatz

Siehe auch



RANG.MITTELW() RANK.AVG()

RANG.MITTELW(*Zahl*; *Bezug*; *Reihenfolge*)

RANG.MITTELW() gibt den Rang zurück, den eine Zahl innerhalb einer Liste von Zahlen einnimmt. Als Rang einer Zahl wird deren Größe, bezogen auf die anderen Werte der jeweiligen Liste, bezeichnet.

Wenn mehrere Werte die gleiche Rangzahl aufweisen, wird die durchschnittliche Rangzahl zurückgegeben.

Zahl (erforderlich) ist die Zahl, deren Rangzahl Sie bestimmen möchten.

Bezug (erforderlich) ist eine Matrix mit Zahlen oder ein Bezug auf eine Liste von Zahlen. Nicht numerische Werte im Bezug werden hierbei ignoriert.

Reihenfolge (optional) ist eine Zahl, die angibt, wie der Rang von Zahl bestimmt werden soll. Ist *Reihenfolge* mit Null belegt oder nicht angegeben, wird der Rang der angegebenen Zahl so bestimmt, als wäre Bezug eine in absteigender Reihenfolge sortierte Liste.

Ist *Reihenfolge* mit einem Wert ungleich Null belegt, wird der Rang von Zahl so bestimmt, als wäre Bezug eine in aufsteigender Reihenfolge sortierte Liste.

Mehr Informationen zum Rang lesen Sie im gleichnamigen Abschnitt auf Seite 489.

Mehr zum Einsatz der Funktion erfahren Sie mithilfe des Praxiseinsatzes der Funktion RANG() auf Seite 489.

Keine

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *Rang.mittelw*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



RGP() LINEST()

Syntax RGP(*Y_Werte*; *X_Werte*; *Konstante*; *Stats*)

Definition Die Funktion RGP() berechnet die Statistik für eine Linie unter Verwendung der Methode der kleinsten Quadrate, zur Berechnung einer geraden Linie, die für die Daten am geeignetsten ist, und gibt eine Matrix zurück, die die Linie beschreibt. Da diese Funktion ein Wertarray liefert, muss die Formel als Matrixformel eingegeben werden.

Die Gleichung einer solchen Geraden lautet:

$$y = mx + b \text{ oder}$$

$$y = m_1x_1 + m_2x_2 + \dots + b \text{ (bei mehreren Bereichen mit } x\text{-Werten)}$$

Dabei ist der abhängige y -Wert eine Funktion der unabhängigen x -Werte. Die m -Werte sind Koeffizienten, die zu den jeweiligen x -Werten gehören, und b ist eine Konstante. Beachten Sie, dass y , x und m Vektoren sein können. Eine von RGP() ausgegebene Matrix hat die Form $\{mn; mn-1; \dots; m1; b\}$. RGP() kann darüber hinaus zusätzliche Regressionskenngrößen bereitstellen (siehe Tabelle 11.4).

Argumente *Y_Werte* (erforderlich) sind die y -Werte, die Ihnen bereits aus der Beziehung $y = mx + b$ bekannt sind.

- ▶ Besteht die Matrix *Y_Werte* aus nur einer Spalte, wird jede Spalte der Matrix *X_Werte* als eigenständige Variable interpretiert
- ▶ Besteht die Matrix *Y_Werte* aus nur einer Zeile, wird jede Zeile der Matrix *X_Werte* als eigenständige Variable interpretiert

X_Werte (optional) sind x -Werte, die Ihnen möglicherweise bereits aus der Beziehung $y = mx + b$ bekannt sind.

- ▶ Die Matrix *X_Werte* kann eine oder mehrere Variablengruppen umfassen. Wird nur eine Variable verwendet, können *Y_Werte* und *X_Werte* Bereiche beliebiger Form sein, solange sie dieselben Dimensionen haben. Werden mehrere Variablen verwendet, muss *Y_Werte* ein Vektor sein, das heißt ein Bereich, der aus nur einer Zeile oder nur einer Spalte besteht.
- ▶ Fehlt die Matrix *X_Werte*, wird an ihrer Stelle die Matrix $\{1; 2; 3; \dots\}$ angenommen, die genauso viele Elemente wie *Y_Werte* enthält.

Konstante (optional) ist ein Wahrheitswert, der angibt, ob die Konstante b den Wert 0 annehmen soll.

- ▶ Ist *Konstante* mit *WAHR* belegt oder nicht angegeben, wird b normal berechnet
- ▶ Ist *Konstante* mit *FALSCH* belegt, wird b gleich 0 gesetzt, und die m -Werte werden so angepasst, dass sie zu der Beziehung $y = mx$ passen

Stats (optional) ist ein Wahrheitswert, der angibt, ob zusätzliche Regressionskenngrößen ausgegeben werden sollen.

- ▶ Ist *Stats* mit *WAHR* belegt, liefert RGP() weitere Regressionskenngrößen, sodass eine wie folgt aufgebaute Matrix zurückgegeben wird: $\{mn; mn-1; \dots; m1; b; sen; sen-1; \dots; se1; seb.r2; sey.F; df.ssreg; ssresid\}$
- ▶ Ist *Stats* mit *FALSCH* belegt oder nicht angegeben, liefert RGP() nur die m -Koeffizienten sowie die Konstante b

Die folgenden Regressionskenngrößen (-statistiken) können zusätzlich ermittelt werden:

Kenngröße (Statistik)	Beschreibung
se_1, se_2, \dots, se_n	Sind die Standardfehler der Koeffizienten $m_1; m_2; \dots; m_n$
se_b	Der Standardfehler der Konstanten b ($se_b = \#NV$, wenn Konstante mit <i>FALSCH</i> belegt ist)
r^2	Das Bestimmtheitsmaß. Vergleicht die berechneten mit den tatsächlichen y -Werten und kann Werte von 0 bis 1 annehmen. Hat es den Wert 1, besteht für die Stichprobe eine vollkommene Korrelation: Ein berechneter y -Wert und der entsprechende tatsächliche y -Wert unterscheiden sich nicht. Im anderen Extremfall, wenn das Bestimmtheitsmaß 0 ist, ist die Regressionsgerade ungeeignet, einen y -Wert vorherzusagen. Informationen darüber, wie r^2 berechnet wird, finden Sie unter BESTIMMTHEITSMASS() auf Seite 352.
se_y	Der Standardfehler des Schätzwerts y (Prognosewert)
F	Die F-Statistik (oder der berechnete F-Wert). Anhand der F-Statistik können Sie entscheiden, ob die zwischen der abhängigen und der unabhängigen Variablen beobachtete Beziehung zufällig ist oder nicht.
df	Der Freiheitsgrad. Mit diesem Freiheitsgrad können Sie den jeweiligen kritischen F-Wert (Quantil F) einer entsprechenden statistischen Tabelle entnehmen. Vergleichen Sie den jeweils auf solche Weise ermittelten kritischen F-Wert mit der von RGP() gelieferten F-Statistik, um das Konfidenzniveau Ihres Modells zu beurteilen.
ss_{reg}	Die Regressions-Quadratsumme
ss_{resid}	Die Residual-Quadratsumme (Summe der Abweichungsquadrate)

Tabelle 11.4

Die Regressionskenngrößen im Überblick

Jede Gerade lässt sich durch ihre Steigung und die jeweilige Anfangsordinate (y -Achsenabschnitt) beschreiben.

Hinweis

1. Steigung (m): Die Steigung einer Geraden (häufig als m bezeichnet) lässt sich aus zwei Punkten der Geraden, (x_1, y_1) und (x_2, y_2) , gemäß der Beziehung $(y_2 - y_1)/(x_2 - x_1)$ berechnen.
2. y -Achsenabschnitt (b): Der y -Achsenabschnitt (häufig als b bezeichnet) ist der y -Wert des Punkts, in dem die Gerade die y -Achse schneidet.

Detaillierte Informationen zur Steigung und zum Achsenabschnitt finden Sie unter STEIGUNG() auf Seite 519 und unter ACHSENABSCHNITT() auf Seite 347.

Eine Gerade wird durch die Gleichung $y = mx + b$ beschrieben. Sobald Sie die Werte von m und b kennen, können Sie jeden Punkt der Geraden berechnen, indem Sie den jeweiligen y - oder x -Wert in die Gleichung einsetzen. Sie können dafür auch die Funktion TREND() verwenden (siehe Seite 528).

Während die Korrelation untersucht, ob die Variablen in einer wechselseitigen Beziehung zueinander stehen und dies durch eine einzige Zahl ausdrückt, versucht man mit der Regressionsanalyse, die Ausprägungen von einer abhängigen Variablen – in unserem Beispiel die Bestellungen – von einer unabhängigen Variablen, also die Websitezugriffe, zu bestimmen.

Hintergrund

Die Regressionsanalyse ist demnach ein Verfahren, mit dem man versucht, ein Merkmal (abhängige Variable) durch ein oder mehrere andere Merkmale (unabhängige Variable) zu erklären. Man geht dabei davon aus, dass die abhängige Variable immer um den gleichen Wert zu- oder abnimmt, wenn sich die unabhängige Variable um eine Einheit ändert (Linearität).

Es wird also eine einseitige Abhängigkeit bestimmt. Welche der beiden Variablen abhängig oder unabhängig ist, müssen Sie selbst abschätzen. Natürlich muss die Abhängigkeit plausibel sein.

Wenn nur eine unabhängige x -Variable vorliegt, können Sie die Steigung und den y -Achsenabschnitt direkt mithilfe der folgenden Formeln ermitteln:

- ▶ Steigung: $=INDEX(RGP(Bekannte_y_Werte;Bekannte_x_Werte);1)$
- ▶ y -Achsenabschnitt: $=INDEX(RGP(Bekannte_y_Werte;Bekannte_x_Werte);2)$

Die Genauigkeit einer von $RGP()$ berechneten Geraden hängt davon ab, wie sehr Ihre Daten streuen. Je linearer die Daten sind, desto genauer ist das von $RGP()$ ermittelte Modell. $RGP()$ verwendet die Methode der kleinsten Quadrate, um die für die jeweiligen Daten beste Anpassung zu ermitteln. Wenn nur eine unabhängige x -Variable vorliegt, werden m und b entsprechend der folgenden Formeln berechnet:

$$m = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2}}$$

Dabei sind x und y Stichprobenmittelwerte, d.h. $x = MITTELWERT(Bekannte_x_Werte)$ und $y = MITTELWERT(Bekannte_y_Werte)$.

Die Regressionsfunktionen $RGP()$ (lineare Regression) und $RKP()$ (exponentielle Regression) berechnen die Koeffizienten von Daten optimal angepasster Geraden bzw. Exponentialkurven.

Trotzdem ist es nach wie vor Ihre Aufgabe, zu entscheiden, welches der beiden Ergebnisse besser zu Ihren Daten passt.

Bei einer Regressionsanalyse berechnet Excel für jeden Punkt das Quadrat der Differenz, die zwischen dem für diesen Punkt berechneten y -Wert und dessen tatsächlichem y -Wert liegt. Die Summe dieser quadrierten Differenzen wird Residual-Quadratsumme (ss_{resid}) genannt.

Danach berechnet Microsoft Excel die Gesamtsumme der Quadrate (ss_{total}). Wenn *Konstante* = WAHR oder nicht angegeben ist, ist die Gesamtsumme der Quadrate gleich der Summe der quadrierten Differenzen, die zwischen den tatsächlichen y -Werten und dem Mittelwert der y -Werte liegen.

Wenn *Konstante* = FALSCH ist, ist die Gesamtsumme der Quadrate gleich der Summe der Quadrate der tatsächlichen y -Werte.

Die Regressions-Quadratsumme (ss_{reg}) lässt sich dann wie folgt berechnen:

$$ss_{reg} = ss_{total} - ss_{resid}$$

Je kleiner die Residual-Quadratsumme im Vergleich zur Gesamtsumme der Abweichungsquadrate ist, desto größer ist der Wert des Bestimmtheitsmaßes (r^2), das anzeigt, wie gut die aus der Regressionsanalyse resultierende Gleichung die zwischen den Variablen bestehende Beziehung beschreibt.

$$r^2 = \frac{ss_{reg}}{ss_{total}}$$

Es gibt Fälle, in denen eine oder mehrere der X -Spalten (unter der Annahme, die x - und y -Werte stehen in Spalten) zusammen mit den anderen X -Spalten keinen zusätzlichen Beitrag zu den Schätzwerten leisten. Anders formuliert bedeutet dies, dass das Herausnehmen dieser X -Spalte(n) zu geschätzten y -Werten führt, die ebenso genau sind.

In einem solchen Fall sollten die redundanten X -Spalten aus dem Regressionsmodell entfernt werden. Dieses Phänomen wird als »Kollinearität« bezeichnet, weil jede redundante X -Spalte als Summe von Vielfachen der nicht redundanten X -Spalten ausgedrückt werden kann. RGP() prüft auf Kollinearität und entfernt alle X -Spalten aus dem Regressionsmodell, die es als redundant ermittelt hat.

Eine entfernte X -Spalte kann in der Ausgabe von RGP() daran erkannt werden, dass sowohl ihre Koeffizienten als auch ihre se -Werte gleich 0 sind. Wurden eine oder mehrere Spalten wegen Redundanz entfernt, wirkt sich dies auf df aus, weil df von der Zahl der X -Spalten abhängt, die tatsächlich zum Berechnen der Schätzwerte verwendet werden.

Ändert sich der Wert von df , weil redundante X -Spalten entfernt wurden, wirkt sich dies auch auf die Werte von sey und F aus. Kollinearität tritt in der Praxis üblicherweise selten auf.

df wird, wenn keine X -Spalten wegen Kollinearität aus dem Modell entfernt wurden, wie folgt berechnet: Gibt es k Spalten und ist $Konstante = WAHR$ oder nicht angegeben, ist

$$df = n - k - 1$$

Ist $Konstante = FALSCH$, lautet die Gleichung

$$df = n - k$$

In beiden Fällen bewirkt jede X -Spalte, die wegen Kollinearität entfernt wurde, dass df um 1 erhöht wird.

Wird eine Matrix-Konstante (wie zum Beispiel X_Werte) als Argument eingegeben, müssen Sie **Punkte** verwenden, um Werte derselben Zeile zu trennen, und ein **Semikolon**, um die Zeilen zu trennen.

Beachten Sie, dass mithilfe einer Regressionsgleichung vorhergesagte y -Werte eventuell keinen Sinn machen, wenn diese außerhalb des Bereichs der y -Werte liegen, die Sie zur Ermittlung der Gleichung verwendet haben.

Tipp

Dieses Szenario ist heute bereits Standard: Ein in der Softwarebranche tätiges Unternehmen verkauft auf der firmeneigenen Website alle vom Unternehmen angebotenen Produkte. Regelmäßig verschickt das Unternehmen Newsletter, um über Neuheiten zu informieren und bereits vorhandene, aber auch potenzielle Neukunden auf die Website und damit auf das Unternehmen aufmerksam zu machen.

Praxiseinsatz

Die Produktbestellungen auf der Website sind im vergangenen Jahr merklich gestiegen. Die Geschäftsführung möchte die Gründe hierfür erforschen. Hängt der Anstieg der Bestellungen mit dem aktiven Marketing und den straffen Werbemaßnahmen der letzten Zeit zusammen? Sind durch die vermehrten Websitezugriffe auch die Bestellungen gestiegen?

Es soll also aufgezeigt werden, ob ein Zusammenhang zwischen den Websitezugriffen und den Bestellungen besteht. Mit einem regelmäßigen Websitecontrolling können Sie wichtige Aussagen über das Verhalten der Nutzer Ihrer Webseite liefern und somit eine effektive Erfolgskontrolle Ihrer Online- bzw. sonstigen Aktivitäten ermöglichen.

Die Marketingabteilung wird damit beauftragt, diese Analyse durchzuführen und erstellt eine Excel-Liste, in der die Websitezugriffe der letzten 18 Monate mit den in diesem Zeitraum eingehenden Bestellungen gegenübergestellt werden.

Anschließend soll eine Regressionsanalyse mit der Funktion `RGP()` durchgeführt werden.

	B	C	D
9	Monat	Webseitenzugriff	Bestellungen
10	Januar 2007	236	98
11	Februar 2007	11593	8000
12	März 2007	18491	6000
13	April 2007	11743	8587
14	Mai 2007	11452	7985
15	Juni 2007	26651	18968
16	Juli 2007	16287	9753
17	August 2007	17750	7857
18	September 2007	19985	13986
19	Oktober 2007	17285	6875
20	November 2007	30369	22765
21	Dezember 2007	19674	9465
22	Januar 2008	28464	19875
23	Februar 2008	25000	15987
24	März 2008	24574	9653
25	April 2008	23141	12986
26	Mai 2008	17700	8543
27	Juni 2008	3702	1654
28	Steigung	0,660130798	

Abbildung 11.111: Die Gegenüberstellung von Webseitenzugriffen und Bestellungen

Die Funktion `RGP()` liefert folgende Ergebnisse:

	B	C	D	E
39	b	m₁	0,660130798	-1383,85617
40	se_b	se₁	0,083627684	1642,68237
41	R²	se_y	0,795684605	2785,72374
42	F	d_f	62,31030061	16
43	SS_{reg}	SS_{resid}	483543930,9	124164108

Abbildung 11.112: Die Berechnung der Regressionsgrößen mithilfe der Funktion `RGP()`

Achtung Formeln, die als Ergebnis eine Matrix liefern, müssen als Matrixformeln eingegeben werden. Das heißt, da Sie über `RGP()` alle zehn Regressionsgrößen auf einmal errechnen können und somit eine Matrix ausgegeben wird, muss die Eingabe der Formel bzw. der Argumente unter bestimmten Kriterien erfolgen.

Markieren Sie zunächst, wie in Abbildung 11.112 dargestellt, die Zellen `D39:E43` und geben Sie anschließend in die Zelle `D39` die Argumente für die Funktion `RGP()` ein. Beenden Sie die Eingabe der Formel durch Klicken auf die Taste `[F2]` und anschließendes Drücken der Tastenkombination `[Strg] + [↕] + [↵]`.

Siehe auch `RKP()`, `TREND()`, `VARIATION()`



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe `Regression.xlsx` auf dem Arbeitsblatt `RGP`.

RKP() LOGEST()

RKP(*Y_Werte*; *X_Werte*; *Konstante*; *Stats*)

Syntax

In Regressionsanalysen berechnet die Funktion RKP() eine Exponentialkurve, die möglichst gut an die von Ihnen bereitgestellten Daten angepasst ist, und liefert ein Wertarray, die diese Kurve beschreibt. Da diese Funktion ein Wertarray liefert, muss die Formel als Matrixformel eingegeben werden.

Definition

Y_Werte (erforderlich) sind die y -Werte, die Ihnen aus der jeweiligen Beziehung $y = b * mx$ bereits bekannt sind.

Argumente

- ▶ Besteht die Matrix *Y_Werte* aus nur einer Spalte, wird jede Spalte der Matrix *X_Werte* als eigenständige Variable interpretiert
- ▶ Besteht die Matrix *Y_Werte* aus nur einer Zeile, wird jede Zeile der Matrix *X_Werte* als eigenständige Variable interpretiert

X_Werte (optional) ist eine optionale Gruppe von x -Werten, die Ihnen aus der Beziehung $y = b * mx$ eventuell bereits bekannt sind.

- ▶ Die Matrix *X_Werte* kann eine oder mehrere Gruppen von Variablen umfassen. Wird nur eine Variable verwendet, können *Y_Werte* und *X_Werte* Bereiche beliebiger Form sein, solange sie dieselben Dimensionen haben. Werden mehrere Variablen verwendet, müssen *Y_Werte* als Zellbereiche vorliegen, wobei sich der Bereich nur über eine Zeile oder eine Spalte erstrecken darf (auch als Vektor bezeichnet).
- ▶ Fehlt die Matrix *X_Werte*, wird an ihrer Stelle die Matrix $\{1;2;3;\dots\}$ angenommen, die genauso viele Elemente wie *Y_Werte* enthält

Konstante (optional) ist ein Wahrheitswert, der angibt, ob die Konstante b den Wert 1 annehmen soll.

- ▶ Ist *Konstante* mit *WAHR* belegt oder nicht angegeben, wird b normal berechnet
- ▶ Ist *Konstante* mit *FALSCH* belegt, wird b gleich 1 gesetzt, und die m -Werte werden gemäß $y = mx$ berechnet

Stats (optional) ist ein Wahrheitswert, der angibt, ob zusätzliche Regressionskenngrößen ausgegeben werden sollen.

- ▶ Ist *Stats* mit *WAHR* belegt, liefert RKP() diese zusätzlichen Regressionskenngrößen, sodass die zurückgegebene Matrix wie folgt aussieht:

```
{mn.mn-1. ... .m1.b;sen.sen-1. ... .se1.seb;r 2.sey;F.df.ssreg.ssresid}
```

- ▶ Ist *Stats* mit *FALSCH* belegt oder nicht angegeben, liefert RKP() nur die m -Koeffizienten und die Konstante b

Detaillierte Informationen zu den Regressionskenngrößen finden Sie unter RGP() auf Seite 492.

Hinweis

Hintergrund Im Gegensatz zur Funktion RGP() versucht die Funktion RKP() die abhängigen y -Werte und die unabhängigen x -Werte durch Berechnung einer Exponentialkurve nach folgender allgemeiner Formel zu beschreiben:

$$y = b \times m^x$$

wobei y und x Vektoren sein können, also

$$y = \left(b \times \left(m_1^{x_1} \right) \times \left(m_2^{x_2} \right) \times \left(m_n^{x_n} \right) \right)$$

Zu jeder Basis m gehört ein entsprechender Exponent x . Sie müssen also auch bei dieser Funktion darauf achten, dass die verwendeten Bezüge oder Werte die gleiche Anzahl an Elementen aufweisen.

Je mehr die grafische Darstellung Ihrer Daten einer Exponentialkurve gleicht, desto besser wird die berechnete Kurve an diese Daten angepasst sein.

Wie RGP() liefert RKP() eine Matrix, deren Elemente die Beziehung zwischen den Werten beschreiben, wobei RGP() Ihre Daten an eine gerade Linie anpasst und RKP() an eine Exponentialkurve.

Wenn Sie nur eine unabhängige x -Variable haben, können Sie die Werte für die Steigung m und den y -Schnittpunkt b mit den folgenden Formeln direkt ermitteln:

- ▶ Steigung m : `INDEX(RKP(Y_Werte;X_Werte);1)`
- ▶ y -Schnittpunkt b : `INDEX(RKP(Y_Werte;X_Werte);2)`

Mithilfe der Gleichung $y = b \cdot m^x$ können Sie zukünftige y -Werte abschätzen. Solche Abschätzungen können Sie aber auch mit der von Excel bereitgestellten Funktion VARIATION() (siehe Seite 549) vornehmen.

Wird eine Matrix-Konstante (wie zum Beispiel `X_Werte`) als Argument eingegeben, müssen Sie **Punkte** verwenden, um Werte derselben Zeile zu trennen, und ein **Semikolon**, um die Zeilen zu trennen.

Tipp Sie sollten daran denken, dass mithilfe einer Regressionsgleichung vorhergesagte y -Werte eventuell ungültig sind, wenn diese außerhalb des Bereichs der y -Werte liegen, mit denen Sie die Gleichung ermittelt haben.

Praxiseinsatz Um Ihnen die Berechnung der Regressionsgrößen über die Funktion RKP() näher zu erläutern, halten wir uns an das bereits unter RGP() auf Seite 492 genannte Beispiel.

Die Produktbestellungen auf der Website eines Softwareherstellers sind im vergangenen Jahr merklich gestiegen. Die Geschäftsführung möchte die Gründe hierfür erforschen. Hängt der Anstieg der Bestellungen mit dem aktiven Marketing und den straffen Werbemaßnahmen der letzten Zeit zusammen? Sind durch die vermehrten Websitezugriffe auch die Bestellungen gestiegen?

Es soll also auch hier, diesmal mithilfe der Funktion RKP(), aufgezeigt werden, ob ein Zusammenhang zwischen den Websitezugriffen und den Bestellungen besteht.

Die Marketingabteilung wird damit beauftragt, diese Analyse durchzuführen und erstellt eine Excel-Liste mit einer Gegenüberstellung der Websitezugriffe der letzten 18 Monate und den in diesem Zeitraum eingegangenen Bestellungen. Anschließend soll eine Regressionsanalyse mit der Funktion RKP() durchgeführt werden.

	B	C	D
9	Monat	Webseitenzugriff	Bestellungen
10	Januar 2007	236	485
11	Februar 2007	11593	5948
12	März 2007	18491	6000
13	April 2007	11743	7465
14	Mai 2007	11452	8364
15	Juni 2007	26651	9486
16	Juli 2007	16287	10837
17	August 2007	17750	11937
18	September 2007	19985	12847
19	Oktober 2007	17285	13857
20	November 2007	30369	14827
21	Dezember 2007	19674	15736
22	Januar 2008	28464	16837
23	Februar 2008	25000	17483
24	März 2008	24574	18938
25	April 2008	23141	19524
26	Mai 2008	17700	20492
27	Juni 2008	3702	21938

Abbildung 11.113: Die Gegenüberstellung der Webseitenzugriffe und Bestellungen

In einer Grafik stellen sich die Bestellungen in Abhängigkeit zu den Webseitenzugriffen folgendermaßen dar:

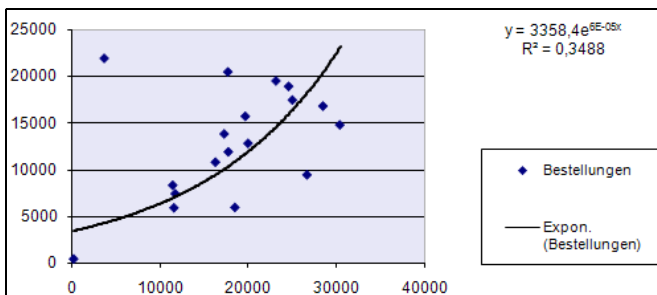


Abbildung 11.114: Die Bestellungen als exponentielle Trendlinie

Wie Sie der Abbildung 11.114 entnehmen können, weisen die Bestellungen in Abhängigkeit zu den Webseitenzugriffen ein exponentielles Wachstum auf. Das heißt, es kann angenommen werden, dass die Bestellungen exponentiell ansteigen, wenn die Anzahl der Webseitenzugriffe steigt.

In diesem Fall bietet sich zur Berechnung der Regressionsgrößen die Funktion RKP() an. Sie liefert folgende Ergebnisse:

	B	C	D	E
31	b	m ₁	1,00006366	3358,432538
32	se _b	se ₁	2,17467E-05	0,427166435
33	R ²	se _y	0,348768458	0,724405218
34	F	d _f	8,568834519	16
35	SS _{reg}	SS _{resid}	4,496606619	8,396206712

Abbildung 11.115: Die Ergebnisse für die exponentielle Regressionsanalyse

Achtung Formeln, die als Ergebnis eine Matrix liefern, müssen als Matrixformeln eingegeben werden. Das heißt, da Sie über RKP() alle zehn Regressionsgrößen auf einmal errechnen können und somit eine Matrix ausgegeben wird, muss die Eingabe der Formel bzw. der Argumente unter bestimmten Kriterien erfolgen.

Markieren Sie zunächst, wie in Abbildung 11.115 dargestellt, die Zellen D31:E35 und geben Sie anschließend in die Zelle D31 die Argumente für die Funktion RKP() ein. Beenden Sie die Eingabe der Formel durch Klicken auf die Taste $\boxed{F2}$ und anschließendes Drücken der Tastenkombination $\boxed{\text{Strg}} + \boxed{\uparrow} + \boxed{\leftarrow}$.

Siehe auch RGP(), TREND(), VARIATION(), SCHÄTZER()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Regression.xlsx* auf dem Arbeitsblatt *RKP*.

SCHÄTZER() FORECAST()

Syntax SCHÄTZER(*x*; *Y_Werte*; *X_Werte*)

Definition Die Funktion SCHÄTZER() gibt den Schätzwert für einen linearen Trend zurück. Der Vorhersagewert ist ein *y*-Wert bei einem gegebenen *x*-Wert. Bei den bekannten Werten handelt es sich um vorhandene *x*- und *y*-Werte, und der neue Wert wird, ausgehend von einer linearen Regression, vorhergesagt. Diese Funktion ermöglicht Ihnen, beispielsweise zukünftige Umsätze, erforderliche Lagerbestände oder Verbrauchertrends vorherzusagen.

Argumente *x* (erforderlich) ist der Datenpunkt, dessen Wert Sie schätzen möchten.

Y_Werte (erforderlich) ist eine abhängige Matrix oder ein abhängiger Datenbereich.

X_Werte (erforderlich) ist eine unabhängige Matrix oder ein unabhängiger Datenbereich.

Hinweis Ist *x* kein numerischer Ausdruck, gibt SCHÄTZER() den Fehlerwert #WERT! zurück.

Sind *Y_Werte* und *X_Werte* leer oder umfassen sie unterschiedlich viele Datenpunkte, gibt SCHÄTZER() den Fehlerwert #NV zurück.

Ist die Varianz von *X_Werte* gleich 0, gibt SCHÄTZER() den Fehlerwert #DIV/0! zurück.

Die Gleichung für SCHÄTZER() lautet: $a+bx$, mit:

$$a = \bar{a} - \bar{b}\bar{x}$$

und

$$b = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2}$$

Dabei sind \bar{x} und \bar{y} die Stichprobenmittelwerte MITTELWERT(*X_Werte*) und MITTELWERT(*Y_Werte*).

Hintergrund Für ausführliche Trendanalysen stellt Excel die Funktionen TREND() und VARIATION() bereit. Diese Funktionen sind in der Bedienung aber recht erklärungsbedürftig und verlangen eine Matrix zur Ausgabe der ermittelten Werte.

Hinweis Mehr zu den Funktionen TREND() und VARIATION() finden Sie auf Seite 528 bzw. auf Seite 549.

Möchten Sie dagegen auf die Schnelle aus einer Zahlenreihe oder einem Bereich eine Prognose für einen bestimmten Wert ermitteln, geht das bequemer mit der Funktion SCHÄTZER().

SCHÄTZER() gibt den Schätzwert für einen linearen Trend zurück. Das heißt, die Funktion versucht anhand von bestehenden Daten einen Trend zu ermitteln und berechnet für den angegebenen x -Wert die entsprechende »Zukunftszahl«. In unserem Beispiel bedeutet dies, dass für den Monat Juli 2008 (x -Wert) die geschätzten Webseitenzugriffe ausgegeben werden.

Als Marketingleiter des Softwareunternehmens haben Sie die Webseitenzugriffe und Onlinebestellungen nun schon ausgiebig analysiert. Da Sie zukunftsorientiert arbeiten, möchten Sie eine Tabelle mit Voraussagen erstellen.

Praxiseinsatz

Vorgegeben sind die kommenden neun Monate – von Juli 2008 bis März 2009 – und gesucht ist sowohl die Anzahl der Webseitenzugriffe als auch die Anzahl der Onlinebestellungen.

Als Modell nehmen Sie an, dass sich die Daten von Januar 2006 bis Juni 2008 annähernd linear verhalten, d.h., Sie haben bereits eine Gerade berechnet, mit der Sie auch Voraussagen über die Zukunft machen könnten (siehe Abbildung 11.116).

Da Sie aber noch etwas mehr von Excel kennen lernen wollen, verwenden Sie daher die Funktion SCHÄTZER().

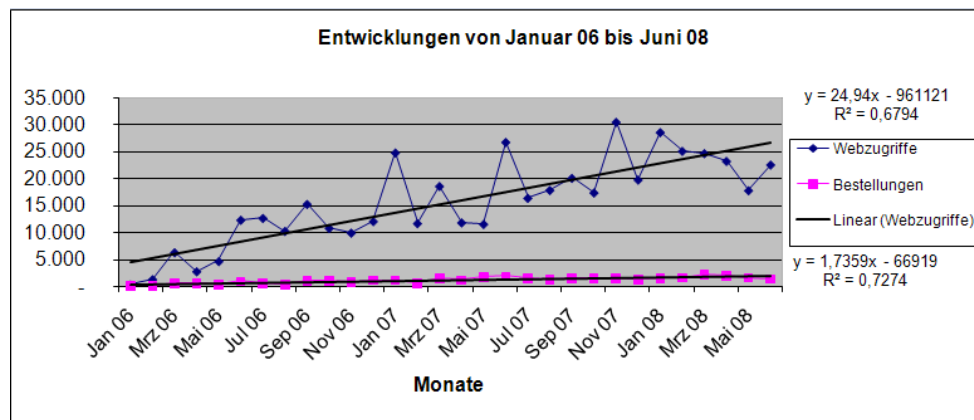


Abbildung 11.116: Die grafische Darstellung der linearen Regressionsgeraden der Bereiche *Webseitenzugriffe* und *Bestellungen*

Zunächst möchten Sie die erwartete Anzahl von Webseitenzugriffen für den Monat Juli 2008 berechnen. Als Argumente für die Funktion SCHÄTZER() verwenden Sie folgende Werte (vergleichen Sie dazu auch die Abbildung 11.117):

- ▶ x = Zelle B40 (Juli 2008) – für diese Zelle suchen Sie die voraussichtliche Anzahl an Webseitenzugriffen
- ▶ Y_Werte = Zellen C10:C39 – die Anzahl der Webseitenzugriffe von Januar 2006 bis Juni 2008 stellen die abhängigen Y_Werte dar
- ▶ X_Werte = Zellen B10:B39 – die Monate Januar 2006 bis Juni 2008 stellen die unabhängigen X_Werte dar

Die Eingabe der Argumente für den gesuchten Trend in Zelle C40 stellt sich entsprechend Abbildung 11.117 dar.

Hinweis Achten Sie bei der Eingabe der Zellbereiche in unserem Beispiel darauf, dass Sie für die Argumente *Y_Werte* und *X_Werte* absolute Pfade angeben. Drücken Sie deshalb die Taste **F4**, nachdem Sie die jeweiligen Zellen markiert haben. Wie in Abbildung 11.117 dargestellt, erscheint daraufhin das **\$**-Zeichen um die Zellangabe. Durch die absolute Adressierung (und gleichzeitige Fixierung) können Sie anschließend die Formel in die noch zu berechnenden Zellen C41:C48 wie gewohnt kopieren.

Die Zelle mit dem gesuchten Argument *x*, also die geschätzte Anzahl an Webseitenzugriffen, wird während des Kopiervorgangs automatisch an die jeweilige Zeile angepasst, da sie relativ adressiert ist.

C40		fx =SCHÄTZER(B40;\$C\$10:\$C\$39;\$B\$10:\$B\$39)	
	B	C	D
g	Monat	Webzugriffe	Bestellungen
10	Januar 2006	236	6
11	Februar 2006	1.221	17
12	März 2006	6.195	456
13	April 2006	2.682	544
14	Mai 2006	4.569	349
15	Juni 2006	12.229	854
16	Juli 2006	12.564	427
17	August 2006	10.157	337
18	September 2006	15.160	899
19	Oktober 2006	10.716	1.011
20	November 2006	9.844	720
21	Dezember 2006	11.975	1.069
22	Januar 2007	24.695	1.070
23	Februar 2007	11.593	498
24	März 2007	18.491	1.401
25	April 2007	11.743	1.076
26	Mai 2007	11.452	1.563
27	Juni 2007	26.651	1.790
28	Juli 2007	16.287	1.367
29	August 2007	17.750	1.138
30	September 2007	19.985	1.352
31	Oktober 2007	17.285	1.343
32	November 2007	30.369	1.430
33	Dezember 2007	19.674	1.140
34	Januar 2008	28.464	1.421
35	Februar 2008	25.000	1.508
36	März 2008	24.574	2.137
37	April 2008	23.141	1.948
38	Mai 2008	17.700	1.521
39	Juni 2008	22.456	1.304
40	Juli 2008	27.258	
41	August 2008	28.031	
42	September 2008	28.804	
43	Oktober 2008	29.552	
44	November 2008	30.326	
45	Dezember 2008	31.074	
46	Januar 2009	31.847	
47	Februar 2009	32.620	
48	März 2009	33.318	

Abbildung 11.117: Die Funktion *SCHÄTZER()* berechnet die Webseitenzugriffe für den Monat Juli 2008

Wie in Abbildung 11.117 dargestellt, erhalten Sie durch einfaches Kopieren der Formel aus Zelle C40 in die noch zu berechnenden Felder C41:C48 den Trend im Bereich der Webseitenzugriffe für die Monate Juli 2008 bis März 2009.

Auf gleiche Weise können Sie mit der Funktion SCHÄTZER() auch eine Prognose für die Onlinebestellungen in Abhängigkeit der soeben errechneten Webseitenzugriffe generieren. Im Unterschied zur Berechnung der geschätzten Webseitenzugriffe müssen Sie jedoch für die Zukunftswerte der Onlinebestellungen folgende Argumente angeben:

- ▶ x = Zelle C40 (Anzahl der Webseitenzugriffe im Juli 2008) – für diese Zelle suchen Sie die voraussichtliche Anzahl an Onlinebestellungen
- ▶ Y_Werte = Zellen D10:D39 – die Anzahl der Onlinebestellungen von Januar 2005 bis Juni 2008 stellen die abhängigen Y_Werte dar
- ▶ X_Werte = Zellen C10:C39 – die Webseitenzugriffe von Januar 2005 bis Juni 2008 stellen die unabhängigen X_Werte dar

Das Ergebnis und die Eingabe der Formel für die Funktion SCHÄTZER() zur Berechnung der voraussichtlichen Onlinebestellungen sehen Sie in Abbildung 11.118.

D40		f _x =SCHÄTZER(C40;SD\$10:SD\$39;SC\$10:SC\$39)	
9	B	C	D
	Monat	Webzugriffe	Bestellungen
40	Juli 2008	27.258	1.706
41	August 2008	28.031	1.748
42	September 2008	28.804	1.791
43	Oktober 2008	29.552	1.832
44	November 2008	30.326	1.875
45	Dezember 2008	31.074	1.916
46	Januar 2009	31.847	1.959
47	Februar 2009	32.620	2.002
48	März 2009	33.318	2.040

Abbildung 11.118: Die Prognosewerte für die Onlinebestellungen, ermittelt durch die Funktion SCHÄTZER()

Geben Sie, wie zuvor im Hinweis erwähnt, für die Argumente Y_Werte und X_Werte absolute Pfade an, können Sie die eingegebene Funktion in Zelle D40 in die Zellen D41:D48 kopieren und erhalten so die Trendwerte für die restlichen Monate – in Abhängigkeit von den Webseitenzugriffen.

RGP(), RKP(), TREND(), VARIATION()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Regression.xlsx* auf dem Arbeitsblatt *Schätzer*.

Siehe auch



SCHIEFE()



SCHIEFE(Zahl1;Zahl2; ...)

Die Funktion SCHIEFE() gibt die Schiefe einer Verteilung zurück. Die Schiefe ist ein Maß für die Asymmetrie einer eingipfligen Häufigkeitsverteilung um ihren Mittelwert. Eine positive Schiefe zeigt eine Verteilung an, deren Gipfel sich tendenziell zu Werten größer dem Mittelwert hin orientiert. Diese Verteilung wird auch linksschiefe Verteilung genannt. Eine negative Schiefe zeigt eine rechtsschiefe Verteilung an, deren Gipfel sich tendenziell zu Werten kleiner dem Mittelwert hin orientiert.

Syntax

Definition

Argumente *Zahl1* (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 Argumente (30 bis Excel 2003), deren Schiefe Sie berechnen möchten. Anstelle der durch ein Semikolon voneinander getrennten Argumente können Sie eine Matrix oder einen Bezug auf eine Matrix angeben.

Hinweis Als Argumente sollten entweder Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Wenn weniger als drei Datenpunkte vorhanden sind oder die Standardabweichung der Stichprobe gleich 0 ist, gibt SCHIEFE() den Fehlerwert #DIV/0! zurück.

Hintergrund SCHIEFE() liefert Ihnen die Schiefe oder, besser ausgedrückt, die Asymmetrie einer eingipfligen Häufigkeitsverteilung zum Mittelwert. Die Schiefe wie auch die Kurtosis charakterisieren zusammen also die Form und den Grad der Symmetrie einer Verteilung

Hinweis Mehr zum Thema *Kurtosis* finden Sie auf Seite 433 dieses Buchs.

Die Schiefe hängt stark von extremen Werten (Ausreißern) ab.

Da die Gaußsche Normalverteilung die Schiefe Null hat und somit immer symmetrisch zum Mittelwert ist, ist die Schiefe ein geeignetes Werkzeug, um eine beliebige Verteilung mit der Normalverteilung zu vergleichen.

Die Normalverteilung hat folgende charakteristische Eigenschaften: Mittelwert, Median und Modus befinden sich in einem Punkt. Die Streuung ist demnach symmetrisch um das arithmetische Mittel herum. Beim ersten Wendepunkt ist *Mittelwert* minus *Standardabweichung*, beim dritten Wendepunkt ist *Mittelwert* plus *Standardabweichung*. Zwischen *Mittelwert* minus *Standardabweichung* und *Mittelwert* plus *Standardabweichung* befinden sich ca. 66 % der Werte.

Schiefe Verteilungen kommen in der Praxis jedoch öfter vor. Bei ihnen sind Mittelwert, Median und Modus nicht mehr im selben Punkt.

Die Formel zur Berechnung der Schiefe lautet wie folgt:

$$\frac{n}{(n-1)(n-2)} \sum \left(\frac{x_i - \bar{x}}{s} \right)^3$$

In der Statistik ist die Schiefe wie folgt definiert:

$$v = \frac{m^3(\mu)}{\sigma^3}$$

Bei dieser Gleichung steht $m^3(\mu)$ für das dritte zentrale Moment und σ für die Standardabweichung.

Ist $v > 0$, so ist die Verteilung rechtsschief, ist $v < 0$, ist die Verteilung linksschief.

Praxiseinsatz Um Ihnen die Funktionsweise der SCHIEFE() zu verdeutlichen, greifen wir auf das Webseitenbeispiel unseres Softwareherstellers zurück. Die Marketingabteilung möchte eine Auswertung des Download- und gesamten Webseitenbereichs machen und hierbei die Schiefe, also die asymmetrische Verteilung der Webseitenklicks um deren jeweiligen Mittelwert berechnen.

Als Ergebnis dieser Berechnung erhalten die Mitarbeiter einen Wert, mit dessen Hilfe die eingipfelige Häufigkeitsverteilung der Werte um den Mittelwert analysiert werden kann.

	B	C	D	E	F
	DATUM	ANZAHL KLICKS IM BEREICH DOWNLOAD	MITTELWERT	ANZAHL KLICKS AUF DER GESAMTEN WEBSEITE	MITTELWERT
10					
11	Jan 07	5000	1725	13987	15163
12	Feb 07	1000	1725	17645	15048
13	Mrz 07	1076	1725	13435	15048
14	Apr 07	1521	1725	14424	15048
15	Mai 07	1790	1725	15000	15048
16	Jun 07	3908	1725	17875	15048
17	Jul 07	1138	1725	16049	15048
18	Aug 07	1352	1725	16734	15048
19	Sep 07	1343	1725	15039	15048
20	Okt 07	1430	1725	23424	15048
21	Nov 07	1140	1725	8974	15048
22	Dez 07	1421	1725	15049	15048
23	Jan 08	1508	1725	15874	15048
24	Feb 08	2137	1725	15987	15048
25	Mrz 08	1948	1725	4980	15048
26	Apr 08	1521	1725	1600	15048
27	Mai 08	426	1725	23424	15048
28	Jun 08	1384	1725	23442	15048
29	Schiefe	2,23		-0,73	

Abbildung 11.119: Die Berechnung der Schiefe für den Downloadbereich und für den gesamten Webseitenbereich

Um eine Vorstellung der Häufigkeitsverteilung sowohl im Downloadbereich als auch für den gesamten Webseitenbereich zu erhalten, werden mit wenigen Mausklicks noch die dazugehörigen Diagramme erstellt. Diese stellen sich entsprechend der Abbildung 11.120 und der Abbildung 11.121 dar.

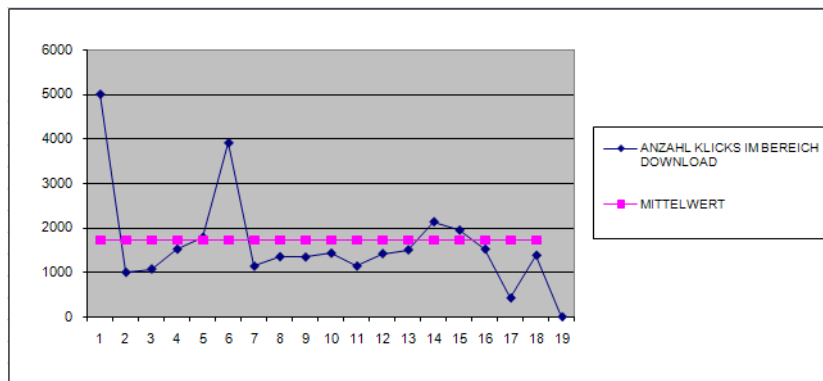


Abbildung 11.120: Die Verteilung der Klicks im Downloadbereich um deren Mittelwert

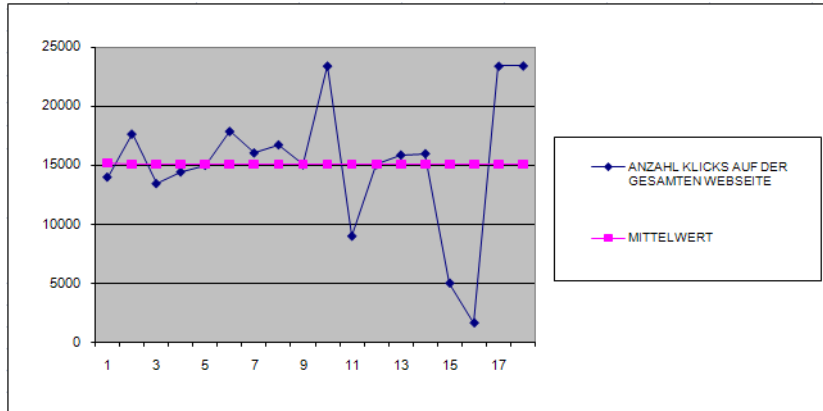


Abbildung 11.121: Die Verteilung der Klicks der gesamten Webseite um deren Mittelwert

Aussagen Was sagen die Ergebnisse der Berechnung der Schiefe aus?

1. Die positive Schiefe für den Downloadbereich mit dem Wert 2,23 bedeutet, dass es sich hierbei um eine Verteilung handelt, deren Gipfel sich tendenziell in Richtung der Werte orientiert, die größer dem Mittelwert sind (linksschiefe Verteilung).

Diese Verteilung nennt sich auch linksgipfelig, weil der höchste Punkt der Verteilung links liegt bzw. die Messdaten sich über einen längeren Bereich in die mathematisch als positiv definierte Richtung erstrecken (siehe Abbildung 11.120).

2. Die negative Schiefe für den gesamten Webseitenbereich mit dem Wert $-0,73$ bedeutet, dass es sich hierbei um eine Verteilung handelt, deren Gipfel sich tendenziell in Richtung der Werte orientiert, die kleiner dem Mittelwert sind (rechtsschiefe Verteilung).

Diese Verteilung nennt sich auch rechtsgipfelig, weil der höchste Punkt der Verteilung vorwiegend auf der mathematisch als negativ (rechts) definierten Richtung liegt (siehe Abbildung 11.121).

Siehe auch KURT(), SCHIEFE.P(), STABW.S(), STDABWN(), VAR.S(), VAR.P()



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Symmetrie.xlsx* auf dem Arbeitsblatt *Schiefe*.

Neu in Excel
2013

SCHIEFE.P()



SKEW.P()

Syntax SCHIEFE.P(*Zahl1*; *Zahl2*; ...)

Definition SCHIEFE.P() gibt die Schiefe einer Verteilung auf Basis einer Grundgesamtheit zurück. Die Darstellung des Asymmetriegrads einer Verteilung um ihren Mittelwert wird durch das Berechnungsergebnis von SCHIEFE.P() beschrieben.

Argumente *Zahl1* (erforderlich); *Zahl2* (optional); ... sind 1 bis 254 Argumente, deren Schiefe, bezogen auf die Grundgesamtheit, Sie berechnen möchten.

Als Argumente sollten entweder Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Hinweis

Wahrheitswerte und Zahlen in Textform, die Sie direkt in die Liste der Argumente eingeben, werden berücksichtigt.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

SCHIEFE.P() verwendet die Standardabweichung einer Grundgesamtheit, nicht einer Stichprobe.

Wenn Argumente ungültige Werte sind, gibt SCHIEFE.P() den Fehlerwert #ZAHL! zurück.

Werden für Argumente ungültige Datentypen verwendet, gibt SCHIEFE.P() den Fehlerwert #WERT! zurück.

Sind weniger als drei Datenpunkte vorhanden oder ist die Standardabweichung der Stichprobe gleich 0, gibt SCHIEFE.P() den Fehlerwert #DIV/0! zurück.

Die Funktion SCHIEFE.P() ist eng verwandt mit der Funktion SCHIEFE() des vorangegangenen Abschnitts.

Hintergrund

Wichtige Hintergrundinformationen zum Thema Schiefe finden Sie im Abschnitt SCHIEFE() auf Seite 503.

Hinweis

SCHIEFE.P() verwendet folgende Gleichung:

$$v = \frac{1}{N} \sum_{i=1}^N \frac{x_i - \bar{x}^0}{\sigma}$$

Um Ihnen die Funktionsweise der SCHIEFE() zu verdeutlichen, greifen wir auf das Webseitenbeispiel unseres Softwareherstellers zurück. In diesem Fall fokussiert sich die Marketingabteilung auf die Auswertung des Downloadbereichs. Neben diversen Berechnungen soll die Schiefe.P, also die asymmetrische Verteilung der Klicks im Downloadbereich auf Basis der Grundgesamtheit errechnet werden.

Praxiseinsatz

Die Grundgesamtheit stellt alle Klicks im Aufzeichnungszeitraum von Januar 2007 bis heute – Januar 2013 – dar. Als Ergebnis dieser Berechnung erhält die Marketingabteilung einen Wert, mit dessen Hilfe der Asymmetriegrad um den Mittelwert analysiert werden kann.

Wie sind die Daten der Grundgesamtheit um den Mittelwert verteilt? Liegen sie tendenziell eher unterhalb oder oberhalb des Mittelwertes?			
DATUM (Jan07-Jan13)	ANZAHL KLICKS IM BEREICH DOWNLOAD	MITTELWERT	Berechnung Schiefe.P
Jan 07	5000	2245	0,398691281
Feb 07	1000	2245	
Mrz 07	1076	2245	
Apr 07	1521	2245	
Mai 07	1790	2245	
Jun 07	3908	2245	
Jul 07	1138	2245	
Aug 07	1352	2245	

Abbildung 11.122: Die Berechnung der Schiefe.P für den Downloadbereich – basierend auf der Grundgesamtheit

Um eine Vorstellung über die Verteilung im Downloadbereich zu erhalten, wird mit wenigen Mausklicks noch das dazugehörige Diagramm erstellt.

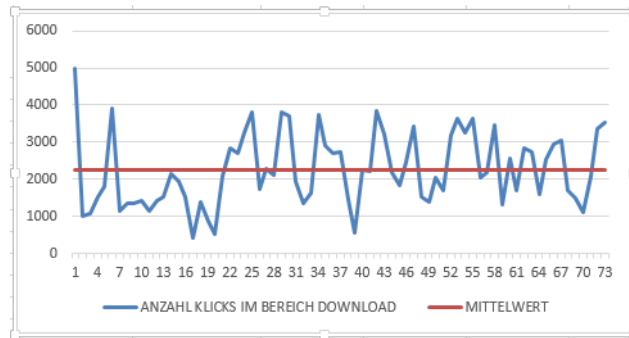


Abbildung 11.123: Die Verteilung der Klicks im Downloadbereich, basierend auf einer Grundgesamtheit, um deren Mittelwert

Aussagen Was sagt das Ergebnis der Berechnung der Schiefe.P aus?

- Die positive Schiefe für die Downloadwerte basierend auf einer Grundgesamtheit mit dem Wert 0,398 bedeutet, dass es sich hierbei um eine Verteilung handelt, deren Gipfel sich tendenziell in Richtung der Werte orientiert, die größer dem Mittelwert sind (links-schiefe Verteilung)

Siehe auch SCHIEFE(), KURT(), STABW.S(), STABWN(), VAR.S(), VAR.P()



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Symmetrie.xlsx* auf dem Arbeitsblatt *Schiefe.P*.

STABW.S() / STABW()



STDEV.S() / STDEV()

Syntax STABW.S(*Zahl1*; *Zahl2*; ...)

Definition Die Funktion STABW.S() schätzt die Standardabweichung ausgehend von einer Stichprobe. Die Standardabweichung ist ein Maß dafür, wie weit die jeweiligen Werte um den Mittelwert (Durchschnitt) streuen.

Argumente *Zahl1* (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 numerische Argumente (30 bis Excel 2003), die einer Stichprobe aus der Grundgesamtheit entsprechen. Anstelle der durch Semikolon voneinander getrennten Argumente können Sie eine Matrix oder einen Bezug auf eine Matrix angeben.

Hinweis STABW.S() geht davon aus, dass die ihr übergebenen Argumente eine Stichprobe darstellen, die aus einer Grundgesamtheit gezogen wurden. Entsprechen die als Argumente übergebenen Daten hingegen einer Grundgesamtheit, sollte die zugehörige Standardabweichung mithilfe der Funktion STABW.N() berechnet werden.

Die berechnete Standardabweichung ist eine erwartungsgetreue Schätzung der Standardabweichung der Stichprobe aus einer Grundgesamtheit. Das bedeutet, es wird durch $n-1$ anstatt durch n geteilt.

Logische Werte wie *WAHR* und *FALSCH* sowie Text werden nicht berücksichtigt. Möchten Sie logische Werte und Text berücksichtigen, verwenden Sie die Funktion STABWA().

Die Standardabweichung ist die Quadratwurzel aus der Varianz. Gegenüber dieser hat die Standardabweichung den Vorteil, dass sie die Streuung in der Maßeinheit des Merkmals wiedergibt.

Hintergrund

Mehr zur Varianz und zu den Streumaßen können Sie unter VAR.S() auf Seite 542 nachlesen.

Hinweis

Beide Maße kommen aus der schließenden Statistik und zwar aus der Gleichung für die Normalverteilung, die dort eine wichtige Rolle spielt.

Sie haben sich allerdings auch in der deskriptiven Statistik etabliert und gegenüber der linearen Abweichung (= arithmetisches Mittel der Abweichung der Reihenwerte vom Mittelwert ohne Berücksichtigung der Vorzeichen), weil sie als zuverlässiger gelten.

In ihrer Höhe hängen Varianz und Standardabweichung von der Maßeinheit ab, in der die Merkmalsausprägung vorliegt. Dieser Faktor muss bei Vergleichen von verschiedenen Verteilungen berücksichtigt werden, und zwar insbesondere dann, wenn Variablen verglichen werden, die sich zusätzlich auf verschiedene Eigenschaftsdimensionen beziehen.

In solchen Fällen wird auf den Variationskoeffizienten zurückgegriffen. Er ergibt sich aus der Division der Standardabweichung durch das arithmetische Mittel und ist ein relatives Streumaß. Je größer der Variationskoeffizient, desto größer ist, relativ zu anderen Verteilungen, die Streuung. Voraussetzung für die Berechnung des Variationskoeffizienten ist das Verhältnisskalenniveau.

Die Standardabweichung gehört zu den wichtigsten Streumaßen. Auch sie beschreibt, wie die Varianz, die Abweichung der Werte vom Mittelwert. Allerdings wird für die Berechnung nicht wie bei der Varianz die Differenz der Werte vom Mittelwert verwendet, sondern das Quadrat der Differenz.

Anders ausgedrückt: Die Standardabweichung ist die Quadratwurzel aus dem arithmetischen Mittel der quadrierten Abweichung der Werte vom arithmetischen Mittel. Sie entspricht der Wurzel aus der Varianz.

STABW() verwendet die folgende Formel:

$$\sqrt{\frac{\sum (x - \bar{x})^2}{(n-1)}}$$

Dabei ist x der Stichprobenmittelwert MITTELWERT(Zahl1;Zahl2;...) und n der Stichprobenumfang.

Die Marketingabteilung des Softwareherstellers hat bereits viel über Statistik und die damit verbundenen möglichen Auswertungen gelernt. Nun möchten sich die Mitarbeiter der Abteilung der Funktion STABW.S() widmen, um noch detailliertere Auswertungen bezüglich der Webseite und der Zugriffe darauf zu generieren.

Praxiseinsatz

Ziel soll es sein, die Standardabweichung ausgehend von einer Stichprobe zu errechnen, um anschließend beurteilen zu können, inwieweit die Anzahl der Zugriffe (Klicks) um den Mittelwert streuen.

Hinweis Beachten Sie, dass es sich bei den zu untersuchenden Daten um eine Stichprobe handelt. Das heißt, die Webseite und somit auch die Zugriffe darauf gibt es bereits seit längerer Zeit. Für die Berechnung der Standardabweichung mit den Funktionen STABW.S() und STABW.A() werden jedoch lediglich 18 Monate, von Januar 07 bis Juni 08, aus dem gesamten Zeitraum betrachtet.

Die Mitarbeiter haben bereits eine Excel-Datei mit der Anzahl der Zugriffe pro Monat auf die verschiedenen Webseitenbereiche erstellt. Neben der Standardabweichung wurden, zur besseren Übersicht für weiterführende Vergleiche, noch die Varianz, die mittlere Abweichung sowie der Mittelwert für die einzelnen Datenbereiche errechnet.

	B	C	D	E	F	G	H	I
12	DATUM	ADVENT	DOWNLOAD	EVENT	TRAINING	WISSEN	PRODUKTE	GESAMT
13	Jan 07		6		19	43	22	90
14	Feb 07		498	38	1319	3000	1170	6025
15	Mrz 07		1401	119	2233	6116	1545	11414
16	Apr 07		1076	16	1903	2860	1168	7023
17	Mai 07		1563	15	1589	3126	1139	7432
18	Jun 07		1790	1853	2428	6682	5083	17836
19	Jul 07		1367	1622	1559	3311	3609	11468
20	Aug 07		1138	2170	1502	3317	4810	12937
21	Sep 07		1352	2611	1628	3542	5581	14714
22	Okt 07	1	1343	538	1440	2971	5506	11799
23	Nov 07	272	1430	1254	2255	4735	11786	21732
24	Dez 07	1874	1140	631	1583	2652	6227	14107
25	Jan 08	42	1421	1515	2224	4837	9500	19539
26	Feb 08	32	1508	1433	1418	4006	12000	20397
27	Mrz 08	10	2137	3114	1298	3128	8000	17687
28	Apr 08	10	1948	2347	1085	3575	8208	17173
29	Mai 08	9	1521	848	765	2139	7739	13021
30	Jun 08		426	440	123	768	778	2535
31	Summe	2250	23065	20564	26371	60808	93871	226929
32	Mittelwert	281,25	1281,39	1209,65	1465,06	3378,22	5215,06	12607,17
33	Mittlere Abweichung	398,2	378,3	827,3	472,6	1093,9	3067,9	4785,7
34	Varianz	422303	279408	966160	444330	2561816	14361231	37396918
35	Standardabweichung	649,85	528,59	982,93	666,58	1600,57	3789,62	6115,30

Abbildung 11.124: Quadrieren Sie das Ergebnis der Standardabweichung, erhalten Sie die Varianz

Aussagen Betrachten wir den Bereich *PRODUKTE* näher, kann folgende Aussage getroffen werden:

Das Ergebnis von 3.789,62 bedeutet, dass die Zugriffe auf die Webseite (Klicks) in Höhe von 3.789,62 Klicks um den Mittelwert, der in diesem Fall 5.215,06 beträgt, streuen. Das heißt, die Klicks sind im Schnitt um 3.789,62 Klicks höher oder niedriger als der Mittelwert.

Quadrieren Sie den unter Standardabweichung errechneten Wert, erhalten Sie die Varianz, wie in Abbildung 11.124 dargestellt.

Für den Bereich *GESAMT*, also die Berechnung der Standardabweichung für **alle** Bereiche in **allen** Monaten des betrachteten Stichprobenzeitraumes lässt sich feststellen, dass die Zugriffe auf die Webseite in Höhe von 6.115,3 Klicks vom Mittelwert abweichen.

Siehe auch STABW.A(), STABWN(), STABWNA(), DBSTABW(), DBSTDABWN()



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Varianz.xlsx* auf dem Arbeitsblatt *Stabw.s*.

STABWA()



STABWA(*Wert1*; *Wert2*; ...)

Die Funktion STABWA() schätzt die Standardabweichung ausgehend von einer Stichprobe. Die Standardabweichung ist ein Maß für die Streuung von Werten im Bezug auf deren Mittelwert (dem Durchschnitt). Ausschlaggebend für die Funktion STABWA() ist, dass Text und Wahrheitswerte wie *WAHR* und *FALSCH* bei der Berechnung berücksichtigt werden.

Wert1 (erforderlich); *Wert2* (optional); ... sind 1 bis 255 Werte (30 bis Excel 2003), die einer Stichprobe aus der Grundgesamtheit entsprechen. Anstelle der durch Semikola voneinander getrennten Argumente können Sie eine Matrix oder einen Bezug auf eine Matrix angeben.

STABWA() geht davon aus, dass die ihr übergebenen Argumente eine Stichprobe aus einer Grundgesamtheit sind. Für den Fall, dass die zugehörigen Daten eine Grundgesamtheit angeben, verwenden Sie zur Berechnung der Standardabweichung die Funktion STABWNA().

Im Gegensatz zur Funktion STABW.S() werden bei STABWA() Text und Wahrheitswerte berücksichtigt. Bei der Berechnung werden Argumente, die den Wert *WAHR* enthalten, zu 1 und Argumente, die Text oder den Wahrheitswert *FALSCH* enthalten, zu 0 evaluiert. Soll die Berechnung weder Text noch Wahrheitswerte berücksichtigen, verwenden Sie demnach die Funktion STABW.S().

Wie bei STABW.S() ist die berechnete Standardabweichung eine erwartungsgereue Schätzung der Standardabweichung der Stichprobe aus einer Grundgesamtheit. Das bedeutet, auch hier wird durch $n-1$ anstatt durch n geteilt.

Da sich die Funktion STABWA() zur Funktion STABW.S() lediglich darin unterscheidet, dass sie Wahrheitswerte sowie Text berücksichtigt, richten wir unser Augenmerk hier vor allem auf die Darstellung der Funktion STABWA() im Beispiel.

Detaillierte Informationen zum Thema Standardabweichung können Sie auf Seite 506 unter STABW.S() nachlesen.

STABWA() verwendet die gleiche Formel wie STABW.S().

$$\sqrt{\frac{\sum (x - \bar{x})^2}{(n-1)}}$$

Dabei ist x der Stichprobenmittelwert MITTELWERT(*Wert1*; *Wert2*; ...) und n der Stichprobenumfang.

Betrachten wir wieder die Auswertung der Zugriffe auf die Webseite des Softwareherstellers. Da es im Betrachtungszeitraum, also in den vergangenen 18 Monaten, verschiedene Probleme mit und auf der Webseite gab, konnten nicht für alle Monate die Zugriffe gezählt werden.

Folgende Probleme lagen im Unternehmen vor:

- ▶ Im Monat März 07 und September 07 gab es Probleme mit dem Webseitenhosting, sodass in beiden Monaten nicht auf die Webseite zugegriffen werden konnte. Diese Monate wurden mit dem Text »Hostingprobleme« versehen.

Syntax

Definition

Argumente

Hinweis

Hintergrund

Praxiseinsatz

- ▶ Im Monat Februar 08 wurden die Inhaltsseiten des Trainingsbereichs umstrukturiert, sodass der gesamte Bereich geschlossen war. Da somit von Extern kein Zugriff auf den Trainingsbereich möglich war, wurde dieser Monat mit dem Wahrheitswert *FALSCH* gekennzeichnet.
- ▶ Im Monat Mai 08 wurden aus bisher noch unerklärter Ursache die Klicks auf die Webseitenbereiche nicht gezählt. Da jedoch ein Zugriff stattgefunden hat, wurde dieser Monat mit dem Wahrheitswert *WAHR* gekennzeichnet.

	B	C	D	E	F	G	H	I
12	DATUM	ADVENT	DOWNLOAD	EVENT	TRAINING	WISSEN	PRODUKTE	GESAMT
13	Jan 07		6		19	43	22	90
14	Feb 07		498	38	1319	3000	1170	6025
15	Mrz 07	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme
16	Apr 07		1076	16	1903	2860	1168	7023
17	Mai 07		1563	15	1589	3126	1139	7432
18	Jun 07		1790	1053	2428	6682	5083	17836
19	Jul 07		1367	1622	1559	3311	3609	11468
20	Aug 07		1138	2170	1502	3317	4810	12937
21	Sep 07	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme
22	Okt 07		1343	538	1440	2971	5506	11799
23	Nov 07		272	1430	1254	4735	11786	21732
24	Dez 07		1874	1140	631	1583	6227	14107
25	Jan 08		42	1421	1515	2224	9500	19539
26	Feb 08	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH
27	Mrz 08		10	2137	3114	1298	3128	8000
28	Apr 08		10	1948	2347	1085	3575	8208
29	Mai 08	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR
30	Jun 08			426	440	123	768	2535
31	Summe	2209	17283	15553	20327	45005	67006	167383
32	Mittelwert	368,17	1234,50	1196,38	1451,93	3214,64	4786,14	11955,93
33	Mittlere Abweichung	501,9	446,1	846,2	489,7	1024,2	2975,8	5331,4
34	Varianz (VAR.S())	554909	353161	1010192	496586	2594171	13504340	43314992
35	STABW.S()	744,92	594,27	1005,08	703,98	1610,64	3674,83	6581,41
36	STABWA()	586,84	740,85	1015,46	874,44	1968,42	3810,32	7699,47

Abbildung 11.125: Unter Berücksichtigung von Text und Wahrheitswerten liefert *STABWA()* ein anderes Ergebnis als *STABW.S()*

Wie Sie in Abbildung 11.125 sehen können, liefert die Funktion *STABWA()* ein anderes Ergebnis als die Berechnung der Standardabweichung über die Funktion *STABW.S()*. Das liegt an der Tatsache, dass *STABWA()*, wie bereits erwähnt, Text und Wahrheitswerte berücksichtigt. Text und der Wahrheitswert *FALSCH* wurden bei der Berechnung auf 0 und der Wahrheitswert *WAHR* auf 1 gesetzt.

Aussagen Betrachten wir den Bereich *PRODUKTE* näher, kann folgende Aussage getroffen werden:

Das Ergebnis von 3.810,32 bedeutet, dass die Zugriffe auf die Webseite (Klicks) in Höhe von 3.810,32 Klicks, unter Berücksichtigung von Text und Wahrheitswerten, um den Mittelwert streuen. Das heißt, die Klicks sind im Schnitt um 3.810,32 Klicks höher oder niedriger als der errechnete Mittelwert für diesen Bereich.

Siehe auch *STABW.S()*, *STABW.N()*, *STABWNA()*, *DBSTABW()*, *DBSTDABWN()*



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Varianz.xlsx* auf dem Arbeitsblatt *Stabwa*.

STABW.N() / STABWN()



STDEV.P() / STDEVQ()

STABW.N(*Zahl1*; *Zahl2*; ...)

Syntax

Die Funktion STABW.N() berechnet die Standardabweichung ausgehend von der Grundgesamtheit. Es wird vorausgesetzt, dass **alle** Werte als Argumente angegeben werden. Die Standardabweichung ist ein Maß dafür, wie weit die jeweiligen Werte um den Mittelwert (Durchschnitt) streuen.

Definition

Zahl1 (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 numerische Argumente (30 bis Excel 2003), die einer Grundgesamtheit entsprechen. Anstelle der durch Semikola voneinander getrennten Argumente können Sie eine Matrix oder einen Bezug auf eine Matrix angeben.

Argumente

Logische Werte wie *WAHR* und *FALSCH* sowie Text werden ignoriert. Möchten Sie logische Werte und Text berücksichtigen, verwenden Sie die Funktion STABW.N().

STABW.N() geht, im Gegensatz zu STABW.S() davon aus, dass die ihr übergebenen Argumente einer Grundgesamtheit entsprechen. Für den Fall, dass die zugehörigen Daten nur eine Stichprobe einer Grundgesamtheit angeben, berechnen Sie die Standardabweichung mit der Funktion STABW.S().

Hinweis

Für Stichproben mit vielen Werten geben die Funktionen STABW.S() und STABW.N() näherungsweise gleiche Werte zurück.

Für eine Stichprobe ist die berechnete Standardabweichung keine erwartungsgerechte Schätzung der Standardabweichung der Grundgesamtheit. Das bedeutet, es wird durch n anstatt durch $n-1$ geteilt.

Da sich die Funktion STABW.N() zur Funktion STABW.S() lediglich darin unterscheidet, dass sie die Standardabweichung nicht auf Basis einer Stichprobe, sondern auf Basis einer Grundgesamtheit berechnet, richten wir unser Augenmerk hier vor allem auf die Darstellung der Funktionen STABW.N() im Beispiel.

Hintergrund

Detaillierte Informationen zum Thema Standardabweichung können Sie auf Seite 509 unter STABW.S() nachlesen.

STABW.N() verwendet folgende Formel:

$$\sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

Dabei ist x der Stichprobenmittelwert MITTELWERT(*Zahl1*; *Zahl2*; ...) und n der Stichprobenumfang.

Kommen wir wieder zur Webseitenbewertung unseres Softwareherstellers und betrachten die Abbildung 11.126 etwas genauer.

Praxiseinsatz

Beachten Sie, dass es sich bei den zu untersuchenden Daten um die Grundgesamtheit handelt. Das heißt, die Webseite besteht erst seit 18 Monaten und somit wurden auch die Zugriffe darauf erst seit 18 Monaten registriert. Für die Berechnung der Standardabweichung mit den Funktionen STABW.N() und STABWNA() wird also die Grundgesamtheit der Daten von Januar 07 bis Juni 08 betrachtet.

Hinweis

	B	C	D	E	F	G	H	I
	DATUM	ADVENT	DOWNLOAD	EVENT	TRAINING	WISSEN	PRODUKTE	GESAMT
13	Jan 07		6		19	43	22	90
14	Feb 07		498	38	1319	3000	1170	6025
15	Mrz 07		1401	119	2233	6116	1545	11414
16	Apr 07		1076	16	1903	2860	1168	7023
17	Mai 07		1563	15	1589	3126	1139	7432
18	Jun 07		1790	1853	2428	6682	5083	17836
19	Jul 07		1367	1622	1559	3311	3609	11468
20	Aug 07		1138	2170	1502	3317	4810	12937
21	Sep 07		1352	2611	1628	3542	5581	14714
22	Okt 07	1	1343	538	1440	2971	5506	11799
23	Nov 07	272	1430	1254	2255	4735	11786	21732
24	Dez 07	1874	1140	631	1583	2652	6227	14107
25	Jan 08	42	1421	1515	2224	4837	9500	19539
26	Feb 08	32	1508	1433	1418	4006	12000	20397
27	Mrz 08	10	2137	3114	1298	3128	8000	17687
28	Apr 08	10	1948	2347	1085	3575	8208	17173
29	Mai 08	9	1521	848	765	2139	7739	13021
30	Jun 08		426	440	123	768	778	2535
31	Summe	2250	23065	20564	26371	60808	93871	226929
32	Mittelwert	281,25	1281,39	1209,65	1465,06	3378,22	5215,06	12607,17
33	Mittlere Abweichung	398,2	378,3	827,3	472,6	1093,9	3067,9	4785,7
34	VAR.P()	369515	263885	909327	419645	2419493	13563385	35319311
35	STABW.S()	649,85	528,59	982,93	666,58	1600,57	3789,62	6115,30
36	STABW.N()	607,88	513,70	953,59	647,80	1555,47	3682,85	5943,01

Abbildung 11.126: Ausgehend von einer Grundgesamtheit liefert die Funktion *STABW.N()* ein anderes Ergebnis als die Funktion *STABW.S()*, die von der Stichprobe einer Grundgesamtheit ausgeht

Deshalb liefern die Funktionen *STABW.N()* und *STABW.S()*, die unterschiedlichen Formeln unterliegen, auch unterschiedliche Ergebnisse.

Aussagen Betrachten wir den Bereich *PRODUKTE* näher, kann folgende Aussage getroffen werden:

Das Ergebnis von 3.682,85 bedeutet, dass die Zugriffe auf die Webseite (Klicks) in Höhe von 3.682,85 Klicks um den Mittelwert, ausgehend von einer Grundgesamtheit, streuen. Das heißt, die Klicks sind im Schnitt um 3.682,85 Klicks höher oder niedriger als der Mittelwert.

Quadrieren Sie den unter *STABW.N()* errechneten Wert, erhalten Sie das Ergebnis von *VAR.P()*, wie in *Abbildung 11.126* dargestellt.

Siehe auch *STABW.S()*, *STABWA()*, *STABWNA()*, *DBSTABW()*, *DBSTDABWN()*



Dieses Beispiel finden Sie im Ordner *\Ms5-235\Kap11* in der Arbeitsmappe *Varianz.xlsx* auf dem Arbeitsblatt *Stabw.n*.

STABWNA() STDEVPA()

Syntax *STABWNA(Wert1;Wert2;...)*

Definition Die Funktion *STABWNA()* berechnet die Standardabweichung ausgehend von einer als Argumente angegebenen Grundgesamtheit, einschließlich Text und logischer Werte. Die Standardabweichung ist ein Maß für die Streuung von Werten bezüglich ihres Mittelwerts (dem Durchschnitt).

Wert1 (erforderlich); *Wert2* (optional); ... sind 1 bis 255 Werte, die einer Grundgesamtheit entsprechen. Anstelle der durch Semikola voneinander getrennten Argumente können Sie eine Matrix oder einen Bezug auf eine Matrix angeben.

Argumente

STABWNA() geht davon aus, dass die ihr übergebenen Argumente einer Grundgesamtheit entsprechen. Für den Fall, dass die zugehörigen Daten nur eine Stichprobe einer Grundgesamtheit angeben, berechnen Sie die Standardabweichung mit der Funktion STABWA().

Hinweis

Entsprechend der Funktion STABWA() werden Argumente, die den Wert *WAHR* enthalten, als 1 und Argumente, die Text oder den Wahrheitswert *FALSCH* enthalten, als 0 berücksichtigt. Soll die Berechnung weder Text noch Wahrheitswerte berücksichtigen, verwenden Sie stattdessen die Funktion STABW.N().

Bei großem Stichprobenumfang geben STABWA() und STABWNA() das gleiche Ergebnis zurück.

Für eine Stichprobe ist die berechnete Standardabweichung keine erwartungsgetreue Schätzung der Standardabweichung der Grundgesamtheit. Das bedeutet, es wird durch n anstatt durch $n-1$ geteilt.

Da wir das Thema Standardabweichung bereits ausführlich auf Seite 509 unter STABW() behandelt haben, möchten wir in diesem Abschnitt unser Augenmerk hauptsächlich auf das Beispiel legen.

Hintergrund

STABWNA() verwendet, wie STABW.N(), folgende Formel:

$$\sqrt{\frac{\sum (x - \bar{x})^2}{n}}$$

Dabei ist x der Stichprobenmittelwert MITTELWERT(*Wert1*; *Wert2*; ...) und n der Stichprobenumfang.

Zur Darstellung der Funktion STABWNA() möchten wir auf das Beispiel aus STABWA() zurückgreifen. In diesem Beispiel hatte der Softwarehersteller folgende Probleme mit seiner Webseite:

Praxiseinsatz

- ▶ Im Monat März 07 und September 07 gab es Probleme mit dem Webseitenhosting, sodass in beiden Monaten nicht auf die Webseite zugegriffen werden konnte. Diese Monate wurden mit dem Text »Hostingprobleme« versehen.
- ▶ Im Monat Februar 08 wurden die Inhaltsseiten des Trainingsbereiches umstrukturiert, sodass der gesamte Bereich geschlossen war. Da somit von extern kein Zugriff auf den Trainingsbereich möglich war, wurde dieser Monat mit dem Wahrheitswert *FALSCH* gekennzeichnet.
- ▶ Im Monat Mai 08 wurden aus bisher noch unerklärter Ursache die Klicks auf die Webseitenbereiche nicht gezählt. Da jedoch ein Zugriff stattgefunden hat, wurde dieser Monat mit dem Wahrheitswert *WAHR* gekennzeichnet.

Beachten Sie, dass es sich, wie unter STABW.N(), bei den zu untersuchenden Daten um die Grundgesamtheit handelt. Das heißt, die Webseite besteht erst seit 18 Monaten und somit wurden auch die Zugriffe darauf erst seit 18 Monaten registriert.

Hinweis

DATUM	ADVENT	DOWNLOAD	EVENT	TRAINING	WISSEN	PRODUKTE	GESAMT
Jan 07		6		19	43	22	90
Feb 07		498	38	1319	3000	1170	6025
Mrz 07	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme
Apr 07		1076	16	1903	2860	1168	7023
Mai 07		1563	15	1589	3126	1139	7432
Jun 07		1790	1853	2428	6682	5083	17836
Jul 07		1367	1622	1559	3311	3609	11468
Aug 07		1138	2170	1502	3317	4810	12937
Sep 07	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme
Okt 07	1	1343	538	1440	2971	5606	11799
Nov 07	272	1430	1254	2255	4735	11786	21732
Dez 07	1874	1140	631	1583	2652	6227	14107
Jan 08	42	1421	1515	2224	4837	9500	19539
Feb 08	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH
Mrz 08	10	2137	3114	1298	3128	8000	17687
Apr 08	10	1948	2347	1085	3575	8208	17173
Mai 08	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR	WAHR
Jun 08		426	440	123	768	778	2535
Summe	2209	17283	15553	20327	45005	67006	167383
Mittelwert	368,17	1234,50	1196,38	1451,93	3214,64	4786,14	11955,93
Mittlere Abweichung	501,9	446,1	846,2	489,7	1024,2	2975,8	5331,4
VARIANZENA()	309942	518360	970510	722160	3659402	13711984	55988426
STABW.N()	680,02	572,66	965,65	678,37	1562,05	3541,15	6342,01
STABWNA()	556,72	719,97	985,14	849,80	1.912,96	3.702,97	7.482,54

Abbildung 11.127: Unter Berücksichtigung von Text und Wahrheitswerten liefert *STABWNA()* ein anderes Ergebnis als *STABW.N()*

Wie Sie in Abbildung 11.127 sehen können, liefert die Funktion *STABWNA()* ein anderes Ergebnis als die Berechnung der Standardabweichung über die Funktion *STABW.N()*. Das liegt an der Tatsache, dass *STABWNA()*, wie bereits erwähnt, Text und Wahrheitswerte berücksichtigt. In unserem Beispiel wurden Text und der Wahrheitswert *FALSCH* auf 0 und der Wahrheitswert *WAHR* auf 1 gesetzt.

Aussagen Betrachten wir den Bereich *PRODUKTE* näher, kann folgende Aussage getroffen werden:

Das Ergebnis von 3.702,97 bedeutet, dass die Zugriffe auf die Webseite (Klicks) in Höhe von 3.702,97 Klicks um den Mittelwert, ausgehend von einer Grundgesamtheit und unter Berücksichtigung von Text und Wahrheitswerten, streuen. Das heißt, die Klicks sind im Schnitt um 3.702,97 Klicks höher oder niedriger als der Mittelwert.

Siehe auch *STABW.S()*, *STABWA()*, *STABW.N()*, *DBSTABW()*, *DBSTDABWN()*



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Varianz.xlsx* auf dem Arbeitsblatt *Stabwna*.

STANDARDISIERUNG() STANDARDIZE()

Syntax *STANDARDISIERUNG(x;Mittelwert;Standabwn)*

Definition Die Funktion *STANDARDISIERUNG()* gibt den standardisierten Wert einer Verteilung zurück, die durch *Mittelwert* und *Standabwn* charakterisiert ist.

Argumente *x* (erforderlich) ist der Wert, den Sie standardisieren möchten.

Mittelwert (erforderlich) ist das arithmetische Mittel der Verteilung.

Standabwn (erforderlich) ist die Standardabweichung der Verteilung.

Hinweis Ist *Standabwn* kleiner oder gleich 0, gibt die Funktion *STANDARDISIERUNG()* den Fehlerwert *#ZAHL!* zurück.

In der Statistik versteht man unter einer Standardisierung die Transformation unterschiedlich skaliertes Zahlenwerte in einen einheitlichen Wertebereich von 0 bis 1, um beispielsweise Vergleiche unterschiedlich verteilter Werte vornehmen zu können.

Die Gauß- oder (Standard-) Normalverteilung (nach Carl Friedrich Gauß) ist ein wichtiger Typ kontinuierlicher Wahrscheinlichkeitsverteilungen. Die besondere Bedeutung der Normalverteilung beruht unter anderem auf dem zentralen Grenzwertsatz, der besagt, dass eine Summe von n unabhängigen, identisch verteilten Zufallsvariablen in der Grenze » n bis unendlich« normalverteilt ist.

Mehr Informationen zur (Standard-) Normalverteilung finden Sie auf Seite 468 unter der Funktion `NORM.VERT()`.

Die *standardisierte* Normalverteilung besitzt den Mittelwert 0 und die Standardabweichung 1.

Die Funktion `STANDARDISIERUNG()` berechnet den standardisierten Wert von x einer Normalverteilung mit bekanntem Mittelwert und Standardabweichung.

Ein standardisierter Wert wird gemäß folgender Gleichung berechnet:

$$Z = \frac{x - \mu}{\sigma}$$

Sie sind Hersteller von Glühbirnen, die Sie europaweit vertreiben, und sind dabei, die Leistung der Glühbirnen zu analysieren. Eine Aufstellung der Messungen haben Sie bereits in Excel gemacht.

	B	C	D
	Testglühbirnen	Gemessene Brennleistung in Stunden (=X)	Standardisierung z-Werte
10			
11	1	1000	-1,728
12	2	1100	-1,555
13	3	1200	-1,382
14	4	1300	-1,209
15	5	1400	-1,037
16	6	1500	-0,864
17	7	1600	-0,691
18	8	1700	-0,518
19	9	1800	-0,346
20	10	1900	-0,173
21	11	2000	0,000
22	12	2100	0,173
23	13	2200	0,346
24	14	2300	0,518
25	15	2400	0,691
26	16	2500	0,864
27	17	2600	1,037
28	18	2700	1,209
29	19	2800	1,382
30	20	2900	1,555
31	21	3000	1,728
32	22	2900	1,555
33	23	2800	1,382
34	24	2700	1,209
35	25	2600	1,037
36	26	2500	0,864
37	27	2400	0,691
38	28	2300	0,518
39	29	2200	0,346
40	30	2100	0,173
41	31	2000	0,000
42	32	1900	-0,173
43	33	1800	-0,346
44	34	1700	-0,518
45	35	1600	-0,691
46	36	1500	-0,864
47	37	1400	-1,037
48	38	1300	-1,209
49	39	1200	-1,382
50	40	1100	-1,555

Abbildung 11.128: Die Aufstellung der Messergebnisse

Hintergrund

Hinweis

Praxiseinsatz

Auch die mittlere Lebenserwartung sowie die dazugehörige Standardabweichung haben Sie bereits errechnet.

Mittelwert	Standardabweichung
2000	579

Abbildung 11.129: Mittelwert und Standardabweichung der Brennleistung

Um alle gemessenen Werte zu standardisieren, verwenden Sie die Funktion STANDARDISIERUNG().

Welche Argumente für die Funktion sind gegeben?

- ▶ x = Werte der Verteilung (gemessene Brennleistungen)
- ▶ *Mittelwert* = 2000 (F13)
- ▶ *Standabwn* = 579 (G13)

Die Ergebnisse können Sie beispielhaft Abbildung 11.130 entnehmen.

B		C	D	E	F	G
Testglühbirnen		Gemessene Brennleistung in Stunden (=X)	Standardisierung z-Werte		Berechnung	
11	1	1000	-1,728		Mittelwert	Standardabweichung
12	2	1100	-1,555		2000	579
13	3	1200	-1,382			
14	4	1300	-1,209			
15	5	1400	-1,037			
16	6	1500	-0,864			
17	7	1600	-0,691			
18	8	1700	-0,518			
19	9	1800	-0,346			
20	10	1900	-0,173			
21	11	2000	0,000			
22	12	2100	0,173			
23	13	2200	0,346			
24	14	2300	0,518			

Abbildung 11.130: Ermitteln Sie die standardisierten Werte mithilfe der Funktion STANDARDISIERUNG()

Mithilfe der Funktion STANDARDISIERUNG() haben Sie alle standardisierten Werte der Verteilung generiert.

Tipp Geben Sie für die Argumente *Mittelwert* und *Standabwn* absolute Zellbezüge an, müssen Sie die Formel nur einmal für einen beliebigen Wert der Verteilung angeben und können sie dann auf die anderen Werte kopieren.

Einen absoluten Zellbezug erhalten Sie durch Drücken der Taste **F4** nach Eingabe der Zelle. Vergleichen Sie hierzu die Bearbeitungsleiste für Funktionen in Abbildung 11.130.

Siehe auch GAUSS(), G.TEST(), NORM.INV(), NORM.VERT(), NORM.S.INV(), NORM.S.VERT()



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Standardisierung*.

STEIGUNG()



SLOPE()

STEIGUNG(*Y_Werte*; *X_Werte*)

Die Funktion STEIGUNG() gibt die Steigung der Regressionsgeraden zurück, die an die in *Y_Werte* und *X_Werte* abgelegten Datenpunkte angepasst ist. Die Steigung entspricht dem Quotienten aus dem jeweiligen vertikalen und dem horizontalen Abstand zweier beliebiger Punkte der Geraden und ist ein Maß für die Änderung entlang der Regressionsgeraden.

Y_Werte (erforderlich) ist eine Matrix oder ein Zellbereich numerisch abhängiger Datenpunkte.

X_Werte (erforderlich) ist eine Reihe unabhängiger Datenpunkte.

Als Argumente müssen entweder Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten. Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Enthalten *Y_Werte* und *X_Werte* keine oder unterschiedlich viele Datenpunkte, gibt STEIGUNG() den Fehlerwert #NV zurück.

Die Berechnung der Steigung mit der Funktion STEIGUNG() findet vor allem im Zusammenhang mit der Regressionsanalyse in Excel Anwendung.

Detaillierte Informationen zur Regressionsanalyse finden Sie unter RGP() auf Seite 492.

Eine lineare Funktion sieht in allgemeiner mathematischer Schreibweise so aus:

$$yt_i = a + bx_i$$

Den einzelnen Variablen kommt hierbei folgende Bedeutung zu:

- ▶ a = Ordinatenabschnitt der Funktion (Schnittpunkt mit der Y-Achse)
- ▶ b = Steigung der Funktion (Tangens des Steigungswinkels)
- ▶ x_i = Werte auf der x-Achse ($i=1,2,\dots,n$)
- ▶ yt_i = Werte auf der y-Achse ($i=1,2,\dots,n$; der Zusatz t steht für theoretisch)

Eine lineare Funktion stellt sich entsprechend der Abbildung 11.131 dar:

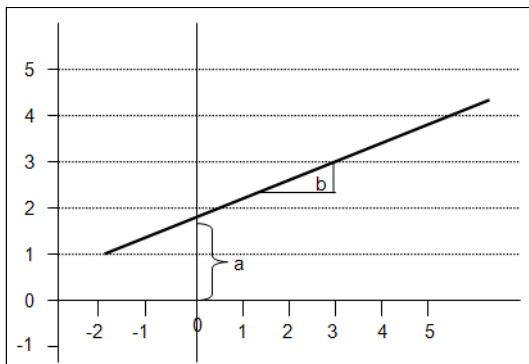


Abbildung 11.131: Die mathematische Darstellung einer linearen Funktion

Syntax

Definition

Argumente

Hinweis

Hintergrund

Hinweis

Wenn der Ordinatenabschnitt a und die Steigung b numerisch vorgegeben sind, ist eine lineare Funktion im Achsenkreuz eindeutig definiert.

Beispiel: $yt_i = 2 + 0,4 * x_i$

Diese Funktion hat den Ordinatenabschnitt $a = 2$ und die Steigung $b = 0,4$. 0,4 bedeutet in diesem Fall, dass die Funktion um 0,4 y -Einheiten steigt, wenn x um eine Einheit zunimmt.

Die Gleichung, nach der nun die Steigung einer Regressionsgeraden berechnet wird, lautet wie folgt:

$$b = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sqrt{\sum(x - \bar{x})^2}}$$

Hierbei sind x und y die Stichprobenmittelwerte MITTELWERT(X_Werte) und MITTELWERT(Y_Werte).

Praxiseinsatz Um die Steigung einer Regressionsgeraden oder einer Trendlinie zu ermitteln, bedienen Sie sich der Tabellenfunktion STEIGUNG().

Die Marketingabteilung unseres Softwareherstellers ist dabei, die firmeninterne Webseite bis ins kleinste Detail zu analysieren. Nachdem nun die Anzahl der Onlinebestellungen der vergangenen 18 Monate der Anzahl der Webseitenzugriffe gegenübergestellt wurden, wollen die Mitarbeiter die Steigung der linearen Trendlinie für die Bestellungen errechnen.

Ziel soll es sein, mit dem Ergebnis Aussagen darüber treffen zu können, wie sich die Regressionsgerade pro Einheit ändert.

	B	C	D
10	Monat	Webseitenzugriff	Bestellungen
11	Januar 2007	236	6
12	Februar 2007	11593	17
13	März 2007	18491	456
14	April 2007	11743	544
15	Mai 2007	11452	349
16	Juni 2007	26651	854
17	Juli 2007	16287	427
18	August 2007	17750	337
19	September 2007	19985	899
20	Oktober 2007	17285	1011
21	November 2007	30369	720
22	Dezember 2007	19674	1069
23	Januar 2008	28464	1070
24	Februar 2008	25000	498
25	März 2008	24574	1401
26	April 2008	23141	1076
27	Mai 2008	17700	1563
28	Juni 2008	3702	1790
29	Steigung		0,014360319

Abbildung 11.132: Die Berechnung der Steigung für die Regressionsgerade, die die Bestellungen in Abhängigkeit von den Webseitenzugriffen zeigt

Die Regressionsgerade in Form eines Diagramms sehen Sie in Abbildung 11.133.

Als Ergebnis der Berechnung der Steigung kann man festhalten, dass pro Webseitenbesuch (Klick) die Bestellungen um 0,0144 steigen. Allgemeiner ausgedrückt heißt das, dass die Funktion um 0,0144 y -Einheiten steigt, wenn x um eine Einheit zunimmt.

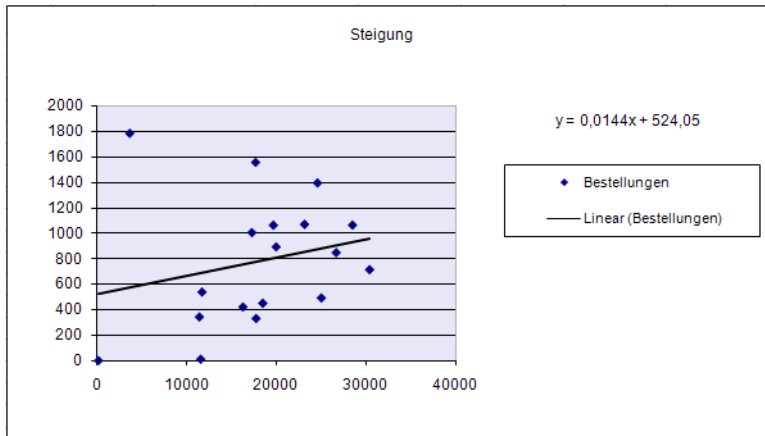


Abbildung 11.133: Die Regressionsgerade mit der Steigung 0,0144

ACHSENABSCHNITT(), BESTIMMTHEITSMASS(), PEARSON(), RGP(), RKP(), STFEHLERYX(), TREND()

Siehe auch

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Regression.xlsx* auf dem Arbeitsblatt *Steigung*.



STFEHLERYX() STEYX()

STFEHLERYX(*Y_Werte*; *X_Werte*)

Syntax

Die Funktion STFEHLERYX() gibt den Standardfehler der geschätzten *y*-Werte für alle *x*-Werte der Regression zurück. Der Standardfehler ist ein Maß dafür, wie groß der Fehler bei der Prognose (Vorhersage) des zu einem *x*-Wert gehörenden *y*-Werts ist.

Definition

Y_Werte (erforderlich) ist eine Matrix oder ein Bereich abhängiger Datenpunkte.

Argumente

X_Werte (erforderlich) ist eine Matrix oder ein Bereich unabhängiger Datenpunkte.

Die Argumente müssen entweder als Zahlen oder Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Hinweis

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Enthalten *Y_Werte* und *X_Werte* keine oder unterschiedlich viele Datenpunkte, gibt die Funktion STFEHLERYX() den Fehlerwert #NV zurück.

Der Standardfehler ist die Streuung von Stichprobenkennwerten um den wahren Wert des gesuchten Parameters in der Grundgesamtheit. Der Mittelwert könnte einen solchen Stichprobenkennwert darstellen. Je größer der Standardfehler, desto breiter ist das Konfidenzintervall, das den wahren Parameter mit vorgegebener Wahrscheinlichkeit enthält.

Hintergrund

Die Größe des Standardfehlers hängt von der Varianz der Messwerte in der Grundgesamtheit ab. Je geringer die Varianz, desto geringer auch der Standardfehler vom Umfang der Stich-

probe. Im Allgemeinen nimmt der Standardfehler proportional zur Quadratwurzel des Stichprobenumfangs ab. Dies bedeutet, dass man, um z.B. einen Standardfehler zu halbieren, den Stichprobenumfang vervierfachen muss.

Die Funktion STFEHLERYX() liefert diesen geschätzten Standardfehler einer linearen Regression und ist ein Maß für die Zuverlässigkeit einer linearen Regression. Je größer der Standardfehler ist, desto stärker weichen die geschätzten Werte von den tatsächlichen Werten – also den Werten aus der Grundgesamtheit – ab.

Für die Argumente Y_Werte und X_Werte geben Sie in der Regel einen Zellbereich an. Hierbei sind die Y_Werte abhängige, und die X_Werte unabhängige Variablen.

Beachten Sie, dass diese Funktion erst die Y_Werte und dann die X_Werte erwartet – nicht umgekehrt.

Ist die Grundgesamtheit mit μ und σ normalverteilt, so gilt für die Schätzfunktion aus Mittelwerten mehrerer Stichproben vom Umfang n :

- ▶ Sie ist normalverteilt,
- ▶ sie hat den Mittelwert μ ,
- ▶ die in Standardabweichungen gemessene Streuung der Stichprobenverteilung wird als Standardfehler des Mittelwerts bezeichnet.

Der Standardfehler definiert also die Standardabweichung aller Stichprobenmittelwerte vom vermuteten Mittelwert der Grundgesamtheit. Ist die Streuung verschiedener Stichprobenmittelwerte annähernd 0, ist auch der Standardfehler etwa 0. Ist die Streuung jedoch groß, wird der Standardfehler auch eher groß ausfallen. Mit größerem Stichprobenumfang verringert sich auch der Standardfehler.

Die Gleichung zur Berechnung des Standardfehlers eines prognostizierten y -Werts lautet:

$$\sqrt{\left(\frac{1}{n-2}\right) \left[\sum (y - \bar{y})^2 - \frac{[\sum (x - \bar{x})(y - \bar{y})]^2}{\sum (x - \bar{x})^2} \right]}$$

Dabei sind x und y die Stichprobenmittelwerte MITTELWERT(X_Werte) und MITTELWERT(Y_Werte) und ist n der Stichprobenumfang.

Praxeiseinsatz Wir möchten Ihnen die Funktion STFEHLERYX() anhand unseres Softwareunternehmens verdeutlichen. Es verkauft auf der firmeneigenen Website alle vom Unternehmen angebotenen Produkte.

Das Unternehmen besteht seit zehn Jahren. Obwohl die Webseite und auch die Möglichkeit der Onlinebestellungen bereits seit Anfang an zur Verfügung stehen, werden erst seit acht Jahren die Webseitenzugriffe zahlenmäßig festgehalten. Die Onlinebestellungen hingegen wurden seither noch nicht in einer separaten Datenbank erfasst.

Der Marketingleiter des Unternehmens hat die Zahlen der letzten 2,5 Jahre vorliegen und möchte diese nun näher analysieren. Dafür hat er die Anzahl der Webseitenzugriffe pro Monat sowie die geschätzten Onlinebestellungen pro Monat in Excel erfasst.

Die Onlinebestellungen sind von den Webseitenzugriffen abhängig. Das heißt, je mehr Webseitenzugriffe, desto mehr Onlinebestellungen. Auch den Mittelwert für beide Komponenten

hat der Marketingleiter bereits errechnet. Für den abhängigen Bereich Onlinebestellungen (y -Werte) ergibt sich ein Mittelwert von 1.121 Onlinebestellungen im Zeitraum von Juli 2007 bis Juni 2008. Vergleichen Sie dazu den Ausschnitt aus der Aufstellung in Abbildung 11.134.

	B	C	D
8	Monat	Webzugriffe	Bestellungen
26	Juni 2007	18.234	1.485
27	Juli 2007	18.769	1.367
28	August 2007	19.736	1.138
29	September 2007	20.333	1.352
30	Oktober 2007	20.987	1.343
31	November 2007	21.323	1.430
32	Dezember 2007	21.999	1.375
33	Januar 2008	22.786	1.421
34	Februar 2008	23.784	1.508
35	März 2008	24.574	1.876
36	April 2008	25.111	1.948
37	Mai 2008	25.789	2.094
38	Juni 2008	26.948	2.134
39	Mittelwert	15.427	1.121

Abbildung 11.134: Der berechnete Mittelwert für die Webseitenzugriffe und die Onlinebestellungen

Nun möchte der Marketingleiter wissen, wie »realitätsnah« der berechnete Mittelwert der Stichprobe für die prognostizierten Onlinebestellungen ist. Die Frage, die er sich stellt, lautet demnach: Wie hoch ist der Standardfehler der geschätzten Onlinebestellungen für alle Webzugriffe? Hierzu verwendet er die Funktion STFEHLERYX().

Das Ergebnis der Berechnung sehen Sie in Abbildung 11.135.

	B	C	D
7		Unabhängig = x-Wert	Abhängig = y-Wert
8	Monat	Webzugriffe	Bestellungen
9	Januar 2006	236	20
10	Februar 2006	1.221	387
11	März 2006	1.563	456
12	April 2006	2.682	544
13	Mai 2006	4.569	349
14	Juni 2006	6.848	854
15	Juli 2006	8.463	427
16	August 2006	10.157	337
17	September 2006	11.837	899
18	Oktober 2006	12.987	1.011
19	November 2006	13.739	720
20	Dezember 2006	14.376	1.069
21	Januar 2007	15.739	1.070
22	Februar 2007	16.123	967
23	März 2007	16.548	1.401
24	April 2007	17.352	1.076
25	Mai 2007	17.986	1.563
26	Juni 2007	18.234	1.485
27	Juli 2007	18.769	1.367
28	August 2007	19.736	1.138
29	September 2007	20.333	1.352
30	Oktober 2007	20.987	1.343
31	November 2007	21.323	1.430
32	Dezember 2007	21.999	1.375
33	Januar 2008	22.786	1.421
34	Februar 2008	23.784	1.508
35	März 2008	24.574	1.876
36	April 2008	25.111	1.948
37	Mai 2008	25.789	2.094
38	Juni 2008	26.948	2.134
39	Mittelwert	15.427	1.121
40	Standardfehler	210,07	

Abbildung 11.135: STFEHLERYX() liefert den Standardfehler für die den x -Werten zugeordneten y -Werten

Wie Sie in der Abbildung 11.135 sehen können, ergibt sich ein Standardfehler von 210,07.

Aussagen Welche Aussagen können auf Basis dieses Ergebnisses getroffen werden?

- ▶ Das Ergebnis von 210,07 sagt aus, dass der errechnete Stichprobenmittelwert der geschätzten Onlinebestellungen für alle Webzugriffe um 210 Bestellungen um den wahren Mittelwert – also dem Mittelwert aus der Grundgesamtheit der Onlinebestellungen – streut

Das Ergebnis von 210,07 ist demnach die Größe für den Fehler bei der Prognose der Onlinebestellungen in Abhängigkeit von den Webseitenzugriffen.

Siehe auch ACHSENABSCHNITT(), BESTIMMTHEITSMASS(), PEARSON(), RGP(), RKP(), STEIGUNG()



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Stfehleryx*.

SUMQUADABW() DEVSQ()

Syntax SUMQUADABW(*Zahl1*; *Zahl2*; ...)

Definition Die Funktion SUMQUADABW() gibt die Summe der quadrierten Abweichungen von Datenpunkten, berechnet auf Basis einer Stichprobe und bezogen auf deren Mittelwert, zurück.

Argumente *Zahl1* (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 Argumente (30 bis Excel 2003), deren quadratische Abweichungen Sie summieren möchten. Anstelle von durch Semikola getrennten Argumenten können Sie auch eine einzeilige oder einspaltige Matrix oder einen Bezug auf eine solche Matrix verwenden.

Hinweis Zulässige Argumente sind Zahlen, Namen von Matrizen oder auch Bezüge zu Zellen, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Hintergrund Generell kann man Zusammenhänge zwischen zwei Variablen durch Koeffizienten beschreiben. Mit der Regressionsrechnung können für intervallskalierte Variablen darüber hinaus Modelle geschätzt werden. Ein Modell ist der (mathematisch) formalisierte Kern einer Theorie. Modelle ermöglichen es, die abhängige Variable durch die unabhängige Variable vorherzusagen. Auf diese Weise können z.B. lineare Zusammenhänge aufgezeigt werden. Das heißt: Je mehr x , desto mehr (oder weniger) y . Zusätzlich kann die Beziehung x zu y durch eine einfache Gerade veranschaulicht werden. Ebenso sind natürlich auch kompliziertere Modelle denkbar.

Die Qualität der Regression wird in der Regel durch r^2 beschrieben.

Hinweis Mehr Informationen zu r^2 , also dem Bestimmtheitsmaß, finden Sie auf Seite 352 dieses Buchs.

Im Bezug auf die Qualität der Regression ist der beste y -Prognosewert für einen beliebigen Fall, ohne weitere Zusatzinformation, der Durchschnitt (quadrierte Abweichungen minimal). Die Abweichung vom Durchschnitt wird auch als Vorhersagefehler bezeichnet. Zur Berechnung der Abweichung vom Durchschnitt können Sie z.B. die Funktion VAR.S() verwenden (siehe Seite 542).

Die Funktion SUMQUADABW() ist eine spezielle Funktion für die Regressionsrechnung – sie berechnet die Summe der quadrierten Abweichungen von deren Stichprobenmittelwert und lässt so weitere Aussagen über die betrachteten Werte zu.

Die Gleichung für die Summe der quadratischen Abweichungen lautet:

$$SUMQUADABW = \sum (x - \bar{x})^2$$

Sie sind immer noch mit der Analyse Ihrer Webseite beschäftigt und wollen weitere Aussagen über den Zusammenhang zwischen den Webseitenzugriffen und den Onlinebestellungen treffen. Sie beschäftigen sich jetzt intensiv mit den Webseitenzugriffen – das heißt mit der unabhängigen Variablen y . Sie wollen wissen, wie hoch die Summe der quadrierten Abweichungen von deren Stichprobenmittelwert ist und bedienen sich der Funktion SUMQUADABW().

Praxiseinsatz

Die Abbildung 11.136 zeigt die Lösung:

B		C
8		Unabhängige Variable
9	Monat	Webseitenzugriff
10	Januar 2007	236
11	Februar 2007	11593
12	März 2007	18491
13	April 2007	11743
14	Mai 2007	11452
15	Juni 2007	26651
16	Juli 2007	16287
17	August 2007	17750
18	September 2007	19985
19	Oktober 2007	17285
20	November 2007	30369
21	Dezember 2007	19674
22	Januar 2008	28464
23	Februar 2008	25000
24	März 2008	24574
25	April 2008	23141
26	Mai 2008	17700
27	Juni 2008	3702
28	Sumquadabw	1109624270

Abbildung 11.136: Die Summe der quadrierten Abweichungen von deren Stichprobenmittelwert

Im Bereich der Webseitenzugriffe liefert die Funktion SUMQUADABW() den Wert 1.109.624.270. Dieser Wert entspricht der Summe der quadrierten Abweichungen von deren Stichprobenmittelwert.

MITTELABW(), STABW.S(), STDABWN(), VAR.S(), VAR.P()

Siehe auch

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Regression.xlsx* auf dem Arbeitsblatt *Sumquadabw*.



T.INV.2S() / TINV()**T.INV.2T() / TINV()**

Hinweis Die Funktion TINV() wird seit Microsoft Excel 2010 durch die Funktion T.INV.2S() ersetzt und zusätzlich durch die Funktion T.INV() ergänzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht. Um die Abwärtskompatibilität von T.INV.2S() zu sichern, ist die Funktionen TINV() weiter unter ihrem alten Namen verfügbar.

Syntax T.INV.2S(*Wahrsch*; *Freiheitsgrade*)

Definition Die Funktion T.INV.2S() gibt den t-Wert der t-Verteilung als eine Funktion der Wahrscheinlichkeit und der Freiheitsgrade zurück.

Argumente *Wahrsch* (erforderlich) ist die zur t-Verteilung gehörige Wahrscheinlichkeit (zweiseitig).
Freiheitsgrade (erforderlich) ist die Anzahl der Freiheitsgrade, durch die die Verteilung gekennzeichnet ist.

Hinweis Ist eines der beiden Argumente kein numerischer Ausdruck, gibt die Funktion T.INV.2S() den Fehlerwert #WERT! zurück.

Ist *Wahrsch* kleiner 0 oder ist *Wahrsch* größer 1, gibt T.INV.2S() den Fehlerwert #ZAHL! zurück.

Ist *Freiheitsgrade* keine ganze Zahl, wird der Dezimalanteil abgeschnitten.

Ist *Freiheitsgrade* kleiner 1, gibt T.INV.2S() den Fehlerwert #ZAHL! zurück.

T.INV.2S() gibt den Wert t zurück, für den gilt: $P(|X| > t) = \text{Wahrsch}$. Dabei ist X eine t-verteilte Zufallsvariable und $P(|X| > t) = P(X < -t \text{ oder } X > t)$.

Ein Quantil der t-Verteilung ist als der t-Wert eines einseitigen Konfidenzintervalls interpretierbar. Aufgrund der Symmetrie der t-Verteilung kann der t-Wert für ein einseitiges Konfidenzintervall durch Ersetzen des Arguments *Wahrsch* durch $2 * \text{Wahrsch}$ ermittelt werden.

Für eine Wahrscheinlichkeit von 0,05 und 10 Freiheitsgraden wird der t-Wert für ein beidseitiges Konfidenzintervall mit =T.INV.2S() (0,05;10) berechnet. Das Ergebnis ist 2,28139.

Der t-Wert für ein einseitiges Konfidenzintervall für die gleiche Wahrscheinlichkeit und die gleiche Anzahl Freiheitsgrade kann mit =T.INV.2S() (2*0,05;10) berechnet werden. Das Ergebnis lautet dann 1,812462.

Bei gegebenem Wert für das Argument *Wahrsch* sucht die Funktion T.INV.2S() den Wert x so, dass $T.VERT.2S(x; \text{Freiheitsgrade}; 2) = \text{Wahrsch}$ gilt. Daher hängt die Genauigkeit von T.INV.2S() von der Genauigkeit von T.VERT.2S() ab. T.INV.2S() geht bei der Suche iterativ vor. Hat die Suche nach 100 Iterationsschritten noch nicht konvergiert, gibt die Funktion den Fehlerwert #NV zurück.

Hintergrund Der t-Wert, der durch die Funktion T.INV.2S() zurückgegeben wird, wird auch als »kritischer Wert« bezeichnet und stellt somit eine Prüfgröße für die aufgestellten Hypothesen dar. Auf Basis dieser Prüfgröße können anschließend weitere Tests – z.B. mit der Funktion T.VERT.2S() – durchgeführt werden, um Aussagen über die Gültigkeit der Nullhypothese zu treffen.

Die Argumente der Funktion sind die Wahrscheinlichkeit und die Anzahl der Freiheitsgrade. Das Argument *Wahrsch* stellt das Signifikanzniveau α dar, das unter anderem durch die Funktion T.VERT.2S() berechnet werden kann. Beim einseitigen t-Test wird dieses verdoppelt. Die Anzahl der Freiheitsgrade berechnet sich nach der Summe der beiden Stichprobenumfänge vermindert um 2.

Mehr Informationen zur Funktion T.VERT.2S() und den t-Tests finden Sie auf der Seite 539 bzw. 533.

Hinweis

Im Rahmen einer medizinischen Studie wird die Verträglichkeit eines Medikaments untersucht. Dazu werden die Testpersonen in zwei Gruppen eingeteilt: Eine Gruppe nimmt die normale Tagesdosis, die andere Gruppe nimmt einmalig zu Beginn eine erhöhte Dosis. Aus privaten Gründen muss eine Testperson den Test frühzeitig abbrechen.

Praxiseinsatz

Das Ziel ist es zu prüfen, ob die erhöhte Dosis den Heilungsprozess beschleunigen kann. Dazu wird die gesamte Behandlungsdauer in Tagen ermittelt.

Die Nullhypothese besagt, dass es *keinen* Unterschied in beiden Gruppen bezüglich des Behandlungserfolgs gibt. Die Gegenhypothese besagt, dass Gruppe 2 schneller wieder gesund ist, da die Behandlungsmethode erfolgreicher als die herkömmliche ist.

Es wird ein *einseitiger t-Test* mit dem *Typ 2*, also dem Mittelwertvergleich von zwei unabhängigen Stichproben, und einem Signifikanzniveau von 0,05 verwendet. Sie wollen nun den kritischen Wert des Stichprobenumfangs über die Funktion T.INV.2S() berechnen. Die Berechnung von T.INV.2S() zeigt die nachfolgende Abbildung 11.137.

	E	C	D	E	F	G	H
7	Gruppe 1 x_1	Gruppe 2 x_2		Berechnete/gegebene Werte	Wert		
8	7	2		Beob. Mittelwert $\bar{x}_1 =$	6,60	=SUMME(B8:B17)/ANZAHL(B8:B17)	
9	8	4		Beob. Mittelwert $\bar{x}_2 =$	4,33	=SUMME(C8:C17)/ANZAHL(C8:C17)	
10	8	3		Signifikanzniveau $\alpha =$	0,050000001	= gegeben	
11	9	5		Stichprobenumfang $N_1 =$	10	=ANZAHL(B8:B17)	
12	6	2		Stichprobenumfang $N_2 =$	9	=ANZAHL(C8:C17)	
13	8	9		Freiheitsgrade = $N_1 + N_2 - 2 =$	17	=F11+F12-2	
14	4	5		Prüfgröße t	2,39049197	= festgelegt	
15	7	6		Kritische Wert	GESUCHT		
16	6	3					
17	3						
18							
19							
20							
21							
22							
23							
24							
25	Ergebnis für T.INV.2S						
	1,739606716						
		=T.INV.2S(F10*2;F13)					

Abbildung 11.137: T.INV.2S() liefert den t-Wert – auch kritischer Wert genannt

Der t-Wert für die Anzahl der untersuchten Stichproben auf Basis einer bestimmten Wahrscheinlichkeit beträgt 1,7396. Dieser kritische Wert ist eine Prüfgröße und dient weiteren Untersuchungen bezüglich der Nullhypothese.

Der t-Wert kann auch aus einer t-Tabelle in Statistikbüchern abgelesen werden.

Hinweis

T.TEST(), T.VERT.2S()

Siehe auch

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *T.inv.2s*.



T.INV() T.INV()

Syntax T.INV(*Wahrsch*; *Freiheitsgrade*)

Definition Die Funktion T.INV() gibt linksseitige Quantile der (Student) t-Verteilung zurück.

Argumente *Wahrsch* (erforderlich) ist die zur t-Verteilung gehörige Wahrscheinlichkeit (zweiseitig).
Freiheitsgrade (erforderlich) ist die Anzahl der Freiheitsgrade, durch die die Verteilung gekennzeichnet ist.

Hinweis Ist eines der Argumente kein numerischer Ausdruck, gibt T.INV() den Fehlerwert #WERT! zurück.

Ist *Wahrsch* ≤ 0 oder ist *Wahrsch* > 1 , gibt T.INV() den Fehlerwert #ZAHL! zurück.

Ist *Freiheitsgrade* keine ganze Zahl, werden die Nachkommastellen abgeschnitten.

Ist *Freiheitsgrade* < 1 , gibt T.INV() den Fehlerwert #ZAHL! zurück.

Hintergrund Hintergrund zum t-Wert bzw. zum kritischen Wert sowie zum t-Test lesen Sie im Zusammenhang mit der Funktion T.INV.2S() auf Seite 526 und auf Seite 533.

Praxiseinsatz Mehr zum Einsatz der Funktion erfahren Sie mithilfe des Praxiseinsatzes der Funktion T.INV.2S() auf Seite 526.

Siehe auch T.TEST(), T.VERT.2S()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *T.inv*.

TREND() TREND()

Syntax TREND(*Y_Werte*; *X_Werte*; *Neue_X_Werte*; *Konstante*)

Definition Die Funktion TREND() liefert Werte, die sich aus einem linearen Trend ergeben. Diese Funktion passt den als Matrizen *Y_Werte* und *X_Werte* übergebenen Werten eine Gerade an (nach der Methode der kleinsten Quadrate). Als Ergebnis liefert die Funktion die auf der Geraden liegenden *y*-Werte, die zu den in *Neue_X_Werte* angegebenen *x*-Werten gehören.

Argumente *Y_Werte* (erforderlich) sind die *y*-Werte, die Ihnen bereits aus der Beziehung $y = mx + b$ bekannt sind.

- ▶ Besteht die Matrix *Y_Werte* aus nur einer Spalte, wird jede Spalte der Matrix *X_Werte* als eigenständige Variable interpretiert
- ▶ Besteht die Matrix *Y_Werte* aus nur einer Zeile, wird jede Zeile der Matrix *X_Werte* als eigenständige Variable interpretiert

X_Werte (optional) sind *x*-Werte, die Ihnen möglicherweise bereits aus der Beziehung $y = mx + b$ bekannt sind.

- ▶ Die Matrix *X_Werte* kann eine oder mehrere Variablengruppen umfassen. Wird nur eine Variable verwendet, können *Y_Werte* und *X_Werte* Bereiche beliebiger Form sein, solange sie dieselben Dimensionen haben. Werden mehrere Variablen verwendet, muss *Y_Werte* ein Vektor sein, das heißt ein Bereich, der aus nur einer Zeile oder nur einer Spalte besteht.

- ▶ Fehlt die Matrix *X_Werte*, wird an ihrer Stelle die Matrix $\{1;2;3;...\}$ angenommen, die genauso viele Elemente wie *Y_Werte* enthält
- ▶ Die Argumente *Y_Werte* und *X_Werte* sind auch bei dieser Funktion Matrizen, die jeweils die gleiche Anzahl an Zeilen (Spalten) haben müssen. Ist dies nicht der Fall, erhalten Sie den Fehlerwert *#BEZUG!*. Ist einer der *y*-Werte gleich Null oder negativ, erhalten Sie den Fehlerwert *#ZAHL!*.

Neue_X_Werte (optional) sind die neuen *x*-Werte, für die die Funktion *TREND()* die zugehörigen *y*-Werte liefern soll.

- ▶ Analog zur Matrix *X_Werte* muss auch *Neue_X_Werte* für jede unabhängige Variable eine eigene Spalte (oder Zeile) bereitstellen. Daher müssen die Matrizen *X_Werte* und *Neue_X_Werte* gleich viele Spalten haben, wenn *Y_Werte* sich in einer einzelnen Spalte befindet. Befindet sich *Y_Werte* in einer einzelnen Zeile, müssen die Matrizen *X_Werte* und *Neue_X_Werte* gleich viele Zeilen haben.
- ▶ Fehlt die Matrix *Neue_X_Werte*, wird angenommen, dass sie mit der Matrix *X_Werte* identisch ist
- ▶ Fehlt sowohl die Matrix *X_Werte* als auch die Matrix *Neue_X_Werte*, werden diese als die Matrix $\{1;2;3;...\}$ angenommen, die genauso viele Elemente wie die Matrix *Y_Werte* enthält

Konstante (optional) ist ein Wahrheitswert, der angibt, ob die Konstante *b* den Wert 0 annehmen soll.

- ▶ Ist *Konstante* mit *WAHR* belegt oder nicht angegeben, wird *b* normal berechnet
- ▶ Ist *Konstante* mit *FALSCH* belegt, wird *b* gleich 0 (Null) gesetzt und *m* so angepasst, dass $y = mx$ gilt

Sie können *TREND()* zur Anpassung von Polynomen verwenden, indem Sie eine lineare Regression bezüglich Potenzen einer Variablen durchführen. Nehmen wir beispielsweise an, die Spalte *A* enthält *y*-Werte und die Spalte *B* enthält *x*-Werte. Sie könnten *x*² in Spalte *C*, *x*³ in Spalte *D* eingeben usw. und anschließend eine Regression für die Spalten *B:D* gegen die Spalte *A* durchführen.

Hinweis

Formeln, die als Ergebnis eine Matrix liefern, müssen als Matrixformeln eingegeben werden.

Wird anstelle eines Arguments (z.B. *X_Werte*) eine Matrixkonstante eingegeben, müssen Sie das Semikolon verwenden, um die zu einer Zeile gehörenden Werte voneinander zu trennen, und Punkte, um die Zeilen selbst voneinander zu trennen.

Wenn bekannt ist, dass verschiedene Werte voneinander abhängig sind, kann man aus den bekannten Werten auf die Zukunft schließen.

Hintergrund

In Excel gibt es eine Reihe statistischer Funktionen, die für die Berechnung eines Trends verwendet werden können. Die statistischen Funktionen berechnen auf der Grundlage von bekannten Werten eine Gerade bzw. Kurve. Wenn Sie die Zeitachse verlängern, können Sie künftige Werte ablesen. Es wird also versucht, die bekannten Werte zu analysieren und mit einer Formel zu beschreiben, die eine Fortschreibung der Werte erlaubt. Vorausgesetzt wird dabei jedoch, dass die Datenreihe lang genug ist, um beispielsweise saisonale Schwankungen ausgleichen zu können.

Auch sonstige Einflüsse, die nicht vorhersehbar sind, aber auf die Entwicklung einen starken Einfluss haben, könnten unter Umständen, trotz tatkräftiger Unterstützung Ihres Rechners, Probleme bereiten.

Zum Beispiel, wenn ein Mitbewerber mit einem neuen Produkt und riesigem Werbeetat ausgerechnet in Ihrer Region Traumumsätze erzielt. Da die Regressionsanalyse davon ausgeht, dass die Daten über eine mathematische Funktion angenähert werden können, bietet Excel mehrere derartige Funktionen an, unter anderem die Funktion TREND().

Diese Funktion verwenden Sie, um einen linearen Trend zu berechnen. Mit der Funktion TREND() können Sie bekannte Werte analysieren. Hierbei werden diese Werte in einer Formel fortgeschrieben und erlauben so Aussagen über zukünftige Entwicklungen.

Die *Y_Werte* und *X_Werte* sind die bereits bekannten Werte aus der Beziehung $y = mx + b$ mit dem Achsenabschnitt *b*, der uns den Schnittpunkt der Geraden mit der Y-Achse angibt, und Steigung *m*, die uns zeigt, um wie viel ein *y*-Wert sich bei einer Einheit der *x*-Wert verändert.

Wenn sich also eine Zahlenreihe immer konstant um einen bestimmten Wert ändert, liegt ein linearer Trend vor.

Praxiseinsatz

Als Marketingleiter eines Softwareunternehmens sind Sie dabei, die firmeneigene Webseite zu analysieren. Die Zugriffe auf die Webseite sowie die Onlinebestellungen sind in der vergangenen Zeit merklich angestiegen.

Da Sie wissen möchten, wie sich beide Komponenten in der Zukunft entwickeln werden, wollen Sie mithilfe der Funktion TREND() die zukünftigen Werte berechnen, um so auf die geschätzte Anzahl der Webseitenzugriffe und der Onlinebestellungen schließen zu können.

Bis Juni 2008 stellen sich die Webseitenzugriffe und Bestellungen wie in Abbildung 11.138 dar.

	B	C	D
8	Monat	Webzugriffe	Bestellungen
9	Januar 2006	236	6
10	Februar 2006	1.221	17
11	März 2006	6.195	456
12	April 2006	2.682	544
13	Mai 2006	4.569	349
14	Juni 2006	12.229	854
15	Juli 2006	12.564	427
16	August 2006	10.157	337
17	September 2006	15.160	899
18	Oktober 2006	10.716	1.011
19	November 2006	9.844	720
20	Dezember 2006	11.975	1.069
21	Januar 2007	24.695	1.070
22	Februar 2007	11.593	498
23	März 2007	18.491	1.401
24	April 2007	11.743	1.076
25	Mai 2007	11.452	1.563
26	Juni 2007	26.651	1.790
27	Juli 2007	16.287	1.367
28	August 2007	17.750	1.138
29	September 2007	19.985	1.352
30	Oktober 2007	17.285	1.343
31	November 2007	30.369	1.430
32	Dezember 2007	19.674	1.140
33	Januar 2008	28.464	1.421
34	Februar 2008	25.000	1.508
35	März 2008	24.574	2.137
36	April 2008	23.141	1.948
37	Mai 2008	17.700	1.521
38	Juni 2008	22.456	1.304
39	Juli 2008		
40	August 2008		
41	September 2008		
42	Oktober 2008		
43	November 2008		
44	Dezember 2008		
45	Januar 2009		
46	Februar 2009		
47	März 2009		

Abbildung 11.138: Übersicht über Webseitenzugriffe und Onlinebestellungen bis Juni 2008

Sie erstellen aus den bisher generierten Werten ein Diagramm, um zu sehen, wie sich ein linearer Trend sowohl für den Bereich Webseitenzugriff als auch für den Bereich Bestellung darstellt – inklusive der jeweiligen Gleichung und dem berechneten Wert für r^2 (siehe Abbildung 11.139).

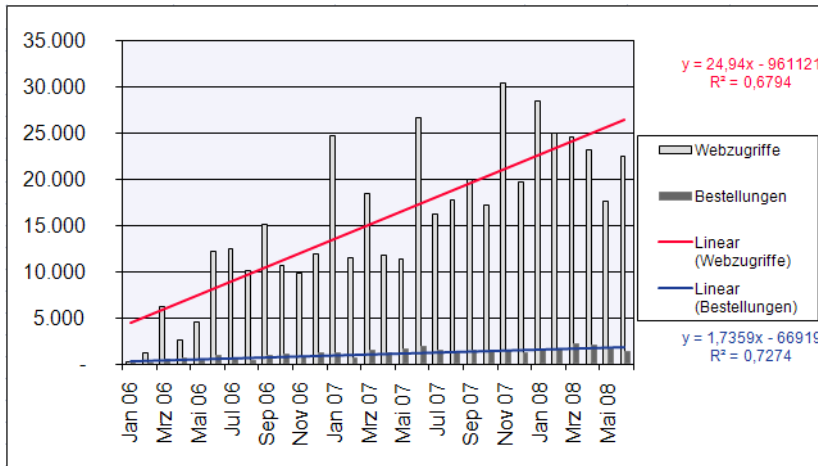


Abbildung 11.139: Die Darstellung der Webseitenzugriffe und Onlinebestellungen in Diagrammform

Betrachtet man die lineare Trendlinie und die dazugehörige Funktion der Bestellungen näher, so kann man die Aussage treffen, dass die Bestellungen pro Monat um 52,872 steigen. Das heißt, jeden Monat kommen ca. 53 Bestellungen hinzu.

Nun wollen Sie aber wissen, wie sich sowohl die Webseitenzugriffe als auch die Onlinebestellungen von Juli 2008 bis März 2009 weiterentwickeln werden (siehe Abbildung 11.140).

Verwenden Sie die Funktion TREND(), haben Sie zur Berechnung der Webseitenzugriffe für die kommenden neun Monate folgende Argumente zur Verfügung:

- ▶ *Y_Werte* = Webseitenzugriffe von Januar 2006 bis Juni 2008
- ▶ *X_Werte* = Monate (Januar 2006 bis Juni 2008)
- ▶ *Neue_X_Werte* = Monate von Juli 2008 bis März 2009
- ▶ *Konstante* = WAHR, das heißt die Konstante b der Gleichung $y = mx + b$ wird normal berechnet

Das Ergebnis stellt sich wie in Abbildung 11.140 dar:

Denken Sie bei der Berechnung der gesuchten Zukunftswerte mit der Funktion TREND() für den Bereich C39:C47 daran (Abbildung 11.140), dass es sich hier um eine Matrixfunktion handelt. Das heißt, die Funktion liefert sofort für alle Monate das gewünschte Ergebnis. Deshalb müssen Sie den gesamten Bereich von C39:C47 markieren, die Formel eingeben und die Eingabe mit der Tastenkombination Strg + ↕ + ↵ beenden.

Sie erkennen Matrixfunktionen an den geschweiften Klammern (siehe die Formel in der Bearbeitungsleiste von Abbildung 11.140).

Hinweis

C39		fx {=TREND(C9:C38;B9:B38;B39:B47;WAHR)}	
	B	C	D
8	Monat	Webzugriffe	Bestellungen
9	Januar 2006	236	6
10	Februar 2006	1.221	17
11	März 2006	6.195	456
12	April 2006	2.682	544
13	Mai 2006	4.569	349
14	Juni 2006	12.229	854
15	Juli 2006	12.564	427
16	August 2006	10.157	337
17	September 2006	15.160	899
18	Oktober 2006	10.716	1.011
19	November 2006	9.844	720
20	Dezember 2006	11.975	1.069
21	Januar 2007	24.695	1.070
22	Februar 2007	11.593	498
23	März 2007	18.491	1.401
24	April 2007	11.743	1.076
25	Mai 2007	11.452	1.563
26	Juni 2007	26.651	1.790
27	Juli 2007	16.287	1.367
28	August 2007	17.750	1.138
29	September 2007	19.985	1.352
30	Oktober 2007	17.285	1.343
31	November 2007	30.369	1.430
32	Dezember 2007	19.674	1.140
33	Januar 2008	28.464	1.421
34	Februar 2008	25.000	1.508
35	März 2008	24.574	2.137
36	April 2008	23.141	1.948
37	Mai 2008	17.700	1.521
38	Juni 2008	22.456	1.304
39	Juli 2008	27.258	
40	August 2008	28.031	
41	September 2008	28.804	
42	Oktober 2008	29.552	
43	November 2008	30.326	
44	Dezember 2008	31.074	
45	Januar 2009	31.847	
46	Februar 2009	32.620	
47	März 2009	33.318	

Abbildung 11.140: Die Berechnung der zukünftigen Webseitenzugriffe mithilfe der Funktion *TREND()*

Nach dem gleichen Verfahren berechnen Sie nun noch, abhängig von den soeben errechneten Trendwerten für die Webseitenzugriffe, die zukünftigen Werte für den Bereich *Bestellungen*.

Das Ergebnis und die Eingabe der Argumente in die Funktion *TREND()* entnehmen Sie bitte der Abbildung 11.141.

Excel ermöglicht Ihnen mit der Funktion *TREND()* nun eine recht gute Entwicklungsprognose für die Bereiche *Webseitenzugriffe* und *Bestellungen*. Voraussetzung ist selbstverständlich, der bisherige Trend hält an.

	B	C	D
8	Monat	Webzugriffe	Bestellungen
9	Januar 2006	236	6
10	Februar 2006	1.221	17
11	März 2006	6.195	456
12	April 2006	2.682	544
13	Mai 2006	4.569	349
14	Juni 2006	12.229	854
15	Juli 2006	12.564	427
16	August 2006	10.157	337
17	September 2006	15.160	899
18	Oktober 2006	10.716	1.011
19	November 2006	9.844	720
20	Dezember 2006	11.975	1.069
21	Januar 2007	24.695	1.070
22	Februar 2007	11.593	498
23	März 2007	18.491	1.401
24	April 2007	11.743	1.076
25	Mai 2007	11.452	1.563
26	Juni 2007	26.651	1.790
27	Juli 2007	16.287	1.367
28	August 2007	17.750	1.138
29	September 2007	19.985	1.352
30	Oktober 2007	17.285	1.343
31	November 2007	30.369	1.430
32	Dezember 2007	19.674	1.140
33	Januar 2008	28.464	1.421
34	Februar 2008	25.000	1.508
35	März 2008	24.574	2.137
36	April 2008	23.141	1.948
37	Mai 2008	17.700	1.521
38	Juni 2008	22.456	1.304
39	Juli 2008	27.258	1.706
40	August 2008	28.031	1.748
41	September 2008	28.804	1.791
42	Oktober 2008	29.552	1.832
43	November 2008	30.326	1.875
44	Dezember 2008	31.074	1.916
45	Januar 2009	31.847	1.959
46	Februar 2009	32.620	2.002
47	März 2009	33.318	2.040

Abbildung 11.141: Die Berechnung der Trendwerte für den Bereich *Bestellungen*

RGF(), RKP(), VARIATION(), SCHÄTZER()

Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Regression.xlsx* auf dem Arbeitsblatt *Trend*.

Siehe auch



T.TEST() / TTEST()



T.TEST() / TTEST()

T.TEST(*Matrix1*; *Matrix2*; *Seiten*; *Typ*)

Die Funktion T.TEST() gibt die Teststatistik eines Studentschen t-Tests zurück. Mithilfe von T.TEST() können Sie testen, ob zwei Stichproben aus zwei Grundgesamtheiten mit demselben Mittelwert stammen.

Matrix1 (erforderlich) ist die erste Datengruppe.

Matrix2 (erforderlich) ist die zweite Datengruppe.

Syntax

Definition

Argumente

Seiten (erforderlich) bestimmt die Anzahl der Endflächen (Schwänze). Ist *Seiten* = 1, verwendet T.TEST() eine Endfläche (einseitiger Test). Ist *Seiten* = 2, verwendet T.TEST() zwei Endflächen (zweiseitiger Test).

Typ (erforderlich) bestimmt den Typ des durchzuführenden t-Tests.

- ▶ Ist der Typ gleich »1«, wird der Test »gepaart« ausgeführt
- ▶ Ist der Typ gleich »2«, wird der Test »Zwei Stichproben, gleiche Varianz (homoskedastisch)« ausgeführt
- ▶ Ist der Typ gleich »3«, wird der Test »Zwei Stichproben, ungleiche Varianz (heteroskedastisch)« ausgeführt

Hinweis Enthalten *Matrix1* und *Matrix2* unterschiedlich viele Datenpunkte und ist *Typ* = 1 (gepaart), gibt TTEST() den Fehlerwert #NV zurück.

Die Argumente *Seiten* und *Typ* werden durch Abschneiden ihrer Nachkommastellen zu ganzen Zahlen gekürzt.

Ist *Seiten* oder *Typ* kein numerischer Ausdruck, gibt T.TEST() den Fehlerwert #WERT! zurück.

Ist *Seiten* mit einem Wert ungleich 1 oder 2 belegt, gibt T.TEST() den Fehlerwert #ZAHL! zurück.

Die Funktion T.TEST() verwendet die Daten aus *Matrix1* und *Matrix2*, um eine nicht negative t-Kenngröße zu berechnen. Wenn *Seiten* = 1 ist, gibt T.TEST() die Wahrscheinlichkeit zurück, die es für einen größeren Wert der t-Kenngröße unter der Annahme gibt, dass *Matrix1* und *Matrix2* »Stichproben aus Grundgesamtheiten mit demselben Mittelwert« sind. Der von T.TEST() zurückgegebene Wert ist, wenn *Seiten* = 2 ist, doppelt so groß wie der Wert, der zurückgegeben wird, wenn *Seiten* = 1 ist, und entspricht der Wahrscheinlichkeit eines größeren absoluten Werts der t-Kenngröße unter der Annahme »identische Mittelwerte der Grundgesamtheiten«.

Hintergrund Die Funktionen der t-Verteilung geben Aufschluss darüber, ob eine oder zwei Stichproben der Normalverteilung entsprechen. So kann beispielsweise getestet werden, ob eine medizinische Behandlungsmethode besser ist als eine andere.

Entwickelt wurde die t-Verteilung, die zu den Wahrscheinlichkeitsverteilungen gehört, 1908 von William Sealey Gosset (Pseudonym Student). Er hatte festgestellt, dass standardisierte normalverteilte Daten nicht mehr normalverteilt sind, wenn die Varianz des Merkmals unbekannt ist und mit der Stichprobenvarianz geschätzt werden muss.

Die t-Verteilung ist vom Mittelwert μ und der Standardabweichung s unabhängig und hängt lediglich vom Freiheitsgrad ab.

Hinweis Die t-Verteilung ist der Standardnormalverteilung sehr ähnlich. Denn wie die Standardnormalverteilung ist auch die t-Verteilung stetig, symmetrisch, glockenförmig und hat einen Variationsbereich von plus/minus unendlich.

Da die Normalverteilung nur für eine große Anzahl an Daten gilt, muss sie in der Regel korrigiert werden. Diese Unsicherheit wird in der t-Verteilung durch die modifizierte und symmetrische Art der Verteilung berücksichtigt. Liegen hohe Freiheitsgrade vor, geht die t-Verteilung in die Normalverteilung über.

Je weniger Freiheitsgrade vorliegen, desto mehr entfernen sich die Integralgrenzen bei vorgegebener Wahrscheinlichkeit und festgelegter Standardabweichung vom Mittelwert, sodass bei zweiseitigem Test das Intervall breiter als 1,96 (für $P = 0,95$) wird.

Die t-Verteilung beschreibt die Verteilung eines Ausdrucks

$$t_m = \frac{N(0,1)}{\sqrt{\frac{\chi_m^2}{m}}}$$

wobei $N(0,1)$ eine standardnormalverteilte Zufallsvariable bedeutet und eine χ^2 -verteilte mit m Freiheitsgraden. Die Zählervariable muss unabhängig von der Nennervariablen sein. Die Dichtefunktion der t-Verteilung ist dann symmetrisch bezüglich ihres Erwartungswerts 0.

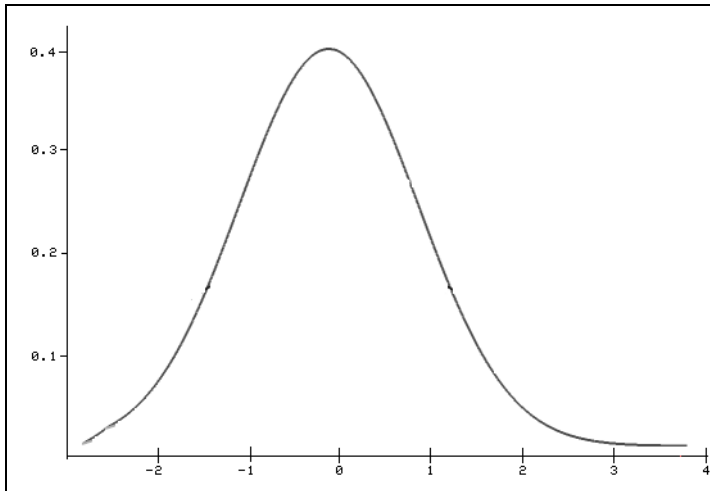


Abbildung 11.142: Dichtefunktion von t-verteilten Zufallsgrößen

Der t-Test erlaubt Hypothesen für kleinere Stichproben, wenn die Grundgesamtheit eine Normalverteilung aufweist, ein bestimmter Mittelwert angenommen wird und die Standardabweichung unbekannt ist.

Es gibt drei verschiedene Typen von t-Tests:

1. Vergleich des Mittelwerts aus der Stichprobe mit dem Mittelwert aus der Grundgesamtheit
Ein Beispiel hierfür wäre der Vergleich des Durchschnittsalters in München mit dem aus der Population der gesamten Bundesrepublik.
2. Mittelwertvergleich aus unabhängigen Stichproben

Als Beispiel kann man hier den Vergleich des durchschnittlichen Einkommens von Männern und Frauen in München nennen. Da die Varianzen der beiden Grundgesamtheiten, aus denen die beiden Stichproben gezogen wurden, meistens unbekannt sind, werden die Varianzen aus der Stichprobe geschätzt. Die Teststatistik T folgt einer t-Verteilung mit k Freiheitsgraden.

Hinweis Sind die Varianzen der Grundgesamtheiten bekannt, aus denen die beiden Stichproben stammen, so ist die Teststatistik normalverteilt.

3. Vergleich der Stichprobenmittelwerte aus voneinander abhängigen Stichproben.

Ein Beispiel hierfür wäre der Bildungsvergleich zwischen Ehepartnern. Dieser Fall liegt z.B. vor, wenn bei den Untersuchungseinheiten einer Stichprobe ein Merkmal zweimal gemessen wird und daraufhin geprüft wird, ob der zweite Messwert höher (oder niedriger) als der erste liegt. Die Größe T folgt einer t -Verteilung mit $n - 1$ Freiheitsgraden. n ist hier die Anzahl der Messwertpaare.

Die für den t -Test relevante Fragestellung lautet:

Mit welcher Wahrscheinlichkeit ist ein gefundener Unterschied der Mittelwerte, errechnet durch eine der drei soeben erläuterten Typen von t -Tests, zufällig zustande gekommen? Und mit welcher Wahrscheinlichkeit begeht man einen Alpha-Fehler, wenn man aufgrund des in der Stichprobe gefundenen Unterschieds der Mittelwerte annimmt, dass dieser Unterschied auch in der Grundgesamtheit vorhanden ist?

Hinweis Der Alpha-Fehler, auch Alpha-Risiko genannt, ist die Wahrscheinlichkeit dafür, dass das Datenmaterial eine entsprechende Ausprägung aus reinem Zufall annimmt. Der Alpha-Fehler liegt sehr oft bei 10 %, 5 % oder weniger als 5 % (Signifikanzniveau) und ist selten größer als 10 %.

In wenigen Worten zusammengefasst wird mit dem t -Test die (Fehler-) Wahrscheinlichkeit der stichprobenbasierten Behauptungen überprüft, d.h., es wird überprüft, mit welcher Wahrscheinlichkeit ein Alpha-Fehler begangen wird.

t-Test-Typen Es gibt zusätzlich zu den verschiedenen Typen von t -Tests noch zwei generelle Unterscheidungsarten:

1. Einseitiger t -Test

Beim einseitigen t -Test werden sogenannte *gerichtete* Hypothesen aufgestellt. Für H_0 und H_1 gilt dann:

$$H_0: \mu_1 - \mu_2 \text{ kleiner oder gleich } 0$$

$$H_1: \mu_1 - \mu_2 > 0$$

Um ein konventionelles Alpha-Niveau (Signifikanzniveau) von 5 % zu erreichen, muss t auf der vorhergesagten Seite der t -Verteilung im extremen 5 %-Bereich liegen.



Abbildung 11.143: Grafische Darstellung eines einseitigen t -Tests

2. Zweiseitiger t-Test

Beim zweiseitigen t-Test werden sogenannte ungerichtete Hypothesen aufgestellt. Für H_0 und H_1 gilt dann:

$$H_0: \mu_1 - \mu_2 = 0$$

$$H_1: \mu_1 - \mu_2 \neq 0$$

Um insgesamt das Alpha-Niveau (Signifikanzniveau) von 5 % zu erreichen, muss t in der t -Verteilung im extremen unteren oder oberen 2,5 %-Bereich liegen.

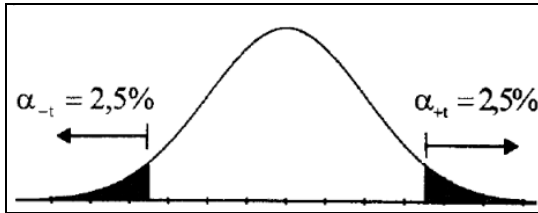


Abbildung 11.144: Grafische Darstellung eines zweiseitigen t-Tests

Einseitiger und zweiseitiger t-Test im Vergleich:

- ▶ t muss beim zweiseitigen Test einen größeren Wert annehmen, um signifikant ($\alpha=0.05$) zu werden
- ▶ Demselben t wird im zweiseitigen Test eine größere Irrtumswahrscheinlichkeit zugeordnet als im einseitigen Test
- ▶ Dasselbe $X_1 - X_2$ wird im einseitigen Test leichter signifikant als im zweiseitigen Test
- ▶ Der einseitige Test hat eine höhere Teststärke

Generell wird einem bestimmten t -Wert im zweiseitigen Test ein doppelt so großer P -Wert wie im einseitigen Test zugeordnet. Dieser P -Wert kann deshalb problemlos in den entsprechenden P -Wert für den einseitigen Test umgerechnet werden – und umgekehrt.

Daraus ergeben sich folgende Formeln:

$$P_{\text{zweiseitig}} = 2 \times P_{\text{einseitig}}$$

$$P_{\text{einseitig}} = \frac{1}{2} P_{\text{zweiseitig}}$$

Im Rahmen einer medizinischen Studie wurde die Verträglichkeit eines Medikaments untersucht. Dazu liegen Ihnen die Testergebnisse sowie einige Erläuterungen vor: Eine Gruppe nahm die normale Tagesdosis, die andere Gruppe nahm einmalig zu Beginn eine erhöhte Dosis ein. Aus privaten Gründen musste eine Testperson den Test frühzeitig abbrechen.

Das Ziel sollte es sein, zu prüfen, ob die erhöhte Dosis den Heilungsprozess beschleunigen kann. Dazu wurde die Behandlungsdauer in Tagen ermittelt.

Die Nullhypothese besagt nun, dass es *keinen* Unterschied in beiden Gruppen bezüglich des Behandlungserfolgs gibt. Die Gegenhypothese besagt, dass Gruppe 2 schneller wieder gesund ist, da die Behandlungsmethode erfolgreicher als die herkömmliche ist.

Praxiseinsatz

Ihnen wurde nun die Aufgabe übertragen, die Testergebnisse zu analysieren und aufgrund Ihrer Auswertungen eine Aussage darüber zu treffen, ob die Nullhypothese angenommen werden kann oder verworfen werden muss.

Da Sie während der Testphase nicht anwesend waren und Ihnen noch einige Hintergrundinformationen fehlen, wollen Sie zunächst mit der Funktion T.TEST() herausfinden, mit wievielperzentiger Wahrscheinlichkeit die Mittelwerte der beiden Stichproben gleich sind. Verglichen mit dem Signifikanzniveau kann daraufhin eine erste Aussage über die Nullhypothese getroffen werden.

Sie führen einen *einseitigen t-Test* mit dem *Typ 2*, also dem Mittelwertvergleich von zwei unabhängigen Stichproben, durch. Das *Signifikanzniveau* beträgt 5 %.

Die Prüfgröße *t* wird mit folgender Formel berechnet:

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\left[\frac{\sum X_1^2 - \frac{(\sum X_1)^2}{N_1}}{N_1 + N_2 - 2} \right] + \left[\frac{\sum X_2^2 - \frac{(\sum X_2)^2}{N_2}}{N_1 + N_2 - 2} \right]} \times \left(\frac{N_1 + N_2}{N_1 \times N_2} \right)}$$

wobei die *X*-Werte über dem Bruchstrich die Mittelwerte von Gruppe 1 und 2, *N*₁ und *N*₂ den Umfang der beiden Stichproben darstellen.

Das Ergebnis der Studie zeigt die nachfolgende Abbildung 11.145.

	B	C	D	E	F
9	Gruppe 1 x₁	Gruppe 2 x₂		Berechnete/gegebene Werte	Wert
10	7	2		Beob. Mittelwert X ₁ ' =	6,60
11	8	4		Beob. Mittelwert X ₂ ' =	4,33
12	8	3		Signifikanzniveau α =	0,05
13	9	5		Stichprobenumfang N ₁ =	10
14	6	2		Stichprobenumfang N ₂ =	9
15	8	9		Freiheitsgrade = N ₁ +N ₂ -2 =	17
16	4	5		Prüfgröße t	2,39049197
17	7	6			
18	6	3			
19	3				
20					
21	Ergebnis für T.TEST				
22	0,01433616	=T.TEST(B10:B19;C10:C19;1,2)			

Abbildung 11.145: Sind die Mittelwerte der beiden Stichproben gleich? Die Berechnung erfolgt über T.TEST().

Da T.TEST() den Wert für eine Wahrscheinlichkeit liefert, lautet das Ergebnis 1,4 %. Sie können nun davon ausgehen, dass mit 1,4%iger Wahrscheinlichkeit die Stichprobenmittelwerte der beiden Stichproben nicht gleich sind. Oder anders ausgedrückt: Mit 1,4%iger Sicherheit sind die Stichproben nicht als gleich zu betrachten. Diese Tatsache haben wir in Abbildung 11.145 bereits in den Zellen F10 und F11 nachgewiesen. Allerdings sind dies nur geschätzte Werte, da es sich um Stichproben handelt.

Da in diesem Beispiel das Ergebnis für T.TEST() < α, muss die Nullhypothese verworfen werden. Das heißt, die Aussage, dass es keinen Unterschied zwischen den beiden Gruppen bezüglich der Behandlungsmethode gibt, wird nicht angenommen.

T.VERT.2S(), T.VERT.RE(), T.VERT(), T.INV(), T.INV.2S

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *T.test*.

Siehe auch



T.VERT.2S() / TVERT() T.DIST.2T() / TDIST()

Die Funktion TVERT() wird seit Microsoft Excel 2010 durch die Funktion T.VERT.2S() ersetzt und zusätzlich durch die Funktionen T.VERT() und T.VERT.RE ergänzt. Damit wird das Ergebnis bzw. die Genauigkeit der Funktionen erhöht.

Hinweis

Um die Abwärtskompatibilität von T.VERT.2S() zu sichern, ist die Funktion TVERT() weiter unter ihrem alten Namen verfügbar.

T.VERT.2S(x;Freiheitsgrade)

Syntax

Die Funktion T.VERT.2S() gibt Werte der Verteilungsfunktion (1-Alpha) einer t-verteiltern Zufallsvariablen zurück. Die t-Verteilung wird für das Testen von Hypothesen bei kleinem Stichprobenumfang verwendet. Sie können diese Funktion anstelle einer Wertetabelle mit den kritischen Werten der t-Verteilung heranziehen.

Definition

x (erforderlich) ist der Wert der Verteilung (Quantil), dessen Wahrscheinlichkeit Sie berechnen möchten.

Argumente

Freiheitsgrade (erforderlich) ist eine ganze Zahl, durch die die Anzahl der Freiheitsgrade bestimmt wird.

Ist eines der Argumente kein numerischer Ausdruck, gibt T.VERT.2S() den Fehlerwert *#WERT!* zurück.

Hinweis

Ist *Freiheitsgrade* < 1, gibt T.VERT.2S() den Fehlerwert *#ZAHL!* zurück.

Ist *x* < 0, gibt T.VERT.2S() den Fehlerwert *#ZAHL!* zurück

Wie unter der Definition bereits erwähnt, berechnen Sie mit der Funktion T.VERT.2S() das Signifikanzniveau (Alpha-Risiko) einer t-verteiltern Zufallsgröße. Mit diesem Signifikanzniveau kann die Wahrscheinlichkeit einer Hypothese beurteilt werden.

Hintergrund

Mehr Informationen zu t-verteiltern Zufallsgrößen und der t-Verteilung im Allgemeinen finden Sie unter der Funktion T.TEST() auf Seite 533.

Hinweis

Interessant wird die Berechnung dieses Signifikanzniveaus, wenn Sie beispielsweise einen kritischen Wert für den gesamten Stichprobenumfang berechnen und für diesen kritischen Wert anschließend das Signifikanzniveau über die Funktion T.VERT.2S() ermitteln.

Mithilfe des Ergebnisses aus der Funktion T.VERT.2S() können Sie eine Aussage über die Gültigkeit der Nullhypothese treffen.

Im Rahmen einer medizinischen Studie wird die Verträglichkeit eines Medikaments untersucht. Dazu werden die Testpersonen in zwei Gruppen eingeteilt: Eine Gruppe nimmt die normale Tagesdosis, die andere Gruppe nimmt einmalig zu Beginn eine erhöhte Dosis. Aus privaten Gründen muss eine Testperson den Test frühzeitig abbrechen.

Praxiseinsatz

Das Ziel ist es zu prüfen, ob die erhöhte Dosis den Heilungsprozess beschleunigen kann. Dazu wird die gesamte Behandlungsdauer in Tagen ermittelt.

Die Nullhypothese besagt, dass es *keinen* Unterschied in beiden Gruppen bezüglich des Behandlungserfolgs gibt. Die Gegenhypothese besagt, dass Gruppe 2 schneller wieder gesund ist, da die Behandlungsmethode erfolgreicher als die herkömmliche ist.

Es wird ein *zweiseitiger t-Test* mit dem *Typ 2*, also dem Mittelwertvergleich von zwei unabhängigen Stichproben, verwendet.

Sie wollen nun das Signifikanzniveau für den kritischen Wert der Stichproben berechnen, um auf Basis dieses Ergebnisses eine Aussage über die Hypothesen treffen zu können. Hierzu verwenden Sie die Funktion T.VERT.2S().

Die Berechnung von T.VERT.2S() zeigt die nachfolgende Abbildung 11.146.

B		C		D	E	F
8	Gruppe 1 x_1	Gruppe 2 x_2			Berechnete/gegebene Werte	Wert
9	7	2			Beob. Mittelwert X_1' =	6,60
10	8	4			Beob. Mittelwert X_2' =	4,33
11	8	3			Signifikanzniveau α =	
12	9	5			Stichprobenumfang N_1 =	10
13	6	2			Stichprobenumfang N_2 =	9
14	8	9			Freiheitsgrade = $N_1 + N_2 - 2$ =	17
15	4	5			Prüfgröße t	2,39049197
16	7	6			Kritische Wert	1,739606716
17	6	3				
18	3					
19						
20	Ergebnis für T.TEST					
21	0,01433616	=T.TEST(B9:B18;C9:C18;1;2)				
22						
23	Ergebnis für T.VERT.2S					
24	0,10000002	=T.VERT.2S(F16;F14)				

Abbildung 11.146: T.VERT.2S() liefert das Signifikanzniveau für den kritischen Wert

Da T.VERT.2S() den Wert für eine Wahrscheinlichkeit liefert, lautet das Ergebnis 10 %. Sie können nun davon ausgehen, dass mit 10%iger Wahrscheinlichkeit die Nullhypothese zutrifft.

Diese Wahrscheinlichkeit ist sehr gering – deshalb kann man davon ausgehen, dass die Nullhypothese – also die Aussage, dass es keine Unterschiede in den beiden Gruppen bezüglich der Behandlungserfolge gibt – verworfen werden muss.

Siehe auch T.TEST(), T.INV(), T.INV.2S



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *T.vert.2s*.

T.VERT() T.DIST()

T.VERT(*x;Freiheitsgrade;kumuliert*)

Gibt die Student-t-Verteilung zurück. Die t-Verteilung wird in der Hypothesenüberprüfung von kleinen Beispieldatasets verwendet. Verwenden Sie diese Funktion anstelle einer Tabelle mit kritischen Werten für die t-Verteilung.

x (erforderlich) ist der numerische Wert, für den die Verteilung ausgewertet werden soll.

Freiheitsgrade (erforderlich) ist eine ganze Zahl, mit der die Anzahl der Freiheitsgrade angegeben wird.

kumuliert (erforderlich) ist der Wahrheitswert, der den Typ der Funktion bestimmt. Ist *kumuliert* mit WAHR belegt, gibt T.VERT() die kumulierte Verteilungsfunktion zurück. Ist *kumuliert* mit FALSCH belegt, gibt die Funktion die Wahrscheinlichkeitsdichtefunktion zurück.

Ist eines der Argumente kein numerischer Ausdruck, gibt T.VERT() den Fehlerwert #WERT! zurück.

Ist *Freiheitsgrade* < 1, gibt T.VERT() den Fehlerwert #ZAHL! zurück.

Ist *x* < 0, gibt T.VERT() den Fehlerwert #ZAHL! zurück.

Mehr Informationen zu t-verteilten Zufallsgrößen und der t-Verteilung im Allgemeinen finden Sie unter der Funktion T.TEST() auf Seite 533.

Mehr zum Einsatz der Funktion erfahren Sie mithilfe des Praxiseinsatzes der Funktion T.VERT.2S() auf Seite 539.

T.TEST(), T.INV(), T.INV.2S

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *T.vert.*

Syntax

Definition

Argumente

Hinweis

Hintergrund

Praxiseinsatz

Siehe auch



T.VERT.RE() T.DIST.RT()

T.VERT.RE(*x;Freiheitsgrade*)

Gibt die (Student) t-Verteilung für die rechte Endfläche zurück. Die t-Verteilung wird für das Testen von Hypothesen bei kleinem Stichprobenumfang verwendet. Verwenden Sie diese Funktion anstelle einer Tabelle mit kritischen Werten für die t-Verteilung.

x (erforderlich) ist der numerische Wert, für den die Verteilung ausgewertet werden soll.

Freiheitsgrade (erforderlich) ist eine ganze Zahl, mit der die Anzahl der Freiheitsgrade angegeben wird.

Ist eines der Argumente kein numerischer Ausdruck, gibt T.VERT.RE() den Fehlerwert #WERT! zurück.

Ist *Freiheitsgrade* < 1, gibt T.VERT.RE() den Fehlerwert #ZAHL! zurück.

Ist *x* < 0, gibt T.VERT.RE() den Fehlerwert #ZAHL! zurück.

Syntax

Definition

Argumente

Hinweis

Hintergrund Mehr Informationen zu t-verteilten Zufallsgrößen und der t-Verteilung im Allgemeinen finden Sie unter der Funktion T.TEST() auf Seite 533.

Praxiseinsatz Mehr zum Einsatz der Funktion erfahren Sie mithilfe des Praxiseinsatzes der Funktion T.VERT.2S() auf Seite 539.

Siehe auch T.TEST(), T.INV(), T.INV.2S



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *T.vert.re*.

VAR.S() / VARIANZ()

 VAR.S() / VAR()

Syntax VAR.S(*Zahl1*; *Zahl2*; ...)

Definition Mit der Funktion VAR.S() wird die Varianz auf der Basis einer Stichprobe geschätzt. Sie ist ein Maß dafür, wie die einzelnen Daten um den Mittelwert verteilt sind, das heißt, wie stark die Daten um den Mittelwert streuen.

Argumente *Zahl1* (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 numerische Argumente (30 bis Excel 2003), die einer Stichprobe aus der Grundgesamtheit entsprechen.

Hinweis Die Funktion VAR.S() geht davon aus, dass die ihr übergebenen Argumente eine Stichprobe, gezogen aus einer Grundgesamtheit, darstellen. Entsprechen die als Argumente übergebenen Daten dagegen einer Grundgesamtheit, sollte die zugehörige Varianz mithilfe der Funktion VAR.P() berechnet werden.

Logische Werte wie *WAHR* und *FALSCH* sowie Text werden hierbei nicht berücksichtigt. Wenn logische Werte und Text berücksichtigt werden sollen, verwenden Sie die Arbeitsblatffunktion VAR.P().

Hintergrund Die gebräuchlichsten Streumaße in der Statistik sind sicher die Varianz und die Standardabweichung. Die Varianz ist ein Maß für die Abweichung einer Zufallsvariable x von ihrem Erwartungswert $E(x)$. Anders ausgedrückt ist die Varianz der Durchschnitt der quadratischen Abweichungen der Einzelwerte vom Durchschnitt, dividiert durch die Anzahl der Werte. Das Ergebnis wird auch als sogenannte empirische Varianz bezeichnet.

Man unterscheidet zunächst:

- ▶ **Varianz einer Zufallsvariablen:** Das ist die durchschnittliche quadratische Abweichung der Ausprägungen vom Durchschnitt in der Grundgesamtheit. Verwendete Funktion ist VAR.P().
- ▶ **Stichprobenvarianz:** Das ist die Varianz von Beobachtungswerten, die als Stichprobe einer Grundgesamtheit entstammen. Diese Varianz wird in der deskriptiven Statistik als Maß für die Streubreite von Daten verwendet. Als inferentielle Varianz dient sie zur Schätzung der unbekanntenen Varianz in der Grundgesamtheit. Verwendete Funktion ist VAR.S().

Wenn es sich bei den untersuchten Daten um eine Stichprobe handelt, wird die Summe der quadrierten Abweichungen nicht durch die Anzahl der Werte (n), sondern durch die um eins reduzierte Anzahl ($n - 1$), dividiert.

VAR.S() verwendet daher die folgende Formel:

$$\frac{\sum (x - \bar{x})^2}{(n-1)}$$

Dabei ist x der Stichprobenmittelwert MITTELWERT(Zahl1;Zahl2;...) und n der Stichprobenumfang.

Beachten Sie bei der Verwendung der Varianz, dass wegen der Quadrierung die Extremwerte eventuell ein größeres Gewicht haben.

Der Nachteil der Varianz liegt vor allem darin, dass sie eine andere Einheit als die Daten besitzt – nämlich die Einheit zum Quadrat. Man verwendet daher oft auch die Standardabweichung, die als Quadratwurzel aus der Varianz definiert ist.

Da die Stichprobenvarianz als Maß für die Streubreite von Daten verwendet wird, taucht sie in der deskriptiven Statistik recht häufig auf.

Praxiseinsatz

Auch die Marketingabteilung des Softwareherstellers hat sich der Funktion VAR.S() angenommen, um noch detaillierte Auswertungen bezüglich der Webseite und der Zugriffe darauf vorzunehmen. Ziel soll es nach abschließender Untersuchung sein, noch genauere Aussagen über Veränderungen treffen zu können und dementsprechend auch Verbesserungen in den einzelnen Bereichen vorzunehmen.

Beachten Sie, dass es sich bei den zu untersuchenden Daten um eine Stichprobe handelt. Das heißt, die Webseite und somit auch die Zugriffe darauf gibt es bereits seit langem. Für die Berechnung der Varianz mit den Funktionen VAR.S() und VARIANZA() werden jedoch lediglich 18 Monate, von Januar 07 bis Juni 08 betrachtet.

	B	C	D	E	F	G	H	I
12	DATUM	ADVENT	DOWNLOAD	EVENT	TRAINING	WISSEN	PRODUKTE	GESAMT
13	Jan 07		6		19	43	22	90
14	Feb 07		498	38	1319	3000	1170	6025
15	Mrz 07		1401	119	2233	6116	1545	11414
16	Apr 07		1076	16	1903	2860	1168	7023
17	Mai 07		1563	15	1589	3126	1139	7432
18	Jun 07		1790	1853	2428	6682	5083	17836
19	Jul 07		1367	1622	1559	3311	3609	11468
20	Aug 07		1138	2170	1502	3317	4810	12937
21	Sep 07		1352	2611	1628	3542	5581	14714
22	Okt 07	1	1343	538	1440	2971	5506	11799
23	Nov 07	272	1430	1254	2255	4735	11786	21732
24	Dez 07	1874	1140	631	1583	2652	6227	14107
25	Jan 08	42	1421	1515	2224	4837	9500	19539
26	Feb 08	32	1508	1433	1418	4006	12000	20397
27	Mrz 08	10	2137	3114	1298	3128	8000	17687
28	Apr 08	10	1948	2347	1085	3575	8208	17173
29	Mai 08	9	1521	848	765	2139	7739	13021
30	Jun 08		426	440	123	768	778	2535
31	Summe	2250	23065	20564	26371	60808	93871	226929
32	Mittelwert	281,25	1281,39	1209,65	1465,06	3378,22	5215,06	12607,17
33	Mittlere Abweichung	398,2	378,3	827,3	472,6	1093,9	3067,9	4785,7
34	Varianz (VAR.S)	422303	279408	966160	444330	2561816	14361231	37396918

Abbildung 11.147: Mithilfe der Funktion VAR.S() wird berechnet, wie die einzelnen Daten auf Basis einer Stichprobe um den Mittelwert verteilt sind

Wie Sie der Abbildung 11.147 entnehmen können, haben die Mitarbeiter der Marketing-Abteilung für den besseren Überblick neben der Varianz sowohl den Mittelwert als auch die mittlere Abweichung berechnet.

Aussagen Betrachten wir den Bereich *DOWNLOAD* näher, kann folgende Aussage getroffen werden:

Die durchschnittlich quadrierte Abweichung der einzelnen Messwerte vom arithmetischen Mittel beträgt im Bereich *DOWNLOAD* 279.408.

Hinweis Würden die Abweichungen nicht quadriert, ergäbe die Summe der Abweichungen den Wert 0, also eine recht unfruchtbare statistische Operation.

Siehe auch `VARIANZA()`, `VAR.P()`, `VARIANZENA()`, `DBVARIANZ()`, `DBVARIANZEN()`



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Varianz.xlsx* auf dem Arbeitsblatt *Var.s*.

VARIANZA() VARA()

Syntax `VARIANZA(Wert1;Wert2;...)`

Definition Die Funktion `VARIANZA()` schätzt die Varianz auf der Basis einer Stichprobe. Wichtig hierbei: Neben Zahlen werden auch Text und Wahrheitswerte wie *WAHR* und *FALSCH* bei der Berechnung berücksichtigt.

Argumente *Wert1* (erforderlich); *Wert2* (optional); ... sind 1 bis 255 Werte (30 bis Excel 2003), die einer Stichprobe aus der Grundgesamtheit entsprechen.

Hinweis `VARIANZA()` geht davon aus, dass die Argumente einer Stichprobe der Grundgesamtheit entsprechen. Stehen die Daten für die ganze Grundgesamtheit, verwenden Sie die Funktion `VARIANZENA()`.

Der Unterschied zwischen den Funktionen `VAR.S()` und `VARIANZA()` liegt darin, dass bei `VARIANZA()` Argumente, die den Wert *WAHR* enthalten, zu 1 ausgewertet und Argumente, die Text oder den Wahrheitswert *FALSCH* enthalten, zu 0 ausgewertet werden.

Das heißt: Wollen Sie, dass bei der Berechnung weder Text noch Wahrheitswerte berücksichtigt werden, müssen Sie die Funktion `VAR.S()` verwenden.

Hintergrund Da sich die Funktion `VARIANZA()` zur Funktion `VAR.S()` lediglich darin unterscheidet, dass sie Wahrheitswerte sowie Text berücksichtigt, richten wir unser Augenmerk hier vor allem auf die Darstellung der `VARIANZA()` im Beispiel.

Detaillierte Informationen zum Thema Varianz können Sie auf Seite 541 unter `VAR.S()` nachlesen.

`VARIANZA()` verwendet die gleiche Formel wie `VAR.S()`.

$$\frac{\sum (x - \bar{x})^2}{(n - 1)}$$

Dabei ist x der Stichprobenmittelwert `MITTELWERT(Wert1;Wert2;...)` und n der Stichprobenumfang.

Betrachten wir wieder die Auswertung der Zugriffe auf die Webseite des Softwareherstellers. Da es im Betrachtungszeitraum, also in den vergangenen 18 Monaten, verschiedene Probleme mit und auf der Webseite gab, konnten nicht für alle Monate die Zugriffe gezählt werden.

Folgende Probleme lagen im Unternehmen vor:

- ▶ Im Monat Mai 07 und August 07 gab es Probleme mit dem Webseitenhosting, sodass in beiden Monaten nicht auf die Webseite zugegriffen werden konnte. Diese Monate wurden mit dem Text »Hostingprobleme« versehen.
- ▶ Im Monat März 08 wurden die Inhaltsseiten des Produktbereichs umgestellt und angepasst, sodass der gesamte Bereich geschlossen war. Da somit von extern kein Zugriff auf den Produktbereich möglich war, wurde dieser Monat mit dem Wahrheitswert *FALSCH* gekennzeichnet.

	B	C	D	E	F	G	H	I
	DATUM	ADVENT	DOWNLOAD	EVENT	TRAINING	WISSEN	PRODUKTE	GESAMT
13	Jan 07		6		19	43	22	90
14	Feb 07		498	38	1319	3000	1170	6025
15	Mrz 07		1401	119	2233	6116	1545	11414
16	Apr 07		1076	16	1903	2660	1168	7023
17	Mai 07	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme
18	Jun 07		1790	1853	2428	6682	5083	17836
19	Jul 07		1367	1622	1559	3311	3609	11468
20	Aug 07	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme
21	Sep 07		1352	2611	1628	3542	5581	14714
22	Okt 07	1	1343	538	1440	2971	5506	11799
23	Nov 07	272	1430	1254	2255	4735	11786	21732
24	Dez 07	1874	1140	631	1583	2652	6227	14107
25	Jan 08	42	1421	1515	2224	4837	9500	19539
26	Feb 08	32	1508	1433	1418	4006	12000	20397
27	Mrz 08	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH
28	Apr 08	10	1948	2347	1085	3575	8208	17173
29	Mai 08	9	1521	848	765	2139	7739	13021
30	Jun 08		426	440	123	768	778	2535
31	Summe	2240	18227	15265	21982	51237	79922	188873
32	Mittelwert	320	1215,13	1090,36	1465,47	3415,80	5328,13	12591,53
33	Mittlere Abweichung	444,0	390,6	714,6	545,2	1277,7	3189,6	5038,2
34	Varianz	478672	275154	714102	536355	3099980	15672494	41646292
35	VARIANZA	343008	443737	763784	757527	4268761	17081614	57612639

Abbildung 11.148: Text und Wahrheitswerte werden bei der Berechnung der Varianz mithilfe der Funktion *VARIANZA()* berücksichtigt

Wie Sie in Abbildung 11.148 sehen können, liefert die Funktion *VARIANZA()* ein anderes Ergebnis als die Berechnung der Varianz über die Funktion *VAR.S()*. Das liegt an der Tatsache, dass *VARIANZA()* wie bereits erwähnt, Text und Wahrheitswerte berücksichtigt. Da es sich in unserem Beispiel um Text und den Wahrheitswert *FALSCH* handelt, wurden beide Werte auf 0 gesetzt.

Betrachten wir den Bereich *DOWNLOAD* näher, kann folgende Aussage getroffen werden:

Aussagen

Die durchschnittlich quadrierte Abweichung der einzelnen Messwerte vom arithmetischen Mittel beträgt im Bereich *DOWNLOAD*, unter Berücksichtigung von Text und Wahrheitswerten, 443.737.

VAR.S(), *VAR.P()*, *VARIANZENA()*, *DBVARIANZ()*, *DBVARIANZEN()*

Siehe auch

Dieses Beispiel finden Sie im Ordner *\\Ms5-235\Kap11* in der Arbeitsmappe *Varianz.xlsx* auf dem Arbeitsblatt *Varianza*.



VAR.P() / VARIANZEN()**VAR.P() / VARP()****Syntax** VAR.P(*Zahl1*; *Zahl2*; ...)**Definition** Die Funktion VAR.P() berechnet die Varianz ausgehend von der Grundgesamtheit.**Argumente** *Zahl1* (erforderlich); *Zahl2* (optional); ... sind 1 bis 255 numerische Argumente (30 bis Excel 2003), die einer Grundgesamtheit entsprechen.**Hinweis** Die Funktion VAR.P() geht davon aus, dass die ihr übergebenen Argumente einer Grundgesamtheit entsprechen. Für den Fall, dass die von Ihnen bereitgestellten Daten nur eine Stichprobe einer Grundgesamtheit angeben, sollte die Varianz mit der Funktion VAR.S() bzw. VARIANZA() berechnet werden.Logische Werte wie *WAHR* und *FALSCH* sowie Text werden nicht berücksichtigt. Möchten Sie logische Werte in Ihrer Berechnung berücksichtigen, verwenden Sie die Arbeitsblatffunktion VARIANZENA().**Hintergrund** Da sich die Funktion VAR.P() zur Funktion VAR.S() lediglich darin unterscheidet, dass sie die Varianz nicht auf Basis einer Stichprobe sondern auf Basis einer Grundgesamtheit berechnet, richten wir unser Augenmerk hier vor allem auf die Darstellung der VAR.P() im Beispiel.

Detaillierte Informationen zum Thema Varianz können Sie auf Seite 541 unter VAR.S() nachlesen.

VAR.P() verwendet folgende Formel:

$$\frac{\sum (x - \bar{x})^2}{n}$$

Dabei ist x der Stichprobenmittelwert MITTELWERT(*Zahl1*; *Zahl2*; ...) und n der Stichprobenumfang.**Praxiseinsatz** Kommen wir wieder zur Webseitenauswertung unseres Softwareherstellers und betrachten Abbildung 11.149 etwas genauer.**Hinweis** Beachten Sie, dass es sich bei den zu untersuchenden Daten um die Grundgesamtheit handelt. Das heißt, die Webseite besteht erst seit 18 Monaten und somit wurden auch die Zugriffe darauf erst seit 18 Monaten registriert. Für die Berechnung der Varianz mit den Funktionen VAR.P() und VARIANZENA() wird also die Grundgesamtheit der Daten von Januar 07 bis Juni 08 betrachtet.

Deshalb liefern die Funktionen VAR.P() und VAR.S(), die unterschiedlichen Formeln unterliegen, auch unterschiedliche Ergebnisse.

Aussagen Betrachten wir den Bereich *DOWNLOAD* näher, kann folgende Aussage getroffen werden:

Die durchschnittlich quadrierte Abweichung der einzelnen Messwerte vom arithmetischen Mittel auf Basis der Grundgesamtheit beträgt 263.885.

	B	C	D	E	F	G	H	I
	DATUM	ADVENT	DOWNLOAD	EVENT	TRAINING	WISSEN	PRODUKTE	GESAMT
12	Jan 07		6		19	43	22	90
13	Feb 07		498	38	1319	3000	1170	6025
14	Mrz 07		1401	119	2233	6116	1545	11414
15	Apr 07		1076	16	1903	2860	1168	7023
16	Mai 07		1563	15	1589	3126	1139	7432
17	Jun 07		1790	1853	2428	6682	5083	17836
18	Jul 07		1367	1622	1559	3311	3609	11468
19	Aug 07		1138	2170	1502	3317	4810	12937
20	Sep 07		1352	2611	1628	3542	5581	14714
21	Okt 07	1	1343	538	1440	2971	5506	11799
22	Nov 07	272	1430	1254	2255	4735	11786	21732
23	Dez 07	1874	1140	631	1583	2652	6227	14107
24	Jan 08	42	1421	1515	2224	4837	9500	19539
25	Feb 08	32	1508	1433	1418	4006	12000	20397
26	Mrz 08	10	2137	3114	1298	3128	8000	17687
27	Apr 08	10	1948	2347	1085	3575	8208	17173
28	Mai 08	9	1521	848	765	2139	7739	13021
29	Jun 08		426	440	123	768	778	2535
30	Summe	2250	23065	20564	26371	60808	93871	226929
31	Mittelwert	281,25	1281,39	1209,65	1465,06	3378,22	5215,06	12607,17
32	Mittlere Abweichung	398,2	378,3	827,3	472,6	1093,9	3067,9	4785,7
33	Varianz	422303	279408	966160	444330	2561816	14361231	37396918
34	VAR.P()	369515	263885	909327	419645	2419493	13563385	35319311

Abbildung 11.149: Im Gegensatz zu VAR.S() wird bei VAR.P() die Grundgesamtheit betrachtet

VAR.S(), VARIANZA(), VARIANZENA(), DBVARIANZ(), DBVARIANZEN()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Varianz.xlsx* auf dem Arbeitsblatt *Var.p.*

Siehe auch



VARIANZENA() VARPA()

VARIANZENA(*Wert1*; *Wert2*; ...)

Die Funktion VARIANZENA() berechnet die Varianz ausgehend von der Grundgesamtheit. Neben Zahlen werden bei der Berechnung auch Text und Wahrheitswerte wie WAHR und FALSCH berücksichtigt.

Wert1 (erforderlich); *Wert2* (optional); ... sind 1 bis 255 Wertargumente (30 bis Excel 2003), die einer Grundgesamtheit entsprechen.

VARIANZENA() geht davon aus, dass die Argumente der Grundgesamtheit entsprechen. Sind die Daten eine Stichprobe aus Grundgesamtheit, berechnen Sie die Varianz mithilfe der Funktion VARIANZA().

WAHR als Argument evaluiert zu 1; Text oder der Wahrheitswert FALSCH evaluiert zu 0.

Wollen Sie bei der Berechnung weder Text noch Wahrheitswerte berücksichtigen, verwenden Sie stattdessen die Funktion VAR.P().

Da wir das Thema Varianz bereits ausführlich auf Seite 541 unter VAR.S() behandelt haben, möchten wir in diesem Abschnitt unser Augenmerk hauptsächlich auf das Beispiel legen.

Syntax

Definition

Argumente

Hinweis

Hintergrund

VARIANZENA() verwendet, wie VAR.P(), folgende Formel:

$$\frac{\sum (x - \bar{x})^2}{n}$$

Dabei ist x der Stichprobenmittelwert MITTELWERT(Wert1;Wert2;...) und n der Stichprobenumfang.

Praxiseinsatz Zur Darstellung der Funktion VARIANZENA() möchten wir auf das Beispiel aus VARIANZA() zurückgreifen. In diesem Beispiel hatte der Softwarehersteller folgende Probleme mit seiner Webseite:

- ▶ Im Monat Mai 07 und August 07 gab es Probleme mit dem Webseitenhosting, sodass in beiden Monaten nicht auf die Webseite zugegriffen werden konnte. Diese Monate wurden mit dem Text »Hostingprobleme« versehen.
- ▶ Im Monat März 08 wurden die Inhaltsseiten des Produktbereichs umgestellt und angepasst, sodass der gesamte Bereich geschlossen war. Da somit von extern kein Zugriff auf den Produktbereich möglich war, wurde dieser Monat mit dem Wahrheitswert *FALSCH* gekennzeichnet.

Hinweis Beachten Sie, dass es sich, wie unter VAR.P(), bei den zu untersuchenden Daten um die Grundgesamtheit handelt. Das heißt, die Webseite besteht erst seit 18 Monaten und somit wurden auch die Zugriffe darauf erst seit 18 Monaten registriert.

	B	C	D	E	F	G	H	I
11	DATUM	ADVENT	DOWNLOAD	EVENT	TRAINING	WISSEN	PRODUKTE	GESAMT
12	Jan 07		6		19	43	22	90
13	Feb 07		498	38	1319	3000	1170	6025
14	Mrz 07		1401	119	2233	6116	1545	11414
15	Apr 07		1076	16	1903	2860	1168	7023
16	Mai 07	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme
17	Jun 07		1790	1853	2428	6682	5083	17836
18	Jul 07		1367	1622	1559	3311	3609	11468
19	Aug 07	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme	Hostingprobleme
20	Sep 07		1352	2611	1628	3542	5581	14714
21	Okt 07	1	1343	538	1440	2971	5506	11799
22	Nov 07	272	1430	1254	2255	4735	11786	21732
23	Dez 07	1874	1140	631	1583	2652	6227	14107
24	Jan 08	42	1421	1515	2224	4837	9500	19539
25	Feb 08	32	1508	1433	1418	4006	12000	20397
26	Mrz 08	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH	FALSCH
27	Apr 08	10	1948	2347	1085	3575	8208	17173
28	Mai 08	9	1521	848	765	2139	7739	13021
29	Jun 08		426	440	123	768	778	2535
30	Summe	2240	18227	15265	21982	51237	79922	188873
31	Mittelwert	320	1215,13	1090,36	1465,47	3415,80	5328,13	12591,53
32	Mittlere Abweichung	444,0	390,6	714,6	545,2	1277,7	3189,6	5038,2
33	VAR.P()	410290	256810	663094	500598	2893314	14627661	38869873
34	VARIANZENA()	308707	419085	718856	715442	4031608	16132635	54411937

Abbildung 11.150: Die Berechnung der Varianz, ausgehend von einer Grundgesamtheit, unter Berücksichtigung von Text und Wahrheitswerten

Wie Sie in Abbildung 11.150 sehen können, liefert die Funktion VARIANZENA() ein anderes Ergebnis als die Berechnung der Varianz über die Funktion VAR.P(). Das liegt an der Tatsache, dass VARIANZENA() wie bereits erwähnt, Text und Wahrheitswerte berücksichtigt. Da es sich in unserem Beispiel um Text und den Wahrheitswert *FALSCH* handelt, wurden beide Werte auf 0 gesetzt.

Betrachten wir den Bereich *DOWNLOAD* näher, kann folgende Aussage getroffen werden:

Die durchschnittlich quadrierte Abweichung der einzelnen Messwerte vom arithmetischen Mittel, auf Basis der Grundgesamtheit und unter Berücksichtigung von Text und Wahrheitswerten beträgt im Bereich *DOWNLOAD* 419.085.

VAR.S(), VARIANZA(), VAR.P(), DBVARIANZ(), DBVARIANZEN()

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap11` in der Arbeitsmappe *Varianz.xlsx* auf dem Arbeitsblatt *Varianzena*.

Aussagen

Siehe auch



VARIATION() GROWTH()

VARIATION(*Y_Werte*; *X_Werte*; *Neue_x_Werte*; *Konstante*)

Die Funktion VARIATION() liefert Werte, die sich aus einem exponentiellen Trend ergeben. VARIATION() liefert die *y*-Werte für eine Reihe neuer *x*-Werte, die Sie mithilfe vorhandener *x*- und *y*-Werte festlegen. Sie können die VARIATION() auch verwenden, um eine zu den vorhandenen *x*- und *y*-Werten passende Exponentialkurve zu ermitteln.

Y_Werte (erforderlich) sind die *y*-Werte, die Ihnen bereits aus der Beziehung $y = b * mx$ bekannt sind.

- ▶ Besteht die Matrix *Y_Werte* aus nur einer Spalte, wird jede Spalte der Matrix *X_Werte* als eigenständige Variable interpretiert
- ▶ Besteht die Matrix *Y_Werte* aus nur einer Zeile, wird jede Zeile der Matrix *X_Werte* als eigenständige Variable interpretiert

X_Werte (optional) sind *x*-Werte, die Ihnen möglicherweise bereits aus der Beziehung $y = b * mx$ bekannt sind.

- ▶ Die Matrix *X_Werte* kann eine oder mehrere Variablengruppen umfassen. Wird nur eine Variable verwendet, können *Y_Werte* und *X_Werte* Bereiche beliebiger Form sein, solange sie dieselben Dimensionen haben. Werden mehrere Variablen verwendet, muss *Y_Werte* ein Vektor sein, das heißt ein Bereich, der aus nur einer Zeile oder nur einer Spalte besteht.
- ▶ Fehlt die Matrix *X_Werte*, wird an ihrer Stelle die Matrix $\{1;2;3;...\}$ angenommen, die genauso viele Elemente wie *Y_Werte* enthält
- ▶ Die Argumente *Y_Werte* und *X_Werte* sind auch bei dieser Funktion Matrizen, die jeweils die gleiche Anzahl an Zeilen (Spalten) haben müssen. Ist dies nicht der Fall, erhalten Sie den Fehlerwert *#BEZUG!*. Ist einer der *y*-Werte gleich Null oder negativ, erhalten Sie den Fehlerwert *#ZAHL!*.

Neue_X_Werte (optional) sind die neuen *x*-Werte, für die die Funktion TREND() die zugehörigen *y*-Werte liefern soll.

- ▶ Analog zur Matrix *X_Werte* muss auch *Neue_X_Werte* für jede unabhängige Variable eine eigene Spalte (oder Zeile) bereitstellen. Daher müssen die Matrizen *X_Werte* und *Neue_X_Werte* gleich viele Spalten haben, wenn *Y_Werte* sich in einer einzelnen Spalte befindet. Befindet sich *Y_Werte* in einer einzelnen Zeile, müssen die Matrizen *X_Werte* und *Neue_X_Werte* gleich viele Zeilen haben.

Syntax

Definition

Argumente

- ▶ Fehlt die Matrix *Neue_X_Werte*, wird angenommen, dass sie mit der Matrix *X_Werte* identisch ist
 - ▶ Fehlt sowohl die Matrix *X_Werte* als auch die Matrix *Neue_X_Werte*, werden diese als die Matrix $\{1;2;3;\dots\}$ angenommen, die genauso viele Elemente wie die Matrix *Y_Werte* enthält
- Konstante* (optional) ist ein Wahrheitswert, der angibt, ob die Konstante *b* den Wert 0 annehmen soll.
- ▶ Ist *Konstante* mit *WAHR* belegt oder nicht angegeben, wird *b* normal berechnet
 - ▶ Ist *Konstante* mit *FALSCH* belegt, wird *b* gleich 0 (Null) gesetzt und *m* so angepasst, dass $y = mx$ gilt

Hinweis Formeln, die ihre Ergebnisse als Matrizen liefern, müssen, nachdem die richtige Anzahl von Zellen markiert ist, als Matrixformeln eingegeben werden.

Wird anstelle eines Arguments (z.B. *X_Werte*) eine Matrixkonstante eingegeben, müssen Sie ein Semikolon verwenden, um die zu einer Zeile gehörenden Werte voneinander zu trennen, und Punkte, um die Zeilen selbst voneinander zu trennen.

Hintergrund Entsprechend der Funktion *TREND()*, die einen linearen Trend – also künftige Werte – aus bereits bestehenden Werten berechnet, können Sie auch einen exponentiellen Trend berechnen. Denn nicht alle Modelle folgen dem linearen Trend.

Hinweis Mehr Informationen zu »Trends« in Daten und der Funktion *TREND()* finden Sie auf Seite 528 dieses Buchs.

Ein exponentieller Trend liegt vor, wenn sich der Wert immer um den gleichen Faktor oder Prozentsatz ändert. Der exponentielle Trend passt den jeweiligen Daten eine Exponentialkurve an. Um ein solches exponentielles Trendmodell zu berechnen, verwenden Sie die Funktion *VARIATION()*.

Praxiseinsatz Bleiben wir bei dem Beispiel wie unter *TREND()* bereits erwähnt. Als Marketingleiter eines Softwareunternehmens analysieren Sie momentan die firmeneigene Webseite. Die Zugriffe auf die Webseite als auch die Onlinebestellungen sind in der vergangenen Zeit merklich angestiegen.

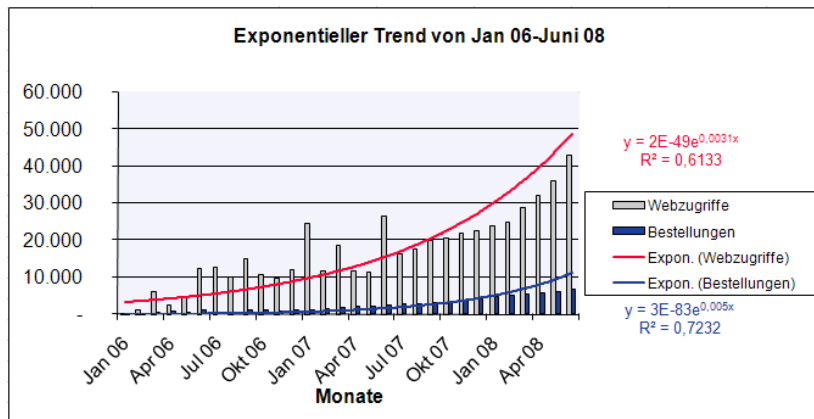


Abbildung 11.151: Die exponentiellen Trendlinien für den Bereich *Webseitenzugriff* und *Onlinebestellung*

Eine Grafik mit eingefügter Trendlinie verdeutlicht Ihnen, dass sich der Anstieg der Webseitenzugriffe und auch der Onlinebestellungen eher exponentiell entwickelt (siehe Abbildung 11.151).

Sie kennen die Zahlen für die Webseitenzugriffe und Onlinebestellungen von Januar 07 bis Juni 08. Jetzt möchten Sie wissen, wie sich beide Komponenten in der Zukunft entwickeln werden – Sie möchten also eine Prognose erstellen.

Zur Berechnung dieses exponentiellen Trends verwenden Sie die Funktion VARIATION().

Bis zum Juni 2008 stellen sich die Webseitenzugriffe und Bestellungen wie in Abbildung 11.152 dar.

	B	C	D
9	Monat	Webzugriffe	Bestellungen
10	Januar 2006	236	6
11	Februar 2006	1.221	17
12	März 2006	6.195	456
13	April 2006	2.682	544
14	Mai 2006	4.569	349
15	Juni 2006	12.229	854
16	Juli 2006	12.564	427
17	August 2006	10.157	337
18	September 2006	15.160	899
19	Oktober 2006	10.716	1.011
20	November 2006	9.844	720
21	Dezember 2006	11.975	1.069
22	Januar 2007	24.695	1.070
23	Februar 2007	11.593	1.284
24	März 2007	18.491	1.572
25	April 2007	11.743	1.874
26	Mai 2007	11.452	2.098
27	Juni 2007	26.651	2.284
28	Juli 2007	16.287	2.534
29	August 2007	17.750	2.764
30	September 2007	19.985	2.987
31	Oktober 2007	20.543	3.173
32	November 2007	21.958	3.587
33	Dezember 2007	22.645	3.982
34	Januar 2008	23.954	4.620
35	Februar 2008	24.932	4.912
36	März 2008	28.743	5.374
37	April 2008	31.948	5.729
38	Mai 2008	35.928	6.046
39	Juni 2008	42.943	6.439
40	Juli 2008		
41	August 2008		
42	September 2008		
43	Oktober 2008		
44	November 2008		
45	Dezember 2008		
46	Januar 2009		
47	Februar 2009		
48	März 2009		

Abbildung 11.152: Übersicht über Webseitenzugriffe und Onlinebestellungen bis Juni 2008

Nun wollen Sie wissen, wie sich sowohl die Webseitenzugriffe als auch die Onlinebestellungen von Juli 2008 bis März 2009 weiterentwickeln werden.

Verwenden Sie die Funktion VARIATION(), haben Sie zur Berechnung der Webseitenzugriffe für die kommenden neun Monate folgende Argumente zur Verfügung:

- ▶ Y_Werte = Webseitenzugriffe von Januar 2006 bis Juni 2008
- ▶ X_Werte = Monate (Januar 2006 bis Juni 2008)
- ▶ $Neue_X_Werte$ = Monate von Juli 2008 bis März 2009
- ▶ $Konstante$ = WAHR, das heißt die Konstante b der Gleichung $y = mx + b$ wird normal berechnet

Das Ergebnis zeigt die Abbildung 11.153.

C40		f _x [=VARIATION(C10:C39;B10:B39;B40:B48;WAHR)]		
	B	C	D	E
9	Monat	Webzugriffe	Bestellungen	
10	Januar 2006	236	6	
11	Februar 2006	1.221	17	
12	März 2006	6.195	456	
13	April 2006	2.682	544	
14	Mai 2006	4.569	349	
15	Juni 2006	12.229	854	
16	Juli 2006	12.564	427	
17	August 2006	10.157	337	
18	September 2006	15.160	899	
19	Oktober 2006	10.716	1.011	
20	November 2006	9.844	720	
21	Dezember 2006	11.975	1.069	
22	Januar 2007	24.695	1.070	
23	Februar 2007	11.593	1.284	
24	März 2007	18.491	1.572	
25	April 2007	11.743	1.874	
26	Mai 2007	11.452	2.098	
27	Juni 2007	26.651	2.284	
28	Juli 2007	16.287	2.534	
29	August 2007	17.750	2.764	
30	September 2007	19.985	2.987	
31	Oktober 2007	20.543	3.173	
32	November 2007	21.958	3.587	
33	Dezember 2007	22.645	3.982	
34	Januar 2008	23.954	4.620	
35	Februar 2008	24.932	4.912	
36	März 2008	28.743	5.374	
37	April 2008	31.948	5.729	
38	Mai 2008	35.928	6.046	
39	Juni 2008	42.943	6.439	
40	Juli 2008	53.742		
41	August 2008	59.183		
42	September 2008	65.175		
43	Oktober 2008	71.551		
44	November 2008	78.796		
45	Dezember 2008	86.504		
46	Januar 2009	95.263		
47	Februar 2009	104.908		
48	März 2009	114.457		

Abbildung 11.153: Die Berechnung der zukünftigen Werte für die exponentiell wachsenden Webseitenzugriffe

Hinweis Denken Sie bei der Berechnung der gesuchten Zukunftswerte mit der Funktion VARIATION() für den Bereich C40:C48 daran (Abbildung 11.153), dass es sich hier um eine Matrixfunktion handelt. Das heißt, die Funktion liefert sofort für alle Monate das gewünschte Ergebnis. Deshalb müssen Sie den gesamten Bereich von C40:C48 markieren, die Formel eingeben und die Eingabe mit der Tastenkombination **[Strg] + [↕] + [↵]** beenden.

Sie erkennen Matrixfunktionen an den geschweiften Klammern in der Bearbeitungsleiste für Funktionen.

Nach dem gleichen Verfahren berechnen Sie nun noch, abhängig von den soeben errechneten Trendwerten für die Webseitenzugriffe, die zukünftigen Werte für den Bereich *Bestellungen*. Ergebnis und Eingabe der Argumente in die Funktion VARIATION() entnehmen Sie bitte der Abbildung 11.154.

	B	C	D	E
9	Monat	Webzugriffe	Bestellungen	
40	Juli 2008	53.742	104.183	
41	August 2008	59.183	201.298	
42	September 2008	65.175	415.757	
43	Oktober 2008	71.551	899.527	
44	November 2008	78.796	2.161.947	
45	Dezember 2008	86.504	5.496.234	
46	Januar 2009	95.263	15.866.575	
47	Februar 2009	104.908	50.993.886	
48	März 2009	114.457	161.983.212	

Abbildung 11.154: Die Berechnung der exponentiellen Trendwerte für den Bereich *Bestellungen*

Excel ermöglicht Ihnen, mit der Funktion VARIATION() eine recht gute Entwicklungsprognose für die Bereiche *Webseitenzugriffe* und *Bestellungen*. Voraussetzung ist selbstverständlich, der bisherige exponentielle Trend hält an.

RGP(), RKP(), TREND(), SCHÄTZER()

Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Regression.xlsx* auf dem Arbeitsblatt *Variation*.

Siehe auch



VARIATIONEN()



PERMUT()

VARIATIONEN(*n*;*k*)

Die Funktion VARIATIONEN() gibt die Anzahl der Möglichkeiten zurück, um *k* Elemente aus einer Menge von *n* Elementen ohne Zurücklegen zu ziehen. Eine Variation ist eine Menge von Elementen oder Ereignissen, deren interne Anordnung oder Reihenfolge relevant ist.

n (erforderlich) ist die Anzahl aller Elemente.

k (erforderlich) gibt an, aus wie vielen Elementen jede Variationsmöglichkeit bestehen soll.

Beide Argumente werden durch Abschneiden ihrer Nachkommastellen zu ganzen Zahlen gekürzt.

Ist *n* oder *k* kein numerischer Ausdruck, gibt VARIATIONEN() den Fehlerwert #WERT! zurück.

Ist *n* kleiner oder gleich 0 bzw. *k* kleiner 0, gibt VARIATIONEN() den Fehlerwert #ZAHL! zurück.

Ist *n* kleiner *k*, gibt VARIATIONEN() den Fehlerwert #ZAHL! zurück.

Die Funktion VARIATIONEN() gehört zum Bereich der Kombinatorik, die sich mit der Bestimmung der Anzahl möglicher Anordnungen von Objekten beschäftigt.

Variationen unterscheiden sich von Kombinationen, für welche die interne Anordnung nicht relevant ist.

Syntax

Definition

Argumente

Hinweis

Hintergrund

Das heißt, bei den Variationen darf die Reihenfolge der gezogenen Elemente nachträglich nicht mehr verändert werden – bei den Kombinationen spielt das keine Rolle.

Mit der Funktion VARIATIONEN() können Sie also zum Beispiel berechnen, wie viele Möglichkeiten es für die ersten drei Plätze eines Wettlaufs mit zehn Teilnehmern gibt. Mit der Funktion KOMBINATIONEN() können Sie hingegen ermitteln, wie viele Möglichkeiten es beim Ziehen der Lottozahlen von 6 Kugeln aus 49 gibt.

Der Unterschied: Während sich die Reihenfolge der gezogenen Lottozahlen beliebig ändern lässt, würden die Läufer protestieren, wenn Sie die Plätze 1 bis 3 nun plötzlich nach den Anfangsbuchstaben der Nachnamen sortieren würden.

Bei den Variationen spielt also die Anordnung eine wichtige Rolle.

Die Funktion VARIATIONEN ermittelt also die Anzahl der geordneten Stichproben vom Umfang k , die man einer Gesamtheit von n verschiedenen Elementen ohne Zurücklegen entnehmen kann.

Die Gleichung zur Berechnung der Anzahl der Variationen lautet:

$$P_{k,n} = \frac{n!}{(n-k)!}$$

Praxiseinsatz

Bleiben wir bei dem Beispiel mit den Wettläufern. An dem Wettrennen nehmen insgesamt zehn Läufer teil. Die ersten drei Personen, die in das Ziel einlaufen, erhalten ein Preisgeld, die anderen nichts. Nun möchten Sie wissen, wie viele unterschiedliche Variationen es für die ersten 3 Plätze bezüglich der Teilnehmer gibt. Hierfür nutzen Sie die Formel VARIATIONEN.

Welche Werte sind für die Argumente n und k der Funktion gegeben?

- ▶ $n = 10$ (Anzahl der Elemente sind in diesem Beispiel die Anzahl der teilnehmenden Läufer)
- ▶ $k = 3$ (die Variationsmöglichkeit soll aus drei Elementen, also den ersten drei Plätzen, bestehen)

Das Ergebnis der Berechnung sehen Sie in Abbildung 11.155.

C21		fx =VARIATIONEN(B18;C18)	
B	C	D	
7	Teilnehmer	Preisgeld	
8	Läufer 1	Platz 1	
9	Läufer 2	Platz 2	
10	Läufer 3	Platz 3	
11	Läufer 4		
12	Läufer 5		
13	Läufer 6		
14	Läufer 7		
15	Läufer 8		
16	Läufer 9		
17	Läufer 10		
18	10	3	
19	=ANZAHL2(B8:B17)		
20			
21	Variationen	720	

Abbildung 11.155: Die Berechnung möglicher »Gewinner«-Variationen

Laut der Berechnung mithilfe der Funktion VARIATIONEN() gibt es 720 verschiedene Möglichkeiten, unter den zehn Teilnehmern die ersten drei Plätze zu belegen.

BINOM.VERT(), BINOM.VERT.BEREICH(), FAKULTÄT(), HYPGEOM.VERT(), KOMBINATIONEN(), BINOM.INV(), NEGBINOM.VERT(); VARIATIONEN2()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Variationen*.

Siehe auch



VARIATIONEN2()



PERMUTATIONA()

VARIATIONEN2(*Zahl*; *gewählte_Zahl*)

Die Funktion VARIATIONEN2() gibt die Anzahl der Möglichkeiten – mit Wiederholungen – für eine angegebene Menge von Objekten zurück, die aus der Gesamtmenge der Objekte ausgewählt werden können.

»Variation« beschreibt eine Menge von Elementen oder Ereignissen, deren interne Anordnung oder Reihenfolge relevant ist.

Zahl (erforderlich) ist eine ganze Zahl zur Angabe der Gesamtzahl aller Elemente.

gewählte_Zahl (erforderlich) gibt über eine ganze Zahl an, aus wie vielen Elementen jede Variation besteht.

Beide Argumente werden durch Abschneiden ihrer Nachkommastellen zu ganzen Zahlen gekürzt.

Wenn numerische Argumente ungültige Werte sind, z.B. wenn die Gesamtzahl 0 (Null) lautet und die gewählte Zahl größer als 0 (Null) ist, gibt die Funktion VARIATIONEN2() den Fehlerwert #ZAHL! zurück.

Wenn für numerische Argumente nicht numerische Datentypen verwendet werden, gibt VARIATIONEN2() den Fehlerwert #WERT! zurück.

Wie schon die Funktion VARIATIONEN() gehört auch die Funktion VARIATIONEN2() zum Bereich der Kombinatorik, die sich mit der Bestimmung der Anzahl möglicher Anordnungen von Objekten beschäftigt.

VARIATIONEN2() unterscheidet sich von VARIATIONEN() darin, dass bei VARIATIONEN2() das Zurücklegen – also Wiederholungen – erlaubt ist.

Grundsätzlich muss man wissen, dass sich Variationen von Kombinationen, für welche die interne Anordnung nicht relevant ist, unterscheiden. Das heißt, bei den Variationen darf die Reihenfolge der gezogenen Elemente nachträglich nicht mehr verändert werden – bei den Kombinationen spielt das keine Rolle.

Die Gleichung zur Berechnung von VARIATIONEN2() lautet, anders als bei VARIATION():

$$VARIATIONEN2 = \text{Ergebnis}^{\text{Auswahl}}$$

Weitere Informationen zum Thema Variationen lesen Sie im Abschnitt VARIATION() ab Seite 549.

**Neu in Excel
2013**

Syntax

Definition

Argumente

Hinweis

Hintergrund

Hinweis

Praxiseinsatz

Mit der Funktion VARIATIONEN2() können Sie also zum Beispiel berechnen, wie viele Möglichkeiten es für die ersten drei Plätze eines Wettlaufs mit zehn Teilnehmern gibt. Mit der Funktion KOMBINATIONEN() können Sie hingegen ermitteln, wie viele Möglichkeiten es beim Ziehen der Lottozahlen von 6 Kugeln aus 49 gibt.

Ein einfaches Beispiel für die Funktion VARIATIONEN2(). Drei Sportler möchten sich in Gruppen aufteilen. Die Frage ist nun, wie viele 2er bzw. 3er »Pärchen-Variationen« können hierbei entstehen.

Welche Werte sind für die Argumente *n* und *k* der Funktion gegeben?

- ▶ **Zahl** = 3, da es sich in diesem Beispiel um die drei Sportler Sara, Helmut und Egbert handelt
- ▶ **gewählte_Zahl** = 2 bzw. 3, da die Variationsmöglichkeit aus zwei bzw. drei Elementen bestehen sollen

Die Abbildung 11.156 verdeutlicht das Ergebnis – sowohl für beide Varianten als auch im Vergleich zur Funktion VARIATIONEN(), bei der das Zurücklegen eingesetzter Werte nicht erlaubt ist.

	B	C	D	E
9	Gruppe		2er Variationen	3er Variationen
10	Sara		9	27
11	Helmut		=VARIATIONEN2(B13;2)	=VARIATIONEN2(B13;3)
12	Egbert		mit zurücklegen	mit zurücklegen
13	3			
14			2er Variationen	3er Variationen
15			6	6
16	Darstellung der 2er Variation mit zurücklegen			
17	Sara-Helmut		=VARIATIONEN(B13;2)	=VARIATIONEN(B13;3)
18	Sara-Egbert		ohne zurücklegen	ohne zurücklegen
19	Sara-Sara			
20	Helmut-Sara			
21	Helmut-Egbert			
22	Helmut-Helmut			
23	Egbert-Sara			
24	Egbert-Helmut			
25	Egbert-Egbert			

Abbildung 11.156: Mögliche Variationen mit und ohne »Zurücklegen«

Laut der Berechnung mithilfe der Funktion VARIATIONEN2() gibt es

- ▶ 9 verschiedene Möglichkeiten, 2er-Kombinationen zu bilden
- ▶ 27 verschiedene Möglichkeiten, 3er-Kombinationen zu bilden

Siehe auch

BINOM.VERT(), BINOM.INV(), FAKULTÄT(), HYPGEOM.VERT(), KOMBINATIONEN(), NEGBINOM.VERT(), VARIATIONEN2()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Variationen2*.

WAHRSCHBEREICH()


PROB()

WAHRSCHBEREICH(*Beob_Werte*; *Beob_Wahrsch*; *Untergrenze*; *Obergrenze*)

Syntax

Die Funktion WAHRSCHBEREICH() gibt die Wahrscheinlichkeit für ein von zwei Werten eingeschlossenes Intervall zurück. Ist das Argument *Obergrenze* nicht angegeben, berechnet diese Funktion die Wahrscheinlichkeit, dass zu *Beob_Werte* gehörende Werte gleich dem Wert von *Untergrenze* sind.

Definition

Beob_Werte (erforderlich) ist ein Bereich von Realisationen der Zufallsvariablen, denen Wahrscheinlichkeiten zugeordnet sind.

Argumente

Beob_Wahrsch (erforderlich) sind die Wahrscheinlichkeiten zu den beobachteten Werten.

Untergrenze (erforderlich) ist die untere Grenze der Werte, deren Wahrscheinlichkeit berechnet werden soll.

Obergrenze (optional) ist die obere Grenze der Werte, deren Wahrscheinlichkeit berechnet werden soll.

Ist eine der zu *Beob_Wahrsch* gehörenden Zahlen kleiner oder gleich 0 bzw. größer 1, gibt die Funktion WAHRSCHBEREICH() den Fehlerwert #ZAH! zurück.

Hinweis

Ist die Summe der zu *Beob_Wahrsch* gehörenden Zahlen größer 1, gibt WAHRSCHBEREICH() den Fehlerwert #ZAH! zurück.

Fehlt das Argument *Obergrenze*, gibt WAHRSCHBEREICH() die Wahrscheinlichkeit für den Wert *Untergrenze* zurück.

Enthalten *Beob_Werte* und *Beob_Wahrsch* unterschiedlich viele Datenpunkte, gibt die Funktion WAHRSCHBEREICH() den Fehlerwert #NV zurück.

Mithilfe der Funktion WAHRSCHBEREICH() können Sie die Wahrscheinlichkeit für ein Intervall von Werten berechnen, deren Einzelwahrscheinlichkeiten jeweils angegeben sind. Anders ausgedrückt summiert die Funktion die jeweiligen Wahrscheinlichkeiten der Werte in diesem Intervall.

Hintergrund

Die Werte entsprechen dem Argument *Beob_Werte*, die zu untersuchen sind, die Wahrscheinlichkeiten bezeichnen das Argument *Beob_Wahrsch*, also die zu den Werten gehörenden Einzelwahrscheinlichkeiten.

Alle Wahrscheinlichkeiten müssen größer oder gleich 0 (0 %) und kleiner oder gleich 1 (100 %) sein und deren Summe muss 1 ergeben.

Sie sind Arzt und bieten Ihren Patienten einen kostenlosen Gesundheitscheck an. Dazu gehört auch der Gang auf die Waage.

Praxiseinsatz

Fünf Ihrer Patienten haben sich bereits wiegen lassen. Aus anderen, früheren Messungen konnten Sie den aktuell gemessenen Werten Wahrscheinlichkeiten zuordnen. Die Werte und deren Wahrscheinlichkeiten haben Sie in einer Tabelle festgehalten (siehe Abbildung 11.157).

	C	D	E
9	Person	Gewicht in kg	Wahrscheinlichkeit
10	1	30	10%
11	2	40	15%
12	3	50	20%
13	4	60	25%
14	5	70	30%

Abbildung 11.157: Die Messwerte sowie die dazugehörigen Wahrscheinlichkeiten

Patient Nr. 6 betritt das Behandlungszimmer. Sie stellen sich folgende Fragen:

1. Wie hoch ist die Wahrscheinlichkeit, dass Patient 6 gewichtsmäßig genau dem Wert 4 entspricht, also 60 kg wiegt?
2. Wie hoch ist die Wahrscheinlichkeit, dass Patient 6 gewichtsmäßig zwischen Wert 4 und 5 liegt, also zwischen 60 und 70 kg wiegt?

Die Antworten liefert die Funktion WAHRSCBEREICH(), dargestellt in Abbildung 11.158.

D16 fx =WAHRSCBEREICH(D10:D14;E10:E14;D13;D14)				
	B	C	D	E
9		Person	Gewicht in kg	Wahrscheinlichkeit
10		1	30	10%
11		2	40	15%
12		3	50	20%
13		4	60	25%
14		5	70	30%
15	Nur Untergrenze	Wahrsbereich 1	0,25	25%
16	Untergrenze/ Obergrenze	Wahrsbereich 2	0,55	55%

Abbildung 11.158: Die Berechnung der Wahrscheinlichkeiten mithilfe der Funktion WAHRSCBEREICH()

Die Antworten auf die Fragen lauten demnach:

1. Die Wahrscheinlichkeit, dass Patient 6 genau 60 kg wiegt, beträgt 25 %.
2. Die Wahrscheinlichkeit, dass Patient 6 zwischen 60 und 70 kg wiegt, beträgt 55 %.

Wie Sie der Abbildung 11.158 entnehmen können, wurde zur Beantwortung von Frage 1 das optionale Argument *Obergrenze* nicht angegeben. Das heißt, WAHRSCBEREICH() hat die Wahrscheinlichkeit für die Untergrenze, in unserem Beispiel 60 kg, berechnet.

Um für Frage 2 eine Antwort zu erhalten, mussten sowohl das Argument *Untergrenze* als auch das Argument *Obergrenze* angegeben werden. Das heißt, es wurde die Wahrscheinlichkeit für einen Wert x innerhalb eines Intervalls, in unserem Beispiel das Intervall zwischen 60 und 70 kg, berechnet.

Siehe auch BINOM.VERT(), BINOM.VERT.BEREICH(), BINOM.INV()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Wahrsbereich*.

WEIBULL.VERT() / WEIBULL()

WEIBULL.DIST() / WEIBULL()

WEIBULL.VERT(*x;Alpha;Beta;kumuliert*)

Die Funktion WEIBULL.VERT() gibt Wahrscheinlichkeiten einer weibullverteilten Zufallsvariablen zurück. Diese Verteilung können Sie bei Zuverlässigkeitsanalysen verwenden, also beispielsweise dazu, die mittlere Lebensdauer eines Geräts zu berechnen.

x (erforderlich) ist der Wert, für den die Funktion ausgewertet werden soll.

Alpha (erforderlich) ist ein Parameter der Verteilung.

Beta (erforderlich) ist ein Parameter der Verteilung.

kumuliert (erforderlich) ist der Wahrheitswert, der den Typ der Funktion bestimmt. Ist *kumuliert* mit *WAHR* belegt, gibt WEIBULL.VERT() den Wert der Verteilungsfunktion zurück, also die Wahrscheinlichkeit, dass die Anzahl zufällig auftretender Ereignisse zwischen 0 und einschließlich *x* liegt. Ist *kumuliert* mit *FALSCH* belegt, gibt WEIBULL.VERT() den Wert der Dichtefunktion zurück.

Ist eines der Argumente *x*, *Alpha* oder *Beta*, kein numerischer Ausdruck, gibt WEIBULL.VERT() den Fehlerwert #WERT! zurück.

Ist *x* kleiner 0, gibt WEIBULL.VERT() den Fehlerwert #ZAHL! zurück.

Ist *Alpha* kleiner oder gleich 0, oder ist *Beta* kleiner oder gleich 0, gibt WEIBULL.VERT() den Fehlerwert #ZAHL! zurück.

Wie viele andere in diesem Buch bereits erwähnte Verteilungen, gehört auch die Weibull-Verteilung zu den statistischen Verteilungen, die beispielsweise zur Untersuchung der Lebensdauer im Bereich der Qualitätssicherung verwendet wird. Hier sind Fragestellungen bezüglich der Materialermüdung von spröden Werkstoffen oder auch Ausfällen von elektronischen Bauteilen relevant.

Benannt ist die Weibull-Verteilung nach dem Schweden Waloddi Weibull (1887 bis 1979).

Die Weibull-Analyse gilt als klassische Zuverlässigkeitsanalyse oder das klassische Lebensdauerdiagramm. Sie hat besonders in der Automobilindustrie eine große Bedeutung. Aus dem sogenannten »Weibull-Netz« lassen sich die charakteristische Lebensdauer, sowie eine »Ausfallwahrscheinlichkeit« von Bauteilen oder anderen Komponenten ablesen.

Verallgemeinert handelt es sich bei der Weibull-Verteilung um eine Exponentialverteilung. Mit der Weibull-Verteilung wird aus verschiedenen Gründen gearbeitet:

- ▶ Sehr viele Verteilungsformen lassen sich mit der Weibull-Verteilung darstellen
- ▶ Die Weibull-Funktion ist mathematisch leicht zu handhaben
- ▶ Zeitabhängige Ausfallmechanismen erscheinen als Gerade

Mehr Informationen zur Exponentialverteilung finden Sie auf Seite 376 unter der Funktion EXPON.VERT().

Syntax

Definition

Argumente

Hinweis

Hintergrund

Hinweis

Sie können mithilfe der Funktion WEIBULL.VERT() sowohl die Dichte- als auch die Verteilungsfunktion der Weibull-Verteilung berechnen.

Zur Berechnung der Dichtefunktion verwenden Sie für das Argument *kumuliert* den Wahrheitswert *FALSCH*. Die Dichtefunktion stellt die erste Ableitung der Verteilungsfunktion nach der Zufallsvariablen (z.B. Zeit bei der Ausfalldichtefunktion) dar. Anders ausgedrückt wird die differentielle Änderung der relativen Häufigkeit pro Skalenabschnitt errechnet.

Zur Berechnung der Verteilungsfunktion verwenden Sie für das Argument *kumuliert* den Wahrheitswert *WAHR*. Die Verteilungsfunktion $F(x)$ gibt die relative Häufigkeit von Ereignissen in kumulierter Form an.

Im eindimensionalen Fall bedeutet das, wie bei einigen anderen Verteilungsfunktionen auch, das $F(x)$ die Wahrscheinlichkeit angibt, dass die Zufallsvariable y höchstens den Wert x annimmt.

Die Gleichung der Verteilungsfunktion einer Weibull-Verteilung lautet wie folgt:

$$F(x, \alpha, \beta) = 1 - e^{-(x/\beta)^\alpha}$$

Die Gleichung der Dichtefunktion einer Weibull-Verteilung lautet wie folgt:

$$f(x, \alpha, \beta) = \frac{\alpha}{\beta^\alpha} x^{\alpha-1} e^{-(x/\beta)^\alpha}$$

Ist *Alpha* = 1, gibt die Funktion WEIBULL.VERT() die Exponentialverteilung mit Folgendem zurück:

$$\lambda = \frac{1}{\beta}$$

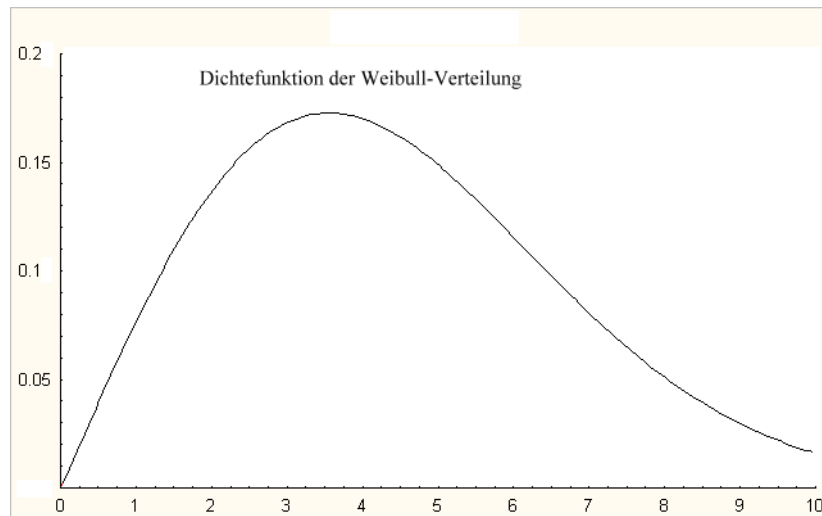


Abbildung 11.159: Die grafische Darstellung der Weibull-Verteilung mit den Argumenten x , $Alpha=5$, $Beta=2$, $kumuliert = Falsch$

Das Verhalten der Ausfallrate hängt demnach vom Parameter β ab:

- ▶ $1 > \beta$ monoton wachsend
- ▶ $1 = \beta$ konstant und somit eine Exponentialverteilung
- ▶ $1 < \beta$ monoton fallend

Gehen Sie folgendermaßen zur Berechnung von WEIBULL.VERT() vor:

Gegeben sind die Werte:

- ▶ 105 = Wert, für den die Funktion ausgewertet werden soll (x)
- ▶ 20 = Alpha-Parameter der Verteilung (*Alpha*)
- ▶ 100 = Beta-Parameter der Verteilung (*Beta*)
- ▶ WAHR = *kumuliert*
- ▶ FALSCH = *kumuliert*

Die Berechnung von WEIBULL.VERT() sehen Sie in Abbildung 11.160.

Praxiseinsatz

E15		fx =WEIBULL.VERT(E10;E11;E12;E13)			
	B	C	D	E	F
9	Bedeutung			Parameter	
10	Wert, für den die Funktion ausgewertet werden soll			105	
11	Alpha-Parameter der Verteilung			20	
12	Beta-Parameter der Verteilung			100	
13	Kumuliert			WAHR	
14	Kumuliert			FALSCH	
15	WEIBULL.VERT() für Kumuliert = WAHR			0,929581	
16	WEIBULL.VERT() für Kumuliert = FALSCH			0,035589	

Abbildung 11.160: Berechnung von WEIBULL.VERT()

Unter Angabe der in Abbildung 11.160 dargestellten Parameter gibt die Funktion WEIBULL.VERT() folgende Ergebnisse zurück:

- ▶ Für *kumuliert* = WAHR liefert WEIBULL.VERT() die Verteilungsfunktion einer Weibull-Verteilung mit dem Wert 0,929581
- ▶ Für *kumuliert* = FALSCH liefert WEIBULL.VERT() die Dichtefunktion einer Weibull-Verteilung mit dem Wert 0,035589

EXPON.VERT()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Wahrscheinlichkeit.xlsx* auf dem Arbeitsblatt *Weibull.vert*.

Siehe auch



ZÄHLENWENN()



Syntax ZÄHLENWENN(*Bereich*; *Kriterien*)

Definition ZÄHLENWENN() zählt die nicht leeren Zellen eines Bereichs, deren Inhalte mit den Suchkriterien übereinstimmen.

Argumente *Bereich* (erforderlich) ist der Zellbereich, von dem Sie wissen möchten, wie viele seiner Zellen einen Inhalt haben, der mit den Suchkriterien übereinstimmt.

Kriterien (erforderlich) gibt die Kriterien in Form einer Zahl, eines Ausdrucks oder einer Zeichenfolge an. Diese Kriterien bestimmen, welche Zellen gezählt werden. Ein Suchkriterium kann als Zahl (z.B. 2000), als Text, z.B. *Kein*, in Bezug auf nicht erzielte Umsätze, oder als Zellbezug (z.B. *B5*) formuliert werden.

Hintergrund Mit dieser Funktion haben Sie eine weitere Möglichkeit, Ihre Daten auf Grundlage einer Bedingung zu analysieren. Warum diese Funktion allerdings unter den statistischen Funktionen und nicht, wie die SUMMEWENN()-Funktion, unter den mathematischen & trigonometrischen Funktionen zu finden ist, lässt sich nicht genau sagen.

Die Funktion ZÄHLENWENN() kann als Suchkriterium nur mit einem konkreten Wert arbeiten. Der Versuch, alle Zellen mit Zahlen kleiner 0 zu suchen, scheitert erst mal. Sie können die Zeichen < und > nur verwenden, wenn Sie die Zeichen sowie den anschließenden Vergleichswert in Anführungsstriche einfassen. Alternativ können Sie < oder > in Anführungsstriche setzen und den darauffolgenden Wert mit einem & anschließen.

Praxiseinsatz 1 Kommen wir wieder auf die Tabelle des Softwareherstellers mit den Umsätzen der letzten zwei Jahre zurück. Unternehmensziel war es, den Umsatz über 200.000 € pro Monat zu bringen – nun steht die Prüfung an, um zu sehen, wie oft dieses Ziel in den vergangenen 24 Monaten erreicht wurde.

Mithilfe der Funktion ZÄHLENWENN() soll das Ergebnis errechnet werden. Für den *Bereich* werden die Zellen C3:C26, für *Kriterien* > 200000 angegeben. Wie in Abbildung 11.161 dargestellt, ergibt sich ein Ergebnis von 15 Umsätzen, die über 200.000 € liegen.

Tipp Obwohl ZÄHLENWENN() eigentlich nur auf die Verwendung mit einer Bedingung ausgerichtet ist, können Sie mit einem Trick trotzdem mehrere Bedingungen angeben, um Ihre Datenmenge zu untersuchen. Möchten Sie mit zwei Bedingungen arbeiten, erstellen Sie für jede der Bedingungen eine ZÄHLENWENN()-Funktion und verknüpfen diese mit einem »+«-Zeichen. Auf unser Beispiel bezogen könnten Sie z.B. nach Werten innerhalb der Tabelle suchen, die zwischen 180.000 € und 200.000 € liegen.

In diesem Fall ist die erste Bedingung größer 180.000 € und die zweite kleiner 200.000 €.

Für jede der beiden Bedingungen erstellen Sie eine ZÄHLENWENN()-Funktion, verknüpfen diese mit einem «+» und setzen diesen Teil in Klammern. Danach ziehen Sie die Gesamtanzahl aller Werte in diesem Bereich vom Ergebnis ab, und erhalten damit die Anzahl der Zahlen, die größer als 180.000 und kleiner als 200.000 sind.

E3 fx =ZÄHLENWENN(C3:C26;">200000")				
A	B	C	D	E
1	Fragestellung: Wie oft war der Umsatz größer als 200000€?			
2	Monat	Umsatz		Ergebnis
3	Januar 07	107.629 €		15
4	Februar 07	185.385 €		
5	März 07	180.807 €		
6	April 07	124.328 €		
7	Mai 07	146.215 €		
8	Juni 07	185.675 €		
9	Juli 07	210.169 €		
10	August 07	221.729 €		
11	September 07	234.187 €		
12	Oktober 07	237.947 €		
13	November 07	210.088 €		
14	Dezember 07	207.791 €		
15	Januar 08	201.097 €		
16	Februar 08	222.460 €		
17	März 08	208.585 €		
18	April 08	206.387 €		
19	Mai 08	218.951 €		
20	Juni 08	95.642 €		
21	Juli 08	126.895 €		
22	August 08	654.988 €		
23	September 08	126.874 €		
24	Oktober 08	524.985 €		
25	November 08	346.852 €		
26	Dezember 08	236.985 €		

Abbildung 11.161: Wie viele Zellen der Tabelle stimmen mit dem Suchkriterium »>200000« überein

ZÄHLENWENN() ist auch mit Platzhalterzeichen möglich.

Praxiseinsatz 2

Der Personalleiter des Softwareherstellers hat eine Liste angelegt, in der die Mitarbeiter eintragen sollen, wann sie Urlaub und wann Gleitzeit beantragt haben. Nun möchten Sie alle Gleitzeittage zählen, doch die Angestellten haben in die Liste sowohl »Gleitzeit« als auch »Gleit«, »Gl« oder nur »G« eingetragen.

Da die ZÄHLENWENN()-Funktion als Suchkriterium auch Platzhalter zulässt, kann der Personalleiter auch hier auf sie zurückgreifen.

E37 fx =ZÄHLENWENN(C37:D43;"G**")					
A	B	C	D	E	F
33	Fragestellung:				
34	Wie viele Gleitzeittage wurden genommen?				
35					
36	Mitarbeiter	1. Jan. 08	2. Jan. 08	Ergebnis	
37	Reinke	Gl	U	6	
38	Jeschke	Gl	Gleitzeit	=ZÄHLENWENN(C37:D43;"G**")	
39	Unverhau	U	Urlaub		
40	Müller	Ur	U		
41	Schmidt	Url	Gleit		
42	Fischer	Gleitzeit	Url		
43	Kammer	Urlaub	Gleit		

Abbildung 11.162: Nutzen Sie als Suchkriterium einen Platzhalter

Wie Sie Abbildung 11.162 beispielhaft sehen können, liefert der Platzhalter »*« in Verbindung mit »G« alle Werte der Tabelle, die mit dem Buchstaben »G« beginnen – in unserem Fall handelt es sich um sechs Einträge und somit um sechs Gleitzeitpunkte, die von den verschiedenen Mitarbeitern in zwei Tagen beantragt wurden.

Siehe auch ANZAHLLEEREZELLEN(), SUMMEWENN(), ZÄHLENWENNS()



Diese Beispiele und das Beispiel aus dem Tipp finden Sie im Ordner \Ms5-235\Kap11 in der Arbeitsmappe *Zählen.xlsx* auf dem Arbeitsblatt *Zählenwenn*.

Neu in Excel
2007

ZÄHLENWENNS() COUNTIFS()

Syntax ZÄHLENWENNS(*Bereich1*; *Kriterien1*; *Bereich2*; *Kriterien2*...)

Hinweis ZÄHLENWENNS() ist eine neue Funktion in Excel 2007!

Definition ZÄHLENWENNS() zählt die Anzahl der Zellen eines Bereichs, die mehreren Kriterien entsprechen.

Argumente *Bereich1* (erforderlich) ist der Bereich, in dem die entsprechenden Kriterien ausgewertet werden sollen. Bei den Zellen des Bereichs muss es sich um Namen von Zellen, Matrizen von Zellen oder Bezüge zu Zellen, die Zahlen enthalten handeln. Leere Zellen und Textwerte werden ignoriert.

Kriterien1 (erforderlich) ist das Kriterium in Form einer Zahl, eines Ausdrucks, Zellbezugs oder Texts, mit dem definiert wird, welche Zellen gezählt werden.

Bereich2... (optional) sind Bereiche von 1 bis 127, in denen die entsprechenden Kriterien ausgewertet werden sollen. Bei den Zellen in den einzelnen Bereichen muss es sich um Namen von Zellen, Matrizen von Zellen oder Bezüge zu Zellen handeln, die Zahlen enthalten. Leere Zellen und Textwerte werden ignoriert.

Kriterien2... (optional) sind 1 bis 127 Kriterien in Form einer Zahl, eines Ausdrucks, Zellbezugs oder Texts, mit denen definiert wird, welche Zellen gezählt werden.

Hinweis Die einzelnen Zellen eines Bereichs werden nur gezählt, wenn alle entsprechenden angegebenen Kriterien für diese Zelle zutreffen.

Wenn Kriterien eine leere Zelle darstellt, behandelt ZÄHLENWENNS() sie als Nullwert.

Die Kriterien dürfen die Platzhalterzeichen »Fragezeichen (?)« und »Sternchen (*)« enthalten. Ein Fragezeichen ersetzt bei Übereinstimmung ein einzelnes Zeichen und ein Sternchen eine beliebige Zeichenfolge. Wenn Sie nach einem tatsächlichen Fragezeichen oder Sternchen suchen möchten, geben Sie eine »Tilde (~)« vor dem zu suchenden Zeichen ein.

Hintergrund Die Funktion ZÄHLENWENNS() ist die Erweiterung der Funktion ZÄHLENWENN. Vorteil der Funktion ZÄHLENWENNS() ist, dass sie die Anzahl der Zellen eines Zellbereichs unter Angabe mehrerer Bedingungen und Kriterien ermitteln kann.

Mehr Informationen zur Funktion ZÄHLENWENN(), als Basis der Funktion ZÄHLENWENNS(), erfahren Sie auf Seite 564.

Kommen wir auf die Tabelle des Software-Herstellers mit den Umsätzen der letzten zwei Jahre zurück. **Praxiseinsatz**

	B	C	D
2	Monat	Jahr	Umsatz
3	Januar	2007	107.629 €
4	Februar	2007	185.385 €
5	März	2007	180.807 €
6	April	2007	124.328 €
7	Mai	2007	146.215 €
8	Juni	2007	185.675 €
9	Juli	2007	210.169 €
10	August	2007	221.729 €
11	September	2007	234.187 €
12	Oktober	2007	237.947 €
13	November	2007	210.088 €
14	Dezember	2007	207.791 €
15	Januar	2008	201.097 €
16	Februar	2008	222.460 €
17	März	2008	208.585 €
18	April	2008	206.387 €
19	Mai	2008	218.951 €
20	Juni	2008	95.642 €
21	Juli	2008	126.895 €
22	August	2008	654.988 €
23	September	2008	126.874 €
24	Oktober	2008	524.985 €
25	November	2008	346.852 €
26	Dezember	2008	236.985 €

Abbildung 11.163: Auszug aus der Umsatztabelle für die beiden Jahre 2007 und 2008

Unternehmensziel war es, den Umsatz über 200.000 € pro Monat zu bringen – nun steht die Prüfung an, um zu sehen, wie oft dieses Ziel in den vergangenen 24 Monaten erreicht wurde. Für den gesamten Zeitraum können Sie dies mit der Funktion ZÄHLENWENN() berechnen.

Das reicht Ihnen jedoch nicht – Sie möchten die beiden Jahre miteinander vergleichen. Suchen also die Anzahl der Monate, in denen ein Umsatz größer als 200.000 € erwirtschaftet wurde und das jeweils für das Jahr 2007 bzw. 2008.

Hierfür nutzen Sie die Funktion ZÄHLENWENN(), da es Ihnen so möglich ist, mehrere Kriterien gleichzeitig anzugeben. In unserem Fall die Kriterien ">200.000" und "2007" bzw. "2008". Die Abbildung 11.164 zeigt das Ergebnis.

	B	C	D	E	F
2	Monat	Jahr	Umsatz		Ergebnis 2007
3	Januar	2007	107.629 €		6
4	Februar	2007	185.385 €		
5	März	2007	180.807 €		
6	April	2007	124.328 €		
7	Mai	2007	146.215 €		
8	Juni	2007	185.675 €		
9	Juli	2007	210.169 €		

Abbildung 11.164: Berechnung der Anzahl der Umsätze höher als 200.000 und getrennt nach Jahr

Wie Sie der Abbildung entnehmen können, ist die Anzahl der Monate, deren Umsatz über 200.000 € lag, im Jahr 2008 um drei gestiegen. Damit hat der Software-Hersteller wohl alles richtig gemacht!

Siehe auch SUMMEWENN(), ZÄHLENWENN(),



Diese Beispiele zur Funktion und das Beispiel aus dem Tipp finden Sie im Ordner `Ms5-235\Kap11` in der Arbeitsmappe *Zählen.xlsx* (Excel 2007/2010) auf dem Arbeitsblatt *Zählenwenns*.

Kapitel 12

Datenbankfunktionen

Argumente in Datenbankfunktionen	569
Arbeiten mit Datenbanken und -sätzen	570
Verwendung von Steuerelementen	572
Die Datenbankfunktionen im Überblick	572
Darstellung der Funktionen in diesem Kapitel	573
DBANZAHL()	573
DBANZAHL2()	575
DBAUSZUG()	577
DBMAX()	578
DBMIN()	580
DBMITTELWERT()	581
DBPRODUKT()	582
DBSTDABW()	584
DBSTDABWN()	586
DBSUMME()	587
DBVARIANZ()	590
DBVARIANZEN()	591
PIVOTDATENZUORDNEN()	593

In diesem Kapitel des Buchs über die Excel-Funktionen wollen wir Ihnen die Datenbankfunktionen näher bringen. Da Datenbanken mittlerweile hauptsächlich auf Tabellen beruhen, bietet sich Excel als Tabellenkalkulationsprogramm und dem darin integrieren Datenbankmodul geradezu an – das heißt, die Excel-Datenbankfunktionen finden immer dann Anwendung, wenn es darum geht, Auswertungen aus einer bereits bestehenden Datenbank zu generieren.

Eine Datenbank in Excel bezeichnet eine Tabelle mit Überschriften (die in der Terminologie der Datenverwaltung *Feldnamen* genannt werden) und einer bestimmten Anzahl von Zeilen. Entsprechend des Feldnamens befinden sich darunter gleichartige Informationen wie beispielsweise Umsätze oder Namen. Diese Informationen innerhalb der einzelnen Spalten werden *Feldinhalte* genannt.

	A	B	C	D	E	F
1	Firma	Ort	Datum	Land	Artikelname	Umsatz
2	Alfreds Futterkiste	Berlin	12.03.08	Deutschland	Chai	kein
3	Alfreds Futterkiste	Berlin	11.01.08	Deutschland	Chang	1.449,00 €
4	Alfreds Futterkiste	Berlin	01.08.08	Deutschland	Aniseed Syrup	1.530,00 €
5	Alfreds Futterkiste	Berlin	18.03.08	Deutschland	Chef Anton's Cajun Seasoning	1.662,00 €
6	Alfreds Futterkiste	Berlin	30.06.08	Deutschland	Chef Anton's Gumbo Mix	1.633,00 €
7	Alfreds Futterkiste	Berlin	02.08.07	Deutschland	Grandma's Boysenberry Spread	1.959,00 €
8	Alfreds Futterkiste	Berlin	24.03.08	Deutschland	Uncle Bob's Organic Dried Pears	1.382,00 €
9	Alfreds Futterkiste	Berlin	08.02.08	Deutschland	Northwoods Cranberry Sauce	1.940,00 €
10	Alfreds Futterkiste	Berlin	28.11.07	Deutschland	Mishi Kobe Niku	1.296,00 €
11	Alfreds Futterkiste	Berlin	21.06.08	Deutschland	Ikura	1.588,00 €
12	Alfreds Futterkiste	Berlin	10.09.07	Deutschland	Queso Cabrales	1.207,00 €
13	Alfreds Futterkiste	Berlin	06.12.07	Deutschland	Queso Manchego La Pastora	1.061,00 €
14	Alfreds Futterkiste	Berlin	18.12.07	Deutschland	Konbu	1.880,00 €

Abbildung 12.1: Der Datenbankbereich besteht aus Feldnamen und Feldinhalten

Eine Tabelle wie in Abbildung 12.1 erkennt Excel intern als Datenbank, das heißt, der Aufbau der Datenbank spielt eine wichtige Rolle. So ist es zum Beispiel wichtig, dass sich Feldnamen nicht wiederholen, die Überschriften in der obersten Reihe stehen und im gesamten Datenbankbereich keine Leerzeilen auftreten.

Ebenso erkennt Excel eine solche Liste automatisch als Datenbank, wenn für den Datenbankbereich typische Aufgaben oder Abfragen durchgeführt werden. Hierzu gehören z.B. die Such- und Sortierfunktion oder das Berechnen von Teilergebnissen. Bei solchen Datenbankaufgaben werden die Spalten als *Datenbankfelder* und die Zeilen als *Datensätze* bezeichnet.

Die Excel-Datenbankfunktionen sind zwar eher weniger bekannte Funktionen, bieten jedoch verschiedenste und sehr effiziente Möglichkeiten, um Tabellen und Listen bzw. deren Dateninhalte nach vorgegebenen Kriterien auszuwerten.

Hinweis Die DB-Funktionen sind ausschließlich auf Excel-Datenbanken anwendbar.

Zusätzlich können Sie beispielsweise PivotTables erstellen oder verschiedene Filter auf die Daten anwenden, um weitere Ergebnisse aus Analysen zu generieren. Ebenso stellt Excel einige Datenbankfunktionen zur Verfügung, mit denen Suchkriterien bei der Auswertung von Bereichen berücksichtigt werden können.

Vorteil ab 2007 Die Verwendung von Excel als Datenbank stieß in den Versionen bis 2003 relativ schnell an ihre Grenzen – eine Tabelle konnte maximal 65.536 Zeilen umfassen. Diese Grenze hat sich ab der Version 2007 nach oben verändert: Sie können hier bis zu 1.048.576 Zeilen je Arbeitsblatt für Ihre Datenlisten nutzen.

Argumente in Datenbankfunktionen

Da die Argumente für die Datenbankfunktionen – mit Ausnahme der Funktion PIVOTDATENZUORDNEN() – immer gleich sind, möchten wir gleich zu Beginn dieses Kapitels näher darauf eingehen. Vor allem interessiert uns in diesem Bereich das Argument *Kriterien*.

Bei den Datenbankfunktionen sind drei Argumente anzugeben:

1. Datenbank

Mit dem erforderlichen Argument *Datenbank* wird der Zellbereich angegeben, den die Liste oder die Datenbank umfasst. In unserem Beispiel wäre das also der Bereich *A1:F7008* mit dem dynamischen Namen *Datenbank*.

2. Feld

Das Argument *Feld* gibt an, welche Spalte in der Funktion verwendet wird. Ein Feld kann als Text zusammen mit der Spaltenbeschriftung aufgeführt werden, und zwar in doppelten Anführungszeichen wie z.B. "Land", "Umsatz" oder "Firma". Alternativ kann ein Feld aber auch als Zahl angegeben werden, die die Position der Spalte innerhalb der Liste bezeichnet wie z.B. 1 für die erste Spalte oder 2 für die zweite Spalte.

Wir empfehlen Ihnen, in Bezug auf das Argument *Feld* mit Namen zu arbeiten. Dies macht die Funktion verständlicher und auch lesbarer.

Tip

3. Kriterien

Das erforderliche Argument *Kriterien* bezeichnet den Zellbereich, der die Feldnamen und darunter die Filterkriterien enthält. Sie können für dieses Argument *Suchkriterien* jeden Bereichs verwenden. Hierbei muss jedoch beachtet werden, dass er mindestens eine Spaltenbeschriftung und mindestens eine Zelle unter der Spaltenbeschriftung umfasst, damit eine Bedingung für die Spalte angegeben werden kann.

Leider ist es nicht möglich, die Kriterien als Matrixausdruck (Matrixkonstante) direkt in die Funktion einzugeben. Sie kommen also nicht umhin, einen Kriterienbereich auf dem Arbeitsblatt anzulegen, um diesen dann als Bereichsbezug in der DB-Funktion zu verwenden.

Hinweis

Im Folgenden möchten wir Ihnen die Suchkriterien und deren Funktionsweise verdeutlichen.

Die Excel-Datenbank für dieses Beispiel finden Sie im Ordner *\Ms5-235\Kap12* in der Arbeitsmappe *DBFUNKTION leer.xlsx* auf dem Arbeitsblatt *Urdaten*.



Die Bedingungen zur Auswertung der Daten sollten der Einfachheit und Übersicht halber ebenfalls in das entsprechende Arbeitsblatt eingetragen werden.

Nehmen wir an, Sie möchten die Kategorien *Land*, *Artikelname* und *Umsatz* näher betrachten:

1. Kopieren Sie aus dem Arbeitsblatt *Urdaten* die entsprechenden Spaltenüberschriften (Feldnamen) und fügen Sie diese rechts neben der Tabelle wieder ein.
2. Tragen Sie nun direkt unter die Kopie der Feldnamen eine Bedingung ein. Als Bedingung können sowohl Texte und Zahlen wie z.B. »Deutschland«, »1500« oder »D*«, als auch Kombinationen aus logischen Operatoren und Begriffen wie z.B. »>K« oder »<2000« verwendet werden.
3. Den Bereich mit den Feldnamen und den darunter befindlichen Filterbedingungen (Kriterienbereich) können Sie nun in den DB-Funktionen verwenden.

Hinweis Der Stern bei »D*« steht in den Suchkriterien als Platzhalter für beliebige, beliebig viele Zeichen, das Fragezeichen für ein einzelnes beliebiges Zeichen.

Die Eingabe der Suchkriterien kann nun wie in Abbildung 12.2 dargestellt aussehen.

G	H	I
Land	Artikelname	Umsatz
Mexiko	Geitost	>1500

Abbildung 12.2: Die Suchkriterien für weitere Datenbankauswertungen als *UND*-Bedingung

Wenn Sie in mehrere Spalten einer Zeile eine Bedingung eintragen (Abbildung 12.2), werden nur die Daten der Datenbank zusammengefasst, die den Bedingungen **aller** Spalten entsprechen. Für die Bedingungen gilt das logische »UND« – man spricht daher auch von *UND-Bedingungen*.

Möchten Sie beispielsweise die Summe der Einträge in der Datenbank erhalten, die sowohl das Kriterium »Mexiko« als auch »Geitost« und »>1500« erfüllen, wäre die in Abbildung 12.2 dargestellte Schreibweise der Suchkriterien – also in einer Zeile – die richtige.

Neben der *UND-Bedingung* können Sie aber auch mit *ODER-Bedingungen* arbeiten. Bei *ODER-Bedingungen* werden die Suchkriterien auf mehrere Zeilen verteilt, wie in Abbildung 12.3 dargestellt.

G	H	I
Land	Artikelname	Umsatz
Mexiko	Geitost	
		>1500

Abbildung 12.3: Die Suchkriterien für weitere Datenbankauswertungen als *ODER*-Bedingung


In diesem Beispiel würde die Summe derer Datensätze innerhalb der Datenbank gebildet, die die Kriterien »Mexiko« UND »Geitost« enthalten ODER »>1500«.

Hinweis Würden Sie als Suchkriterium »Cha« angeben, würden sowohl Datensätze mit dem Artikelnamen »Chai« als auch »Chang« in die Berechnung eingehen. Für eine exakte Übereinstimmung tragen Sie den Wert in Anführungszeichen und einem Gleichheitszeichen ein. Auf diese Weise findet »="Chai"« nur die Artikel, die exakt diesem Eintrag entsprechen.

Arbeiten mit Datenbanken und -sätzen

Vorsicht mit Leerzeichen Bei der Groß- und Kleinschreibung zeigt sich Excel sehr tolerant. Anders ist der Fall aber mit Leerzeichen, die versehentlich am Ende eines Eintrags stehen. Wenn Sie beim Filtern keine Datensätze finden, liegt das vielleicht an solch einem versehentlich eingetragenen Leerzeichen. Dieses Leerzeichen kann in einem Datensatz oder in den Suchkriterien zu Problemen führen. Wenn Sie sicher sind, dass Daten mit den entsprechenden Suchkriterien vorhanden sind, aber trotzdem keine Daten ausgewertet werden, versuchen Sie es mit einem angehängten Fragezeichen oder dem Stern als Platzhalter bzw. sorgen Sie dafür, dass Ihre Datenbestände bereinigt werden. Dazu stehen Ihnen die in Kapitel 7 vorgestellten Textfunktionen GLÄTTEN() oder SÄUBERN() zur Verfügung.

Wenn Sie bei den Berechnungen alle Datensätze berücksichtigen wollen, die in einem Feld einen Eintrag haben, verwenden Sie als Suchkriterium für dieses Feld die Zeichenfolge »<>«. Wenn Sie diese Zeichenfolge z.B. für den Nachnamen verwenden, werden alle Datensätze berücksichtigt, bei denen das Feld *Nachname* nicht leer ist.

Wenn Sie alle Datensätze berücksichtigen wollen, bei denen z.B. das Feld *Umsatz* leer ist, tragen Sie für das Suchkriterium ein Gleichheitszeichen » = « ein und drücken die -Taste.

Wenn Sie mehrere Bedingungen für ein Feld festlegen wollen, können Sie dies erreichen, indem Sie einen Feldnamen mehrfach verwenden und hierfür Bedingungen festlegen. Sie können aber auch berechnete Kriterien festlegen.

Für berechnete Kriterien ist es wichtig, dass Sie hier keinen Feldnamen als Überschrift verwenden dürfen. Tragen Sie einen beliebigen anderen Begriff wie z.B. »Kriterien« ein.

Nachfolgend stellen wir Ihnen die Operatoren und Platzhalter im Überblick vor.

Operator	Beispiel	Angezeigt werden als Ergebnis alle Datensätze, deren Werte in der gewählten Spalte ...
> (größer als)	> 5000	... größer als 5.000 sind
< (kleiner als)	< 5000	... kleiner als 5.000 sind
>= (größer als oder gleich)	>= 5000	... 5.000 entsprechen oder größer sind
<= (kleiner als oder gleich)	<= 5000	... 5.000 entsprechen oder kleiner sind
< > (ungleich)	< > 5000	... einen anderen Wert als 5.000 aufweisen
= (gleich)	= 5000	... genau den Wert 5.000 aufweisen Die Eingabe des Gleichheitszeichens bei den Suchkriterien ist in diesem Fall nicht erforderlich

Alle Datensätze mit oder ohne Inhalt berücksichtigt

Berechnete Kriterien

Tabelle 12.1
Übersicht zu Operatoren

Operatoren und Platzhalter im Überblick

Platzhalter	Beispiel	Angezeigt werden als Ergebnis alle Datensätze, die in der gewählten Spalte ...
* (beliebig viele Zeichen)	D*	... mit D beginnen
=* (beliebig viele Zeichen, kombiniert mit einer exakten Zeichenfolge)	"=*ecker"	... mit einem oder mehreren beliebigen Zeichen beginnen und mit <i>ecker</i> enden, z.B. Becker oder Wecker. Meckerheini würde nicht angezeigt, da hier nach <i>ecker</i> weitere Zeichen folgen.
? (einzelnes Zeichen)	?ecker	... mit einem beliebigen Zeichen, gefolgt von <i>ecker</i> , beginnen, z.B. Becker, Wecker und Meckerheini. Hochecker würde nicht angezeigt, da hier vor <i>ecker</i> mehrere Zeichen stehen.
=? (einzelnes Zeichen, kombiniert mit einer exakten Zeichenfolge)	"=?ecker"	... mit einem beliebigen Zeichen beginnen und mit <i>ecker</i> enden, z.B. Becker oder Wecker. Meckerheini und Hochecker würden nicht angezeigt.

Tabelle 12.2
Übersicht zu Platzhaltern

Bei der Verwendung von Platzhaltern in Kombination mit einer exakten Zeichenfolge muss der Suchbegriff in Anführungszeichen gesetzt werden, z.B. "=?ecker", da Excel nach einem Gleichheitszeichen (=) eine Formel erwartet – es sei denn, dem Gleichheitszeichen folgt eine Zeichenfolge in Anführungszeichen. Zusätzlich muss das Ganze noch von einem Gleichheitszeichen eingeleitet werden, da die »Suchformel« sonst nicht als solche erkannt würde und beispielsweise nach der Zeichenfolge "=?ecker" gesucht würde.

Verwendung von Steuerelementen

Unter Steuerelementen versteht man Schaltflächen, Listenfelder, Kontrollkästchen, Optionsfelder usw. Sie sind Bestandteil der integrierten Dialogfelder in Excel. Im Rahmen einer VBA-Programmierung können sie vom Anwender in benutzerdefinierten Dialogfeldern verwendet werden.

Excel verwendet Steuerelemente auch in Arbeitsblättern. Die Funktion »AutoFilter« verwendet beispielsweise Listenfelder zur Auswahl eines Filterkriteriums.

AutoFilter ab Excel 2007

Um den AutoFilter in Excel 2013 zu setzen, gehen Sie wie folgt vor:

1. Öffnen Sie die Registerkarte *DATEN*.
2. Klicken Sie in der Gruppe *Sortieren und Filtern* auf die Schaltfläche *Filtern*.

Wie in Abbildung 12.4 dargestellt, wird die Filterfunktion automatisch eingefügt und steht zum Filtern nach bestimmten Kriterien bereit.

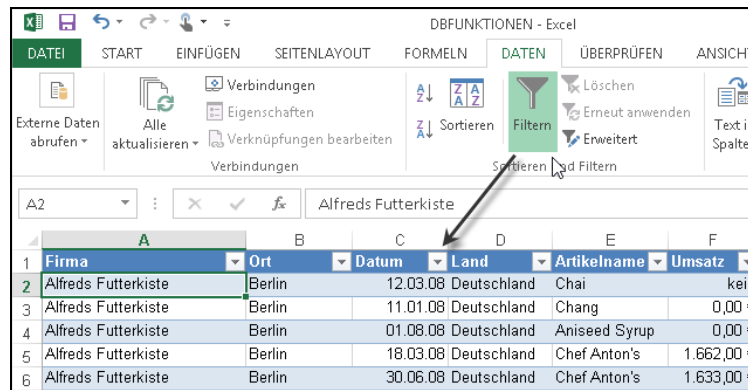


Abbildung 12.4: Aktivierte AutoFilter-Funktion

Die Datenbankfunktionen im Überblick

Nachdem wir nun die Grundlagen erläutert haben, können wir uns an die Datenbankfunktionen, die damit verbundenen Fragestellungen und die Auswertungen der Daten begeben. Welche Funktionen gehören in den Bereich der Datenbanken¹?

Tabelle 12.3
Übersicht über die Datenbankfunktionen

Die Funktion ermittelt ...
DBANZAHL()	... die Anzahl der Felder, die die Bedingung(en) erfüllen
DBANZAHL2()	... die Anzahl der nicht leeren Felder, die die Bedingung(en) erfüllen
DBAUSZUG()	... einen einzelnen Wert, der die Bedingung(en) erfüllt
DBMAX()	... den größten Wert, der die Bedingung(en) erfüllt

¹ Weil sie unmittelbar mit dem Thema »Datenanalyse« zu tun hat, beschreiben wir am Ende des Kapitels zusätzlich die Funktion PIVOTDATENZUORDNEN(), welche Sie noch einmal im Kapitel 9 finden.

Die Funktion ermittelt ...
DBMIN()	... den kleinsten Wert, der die Bedingung(en) erfüllt
DBMITTELWERT()	... den Mittelwert der Werte, die die Bedingung(en) erfüllen
DBPRODUKT()	... das Produkt der Werte, die die Bedingung(en) erfüllen
DBSTDABW()	... geschätzt (aus einer Stichprobe) die Standardabweichung der Werte, die die Bedingung(en) erfüllen
DBSTDABWN()	... exakt die Standardabweichung der Werte, die die Bedingung(en) erfüllen
DBSUMME()	... die Summe der Werte, die die Bedingung(en) erfüllen
DBVARIANZ()	... geschätzt (aus einer Stichprobe) die Varianz der Werte, die die Bedingung(en) erfüllen
DBVARIANZEN()	... exakt die Varianz der Werte, die die Bedingung(en) erfüllen
PIVOTDATENZUODNEN()	... Werte aus einem PivotTable-Bericht

Darstellung der Funktionen in diesem Kapitel

Im Folgenden werden wir uns mit den verschiedenen Datenbankfunktionen beschäftigen. Um die Funktionen gezielt erklären zu können, werden wir uns für alle Funktionen an einem Beispiel orientieren, sodass zum einen die Übersicht gewahrt bleibt und Sie zum anderen die Möglichkeit haben, die Ergebnisse auf Basis eines Fallbeispiels miteinander zu vergleichen.

Sie sind Inhaber eines Großhandelsunternehmens und haben eine Datenbank angelegt, in der 76 Ihrer Produkte aufgeführt sind. Diese 76 Produkte vertreiben Sie an verschiedene Einzelhändler in verschiedenen Ländern.

Ebenfalls erfasst sind das Herkunftsland des Einzelhändlers sowie das Verkaufsdatum und der getätigte Umsatz. Nun möchten Sie mithilfe der Datenbankfunktionen Ihre Datenbank bis ins Detail analysieren und dadurch neue Erkenntnisse über Umsätze usw. generieren.

Sie machen sich an die Arbeit ...

Das Szenario

DBANZAHL() DCOUNT()

DBANZAHL(*Datenbank*; *Feld*; *Kriterien*)

Die Funktion DBANZAHL() zählt die Anzahl von Zellen in einer Spalte einer Liste oder Datenbank, die Zahlen enthalten und den angegebenen Bedingungen entsprechen. Das *Feld*-Argument ist hierbei optional. Wenn das Argument *Feld* weggelassen wird, zählt DBANZAHL() alle Datensätze, die den Kriterien entsprechen.

Datenbank (erforderlich) ist der Zellbereich, den die Liste oder Datenbank umfasst.

Feld (optional) gibt an, welche Spalte in der Funktion verwendet wird. Ein Feld kann als Text zusammen mit der Spaltenbeschriftung aufgeführt werden, und zwar in doppelten Anführungszeichen wie z.B. "Land", "Umsatz" oder "Firma". Alternativ kann ein Feld aber auch als Zahl angegeben werden, die die Position der Spalte innerhalb der Liste bezeichnet, wie z.B. 1 für die erste Spalte, 2 für die zweite usw.

Syntax

Definition

Argumente

Kriterien (erforderlich) bezeichnet den Zellbereich, der die Feldnamen und darunter die Filterkriterien enthält. Sie können für dieses Argument Suchkriterien jeden Bereichs verwenden. Hierbei muss jedoch beachtet werden, dass er mindestens eine Spaltenbeschriftung und mindestens eine Zelle unter der Spaltenbeschriftung umfasst, damit eine Bedingung für die Spalte angegeben werden kann.

Hintergrund Die Funktion DBANZAHL() übernimmt eine einfache, jedoch oftmals sehr arbeitszeit-sparende Aufgabe. Denn gerade bei Datenbanken ist ein Abzählen »von Hand« nicht mehr möglich. Möchten Sie also Datensätze in Ihrer Datenbank zählen, die bestimmten Kriterien unterliegen, verwenden Sie die Funktion DBANZAHL().

Hinweis Bei einer Zählung werden alle Argumente berücksichtigt, die entweder Zahlen, Null, Wahrheitswerte, Datumsangaben oder Zahlen in Textform sind. Argumente, die mit einem Fehlerwert oder mit einem Text belegt sind, der nicht in eine Zahl umgewandelt werden kann, werden hierbei ignoriert.

Praxiseinsatz Als Inhaber des Großhandelsunternehmens möchten Sie wissen, bei wie vielen Lieferungen innerhalb der erfassten Einträge in Ihrer Datenbank Rechnungen gestellt wurden – also Umsätze getätigt wurden, deren Betrag größer Null ist.



Um dieses Beispiel selbst auf Ihrem Rechner nachstellen zu können, finden Sie im Ordner \Ms5-235\Kap12 in der Arbeitsmappe DBFUNKTIONEN.xlsx auf dem Arbeitsblatt *Urdaten* die entsprechende Datenbank.

Sie öffnen ein neues Arbeitsblatt, in dem Sie zunächst den Kriterienbereich definieren – in diesem Fall ist dies das Feld *Umsatz* aus den *Urdaten*. Des Weiteren ist das Suchkriterium festzulegen, also >0 .

C11		fx =DBANZAHL(Datenbank;B10;B10:B11)			
	B	C	D	E	F
8					
9	Suchkriterium	Ergebnis			
10	Umsatz				
11	>0	7000			

Abbildung 12.5: Es wird die Anzahl der Einträge ermittelt, deren Umsatz größer 0 ist

Mit der Formel DBANZAHL() können Sie nun leicht die Anzahl der Einträge mit einem *Umsatz* >0 innerhalb der Datenbank errechnen.

Für das Argument *Datenbank* geben Sie den Zellbereich an, den die Datenbank umfasst, also beispielsweise *A1:F7008*. In unserem Beispiel trägt dieser Zellbereich den dynamischen Namen *Datenbank*, da es auf Dauer betrachtet zu aufwendig wäre, bei jeder Eingabe des Datenbankbereichs die Zellen von *A1:F7008* manuell zu markieren.

Das gleiche Ergebnis erzielen Sie auch, wenn Sie dem Bereich *B10:B11* – also den Suchkriterien, wie in Abbildung 12.5 dargestellt, den Namen *SUCH1* geben und das Argument *Feld*, in diesem Beispiel also der *Umsatz*, nicht durch die entsprechende Zelle *B10* angeben, sondern das Wort »Umsatz« selbst eintragen. Achten Sie hierbei darauf, dass der Feldname »Umsatz« in Anführungszeichen gesetzt wird.

	C18	=DBANZAHL(Datenbank;"Umsatz";SUCH1)			
	B	C	D	E	F
16	Suchkriterium	Ergebnis			
17	Umsatz				
18	>0	7000			

Abbildung 12.6: Die veränderte Eingabe der Argumente für die Funktion *DBANZAHL()*

Wie die Abbildung 12.6 darstellt, erhalten Sie auch hier das Ergebnis *7000*.

Das heißt, es befinden sich 7.000 Einträge in der Datenbank, deren Umsatz größer als 0 war und somit fakturiert wurden. Auf diese Weise können Sie in wenigen Minuten beispielsweise auch die Datensätze zählen, deren *Umsatz >1500* ist.

ANZAHL(), ANZAHL2(), DBANZAHL2(), ZÄHLENWENN(), ANZAHLLEEREZELLEN()

Das zugehörige Beispiel finden Sie im Ordner *Ms5-235\Kap12* in der Arbeitsmappe *DBFUNKTIONEN.xlsx* auf dem Arbeitsblatt *DBANZAHL*.

Siehe auch



DBANZAHL2() DCOUNTA()

DBANZAHL2(Datenbank;Feld;Kriterien)

Die Funktion *DBANZAHL2()* zählt die Anzahl von Zellen in einer Spalte einer Liste oder Datenbank, die nicht leer sind und den angegebenen Bedingungen entsprechen. Das *Feld*-Argument ist optional. Wird dieses Argument weggelassen, zählt *DBANZAHL2()* alle Datensätze, die den Kriterien entsprechen.

Datenbank (erforderlich) ist der Zellbereich, den die Liste oder Datenbank umfasst.

Feld (optional) gibt an, welche Spalte in der Funktion verwendet wird.

Kriterien (erforderlich) bezeichnet den Zellbereich, der die Feldnamen und darunter die Filterkriterien enthält.

Möchten Sie Datensätze in Ihrer Datenbank zählen, die nicht nur Zahlenwerte enthalten, verwenden Sie die Funktion *DBANZAHL2()*. Der Unterschied zur Funktion *DBANZAHL()* liegt darin, dass *DBANZAHL2()* jede beliebige Art von Information, auch leeren Text (""), zählt. Nicht mit einberechnet werden leere Zellen (Inhalt Null).

Da Sie Ihr Großhandelsgewerbe noch nicht so lange betreiben, möchten Sie wissen, wie viele der in der Datenbank erfassten Rechnungen an Unternehmen innerhalb Deutschlands gingen.

Des Weiteren wollen Sie herausbekommen, wie viele Rechnungen insgesamt geschrieben wurden, um anschließend den prozentualen Anteil der Rechnungen an Unternehmen innerhalb Deutschlands zur Gesamtanzahl an Rechnungen zu generieren.

Sie öffnen ein neues Arbeitsblatt, in dem Sie zunächst den Kriterienbereich innerhalb der Datenbank definieren – in diesem Fall ist dies das Feld *Land* aus den Urdaten. Des Weiteren ist das Suchkriterium festzulegen, also *Deutschland*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

F11		fx =DBANZAHL2(Datenbank;E10;E10:E11)			
	E	F	G	H	
9	Suchkriterium	Ergebnis			
10	Land				
11	Deutschland	847			

Abbildung 12.7: Es wird die Anzahl der Einträge ermittelt, die als Land »Deutschland« enthalten

Da die Formel DBANZAHL2() auch Text zählt, wird mit ihrer Hilfe die Anzahl aller Einträge ermittelt, die dem Suchkriterium »Deutschland« im Suchbereich »Land« entsprechen.

Für das Argument *Datenbank* geben Sie den Zellbereich an, den die Datenbank umfasst, also beispielsweise A1:F7008. In unserem Beispiel trägt dieser Zellbereich treffenderweise den dynamischen Namen *Datenbank*.

Das gleiche Ergebnis erzielen Sie auch, wenn Sie dem Bereich B10:B11 – also den Suchkriterien, wie in Abbildung 12.8 dargestellt, den Namen *SUCH2* geben und das Argument *Feld*, in diesem Beispiel also das *Land*, nicht durch die entsprechende Zelle B10 angeben, sondern das Wort »Land« selbst eintragen. Achten Sie hierbei darauf, dass der Feldname »Land« in Anführungszeichen gesetzt wird.

C11		fx =DBANZAHL2(Datenbank;"LAND";SUCH2)			
	B	C	D	E	
9	Suchkriterium	Ergebnis			
10	Land				
11	Deutschland	847			

Abbildung 12.8: Die veränderte Eingabe der Argumente für die Funktion DBANZAHL2()

Wie in Abbildung 12.8 darstellt, erhalten Sie auch hier das Ergebnis 847, d.h., es befinden sich 847 Einträge in der Datenbank, die im Suchbereich *Land* das Kriterium *Deutschland* erfüllen. Es wurden also 847 Rechnungen an Unternehmen innerhalb Deutschlands versandt.

Hinweis Für ganz Aufmerksame unter Ihnen sei angemerkt, dass die gezeigte Ermittlung der in Deutschland gestellten Rechnungen die Umsatzeinträge nicht berücksichtigt. Es ist wenig wahrscheinlich, dass für die Datensätze ohne Umsatz Rechnungen gestellt worden sind. Sollten Sie auf diesen Umstand Wert legen, erweitern Sie den Kriterienbereich um das *Umsatz*-Feld und notieren dort die in Abbildung 12.9 gezeigte Bedingung und Formel.

J11		fx =DBANZAHL2(Datenbank;H10;H10:I11)			
	H	I	J	K	L
9	Suchkriterium	Suchkriterium	Ergebnis		
10	Land	Umsatz			
11	Deutschland	>0	843		

Abbildung 12.9: Da bei den Null-Umsätzen keine Rechnung gestellt wird, muss das zweite Filterkriterium hinzu, um die korrekte Zahl der Rechnungen zu ermitteln

Nun berechnen Sie auf gleiche Art und Weise die Gesamtzahl an Rechnungen, die gestellt wurden. Der Suchbereich verändert sich in diesem Fall nicht. Als Suchkriterium wird auf eine leere Zelle Bezug genommen, sodass alle Einträge im Suchbereich *Land* gezählt werden.

F18		=DBANZAHL2(Datenbank;E17;E17:E18)		
	E	F	G	H
16	Suchkriterium	Ergebnis		
17	Land			
18		7007		

Abbildung 12.10: Mit *DBANZAHL2()* wird die Anzahl aller gestellten Rechnungen ermittelt

Nachdem Sie nun die Gesamtzahl aller Rechnungen in Höhe von 7.007 ermittelt haben, können Sie nun noch den prozentualen Anteil der Rechnungen an Unternehmen innerhalb Deutschlands zur Gesamtanzahl an Rechnungen generieren.

Anzahl der Rechnungen an Deutschland anteilig zur Gesamtanzahl			
12,09%	=C11/F18		
12,09%	=DBANZAHL2(Datenbank;"LAND";SUCH2)/DBANZAHL2(Datenbank;"LAND";SUCH3)		

Abbildung 12.11: Die prozentuale Verteilung der Rechnungen

Wie Sie in Abbildung 12.11 sehen, können Sie den prozentualen Anteil auch unter Verwendung der Funktion *DBANZAHL2()* errechnen. Dies ist mit Sicherheit der schwierigere Weg, aber dennoch möglich. Sie wissen nun, dass Sie 12,09 % Ihrer Rechnungen an Unternehmen innerhalb Deutschlands verschicken. Der Rest verteilt sich auf Unternehmen im Ausland.

ANZAHL(), ANZAHL2(), DBANZAHL(), ZÄHLENWENN(), ANZAHLLEEREZELLEN()

Siehe auch

Das zugehörige Beispiel finden Sie im Ordner `Ms5-235\Kap12` in der Arbeitsmappe *DBFUNKTIONEN.xlsx* auf dem Arbeitsblatt *DBANZAHL2*.



DBAUSZUG() DGET()

DBAUSZUG(Datenbank;Feld;Kriterien)

Die Funktion *DBAUSZUG()* extrahiert einen einzelnen Wert aus einer Spalte einer Liste oder Datenbank, der den angegebenen Bedingungen entspricht.

Datenbank (erforderlich) ist der Zellbereich, den die Liste oder Datenbank umfasst.

Feld (erforderlich) gibt an, welche Spalte in der Funktion verwendet wird.

Kriterien (erforderlich) bezeichnet den Zellbereich, der die Feldnamen und darunter die Filterkriterien enthält.

Wenn kein Datensatz den Kriterien entspricht, gibt *DBAUSZUG()* den Fehlerwert *#WERT!* zurück.

Wenn mehr als ein Datensatz den Kriterien entspricht, gibt *DBAUSZUG()* den Fehlerwert *#ZAHL!* zurück.

Suchen Sie **einen** bestimmten Wert innerhalb Ihrer Datenbank, also den Inhalt eines Felds eines Datensatzes unter Beachtung bestimmter Kriterien, können Sie hierzu die Funktion *DBAUSZUG()* nutzen.

Mehr über die Eingabemöglichkeiten von Suchkriterien finden Sie im Abschnitt »Argumente in Datenbankfunktionen« in diesem Kapitel ab Seite 569.

Syntax

Definition

Argumente

Hinweis

Hintergrund

Hinweis

Praxiseinsatz Im Vertrieb Ihres Großhandelsunternehmens ging eine Beschwerde von einem Ihrer Kunden – dem deutschen Unternehmen »Blauer See Delikatessen« – ein. Dieser Kunde hat das erste Mal bei Ihnen Nuss-Nougat-Creme bestellt und nun gesehen, dass diese bereits schimmelt.

Um beim Hersteller dieser Nuss-Nougat-Creme ebenfalls Beschwerde einlegen zu können und um nachzufragen, ob bereits etwaige Produktionsfehler bekannt sind, wollen Sie das Datum herausbekommen, an dem Sie die Nuss-Nougat-Creme an die Firma »Blauer See Delikatessen« verkauft haben. Hierbei hilft Ihnen die Funktion DBAUSZUG().

F11					fx =DBAUSZUG(Datenbank;E10;B10:D11)
	B	C	D	E	F
9	Suchkriterien			Suchfeld	Ergebnis
10	Firma	Land	Artikelname	Datum	
11	Blauer See Delikatessen	Deutschland	NuNuCa Nuß-Nougat-Creme		10.02.2008

Abbildung 12.12: Die Funktion DBAUSZUG() liefert das Datum, an dem die fehlerhafte Nuss-Nougat-Creme geliefert wurde

Da das Unternehmen »Blauer See Delikatessen« das erste Mal bei Ihnen Nuss-Nougat-Creme bestellt hat, können Sie also sicher sein, dass die Funktion DBAUSZUG() ein Ergebnis liefert.

Wie Sie in Abbildung 12.12 sehen können, liefert DBAUSZUG() unter Angabe der Suchkriterien *Firmennamen*, *Land* und *Artikelname* (B10:D11) das gesuchte Datum, was in diesem Beispiel das Suchfeld darstellt. Es handelt sich um den 10. Februar 2008.

Auf diese Weise können Sie in Minutenschnelle, auch unter Angabe mehrerer Suchkriterien, aus Ihrer Datenbank einzelne Werte generieren.

Siehe auch DBMAX(), DBMIN()



Das zugehörige Beispiel finden Sie im Ordner \Ms5-235\Kap12 in der Arbeitsmappe DBFUNKTIONEN.xlsx auf dem Arbeitsblatt DBAUSZUG.

DBMAX() DMAX()

Syntax DBMAX(*Datenbank*;Feld;Kriterien)

Definition Die Funktion DBMAX() liefert die größte Anzahl bzw. den größten Wert in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entspricht.

Argumente *Datenbank* (erforderlich) ist der Zellbereich, den die Liste oder Datenbank umfasst.

Feld (erforderlich) gibt an, welche Spalte in der Funktion verwendet wird.

Kriterien (erforderlich) bezeichnet den Zellbereich, der die Feldnamen und darunter die Filterkriterien enthält.

Hintergrund Für die Ermittlung der größten bzw. kleinsten Werte innerhalb einer Datenbank unter Verwendung bestimmter Kriterien nutzen Sie die Funktionen DBMAX() und DBMIN(). Mithilfe dieser Funktionen erhalten Sie in wenigen Minuten aus einer Datenbank die Zahl, die beispielsweise der höchsten Produktionsmenge eines bestimmten Produkts innerhalb der letzten fünf Jahre entspricht.

Als Inhaber des Großhandelsunternehmens haben Sie bereits viele Kunden akquiriert und auch viele Umsätze mit den im Sortiment zur Verfügung stehenden Produkten erzielt.

Sie möchten nun alle drei Komponenten analysieren und zunächst wissen, mit welchem Produkt und mit welchem Kunden Sie den größten Umsatz in Bezug auf eine Bestellung innerhalb Deutschlands gemacht haben. Hierzu verwenden Sie die Funktion DBMAX().

	B	C	D	E
9	Suchkriterien		Suchfeld	Ergebnis
10	Artikelname	Land	Umsatz	
11		Deutschland		2000

Abbildung 12.13: Die Funktion *DBMAX()* liefert den größten Wert in einer Liste unter Verwendung bestimmter Kriterien

Wie Sie in Abbildung 12.13 sehen können, werden als Suchkriterien für die Errechnung des größten Werts der Artikelname und das Land verwendet. Da Sie nur den größten Umsatz innerhalb Deutschlands suchen, wird »Deutschland« auch entsprechend angegeben (*C11*). Für das Produkt bzw. den Artikelnamen werden keine weiteren Angaben gemacht, da alle zum Verkauf stehenden Produkte in die Berechnung mit einbezogen werden sollen.

Als Ergebnis liefert *DBMAX()* den Wert *2000*. Das heißt, der höchste Umsatz für eine Bestellung innerhalb Deutschlands beträgt 2.000 €.

Falls Sie nun noch wissen wollen, welches Unternehmen mit welchem Produkt Ihnen diesen Umsatz eingebracht hat, können Sie beispielsweise mit der Funktion *DBAUSZUG()* weitere Berechnungen anstellen. Die Abbildung 12.14 zeigt Ihnen einen möglichen Lösungsweg.

	B	C	D	E
14	Wie heißt das Unternehmen, das eine Bestellung für 2000€ aufgegeben hat?			
15				
16	Suchkriterien		Suchfeld	Ergebnis
17	Land	Umsatz	Firma	
18	Deutschland	2000		Die Wandernde Kuh
19				=DBAUSZUG(Datenbank;D17;B17:C18)
20				
21	Mit welchem Produkt hat dieses Unternehmen den Umsatz gemacht?			
22				
23	Suchkriterien		Suchfeld	Ergebnis
24	Firma	Umsatz	Artikelname	
25	Die Wandernde Kuh	2000		Alice Mutton
26				=DBAUSZUG(Datenbank;D24;B24:C25)

Abbildung 12.14: Zur weiteren Analyse wird die Funktion *DBAUSZUG()* herangezogen

Auf diese Weise können Sie mit wenig Zeitaufwand aussagekräftige Analysen erstellen und Ihre Umsätze und Kunden genau unter die Lupe nehmen.

DBMIN(), MAX(), MIN()

Das zugehörige Beispiel finden Sie im Ordner *Ms5-235\Kap12* in der Arbeitsmappe *DBFUNKTIONEN.xlsx* auf dem Arbeitsblatt *DBMAX*.

Siehe auch



DBMIN() DMIN()

Syntax DBMIN(*Datenbank*;Feld;Kriterien)

Definition Die Funktion DBMIN() liefert die kleinste Anzahl bzw. den kleinsten Wert in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entspricht.

Argumente *Datenbank* (erforderlich) ist der Zellbereich, den die Liste oder Datenbank umfasst.
Feld (erforderlich) gibt an, welche Spalte in der Funktion verwendet wird.

Kriterien (erforderlich) bezeichnet den Zellbereich, der die Feldnamen und darunter die Filterkriterien enthält.

Hintergrund Für die Ermittlung der kleinsten bzw. größten Werte innerhalb einer Datenbank unter Verwendung bestimmter Kriterien nutzen Sie die Funktionen DBMIN() und DBMAX(). Mithilfe dieser Funktionen erhalten Sie in wenigen Minuten aus einer Datenbank die Zahl, die bspw. dem kleinsten Umsatz für ein bestimmtes Produkt innerhalb der letzten fünf Jahre entspricht.

Praxiseinsatz Ihr Großhandelsunternehmen läuft recht gut. Sie machen gute Umsätze und der Verkauf läuft. Dennoch wollen Sie analysieren, mit welchem Produkt Sie in der vergangenen Zeit den geringsten Umsatz gemacht haben, um anschließend eventuell das Sortiment zu ändern oder das entsprechende Produkt auszutauschen.

Deshalb wollen Sie Schritt für Schritt für jedes Unternehmen das Produkt ermitteln, das den geringsten Umsatz bringt. Hierzu verwenden Sie die Funktion DBMIN().

	B	C	D
9	Firma	Umsatz	Ergebnis
10	Alfreds Futterkiste	>0	1002

Abbildung 12.15: Die Funktion *DBMIN()* liefert den kleinsten Umsatz für das Unternehmen »Alfreds Futterkiste«

Wie Sie in Abbildung 12.15 sehen können, werden als Suchkriterien für die Errechnung des kleinsten Werts der Firmenname und der Umsatz größer 0 verwendet. Da in der Datenbank auch Einträge mit einem Umsatz von 0 € vorhanden sind, muss das Kriterium >0 angegeben werden.

DBMIN() liefert als kleinsten Wert einen Umsatz von 1.002 € für das Unternehmen »Alfreds Futterkiste«. Da Sie noch wissen wollen, mit welchem Produkt »Alfreds Futterkiste« den kleinsten Umsatz erbracht hat, bedienen Sie sich der Funktion *DBAUSZUG()*.

Die Abbildung 12.16 zeigt Ihnen den Lösungsweg.

	B	C	D	E
14	Mit welchem Produkt wurde dieser Umsatz gemacht?			
15				
16	Umsatz	Firma	Artikelname	Ergebnis
17	1002	Alfreds Futterkiste		Geitost

Abbildung 12.16: Errechnen Sie das gesuchte Produkt mit *DBAUSZUG()*

Auf diese Weise können Sie nun jedes Unternehmen für sich analysieren, die Produkte mit dem geringsten Umsatz ermitteln und gegebenenfalls Ihr Sortiment anpassen.

DBMAX(), MAX(), MIN()

Das zugehörige Beispiel finden Sie im Ordner `Ms5-235\Kap12` in der Arbeitsmappe `DBFUNKTIONEN.xlsx` auf dem Arbeitsblatt `DBMIN`.

Siehe auch



DBMITTELWERT() DAVERAGE()

DBMITTELWERT(*Datenbank;Datenbankfeld;Suchkriterien*)

DBMITTELWERT() liefert das arithmetische Mittel aus den Werten einer Listen- oder Datenbankspalte, die den von Ihnen angegebenen Bedingungen entsprechen.

Datenbank (erforderlich) ist der Zellbereich, aus dem die Liste oder Datenbank besteht.

Datenbankfeld (erforderlich) gibt an, welche Spalte in der Funktion verwendet wird.

Suchkriterien (erforderlich) ist der Zellbereich, der die angegebenen Bedingungen enthält.

Das arithmetische Mittel ist der bekannteste Mittelwert und hat damit auch die größte Akzeptanz unter den Nicht-Statistikern. Da der Mittelwert recht einfach zu berechnen ist und auf allen Werten beruht, in diesem Fall auf allen Werten, die den vorgegebenen Kriterien entsprechen, spielt er in der schließenden Statistik eine wichtige Rolle.

Um den Mittelwert zu errechnen, werden die Einzelwerte eines Wertbereichs addiert und durch die Anzahl der Zahlenwerte dividiert.

Mehr Informationen zum Mittelwert finden Sie in Kapitel 8.

Mit der Funktion DBMITTELWERT() errechnen Sie den Mittelwert der ausgewählten Datenbankeinträge.

Es ist Jahresende und Sie wollen sich als Geschäftsführer des Großhandelsunternehmens näher mit den Umsätzen der vergangenen Jahre befassen. Um zunächst einen Überblick zu erhalten, welches Ihrer Produkte in welchem Land im Schnitt am meisten Umsatz gebracht hat, wollen Sie mithilfe der Funktion DBMITTELWERT() die entsprechenden Werte errechnen.

Syntax

Definition

Argumente

Hintergrund

Hinweis

Praxiseinsatz

	B	C	D	E
7	Wie hoch ist der durchschnittliche Umsatz mit Chang in Deutschland?			
8				
9	Suchkriterien		Datenbankfeld	Ergebnis
10	Land	Artikelname	Umsatz	
11	Deutschland	Chang		1473,00

Abbildung 12.17: Die Funktion `DBMITTELWERT()` liefert den durchschnittlichen Umsatz für den Artikel »Chang« in Deutschland

Um den durchschnittlichen Umsatz für »Chang« in Deutschland zu erhalten, werden als Suchkriterien das Land *Deutschland* und der Artikelname *Chang* festgelegt. Da nach dem durchschnittlichen Umsatz gesucht wird, dient *Umsatz* als Datenbankfeld. Wie Sie in Abbildung 12.17 sehen können, liefert `DBMITTELWERT()` den Wert *1.473,00 €*.

Auf gleiche Weise wollen Sie nun noch errechnen, wie hoch der durchschnittliche Gesamtumsatz mit dem Produkt *Chang* war.

E18		fx =DBMITTELWERT(Datenbank;D17;B17:C18)		
	B	C	D	E
14	Wie hoch ist der gesamte durchschnittliche Umsatz mit Chang?			
15				
16	Suchkriterien		Datenbankfeld	Ergebnis
17	Land	Artikelname	Umsatz	
18		Chang		1560,02

Abbildung 12.18: Die Funktion `DBMITTELWERT()` liefert den durchschnittlichen Umsatz für den Artikel »Chang«

Um den durchschnittlichen Umsatz für *Chang* nicht länderbezogen zu berechnen, wird in diesem Fall eine Angabe für das Suchkriterium *Land* weggelassen. Das heißt, es werden alle unter dem Suchkriterium *Land* aufgeführten Länder mit in die Berechnung einbezogen. `DBMITTELWERT()` ermittelt dann einen Umsatz von *1.560,02 €*.

Auf diese Weise können Sie eine Vielzahl von Berechnungen erstellen und Vergleiche zwischen einzelnen durchschnittlichen Umsätzen für bestimmte Produkte in verschiedenen Ländern ziehen. Ebenso können Sie beispielsweise den prozentualen Anteil eines durchschnittlichen Umsatzes für ein Produkt im Vergleich zum durchschnittlichen Gesamtumsatz dieses Produkts errechnen.

Siehe auch `MITTELWERT()`, `MEDIAN()`



Das zugehörige Beispiel finden Sie im Ordner `\Ms5-235\Kap12` in der Arbeitsmappe `DBFUNKTIONEN.xlsx` auf dem Arbeitsblatt `DBMITTELWERT`.

DBPRODUKT() DPRODUCT()

Syntax `DBPRODUKT(Datenbank;Feld;Kriterien)`

Definition Die Funktion `DBPRODUKT()` multipliziert die Werte in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen.

Argumente *Datenbank* (erforderlich) ist der Zellbereich, den die Liste oder Datenbank umfasst.
Feld (erforderlich) gibt an, welche Spalte in der Funktion verwendet wird.

Kriterien (erforderlich) bezeichnet den Zellbereich, der die angegebenen Bedingungen enthält.

Hintergrund Der Begriff Produkt bezeichnet in der Mathematik das Ergebnis einer Multiplikation von Zahlen oder bestimmter anderer Verknüpfungen. Mit der Funktion `DBPRODUKT()` können Sie, unter Angabe bestimmter Kriterien, Werte einer Liste – also Werte aus einer Spalte – miteinander multiplizieren und so in Kürze deren Produkt generieren.

Da der Einsatz der Funktion DBPRODUKT() in Bezug auf unser Beispiel keinen Sinn machen würde, möchten wir in diesem Fall auf ein anderes Beispiel zurückgreifen.

Praxiseinsatz

Wir möchten an dieser Stelle erwähnen, dass es uns in diesem Fall sehr schwer gefallen ist, ein sinnvolles Anwendungsbeispiel zu finden. Uns sind keine Anwendungen für diese Funktion bekannt, da das Ausmultiplizieren eines Datenbankfelds in den uns bekannten Anwendungen nicht benötigt wird. Bei unseren Recherchen stießen wir auf das von Microsoft in der Hilfe dargestellte Beispiel. Welche Aussage jedoch das Produkt aller Erträge der Apfelbäume zwischen 1,5 und 2,5 m Höhe haben soll, erklären weder Microsoft in der Excel-Hilfe noch andere Quellen im Web.

Hinweis

Andere, in der jüngeren Literatur beschriebene Anwendungsbeispiele sind grundsätzlich auch mit den anderen Funktionen zu lösen und erfordern nicht zwingend DBPRODUKT(). Deshalb beschränken wir uns auf eine ganz simple Darstellung einer evtl. denk-, allerdings auch streitbaren, Anwendung der Funktion.

Stellen Sie sich vor, Sie sind Immobilienmakler und haben eine Datenbank mit den von Ihnen zum Verkauf angebotenen Wohnungen angelegt (Abbildung 12.19).

	B	C	D	E
9	Wohnung	Zimmer	Abmaße für	Maße in m
10	Schusterstraße 5	1	Länge	3,50
11	Schusterstraße 5	1	Breite	2,10
12	Schusterstraße 5	2	Länge	5,20
13	Schusterstraße 5	2	Breite	2,45
14	Schusterstraße 5	3	Länge	3,60
15	Schusterstraße 5	3	Breite	4,10
16	Schusterstraße 5	4	Länge	4,70
17	Schusterstraße 5	4	Breite	3,30
18	Ankerstraße 23	1	Länge	3,50
19	Ankerstraße 23	1	Breite	2,10
20	Ankerstraße 23	2	Länge	5,20
21	Ankerstraße 23	2	Breite	2,45
22	Ankerstraße 23	3	Länge	3,60
23	Ankerstraße 23	3	Breite	4,10
24	Maxstraße 134	1	Länge	4,70
25	Maxstraße 134	1	Breite	3,30
26	Maxstraße 134	2	Länge	3,50
27	Maxstraße 134	2	Breite	2,10
28	Maxstraße 134	3	Länge	5,20
29	Maxstraße 134	3	Breite	2,45
30	Maxstraße 134	4	Länge	3,60
31	Maxstraße 134	4	Breite	4,10
32	Maxstraße 134	5	Länge	4,70
33	Maxstraße 134	5	Breite	3,30
34	Maxstraße 134	6	Länge	3,50
35	Maxstraße 134	6	Breite	2,10

Abbildung 12.19: Die Immobiliendatenbank, aufgeschlüsselt nach den einzelnen Zimmern

Sie haben für jede Wohnung und für jedes darin befindliche Zimmer *Länge* und *Breite* angegeben und möchten nun durch eine einfache Eingabe die zur Verfügung stehenden Quadratmeter pro Zimmer errechnen. So können Sie Ihren Kunden im Handumdrehen Auskunft über die genaue Größe der Zimmer geben.

Um die Quadratmeterzahl zu berechnen, verwenden Sie die Funktion DBPRODUKT(). Abbildung 12.20 zeigt einen möglichen Lösungsweg.

J11		fx =DBPRODUKT(B9:E35;I10;G10:H11)	
	G	H	I
9	Suchkriterien		Datenbankfeld
10	Wohnung	Zimmer	Maße in m
11	Schusterstraße 5	1	7,35

Abbildung 12.20: Die zur Verfügung stehenden Quadratmeter werden unter bestimmten Bedingungen berechnet

In unserem Beispiel wird für das Argument *Datenbank* der Bereich *B9:E35* markiert. Als *Kriterien* wurden die Straße sowie eines der Zimmer angegeben und das Argument *Feld* wird durch die Zelle *I10* also *Maße in m* ausgedrückt.

Auf diese Weise erhalten Sie die in Zimmer »1« der Wohnung in der »Schusterstraße 5« zur Verfügung stehenden Quadratmeter von 7,35 m².

Ändern Sie die in Zelle *H11* angegebene Zahl oder geben Sie einen anderen Straßennamen ein, erhalten Sie automatisch die aktualisierte, und unter den neuen Bedingungen berechnete, Quadratmeterzahl.

Siehe auch DBSUMME(), PRODUKT(), SUMMEWENN()



Das zugehörige Beispiel finden Sie im Ordner *\Ms5-235\Kap12* in der Arbeitsmappe *DBFUNKTIONEN.xlsx* auf dem Arbeitsblatt *DBPRODUKT*.

DBSTDABW() DSTDEV()

Syntax DBSTABW(*Datenbank*;Feld;Kriterien)

Definition Die Funktion DBSTDABW() schätzt die Standardabweichung einer Grundgesamtheit ausgehend von den Zahlen aus einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen.

Argumente *Datenbank* (erforderlich) ist der Zellbereich, den die Liste oder Datenbank umfasst.

Feld (erforderlich) gibt an, welche Spalte in der Funktion verwendet wird.

Kriterien (erforderlich) bezeichnet den Zellbereich, der die angegebenen Bedingungen enthält.

Hintergrund Die Standardabweichung gehört zu den wichtigsten Streumaßen. Wie die Varianz beschreibt auch sie die Abweichung der Werte vom Mittelwert. Allerdings wird für die Berechnung nicht wie bei der Varianz die Differenz der Werte vom Mittelwert verwendet, sondern das Quadrat der Differenz.

Anders ausgedrückt: Die Standardabweichung ist die Quadratwurzel aus dem arithmetischen Mittel der quadrierten Abweichung der Werte vom arithmetischen Mittel. Sie entspricht der Wurzel aus der Varianz.

Hinweis Mehr Informationen zum Thema »Standardabweichung und Varianz« finden Sie in Kapitel 11.

Als Inhaber des Großhandelsunternehmens haben Sie bereits einige Auswertungen erstellt. Nun widmen Sie sich der Funktion DBSTDABW(), um noch detaillierter Auswertungen bezüglich Ihrer getätigten Umsätze zu generieren.

Praxiseinsatz

Ziel soll es sein, die Standardabweichung, ausgehend von einer Stichprobe, unter Verwendung bestimmter Kriterien zu errechnen, um anschließend beurteilen zu können, inwieweit die Umsätze für ein bestimmtes Produkt innerhalb eines bestimmten Landes um den Mittelwert streuen.

Beachten Sie, dass es sich bei den zu untersuchenden Daten um eine Stichprobe handelt. Das heißt, es wurden bereits mehr Umsätze mit diesem Produkt im entsprechenden Land getätigt. Es wird jedoch lediglich eine Stichprobe daraus betrachtet.

Hinweis

E11			
fx =DBSTDABW(Datenbank;D10;B10:C11)			
B	C	D	E
8	In welcher Höhe streuen die Umsätze für das Produkt <i>Queso Cabrales</i> innerhalb Deutschlands um deren Mittelwert ausgehend von einer Stichprobe?		
9			
10	Land	Artikelname	Umsatz
11	Deutschland	Queso Cabrales	248,15

Abbildung 12.21: Die Berechnung der Standardabweichung, ausgehend von einer Stichprobe

Um eine Aussage treffen zu können, fehlt nun noch die Berechnung des dazugehörigen Mittelwerts.

E18			
fx =DBMITTELWERT(Datenbank;D17;B17:C18)			
B	C	D	E
15	Wie hoch ist der Mittelwert für die Umsätze mit dem Produkt <i>Queso Cabrales</i> innerhalb Deutschlands?		
16			
17	Land	Artikelname	Umsatz
18	Deutschland	Queso Cabrales	1448,27

Abbildung 12.22: Der Mittelwert für den Umsatz des Produkts »Queso Cabrales« innerhalb Deutschlands

Folgende Aussage kann nun getroffen werden: Das Ergebnis der Berechnung der Standardabweichung ausgehend von einer Stichprobe und unter Beachtung bestimmter Kriterien von 248,15 bedeutet, dass die Umsätze von *Queso Cabrales* innerhalb Deutschlands in Höhe von 248,15 € um den dazugehörigen Mittelwert, der in diesem Fall 1.448,27 € beträgt, streuen. Das heißt, die Umsätze sind im Schnitt um 248,15 € höher oder niedriger als der Mittelwert. DBSTDABW(), STABWA(), STABW.N(), STABWNA()

Aussagen

Das zugehörige Beispiel finden Sie im Ordner \Ms5-235\Kap12 in der Arbeitsmappe DBFUNKTIONEN.xlsx auf dem Arbeitsblatt DBSTDABW.

Siehe auch



DBSTDABWN() DSTDEVP()

Syntax DBSTDABWN(*Datenbank*; *Feld*; *Kriterien*)

Definition Die Funktion DBSTDABWN() berechnet die Standardabweichung einer vollständigen Grundgesamtheit ausgehend von den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen.

Argumente *Datenbank* (erforderlich) ist der Zellbereich, den die Liste oder Datenbank umfasst.

Feld (erforderlich) gibt an, welche Spalte in der Funktion verwendet wird.

Kriterien (erforderlich) bezeichnet den Zellbereich, der die angegebenen Bedingungen enthält.

Hintergrund Die Funktion DBSTDABWN() unterscheidet sich zur Funktion DBSTDABW() lediglich darin, dass sie die Standardabweichung nicht auf Basis einer Stichprobe berechnet, sondern auf Basis einer Grundgesamtheit unter Beachtung der angegebenen Kriterien.

Hinweis Mehr Informationen zum Thema »Standardabweichung« finden Sie in Kapitel 11 sowie in der Beschreibung zur Funktion DBSTDABW() in diesem Kapitel auf Seite 584.

Praxiseinsatz Als Inhaber des Großhandelsunternehmens sind Sie immer noch mit der Auswertung Ihrer Excel-Datenbank beschäftigt. Sie haben bereits die Standardabweichung für den Umsatz eines bestimmten Produkts mit der Funktion DBSTDABW() berechnet. Nun möchten Sie das Gleiche auf Basis einer Grundgesamtheit berechnen.

Ziel soll es also sein, die Standardabweichung ausgehend von einer Grundgesamtheit und unter Verwendung bestimmter Kriterien zu errechnen, um anschließend beurteilen zu können, inwieweit die Umsätze für ein bestimmtes Produkt innerhalb eines bestimmten Landes um den Mittelwert streuen.

Hinweis Beachten Sie, dass es sich bei den zu untersuchenden Umsätzen um die Grundgesamtheit handelt. Das heißt, die Umsätze die untersucht werden, entsprechen **allen** bisher getätigten Umsätzen. Deshalb liefern die Funktionen DBSTDABW() und DBSTDABWN() auch unterschiedliche Ergebnisse.

E11		fx =DBSTDABWN(Datenbank;D10;B10:C11)		
B	C	D	E	
8	In welcher Höhe streuen die Umsätze für das Produkt <i>Queso Cabrales</i> innerhalb Deutschlands um deren Mittelwert ausgehend von einer Grundgesamtheit?			
9				
10	Land	Artikelname	Umsatz	Ergebnis
11	Deutschland	Queso Cabrales		236,60

Abbildung 12.23: Die Berechnung der Standardabweichung, ausgehend von einer Grundgesamtheit

Um eine Aussage treffen zu können, fehlt nun noch die Berechnung des dazugehörigen Mittelwerts.

E18		fx =DBMITTELWERT(Datenbank;D17;B17:C18)		
B	C	D	E	
15	Wie hoch ist der Mittelwert für die Umsätze mit dem Produkt <i>Queso Cabrales</i> innerhalb Deutschlands?			
16				
17	Land	Artikelname	Umsatz	Ergebnis
18	Deutschland	Queso Cabrales		1448,27

Abbildung 12.24: Der Mittelwert für den Umsatz des Produkts »Queso Cabrales« innerhalb Deutschlands

Folgende Aussage kann nun getroffen werden: Das Ergebnis der Berechnung der Standardabweichung ausgehend von einer Grundgesamtheit und unter Beachtung bestimmter Kriterien von 236,60 bedeutet, dass die Umsätze von *Queso Cabrales* innerhalb Deutschlands in Höhe von 236,60 € um den dazugehörigen Mittelwert, der in diesem Fall 1.448,27 € beträgt, streuen. Das heißt, die Umsätze sind im Schnitt um 236,60 € höher oder niedriger als der Mittelwert.

DBSTABW(), STABW.S(), STABWA(), STABWNA()

Das zugehörige Beispiel finden Sie im Ordner `Ms5-235\Kap12` in der Arbeitsmappe `DBFUNKTIONEN.xlsx` auf dem Arbeitsblatt `DBSTDABWN`.

Aussagen

Siehe auch



DBSUMME() DSUM()

DBSUMME(*Datenbank;Feld;Kriterien*)

Die Funktion DBSUMME() addiert die Zahlen in einer Spalte aus einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen.

Datenbank (erforderlich) ist der Zellbereich, den die Liste oder Datenbank umfasst.

Feld (erforderlich) gibt an, welche Spalte in der Funktion verwendet wird.

Kriterien (erforderlich) bezeichnet den Zellbereich, der die angegebenen Bedingungen enthält.

Unter Verwendung der Datenbankfunktion DBSUMME() können Sie einen Bereich mit Suchkriterien versehen und bei Übereinstimmung die Werte aus diesem Bereich addieren. Die Angabe eines Suchkriteriums ist hierbei erforderlich.

Wenn Sie sich für Summen aus Daten interessieren, die nicht nur einer Bedingung oder einem Kriterium genügen, bietet Ihnen die Funktion DBSUMME() die besten Voraussetzungen.

Die Datenbank, die Sie zur Kontrolle aller getätigten Umsätze innerhalb Ihres Großhandelsunternehmens angelegt haben, enthält nun bereits viele Einträge. Sie sind dabei, die Inhalte bis ins Detail zu analysieren.

Sie haben mithilfe der Funktion DBMITTELWERT() den durchschnittlichen Umsatz für das Produkt *Chai* in Deutschland ermittelt.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

E24				
=DBMITTELWERT(Datenbank;D23;B23:C24)				
	B	C	D	E
21	Wie hoch ist der durchschnittliche Umsatz für Chai in Deutschland?			
22				
23	Land	Artikelname	Umsatz	Ergebnis
24	Deutschland	Chai		1486,67

Abbildung 12.25: Der durchschnittliche Umsatz für das Produkt »Chai« in Deutschland

Mehr Informationen zum Thema »Mittelwert« finden Sie in Kapitel 11.

Hinweis

Um nun herauszubekommen, wie hoch die Summe der Umsätze ist, die über diesem Mittelwert liegt, bedienen Sie sich der Funktion `DBSUMME()`. Das heißt, Sie berechnen die Summe der Umsätze unter Berücksichtigung der Kriterien

- ▶ Land = Deutschland
- ▶ Artikelname = Chai
- ▶ Umsatz = >1486,67 €

Die Abbildung 12.26 zeigt die Lösung.

E10		fx =DBSUMME(Datenbank;D9;B9:D10)		
	B	C	D	E
7	Wie hoch ist die Summe der Umsätze über 1486,67€ für das Produkt Chai in Deutschland?			
8				
9	Land	Artikelname	Umsatz	Ergebnis
10	Deutschland	Chai	>1486,67	9549,00

Abbildung 12.26: Die Berechnung der Umsatzsumme mit bestimmten Kriterien

Für das Argument *Datenbank* geben Sie den Zellbereich an, den die Datenbank umfasst. Also bspw. von Zelle `A1:F7008`. In unserem Beispiel trägt dieser Zellbereich den dynamischen Namen *Datenbank*.

Das gleiche Umsatzergebnis erzielen Sie also auch, wenn Sie dem Bereich `B9:D10` – also den Suchkriterien, wie in Abbildung 12.26 dargestellt – den Namen *SUCH4* geben und das Argument *Feld*, in diesem Beispiel also der Umsatz, nicht durch die entsprechende Zelle `D9` angeben, sondern das Wort »Umsatz« selbst eintragen. Achten Sie hierbei darauf, dass der Feldname »Umsatz« in Anführungszeichen gesetzt wird.

E17		fx =DBSUMME(Datenbank;"Umsatz";SUCH4)		
	B	C	D	E
15				
16	Land	Artikelname	Umsatz	Ergebnis
17	Deutschland	Chai	>1486,67	9549,00

Abbildung 12.27: Die Berechnung der Umsatzsumme mithilfe von Namen

Bisher haben Sie in Bezug auf die Kriterien immer mit UND-Bedingungen gearbeitet. Natürlich haben Sie auch die Möglichkeit, ODER-Bedingungen aufzustellen.

Stellen Sie sich vor, Sie haben für den Gesamtumsatz für *Chai* in *Deutschland* bereits die Standardabweichung ausgehend von einer Stichprobe, wie in Abbildung 12.28 dargestellt, ermittelt.

Hinweis Mehr Informationen zum Thema »Standardabweichung« finden Sie in Kapitel 11.

E31		fx =DBSTDABW(Datenbank;D30;B30:C31)		
	B	C	D	E
28	Wie hoch ist die Standardabweichung (Stichprobe) in Bezug auf den Umsatz für Chai in Deutschland?			
29				
30	Land	Artikelname	Umsatz	Ergebnis
31	Deutschland	Chai		232,16

Abbildung 12.28: Die Standardabweichung im Bezug auf den Umsatz für das Produkt »Chai« in Deutschland

Das heißt, die Umsätze von *Chai* innerhalb Deutschlands streuen in Höhe von 232,16 € um den dazugehörigen Mittelwert, der in diesem Fall 1.486,67 € beträgt. Das heißt, die Umsätze sind im Schnitt um 232,16 € höher oder niedriger als der dazugehörige Mittelwert.

Nun wollen Sie berechnen, wie hoch die Umsätze für das Produkt *Chai* in Deutschland sind – allerdings möchten Sie nur die Umsätze summieren, die entweder über dem Mittelwert plus der Standardabweichung oder unter dem Mittelwert abzüglich der Standardabweichung liegen.

Das heißt, folgende Kriterien müssen erfüllt sein:

- ▶ Land = Deutschland UND
- ▶ Artikelname = Chai UND
- ▶ Umsatz = > 1718,83 € (Mittelwert plus Standardabweichung)

ODER

- ▶ Land = Deutschland UND
- ▶ Artikelname = Chai UND
- ▶ Umsatz = < 1254,51 € (Mittelwert minus Standardabweichung)

Das Ergebnis ist in Abbildung 12.29 dargestellt.

E38		fx =DBSUMME(Datenbank;D36;B36:D38)		
	B	C	D	E
34	Wie hoch ist die Summe der Umsätze für Chai in Deutschland deren Wert entweder über 1718,83€ oder unter 1254,51€ liegen?			
35				
36	Land	Artikelname	Umsatz	Ergebnis
37	Deutschland	Chai	>1718,83	
38	Deutschland	Chai	<1254,51	2906

Abbildung 12.29: Die Berechnung der Summe mit einer *ODER*-Bedingung

Kontrollieren Sie das berechnete Ergebnis von 2.906 € mit den Urdaten, erkennen Sie dessen Richtigkeit. Es gibt lediglich zwei Umsätze für *Chai* in Deutschland, die den angegebenen Kriterien entsprechen.

	A	B	C	D	E	F
1	Firma	Ort	Datum	Land	Artikelname	Umsatz
2	Alfreds Futterkiste	Berlin	12.03.08	Deutschland	Chai	kein
387	Blauer See Delikatessen	Mannheim	22.11.07	Deutschland	Chai	kein
1234	Drachenblut Delikatessen	Aachen	17.01.08	Deutschland	Chai	1.471,00 €
1850	Frankenversand	München	08.10.07	Deutschland	Chai	1.516,00 €
2928	Königlich Essen	Brandenburg	10.03.08	Deutschland	Chai	1.535,00 €
3313	Lehmanns Marktstand	Frankfurt a.M.	06.02.08	Deutschland	Chai	1.354,00 €
3929	Morgenstern Gesundkost	Leipzig	21.08.07	Deutschland	Chai	1.510,00 €
4237	Ottilies Käseladen	Köln	10.10.07	Deutschland	Chai	1.006,00 €
4776	QUICK-Stop	Cunewalde	25.11.07	Deutschland	Chai	1.568,00 €
6008	Toms Spezialitäten	Münster	12.12.07	Deutschland	Chai	1.520,00 €
6547	Die Wandernde Kuh	Stuttgart	24.02.08	Deutschland	Chai	1.900,00 €

Abbildung 12.30: Der Vergleich mit den Urdaten

Auf diese Weise können Sie mit der Funktion DBSUMME() und unter Berücksichtigung bestimmter Kriterien, die entweder durch eine UND-Bedingung oder aber durch eine ODER-Bedingung miteinander verknüpft sind, Ihre Umsätze analysieren.

Siehe auch SUMMEWENN(), SUMME()



Das zugehörige Beispiel finden Sie im Ordner \Ms5-235\Kap12 in der Arbeitsmappe DBFUNKTIONEN.xlsx auf dem Arbeitsblatt DBSUMME.

DBVARIANZ() DVAR()

Syntax DBVARIANZ(*Datenbank*;Feld;Kriterien)

Definition Die Funktion DBVARIANZ() schätzt die Varianz ausgehend von einer Stichprobe mit den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen.

Argumente *Datenbank* (erforderlich) ist der Zellbereich, den die Liste oder Datenbank umfasst.

Feld (erforderlich) gibt an, welche Spalte in der Funktion verwendet wird.

Kriterien (erforderlich) bezeichnet den Zellbereich, der die angegebenen Bedingungen enthält.

Hintergrund Die gebräuchlichsten Streumaße in der Statistik sind sicher die *Varianz* und die *Standardabweichung*. Die Varianz ist ein Maß für die Abweichung einer Zufallsvariable x von ihrem Erwartungswert $E(x)$. Anders ausgedrückt ist die Varianz der Durchschnitt der quadratischen Abweichungen der Einzelwerte vom Durchschnitt, dividiert durch die Anzahl der Werte. Das Ergebnis wird auch als sogenannte »Empirische Varianz« bezeichnet.

Hinweis Mehr Informationen zum Thema »Varianz« finden Sie in Kapitel 11.

Praxiseinsatz Da die Stichprobenvarianz als Maß für die Streubreite von Daten verwendet wird, taucht sie in der deskriptiven Statistik recht häufig auf.

Auch Sie, als Inhaber eines Großhandelsunternehmens, haben sich der Funktion DBVARIANZ() angenommen, um noch detailliertere Auswertungen bezüglich Ihrer Datenbank zu generieren.

Kommen wir auf unser Beispiel aus der Beschreibung zur Funktion DBSTDABW() auf Seite 584 zurück. Ziel soll es nun sein, anstatt der Standardabweichung die Varianz, ausgehend von einer Stichprobe, unter Verwendung bestimmter Kriterien zu errechnen, um anschließend beurteilen zu können, inwieweit die Umsätze für ein bestimmtes Produkt innerhalb eines bestimmten Landes um den Mittelwert streuen.

Beachten Sie, dass es sich bei den zu untersuchenden Daten um eine Stichprobe handelt. Das heißt, es wurden bereits mehr Umsätze mit diesem Produkt im entsprechenden Land getätigt. Es wird jedoch lediglich eine Stichprobe daraus betrachtet.

Hinweis

	B	C	D	E
7	Wie hoch ist die Varianz in Bezug auf den Umsatz für das Produkt Queso Cabrales innerhalb Deutschlands ausgehend von einer Stichprobe?			
8				
9	Land	Artikelname	Umsatz	Ergebnis
10	Deutschland	Queso Cabrales		61576,02

Abbildung 12.31: Die Berechnung der Varianz ausgehend von einer Stichprobe

Um eine Aussage treffen zu können, fehlt nun noch die Berechnung des dazugehörigen Mittelwerts.

	B	C	D	E
14	Wie hoch ist der Mittelwert für die Umsätze mit dem Produkt Queso Cabrales innerhalb Deutschlands?			
15				
16	Land	Artikelname	Umsatz	Ergebnis
17	Deutschland	Queso Cabrales		1448,27

Abbildung 12.32: Der Mittelwert für den Umsatz des Produkts »Queso Cabrales« innerhalb Deutschlands

Folgende Aussage kann nun getroffen werden: Das Ergebnis der Berechnung der Varianz ausgehend von einer Stichprobe und unter Berücksichtigung der Kriterien bedeutet, dass die durchschnittlich quadrierte Abweichung der einzelnen Messwerte 61.576,02 (€²) vom arithmetischen Mittel beträgt.

Aussagen

Ziehen Sie die Wurzel aus der soeben berechneten Varianz, erhalten Sie das Ergebnis für die *Standardabweichung* (siehe die Funktion DBSTDABW() auf Seite 584).

DBVARIANZEN(), VAR.S(), VARIANZA(), VAR.P(), VARIANZENA()

Siehe auch

Das zugehörige Beispiel finden Sie im Ordner \Ms5-235\Kap12 in der Arbeitsmappe DBFUNKTIONEN.xlsx auf dem Arbeitsblatt DBVARIANZ.



DBVARIANZEN() DVARP()

DBVARIANZEN(*Datenbank;Feld;Kriterien*)

Syntax

Die Funktion DBVARIANZEN() berechnet die Varianz einer vollständigen Grundgesamtheit ausgehend von den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen.

Definition

- Argumente** *Datenbank* (erforderlich) ist der Zellbereich, den die Liste oder Datenbank umfasst.
Feld (erforderlich) gibt an, welche Spalte in der Funktion verwendet wird.
Kriterien (erforderlich) bezeichnet den Zellbereich, der die angegebenen Bedingungen enthält.
- Hintergrund** Die Funktion DBVARIANZEN() unterscheidet sich zur Funktion DBVARIANZ() lediglich darin, dass sie die Standardabweichung nicht auf Basis einer Stichprobe, sondern auf Basis einer Grundgesamtheit unter Beachtung bestimmter Kriterien berechnet.
- Hinweis** Mehr Informationen zum Thema »Varianz« finden Sie in Kapitel 11 sowie in der Beschreibung zur Funktion DBVARIANZ() in diesem Kapitel auf Seite 590.
- Praxiseinsatz** Als Inhaber des Großhandelsunternehmens sind Sie immer noch mit der Auswertung Ihrer Excel-Datenbank beschäftigt. Sie haben bereits die Varianz für den Umsatz eines bestimmten Produkts mit der Funktion DBVARIANZ() berechnet. Nun möchten Sie das Gleiche auf Basis einer Grundgesamtheit berechnen.
- Ziel soll es also sein, die Varianz, ausgehend von einer Grundgesamtheit und unter Verwendung bestimmter Kriterien zu errechnen, um anschließend beurteilen zu können, inwieweit die Umsätze für ein bestimmtes Produkt innerhalb eines bestimmten Landes um den Mittelwert streuen.
- Hinweis** Beachten Sie, dass es sich bei den zu untersuchenden Umsätzen um die Grundgesamtheit handelt. Das heißt, die Umsätze die untersucht werden, entsprechen **allen** bisher getätigten Umsätzen. Deshalb liefern die Funktionen DBVARIANZ() und DBVARIANZEN() auch unterschiedliche Ergebnisse.

E10		fx =DBVARIANZEN(Datenbank;D9;B9:C10)		
	B	C	D	E
7	Wie hoch ist die Varianz im Bezug auf den Umsatz für das Produkt Queso Cabrales innerhalb Deutschlands ausgehend von einer Grundgesamtheit?			
8				
9	Land	Artikelname	Umsatz	Ergebnis
10	Deutschland	Queso Cabrales		55978,20

Abbildung 12.33: Die Berechnung der Varianz, ausgehend von einer Grundgesamtheit

Um eine Aussage treffen zu können, fehlt nun noch die Berechnung des dazugehörigen Mittelwerts.

E17		fx =DBMITTELWERT(Datenbank;D16;B16:C17)		
	B	C	D	E
16	Land	Artikelname	Umsatz	Ergebnis
17	Deutschland	Queso Cabrales		1448,27

Abbildung 12.34: Der Mittelwert für den Umsatz des Produkts »Queso Cabrales« innerhalb Deutschlands

- Aussagen** Die folgende Aussage kann nun getroffen werden: Das Ergebnis der Berechnung der Varianz ausgehend von einer Grundgesamtheit und unter Berücksichtigung bestimmter Kriterien bedeutet, dass die durchschnittlich quadrierte Abweichung der einzelnen Messwerte 55.978,20 (€²) vom arithmetischen Mittel beträgt.

Wie schon bei der Funktion DBVARIANZ() gilt auch hier: Ziehen Sie die Wurzel aus der soeben berechneten Varianz, erhalten Sie das Ergebnis für die *Standardabweichung* (ausgehend von einer Grundgesamtheit) – siehe die Funktion DBSTDABWN() auf Seite 586.

DBVARIANZ(), VAR.S(), VARIANZA(), VAR.P(), VARIANZENA()

Das zugehörige Beispiel finden Sie im Ordner \Ms5-235\Kap12 in der Arbeitsmappe DBFUNKTIONEN.xlsx auf dem Arbeitsblatt DBVARIANZEN.

Siehe auch



PIVOTDATENZUORDNEN() GETPIVOTDATA()

PIVOTDATENZUORDNEN(*Datenfeld*; *PivotTable*; *Feld1*; *Element1*; *Feld2*; *Element2*; ...)

Syntax

Die Funktion PIVOTDATENZUORDNEN() gibt Daten aus einem PivotTable-Bericht zurück. Sie können PIVOTDATENZUORDNEN() verwenden, um Datenzusammenfassungen aus einem PivotTable-Bericht abzurufen, vorausgesetzt, die Datenzusammenfassung im Bericht ist sichtbar.

Definition

Sie können eine einfache PIVOTDATENZUORDNEN()-Formel schneller eingeben, indem Sie ein Gleichheitszeichen in die Zelle eintippen, in der Sie den Wert zurückgeben möchten. Anschließend klicken Sie im PivotTable-Bericht auf die Zelle, die die Daten enthält, die Sie zurückgeben möchten.

Tipp

Datenfeld (erforderlich) stellt den Namen in Anführungszeichen für das Datenfeld dar, das die Daten enthält, die Sie abrufen möchten.

Argumente

PivotTable (erforderlich) stellt einen Bezug auf eine Zelle, einen Zellbereich oder einen benannten Zellbereich in einem PivotTable-Bericht dar. Diese Informationen werden dazu verwendet, zu ermitteln, welcher PivotTable-Bericht die Daten enthält, die Sie abrufen möchten.

Feld1, Element1, Feld2, Element2 (optional) stehen für Paare aus Feld- und Elementnamen (zwischen 1 und 14), die die Daten beschreiben, die Sie abrufen möchten. Diese Paare können in einer beliebigen Reihenfolge auftreten. Feld- und Elementnamen, die nicht aus Datumsangaben oder Zahlen bestehen, werden in Anführungszeichen eingeschlossen. Für OLAP-PivotTable-Berichte können Elemente den Quellnamen der Dimension sowie den Quellnamen des Elements enthalten. Ein Paar aus Feld und Element könnte für eine OLAP-PivotTable wie folgt aussehen:

"[Produkt]", "[Produkt].[Alle Produkte].[Lebensmittel].[Backwaren]"

Berechnete Felder oder Elemente und benutzerdefinierte Berechnungen werden von der Funktion PIVOTDATENZUORDNEN() berücksichtigt.

Hinweis

Ist *PivotTable* ein Bereich, der zwei oder mehr PivotTable-Berichte enthält, werden die Daten aus dem in diesem Bereich zuletzt erstellten Bericht abgerufen.

Wenn die Feld- und Elementargumente eine einzelne Zelle beschreiben, wird deren Wert zurückgegeben, unabhängig davon, ob es sich um eine Zeichenfolge, eine Zahl, einen Fehler o.Ä. handelt.

Wenn ein Element eine Datumsangabe enthält, muss der Wert als serielle Nummer angegeben oder mithilfe der Funktion DATUM() gefüllt worden sein, damit der Wert erhalten bleibt, wenn das Arbeitsblatt mit anderen Ländereinstellungen geöffnet wird. Zeitangaben können als Dezimalwerte oder mithilfe der Funktion ZEIT() eingegeben werden.

Ist *PivotTable* kein Bereich, in dem sich ein PivotTable-Bericht befindet, gibt PIVOTDATENZUORDNEN() die Fehlermeldung #BEZUG! zurück.

Wenn die Argumente kein sichtbares Feld beschreiben oder sie ein Seitenfeld umfassen, das nicht angezeigt wird, gibt PIVOTDATENZUORDNEN() die Fehlermeldung #BEZUG! zurück.

Hintergrund

Die PivotTable-Funktion von Excel ist ein mächtiges Instrument zur Datenanalyse. In einer PivotTable lassen sich die Daten einer Datenbank immer wieder anders anordnen und aus verschiedenen Perspektiven betrachten. So können die Daten gruppiert, ausgeblendet, gefiltert und ausgewertet werden, ohne dass man den Datenbestand selber, also die Excel-Tabelle, verändert.

Das Besondere an PivotTables ist, dass Sie das Aussehen der Daten in Sekundenschnelle verändern können. Darüber hinaus bieten PivotTables eine Vielfalt an Layoutfunktionen, sodass Sie Ihre Daten aus den verschiedensten Perspektiven betrachten können.

Sinnvoll ist der Einsatz dieses Mittels nur bei Datenbanken oder Listen, die gleichartige Elemente enthalten. Denn nur dann macht die Zusammenfassung der Daten nach verschiedenen Kriterien die Übersicht über dieselben leichter. Wenn z.B. fünfmal *München* und dreimal *Stuttgart* mit den dazugehörigen Umsatzzahlen vorkommt, lassen sich diese Städte jeweils zusammenfassen, um sich die Anzahl der Bestellungen oder die Summe der Umsätze anzeigen zu lassen. Kämen sowohl *Stuttgart* also auch *München* jeweils nur einmal vor, ergibt eine solche Zusammenfassung nicht viel Sinn, da man die Daten der Tabelle direkt entnehmen kann.

Eine PivotTable stellt die Daten in aller Regel mit Zeilen- und Spaltenüberschriften dar. Sie ist eine Weiterentwicklung der »Kreuztabellen« aus früheren Excel-Versionen.

Mithilfe der Funktion PIVOTDATENZUORDNEN() ist es nun möglich, Daten, die in einem PivotTable-Bericht zusammengefasst wurden, direkt aus dem Bericht herauszulesen. Man kann auf jedes Ergebnis der PivotTable von der aktuellen oder einer anderen Arbeitsmappe Bezug nehmen.

Praxiseinsatz

Wieder endet ein Geschäftsjahr Ihres Großhandelsunternehmens. Sie haben Ihre Datenbank bereits in alle Richtungen ausgewertet und möchten zu guter Letzt noch eine PivotTable erstellen, um daraus mit der Funktion PIVOTDATENZUORDNEN() auf einen Blick zu sehen, mit welchem Produkt welcher Umsatz erzielt wurde.

Als Erstes erstellen Sie deshalb aus den Urdaten über den Befehl *PivotTable* auf der Registerkarte *EINFÜGEN* und unter Angabe des Bereichs *Datenbank* (Urdaten), eine PivotTable entsprechend der Abbildung 12.35.

	A	B	C	D	E	F	G	H
2	Ort	(Alle)						
3	Firma	(Alle)						
4	Land	(Alle)						
5								
6	Summe von Umsatz	Datum						
7	Artikelname	Jan	Feb	Mrz	Apr	Mai	Jun	Jul
8	Alice Mutton	12.223,00 €	7.452,00 €	10.808,00 €	11.135,00 €	14.478,00 €	9.673,00 €	1.928,00 €
9	Aniseed Syrup	2.954,00 €	10.747,00 €	20.229,00 €	11.331,00 €	21.443,00 €	14.165,00 €	
10	Boston Crab Meat	12.307,00 €	12.320,00 €	5.844,00 €	9.265,00 €	17.178,00 €	15.106,00 €	1.855,00 €
11	Camembert Pierrot	21.890,00 €	7.940,00 €	12.308,00 €	14.503,00 €	11.752,00 €	16.767,00 €	4.698,00 €
12	Camaron Tigres	9.986,00 €	8.938,00 €	5.986,00 €	15.200,00 €	9.526,00 €	23.850,00 €	6.370,00 €
13	Chai	15.530,00 €	7.763,00 €	11.687,00 €	11.783,00 €	12.591,00 €	7.574,00 €	4.073,00 €
14	Chang	9.727,00 €	8.034,00 €	3.023,00 €	10.533,00 €	22.458,00 €	13.617,00 €	2.166,00 €
15	Chartreuse verte	9.137,00 €	9.772,00 €	21.558,00 €	9.777,00 €	16.522,00 €	10.766,00 €	4.569,00 €
16	Chef Anton's Cajun Seasoning	12.281,00 €	8.853,00 €	14.697,00 €	14.375,00 €	16.055,00 €	12.356,00 €	
17	Chef Anton's Gumbo Mix	13.714,00 €	6.492,00 €	11.174,00 €	9.034,00 €	13.180,00 €	18.012,00 €	6.756,00 €
18	Chocolade	4.322,00 €	20.424,00 €	11.040,00 €	8.035,00 €	20.094,00 €	7.769,00 €	1.043,00 €

Abbildung 12.35: Die PivotTable aus den Urdaten

Die Anordnung der Daten spielt für das spätere Einsetzen der Funktion PIVOTDATENZUORDNEN() keine Rolle.

Um Ihnen bereits vorab das Ergebnis der Funktion PIVOTDATENZUORDNEN() zu verdeutlichen, vergleichen Sie die Abbildung 12.36.

Chang	Monat	Umsatz
Carnarvon Tigers	1	9.727,00 €
Chai	2	8.034,00 €
Chang	3	3.023,00 €
Chartreuse verte	4	10.533,00 €
Chef Anton's Cajun Seasoning	5	22.458,00 €
Chef Anton's Gumbo Mix	6	13.617,00 €
Chocolade	7	2.166,00 €
Côte de Blaye	8	10.692,00 €
	9	16.200,00 €
	10	15.003,00 €
	11	13.626,00 €
	12	16.820,00 €

Abbildung 12.36: Nach Auswahl eines Produkts wird der Umsatz in den Monaten Januar bis Dezember für alle Länder angezeigt


Ihr Ziel ist es also, sich nach Auswahl eines Ihrer Produkte über das integrierte Kombinationsfeld alle dazugehörigen Umsätze eines Jahrs für alle Länder anzeigen zu lassen.

Nun öffnen Sie ein neues Arbeitsblatt und klicken dort in eine beliebige Zelle. Um zunächst die Funktion PIVOTDATENZUORDNEN() zu testen, schreiben Sie in die markierte Zelle ein Gleichheitszeichen »=« und klicken in eine Zelle, wiederum beliebig, Ihrer zuvor erstellten PivotTable.

Automatisch wird nun die Funktion PIVOTDATENZUORDNEN() angewandt (Abbildung 12.37).

BESTIMMTHEITSMASS		=PIVOTDATENZUORDNEN("Umsatz";Pivot aus Urdaten!\$A\$6;"Datum";1;"Artikelname";"Chai")						
	A	B	C	D	E	F	G	
5								
6	Summe von Umsatz	Datum						
7	Artikelname	Jan	Feb	Mrz	Apr	Mai	Jun	
8	Alice Mutton	12.223,00 €	7.452,00 €	10.808,00 €	11.135,00 €	14.478,00 €	9.673,00 €	
9	Aniseed Syrup	2.954,00 €	10.747,00 €	20.229,00 €	11.331,00 €	21.443,00 €	14.165,00 €	
10	Boston Crab Meat	12.307,00 €	12.320,00 €	5.844,00 €	9.265,00 €	17.178,00 €	15.106,00 €	
11	Camembert Pierrot	21.890,00 €	7.940,00 €	12.308,00 €	14.503,00 €	11.752,00 €	16.767,00 €	
12	Carnarvon Tigers	9.986,00 €	8.938,00 €	5.986,00 €	15.200,00 €	9.526,00 €	23.850,00 €	
13	Chai	15.530,00 €	7.763,00 €	11.687,00 €	11.783,00 €	12.591,00 €	7.574,00 €	
14	Chang	9.727,00 €	8.034,00 €	3.023,00 €	10.533,00 €	22.458,00 €	13.617,00 €	

Abbildung 12.37: Die Funktion PIVOTDATENZUORDNEN() wird durch Klicken in die PivotTable aktiviert

Bestätigen Sie die markierte Zelle, in diesem Fall die Zelle E10, durch Klicken auf die -Taste, erscheint der Wert von 15.530 € in Ihrem neuen Arbeitsblatt, wie in Abbildung 12.38 dargestellt.

E10		=PIVOTDATENZUORDNEN("Umsatz";Pivot aus Urdaten!\$A\$6;"Datum";1;"Artikelname";"Chai")				
	E	F	G	H	I	
10	15530					

Abbildung 12.38: Die Daten der PivotTable wurden der Zelle A1 zugeordnet

Wie sind die Argumente der Funktion PIVOTDATENZUORDNEN() hier zu deuten?

=PIVOTDATENZUORDNEN("Umsatz";'Pivot aus Urdaten'!\$A\$6;"Datum";1;"Artikelname";"Chai")

- ▶ *"Umsatz"*: Stellt das erforderliche Datenfeld dar, dass die Daten enthält, die Sie abrufen wollen – in diesem Fall also den Umsatz
- ▶ *'Pivot aus Urdaten'!\$A\$6*: Stellt den Bezug auf die Zelle A6 (oder auch den Zellbereich) innerhalb der PivotTable dar, um zu definieren, welcher PivotTable-Bericht die Daten enthält, die Sie abrufen möchten
- ▶ *"Datum";1*: Stellt das erste Paar aus Feld- und Elementnamen dar. Das heißt, es soll der Umsatz für den Datumswert Januar zurückgegeben werden. Die Zahl 1 steht für den Monat Januar, die Zahl 2 für Februar usw. (siehe Abbildung 12.37 auf Seite 595).
- ▶ *"Artikelname";"Chai"*: Stellt das zweite Paar aus Feld- und Elementnamen dar. Es wird also festgelegt, dass der berechnete Umsatz für das Feld *Artikelname* und dessen Element *Chai* zurückgegeben werden soll.

Sie haben sich nun einen ersten Überblick über die Funktionsweise von PIVOTDATENZUORDNEN() verschafft und machen sich ans Werk, Ihre Umsatzübersicht zu erstellen.

Dazu kopieren Sie zunächst alle Artikelnamen aus der Urdatenliste und fügen Sie in das neu angelegte Arbeitsblatt ein, wie in Abbildung 12.39 dargestellt.

Artikelnamen aus Urdaten
Alice Mutton
Aniseed Syrup
Boston Crab Meat
Camembert Pierrot
Carnarvon Tigers
Chai
Chang
Chartreuse verte
Chef Anton's Cajun Seasoning
Chef Anton's Gumbo Mix
Chocolate
Côte de Blaye
Escargots de Bourgogne
Filo Mix
Flotemysost
Geitost

Abbildung 12.39: Die kopierten Artikelnamen aus den Urdaten

Je nachdem, ob Sie mit Excel 2007 oder mit Excel 2010/2013 arbeiten, gehen Sie nun folgendermaßen vor, um ein Kombinationsfeld in die Arbeitsmappe einzufügen:

Kombinationsfeld in Excel 2007

Wird bei Ihnen die Registerkarte *Entwicklertools* nicht angezeigt, müssen Sie diese zunächst aktivieren.

Klicken Sie auf die *Office*-Schaltfläche und anschließend auf die Schaltfläche *Excel-Optionen*. Aktivieren Sie danach in der Kategorie *Häufig verwendet* das Kontrollkästchen *Entwickleregisterkarte in der Multifunktionsleiste anzeigen* und klicken Sie auf *OK*.

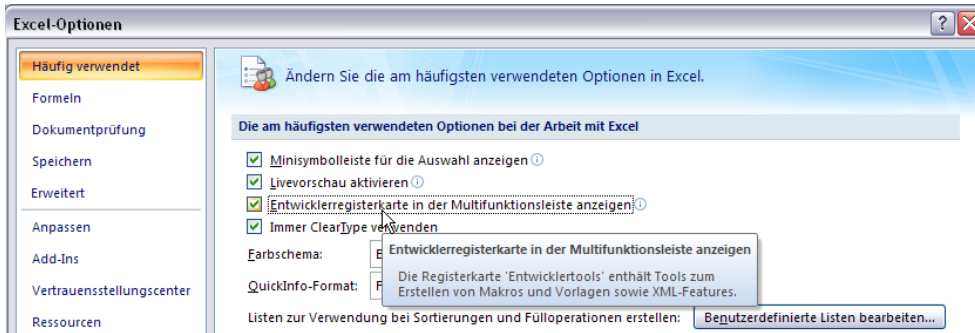


Abbildung 12.40: Aktivieren Sie die Entwicklertools in 2007

Klicken Sie auf die Registerkarte *DATEI* und anschließend auf *Optionen*. Wählen Sie die Kategorie *Menüband anpassen* und aktivieren Sie anschließend das Kontrollkästchen *Entwicklertools* im rechten Listenfeld *Hauptregisterkarten*. Bestätigen Sie mit *OK*.

Kombinationsfeld in Excel 2010/2013

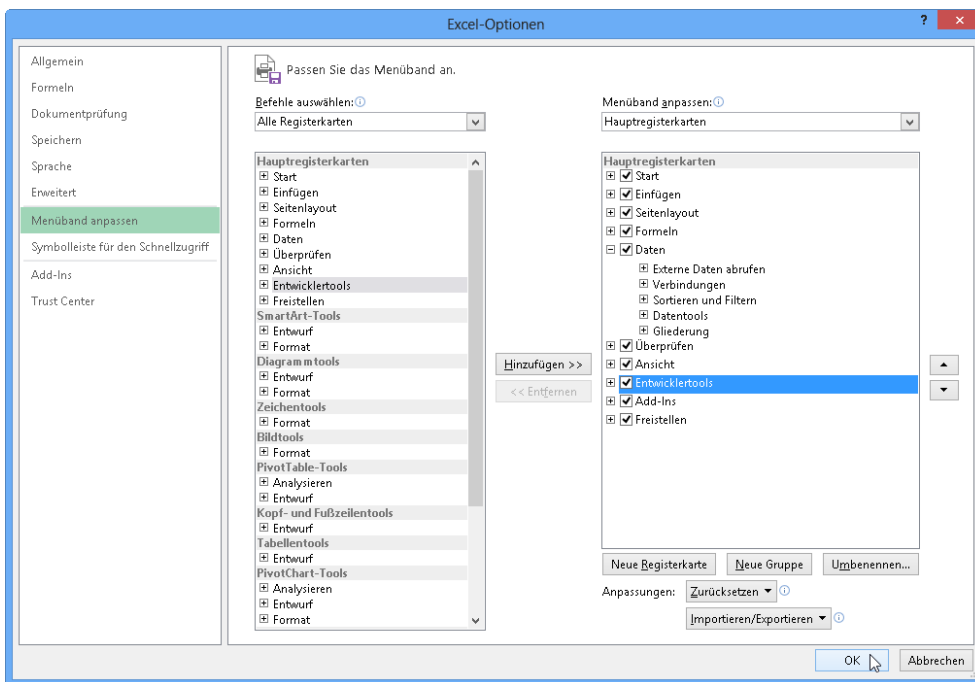


Abbildung 12.41: Aktivieren Sie die Entwicklertools in 2010/2013

Sie sehen nun im Menüband die Registerkarte *ENTWICKLERTOOLS*. Klicken Sie hier in der Gruppe *Steuerelemente* auf die Schaltfläche *Einfügen* und fügen Sie das Kombinationsfeld ein.

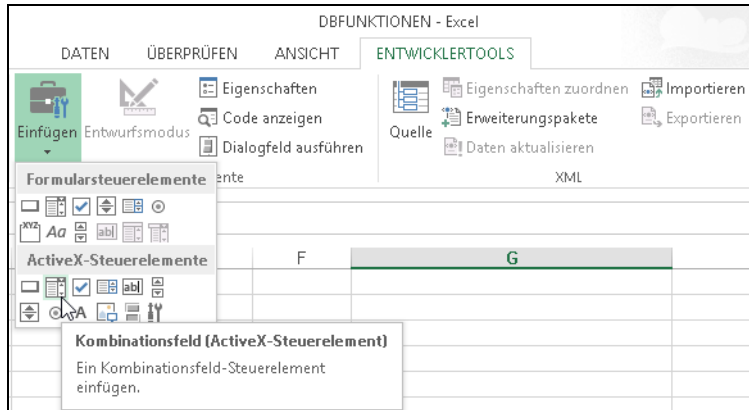


Abbildung 12.42: Die Entwicklertools stellen Formularfelder bereit

Es öffnet sich die entsprechende Symbolleiste. Anschließend ziehen Sie über die Symbolleiste *Formular* ein *Kombinationsfeld* in das Arbeitsblatt und ordnen diesem in den Eigenschaften (zu öffnen über einen Doppelklick) unter *Eingabebereich* die zuvor in das Arbeitsblatt kopierten Artikelnamen und unter *Zellverknüpfung* eine beliebige leere Zelle im Arbeitsblatt zu.

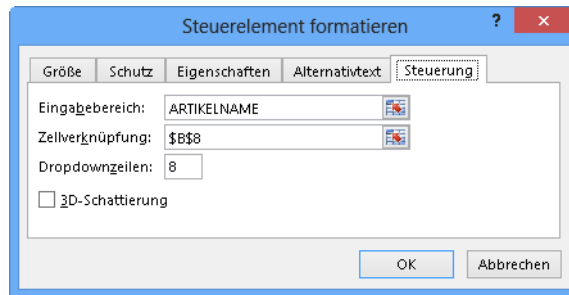


Abbildung 12.43: Dem Kombinationsfeld Eigenschaften zuweisen

Hinweis In unserem Beispiel haben wir den Artikelnamen einen *Namen* zugeordnet. Daher wird in Abbildung 12.43 nicht ein Zellbereich, also beispielsweise von *H12:H88* angegeben, sondern der Name *Artikelname*. Beides ist natürlich möglich.

Klicken Sie nun auf den Dropdownpfeil des Kombinationsfelds, stehen bereits sämtliche Artikelnamen zur Auswahl zur Verfügung. Die zuvor in den Eigenschaften des Kombinationsfelds angegebene Zellverknüpfung gibt die Stelle wieder, an der sich der ausgewählte Eintrag befindet.

	A	B	C
8			6
9		Chai	

Abbildung 12.44: Der Eintrag »Chai« im Kombinationsfeld befindet sich an sechster Stelle

Aus dieser Zahl wollen Sie nun eine Art Überschrift für Ihre Auswertung generieren. Das erreichen Sie mit der Funktion INDEX().

Klicken Sie wieder in eine leere Zelle Ihres Tabellenblatts und definieren Sie die Argumente der Funktion INDEX() wie in der folgenden Abbildung 12.45 dargestellt.

	A	B	C
7		Chai	
8			6
9		Chai	


Abbildung 12.45: Erstellen Sie über die Funktion INDEX() eine Art Überschrift für Ihre Auswertung

Folgende Argumente sind hier angegeben:

- ▶ *Matrix* = Liste der Artikelnamen
- ▶ *Zeile* = B8 markiert die Zeile bzw. Zelle, aus der der Wert geliefert werden soll
- ▶ *Spalte* = Das Argument *Spalte* muss in diesem Fall nicht angegeben werden, da die Zeile als Argument bereits benannt wurde. Wäre *Zeile* nicht definiert, müsste *Spalte* angegeben werden.

Egal, welchen Artikel Sie nun aus dem Kombinationsfeld wählen, wird dieser als eine Art Überschrift angezeigt.

Um jetzt eine Umsatzanzeige für die einzelnen Monate des ausgewählten Produkts zu erstellen, legen Sie, wie in Abbildung 12.36 auf Seite 595 dargestellt, eine zweisepaltige Tabelle mit den Spaltennamen *Monat* und *Umsatz* an. Tragen Sie in die Spalte *Monat* die Zahlen 1 bis 12 ein und klicken Sie anschließend in die Umsatzzelle für den Monat 1, sprich Januar.

Tragen Sie, wie vorhin bereits beschrieben, in diese Zelle ein Gleichheitszeichen ein und klicken Sie in der PivotTable auf eine Zelle unter dem Monat *Januar*. Bestätigen Sie mit der -Taste. Vergleichen Sie dazu die Abbildung 12.37 auf Seite 595 und die Abbildung 12.38 auf Seite 595.

Für welches Produkt Sie sich in diesem Schritt entscheiden, ist egal, da der Bezug zu dem gewünschten Produkt nicht direkt aus der PivotTable hergestellt werden soll, sondern über das Kombinationsfeld.

Hinweis

Der erste Eintrag in der Umsatzzelle für den Monat *Januar* stellt sich wie in Abbildung 12.46 gezeigt dar.

	A	B	C	D	E	F	G
10		Monat	Umsatz				
11		1	12.223,00 €				
12		2					

Abbildung 12.46: Der Umsatz für das Produkt »Alice Mutton« im Januar

Nun müssen Sie nur noch die Formel für die Funktion PIVOTDATENZUORDNEN() etwas ändern, und Ihr Umsatzcockpit ist fertig.

Da Sie für den Monat Januar nicht immer den Umsatz des Produkts bzw. des Artikelnamens *Alice Mutton* angezeigt bekommen wollen, sondern die Ausgabe davon abhängig machen wollen, welchen Artikelnamen Sie in dem Kombinationsfeld ausgewählt haben, müssen Sie den Zellbezug ändern.

Deshalb löschen Sie den Elementnamen *Alice Mutton* samt Anführungsstrichen aus der Formel und ersetzen es durch die Zelle, die Sie zur Überschriftenzelle Ihrer Auswertung gemacht haben.

In der folgenden Abbildung 12.47 wäre das die Zelle B7.

Wichtig Drücken Sie nach Eingabe der Zelle die Taste **[F4]**, um einen *absoluten* Bezug herzustellen, und bestätigen Sie anschließend mit der **[↵]**-Taste.

Monat	Umsatz
1	15.530,00 €

Abbildung 12.47: Die Umsatzausgabe wird davon abhängig gemacht, welches Produkt im Kombinationsfeld ausgewählt ist

Wie Sie in der Formel der Abbildung 12.47 erkennen können, wird der Feldname *"Datum"* durch den Elementnamen *B12* ergänzt.

Nun müssen Sie nur noch die in Zelle C12 bereits eingetragene Formel bis zum Monat Dezember herunterziehen und erhalten auf diese Weise für alle Monate den entsprechenden Umsatz – abhängig von dem im Kombinationsfeld ausgewählten Produkt. Vergleichen Sie dazu die Abbildung 12.36 auf Seite 595.

Wie in diesem Beispiel dargestellt, können Sie mit der Funktion **PIVOTDATENZUORDNEN()** auf einfache Art und Weise übersichtliche und einfach zu handhabende Cockpits bzw. Auswertungen Ihrer Datenbank erstellen und haben alle Umsätze »per Knopfdruck« parat.

Siehe auch Keine



Das zugehörige Beispiel finden Sie im Ordner `\\Ms5-235\Kap12` in der Arbeitsmappe `DBFUNKTIONEN.xlsx` auf dem Arbeitsblatt `PIVOTDATENZUORDNEN`.

Kapitel 13

Cubefunktionen

CUBEELEMENT()	604
CUBEELEMENTEIGENSCHAFT()	607
CUBEKPIELEMENT()	608
CUBEMENGE()	610
CUBEMENGENANZAHL()	612
CUBERANGELEMENT()	612
CUBEWERT()	614



Cubefunktionen gibt es seit Excel 2007. Sie werden immer dann eingesetzt, wenn eine Datenverbindung zu einem Cube¹ der Analysis Services von SQL Server (ab Version 2005) innerhalb einer Arbeitsmappe vereinbart wurde. Das ist unter anderem dann der Fall, wenn der Anwender eine PivotTable eingerichtet hat, deren Datenherkunft eben dieser Cube ist. Ein Cube unterscheidet sich von aufbereiteten Abfragen, wie sie aus Access oder SQL Server gewonnen werden können dadurch, dass in ihm bereits verschiedene Gruppierungen der Daten in Hierarchien vorgenommen wurden und für diese Kennzahlen (Measures) berechnet und im Cube gespeichert wurden. Das erzeugt beim Anwender zwei Vorteile: Die ihn interessierenden Daten liegen ohne weitere rechnerische oder gar kompliziertere mathematische Vorbereitungen entsprechend aufbereitet vor und den Hauptteil der Zeit zum Berechnen der Auswertungen hat der Server bereits verbracht, nur ein Bruchteil dieser Arbeit verbleibt für Excel. Diese Vorteile haben aber auch u.U. eine Schattenseite: Der Einsatz von berechneten Feldern/Elementen für die PivotTable selbst ist nicht mehr möglich, die entsprechende Schaltfläche bleibt inaktiv².

Der Einsatz der Cubefunktionen setzt eine von zwei (technischen) Bedingungen voraus:

- ▶ beim Einrichten der Arbeitsmappe besteht eine Verbindung zu den Analysis Services eines SQL Servers oder
- ▶ die Daten liegen in einem Offline-Cube (also ohne Verbindung zum Server) im Dateisystem des Anwenders bereit.

Diese Voraussetzungen schränken sicher den Nutzerkreis der Funktionen erheblich ein. Damit Sie dennoch etwas experimentieren können, finden Sie innerhalb der Begleitdateien zum Buch einen Offline-Cube und Datenverbindungsdateien für das in Kapitel 1 beschriebene Beispiel.



Abbildung 13.1: In begleiteten Schritten zum Offline-Cube

Tip Ein solcher Offline-Cube (Dateiendung `.cub`) kann mithilfe von Excel erstellt und auch später in seinem Aufbau modifiziert werden. Zu beachten ist, dass bei großen Datenmengen zur Erstellung u.U. viel Zeit und Speicherplatz auf dem lokalen Computer benötigt werden.

1 Genauer kann man von OLAP-Cubes sprechen, wobei OLAP für *Online Analytical Processing* steht.

2 Excel 2013 stellt als Ersatz die Möglichkeit bereit, die Auswertung selbst definierter MDX-Ausdrücke ins Spiel zu bringen. Diese arbeiten wohl nur mit Cubes, die direkt vom SQL Server kommen, und nicht mit solchen, die in den Beispieldateien zu diesem Buch als Unterstützung beigefügt wurden.

Besteht eine Verbindung zu den Analysis Services, die über MS Query (*DATEN/Externe Daten abrufen/Aus anderen Quellen/Aus Microsoft Query*³) oder den Datenverbindungs-Assistenten (*EINFÜGEN/PivotTable/Externe Datenquelle verwenden*) eingerichtet wurde, gelingt nach dem Anlegen einer PivotTable über die Schaltfläche aus Abbildung 13.1 der Übergang in ein Dialogfeld. In diesem wird Schritt für Schritt das Anlegen des Cubes begleitet.

Wollen Sie die Datenverbindungsdateien (Endung *.odc* für den Zugriff über Arbeitsmappen-Verbindungen und *.oqy* für den Zugriff über MS Query) aus den Beispieldateien zum Zugriff auf die Beispiel-Cubedatei benutzen, so sind diese jedoch vor ihrem Einsatz anzupassen. Das liegt daran, dass im allgemeinen Pfade zu Datenbanken stets absolut angegeben werden. Mithilfe des Windows-Editors passen Sie also etwas wie

```
Data Source='X:\Dokumente\Kap13\Buch\CubeTest.cub';  
Location='X:\Dokumente\Kap13\Buch\CubeTest.cub';
```

entsprechend der von Ihnen verwendeten Pfade an. Die vorbereitete Arbeitsmappe dient als Muster für mögliche Berechnungen. Um unnötige weitere Anpassungen dieser Mappe zu vermeiden, gestalten Sie am besten Ihre eigene Auswertungsmappe in folgenden Schritten:⁴

- ▶ *EINFÜGEN/PivotTable/Externe Datenquelle verwenden* (hier können Sie nach weiteren Elementen suchen und die vorbereiteten Datenverbindungsdateien oder die *cub*-Datei selbst einsetzen)
- ▶ Anlegen des Layouts und Erfassen der zahlenmäßigen Inhalte der Datenquelle
- ▶ Einsatz der Cubefunktionen

In der Excel-StandardEinstellung ist beim Öffnen einer Mappe mit Datenverbindungen die Verwendung dieser Verbindung in der Statusleiste explizit zu erlauben. Nutzen Sie Excel ab Version 2010, wird das Dokument nach der Aktivierung vertrauenswürdig und die Aktivierung muss so lange nicht mehr bestätigt werden, bis im Sicherheitscenter bzw. Trust Center die Vertrauenswürdigkeit aller Dokumente zurückgesetzt wird.

Nicht immer ist es der Anwender, der manuell Cubefunktionen ins Arbeitsblatt einfügt. Verwenden Sie den Befehl *In Formeln konvertieren* aus Abbildung 13.1, wandelt Excel einen Teil der PivotTable – auf Wunsch auch die gesamte – in eine Tabelle um, die zunächst unformatiert ist, aber den gleichen Inhalt wie die PivotTable hat. Der Vorteil⁵ dabei ist, dass das gesamte Layout, also die Anordnung von Spalten und Zeilen fixiert wird. Auch die Filter können in diese Fixierung mit eingeschlossen werden.

Hinweis

Hat man das Prinzip der Formeln erkannt, gelingt es, Pivot-ähnliche Strukturen vollkommen individuell aufzubauen und somit flexibel gestaltete Auswertungen von Daten zu schaffen.

Die Detailbeschreibung der Funktionen bezieht sich in allen Fällen auf das in Abbildung 13.2 dargestellte Beispiel des Umsatzes von Süßwaren (Schokolade und Kekse, zusammengefasst in der Dimension *Produkte*). Dieser Umsatz wird im Cube erfasst als Kennzahlengruppe *Umsatz* mit den Kennzahlen *Umsatz*, *BruttoUmsatz* und *Umsatz Count* in zwei Filialgruppen zu je zwei Filialen (zusammengefasst in der Dimension *Filialen*) in drei aufeinander folgenden Jahren

³ Hier gibt es ein Dialogfeld, welches hilft, OLAP-Cubes abzurufen.

⁴ Ganz schnell gelingt es, eine Mappe zu erstellen, wenn Sie auf die *.odc*-Datei im Windows-Explorer doppelt klicken. Dann sollte sich Excel ab Version 2007 öffnen, die Datenverbindung ist von Ihnen zu akzeptieren und los geht es mit dem Layout der PivotTable.

⁵ Je nach Ausgangslage wird das eventuell der eine oder andere Nutzer auch als Nachteil empfinden.

(zusammengefasst in der Dimension *Jahre*). Damit bleiben die Formeldetails übersichtlich, die Reichweite der Möglichkeiten (detaillierte Monatsumsätze für Produktgruppen mit zahlreichen Produkten und Filialen, die über ein ganzes Land bis auf den einzelnen Verkäufer strukturiert sind) wird vorstellbar.

GruppenName	FilialName	Jahr	Produkt	Umsatz
Nord	NordOst	2010	Schokolade	1000
Nord	NordOst	2010	Kekse	1500
Nord	NordWest	2010	Schokolade	800
Nord	NordWest	2010	Kekse	1700
Süd	SüdOst	2010	Schokolade	1200
Süd	SüdOst	2010	Kekse	1100
Süd	SüdWest	2010	Schokolade	1300
Süd	SüdWest	2010	Kekse	1100
Nord	NordOst	2011	Schokolade	1040
Nord	NordOst	2011	Kekse	1560
Nord	NordWest	2011	Schokolade	824
Nord	NordWest	2011	Kekse	1751
Süd	SüdOst	2011	Schokolade	1260
Süd	SüdOst	2011	Kekse	1155
Süd	SüdWest	2011	Schokolade	1352
Süd	SüdWest	2011	Kekse	1144
Nord	NordOst	2012	Schokolade	1081
Nord	NordOst	2012	Kekse	1622
Nord	NordWest	2012	Schokolade	848
Nord	NordWest	2012	Kekse	1803
Süd	SüdOst	2012	Schokolade	1323
Süd	SüdOst	2012	Kekse	1212
Süd	SüdWest	2012	Schokolade	1406
Süd	SüdWest	2012	Kekse	1189

Abbildung 13.2: Der Umsatz von Süßwaren in verschiedenen Filialen

Tabelle 13.1
Cubefunktionen
in der Übersicht

Funktion	Beschreibung
CUBEELEMENT()	Gibt ein Element (eine Zelle) aus einem Cube zurück
CUBEELEMENTEIGENSCHAFT()	Gibt den Wert einer Elementeigenschaft aus dem Cube zurück
CUBEKPIELEMENT()	Gibt die Eigenschaft eines sog. Key Performance Indicators (KPI, im Cube definierte Leistungskennzahl) zurück
CUBEMENGE()	Gibt eine berechnete Menge (Set oder Satz) von Elementen zurück
CUBEMENGENANZAHL()	Gibt die Anzahl der Elemente aus einer Elementmenge (Satz von Elementen) zurück
CUBERANGELEMENT()	Gibt das <i>n</i> -te Element einer geordneten Menge zurück
CUBEWERT()	Gibt den Wert eines Elements (Zelle) aus einem Cube zurück

CUBEELEMENT() CUBEMEMBER()

Syntax CUBEELEMENT(*Verbindung;Element_Ausdruck;Beschriftung*)

Definition Die Funktion gibt ein Element (eine Zelle) aus einem Cube zurück. Sie kann verwendet werden, um die Existenz eines solchen Elements zu prüfen und das gefundene Element an andere Funktionen mittels eines Zellbezugs weiterzugeben.

Argumente *Verbindung* (erforderlich) ist eine Zeichenkette mit dem Namen der Arbeitsmappenverbindung zum Cube. Nach dem Schreiben des ersten Anführungszeichens werden vorhandene Datenverbindungen kontextsensitiv eingeblendet (Abbildung 13.3).

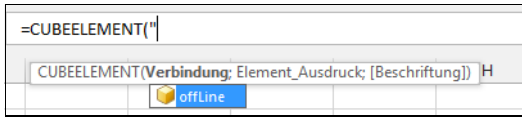


Abbildung 13.3: Kontextsensitivität beim Schreiben von Formeln ist ein starkes Hilfsmittel

Element_Ausdruck (erforderlich) definiert die Lage des Elements im Cube durch die Angabe eines multidimensionalen Ausdrucks (MDX⁶). Dieser kann direkt eingegeben werden oder sich in einer Zelle befinden, auf die dann Bezug genommen wird. Die Verwendung von Tupeln im Ausdruck ist erlaubt. Auch beim Schreiben der Elemente reagiert die Eingabezeile kontextsensitiv.

Beschriftung (optional) ersetzt bei der Anzeige des Ergebnisses in der Excel-Zelle die eventuell vorhandene Beschriftung des Elements im Cube. Wurde ein Tupel verwendet, liefert die Funktion die (sofern vorhanden) Beschriftung des letzten Elements im Tupel.

In der Zelle, die die Funktion enthält, wird während der Datenabfrage vorübergehend die Meldung *#DATEN_ABRUFEN...* angezeigt.

Hintergrund

Wird CUBEELEMENT() als Argument für eine andere Cubefunktion verwendet, wird der MDX-Ausdruck im Argument und nicht der angezeigte Wert verwendet.

Mögliche Fehlerwerte und Meldungen geben darüber Auskunft, was u.U. nicht beachtet wurde oder warum ein Element nicht gefunden werden kann⁷:

- ▶ Steht der OLAP-Server (oder der Offline-Cube) nicht zur Verfügung, erfolgt eine Fehlermeldung. Der Zellinhalt der betroffenen Zellen ändert sich nicht.
- ▶ Wenn *Element_Ausdruck* mehr als 255 Zeichen lang ist⁸, gibt CUBEELEMENT() den Fehlerwert *#WERT!* zurück
- ▶ CUBEELEMENT() gibt unter folgenden Bedingungen den Fehlerwert *#NV* zurück:
 - ▶ Der Verbindungsname entspricht keiner gültigen Arbeitsmappenverbindung⁹
 - ▶ Die *Element_Ausdruck*-Syntax enthält Fehler
 - ▶ Das durch die MDX-Abfrage angegebene Element ist nicht im Cube vorhanden
 - ▶ Für die angegebenen Werte im Tupel gibt es keine gemeinsame Schnittmenge¹⁰
 - ▶ Der Menge der angeforderten Elemente enthält mindestens ein Element mit einer anderen Dimensionalität als die der anderen Elemente
 - ▶ CUBEELEMENT() kann den Fehlerwert *#NV* zurückgeben, wenn die Verbindung zur Datenquelle abgerissen ist und nicht neu aufgebaut werden kann

⁶ MDX steht für *Multidimensional Expressions*.

⁷ Wegen der Kontextsensitivität während der Eingabe der Formeln lassen sich die meisten Fehler vermeiden.

⁸ Diese Beschränkung für ein Argument oder eine Funktion kann dadurch umgangen werden, dass die benötigten Informationen als Werte in Zellen stehen, auf die in der Funktion selbst Bezug genommen wird.

⁹ Werden bei Angabe des Namens die Anführungsstriche vergessen, so erscheint hier wie auch bei allen anderen Cubefunktionen der Fehlerwert *#NAME?*.

¹⁰ Stellt man sich unter einem Cube in der Tat einen mehrdimensionalen Würfel vor, geben die Ausdrücke im Tupel die »Marschrichtung« entlang jeder Achse vor. Ziel des Tupels muss es sein, mindestens einen gemeinsamen Schnittpunkt zu finden.

Praxiseinsatz Zum besseren Verständnis des Einsatzes der Funktionen in diesem Kapitel schauen Sie sich die PivotTable unseres Beispiels an (Abbildung 13.4).

	A	B	C	D
1	Jahr	All		
2				
3	Umsatz	Spaltenbeschriftungen		
4	Zeilenbeschriftungen	Kekse	Schokolade	Gesamtergebnis
5	Nord	9936	5593	15529
6	NordOst	4682	3121	7803
7	NordWest	5254	2472	7726
8	Süd	6900	7841	14741
9	SüdOst	3467	3783	7250
10	SüdWest	3433	4058	7491
11	Gesamtergebnis	16836	13434	30270

Abbildung 13.4: Umsätze an Süßwaren

Und so liefert die Formel, die eine einzelne Zelle sucht,

```
=CUBELEMENT("offLine"; "[Produkte].[Produkt].[A11].[Kekse]")
```

im Ergebnis das Element *Kekse*, das die gleiche Beschriftung hat. Nutzen Sie ein Tupel in der Form

```
=CUBELEMENT("offLine"; ([Filialen].[Filiale].[A11].[NordOst], [Produkte].[A11].[Kekse], [Jahre].[2011]))
```

entsteht im Ergebnis der Anzeige *2011* (der Inhalt besteht aus den Informationen zum Verkauf von Keksen im Jahre 2011 durch die Filiale NordOst) und versuchen Sie mit

```
=CUBELEMENT("offLine"; ([Filialen].[Gruppe].[A11].[Nord], [Filialen].[Filiale].[A11].[NordOst]))
```

einen leeren Schnittpunkt zu finden, wird das mit dem Fehler *#NV* quittiert.

Statt des englischen *All*, welches aus

```
=CUBELEMENT("offLine"; "[Produkte].[Produkt].[A11]")
```

entsteht, lässt sich mit

```
=CUBELEMENT("offLine"; "[Produkte].[Produkt].[A11]"; "Gesamtergebnis")
```

das Wort *Gesamtergebnis* anzeigen. Wichtig ist es nun zu verstehen, dass das, was in der Zelle mit der Formel angezeigt wird, genutzt werden kann, um Zellbeschriftungen zu erstellen. Der wirkliche Inhalt der Zelle ist viel informativer, wie der Einsatz der Funktion *CUBEWERT()* beweisen wird, wenn diese auf Zellen mit *CUBELEMENT()*-Einträgen Bezug nimmt.

Siehe auch Alle Cubefunktionen, *PIVOTDATENZUORDNEN()*

CUBEELEMENTEIGENSCHAFT()

CUBEMEMBERPROPERTY()

CUBEELEMENTEIGENSCHAFT(*Verbindung*; *Element_Ausdruck*; *Eigenschaft*)

Die Funktion gibt den Wert einer Elementeigenschaft aus dem Cube zurück. Damit kann ebenso geprüft werden, ob das Element im Cube vorhanden ist. Im Erfolgsfall wird die Eigenschaft als Wert der Excel-Zelle angezeigt.

Verbindung (erforderlich) ist eine Zeichenkette mit dem Namen der Arbeitsmappenverbindung zum Cube. Nach dem Schreiben des ersten Anführungszeichens werden vorhandene Datenverbindungen kontextsensitiv eingeblendet (siehe Abbildung 13.3 auf Seite 605).

Element_Ausdruck (erforderlich) definiert die Lage des Elements im Cube durch die Angabe eines multidimensionalen Ausdrucks (MDX). Dieser kann direkt eingegeben werden oder sich in einer Zelle befinden, auf die dann Bezug genommen wird. Die Verwendung von Tupeln im Ausdruck ist erlaubt. Auch beim Schreiben der Elemente reagiert die Eingabezeile kontextsensitiv.

Eigenschaft (erforderlich) ist der Name der Eigenschaft, deren Wert abgefragt wird.

Ob ein Element Eigenschaften hat, kann mithilfe der OLAP-Tools einer PivotTable ermittelt werden, die ihre Daten aus dem gegebenen Cube bezieht (Abbildung 13.5).

Syntax

Definition

Argumente

Hintergrund

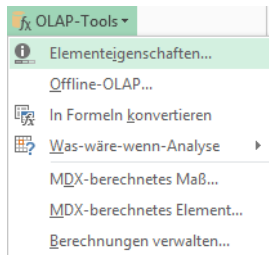


Abbildung 13.5: Nicht alle Elemente haben Eigenschaften

Im vorliegenden Beispiel (Abbildung 13.4 auf Seite 606) sind es die Filialen, die durch ihre Zugehörigkeit zu einer Gruppe die Eigenschaft *Gruppe* mit den möglichen Werten *Nord* oder *Süd* haben.

In der Zelle, die die Funktion enthält, wird während der Datenabfrage vorübergehend die Meldung *#DATEN_ABRUFEN...* angezeigt.

Mögliche Fehlerwerte und Meldungen geben darüber Auskunft, was u.U. nicht beachtet wurde oder warum ein Element nicht gefunden werden kann:

- ▶ Steht der OLAP-Server (oder der Offline-Cube) nicht zur Verfügung, erfolgt eine Fehlermeldung. Der Zelleninhalt der betroffenen Zellen ändert sich nicht.
- ▶ Wenn der Verbindungsname keiner gültigen Arbeitsmappenverbindung entspricht, gibt CUBEELEMENTEIGENSCHAFT() den Fehlerwert *#NV* zurück

- ▶ Wenn die Syntax von *Element_Ausdruck* nicht korrekt ist oder das hierdurch angegebene Element nicht im Cube vorhanden ist, gibt CUBELEMENTEIGENSCHAFT() den Fehlerwert #NV zurück
- ▶ CUBELEMENTEIGENSCHAFT() kann den Fehlerwert #NV zurückgeben, wenn die Verbindung zur Datenquelle abgerissen ist und nicht neu aufgebaut werden kann

Praxiseinsatz Wie bereits erwähnt, haben nur die Filialen in der PivotTable unseres Beispiels (Abbildung 13.4 auf Seite 606) eine Eigenschaft, diese heißt *Gruppe*. Und damit liefert

```
=CUBELEMENTEIGENSCHAFT("offLine";"[Filialen].[Filiale].[A11].[NordOst]";"Gruppe")
```

den Wert *Nord* und

```
=CUBELEMENTEIGENSCHAFT("offLine";"[Filialen].[Filiale].&[3]";"Gruppe")
```

den Wert *Süd*. In der letzten Formel wurde statt des Namens der Filiale die Nummer ihres Auftretens in der Liste eingesetzt¹¹.

Siehe auch Alle Cubefunktionen, PIVOTDATENZUORDNEN()

CUBEKPIELEMENT()



CUBEKPIMEMBER()

Syntax CUBEKPIELEMENT(*Verbindung*;KPI_Name;KPI_Eigenschaft;Beschriftung)

Definition Diese Funktion gibt die Eigenschaft eines sogenannten Key Performance Indicators (KPI, Leistungskennzahl) zurück und zeigt dessen KPI-Namen in der Zelle an¹².

Argumente *Verbindung* (erforderlich) ist eine Zeichenkette mit dem Namen der Arbeitsmappenverbindung zum Cube. Nach dem Schreiben des ersten Hochkommata werden vorhandene Datenverbindungen kontextsensitiv eingeblendet (Abbildung 13.3 auf Seite 605).

KPI_Name (erforderlich) definiert den Namen des KPI im Cube.

Ein KPI besteht aus mehreren Komponenten (je nach Definition des KPI auf dem Server). Diese werden durch das erforderliche Argument *KPI_Eigenschaft* mithilfe einer Ziffer abgefragt. Diese Ziffer ergibt sich aus Tabelle 13.2.

Tabelle 13.2
Die Bedeutung der Ziffern für das dritte Argument der Funktion

Argument	Zu verwendender MDX-Ausdruck	Beschreibung
1	[KPI-Name Value]	Auf dem Server berechneter Wert
2	[KPI-Name Goal]	Hinterlegter Zielwert
3	[KPI-Name Status]	Der auf einen Zeitpunkt bezogene Zustand
4	[KPI-Name Trend]	Über einen Zeitraum berechnete Werte
5	[KPI-Name Weight]	Information über die Bedeutung des KPI
6	[KPI-Name CurrentTimeMember]	Zeitliche Einordnung des KPI

¹¹ Eine kurze Einführung in die Syntax von MDX finden Sie unter anderem in *Business Intelligence und Reporting mit Microsoft SQL Server 2008* (Autorenteam, erschienen bei Microsoft Press, 2009, ISBN-13: 978-3-86645-657-0).

¹² Diese Indikatoren müssen durch Projekte auf den Analysis Services von Microsoft SQL Server ab Version 2005 erstellt werden und definieren u.a. quantifizierbare Größen, mit deren Hilfe Unternehmensleistungen überwacht werden können.

Beschriftung (optional) ersetzt bei der Anzeige des Ergebnisses in der Excel-Zelle die eventuell vorhandene Beschriftung der KPI-Komponente im Cube.

In der Zelle, die die Funktion enthält, wird während der Datenabfrage vorübergehend die Meldung *#DATEN_ABRUFEN...* angezeigt.

Hintergrund

Mögliche Fehlerwerte und Meldungen geben darüber Auskunft, was u.U. nicht beachtet wurde oder warum ein Element nicht gefunden werden kann:

- ▶ Steht der OLAP-Server (oder der Offline-Cube) nicht zur Verfügung, erfolgt eine Fehlermeldung. Der Zelleninhalt der betroffenen Zellen ändert sich nicht.
- ▶ Wenn der Verbindungsname keiner gültigen Arbeitsmappenverbindung entspricht, gibt CUBEKPIELEMENT() den Fehlerwert *#NV* zurück
- ▶ CUBEKPIELEMENT() gibt den Fehlerwert *#NV* zurück, wenn *KPI_Name* oder *KPI_Eigenschaft* ungültig ist
- ▶ CUBEKPIELEMENT() kann den Fehlerwert *#NV* zurückgeben, wenn die Verbindung zur Datenquelle abgerissen ist und nicht neu aufgebaut werden kann

Sie können die Funktion mit der Funktion CUBEWERT() kombinieren. Dazu geben Sie Letzterer als zweites Argument die CUBEKPIELEMENT()-Funktion oder einen Zellbezug auf eine Zelle mit der CUBEKPIELEMENT()-Funktion mit.

Im durchgehenden Beispiel zu diesem Kapitel wurde im Cube ein KPI namens *Durchschnitt* hinterlegt. Dieser berechnet den Mittelwert aus dem Gesamtumsatz und der Anzahl der Umsätze ohne Nachkommastellen. Beide Größen liegen auch als Kennziffern (Measures) im Cube vor, lassen sich aber in der PivotTable nicht zum Berechnen von Feldern einsetzen (die Option berechneter Felder fehlt bei Cube-basierten Daten). Der Zielwert (Goal) beläuft sich ohne weitere Unterscheidungen der Einfachheit halber auf 1.500 €. Die Abbildung 13.6 zeigt die Details im Falle von Keksen.

Praxiseinsatz

	A	B	C	D
1	Jahr	All		
2				
3		Spaltenbeschriftungen		
4		Umsatz	Durchschnitt	Durchschnitt Ziel
5	Zeilenbeschriftungen	Kekse	Kekse	Kekse
6	Nord	9936	1656	1500
7	NordOst	4682	1561	1500
8	NordWest	5254	1751	1500
9	Süd	6900	1150	1500
10	SüdOst	3467	1156	1500
11	SüdWest	3433	1144	1500

Abbildung 13.6: Der KPI *Durchschnitt* im Einsatz

Die Formel

```
=CUBEKPIELEMENT("offLine";"Durchschnitt";1)
```

zeigt nun das Wort *Durchschnitt* an. Mit

```
=CUBEWERT("offLine";CUBEKPIELEMENT("offLine";"Durchschnitt";1))
```

erhalten Sie die Zahl *1261* (das ist der gerundete Durchschnitt über alle Umsätze). In der zweiten Formel können Sie als zweites Argument auch den Bezug zu der Zelle geben, in der die erste der beiden Formeln steht.

Den Zielwert der Kenngröße *Durchschnitt* erhalten Sie aus

```
=CUBEWERT("offLine";CUBEKPIELEMENT("offLine";"Durchschnitt";2))
```

Entscheidend ist hier der Wert 2 im letzten Argument.

Wichtig ist es also auch in diesem Fall zu verstehen, dass die Anzeige in der Zelle mit der Formel genutzt werden kann, um Zellbeschriftungen zu erstellen. Der wirkliche Inhalt der Zelle ist viel informativer, wie der Einsatz der Funktion CUBEWERT() zeigt.

Weitere Komponenten des genannten KPI liegen im Beispiel-Cube nicht vor.

Siehe auch Alle Cubefunktionen, PIVOTDATENZUORDNEN()

CUBEMENGE() CUBESET()

Syntax CUBEMENGE(*Verbindung*;Menge_Ausdruck;Beschriftung;Sortier_reihenfolge;Sortieren_nach)

Definition Die Funktion gibt eine berechnete Menge (Set oder Satz) von Elementen zurück, indem eine Berechnungsvorschrift an den Cube (auf dem Server gesendet) wird, der daraufhin die Menge erstellt und anschließend an Microsoft Excel zurückgibt. Damit besteht auch bei dieser Funktion ein Unterschied zwischen dem, was in der Zelle angezeigt wird und dem, was die Zelle tatsächlich beinhaltet.

Argumente *Verbindung* (erforderlich) ist eine Zeichenkette mit dem Namen der Arbeitsmappenverbindung zum Cube. Nach dem Schreiben des ersten Hochkommata werden vorhandene Datenverbindungen kontextsensitiv eingeblendet (Abbildung 13.3 auf Seite 605).

Menge_Ausdruck (erforderlich) definiert die Elementmenge im Cube durch die Angabe eines multidimensionalen Ausdrucks (MDX). Dieser kann direkt eingegeben werden oder sich in einer Zelle befinden, auf die dann Bezug genommen wird. Die Verwendung von Tupeln im Ausdruck ist erlaubt. Auch beim Schreiben der Elemente reagiert die Eingabezeile kontextsensitiv.

Beschriftung (optional) ersetzt bei der Anzeige des Ergebnisses in der Excel-Zelle die eventuell vorhandene Beschriftung des Elements im Cube. Wurde ein Tupel verwendet, liefert die Funktion die (sofern vorhanden) Beschriftung des letzten Elements im Tupel.

Sortier_reihenfolge (optional) ist die Art einer eventuell auszuführenden Sortierung, die Werte sind ganze Zahlen nach der Tabelle 13.3. Während des Schreibens der Form werden Sie, wie in Abbildung 13.7 festgehalten, kontextsensitiv unterstützt.

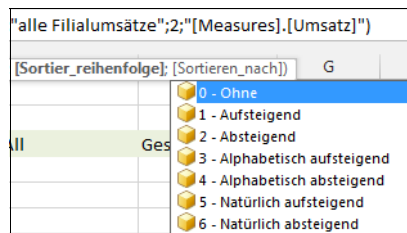


Abbildung 13.7: Die Tooltips zeigen an, was zu tun ist

Argument	Beschreibung	Auswirkung auf das fünfte Argument
0	Belässt die Menge in der Reihenfolge im Cube	Wird ignoriert
1	Sortiert aufsteigend nach dem fünften Argument	Ist erforderlich
2	Sortiert absteigend nach dem fünften Argument	Ist erforderlich
3	Sortiert alphabetisch aufsteigend	Wird ignoriert
4	Sortiert alphabetisch absteigend	Wird ignoriert
5	Sortiert natürlich aufsteigend	Wird ignoriert
6	Sortiert natürlich absteigend	Wird ignoriert

Tabelle 13.3
Bedeutung
der Ziffern für
das vierte
Argument der
Funktion

Der Standardwert des vierten Arguments ist 0. Mit einer alphabetischen Sortierung für eine Menge, die aus Tupeln definiert wird, wird anhand des letzten Elements im Tupel sortiert. Weitere Informationen zu diesen verschiedenen Sortierreihenfolgen finden Sie im Microsoft SQL Analysis Services-Hilfesystem.

Das in Abhängigkeit vom vierten Argument erforderliche Argument *Sortieren_nach* definiert die Werte in der Menge, nach denen sortiert werden soll. Wird es in den Fällen, in denen es erforderlich ist, weggelassen, reagiert die Funktion mit dem Fehlerwert *#WERT!*.

In der Zelle, die die Funktion enthält, wird während der Datenabfrage vorübergehend die Meldung *#DATEN_ABRUFEN...* angezeigt.

Hintergrund

Wird CUBEMENGE() als Argument für eine andere Cubefunktion verwendet, werden die Anweisungen zur Zurückgabe der Menge und nicht der angezeigte Wert verwendet.

Mögliche Fehlerwerte und Meldungen geben darüber Auskunft, was u.U. nicht beachtet wurde oder warum ein Element nicht gefunden werden kann¹³:

- ▶ Steht der OLAP-Server (oder der Offline-Cube) nicht zur Verfügung, erfolgt eine Fehlermeldung. Der Zellinhalt der betroffenen Zellen ändert sich nicht.
- ▶ Wenn *Element_Ausdruck* mehr als 255 Zeichen lang ist¹⁴, gibt CUBEMENGE() den Fehlerwert *#WERT!* zurück
- ▶ CUBEMENGE() gibt unter folgenden Bedingungen den Fehlerwert *#NV* zurück:
 - ▶ Der Verbindungsname entspricht keiner gültigen Arbeitsmappenverbindung
 - ▶ Die *Element_Ausdruck*-Syntax enthält Fehler
 - ▶ Die durch die MDX-Abfrage angegebene Menge ist nicht im Cube vorhanden
 - ▶ Der Satz enthält mindestens ein Element mit einer anderen Dimensionalität als die anderen Elemente
 - ▶ CUBEMENGE() kann den Fehlerwert *#NV* zurückgeben, wenn die Verbindung zur Datenquelle abgerissen ist und nicht neu aufgebaut werden kann

¹³ Wegen der Kontextsensitivität während der Eingabe der Formeln lassen sich die meisten Fehler vermeiden.

¹⁴ Diese Beschränkung für ein Argument oder eine Funktion kann dadurch umgangen werden, dass die benötigten Informationen als Werte in Zellen stehen, auf die in der Funktion selbst Bezug genommen wird.

Praxiseinsatz Im Beispiel zu diesem Kapitel entsteht durch

```
=CUBEMENGE("offLine";"[Filialen].[Filiale].Children";
"alle Filialumsätze";2;"[Measures].[Umsatz]")
```

die geordnete Menge aller Filialen, wobei zur Ordnung der Umsatz (über alle Produkte und alle Jahre) herangezogen wird. Die Filiale mit dem größten Umsatz steht an erster Stelle.

Wichtig ist es zu verstehen, dass die Anzeige in der Zelle mit der Formel¹⁵ genutzt werden kann, um Zellbeschriftungen zu erstellen. Der wirkliche Inhalt der Zelle ist viel informativer, wie der Einsatz der Funktionen CUBEMENGENANZAHL(), CUBERANGELEMENT() und CUBEWERT() beweist, wenn diese auf die Zellen mit den CUBEMENGE()-Einträgen Bezug nimmt.

Siehe auch Alle Cubefunktionen, PIVOTDATENZUORDNEN()

CUBEMENGENANZAHL()



CUBESETCOUNT()

Syntax CUBEMENGENANZAHL(*Menge*)

Definition Die Funktion gibt die Anzahl der Elemente aus einer Elementmenge (Satz von Elementen) zurück.

Argumente *Menge* (erforderlich) wird mithilfe der Funktion CUBEMENGE() definiert oder ist der Bezug auf eine Zelle, in der durch CUBEMENGE() Elemente des Cubes bereit gehalten werden.

Hintergrund In der Zelle, die die Funktion enthält, wird während der Datenabfrage vorübergehend die Meldung #DATEN_ABRUFEN... angezeigt.

Das Ergebnis ist eine nicht negative ganze Zahl. Wird durch das Argument ein Fehler erzeugt, wird dieser auch als Ergebnis angezeigt.

Praxiseinsatz Nehmen Sie im Beispiel dieses Kapitels mit der Funktion CUBEMENGENANZAHL() Bezug auf eine Zelle, in der durch

```
=CUBEMENGE("offLine";"[Filialen].[Filiale].Children";
"alle Filialumsätze";2;"[Measures].[Umsatz]")
```

die geordnete Menge der Filialen bereit gehalten wird, entsteht das Ergebnis 4. Das gleiche Ergebnis erhalten Sie, wenn Sie die genannte Formel als Argument übergeben. Beachten Sie, dass das Schlüsselwort *Children* bei der Formeleingabe nicht kontextsensitiv angezeigt wird.

Siehe auch Alle Cubefunktionen, PIVOTDATENZUORDNEN()

CUBERANGELEMENT()



CUBERANKEDMEMBER()

Syntax CUBERANGELEMENT(*Verbindung;Menge_Ausdruck;Rang;Beschriftung*)

Definition Die Funktion gibt das n-te Element einer geordneten Menge zurück.

Argumente *Verbindung* (erforderlich) ist eine Zeichenkette mit dem Namen der Arbeitsmappenverbindung zum Cube. Nach dem Schreiben des ersten Hochkommas werden vorhandene Datenverbindungen kontextsensitiv eingblendet (Abbildung 13.3 auf Seite 605).

¹⁵ Im vorliegenden Fall wird *alle Filialumsätze* angezeigt

Menge_Ausdruck (erforderlich) definiert die Elementmenge im Cube durch die Angabe eines multidimensionalen Ausdrucks (MDX). Dieser kann direkt eingegeben werden oder sich in einer Zelle befinden, auf die dann Bezug genommen wird. Die Verwendung von Tupeln im Ausdruck ist erlaubt. Auch beim Schreiben der Elemente reagiert die Eingabezeile kontextsensitiv.

Rang (erforderlich) ist eine nicht negative ganze Zahl, die die Position des gesuchten Elements in der Menge bestimmt.

Beschriftung (optional) ersetzt bei der Anzeige des Ergebnisses in der Excel-Zelle die eventuell vorhandene Beschriftung des Elements im Cube. Wurde ein Tupel verwendet, liefert die Funktion die (sofern vorhanden) Beschriftung des letzten Elements im Tupel.

In der Zelle, die die Funktion enthält, wird während der Datenabfrage vorübergehend die Meldung `#DATEN_ABRUFEN...` angezeigt.

Hintergrund

Mögliche Fehlerwerte und Meldungen geben darüber Auskunft, was u.U. nicht beachtet wurde oder warum ein Element nicht gefunden werden kann¹⁶:

- ▶ Steht der OLAP-Server (oder der Offline-Cube) nicht zur Verfügung, erfolgt eine Fehlermeldung. Der Zelleninhalt der betroffenen Zellen ändert sich nicht.
- ▶ Wenn *Menge_Ausdruck* mehr als 255 Zeichen lang ist¹⁷, gibt CUBERANGELEMENT() den Fehlerwert `#WERT!` zurück
- ▶ CUBERANGELEMENT() gibt unter folgenden Bedingungen den Fehlerwert `#NV` zurück:
 - ▶ Der Verbindungsname entspricht keiner gültigen Arbeitsmappenverbindung
 - ▶ Die *Menge_Ausdruck*-Syntax enthält Fehler
 - ▶ Die durch die MDX-Abfrage angegebene Menge ist nicht im Cube vorhanden
 - ▶ CUBERANGELEMENT() kann den Fehlerwert `#NV` zurückgeben, wenn die Verbindung zur Datenquelle abgerissen ist und nicht neu aufgebaut werden kann

Nehmen Sie im Beispiel dieses Kapitels mit der Funktion

Praxiseinsatz

```
=CUBERANGELEMENT("offLine";B9;1)
```

Bezug auf eine Zelle (hier B9), in der durch

```
=CUBEMENGE("offLine";"[Filialen].[Filiale].Children";  
"alle Filialumsätze";2;"[Measures].[Umsatz]")
```

die geordnete Menge der Filialen bereit gehalten wird, entsteht das Ergebnis *NordOst*. Diese Filiale ist über alle Produkte und Jahre die umsatzstärkste.

Mit der verschachtelten Formel

```
=CUBERANGELEMENT("offLine";CUBEMENGE("offLine";("[Filialen].[Filiale].[A11].[NordOst],  
[Jahre].Children");"alle Filialumsätze";2;"[Measures].[Umsatz]");3)
```

ermitteln Sie das umsatzschwächste Jahr (Position 3) dieser Filiale, nämlich 2010.

Alle Cubefunktionen, PIVOTDATENZUORDNEN()

Siehe auch

¹⁶ Wegen der Kontextsensitivität während der Eingabe der Formeln lassen sich die meisten Fehler vermeiden.

¹⁷ Diese Beschränkung für ein Argument oder eine Funktion kann dadurch umgangen werden, dass die benötigten Informationen als Werte in Zellen stehen, auf die in der Funktion selbst Bezug genommen wird.

CUBEWERT() CUBEVALUE()

Syntax CUBEWERT(*Verbindung*; *Element_Ausdruck1*; *Element_Ausdruck2*;...)

Definition Die Funktion gibt den Wert eines Elements (Zelle) aus einem Cube zurück.

Argumente *Verbindung* (erforderlich) ist eine Zeichenkette mit dem Namen der Arbeitsmappenverbindung zum Cube. Nach dem Schreiben des ersten Anführungszeichens werden vorhandene Datenverbindungen kontextsensitiv eingeblendet (Abbildung 13.3 auf Seite 605).

Element_Ausdruck (erforderlich) definiert die Lage des Elements im Cube durch die Angabe eines multidimensionalen Ausdrucks (MDX). Dieser kann direkt eingegeben werden oder sich in einer Zelle befinden, auf die dann Bezug genommen wird. Die Verwendung von Tupeln im Ausdruck ist erlaubt. *Element_Ausdruck* kann auch eine Menge darstellen, wie sie mit der CUBEMENGE()-Funktion definiert wird. Wenn keine Kennzahl (Measure) in *Element_Ausdruck* angegeben ist, wird die Standardkennzahl des Cubes verwendet.

Durch die mögliche Wiederholung des Arguments können Schnittmengen definiert werden. Die Verwendung von Tupeln ist erlaubt.

Auch beim Schreiben der Elemente reagiert die Eingabezeile kontextsensitiv.

Hintergrund In der Zelle, die die Funktion enthält, wird während der Datenabfrage vorübergehend die Meldung #DATEN_ABRUFEN... angezeigt.

Wird CUBEWERT() als Argument für eine andere Cubefunktion verwendet, wird der MDX-Ausdruck im Argument und nicht der angezeigte Wert verwendet.

Mögliche Fehlerwerte und Meldungen geben darüber Auskunft, was u.U. nicht beachtet wurde oder warum ein Element nicht gefunden werden kann¹⁸:

- ▶ Steht der OLAP-Server (oder der Offline-Cube) nicht zur Verfügung, erfolgt eine Fehlermeldung. Der Zellinhalt der betroffenen Zellen ändert sich nicht.
- ▶ Wenn der Verbindungsname keiner gültigen Arbeitsmappenverbindung entspricht, gibt CUBEWERT() den Fehlerwert #NV zurück
- ▶ Wenn *Element_Ausdruck* mehr als 255 Zeichen lang ist¹⁹, gibt CUBEWERT() den Fehlerwert #WERT! zurück.
- ▶ CUBEWERT() gibt unter folgenden Bedingungen den Fehlerwert #NV zurück:
 - ▶ Die *Element_Ausdruck*-Syntax enthält Fehler
 - ▶ Das durch die MDX-Abfrage angegebene Element ist nicht im Cube vorhanden
 - ▶ Für die angegebenen Werte im Tupel gibt es keine gemeinsame Schnittmenge²⁰
 - ▶ Der Menge der angeforderten Elemente enthält mindestens ein Element mit einer anderen Dimensionalität als die der anderen Elemente
 - ▶ CUBEWERT() kann den Fehlerwert #NV zurückgeben, wenn die Verbindung zur Datenquelle abgerissen ist und nicht neu aufgebaut werden kann

¹⁸ Wegen der Kontextsensitivität während der Eingabe der Formeln lassen sich die meisten Fehler vermeiden.

¹⁹ Diese Beschränkung für ein Argument oder eine Funktion kann dadurch umgangen werden, dass die benötigten Informationen als Werte in Zellen stehen, auf die in der Funktion selbst Bezug genommen wird.

²⁰ Stellt man sich unter einem Cube in der Tat einen mehrdimensionalen Würfel vor, geben die Ausdrücke im Tupel die »Marschrichtung« entlang jeder Achse vor. Ziel des Tupels muss es sein, mindestens einen gemeinsamen Schnittpunkt zu finden.

Im durchgängigen Beispiel dieses Kapitels können Sie mit

```
=CUBEWERT("offLine";"[Measures].[BruttoUmsatz]";"[Filialen].[Filiale].[A11].[NordOst]";
"[Jahre].[Jahr].[A11].[2010]";"[Produkte].[Produkt].[A11].[Kekse]")
```

den Kekse-Bruttoumsatz der NordOst-Filiale im Jahre 2010 ermitteln. Dieser lautet *1.785 €*²¹. Das gleiche Ergebnis erhalten Sie, wenn Sie ein Tupel verwenden (die Argumente der vorigen Formel werden in Klammern zusammengefasst, das Trennzeichen ist dann das Komma):

```
=CUBEWERT("offLine";([Measures].[BruttoUmsatz],[Filialen].[Filiale].[A11].[NordOst],
[Jahre].[Jahr].[A11].[2010],[Produkte].[Produkt].[A11].[Kekse]))
```

Haben Sie in der Zelle B3 die Formel

```
=CUBEELEMENT("offLine";"[Produkte].[Produkt].[A11].[Kekse]")
```

eingetragen, liefert

```
=CUBEWERT("offLine";B3)
```

den Gesamtumsatz an Keksen, nämlich *16.836 €*.

Sie können die Liste der Beispiele um jene ergänzen, die zur Funktion CUBEKPIELEMENT() angegeben wurden. Und auch die Funktionen CUBERANGELEMENT() und CUBEMENGE() sind mit

```
=CUBEWERT("offLine";CUBERANGELEMENT("offLine";CUBEMENGE("offLine";[Filialen]
.[Filiale].Children;"alle Filialumsätze";2;"[Measures].[Umsatz]");1))
```

vertreten. Hier lautet das Ergebnis *7.803 €*, das ist der Gesamtumsatz der besten Filiale hinsichtlich der Einzelumsätze, nämlich *NordOst*.

Alle Cubefunktionen, PIVOTDATENZUORDNEN()

Die Dateien für das durchgängige Beispiel dieses Kapitels finden Sie im Ordner `\Ms5-235\Kap13`.

Praxiseinsatz

Siehe auch



²¹ Hier wurde der Nettoumsatz im Cube mit 1,19 multipliziert.

Kapitel 14

Finanzmathematische Funktionen

AMORDEGRK()	623
AMORLINEARK()	625
AUFGELZINS()	627
AUFGELZINSF()	629
AUSZAHLUNG()	630
BW()	631
DIA()	634
DISAGIO()	635
DURATION()	636
EFFEKTIV()	638
GDA()	640
GDA2()	641
IKV()	642
ISPMT()	644
KAPZ()	645
KUMKAPITAL()	646
KUMZINSZ()	647
KURS()	649

KURSDISAGIO()	653	UNREGLE.KURS()	681
KURSFÄLLIG()	654	UNREGLE.REND()	683
LIA()	656	VDB()	684
MDURATION()	657	XINTZINSFUSS()	686
NBW()	658	XKAPITALWERT()	688
NOMINAL()	660	ZINS()	690
NOTIERUNGBRU()	660	ZINSSATZ()	692
NOTIERUNGDEZ()	661	ZINSTERMNZ()	694
PDURATION()	662	ZINSTERMTAGE()	695
QIKV()	663	ZINSTERMTAGNZ()	696
RENDITE()	665	ZINSTERMTAGVA()	697
RENDITEDIS()	669	ZINSTERMVZ()	698
RENDITEFÄLL()	671	ZINSTERMZAHL()	699
RMZ()	672	ZINSZ()	700
TBILLÄQUIV()	674	ZSATZINVEST()	701
TBILLKURS()	675	ZW()	702
TBILLRENDITE()	676	ZW2()	704
UNREGER.KURS()	677	ZZR()	705
UNREGER.REND()	680		

Grundlegende finanzmathematische Aufgabenstellungen lassen sich in folgende Gebiete aufteilen:

► **Probleme der einfachen Zinsrechnung**

Einfache Zinsrechnung zeichnet sich dadurch aus, dass Zinsen zum Zeitpunkt ihrer Fälligkeit nicht dem Kapital hinzugeschlagen werden. Diese Art der Berechnung findet vor allem bei Zeiträumen Anwendung, die unterhalb eines Jahrs, evtl. sogar im Tagesbereich, liegen.

Die Finanzmathematik kennt die verschiedensten Zinsbegriffe. In jedem Falle handelt es sich bei Zinsen um den Preis für geliehenes (Leihzinsen) oder verliehenes Kapital (Guthabenzinsen). Um die Höhe dieses Preises anzugeben, wird ein relativer Preis in Form eines Zinsfußes (Preis für 100 Geldeinheiten) bzw. Zinssatzes (Preis für eine Geldeinheit) formuliert. Dieser bezieht sich immer auf eine bestimmte Periodenlänge, im allgemeinen ein ganzes Jahr. In diesem Falle lautet der Zusatz *p.a.* (»per annum«).

Zwei prinzipiell verschiedene Zinsberechnungen begegnen Ihnen in diesem Kapitel immer wieder: nachschüssige und vorschüssige Verzinsung.

Bei der nachschüssigen Verzinsung werden die Zinsen am Ende der Zinsperiode vom Kapital, um welches es an deren Anfang ging, mithilfe des Zinssatzes ermittelt und bezahlt. Bei vorschüssiger Verzinsung werden die Zinsen vom Kapital, welches am Ende fällig ist, mithilfe des Zinssatzes berechnet und am Anfang der Periode bezahlt.

► **Probleme der Zinseszinsrechnung**

In diesem Fall werden Zinsen bei Fälligkeit dem Kapital zugeschlagen. Das führt zum sogenannten Zinseszinsseffekt. Da oft auf Formeln abgezielt wird, wird ebenso oft angenommen, dass der Zinssatz sich über den gesamten Betrachtungszeitraum nicht ändert.

► **Probleme der Rentenrechnung**

Unter einer Rente versteht man eine regelmäßig wiederkehrende Zahlung konstanter Höhe. Alle in Excel verfügbaren Funktionen gehen davon aus, dass die Rentenzahlungsstermine mit den Zinsterminen übereinstimmen. Unterschieden wird nur zwischen vorschüssigen Renten am Anfang der Zinsperioden und nachschüssigen, deren Zahlung am Ende der Perioden erfolgt.

► **Probleme der Tilgungsrechnung**

Die Finanzmathematik kennt wenigstens drei prinzipielle Formen der Tilgung von Krediten – Tilgung durch Einmalzahlung am Ende, Tilgung durch Ratenzahlung mit konstanten Tilgungsbeträgen und Tilgung durch Annuitätenzahlung, bei der die Summe aus Tilgungsbetrag plus Zinsen konstant bleibt.

Die letzte Form ist im Prinzip eine Form der Rentenrechnung.

Die Funktionen, die Excel zur Tilgungsrechnung bereitstellt, beziehen sich ausschließlich auf die erste und die dritte Form. Was nicht heißen soll, dass sich etwas anderes auf Tabellenblättern unter Einsatz verschiedenster Mittel auch ohne Funktionen nicht nachbilden lässt.

► **Probleme der Kursrechnung**

Kurs- und Renditerechnungen, vor allem für festverzinsliche Wertpapiere (Anleihen), sind ein besonders anspruchsvolles Thema der Finanzmathematik. Sehr viele der integrierten Funktionen beschäftigen sich mit diesem Schwerpunkt. Dabei definiert sich der Kurs immer als relativer Barwert zukünftiger Leistungen nach eventuellem Abzug sogenannter Stückzinsen. Die Rendite ist die Kennzahl (als Zinssatz), die einen am Markt vorhandenen Kurs realisiert.

► **Probleme der Investitionsrechnung**

Investitionsrechnung tritt oft in den Formen der statischen und der dynamischen Investitionsanalyse auf. Zur ersten Form gehören Methoden wie Kosten- und Gewinnvergleich sowie Amortisationsrechnung. Diese beruhen auf Daten der Kosten- und Leistungsrechnung. Dynamische Methoden berücksichtigen Zinseszinsseffekte und bewerten Geld, welches aus Ein- und Auszahlungen stammt. Excel verfügt über einige Funktionen zum Bearbeiten von Aufgaben der dynamischen Investitionsrechnung (Kapitalwertmethode und Methode des internen Zinssatzes).

► **Probleme der Abschreibungsrechnung**

Etwas ist für das Verständnis finanzmathematischer Vorgänge wichtig: Zahlungen werden nie allein durch ihre Höhe, sondern immer auch durch den Zeitpunkt, an dem sie erfolgen, bewertet. Es ist ein Unterschied, ob ein Schuldner seine Schuld heute oder erst in einem Jahr bezahlt. Je länger der Abstand, desto höher der Betrag, der fällig wird: zur Schuld kommen noch die Zinsen. Manchmal wird dieses Prinzip auch anders herum formuliert: Spätes Geld ist wenig wert. Das hat nichts mit dem Begriff der Inflation zu tun, sondern damit, dass es finanzmathematisch bei einem Jahreszinssatz von 10 % unerheblich ist, ob in einem Jahr 110 Geldeinheiten oder bereits heute 100 Geldeinheiten den Besitzer wechseln.

In diesem Sinne sind Probleme der Abschreibungsrechnung eigentlich keine Probleme finanzmathematischer Überlegungen. Traditionell werden Sie allerdings in Fachbüchern mit den anderen Problemen besprochen und auch Excel hält die entsprechenden Funktionen in einer gemeinsamen Gruppe parat. Die Bedeutung der Abschreibungsmethoden, die Excel zur Verfügung stellt, ist für die Praxis in Deutschland, die durch das Steuerrecht bestimmt wird, eher unerheblich.

Die Sprache der Excel-Hilfe zu Themen der Finanzmathematik ist nicht unbedingt an allen Stellen die Sprache der Finanzmathematik selbst. Gelegentlich wurde die Excel-Hilfe auch etwas unglücklich übersetzt bzw. die gegebenen Erläuterungen stimmen so nicht unbedingt. Vergleichen Sie sie deshalb mit den in diesem Kapitel aufgeschriebenen Hintergrundinformationen und ziehen Sie im Zweifelsfall die Beispiele und auch weitere Fachliteratur zur Klarstellung heran.



Im Ordner `\Ms5-235\Kap14` der Beispieldateien zum Buch finden Sie Arbeitsmappen, deren Namen genau den genannten Gebieten entsprechen und mit denen versucht wird, eine Gliederung in die Verwendung der Funktionen zu bringen.

Funktion	Beschreibung
AMORDEGRK()	Berechnet den Abschreibungsbetrag für Wirtschaftsgüter auf der Basis des französischen Buchführungssystems
AMORLINEARK()	Berechnet den Abschreibungsbetrag in der gewünschten Periode bei linearer Abschreibung
AUFGELZINS()	Berechnet die aufgelaufenen Zinsen einer gesamtfälligen Schuld im unterjährlichen Bereich mit periodischen Zinszahlungen
AUFGELZINSF()	Berechnet die aufgelaufenen Zinsen einer gesamtfälligen Schuld im unterjährlichen Bereich mit einer einmaligen Zinszahlung im Jahr
AUSZAHLUNG()	Ermittelt den Rückzahlungsbetrag einer durch einfache, vorschüssige Abzinsung entstandenen Einzahlung
BW()	Berechnet den Barwert eines regelmäßigen Zahlungsstroms unter Beachtung eventueller einmaliger Zahlungen am Ende
DIA()	Ermittelt die Abschreibungsbeträge eines Wirtschaftsgutes nach der arithmetisch-degressiven Methode
DISAGIO()	Berechnet den vorschüssigen Zinssatz (Disagio) bei vorgegebenem Bar- bzw. Endwert und vorgegebener Laufzeit
DURATION()	Berechnet die (durchschnittliche) Kapitalbindungsdauer eines festverzinslichen Wertpapiers
EFFEKTIV()	Berechnet aus einem (nachsüssigen) Nominalzinssatz den dazu äquivalenten effektiven (nachsüssigen) Jahreszinssatz
GDA()	Berechnet die Abschreibungsbeträge nach der Methode einer Mehrfachraten-Abschreibung
GDA2()	Berechnet Abschreibungsbeträge für ein Wirtschaftsgut nach der Methode der geometrisch-degressiven Abschreibung
IKV()	Berechnet eine Kenngröße der dynamischen Investitionsrechnung: den internen Zinssatz
ISPMT()	Berechnet bei gegebenem Jahreszinssatz und einfacher nachsüssiger Verzinsung die Summe der Zinsen
KAPZ()	Ermittelt den Anteil an einer Annuität, der zur Tilgung eines Darlehens benutzt wird
KUMKAPITAL()	Berechnet den Tilgungsanteil, der zwischen zwei Zeitpunkten gezahlt wird
KUMZINSZ()	Berechnet die kumulierten Zinsen, die zwischen zwei Zeitpunkten gezahlt werden
KURS()	Berechnet den Kurs eines festverzinslichen Wertpapiers (Anleihe)
KURSDISAGIO()	Berechnet den Auszahlungsbetrag eines mit vorschüssiger Verzinsung einfach abgezinsten Wertpapiers
KURSFÄLLIG()	Gibt den Kurs (als Prozentfuß, d.h. bezogen auf einen Nennwert) eines Wertpapiers zurück
LIA()	Berechnet die Abschreibungsbeträge von Wirtschaftsgütern nach der Methode der linearen Abschreibung
MDURATION()	Berechnet die als »modified duration« bekannte Kennzahl für festverzinsliche Wertpapiere
NBW()	Berechnet den Nettobarwert der mit einer Investition verbundenen zukünftigen Periodenüberschüsse (cash flow)

Tabelle 14.1
Die finanz-
mathematischen
Funktionen in
der Übersicht

Funktion	Beschreibung
NOMINAL()	Errechnet aus einem gegebenen Effektivzinssatz den Nominalzinssatz
NOTIERUNGBRU()	Wandelt die Dezimalstellen einer Dezimalzahl in den Zähler eines Bruchs mit vorgegebenem Nenner um und stellt diese in den Nachkommastellen dar
NOTIERUNGDEZ()	Wandelt einen als Zähler eines Bruchs zu interpretierenden Nachkommaanteil einer Zahl bei gegebenem Nenner in eine Dezimalzahl um
PDURATION()	Berechnet die Laufzeit einer Investition bei gegebenem Zinssatz sowie eingesetztem und erwartetem Kapital
QIKV()	Berechnet die interne Verzinsung einer Investition, wobei negative Periodenüberschüsse mit einem anderen Zinssatz als positive bewertet werden
RENDITE()	Ermittelt die Rendite eines festverzinslichen Wertpapiers (Anleihe)
RENDITEDIS()	Berechnet die jährliche Rendite (als nachschüssigen Zinssatz) für ein Wertpapier
RENDITEFÄLL()	Berechnet die jährl. Rendite eines festverzinsl. Wertpapiers im unterjährlichen Bereich (ohne Zinseszinsseffekte)
RMZ()	Liefert die Höhe der Rente (regelmäßige Zahlung) eines der Rentenrechnung zuzuordnenden Vorgangs
TBILLÄQUIV()	Errechnet den äquivalenten nachschüssigen Jahreszinssatz auf der Basis von 365 Tagen für einen geg. vorschüssigen Jahreszinssatz auf Basis von 360 Tagen
TBILLKURS()	Liefert den Kurs eines abgezinsten Wertpapiers als Prozentfuß
TBILLRENDITE()	Liefert die Rendite eines abgezinsten Wertpapiers als nachschüssigen Jahreszinssatz
UNREGER.KURS()	Berechnet den Kurs eines festverzinsl. Wertpapiers unter Berücksichtigung einer ersten Zinsperiode
UNREGER.REND()	Berechnet die Rendite eines festverzinsl. Wertpapiers für den Zeitraum vom Abrechnungstag bis zum Fälligkeitstermin
UNREGLE.KURS()	Berechnet den Kurs eines festverzinsl. Wertpapiers in einer letzten Zinsperiode
UNREGLE.REND()	Berechnet die Rendite eines festverzinsl. Wertpapiers in einer letzten Zinsperiode
VDB()	Berechnet Abschreibungsbeträge nach der geometrisch-degressiven Abschreibung
XINTZINSFUSS()	Liefert den internen Zinssatz einer Reihe nicht notwendig periodisch anfallender Zahlungen im unterjährlichen Bereich
XKAPITALWERT()	Gibt den Kapitalwert einer Reihe nicht notwendig periodisch anfallender Überschüsse im unterjährlichen Bereich zurück
ZINS()	Ermittelt den zutreffenden Zinssatz bei Aufgaben der Zinseszinsrechnung sowie der Rentenrechnung
ZINSSATZ()	Berechnet den äquivalenten nachschüssigen Zinssatz für ein Wertpapier, welches mit einem Abschlag für eine in der Regel unterjährliche Laufzeit ausgestattet wurde
ZINSTERMNZ()	Ermittelt das Datum der ersten Kuponzinsszahlung, nachdem ein festverzinsl. Wertpapier den Besitzer gewechselt hat
ZINSTERMTAGE()	Ermittelt die Anzahl der Tage der Zinsperiode, in die der Besitzwechsel eines festverzinsl. Wertpapiers fällt
ZINSTERMTAGNZ()	Ermittelt die Anzahl der Tage zw. dem Tag des Besitzerwechsels eines festverzinsl. Wertpapiers und dem nächsten Zinstermin



Funktion	Beschreibung
ZINSTERMTAGVA()	Ermittelt die Anzahl der Tage, die seit der letzten Zinszahlung bis zum Besitzwechsel eines festverzinsl. Wertpapiers vergangen sind
ZINSTERMVZ()	Ermittelt das Datum der bislang letzten Zinszahlung vor Besitzwechsel eines festverzinslichen Wertpapiers
ZINSTERMZAHL()	Ermittelt die Anzahl der Zinszahlungen, die der neue Besitzer nach Kauf eines festverzinsl. Wertpapiers vor sich hat
ZINSZ()	Ermittelt den Anteil an einer Annuität, der zur Zinszahlung für Darlehen benutzt wird
ZSATZINVEST()	Berechnet den Zinssatz, zu welchem ein Kapital investiert werden muss, damit es einen bestimmten Zukunftswert besitzt
ZW()	Berechnet den Zukunftswert eines regelmäßigen Zahlungsstroms unter Beachtung einmaliger Zahlungen am Anfang des betrachteten Zeitraums
ZW2()	Errechnet den Endwert für ein Kapital mit variablen Periodenzinssätzen
ZZR()	Berechnet die Laufzeit eines Vorgangs der Zinseszinsrechnung, Rentenrechnung oder Tilgungsrechnung

AMORDEGRK() AMORDEGRC()

AMORDEGRK(*Ansch_Wert;Kaufdatum;Erster_Zinstermin;Restwert;Termin;Satz;Basis*)

Syntax

Diese Funktion berechnet in Abhängigkeit von der gewünschten Periode den Abschreibungsbetrag für Wirtschaftsgüter auf der Basis des französischen Buchführungssystems.

Definition

Der berechnete Betrag wird auf eine ganze Zahl gerundet.

Ansch_Wert (erforderlich) verlangt die Anschaffungskosten (Nettokaufpreis plus Anschaffungsnebenkosten minus Anschaffungskostenminderungen) eines Wirtschaftsguts. Verwenden Sie keine Zahl, erscheint der Fehlerwert #WERT!. Ist die eingegebene Zahl negativ, erscheint die Fehlermeldung #ZAHL!.

Argumente

Kaufdatum (erforderlich) ist das Anschaffungsdatum des Wirtschaftsguts, also der Beginn der Abschreibung.

Erster_Zinstermin (erforderlich) meint das Datum am Ende der ersten Periode im Abschreibungszeitraum. Diese Periode bekommt die Nummer 0 (Null).

Restwert (erforderlich) ist der vorgesehene Restwert des Wirtschaftsguts nach den Jahren der Abschreibung. Verwenden Sie hier eine Zahl größer als *Kosten*, wird der Fehlerwert #ZAHL! ausgegeben. Der gleiche Fehlerwert erscheint bei Verwendung einer negativen Zahl.

Termin (erforderlich) gibt den Zeitraum an, für welchen der Abschreibungsbetrag ermittelt werden soll. Verwenden Sie nur ganze Zahlen größer oder gleich Null.

Satz (erforderlich) ist der (zunächst lineare) Abschreibungsprozentsatz.

Basis (optional) definiert die zu verwendende Methode bei der Zählung von Tagen gemäß der Tabelle 14.2.¹

¹ Die Namen der Argumente folgen der Version Excel 2013. Diese sind besser gewählt als in den Vorversionen, nur dass der erste Termin nichts mit Zinsen zu tun hat.

Tabelle 14.2
Methoden des
Zählens von
Tagen im Jahr

Basis	Methode	Bedeutung
0	30/360 (NASD-Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Ist das Ausgangsdatum der 31. eines Monats, wird dieses Datum zum 30. desselben Monats. Ist das Enddatum der 31. eines Monats und das Ausgangsdatum ein Datum vor dem 30. eines Monats, wird das Enddatum zum 1. des darauf folgenden Monats. In allen anderen Fällen wird das Enddatum zum 30. desselben Monats.
1	Genau/Genau	Jeder Monat hat so viele Tage wie im Kalender stehen, die Summe über alle Monate ergibt die Anzahl der Tage im Jahr
3	Genau/365 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 365 Tage.
4	30/360 (europäische Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Jedes Ausgangs- und Enddatum, das auf den 31. eines Monats fällt, wird zum 30. desselben Monats.

Hintergrund Abschreibung hat den Sinn, den Werteverlust eines Wirtschaftsguts zu ermitteln und sichtbar zu machen. Davon zu unterscheiden ist der Begriff der Absetzung für Abnutzung, der die Kosten für den Erwerb eines Wirtschaftsguts als Betriebsausgaben steuerlich erfasst.

Der durch *Satz* gegebene Abschreibungsprozentsatz wird zunächst als linearer Abschreibungsatz interpretiert und bestimmt so die Dauer der Abschreibung (5 % gleich 20 Jahre, 10 % gleich 10 Jahre, 20 % gleich 5 Jahre usw.). Degressive Abschreibung entsteht durch eine zusätzliche Gewichtung (Multiplikation) des Abschreibungssatzes mit einem Faktor nach folgender Regel:

- ▶ Faktor 1,5 bei einem *Satz* größer 25 % (das entspricht einer Abschreibungsdauer von drei bis vier Jahren)
- ▶ Faktor 2 bei einem *Satz* zwischen 16,66 % und 25 % (das entspricht einer Abschreibungsdauer von fünf bis sechs Jahren)
- ▶ Faktor 2,5 bei einem *Satz* kleiner 16,67 % (das entspricht einer Abschreibungsdauer von mehr als sechs Jahren)

Wird auf einen Restwert von 0 (Null) abgeschrieben, kann dieser nicht in der Anzahl der geforderten Perioden erreicht werden (er wird zwar immer kleiner, verschwindet aber erst dann, wenn durch Runden auch die Nachkommastellen verschwinden). Deshalb wird in diesem Fall der in der dritten Periode vor Schluss verbleibende Restwert auf die beiden letzten Perioden aufgeteilt.

Wird auf einen Restwert größer als 0 (Null) abgeschrieben, endet die Abschreibung nach obigem Vorgehen in der Periode, in welcher ein Buchwert kleiner oder gleich dem vorgegebenen Restwert erreicht wird. Die Funktion liefert in den Folgeperioden den Abschreibungswert 0 (Null). Sie müssen allerdings den letzten Buchwert als letzten Abschreibungswert verwenden.

Eine Besonderheit liegt im Abschreibungsbetrag der ersten Periode (diese bekommt die Nummer 0). Diese wird nur anteilig am Jahr berücksichtigt, wobei die Zählung der Tage wie in der obigen Tabelle erfolgt.

Wegen des Gewichtungsfaktors kann die Funktion nicht nach dem in Deutschland geltenden Steuerrecht eingesetzt werden.

Ein am 6.6.2012 erworbenes Wirtschaftsgut mit Anschaffungskosten von 1.000,00 € wird im ersten Jahr bei einem Abschreibungsprozentsatz von 10 % unabhängig vom gedachten Restwert um 142,00 € abgeschrieben. Das ergibt sich aus der Formel

```
=AMORDEGRK(1000;"6.6.2012";"31.12.2012";0;0;10%;4)
```

wobei sich hinter den Zahlen in der Regel die Bezüge auf die Zellen befinden, in denen diese Zahlen stehen.

Sie können versuchen, diesen Betrag auszurechnen, indem Sie die Formel =TAGE360("6.6.2012";"31.12.2012";WAHR) zur Ermittlung der Tage nutzen. Die Formel sollte aussehen wie

```
=RUNDEN(1000*10%*2,5*TAGE360("6.6.2012";"31.12.2012";WAHR)/360;0)
```

Die späteren Abschreibungsbeträge ermitteln sich einfacher aus dem jeweiligen Buchwert am Ende der vorhergehenden Periode multipliziert mit 10 % mal 2,5.

AMORLINEARK(), TAGE360(), ZINSTERMTAGVA()

Einzelheiten zur Umsetzung des Beispiels finden Sie auf dem Arbeitsblatt *AMORDEGRK* der Arbeitsmappe *Abschreibungsrechnung.xlsx* im Ordner *\Ms5-235\Kap14*.

Praxiseinsatz

Siehe auch



AMORLINEARK()



AMORLINC()

AMORLINEARK(*Ansch_Wert*; *Kaufdatum*; *Erster_Zinstermin*; *Restwert*; *Termin*; *Satz*; *Basis*)

Diese Funktion berechnet den Abschreibungsbetrag in der gewünschten Periode bei linearer Abschreibung von Wirtschaftsgütern. Die Berechnungen beruhen zwar auf dem französischen Buchführungssystem, lassen sich aber mit ein paar Zusatzüberlegungen für die Abschreibungsrechnung nach derzeit geltendem Steuerrecht in Deutschland anpassen.

Ansch_Wert (erforderlich) verlangt die Anschaffungskosten (Nettokaufpreis plus Anschaffungsnebenkosten minus Anschaffungskostenminderungen) eines Wirtschaftsguts. Verwenden Sie keine Zahl, erscheint der Fehlerwert #WERT!. Ist die eingegebene Zahl negativ, erscheint die Fehlermeldung #ZAHL!.

Kaufdatum (erforderlich) ist das Anschaffungsdatum des Wirtschaftsguts, also der Beginn der Abschreibung.

Erster_Zinstermin (erforderlich) meint das Datum am Ende der ersten Periode im Abschreibungszeitraum. Diese Periode bekommt die Nummer 0 (Null).

Restwert (erforderlich) ist der vorgesehene Restwert des Wirtschaftsguts nach den Jahren der Abschreibung. Verwenden Sie hier eine Zahl größer als *Kosten*, wird der Fehlerwert #ZAHL! ausgegeben. Der gleiche Fehlerwert erscheint bei Verwendung einer negativen Zahl.

Termin (erforderlich) gibt den Zeitraum an, für welchen der Abschreibungsbetrag ermittelt werden soll. Verwenden Sie nur ganze Zahlen größer oder gleich Null.

Satz (erforderlich) ist der (zunächst lineare) Abschreibungsprozentsatz. Dieser wird in der Regel als Reziprokes der geplanten Abschreibungsdauer ermittelt (10 Jahre ergeben 10 %, 5 Jahre 20 % usw.)

Syntax

Definition

Argumente

Basis (optional) definiert die zu verwendende Methode bei der Zählung von Tagen gemäß der Tabelle 14.2.²

Hintergrund Eine Abschreibung hat den Sinn, den Werteverlust eines Wirtschaftsguts zu ermitteln und sichtbar zu machen. Davon zu unterscheiden ist der Begriff der Absetzung für Abnutzung, der die Kosten für den Erwerb eines Wirtschaftsguts als Betriebsausgaben steuerlich erfasst.

Der durch *Satz* gegebene Abschreibungsprozentsatz ist ein linearer Abschreibungssatz und bestimmt so die Dauer der Abschreibung.

Eine Besonderheit liegt im Abschreibungsbetrag der ersten Periode (diese bekommt die Nummer 0). Diese wird nur anteilig am Jahr berücksichtigt, wobei die Zählung der Tage nach verschiedenen Methoden erfolgen kann.

Wird auf einen Restwert von 0 (Null) abgeschrieben und ist die erste Periode kein ganzes Jahr, entsteht im Endergebnis eine Periode mehr als ursprünglich gedacht. Der aus der verkürzten ersten Periode verbleibende »Überhang« wird in der letzten Periode abgeschrieben.

Wird auf einen Restwert größer als 0 (Null) abgeschrieben, endet die Abschreibung nach obigem Vorgehen in der Regel vor Erreichen der ursprünglich gedachten letzten Periode. Das liegt daran, dass der Abschreibungsprozentsatz immer auf die Kosten und nicht – wie in Deutschland üblich – auf die Differenz von Kosten und Restwert angewendet wird.

Wollen Sie die Funktion nach dem zum Zeitpunkt des Erscheinens dieses Buchs in Deutschland geltenden Steuerrecht verwenden, können Sie im Falle der vollständigen Abschreibung (*Restwert* = 0) mit *Basis* = 4 und einem Anschaffungsdatum (Argument *Kaufdatum*) arbeiten, welches auf den ersten Tag des Monats der tatsächlichen Anschaffung fällt. Wählen Sie dann als Datum für *Erster_Zinstermin* den 1. Januar des Folgejahrs. Dadurch ist sichergestellt, dass der angefangene Monat der Anschaffung mit $\frac{1}{12}$ von *Ansch_Wert* berücksichtigt wird. Durch entsprechende Beispiele überzeugen Sie sich schnell, dass es nicht zutreffend ist, als Anschaffungsdatum den letzten Tag des Vormonats bei Ende der Periode zum 31.12. zu wählen (so liegen etwa vom 28.2. bis Jahresende 302 und nicht 300 Tage).

Praxiseinsatz Sie kaufen am 5.10.2013 einen PC für 3.000 €. Der Abschreibungsprozentsatz beträgt wegen der dreijährigen Abschreibungsdauer 33,333 %. Die folgende Formel berechnet die Abschreibung des ersten Jahrs (Periode 0):

```
=AMORLINEARK(3000;DATUM(JAHR("5.10.2013");MONAT("5.10.2013");1);"1.1.2014";0;0;33,333%;4)
```

wobei statt der konkreten Werte die Zellbezüge auf Zellen mit diesen Werten stehen sollten.

Sie können das Ergebnis von 250,00 € auch durch

```
=3000*33,333%*TAGE360(DATUM(JAHR("5.10.2013");MONAT("5.10.2013");1);"1.1.2014";WAHR)/360
```

nachbilden. Statt des Einsatzes von TAGE360() steht Ihnen die einfache Verwendung von

```
=3000*33,333%*(13-MONAT("5.10.2013"))/12
```

als Alternative zur Verfügung.

Siehe auch AMORDEGRK(), TAGE360(), ZINSTERMTAGVA()



Das Beispiel sowie Formeln zu den Hintergrundinformationen finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Abschreibungsrechnung.xlsx* auf dem Arbeitsblatt *AMORLINEARK*.

² Die Namen der Argumente folgen der Version Excel 2013. Diese sind besser gewählt als in den Vorversionen, nur dass der erste Termin nichts mit Zinsen zu tun hat.

AUFGELZINS()



AUFGELZINS(*Emission*; *Erster_Zinstermin*; *Abrechnung*; *Satz*; *Nennwert*; *Häufigkeit*; *Basis*; *Berechnungsmethode*)

Syntax

Die Funktion AUFGELZINS() berechnet die aufgelaufenen Zinsen einer gesamt-fälligen Schuld im unterjährlichen Bereich bzw. die Stückzinsen eines festverzinslichen Wertpapiers mit periodischen Zinszahlungen.

Definition

Emission (erforderlich) gibt das Datum der Schuldvereinbarung bzw. der Ausgabe des Wertpapiers an.

Argumente

Erster_Zinstermin (erforderlich) ist im unterjährlichen Fall mit dem Datum der Fälligkeit identisch und ergibt sich sonst aus der durch Häufigkeit definierten Periodenzahl.

Abrechnung (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Satz (erforderlich) ist der vereinbarte jährliche Zinssatz (Kuponzinssatz) des Schuld-papiers/der Anleihe.³

Nennwert (optional) ist der Nominalwert des Wertpapiers. Wird er nicht angegeben, rechnet Excel mit 1000 (Geldeinheiten).

Häufigkeit (erforderlich) bestimmt die Anzahl der Zinszahlungen innerhalb eines Jahresabstands. Zulässig sind die ganzen Zahlen 1 = jährlich; 2 = halbjährlich; 4 = vierteljährlich.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.3. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

Basis	Methode	Bedeutung
0	30/360 (NASD-Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Ist das Ausgangsdatum der 31. eines Monats, wird dieses Datum zum 30. desselben Monats. Ist das Enddatum der 31. eines Monats und das Ausgangsdatum ein Datum vor dem 30. eines Monats, wird das Enddatum zum 1. des darauf folgenden Monats. In allen anderen Fällen wird das Enddatum zum 30. desselben Monats.
1	Genau/Genau	Jeder Monat hat so viele Tage wie im Kalender stehen, die Summe über alle Monate ergibt die Anzahl der Tage im Jahr
2	Genau/360 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 360 Tage.
3	Genau/365 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 365 Tage.
4	30/360 (europäische Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Jedes Ausgangs- und Enddatum, das auf den 31. eines Monats fällt, wird zum 30. desselben Monats.

Tabelle 14.3

Verschiedene Methoden zur Ermittlung der Anzahl der Tage eines Jahrs

Berechnungsmethode ist ein optionaler Wahrheitswert, der darüber entscheidet, ob die Zinsen für die gesamte bisherige Laufzeit berechnet werden sollen (WAHR oder weggelassen) oder ob nur die Zinsen seit der letzten Zinszahlung angegeben werden sollen (FALSCH).

³ In den Versionen vor Excel 2013 heißt dieses Argument *Nominalzinssatz*.

Alle Argumente der Funktion, die ein Datum betreffen, verwenden eine Datumsangabe ohne Uhrzeit. Mit anderen Worten, gebrochene Zahlen werden abgerundet. Die beiden Argumente *Häufigkeit* und *Basis* verlangen ebenfalls ganze Zahlen und schneiden Nachkommastellen ab.

Werden ungültige Datumseinträge verwendet bzw. keine Zahlen dort angegeben, wo sie erforderlich sind, liefert die Funktion den Fehlerwert *#WERT!*. Im Falle ungültiger Zahlen für die Nicht-Datums-Argumente lautet die Fehlermeldung *#ZAHL!*.

Hintergrund Zinsen für festverzinsliche Anleihen werden entsprechend dem Argument *Häufigkeit* einmal oder mehrmals im Jahr ausgezahlt. Wechselt das Wertpapier zwischen den Zinsstichtagen den Besitzer, muss der Käufer dem Verkäufer neben dem Kurs die bis dato angefallenen Stückzinsen bezahlen (das ergibt den Kaufpreis). Gibt es mehr als einen Zinstermin im Jahresabstand, wird zur Berechnung der Stückzinsen der Nominalzinssatz durch die Anzahl der Zinstermine geteilt. Mit diesem (sogenannten relativen) Zinssatz werden dann die Tage der angebrochenen Zinsperiode bewertet, wobei sich die Zählweise der Tage nach dem Argument *Basis* richtet.

Praxiseinsatz Eine gesamtällige Schuld über 1.000,00 € wird am 1.6.2013 mit einem Nominalzinssatz von 4 % p. a. und die Rückzahlung zum 1.12.2013 vereinbart. Am 9.8.2013 erfolgt ein Gläubigerwechsel. Zu diesem Termin sind Realzinsen von 4,5 % p. a. für die ausstehende Zeit bis zur Rückzahlung üblich. Welchen Preis zahlt der Übernehmende?

Um es vorwegzunehmen: Sie können das Ergebnis einfach mithilfe der Funktionen *KURSFÄLLIG()* und *AUFGELZINS()* bestimmen. Wollen Sie aber nachbilden, was diese Funktionen leisten, sind folgende Überlegungen sinnvoll:

1. Welche Zinsen sind bis heute aufgelaufen und dem bisherigen Gläubiger zu zahlen? Die Antwort von 7,56 € (nach Rundung) bringt

=AUFGELZINS(C2;C4;C3;4%;1000;1;4)

wenn in Zelle C2 das Datum vom 1.6.2013, in Zelle C3 das vom 9.8.2013 und in Zelle C4 das vom 1.12.2013 steht. Sie prüfen das, indem Sie die seit der Schuldvereinbarung vergangenen Tage zählen (es sind 68, wie auch *TAGE360()* und *ZINSTERMTAGVA()* beweisen) und den Zinsanteil durch $1000 \cdot 68 / 360 \cdot 4\%$ bestimmen.

2. Welchen Wert hat die Auszahlung von 1.000,00 € plus Zinsen (in Höhe von 4%/2 auf 1000 wegen des Abstandes von nur einem halben Jahr) zum 1.12.2013 bei der gegebenen Realverzinsung am 9.8.2013? Das berechnen Sie durch Abzinsen des Zukunftswerts von 1.020,00 € auf den Tag vom 9.8.2013. Hierzu ermitteln Sie, dass zwischen beiden Terminen ein Abstand von 112 Tagen (bei *Basis* 4) ist. Bei dieser Ermittlung hilft Ihnen *TAGE360()* bzw. die Funktion *ZINSTERMTAGNZ()*, die auch andere als die 30/360-Tage-Zählung kennt. Der Barwert beträgt 1.005,92 €.
3. Dieser Barwert (Preis) ergibt sich auch, wenn Sie das 10-fache des Ergebnisses von *KURSFÄLLIG()* nehmen und die Stückzinsen addieren. Das 10-fache ist deshalb zu nehmen, weil *KURSFÄLLIG()* mit einem Nennwert von 100 arbeitet.

Siehe auch *AUFGELZINSF()*, *KURSFÄLLIG()*, *ZINSTERMTAGVA()*, *ZINSTERMTAGNZ()*, *TAGE360()*



Dieses Beispiel finden Sie im Ordner *\Ms5-235\Kap14* in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* auf dem Arbeitsblatt *AUFGELZINS*.

AUFGELZINSF()



ACCRINTM()

AUFGELZINSF(*Emission*; *Abrechnung*; *Nominalzins*; *Nennwert*; *Basis*)

Die Funktion AUFGELZINSF() berechnet die aufgelaufenen Zinsen einer gesamt-fälligen Schuld im unterjährlichen Bereich bzw. die Stückzinsen eines festverzinslichen Wertpapiers mit einer einmaligen Zinszahlung im Jahr.

Emission (erforderlich) ist das Datum der Ausgabe des Wertpapiers.

Abrechnung (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Nominalzins (erforderlich) ist der vereinbarte Zinssatz des Wertpapiers, bezogen auf ein ganzes Jahr.

Nennwert (optional) ist der Nominalwert des Wertpapiers. Wird er nicht angegeben, rechnet Excel (entgegen der Information der Excel-Hilfe bis Excel 2003⁴) mit 1000 (Geldeinheiten).

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.3.

Alle Argumente der Funktion, die ein Datum betreffen, verwenden eine Datumsangabe ohne Uhrzeit. Mit anderen Worten, gebrochene Zahlen werden abgerundet. Die beiden Argumente *Häufigkeit* und *Basis* verlangen ebenfalls ganze Zahlen und schneiden Nachkommastellen ab.

Werden ungültige Datumseinträge verwendet bzw. keine Zahlen dort angegeben, wo sie erforderlich sind, liefert die Funktion den Fehlerwert #WERT!. Im Falle ungültiger Zahlen für die Nicht-Datums-Argumente lautet die Fehlermeldung #ZAHL!.

Diese Funktion ist eine vereinfachte Form der Funktion AUFGELZINS(). Der Unterschied besteht in der Vorgabe nur einer Zinsperiode, die mit einem ganzen Jahr (nicht notwendigerweise das Kalenderjahr) identisch ist. Aufgrund der genannten Gemeinsamkeit gelten alle zutreffenden Hintergrundinformationen zu AUFGELZINS().

Die Stückzinsen aus dem Beispiel zur Funktion AUFGELZINS() im vorigen Abschnitt berechnen sich nun nach der Formel

```
=AUFGELZINSF(C2;C3;4%;1000;4)
```

wobei in C2 das Ausgabedatum und in C3 das Datum des Besitzwechsels steht.

Die Bundesanleihe WKN 113517 wurde am 25.10.2000 mit einem Nominalzinssatz von 5,5 % bei einer Zinszahlung im Jahr begeben und ist am 4.1.2031 fällig (Quelle: http://www.bundesbank.de/download/kredit/kredit_emission_chronologie.pdf).

Am 30.8.2010 lag der Kurs bei 143,27. Wie viel musste ein Käufer (ohne Gebühren) zum Erwerb zahlen?

Zur Lösung beachten Sie, dass Sie als *Emission* nicht den 25.10.2000 nehmen können, sondern den Tag der letzten Zinszahlung – das ist der 4.1.2010 – einsetzen müssen:

```
=AUFGELZINSF(C24;C25;5,5%;100;4)
```

(in C24 steht der 4.1.2010 und in C25 der 30.8.2010). Zum Kaufkurs von 143,27 sind also noch 3,61 € Stückzinsen pro Anleiheanteil hinzuzufügen.

⁴ Mit Excel 2007 wurden etliche Fehler der Beschreibung von Funktionen, die bis Version 2003 im Add-In *Analyse-Funktionen* umgesetzt wurden, bereinigt.

Syntax

Definition

Argumente

Hintergrund

**Praxiseinsatz
Gesamt-fällige
Schuld**

**Bundes-
anleihen**

Tipp Um die Rendite einer Anlage in einem solchen Wertpapier zu bestimmen, ist die Funktion `RENDITE()` geeignet. Diese arbeitet im Gegensatz zu `RENDITEFÄLL()`, die im unterjährlichen Bereich Anwendung findet, mit Zinseszinsseffekten.

Siehe auch `AUFGELZINS()`



Die Beispiele finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* auf dem Arbeitsblatt `AUFGELZINSF`.

AUSZAHLUNG() RECEIVED()

Syntax `AUSZAHLUNG(Abrechnung;Fälligkeit;Anlage;Disagio;Basis)`

Definition Diese Funktion ermittelt den Rückzahlungsbetrag einer durch einfache, vorschüssige Abzinsung entstandenen Einzahlung.

Argumente *Abrechnung* (erforderlich) ist der Tag, an dem die Einzahlung (Investition seitens des Käufers, Kreditaufnahme seitens des Verkäufers) vorgenommen wird.

Fälligkeit (erforderlich) markiert das Datum, an welchem die Rückzahlung der Einzahlung erfolgt.

Anlage (erforderlich) informiert über die Höhe der Einzahlung (des Darlehens).

Disagio (erforderlich) gibt den Prozentsatz der vorschüssigen Verzinsung an.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.3.

Alle Argumente der Funktion, die ein Datum betreffen, verwenden eine Datumsangabe ohne Uhrzeit, gebrochene Zahlen werden also abgerundet. Das Argument *Basis* verlangt ebenfalls eine ganze Zahl und schneidet Nachkommastellen ab.

Werden ungültige Datumseinträge verwendet bzw. keine Zahlen dort angegeben, wo sie erforderlich sind, liefert die Funktion den Fehlerwert `#WERT!`. Im Falle ungültiger Zahlen für die Nicht-Datums-Argumente lautet die Fehlermeldung `#ZAHL!`.

Wichtig Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der `DATUM()`-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Hintergrund Das Prinzip der vorschüssigen Verzinsung besteht in der Annahme, dass sich ein Anfangskapital (einzuzahlender Betrag, Darlehen) aus dem Rückzahlungsbetrag minus die Zinsen auf diesen Betrag ergeben. Dieses Prinzip wird vor allem im unterjährlichen Bereich verwendet und unterscheidet sich von dem Prinzip, welches einem Sparbuch oder einem Hypothekendarlehen zugrunde liegt. Dort werden nämlich die Zinsen am Ende einer Periode auf Basis des Anfangskapitals berechnet (nachsüssige Verzinsung).

Zwischen den Funktionen zu diesem Themenkreis besteht die Verbindung

$$AUSZAHLUNG() - AUSZAHLUNG() \cdot DISAGIO() \cdot \frac{TAGE360()}{360} = KURSDISAGIO()$$

(hier notiert für den Fall von *Basis* = 4). Das ist aber gerade das Prinzip vorschüssiger Verzinsung.

Die Funktion AUSZAHLUNG() löst die obige Gleichung für den Fall auf, dass die anderen Angaben vorliegen.

Am 10.5.2013 legt ein Geschäftsmann seiner Bank einen Wechsel mit einer (Rest-) Laufzeit von zwei Monaten vor. Diese hat einen Diskontsatz von 5 % p. a. angesetzt und (ohne Gebühren zu fordern) dem Konto 4.958,33 € gutgeschrieben. Wie hoch war die Wechselsumme?

Diese Aufgabenstellung ist natürlich nicht ganz typisch. Gewöhnlich wird man die Wechselsumme (Rückzahlungsbetrag, die »Erfinder« von Excel nennen dies AUSZAHLUNG()) kennen und nach dem Auszahlungsbetrag (dieser heißt KURSDISAGIO()) fragen.

Sie lesen die Gleichung aus den Hintergrundinformationen als

$$Wechselsumme - Wechselsumme \cdot Diskontsatz \cdot \frac{TAGE360()}{360} = Gutschrift$$

und verwenden

=AUSZAHLUNG("10.5.2013";"10.7.2013";4958,33;5%;4)

um zum Ergebnis 5.000 € zu gelangen.

Natürlich stehen die konkreten Zahlen in Zellen und die Formel verwendet Zellbezüge.

KURSDISAGIO(), DISAGIO(), RENDITEDIS(), ZINSSATZ()

Das Beispiel zur *Wechselrechnung* finden Sie auf dem gleichnamigen Arbeitsblatt der Arbeitsmappe *Einfache Zinsrechnung.xlsx* im Ordner *Ms5-235\Kap14*.

Hinweis

**Praxiseinsatz
Wechsel-
rechnung**

Siehe auch



BW() PV()

BW(*Zins*; *Zzr*; *Rmz*; *Zw*; *F*)

Die Funktion BW() berechnet den Barwert eines regelmäßigen Zahlungsstroms unter Beachtung eventueller einmaliger Zahlungen am Ende des betrachteten Zeitraums nach dem finanzmathematischen Äquivalenzprinzip:

Leistung des Schuldners + Leistung des Gläubigers = 0

Zins (erforderlich) legt den (konstanten) Periodenzinssatz als nachschüssigen Zinssatz fest.

Zzr (erforderlich) bestimmt die Anzahl der Zinsperioden. Dabei wird davon ausgegangen, dass eventuelle regelmäßige Zahlungen (Argument *Rmz* ist größer Null) am Ende oder Anfang der Zinsperioden erfolgen.

Rmz (erforderlich/optional, siehe Hinweis) informiert über die Höhe der regelmäßigen Zahlung und kann als Rente oder Annuität interpretiert werden.

Syntax

Definition

Argumente

Z_w (optional/erforderlich, siehe Hinweis) ist der Kontostand, der sich am Ende des Vorgangs einstellen soll (etwa ein Restguthaben bei Auszahlungsplänen oder eine Abschlusstilgung in Höhe der Restschuld bei Krediten).

F (optional) legt fest, ob die regelmäßigen Zahlungen am Ende der Perioden ($F = 0$ oder nicht angegeben) oder aber am Anfang der Perioden ($F = 1$) erfolgen.

Hinweis Wird das Argument Rmz weggelassen, ist die Angabe von Z_w zwingend. Ist Z_w vorhanden, kann Rmz weggelassen werden. Das Weglassen der Argumente wirkt so, als ob sie mit 0 (Null) angegeben wurden.

Wichtig Das genannte finanzmathematische Äquivalenzprinzip bedeutet, dass hinsichtlich des Vorzeichens zwischen Aus- und Einzahlungen – Kreditaufnahme und Tilgung, Investition und Desinvestition (auch Devestition) – unterschieden werden muss. Hier weicht Excel vom in der finanzmathematischen Literatur gebräuchlichen Äquivalenzprinzip

Leistung des Schuldners = Leistung des Gläubigers
ab.

Hintergrund Die fünf Funktionen $BW()$ = Barwert, $ZW()$ = Zukunftswert, $RMZ()$ = regelmäßige Zahlung, $ZZR()$ = Zins- oder Zahlungszeiträume und $ZINS()$ = Zinssatz stehen in Umsetzung des obigen Äquivalenzprinzips in folgendem Zusammenhang:

$$BW \cdot (1 + ZINS)^{ZZR} + RMZ \cdot (1 + F \cdot ZINS) \cdot \frac{(1 + ZINS)^{ZZR} - 1}{ZINS} + ZW = 0$$

Der Barwert wird bis zum Ende aufgezinst, ebenso die regelmäßigen Zahlungen. Zum Ende wird die Summe mit dem Zukunftswert verglichen.

Die Nutzung einer dieser Funktionen ist gleichbedeutend mit der jeweiligen finanzmathematischen Grundaufgabe: Berechnung eines Werts aus obiger Gleichung, wenn die anderen Werte bekannt sind. Die Funktionen stellen also die Gleichung nach der gleichnamigen Größe um (außer bei $ZINS$ – hier muss eine Näherungsrechnung durchgeführt werden).

Hinweis In der Praxis erfolgt die Angabe des Zinssatzes nahezu immer als Jahreszinssatz. Die genannten Funktionen arbeiten nur dann korrekt, wenn die Perioden zum Zinssatz passen. Bei unterjährlicher Verzinsung wird in der Regel der Jahreszinssatz gleichmäßig auf die Perioden aufgeteilt: 12 Monate zu einem Zwöftel des Zinssatzes, 3 Monate zu einem Viertel und ein halbes Jahr zur Hälfte.

Praxiseinsatz Die folgenden Beispiele orientieren sich in den Überschriften am »finanzmathematischen Sprachgebrauch«.

Zinseszinsrechnung Jemand möchte im Alter ein kleines finanzielles Polster haben und beschließt, eine gerade angefallene Erbschaft in Höhe von 10.000,00 € auf 15 Jahre zu einem Zinssatz von 5 % fest anzulegen. Er hofft auf eine Auszahlung von wenigsten 25.000,00 €. Gelingt der Plan?

Eine Berechnung nach

$$=BW(5\%;15;;25000)$$

liefert –12.025,43 €. Das heißt, das ist der Betrag, der einzuzahlen wäre (deshalb das Minuszeichen), damit nach 15 Jahren die geforderten 25.000,00 € aufgelaufen sind. Der Plan misslingt also.

Jemand hat bis zum 60. Lebensjahr 100.000,00 € angespart und möchte sich eine zusätzliche Rente in Höhe von 750,00 € monatlich auszahlen lassen. Reicht das angesparte Geld, wenn 4,5 % p.a. auf das noch vorhandene Guthaben gezahlt werden und 15 Jahre Zahlungen geplant sind?

Sie ermitteln den Barwert der Rente aus

$$=BW(4,5\%/12;15*12;750)$$

zu -98.040,08 €. Es bleibt also nach 15 Jahren sogar ein kleiner Rest, da die vorhandene Summe größer ist. Das Minuszeichen interpretieren Sie als Verzicht auf sofortige Auszahlung.

Die Rechnung geht von Zinseszinsen innerhalb eines Jahrs aus. Auf einem normalen Sparbuch lässt sich der Zinseszinsseffekt allerdings nicht umsetzen, sodass das Ergebnis etwas »theoretisch« ist.

Jemand kann monatlich 1.000,00 € zur Rückzahlung (Tilgung plus Zinsen) eines Kredits einsetzen. Wie hoch kann dieser ausfallen, wenn an eine Laufzeit von 30 Jahren zum festen Zinssatz von 5,5 % p.a. gedacht ist?

Auch hier führt die Barwertformel zum Ziel (Tilgungsrechnung in diesem Sinne ist Rentenrechnung). Sie ermitteln 176.121,76 € über

$$=BW(5,5\%/12;30*12;-1000)$$

Anders als etwa beim Sparbuch erfolgt bei Hypothekendarlehen tatsächlich eine monatliche Verzinsung zu einem Zwölftel des vereinbarten Jahreszinssatzes (Nominalzins).

Die Verwendung von Formeln unterscheidet sich von der ausführlichen Notierung eines Kontoverlaufs mit den gegebenen Ein- und Auszahlungen möglicherweise durch Rundungsfehler. Das liegt daran, dass auf einem tatsächlich geführten Konto Rundungen auf zwei Stellen nach dem Komma erfolgen. Gestalten Sie ein solches Konto unter Excel nach, ist in den Zwischenschritten, die konkretes Geld bedeuten, jeweils die Funktion RUNDEN() einzusetzen. Eine ledigliche Begrenzung der Anzeige des Zellwerts auf zwei Stellen ist oft nicht korrekt.

Die Kapitalwertmethode der Investitionsrechnung bestimmt den Barwert zukünftiger Einnahmeüberschüsse und stellt diesen den Anschaffungskosten gegenüber. Gerechnet wird dabei mit einem Kalkulationszinssatz von (oft) 10 %. Fallen die Einnahmeüberschüsse als gedachte regelmäßige jährliche Zahlungen an und verfügt das Investitionsgut nach der Desinvestitionsdauer über einen Restwert, kann der Kapitalwert mittels BW() nach dem Muster

$-BW(10\%;\text{Anzahl der Jahre};\text{Jährlicher Einnahmeüberschuss};\text{Restwert})$ -Anschaffungskosten ermittelt werden.

Die Investition ist ökonomisch sinnvoll, wenn der berechnete Betrag positiv ausfällt.

Einige Anwendungen der Funktion BW() finden Sie auch den Beispielen zu KURS() und RENDITE().

RMZ(), ZINS(), ZINSZ(), ZW(), ZZR(), KAPZ(), KUMKAPITAL(), NBW(), PDURATION()

Die Arbeitsmappen *Zinseszinsrechnung.xlsx*, *Rentenrechnung.xlsx* und *Tilgungsrechnung.xlsx* im Ordner *Ms5-235\Kap14* halten die Beispiele auf dem jeweiligen Arbeitsblatt *BW* bereit.

Rentenrechnung

Tilgungsrechnung (Annuitätentilgung)

Wichtig

Investitionsrechnung

Kursrechnung

Siehe auch



DIA() SYD()

Syntax DIA(*Ansch_Wert*; *Restwert*; *Nutzungsdauer*; *Zr*)

Definition Diese Funktion ermittelt die Abschreibungsbeträge eines Wirtschaftsguts nach der arithmetisch-degressiven Methode (auch digitale Methode genannt).

Argumente *Ansch_Wert* (erforderlich) verlangt die Anschaffungskosten (Nettokaufpreis plus Anschaffungsnebenkosten minus Anschaffungskostenminderungen) eines Wirtschaftsguts. Verwenden Sie keine Zahl, erscheint der Fehlerwert #WERT!. Ist die eingegebene Zahl negativ, lassen sich die berechneten Werte nicht interpretieren.

Restwert (erforderlich) gibt den Wert des Wirtschaftsguts am Ende der Abschreibungsdauer an. Verwenden Sie keine Zahl, erscheint der Fehlerwert #WERT!. Ist die eingegebene Zahl negativ, erfolgt die Fehlermeldung #ZAHL!.

Nutzungsdauer (erforderlich) verlangt die Anzahl der Perioden, in denen das Wirtschaftsgut abgeschrieben wird. Sie sollten für dieses Argument eine ganze Zahl größer 0 (Null) verwenden.

Zr (erforderlich) gibt die Periode innerhalb der Abschreibungsdauer an, für die der Abschreibungsbetrag ermittelt werden soll. Auch hier sollten Sie eine ganze Zahl größer 0 (Null) verwenden.

Hintergrund Abschreibung hat den Sinn, den Werteverlust eines Wirtschaftsguts zu ermitteln und sichtbar zu machen. Davon zu unterscheiden ist der Begriff der Absetzung für Abnutzung, der die Kosten für den Erwerb eines Wirtschaftsguts als Betriebsausgaben steuerlich erfasst.

Arithmetisch-degressive Abschreibung beruht auf einer konstanten Minderung des Abschreibungsbetrags von Periode zu Periode. Der erste Abschreibungsbetrag wird aus der Formel

$$\text{erste Abschreibung} = \frac{2 \cdot (\text{Anschaffungskosten} - \text{Restwert})}{\text{Abschreibungsdauer} + 1}$$

ermittelt. Der konstante Betrag, der die Abschreibungsbeträge jeweils mindert, entsteht aus der Teilung des ersten Abschreibungsbetrags durch die Gesamtzahl der Abschreibungsperioden.

Wichtig Die arithmetisch-degressive Abschreibungsmethode ist im deutschen Steuerrecht nicht zulässig.

Praxiseinsatz Ein Wirtschaftsgut mit Anschaffungskosten von 1.000,00 € soll in 5 Jahren arithmetisch-degressiv auf einen Restwert von 0 € abgeschrieben werden. Sie können hierzu in jeder Periode den Abschreibungsbetrag mittels DIA() berechnen und vom Buchwert der Vorperiode abziehen.

Alternativ können Sie aber auch einen Abschreibungsplan erstellen, der die oben genannten Formeln zur Berechnung des ersten Abschreibungsbetrags sowie die Vorgehensweise zur Berechnung der anderen Beträge umsetzt. Die Erstellung eines Abschreibungsplans nach eigenem Muster ist vor allem bei der Anwendung steuerlich zulässiger Methoden (Zwölfstel-Regelung seit 2004) zu empfehlen, da die integrierten Funktionen im Allgemeinen nicht anwendbar sind.

Einzelheiten zu diesem Beispiel entnehmen Sie bitte der zugehörigen Arbeitsmappe.

Siehe auch GDA(), GDA2(), LIA(), VDB(), AMORLINEARK()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Abschreibungsrechnung.xlsx* auf dem Arbeitsblatt *DIA*.

DISAGIO()



DISAGIO(*Abrechnung;Fälligkeit;Kurs;Rückzahlung;Basis*)

Diese Funktion berechnet den vorschüssigen Zinssatz (prozentualer Abschlag, Diskont, Disagio) bei vorgegebenem Bar- bzw. Endwert und vorgegebener Laufzeit (einfache Verzinsung).

Abrechnung (erforderlich) ist der Tag, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist der Tag, an dem die Leistungen aus dem Wertpapier erfolgen müssen.

Kurs (erforderlich) gibt an, zu welchem Preis der Besitzwechsel erfolgt.

Rückzahlung (erforderlich) beschreibt die Leistung am Fälligkeitstag.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.4. Wird dieses Argument weggelassen, rechnet Excel mit $Basis = 0$.

Basis	Methode	Bedeutung
0	30/360 (NASD-Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Ist das Ausgangsdatum der 31. eines Monats, wird dieses Datum zum 30. desselben Monats. Ist das Enddatum der 31. eines Monats und das Ausgangsdatum ein Datum vor dem 30. eines Monats, wird das Enddatum zum 1. des darauf folgenden Monats. In allen anderen Fällen wird das Enddatum zum 30. desselben Monats.
1	Genau/Genau	Jeder Monat hat so viele Tage wie im Kalender stehen, die Summe über alle Monate ergibt die Anzahl der Tage im Jahr
2	Genau/360 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 360 Tage.
3	Genau/365 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 365 Tage.
4	30/360 (Europäische Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Jedes Ausgangs- und Enddatum, das auf den 31. eines Monats fällt, wird zum 30. desselben Monats.

Syntax

Definition

Argumente

Tabelle 14.4

Verschiedene Methoden zur Ermittlung der Anzahl der Tage eines Jahrs

Alle Argumente der Funktion, die ein Datum betreffen, verwenden eine Datumsangabe ohne Uhrzeit, gebrochene Zahlen werden also abgerundet. Das Argument *Basis* verlangt ebenfalls eine ganze Zahl und schneidet Nachkommastellen ab.

Werden ungültige Datumseinträge verwendet bzw. keine Zahlen dort angegeben, wo sie erforderlich sind, liefert die Funktion den Fehlerwert *#WERT!*. Im Falle ungültiger Zahlen für die Nicht-Datums-Argumente lautet die Fehlermeldung *#ZAHL!*.

Das Prinzip der vorschüssigen Verzinsung besteht in der Annahme, dass sich ein Anfangskapital (einzuzahlender Betrag, Darlehen) aus dem Rückzahlungsbetrag minus den Zinsen auf diesen Betrag ergeben. Dieses Prinzip wird vor allem im unterjährlichen Bereich verwendet und unterscheidet sich von dem Prinzip, welches einem normalen Sparbuch oder einem Hypothekendarlehen zugrunde liegt. Dort werden nämlich die Zinsen am Ende einer Periode auf Basis des Anfangskapitals berechnet (nachsüssige Verzinsung).

Der Hinweis in der Excel-Hilfe auf den Nennwert in den Argumenten *Kurs* und *Rückzahlung* ist irrelevant, da DISAGIO() eine Verhältniszahl liefert.

Hintergrund

Hinweis Zwischen den Funktionen zu diesem Themenkreis besteht die Verbindung

$$AUSZAHLUNG() - AUSZAHLUNG() \cdot DISAGIO() \cdot \frac{TAGE360()}{360} = KURSDISAGIO()$$

(hier notiert für den Fall von *Basis* = 4). Das ist aber gerade das Prinzip vorschüssiger Verzinsung.

Die Funktion DISAGIO() löst die obige Gleichung für den Fall auf, dass die anderen Angaben vorliegen.

Die Funktion DISAGIO() steht mit der Funktion ZINSSATZ() in folgender Beziehung:

$$(1 + ZINSSATZ()) \cdot (1 - DISAGIO()) = 1$$

Diese Beziehung erlaubt es, aus einem vorschüssigen Jahreszinssatz den dazu äquivalenten nachschüssigen Jahreszinssatz auszurechnen. Äquivalenz bedeutet dabei, dass zwei Anleger, die das gleiche Kapital einsetzen – der eine mit vorschüssiger, der andere mit nachschüssiger Verzinsung – am Ende eines Jahrs auch eine gleiche Rückzahlung bekommen. Im Wesentlichen gleichwertig zu ZINSSATZ() ist die Funktion RENDITEDIS(), die sich nur in der Bezeichnung der Argumente unterscheidet.

Praxiseinsatz Am 10.5.2013 legt ein Geschäftsmann seiner Bank einen Wechsel mit einer Wechselsumme über 5.000,00 € und einer (Rest-)Laufzeit von 2 Monaten vor. Diese schreibt (ohne Gebühren zu fordern) dem Konto 4.958,33 € gut. Wie hoch ist der Diskontsatz?

Wechselrechnung

Sie lesen die Gleichung aus den Hintergrundinformationen als

$$\text{Wechselsumme} - \text{Wechselsumme} \cdot \text{Diskontsatz} \cdot \frac{TAGE360()}{360} = \text{Gutschrift}$$

und verwenden

=DISAGIO("10.5.2013";"10.7.2013";4958,33;5000;4)

zur Erlangung des Ergebnisses von 5 %.

Natürlich stehen die konkreten Zahlen in Zellen und die Formel verwendet Zellbezüge.

Siehe auch KURSDISAGIO(), RENDITEDIS(), AUSZAHLUNG()



Die Beispiele zur Wechselrechnung finden Sie auf den gleichnamigen Arbeitsblättern der Arbeitsmappe *Einfache Zinsrechnung.xlsx* im Ordner \Ms5-235\Kap14.

DURATION() DURATION()

Syntax DURATION(*Abrechnung*; *Fälligkeit*; *Nominalzins*; *Rendite*; *Häufigkeit*; *Basis*)

Definition Diese Funktion berechnet die (durchschnittliche) Kapitalbindungsdauer eines festverzinslichen Wertpapiers. Nach dem Entwickler dieser Kennzahl wird sie auch manchmal als Macauley-Duration bezeichnet.

Argumente *Abrechnung* (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum, an dem die Rückzahlung des durch das Wertpapier verbrieften Darlehens erfolgt.

Nominalzins (erforderlich) informiert über den vereinbarten Jahreszinssatz als Preis für das geliehene Geld.

Rendite (erforderlich) ist der am Abrechnungstag geltende Marktzinssatz, zu dem bei der Berechnung der Duration alle zukünftigen Zahlungen abgezinst werden.

Da es durchaus üblich ist, festverzinsliche Wertpapiere mit mehreren Zinsterminen im Jahr auszustatten, informiert das erforderliche Argument **Häufigkeit** über deren Anzahl. Zulässige Werte sind: 1 = jährlich; 2 = halbjährlich; 4 = vierteljährlich.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.4. Wird es weggelassen, rechnet Excel mit $Basis = 0$.

Alle Argumente der Funktion, die ein Datum betreffen, verwenden eine Datumsangabe ohne Uhrzeit, gebrochene Zahlen werden also abgerundet. Die Argumente **Häufigkeit** und **Basis** verlangen ebenfalls eine ganze Zahl und schneiden Nachkommastellen ab.

Werden ungültige Datumseinträge verwendet bzw. keine Zahlen dort angegeben, wo sie erforderlich sind, liefert die Funktion den Fehlerwert **#WERT!**. Im Falle ungültiger Zahlen für die Nicht-Datums-Argumente lautet die Fehlermeldung **#ZAHL!**.

Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Wichtig

Die Duration nach Macauley ist ein gewichteter arithmetischer Mittelwert. Es werden dabei die Barwerte der garantierten zukünftigen Erträge (Zinsen und Rückzahlung) aus dem aktuellen Marktzins für die Gesamtlaufzeit bis zur Fälligkeit durch Abzinsen berechnet und mit dem zeitlichen Abstand bis zu ihrer Zahlung gewichtet. Durch etwas kompliziertere mathematische Betrachtungen kann dann bewiesen werden, dass die so berechnete »durchschnittliche Kapitalbindungsdauer« gleichzeitig den Zeitpunkt bestimmt, bei dem eine (theoretische) Immunsierung gegen Zinsänderungen eintritt. Der mit dem aktuellen Marktzins berechnete Zukunftswert aller Zahlungen zum Zeitpunkt, der durch die Duration geliefert wird, ist der Mindestwert, der unabhängig von sich ändernden Marktzinsen zu diesem Zeitpunkt in der Tat erzielt wird.

Hintergrund

Das Ergebnis wird nun verwendet, um Anleihen (festverzinsliche Wertpapiere) mit nahezu gleichen Konditionen (Rendite und Laufzeit) miteinander zu vergleichen. Der Anleihe mit der geringeren Kapitalbindungsdauer ist beim Erwerb der Vorzug zu geben, da erwartet wird, das Risiko minimal zu halten.

Die Kennzahl ist natürlich kein Garant für die Zukunft. Kurse und damit Renditen am Markt stellen sich durch Angebot und Nachfrage ein, deren Ursachen sehr unterschiedlich sein können. Eine heute getroffene Bewertung wird deshalb in nicht ferner Zukunft wieder korrigiert werden. Ein mögliches Kriterium kann auch dann wieder die Duration sein.

Praxiseinsatz Am 29.1.2013 gab es für zwei Bundeswertpapiere folgende Informationen (<http://www.deutsche-finanzagentur.de>):

Wertpapier	Nominalzins	Fälligkeit	Kurs	Rendite
Bundesanleihe von 05	3,50 %	4.7.2015	109,354	0,37 %
Bundesobligation Serie 159	2,00 %	26.2.2016	106,806	0,38 %

Eine Berechnung der Duration ergibt Folgendes:

Wertpapier	Duration
Bundesanleihe von 05	2,33 Jahre
Bundesobligation Serie 157	2,96 Jahre

Es ist die Bundesanleihe vorzuziehen. Der Unterschied scheint hier bezogen auf die Laufzeit doch recht gering. Für andere Schuldner und andere Konditionen sowie unter Einbeziehung steuerlicher Aspekte (die Rendite ist in Abhängigkeit vom Steuersatz zu reduzieren) kann sich auch ein anderer Risikovorteil abzeichnen.

Siehe auch MDURATION(), RENDITE(), KURS()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Kursrechnung.xlsx* auf dem Arbeitsblatt *DURATION*.

EFFEKTIV() EFFECT()

Syntax EFFEKTIV(*Nominalzins*; *Perioden*)

Definition Diese Funktion berechnet aus einem (nachsüssigen) Nominalzinssatz⁵, der auf verschiedene, gleich lange Perioden innerhalb eines Jahrs gleichmäßig aufgeteilt wird (das führt zum Begriff des relativen Zinssatzes), den dazu äquivalenten effektiven (nachsüssigen) Jahreszinssatz (auch konformer Zinssatz genannt). Obwohl nicht üblich, kann hier das »Jahr« auch durch eine andere Zeiteinheit ersetzt werden.

Argumente *Nominalzins* (erforderlich) gibt den Jahreszinssatz an, der durch Aufteilung auf die Perioden zu einer jeweils unterjährlichen Verzinsung genutzt wird.

Perioden (erforderlich) ist die Anzahl der Zinstage im Jahr. Gewöhnlich sind dies 2, 4 oder 12, in speziellen Fällen auch 365.

Jedes der Argumente muss sich als Zahl auswerten lassen, ansonsten erscheint der Fehlerwert #WERT!. Das Argument *Perioden* wird durch Abschneiden von Nachkommastellen stets zu einer ganzen Zahl gemacht. Ist *Nominalzins* kleiner oder gleich 0 (Null) oder *Perioden* kleiner 1, gibt EFFEKTIV() den Fehlerwert #ZAHL! zurück.

Hintergrund Bei verschiedenen Finanzgeschäften (Hypothekendarlehen, Bausparen, Guthabenverzinsung bei bestimmten Girokonten, Kontokorrentkonten, Dispokredite u.a.m.) wird zwar ein Jahreszinssatz angegeben, dieser dient aber nur zur Vorbereitung weiterer Modalitäten. Diese beste-

⁵ Hierfür hat sich in letzter Zeit auch der Begriff »Sollzinssatz« durchgesetzt.

hen darin, dass nicht jährlich verzinst wird, sondern in unterjährlichen Perioden. Der dann angewendete Zinssatz entsteht durch Teilung des Nominalzinssatzes durch die Periodenzahl.

Um Vergleichbarkeit verschiedener Konditionen zu haben, wird der Zinssatz, der für einen feststehenden Betrag bei einmaliger Verzinsung das gleiche Ergebnis wie die unterjährliche Verzinsung liefert, als effektiver Jahreszinssatz bezeichnet. Zwischen beiden Zinssätzen besteht die Beziehung

$$1 + \text{Effektivzinssatz} = \left(1 + \frac{\text{Nominalzinssatz}}{\text{Periodenzahl}} \right)^{\text{Periodenzahl}}$$

Ein Sparer hat auf einem Konto die Möglichkeit monatlicher Verzinsung und dort rechtzeitig vor Jahresanfang 1.000,00 € eingezahlt. Der angegebene nominale Jahreszinssatz beträgt 5 %. Wie hoch ist das Guthaben am Ende des Jahrs, welche Effektivverzinsung liegt vor?

Zur Lösung der Aufgabe können Sie ein Konto notieren, auf welchem die Zinsgutschriften monatlich erfolgen (achten Sie dabei auf korrektes Runden auf zwei Stellen nach dem Komma). Der Kontostand am Ende beträgt 1.051,16 €.

Zum gleichen Ergebnis gelangen Sie, wenn Sie die Funktion ZW() mit den zutreffenden Argumenten benutzen:

=ZW(5%/12;12;;-1000)

Setzen Sie den Kontoendstand zum eingesetzten Kapital ins Verhältnis, ergibt sich eine Verzinsung mit 5,12 %. Das ist aber genau das, was auch =EFFEKTIV(5%;12) ausrechnet.

Eine Bank wirbt mit einem Hypothekendarlehen, welches einen Nominalzins (Sollzinssatz) von 2,42 % angibt und bei monatlicher Tilgung und bei vollständiger Kreditauszahlung (ohne Disagio) einen (anfänglichen) effektiven Jahreszinssatz von 2,45 % bei fünfjähriger Zinsbindung verspricht.

Sie können diesen effektiven Jahreszinssatz in diesem Fall zwar mithilfe von EFFEKTIV() nachbilden, aber das ist nicht die Art, wie die Bank die Zahl ermittelt. Die Übereinstimmung bis auf zwei Stellen nach dem Komma ist Zufall. Die Effektivzinsermittlung muss nach der seit dem Jahr 2002 in Deutschland geltenden Preisangabenverordnung vorgenommen werden.

Diese Verordnung ist Nachfolger jener aus dem Jahre 1985, in welcher gefordert wurde, Kredite auf einem nach deutschen Gepflogenheiten eingerichteten Sparbuch (30/360 Tage) nachzubilden. Der Zinssatz, der dann Schuldner und Gläubiger gleichstellte, wurde als (anfänglicher) effektiver Jahreszinssatz bezeichnet. Anfänglich in den Fällen, in denen er nur die Zeit der Zinsbindung berücksichtigte.

Die Verordnung des Jahrs 2002 fordert zur Bewertung international übliche Tagezählung (taggenau/365) und die weitgehende Verwendung gebrochener Laufzeiten zum Erfassen von Zinseszinsseffekten. Eine genaue Diskussion kann in diesem Buch nicht erfolgen. Excel stellt für Berechnungen nach der Preisangabenverordnung keine Funktionen bereit, jedoch lassen sich die Anforderungen nach eingehendem Studium der Verordnung in umfangreicheren Arbeitsblättern umsetzen.

NOMINAL(), RENDITE(), ZINS()

Die beiden Beispiele finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Zinseszinsrechnung.xlsx* auf dem Arbeitsblatt *EFFEKTIV*.

**Praxiseinsatz
Ein Spar-
vorgang**

**Hypotheken-
darlehen**

Siehe auch



GDA() DDB()

Syntax GDA(*Ansch_Wert*; *Restwert*; *Nutzungsdauer*; *Periode*; *Faktor*)

Definition Diese Funktion berechnet die Abschreibungsbeträge nach der Methode einer Mehrfachraten-Abschreibung.

Argumente *Ansch_Wert* (erforderlich) verlangt die Anschaffungskosten (Nettokaufpreis plus Anschaffungsnebenkosten minus Anschaffungskostenminderungen) eines Wirtschaftsguts. Verwenden Sie keinen als Zahl auswertbaren Wert, erscheint der Fehlerwert #WERT!. Ist die eingegebene Zahl negativ, erfolgt die Fehlermeldung #ZAHL!.

Restwert (erforderlich) entspricht dem Wert des Wirtschaftsguts am Ende der Abschreibungsdauer. Verwenden Sie keine Zahl, erscheint der Fehlerwert #WERT!. Ist die eingegebene Zahl negativ, erfolgt die Fehlermeldung #ZAHL!.

Nutzungsdauer (erforderlich) verlangt die Anzahl der Perioden, in denen das Wirtschaftsgut abgeschrieben wird. Sie sollten für dieses Argument eine ganze Zahl größer 0 (Null) verwenden.

Periode (erforderlich) ist die Periode innerhalb der Abschreibungszeit, für die der Abschreibungsbetrag ermittelt wird. Es wird eine positive ganze Zahl verlangt, die nicht größer als *Nutzungsdauer* ist.

Faktor (optional) ist der Multiplikator für den (gedachten linearen) Abschreibungssatz (Reziprokes der Abschreibungsdauer). Wird er nicht angegeben, rechnet Excel mit der Zahl 2.

Hintergrund Abschreibung hat den Sinn, den Werteverlust eines Wirtschaftsguts zu ermitteln und sichtbar zu machen. Davon zu unterscheiden ist der Begriff der Absetzung für Abnutzung, der die Kosten für den Erwerb eines Wirtschaftsguts als Betriebsausgaben steuerlich erfasst.

Die Methode zielt zunächst auf lineare Abschreibung. Der so ermittelte Abschreibungsprozentsatz als Reziprokes der Abschreibungsdauer wird allerdings mit einem weiteren Faktor multipliziert und auf den jeweiligen Buchwert angewendet. Dadurch kommt es im Endeffekt zu geometrischer Abschreibung. Im Fall eines verwendeten Faktors gleich 2 heißt das Verfahren »Methode der degressiven Doppelraten-Abschreibung«.

Wichtig Die hier genannte Abschreibungsmethode ist im deutschen Steuerrecht nicht zulässig.

Praxiseinsatz Ein Wirtschaftsgut mit Anschaffungskosten von 1.000,00 € soll in 5 Jahren nach der genannten Methode auf einen Restwert von 100,00 € abgeschrieben werden (dieser Restwert hat für die Berechnung allerdings keine Bedeutung). Sie können dazu in jeder Periode den Abschreibungsbetrag mittels GDA() berechnen und vom Buchwert der Vorperiode abziehen.

Alternativ können Sie aber auch einen Abschreibungsplan erstellen, der die oben genannte Vorgehensweise zur Berechnung umsetzt. Die Erstellung eines Abschreibungsplans nach eigenem Muster ist vor allem bei der Anwendung steuerlich zulässiger Methoden (Zwölfstel-Regelung seit 2004) zu empfehlen, da die integrierten Funktionen im Allgemeinen nicht anwendbar sind.

Einzelheiten zu diesem Beispiel entnehmen Sie bitte der zugehörigen Arbeitsmappe.

Siehe auch DIA(), GDA2(), LIA, VDB(), AMORLINEARK()



Das Beispiel finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Abschreibungsrechnung.xlsx* auf dem Arbeitsblatt *GDA*.

GDA2() DB()

GDA2(*Ansch_Wert*; *Restwert*; *Nutzungsdauer*; *Periode*; *Monate*)

Diese Funktion berechnet Abschreibungsbeträge für ein Wirtschaftsgut nach der Methode der geometrisch-degressiven Abschreibung, wobei der Bruchteil eines Jahrs (in vollen Monaten) in der ersten Abschreibungsperiode berücksichtigt wird.

Ansch_Wert (erforderlich) verlangt die Anschaffungskosten (Nettokaufpreis plus Anschaffungsnebenkosten minus Anschaffungskostenminderungen) eines Wirtschaftsguts. Verwenden Sie keinen als Zahl auswertbaren Wert, erscheint der Fehlerwert #WERT!. Ist die eingegebene Zahl negativ, erfolgt die Fehlermeldung #ZAHL!.

Restwert (erforderlich) entspricht dem Wert des Wirtschaftsguts am Ende der Abschreibungsdauer. Verwenden Sie keine Zahl, erscheint der Fehlerwert #WERT!. Ist die eingegebene Zahl negativ, erfolgt die Fehlermeldung #ZAHL!.

Nutzungsdauer (erforderlich) verlangt die Anzahl der Perioden, in denen das Wirtschaftsgut abgeschrieben wird. Sie sollten für dieses Argument eine ganze Zahl größer 0 (Null) verwenden.

Periode (erforderlich) ist die Periode innerhalb der Abschreibungszeit, für die der Abschreibungsbetrag ermittelt wird. Es wird eine positive ganze Zahl verlangt, die nicht größer als *Nutzungsdauer* ist.

Monate (optional) gibt die Dauer einer zu berücksichtigenden Teilperiode im Anschaffungsjahr in ganzen Monaten an. Wird es weggelassen, rechnet Excel mit dem vollen Jahr (12 Monate).

Abschreibung hat den Sinn, den Werteverlust eines Wirtschaftsguts zu ermitteln und sichtbar zu machen. Davon zu unterscheiden ist der Begriff der Absetzung für Abnutzung, der die Kosten für den Erwerb eines Wirtschaftsguts als Betriebsausgaben steuerlich erfasst.

Für den geometrisch-degressiven Abschreibungsprozentsatz gilt die Formel

$$\text{Abschreibungssatz} = 1 - \sqrt[\text{Abschreibungsdauer}]{\frac{\text{Restwert}}{\text{Anschaffungskosten}}}$$

die auch erklärt, weshalb ein Restwert von 0 (Null) ohne Sinn ist – die Abschreibung würde vollständig im ersten Jahr erfolgen. In der Regel kann in solchen Fällen von einem Erinnerungswert von 1,00 € ausgegangen werden.

Der berechnete Abschreibungssatz wird (in Excel) auf drei Stellen nach dem Komma gerundet und jede Periode auf den Buchwert angewendet. Der so bestimmte Abschreibungsbetrag mindert den Buchwert für die nächste Periode.

Ist die erste Periode kein ganzes Jahr lang, wird der Abschreibungssatz für diese Periode entsprechend gewölftelt.

Die geometrisch-degressive Abschreibungsmethode war im deutschen Steuerrecht nur noch bis 2007 zulässig, und dies auch nur mit Einschränkungen. So darf der Abschreibungsprozentsatz, der für die Zeit der geometrisch-degressiven Abschreibung konstant bleiben muss, maximal das doppelte des linearen Abschreibungssatzes betragen und dabei 20 % nicht übersteigen. Dies wurde für kurze Zeit aufgehoben. Im Jahre 2010 gab es wieder diese Abschreibemöglichkeiten, seit 2011 nicht mehr.

Die Funktion GDA2() ist zur Berechnung steuerlicher Absetzungsbeiträge in diesem Sinne nicht geeignet.

Syntax

Definition

Argumente

Hintergrund

Hinweis

Praxiseinsatz Ein Wirtschaftsgut mit Anschaffungskosten von 1.000,00 € soll in 5 Jahren geometrisch-degressiv auf einen Restwert von 100,00 € abgeschrieben werden. Sie können hierzu in jeder Periode den Abschreibungsbetrag mittels GDA2() berechnen und vom Buchwert der Vorperiode abziehen.

Alternativ können Sie aber auch einen Abschreibungsplan erstellen, der die oben genannten Formeln zur Berechnung des ersten Abschreibungsbetrags sowie die Vorgehensweise zur Berechnung der anderen Beträge umsetzt.

Einzelheiten zu diesem Beispiel entnehmen Sie bitte der zugehörigen Arbeitsmappe. Diese enthält auch einen Abschreibungsplan, der bis 2007 und 2009/2010 in Deutschland steuerlich zulässig ist. Die Bestimmung des linearen Abschreibungssatzes geschieht dabei nach der Formel

$$\text{Abschreibungssatz} = \frac{\text{Anschaffungskosten} - \text{Restwert}}{\text{Anschaffungskosten}} \cdot \frac{1}{\text{Abschreibungsdauer}}$$

Siehe auch DIA(), GDA(), LIA(), VDB(), AMORLINEARK()



Das Beispiel mit seinen Ergänzungen finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Abschreibungsrechnung.xlsx* auf dem Arbeitsblatt *GDA2*.

IKV()



IRR()

Syntax IKV(*Werte*; *Schätzwert*)

Definition Diese Funktion berechnet eine Kenngröße der dynamischen Investitionsrechnung: den internen Zinssatz. Dieser macht den Barwert aller Aus- und Einzahlungen in Zusammenhang mit einer Investition zu Null.

Argumente *Werte* (erforderlich) sind die lückenlos in einer Spalte angeordneten (tatsächlichen und/oder erwarteten) Überschüsse aus Aus- und Einzahlungen. Jeder Wert vertritt eine Periode (in der Regel ein Jahr) in aufsteigender Ordnung und ohne Lücken. Negative Überschüsse werden mit einem Minuszeichen versehen.

Da eine Berechnung des internen Zinssatzes aus mathematischen Gründen für mehr als zwei Perioden nur näherungsweise erfolgen kann, kann diese Näherungsrechnung zu verschiedenen, teilweise auch unsinnigen (Zinssatz ist negativ) Ergebnissen führen. Sie können die Berechnung durch das optionale Argument *Schätzwert* als Ausgangspunkt für die Näherungsrechnung beeinflussen. Verzichten Sie auf das Argument, rechnet Excel mit einem Ausgangspunkt von 10 %. Das ist ein in der Praxis durchaus üblicher Kalkulationszinssatz, in dessen »Nähe« die Verzinsung einer ökonomisch realen Investition auch liegt.

Enthalten Zellen des Arguments *Werte* keine Zahlen oder sind leer, rechnet Excel so, als ob diese Zellen nicht vorhanden sind. Liefert die Funktion den Fehlerwert #ZÄH!, war die oben genannte Näherungsberechnung mit dem gedachten *Schätzwert* für den internen Zinssatz erfolglos. Das kann seine Ursache sowohl im eingesetzten *Schätzwert*, aber auch in den Daten selbst haben (etwa nur Auszahlungen oder nur Einzahlungen).

Hintergrund Als Kapitalwert einer Investition wird der Barwert aller Aus- und Einzahlungen (auch als Nettobarwert bezeichnet) verstanden. Dabei wird davon ausgegangen, dass die Zahlungen am Ende eines Jahrs erfolgen (die Startperiode bekommt die Nummer 0, in ihr liegt gewöhnlich nur eine Auszahlung vor). Diese Zahlungen werden zu einem Kalkulationszinssatz, der

in der Praxis bei 10 % plus/minus Risikozuschlägen und Risikoabschlägen liegt, abgezinst. Eine Ausnahme sind Finanzinvestitionen, wo mit der für die Laufzeit üblichen Markttrendite gerechnet werden kann.

Die interne Kapitalverzinsung (interner Zinssatz) ist nun jener Wert, der den Kapitalwert zu Null macht:

$$\sum_{n=0}^{\text{Dauer}} \frac{\text{Einzahlungen minus Auszahlungen im Jahre } n}{(1 + \text{IKV})^n} = 0$$

Eine besondere Form der Investition sind Investitionen, bei denen Geld ver- oder geliehen wird. Anstelle von interner Kapitalverzinsung spricht man bei Krediten von Effektivzins, bei Sparanlagen (hier ist der Sparer oder Anleger der Kreditgeber) von Rendite.

Die Anschaffungskosten für eine Maschine belaufen sich auf 80.000,00 €. Die erwarteten jährlichen Überschüsse (Einzahlungen minus Auszahlungen) werden wie in Tabelle 14.5 geschätzt.

Jahr	Überschuss (in €)
1	15.000
2	19.000
3	25.000
4	27.000
5	17.000
6	7.000

Praxiseinsatz Sachinvestition

Tabelle 14.5
Geschätzte jährliche Überschüsse beim Einsatz einer Maschine

Ist diese Investition empfehlenswert, wenn die Verzinsung der eingesetzten Summe mit wenigstens 10 % erfolgen soll?

Zur Lösung notieren Sie in der ersten Zeile die Anschaffungskosten (mit negativem Vorzeichen) und darunter lückenlos die Daten aus Tabelle 14.5. Die Anwendung von IKV() führt zu einem internen Zinssatz von 10,47 %, der leicht über dem geforderten liegt.

Beachten Sie, dass es bei Sachinvestitionen, deren zukünftige Überschüsse nur geschätzt sind, nicht unbedingt auf Stellen nach dem Komma ankommt.

Auf die gleiche Weise lassen sich Finanzinvestitionen bewerten. Jemand bekommt auf ein angelegtes Kapital nach einem Jahr 1,75 %, nach zwei Jahren 2,25% und nach 3 Jahren 3,8 % Zinsen. Wie hoch ist die Rendite, wenn die Zinsen dem Kapital nicht zugeschlagen werden?

Finanz- investition

Dieses Problem können Sie in Excel unmittelbar nachbilden. Der Aufbau der Tabelle kann dabei wie in Abbildung 14.1 aussehen, IKV() liefert das gewünschte Ergebnis.

	A	B	C	D	E	F
16						
17		Jahr	Zinssätze	Ausgaben	Einnahmen	Überschuss
18		0		100,00		-100,00
19		1	1,75%		1,75	1,75
20		2	2,25%		2,25	2,25
21		3	3,80%		103,80	103,80
22						
23				Rendite	IKV	2,58%

Abbildung 14.1: Renditeberechnung durch IKV()

Siehe auch NBW(), QIKV(), XINTZINSFUSS(), XKAPITALWERT(), BW(), ZINS()



Die beiden Beispiele finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Investitionsrechnung.xlsx* auf dem Arbeitsblatt *IKV*.

ISPMT()



ISPMT()

Syntax ISPMT(*Rate*; *Per*; *Nper*; *Pv*)

Definition Diese Funktion berechnet bei gegebenem Jahreszinssatz und einfacher nachschüssiger Verzinsung die Summe der Zinsen, die auf einen bestimmten Teil des Jahrs fallen. Es sind auch andere Zeiträume als ein Jahr denkbar, aber kaum üblich.

Sie wurde aus Gründen der Kompatibilität zu Lotus 1-2-3 aufgenommen.

Argumente *Rate* (erforderlich) ist der nachschüssige Jahreszinssatz, der gleichmäßig auf die Gesamtperiodenzahl im Jahr aufgeteilt werden soll.

Per (erforderlich) gibt die Nummer der Periode an (beginnend ab Null), von deren Ende aus die Zinsberechnung bis zum Jahresende erfolgen soll.

Nper (erforderlich) ist die Gesamtzahl der Perioden innerhalb des Jahrs und bestimmt den Teiler für den Jahreszinssatz.

Pv (erforderlich) ist der Gegenwartswert (Barwert), dessen nachschüssige Verzinsung erfolgen soll.

Hintergrund Die Aufteilung eines Jahreszinssatzes auf einen Periodenzinssatz unterjährlicher Zeitabschnitte ist ein oft durchgeführter Vorgang (Überziehung von Girokonten, Kontokorrentkonten, Hypothekendarlehen). Allerdings wird meist der unterjährliche Zinssatz benutzt, um auch innerhalb des Jahrs mit Zinseszinsseffekten zu verzinsen. Das wird durch die Funktion nicht berücksichtigt.

Praxiseinsatz Das, was die Funktion tut, lässt sich am besten an einem Sparbuch verdeutlichen. Bringen Sie am 30.4. eines Jahrs 100,00 € auf ein Sparbuch mit 6 % Zinsen, stehen Ihnen diese (falls der Betrag bis zum Jahresende auf dem Konto bleibt) für 8/12 des Jahrs zu. Den Zinsertrag können Sie durch

`=ISPMT(6%;4;12;-100)`

als 4,00 € ermitteln.

Bringen Sie den gleichen Betrag am 15.5. zur Kasse, sind bereits 135 Zinstage des Jahrs vergangen und Sie bekommen noch für 225 Tage Zinsen. Über

`=ISPMT(6%;135;360;-100)`

errechnen Sie einen Betrag von 3,75 €.

Siehe auch AUFGELZINS(), AUFGELZINSF(), KAPZ(), KUMZINS()



Das Beispiel finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* auf dem Arbeitsblatt *ISPMT*.

KAPZ()



PPMT()

KAPZ(*Zins*; *Zr*; *Zzr*; *Bw*; *Zw*; *F*)

Diese Funktion ermittelt den Anteil an einer Annuität, der zur Tilgung eines Darlehens benutzt wird, welches nach dem Prinzip der Annuitätentilgung zurückgezahlt wird.

Zins (erforderlich) benennt den Nominalzins des Kredits.

Zr (erforderlich) enthält die für die gewünschte Periode zutreffende Nummer.

Zzr (erforderlich) gibt die Gesamtzahl der Perioden, in denen der Kredit zurückgezahlt wird, an.

Bw (erforderlich) erfasst die Darlehenssumme.

Zw (optional) bewertet Darlehen, deren Rückzahlung nach der Gesamtperiodenzahl noch nicht vollständig erfolgt ist. Ein solcher Effekt tritt etwa ein, wenn Hypothekendarlehen mit einer Zinsbindungsfrist ausgezahlt werden.

F (optional) legt fest, ob die Zahlungen zu Beginn der Perioden zinswirksam werden ($F = 1$) oder erst an deren Ende ($F = 0$). Wird das Argument weggelassen, rechnet Excel so, als ob es 0 (Null) wäre.

Die Tilgung von Krediten kennt verschiedene Formen. Eine davon ist, dass der Schuldner jede Periode den gleichen Betrag zahlt. Dieser setzt sich aus einem Tilgungsanteil (der sich mit fortschreitender Zeit zunehmend vergrößert) und einem Zinsanteil (der sich immer mehr verkleinert) zusammen. Die Veränderung der Zusammensetzung hängt damit zusammen, dass die abzutragende Schuld mit der Zeit immer geringer wird. Diese Form wird als Annuitätentilgung bezeichnet, der Gesamtbetrag heißt – auch wenn seine Zahlung nicht unbedingt jährlich erfolgt – Annuität.

Der Tilgungsanteil der ersten Annuität ergibt sich aus dieser nach Abzug des Zinsanteils. Dieser ist aber gerade die Kreditsumme multipliziert mit dem für die Periode festgelegten Nominalzinssatz. Die Tilgungsanteile der weiteren Perioden bestimmen sich (nach einigen mathematischen Zwischenschritten) aus

$$(\text{Tilgungsanteil der ersten Periode}) \cdot (1 + \text{Nominalzinssatz})^{\text{Periodennummer} - 1}$$

Sehr oft wird bei Kreditverträgen ein Jahreszinssatz als Nominalzinssatz formuliert, die Zahlung erfolgt aber unterjährig. In solchen Fällen müssen Sie den unterjährlichen Periodenzinssatz ermitteln, indem Sie den Jahreszinssatz durch die Anzahl der Zahlungen im Jahr dividieren. Die Gesamtperiodenzahl ergibt sich dann aus den Perioden pro Jahr multipliziert mit der Anzahl der Jahre.

Im Beispiel »Tilgungsrechnung« zur Funktion BW() standen folgende Daten fest: Ein Schuldner nimmt einen Kredit in Höhe von 176.121,76 € auf, den er bei einem Zinssatz von 5,5 % innerhalb von 30 Jahren jeden Monat mit 1.000,00 € bedient. Der Tilgungsplan weist im 18. Monat einen Zinsanteil von 791,64 € aus. Damit beträgt der Tilgungsanteil in diesem Monat 208,36 €. Sie können dieses Ergebnis bis auf eventuelle Rundungsfehler durch die Funktion KAPZ() überprüfen:

=KAPZ(5,5%/12;18;30*12;-176121,76)

(die konkreten Zahlen entstehen aus Zellbezügen zu den Zellen, in denen sich die Daten befinden).

Syntax

Definition

Argumente

Hintergrund

Hinweis

Praxiseinsatz

Rundungsfehler treten bei der Benutzung der integrierten Funktionen oft auf. Diese können nämlich nicht berücksichtigen, was am Bankschalter tatsächlich passiert: Es gibt nur Geldbeträge mit zwei Stellen nach dem Komma. Schreiben Sie also einen Tilgungsplan Monat für Monat auf, müssen Sie die Funktion RUNDEN() einsetzen, damit Sie das wirkliche Geschehen auch richtig abbilden.

Siehe auch BW(), RMZ(), ZINS(), ZW(), ZZR(), KUMKAPITAL(), ZINSZ()



Das Beispiel befindet sich auf dem Arbeitsblatt *BW* der Arbeitsmappe *Tilgungsrechnung.xlsx* im Ordner `\Ms5-235\Kap14`.

KUMKAPITAL() CUMPRINC()

Syntax KUMKAPITAL(*Zins*; *Zzr*; *Bw*; *Zeitraum_Anfang*; *Zeitraum_Ende*; *F*)

Definition Diese Funktion berechnet den Tilgungsanteil, der zwischen zwei Zeitpunkten gezahlt wird, wenn ein Darlehen als Annuitätendarlehen getilgt wird.

Argumente *Zins* (erforderlich) benennt den Nominalzins des Kredits.

Zzr (erforderlich) gibt die Gesamtzahl der Perioden, in denen der Kredit zurückgezahlt wird, an.

Bw (erforderlich) erfasst die Darlehenssumme.

Zeitraum_Anfang (erforderlich) verlangt die erste in die Berechnung einfließende Periode, das ebenfalls erforderliche Argument *Zeitraum_Ende* die letzte.

Mithilfe des (hier ausnahmsweise) erforderlichen Fälligkeitsarguments *F* kann angegeben werden, ob die Zahlungen zu Beginn der Perioden zinswirksam werden ($F = 1$) oder erst an deren Ende ($F = 0$).

Werden die Werte von Argumenten der Funktion, die sinnvollerweise ganze Zahlen verlangen, als gebrochene Zahlen ausgewertet, werden deren Nachkommastellen abgeschnitten.

Die Argumente *Zins*, *Zzr* und *Bw* werden als positive Zahlen erwartet, ansonsten gibt KUMKAPITAL() den Fehlerwert #ZAHL! zurück.

Auch die anderen Argumente müssen »vernünftig« ausfallen: *Zeitraum_Anfang* größer oder gleich 1 sowie *Zeitraum_Ende* größer oder gleich 1 und dabei nicht kleiner als *Zeitraum_Anfang*.

F muss sich zu 0 oder 1 auswerten lassen.

Hintergrund Die Tilgung von Krediten kennt verschiedene Formen. Eine davon ist, dass der Schuldner jede Periode den gleichen Betrag zahlt. Dieser setzt sich aus einem Tilgungsanteil (der sich mit fortschreitender Zeit zunehmend vergrößert) und einem Zinsanteil (der sich immer mehr verkleinert) zusammen. Die Veränderung der Zusammensetzung hängt damit zusammen, dass die abzutragende Schuld mit der Zeit immer geringer wird. Diese Form wird als Annuitätentilgung bezeichnet, der Gesamtbetrag heißt – auch wenn seine Zahlung nicht unbedingt jährlich erfolgt – Annuität.

Die Funktion berechnet den (auch kumulierten) Anteil, der mit Zahlung der Annuitäten zur Kredittilgung verwendet wurde, also keine Zinszahlungen darstellt. Insbesondere kann durch Summierung über alle vergangenen Perioden die verbliebene Restschuld eines ursprünglichen Darlehens bestimmt werden.

Sehr oft wird bei Kreditverträgen ein Jahreszinssatz als Nominalzinssatz formuliert, die Zahlung erfolgt aber unterjährlich. In solchen Fällen müssen Sie den unterjährlichen Periodenzinssatz ermitteln, indem Sie den Jahreszinssatz durch die Anzahl der Zahlungen im Jahr dividieren. Die Gesamtperiodenzahl ergibt sich dann aus den Perioden pro Jahr multipliziert mit der Anzahl der Jahre.

Im Beispiel »Tilgungsrechnung« zur Funktion BW() standen folgende Daten fest: Ein Schuldner nimmt einen Kredit in Höhe von 176.121,76 € auf, den er bei einem Zinssatz von 5,5 % innerhalb von 30 Jahren jeden Monat mit 1.000,00 € bedient. Der Tilgungsplan weist im 19. Monat eine Restschuld von 172.513,25 € aus, damit wurden also 3.608,51 € getilgt. Sie können dieses Ergebnis bis auf eventuelle Rundungsfehler durch die Funktion KUMKAPITAL() überprüfen:

```
--KUMKAPITAL(5,5%/12;30*12;176121,76;1;18;0)
```

(die konkreten Zahlen entstehen aus Zellbezügen zu den Zellen, in denen sich die Daten befinden). Das Vorzeichen des Ergebnisses ist negativ, da der Zahlungsstrom in die andere Richtung als der Barwert (Kreditsumme) geht.

Rundungsfehler treten bei der Benutzung der integrierten Funktionen oft auf. Diese können nämlich nicht berücksichtigen, was am Bankschalter tatsächlich passiert: Es gibt nur Geldbeträge mit zwei Stellen nach dem Komma. Schreiben Sie also einen Tilgungsplan Monat für Monat auf, müssen Sie die Funktion RUNDEN() einsetzen, damit Sie das wirkliche Geschehen auch richtig abbilden.

Hypothekendarlehen werden oft mit einer Zinsbindung vereinbart, die Zeit bis zum Wechsel der Zinsbedingungen reicht aber zur vollständigen Tilgung nicht aus. In einem solchen Fall kann der Kreditnehmer anhand der Restschuld bestimmen, wie hoch sein Risiko im Falle dann gestiegener Zinsen ist, und so ggf. auch verschiedene Darlehensangebote miteinander vergleichen. Will er keinen Tilgungsplan im Detail aufschreiben, kann er die Restschuld ermitteln, indem er von der Kreditsumme den innerhalb der Zinsbindungsfrist getilgten Betrag abzieht. Diesen Betrag rechnet aber KUMKAPITAL() aus. Die konkrete Rechnung erfolgt wie im vorigen Beispiel.

BW(), RMZ(), ZINS(), ZW(), ZZR(), ZINSZ(), KUMZINSZ()

Das erste Beispiel befindet sich auf dem Arbeitsblatt *BW* der Arbeitsmappe *Tilgungsrechnung.xlsx* im Ordner *\Ms5-235\Kap14*.

KUMZINSZ() CUMIPMT()

KUMZINSZ(*Zins*;Zzr;Bw;Zeitraum_Anfang;Zeitraum_Ende;F)

Diese Funktion berechnet die kumulierten Zinsen, die zwischen zwei Zeitpunkten gezahlt werden, wenn ein Darlehen als Annuitätendarlehen getilgt wird.

Zins (erforderlich) benennt den Nominalzins des Kredits.

Zzr (erforderlich) gibt die Gesamtzahl der Perioden, in denen der Kredit zurückgezahlt wird, an.

Bw (erforderlich) gibt die Darlehenssumme an.

Zeitraum_Anfang (erforderlich) verlangt die erste in die Berechnung einfließende Periode, das ebenfalls erforderliche Argument *Zeitraum_Ende* die letzte.

Mithilfe des (hier ausnahmsweise) erforderlichen Fälligkeitsarguments *F* kann angegeben werden, ob die Zahlungen zu Beginn der Perioden zinswirksam werden ($F = 1$) oder erst an deren Ende ($F = 0$).

Hinweis

Praxiseinsatz Zinsbindung über die gesamte Laufzeit

Zinsbindung nur am Anfang

Siehe auch



Syntax

Definition

Argumente

Werden die Werte von Argumenten der Funktion, die sinnvollerweise ganze Zahlen verlangen, als gebrochene Zahlen ausgewertet, werden deren Nachkommastellen abgeschnitten.

Die Argumente *Zins*, *Zzr* und *Bw* werden als positive Zahlen erwartet, ansonsten gibt KUMKAPITAL() den Fehlerwert #ZAHL! zurück.

Auch die anderen Argumente müssen »vernünftig« ausfallen: *Zeitraum_Anfang* größer oder gleich 1 sowie *Zeitraum_Ende* größer oder gleich 1 und dabei nicht kleiner als *Zeitraum_Anfang*.

F muss sich zu 0 oder 1 auswerten lassen.

Hintergrund Die Tilgung von Krediten kennt verschiedene Formen. Eine davon ist, dass der Schuldner jede Periode den gleichen Betrag zahlt. Dieser setzt sich aus einem Tilgungsanteil (der sich mit fortschreitender Zeit zunehmend vergrößert) und einem Zinsanteil (der sich immer mehr verkleinert) zusammen. Die Veränderung der Zusammensetzung hängt damit zusammen, dass die abzutragende Schuld mit der Zeit immer geringer wird. Diese Form wird als Annuitätentilgung bezeichnet, der Gesamtbetrag heißt – auch wenn seine Zahlung nicht unbedingt jährlich erfolgt – Annuität.

Während das Zusammenzählen von Tilgungsanteilen korrekt ist (es wird so die Restschuld ermittelt), ist das Zusammenzählen der Zinsen ohne finanzmathematische Bedeutung. Dies wird zwar gern gemacht (auch von Kreditinstituten), um Kredite miteinander zu vergleichen, es ist aber kein finanzmathematischer Vergleich. Dieser kann nur, wie es die Preisangabenverordnung aus dem Jahr 2002 vorschreibt, durch einen Barwertvergleich der Gläubiger- mit den Schuldnerleistungen erfolgen. Wenn also Zinsen sinnvoll addiert werden, dann nur nach deren Bewertung zum Zeitpunkt des Vertragsabschlusses. Wer einen Kredit über 100.000 Geldeinheiten aufnimmt, zahlt auch nur diese zurück. Wenn er das später tut, fallen Zinsen an.

Hinweis Sehr oft wird bei Kreditverträgen ein Jahreszinssatz als Nominalzinssatz (Sollzinssatz) formuliert, die Zahlung erfolgt aber unterjährig. In solchen Fällen müssen Sie den unterjährlichen Periodenzinssatz ermitteln, indem Sie den Jahreszinssatz durch die Anzahl der Zahlungen im Jahr dividieren. Die Gesamtperiodenzahl ergibt sich dann aus den Perioden pro Jahr multipliziert mit der Anzahl der Jahre.

Praxiseinsatz Im Beispiel »Tilgungsrechnung« zur Funktion BW() standen folgende Daten fest: Ein Schuldner nimmt einen Kredit in Höhe von 176.121,76 € auf, den er bei einem Zinssatz von 5,5 % innerhalb von 30 Jahren jeden Monat mit 1.000,00 € bedient.

Der Tilgungsplan weist im 19. Monat eine Restschuld von 172.513,25 € aus. Der Kredit wurde bisher mit 18.000,00 € bedient und Sie sind (trotz der oben geäußerten Bedenken) an der Summe interessiert, die nur Zinsen darstellt. Sie können an der Restschuld erkennen, dass 3.608,51 € getilgt wurden, also muss die Differenz in Höhe von 14.391,49 € der Zinsanteil sein. Sie können dieses Ergebnis bis auf eventuelle Rundungsfehler durch die Funktion KUMZINSZ() überprüfen:

=-KUMZINSZ(5,5%/12;30*12;176121,76;1;18;0)

(die konkreten Zahlen entstehen aus Zellbezügen zu den Zellen, in denen sich die Daten befinden). Das Vorzeichen des Ergebnisses ist negativ, da der Zahlungsstrom in die andere Richtung als der Barwert (Kreditsumme) geht.

Rundungsfehler treten bei der Benutzung der integrierten Funktionen oft auf. Diese können nämlich nicht berücksichtigen, was am Bankschalter tatsächlich passiert: Es gibt nur Geldbeträge mit zwei Stellen nach dem Komma. Schreiben Sie also einen Tilgungsplan Monat für

Monat auf, müssen Sie die Funktion RUNDEN() einsetzen, damit Sie das wirkliche Geschehen auch richtig abbilden.

BW(), RMZ(), ZINS(), ZW(), ZZR(), ZINSZ(), KUMKAPITAL()

Sie finden dieses Beispiel auf dem Arbeitsblatt *BW* der Arbeitsmappe *Tilgungsrechnung.xlsx* im Ordner *Ms5-235\Kap14*.

Siehe auch



KURS() PRICE()

KURS(*Abrechnung;Fälligkeit;Zins;Rendite;Rückzahlung;Häufigkeit;Basis*)

Berechnet den Kurs eines festverzinslichen Wertpapiers (Anleihe), also den Kaufpreis ohne Stückzinsen.

Abrechnung (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum, an dem die Rückzahlung des durch das Wertpapier verbrieften Darlehens erfolgt.

Zins (erforderlich) informiert über den vereinbarten Jahreszinssatz als Preis für das geliehene Geld (Kuponzinssatz).

Rendite (erforderlich) ist der am Abrechnungstag geltende Marktzinssatz, zu dem bei der Berechnung des Kurses alle zukünftigen Zahlungen abgezinst werden.

Rückzahlung (erforderlich) ist der Prozentfuß bezogen auf den Nennwert des Wertpapiers (also so, als ob dieser 100 Geldeinheiten wäre), zu welchem am Fälligkeitstag zurückgezahlt wird.

Da es durchaus üblich ist, festverzinsliche Wertpapiere mit mehreren Zinsterminen im Jahr auszustatten, informiert das erforderliche Argument **Häufigkeit** über deren Anzahl. Zulässige Werte sind: 1 = jährlich; 2 = halbjährlich; 4 = vierteljährlich.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.6. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

Syntax

Definition

Argumente

Basis	Methode	Bedeutung
0	30/360 (NASD-Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Ist das Ausgangsdatum der 31. eines Monats, wird dieses Datum zum 30. desselben Monats. Ist das Enddatum der 31. eines Monats und das Ausgangsdatum ein Datum vor dem 30. eines Monats, wird das Enddatum zum 1. des darauf folgenden Monats. In allen anderen Fällen wird das Enddatum zum 30. desselben Monats.
1	Genau/Genau	Jeder Monat hat so viele Tage wie im Kalender stehen, die Summe über alle Monate ergibt die Anzahl der Tage im Jahr
2	Genau/360 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 360 Tage.
3	Genau/365 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 365 Tage.
4	30/360 (Europäische Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Jedes Ausgangs- und Enddatum, das auf den 31. eines Monats fällt, wird zum 30. desselben Monats.

Tabelle 14.6

Verschiedene Methoden zur Ermittlung der Anzahl der Tage eines Jahres

An die Argumente von KURS() werden folgende Anforderungen gestellt:

- ▶ Datumsangaben verlangen diese ohne Anhabe der Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, gibt KURS() den Fehlerwert #ZAHL! zurück
- ▶ Für *Rendite* und *Zins* werden nicht negative Zahlen, für *Rückzahlung* sogar positive Zahlen verlangt. Andernfalls gibt KURS() den Fehlerwert #ZAHL! zurück.
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt KURS() den Fehlerwert #ZAHL! zurück. Das Gleiche geschieht, wenn der Abrechnungstermin nach der Fälligkeit liegt.

Hintergrund Um das finanzmathematische Äquivalenzprinzip

Leistung des Gläubigers = Leistung des Schuldners

bezogen auf den »Startpunkt« des Geschäfts umzusetzen, ist der Kurs eines festverzinslichen Wertpapiers (Anleihe) plus eventuell zu zahlende Stückzinsen gleich dem Barwert der zukünftigen in diesem Wertpapier verbrieften Leistungen des Emittenten (Schuldners). Der Kurs wird als Prozentfuß bezogen auf den Nennwert des Wertpapiers angegeben, also so, als ob der Nennwert 100 Geldeinheiten beträgt.

Die Barwertberechnung ist dann kein Problem, wenn das Kaufdatum (Besitzwechsel) eines mit Jahreskupon versehenen Wertpapiers mit dem Tag der Zinszahlung einhergeht. In diesem Fall gilt es nur ganze Jahre zu berücksichtigen. Nicht so einfach ist die Situation, wenn der Besitzwechsel zwischen Zinsterminen erfolgt bzw. Zinszahlungen mehrfach im Jahr vereinbart sind. Die Finanzmathematik kennt verschiedene Arten des Umgangs mit gebrochenen Jahresanteilen. Die bekanntesten sind die Methoden Moosmüller und Braess/Fangmeyer sowie die ISMA-Methode. ISMA steht für International Securities Market Association, eine Organisation, die aus der Association of International Bond Dealers (AIBD) hervorgegangen ist.

Die ISMA-Methode liefert für eine einmalige Zinszahlung im Jahr das Gleiche wie KURS() und lässt sich für Halb- und Vierteljahreskupsens leicht mit KURS() nachbilden.

Das genannte Äquivalenzprinzip durch Barwertbildung (Abzinsen zukünftiger Leistungen) lautet bei der Kursberechnung durch Excel in der von Excel und seinen Funktionen entlehnten Terminologie

$$\begin{aligned}
 KURS = & \frac{\text{Rückzahlung}}{\left(1 + \frac{\text{Rendite}}{\text{Häufigkeit}}\right)^{\text{ZINSTERMZAHL} - 1 + \frac{\text{ZINSTERMTAGNZ}}{\text{Tage der Zinsperiode}}} + \\
 & + \sum_{K=1}^{\text{ZINSTERMZAHL}} \frac{\frac{\text{ZINS}}{\text{Häufigkeit}} \cdot 100}{\left(1 + \frac{\text{Rendite}}{\text{Häufigkeit}}\right)^{K-1 + \frac{\text{ZINSTERMTAGNZ}}{\text{Tage der Zinsperiode}}} - \\
 & - \frac{\text{ZINS}}{\text{Häufigkeit}} \cdot 100 \cdot \frac{\text{ZINSTERMTAGVA}}{\text{Tage der Zinsperiode}}
 \end{aligned}$$

wobei mit *ZINSTERMZAHL* die Anzahl der Zinszahlungen nach dem Kauf, mit *ZINSTERMTAGNZ* die Anzahl der Tage bis zum nächsten Zinstermin und mit *ZINSTERMTAGVA* die Anzahl der Tage seit dem letzten Zinstermin bezeichnet wurden.

Ist *Häufigkeit* = 1, stellt diese Formel gerade auch die ISMA-Methode dar. Ist die *Häufigkeit* gleich 2 oder 4, rechnet Excel mit einer gleichmäßigen Verteilung der *Rendite* auf die Perioden innerhalb des Jahrs. Die ISMA-Methode dagegen verwendet eine periodengerechte Umlegung nach dem Zusammenhang zwischen Nominal- und Effektivzins, den Sie in den Erläuterungen zur Funktion *EFFEKTIV()* bzw. *NOMINAL()* finden:

$$\text{Rendite der Periode} = \frac{\text{NOMINAL}(\text{Rendite im Jahr}; \text{Häufigkeit})}{\text{Häufigkeit}}$$

Deshalb können Sie im Falle von mehreren Zinszahlungen pro Jahr den Kurs nach der ISMA-Methode mithilfe von *KURS()* berechnen, indem Sie vorher eine Umrechnung der »effektiven« *Rendite* in eine »nominale« vornehmen (durch Anwenden der Funktion *NOMINAL()*).

Die oben gegebenen Erläuterungen gelten nur, wenn mehr als ein einziger Zinstermin bis zur Fälligkeit verbleibt. Andernfalls rechnet Excel nicht nach der obigen Formel für den Kurs, sondern benutzt statt der gebrochenen Laufzeit eine gebrochene Rendite. *KURS()* liefert dann das gleiche Ergebnis wie *KURSFÄLLIG()*.

Achtung

Die 3,500 %-Bundesanleihe von 05 rentierte am 30.1.2013 mit 0,39 %. Ihre Fälligkeit ist am 4.1.2016. Wie viel muss jemand bezahlen, der Papiere im Nennwert von 1.000,00 € erwerben will?

**Praxiseinsatz
Jahreskupon**

Diese Aufgabenstellung ist nicht ganz typisch, da sich Kurse durch Angebot und Nachfrage bilden, die Rendite wird dann aus dem Kurs ausgerechnet. Es soll aber das Prinzip an einem Wertpapier demonstriert werden, dessen Verhalten täglich unter <http://www.deutsche-finanzagentur.de> nachzulesen ist.

Hinweis

Kursberechnungen dieser Art sind denkbar, wenn ein Emittent ein Wertpapier begibt, dieses mit einem »runden« Nominalzins versieht ($\frac{1}{8}$ -Abstände) und es mit gewissen Renditevorstellungen mit Zu- oder Abschlägen beim Kurs auf den Markt bringen will.

Wollen Sie den ISMA-Kurs nach der Tagezahl-Methode taggenau/taggenau bestimmen, ohne die Funktion *KURS()* zu verwenden (Sie wollen verstehen, was *KURS()* genau umsetzt), können Sie ein Arbeitsblatt wie in Abbildung 14.2 vorbereiten.

In die ersten sieben Zellen von *C5* bis *C11* geben Sie die bekannten Daten ein. Vorbereitend notieren Sie die Tagezahlweise mit 1 (das entspricht *Basis* = 1 in allen verwendeten integrierten Funktionen).

Als Nächstes ermitteln Sie mit *=ZINSTERMNZ(C5;C7;1;C11)* den nächsten Zinstermin: 4.1.2014. Die Formel *=ZINSTERMTAGNZ(C5;C7;1;C11)* hilft Ihnen beim Zählen der Tage bis zu diesem Zinstermin: 339. Zu Berechnung der Stückzinsen brauchen Sie die Tage seit dem letzten Zinstermin (*=ZINSTERMTAGVA(C5;C7;1;C11)* zählt hier 26), sodass Sie über *=C8*C15/C16*C9* einen Betrag von 2,49 € fast im Kopf bestimmen können. Die Zahl der verbleibenden Zinstermine, die Sie zum Abzinsen der zukünftige Leistungen brauchen, wird durch *=ZINSTERMZAHL(C5;C7;1;C11)* zu 3 ermittelt.

	A	B	C	D
1				
2		Bundesanleihe von 05		
3		Jahreskupon (Excel=ISMA)		
4				
5		Kaufdatum	30.01.2013	
6		Rückzahlung	100,00%	
7		Fälligkeit	04.01.2016	
8		Nominalzins	3,500%	
9		Nennwert	1.000,00 €	
10		Rendite	0,39%	
11		Tagezählweise	1	
12				
13		vorhergehender Zinstermin	04.01.2013	
14		nächster Zinszahlungstermin	04.01.2014	
15		Jahresrest in Tagen	339	
16		Tage für Stückzinsen	26	
17		Tage im Jahr	365	
18		Stückzinsen	2,49 €	
19		Anzahl der Zinstermine	3	
20				
21		BW der Zinsen am nächsten Zinstermin	104,59 €	
22		BW der Rückzahlung am nächsten Zinstermin	992,25 €	
23				
24		Barwert aller Leistungen	1.092,88 €	
25				
26		Kurs	109,03868	[%]
27				
28		KURS	109,03868	[%]

Abbildung 14.2: Vorbereitung von Kurs- und Renditeberechnungen

Nun geht es ans Abzinsen. Hier hilft Ihnen die Funktion $BW()$, und zwar mit $=BW(C10;C18-1; -C8*C9; -C8*C9;1)$ als Aufgabe der Rentenrechnung zur Berechnung des Barwerts zukünftiger Zinszahlungen (Ergebnis 104,59 €) sowie als Aufgabe der Zinseszinsrechnung zum Abzinsen des Rückzahlbetrags über $=BW(C10;C18-1;0; -C6*C9)$ zu 992,25 €.

Die Summe der beiden Barwerte ist nun, entsprechend der Vorgabe von ISMA, über die gebrochene Laufzeit von 334 aus 365 Tagen mit einem gebrochenen Exponenten in der Zinsformel zu ermitteln: $=(C20+C21)/(1+C10)^(C14/C16)$ liefert 1.092,88 €.

Der Kurs ist der Prozentfuß (also eine Zahl bezogen auf einen Nennwert von 100), der vom gerade berechneten Barwert noch den Wert der Stückzinsen abziehen muss: $=(C23-C17)/C9*100$.

Im Ergebnis kommt nun das heraus, was auch $=KURS(C5;C7;C8;C10;C6*100;1;C11)$ ermittelt: 109,039.

Beachten Sie, dass es wegen der geringen Rendite zu Abweichungen des berechneten Kurses zu den Angaben am jeweiligen Tag kommen kann (Rundungsfehler).

Halbjahreskupon

In Deutschland gibt es nur wenige Anleihen, die mehr als einen Zinstermin pro Jahr haben. In der Beispieldatei zum Buch wurde deshalb die Anleihe des vorigen Beispiels fiktiv mit einem Halbjahreskupon ausgestattet und die Berechnungen durchgeführt. Der Excel-Kurs unterscheidet sich von dem nach ISMA in Nachkommastellen. Wollen Sie den ISMA-Kurs bestimmen, ist der Marktzins *Rendite* mittels $NOMINAL(Rendite;Häufigkeit)$ periodengerecht aufzuteilen und in die Funktion $KURS()$ einzusetzen.

RENDITE(), UNREGER.KURS(), UNREGER.REND(), UNREGLE.KURS(),
UNREGLE.REND()

Die beiden Beispiele finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Kursrechnung.xlsx* auf dem Arbeitsblatt *KURS*.

Siehe auch



KURSDISAGIO() PRICEDISC()

KURSDISAGIO(*Abrechnung*; *Fälligkeit*; *Disagio*; *Rückzahlung*; *Basis*)

Diese Funktion berechnet den Auszahlungsbetrag eines mit vorschüssiger Verzinsung einfach abgezinsten Wertpapiers.

Abrechnung (erforderlich) ist der Tag, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) erfasst den Tag, an dem die Leistungen aus dem Wertpapier erfolgen müssen.

Disagio (erforderlich) gibt an, zu welchem Zinssatz die Abzinsung (Abschlag, Diskont) erfolgt.

Rückzahlung (erforderlich) beschreibt die Leistung am Fälligkeitstag.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.6. Wird es weggelassen, rechnet Excel mit $Basis = 0$.

Alle Argumente der Funktion, die ein Datum betreffen, verwenden eine Datumsangabe ohne Uhrzeit, gebrochene Zahlen werden also abgerundet. Das Argument *Basis* verlangt ebenfalls eine ganze Zahl und schneidet Nachkommastellen ab.

Werden ungültige Datumseinträge verwendet bzw. keine Zahlen dort angegeben, wo sie erforderlich sind, liefert die Funktion den Fehlerwert **#WERT!**. Im Falle ungültiger Zahlen für die Nicht-Datums-Argumente lautet die Fehlermeldung **#ZAHL!**.

Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Der Hinweis auf den Nennwert ist falsch.

Das Prinzip der vorschüssigen Verzinsung besteht in der Annahme, dass sich ein Anfangskapital (einzuzahlender Betrag, Darlehen) aus dem Rückzahlungsbetrag minus die Zinsen auf diesen Betrag ergeben. Dieses Prinzip wird vor allem im unterjährlichen Bereich verwendet und unterscheidet sich von dem Prinzip, welches einem Sparbuch oder einem Hypothekendarlehen zugrunde liegt. Dort werden nämlich die Zinsen am Ende einer Periode auf Basis des Anfangskapitals berechnet (nachsüssige Verzinsung).

Syntax

Definition

Argumente

Wichtig

Hintergrund

Hinweis Zwischen den Funktionen zu diesem Themenkreis besteht die Verbindung

$$AUSZAHLUNG() - AUSZAHLUNG() \cdot DISAGIO() \cdot \frac{TAGE360()}{360} = KURSDISAGIO()$$

(hier notiert für den Fall von *Basis* = 4). Das ist aber gerade das Prinzip vorschüssiger Verzinsung.

Das von der Funktion verwendete Argument *Rückzahlung* ist hier der Funktion AUSZAHLUNG() »zuzuordnen«.

Praxiseinsatz Wechselrechnung Am 10.5.2013 legt ein Geschäftsmann seiner Bank einen Wechsel in Höhe von 5.000 € mit einer (Rest-) Laufzeit von 2 Monaten vor. Diese setzt einen Diskontsatz von 5 % p.a. an und fordert keine Gebühren. Wie hoch ist die Summe, die dem Konto gutgeschrieben wird?

Sie lesen die Gleichung aus den Hintergrundinformationen als

$$\text{Wechselsumme} - \text{Wechselsumme} \cdot \text{Diskontsatz} \cdot \frac{TAGE360()}{360} = \text{Gutschrift}$$

und verwenden

=KURSDISAGIO("10.5.2013";"10.7.2013";5%;5000;4)

zur Erlangung des Ergebnisses von 4.958,33 €.

Natürlich stehen die konkreten Zahlen in Zellen und die Formel verwendet Zellbezüge.

Siehe auch DISAGIO(), AUSZAHLUNG(), RENDITEDIS(), ZINSSATZ()



Das Beispiel finden Sie auf den Arbeitsblättern *Wechselrechnung* in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* im Ordner *\Ms5-235\Kap14*.

KURSFÄLLIG() PRICEMAT()

Syntax KURSFÄLLIG(*Abrechnung*; *Fälligkeit*; *Emission*; *Zins*; *Rendite*; *Basis*)

Definition Gibt den Kurs (als Prozentfuß, d.h. bezogen auf einen Nennwert) eines Wertpapiers zurück, welches mit einfacher nachschüssiger Verzinsung ausgestattet ist (keine Zinseszinsseffekte).

Argumente *Abrechnung* (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) erwartet den Fälligkeitstermin des Wertpapiers.

Emission (erforderlich) ist das Datum der Ausgabe des Wertpapiers.

Zins (erforderlich) ist der vereinbarte Nominalzinssatz des Wertpapiers, bezogen auf ein ganzes Jahr.

Rendite (erforderlich) erwartet den gegenwärtig vorhandenen Marktzinssatz bzw. die erwartete Rendite des Wertpapiers (als Zinssatz bezogen auf ein ganzes Jahr).

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.6. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

An die Argumente der Funktion werden folgende Forderungen gestellt:

- ▶ *Abrechnung*, *Fälligkeit*, und *Emission* verlangen eine Datumsangabe ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Ebenso wird aus *Basis* durch Abschneiden der Nachkommastellen eine ganze Zahl.
- ▶ Sind *Abrechnung*, *Fälligkeit* oder *Emission* kein zulässiges Datum, gibt KURSFÄLLIG() den Fehlerwert #WERT! zurück. Der Abrechnungstermin muss vor dem Fälligkeitstermin liegen.
- ▶ Sind *Zins* oder *Rendite* kleiner 0, liefert KURSFÄLLIG() den Fehlerwert #ZAHL!
- ▶ *Basis* muss sich zu einer Zahl zwischen 0 und 4 auswerten lassen, sonst gibt KURSFÄLLIG() den Fehlerwert #ZAHL! zurück

Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Um das finanzmathematische Äquivalenzprinzip

Leistung des Gläubigers = Leistung des Schuldners

bezogen auf den »Startpunkt« des Geschäfts umzusetzen, ist der Kurs eines festverzinslichen Wertpapiers plus eventuell zu zahlende Stückzinsen gleich dem Barwert der zukünftigen in diesem Wertpapier verbrieften Leistungen des Emittenten (Schuldners). Der Kurs wird als Prozentfuß bezogen auf den Nennwert des Wertpapiers angegeben, also so, als ob der Nennwert 100 Geldeinheiten beträgt. Im vorliegenden Fall besteht die zukünftige Leistung in der Rückzahlung plus Nominalzinsen. Zinseszinsseffekte treten nicht auf, da nur Bewertungen im unterjährlichen Bereich vorgenommen werden. Beim Abzinsen wird nicht mit gebrochener Laufzeit im Exponenten (ISMA-Methode), sondern mit einem gebrochenen, der Laufzeit entsprechenden Zinssatz gerechnet.

Eine gesamtfällige Schuld über 1.000,00 € wird am 1.6.2013 mit einem Nominalzinssatz von 4 % p.a. und Rückzahlung zum 1.12.2013 vereinbart. Am 9.8.2013 erfolgt ein Gläubigerwechsel. Zu diesem Termin sind Realzinsen von 2,5 % p.a. für die ausstehende Zeit bis zur Rückzahlung üblich. Welchen Preis zahlt der Übernehmende?

Den Kurs von 100,46, der wegen des Nennwerts von 1.000,00 € mit 10 zu multiplizieren ist, erhalten Sie über

=KURSFÄLLIG(C3;C4;C2;4%;4,5%;4)

wenn C3 das Kaufdatum, C4 das Fälligkeitsdatum und C2 den Emissionstag bereithalten.

Hinzukommen Stückzinsen in Höhe von 7,56 €, die Sie mithilfe von AUFGELZINS() oder »per Hand« berechnen können. Weitere Erläuterungen finden Sie im Beispiel zur Funktion AUFGELZINS().

AUFGELZINS(), RENDITEFÄLL(), ZINSTERMTAG()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* auf dem Arbeitsblatt *AUFGELZINS*.

Wichtig

Hintergrund

Praxiseinsatz

Siehe auch



LIA()



SLN()

Syntax LIA(*Ansch_Wert*; *Restwert*; *Nutzungsdauer*)

Definition Die Funktion berechnet die Abschreibungsbeträge von Wirtschaftsgütern nach der Methode der linearen Abschreibung.

Argumente *Ansch_Wert* (erforderlich) verlangt die Anschaffungskosten (Nettokaufpreis plus Anschaffungsnebenkosten minus Anschaffungskostenminderungen) eines Wirtschaftsguts. Verwenden Sie keine Zahl, erscheint der Fehlerwert #WERT!, ist die eingegebene Zahl negativ, lassen sich die berechneten Werte nicht interpretieren.

Restwert (erforderlich) entspricht dem Wert des Wirtschaftsguts am Ende der Abschreibungsdauer. Verwenden Sie keine Zahl, erscheint der Fehlerwert #WERT!. Ist die eingegebene Zahl negativ, erfolgt die Fehlermeldung #ZAHL!.

Nutzungsdauer (erforderlich) verlangt die Anzahl der Perioden, in denen das Wirtschaftsgut abgeschrieben wird. Sie sollten für dieses Argument eine ganze Zahl größer 0 (Null) verwenden.

Hintergrund Abschreibung hat den Sinn, den Werteverlust eines Wirtschaftsguts zu ermitteln und sichtbar zu machen. Davon zu unterscheiden ist der Begriff der Absetzung für Abnutzung, der die Kosten für den Erwerb eines Wirtschaftsguts als Betriebsausgaben steuerlich erfasst.

Der Abschreibungsbetrag, der in jeder Periode konstant ist, bestimmt sich aus der Differenz zwischen Anschaffungskosten und Restwert, aufgeteilt auf die Anzahl der Perioden. Dieser Betrag mindert den Buchwert in jeder Periode.

Die lineare Abschreibungsmethode ist im deutschen Steuerrecht zulässig. Da aber ein gebrochener Jahresanteil der ersten Periode gesondert zu berücksichtigen ist, ist die Funktion LIA() im Allgemeinen nicht einsetzbar. Sie müssen den Betrag der ersten Abschreibungsperiode »per Hand« anpassen und schreiben den Rest mithilfe von LIA() auf.

Praxiseinsatz Ein Wirtschaftsgut mit Anschaffungskosten von 1.000,00 € soll in 5 Jahren linear auf einen Restwert von 100,00 € abgeschrieben werden. Sie können hierzu in jeder Periode den Abschreibungsbetrag mittels LIA() berechnen und vom Buchwert der Vorperiode abziehen.

Alternativ können Sie aber auch einen Abschreibungsplan erstellen, der die oben genannten Formeln zur Berechnung des ersten Abschreibungsbetrags sowie die Vorgehensweise zur Berechnung der anderen Beträge umsetzt. Die Erstellung eines Abschreibungsplans nach eigenem Muster ist vor allem dann notwendig, wenn wegen der Zwölfstel-Regelung seit 2004 der unmittelbare Einsatz der integrierten Funktionen versagt.

Einzelheiten zu diesem Beispiel entnehmen Sie bitte der zugehörigen Arbeitsmappe.

Siehe auch DIA(), GDA(), GDA2() VDB(), AMORLINEARK()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Abschreibungsrechnung.xlsx* auf dem Arbeitsblatt *LIA*.

MDURATION()



MDURATION(*Abrechnung*; *Fälligkeit*; *Nominalzins*; *Rendite*; *Häufigkeit*; *Basis*)

Diese Funktion berechnet die als »modified duration« bekannte Kennzahl für festverzinsliche Wertpapiere.

Abrechnung (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum, an dem die Rückzahlung des durch das Wertpapier verbrieften Darlehens erfolgt.

Coupon (erforderlich)⁶ informiert über den vereinbarten Jahreszinssatz als Preis für das geliehene Geld.

Rendite (erforderlich) ist der am Abrechnungstag geltende Marktzinssatz, zu dem bei der Berechnung der Duration alle zukünftigen Zahlungen abgezinst werden.

Da es durchaus üblich ist, festverzinsliche Wertpapiere mit mehreren Zinsterminen im Jahr auszustatten, informiert das erforderliche Argument *Häufigkeit* über deren Anzahl. Zulässige Werte sind: 1 = jährlich; 2 = halbjährlich; 4 = vierteljährlich.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.6. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

Alle Argumente der Funktion, die ein Datum betreffen, verwenden eine Datumsangabe ohne Uhrzeit, gebrochene Zahlen werden also abgerundet. Die Argumente *Häufigkeit* und *Basis* verlangen ebenfalls eine ganze Zahl und schneiden Nachkommastellen ab.

Werden ungültige Datumseinträge verwendet bzw. keine Zahlen dort angegeben, wo sie erforderlich sind, liefert die Funktion den Fehlerwert #WERT!. Im Falle ungültiger Zahlen für die Nicht-Datums-Argumente lautet die Fehlermeldung #ZAHL!.

Volatilität, das heißt Veränderungen von Kursen in Reaktion auf veränderte Marktzinsen, ist eine wichtige Kennzahl zum Risikomanagement von Portfolios. Im Gegensatz zu Aktien und Optionen verringert sich die »Schwankungsbreite« von festverzinslichen Wertpapieren gegen Ende deren Laufzeit, da der Auszahlungsbetrag für den Fälligkeitstermin feststeht.

Mithilfe der mathematischen Differenzialrechnung kann festgestellt werden, dass für festverzinsliche Wertpapiere gilt:

$$\text{relative Kursänderung} = - \frac{\text{Duration}}{(1 + \text{Marktzins})} \cdot \text{Marktzinsänderung}$$

wobei Duration die Macauley-Duration des Wertpapiers ist (siehe Funktion DURATION()).

MDURATION() liefert nun gerade den Faktor, mit dem die Marktzinsänderung zu multiplizieren ist (ohne Vorzeichen), um die relative Kursänderung zu ermitteln.

Erfolgen mehrere Zinszahlungen im Jahr, wird der in der Regel als Jahreszinssatz bekannte Marktzins gleichmäßig auf die Zinsperioden aufgeteilt.

Syntax

Definition

Argumente

Hintergrund

⁶ Vor Excel 2013 heißt dieses Argument *Nominalzins*.

Praxiseinsatz Die 3,500 %-Bundesanleihe von 05 rentierte am 30.1.2013 mit 0,39 %. Ihre Fälligkeit ist am 4.1.2016. Ihr Kurs betrug an diesem Tag rechnerisch 109,039 (Beispiel *Jahreskupon* zur Funktion KURS()).

Führen Sie die Kursberechnungen aus dem genannten Beispiel noch einmal durch, aber nun mit einer um 0,5 Prozentpunkte auf 0,89 % nach oben korrigierten Rendite (das ist die von Ihnen erwartete Marktziinsänderung für die nahe Zukunft), ändert sich der Kurs auf 107,511. Dies entspricht einem Kursverlust von 1,41 %.

Eine Näherung dieses Ergebnisses erhalten Sie, wenn Sie die modifizierte Duration über MDURATION() ausrechnen. Diese beträgt mit den Daten der Anleihe 2,821. Multiplizieren Sie dies mit den 0,5 Prozentpunkten der Marktziinsänderung, erhalten Sie eine relative Kursänderung von 1,41 %. Wegen des Vorzeichens führt das zu einem Kursverlust von 1,538, was einem (genäherten) Kurs der Zukunft von 107,501 entspricht.

Die Kennzahl »modified duration« kann also begleitend zum Kurs einer Anleihe geliefert werden, damit ein Anleger in der Lage ist, ohne ausführliche Um- und Neuberechnungen die Struktur eines Depots unter Risikogesichtspunkten zu bewerten.

Siehe auch DURATION(), KURS(), BW()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Kursrechnung.xlsx* auf dem Arbeitsblatt MDURATION.

NBW() NPV()

Syntax NBW(*Zins*; *Wert1*; *Wert2*; ...)

Definition Diese Funktion berechnet den Nettobarwert der mit einer Investition verbundenen zukünftigen Periodenüberschüsse (cash flow) auf Basis eines vorgegebenen Kalkulationszinssatzes.

Argumente *Zins* (erforderlich) ist der vom Investierenden vorgegebene Kalkulationszinssatz.

Wert1; *Wert2*; ... (erforderlich) sind die lückenlos in einer Spalte angeordneten (tatsächlichen und erwarteten) Überschüsse aus Aus- und Einzahlungen. Jeder Wert vertritt das Ende einer Periode (in der Regel ein Jahr) in aufsteigender Ordnung und ohne Lücken. Negative Überschüsse werden mit einem Minuszeichen versehen.

Enthalten Zellen des Arguments *WertN* keine Zahlen oder sind leer, rechnet Excel so, als ob diese Zellen nicht vorhanden sind. Dies ist auch so, wenn eventuelle Zellbezüge im Argument auf solche Zellen verweisen.

Hintergrund Dynamische Methoden der Investitionsrechnung gehen im Unterschied zu statischen Methoden, die Kosten und Erlöse einer Investition berücksichtigen, von (geschätzten und prognostizierten) Ein- und Auszahlungen und deren Verzinsung aus. Beide Zahlungsrichtungen werden dabei mit einem einheitlichen Kalkulationszinssatz bewertet, der aus den Erfahrungen des Investierenden resultiert. Als Kapitalwert wird die Summe aller abgezinsten Periodenüberschüsse bezeichnet.

$$\text{Kapitalwert} = \sum_{n=0}^{\text{Dauer}} \frac{\text{Ein} - \text{minus Auszahlungen im Jahre } n}{(1 + \text{Zins})^n}$$

Eine Investition gilt als wirtschaftlich sinnvoll, wenn der Kapitalwert nicht negativ ist, da dann wenigstens das eingesetzte Kapital plus dessen Verzinsung zurückkommt.

Der erste Wert der *Werte*-Argumente von NBW() vertritt das Ende der ersten Periode. Der Kapitalwert der Investition ermittelt sich deswegen aus NBW() abzüglich der am Anfang der ersten Periode entstehenden Auszahlungen.

Wichtig

Sind die Periodenüberschüsse konstant, kann als Alternative zu NBW() die Funktion BW() zur Berechnung des Nettobarwerts herangezogen werden. Da BW() auch einen Zukunftswert als Argument erlaubt, lässt sich ein auf einem Restwert der Investition beruhender und von den konstanten Überschüssen abweichender Überschuss am Ende der Investitionsdauer ebenfalls berücksichtigen.

Hinweis

Vergleichen Sie zu den folgenden Beispielen auch die Erläuterungen zu IKV() und die dortigen Beispiele.

Praxiseinsatz

Die Anschaffungskosten für eine Maschine belaufen sich auf 80.000,00 €. Die erwarteten jährlichen Überschüsse (Einzahlungen minus Auszahlungen) werden wie in Tabelle 14.7 geschätzt.

Sachinvestition

Jahr	Überschuss (in €)
1	15.000
2	19.000
3	25.000
4	27.000
5	17.000
6	7.000

Tabelle 14.7
Geschätzte jährliche Überschüsse beim Einsatz einer Maschine

Ist diese Investition wirtschaftlich sinnvoll, wenn (aus Erfahrung) ein Kalkulationszinssatz von 10 % (im Jahr) angesetzt wird.

Zur Lösung notieren Sie in einer Zelle die Anschaffungskosten und an anderer Stelle lückenlos in einer Spalte die Werte aus Tabelle 14.7. Die Anwendung von NBW() führt zu einem Betrag von rund 81.070 €, der die Anschaffungskosten leicht übersteigt. Die Verzinsung erfolgt also voraussichtlich zu einem etwas besseren Zinssatz als erwartet.

Beachten Sie, dass es bei Sachinvestitionen, deren zukünftige Überschüsse nur geschätzt sind, nicht unbedingt auf Stellen nach dem Komma ankommt.

Auf die gleiche Weise lassen sich Finanzinvestitionen bewerten: Jemand bekommt auf ein angelegtes Kapital nach einem Jahr 1,75 %, nach zwei Jahren 2,25% und nach 3 Jahren 3,8 % Zinsen. Er vermutet eine Gesamtverzinsung zu 2,5 %.

Finanzinvestition

In der Beispieldatei zum Buch finden Sie Berechnungen, die zeigen, dass der Nettobarwert für einen Anlagebetrag von 100,00 € dann sogar bei 100,24 € liegt und damit die Investition bei dieser erwartenden Verzinsung sinnvoll ist.

BW(), IKV(), QIKV(), XINTZINSFUSS(), XKAPITALWERT()

Siehe auch

Die beiden Beispiele finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Investitionsrechnung.xlsx* auf dem Arbeitsblatt *NBW*.



NOMINAL()



NOMINAL()

Syntax NOMINAL(*Effektiver_Zins;Perioden*)

Definition Diese Funktion errechnet aus einem gegebenen Effektivzinssatz den Nominalzinssatz, der finanzmathematisch zur Äquivalenz führt.

Argumente *Effektiver_Zins* (erforderlich) beschreibt den vorgegebenen effektiven Jahreszinssatz, der sich durch Zinseszinsseffekte bei unterjährlicher Verzinsung ergibt.

Perioden (erforderlich) liefert die Anzahl der Zinsperioden im Jahr.

Das Argument *Perioden* wird im Falle nicht einer gebrochenen Zahl durch Abschneiden der Stellen eingekürzt. Das Ergebnis muss größer als 0 (Null) sein, sonst gibt NOMINAL() den Fehlerwert #ZAHL! zurück. Ist eines der Argumente kein numerischer Ausdruck oder *Effektiver_Zins* nicht positiv, liefert NOMINAL() ebenfalls den Fehlerwert #ZAHL!.

Hintergrund Bei verschiedenen Finanzgeschäften (Hypothekendarlehen, Bausparen, Guthabenverzinsung bei bestimmten Girokonten, Kontokorrentkonten, Dispokredite u.a.m.) wird zwar ein Jahreszinssatz angegeben, dieser dient aber nur zur Vorbereitung weiterer Modalitäten. Diese bestehen darin, dass nicht jährlich verzinst wird, sondern in unterjährlichen Perioden. Der dann angewendete Zinssatz entsteht durch Teilung des Nominalzinssatzes durch die Periodenzahl.

Um Vergleichbarkeit verschiedener Konditionen zu haben, wird der Zinssatz, der bei einmaliger Verzinsung das gleiche Ergebnis wie die unterjährliche Verzinsung liefert, als effektiver Jahreszinssatz bezeichnet. Zwischen beiden Zinssätzen besteht die Beziehung

$$1 + \text{Effektivzinssatz} = \left(1 + \frac{\text{Nominalzinssatz}}{\text{Periodenzahl}} \right)^{\text{Periodenzahl}}$$

Die Funktion NOMINAL() löst die obige Gleichung nach dem Nominalzinssatz auf.

Praxiseinsatz Zusammenhänge Die Zusammenhänge zwischen Nominal- und Effektivzinssatz werden in den Beispielen zur Funktion EFFEKTIV() deutlich. Dort wird auch auf die Problematik der Preisangabenverordnung in Deutschland eingegangen.

ISMA-Kurs und ISMA-Rendite Im Beispiel zum Halbjahreskupon zur Anwendung der Funktion KURS() finden Sie die Nachbildung des ISMA-Kurses durch die integrierte Excel-Funktion. Hierzu ist es notwendig, die zu verwendende Rendite vor Anwendung der Funktion mittels NOMINAL() umzurechnen.

Siehe auch EFFEKTIV()

NOTIERUNGBRU()



DOLLARFR()

Syntax NOTIERUNGBRU(*Zahl;Teiler*)

Definition Diese Funktion wandelt die Dezimalstellen einer Dezimalzahl in den Zähler eines Bruchs mit vorgegebenem Nenner um und stellt diese in den Nachkommastellen dar.

Argumente *Zahl* (erforderlich) ist die Dezimalzahl, deren Dezimalstellen umgewandelt werden sollen.

Teiler (erforderlich) definiert den Nenner, für den der Zähler ermittelt wird.

Ist *Teiler* keine ganze Zahl, werden die Nachkommastellen abgeschnitten. Außerdem wird ein positiver Teiler erwartet. Ist er gleich 0 (Null), gibt NOTIERUNGBRU() den Fehlerwert #DIV/0! zurück. Im Falle negativer Zahlen lautet die Fehlermeldung #ZAHL!.

Zwei Beispiele: Die Aktiennotierungen an den großen Börsen in den USA wurden bis 2001⁷ aus traditionellen Gründen mithilfe natürlicher Brüche vorgenommen. Zinsfestlegungen in Deutschland geschehen in der Regel in Achtel-Schritten.

Die Funktion hilft bei der Umwandlung in die genannte Darstellung und erleichtert damit einen Vergleich der Notierungen.

Die Abbildung 14.3 zeigt die Wirkungen der Funktion.

	A	B	C	D	E
2					
3		Nenner	8	4	2
4		8,250	8,2	8,1	8,05
5		1,125	1,1	1,05	1,025

Abbildung 14.3: Umrechnung in Achtel, Viertel und Halbe

Die Ergebnisse sind zu lesen als: 8,25 ist gleich $8\frac{1}{4}$ und 1,125 ist gleich $1\frac{1}{8}$.

Die Ergebnisse der Umwandlung sind nur zum Lesen da, nicht zum Weiterrechnen. Multiplizieren Sie das Ergebnis in der Spalte *D* etwa mit 10, entsteht im ersten Fall 81 und nicht etwa 82,5.

NOTIERUNGDEZ()

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Sonstiges.xlsx* auf dem Arbeitsblatt *NOTIERUNGBRU*.

NOTIERUNGDEZ()



NOTIERUNGDEZ(*Zahl*; *Teiler*)

Diese Funktion wandelt einen als Zähler eines Bruchs zu interpretierenden Nachkommanteil einer Zahl bei gegebenem Nenner in eine Dezimalzahl um.

Zahl (erforderlich) ist die Zahl, deren Dezimalstellen als Zähler eines Bruchs interpretiert werden.

Teiler (erforderlich) ist der Nenner des Bruchs, für den der Dezimalanteil von *Zahl* als Zähler interpretiert wird.

Für *Teiler* wird eine ganze Zahl erwartet, Nachkommastellen werden abgeschnitten. Ist *Teiler* kleiner 0 (Null), gibt NOTIERUNGDEZ() den Fehlerwert *#ZAHL!* zurück. Im Falle von gleich 0 lautet der Fehlerwert *#DIV/0!*.

Zwei Beispiele: Die Aktiennotierungen an den großen Börsen in den USA wurden bis 2001 aus traditionellen Gründen mithilfe natürlicher Brüche vorgenommen. Zinsfestlegungen in Deutschland geschehen in der Regel in Achtel-Schritten.

Die Funktion hilft bei der Umwandlung aus der genannten Darstellung und erleichtert damit einen Vergleich der Notierungen.

Hintergrund

Praxiseinsatz

Achtung

Siehe auch



Neu in Excel 2013

Syntax

Definition

Argumente

Hintergrund

⁷ Quelle: <http://www.infoplease.com/spot/stockdecimal1.html>.

Praxiseinsatz Die Abbildung 14.4 zeigt die Wirkungen der Funktion.

	A	B	C	D	E	F
1						
2		Nenner	1,1	2,2	1,01	2,02
3		2	1,5	3	1,05	2,1
4		4	1,25	2,5	1,025	2,05
5		8	1,125	2,25	1,0125	2,025
6		16	1,625	3,25	1,0625	2,125

Abbildung 14.4: Umrechnung aus Halben, Vierteln, Achteln, Sechzehnteln

Die Ergebnisse der Spalte C interpretieren die erste Zahl 1,1 der Reihe nach als $1\frac{1}{2}$, $1\frac{1}{4}$ und $1\frac{1}{8}$. Problematisch ist die Interpretation als $1\frac{1}{16}$, da hier die Zahl der Dezimalstellen (es ist eine einzige) nur die Interpretation bis zu Zehnteln erlaubt. Sie sehen die Lösung in der Spalte E. In Spalte D wird die Zahl 2,2 interpretiert – die Zahlen vor dem Komma sind die Ganzen, die nach dem Komma geben die Zahl der Viertel, Achtel usw. an. Auch hier können die Sechzehntel nicht korrekt interpretiert werden.

Achtung Die Zahlen, die umgewandelt werden sollen, können in der Regel nur per Hand eingegeben werden und nicht als Ergebnisse von Rechnungen – außer durch NOTIERUNGBRU() – entstanden sein. Mit den Umwandlungsergebnissen kann aber – im Gegensatz zu NOTIERUNGBRU() – »normal« weitergerechnet werden.

Siehe auch NOTIERUNGBRU()



Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap14` in der Arbeitsmappe `Sonstiges.xlsx` auf dem Arbeitsblatt `NOTIERUNGDEZ`.

PDURATION()



PDURATION()

Syntax PDURATION(*Zins*; *Bw*; *Zw*)

Definition Diese Funktion berechnet die Laufzeit einer Investition bei gegebenem Zinssatz sowie eingesetztem und erwartetem Kapital.

Argumente *Zins* (erforderlich) ist der veranschlagte Zinssatz pro Periode.

Bw (erforderlich) ist das eingesetzte Kapital.

Zw (erforderlich) ist das zu erhaltende Kapital.

Hintergrund Die fünf Funktionen `BW()` = Barwert, `ZW()` = Zukunftswert, `RMZ()` = regelmäßige Zahlung, `ZZR()` = Zins- oder Zahlungszeiträume und `ZINS()` = Zinssatz stehen in Umsetzung des obigen Äquivalenzprinzips in folgendem Zusammenhang:

$$BW \cdot (1 + ZINS)^{ZZR} + RMZ \cdot (1 + F \cdot ZINS) \cdot \frac{(1 + ZINS)^{ZZR} - 1}{ZINS} + ZW = 0$$

Der Barwert wird bis zum Ende aufgezinst, ebenso die regelmäßigen Zahlungen. Zum Ende wird die Summe mit dem Zukunftswert verglichen.

Die Nutzung einer dieser Funktionen ist gleichbedeutend mit der jeweiligen finanzmathematischen Grundaufgabe: Berechnung eines Werts aus obiger Gleichung, wenn die anderen Werte bekannt sind.

Die Funktion PDURATION() reduziert diese Formel für den Fall $RMZ=0$ (das Kapital bleibt konstant in der Verzinsung, es gibt keine Zuzahlungen oder Entnahmen) auf die Auflösung nach ZZR:

$$BW \cdot (1 + ZINS)^{ZZR} - ZW = 0$$

Abweichend werden jedoch der Bar- und der Zukunftswert als positive Zahlen gefordert.

Die Funktion PDURATION() steht in keinem Zusammenhang zu DURATION() oder MDURATION().

Aufgaben zur Zinseszinsrechnung mithilfe der Funktion ZZR() können auch mithilfe von PDUARTION() behandelt werden.

BW(), RMZ(), ZINS(), ZSATZINVEST(), ZW(), ZZR(),

Ein Beispiel finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Zinseszinsrechnung.xlsx* auf dem Arbeitsblatt *PDURATION*.

Hinweis

Praxiseinsatz

Siehe auch



QIKV() MIRR()

QIKV(*Werte;Investition;Reinvestition*)

Diese Funktion berechnet die interne Verzinsung einer Investition, wobei negative Periodenüberschüsse mit einem anderen Zinssatz als positive bewertet werden (»qualifizierte interne Kapitalverzinsung«).

Werte (erforderlich) verlangt die lückenlos in einer Spalte angeordneten (tatsächlichen oder erwarteten) Überschüsse aus Aus- und Einzahlungen. Jeder Wert vertritt das Ende einer Periode (in der Regel ein Jahr) in aufsteigender Ordnung und ohne Lücken. Negative Überschüsse sollen mit einem Minuszeichen versehen sein. Es werden wenigstens zwei Werte erwartet, davon mindestens einer positiv und einer negativ.

Investition (erforderlich) ist der Zinssatz, mit dem die negativen Beträge (Auszahlungen) auf ihren Barwert reduziert (abgezinst) werden.

Reinvestition (erforderlich) ist der Zinssatz, mit dem die positiven Beträge (Einzahlungen) auf ihren Endwert aufgezinst werden.

Enthalten Zellen des Arguments *Werte* keine Zahlen oder sind leer, rechnet Excel so, als ob diese Zellen nicht vorhanden sind. Das ist auch so, wenn eventuelle Zellbezüge im Argument auf solche Zellen verweisen.

Die Benutzung der Methode der internen Kapitalverzinsung zur Bewertung von Investitionen ist Gegenstand der dynamischen Investitionsrechnung. Wird dabei wie in den Erläuterungen zu den Funktionen IKV() und NBW() vorgegangen, gibt es Vor- und Nachteile. Ein Vorteil der Kapitalwertbildung durch Abzinsen aller zukünftigen Zahlen ist der nach oben offene Zeithorizont, der keine Überlegungen zur Wiederanlegbarkeit von Erträgen fordert. Ein Nachteil ist der dabei stets vorausgesetzte sich nicht ändernde Zinssatz zur Bewertung.

Syntax

Definition

Argumente

Hintergrund

Die Methode des »qualifizierten internen Zinssatzes« schränkt den Zeithorizont nach oben ein, lässt aber dafür etwas Spielraum bei den Bewertungszinssätzen. Alle Auszahlungen werden zum »Fremdkapitalzinssatz« durch Abzinsen zu ihrem heutigen Wert, also dem Barwert beurteilt. Alle Einzahlungen werden wie Anlagen behandelt, das Ergebnis ist ihr Zukunftswert (der durch Aufzinsen entsteht) zum Ende der Investition.

Der qualifizierte interne Zinssatz ist der, der durch Verzinsung des Barwerts der Auszahlungen gerade zum Zukunftswert der Einzahlungen führt – er beantwortet also die Frage, zu welchem Zinssatz ausgegebenes Geld am Ende wieder da ist.

Praxiseinsatz Jemand erwirbt eine Bundesobligation, die 5 Jahre lang jährlich 4,25 % Zinsen abwirft. Eine Wiederanlage der Zinsen ist aber aus verschiedenen Gründen nur zu 2 % möglich.

Während die interne Kapitalverzinsung (berechnet mittels IKV()) hier 4,25 % beträgt, bringt der Mischzins über QIKV() nur 4,08 % im Jahr.

	A	B	C	D	E	F	G
1							
2		Zinssatz der Obligation	(Investition)	4,25%			
3		Wiederanlagezinssatz	(Reinvestition)	2,00%			
4							
5		Anlagebetrag		100,00 €			
6							
7		Jahr	QIKV	Anlage	Wiederanlage		
8		0	- 100,00 €	- 100,00 €	- €		
9		1	4,25 €	- €	4,25 €		
10		2	4,25 €	- €	4,25 €		
11		3	4,25 €	- €	4,25 €		
12		4	4,25 €	- €	4,25 €		
13		5	104,25 €	- €	104,25 €		
14							
15		IKV	4,25%		Barwert der Anlage	- 100,00 €	
16					Endwert der Wiederanlage	122,12 €	
17		QIKV	4,08%				
18					Mischzins	4,08%	

Abbildung 14.5: Hinter QIKV() verbirgt sich eine »Mischzins-Kalkulation«

Die Abbildung 14.5 zeigt die Interpretation: Einer Ausgabe von 100,00 € steht am Ende ein (durch Aufzinsung entstandenes) Ergebnis von 122,12 € gegenüber.

Siehe auch IKV(), NBW(), ZINS(), XINTZINSFUSS(), XKAPITALWERT()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Investitionsrechnung.xlsx* auf dem Arbeitsblatt QIKV.

RENDITE()



RENDITE(*Abrechnung*; *Fälligkeit*; *Zins*; *Kurs*; *Rückzahlung*; *Häufigkeit*; *Basis*)

Diese Funktion ermittelt die Rendite eines festverzinslichen Wertpapiers (Anleihe).

Abrechnung (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum, an dem die Rückzahlung des durch das Wertpapier verbrieften Darlehens erfolgt.

Zins (erforderlich) informiert über den vereinbarten Jahreszinssatz als Preis für das geliehene Geld (Kuponzinssatz).

Kurs (erforderlich) nimmt den Kurs des Wertpapiers am Abrechnungstag auf (ohne Stückzinsen).

Rückzahlung (erforderlich) ist der Prozentfuß bezogen auf den Nennwert des Wertpapiers (also so, als ob dieser 100 Geldeinheiten wäre), zu welchem am Fälligkeitstag zurückgezahlt wird.

Da es durchaus üblich ist, festverzinsliche Wertpapiere mit mehreren Zinsterminen im Jahr auszustatten, informiert das erforderliche Argument *Häufigkeit* über deren Anzahl. Zulässige Werte sind: 1 = jährlich; 2 = halbjährlich; 4 = vierteljährlich.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.8. Wird dieses Argument weggelassen, rechnet Excel mit *Basis* = 0.

Basis	Methode	Bedeutung
0	30/360 (NASD-Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Ist das Ausgangsdatum der 31. eines Monats, wird dieses Datum zum 30. desselben Monats. Ist das Enddatum der 31. eines Monats und das Ausgangsdatum ein Datum vor dem 30. eines Monats, wird das Enddatum zum 1. des darauf folgenden Monats. In allen anderen Fällen wird das Enddatum zum 30. desselben Monats.
1	Genau/Genau	Jeder Monat hat so viele Tage wie im Kalender stehen, die Summe über alle Monate ergibt die Anzahl der Tage im Jahr
2	Genau/360 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 360 Tage.
3	Genau/365 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 365 Tage.
4	30/360 (Europäische Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Jedes Ausgangs- und Enddatum, das auf den 31. eines Monats fällt, wird zum 30. desselben Monats.

Syntax

Definition

Argumente

Tabelle 14.8

Verschiedene Methoden zur Ermittlung der Anzahl der Tage eines Jahres

An die Argumente von RENDITE() werden folgende Anforderungen gestellt:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, gibt RENDITE() den Fehlerwert #WERT! zurück

- ▶ Für *Zins* werden nicht negative Zahlen, für *Rückzahlung* sogar positive Zahlen verlangt. Andernfalls gibt RENDITE() den Fehlerwert #ZAHL! zurück.
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt RENDITE() den Fehlerwert #ZAHL! zurück. Das Gleiche geschieht, wenn der Abrechnungstermin nach der Fälligkeit liegt.

Wichtig Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Hintergrund Um das finanzmathematische Äquivalenzprinzip

Leistung des Gläubigers = Leistung des Schuldners

bezogen auf den »Startpunkt« des Geschäfts umzusetzen, ist der Kurs eines festverzinslichen Wertpapiers (Anleihe) plus eventuell zu zahlende Stückzinsen gleich dem Barwert der zukünftigen in diesem Wertpapier verbrieften Leistungen des Emittenten (Schuldners). Der Kurs wird als Prozentfuß bezogen auf den Nennwert des Wertpapiers angegeben, also so, als ob der Nennwert 100 Geldeinheiten beträgt.

Da der Kurs in der Regel durch Angebot und Nachfrage am Markt entsteht, dient das Äquivalenzprinzip dazu, diejenige Verzinsung zu errechnen, mit der ein Kurs durch Barwertberechnung zu bilden wäre, der dem am Markt entspricht. Diese innere Verzinsung heißt Rendite. Eine Auflösungsformel hierzu gibt es im Allgemeinen nicht. Es muss ein mathematisches Näherungsverfahren verwendet werden. Mit einem ausführlichen Rechenschema wie in den unten aufgeführten Beispielen hält Excel die Zielwertsuche zur Auflösung der Gleichung bereit.

Die Barwertberechnung ist dann ohne Probleme, wenn das Kaufdatum (Besitzwechsel) eines mit Jahreskupon versehenen Wertpapiers mit dem Tag der Zinszahlung einhergeht. In diesem Fall gibt es nur ganze Jahre zu berücksichtigen. Nicht so einfach ist die Situation, wenn der Besitzwechsel zwischen Zinsterminen erfolgt bzw. Zinszahlungen mehrfach im Jahr vereinbart sind. Die Finanzmathematik kennt verschiedene Arten des Umgangs mit gebrochenen Jahresanteilen. Die bekanntesten sind die Methoden Moosmüller und Braess/Fangmeyer sowie die ISMA-Methode. ISMA steht für International Securities Market Association, eine Organisation, die aus der Association of International Bond Dealers (AIBD) hervorgegangen ist.

Die ISMA-Methode liefert für eine einmalige Zinszahlung im Jahr das Gleiche wie RENDITE() und lässt sich für Halb- und Vierteljahreskups mit RENDITE() nachbilden.

Das genannte Äquivalenzprinzip durch Barwertbildung (Abzinsen zukünftiger Leistungen) lautet bei der Kursberechnung durch Excel in der von Excel und seinen Funktionen entlehnten Terminologie

$$\begin{aligned}
 KURS = & \frac{\text{Rückzahlung}}{\left(1 + \frac{\text{Rendite}}{\text{Häufigkeit}}\right)^{\text{ZINSTERMZAHL} - 1 + \frac{\text{ZINSTERMTAGNZ}}{\text{Tage der Zinsperiode}}} + \\
 & + \sum_{K=1}^{\text{ZINSTERMZAHL}} \frac{\frac{\text{ZINS}}{\text{Häufigkeit}} \cdot 100}{\left(1 + \frac{\text{Rendite}}{\text{Häufigkeit}}\right)^{K-1 + \frac{\text{ZINSTERMTAGNZ}}{\text{Tage der Zinsperiode}}} - \\
 & - \frac{\text{ZINS}}{\text{Häufigkeit}} \cdot 100 \cdot \frac{\text{ZINSTERMTAGVA}}{\text{Tage der Zinsperiode}}
 \end{aligned}$$

wobei mit *ZINSTERMZAHL* die Anzahl der Zinszahlungen nach dem Kauf, mit *ZINSTERMTAGNZ* die Anzahl der Tage bis zum nächsten Zinstermin und mit *ZINSTERMTAGVA* die Anzahl der Tage seit dem letzten Zinstermin bezeichnet wurden.

Ist *Häufigkeit* = 1, stellt diese Formel gerade auch die ISMA-Methode dar. Ist die *Häufigkeit* gleich 2 oder 4, rechnet Excel mit einer gleichmäßigen Verteilung der *Rendite* auf die Perioden innerhalb des Jahrs. Die ISMA-Methode dagegen verwendet eine periodengerechte Umlegung nach dem Zusammenhang zwischen Nominal- und Effektivzins, den Sie in den Erläuterungen zur Funktion *EFFEKTIV()* bzw. *NOMINAL()* finden:

$$\text{Rendite der Periode} = \frac{\text{NOMINAL}(\text{Rendite im Jahr}; \text{Häufigkeit})}{\text{Häufigkeit}}$$

Deshalb können Sie im Falle von mehreren Zinszahlungen pro Jahr die Rendite nach der ISMA-Methode mithilfe von *RENDITE()* berechnen, indem Sie im Nachhinein eine Umrechnung der Excel-Rendite in eine »effektive« vornehmen (durch Anwenden der Funktion *EFFEKTIV()*).

Die oben gegebenen Erläuterungen gelten nur, wenn mehr als eine einzige Zinsperiode bis zur Fälligkeit verbleibt. Anderenfalls rechnet Excel nicht nach der obigen Formel für den Kurs, sondern benutzt statt der gebrochenen Laufzeit eine gebrochene Rendite (einfache Verzinsung). *RENDITE()* liefert dann das gleiche Ergebnis wie *RENDITEFÄLL()*.

Achtung

Die 3,500 %-Bundesanleihe von 05 wurde am 30.1.2013 zu einem Kurs von 109,010 gehandelt. Ihre Fälligkeit ist am 4.1.2016. Welche Rendite konnten Sie in den entsprechenden Mitteilungen oder auch Tageszeitungen finden?

**Praxiseinsatz
Jahreskupon**

Wollen Sie die ISMA-Rendite nach der Tagezahl-Methode taggenau/taggenau bestimmen, ohne die Funktion *RENDITE()* zu verwenden (Sie wollen verstehen, was *RENDITE()* genau umsetzt), können Sie ein Arbeitsblatt wie in Abbildung 14.6 vorbereiten.

	A	B	C
1			
2		Jahreskupon	
3			
4		ISMA=Excel	
5			
6	Kaufdatum		30.01.2013
7	Kaufkurs		109,010
8	Rückzahlung		100,00
9	Fälligkeit		04.01.2016
10	Nominalzins		3,500%
11	Tagezählweise		1
12	Häufigkeit		1
13			
14	Nächster Zinszahlungstermin		04.01.2014
15	Jahresrest in Tagen		339
16	Tage für Stückzinsen		26
17	Tage im Jahr		365
18	Stückzinsen		0,25
19	Anzahl der Zinstermine		3
20	BW der Zinsen am nächsten Zinstermin		10,45
21	BW der Rückzahlung am nächsten Zinstermin		99,01
22			
23	Kaufpreis		109,26
24	Barwert der zukünftigen Zahlungen		108,95
25	Differenz		0,3099
26			
27	Rendite p.a.		0,5000%

Abbildung 14.6: Vorbereitung von Kurs- und Renditeberechnungen

In die ersten sieben Zellen von C6 bis C10 geben Sie die bekannten Daten ein. Vorbereitend notieren Sie die Tagezählweise mit 1 (das entspricht $Basis = 1$ in allen verwendeten integrierten Funktionen) sowie die Häufigkeit der Kuponzahlungen.

Als Nächstes ermitteln Sie mit =ZINSTERMNZ(C6;C9;C12;C11) den nächsten Zinstermin: 4.1.2014. Die Formel =ZINSTERMTAGNZ(C6;C9;C12;C11) hilft Ihnen beim Zählen der Tage bis zu diesem Zinstermin: 339. Zur Berechnung der Stückzinsen brauchen Sie die Tage seit dem letzten Zinstermin (=ZINSTERMTAGVA(C6;C9;C12;C11) zählt hier 26), sodass Sie über $C10 * C16 / C17 * C8$ einen Betrag von 0,25 fast im Kopf bestimmen können. Die Zahl der verbleibenden Zinstermine, die Sie zum Abzinsen der zukünftige Leistungen brauchen, wird durch =ZINSTERMZAHL(C6;C9;C12;C11) zu 3 ermittelt.

Nun geht es ans Abzinsen. Hier hilft Ihnen die Funktion BW(), und zwar mit =BW(C27;C19-1;-C10*C8;-C10*C8;1) als Aufgabe der Rentenrechnung zur Berechnung des Barwerts zukünftiger Zinszahlungen (Ergebnis 10,45) sowie als Aufgabe der Zinseszinsrechnung zum Abzinsen des Rückzahlbetrags über =BW(C27;C19-1;0;-C8) zu 99,01. Bei diesem Abzinsen verwenden Sie eine zunächst gedachte Rendite von 0,5 % in C27.

Die Summe der beiden Barwerte ist nun, entsprechend der Vorgabe von ISMA, über die gebrochene Laufzeit von 339 aus 365 Tagen mit einem gebrochenen Exponenten in der Zinsformel zu ermitteln: =BW(C27;C15/C17;0;-C21-C20) liefert 108,95.

Sie vergleichen nun die Summe aus Kurs und zu zahlenden Stückzinsen (C23) mit der Summe der berechneten Barwerte (C24) durch Differenzbildung in C25 (es entsteht der Wert

0,3099). Auf diese Zelle lassen Sie die Zielwertsuche mit Zielwert 0 und veränderbarer Zelle C27 (dort steht ihre bisher angenommene Rendite) wirken.

Im Ergebnis kommt dann das heraus, was auch =RENDITE(C6;C9;C10;C7;C8;C12;C11) ermittelt: 0,399 %.

In Deutschland gibt es wohl nur wenige Anleihen, die mehr als einen Zinstermin pro Jahr haben. In der Beispieldatei zum Buch wurde deshalb die Anleihe des vorigen Beispiels fiktiv mit einem Halbjahreskupon ausgestattet und die Berechnungen durchgeführt. Der Excel-Kurs unterscheidet sich von dem nach ISMA in Nachkommastellen. Wollen Sie die ISMA-Rendite bestimmen, ist die durch RENDITE() errechnete Zahl (sie ist das Doppelte der korrekten ISMA-Halbjahresrendite) mittels EFFEKTIV(RENDITE();Häufigkeit) periodengerecht auszuwerten.

KURS(), ZINS(), DURATION(), MDURATION()

Die beiden Beispiele finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe Kursrechnung.xlsx auf dem Arbeitsblatt RENDITE.

Halbjahreskupon

Siehe auch



RENDITEDIS() YIELDDISC()

RENDITEDIS(*Abrechnung*; *Fälligkeit*; *Kurs*; *Rückzahlung*; *Basis*)

Diese Funktion berechnet die jährliche Rendite (als nachschüssigen Zinssatz) für ein Wertpapier, welches mit einem Abschlag (Diskont, Disagio) für eine in der Regel unterjährliche Laufzeit ausgestattet wurde.

Diese Funktion unterscheidet sich von der Funktion ZINSSATZ() nur durch die Benennung der Argumente.

Abrechnung (erforderlich) ist der Tag, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) erfasst den Tag, an dem die Leistungen aus dem Wertpapier erfolgen müssen.

Kurs (erforderlich) ist der Kaufpreis des Wertpapiers am Abrechnungstag.

Rückzahlung (erforderlich) beschreibt die Leistung am Fälligkeitstag.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.8. Wird es weggelassen, rechnet Excel mit $Basis = 0$.

Alle Argumente der Funktion, die ein Datum betreffen, verwenden eine Datumsangabe ohne Uhrzeit, gebrochene Zahlen werden also abgerundet. Das Argument *Basis* verlangt ebenfalls eine ganze Zahl und schneidet Nachkommastellen ab.

Folgendes ist zu beachten:

- ▶ Werden ungültige Datumseinträge verwendet bzw. keine Zahlen dort angegeben, wo sie erforderlich sind, liefert die Funktion den Fehlerwert #WERT!
- ▶ Für *Kurs* und *Rückzahlung* werden positive Zahlen verlangt. Andernfalls gibt RENDITEDIS() den Fehlerwert #ZAHL! zurück.
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt RENDITEDIS() den Fehlerwert #ZAHL! zurück. Das Gleiche geschieht, wenn der Abrechnungstermin nach der Fälligkeit liegt.

Syntax

Definition

Argumente

Wichtig Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Die Hinweise auf einen Nennwert von 100 sind falsch.

Hintergrund Vom Grundsatz her gehört diese Funktion in den Umkreis der vorschüssigen Verzinsung. Das Prinzip besteht in der Annahme, dass sich ein Anfangskapital (einzuzahlender Betrag, Darlehen) aus dem Rückzahlungsbetrag minus die Zinsen auf diesen Betrag ergeben. Dieses Prinzip wird vor allem im unterjährlichen Bereich verwendet und unterscheidet sich von dem Prinzip, welches einem Sparbuch oder einem Hypothekendarlehen zugrunde liegt. Dort werden nämlich die Zinsen am Ende einer Periode auf Basis des Anfangskapitals berechnet (nachschüssige Verzinsung).

Um die Vergleichbarkeit von Wertpapieren mit nachschüssiger Verzinsung herzustellen, wird die Differenz zwischen Rückzahlung und Kurs ins Verhältnis zum Kurs gesetzt. Das Ergebnis entspricht dem relativen Zuwachs bis zur Fälligkeit, also einem nachschüssigen Zinssatz, der hier, wie oft in ähnlichen Situationen, den Namen Rendite erhält. Die Berechnung erfolgt ohne Zinseszinsseffekte, der Zinssatz ist unter diesem Aspekt noch auf ein Jahr (bei entsprechender Tagezählung) hochzurechnen.

Praxiseinsatz

Wechselrechnung Wechsel haben eine lange Tradition im Geldverleih, sie stellen eine Form des Lieferantenkredits dar. Es erfolgt vorschüssige Verzinsung. Will nun ein Geschäftsmann die Aufnahme eines Lieferantenkredits mittels Wechsel gegen die Effektivverzinsung seines Kontokorrentkredits vergleichen, muss er den vorschüssigen Zinssatz äquivalent umrechnen. Dazu kann er RENDITEDIS() (oder auch ZINSSATZ()) verwenden.

Am 10.5.2013 legt ein Geschäftsmann seiner Bank einen Wechsel in Höhe von 5.000 € mit einer (Rest-) Laufzeit von 2 Monaten vor. Diese setzt einen Diskontsatz von 5 % p.a. an und fordert keine Gebühren. Er erhält eine Gutschrift von 4.958,33 € (das ist 5.000,00 € minus 5 % davon). Wie hoch ist der Preis als nachschüssiger Zinssatz, mit dem sich die Bank die sofortige Gutschrift bezahlen lässt?

Mit

```
=RENDITEDIS("10.5.2013";"10.7.2013";4958,33;5000;4)
```

berechnen Sie 5,04 % als Jahreszinssatz.

Die konkreten Zahlen stehen natürlich in Zellen und die Formel verwendet Zellbezüge.

Siehe auch AUSZAHLUNG(), DISAGIO(), KURSDISAGIO(), ZINSSATZ()



Dieses Beispiel finden Sie auf dem Arbeitsblatt *Wechselrechnung* in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* im Ordner *\Ms5-235\Kap14*.

RENDITEFÄLL()



RENDITEFÄLL(*Abrechnung;Fälligkeit;Emission;Zins;Kurs;Basis*)

Berechnet die jährliche Rendite eines festverzinslichen Wertpapiers im unterjährlichen Bereich (ohne Zinseszinsseffekte).

Abrechnung (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) erwartet den Fälligkeitstermin des Wertpapiers.

Emission (erforderlich) ist das Datum der Ausgabe des Wertpapiers.

Zins (erforderlich) ist der vereinbarte Nominalzinssatz des Wertpapiers bezogen auf ein ganzes Jahr.

Kurs (erforderlich) ist der Prozentfuß bezogen auf den Nennwert des Wertpapiers (also so, als ob dieser 100 Geldeinheiten wäre).

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.8. Wird es weggelassen, rechnet Excel mit $Basis = 0$.

An die Argumente der Funktion werden folgende Forderungen gestellt:

- ▶ *Abrechnung*, *Fälligkeit*, und *Emission* verlangen eine Datumsangabe ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Ebenso wird aus *Basis* durch Abschneiden der Nachkommastellen eine ganze Zahl.
- ▶ Sind *Abrechnung*, *Fälligkeit* oder *Emission* kein zulässiges Datum, gibt RENDITEFÄLL() den Fehlerwert #WERT! zurück. Der Abrechnungstermin muss vor dem Fälligkeitstermin liegen.
- ▶ Ist *Zins* oder *Kurs* kleiner 0, liefert RENDITEFÄLL() den Fehlerwert #ZAHL!
- ▶ *Basis* muss sich zu einer Zahl zwischen 0 und 4 auswerten lassen, sonst gibt RENDITEFÄLL() den Fehlerwert #ZAHL! zurück

Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Um das finanzmathematische Äquivalenzprinzip

Leistung des Gläubigers = Leistung des Schuldners

bezogen auf den »Startpunkt« des Geschäfts umzusetzen, ist der Kurs eines festverzinslichen Wertpapiers plus eventuell zu zahlende Stückzinsen gleich dem Barwert der zukünftigen in diesem Wertpapier verbrieften Leistungen des Emittenten (Schuldners). Der Kurs wird als Prozentfuß bezogen auf den Nennwert des Wertpapiers angegeben, also so, als ob der Nennwert 100 Geldeinheiten beträgt. Im vorliegenden Fall besteht die zukünftige Leistung in der Rückzahlung plus Nominalzinsen. Zinseszinsseffekte treten nicht auf, da nur Bewertungen im unterjährlichen Bereich vorgenommen werden. Beim Abzinsen wird nicht mit gebrochener Laufzeit im Exponenten (ISMA-Methode), sondern mit einem gebrochenen, der Laufzeit entsprechenden Zinssatz gerechnet.

Syntax

Definition

Argumente

Wichtig

Hintergrund

Um die so definierte Rendite auszurechnen, wird die Differenz zwischen den zukünftigen Leistungen des Schuldners und der Gegenwartsleistung des Gläubigers (Käufer des Wertpapiers) ins Verhältnis zu dieser Gegenwartsleistung gesetzt:

$$\frac{(\text{Rückzahlung} + \text{Zinsen}) - (\text{Kurs} + \text{Stückzinsen})}{\text{Kurs} + \text{Stückzinsen}}$$

Dabei berechnen sich die Zinsen aus dem jährlichen Nominalzinssatz bezogen auf die Differenz zwischen Fälligkeits- und Emissionsdatum. Die Stückzinsen ermitteln sich genau so, nur dass hier die Zeitspanne zwischen Kaufzeitpunkt und Emissionsdatum entscheidend ist. Die so ermittelte Verhältniszahl wird durch proportionales Hochrechnen auf das Jahr zur Rendite.

Praxiseinsatz Eine Bundesanleihe von 2003 wurde mit einem Kupon von 4,250 % ausgestattet, der Fälligkeitstermin wurde auf den 4.1.2014 festgelegt. Am 30.1.2013 betrug der Kurs 103,790. Wie hoch war zu diesem Zeitpunkt die Rendite? Durch Einsetzen der entsprechenden bekannten Daten in `RENDITEFÄLL()` liefert 0,15 %. Das gleiche Ergebnis erhalten Sie auch mit der Funktion `RENDITE()`.

Siehe auch `KURSFÄLLIG()`



Das Beispiel finden Sie auf dem Arbeitsblatt `RENDTEFÄLL` in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* im Ordner `\Ms5-235\Kap14`.

RMZ() PMT()

Syntax `RMZ(Zins;Zzr;Bw;Zw;F)`

Definition Diese Funktion liefert die Höhe der Rente (regelmäßige Zahlung) eines der Rentenrechnung zuzuordnenden Vorgangs. Im Falle der Tilgungsrechnung ist hier die Annuität gemeint, falls das Darlehen mittels Annuitätentilgung zurückgezahlt wird.

Wichtig Das von Excel verwendete finanzmathematische Äquivalenzprinzip

Leistung des Schuldners + Leistung des Gläubigers = 0

bedeutet, dass hinsichtlich des Vorzeichens zwischen Aus- und Einzahlungen (Kreditaufnahme und Tilgung, Investition und Desinvestition) unterschieden werden muss. Excel weicht hier vom in der finanzmathematischen Literatur gebräuchlichen Äquivalenzprinzip

Leistung des Schuldners = Leistung des Gläubigers

ab.

Argumente *Zins* (erforderlich) gibt den Periodenzinssatz an. Oft handelt es sich dabei um einen Jahreszinssatz.

Zzr (erforderlich) erfasst die Anzahl der Zinstermine über die gesamte Laufzeit. Es wird angenommen, dass diese Termine mit den Rentenzahlungsterminen zusammenfallen.

Bw (erforderliche/optional) ist der Anfangswert einer Richtung der Zahlungen. Für Auszahlpläne handelt es sich um den Kontostand am Anfang der Betrachtungen, bei Tilgungsplänen um die Kreditsumme.

Zw (optional/erforderlich) ist der Endwert einer Richtung der Zahlungen. Bei Sparvorgängen ist es das angestrebte Sparziel, bei Auszahlungsplänen ein gewünschter Kontostand am Ende der Betrachtungen, bei Tilgungsplänen die im gegebenen Zeitraum angestrebte Restschuld.

F (optional) legt fest, ob die regelmäßigen Zahlungen zu Beginn der Zinsperioden ($F = 1$) oder an deren Ende ($F = 0$) erfolgen. Wird das Argument nicht angegeben, rechnet Excel mit $F = 0$.

Eines der beiden Argumente Bw und Zw muss unbedingt angegeben werden.

Die fünf Funktionen $BW()$ = Barwert, $ZW()$ = Zukunftswert, $RMZ()$ = regelmäßige Zahlung, $ZZR()$ = Zins- oder Zahlungszeiträume und $ZINS()$ = Zinssatz stehen in Umsetzung des von Excel verwendeten Äquivalenzprinzips in folgendem Zusammenhang:

$$BW \cdot (1 + ZINS)^{ZZR} + RMZ \cdot (1 + F \cdot ZINS) \cdot \frac{(1 + ZINS)^{ZZR} - 1}{ZINS} + ZW = 0$$

Der Barwert wird bis zum Ende aufgezinst, ebenso die regelmäßigen Zahlungen. Zum Ende wird die Summe mit dem Zukunftswert verglichen.

Die Nutzung einer dieser Funktionen ist gleichbedeutend mit der jeweiligen finanzmathematischen Grundaufgabe: Berechnung eines Werts aus obiger Gleichung, wenn die anderen Werte bekannt sind. Die Funktionen stellen also die Gleichung nach der gleichnamigen Größe um (außer bei $ZINS$ – hier wird eine Näherungsrechnung durchgeführt).

Gewöhnlich erfolgen die Angaben des Zinssatzes als Jahreszinssatz. Die genannten Funktionen arbeiten nur dann korrekt, wenn die Perioden zum Zinssatz passen. Bei unterjährlicher Verzinsung wird in der Regel der Jahreszinssatz gleichmäßig auf die Perioden aufgeteilt: 12 Monate zu einem Zwölftel des Zinssatzes, 3 Monate zu einem Viertel und ein halbes Jahr zur Hälfte.

Die folgenden Beispiele orientieren sich in den Überschriften am »finanzmathematischen Sprachgebrauch«.

Jemand hat bis zum 60. Lebensjahr 100.000,00 € angespart und möchte sich monatlich eine zusätzliche Rente auszahlen lassen. Es gelingt, für 15 Jahre 4,5 % p.a. auf das jeweils vorhandene Guthaben zu vereinbaren. Wie hoch fällt die Rente aus, wenn Kapitalverzehr geplant ist bzw. wie hoch fällt sie aus, wenn am Ende noch 10.000,00 € auf dem Konto sein sollen?

Im ersten Fall ermitteln Sie die Rente aus

$$=RMZ(4,5\%/12;15*12;-100000)$$

zu 764,99 €, im zweiten aus

$$=RMZ(4,5\%/12;15*12;-100000;10000)$$

zu 725,99 €. Das Minuszeichen bei Bw interpretieren Sie als Verzicht auf sofortige Auszahlung, Zw und RMZ »zeigen« dann in die andere Zahlungsrichtung.

Die Rechnung geht von Zinseszinsen innerhalb eines Jahrs aus. Auf einem Sparbuch lässt sich der Zinseszinsseffekt allerdings so nicht umsetzen, sodass das Ergebnis etwas »theoretisch« ist.

Jemand hat ein Annuitätendarlehen über 100.000,00 € aufgenommen. Es wurde ein Zinssatz von 5,5 % p.a. für eine Zinsbindungsfrist von 5 Jahren vereinbart. Die Bank teilt mit, dass die Restschuld nach 5 Jahren noch 80.000 € beträgt.

Hintergrund

Hinweis

Praxiseinsatz

**Renten-
rechnung**

**Tilgungs-
rechnung
(Annuitäten-
tilgung)**

Wie groß ist die (konstante) monatliche Belastung (Tilgung plus Zinsen) für den Schuldner?
 $=\text{RMZ}(5,5\%/12;5*12;100000;-80000)$

liefert die Lösung: 748,69 € (mit Minuszeichen).

Anders als beim Sparbuch erfolgt bei Hypothekendarlehen eine monatliche Verzinsung zu einem Zwölftel des vereinbarten Jahreszinssatzes (Nominalzins).

Wichtig Die Verwendung von Formeln unterscheidet sich von der ausführlichen Notierung eines Konto- verlaufs mit den gegebenen Ein- und Auszahlungen möglicherweise durch Rundungsfehler. Das liegt daran, dass auf einem tatsächlich geführten Konto Rundungen auf zwei Stellen nach dem Konto erfolgen. Gestalten Sie ein solches Konto unter Excel nach, ist in den Zwischenschritten, die konkretes Geld bedeuten, jeweils die Funktion RUNDEN() einzusetzen. Eine ledigliche Begrenzung der Anzeige des Zellwerts auf zwei Stellen ist oft nicht korrekt.

Siehe auch BW(), KAPZ(), PDURATION(), ZINS(), ZINSZ(), ZSATZINVEST(), ZW(), ZZR()



Die Arbeitsmappen *Rentenrechnung.xlsx* und *Tilgungsrechnung.xlsx* im Ordner \Ms5-235\Kap14 halten die Beispiele auf dem jeweiligen Arbeitsblatt RMZ bereit.

TBILLÄQUIV() TBILLEQ()

Syntax TBILLÄQUIV(*Abrechnung*; *Fälligkeit*; *Abzinsungssatz*)

Definition Diese Funktion errechnet den äquivalenten nachschüssigen Jahreszinssatz auf der Basis von 365 Tagen für einen gegebenen vorschüssigen Jahreszinssatz auf der Basis von 360 Tagen. Letzterer wird im Allgemeinen als Diskontsatz auf den Nennwert formuliert (Disagio). Anwendungsgebiet sind amerikanische Schatzanweisungen (Treasury Bills).

Argumente *Abrechnung* (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum der Rückzahlung des verbrieften Nennwerts.

Abzinsungssatz (erforderlich) ist der prozentuale Abschlag vom Nominalwert (Disagio) des Wertpapiers⁸.

Folgende Anforderungen werden gestellt:

- ▶ *Abrechnung* und *Fälligkeit* verlangen eine Datumsangabe ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Unzulässige Datumsangaben werden mit dem Fehlerwert #WERT! quittiert.
- ▶ Ist *Abzinsungssatz* nicht positiv, gibt TBILLÄQUIV() den Fehlerwert #ZAHL! zurück
- ▶ Der gleiche Fehler wird erzeugt, wenn *Abrechnung* nach *Fälligkeit* liegt oder *Fälligkeit* mehr als ein Jahr Abstand zu *Abrechnung* hat

Wichtig Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

⁸ Vor Excel 2013 heißt dieses Argument *Diskont*.

Die Äquivalenz zwischen vorschüssiger und nachschüssiger Verzinsung wird in diesem Falle durch die Gleichung

$$\left(1 + \frac{\text{Tage}}{365} \cdot \text{TBILLÄQUIV}\right) \cdot \left(1 - \frac{\text{Tage}}{360} \cdot \text{Abzinsungssatz}\right) = 1$$

hergestellt. Dabei ist *Tage* der taggenaue Abstand zwischen *Fälligkeit* und *Abrechnung*

In der Beispieldatei zum Buch finden Sie einige Zahlenexperimente, die durch »Handrechnung« die obige Formel nachgestalten und die Ergebnisse mit denen von `RENDITEDIS()` und `ZINSSATZ()` vergleicht, deren Aufgaben ebenfalls in der Umwandlung vorschüssiger Zinssätze in nachschüssige besteht.

TBILLKURS(), TBILLRENDITE()

Die genannten Vergleiche finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* auf dem Arbeitsblatt *TBILL*.

Hintergrund

Praxiseinsatz

Siehe auch



TBILLKURS() TBILLPRICE()

TBILLKURS(*Abrechnung*; *Fälligkeit*; *Abzinsungssatz*)

Diese Funktion liefert den Kurs eines abgezinsten Wertpapiers als Prozentfuß, so, als ob der Nennwert 100 Geldeinheiten beträgt.

Anwendungsgebiet sind amerikanische Schatzanweisungen (Treasury Bills).

Abrechnung (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum der Rückzahlung des verbrieften Nennwerts.

Abzinsungssatz (erforderlich) ist der prozentuale Abschlag vom Nominalwert (Diskont) des Wertpapiers⁹.

Folgende Anforderungen werden gestellt:

- ▶ *Abrechnung* und *Fälligkeit* verlangen eine Datumsangabe ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Unzulässige Datumsangaben werden mit dem Fehlerwert `#WERT!` quittiert.
- ▶ Ist *Abzinsungssatz* nicht positiv, gibt TBILLKURS() den Fehlerwert `#ZAHL!` zurück
- ▶ Der gleiche Fehler wird erzeugt, wenn *Abrechnung* nach *Fälligkeit* liegt oder *Fälligkeit* mehr als ein Jahr Abstand zu *Abrechnung* hat

Die Berechnung des Kurses ist hier einfache Prozentrechnung: Der vorschüssige Jahreszinssatz (Disagio) wird auf die tatsächliche Laufzeit heruntergebrochen. Die Tageszählung erfolgt mit 360 Tagen im Jahr, wobei jeder Monat so viele Tage hat wie im Kalender stehen. Das entspricht der Option *Basis* = 2 bei den Funktionen, die ein solches Argument wie in Tabelle 14.8 benutzen.

$$\text{TBILLKURS} = 100 * \left(1 - \frac{\text{Tage}}{360} \cdot \text{Abzinsungssatz}\right)$$

⁹ Vor Excel 2013 heißt dieses Argument *Disagio*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz In der Beispieldatei zum Buch finden Sie einige Zahlenexperimente, die die Äquivalenz zur Funktion KURSDISAGIO() zeigen, falls dort die Tageszählung mit *Basis* = 2 erfolgt.

Siehe auch TBILLÄQUIV(), TBILLRENDITE()



Die genannten Vergleiche finden Sie im Ordner \\Ms5-235\Kap14 in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* auf dem Arbeitsblatt *TBILL*.

TBILLRENDITE() TBILLYIELD()

Syntax TBILLRENDITE(*Abrechnung*; *Fälligkeit*; *Pr*)

Definition Diese Funktion liefert die Rendite eines abgezinsten Wertpapiers als nachschüssigen Jahreszinssatz.

Anwendungsgebiet sind amerikanische Schatzanweisungen (Treasury Bills).

Argumente *Abrechnung* (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum der Rückzahlung des verbrieften Nennwerts.

Pr (erforderlich) ist der Kurs des Wertpapiers am Abrechnungstag als Prozentfuß, also so zu interpretieren, als ob der Nennwert des Papiers 100 Geldeinheiten beträgt¹⁰.

Folgende Anforderungen werden gestellt:

- ▶ *Abrechnung* und *Fälligkeit* verlangen eine Datumsangabe ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Unzulässige Datumsangaben werden mit dem Fehlerwert #WERT! quittiert.
- ▶ Ist *Kurs* nicht positiv, gibt TBILLRENDITE() den Fehlerwert #ZAHL! zurück
- ▶ Der gleiche Fehler wird erzeugt, wenn *Abrechnung* nach *Fälligkeit* liegt oder *Fälligkeit* mehr als ein Jahr Abstand zu *Abrechnung* hat

Wichtig Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Hintergrund Die Berechnung der Rendite ist hier eine einfache Prozentrechnung: Der Gewinn zum Tage der Fälligkeit, der sich aus der Differenz zwischen 100 und dem bezahlten Kaufkurs ergibt, wird ins Verhältnis zum Kaufkurs gesetzt (das entspricht nachschüssiger Verzinsung). Da das Papier weniger als ein Jahr gehalten wird, ist der Zins zu Vergleichszwecken auf ein ganzes Jahr hochzurechnen:

$$\text{TBILLRENDITE} = \frac{100 - \text{Kurs}}{\text{Kurs}} * \frac{360}{\text{Tage}}$$

¹⁰ Vor Excel 2013 heißt dieses Argument *Kurs*.

Die Tageszählung erfolgt mit 360 Tagen im Jahr, wobei jeder Monat so viele Tage hat wie im Kalender stehen. Das entspricht der Option *Basis = 2* bei den Funktionen, die ein solches Argument wie in Tabelle 14.8 benutzen.

In der Beispieldatei zum Buch finden Sie einige Zahlenexperimente, die durch »Handrechnung« die obige Formel nachgestalten und die Ergebnisse mit denen von `RENDITEDIS()` und `ZINSSATZ()` vergleichen, deren Aufgaben ebenfalls in der Umwandlung vorschüssiger Zinssätze in nachschüssige Zinssätze (Renditen) besteht. Beide Funktionen liefern das Ergebnis `TBILLRENDITE()`, wenn die entsprechende Tageszählung (*Basis = 2*) verwendet wird.

`TBILLÄQUIV()`, `TBILLKURS()`

Die genannten Vergleiche finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* auf dem Arbeitsblatt *TBILL*.

Praxiseinsatz

Siehe auch



UNREGER.KURS() ODDPRICE()

`UNREGER.KURS(Abrechnung;Fälligkeit;Emission;Erster_Zinstermin;Zins;Rendite; Rückzahlung;Häufigkeit;Basis)`

Syntax

Berechnet den Kurs eines festverzinslichen Wertpapiers unter Berücksichtigung einer ersten Zinsperiode, die kürzer oder länger als die restlichen regelmäßigen Zinsperioden (Viertel-, halbe oder ganze Jahre) ausfällt.

Definition

Abrechnung (erforderlich) nimmt das Datum, an dem die Anleihe in den Besitz des Käufers übergeht.

Argumente

Fälligkeit (erforderlich) ist der Tag, an dem die Rückzahlung des verbrieften Betrags erfolgt.

Emission (erforderlich) ist das Datum der Ausgabe des Wertpapiers.

Erster_Zinstermin (erforderlich) ist das Datum der ersten Verzinsung.

Zins (erforderlich) informiert über die Nominalverzinsung (Jahreszinssatz) der Anleihe.

Rendite (erforderlich) ist der am Markt gültige Zinssatz für Anleihen der gegebenen Laufzeit.

Rückzahlung (erforderlich) ist der Prozentfuß der Rückzahlung bezogen auf den Nennwert des Wertpapiers (also so, als ob dieser 100 Geldeinheiten beträgt).

Häufigkeit (erforderlich) ist die Anzahl der regelmäßigen Zinszahlungen im Laufe eines Jahr (jährlich, halbjährlich, vierteljährlich).

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.9. Wird es weggelassen, rechnet Excel mit *Basis = 0*.

Tabelle 14.9
Verschiedene
Methoden zur
Ermittlung der
Anzahl der Tage
eines Jahrs

Basis	Methode	Bedeutung
0	30/360 (NASD-Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Ist das Ausgangsdatum der 31. eines Monats, wird dieses Datum zum 30. desselben Monats. Ist das Enddatum der 31. eines Monats und das Ausgangsdatum ein Datum vor dem 30. eines Monats, wird das Enddatum zum 1. des darauf folgenden Monats. In allen anderen Fällen wird das Enddatum zum 30. desselben Monats.
1	Genau/Genau	Jeder Monat hat so viele Tage wie im Kalender stehen, die Summe über alle Monate ergibt die Anzahl der Tage im Jahr
2	Genau/360 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 360 Tage.
3	Genau/365 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 365 Tage.
4	30/360 (Europäische Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Jedes Ausgangs- und Enddatum, das auf den 31. eines Monats fällt, wird zum 30. desselben Monats.

Folgendes ist zu beachten:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, gibt UNREG.KURS() den Fehlerwert #WERT! zurück
- ▶ Für *Zins* und *Rendite* werden nicht negative Zahlen verlangt. Andernfalls gibt UNREG.KURS () den Fehlerwert #ZAHL! zurück.
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt UNREG.KURS () den Fehlerwert #ZAHL! zurück
- ▶ Das Gleiche geschieht, wenn die Reihenfolge *Fälligkeit größer Erster_Zinstermin größer Abrechnung größer Emission* nicht eingehalten wird

Wichtig Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Hintergrund Um das finanzmathematische Äquivalenzprinzip

Leistung des Gläubigers = Leistung des Schuldners

bezogen auf den »Startpunkt« des Geschäfts umzusetzen, ist der Kurs eines festverzinslichen Wertpapiers (Anleihe) plus eventuell zu zahlende Stückzinsen gleich dem Barwert der zukünftigen in diesem Wertpapier verbrieften Leistungen des Emittenten (Schuldners). Der Kurs wird als Prozentfuß bezogen auf den Nennwert des Wertpapiers angegeben, also so, als ob der Nennwert 100 Geldeinheiten beträgt.

Die Barwertberechnung ist dann ohne Probleme, wenn das Kaufdatum (Besitzwechsel) eines mit Jahreskupon versehenen Wertpapiers mit dem Tag der Zinszahlung einhergeht. In diesem Fall gibt es nur ganze Jahre zu berücksichtigen. Nicht so einfach ist die Situation, wenn der Besitzwechsel zwischen Zinsterminen erfolgt bzw. Zinszahlungen mehrfach im Jahr vereinbart sind. Die Finanzmathematik kennt verschiedene Arten des Umgangs mit gebrochenen Jahresanteilen. Die bekanntesten sind die Methoden Moosmüller und Braess/Fangmeyer sowie die ISMA-Methode. ISMA steht für International Securities Market Association, eine Organisation, die aus der Association of International Bond Dealers (AIBD) hervorgegangen ist.

Über den Zusammenhang zwischen der ISMA-Methode und den Berechnungen der Excel-Funktionen lesen Sie bitte die Hintergrundinformationen zu KURS() und RENDITE().

Die in der Excel-Hilfe angegebene Formel reduziert sich für den Fall einer einzigen regelmäßigen Zinszahlung im Jahr (das Jahr zu 360 Tagen) und einer verkürzten ersten Zinsperiode zu

$$\begin{aligned} \text{UNREGER.KURS} = & \left[\frac{\text{Rückzahlung}}{(1 + \text{Rendite})^{N-1+DSC/360}} \right] + \left[\frac{100 \cdot \text{Zins} \cdot \text{DFC} / 360}{(1 + \text{Rendite})^{DSC/360}} \right] + \\ & + \left[\sum_{k=2}^N \frac{100 \cdot \text{Zins}}{(1 + \text{Rendite})^{k-1+DSC/360}} \right] - \left[100 \cdot \text{Zins} \cdot \frac{A}{360} \right] \end{aligned}$$

(N ist die Gesamtzahl der Zinstermine, A der Abstand zwischen Emissionstag und Abrechnungstag, DSC die Zeit vom Abrechnungstag bis zum ersten Zinstag und DFC die Zeit zwischen erstem Zinstag und Emissionsdatum).

Sie sehen, dass es als Besonderheit gegenüber der Kursformel aus den Hintergrundinformationen zur Funktion KURS() die Sonderbehandlung der ersten Periode gibt. Sie fällt aus der Summation und Barwertbildung der Folgeperioden heraus, weil die Zinszahlung nicht zum vollen Kupon erfolgt.

Ist die Häufigkeit größer als 1 (also 2, oder 4), ersetzen Sie Zins und Rendite durch die gleichmäßigen Aufteilungen auf die unterjährlichen Perioden.

Um die Formel der Excel-Hilfe für eine erste Zinsperiode, die länger ist als die restlichen »normalen«, zu verstehen, beachten Sie einfach, dass es nun in der ersten Periode zu mehreren fiktiven Zinszahlungen kommt, die am Ende der Periode in die Tat umgesetzt werden müssen.

Ähnlich ist mit den Stückzinsen in der ersten Periode zu verfahren.

Sobald der erste Zinstermin überschritten ist, findet die Funktion keine Verwendung mehr.

In der Beispieldatei zum Buch finden Sie ein Beispiel, welches einen Rechenweg für eine fiktive Anleihe mit verkürzter erster Zinsperiode aufbaut, der ähnlich dem aus Abbildung 14.2 (siehe Seite 652) verläuft. Das Ergebnis ist das gleiche, wie es auch durch Anwendung der Funktion UNREGER.KURS() geliefert wird.

KURS(), RENDITE(), UNREGER.REND(), UNREGLE.KURS(), UNREGLE.REND()

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Kursrechnung.xlsx* auf dem Arbeitsblatt *UNREGER.KURS*.

Praxiseinsatz

Siehe auch



UNREGER.REND() ODDFIELD()

Syntax UNREGER.REND(*Abrechnung*; *Fälligkeit*; *Emission*; *Erster_Zinstermin*; *Zins*; *Kurs*; *Rückzahlung*; *Häufigkeit*; *Basis*)

Definition Diese Funktion berechnet die Rendite eines festverzinslichen Wertpapiers für den Zeitraum vom Abrechnungstag bis zum Fälligkeitstermin. Dabei wird berücksichtigt, dass die erste Zinsperiode kürzer oder länger als die restlichen regelmäßigen Zinsperioden ist.

Argumente *Abrechnung* (erforderlich) entspricht dem Datum, an dem die Anleihe in den Besitz des Käufers übergeht.

Fälligkeit (erforderlich) ist der Tag, an dem die Rückzahlung des verbrieften Betrags erfolgt.

Emission (erforderlich) ist das Datum der Ausgabe des Wertpapiers.

Erster_Zinstermin (erforderlich) ist das Datum der ersten Verzinsung.

Zins (erforderlich) informiert über die Nominalverzinsung (Jahreszinssatz) der Anleihe.

Kurs (erforderlich) ist der Kurs des Wertpapiers am Abrechnungstag, angegeben als Prozentfuß auf den Nennwert des Wertpapiers (also so, als ob dieser 100 Geldeinheiten beträgt).

Rückzahlung (erforderlich) ist der Prozentfuß der Rückzahlung bezogen auf den Nennwert des Wertpapiers (also so, als ob dieser 100 Geldeinheiten beträgt).

Häufigkeit (erforderlich) ist die Anzahl der regelmäßigen Zinszahlungen im Laufe eines Jahrs (jährlich, halbjährlich, vierteljährlich).

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.9. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

Folgendes ist zu beachten:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, gibt UNREGER.REND() den Fehlerwert #WERT! zurück
- ▶ Für *Zins* und *Rendite* werden nicht negative Zahlen verlangt. Andernfalls gibt UNREGER.REND() den Fehlerwert #ZAHL! zurück.
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt UNREGER.REND() den Fehlerwert #ZAHL! zurück
- ▶ Das Gleiche geschieht, wenn die Reihenfolge *Fälligkeit* größer *Erster_Zinstermin* größer *Abrechnung* größer *Emission* nicht eingehalten wird

Wichtig Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Hintergrund Lesen Sie hierzu die Hintergrundinformationen zur Funktion UNREGER.KURS().

Die Funktion UNREGER.REND() ermittelt den Wert der Rendite so, dass sich ein gewünschter (bzw. am Markt zu zahlender) Kurs einstellt und informiert den Käufer so über die zu erwartende Effektivverzinsung der Anlage.

In der Beispieldatei zum Buch finden Sie ein Beispiel, welches einen Rechenweg für eine fiktive Anleihe mit verkürzter erster Zinsperiode aufbaut, der ähnlich dem aus Abbildung 14.6 (siehe Seite 668) verläuft. Dabei folgt der Rechenweg der Berechnung des Kurses (wie im Beispiel zu UNREGER.KURS()). Die gesuchte Rendite wird dann mittels Zielwertsuche (eine gedachte Rendite ändert sich so, dass der gewünschte Kurs berechnet wird) ermittelt. UNREGER.REND() liefert das gleiche Ergebnis.

KURS(), RENDITE(), UNREGER.KURS(), UNREGLE.KURS(), UNREGLE.REND()

Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Kursrechnung.xlsx* auf dem Arbeitsblatt *UNREGER.REND*.

Praxiseinsatz

Siehe auch



UNREGLE.KURS() ODDLPRICE()

UNREGLE.KURS(*Abrechnung;Fälligkeit;Letzter_Zinstermin;Zins;Rendite;Rückzahlung;Häufigkeit;Basis*)

Syntax

Diese Funktion berechnet den Kurs eines festverzinslichen Wertpapiers in einer letzten Zinsperiode, deren Länge sich von den regelmäßigen Zinsperioden der Vergangenheit unterscheidet. Zinseszinsseffekte werden nicht berücksichtigt.

Definition

Das erforderliche Argument *Abrechnung* nimmt das Datum, an dem die Anleihe in den Besitz des Käufers übergeht.

Argumente

Das Datum der *Fälligkeit* (erforderlich) ist der Tag, an dem die Rückzahlung des verbrieften Betrags erfolgt.

Das Argument *Letzter_Zinstermin* (erforderlich) ist das Datum der letzten (regelmäßigen) Zinszahlung.

Zins (erforderlich) informiert über die Nominalverzinsung (Jahreszinssatz) der Anleihe.

Das Argument *Rendite* (erforderlich) ist der am Markt gültige Jahreszinssatz für Anleihen der gegebenen Laufzeit.

Rückzahlung (erforderlich) ist der Prozentfuß der Rückzahlung bezogen auf den Nennwert des Wertpapiers (also so, als ob dieser 100 Geldeinheiten beträgt).

Häufigkeit (erforderlich) ist die Anzahl der regelmäßigen Zinszahlungen im Laufe eines Jahrs (jährlich, halbjährlich, vierteljährlich).

Das optionale Argument *Basis* entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.9. Wird es weggelassen, rechnet Excel mit *Basis = 0*.

Folgendes ist zu beachten:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, gibt *UNREGLE.KURS()* den Fehlerwert *#ZAHL!* zurück
- ▶ Für *Zins* und *Rendite* werden nicht negative Zahlen verlangt. Andernfalls gibt *UNREGLE.KURS()* den Fehlerwert *#ZAHL!* zurück.

- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt *UNREGLE.KURS()* den Fehlerwert *#ZAHL!* zurück
- ▶ Das Gleiche geschieht, wenn die Reihenfolge *Fälligkeit* größer *Abrechnung* größer *Letzter_Zinstermin* nicht eingehalten wird

Wichtig Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Hintergrund Um das finanzmathematische Äquivalenzprinzip

Leistung des Gläubigers = Leistung des Schuldners

bezogen auf den »Startpunkt« des Geschäfts umzusetzen, ist der Kurs eines festverzinslichen Wertpapiers (Anleihe) plus eventuell zu zahlende Stückzinsen gleich dem Barwert der zukünftigen in diesem Wertpapier verbrieften Leistungen des Emittenten (Schuldners). Der Kurs wird als Prozentfuß bezogen auf den Nennwert des Wertpapiers angegeben, also so, als ob der Nennwert 100 Geldeinheiten beträgt.

Die Barwertberechnung ist dann ohne Probleme, wenn das Kaufdatum (Besitzwechsel) eines mit Jahreskupon versehenen Wertpapiers mit dem Tag der Zinszahlung einhergeht. In diesem Fall gibt es nur ganze Jahre zu berücksichtigen. Nicht so einfach ist die Situation, wenn der Besitzwechsel zwischen Zinsterminen erfolgt bzw. Zinszahlungen mehrfach im Jahr vereinbart sind. Die Finanzmathematik kennt verschiedene Arten des Umgangs mit gebrochenen Jahresanteilen. Die bekanntesten sind die Methoden Moosmüller und Braess/Fangmeyer sowie die ISMA-Methode. ISMA steht für International Securities Market Association, eine Organisation, die aus der Association of International Bond Dealers (AIBD) hervorgegangen ist.

Über den Zusammenhang zwischen der ISMA-Methode und den Berechnungen der Excel-Funktionen lesen Sie bitte in den Hintergrundinformationen zu *KURS()* und *RENDITE()*.

Im vorliegenden Fall lässt sich das durch Excel angewendete Prinzip wegen der einfachen Verzinsung (keine Zinseszinsen) bei vorgegebenem Marktzins (Jahresrendite) so formulieren:

$$\text{Kurs} + \text{Stückzinsen} = \frac{\text{Nennwert} + \text{Zinsen bei Fälligkeit}}{1 + \text{anteilige Rendite}}$$

wobei sich die Stückzinsen aus den anteiligen Zinsen seit dem letzten Zinstermin und die anteilige Rendite sich aus den Tagen bis zur Fälligkeit (bezogen auf die Gesamtstage im Jahr) ergeben. Außerhalb des letzten Zeitraums bis zur Fälligkeit findet die Funktion keine Anwendung.

Praxiseinsatz In der Beispieldatei zum Buch finden Sie ein Beispiel, welches einen Rechenweg für eine fiktive Anleihe mit der geforderten Ausstattung aufbaut, der ähnlich dem aus Abbildung 14.2 (siehe Seite 652) verläuft. Das Ergebnis ist das gleiche, wie es auch durch Anwendung der Funktion *UNREGLE.KURS()* geliefert wird.

Siehe auch *UNREGLE.REND()*, *KURSDISAGIO()*, *RENDITEDIS()*, *UNREGER.KURS()*, *UNREGER.REND()*, *RENDITE()*, *KURS()*



Dieses Beispiel finden Sie im Ordner *\Ms5-235\Kap14* in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* auf dem Arbeitsblatt *UNREGLE.KURS*.

UNREGLE.REND()



ODDLYIELD()

UNREGLE.REND(*Abrechnung*; *Fälligkeit*; *Letzter_Zinstermin*; *Zins*; *Kurs*; *Rückzahlung*; *Häufigkeit*; *Basis*)

Syntax

Diese Funktion berechnet die Rendite eines festverzinslichen Wertpapiers in einer letzten Zinsperiode, deren Länge sich von den regelmäßigen Zinsperioden der Vergangenheit unterscheidet. Zinseszinsseffekte werden nicht berücksichtigt.

Definition

Abrechnung (erforderlich) nimmt das Datum, an dem die Anleihe in den Besitz des Käufers übergeht.

Argumente

Fälligkeit (erforderlich) ist der Tag, an dem die Rückzahlung des verbrieften Betrags erfolgt.

Letzter_Zinstermin (erforderlich) ist das Datum der letzten (regelmäßigen) Zinszahlung vor dem Kauftermin.

Zins (erforderlich) informiert über die Nominalverzinsung (Jahreszinssatz) der Anleihe.

Kurs (erforderlich) ist der Kurs des Wertpapiers am Abrechnungstag, angegeben als Prozentfuß auf den Nennwert des Wertpapiers (also so, als ob dieser 100 Geldeinheiten beträgt).

Rückzahlung (erforderlich) ist der Prozentfuß der Rückzahlung bezogen auf den Nennwert des Wertpapiers (also so, als ob dieser 100 Geldeinheiten beträgt).

Häufigkeit (erforderlich) ist die Anzahl der regelmäßigen Zinszahlungen im Laufe eines Jahrs (jährlich, halbjährlich, vierteljährlich).

Das optionale Argument *Basis* entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.9. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

Folgendes ist zu beachten:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, gibt UNREGLE.REND() den Fehlerwert #ZAHL! zurück
- ▶ Für *Zins* und *Kurs* werden nicht negative Zahlen verlangt. Andernfalls gibt UNREGLE.REND() den Fehlerwert #ZAHL! zurück
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt UNREGLE.REND() den Fehlerwert #ZAHL! zurück
- ▶ Das Gleiche geschieht, wenn die Reihenfolge *Fälligkeit* größer *Abrechnung* größer *Letzter_Zinstermin* nicht eingehalten wird

Die Excel-Hilfe formuliert folgenden Hinweis:

Wichtig

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Lesen Sie hierzu die Hintergrundinformationen zur Funktion UNREGLE.KURS().

Hintergrund

Die Funktion UNREGLE.REND() ermittelt den Wert der Rendite so, dass sich ein gewünschter (bzw. am Markt zu zahlender) Kurs einstellt, und informiert den Käufer so über die zu erwartende Effektivverzinsung der Anlage. Wegen des verwendeten Prinzips der einfachen

Verzinsung (keine Zinseszinsseffekte) lässt sich die Kursformel nach der Rendite auflösen. Die Formel der Excel-Hilfe liest sich »in Worten« so:

$$\text{Rendite} = \frac{(\text{Rückzahlung} + \text{Zinsen bei Fälligkeit}) - (\text{Kurs} + \text{Stückzinsen})}{\text{Kurs} + \text{Stückzinsen}} \cdot \frac{1}{\text{Periodenanteil}}$$

Der Periodenanteil bezogen auf ein Jahr führt dazu, dass *Rendite* ein Jahreszinssatz ist. Die Zinsen bei Fälligkeit ergeben sich aus den Zinsen seit dem letzten Zinstermin, die Stückzinsen berechnen sich anteilig bezogen auf den Zeitpunkt der letzten Zinszahlung bis zum Kauf.

Praxiseinsatz In der Beispieldatei zum Buch finden Sie ein Beispiel, welches einen Rechenweg für eine fiktive Anleihe mit veränderter letzter Zinsperiode aufbaut, der ähnlich dem aus Abbildung 14.6 (siehe Seite 668) verläuft. Dabei folgt der Rechenweg der Berechnung des Kurses (wie im Beispiel zu UNREGLE.KURS()). Die gesuchte Rendite wird dann mittels Zielwertsuche (eine gedachte Rendite ändert sich so, dass der gewünschte Kurs berechnet wird) ermittelt. UNREGLE.REND() liefert das gleiche Ergebnis.

Siehe auch UNREGLE.KURS(), KURSDISAGIO(), RENDITEDIS(), UNREGER.KURS(), UNREGER.REND(), RENDITE(), KURS()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* auf dem Arbeitsblatt *UNREGLE.REND*.

VDB() VDB()

Syntax VDB(*Ansch_Wert*; *Restwert*; *Nutzungsdauer*; *Anfang*; *Ende*; *Faktor*; *Nicht_wechseln*)

Definition Diese Funktion berechnet Abschreibungsbeträge nach der geometrisch-degressiven Abschreibung. Optional kann ein (durch steuerrechtliche Regelungen vorgeschlagener) Wechsel zur linearen Abschreibung ausgewertet werden.

Argumente *Ansch_Wert* (erforderlich) verlangt die Anschaffungskosten (Nettokaufpreis plus Anschaffungsnebenkosten minus Anschaffungskostenminderungen) eines Wirtschaftsguts. Verwenden Sie keinen als Zahl auswertbaren Wert, erscheint der Fehlerwert #WERT!. Ist die eingegebene Zahl negativ, erfolgt die Fehlermeldung #ZAHL!.

Restwert (erforderlich) entspricht dem Wert des Wirtschaftsguts am Ende der Abschreibungsdauer. Verwenden Sie keine Zahl, erscheint der Fehlerwert #WERT!. Ist die eingegebene Zahl negativ, erfolgt die Fehlermeldung #ZAHL!.

Nutzungsdauer (erforderlich) verlangt die Anzahl der Perioden, in denen das Wirtschaftsgut abgeschrieben wird. Sie sollten für dieses Argument eine ganze Zahl größer 0 (Null) verwenden.

Anfang (erforderlich) ist die Periode innerhalb der *Abschreibungsdauer*, an deren Ende die Berechnung der (eventuell über mehrere Perioden summierten) Abschreibungsbeträge beginnen soll. Die erste Periode hat die Nummer 0 (Null).

Ende (erforderlich) ist die letzte Periode innerhalb der *Abschreibungsdauer*, an deren Ende die Berechnung der (eventuell über mehrere Perioden summierten) Abschreibungsbeträge enden soll.

Faktor (optional) ist der Multiplikator für den (gedachten linearen) Abschreibungssatz (Reziprokes der Abschreibungsdauer). Wird er nicht angegeben, rechnet Excel mit der Zahl 2.

Nicht_wechseln (optional) ist ein Wahrheitswert, der angibt, ob (wegen größerer Abschreibungsbeträge, siehe die Erläuterungen weiter unten) zur linearen Abschreibung gewechselt werden soll. Wenn dieses Argument *FALSCH* ist oder nicht angegeben wird, wird der (eventuell summierte) Abschreibungsbetrag durch den Wechsel zur linearen Abschreibung ermittelt.

Abschreibung hat den Sinn, den Werteverlust eines Wirtschaftsguts zu ermitteln und sichtbar zu machen. Davon zu unterscheiden ist der Begriff der Absetzung für Abnutzung, der die Kosten für den Erwerb eines Wirtschaftsguts als Betriebsausgaben steuerlich erfasst.

Der Abschreibungsprozentsatz, den die Funktion VDB() nutzt, bestimmt sich aus dem Reziproken der Abschreibungsdauer multipliziert mit einem Faktor. Wird dieser nicht angegeben, wird mit der Zahl 2 gerechnet. Der Buchwert am Anfang der Abschreibungsperiode wird mit dem Abschreibungsprozentsatz multipliziert. Der entstehende Betrag mindert den Buchwert zum Ende der Periode.

Optional prüft die Funktion, ob ein solcher Abschreibungsbetrag kleiner ist als der, der durch Wechsel zu einer linearen Abschreibung (auf den ursprünglichen Restwert) entstehen würde und wechselt im »Ja-Fall« zur linearen Abschreibung.

Diese Form der Abschreibungsmethode (einschließlich des Wechsels zur linearen Abschreibung) ist zum Zeitpunkt der Niederschrift dieses Kapitels im deutschen Steuerrecht bis zum Steuerjahr 2007 zulässig, wobei an den anfänglichen Abschreibungsprozentsatz Forderungen im Einkommensteuergesetz gemacht werden. Er darf nicht mehr als das Doppelte des linearen Satzes und nicht mehr als 20 % betragen. Diese Forderungen können mithilfe des *Faktors* u.U. reguliert werden.

Da aber ein eventueller gebrochener Jahresanteil der ersten Periode gesondert zu berücksichtigen ist, ist die Funktion VDB() im Allgemeinen ohne Zusatzrechnungen »per Hand« nicht einsetzbar. Ein Aufschreiben des Abschreibungsplanes unter Nutzung der Prozentrechnung und arithmetischer Verknüpfung einzelner Zellen sollte da eher zum Ziele führen.

Ein Wirtschaftsgut mit Anschaffungskosten von 1.000,00 € soll in 10 Jahren geometrisch-degressiv auf einen Restwert von 100,00 € abgeschrieben werden. Sie können hierzu in jeder Periode den Abschreibungsbetrag mittels VDB() berechnen und vom Buchwert der Vorperiode abziehen.

Alternativ können Sie aber auch einen Abschreibungsplan erstellen, der die oben genannten Formeln zur Berechnung des ersten Abschreibungsbetrags sowie die Vorgehensweise zur Berechnung der anderen Beträge umsetzt. Die Erstellung eines Abschreibungsplans nach eigenem Muster ist vor allem bei der Anwendung steuerlich zulässiger Methoden (Zwölfstel-Regelung seit 2004, wechselndes Steuerrecht in den Jahren 2009/2010 hinsichtlich degressiver Abschreibungen und neuer Wechsel 2011) zu empfehlen, da die integrierten Funktionen im Allgemeinen nicht anwendbar sind.

Einzelheiten zu diesem Beispiel entnehmen Sie bitte der zugehörigen Arbeitsmappe.

DIA(), GDA(), GDA2(), LIA()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Abschreibungsrechnung.xlsx* auf dem Arbeitsblatt *VDB*.

Hintergrund

Praxiseinsatz

Siehe auch



XINTZINSFUSS()



Syntax XINTZINSFUSS(*Werte*; *Zeitpunkte*; *Schätzwert*)

Definition Diese Funktion liefert den internen Zinssatz (und nicht, wie der Name suggeriert, den Zinsfuß) einer Reihe nicht notwendig periodisch anfallender Zahlungen im unterjährlichen Bereich.

Argumente *Werte* (erforderlich) sind die lückenlos in einer Spalte angeordneten (tatsächlichen und/oder erwarteten) Überschüsse aus Aus- und Einzahlungen. Jeder Wert vertritt einen Zeitpunkt. Negative Überschüsse werden mit einem Minuszeichen versehen.

Zeitpunkte (erforderlich) sind die Termine, für die die Überschüsse zutreffen. Die Anordnung muss mit der der Zahlungen im Argument *Werte* übereinstimmen. Der erste Termin gibt den Startpunkt der Bewertung, alle anderen müssen (nicht unbedingt selbst geordnet) nach diesem Termin liegen (der Fehlerwert #ZAH! zeigt eine mögliche Nichtbeachtung dieser Forderung)¹¹.

Da eine Berechnung des internen Zinssatzes aus mathematischen Gründen für mehr als zwei Perioden nur näherungsweise erfolgen kann, kann diese Näherungsrechnung zu verschiedenen, teilweise auch unsinnigen Ergebnissen (Zinssatz ist negativ) führen. Sie können die Berechnung durch das optionale Argument *Schätzwert* als Ausgangspunkt für die Näherungsrechnung beeinflussen.

Beachten Sie weiterhin:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten
- ▶ Ungültige Datumsangaben werden mit dem Fehlerwert #WERT! quittiert
- ▶ XINTZINSFUSS() erwartet mindestens eine positive und mindestens eine negative Zahlung, andernfalls gibt die Funktion den Fehlerwert #ZAH! zurück. Die erste Zahlung selbst muss nicht negativ sein.
- ▶ Umfassen *Werte* und *Zeitpunkte* unterschiedlich viele Werte, gibt XINTZINSFUSS() ebenfalls den Fehlerwert #ZAH! zurück

Hintergrund Obwohl »klassische« Investitionsrechnung mit Jahresperioden rechnet, gibt es eine Reihe von Aufgaben, die im unterjährlichen Bereich zu bearbeiten sind. Dazu zählt vor allem die Bewertung kurzfristiger Finanzgeschäfte.

Die Berechnungen hier gehen von der Barwertformel für Zahlungen aus, die bei gebrochenen Jahresperioden nicht mit einem gebrochenen Zinssatz, sondern mit gebrochener Laufzeit rechnet. Basis bleibt trotz allem ein ganzes Jahr, was vermutlich seine Ursache darin hat, dass ein »Tageszinssatz« (oder noch kleinere Perioden) mit Nullen nach dem Komma nicht die notwendige Vorstellungskraft bei der Interpretation hervorruft. Der Barwert (auch Kapitalwert genannt) berechnet sich aus

$$\text{Barwert} = \sum_{k=1}^{\text{Periodenanzahl}} \frac{\text{Zahlungen zum Zeitpunkt}_k}{(1 + \text{Jahreszinssatz})^{(\text{Zeitpunkt}_k - \text{Zeitpunkt}_1)/365}}$$

Der Jahreszinssatz, der den Barwert zu 0 (Null) macht, heißt interner Zinssatz (Rendite oder Effektivzinssatz).

¹¹ Vor Excel 2013 heißt dieses Argument *Zeitpunkte*.

Ratenkreditgeschäfte (Konsumentenkredite) verlangen Zahlungen in regelmäßigen Abständen im unterjährlichen Bereich. XINTZINSFUSS() kann solche Zahlungen verarbeiten, dennoch ist es keine gute Funktion, um die Forderungen der in Deutschland geltenden Preisangabenverordnung von 2002 zur Angabe von Effektivzinssätzen zu erfüllen. Anders als die Verordnung von 1985, die die Nachbildung von Krediten auf einem Konto analog zum Sparbuch forderte, verlangt die Verordnung aus dem Jahr 2002 die Bewertung aller Zahlungen als Barwerte im Sinne der obigen Formel (Abzinsen durch gebrochene Laufzeiten, nicht durch gebrochene Zinssätze). Jedoch stimmt die Tageszählung nicht: Die Preisangabenverordnung setzt in diesem Fall einen Monat mit $365/12=30,4166\dots$ Tagen an.

Die zur Berechnung verwendeten Werte können nicht »beliebig« gewählt werden. Negative Renditen deuten darauf hin, dass mehr Geld in eine Richtung geflossen ist, aus der es später nicht zurückkommt. Das kann die Ursache sowohl in »falschen« Beträgen als auch »falschen« Zeitpunkten haben. Investiert jemand 100 Geldeinheiten und bekommt später dreimal 33 zurück, kann das nicht in Ordnung sein. Anders aber, wenn er erst dreimal 33 investiert, um dann später 100 zu erhalten. Zahlenreihen mit solchen Effekten lassen sich beliebig erzeugen und »verschachteln«.

Hinweis

Die Gleichung der obigen Formel gestattet eine Auflösung nur in wenigen Sonderfällen. Excel muss also mit einem mathematischen Näherungsverfahren arbeiten. Dieses bewegt sich, eine gute »Startposition« vorausgesetzt, Schritt für Schritt auf die Lösung – falls es eine solche überhaupt gibt – zu. Sie können diese Iteration durch die Mitgabe eines geeigneten Schätzwerts für den internen Zinssatz unterstützen, die Berechnung einer Lösung aber nicht erzwingen.

In Situationen, die einem »normalen« Alltag bei der Bewertung von Finanzinvestitionen entspricht, sollte es kaum zu Problemen kommen. Die Bewertung von Sachinvestitionen empfiehlt sich wegen deren »Ungewissheit der Überschüsse« ohnehin nicht in gebrochenen Jahren.

Ein Händler wirbt mit dem Finanzkauf (= Ratenkredit oder Konsumentenkredit) einer Waschmaschine im Wert von 599,00 €. Er verlangt dafür 12 Monate lang eine Zahlung von 52,48 €.

**Praxiseinsatz
Konsumenten-
kredit**

Wie hoch ist die Effektivverzinsung?

Sie tragen dazu in einer Spalte 13 Termine im Abstand von einem Monat ein. Neben dem ersten notieren Sie den »Barpreis« mit positivem Vorzeichen, darunter die Monatsraten – mit negativen Vorzeichen.

Die Anwendung von XINTZINSFUSS() liefert dann 9,80 %. Das ist allerdings nicht die Zahl, die der Händler nach der Preisangabenverordnung von 2002 angibt. In der Arbeitsmappe der Beispieldatei zum Buch finden Sie die Berechnung, die die korrekte Monatslänge verwendet und durch Zielwertsuche den Barwert aus der obigen Formel zu 0 (Null) macht. Es sind 9,76 %.

Versicherungsgesellschaften bieten Kunden die Möglichkeit, ihre Prämien mit einem Aufschlag in Raten zu zahlen¹². Gewöhnlich wird dabei ein Aufschlag von 3 % bei halbjährlicher, 5 % bei vierteljährlicher und 8 % bei monatlicher Zahlung fällig.

**Versicherungs-
prämie auf
Raten**

Zu welchem Effektivzinssatz muss der Kunde den so eingeräumten Kredit bezahlen?

¹² Inzwischen gibt es auch Angebote, die statt mit Aufschlägen mit Abschlägen (Skonto) arbeiten. Solche Verträge lassen sich analog bewerten.

Eine Möglichkeit besteht in der Bewertung mittels der obigen Barwertformel. Andere Möglichkeiten bestehen in der Nachbildung auf einem Sparbuch oder eine Bewertung nach Preisangabenverordnung 2002, die hier allerdings nicht greift, einen Effektivzins muss die Versicherungsgesellschaft wohl nicht nennen.

Im Falle einer gesamt-fälligen (jährlichen) Prämie von 1.000,00 € stehen dieser bei vierteljährlicher Zahlungsweise alternativ vier Zahlungen zu je 262,50 € gegenüber. Der Versicherungsnehmer kann das zunächst »gesparte« Geld von 737,50 € zur Bank bringen, um alle drei Monate von dort 262,50 € abzuheben. XINTZINSFUSS() informiert darüber, zu welchem (jährlichen) Zinssatz die Bank das Guthaben verzinsen müsste (Zinszahlungen alle drei Monate), damit das Konto am Ende nicht überzogen werden muss und die letzte Rate zur Verfügung steht. Das sind aber »stolze« 14,2 %. Der Versicherungsnehmer kann sich deshalb durchaus überlegen, die Gesamtprämie unter Nutzung eines Diskontkredits sofort zu zahlen.

Siehe auch XKAPITALWERT(), IKV(), QIKV(), ZINS()



Dies Beispiele finden Sie im Ordner `Ms5-235\Kap14` in der Arbeitsmappe *Investitionsrechnung.xlsx* auf dem Arbeitsblatt *XINTZINSFUSS*.

XKAPITALWERT() XNPV()

Syntax XKAPITALWERT(*Zins*; *Werte*; *Zeitpunkte*)

Definition Diese Funktion gibt den Kapitalwert einer Reihe nicht notwendig periodisch anfallender Überschüsse (als Differenz zwischen Ein- und Auszahlungen) im unterjährlichen Bereich zurück.

Argumente *Zins* (erforderlich) ist der Jahreszinssatz, der in die Berechnung eingeht. Es kann sich dabei um einen Guthabenzinssatz, einen Zinssatz auf einem Kontokorrentkonto oder einen Marktzinssatz handeln, der die Ausgangssituation am besten beschreibt.

Werte (erforderlich) sind die lückenlos in einer Spalte angeordneten (tatsächlichen und/oder erwarteten) Überschüsse aus Aus- und Einzahlungen. Jeder Wert vertritt einen Zeitpunkt. Negative Überschüsse werden mit einem Minuszeichen versehen.

Zeitpunkte (erforderlich) sind die Termine, für die die Überschüsse zutreffen. Die Anordnung muss mit der der Zahlungen im Argument *Werte* übereinstimmen. Der erste Termin gibt den Startpunkt der Bewertung, alle anderen müssen (nicht unbedingt selbst geordnet) nach diesem Termin liegen (der Fehlerwert *#ZAHL!* zeigt eine mögliche Nichtbeachtung dieser Forderung)¹³.

Beachten Sie außerdem:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten
- ▶ Ungültige Datumsangaben werden mit dem Fehlerwert *#WERT!* quittiert
- ▶ XKAPITALWERT() erwartet mindestens eine positive und mindestens eine negative Zahlung, andernfalls gibt die Funktion den Fehlerwert *#ZAHL!* zurück. Die erste Zahlung selbst muss nicht negativ sein.
- ▶ Umfassen *Werte* und *Zeitpunkte* unterschiedlich viele Werte, gibt XKAPITALWERT() ebenfalls den Fehlerwert *#ZAHL!* zurück

¹³ Vor Excel 2013 heißt dieses Argument *Zeitpunkte*.

Lesen Sie bitte hierzu die Informationen zur Funktion XINTZINSFUSS(). Der in der dortigen Formel ermittelte Barwert ist gerade der Kapitalwert, den XKAPITALWERT() bei gegebener Verzinsung errechnet.

Ein Händler bietet seinen Kunden an, Rechnungen später bezahlen zu können oder bei Sofortzahlung einen Skontobetrag abzuziehen. Die Möglichkeit des Späterzahlens stellt also eine Form des »Geldverleihs« dar.

Angenommen, es gibt drei Rechnungen mit den Daten aus Tabelle 14.10.

Datum	Betrag	Skonto	Zahlungsziel
2.5.2013	700,00 €	0,5 %	14 Tage
3.8.2013	300,00 €	1 %	4 Wochen
7.11.2013	250,00 €	2 %	2 Monate

Hintergrund

Praxiseinsatz

Tabelle 14.10
Fiktive Rechnungsbeträge mit Skontobedingungen

Ist das Gewähren von Kredit in dieser Form finanzmathematisch sinnvoll, wenn es Anlageformen gibt, die sich mit 10 % im Jahr rentieren?

Die finanzmathematische Lösung (diese beachtet weder Fragen der Kundenbeziehungen noch solche der Liquidität oder Bonität) bringt der Einsatz von XKAPITALWERT() wie in Abbildung 14.7.

	A	B	C	D	E	F
1						
2		Zinssatz	10,00%			
3						
4		Datum	Betrag	Skonto	Frist (Tage)	Konto
5		02.05.2013	700,00 €	0,50%	14	696,50 €
6		16.05.2013				700,00 €
7		03.08.2013	300,00 €	1,00%	28	297,00 €
8		31.08.2013				300,00 €
9		07.11.2013	250,00 €	2,00%	60	245,00 €
10		06.01.2014				250,00 €
11						
12		XKAPITALWERT	2,80			
13						
14		Rendite	13,58%			

Abbildung 14.7: Kreditvergabe durch Skonti

Angenommen, der Kunde schöpft die Skantomöglichkeit nicht aus. Dann ist das so, als ob mit Lieferung der Händler das Geld zu einer fiktiven Bank bringt und dort mit Zahlungseingang wieder holt. Eine Bewertung der Bank nach dem hier verwendeten Barwertprinzip zeigt einen positiven Kapitalwert. Der Händler hat also mehr Zinseffekte als 10 % pro Jahr.

Der Einsatz von XINTZINSFUSS() beziffert den Effektivzins auf 13,58 % im Jahr. Einen solchen Eindruck bekommt man auch durch Überschlagsrechnung, indem man die Skontosätze (die im Prinzip vorschüssige Zinssätze sind) im Kopf hochrechnet: 1 % für einen Monat bringt 12 % fürs ganze Jahr.

Siehe auch XINTZINSFUSS(), NBW(), BW()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Investitionsrechnung.xlsx* auf dem Arbeitsblatt *XKAPITALWERT*.

ZINS() RATE()

Syntax ZINS(*Zzr*; *Rmz*; *Bw*; *Zw*; *F*; *Schätzwert*)

Definition Diese Funktion ermittelt den zutreffenden Zinssatz bei Aufgaben der Zinseszinsrechnung sowie der Rentenrechnung (auch Tilgungsrechnung nach dem Prinzip der Annuitätentilgung).

Argumente *Zzr* (erforderlich) legt die Anzahl der Zinsperioden fest. Dabei wird davon ausgegangen, dass eventuelle regelmäßige Zahlungen (Argument *Rmz* ist größer Null) am Ende (oder Anfang) der Zinsperioden erfolgen.

Rmz (erforderlich/optional, siehe Hinweis) informiert über die Höhe der regelmäßigen Zahlung und kann als Rente interpretiert werden.

Bw (erforderliche/optional, siehe Hinweis) ist der Anfangswert einer Richtung der Zahlungen. Für Auszahlpläne handelt es sich um den Kontostand am Anfang der Betrachtungen, bei Tilgungsplänen um die Kreditsumme.

Zw (optional/erforderlich, siehe Hinweis) ist der Zukunftswert, der am Ende des betrachteten Zeitraums vorliegen soll.

F (optional) legt fest, ob die regelmäßigen Zahlungen am Ende der Perioden ($F = 0$ oder nicht angegeben) oder aber am Anfang der Perioden ($F = 1$) erfolgen.

Schätzwert (optional) kann angegeben werden, um die notwendige Näherungsrechnung (siehe Hintergrundinformationen) in Gang zu bringen. Wenn Sie keinen Wert für *Schätzwert* angeben, rechnet Excel mit einem Wert von 10 %.

Hinweis Jeweils zwei der drei Größen *Rmz*, *Bw* und *Zw* müssen angegeben werden und dabei verschieden von 0 (Null) sein. Die Vorzeichen richten sich nach der »Richtung des Geldflusses« (vgl. die unten stehende Formel).

Wichtig Das genannte finanzmathematische Äquivalenzprinzip bedeutet, dass hinsichtlich des Vorzeichens zwischen Aus- und Einzahlungen (Kreditaufnahme und Tilgung, Investition und Desinvestition) unterschieden werden muss. Hier weicht Excel vom in der finanzmathematischen Literatur gebräuchlichen Äquivalenzprinzip

Leistung des Schuldners = Leistung des Gläubigers
ab.

Hintergrund Die fünf Funktionen BW() = Barwert, ZW() = Zukunftswert, RMZ() = regelmäßige Zahlung, ZZR() = Zins- oder Zahlungszeiträume und ZINS() = Zinssatz, stehen in Umsetzung des obigen Äquivalenzprinzips in folgendem Zusammenhang:

$$BW \cdot (1 + ZINS)^{ZZR} + RMZ \cdot (1 + F \cdot ZINS) \cdot \frac{(1 + ZINS)^{ZZR} - 1}{ZINS} + ZW = 0$$

Der Barwert wird bis zum Ende aufgezinst, ebenso die regelmäßigen Zahlungen. Zum Ende wird die Summe mit dem Zukunftswert verglichen.

Die Nutzung einer dieser Funktionen ist gleichbedeutend mit der jeweiligen finanzmathematischen Grundaufgabe: Berechnung eines Werts aus obiger Gleichung, wenn die anderen Werte bekannt sind.

Die Funktionen stellen also die Gleichung nach der gleichnamigen Größe um (außer bei ZINS – hier wird eine Näherungsrechnung durchgeführt).

Die Rechnung als Näherungsrechnung wird durch Excel vom genannten Schätzwert aus gestartet und sollte Schritt für Schritt an die Lösung heranführen, wenn es denn eine solche überhaupt gibt. Durch Variation des Schätzwerts können Sie Excel bei der Lösungssuche unterstützen. Die Lösbarkeit der Gleichung lässt sich dadurch aber nicht erzwingen.

Gewöhnlich erfolgen die Angaben des Zinssatzes als Jahreszinssatz. Die genannten Funktionen arbeiten nur dann korrekt, wenn die Perioden zum Zinssatz passen. Bei unterjährlicher Verzinsung wird in der Regel der Jahreszinssatz gleichmäßig auf die Perioden aufgeteilt: 12 Monate zu einem Zwölftel des Zinssatzes, 3 Monate zu einem Viertel und ein halbes Jahr zur Hälfte.

Hinweis

Die folgenden Beispiele orientieren sich in einigen Überschriften am »finanzmathematischen Sprachgebrauch«.

Praxiseinsatz

Jemand möchte im Alter ein kleines finanzielles Polster haben und beschließt, eine gerade angefallene Erbschaft in Höhe von 10.000,00 € auf 15 Jahre fest anzulegen. Er hofft auf eine Auszahlung von wenigstens 25.000,00 €. Bei welchem Zinssatz gelingt der Plan?

Zinseszinsrechnung

Eine Berechnung nach

=ZINS(15;;-10000;25000)

liefert einen erforderlichen Jahreszinssatz von 6,3 %.

Bundesschatzbriefe vom Typ B sind solche, bei denen gezahlte Zinsen zu den vereinbarten Konditionen wieder angelegt werden. Im Beispiel zur Funktion ZW2() finden Sie eine Rechnung, die ZINS() benutzt, um die durch die Bundesbank veröffentlichten Renditen zu bestätigen.

Bundesschatzbriefe Typ B

Jemand hat bis zum 60. Lebensjahr 100.000,00 € angespart und möchte sich eine zusätzliche Rente in Höhe von 750,00 € monatlich auszahlen lassen. Zu welchem Zinssatz muss das jeweils noch vorhandene Guthaben verzinst werden, wenn 15 Jahre lang Zahlungen am Monatsanfang geplant sind?

Rentenrechnung

Sie ermitteln den erforderlichen Monatszinssatz der Rente aus

=ZINS(15*12;-750;100000;;1)

zu 0,354796 % (die 1 in der Formel zielt auf den Monatsanfang der Zahlungen). Die Rechnung geht von Zinseszinsen innerhalb eines Jahrs aus. Auf einem Sparguthaben lässt sich der Zinseszinsseffekt allerdings nicht umsetzen, sodass das dortige Ergebnis etwas »theoretisch« ist. Außerdem ist bei der Angabe des (12-fachen) Nominalzinsses für ein ganzes Jahr dieser auf zwei Stellen nach dem Komma beschränkt: 4,26 % p.a. Ein konkretes Konto muss dann mit einem Zwölftel dieser Zahl geführt werden.

Jemand kann monatlich 1.000,00 € zur (monatlich konstanten) Rückzahlung (Tilgung plus Zinsen) eines Kredits einsetzen. Zu welchem Zinssatz müsste dieser verzinst werden, wenn 175.000,00 € im Laufe von 30 Jahren zurückgezahlt werden sollen?

Tilgungsrechnung (Annuitätentilgung)

Tilgungsrechnung in diesem Sinne ist Rentenrechnung. Der Einsatz von ZINS() führt über
 $=\text{ZINS}(30*12; -1000; 175000)$

auf 0,46316 %. Anders als beim Sparbuch erfolgt beim Hypothekendarlehen eine monatliche Verzinsung zu einem Zwölftel des vereinbarten Jahreszinssatzes (Nominalzins). Allerdings ist der durch ZINS() berechnete Zinssatz noch durch Multiplikation mit 12 auf ein Jahr hochzurechnen und das Konto mit dem daraus resultierenden Nominalzins von 5,56 % p.a. zu führen. Zur monatlichen Berechnung der Zinsen wird dann ein Zwölftel dieses Zinssatzes angewendet. Dies führt zu einer verbleibenden Restschuld nach 30 Jahren.

Hinweis Mit ZINS() können Sie den durch die Preisangabenverordnung des Jahrs 2000 notwendig anzugebenden effektiven Jahreszinssatz für Hypothekendarlehen nur näherungsweise berechnen. Das liegt daran, dass ZINS() einen gebrochenen Zinssatz bei unterjährlichen Zinsperioden ermittelt, die Preisangabenverordnung aber die Nutzung gebrochener Jahresperioden verlangt.

Wichtig Die Verwendung von Formeln unterscheidet sich von der ausführlichen Notierung eines Kontoverlaufs mit den gegebenen Ein- und Auszahlungen möglicherweise durch Rundungsfehler. Das liegt daran, dass auf einem tatsächlich geführten Konto Rundungen auf zwei Stellen nach dem Komma erfolgen. Gestalten Sie ein solches Konto unter Excel nach, ist in den Zwischenschritten, die konkretes Geld bedeuten, jeweils die Funktion RUNDEN() einzusetzen. Eine ledigliche Begrenzung der Anzeige des Zellwerts auf zwei Stellen ist oft nicht korrekt.

Die Beispiele in der Beispieldatei zum Buch lassen Spielraum zum Experimentieren mit den Rundungseffekten, da die Kontopläne bereits vorbereitet sind.

Investitionsrechnung Sind bei Investitionsbewertungen die jährlichen zukünftigen Überschüsse konstant, können Sie statt der Funktion IKV() auch die Funktion ZINS() zur Bestimmung der internen Kapitalverzinsung verwenden. Details zur Methode des internen Zinssatzes finden Sie bei den Erläuterungen zu IKV().

Siehe auch BW(), IKV(), PDURATION(), RMZ(), ZW(), ZZR(), ZSATZINVEST()



Die Arbeitsmappen *Zinseszinsrechnung.xlsx*, *Rentenrechnung.xlsx* und *Tilgungsrechnung.xlsx* im Ordner `\Ms5-235\Kap14` halten die Beispiele mit entsprechender Überschrift auf dem jeweiligen Arbeitsblatt *ZINS* bereit.

ZINSSATZ() INTRATE()

Syntax ZINSSATZ(*Abrechnung*; *Fälligkeit*; *Anlage*; *Rückzahlung*; *Basis*)

Definition Diese Funktion berechnet den äquivalenten nachschüssigen Zinssatz für ein Wertpapier, welches mit einem Abschlag (Diskont, Disagio) für eine in der Regel unterjährliche Laufzeit ausgestattet wurde.

Sie unterscheidet sich von der Funktion RENDITEDIS() nur durch die Benennung der Argumente.

Argumente *Abrechnung* (erforderlich) ist der Tag, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) erfasst den Tag, an dem die Leistungen aus dem Wertpapier erfolgen müssen.

Anlage (erforderlich) ist der Kaufpreis des Wertpapiers am Abrechnungstag.

Rückzahlung (erforderlich) beschreibt die Leistung am Fälligkeitstag.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.11. Wird es weggelassen, rechnet Excel mit $Basis = 0$.

Basis	Methode	Bedeutung
0	30/360 (NASD-Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Ist das Ausgangsdatum der 31. eines Monats, wird dieses Datum zum 30. desselben Monats. Ist das Enddatum der 31. eines Monats und das Ausgangsdatum ein Datum vor dem 30. eines Monats, wird das Enddatum zum 1. des darauf folgenden Monats. In allen anderen Fällen wird das Enddatum zum 30. desselben Monats.
1	Genau/Genau	Jeder Monat hat so viele Tage wie im Kalender stehen, die Summe über alle Monate ergibt die Anzahl der Tage im Jahr
2	genau/360 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 360 Tage.
3	Genau/365 Tage	Jeder Monat hat so viele Tage wie im Kalender stehen. Ein Jahr hat immer 365 Tage.
4	30/360 (Europäische Methode)	Jeder Monat hat 30 Tage, ein Jahr 12 Monate. Jedes Ausgangs- und Enddatum, das auf den 31. eines Monats fällt, wird zum 30. desselben Monats.

Tabelle 14.11
Verschiedene Methoden zur Ermittlung der Anzahl der Tage eines Jahres

Alle Argumente der Funktion, die ein Datum betreffen, verwenden eine Datumsangabe ohne Uhrzeit, gebrochene Zahlen werden also abgerundet. Das Argument *Basis* verlangt ebenfalls eine ganze Zahl und schneidet Nachkommastellen ab.

Folgendes ist zu beachten:

- ▶ Werden ungültige Datumseinträge verwendet bzw. keine Zahlen dort angegeben, wo sie erforderlich sind, liefert die Funktion den Fehlerwert **#WERT!**
- ▶ Ist *Anlage* kleiner oder gleich 0 oder ist *Rückzahlung* kleiner oder gleich 0, gibt ZINSSATZ() den Fehlerwert **#ZAHL!** zurück
- ▶ Ist *Basis* kleiner 0 oder größer 4, gibt ZINSSATZ() den Fehlerwert **#ZAHL!** zurück
- ▶ Ist *Abrechnung* größer oder gleich *Fälligkeit*, gibt ZINSSATZ() den Fehlerwert **#ZAHL!** zurück

Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Vom Grundsatz her gehört diese Funktion in den Umkreis der vorschüssigen Verzinsung. Das Prinzip besteht in der Annahme, dass sich ein Anfangskapital (einzuzahlender Betrag, Darlehen) aus dem Rückzahlungsbetrag minus die Zinsen auf diesen Betrag ergeben. Dieses Prinzip wird vor allem im unterjährlichen Bereich verwendet und unterscheidet sich von dem Prinzip, welches einem Sparbuch oder einem Hypothekendarlehen zugrunde liegt. Dort werden nämlich die Zinsen am Ende eine Periode auf Basis des Anfangskapitals berechnet (nachsüssige Verzinsung).

Wichtig

Hintergrund

Um Vergleichbarkeit von Wertpapieren mit nachschüssiger Verzinsung herzustellen, wird die Differenz zwischen Rückzahlung und Kurs ins Verhältnis zum Kurs gesetzt. Das Ergebnis entspricht dem relativen Zuwachs bis zur Fälligkeit, also einem nachschüssigen Zinssatz, der hier, wie oft in ähnlichen Situationen, den Namen Rendite erhält. Die Berechnung erfolgt ohne Zinseszinsseffekte, der Zinssatz ist unter diesem Aspekt noch auf ein Jahr (bei entsprechender Tagezählung) hochzurechnen.

Die Funktion `DISAGIO()` steht mit der Funktion `ZINSSATZ()` in folgender Beziehung:

$$(1 + \text{ZINSSATZ}()) \cdot (1 - \text{DISAGIO}()) = 1$$

Diese Beziehung erlaubt es, aus einem vorschüssigen Jahreszinssatz den dazu äquivalenten nachschüssigen Jahreszinssatz auszurechnen. Äquivalenz bedeutet dabei, dass zwei Anleger, die das gleiche Kapital einsetzen – der eine mit vorschüssiger, der andere mit nachschüssiger Verzinsung – am Ende eines Jahrs auch eine gleiche Rückzahlung bekommen.

Praxiseinsatz Als Beispiele für diese Funktion können Sie die Beispiele zu `RENDITEDIS()` nehmen und stattdessen die Funktion `ZINSSATZ()` für die Berechnungen verwenden.

Siehe auch `AUSZAHLUNG()`, `DISAGIO()`, `KURSDISAGIO()`, `RENDITEDIS()`, `TBILLÄQUIV()`



Sie finden die Beispiele im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Einfache Zinsrechnung.xlsx* auf dem Arbeitsblatt *Wechselrechnung*.

ZINSTERMNZ() COUPNCD()

Syntax `ZINSTERMNZ(Abrechnung; Fälligkeit; Häufigkeit; Basis)`

Definition Diese Funktion ermittelt das Datum der ersten Kuponzinszahlung, nachdem ein festverzinsliches Wertpapier mit regelmäßigen Zinsterminen den Besitzer gewechselt hat.

Argumente *Abrechnung* (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum, an dem die Rückzahlung des durch das Wertpapier verbrieften Darlehens erfolgt.

Da es durchaus üblich ist, festverzinsliche Wertpapiere mit mehreren Zinsterminen im Jahr auszustatten, informiert das erforderliche Argument *Häufigkeit* über deren Anzahl. Zulässige Werte sind: 1 = jährlich; 2 = halbjährlich; 4 = vierteljährlich.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.11. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

An die Argumente werden folgende Anforderungen gestellt:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, wird der Fehlerwert `#WERT!` zurückgegeben
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt die Funktion den Fehlerwert `#ZAHL!` zurück. Das Gleiche geschieht, wenn der Abrechnungstermin nach der Fälligkeit liegt.

Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Regelmäßige Kuponzinszahlungen eines festverzinslichen Wertpapiers (Anleihe) richten sich nach dem Fälligkeitsdatum und der Häufigkeit innerhalb eines Jahrs.

Unterschiedliche Ergebnisse im unterjährlichen Bereich können dabei durch unterschiedliche Tagezählung im Jahr entstehen.

Diese Funktion wurde in den ausführlichen Rechnungen der Beispiele zu KURS() und RENDITE() verwendet.

ZINSTERMTAGE(), ZINSTERMTAGNZ(), ZINSTERMTAGVA(), ZINSTERMVZ(), ZINSTERMZAHL(), KURS(), RENDITE(), TAGE360()

Die genannten Beispiele finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Kursrechnung.xlsx* auf den Arbeitsblättern *KURS* und *RENDITE*.

Wichtig

Hintergrund

Praxiseinsatz

Siehe auch



ZINSTERMTAGE() COUPDAYS()

ZINSTERMTAGE(*Abrechnung*; *Fälligkeit*; *Häufigkeit*; *Basis*)

Diese Funktion ermittelt die Anzahl der Tage der Zinsperiode, in die der Besitzwechsel eines festverzinslichen Wertpapiers mit regelmäßigen Zinsterminen fällt.

Abrechnung (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum, an dem die Rückzahlung des durch das Wertpapier verbrieften Darlehens erfolgt.

Da es durchaus üblich ist, festverzinsliche Wertpapiere mit mehreren Zinsterminen im Jahr auszustatten, informiert das erforderliche Argument *Häufigkeit* über deren Anzahl. Zulässige Werte sind: 1 = jährlich; 2 = halbjährlich; 4 = vierteljährlich.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.11. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

An die Argumente werden folgende Anforderungen gestellt:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, wird der Fehlerwert #WERT! zurückgegeben
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt die Funktion den Fehlerwert #ZAH! zurück. Das Gleiche geschieht, wenn der Abrechnungstermin nach der Fälligkeit liegt.

Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Syntax

Definition

Argumente

Wichtig

Hintergrund Regelmäßige Kuponzinszahlungen eines festverzinslichen Wertpapiers (Anleihe) richten sich nach dem Fälligkeitsdatum und der Häufigkeit innerhalb eines Jahrs.

Unterschiedliche Ergebnisse im unterjährlichen Bereich können dabei durch unterschiedliche Tagezählung im Jahr entstehen.

Praxeinsatz Diese Funktion wurde in den ausführlichen Rechnungen der Beispiele zu KURS() und RENDITE() verwendet.

Siehe auch ZINSTERMNZ(), ZINSTERMTAGNZ(), ZINSTERMTAGVA(), ZINSTERMVZ(), ZINSTERMZAHL(), KURS(), RENDITE(), TAGE360()



Die genannten Beispiele finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Kursrechnung.xlsx* auf den Arbeitsblättern *KURS* und *RENDITE*.

ZINSTERMTAGNZ() COUPDAYSNC()

Syntax ZINSTERMTAGNZ(*Abrechnung*; *Fälligkeit*; *Häufigkeit*; *Basis*)

Definition Diese Funktion ermittelt die Anzahl der Tage zwischen dem Tag des Besitzerwechsels eines festverzinslichen Wertpapiers mit regelmäßigen Zinsterminen und dem nächsten Zinsternin.

Argumente *Abrechnung* (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum, an dem die Rückzahlung des durch das Wertpapier verbrieften Darlehens erfolgt.

Da es durchaus üblich ist, festverzinsliche Wertpapiere mit mehreren Zinsterminen im Jahr auszustatten, informiert das erforderliche Argument *Häufigkeit* über deren Anzahl. Zulässige Werte sind: 1 = jährlich; 2 = halbjährlich; 4 = vierteljährlich.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.11. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

An die Argumente werden folgende Anforderungen gestellt:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, wird der Fehlerwert #WERT! zurückgegeben
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt die Funktion den Fehlerwert #ZAHL! zurück. Das Gleiche geschieht, wenn der Abrechnungstermin nach der Fälligkeit liegt.

Wichtig Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Regelmäßige Kuponzinszahlungen eines festverzinslichen Wertpapiers (Anleihe) richten sich nach dem Fälligkeitsdatum und der Häufigkeit innerhalb eines Jahrs.

Das Ergebnis der Funktion kann genutzt werden, um die Zinsen auszurechnen, die dem neuen Besitzer aus dem Rest der Zinsperiode noch zustehen, oder eine Barwertberechnung von Leistungen zum Zinstermin über eine gebrochene Periode vorzunehmen. Unterschiedliche Ergebnisse im unterjährlichen Bereich können dabei durch unterschiedliche Tagezählung im Jahr entstehen.

Diese Funktion wurde in den ausführlichen Rechnungen der Beispiele zu KURS() und RENDITE() verwendet.

ZINSTERMNZ(), ZINSTERMTAGE(), ZINSTERMTAGVA(), ZINSTERMVZ(), ZINSTERMZAHL(), KURS(), RENDITE(), TAGE360()

Die genannten Beispiele finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Kursrechnung.xlsx* auf dem Arbeitsblatt *KURS* und *RENDITE*.

Hintergrund**Praxiseinsatz****Siehe auch**

ZINSTERMTAGVA() COUPDAYBS()

ZINSTERMTAGVA(*Abrechnung*; *Fälligkeit*; *Häufigkeit*; *Basis*)

Diese Funktion ermittelt die Anzahl der Tage, die seit der letzten Zinszahlung bis zum Besitzwechsel eines festverzinslichen Wertpapiers mit regelmäßigen Zinsterminen vergangen sind.

Abrechnung (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum, an dem die Rückzahlung des durch das Wertpapier verbrieften Darlehens erfolgt.

Da es durchaus üblich ist, festverzinsliche Wertpapiere mit mehreren Zinsterminen im Jahr auszustatten, informiert das erforderliche Argument **Häufigkeit** über deren Anzahl. Zulässige Werte sind: 1 = jährlich; 2 = halbjährlich; 4 = vierteljährlich.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.11. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

An die Argumente werden folgende Anforderungen gestellt:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, wird der Fehlerwert **#WERT!** zurückgegeben
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt die Funktion den Fehlerwert **#ZAHL!** zurück. Das Gleiche geschieht, wenn der Abrechnungstermin nach der Fälligkeit liegt.

Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Syntax**Definition****Argumente****Wichtig**

Hintergrund Regelmäßige Kuponzinszahlungen eines festverzinslichen Wertpapiers (Anleihe) richten sich nach dem Fälligkeitsdatum und der Häufigkeit innerhalb eines Jahrs.

Das Ergebnis der Funktion kann verwendet werden, um die Zinsen auszurechnen, die der zukünftige Besitzer für die bereits vergangenen Tage der aktuellen der Zinsperiode (sogenannte Stückzinsen) zusätzlich zum Kurs bei Kauf bezahlen muss. Unterschiedliche Ergebnisse im unterjährlichen Bereich können dabei durch unterschiedliche Tagezählung im Jahr entstehen.

Praxiseinsatz Diese Funktion wurde in den ausführlichen Rechnungen der Beispiele zu KURS() und RENDITE() verwendet.

Siehe auch ZINSTERMNZ(), ZINSTERMTAGE(), ZINSTERMTAGNZ(), ZINSTERMVZ(), ZINSTERMZAHL(), KURS(), RENDITE(), TAGE360()



Die genannten Beispiele finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Kursrechnung.xlsx* auf dem Arbeitsblatt *KURS* und *RENDITE*.

ZINSTERMVZ() COUPPCD()

Syntax ZINSTERMVZ(*Abrechnung*; *Fälligkeit*; *Häufigkeit*; *Basis*)

Definition Diese Funktion ermittelt das Datum der bislang letzten Zinszahlung vor Besitzwechsel eines festverzinslichen Wertpapiers mit regelmäßigen Zinsterminen.

Argumente *Abrechnung* (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum, an dem die Rückzahlung des durch das Wertpapier verbrieften Darlehens erfolgt.

Da es durchaus üblich ist, festverzinsliche Wertpapiere mit mehreren Zinsterminen im Jahr auszustatten, informiert das erforderliche Argument *Häufigkeit* über deren Anzahl. Zulässige Werte sind: 1 = jährlich; 2 = halbjährlich; 4 = vierteljährlich.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.11. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

An die Argumente werden folgende Anforderungen gestellt:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, wird der Fehlerwert #WERT! zurückgegeben
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt die Funktion den Fehlerwert #ZAHL! zurück. Das Gleiche geschieht, wenn der Abrechnungstermin nach der Fälligkeit liegt.

Wichtig Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Regelmäßige Kuponzinszahlungen eines festverzinslichen Wertpapiers (Anleihe) richten sich nach dem Fälligkeitsdatum und der Häufigkeit innerhalb eines Jahrs.

Das Ergebnis der Funktion kann verwendet werden, um Stückzinsen »per Hand« auszurechnen (siehe die Erläuterungen zu ZINSTERMTAGVA()) oder den Termin auf einem Arbeitsblatt zur besseren Orientierung mit aufzunehmen. Unterschiedliche Ergebnisse im unterjährlichen Bereich können dabei durch unterschiedliche Tagezählung im Jahr entstehen.

Diese Funktion wurde zur Verdeutlichung der Überlegungen in den ausführlichen Rechnungen der Beispiele zu KURS() verwendet.

ZINSTERMNZ(), ZINSTERMTAGE(), ZINSTERMTAGVA(), ZINSTERMTAGNZ(), ZINSTERMZAHL(), KURS(), RENDITE(), TAGE360()

Die genannten Beispiele finden Sie im Ordner `\Ms5-235\Kap14` in der Arbeitsmappe *Kursrechnung.xlsx* auf dem Arbeitsblatt *KURS*.

Hintergrund**Praxiseinsatz****Siehe auch**

ZINSTERMZAHL() COUPNUM()

ZINSTERMZAHL(*Abrechnung*; *Fälligkeit*; *Häufigkeit*; *Basis*)

Diese Funktion ermittelt die Anzahl der Zinszahlungen (Kupontermine), die der neue Besitzer nach Kauf eines festverzinslichen Wertpapiers mit regelmäßigen Zinsterminen vor sich hat.

Abrechnung (erforderlich) ist das Datum, an dem das Wertpapier den Besitzer wechselt.

Fälligkeit (erforderlich) ist das Datum, an dem die Rückzahlung des durch das Wertpapier verbrieften Darlehens erfolgt.

Da es durchaus üblich ist, festverzinsliche Wertpapiere mit mehreren Zinsterminen im Jahr auszustatten, informiert das erforderliche Argument *Häufigkeit* über deren Anzahl. Zulässige Werte sind: 1 = jährlich; 2 = halbjährlich; 4 = vierteljährlich.

Basis (optional) entscheidet über die Methode zur Ermittlung von Tagen im Jahr nach Tabelle 14.11. Wird es weggelassen, rechnet Excel mit *Basis* = 0.

An die Argumente werden folgende Anforderungen gestellt:

- ▶ Datumsangaben verlangen diese ohne Uhrzeit, Nachkommastellen werden also abgeschnitten. Auch die Argumente *Häufigkeit* und *Basis* werden zu ganzen Zahlen eingekürzt.
- ▶ Lassen sich Datumsargumente nicht zu einem zulässigen Datum auswerten, wird der Fehlerwert **#WERT!** zurückgegeben
- ▶ Lässt sich *Häufigkeit* nicht zu 1, 2 oder 4 sowie *Basis* nicht zu einer Zahl zwischen 0 und 4 auswerten, gibt die Funktion den Fehlerwert **#ZAHL!** zurück. Das Gleiche geschieht, wenn der Abrechnungstermin nach der Fälligkeit liegt.

Die Excel-Hilfe formuliert folgenden Hinweis:

»Datumsangaben sollten mit der DATUM()-Funktion oder als Ergebnisse anderer Formeln oder Funktionen eingegeben werden. [...] Es können Probleme auftreten, wenn Datumsangaben als Text eingegeben werden.«

Syntax**Definition****Argumente****Wichtig**

Hintergrund Regelmäßige Kuponzinszahlungen eines festverzinslichen Wertpapiers (Anleihe) richten sich nach dem Fälligkeitsdatum und der Häufigkeit innerhalb eines Jahrs.

Das Ergebnis der Funktion kann verwendet werden, um den Barwert der zukünftigen Leistungen aus dem Wertpapier (Kurs) zum Tage des nächsten Zinstermins zu berechnen. Vergleichen Sie hierzu die Erläuterungen zum ausführlichen Rechenbeispiel der Funktion KURS() bzw. RENDITE(). Unterschiedliche Ergebnisse im unterjährlichen Bereich können dabei durch unterschiedliche Tagezählung im Jahr entstehen.

Praxiseinsatz Diese Funktion wurde in den ausführlichen Rechnungen der Beispiele zu KURS() und RENDITE() verwendet.

Siehe auch ZINSTERMNZ(), ZINSTERMTAGE(), ZINSTERMTAGNZ(), ZINSTERMTAGVA(), ZINSTERMVZ(), KURS(), RENDITE(), TAGE360()



Die genannten Beispiele finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Kursrechnung.xlsx* auf dem Arbeitsblatt *KURS* und *RENDITE*.

ZINSZ() IPMT()

Syntax ZINSZ(*Zins*; *Zr*; *Zzr*; *Bw*; *Zw*; *F*)

Definition Diese Funktion ermittelt den Anteil an einer Annuität, der zur Zinszahlung für Darlehen benutzt wird, welches nach dem Prinzip der Annuitätentilgung zurückgezahlt wird.

Argumente *Zins* (erforderlich) benennt den Nominalzins des Kredits.

Zr (erforderlich) wird die für die gewünschte Periode zutreffende Nummer übergeben.

Zzr (erforderlich) gibt die Gesamtzahl der Perioden, in denen der Kredit zurückgezahlt wird, an.

Bw (erforderlich) erfasst die Darlehenssumme.

Zw (optional) bewertet Darlehen, deren Rückzahlung nach der Gesamtperiodenzahl noch nicht vollständig erfolgt ist. Ein solcher Effekt tritt etwa ein, wenn Hypothekendarlehen mit einer Zinsbindungsfrist ausgezahlt werden.

F (optional) kann angeben, ob die Zahlungen zu Beginn der Perioden zinswirksam werden ($F = 1$) oder erst an deren Ende ($F = 0$). Wird das Argument weggelassen, rechnet Excel so, als ob es 0 (Null) wäre.

Hintergrund Die Tilgung von Krediten kennt verschiedene Formen. Eine davon ist, dass der Schuldner jede Periode den gleichen Betrag zahlt. Dieser setzt sich aus einem Tilgungsanteil (der mit fortschreitender Zeit immer größer wird) und einem Zinsanteil (der immer kleiner wird) zusammen. Die Veränderung der Zusammensetzung hängt damit zusammen, dass die abzutragende Schuld mit der Zeit immer geringer wird. Diese Form wird als Annuitätentilgung bezeichnet, der Gesamtbetrag wird, auch wenn seine Zahlung nicht unbedingt jährlich erfolgt, als Annuität bezeichnet.

Der Tilgungsanteil der ersten Annuität ergibt sich aus dieser nach Abzug des Zinsanteils. Dieser ist aber gerade die Kreditsumme multipliziert mit dem für die Periode festgelegten Nominalzinssatz. Die Tilgungsanteile der weiteren Perioden bestimmen sich (nach einigen mathematischen Zwischenschritten) aus

$$(\text{Tilgungsanteil der ersten Periode}) \cdot (1 + \text{Nominalzinssatz})^{\text{Periodennummer} - 1}$$

Die Zinsanteile dieser Perioden können durch Abzug des Tilgungsanteils von der Annuität berechnet werden.

Sehr oft wird bei Kreditverträgen ein Jahreszinssatz als Nominalzinssatz formuliert, die Zahlung erfolgt aber unterjährlich. In solchen Fällen müssen Sie den unterjährlichen Periodenzinssatz ermitteln, indem Sie den Jahreszinssatz durch die Anzahl der Zahlungen im Jahr dividieren. Die Gesamtperiodenzahl ergibt sich dann aus den Perioden pro Jahr multipliziert mit der Anzahl der Jahre.

Hinweis

Im Beispiel »Tilgungsrechnung« zur Funktion BW() standen folgende Daten fest: Ein Schuldner nimmt einen Kredit in Höhe von 176.121,76 € auf, den er bei einem Zinssatz von 5,5 % innerhalb von 30 Jahren jeden Monat mit 1.000,00 € bedient. Der Tilgungsplan weist im 18. Monat einen Zinsanteil von 791,64 € aus. Sie können dieses Ergebnis bis auf eventuelle Rundungsfehler durch die Funktion ZINSZ() überprüfen:

Praxiseinsatz

=ZINSZ(5,5%/12;18;30*12;176121,76)

(die konkreten Zahlen entstehen aus Zellbezügen zu den Zellen, in denen sich die Daten befinden).

Das sich ergebende negative Vorzeichen des Betrags ist der anderen »Richtung« der Zahlung zuzuschreiben.

Rundungsfehler treten bei der Benutzung der integrierten Funktionen oft auf. Diese können nämlich nicht berücksichtigen, was am Bankschalter tatsächlich passiert: Es gibt nur Geldbeträge mit zwei Stellen nach dem Komma. Schreiben Sie also einen Tilgungsplan Monat für Monat auf, müssen Sie die Funktion RUNDEN() einsetzen, damit Sie das wirkliche Geschehen auch richtig abbilden.

BW(), KAPZ(), KUMKAPITAL(), RMZ(), ZINS(), ZW(), ZZR(), ZSATZINVEST()

Siehe auch

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Tilgungsrechnung.xlsx* auf dem Arbeitsblatt *BW*.



ZSATZINVEST() RRI()

ZSATZINVEST(*Zzr*; *Bw*; *Zw*)

Syntax

Diese Funktion berechnet den Zinssatz, zu welchem ein Kapital investiert werden muss, damit es über einen gegebenen Zeitraum von Zinsperioden einen bestimmten Zukunftswert besitzt.

Definition

Zzr (erforderlich) ist die Anzahl der Zinsperioden.

Argumente

Bw (erforderlich) ist die Größe des eingesetzten Kapitals.

Zw (erforderlich) ist der geforderte Endstand.

Die fünf Funktionen BW() = Barwert, ZW() = Zukunftswert, RMZ() = regelmäßige Zahlung, ZZR() = Zins- oder Zahlungszeiträume und ZINS() = Zinssatz, stehen in Umsetzung des obigen Äquivalenzprinzips in folgendem Zusammenhang:

Hintergrund

$$BW \cdot (1 + ZINS)^{ZZR} + RMZ \cdot (1 + F \cdot ZINS) \cdot \frac{(1 + ZINS)^{ZZR} - 1}{ZINS} + ZW = 0$$

Der Barwert wird bis zum Ende aufgezinst, ebenso die regelmäßigen Zahlungen. Zum Ende wird die Summe mit dem Zukunftswert verglichen.

Die Nutzung einer dieser Funktionen ist gleichbedeutend mit der jeweiligen finanzmathematischen Grundaufgabe: Berechnung eines Werts aus obiger Gleichung, wenn die anderen Werte bekannt sind.

Die Funktion ZSATZINVEST() reduziert diese Formel für den Fall $RMZ=0$ (das Kapital bleibt unverändert, es erfolgen außer Zinsgutschriften keine anderen Zahlungen) auf die Auflösung nach ZINS:

$$BW \cdot (1 + ZSATZINVEST)^{ZZR} - ZW = 0$$

Abweichend werden jedoch der Bar- und der Zukunftswert als positive Zahlen gefordert.

Praxiseinsatz Aufgaben zur Zinseszinsrechnung mithilfe von ZINS() lassen sich auch durch ZSATZINVEST() behandeln.

Auch das Beispiel zur Funktion ZW2() kann mithilfe von ZSATZINVEST() bearbeitet werden.

Siehe auch BW(), PDURATION(), RMZ(), ZZR(), ZINS(), ZW(), ZW2()



Ein Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Zinseszinsrechnung.xlsx* auf dem Arbeitsblatt ZSATZINVEST.

ZW() FV()

Syntax ZW(*Zins*; *Zzr*; *Rmz*; *Bw*; *F*)

Definition Die Funktion ZW() berechnet den Zukunftswert (Endwert) eines regelmäßigen Zahlungsstroms unter Beachtung einmaliger Zahlungen am Anfang des betrachteten Zeitraums nach dem finanzmathematischen Äquivalenzprinzip

Leistung des Schuldners + Leistung des Gläubigers = 0

Argumente *Zins* (erforderlich) legt den (konstanten) Periodenzinssatz als nachschüssigen Zinssatz fest. *Zzr* (erforderlich) legt die Anzahl der Zinsperioden fest. Dabei wird davon ausgegangen, dass eventuelle regelmäßige Zahlungen (Argument *Rmz* ist größer Null) am Ende oder Anfang der Zinsperioden erfolgen.

Rmz (erforderlich/optional, siehe Hinweis) informiert über die Höhe der regelmäßigen Zahlung und kann als Rente oder Annuität interpretiert werden.

Bw (optional/erforderlich, siehe Hinweis) ist der Anfangswert des Vorgangs im betrachteten Zeitraum (ein Kontostand mit Guthaben oder der Betrag eines Darlehens).

F (optional) legt fest, ob die regelmäßigen Zahlungen am Ende der Perioden ($F = 0$ oder nicht angegeben) oder aber am Anfang der Perioden ($F = 1$) erfolgen.

Hinweis Wird das Argument *Rmz* weggelassen, ist die Angabe von *Bw* zwingend. Ist *Bw* vorhanden, kann *Rmz* weggelassen werden. Das Weglassen der Argumente wirkt so, als ob sie mit 0 (Null) angegeben wurden.

Das genannte finanzmathematische Äquivalenzprinzip bedeutet, dass hinsichtlich des Vorzeichens zwischen Aus- und Einzahlungen (Kreditaufnahme und Tilgung, Investition und Desinvestition) unterschieden werden muss. Hier weicht Excel vom in der finanzmathematischen Literatur gebräuchlichen Äquivalenzprinzip

Wichtig

Leistung des Schuldners = Leistung des Gläubigers
ab.

Die fünf Funktionen $BW()$ = Barwert, $ZW()$ = Zukunftswert, $RMZ()$ = regelmäßige Zahlung, $ZZR()$ = Zins- oder Zahlungszeiträume und $ZINS()$ = Zinssatz, stehen in Umsetzung des obigen Äquivalenzprinzips in folgendem Zusammenhang:

Hintergrund

$$BW \cdot (1 + ZINS)^{ZZR} + RMZ \cdot (1 + F \cdot ZINS) \cdot \frac{(1 + ZINS)^{ZZR} - 1}{ZINS} + ZW = 0$$

Der Barwert wird bis zum Ende aufgezinst, ebenso die regelmäßigen Zahlungen. Zum Ende wird die Summe mit dem Zukunftswert verglichen.

Die Nutzung einer dieser Funktionen ist gleichbedeutend mit der jeweiligen finanzmathematischen Grundaufgabe: Berechnung eines Werts aus obiger Gleichung, wenn die anderen Werte bekannt sind. Die Funktionen stellen also die Gleichung nach der gleichnamigen Größe um (außer bei $ZINS$ – hier wird eine Näherungsrechnung durchgeführt).

Gewöhnlich erfolgen die Angaben des Zinssatzes als Jahreszinssatz. Die genannten Funktionen arbeiten nur dann korrekt, wenn die Perioden zum Zinssatz passen. Bei unterjährlicher Verzinsung wird in der Regel der Jahreszinssatz gleichmäßig auf die Perioden aufgeteilt: 12 Monate zu einem Zwölftel des Zinssatzes, 3 Monate zu einem Viertel und ein halbes Jahr zur Hälfte.

Hinweis

Die folgenden Beispiele orientieren sich in den Überschriften am »finanzmathematischen Sprachgebrauch«.

Praxiseinsatz

Jemand möchte im Alter ein kleines finanzielles Polster haben und beschließt, eine gerade angefallene Erbschaft in Höhe von 10.000,00 € auf 15 Jahre zu einem Zinssatz von 4,5 % fest anzulegen. Wie hoch ist das Guthaben nach Ablauf der Zeit?

Zinseszinsrechnung

Eine Berechnung nach

$$=ZW(4,5\%;15;;-10000)$$

liefert 19.352,82 €. Der Betrag der Erbschaft erhält deshalb das Minuszeichen, weil er zunächst abgegeben wird.

Jemand möchte bis zu seinem 60. Lebensjahr zu Beginn jeden Monats 750,00 € sparen. Es ist möglich, ein Konto zu benutzen, welches 4,5 % p.a. abwirft. Es ist gedacht, diesen Sparplan 15 Jahre lang durchzuhalten. Wie hoch ist das Guthaben am Ende?

Rentenrechnung

Sie ermitteln den Zukunftswert der »Rente« (das ist hier eine regelmäßige Sparrate) aus

$$=ZW(4,5\%/12;15*12;-750;;1)$$

zu 193.032,17 €. Die 1 steht für das vorschüssige Zahlen der Sparrate, das Minuszeichen für die Abgabe des Geldes bei der Bank.

Die Rechnung geht von Zinseszinsen innerhalb eines Jahrs aus. Auf einem Sparbuch lässt sich der Zinseszinseffekt allerdings nicht umsetzen, sodass das Ergebnis etwas »theoretisch« ist.

Tilgungsrechnung (Annuitätentilgung)

Jemand kann monatlich 1.000,00 € zur Rückzahlung (Tilgung plus Zinsen) eines Kredits einsetzen. Bei einem Zinssatz von 5,5 % p.a. beträgt die Zinsbindungsfrist zunächst 5 Jahre. Wie hoch ist die Restschuld nach dieser Zeit?

Auch hier führt ZW() zum Ziel (Tilgungsrechnung in diesem Sinne ist Rentenrechnung). Sie ermitteln 62.689,55 € über

=ZW(5,5%/12;5*12;-1000;100000)

Der Betrag erhält ein Minuszeichen, weil er noch zu tilgen ist.

Anders als beim Sparbuch erfolgt bei Hypothekendarlehen eine monatliche Verzinsung zu einem Zwölftel des vereinbarten Jahreszinssatzes (Nominalzins).

Wichtig

Die Verwendung von Formeln unterscheidet sich von der ausführlichen Notierung eines Kontoverlaufs mit den gegebenen Ein- und Auszahlungen möglicherweise durch Rundungsfehler. Das liegt daran, dass auf einem tatsächlich geführten Konto Rundungen auf zwei Stellen nach dem Konto erfolgen. Gestalten Sie ein solches Konto unter Excel nach, ist in den Zwischenschritten, die konkretes Geld bedeuten, jeweils die Funktion RUNDEN() einzusetzen. Eine ledigliche Begrenzung der Anzeige des Zellwerts auf zwei Stellen ist oft nicht korrekt.

Siehe auch

BW(), KAPZ(), PDURATION(), RMZ(), ZINS(), ZSATZINVEST(), ZZR()



Die Arbeitsmappen *Zinseszinsrechnung.xlsx*, *Rentenrechnung.xlsx* und *Tilgungsrechnung.xlsx* bzw. *Tilgungsrechnung.xls* im Ordner \Ms5-235\Kap14 halten die Beispiele mit entsprechender Überschrift auf dem jeweiligen Arbeitsblatt ZW bereit.

ZW2() FVSCHEDULE()

Syntax ZW2(*Kapital*;Zinsen)

Definition Diese Funktion errechnet den Endwert für ein Kapital mit variablen Periodenzinssätzen.

Argumente *Kapital* (erforderlich) benötigt die Angabe des Kapitals, welches in der Zukunft einer Verzinsung unterworfen werden soll.

Zinsen (erforderlich) verlangt eine lückenlose, die Reihenfolge beachtende Liste mit den Zinssätzen für die Perioden. Das kann ein Zellbezug oder eine Matrix (in geschweifte Klammern eingeschlossene, durch Semikolon getrennte Werte) sein. Nicht als Zahlen auswertbare Werte verursachen den Fehlerwert #WERT!. Leere Zellen werden so behandelt, als wären sie mit Nullen belegt.

Hintergrund Die Berechnung erfolgt unter Beachtung, dass Zinsen am Ende der Periode dem Kapital hinzugeschlagen werden:

$$\text{Endwert in einer Periode} = (\text{Anfangswert in der Periode}) \cdot (1 + \text{Periodenzinssatz})$$

Dadurch werden natürlich Zinseszinsseffekte abgebildet.

Die Resultate sind aber nicht in übersichtlichen Formeln darstellbar, ein Zugeständnis, welches gemacht werden muss, sollen einzelne Perioden besser (genauer als mit einem konstanten Zinssatz über die gesamte Zeit) bewertet werden können.

Jemand bekommt auf ein angelegtes Kapital nach einem Jahr 1,75 %, nach zwei Jahren 2,25 % und nach 3 Jahren 3,8 % Zinsen. Wie hoch ist die Rendite, wenn die Zinsen dem Kapital zugeschlagen werden?

Das lässt sich in einem Arbeitsblatt unter Verwendung von ZW2() und ZINS() wie in Abbildung 14.8 bestimmen. Auch ZSATZINVEST() führt hier zur Lösung.

	A	B	C	D	E	F
3						
4		Anlagebetrag		100,00 €		
5						
6		Laufzeit	Zinssätze	ZW2	ZINS	ZSATZINVEST
7		1 Jahre	1,75%	101,75 €	1,75%	1,75%
8		2 Jahre	2,25%	104,04 €	2,00%	2,00%
9		3 Jahre	3,80%	107,99 €	2,60%	2,60%

Abbildung 14.8: Renditeberechnung bei wechselnden Zinssätzen

Die Funktion ZW2() ermittelt den Stand der Dinge in den einzelnen Jahren durch Verwendung der Zinssätze der Spalte C vom Anfang bis zum jeweiligen Jahr. ZINS() bzw. ZSATZINVEST() ermitteln daraus und aus dem Anlagebetrag die Rendite.

Diese lässt sich so interpretieren: Würde der Betrag zu jährlich konstantem Zinssatz (gleich Rendite) angelegt, entstünde dasselbe Ergebnis wie durch die wechselnden Zinssätze.

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap14 in der Arbeitsmappe *Zinseszinsrechnung.xlsx* auf dem Arbeitsblatt ZW2.



ZZR() NPER()

ZZR(*Zins;Rmz;Bw;Zw;F*)

Die Funktion ZZR() berechnet die Laufzeit eines Vorgangs der Zinseszinsrechnung, Rentenrechnung oder Tilgungsrechnung. Dieser geht von eventuellen regelmäßigen Zahlungen gleicher Höhe und/oder einmaliger Zahlungen am Anfang/Ende des betrachteten Zeitraums nach dem finanzmathematischen Äquivalenzprinzip

Leistung des Schuldners + Leistung des Gläubigers = 0

aus.

Zins (erforderlich) legt den (konstanten) Periodenzinssatz als nachschüssigen Zinssatz fest.

Rmz (erforderlich/optional, siehe Hinweis) informiert über die Höhe der regelmäßigen Zahlungen und kann als Rente oder Annuität interpretiert werden.

Bw (erforderlich/optional, siehe Hinweis) ist der Anfangswert einer Richtung der Zahlungen. Für Auszahlpläne handelt es sich um den Kontostand am Anfang der Betrachtungen, bei Tilgungsplänen um die Kreditsumme.

Zw (optional/erforderlich, siehe Hinweis) ist der Kontostand, der sich am Ende des Vorgangs einstellen soll (etwa ein Restguthaben bei Auszahlungsplänen oder eine Abschlusstilgung bei Krediten in Höhe der Restschuld).

Syntax

Definition

Argumente

F (optional) legt fest, ob die regelmäßigen Zahlungen am Ende der Perioden ($F = 0$ oder nicht angegeben) oder aber am Anfang der Perioden ($F = 1$) erfolgen.

Hinweis Jeweils zwei der drei Größen Rmz , Bw und Zw müssen angegeben werden und dabei verschieden von 0 (Null) sein. Die Vorzeichen richten sich nach der »Richtung des Geldflusses« (vgl. die unten stehende Formel).

Wichtig Das genannte finanzmathematische Äquivalenzprinzip bedeutet, dass hinsichtlich des Vorzeichens zwischen Aus- und Einzahlungen (Kreditaufnahme und Tilgung, Investition und Desinvestition) unterschieden werden muss. Hier weicht Excel vom in der finanzmathematischen Literatur gebräuchlichen Äquivalenzprinzip

Leistung des Schuldners = Leistung des Gläubigers
ab.

Hintergrund Die fünf Funktionen $BW()$ = Barwert, $ZW()$ = Zukunftswert, $RMZ()$ = regelmäßige Zahlung, $ZZR()$ = Zins- oder Zahlungszeiträume und $ZINS()$ = Zinssatz, stehen in Umsetzung des obigen Äquivalenzprinzips in folgendem Zusammenhang:

$$BW \cdot (1 + ZINS)^{ZZR} + RMZ \cdot (1 + F \cdot ZINS) \cdot \frac{(1 + ZINS)^{ZZR} - 1}{ZINS} + ZW = 0$$

Der Barwert wird bis zum Ende aufgezinst, ebenso die regelmäßigen Zahlungen. Zum Ende wird die Summe mit dem Zukunftswert verglichen.

Die Nutzung einer dieser Funktionen ist gleichbedeutend mit der jeweiligen finanzmathematischen Grundaufgabe: Berechnung eines Werts aus obiger Gleichung, wenn die anderen Werte bekannt sind. Die Funktionen stellen also die Gleichung nach der gleichnamigen Größe um (außer bei $ZINS$ – hier wird eine Näherungsrechnung durchgeführt).

Hinweis Gewöhnlich erfolgen die Angaben des Zinssatzes als Jahreszinssatz. Die genannten Funktionen arbeiten nur dann korrekt, wenn die Perioden zum Zinssatz passen. Bei unterjährlicher Verzinsung wird in der Regel der Jahreszinssatz gleichmäßig auf die Perioden aufgeteilt: 12 Monate zu einem Zwölftel des Zinssatzes, 3 Monate zu einem Viertel und ein halbes Jahr zur Hälfte.

Praxiseinsatz Die folgenden Beispiele orientieren sich in den Überschriften am »finanzmathematischen Sprachgebrauch«.

Zinseszinsrechnung Jemand möchte im Alter ein kleines finanzielles Polster haben und beschließt, eine gerade angefallene Erbschaft in Höhe von 10.000,00 € zu einem Zinssatz von 4,5 % fest anzulegen. Er hofft auf eine Auszahlung von wenigsten 25.000,00 €. Wie lange muss das Geld liegen bleiben?

Eine Berechnung nach

$$=ZZR(4,5\%;; -10000;25000)$$

liefert 20,82 Jahre. Damit ist am Ende des 20. Jahrs das Ziel noch nicht erreicht, am Ende des 21. aber bereits überschritten. Die konkreten Kontostände lassen sich mit den Zahlen 20 bzw. 21 für das Laufzeitargument mittels $ZW()$ berechnen. Oder Sie nutzen einen Kontoplan.

Rentenrechnung Jemand hat bis zum 60. Lebensjahr 100.000,00 € angespart und möchte sich eine zusätzliche Rente in Höhe von 750,00 € monatlich auszahlen lassen. Wie lange reicht das angesparte Geld, wenn 4,5 % p.a. auf das noch vorhandene Guthaben gezahlt werden?

Sie ermitteln die Laufzeit der Rente aus

=ZZR(4,5%/12;-750;100000;;1)

zu 184,19 Monaten. Damit ist am Ende des 184. Monats das Konto soweit abgeschmolzen, dass eine weitere Abhebung nicht erfolgen kann.

Die Rechnung geht von Zinseszinsen innerhalb eines Jahrs aus. Auf einem Sparbuch lässt sich der Zinseszinsseffekt allerdings nicht umsetzen, sodass das Ergebnis etwas »theoretisch« bleibt.

Jemand zahlt monatlich 1.000,00 € zur Rückzahlung (Tilgung plus Zinsen) eines Kredits. Der Zinssatz ist über die gesamte Laufzeit mit 5,5 % p.a. vereinbart. Wie lange muss der Kredit bedient werden, wenn die Kreditsumme 175.000,00 € beträgt?

Auch hier führt die Anwendung von ZZR() zum Ziel (Tilgungsrechnung in diesem Sinne ist Rentenrechnung). Sie ermitteln 354,24 Monate über

=ZZR(5,5%/12;-1000;175000)

Damit erfolgt im 355. Monat eine letzte Zahlung, die geringer als die regelmäßigen ausfällt.

Dieses Beispiel zeigt, dass Excel auch mit sogenannten Prozentannuitäten in der Tilgungsrechnung umgehen kann. Bei diesen ermittelt sich die Höhe der Rückzahlung nicht aus der Laufzeit (siehe RMZ()), sondern durch die Angabe eines Tilgungssatzes, der die erste Tilgungsrate bestimmt (Stichwort: Kredit mit 1 % Tilgung). Die Summe aus der ersten Tilgungsrate und den Zinsen der ersten Periode ist dann der regelmäßige Rückzahlungsbetrag.

Anders als beim Sparbuch erfolgt beim Hypothekendarlehen eine monatliche Verzinsung zu einem Zwölftel des vereinbarten Jahreszinssatzes (Nominalzins).

Die Verwendung von Formeln unterscheidet sich von der ausführlichen Notierung eines Kontoverlaufs mit den gegebenen Ein- und Auszahlungen möglicherweise durch Rundungsfehler. Dies liegt daran, dass auf einem tatsächlich geführten Konto Rundungen auf zwei Stellen nach dem Konto erfolgen. Gestalten Sie ein solches Konto unter Excel nach, ist in den Zwischenschritten, die konkretes Geld bedeuten, jeweils die Funktion RUNDEN() einzusetzen. Eine ledigliche Begrenzung der Anzeige des Zellwerts auf zwei Stellen ist oft nicht korrekt.

BW(), KAPZ(), PDURATION(), RMZ(), ZINS(), ZINSZ(), ZSATZINVEST(), ZW()

Die Arbeitsmappen *Zinseszinsrechnung.xlsx*, *Rentenrechnung.xlsx* und *Tilgungsrechnung.xlsx* im Ordner \Ms5-235\Kap14 halten die Beispiele mit entsprechender Überschrift auf dem jeweiligen Arbeitsblatt ZZR bereit.

**Tilgungsrechnung
(Annuitätentilgung)**

Hinweis

Wichtig

Siehe auch



Kapitel 15

Mathematische und trigonometrische Funktionen

ABRUNDEN()	714
ABS()	715
AGGREGAT()	716
ARABISCH()	718
ARCCOS()	719
ARCCOSHYP()	721
ARCCOT()	722
ARCCOTHYP()	724
ARCSIN()	725
ARCSINHYP()	726
ARCTAN()	727
ARCTAN2()	729
ARCTANHYP()	730
AUFRUNDEN()	731
BASIS()	732

BOGENMASS()	733	POTENZREIHE()	765
COS()	734	PRODUKT()	767
COSEC()	736	QUADRATESUMME()	767
COSECHYP()	737	QUOTIENT()	768
COSHYP()	738	REST()	769
COT()	739	RÖMISCH()	771
COTHYP()	740	RUNDEN()	772
DEZIMAL()	742	SEC()	773
EXP()	743	SECHYP()	775
FAKULTÄT()	744	SIN()	776
GANZZAHL()	745	SINHYP()	778
GERADE()	745	SUMME()	779
GGT()	746	SUMMENPRODUKT()	780
GRAD()	747	SUMMEWENN()	781
ISO.OBERGRENZE()	748	SUMMEWENNS()	783
KGV()	749	SUMMEX2MY2()	785
KOMBINATIONEN()	750	SUMMEX2PY2()	786
KOMBINATIONEN2()	751	SUMMEXMY2()	787
KÜRZEN()	752	TAN()	788
LN()	752	TANHYP()	790
LOG()	753	TEILERGEBNIS()	792
LOG10()	754	UNGERADE()	794
MDET()	755	UNTERGRENZE()	794
MEINHEIT()	756	UNTERGRENZE.GENAU()	794
MINV()	757	UNTERGRENZE.MATHEMATIK()	795
MMULT()	759	VORZEICHEN()	797
OBERGRENZE()	760	VRUNDEN()	798
OBERGRENZE.GENAU()	760	WURZEL()	799
OBERGRENZE.MATHEMATIK()	761	WURZELPI()	800
PI()	763	ZUFALLSBEREICH()	800
POLYNOMIAL()	763	ZUFALLSZAHL()	801
POTENZ()	764	ZWEIFAKULTÄT()	802

Grundlegende mathematische und trigonometrische Aufgabenstellungen lassen sich in folgende Gebiete aufteilen:

1. Funktionen für allgemeine mathematische Aufgabenstellungen
2. Funktionen für trigonometrische Aufgabenstellungen
3. Sonstige Funktionen

Formeln für allgemeine mathematische Aufgabenstellungen

Dies ist der Bereich mit den meisten Funktionen innerhalb dieses Kapitels. Wir unterteilen ihn noch einmal wie folgt:

- ▶ **Grundlegende Rechenarten** Neben den Grundrechenarten Addieren, Subtrahieren, Multiplizieren und Dividieren werden in diesem Kapitel noch das Potenzieren, Radizieren und Logarithmieren behandelt. Die in diesem Kapitel dargestellten Rechenarten bilden die Grundlage für die Algebra.
- ▶ **Runden** Beim Runden gilt: Alle Werte < 5 werden abgerundet und ≥ 5 werden aufgerundet. Es gibt jedoch eine Vielzahl von Gründen, um von diesem Grundsatz abzuweichen, sei es, um Preise auf ein Vielfaches, von z.B. 5, zu runden, um einheitliche Beträge zu erhalten oder ein striktes Aufrunden mit dem Ziel, Werte innerhalb eines Bereichs zu vereinheitlichen.
- ▶ **Matrizenrechnung** Matrizen sind eine Anordnung von Zahlen in Tabellenform. Eine Matrix lässt sich in Zeilen und Spalten einteilen, welche auch als Vektoren bezeichnet werden (Zeilen- und Spaltenvektor). Die Matrix bildet die Grundlage für die lineare Algebra. In der linearen Algebra werden mithilfe von Matrizen Gleichungen mit mehreren Variablen berechnet.

Funktionen für trigonometrische Aufgabenstellungen

Als ein Unterbereich der Geometrie beschäftigt sich die Trigonometrie mit der Berechnung von Dreiecken. Als wichtige Grundlage dafür dienen die Winkel und Seitenlängen eines rechtwinkligen Dreiecks. Die längste Seite eines rechtwinkligen Dreiecks liegt dem rechten Winkel gegenüber und wird als Hypotenuse bezeichnet. Die beiden anderen (kürzeren) Seiten sind die Katheten. Die dem zu berechnenden Winkel anliegende Kathete bezeichnet man als Ankathete, die dem Winkel gegenüber liegende Kathete als Gegenkathete. Durch die sog. Winkelfunktionen lassen sich über die Winkel eines rechtwinkligen Dreiecks die Verhältnisse von Katheten und Hypotenuse bestimmen. Mit den Arcus-Funktionen oder inversen Winkelfunktionen lassen sich aus den Verhältnissen von Hypotenuse und Katheten die Winkel berechnen. Erläutert werden die Winkelfunktionen mittels eines Vollkreises ($=360^\circ$), anhand dessen sich alle Möglichkeiten der Winkelkonstellationen darstellen lassen.

Sonstige Funktionen

Die Funktionen ARABISCH() und RÖMISCH() zum Umwandeln von Zahlenwerten in römische Zahlen (mit denen Sie allerdings keine Berechnungen durchführen können, weil das Ergebnis ein Text ist) gehören in diese Kategorie. Wir beschreiben sie hier, weil sie von Microsoft so kategorisiert wurden. Nach unserer Meinung gehören sie eigentlich in die Kategorie der »Text- und Datenfunktionen« (siehe Kapitel 7).

Tabelle 15.1

Die Mathematisch trigonometrischen Funktionen in der Übersicht

Funktion	Beschreibung
ABRUNDEN()	Rundet einen Wert ab
ABS()	Liefert den absoluten Wert einer Zahl
AGGREGAT()	Gibt das Aggregat in einer Liste oder Matrix zurück
ARABISCH()	Wandelt eine römische Zahl in eine arabische Zahl um
ARCCOS()	Berechnet den Winkel dessen Kosinus Zahl ist
ARCCOSHYP()	Liefert den umgekehrten hyperbolischen Kosinus einer Zahl
ARCCOT()	Gibt den Arkuskotangens einer Zahl zurück
ARCCOTHYP()	Gibt den hyperbolischen Arkuskotangens einer Zahl zurück
ARCSIN()	Gibt den Arkussinus oder auch umgekehrten Sinus einer Zahl zurück
ARCSINHYP()	Errechnet den umgekehrten hyperbolischen Sinus einer Zahl
ARCTAN()	Liefert den Arkustangens oder auch umgekehrten Tangens einer Zahl zurück
ARCTAN2()	Gibt den Arkustangens oder auch umgekehrten Tangens ausgehend von einer x- und einer y-Koordinate zurück
ARCTANHYP()	Berechnet den umgekehrten hyperbolischen Tangens einer Zahl
AUFRUNDEN()	Rundet eine Zahl auf
BASIS()	Konvertiert eine Zahl in eine Textdarstellung mit der angegebenen Basis
BOGENMASS()	Wandelt Grad in Bogenmaß (Radiant) um
COS()	Gibt den Kosinus einer Zahl zurück
COSEC()	Gibt den Kosekans eines Winkels zurück
COSECHYP()	Gibt den hyperbolischen Kosekans eines Winkels zurück
COSHYP()	Berechnet den hyperbolischen Kosinus einer Zahl
COT()	Gibt den Kotangens eines Winkels zurück
COTHYP()	Gibt den hyperbolischen Kotangens einer Zahl zurück
DEZIMAL()	Konvertiert eine Textdarstellung einer Zahl mit einer angegebenen Basis in eine Dezimalzahl
EXP()	Potenziert eine Basis mit dem angegebenen Wert
FAKULTÄT()	Liefert die Fakultät zu Zahl mit einer Schrittlänge von 1
GANZZAHL()	Rundet eine Zahl auf die nächstkleinere ganze Zahl ab
GERADE()	Rundet eine Zahl auf die nächste gerade ganze Zahl auf
GGT()	Berechnet den größten gemeinsamen Teiler
GRAD()	Wandelt Bogenmaß (Radiant) in Grad um
ISO.OBERGRENZE()	Rundet eine Zahl auf die nächste Ganzzahl oder auf das kleinste Vielfache des angegebenen Schritts auf
KGV()	Errechnet das kleinste gemeinsame Vielfache
KOMBINATIONEN()	Berechnet, wie viele Gruppen aus einer bestimmten Anzahl von Elementen gebildet werden können
KOMBINATIONEN2()	Gibt die Anzahl der Kombinationen mit Wiederholung für eine angegebene Anzahl von Elementen zurück



Funktion	Beschreibung
KÜRZEN()	Kürzt eine Zahl auf die angegebene Anzahl an Stellen hinter dem Komma
LN()	Gibt den natürlichen Logarithmus einer Zahl zurück
LOG()	Errechnet den Logarithmus einer Zahl zu einer beliebigen Basis
LOG10()	Berechnet den Logarithmus einer Zahl zur Basis 10
MDET()	Errechnet die Determinante einer Matrix
MEINHEIT()	Gibt die Einheitsmatrix für die angegebene Größe zurück
MINV()	Gibt die Inverse einer Matrix zurück
MMULT()	Berechnet das Produkt zweier Matrizen
OBERGRENZE()	Rundet eine Zahl betragsmäßig auf das kleinste Vielfache eines anderen Werts auf
OBERGRENZE.GENAU()	Rundet eine Zahl unabhängig vom Vorzeichen auf
OBERGRENZE.MATHEMATIK()	Rundet eine Zahl auf die nächste Ganzzahl oder auf das kleinste Vielfache des angegebenen Schritts auf
PI()	Gibt die Konstante PI zurück
POLYNOMIAL()	Gibt den Polynomkoeffizienten einer Gruppe zurück
POTENZ()	Potenziert eine Zahl
POTENZREIHE()	Gibt die Summe von Potenzen zurück. Wird zur Berechnung von Potenzreihen und dichotomen Wahrscheinlichkeiten verwendet.
PRODUKT()	Einfache Multiplikation von Werten
QUADRATSUMME()	Summiert die quadrierten Argumente
QUOTIENT()	Gibt den ganzzahligen Anteil einer Division zurück
REST()	Gibt den Rest einer Division zurück
RÖMISCH()	Wandelt eine arabische Zahl in eine römische Zahl als Text um
RUNDEN()	Rundet eine Zahl auf eine bestimmte Anzahl von Dezimalstellen
SEC()	Gibt den Sekans eines Winkels zurück
SECHYP()	Gibt den hyperbolischen Sekans eines Winkels zurück
SIN()	Berechnet den Sinus einer Zahl
SINHYP()	Gibt den hyperbolischen Sinus einer Zahl zurück
SUMME()	Bildet die Summe von Argumenten
SUMMENPRODUKT()	Multipliziert die einander entsprechenden Komponenten einer angegebenen Matrix und bildet die Summe dieser Produkte
SUMMEWENN()	Addiert die Zahlen, die mit einem Suchkriterium übereinstimmen
SUMMEWENNS()	Addiert die Zahlen, die mit mehreren Suchkriterien übereinstimmen
SUMMEX2MY2()	Summiert für zusammengehörige Komponenten zweier Matrizen die Differenzen der Quadrate
SUMMEX2PY2()	Summiert für zusammengehörige Komponenten zweier Matrizen die Summen der Quadrate

Funktion	Beschreibung
SUMMEXMY2()	Summiert für zusammengehörige Komponenten zweier Matrizen die quadrierten Differenzen
TAN()	Berechnet den Tangens einer Zahl
TANHYP()	Gibt den hyperbolischen Tangens einer Zahl zurück
TEILERGEBNIS()	Berechnet ein Teilergebnis in einer Liste oder Datenbank
UNGERADE()	Rundet eine Zahl auf die nächste ungerade ganze Zahl auf
UNTERGRENZE()	Rundet eine Zahl ab auf das Vielfache einer anderen Zahl
UNTERGRENZE.GENAU()	Rundet eine Zahl unabhängig vom Vorzeichen ab
UNTERGRENZE.MATHEMATIK()	Rundet eine Zahl auf die nächste Ganzzahl oder auf das kleinste Vielfache des angegebenen Schritts ab
VORZEICHEN()	Gibt das Vorzeichen einer Zahl zurück
VRUNDEN()	Gibt eine auf das gewünschte Vielfache gerundete Zahl zurück
WURZEL()	Berechnet die Quadratwurzel einer Zahl
WURZELPI()	Errechnet die Wurzel aus einer mit Pi multiplizierten Zahl
ZUFALLSBEREICH()	Gibt eine ganze Zufallszahl aus dem festgelegten Bereich zurück
ZUFALLSZAHL()	Gibt eine Zufallszahl zwischen 0 und 1 zurück mit bis zu 16 Stellen hinter dem Komma
ZWEIFAKULTÄT()	Berechnet die Fakultät zu einer Zahl mit Schrittlänge 2

ABRUNDEN()



ROUNDDOWN()

Syntax ABRUNDEN(*Zahl*; *Anzahl_Stellen*)

Definition Durch die Funktion ABRUNDEN() wird eine *Zahl* auf eine in *Anzahl_Stellen* festgelegte Anzahl an Stellen abgerundet.

Argumente *Zahl* (erforderlich) ist eine reelle Zahl, die abgerundet werden soll.

Anzahl_Stellen (erforderlich) gibt an, auf wie viele Dezimalstellen die Zahl abgerundet werden soll.

Hintergrund Im Gegensatz zu RUNDEN(), wo bei Zahlen < 5 ab und ≥ 5 aufgerundet wird, wird bei ABRUNDEN() konsequent auf die vorstehende Ziffer abgerundet. Im Prinzip wird *Zahl* an einer bestimmten Stelle einfach abgeschnitten. Diesen Schnittpunkt legt man mit *Anzahl_Stellen* fest. Ist *Anzahl_Stellen* ein negativer Wert, wird auf die entsprechende Anzahl von Stellen vor dem Komma abgerundet.

Ist *Anzahl_Stellen*:

- ▶ größer gleich 0 (null), wird die jeweilige Zahl entsprechend der angegebenen Anzahl von Dezimalstellen abgerundet.
- ▶ gleich 0, wird *Zahl* auf die nächste ganze Zahl abgerundet.
- ▶ kleiner 0, wird die jeweilige Zahl links des Kommas abgerundet.

Bei einer Preiskalkulation sollen die errechneten Endbeträge 15,482 € und 83,2578 € auf zwei Stellen hinter dem Komma abgerundet werden:

=ABRUNDEN(15,482;2) ergibt 15,48 €.

=ABRUNDEN(83,2578;2) ergibt 83,25 €.

Hier noch weitere Beispiele, um die Funktionsweise zu dokumentieren:

=ABRUNDEN(1,9;0) ergibt einen Wert von 1.

=ABRUNDEN(-2,846;2) ergibt einen Wert von -2,84.

=ABRUNDEN(18;-1) ergibt einen Wert von 10.

AUFRUNDEN(), GANZZAHL(), REST(), VRUNDEN(), OBERGRENZE.MATHEMATIK(), RUNDEN(), KÜRZEN(), UNTERGRENZE.MATHEMATIK()

Weitere Beispiele zum Testen finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt ABRUNDEN.



Siehe auch

ABS()



ABS()

ABS(*Zahl*)

Liefert den Absolutwert einer *Zahl*. Der Absolutwert einer Zahl ist die Zahl ohne ihr Vorzeichen.

Zahl (erforderlich) ist die reelle Zahl, deren Absolutwert Sie ermitteln möchten.

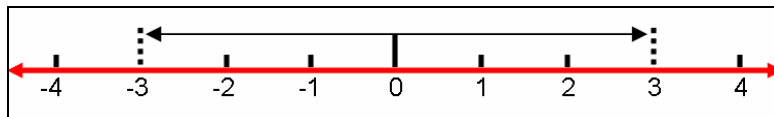


Abbildung 15.1: Gleicher Abstand von 3 zur 0

Die ganzen Zahlen +3 und -3 stehen auf einem Zahlenstrahl auf unterschiedlichen Seiten des Nullwerts. Sie sind aber beide um die gleichen Einheiten (Betrag oder Wert) vom Nullpunkt entfernt. Die Zahlen haben beide den gleichen absoluten Wert.

Absolute Werte werden durch zwei senkrechte Striche gekennzeichnet: $|a|$. Es gilt:

$$|a| = |-a|$$

Absolute Werte finden sich in einer Vielzahl von mathematischen Formeln wieder, z.B. in der mathematischen Definition des Areatangens (siehe die Funktion ARCTANHYP() auf Seite 730). Hier wird der Wert x als absoluter Wert definiert:

$$y = \operatorname{arctanh} x = \frac{1}{2} \ln \frac{1+x}{1-x} \text{ mit } |x| < 1$$

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Hier noch ein paar weitere Beispiele, um die Funktion zu dokumentieren:

=ABS(-18) ergibt einen absoluten Wert von 18.

=ABS(-3,47) ergibt einen absoluten Wert von 3,47.

=ABS(5) ergibt einen absoluten Wert von 5.

Siehe auch IMABS() Technische Funktionen, VORZEICHEN()



Diese Beispiele finden Sie im Ordner `Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `ABS`.

AGGREGAT() AGGREGATE()

Syntax Enthält zwei Versionen:

- ▶ Bezugsversion
AGGREGAT(*Funktion*;Optionen;Bezug1;Bezug2; ...)
- ▶ Matrixversion
AGGREGAT(*Funktion*;Optionen;Array;k)

Definition Gibt ein Aggregat in einer Liste oder einer Datenbank zurück.

Argumente *Funktion* (erforderlich) Ein Wert von 1 bis 19, der die zu verwendende Funktion angibt:

Funktion	Funktion
1	MITTELWERT()
2	ANZAHL()
3	ANZAHL2()
4	MAX()
5	MIN()
6	PRODUKT()
7	STABW.S()
8	STABW.N()
9	SUMME()
10	VAR.S()
11	VAR.P()
12	MEDIAN()
13	MODUS.EINF()
14	KGRÖSSTE()
15	KKLEINSTE()
16	QUANTIL.INKL()
17	QUARTILE.INKL()
18	QUANTIL.EXKL()
19	QUARTILE.EXKL()

Option (erforderlich) Ein numerischer Wert, der bestimmt, welche Werte im Berechnungsbereich ignoriert werden sollen.

Option	Verhalten
0 oder nicht angegeben	Geschachtelte TEILERGEBNIS()- und AGGREGAT()-Funktionen ignorieren
1	Ausgeblendete Zeilen, geschachtelte TEILERGEBNIS()- und AGGREGAT()-Funktionen ignorieren
2	Fehlerwerte, geschachtelte TEILERGEBNIS()- und AGGREGAT()-Funktionen ignorieren
3	Ausgeblendete Zeilen, Fehlerwerte, geschachtelte TEILERGEBNIS()- und AGGREGAT()-Funktionen ignorieren
4	Nichts ignorieren
5	Ausgeblendete Zeilen ignorieren
6	Fehlerwerte ignorieren
7	Ausgeblendete Zeilen und Fehlerwerte ignorieren

Bezug1 (erforderlich) Das erste numerische Argument für Funktionen, die mehrere numerische Argumente nutzen, für die Sie den Aggregatwert ermitteln möchten.

Bezug2 ... (optional) Die numerischen Argumente 2 bis 253, deren Aggregatwert Sie berechnen möchten.

In der Matrixversion ist **Array** eine Matrix, eine Matrixformel oder ein Bezug auf einen Zellbereich, für den Sie den Aggregatwert berechnen möchten. Der optionale Parameter *k* steht in diesem Fall für die Position in der Matrix. Gemeint ist je nach angegebener *Funktion* der *k*-größte oder -kleinste Wert bzw. das *k*-Quantil oder -Quartil, z.B. gibt der Wert 3 für *k* bei Verwendung der *Funktion* Nr. 14 KGRÖSSTE() den drittgrößten Wert wieder.

Wenn ein zweites Bezugsargument erforderlich, aber nicht angegeben ist, gibt die Funktion AGGREGAT() den Fehler #WERT! zurück. Auch, wenn Sie einen 3D-Bezug verwenden, gibt AGGREGAT() den #WERT! zurück.

Hinweis

Sehr nützlich: Gibt es innerhalb der Bezüge weitere AGGREGAT()-Funktionen, werden diese geschachtelten Aggregate ignoriert, damit sie nicht mehrfach berücksichtigt werden. Dasselbe gilt, wenn die Bezüge Teilergebnisse enthalten.

Bitte beachten Sie: Die Funktion AGGREGAT() wurde für Datenspalten oder vertikale Bereiche konzipiert, nicht für Zeilen oder horizontale Bereiche. Das Ausblenden einer Spalte hat **keine** Auswirkungen auf das Ergebnis des Aggregats. Wird hingegen eine Zeile im vertikalen Bereich ausgeblendet, wirkt sich das auf das Aggregat aus.

Die AGGREGAT()-Funktion ist eine sehr mächtige Funktion. Sie wurde von Microsoft zu Excel 2010 hinzugefügt, um bestimmte Beschränkungen bei den 19 enthaltenen Funktionen zu überwinden. Die meisten von ihnen geben z.B. einen Fehler zurück, wenn in den zu berechnenden Bezügen ein Fehler enthalten ist. In früheren Excel-Versionen war hierfür eine aufwendige WENNFehler()-Abfrage notwendig. Auch das Verhalten hinsichtlich der Behandlung von ausgeblendeten Zeilen kann nun gesteuert werden.

Hintergrund

Zusätzlich ist es mit der AGGREGAT()-Funktion jetzt möglich, Formeln über Wertebereiche mit Teilergebnissen zu berechnen, ohne die Teilergebnisse in das Ergebnis mit einzubeziehen.

1	2	3	A	B	C
				Kategorie	Werte
				A	27
				A	43
				A	57
				A	#DIV/0!
				A Ergebnis	#DIV/0!
				B	32
				B	12
				B	50
				B	5
				B	43
				B	59
				B Ergebnis	201
				Gesamtergebnis	#DIV/0!

Abbildung 15.2: Die TEILERGEBNIS()-Funktion kann die Fehler nicht ignorieren

Wollte man eine Summe über die Werte der Spalte C bilden, käme man durch die einfache SUMME() nicht weiter. Auch die Funktion TEILERGEBNIS() führt aufgrund der Fehlerwerte zu keiner Lösung. Mit der Funktion

```
=AGGREGAT(9;3;C18:C29)
```

erhält man trotz Fehlerwert und Teilergebnissen ein verwertbares Ergebnis. Zusätzlich besteht noch die Möglichkeit, ausgeblendete Zellen zu ignorieren, sodass eine Summe nur über den sichtbaren Bereich gebildet wird. So kann man mit einem Klick ganze Bereiche im Ergebnis berücksichtigen oder ausblenden.

Praxiseinsatz Der Einsatz in der Praxis ist sehr vielfältig, z.B. Kalkulationsschemata oder Messreihen bei denen trotz Fehlerwerten und Teilergebnissen ein Gesamtergebnis berechnet werden soll.

Siehe auch TEILERGEBNIS()



Weitere Beispiele zum Testen finden Sie im Ordner `Ms5-235\Kap15` in der Arbeitsmappe `AGGREGAT.xlsx` (ab Excel 2010).

Neu in Excel 2013

ARABISCH()



ARABIC()

Syntax ARABISCH(*Text*)

Definition ARABISCH() wandelt eine römische Zahl in eine arabische Zahl um.

Argumente *Text* (erforderlich) enthält die römische Zahl als Zeichenfolge, die in Anführungszeichen geschrieben werden muss. Die Zeichenfolge darf maximal 255 Zeichen enthalten. Groß- oder Kleinschreibung spielt keine Rolle.

Das römische Zahlensystem enthält die Grundsymbole:

I = 1; V = 5; X = 10; L = 50; C = 100; D = 500; M = 1000

Die Schreibweise verlangt, von links nach rechts mit absteigendem Zahlenwert die Grundsymbole anzuordnen und die Zahlenwerte zu addieren. Steht dagegen ein Symbol mit einem kleineren Zahlenwert vor dem mit einem größeren, wird sein Wert von dem folgenden größeren subtrahiert. Diese Arbeit nimmt uns die Funktion ARABISCH() ab.

Obwohl negative römische Zahlen sehr ungewöhnlich sind, wird die Berechnung von negativen römischen Zahlenwerten unterstützt. Setzen Sie ein Minuszeichen vor den Text der jeweiligen römischen Zahl (beispielsweise *"-MMXI"*).

Führende und nachfolgende Leerzeichen werden ignoriert.

Aufgrund der Beschränkung auf maximal 255 Zeichen ist die größte darstellbare Zahl 255.000.

Beachten Sie bitte auch die Ausführungen bei der Funktion RÖMISCH() auf Seite 771!

=ARABISCH("XIXVIII") ergibt 27.

=ARABISCH("MMXIII") ergibt 2013.

=ARABISCH("MmdXIXVIII") ergibt 2527.

BASIS(), DEZIMAL(), RÖMISCH()

Diese Beispiele finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `ARABISCH`.

ARCCOS()



ARCCOS(*Zahl*)

Die Funktion gibt den Arkuskosinus von *Zahl* zurück. Der Arkuskosinus ist der Winkel, dessen Kosinus *Zahl* ist.

Zahl (erforderlich) ist der Kosinus des jeweiligen Winkels und muss zwischen -1 und $+1$ liegen.

Mithilfe der ARCCOS()-Funktion lässt sich mit der Angabe des Kosinus (also dem Verhältnis von Ankathete zu Hypotenuse) der anliegende Winkel berechnen. Das Ergebnis wird als Bogenmaß ausgegeben und muss, wenn es als Grad-Wert vorliegen soll, mit der Funktion GRAD() umgerechnet werden. Da jeder Winkel in einem Vollkreis zweimal vorkommt (Supplementwinkel), wird der Wertebereich für den Arkuskosinus eingeschränkt:

- ▶ Wird der Winkel als Bogenmaß berechnet, erstreckt sich der Wertebereich von 0 (null) bis π (Pi).
- ▶ Wird der Winkel in Grad angegeben, erstreckt sich der Wertebereich von 0° bis 180° .

Der Arkuskosinus gehört zu streng monotonen Funktionen, da er im Definitionsbereich streng monoton abnimmt.

Hintergrund

Hinweis

Praxiseinsatz

Siehe auch



Syntax

Definition

Argumente

Hintergrund

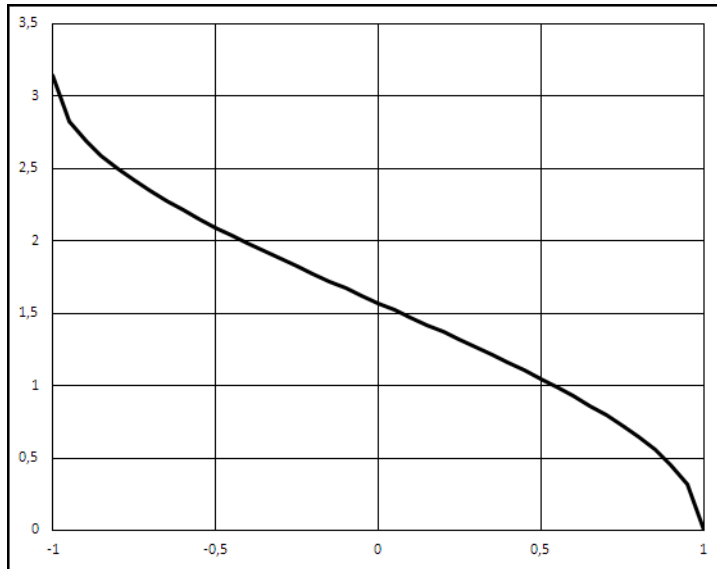


Abbildung 15.3: Darstellung der ARCCOS()-Funktion

Hinweis Die Arkuskosinus-Funktion gehört zu den Umkehrfunktionen trigonometrischer Funktionen, die aus einem Winkelfunktionswert den zugehörigen Winkel berechnen. Man nennt sie auch Arkusfunktionen, inverse Winkelfunktionen oder zyklometrische Funktionen.

Praxiseinsatz Eine 10 m lange Leiter ist an eine Hauswand gelehnt. Das untere Ende ist 2,5 m von der Mauer entfernt. Wie groß ist der Neigungswinkel?

Zuerst wird der Kosinus aus dem Verhältnis von Ankathete (Abstand zur Wand) zur Hypotenuse (Länge der Leiter) berechnet:

$$\cos \alpha = \frac{2,5}{10} = 0,25$$

Anschließend wird aus dem Kosinus der Winkel α ermittelt:

$$= \text{GRAD}(\text{ARCCOS}(2,5/10))$$

Der Neigungswinkel beträgt $75,52^\circ$ oder 75° und $30'$. Da das Ergebnis der ARCCOS()-Funktion im Bogenmaß ausgegeben wird, rechnen wir noch in Grad um.

Hinweis Das etwa halbe Grad ($^\circ$) hinter dem Komma bei der Winkelangabe $75,52^\circ$ kann auch in Winkelminuten angegeben werden: Ein ($^\circ$) wird in 60 Winkelminuten ($'$) eingeteilt. Deshalb kann man den Winkel auch durch $75^\circ 30'$ ausdrücken. Diese Angabe findet man auch z.B. auf Seekarten wieder, in denen die Längen- bzw. Breitengrade entsprechend gekennzeichnet sind.

Gegeben ist ein Kosinus von 0,707106781. Der sich daraus ergebende Winkel kann sowohl 45° als auch 315° betragen. Da sich der Definitionsbereich auf 0° bis 180° beschränkt, lautet das Ergebnis 45°. Die entsprechende Formel sieht in Excel wie folgt aus:

=GRAD(ARCCOS(0,707106781))

Weil Excel das Ergebnis in Bogenmaß ausgibt, muss eine Umrechnung des Ergebnisses mithilfe der Funktion GRAD() erfolgen.

=ARCCOS(0) ergibt ein Bogenmaß von 1,570796327 (= 90°).

=ARCCOS(-0,5) ergibt ein Bogenmaß von 2,094395102 (=120°).

=ARCCOS(-1) ergibt ein Bogenmaß von 3,141592654 (= 180°).

ARCSIN(), COS(), GRAD(), SIN()

Die Tabelle für die grafische Darstellung aus Abbildung 15.3 und die oben genannten Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt ARCCOS.

**Weitere
Beispiele**

Siehe auch



ARCCOSHYP()



ACOSH()

ARCCOSHYP(*Zahl*)

ARCCOSHYP() liefert den umgekehrten hyperbolischen Kosinus (math.: *arcosh*) einer Zahl.

Zahl (erforderlich) kann jede beliebige reelle Zahl sein, die größer gleich 1 ist.

Die inversen Hyperbelfunktionen sind die Umkehrfunktionen zu den hyperbolischen Funktionen. Die Hyperbelfunktionen sinh, tanh und coth sind streng monoton, sodass sie genau eine inverse Funktion besitzen. Dagegen hat die Funktion cosh zwei symmetrisch zum positiven Zweig der Ordinate liegende Monotonintervalle und demnach auch zwei Umkehrfunktionen:

$y = \operatorname{arcosh} x$

und

$y = -\operatorname{arcosh} x$, wobei

$$\operatorname{arcosh} x = \ln\left(x + \sqrt{x^2 - 1}\right) \text{ mit } x \in [1, \infty) \text{ ist.}$$

Der Graph beginnt im Punkt(1,0) und wächst bzw. fällt dann monoton.

Die Umkehrfunktionen der hyperbolischen Funktionen tragen nicht den gleichen Präfix wie die Umkehrfunktionen der trigonometrischen Funktionen »Arkus«, sondern »Area«. Ihre korrekten Namen lauten also Areasinus, Areakosinus, Areatangens usw. Der Hinweis auf die Zugehörigkeit zu den Hyperbelfunktionen erübrigt sich, weil die Areafunktionen laut Definition deren Umkehrfunktionen sind. Der Name »Area« (lat. Fläche) leitet sich von der geometrischen Definition dieser Funktionen ab, nach der sie die Fläche eines Hyperbelsektors bezeichnen.

Dessen ungeachtet tragen die Areafunktionen in Excel den Präfix ARC.

Syntax

Definition

Argumente

Hintergrund

Hinweis



Abbildung 15.4: Die grafische Darstellung des Areakosinus zeigt nur den monoton steigenden, als den im 1. Quadranten gelegenen Zweig; der monoton fallende Zweig liegt spiegelbildlich unter der Abszisse

Praxiseinsatz Zwei parallele elektrische Leiter mit dem Durchmesser d im Abstand a mit der Länge l haben die Kapazität C

$$C = \frac{\pi \cdot \varepsilon \cdot l}{\text{arcosh} \frac{a}{d}}$$

mit $\varepsilon = \varepsilon_0 \varepsilon_r$ als Dielektrizitätskonstante des Mediums.

Hier noch weitere Beispiele, um die Funktionsweise zu illustrieren:

=ARCCOSHYP(1) ergibt 0.

=ARCCOSHYP(2) ergibt 1,3169579.

=ARCCOSHYP(8) ergibt 2,7686594.

Siehe auch ARCSINHYP(), ARCTANHYP(), COSHYP, SINHYP(), TANHYP()



Die Tabelle für die grafische Darstellung aus Abbildung 15.4, das Beispiel aus der Elektrizitätslehre und die übrigen Beispiele finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `ARCCOSHYP.xlsx` auf den Arbeitsblättern *Graph* und *Kapazität*.

Neu in Excel
2013

ARCCOT()

 **ACOT()**

Syntax ARCCOT(*Zahl*)

Definition Die Funktion ARCCOT() gibt den Arkuskotangens einer Zahl als Bogenmaß im Bereich 0 bis π zurück.

Argumente *Zahl* (erforderlich) ist der Kotangens des gewünschten Winkels.

Der Arkuskotangens einer Zahl ist der Winkel, dessen Kotangens diese Zahl darstellt. Er ist demnach die Umkehrfunktion des Kotangens. Das Ergebnis (der »Hauptwert« des Arkuskotangens) wird im Bogenmaß erhalten. Wenn Sie es lieber im Gradmaß haben wollen, können Sie dies mit der Funktion GRAD() erreichen (siehe Seite 747).

Wegen der Periodizität des Kotangens ist der Arkuskotangens nicht eindeutig. Man beschränkt daher die Hauptwerte des Arkuskotangens auf das (offene) Intervall $]0;\pi[$. Die Excel-Funktion ARCCOT() lässt aber den Hauptwert π zu; sie arbeitet also in dem halboffenen Intervall $]0;\pi]$.

Die Arkuskotangens-Funktion gehört zu den Umkehrfunktionen trigonometrischer Funktionen, die aus einem Winkelfunktionswert den zugehörigen Winkel berechnen. Man nennt sie auch Arkusfunktionen, inverse Winkelfunktionen oder zyklometrische Funktionen.

Hintergrund

Hinweis

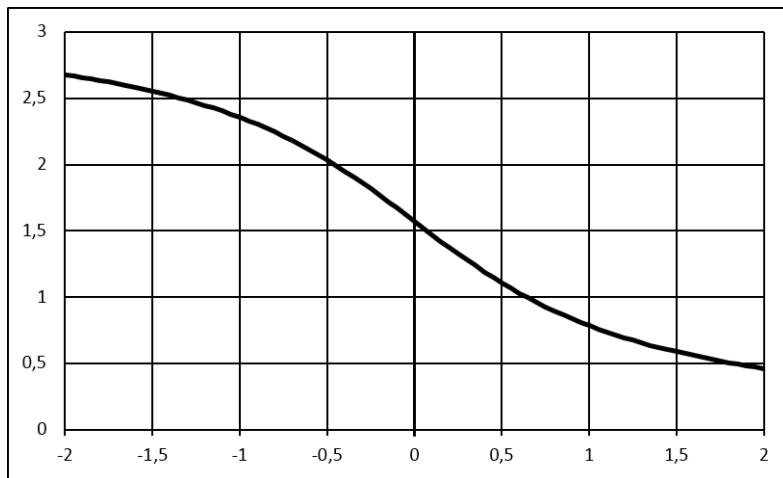


Abbildung 15.5: Graph der ARCCOT-Funktion

=ARCCOT(1) ergibt 0,785398163 (= 45°).

=ARCCOT(-81622) ergibt 3,14159265358979 (= π).

=GRAD(ARCCOT(0,5)) ergibt 63,43494882°.

ARCTAN(), COT(), GRAD(), TAN()

Die Tabelle für die grafische Darstellung aus Abbildung 15.5 und die oben genannten Beispiele finden Sie im Ordner `Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `ARCCOT`.

Praxiseinsatz

Siehe auch



Neu in Excel
2013

ARCCOTHYP()



ACOTH()

Syntax ARCCOTHYP(*Zahl*)

Definition ARCCOTHYP() liefert den umgekehrten hyperbolischen Kotangens (math.: *arcoth*) einer Zahl.

Argumente *Zahl* (erforderlich) kann jede beliebige reelle Zahl sein. Der Betrag von *Zahl* muss größer als 1 sein, d.h. $|x| > 1$.

Hintergrund Die inversen Hyperbelfunktionen sind die Umkehrfunktionen zu den hyperbolischen Funktionen. Die Hyperbelfunktionen *sinh*, *tanh* und *coth* sind streng monoton, sodass sie genau eine inverse Funktion besitzen.

Aus der Definition von *sinh* (siehe Seite 778) und *cosh* (siehe Seite 738) lässt sich die Definition für *coth* herleiten und daraus ergibt sich die Definition für *arcoth*:

$$y = \frac{1}{2} \ln \frac{x+1}{x-1} \quad \text{mit } -\infty < x < +\infty$$

Der Areakotangens ist eine ungerade und nur für $|x| > 1$ definierte Funktion. Der Graph ist im gesamten Definitionsbereich monoton fallend.

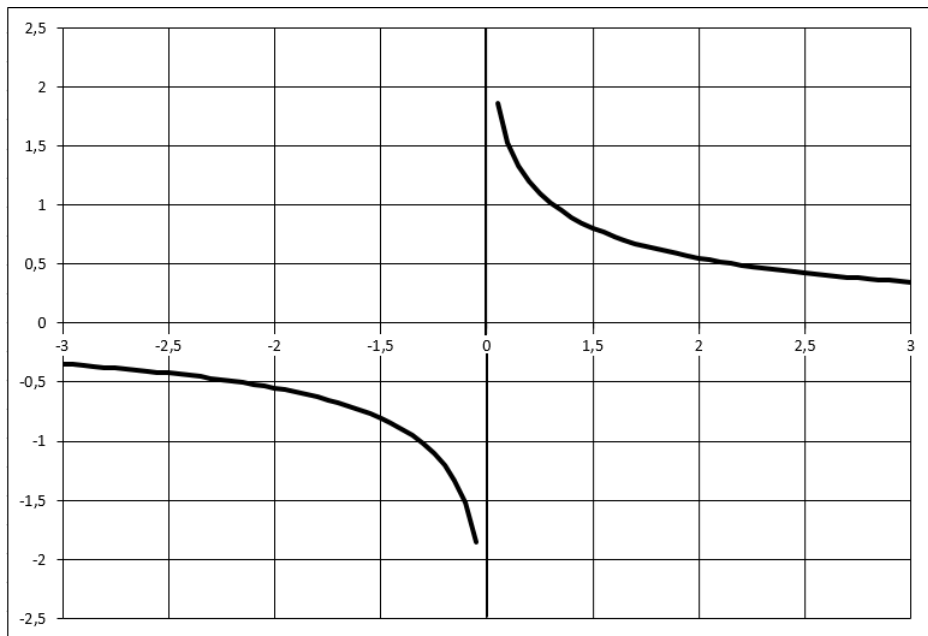


Abbildung 15.6: Darstellung der ARCCOTHYP()-Funktion

Hinweis Die Umkehrfunktionen der hyperbolischen Funktionen tragen nicht den gleichen Präfix wie die Umkehrfunktionen der trigonometrischen Funktionen »Arkus«, sondern »Area«. Ihre korrekten Namen lauten also Areasinus, Areakosinus, Areatangens usw. Der Hinweis auf die Zugehörigkeit zu den Hyperbelfunktionen erübrigt sich, weil die Areafunktionen laut Definition deren Umkehrfunktionen sind. Der Name »Area« (lat. Fläche) leitet sich von der geometrischen Definition dieser Funktionen ab, nach der sie die Fläche eines Hyperbelsektors bezeichnen.

=ARCCOTHYP(1,1) ergibt 1,522261219.

=ARCCOTHYP(-1,1) ergibt -1,522261219.

=ARCCOTHYP(6) ergibt 0,168236118.

ARCSINHYP(), ARCTANHYP(), COSHYP, COTHYP(), SINHYP(), TANHYP()

Die Tabelle für die grafische Darstellung aus Abbildung 15.6 und die oben genannten Beispiele finden Sie im Ordner `MS5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `ARCCOTHYP`.

Praxiseinsatz

Siehe auch



ARCSIN()



ARCSIN(*Zahl*)

Mithilfe der ARCSIN()-Funktion lässt sich mit der Angabe des Sinus (also das Verhältnis von Hypotenuse zur Gegenkathete) der anliegende Winkel berechnen.

Zahl (erforderlich) ist der Sinus des Winkels, dessen Arkussinus berechnet werden soll. Der entsprechende Sinuswert darf zwischen -1 und $+1$ liegen.

Die Formel ARCSIN() berechnet aus einem Sinuswert den zugrunde liegenden Winkel. Das Ergebnis wird in Bogenmaß ausgegeben. Soll das Ergebnis als Grad vorliegen, muss es entweder mit `180/PI()` multipliziert oder mithilfe der `GRAD()`-Funktion umgewandelt werden.

Da jeder Winkel in einem Vollkreis zweimal vorkommt (Supplementwinkel), wird der Wertebereich für den Arkussinus eingeschränkt:

- ▶ Wird der Winkel als Bogenmaß berechnet, erstreckt sich der Wertebereich von $-\pi/2$ bis $+\pi/2$.
- ▶ Wird der Winkel in Grad angegeben, erstreckt sich der Wertebereich von -90° bis $+90^\circ$.

Syntax

Definition

Argumente

Hintergrund

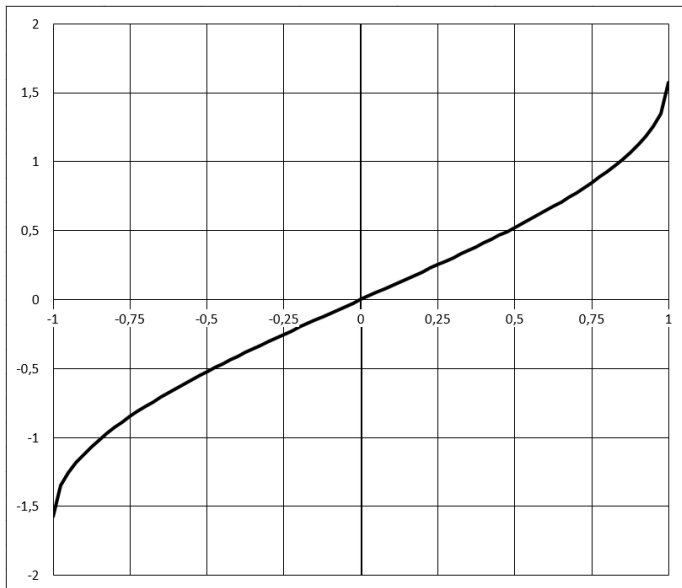


Abbildung 15.7: Darstellung der ARCSIN()-Funktion

Praxiseinsatz In der Praxis wird die Funktion ARCSIN() verwendet, wenn von einem rechtwinkligen Dreieck die Gegenkathete und die Hypotenuse bekannt sind und Sie die Winkel berechnen möchten:

$$\theta = \arcsin\left(\frac{\text{Gegenkathete}}{\text{Hypotenuse}}\right)$$

Hier einige Beispiele, um die Funktion zu dokumentieren:

=ARCSIN(1) ergibt ein Bogenmaß von 1,570796327 (= 90°).

=ARCSIN(0,5) ergibt ein Bogenmaß von 0,523598776 (= 30°).

=ARCSIN(-0,5) ergibt ein Bogenmaß von -0,523598776 (= -30°)

Siehe auch ARCCOS(), ARCSINHYP(), GRAD(), PI(), SIN()



Die Tabelle für die grafische Darstellung aus Abbildung 15.7 und die oben genannten Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt ARCSIN.

ARCSINHYP() ASINH()

Syntax ARCSINHYP(*Zahl*)

Definition Gibt den umgekehrten hyperbolischen Sinus (math.: *arsinh*) einer Zahl zurück.

Argumente *Zahl* (erforderlich) ist eine beliebige reelle Zahl.

Hintergrund Die inversen Hyperbelfunktionen sind die Umkehrfunktionen zu den hyperbolischen Funktionen. Die mathematische Definition des Areasinus ist:

$$y = \operatorname{arsinh} x = \ln\left(x + \sqrt{x^2 + 1}\right) \text{ mit } x \in \mathfrak{R}.$$

Hinweis Die Umkehrfunktionen der hyperbolischen Funktionen tragen nicht das gleiche Präfix wie die Umkehrfunktionen der trigonometrischen Funktionen »Arkus«, sondern »Area«. Ihre korrekten Namen lauten also Areasinus, Areakosinus, Areatangens usw. Der Hinweis auf die Zugehörigkeit zu den Hyperbelfunktionen erübrigt sich, weil die Areafunktionen laut Definition deren Umkehrfunktionen sind. Der Name »Area« (lat. Fläche) leitet sich von der geometrischen Definition dieser Funktionen ab, nach der sie die Fläche eines Hyperbelsektors bezeichnen.

Hier einige Beispiele, um die Funktion zu dokumentieren:

=ARCSINHYP(0) ergibt 0.

=ARCSINHYP(1) ergibt 0,88137359.

=ARCSINHYP(-1) ergibt -0,88137359.

=ARCSINHYP(10) ergibt 2,99822295.

=ARCSINHYP(-10) ergibt -2,99822295

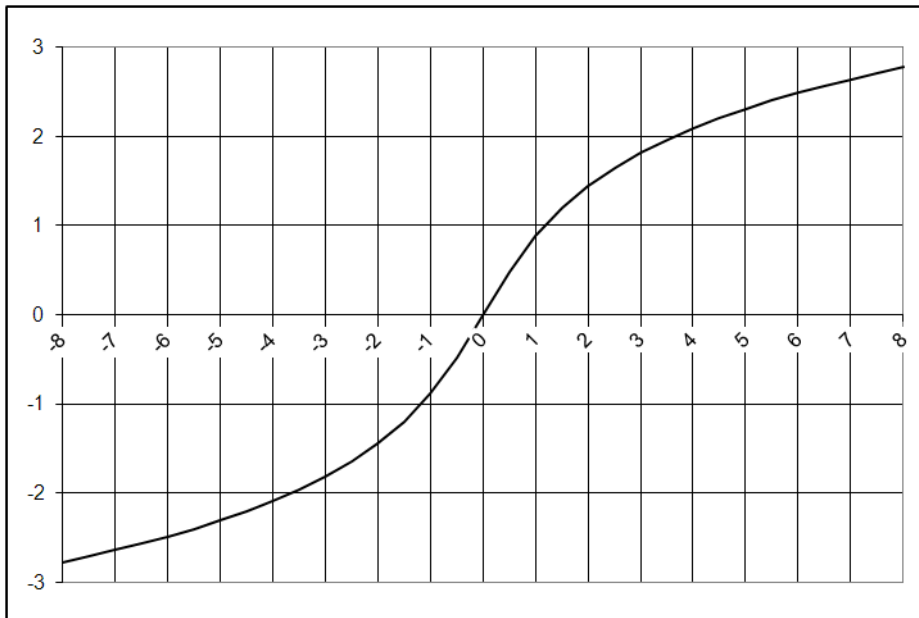


Abbildung 15.8: Der Arcsinus ist ungerade und eine zwischen $-\infty$ und $+\infty$ streng monoton wachsende Funktion

ARCCOSHYP(), ARCTANHYP(), COSHYP, SINHYP(), TANHYP()

Die Tabelle für die grafische Darstellung aus Abbildung 15.8 und die oben genannten Beispiele finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `ARCSINHYP`.

Siehe auch



ARCTAN()



ARCTAN(*Zahl*)

Gibt den Arkustangens oder auch umgekehrten Tangens einer *Zahl* zurück.

Zahl (erforderlich) ist der Tangens des Winkels, den Sie berechnen möchten.

Mithilfe der ARCTAN()-Funktion lässt sich mit der Angabe des Tangens (also das Verhältnis von Gegenkathete zur Ankathete) der anliegende Winkel berechnen. Anders gesagt: Der Arkustangens ist der Winkel, dessen Tangens *Zahl* ist. Der als Ergebnis zurückgegebene Winkel wird im Bogenmaß (Radiant) mit einem Wert zwischen $-\pi/2$ und $+\pi/2$ ausgegeben. Soll der Arkustangens in Grad ausgedrückt werden, muss das Ergebnis mit `180/PI()` multipliziert oder mithilfe der `GRAD()`-Funktion umgewandelt werden.

Da jeder Winkel in einem Vollkreis zweimal vorkommt (Supplementwinkel), wird der Wertebereich für den Arkustangens eingeschränkt:

- ▶ Wird der Winkel als Bogenmaß berechnet, erstreckt sich der Wertebereich von $-\pi/2$ bis $+\pi/2$.
- ▶ Wird der Winkel in Grad angegeben, erstreckt sich der Wertebereich von -90° bis $+90^\circ$ Grad.

Syntax

Definition

Argumente

Hintergrund

Werden die einzelnen Werte des Arkustangens aus dem Einheitskreis in ein Koordinatensystem übertragen, ergibt sich folgender Graph:

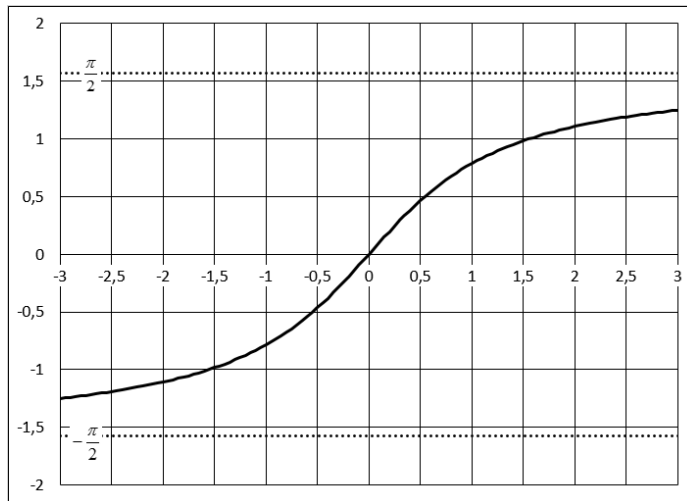


Abbildung 15.9: Darstellung der ARCTAN()-Funktion

Achtung Die ursprüngliche mathematische Schreibweise *arctan* oder *atan* wird heute immer mehr durch die Schreibweise \tan^{-1} ersetzt, was aber mit dem Kehrwert des Tangens, dem Kotangens, verwechselt werden kann.

Praxiseinsatz In der Praxis wird die Funktion ARCTAN() verwendet, wenn von einem rechtwinkligen Dreieck die Gegen- und Ankathete bekannt sind und Sie die Winkel berechnen möchten:

$$\theta = \arctan\left(\frac{\text{Gegenkathete}}{\text{Ankathete}}\right)$$

Hier weitere Beispiele, um die Funktion zu dokumentieren:

Gegeben ist ein Tangens von 1. Der sich daraus ergebende Winkel kann sowohl 45° als auch 225° betragen. Da sich der Definitionsbereich von -90° bis +90° beschränkt, lautet das Ergebnis 45°. Die entsprechende Formel sieht dann in Excel wie folgt aus:

=GRAD(ARCTAN(1))

Weil Excel das Ergebnis als Bogenmaß ausgibt, erfolgt eine Umrechnung des Ergebnisses durch die Funktion GRAD().

=GRAD(ARCTAN(-1)) ergibt einen Winkel α von -45°.

=GRAD(ARCTAN(0,75355405)) ergibt einen Winkel α von 37°.

=GRAD(ARCTAN(4,010780934)) ergibt einen Winkel α von 76°.

Siehe auch ARCTAN2(), ARCTANHYP(), PI(), TAN()



Die Tabelle für die grafische Darstellung aus Abbildung 15.9 und die oben genannten Beispielwerte finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `ARCTAN`.

ARCTAN2()



ARCTAN2(*x_Koordinate*; *y_Koordinate*)

Diese Funktion gibt den Arkustangens ausgehend von einer X/Y-Koordinate zurück. Dieser Arkustangens ist der Winkel zwischen der X-Achse und der Linie, die durch den Koordinatenursprung (0; 0) und den Punkt verläuft, der durch die X/Y-Koordinaten bestimmt ist.

Der als Ergebnis zurückgegebene Winkel wird im Bogenmaß (Radiant) mit einem Wert zwischen $-\pi$ und $+\pi$ (ausgenommen $-\pi$) ausgegeben.

x_Koordinate (erforderlich) ist die X-Koordinate des Punkts.

y_Koordinate (erforderlich) ist die Y-Koordinate des Punkts.

Der Arkustangens2 ist neben ARCTAN() eine weitere inverse Funktion (Umkehrfunktion) der TAN()-Funktion. Das Ergebnis wird jedoch nicht auf Grundlage des Tangens berechnet, sondern mithilfe einer XY-Koordinate (siehe Abbildung 15.10). Das Ergebnis wird als Bogenmaß ausgegeben. Soll der Arkustangens in Grad ausgedrückt werden, müssen Sie das jeweilige Ergebnis mit $180/\text{PI}()$ multiplizieren oder die GRAD()-Funktion verwenden.

Ein positives Ergebnis entspricht einem Winkel der, bezogen auf die x-Achse, gegen den Uhrzeigersinn abgemessen wird; ein negatives Ergebnis entspricht einem im Uhrzeigersinn abgemessenen Winkel.

Das Ergebnis der Formel $-\text{ARCTAN2}(a;b)$ entspricht dem Ergebnis von $\text{ARCTAN}(b/a)$ mit dem Unterschied, dass a in ARCTAN2() den Wert 0 annehmen darf.

Ist sowohl der Wert der *x_Koordinate* als auch der Wert der *y_Koordinate* gleich 0 (null), gibt die Funktion ARCTAN2() den Fehlerwert #DIV/0! zurück.

Der Graph der ARCTAN2()-Funktion entspricht dem der ARCTAN()-Funktion (siehe Abbildung 15.9 auf Seite 728).

Syntax

Definition

Argumente

Hintergrund

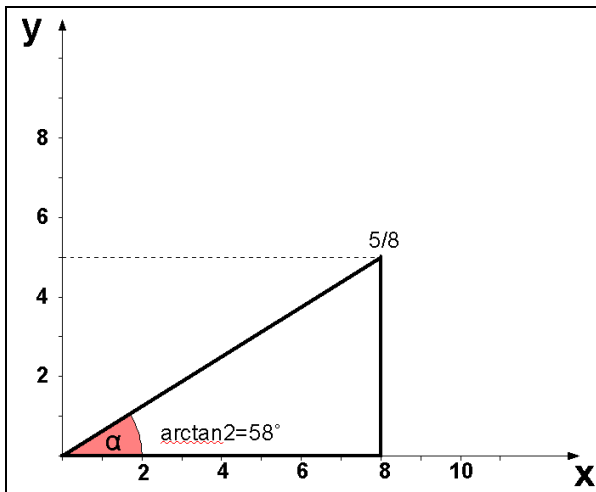


Abbildung 15.10: Darstellung der ARCTAN2-Funktion

Praxiseinsatz In der Praxis lässt sich die Funktion anwenden, wenn als Ergebnis einer Berechnung XY-Koordinaten ausgegeben werden und daraus der Winkel α berechnet werden soll.

=GRAD(ARCTAN2(1;2)) ergibt einen Winkel α von 63,435°.

=GRAD(ARCTAN2(-6;-2)) ergibt einen Winkel α von -161,565°.

=GRAD(ARCTAN2(11;2)) ergibt einen Winkel α von 10,305°.

Siehe auch ARCTANHYP(), ATAN(), PI(), TAN()



Diese Beispiele finden Sie im Ordner Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt ARCTAN2.

ARCTANHYP() ATANH()

Syntax ARCTANHYP(*Zahl*)

Definition ARCTANHYP() gibt den umgekehrten hyperbolischen Tangens (math.: *artanh*) einer Zahl zurück. Für *Zahl* sind nur Werte zulässig, die im Bereich $-1 < \text{Zahl} < +1$ liegen.

Argumente *Zahl* (erforderlich) kann jede beliebige reelle Zahl sein, die im Bereich $-1 < \text{Zahl} < +1$ liegt.

Hintergrund Die inversen Hyperbelfunktionen sind die Umkehrfunktionen zu den hyperbolischen Funktionen. Die mathematische Definition des Areatangens lautet

$$y = \operatorname{ar\,tanh} x = \frac{1}{2} \ln \frac{1+x}{1-x} \quad \text{mit } |x| < 1.$$

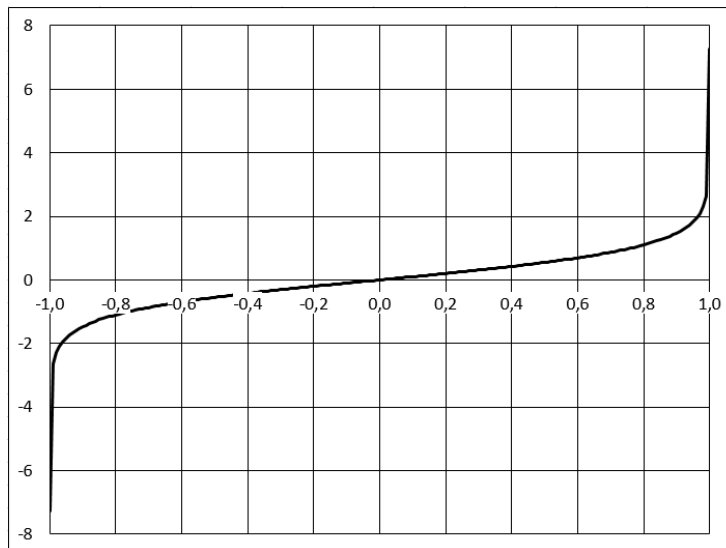


Abbildung 15.11: Die Funktion *Areatangens* ist ungerade und nur für $|x| < 1$ definiert

Die Umkehrfunktionen der hyperbolischen Funktionen tragen nicht den gleichen Präfix wie die Umkehrfunktionen der trigonometrischen Funktionen »Arkus«, sondern »Area«. Ihre korrekten Namen lauten also Areasinus, Areakosinus, Areatangens usw. Der Hinweis auf die Zugehörigkeit zu den Hyperbelfunktionen erübrigt sich, weil die Areafunktionen laut Definition deren Umkehrfunktionen sind. Der Name »Area« (lat. Fläche) leitet sich von der geometrischen Definition dieser Funktionen ab, nach der sie die Fläche eines Hyperbelsektors bezeichnen.

Hier einige Beispiele, um die Funktion zu dokumentieren:

=ARCTANHYP(0) ergibt 0.

=ARCTANHYP(0,5) ergibt 0,54930614.

=ARCTANHYP(-0,5) ergibt -0,54930614.

=ARCTANHYP(0,999) ergibt 3,80020117.

=ARCTANHYP(-0,999) ergibt -3,80020117.

ARCCOSHYP(), ARCSINHYP(), TANHYP()

Die Tabelle für die grafische Darstellung aus Abbildung 15.11 und die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *ARCTANHYP*.

Hinweis

Siehe auch



AUFRUNDEN() ROUNDUP()

AUFRUNDEN(*Zahl*; *Anzahl_Stellen*)

Diese Funktion rundet eine Zahl auf eine beliebige Anzahl an Stellen auf.

Zahl (erforderlich) ist eine reelle Zahl, die Sie aufrunden möchten.

Anzahl_Stellen (erforderlich) gibt an, auf wie viele Dezimalstellen die Zahl gerundet werden soll.

Im Gegensatz zur Funktion RUNDEN(), wo bei Zahlen < 5 ab- und ≥ 5 aufgerundet wird, wird beim AUFRUNDEN konsequent die vorstehende Ziffer aufgerundet. Mithilfe des Arguments *Anzahl_Stellen* wird die Anzahl der Dezimalstellen festgelegt, auf die das Argument *Zahl* gerundet werden soll.

Ist der in *Anzahl_Stellen* angegebene Wert:

- ▶ größer oder gleich 0 (null), wird die jeweilige Zahl entsprechend der angegebenen Anzahl von Dezimalstellen aufgerundet.
- ▶ gleich 0, wird Zahl auf die nächste ganze Zahl aufgerundet.
- ▶ kleiner 0, wird die jeweilige Zahl links vom Komma aufgerundet.

Sie möchten ein Zimmer tapezieren und es soll ermittelt werden, wie viele Rollen Tapete benötigt werden, um eine Fläche von 43 m^2 zu tapezieren. Die Länge einer Rolle Tapete beträgt 10,5 m, die Breite 0,53 m. Das Ergebnis lautet: 3 Rollen. Die entsprechende Formel sieht in Excel wie folgt aus:

=AUFRUNDEN(43/10,5*0,53;0)

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Zuerst wird die Anzahl der zu tapezierenden Fläche (43 m^2) durch die Größe der Tapetenrolle ($10,5 \text{ m} \times 0,53 \text{ m}$) geteilt. Das Ergebnis muss auf eine ganze Zahl aufgerundet werden, da man in der Regel ganze Rollen im Fachmarkt erhält. Wenn Sie wollen, entwickeln Sie dieses Beispiel weiter, indem Sie die Bahnlängen und geschätzten Verschnitt mit einbeziehen.

Hier weitere Beispiele, um die Funktion zu dokumentieren:

=AUFRUNDEN(1,9;0) ergibt einen Wert von 2.

=AUFRUNDEN(-2,8;0) ergibt einen Wert von -3.

=AUFRUNDEN(18;-1) ergibt einen Wert von 20.

Siehe auch ABRUNDEN(), GANZZAHL(), REST(), VRUNDEN(), OBERGRENZE.MATHEMATIK(), RUNDEN(), KÜRZEN(), UNTERGRENZE.MATHEMATIK()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *AUFRUNDEN*.

Neu in Excel
2013

BASIS() BASE()

Syntax BASIS(*Zahl*; *Basis*; *Mindestlänge*)

Definition Die Funktion wandelt eine *Zahl* in ihre Textdarstellung mit der angegebenen Basis um.

Argumente *Zahl* (erforderlich) ist die Zahl, die Sie umwandeln möchten. *Zahl* muss im Bereich von 0 bis $2^{53}-1$ liegen. Verwenden Sie nicht ganze Zahlen, werden die Nachkommastellen ignoriert.

Basis (erforderlich) darf im Bereich von 2 bis 36 liegen und ist die Basis des Zahlenformats, in das Sie *Zahl* umwandeln möchten.

Mindestlänge (optional) bestimmt die Anzahl der Zahlenstellen des Ergebnisses, also damit die Anzahl der führenden Nullen. Der angegebene Wert muss größer oder gleich null sein.

Hintergrund Diese Funktion ist das universelle Gegenstück zu den Umwandlungsfunktionen DEZINBIN(), DEZINHEX() und DEZINOKT() in der Kategorie »Konstruktion« (siehe Kapitel 16). Dort sind auch einige Zahlensysteme zum Teil ausführlich beschrieben.

Ist eine Basis größer als 10, reicht der Ziffernvorrat unseres Dezimalsystems nicht mehr zur Darstellung derartiger Zahlen aus und es werden Buchstaben zu Hilfe genommen, die dann höherwertige Ziffern repräsentieren (siehe Tabelle 15.2 und weitere Erklärungen auf Seite 742).

Praxiseinsatz =BASIS(12;2) ergibt 1100.

=BASIS(67;8) ergibt 103.

=BASIS($2^{53}-1$;16) ergibt 1FFFFFFFFFFFFFFF.

=BASIS(3,92;2;4) ergibt 0011

Dies entspricht =BASIS(3;2;4) (ohne Dezimalstellen).

=BASIS(2333;26;8) ergibt 000003BJ.

Siehe auch DEZINBIN(), DEZINHEX(), DEZINOKT()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *BASIS*.

BOGENMASS()



BOGENMASS(*Winkel*)

Diese Funktion wandelt Grad in Bogenmaß (Radiant) um.

Winkel (erforderlich) ist ein in Grad gegebener Winkel, den Sie umwandeln möchten.

Der Abschnitt eines Kreises lässt sich sowohl durch den Winkel (1 Vollkreis = 360°) – siehe die Funktion GRAD() – als auch durch das Verhältnis von Kreisumfang zu Kreisbogen bestimmen. Dieser Quotient wird als Bogenmaß bzw. Radiant bezeichnet.

Da Excel für die Berechnung der Winkelfunktionen vom Bogenmaß ausgeht, benötigen Sie die Funktion sehr häufig für die Angabe der Argumente in den Winkelfunktionen; siehe Funktionen COS(), SIN().

Es gilt: Kreisumfang ($2\pi r$) / Kreisbogen (b) = Vollwinkel (360°) / Zentrierwinkel (α)

Die Maßeinheit für das Bogenmaß ist der Radiant (kurz: rad). Der Vollumfang eines Kreises entspricht 360° oder auch als Bogenmaß $2\pi = 6,28318531$ rad. Ein Halbkreis entspricht $\pi = 3,14159265$ rad.

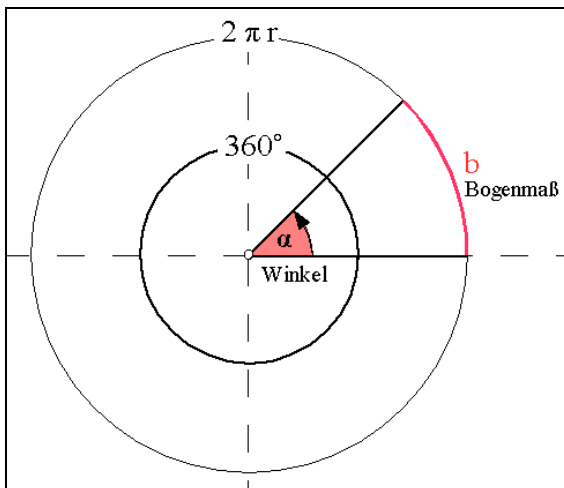


Abbildung 15.12: Darstellung des Bogenmaßes im Vollkreis

Eine interessante Anwendung ist die Maßeinheit »Seemeilen«. Der Erdumfang beträgt 360° mal 60', was 21.600 Bogenminuten oder Seemeilen entspricht. Der Erdumfang ist $360^\circ \cdot 60' = 21.600$ Bogenminuten oder Seemeilen.

Hier noch ein paar Beispiele, um die Funktion zu dokumentieren:

=BOGENMASS(1) ergibt 0,017453293 rad.

=BOGENMASS(45) ergibt 0,785398163 rad.

=BOGENMASS(90) ergibt 1,570796327 rad.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch GRAD()



Die oben genannten Beispielwerte finden Sie im Ordner `Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `BOGENMASS`.

COS() COS()

Syntax COS(Zahl)

Definition Gibt den Kosinus einer Zahl zurück.

Argumente Zahl (erforderlich) ist der im Bogenmaß angegebene Winkel, dessen Kosinus Sie berechnen möchten.

Hintergrund In einem rechtwinkligen Dreieck ist das Verhältnis von Ankathete zur Hypotenuse abhängig vom Öffnungswinkel. Das Verhältnis der Ankathete eines Winkels zur Hypotenuse ist der Kosinus (*cos*) eines Winkels.

$$\cos \alpha = \frac{b}{c}; \cos \beta = \frac{a}{c}$$

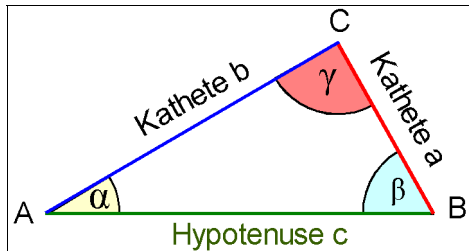


Abbildung 15.13: Seitenverhältnisse im rechtwinkligen Dreieck

Bei der Darstellung der Abhängigkeit des Kosinus vom Winkel α in einem Einheitskreis ($c = 1$) nimmt der Winkel α von 0° bis 90° zu, so nimmt der Kosinus von 1 bis 0 ab.

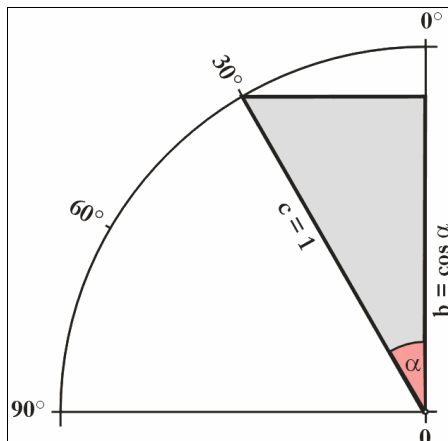


Abbildung 15.14: Kosinus im Einheitskreis

Beim Übertrag der Werte in ein Koordinatensystem, mit den Werten des Winkels α auf der x-Achse und dem Sinus von α auf y-Achse, ergibt sich folgende Kurve für den gesamten Kreis.

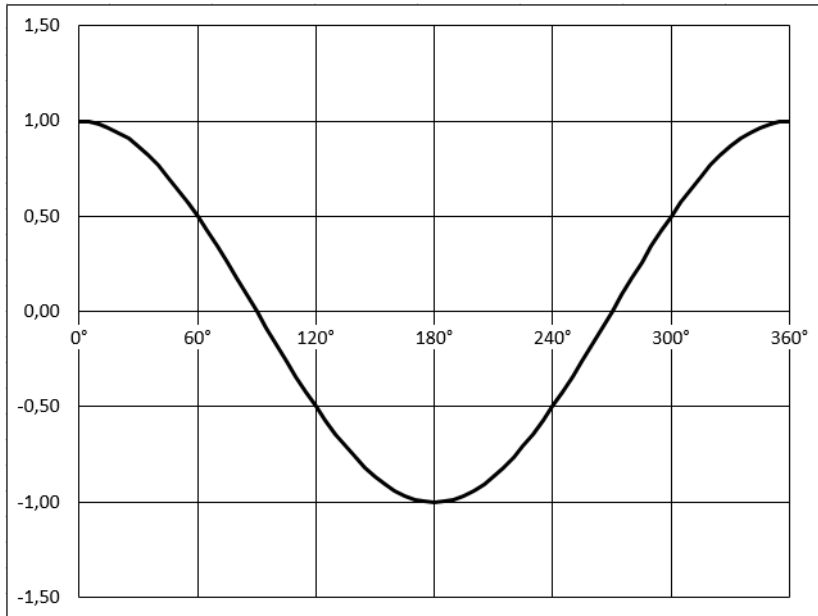


Abbildung 15.15: Die Kosinuskurve

Eine ICE-Strecke muss über eine Brücke geführt werden. Berechnen Sie die Länge der zu einer Brücke ansteigenden Strecke (s), wenn man 1.500 m (l) vor der Brücke mit der Steigung beginnt und der Steigungswinkel α 1° betragen soll.

Praxiseinsatz

Mit der Umstellung der Formel $\cos \alpha = \text{Ankathete } (l) / \text{Hypotenuse } (s)$ nach der Strecke s :

$$s = l / \cos \alpha$$

erhalten Sie die Berechnungsformel. In Excel stellt sich die Formel so dar:

$$=1500/\text{COS}(\text{BOGENMASS}(1))$$

Die Berechnung ergibt eine Steigungstrecke von 1.500,23 m, also eine um ganze 23 cm längere Strecke als die Entfernung selbst.

ARCCOS(), COSHYP(), PI(), SIN()

Siehe auch

Die Tabelle für die grafische Darstellung aus Abbildung 15.15 und das Beispiel finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `COS`. Das ICE-Beispiel haben wir auf dem Arbeitsblatt `trigon` in derselben Arbeitsmappe dargestellt.



Neu in Excel
2013

COSEC()



Syntax COSEC(*Zahl*)

Definition Die Funktion COSEC() berechnet den Kosekans des im Bogenmaß angegebenen Winkels.

Argumente *Zahl* (erforderlich) ist der Winkel, angegeben im Bogenmaß, dessen Kosekans berechnet werden soll. Der absolute Wert darf maximal 134.217.727 ($2^{27}-1$) groß sein.

Hintergrund In einem rechtwinkligen Dreieck ist das Verhältnis von Hypotenuse zur Ankathete abhängig vom Öffnungswinkel. Das Verhältnis der Hypotenuse eines Winkels zur Ankathete ist der Kosekans (*csc*) eines Winkels und damit der Kehrwert der Sinusfunktion.

$$\text{csc } \alpha = \frac{c}{a} \quad \dots \quad \text{csc } \beta = \frac{c}{b}$$

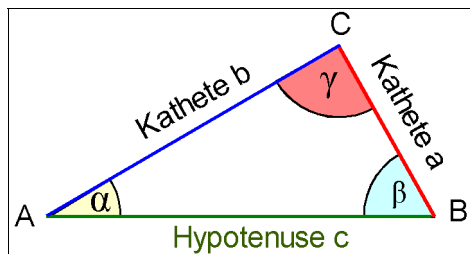


Abbildung 15.16: Seitenverhältnisse im rechtwinkligen Dreieck

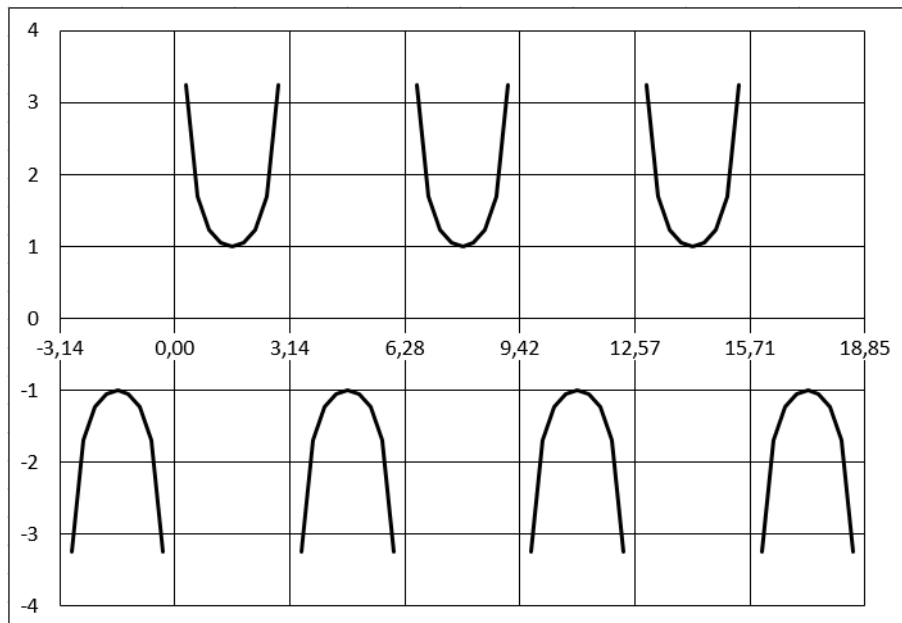


Abbildung 15.17: Graph des Kosekans

=COSEC(0,5) ergibt 2,085829643.

=COSEC(PI()/2) ergibt 1.

COS(), SEC(), SIN()

Die Tabelle für die grafische Darstellung aus Abbildung 15.17 und das Beispiel finden Sie im Ordner \\Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *COSEC*.

Praxiseinsatz

Siehe auch



COSECHYP()



COSECHYP(*Zahl*)

Die Funktion COSECHYP() gibt den hyperbolischen Kosekans (Kosekans Hyperbolicus, Hyperbelkosekans, math. *csch*) einer reellen Zahl zurück.

Zahl (erforderlich) ist der Winkel, angegeben im Bogenmaß, dessen Kosekans Hyperbolicus berechnet werden soll.

Die Funktion *csch* (Kosekans Hyperbolicus) gehört zu den Hyperbelfunktionen und stellt die Kehrwertfunktion des Sinus Hyperbolicus dar. Sie ist definiert als

$$\operatorname{csch} x = \frac{1}{\sinh x} = \frac{2}{e^x - e^{-x}}$$

**Neu in Excel
2013**

Syntax

Definition

Argumente

Hintergrund

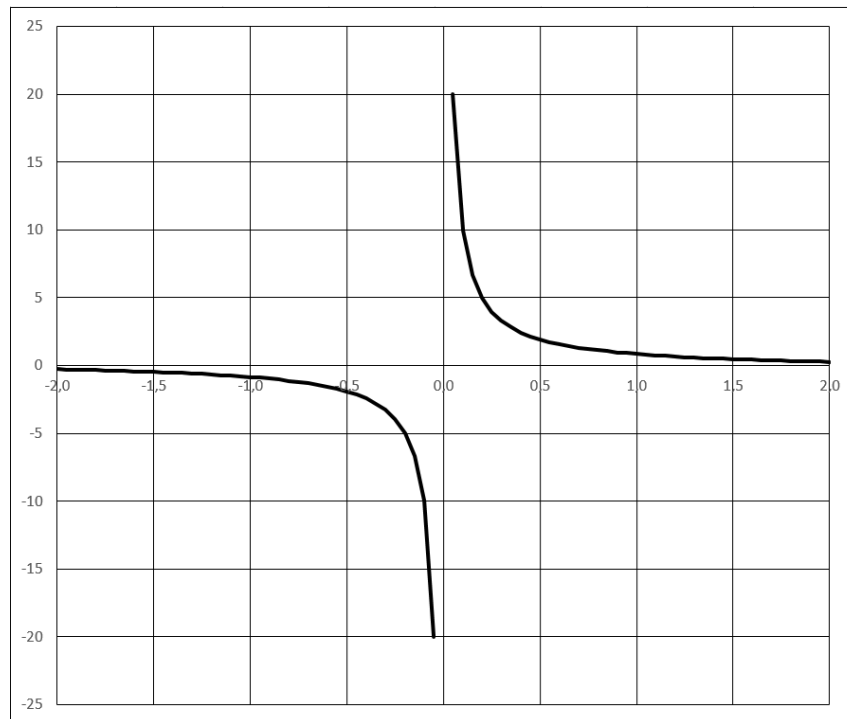


Abbildung 15.18: Darstellung der COSECHYP()-Funktion

Praxiseinsatz =COSECHYP(0,001) ergibt 999,9998333.
 =COSECHYP(0,5) ergibt 1,919034751.
 =COSECHYP(PI()/2) ergibt 0,434537208.

Siehe auch COSEC(), SECHYP(), SIN(), SINHYP()



Die Tabelle für die grafische Darstellung aus Abbildung 15.18 und die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *COSECHYP*.

COSHYP()



Syntax COSHYP(*Zahl*)

Definition Die Funktion COSHYP() gibt den hyperbolischen Kosinus (Hyperbelkosinus, Kosinus Hyperbolicus, math. *cosh*) einer reellen Zahl zurück.

Argumente *Zahl* (erforderlich) ist der Winkel, angegeben im Bogenmaß, dessen Kosinus Hyperbolicus berechnet werden soll.

Hintergrund Der hyperbolische Kosinus gehört zu den Hyperbelfunktionen, die, ebenso wie die Kreisfunktionen (z.B. Sinus und Tangens), für alle reellen und komplexen Zahlen definiert sind.

Microsoft Excel erlaubt aber nur reelle Argumente für die Hyperbelfunktionen in der Kategorie »Mathematik & Trigonometrie«. Für komplexe Argumente benutzen Sie bitte die Funktionen in der Kategorie »Konstruktion«, z.B. IMCOSHYP() (siehe Kapitel 16).

Die Definition für den hyperbolischen Kosinus lautet:

$$\cosh x = \frac{e^x + e^{-x}}{2}$$

Seine grafische Darstellung zeigt die folgende Abbildung 15.19:

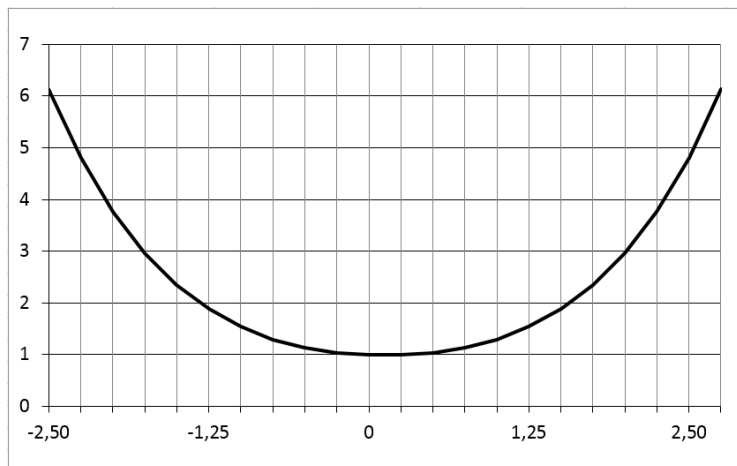


Abbildung 15.19: Der Hyperbelkosinus ist eine gerade Funktion, die symmetrisch zur y-Achse liegt, weil sie zwei Monotonieintervalle besitzt

Der hyperbolische Kosinus wird in der Technik und in den Naturwissenschaften, vor allem in der Forschung und Entwicklung, häufig gebraucht. Seinen größten Bekanntheitsgrad hat der Hyperbelkosinus als »Kettenlinie« oder »Katenoide«. Sie beschreibt die Form einer an zwei Punkten befestigten, frei hängenden Kette. Die Gleichung der Kettenlinie lautet:

$$y = a \cosh \frac{x}{a}$$

mit a als Abstand des Scheitelpunkts von der Nulllinie (des tiefsten Punkts).

ARCCOSHYP(), SINHYP(), TANHYP()

Die Tabelle für die grafische Darstellung aus Abbildung 15.19 und das Beispiel »Kettenlinie« finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe *COSHYP.xlsx* auf den Arbeitsblättern *Graph* und *Kettenlinie*.

Praxiseinsatz

Siehe auch



COT() COT()

COT(*Zahl*)

Die Funktion COT() gibt den Kotangens eines Winkels zurück.

Zahl (erforderlich) ist der Winkel, angegeben im Bogenmaß, dessen Kotangens berechnet werden soll, und kann jede beliebige reelle Zahl mit einem Betrag kleiner 2^{27} außer 0 sein; an den übrigen Unstetigkeitsstellen gibt Excel sehr große bzw. sehr kleine Werte aus (s. Abbildung 15.21, das letzte Beispiel bei *Praxiseinsatz* und die Erklärung im Achtung-Einschub danach).

Die Funktion COT(), math. *cot*, gehört zu den trigonometrischen Funktionen.

In einem rechtwinkligen Dreieck ist der Kotangens das Verhältnis Ankathete zu Gegenkathete. Es gilt:

$$\cot \alpha = \frac{b}{a} = \frac{\cos \alpha}{\sin \alpha} = \frac{1}{\tan \alpha}$$

Der Kotangens ist die Kehrwertfunktion des Tangens.

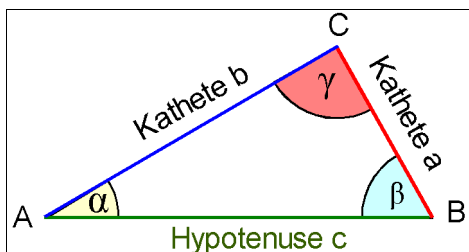


Abbildung 15.20: Das rechtwinklige Dreieck mit den Bezeichnungen der Seiten und Winkel

Neu in Excel 2013

Syntax

Definition

Argumente

Hintergrund

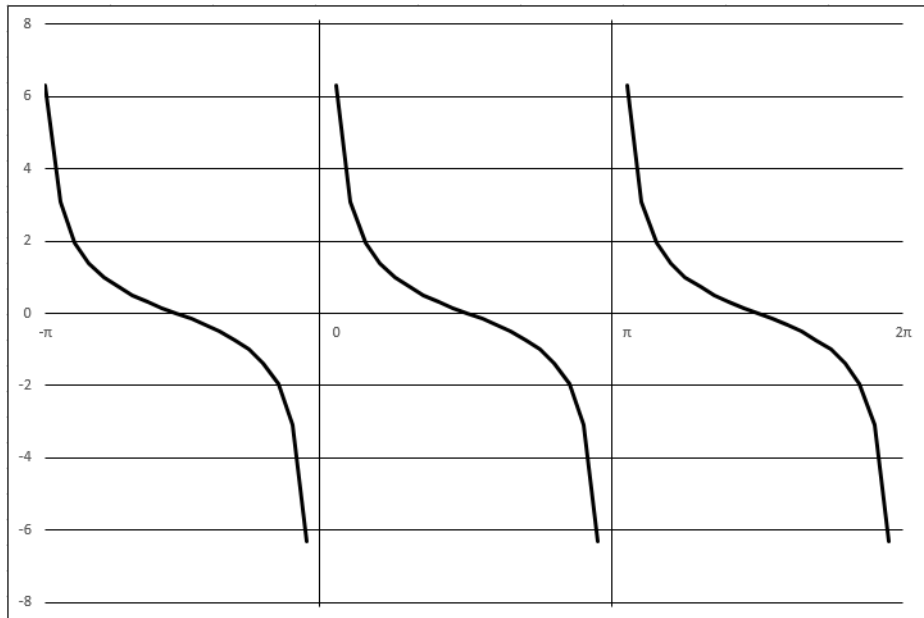


Abbildung 15.21: Der Graph der Kotangensfunktion

Praxiseinsatz =COT(1) ergibt 0,642092616.
 =COT(2) ergibt -0,457657554.
 =COT(PI()) ergibt -8,16228E+15.

Achtung Der Kotangens ist für alle reellen Zahlen mit Ausnahme ganzzahliger Vielfache von π definiert. Abgesehen von Null akzeptiert Excel jedoch ganzzahlige Vielfache von π wegen der technisch bedingten Rundungsfehler. Der Rückgabewert der COT()-Funktion ist daher kritisch zu prüfen: Er könnte in Ihren Berechnungen Fehler verursachen.

Siehe auch COS(), COSEC(), SEC(), SIN(), TAN()



Die Tabelle für die grafische Darstellung aus Abbildung 15.21 und die Beispiele finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `COT`.

Neu in Excel
2013

COTHYP()

 **COTH()**

Syntax COTHYP(*Zahl*)

Definition Die Funktion COTHYP() gibt den hyperbolischen Kotangens (Hyperbelkotangens, Kotangens Hyperbolicus, math. *coth*) einer reellen Zahl zurück.

Argumente *Zahl* (erforderlich) ist der Winkel, angegeben im Bogenmaß, dessen Kotangens Hyperbolicus berechnet werden soll, und kann jede beliebige reelle Zahl außer 0 sein.

Der hyperbolische Kotangens \coth ist die Kehrwertfunktion des hyperbolischen Tangens \tanh ; seine Definition lautet:

$$\coth x = \frac{1}{\tanh x} = \frac{e^x + e^{-x}}{e^x - e^{-x}}$$

Der hyperbolische Kotangens gehört zu den Hyperbelfunktionen, die, ebenso wie die Kreisfunktionen (z.B. Sinus und Tangens), für alle reellen und komplexen Zahlen definiert sind.

Microsoft Excel erlaubt aber nur reelle Argumente für die Hyperbelfunktionen in der Kategorie »Mathematik & Trigonometrie«. Für komplexe Argumente benutzen Sie bitte die Funktionen in der Kategorie »Konstruktion«, z.B. IMCOSHYP() (siehe Kapitel 16). Ein direktes komplexes Gegenstück zur COTHYP()-Funktion gibt es nicht. Man muss dann über den folgenden Zusammenhang auf die Funktionen IMCOSHYP() und IMSINHYP() zurückgreifen:

$$\coth z = \frac{\cosh z}{\sinh z}$$

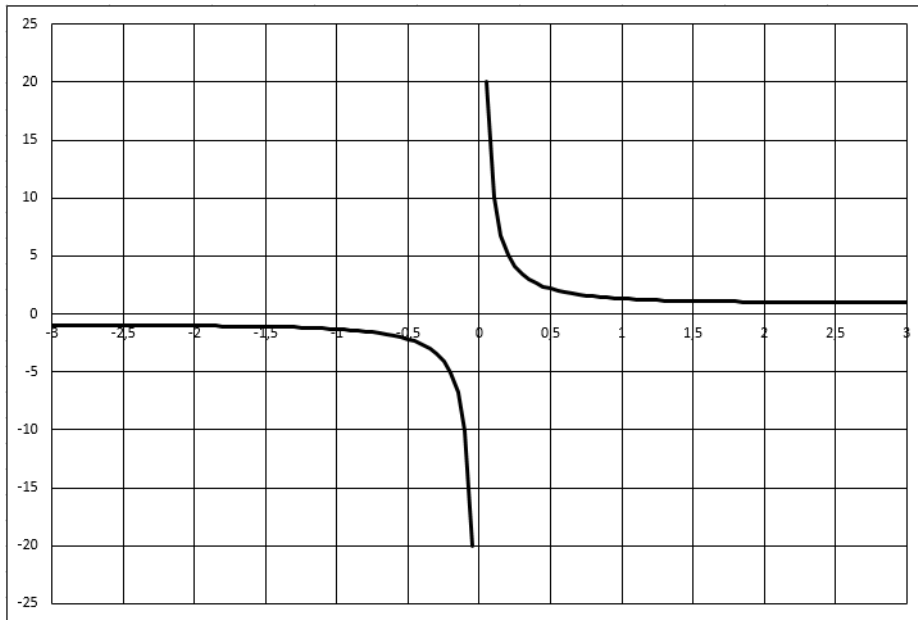


Abbildung 15.22: Der Graph der COTHYP()-Funktion, der sich für große Argumente asymptotisch +1 bzw. -1 nähert

Der hyperbolische Kotangens wird in der Physik überraschend häufig gebraucht, so beispielsweise, um die Zeitfunktion der Geschwindigkeit bei einem Wurf nach unten zu beschreiben, wenn Newton-Reibung berücksichtigt wird.

=COTHYP(1) ergibt 1,313035285.

=COTHYP(-1) ergibt -1,313035285.

=COTHYP(PI()) ergibt 1,003741873.

Siehe auch COSECHYP(), COSHYP(), SECHYP(), SINHYP(), TANHYP()



Die Tabelle für die grafische Darstellung aus Abbildung 15.22 und die Beispiele finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `COTHYP`.

Neu in Excel
2013

DEZIMAL()



DECIMAL()

Syntax DEZIMAL(*Text*; *Basis*)

Definition Die Funktion gibt die dezimale Darstellung einer Zahl mit einer beliebigen Basis zurück.

Argumente *Text* (erforderlich) ist eine maximal 255 Zeichen lange Textdarstellung einer ganzen Zahl, die mit Anführungszeichen eingegeben werden muss. Groß- oder Kleinschreibung ist nicht zu beachten.

Basis (erforderlich) ist eine ganze Zahl von 2 bis 36, die die Basis der mit *Text* dargestellten Zahl angibt.

Hintergrund Die Funktion DEZIMAL() ist die Umkehrfunktion der Funktion BASIS() auf Seite 732. Eine im Textformat vorliegende Zahl zu einer beliebigen Basis (≤ 36) können Sie mit dieser Funktion in das Dezimalsystem konvertieren.

Ist eine Basis größer als 10, reicht der Ziffernvorrat unseres Dezimalsystems nicht mehr zur Darstellung derartiger Zahlen aus und es werden Buchstaben zu Hilfe genommen, die dann höherwertige Ziffern repräsentieren.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	G	H	I
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	10

Tabelle 15.2: Gegenüberstellung von Dezimalzahlen (1. und 3. Reihe) und Zahlen mit einer Basis, die größer als 17 ist (2. und 4. Reihe)

Sobald die einstelligen Zeichen für eine Basis ausgeschöpft sind (bei 36 auf jeden Fall), müssen größere Zahlen mit zwei oder mehr Zeichen gebildet werden. Jede Stelle steht für eine Potenz der Basis, die mit dem jeweiligen Ziffernwert dieser Stelle multipliziert wird. Die Summation über alle Stellen ergibt die Zahl.

Derartige Zahlensysteme nennt man polyadisch, ein nicht-polyadisches Zahlensystem ist z.B. das römische Zahlensystem. Das Bildungsgesetz für polyadische Zahlensysteme lautet

$$a = \sum_{i=0}^n z_i B^i \quad (n \geq 0, n \text{ ganz})$$

mit B als Basis, n als Anzahl der Stellen minus 1 und z_i als Ziffern der Zahl.

So ergibt die Zahl 10 zur Basis 36 die Rechenvorschrift $a = 1 \cdot 36^1 + 0 \cdot 36^0 = 36$

=DEZIMAL("101010";2) ergibt 42.

=DEZIMAL("1FFFFFFFFFFFF";16) ergibt 9,00719925E+15 (= $2^{53}-1$).

=DEZIMAL("777";8) ergibt 511.

=DEZIMAL("777";16) ergibt 1911.

=DEZIMAL("H";20) ergibt 17 (siehe Tabelle 15.2).

ARABISCH(), BASIS(), RÖMISCH()

Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *DEZIMAL*.

Praxiseinsatz

Siehe auch



EXP()



EXP()

EXP(*Zahl*)

Die Funktion EXP() potenziert die Basis e mit der als Argument angegebenen *Zahl*.

Zahl (erforderlich) ist der Exponent zur Basis e

Das Potenzieren gehört, wie das Radizieren und Logarithmieren, zu den Rechenarten der dritten Stufe. Das Potenzieren entsteht für natürliche Zahlen genauso aus dem Multiplizieren wie das Multiplizieren aus dem Addieren. Die Funktion EXP() verwendet für das Potenzieren als Basis die transzendent-irrationale Eulersche Zahl e. Diese Konstante ist die Basis des natürlichen Logarithmus und hat den Wert $2,71828182845904$.

Diese, nach dem Schweizer Mathematiker Leonhard Euler benannte Zahl, ist eine irrationale und transzendente Zahl, d.h. wie die Kreiszahl Pi, siehe auch Funktion PI(), lässt sie sich weder als Bruch zweier natürlicher Zahlen noch als Lösung einer algebraischen Gleichung endlichen Grades darstellen. Die Eulersche Zahl spielt in der Differenzial- und Integralrechnung eine wichtige Rolle.

EXP() ist die Umkehrfunktion zu LN(), welche den natürlichen Logarithmus zurückgibt, und rechnet nach dieser Formel:

$$x = e^{\text{Zahl}}, \text{ aufgelöst als Produkt: } x = e_{\text{Zahl}} \cdot e_{\text{Zahl}-1} \cdot \dots \cdot e_2 \cdot e_1$$

Um eine Potenz mit einer anderen Basis zu berechnen, müssen Sie den Potenzierungsoperator (^) verwenden. Die achte Potenz von 2 (Basis) ermitteln Sie in Excel so:

=2^8, was 256 ergibt.

Mehr zu den Logarithmensystemen können Sie unter der Funktion LN() nachlesen.

Die Exponentialfunktion findet in der Praxis neben der Wahrscheinlichkeitsberechnung (Stochastik) z.B. auch Anwendung in der Physik bei der Berechnung des radioaktiven Zerfalls oder in der Biologie bei der Berechnung von »exponentiellem« Wachstum von Organismen.

Weitere Informationen würden den Rahmen dieses Buchs sprengen. Wenden Sie sich bitte der entsprechenden Spezialliteratur zu.

Syntax

Definition

Argumente

Hintergrund

Hinweis

Praxiseinsatz

Hier noch zwei Beispiele, um die Funktion zu dokumentieren:

=EXP(1) ergibt den Wert 2,71828183

=EXP(2) ergibt den Wert 7,3890561

Siehe auch IMEXP(), LN(), LOG(), POTENZ()



Diese Beispiele und den Graphen der Exponentialfunktion finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *EXP*.

FAKULTÄT() FACT()

Syntax FAKULTÄT(*Zahl*)

Definition Diese Funktion gibt die Fakultät zu *Zahl* mit einer Schrittlänge von 1 zurück.

Argumente *Zahl* (erforderlich) ist die nicht negative Zahl, deren Fakultät Sie berechnen möchten. Ist *Zahl* keine ganze Zahl, werden die Nachkommastellen abgeschnitten und nicht gerundet.

Hintergrund Die Fakultät ist eine in der Mathematik immer dann verwendete Funktion, wenn es um Produktfolgen, hier speziell die »Partialproduktfolge der Folge der natürlichen Zahlen«, geht. Die Fakultät einer Zahl ergibt sich aus dem Produkt aller natürlichen Zahlen von 1 bis zum Argument *Zahl*.

Für die Zahl 4 wird beispielsweise die Fakultät wie folgt ermittelt:

$$4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$$

Die mathematische Definition lautet:

$$n! = n \cdot (n-1) \cdot (n-2) \cdot \dots \cdot 3 \cdot 2 \cdot 1$$

Zweckmäßigerweise wird definiert: $0! = 1$.

Praxiseinsatz In der abzählenden Kombinatorik spielen Fakultäten eine wichtige Rolle, weil $n!$ die Anzahl der Möglichkeiten ist, n unterscheidbare Gegenstände in einer Reihe anzuordnen.

Beispiel: Bei einem Marathon starten 4 Läufer. Wie viele Möglichkeiten gibt es für die Reihenfolge beim Zieleinlauf dieser Läufer, wenn alle Läufer das Ziel erreichen?

Lösung: Für den ersten Platz kommen alle 4 Läufer in Frage. Ist der erste Läufer angekommen, können nur noch 3 Läufer um den zweiten Platz konkurrieren. Ist auch der zweite Platz vergeben, kommen für den 3. Platz nur noch 2 Läufer in Frage, für den 4. und letzten Platz dann nur noch einer. Es gibt also $4! = 24$ verschiedene Ranglisten für den Zieleinlauf.

Die Formel in Excel lautet =Fakultät(4)

Siehe auch ZWEIFAKULTÄT(), PRODUKT()



Eine Beispielreihe finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *FAKULTÄT*.

GANZZAHL() INT()

GANZZAHL(*Zahl*)

Die Funktion rundet das Argument *Zahl* auf die nächstkleinere ganze Zahl ab.

Zahl (erforderlich) ist die reelle Zahl, die Sie auf die nächste ganze Zahl abrunden möchten.

Zu den reellen Zahlen zählen alle Dezimalzahlen, d.h. alle Zahlen, die sich durch die Ziffern 1,2, ...9,0 sowie dem »«,« (Komma) und den Vorzeichen »+« und »-« darstellen lassen. Beim Zahlenbereich der »Reellen Zahlen« handelt es sich um die Erweiterung des Bereichs der »Rationalen Zahlen« um die »Irrationalen Zahlen«. Diese Erweiterung ist notwendig, um die Rechenoperationen der dritten Stufe – Radizieren, Logarithmieren und Potenzieren – uneingeschränkt im Bereich der absolut rationalen Zahlen ausführbar zu machen.

»Ganze Zahlen« gehören zu den rationalen Zahlen. Zu ihnen gehören alle positiven und negativen Zahlen sowie die Zahl 0 (null). Im Bereich der positiven Zahlen schneidet die Funktion GANZZAHL() die Nachkommastellen ab, bei den negativen Zahlen rundet sie auf die nächstkleinere Zahl ab.

Bei der Berechnung der Steuererklärung soll das Ergebnis in ganzen Eurobeträgen erscheinen und (zu Ihren Gunsten) abgerundet werden.

Hier weitere Beispiele, um die Funktion zu dokumentieren:

=GANZZAHL(4,3) ergibt einen Wert von 4.

=GANZZAHL(-2,51) ergibt einen Wert von -3.

=GANZZAHL(78,8) ergibt einen Wert von 78.

ABRUNDEN(), AUFRUNDEN(), REST(), VRUNDEN(), OBERGRENZE.MATHEMATIK()
RUNDEN(), KÜRZEN(), UNTERGRENZE.MATHEMATIK()

Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *GANZZAHL*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



GERADE() EVEN()

GERADE(*Zahl*)

Rundet eine Zahl auf die nächste gerade ganze Zahl auf.

Zahl (erforderlich) ist der Wert, der auf die nächste gerade Zahl aufgerundet werden soll.

Im Gegensatz zur Funktion RUNDEN(), wo bei Zahlen < 5 ab- und ≥ 5 aufgerundet wird, rundet die Funktion GERADE() konsequent auf die nächste gerade ganze Zahl auf. Unabhängig davon, welches Vorzeichen *Zahl* hat, wird ein Wert aufgerundet, wenn die Anpassung von null ausgehend erfolgt. Ist *Zahl* eine gerade ganze Zahl, erfolgt keine Rundung.

Als gerade Zahl werden alle Zahlen bezeichnet, die ganzzahlig durch 2 teilbar sind. Sie können diese Funktion für die Verarbeitung von Elementen verwenden, die paarweise auftreten.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz Zwei Erben wollen Ihre Erbschaft zusammenrechnen. Um die Aufteilung zu vereinfachen, sollen alle Beträge durch zwei teilbar sein.

Hier einige Beispiele, um die Funktion zu dokumentieren:

=GERADE(4,5) ergibt einen Wert von 6.

=GERADE(-6,3) ergibt einen Wert von -8.

=GERADE(59959) ergibt einen Wert von 59.960.

Siehe auch GANZZAHL(), ISTGERADE() (Informationsfunktion), ISTUNGERADE() (Informationsfunktion), OBERGRENZE(), OBERGRENZE.GENAU(), RUNDEN(), KÜRZEN(), UNGERADE(), UNTERGRENZE(), UNTERGRENZE.GENAU()



Die Beispiele finden Sie im Ordner `Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `GERADE`.

GGT() GCD()

Syntax GGT(*Zahl1*; *Zahl2*; ...)

Definition Gibt den größten gemeinsamen Teiler für die Wertereihe *Zahl1*; *Zahl2*; ... zurück.

Argumente *Zahl1*; *Zahl2*; ... sind 1 bis 255 ganze Zahlen (erforderlich ist mindestens eine Zahl).

Hintergrund Der größte gemeinsame Teiler (ggT) spielt in der Mathematik eine große Rolle, z.B. beim Kürzen von Brüchen. Der größte gemeinsame Teiler ist die ganze Zahl, durch die sowohl *Zahl1* als auch *Zahl2* usw. dividiert werden können – ohne dass ein Rest bleibt.

Excel berechnet den ggT auf der Grundlage von ganzen Zahlen, Werte nach einem Komma werden abgeschnitten. Wird in die Formel ein nicht numerischer Ausdruck mit einbezogen, gibt GGT() den Fehlerwert `#WERT!` zurück. Ist eines der Argumente kleiner als 0 (null), gibt GGT() den Fehlerwert `#ZAHL!` zurück.

Es gibt zwei Möglichkeiten zur Berechnung des ggT:

1. Die Primfaktorenzerlegung

Der ggT ist das Produkt aus den Primfaktoren, die in allen Zahlen gemeinsam vorkommen. Diese werden dabei jeweils nur einmal als Faktor verwendet. Der ggT lässt sich also über die Primfaktorzerlegung von *m* und *n* berechnen.

Beispiel:

$$m = 36 = 2^2 \cdot 3^2$$

$$n = 120 = 2^2 \cdot 3 \cdot 5$$

Anschließend werden die Faktoren, die in beiden Zahlen vorkommen, miteinander multipliziert:

2. Euklidischer Algorithmus (Euklid, griechischer Mathematiker um 325 v.u.Z.)

Diese Methode ist nur für die Ermittlung des ggT von zwei Zahlen geeignet. Man dividiert die größere der beiden Zahlen, von denen man den ggT ermitteln möchte, durch die kleinere. Anschließend wird der Divisor durch den verbleibenden Rest dividiert. Dieser Vorgang wird wiederholt, bis die Division aufgeht. Der letzte Divisor ist der ggT der beiden Ausgangszahlen.

Aus dem ggT sowie dem Produkt von m und n lässt sich auch das kgV (kleinstes gemeinsames Vielfaches) – siehe Funktion KGV() – bestimmen:

$$\text{kgV}(m, n) = \frac{m \cdot n}{\text{ggT}(m, n)}$$

Ein Zimmer der Länge 7,65 m und der Breite 3,60 m soll mit möglichst großen quadratischen Platten ausgelegt werden. Welche Seitenlänge müssen die Platten haben?

=GGT(765;360) ergibt 45, d.h. die Seitenlänge einer Platte sollte 45 cm betragen. Weil diese Funktion nur mit ganzen Zahlen rechnet, muss man die Werte in cm als Argumente angeben.

KGV()

Die Beispiele finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `GGT`.

Hinweis

Praxiseinsatz

Siehe auch



GRAD() DEGREES()

GRAD(*Winkel*)

Die Funktion wandelt Bogenmaße (Radian) in Gradzahlen um.

Winkel (erforderlich) ist ein in Bogenmaß (Radian) angegebener Winkel, den Sie umwandeln möchten.

Als Grad (°) wird die Maßeinheit für einen Winkel bezeichnet. Ein Vollkreis entspricht einem Winkel von 360°. Unterteilt wird ein Grad in 60 (Winkel-)Minuten ('), wobei eine Minute in 60 (Winkel-)Sekunden (") eingeteilt wird.

In der Mathematik und Physik wird der Winkel auch als Bogenmaß angegeben. Dabei entspricht der Vollkreis einem Maß von 2π . Ein Winkel von π entspricht 180°.

In Excel spielt diese Funktion eine besondere Rolle. Da alle Formeln der Winkelfunktionen mit dem Bogenmaß rechnen, müssen die Ergebnisse bei Bedarf in Grad umgerechnet werden – siehe die Funktionen ARCCOS() und ARCSIN().

Bei Kreisberechnungen werden die Ergebnisse meistens als Bogenmaß ausgegeben. Siehe zum Beispiel die Funktion ARCTAN() oder ARCTAN2(). Mit der GRAD()-Funktion wird das Ergebnis in Grad umgerechnet z.B.:

=GRAD(ARCCOS(7,5/15))

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Hier weitere Beispiele, um die Funktion zu dokumentieren:

=GRAD(1) ergibt 57,296 Grad.

=GRAD(0,785) ergibt 45 Grad.

=GRAD(1,570796327) ergibt 90 Grad.

=GRAD(PI()/2) ergibt 90 Grad.

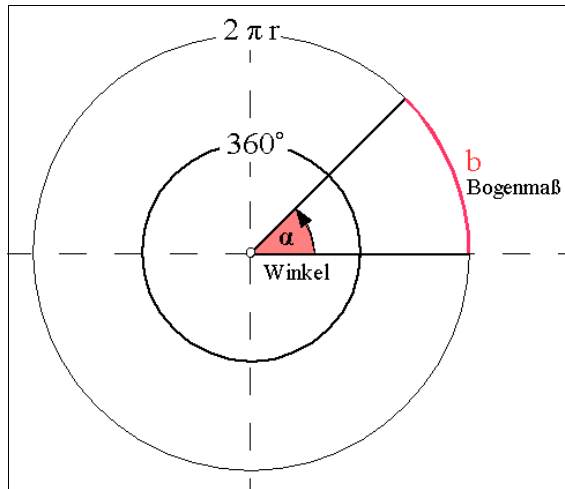


Abbildung 15.23: Gegenüberstellung von Winkel und Bogenmaß

Siehe auch BOGENMASS()



Die Beispiele finden Sie im Ordner `Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `GRAD`.

ISO.OBERGRENZE()



ISO.CEILING()

Syntax ISO.OBERGRENZE(*Zahl*; *Schritt*)

Definition Rundet eine Zahl auf die nächste ganze Zahl oder das nächste Vielfache von *Schritt* auf. Die Zahl wird unabhängig vom Vorzeichen aufgerundet.

Argumente *Zahl* (erforderlich) ist der Wert, den Sie runden möchten.

Schritt (optional) ist das Vielfache, auf das Sie runden möchten.

Wenn *Schritt* ausgelassen wird, ist der Standardwert 1.

Hintergrund Die Funktion ISO.OBERGRENZE() liefert die gleichen Ergebnisse wie die Funktionen OBERGRENZE.GENAU() und OBERGRENZE.MATHEMATIK() bei gleicher Argumentübergabe.

Sie ist bei Excel 2010 aus Kompatibilitätsgründen eingeführt worden, wird aber wohl in Zukunft durch die einfacher einzusetzende Funktion OBERGRENZE.MATHEMATIK() auf Seite 761 ersetzt werden. Lesen Sie bitte dort weitere Erklärungen.

Die Funktion ISO.OBERGRENZE() ist weder in den Kategorien des Menübands noch in den Listen des Funktions-Assistenten aufgeführt. Sie müssen sie händisch mit dem kompletten Namen in die Formel eingeben. Auch das Listenfeld, das beim AutoVervollständigen für Formeln erscheint, weist diese Funktion nicht aus.

Hinweis

KGV() LCM()

KGV(*Zahl1*; *Zahl2*; ...)

Syntax

Die Funktion gibt das kleinste gemeinsame Vielfache (kgV) der als Argumente angegebenen ganzen Zahlen zurück. Als kgV wird die kleinste positive ganze Zahl bezeichnet, die ein Vielfaches aller ganzzahligen Argumente *Zahl1*, *Zahl2* usw. ist. KGV() können Sie verwenden, wenn Sie Brüche addieren müssen, die unterschiedliche Nenner haben.

Definition

Zahl1; *Zahl2*; ... sind 1 bis 255 ganze Zahlen, deren kleinstes gemeinsames Vielfaches Sie berechnen möchten (erforderlich ist mindestens eine Zahl).

Argumente

Das kgV spielt in der Mathematik eine große Rolle, z.B. beim Addieren von Brüchen. Das kgV ist die kleinste ganze Zahl, die gleichzeitig durch *Zahl1* (*m*) und *Zahl2* (*n*) ohne Rest teilbar ist.

Hintergrund

Excel berechnet das kgV auf der Grundlage von ganzen Zahlen. Werte nach einem Komma werden abgeschnitten. Wird in die Formel ein nicht numerischer Ausdruck mit einbezogen, gibt KGV() den Fehlerwert #WERT! zurück. Ist eines der Argumente kleiner als 0 (null), gibt KGV() den Fehlerwert #ZAHL! zurück.

Das kgV lässt sich ebenso wie der ggT über die Primfaktorenzerlegung von *m* und *n* berechnen.

Beispiel:

$$m = 36 = 2^2 \cdot 3^2$$

$$n = 120 = 2^3 \cdot 3 \cdot 5$$

Anschließend werden sämtliche vorkommenden Primfaktoren mit der jeweils höchsten Potenz miteinander multipliziert:

$$2^3 \cdot 3^2 \cdot 5 = 360$$

Aus dem kgV sowie dem Produkt von *m* und *n* lässt sich der ggT – siehe Funktion GGT() – bestimmen:

Hinweis

$$ggT(m, n) = \frac{m \cdot n}{kgV(m, n)}$$

Zwei Menschen laufen nebeneinander. Läufer 1 hat eine Schrittlänge von 1,10 m, Läufer 2 hat eine Schrittlänge von 1,43 m. Sie laufen gleichzeitig los. Nach welcher Wegstrecke treten beide zum ersten Mal wieder gleichzeitig auf?

Praxiseinsatz

=KGV(110;143) ergibt 1430, d.h. nach 14,30 m treten beide wieder gleichzeitig auf. Weil diese Funktion nur mit ganzen Zahlen rechnet, muss man die Werte in cm als Argumente angeben.

GGT()

Siehe auch

Die Primfaktorenzerlegung finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt GGT.



KOMBINATIONEN()



Syntax KOMBINATIONEN($n;k$)

Definition Die Funktion gibt die Anzahl der möglichen Kombinationen ohne Wiederholung von k Elementen aus einer Menge von n Elementen zurück.

Argumente n (erforderlich) ist die Anzahl der Elemente.

k (erforderlich) gibt an, aus wie vielen Elementen jede Kombination bestehen soll.

Da KOMBINATIONEN() nur mit ganzen Zahlen rechnet, kürzt Excel numerische Werte zu ganzen Zahlen.

Hinweise Verwenden Sie KOMBINATIONEN(), um zu berechnen, wie viele Gruppen aus einer bestimmten Anzahl von Elementen gebildet werden können, wenn jede Kombination dasselbe Element **nur einmal** enthalten darf.

Die Kombination ist eine beliebige Menge oder Teilmenge von Elementen. Im Gegensatz zur Permutation oder Variation (siehe auch Funktion VARIATIONEN() in Kapitel 11) spielt bei der Kombination die interne Reihenfolge der Elemente keine Rolle, d.h. die Anordnungen werden außer Acht gelassen. Die Elementfolge »ABC« ist gleichwertig mit »BCA« oder »CAB«. Es muss also weniger Kombinationen als Variationen oder Permutationen geben.

Die Anzahl möglicher Kombinationen ohne Wiederholung wird wie folgt berechnet, wobei n allen Elementen und k der Anzahl der ausgewählten Elemente entspricht:

$$C_n^{(k)} = \binom{n}{k} = \frac{P_{k,n}}{k!} = \frac{n!}{k!(n-k)!}$$

wobei:

$$P_{k,n} = \frac{n!}{(n-k)!}$$

Praxiseinsatz Für die Organisation eines Fußballturniers soll ermittelt werden, wie viele Spiele stattfinden müssen, damit alle Mannschaften einmal gegeneinander gespielt haben.

Folgende Mannschaften haben sich angemeldet: A, B, C, D

Möglichkeiten:

1. A + B
2. A + C
3. A + D
4. B + C
5. B + D
6. C + D

Das Ergebnis lautet 6. Die Formel in Excel sieht so aus:

=KOMBINATIONEN(4;2)

BINOMVERT() (Statistische Funktionen), FAKULTÄT(), HYPGEOMVERT() (Statistische Funktionen), KOMBINATIONEN2(), KRITBINOM() (Statistische Funktionen), NEGBINOMVERT() (Statistische Funktionen), VARIATIONEN()

Das Beispiel finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *KOMBIN*.

Siehe auch



KOMBINATIONEN2() COMBINA()

KOMBINATIONEN2($n;k$)

Die Funktion gibt die Anzahl der möglichen Kombinationen mit Wiederholungen von k Elementen aus einer Menge von n Elementen zurück.

n (erforderlich) ist die Anzahl der Elemente.

k (erforderlich) gibt an, aus wie vielen Elementen jede Kombination bestehen soll.

Da KOMBINATIONEN2() nur mit Werten ohne Komma rechnet, kürzt Excel numerische Werte zu ganzen Zahlen.

Verwenden Sie KOMBINATIONEN2(), um zu berechnen, wie viele Gruppen aus einer bestimmten Anzahl von Elementen gebildet werden können, wenn jede Kombination dasselbe Element **mehrfach** enthalten darf.

Die Kombination ist eine beliebige Menge oder Teilmenge von Elementen. Im Gegensatz zur Permutation oder Variation (siehe auch Funktion VARIATIONEN() in Kapitel 11) spielt bei der Kombination die interne Reihenfolge der Elemente keine Rolle, d.h. die Anordnungen werden außer Acht gelassen. Die Elementfolge »ABC« ist gleichwertig mit »BCA« oder »CAB«. Es muss also weniger Kombinationen als Variationen oder Permutationen geben.

Die Anzahl möglicher Kombinationen mit Wiederholung wird wie folgt berechnet, wobei n allen Elementen und k der Anzahl der ausgewählten Elemente entspricht:

$${}^w C_n^{(k)} = \binom{n+k-1}{k}$$

Mit k Würfeln sind ${}^w C_6^{(k)} = \binom{6+k-1}{k}$ verschiedene Würfe möglich.

Für zwei Würfel gilt demnach ${}^w C_6^{(2)} = \binom{6+2-1}{2} = \binom{7}{2} = 21$.

Für den Ausdruck $\binom{7}{2}$ lesen Sie bitte bei KOMBINATIONEN() ab Seite 750 nach.

BINOMVERT() (Statistische Funktionen), FAKULTÄT(), HYPGEOMVERT() (Statistische Funktionen), KOMBINATIONEN(), KRITBINOM() (Statistische Funktionen), NEGBINOMVERT() (Statistische Funktionen), VARIATIONEN()

Das Beispiel finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *KOMBIN*.

**Neu in Excel
2013**

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



KÜRZEN()



Syntax KÜRZEN(*Zahl*; *Anzahl_Stellen*)

Definition Die Funktion schneidet die Kommastellen der Zahl ab und gibt als Ergebnis eine ganze Zahl zurück.

Argumente *Zahl* (erforderlich) ist die Zahl, deren Stellen Sie abschneiden möchten.

Anzahl_Stellen (optional) ist eine Zahl, die angibt, wie viele Nachkommastellen erhalten bleiben sollen. Liegt keine Eingabe vor, ist die Voreinstellung für *Anzahl_Stellen* 0 (null). Es sind positive und negative Werte für *Anzahl_Stellen* möglich.

Hintergrund Bei der Funktion KÜRZEN() werden die restlichen Zellen hinter dem durch *Anzahl_Stellen* bestimmten Schnittpunkt einfach abgeschnitten. Wird kein Schnittpunkt angegeben, wird hinter der ganzen Zahl abgetrennt.

Die Funktionen KÜRZEN() und ABRUNDEN() geben das gleiche Ergebnis zurück. Beide Funktionen unterscheiden sich nur darin, dass die Anzahl der Stellen, auf die abgerundet werden soll, bei der Funktion KÜRZEN() angegeben werden **kann** und bei der Funktion ABRUNDEN() angegeben werden **muss**.

Praxiseinsatz Als Praxiseinsatz bietet sich das Kürzen von monetären Beträgen z.B. auf 2 Stellen hinter dem Komma oder auf volle Beträge an.

=KÜRZEN(234,4692354;2) ergibt 234,46.

=KÜRZEN(234,4692354;-1) ergibt 230.

=KÜRZEN(9,3) ergibt einen Wert von 9.

=KÜRZEN(-9,3) ergibt einen Wert von -9.

=KÜRZEN(18,628;2) ergibt einen Wert von 18,62.

Siehe auch GANZZAHL(), REST(), OBERGRENZE.MATHEMATIK(), RUNDEN(), UNTERGRENZE.MATHEMATIK()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *KÜRZEN*.

LN()



Syntax LN(*Zahl*)

Definition Die Funktion gibt den natürlichen Logarithmus einer Zahl zurück.

Argumente *Zahl* (erforderlich) ist die positive reelle Zahl, deren natürlichen Logarithmus Sie berechnen möchten.

Hintergrund Bei einer Gleichung $b = ax$, wobei a und b bekannt sind, lässt sich der Wert x durch das sog. Logarithmieren ermitteln:

$$x = \log_a b$$

Beispiel:

$$2 = \log_3 9$$

Der Wert x wird als Logarithmus der Basis a mit dem Numerus b bezeichnet.

Alle Logarithmen mit gleicher Basis bilden ein Logarithmensystem. Als Basis können alle positiven Zahlen außer 0 und 1 verwendet werden.

Die Logarithmen mit der Basis 2 (binär) kürzt man mit lb ab.

In der Naturwissenschaft wird vielfach die Eulersche Zahl e (2,71828182845904) als Basis verwendet. Weitere Informationen zur Zahl e finden Sie unter der Funktion EXP(). Die Abkürzung für Logarithmen mit der Basis e ist ln (natürliche Logarithmen).

Excel bietet die Funktion LN() zur Berechnung von natürlichen Logarithmen, wobei LN() die Umkehrfunktion zu EXP() ist.

Der ln steht im folgenden Zusammenhang zum log :

$$\log_b x = \frac{\ln x}{\ln b}$$

Um das Rechnen einfacher zu gestalten, wählte der Engländer Henry Briggs die Basis 10. Man nennt diese Logarithmen Briggsche, dekadische oder Zehnerlogarithmen – siehe LOG10().

Er wird unter anderem zur Beschreibung von Wachstums- oder Zerfallsprozessen benutzt, ähnlich wie die Funktion EXP().

Hier einige Beispiele, um die Funktion zu dokumentieren:

=LN(86) ergibt den natürlichen Logarithmus von 86 = 4,4543473.

=LN(2,7182818) ergibt den natürlichen Logarithmus zur Konstante $e = 1$.

=LN(EXP(3)) ergibt den natürlichen Logarithmus der Basis potenziert mit 3 = 3.

EXP(), IMLN() (Technische Funktionen), IMLOG10() (Technische Funktionen), IMLOG2() (Technische Funktionen), LOG(), LOG10()

Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt LN_LOG.

Praxiseinsatz

Siehe auch



LOG() LOG()

LOG(*Zahl*; *Basis*)

Die Funktion gibt den Logarithmus von *Zahl* zu der angegebenen *Basis* zurück.

Zahl (erforderlich) ist die positive reelle Zahl, deren Logarithmus Sie berechnen möchten.

Basis (erforderlich) ist die Basis des Logarithmus. Wenn das Argument *Basis* fehlt, wird es als 10 angenommen.

Der Logarithmus ist die Umkehrung der Exponentialfunktion.

Beispiel:

Die Exponentialfunktion $a = b^x$ lässt sich mit dem Logarithmus wie folgt umstellen:

$$x = \log_b a$$

Syntax

Definition

Argumente

Hintergrund

Der Logarithmus eines Produkts (Quotienten) entspricht der Summe (Differenz) der Logarithmen aus ihren Faktoren (dem Dividenden und dem Divisor):

$$\lg(a \cdot b) = \lg a + \lg b$$

$$\lg\left(\frac{a}{b}\right) = \lg a - \lg b$$

Durch die Logarithmen werden:

- ▶ Multiplikation und Division auf Addition und Subtraktion,
- ▶ Potenzieren und Radizieren auf Multiplikation und Division

zurückgeführt.

Im Gegensatz zu den Funktionen LN() und LOG10() ist bei der Funktion LOG() die Möglichkeit, die Basis selber zu bestimmen. Wird die Basis nicht angegeben, entspricht LOG() der Funktion LOG10().

Praxiseinsatz =LOG(10;10) ergibt 1.
 =LOG(2;10) ergibt 0,30103.
 =LOG(2) ergibt 0,30103.
 =LOG(2;5) ergibt 0,430676558.
 =LOG(2;3) ergibt 0,630929754.

Siehe auch EXP(), IMLN() (Technische Funktionen), IMLOG10() (Technische Funktionen), IMLOG2() (Technische Funktionen), LN(), LOG10()



Weitere Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf den Arbeitsblättern LN_LOG.

LOG10() LOG10()

Syntax LOG10(*Zahl*)

Definition Die Funktion gibt den dekadischen Logarithmus (Basis 10) von *Zahl* zurück.

Argumente *Zahl* (erforderlich) ist die positive reelle Zahl, deren Logarithmus zur Basis 10 Sie berechnen möchten.

Hintergrund Die Funktion LOG10() bildet den sogenannten »Zehnerlogarithmus« oder auch »Dekadischen Logarithmus« mit der Zahl 10 als Basis.

Praxiseinsatz In der Praxis wird der dekadische Logarithmus häufig bei naturwissenschaftlichen Funktionen angewendet, z.B. zur Berechnung der Lautstärke in Dezibel »dB«:

$$L = \left(\log \frac{P_2}{P_1}\right) B = 10 \left(\log \frac{P_2}{P_1}\right) dB$$

Für weitere Informationen wenden Sie sich bitte der entsprechenden Fachliteratur zu.

Hier noch einige Beispiele, um die Funktion zu dokumentieren:

=LOG10(2) ergibt den Wert 0,30103.

=LOG10(6) ergibt den Wert 0,77815125.

=LOG10(2,6) ergibt den Wert 0,414973348.

=LOG10(10) ergibt den Wert 1.

EXP(), IMLN() (Technische Funktionen), IMLOG10() (Technische Funktionen), IMLOG2() (Technische Funktionen), LN(), LOG()

Siehe auch

Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *LN_LOG*.



MDET() MDETERM()

MDET(*Matrix*)

Gibt die Determinante einer Matrix zurück.

Matrix (erforderlich) ist eine quadratische Matrix, d.h. die Anzahl der Zeilen und Spalten ist identisch.

Voraussetzung zur Ermittlung der Determinanten ist eine quadratische Matrix als Berechnungsgrundlage. In Excel kann Matrix als

- ▶ Zellbereich (zum Beispiel A1:C3),
- ▶ Matrix-Konstante (zum Beispiel {1.2.3;4.5.6;7.8.9}) oder
- ▶ Name, der einem Zellbereich oder einer Matrix-Konstanten zugeordnet ist
- ▶ eingegeben werden.

Sind die zu einer Matrix gehörenden Zellen leer oder enthalten diese Text, gibt MDET() den Fehlerwert #WERT! zurück. MDET() gibt ebenfalls den Fehlerwert #WERT! zurück, wenn *Matrix* nicht quadratisch ist.

Mithilfe der Determinanten lässt sich bestimmen, ob ein lineares Gleichungssystem eine eindeutige Lösung hat. Ist der Wert der Determinanten gleich 0, hat die Gleichung keine oder keine eindeutige Lösung.

Zur Berechnung der Determinanten einer Matrix A werden folgende Formeln angewendet:

1. A ist eine 1x1-Matrix:

$$\det(A) = a_{1,1}$$

2. A ist eine 2x2-Matrix:

$$\det(A) = a_{1,1} \cdot a_{2,2} - a_{2,1} \cdot a_{1,2}$$

3. A ist eine 3x3-Matrix:

$$\det(A) = a_{1,1} \cdot a_{2,2} \cdot a_{3,3} + a_{1,3} \cdot a_{2,1} \cdot a_{3,2} + a_{3,1} \cdot a_{2,2} \cdot a_{1,3} - a_{1,1} \cdot a_{2,3} \cdot a_{3,2} - a_{1,2} \cdot a_{2,1} \cdot a_{3,3}$$

Syntax

Definition

Argumente

Hintergrund

Als allgemeine Formel für eine n -mal- n -Matrix hat Gottfried Wilhelm Leibniz (Deutscher Philosoph und Mathematiker, 1646 bis 1716) die nach ihm benannte Leibniz-Formel entwickelt:

$$\det(A) = \sum_{\sigma \in S_n} (\text{sgn}(\sigma)) \prod_{i=1}^n a_{i,\sigma(i)}$$

Praxiseinsatz Siehe das Praxisbeispiel zur Funktion MINV().

Siehe auch MEINHEIT(), MINV(), MMULT(), MTRANS() (Matrixfunktionen)



Beispiele zu dieser Funktion finden Sie im Ordner `Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf den Arbeitsblättern *Matrix I* und *Matrix II*.

Neu in Excel
2013

MEINHEIT()



MUNIT()

Syntax MEINHEIT(*Größe*)

Definition Die Funktion MEINHEIT() gibt die Einheitsmatrix für die angegebene Größe zurück.

Argumente *Größe* (erforderlich) ist eine ganze Zahl > 0 , die die Anzahl der Spalten bzw. Zeilen der Einheitsmatrix bestimmt.

Hintergrund Eine Matrix ist eine rechteckige Anordnung von $n \cdot m$ Zahlen (n Zeilen, m Spalten), die in runde Klammern eingeschlossen sind. Ist $n = m$, handelt es sich um eine quadratische Matrix. Bei einer quadratischen Diagonalmatrix sind alle Glieder außerhalb der Hauptdiagonalen (von links oben nach rechts unten) 0.

Eine Einheitsmatrix ist eine quadratische Diagonalmatrix, bei der alle Glieder der Hauptdiagonalen 1 sind.

Praxiseinsatz Die Funktion MEINHEIT() ist eine Matrixfunktion, was bedeutet, dass sie nicht in einer einzelnen Zelle oder einer einzelnen Formel benutzt werden kann. Sie wirkt sich wie alle Matrixfunktionen auf einen rechteckigen Bereich in dem Arbeitsblatt aus – der Matrix eben.

Um mit Matrixfunktionen zu arbeiten, markieren sie den betreffenden Bereich, geben die Formel mit der Matrixfunktion in die weiß unterlegte Zelle ein und schließen die Eingabe mit der Tastenkombination `[↵]+[Strg]+[↵]` ab. Der betreffende Bereich wird von nun als eine Einheit, die Matrix, behandelt: Sie können die Formeln in diesem Bereich nur in ihrer Gesamtheit mit Matrixfunktionen in den Formeln verändern. Zur Kenntlichmachung sind alle Formeln der Matrix in geschweifte Klammern eingeschlossen, z.B. {MEINHEIT(4)}.

Hinweis Wenn Sie einen Bereich markieren, ist dieser Bereich grau ausgefüllt. Eine Zelle ist jedoch weiß unterlegt. Die Position dieser hervorgehobenen Zelle innerhalb des Bereichs ist je nach Vorgehensweise unterschiedlich, spielt aber keine Rolle. Die Matrixfunktion füllt den Bereich immer in der linken oberen Ecke beginnend nach rechts unten aus. Haben Sie den Bereich zu groß gewählt, werden die überschüssigen Zellen mit dem Fehlerwert #NV gefüllt.

Wenn Sie in Ihrem Arbeitsblatt eine 4x4-Einheitsmatrix in den Zellen B4 bis E7 anlegen wollen, markieren Sie bitte den Bereich, geben (in die weiß unterlegte Zelle) die Formel =MEINHEIT(4) ein und schließen die Formeleingabe mit der Tastenkombination `[↵]+[Strg]+[↵]` ab.

Die Zellen B4, C5, D6 und E7 sind nun mit Einsen gefüllt, die übrigen Zellen des markierten Bereichs mit Nullen: Ihre 4x4-Einheitsmatrix!

INDEX(), MDET(), MINV(), MMULT(), MTRANS() (Suchfunktionen)

Das Beispiel finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *MEINHEIT*.

Siehe auch



MINV()



MINVERSE()

MINV(*Matrix*)

Die Funktion gibt die Inverse einer Matrix (die zu einer Matrix gehörende Kehrmatrix) zurück.

Matrix (erforderlich) ist eine quadratische Matrix (die Anzahl der Zeilen und Spalten ist identisch).

Die Inverse einer Matrix wird verwendet, um lineare Gleichungen mit mehreren Variablen zu lösen. Voraussetzung für die Invertierbarkeit einer Matrix ist eine quadratische Matrix. Außerdem darf es sich nicht um eine singuläre Matrix handeln, d.h. die Determinante der Matrix ist $\neq 0$, siehe die Funktion MDET().

In der Mathematik wird die Inverse einer Matrix durch ein hoch gestelltes -1 gekennzeichnet (Matrix A wird zu A^{-1}).

Die Lösung der Gleichung $Ax=b$ ergibt $x=A^{-1}b$.

Manuell errechnen lässt sich die Inverse über folgende Verfahren:

1. Gaußsches Eliminationsverfahren

Beim Gaußschen Eliminationsverfahren handelt es sich um eine häufig angewendete Methode zur manuellen Berechnung einer Inversen. Eine detaillierte Erklärung würde den Rahmen dieses Buchs sprengen. Bitte schauen Sie in der einschlägigen Literatur nach.

2. Cramersche Regel

Die Cramersche Regel – nach Gabriel Cramer (1704 bis 1752) – ist aufgrund der erforderlichen meist aufwendigen Berechnung nur von theoretischem Interesse. Für Dimensionen ≤ 3 kann sie durchaus für die schnelle Lösungsberechnung herangezogen werden. Bitte wenden Sie sich auch hier der einschlägigen Literatur zu.

3. Mithilfe der Formel:

$$A^{-1} = \frac{1}{\det(A)} \cdot A^{\#}$$

Für die Formel werden die Determinante – $\det(A)$ siehe Funktion MDET() – sowie die komplementäre Matrix ($A^{\#}$) benötigt.

In die Bearbeitungsleiste von Excel kann eine *Matrix* als

- ▶ Zellbereich, beispielsweise als A1:C3,
- ▶ Matrixkonstante, beispielsweise {1.2.3;4.5.6;7.8.9} oder
- ▶ Name für eine dieser beiden Möglichkeiten

angegeben werden.

Syntax

Definition

Argumente

Hintergrund

Wenn Zellen in der Matrix leer sind oder Text enthalten, gibt MINV() den Fehlerwert #WERT! zurück. MINV gibt außerdem den Fehlerwert #WERT! zurück, wenn die Anzahl der Zeilen und Spalten in Matrix nicht gleich ist.

Da es sich bei dem Ergebnis der Formel MINV() um eine Matrix handelt, muss die Formel als Matrixformel eingegeben werden – Abschluss der Eingabe mit den Tasten $\boxed{\text{Strg}} + \boxed{\uparrow} + \boxed{\leftarrow}$.

MINV() wird mit einer Genauigkeit von ungefähr 16 Stellen berechnet; dies kann zu einem kleinen numerischen Fehler führen, wenn nicht richtig gerundet wird.

Für einige quadratische Matrizen kann keine Inverse ermittelt werden; die Funktion MINV() gibt in diesem Fall den Fehlerwert #ZAH! zurück. Die Determinante für eine nicht invertierbare Matrix ist 0.

Praxiseinsatz Aus folgender Gleichung soll der Wert x berechnet werden:

$$Ax = b$$

wobei:

	A	B	C	D	E	F	G
2							
3		1	-2	-2			1
4	A =	-2	1	0		b =	2
5		-1	-2	-3			3
6							

Abbildung 15.24: Variablen der Gleichung

Lösung:

- Umstellung der Formel nach x:

$$x = A^{-1}b$$

- Um zu kontrollieren, ob die Gleichung eindeutig lösbar ist, kann mit MDET() die Determinante für die Matrix A gebildet werden. Ist die Determinante ungleich 0 besteht eine eindeutige Lösung, die mithilfe der inversen Matrix ermittelt werden kann.

Die Formel für die Determinante lautet: =MDET(B3:D5)

Ergebnis: die Determinante von A ist = -1

Fazit: Die Determinante von A ist ungleich 0, also besteht für die Matrix A eine eindeutige Lösung.

- Berechnung des inversen Elements von $A = A^{-1}$

Hierfür wird in einen Zellenbereich (B9:D11) folgende Matrixformel in Bezug auf die Matrix A eingegeben: {=MINV(B3:D5)}

	A	B	C	D	E
8					
9		3	2	-2	
10	A ⁻¹ =	6	5	-4	
11		-5	-4	3	
12					

Abbildung 15.25: Berechnung des inversen Elements

4. Um zu kontrollieren, ob die inverse Matrix auch korrekt gebildet worden ist, wird A mit A^{-1} multipliziert. Die Formel (im Bereich B14:D16) lautet: `{=MMULT(B3:D5;B9:D11)}`. Das Ergebnis muss eine Einheitsmatrix sein:

	A	B	C	D	E
12					
13	Kontrolle:				
14		1	0	0	
15	$AA^{-1} =$	0	1	0	
16		0	0	1	
17					

Abbildung 15.26: Kontrolle durch Einheitsmatrix

(Wir sehen Excel hat richtig gerechnet ;)

5. Zum Schluss wird noch A^{-1} mit b multipliziert. Die Formel lautet `{=MMULT(B9:D11;G3:G5)}` und das Ergebnis ist x.

	A	B	C
17			
18		1	
19	x =	4	
20		-4	
21			

Abbildung 15.27: Das Ergebnis der Gleichung

INDEX(), MDET(), MEINHEIT(), MMULT(), MTRANS() (Suchfunktionen)

Beispiele zu dieser Funktion finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf den Arbeitsblättern *Matrix I* und *Matrix II*.

Siehe auch



MMULT() MMULT()

MMULT(*Matrix1*; *Matrix2*)

Gibt das Produkt zweier Matrizen zurück. Das Ergebnis ist eine Matrix, die dieselbe Anzahl von Zeilen wie *Matrix1* und dieselbe Anzahl von Spalten wie *Matrix2* hat.

Matrix1 (erforderlich) und *Matrix2* (erforderlich) sind die Matrizen, die Sie multiplizieren möchten.

In die Bearbeitungsleiste von Excel kann *Matrix1* und *Matrix2*

- ▶ als Zellbereich, beispielsweise als A1:C3,
- ▶ als Matrixkonstante, beispielsweise {1.2.3;4.5.6;7.8.9} oder
- ▶ als Name für eine dieser beiden Möglichkeiten

angegeben werden.

Das Produkt zweier Matrizen kann nur berechnet werden, wenn die Anzahl der Spalten der ersten Matrix gleich der Anzahl der Zeilen der zweiten Matrix ist. Beide Matrizen dürfen nur Zahlen enthalten.

Syntax

Definition

Argumente

Hintergrund

Zur Berechnung des Produkts werden die Zeilenelemente der ersten Matrix mit den entsprechenden Spaltenelementen der zweiten Matrix multipliziert.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \end{pmatrix} = \begin{pmatrix} (a_{11} \cdot b_{12} + a_{12} \cdot b_{21} + a_{13} \cdot b_{31}) & (a_{11} \cdot b_{12} + a_{12} \cdot b_{22} + a_{13} \cdot b_{32}) \\ (a_{21} \cdot b_{11} + a_{22} \cdot b_{21} + a_{23} \cdot b_{31}) & (a_{21} \cdot b_{12} + a_{22} \cdot b_{22} + a_{23} \cdot b_{32}) \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

Praxeinsatz Siehe das Praxisbeispiel zur Funktion MINV().

Siehe auch MDET(), MEINHEIT(), MINV(), MTRANS() (Suchfunktionen)



Beispiele zu dieser Funktion finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf den Arbeitsblättern *Matrix I* und *Matrix II*.

OBERGRENZE() CEILING()

Syntax OBERGRENZE(*Zahl*; *Schritt*)

Definition Rundet eine Zahl betragsmäßig auf das kleinste Vielfache von *Schritt* auf.

Argumente *Zahl* (erforderlich) ist der Wert, den Sie runden möchten.

Schritt (erforderlich) ist das Vielfache, auf das Sie runden möchten.

Hintergrund Die Funktion OBERGRENZE(), genauso wie die Funktion OBERGRENZE.GENAU() (s.u.), sollen durch die einfacher einzusetzende Funktion OBERGRENZE.MATHEMATIK() auf Seite 761 ersetzt werden. Lesen Sie bitte dort weitere Erklärungen.

OBERGRENZE.GENAU() CEILING.PRECISE()

Syntax OBERGRENZE.GENAU(*Zahl*; *Schritt*)

Definition Rundet eine Zahl auf die nächste ganze Zahl oder das nächste Vielfache von *Schritt* auf. Die Zahl wird unabhängig vom Vorzeichen aufgerundet.

Argumente *Zahl* (erforderlich) ist der Wert, den Sie runden möchten.

Schritt (optional) ist das Vielfache, auf das Sie runden möchten.

Wenn *Schritt* ausgelassen wird, ist der Standardwert 1.

Hintergrund Die Funktion OBERGRENZE.GENAU(), genauso wie die Funktion OBERGRENZE() (s.o.), sollen durch die einfacher einzusetzende Funktion OBERGRENZE.MATHEMATIK() auf Seite 761 ersetzt werden. Lesen Sie bitte dort weitere Erklärungen.

OBERGRENZE.MATHEMATIK()



CEILING.MATH()

OBERGRENZE.MATHEMATIK (*Zahl*; *Schritt*; *Modus*)

Rundet eine Zahl auf die nächste ganze Zahl oder auf das nächste Vielfache von *Schritt* auf.

Zahl (erforderlich) ist die Zahl, die gerundet werden soll. *Zahl* muss kleiner als $+9,99^{+307}$ und größer als $-2,229^{-308}$ sein.

Schritt (optional) ist der Wert für das Vielfache, auf das *Zahl* aufgerundet werden soll. Wenn *Schritt* ausgelassen wird, ist der Standardwert +1 bei einer positiven *Zahl* und -1 bei einer negativen *Zahl*.

Modus (optional) steuert für *Zahl* $\neq 0$, ob *Zahl* in Richtung des größeren oder des kleineren Werts gerundet wird.

Diese Funktion gibt Ihnen die Möglichkeit, Zahlen auf eine bestimmte Intervallgrenze aufzurunden, wobei Sie das Intervall angeben können. Diese Herangehensweise kommt besonders bei kaufmännischen Berechnungen vor. Natürlich wäre diese Art der Berechnung auch mit den anderen Runden-Funktionen in einer eigenen Formelkonstruktion möglich, aber OBERGRENZE.MATHEMATIK() stellt den einfachsten Weg dar.

Die Funktionen OBERGRENZE.GENAU() aus Excel 2010 und OBERGRENZE() wurden in Excel 2013 durch die neue Funktion OBERGRENZE.MATHEMATIK() ergänzt, um das Verhalten im Fall von negativen Werten besser steuern zu können.

Die Funktion OBERGRENZE() können Sie weiter nutzen; sie befindet sich jetzt in der Kategorie »Kompatibilität«.

Die Funktion OBERGRENZE.GENAU() ist nun weder in den Kategorien des Menübands noch in den Listen des Funktions-Assistenten aufgeführt. Sie können sie aber ebenfalls weiter nutzen; Sie müssen sie dann händisch mit dem kompletten Namen in die Formel eingeben.

Bei der Funktion OBERGRENZE.MATHEMATIK() kann über das Argument *Modus* die Rundungsrichtung bei negativem Argument *Zahl* geändert werden:

- ▶ Bei *Modus* = 0 (*FALSCH* oder weggelassen) bewegt sich das Rundungsergebnis zu größeren Zahlen.
- ▶ Bei *Modus* = -1 (*WAHR* oder eine andere von 0 verschiedene Zahl) bewegt sich das Ergebnis zu größeren absoluten Beträgen, also von der 0 weg.

Bei der Funktion OBERGRENZE.GENAU() wird immer zu größeren absoluten Beträgen (»aufwärts«) gerundet.

Bei einem positiven Argument *Zahl* wird *Modus* ignoriert.

Die Funktion OBERGRENZE() arbeitet etwas uneinheitlich (siehe Tabelle 15.3).

Sie möchten in einer Tariftable die Entgelte entsprechend der vereinbarten Tarifierhöhung berechnen. Dabei sollen alle Ergebnisse auf ganze 50 Cent aufgerundet werden. Anders formuliert: Es sind nur Beträge mit einer 0 (null) oder einer 5 nach dem Komma möglich.

Neu in Excel
2013

Syntax

Definition

Argumente

Hintergrund

Hinweis

Praxiseinsatz

Die Formel

=OBERGRENZE.MATHEMATIK(Betrag;0,5)

löst diese Rechenaufgabe.

Hier noch ein paar Beispiele, um die Funktion zu dokumentieren:

=OBERGRENZE.MATHEMATIK(2345,47;0,5) ergibt 2.345,50 €.

=OBERGRENZE.MATHEMATIK(2345,67;0,5) ergibt 2.346,00 €.

Tabelle 15.3
Vergleich der drei Varianten der Funktion OBERGRENZE

Formel	Ergebnis
=OBERGRENZE.MATHEMATIK(3,2;1)	4
=OBERGRENZE.MATHEMATIK(3,2;-1)	4
=OBERGRENZE.MATHEMATIK(-3,2;1)	-3
=OBERGRENZE.MATHEMATIK(-3,2;-1)	-3
=OBERGRENZE.MATHEMATIK(3,2;1;1)	4
=OBERGRENZE.MATHEMATIK(3,2;-1;1)	4
=OBERGRENZE.MATHEMATIK(-3,2;1;1)	-4
=OBERGRENZE.MATHEMATIK(-3,2;-1;1)	-4
=OBERGRENZE.MATHEMATIK(3,2;1;-1)	4
=OBERGRENZE.MATHEMATIK(3,2;-1;-1)	4
=OBERGRENZE.MATHEMATIK(-3,2;1;-1)	-4
=OBERGRENZE.MATHEMATIK(-3,2;-1;-1)	-4
=OBERGRENZE(3,2;1)	4
=OBERGRENZE(3,2;-1)	#ZAHL!
=OBERGRENZE(-3,2;1)	-3
=OBERGRENZE(-3,2;-1)	-4
=OBERGRENZE.GENAU(3,2;1)	4
=OBERGRENZE.GENAU(3,2;-1)	4
=OBERGRENZE.GENAU(-3,2;1)	-3
=OBERGRENZE.GENAU(-3,2;-1)	-3

Verwandte Funktionen

ABRUNDEN(), AUFRUNDEN(), GANZZAHL(), GERADE(), ISO.OBERGRENZE(), KÜRZEN(), OBERGRENZE(), OBERGRENZE.GENAU(), RUNDEN(), UNGERADE(), UNTERGRENZE(), UNTERGRENZE.GENAU(), UNTERGRENZE.MATHEMATIK()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt OBERGRENZE.

PI()



PI()

PI()

Gibt die mathematische Konstante Pi ($\pi=3,14159265358979$) mit einer Genauigkeit von 15 Stellen zurück.

Keine

Pi (π) ist die sog. Kreiszahl. Sie beschreibt das Verhältnis des Umfangs u eines Kreises zu seinem Durchmesser (d) bzw. Radius (r):

$$d \cdot \pi = u$$

oder

$$2r \cdot \pi = u$$

π ist eine unendliche Zahl, die nur annähernd bestimmt werden kann. Excel rechnet mit 15 Nachkommastellen. Gerundet auf 100 Nachkommastellen hat π folgenden Wert:

3,141 592 653 589 793 238 462 643 383 279 502 884 197 169 399 375 105 820 974 944 592 307
816 406 286 208 998 628 034 825 342 117 067 0

Die Funktion PI() findet in der Praxis immer dann Anwendung, wenn eine Kreisberechnung eine Rolle spielt z.B.

In der Geometrie:

- ▶ Volumen eines Zylinders (r = Radius, h = Höhe) : $V = r^2 \pi h$
- ▶ Oberfläche einer Kugel (r = Radius): $A_0 = 4\pi r^2$

In der Arithmetik:

- ▶ Eulersche Identität: $e^{i\varphi} = \cos(\varphi) + i \sin(\varphi)$

In der Physik:

- ▶ Kreisbewegung: $\omega = 2\pi f$ (Winkelgeschwindigkeit = 2π mal Umlaufgeschwindigkeit)

COS(), COSEC(), COT(), SEC(), SIN(), TAN()

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch

POLYNOMIAL()



MULTINOMINAL()

POLYNOMIAL(Zahl1;Zahl2; ...)

POLYNOMIAL() gibt den Polynomkoeffizienten einer Gruppe von Zahlen zurück.

Die Formel für die POLYNOMIAL()-Funktion lautet:

$$\text{POLYNOMIAL}(a, b, c) = \frac{(a+b+c)!}{a! \cdot b! \cdot c!}$$

Zahl1;Zahl2;... (erforderlich) sind 1 bis 255 Werte, deren Polynomkoeffizienten berechnet werden sollen.

Die Funktion POLYNOMIAL() berechnet den Polynomkoeffizienten für einen Satz von Zahlen zur Bestimmung der Permutationen, also der Anordnungsmöglichkeiten von Gruppen identischer Elemente unter Beachtung der Reihenfolge. Die Beispiele unten verdeutlichen dies.

Syntax

Definition

Argumente

Hintergrund

Ferner lässt sich damit die Polynomialverteilung ermitteln. Dies stellt die Verallgemeinerung des Spezialfalls Binomialkoeffizienten bzw. Binomialverteilung dar.

Praxiseinsatz Es sollen die Möglichkeiten berechnet werden, in welcher Reihenfolge 5 Personen, nämlich zwei Frauen ($a=2$), zwei Männer ($b=2$) und ein Kind ($c=1$) auf einer Bank mit 5 Sitzplätzen sitzen können. Folgende Formel führt zum Ergebnis:

=POLYNOMIAL(2;2;1) ergibt 30, d.h. es stehen 30 verschiedene Sitzmöglichkeiten zur Verfügung.

Siehe auch FACTDOUBLE(), FAKULTÄT(), ZWEIFAKULTÄT()

POTENZ() POWER()

Syntax POTENZ(*Zahl*; *Potenz*)

Definition Gibt als Ergebnis eine potenzierte Zahl zurück.

Argumente *Zahl* (erforderlich) ist die Zahl, die Sie mit dem Exponenten potenzieren möchten. Es sind alle reellen Zahlen zulässig.

Potenz (erforderlich) ist der Exponent, mit dem Sie die *Zahl* potenzieren möchten.

Hintergrund Potenzieren ist das (wiederholte) Multiplizieren einer Zahl mit sich selbst:

$$a \cdot a \cdot a \cdot \dots \cdot a = a^n$$

(Die Anzahl mit der a mit sich selbst multipliziert wird, nennt man die Potenz n von a .)

a wird als Basis bzw. Grundzahl und

n wird als Exponent bzw. Hochzahl bezeichnet.

Das Ergebnis ist die Potenz. Sowohl der Logarithmus als auch die Wurzelfunktion sind Umkehrfunktionen des Potenzierens.

Alternativ zur Funktion POTENZ() kann der Operator »^« zum Potenzieren einer Zahl verwendet werden, zum Beispiel =5^2 anstelle von =POTENZ(5;2).

Praxiseinsatz Für die digitale Verarbeitung am Computer kommt das Dualsystem zum Einsatz. Die Basis für das Dualsystem liefert die Zahl 2. Die Größeneinheiten von digitalen Speichermedien sind daher die Potenzen zur Basis 2 (Zweierpotenzen). Aus diesem Grund verwenden einige Softwarehersteller ein Kilobyte mit 2^{10} Byte = 1.024 Byte und nicht 1.000 Byte, wie es die SI-konforme Bedeutung eines KB fordert.

Hier noch ein paar weitere Beispiele, um die Funktion zu dokumentieren:

=POTENZ(3;2) entspricht 3^2 – Ergebnis: 9.

=POTENZ(3,2;3) entspricht $3,2^3$ – Ergebnis: 32,768.

=POTENZ(7;1,33) entspricht $7^{1,33}$ – Ergebnis: 13,30394354.

Siehe auch WURZEL()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt POTENZ.

POTENZREIHE()



SERIESSUM()

POTENZREIHE($x;n;m$;Koeffizienten)

POTENZREIHE($x;n;m;\{a_1;a_2;a_3;a_4;\dots;a_k\}$) gibt folgende Summe von Potenzen zurück:

$$y = a_1x^n + a_2x^{n+m} + a_3x^{n+2m} + a_4x^{n+3m} + \dots + a_kx^{n+km}$$

Die Anzahl k der Koeffizienten bestimmt die Anzahl der Glieder in der Potenzreihe.

x (erforderlich) ist der Wert der unabhängigen Variablen der Potenzreihe.

n (erforderlich) ist die Potenz von x im ersten Term der Potenzreihe.

m (erforderlich) ist das Inkrement, um das Sie n in jedem Glied der Reihe vergrößern möchten.

Koeffizienten (erforderlich) ist eine Matrix oder eine Bereichsangabe, die eine Gruppe von Koeffizienten enthält, mit denen die aufeinanderfolgenden Potenzen der Variablen x multipliziert werden. Die Anzahl der in der Matrix oder dem Bereich angegebenen Werte bestimmt, wie viele Glieder (Potenzen) die jeweilige Potenzreihe umfasst. Sind in Koeffizienten beispielsweise drei Werte angegeben, besteht die zugehörige Potenzreihe aus drei Gliedern.

Unter einer Potenzreihe versteht man in der Analysis eine unendliche Reihe in der Form:

$$\sum_{n=0}^{\infty} a_n (x - x_0)^n$$

$$(a_n)_{n \in \mathbb{N}_0}$$

ist als eine beliebige Folge von reellen oder komplexen Zahlen definiert. Der Wert x_0 wird als Entwicklungspunkt der Potenzreihe definiert.

Viele Funktionen können mithilfe einer Potenzreihenentwicklung angenähert werden. Die Genauigkeit des erreichten Ergebnisses hängt von der Anzahl der Glieder der Potenzreihe ab; je mehr Glieder sie aufweist, umso exakter ist das Ergebnis.

Die folgenden Beispiele verdeutlichen das Einsatzgebiet der Funktion.

Beispiel 1:

=POTENZREIHE(1;0;1;{1;1;0,5;0,16666667;0,04166667;0,00833333;0,00019841})

ergibt 2,71686508. Hierin wurden eingesetzt:

$$x = 1$$

$$n = 0$$

$$m = 1$$

$$a_1 = 1$$

$$a_2 = 1$$

$$a_3 = 0,5 = 1/2!$$

$$a_4 = 0,16666667 = 1/3!$$

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

$$a_5 = 0,04166667 = 1/4!$$

$$a_6 = 0,008333333 = 1/5!$$

$$a_7 = 0,00019841 = 1/6!$$

Das stellt die ersten sieben Glieder der Reihe

$$1 + 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} + \frac{1}{120} + \frac{1}{720} + \dots$$

zur Berechnung der Eulerschen Zahl e dar.

Beispiel 2:

In dieser Tabelle wird die im Kasten dargestellte Potenzreihe ausgewertet (siehe Abbildung 15.28). Die Funktionsargumente sind namentlich und mit ihren Werten im Block B4:C10 zusammengefasst.

	A	B	C	D	E	F	G
1							
2	Potenzreihe						
3							
4	x	0,785398			$y = \frac{x}{1!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$ ergibt		
5	n	1					
6	m	2					
7	a ₁	1					
8	a ₂	-0,1666667			0,70710635		
9	a ₃	0,008333333					
10	a ₄	-0,0001984			45,00		
11					0,70710667		
12							
13							
14		0,785398 rad $\hat{=}$ 45,00°					
15							

Abbildung 15.28: Eine Potenzreihe mit einem ziemlich bekannten Ergebnis

In der Zelle D7 sehen Sie das Ergebnis, das uns die Potenzreihen-Funktion mit den gegebenen Argumenten auswirft. Vielleicht kommt Ihnen diese Zahl bekannt vor. – Wenn nicht, dann betrachten Sie doch einmal die Zellen D9 und D10 und ihre Inhalte!

In D9 steht die Formel

$$=GRAD(B3)$$

die den x-Wert in B3 ins Gradmaß (45°) umwandelt. Dann muss also in B3 ein Wert im Bogenmaß stehen. In D10 steht die Formel

$$=SIN(B3)$$

die aus den 45° den Sinus berechnet. Das Ergebnis stimmt bis zur sechsten Nachkommastelle mit dem Wert überein, den die Potenzreihen-Funktion errechnete.



Diese Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt POTENZREIHE.

PRODUKT() PRODUCT()

PRODUKT(*Zahl1*; *Zahl2*; ...)

Diese Funktion multipliziert die Argumente und gibt das Produkt zurück.

Zahl1 (erforderlich); *Zahl2*; ... sind 1 bis 255 Zahlen, die Sie multiplizieren möchten.

Als Produkt wird das Ergebnis einer Multiplikation bezeichnet, wobei die Werte, die miteinander multipliziert werden, Faktoren heißen:

$$a_1 \cdot a_2 \cdot a_3 \cdot a_4 \cdot \dots = b$$

In der Mathematik gibt es analog zum Summensymbol Σ das Symbol Π für Summe, um das Produkt über mehrere Faktoren darzustellen.

Multiplikation von Beträgen z.B. auf einer Einkaufsliste, wie in der Abbildung 15.29 dargestellt.

	A	B	C	D	E	F
1	Obst	Anzahl	Einzelpreis	Gesamtpreis		
2	Äpfel	5	0,45 €	2,25 €		
3	Melonen	1	2,30 €	2,30 €		
4	Birnen	3	0,53 €	1,59 €		

Abbildung 15.29: Anwendung der Funktion PRODUKT()

Syntax
Definition
Argumente
Hintergrund

Praxiseinsatz

FAKULTÄT(), SUMME(), SUMMENPRODUKT(), ZWEIFAKULTÄT()

Siehe auch

QUADRATESUMME() SUMSQ()

QUADRATESUMME(*Zahl1*; *Zahl2*; ...)

Summiert die quadrierten Argumente.

Zahl1 (erforderlich); *Zahl2*; ... sind 1 bis 255 Argumente, deren Quadrate Sie summieren möchten. Anstelle von durch ein Semikolon getrennten Argumenten können Sie auch eine einzeilige oder einspaltige Matrix oder einen Bezug auf eine solche Matrix verwenden.

QUADRATESUMME() bildet die Quadratzahlen der übergebenen Argumente und summiert diese auf. Die Formel dafür lautet:

$$\sum(a^2, b^2, c^2, d^2 \dots)$$

Im rechtwinkligen Dreieck mit den Katheten *a* und *b* und der Hypotenuse *c* gilt der Satz des Pythagoras:

$$a^2 + b^2 = c^2$$

Nach Ziehen der Quadratwurzel kann man aus der Länge der Katheten die Hypotenuse berechnen:

$$c = \sqrt{a^2 + b^2}$$

Syntax
Definition
Argumente
Hintergrund

Praxiseinsatz

Mit $a=4$ und $b=3$ erhalten wir die entsprechende Excel-Formel
`=WURZEL(QUADRATESUMME(4;3))`
 und damit als Ergebnis $c=5$.

Hier noch ein paar Beispiele, um die Funktion zu dokumentieren:

`=QUADRATESUMME(2;3;4)` ergibt 29

`=QUADRATESUMME(5;6)` ergibt 61

`=QUADRATESUMME(12;3)` ergibt 153

Siehe auch SUMME(), SUMMENPRODUKT(), WURZEL()



Die Beispiele finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `QUADRATESUMME`.

QUOTIENT() QUOTIENT()

Syntax QUOTIENT(*Zähler*; *Nenner*)

Definition Diese Funktion gibt den ganzzahligen Anteil einer Division zurück.

Argumente *Zähler* (erforderlich) ist der Dividend.

Nenner (erforderlich) ist der Divisor.

Hintergrund Die Division ist die Umkehrung der Multiplikation. Sie kann durch einen Doppelpunkt $a : b$ oder als Bruch

$$\frac{a(\text{Zähler, Divident})}{b(\text{Nenner, Divisor})} \left. \vphantom{\frac{a}{b}} \right\} \text{Quotient}$$

dargestellt werden, d.h. der Doppelpunkt und der Bruchstrich sind gleichbedeutende Rechenzeichen. In Excel-Formeln verwenden Sie jedoch den Schrägstrich (/) als Operator für die Division.

Hinweis Mehr zu den Operatoren in Excel-Formeln finden Sie in Kapitel 2.

Die Teilungszahl (a) heißt »Zähler« oder »Dividend«, der Teiler (b) »Nenner« oder »Divisor«. Beide zusammen bilden einen »Quotienten«.

Diese Funktion können Sie immer dann verwenden, wenn Sie die Nachkommastellen (den Rest) einer Division löschen möchten.

Eine Alternative dazu wäre die Funktion GANZZAHL() mit der Division als Argument:

`=QUOTIENT(1;4)` ist identisch mit `=GANZZAHL(1/4)`

Praxiseinsatz Es sollen 587 Teile verpackt werden. In jeden Karton passen 12 Teile. Wie viele Kartons werden voll?

`=QUOTIENT(587;12)` ergibt 48, also 48 Kartons werden vollständig gefüllt.

Weitere Beispiele:

=QUOTIENT(7;3) ergibt 2.

=QUOTIENT(5,87;2) ergibt 2.

=QUOTIENT(-5;2) ergibt -2.

GANZZAHL(), REST()

Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *QUOTIENT*.

Siehe auch



REST() MOD()

REST(*Zahl*; *Divisor*)

Die Funktion gibt den Rest einer Division zurück. Das Ergebnis hat dasselbe Vorzeichen wie *Divisor*.

Zahl (erforderlich) ist die Zahl, für die der Rest einer Division gesucht wird.

Divisor (erforderlich) ist die Zahl, durch die Zahl dividiert werden soll.

Bei bestimmten Berechnungen müssen Sie den Rest einer Division betrachten. Man spricht auch vom »Modulo«-Rechnen (*mod*). Hierfür steht Ihnen die REST()-Funktion zur Verfügung.

Wenn Divisor den Wert 0 (null) hat, gibt REST() den Fehlerwert #DIV/0! zurück.

Die REST()-Funktion kann auch mithilfe der GANZZAHL()-Funktion formuliert werden:

=Zähler-Divisor*GANZZAHL(Zähler/Divisor)

Diese Formel steht für die am häufigsten in Softwaresystemen implementierte Modulo-Berechnungsmethode. Es gibt nämlich zwei Varianten der Modulo-Berechnung, die für negative Argumente unterschiedliche Ergebnisse liefern:

► Mathematische Methode (Excel-REST()-Funktion)

$$Z \bmod D = Z - D * \lceil Z/D \rceil$$

wobei $\lceil Z/D \rceil$ die Division in Gaußschen Klammern beschreibt (z.B. $\lceil -4,3 \rceil = -5$). Dies entspricht dem Ergebnis der Funktion =GANZZAHL(Z/D). Für die mathematische Methode gilt hinsichtlich negativer Werte:

$$(\cdot Z) \bmod D \neq \cdot (Z \bmod D)$$

Beispiel: $-7 \bmod 3 = 2$ $-1 = -(7 \bmod 3)$

Das Ergebnis hat immer dasselbe Vorzeichen wie der Divisor (D).

Syntax

Definition

Argumente

Hintergrund

► **Symmetrische Methode**

$$Z \text{ mod } D = Z - D * (Z \text{ div } D)$$

Dabei bezeichnet $(Z \text{ div } D)$ den **zur Null hin** gerundeten Quotienten (z.B. $-4,3 \approx -4$).
Hinsichtlich negativer Werte gilt:

$$(\cdot Z) \text{ mod } D = -(Z \text{ mod } D)$$

Beispiel: $-7 \text{ mod } 3 = -1 = -1 = -(7 \text{ mod } 3)$

Das Ergebnis hat immer dasselbe Vorzeichen wie der Zähler (Z).

Achtung Bei Excel 2010 hatte sich bei dieser Funktion ein Berechnungsfehler eingeschlichen, der auch in der Version 2013 noch zu finden ist.

Wenn das Argument *Divisor* kleiner 1 und das Argument *Zahl* ein ganzzahliges Vielfaches von *Divisor* ist und deswegen *Zahl* ohne Rest durch *Divisor* teilbar sein müsste, liefert REST() als Ergebnis den Wert von *Divisor* z.B. =REST(10;0,2) ergibt 0,2.

Sie können sich bei *Divisor*-Werten kleiner als 1 mit dieser Formel behelfen (*Zahl* in A1 und *Divisor* in B1):

$$=A1-B1*GANZZAHL(A1/B1)$$

Praxiseinsatz Die Funktion REST() findet häufig zusammen mit anderen Formeln Anwendung, z.B. bei der Addition jeder zweiten Zeile:

	A	B	C	D	E
1					
2		Name	Taschengeld		
3		Uwe	10,00 €		
4		Kai	12,00 €		
5		Britta	9,00 €		
6		Finn	15,00 €		
7		Susi	12,00 €		
8		Peter	8,00 €		
9					
10			35,00 €	Summe jeder 2. Zeile	

Abbildung 15.30: Addition jeder zweiten Zeile

Die Formel lautet $\{=SUMME(WENN(REST(ZEILE(B2:B7);2)=0;B2:B7))\}$. Da es sich hier um eine Matrixformel handelt, muss die Formel mit $\text{Strg} + \text{↵} + \text{↵}$ eingegeben werden.

Hier weitere Beispiele, um die Funktion zu dokumentieren:

=REST(7;3) ergibt einen Wert von 1.

=REST(-7;3) ergibt einen Wert von 2 (Der Wert ist positiv, da Divisor positiv ist).

=REST(7;-3) ergibt einen Wert von -2 (Der Wert ist negativ, da Divisor negativ ist).

=REST(12,56;3,2) ergibt einen Wert von 2,96.

Siehe auch ABRUNDEN(), AUFRUNDEN(), GANZZAHL(), KÜRZEN(), OBERGRENZE.MATHEMATIK(), REST(), RUNDEN(), UNTERGRENZE.MATHEMATIK(), VRUNDEN()



Die Beispiele finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `REST`.

RÖMISCH()



RÖMISCH(*Zahl*;*Typ*)

Die Funktion konvertiert eine arabische Zahl in eine römische Zahl im Textformat.

Zahl (erforderlich) ist die arabische Zahl, die Sie umwandeln möchten. Es sind Zahlenangaben von 0 (null) bis 3999 möglich. Negative Werte und Zahlen >3999 ergeben die Fehlermeldung #WERT!.

Typ (optional) ist eine Zahl oder Wahrheitswert, die den Typ der römischen Zahl festlegt. Die Schreibweise der römischen Zahlen reicht von klassisch bis vereinfacht, wobei die Zeichenfolge kürzer wird, sobald ein höherer Typ vorliegt. Hier die möglichen Typangaben und ihre Bedeutung:

Typ	Schreibweise
0	Klassisch
1	Kürzer
2	Kürzer
3	Kürzer
4	Vereinfacht
WAHR	Klassisch
FALSCH	Vereinfacht

Syntax

Definition

Argumente

Tabelle 15.4

Typen der römischen Zahlen

Arabisch	Römisch				
	0	1	2	3	4
3.999	Klassisch MMMCMXCIX	kürzer MMMLMVLIV	kürzer MMMXMIX	kürzer MMMVMIV	vereinfacht MMMIM

Abbildung 15.31: Die unterschiedlichen Zahltypen und ihre Auswirkung auf die Konvertierung der größten möglichen Zahl

Bei den römischen Zahlzeichen unterscheidet man in Grundzeichen und Hilfszeichen. Die Hilfszeichen sind historisch später entstanden, weil die Römer gemerkt haben, dass die Grundzeichen manchmal zu unpraktisch großen Zahlendarstellungen führen.

Hintergrund

Grundzeichen		Hilfszeichen	
Zeichen	Wertigkeit	Zeichen	Wertigkeit
I	1	V	5
X	10	L	50
C	100	D	500
M	1000		

Tabelle 15.5

Römische Zahlen und deren dezimale Entsprechung

Regeln zur Bildung von römischen Zahlen:

1. Gleiche Ziffern nebeneinander werden addiert. Es dürfen höchstens drei Grundzeichen nebeneinander stehen. Beispiel: III = 3
2. Kleinere Ziffern rechts von größeren werden addiert, links von größeren subtrahiert. Hilfszeichen dürfen nicht subtrahiert werden. Beispiel: XI = 11, IX = 9, XLV = 45
3. Die Grundzeichen I, X, C dürfen nur von dem nächst höheren Hilfs- oder Grundzeichen subtrahiert werden. Beispiele: CD = 400, CM = 900

Die heutzutage gelehrt Regeln für römische Zahlenzeichen waren in Mitteleuropa bis in das 16. Jahrhundert allgemein gebräuchlich und wurden bis dahin ständig angepasst. Arabische Ziffern setzten sich erst mit Beginn des Buchdrucks durch. So kannten die Römer z.B. nicht die »Subtraktionsregel«. Noch heute befindet auf Zifferblättern von Uhren mit römischen Ziffern die ursprüngliche Schreibweise der Zahl 4: IIII.

Neben der klassischen Schreibweise der römischen Ziffern sind weitere, kürzere Schreibweisen möglich. Excel trägt dem durch die verschiedenen Typen Rechnung.

Praxiseinsatz Mit der Funktion RÖMISCH() können Sie arabische Zahlen in römische Zahlen umwandeln, wie sie z.B. noch in Überschriften von Kapiteln oder Aufzählungen verwendet werden:

1 Quellenverzeichnis	I Quellenverzeichnis
2 Inhaltsverzeichnis	II Inhaltsverzeichnis
3 Einleitung	III Einleitung

usw.

Hier weitere Beispiele, um die Funktion zu dokumentieren:

=RÖMISCH(999;0) ergibt CMXCIX – klassisch (Typ 0, WAHR oder leer)

=RÖMISCH(999;1) ergibt LMVLIV – kürzer (Typ 1)

=RÖMISCH(999;2) ergibt XMIX – kürzer (Typ 2)

=RÖMISCH(999;3) ergibt VMIV – kürzer (Typ 3)

=RÖMISCH(999;4) ergibt IM – vereinfacht (Typ 4)

Siehe auch ARABISCH(), DEZIMAL()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *RÖMISCH*.

RUNDEN() ROUND()

Syntax RUNDEN(*Zahl*; *Anzahl_Stellen*)

Definition Die Funktion rundet eine Zahl auf eine bestimmte Anzahl von Dezimalstellen.

Argumente *Zahl* (erforderlich) ist die Zahl, die Sie auf- oder abrunden möchten.

Anzahl_Stellen (erforderlich) gibt an, auf wie viele Dezimalstellen Sie die Zahl auf- oder abrunden möchten.

Das Runden ist aus unserem Zahlensystem nicht wegzudenken. Die meisten Zahlenwerte, die uns begegnen, sind in ihrer Entstehung irgendwann einmal gerundet worden. Die Gründe sind vielfältig: Große oder lange Zahlen lassen sich durch Runden in einem übersichtlichen, leichter rechenbaren Rahmen halten, z.B. bei Bevölkerungsstatistiken oder die Zahl Pi (π). Ebenso werden (fast) alle Preise auf zwei Stellen hinter dem Komma gerundet, da die kleinste bezahlbare Einheit 1 Cent ist.

Beim Runden wird die Zahl direkt hinter der zu rundenden Ziffer betrachtet. Ist diese größer als 4, wird aufgerundet. Ist diese Zahl kleiner oder gleich 4, wird abgerundet. Beispiele:

3,2549 € werden auf 3,25 € gerundet.

3,2551 € werden auf 3,26 € gerundet.

Negative Beträge werden von der Null weg aufgerundet.

-3,2549 € werden auf -3,25 € gerundet.

-3,2551 € werden auf -3,26 € gerundet.

Ist die Angabe in *Anzahl_Stellen* größer 0 (null), wird Zahl auf die angegebene Anzahl von Dezimalstellen gerundet. Ist sie gleich 0, wird Zahl auf die nächste ganze Zahl gerundet. Ist der Wert in *Anzahl_Stellen* kleiner 0, wird der links des Dezimalzeichens stehende Teil von Zahl gerundet.

In einer Preiskalkulation soll folgender Wert auf zwei Stellen hinter dem Komma ausgewiesen werden: 14,5943 €

Die Formel lautet =RUNDEN(14,5943;2) und ergibt den gerundeten Wert 14,59 €.

ABRUNDEN(), AUFRUNDEN(), GANZZAHL(), KÜRZEN(), OBERGRENZE.MATHEMATIK(), QUOTIENT(), REST(), UNTERGRENZE.MATHEMATIK(), VRUNDEN()

Eine Beispielreihe mit alternativen Prüfberechnungen finden Sie im Ordner `Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `RUNDEN`.

Hintergrund

Praxiseinsatz

Siehe auch



SEC()



SEC()

SEC(*Zahl*)

Die Funktion SEC() ist eine trigonometrische Funktion und gibt den Sekans einer Zahl zurück.

Zahl (erforderlich) ist der im Bogenmaß angegebene Winkel, dessen Sekans Sie berechnen möchten. Der maximale Wert für *Zahl* ist $\pm 134.217.727 (= \pm(2^{27} - 1))$.

In einem rechtwinkligen Dreieck ist das Verhältnis von Hypotenuse zu Ankathete abhängig vom Öffnungswinkel. Das Verhältnis der Hypotenuse eines Winkels zur Ankathete ist der Sekans (math. *sec*) eines Winkels. Der Sekans ist somit die Kehrwertfunktion des Kosinus.

$$\sec \alpha = \frac{c}{b}$$

$$\sec \beta = \frac{c}{a}$$

Neu in Excel 2013

Syntax

Definition

Argumente

Hintergrund

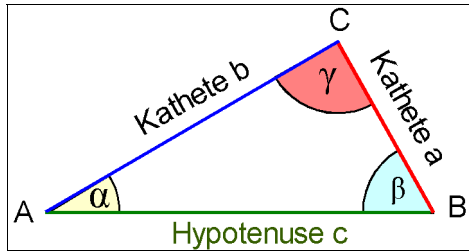


Abbildung 15.32: Seitenverhältnisse im rechtwinkligen Dreieck

Die Funktion SEC() benötigt als Rechengrundlage das Bogenmaß. Mithilfe der Funktion BOGENMASS() können Sie einen im Gradmaß vorliegenden Winkel in das Bogenmaß überführen.

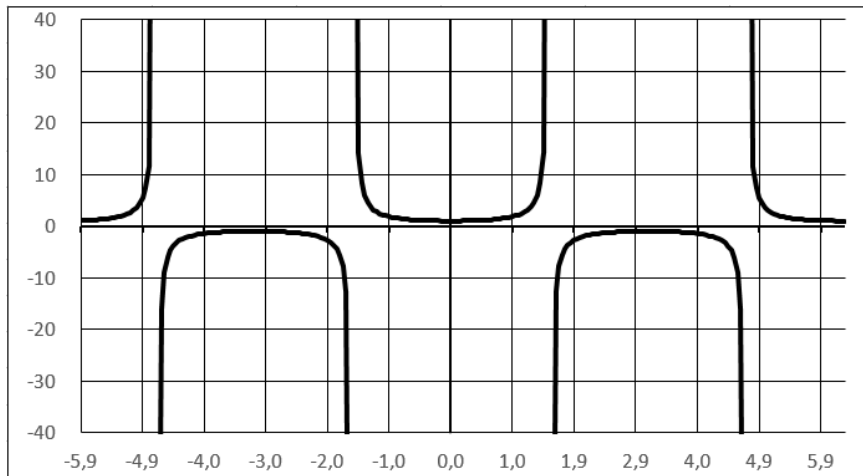


Abbildung 15.33: Der Graph der Sekansfunktion SEC()

Praxiseinsatz Die Sekansfunktion setzt man ein, um Formeln, die Terme mit dem Kehrwert der Kosinusfunktion ($1/\cos$) enthalten, zu vereinfachen, wie z.B. im MUF-Faktor $\sec \Phi$ im Secans-Gesetz aus der Kurzwellenfernübertragung (MUF: Maximum Usable Frequencies; Φ : Einfallswinkel in die Ionosphäre).

Hier noch einige Zahlenbeispiele:

=SEC(0) ergibt 1.

=SEC(0,5) ergibt 1,139493927.

=SEC(PI()) ergibt -1.

=SEC(PI()/2) ergibt 1,63246E+16.

Siehe auch COSEC(), COS(), SIN()



Die Tabelle für die grafische Darstellung aus Abbildung 15.33 und Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt SEC.

SECHYP()

SECHYP(*Zahl*)

Die Funktion SECHYP() gibt den hyperbolischen Sekans (Sekans Hyperbolicus, Hyperbelsekans, math. *sech*) einer reellen Zahl zurück.

Zahl (erforderlich) ist der Winkel, angegeben im Bogenmaß, dessen Sekans Hyperbolicus berechnet werden soll.

Die Funktion *sech* (Sekans Hyperbolicus) gehört zu den Hyperbelfunktionen und stellt die Kehrwertfunktion des Kosinus Hyperbolicus dar. Sie ist definiert als

$$\operatorname{sech} x = \frac{1}{\cosh x} = \frac{2}{e^x + e^{-x}}$$

Die Funktion SECHYP() benötigt ein Argument im Bogenmaß. Mithilfe der Funktion BOGENMASS() können Sie einen im Gradmaß vorliegenden Winkel in das Bogenmaß umrechnen.

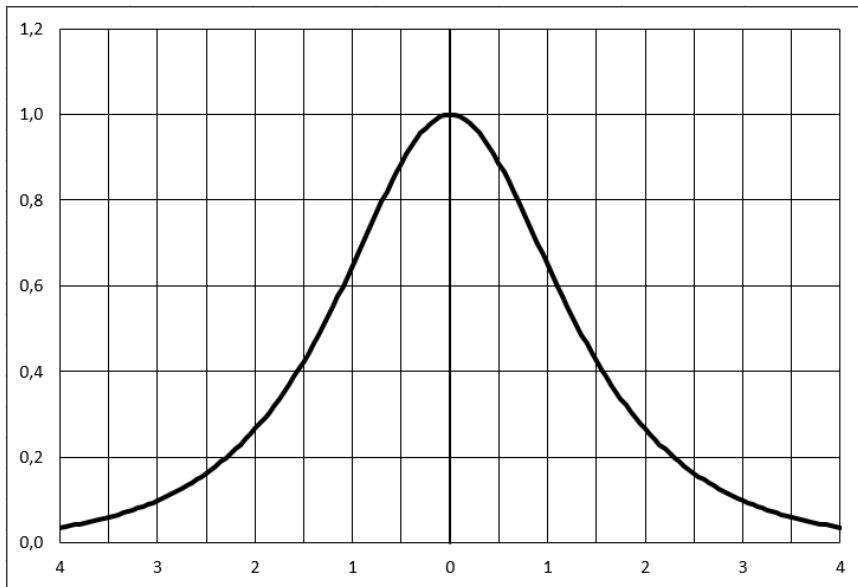


Abbildung 15.34: Graph der »Sekans Hyperbolicus«-Funktion SECHYP()

Der Sekans Hyperbolicus wird in der Naturwissenschaft und Technik durchaus benötigt. Als Beispiele ohne nähere Erläuterungen seien hier genannt: zur Berechnung elektrostatischer Gitterpotenziale, bei optischen und elektrischen Impulsen als Impulsform (meistens quadriert: sech^2) und bei der Berechnung der Beschleunigung im freien Fall mit Luftwiderstand.

=SECHYP(0) ergibt 1.

=SECHYP(0,5) ergibt 0,886818884.

=SECHYP(1) ergibt 0,648054274.

=SECHYP(PI()) ergibt 0,086266738.

Neu in Excel
2013

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch PI(), WURZEL()



Die Tabelle für die grafische Darstellung aus Abbildung 15.34 und die Beispiele finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `SECHYP`.

SIN() SIN()

Syntax SIN(*Zahl*)

Definition Gibt den Sinus einer Zahl zurück.

Argumente *Zahl* (erforderlich) ist der im Bogenmaß angegebene Winkel, dessen Sinuswert Sie berechnen möchten.

Hintergrund In einem rechtwinkligen Dreieck ist das Verhältnis von Hypotenuse zu Gegenkathete abhängig vom Öffnungswinkel. Das Verhältnis der Gegenkathete eines Winkels zur Hypotenuse ist der Sinus (sin) eines Winkels.

$$\sin \alpha = \frac{a}{c}, \sin \beta = \frac{b}{c}$$

Die Funktion SIN() benötigt als Rechengrundlage das Bogenmaß. Liegt der Winkel im Gradmaß vor, müssen Sie ihn durch Multiplizieren mit PI()/180 oder der Funktion BOGENMASS() in das Bogenmaß überführen.

Bei der Darstellung der Abhängigkeit des Sinus vom Winkel α in einem Einheitskreis ($c = 1$) nimmt der Sinus bis zu einem Winkel von 90° zu und entspricht in diesem Winkel genau dem Wert 1.

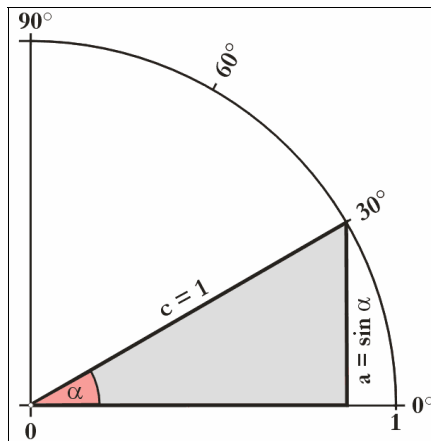


Abbildung 15.35: Sinus im Einheitskreis

Beim Übertrag der Werte in ein Koordinatensystem, mit den Werten des Winkels α auf der x -Achse und dem Sinus von α auf der y -Achse, ergibt sich folgende Kurve für den gesamten Kreis.

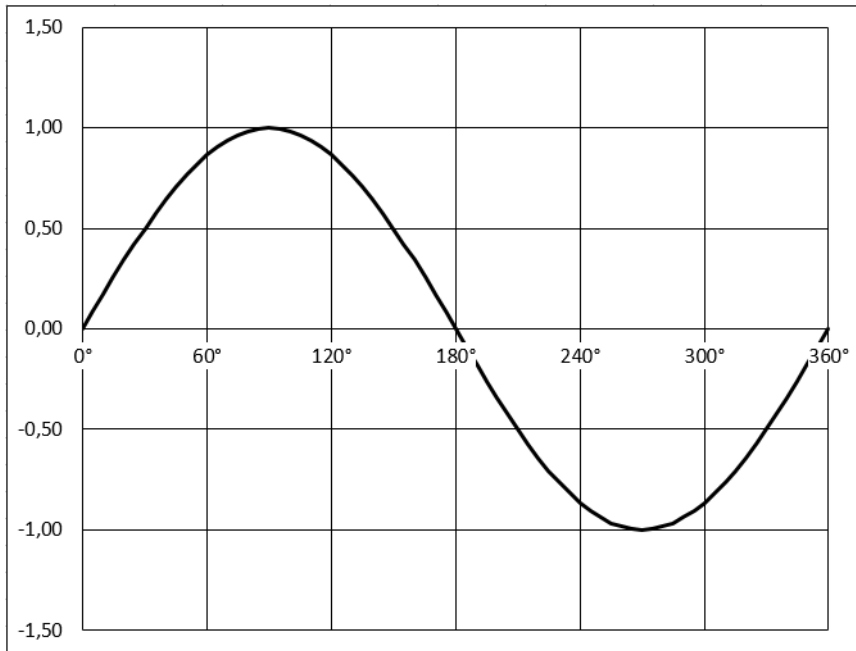


Abbildung 15.36: Sinuskurve

Es soll die Höhe eines Berges berechnet werden. Die Entfernung von einer Beobachtungsstation am Boden zum Berggipfel beträgt 3,7 km. Der Gipfel erscheint unter einem Winkel von 19,5°.

Praxiseinsatz

Die Formel für den Sinus lautet:

$$\sin = \frac{\text{Gegenkathete}}{\text{Hypotenuse}}$$

Die Formel wird mit den vorhandenen Daten ausgefüllt,

$$\sin(19,5) = \frac{\text{Höhe}}{3,7 \text{ km}}$$

und für die zu berechnende Höhe umgestellt.

$$\text{Höhe} = \sin(19,5) \cdot 3,7 \text{ km}$$

Das Ergebnis lautet:

$$\text{Höhe} = 1,23508538 \text{ km} \approx 1,235 \text{ km} = 1235 \text{ m}$$

In die Bearbeitungsleiste von Excel wird dafür folgende Formel eingegeben:

$$=\text{SIN}(\text{BOGENMASS}(19,5))*3,7$$

Damit Excel den Sinus des Winkels von $19,5^\circ$ berechnen kann, muss dieser zuerst in ein Bogenmaß umgerechnet werden. Anschließend wird der Sinus mit der Hypotenuse multipliziert.

Damit das Ergebnis gleich in Meter ausgegeben wird, muss das Ergebnis auf drei Stellen gerundet und mit 1.000 multipliziert werden:

```
=RUNDEN((SIN(BOGENMASS(19,5))*3,7);3)*1000
```

Siehe auch ARCSIN(), COS(), COSEC(), PI()



Die Tabelle für die grafische Darstellung aus Abbildung 15.36 und das Beispiel finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *SIN*. Das Berg-Beispiel haben wir auf dem Arbeitsblatt *trigon* in derselben Arbeitsmappe dargestellt.

SINHYP() SINH()

Syntax SINHYP(*Zahl*)

Definition SINHYP() gibt den hyperbolischen Sinus (Hyperbelsinus, Sinus Hyperbolicus, math. *sinh*) einer reellen Zahl zurück.

Argumente *Zahl* (erforderlich) ist eine beliebige reelle Zahl.

Hintergrund Der hyperbolische Sinus gehört zu den Hyperbelfunktionen, die, ebenso wie die Kreisfunktionen (z.B. Sinus und Tangens), für alle reellen und komplexen Zahlen definiert sind.

Microsoft Excel erlaubt aber nur reelle Argumente für die Hyperbelfunktionen in der Kategorie »Mathematisch-trigonometrische Funktionen«. Für komplexe Argumente benutzen Sie bitte die Funktionen in der Kategorie »Konstruktion«, z.B. IMSINHYP() (siehe Kapitel 16).

Die Definition für den hyperbolischen Sinus lautet:

$$\sinh x = \frac{e^x - e^{-x}}{2}$$

Seine grafische Darstellung zeigt die Abbildung 15.37. Der hyperbolische Sinus wird in der Technik und in den Naturwissenschaften, vor allem in der Forschung und Entwicklung, häufig gebraucht. Populärwissenschaftliche Anwendungen sind nicht bekannt.

Hier einige Beispiele, um die Funktion zu dokumentieren:

```
=SINHYP(0) ergibt 0.
```

```
=SINHYP(0,5) ergibt 0,52109531.
```

```
=SINHYP(-0,5) ergibt -0,52109531.
```

Tipp

Das letzte Beispiel aus der Microsoft Excel-Hilfe befasst sich mit einer empirischen kumulativen Verteilungsfunktion. Derartige Verteilungsfunktionen werden auf reale Messreihen angewendet, wenn das zu beobachtende System oder seine Komponenten sich nicht »ideal« verhalten, d.h. wenn es nicht die z.B. für die Normal- oder Poisson-Verteilung notwendigen Bedingungen und Annahmen erfüllt.

Oft kann dann eine besser geeignete kumulative¹ Verteilungsfunktion auf empirischem Weg gefunden werden – wie eben eine auf dem Hyperbelsinus basierende Funktion. Ein darauf basierendes Beispiel mit der Funktion SINHYP() finden Sie innerhalb der Beispieldateien zum Buch.

¹ Die Angabe »kumulativ« soll helfen, Verwechslungen mit der rein mathematischen Funktion »Wahrscheinlichkeitsdichte« zu vermeiden.

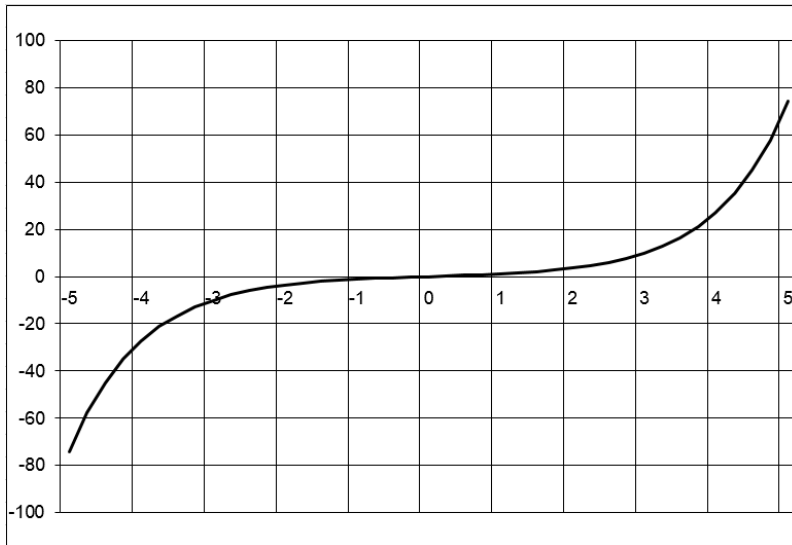


Abbildung 15.37: Der Hyperbelsinus ist ungerade und eine zwischen $-\infty$ und $+\infty$ monoton wachsende Funktion

ARCSINHYP(), COSHYP(), TANHYP()

Die Tabelle für die grafische Darstellung aus Abbildung 15.37 und das Beispiel aus dem Tipp finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `SINHYP.xlsx` auf den Arbeitsblättern *Graph* und *Verteilung*.

Siehe auch



SUMME() SUM()

SUMME(*Zahl1*; *Zahl2*; ...)

Die (sicher meistbenutzte) Excel-Funktion summiert die Argumente.

Zahl1 (erforderlich); *Zahl2*; ... sind 1 bis 255 Argumente, deren Summe Sie berechnen möchten.

Das Ergebnis einer Addition wird als Summe bezeichnet. Das Rechenzeichen für die Addition ist das Zeichen »+« (Plus). In der Mathematik ist ebenso das »Σ« als sog. Summenzeichen gebräuchlich:

$$x = a + b + c + d + \dots$$

oder

$$x = \sum(a, b, c, d \dots)$$

In der Funktion SUMME() werden Zahlen, Wahrheitswerte und Zahlen in Textform als Argumente berücksichtigt – soweit sie direkt in die Formel eingetragen werden. Beispiele:

- ▶ Das Argument *WAHR* wird in den Wert 1 umgewandelt.
- ▶ "12" wird als der Wert 12 berücksichtigt, obwohl der Wert durch die Anführungszeichen als Text gekennzeichnet ist.

Syntax

Definition

Argumente

Hintergrund

Hinweis Werden Text oder Wahrheitswerte über einen Zellenbezug in die Formel eingebunden, werden sie nicht berücksichtigt.

Ist als Argument eine Matrix oder ein Bezug angegeben, werden nur die Elemente dieser Matrix oder dieses Bezugs berücksichtigt, die Zahlen sind. Alle anderen Elemente, wie leere Zellen, Wahrheitswerte, Texte oder Fehlerwerte werden ignoriert.

Alle Fehlerwerte oder als Text angegebene Argumente, die nicht in Zahlen umgewandelt werden können, führen zu Fehlern.

Praxiseinsatz Der Praxiseinsatz ist so vielfältig, dass in Excel für diese Funktion das Summen-Symbol gleich mehrfach vorkommt: In der Startleiste und in der Leiste für die Funktionen. Kein Anwender von Excel wird diese Anwendung missen wollen und sei es nur, um »mal eben« seine Einnahmen oder Ausgaben zusammenzuzählen z.B.: =Summe(3,25;5,30;4,70) Das Ergebnis lautet 13,25.

Weitere Beispiele:

=Summe(1;4;7) ergibt den Wert 12.

=Summe(A2:B5) addiert die Werte im Zellbereich A2 bis B5.

=Summe(A2:B5;12) addiert zu der Summe des Zellbereichs A2 bis B5 den Wert 12.

Siehe auch ANZAHL() (Statistische Funktionen), ANZAHL2() (Statistische Funktionen), MITTELWERT(), PRODUKT(), SUMMENPRODUKT()

SUMMENPRODUKT() SUMPRODUCT()

Syntax SUMMENPRODUKT(*Matrix1*; *Matrix2*; *Matrix3*; ...)

Definition Die Funktion multipliziert die einander entsprechenden Komponenten der angegebenen Matrizen miteinander und gibt die Summe dieser Produkte zurück.

Argumente *Matrix1* (erforderlich); *Matrix2*; ... sind 2 bis 255 Matrizen, deren Komponenten Sie zunächst multiplizieren und anschließend addieren möchten.

Hintergrund Beim Summenprodukt werden die einzelnen Werte einer Matrix miteinander multipliziert und die Ergebnisse addiert. Die Formel dafür lautet:

$$\begin{array}{cc} a_1 & b_1 \\ a_2 \cdot b_2 = (a_1 \cdot b_1 + a_2 \cdot b_2 + a_3 \cdot b_3) \\ a_3 & b_3 \end{array}$$

So lässt sich z.B. ein Gesamtpreis ermitteln, wenn unterschiedliche Produkte mit Anzahl und Preis in einer Tabelle stehen.

Die als Argumente angegebenen Matrizen müssen bezüglich der Zeilen- und Spaltenanzahl identisch sein. Ist dies nicht der Fall, gibt SUMMENPRODUKT() den Fehlerwert #WERT! zurück

Matrizelemente, die keine numerischen Ausdrücke sind, behandelt SUMMENPRODUKT() so, als wären sie mit 0 (null) belegt.

Es soll für folgende Produkte ein Gesamtpreis ermittelt werden:

12 Äpfel à 0,39 €

15 Birnen à 0,42 €

Das Ergebnis wird wie folgt ermittelt:

$$\text{Gesamtpreis} = 12 \cdot 0,39 + 15 \cdot 0,42 = 10,98$$

In Excel werden die Ausgangswerte in zwei Matrizen dargestellt:

	A	B	C
1		Menge	Preis
2	Äpfel	12	0,39 €
3	Birnen	15	0,42 €
4			
5		Gesamtpreis:	10,98 €

Abbildung 15.38: Menge/Preis-Matrizen

In die Bearbeitungsleiste von Excel geben Sie für den Gesamtpreis folgende Formel ein:

=SUMMENPRODUKT(B2:B3;C2:C3)

MMULT(), PRODUKT(), SUMME()

Das (erweiterte) Beispiel finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `SUMMENPRODUKT`.

Praxiseinsatz

Siehe auch



SUMMEWENN()



SUMIF()

SUMMEWENN(*Bereich*; *Kriterien*; *Summe_Bereich*)

Diese Funktion addiert alle Zahlen in *Summe_Bereich*, die mit dem angegebenen Suchkriterium *Kriterien* im Zellbereich *Bereich* übereinstimmen.

Bereich (erforderlich) ist der Zellbereich, in dem Sie die Kriterien suchen möchten.

Kriterien (erforderlich) gibt die Kriterien in Form einer Zahl, eines Ausdrucks oder einer Zeichenfolge an. Diese Kriterien bestimmen, welche Zellen addiert werden. Zum Beispiel kann das Argument Kriterien als 32, "32", ">32", "Äpfel" oder als Funktion z.B. HEUTE() formuliert werden.

Summe_Bereich (optional) gibt die tatsächlich zu addierenden Zellen an. Ist *Summe_Bereich* nicht angegeben, wird über *Bereich* summiert.

Die Funktion SUMMEWENN() ist ein sehr interessantes Instrument, um aus einer Auflistung von Werten bestimmte Werte herauszufiltern und diese zu addieren. Werte aus einem festgelegten Bereich werden nur dann addiert, wenn Sie mit dem Suchkriterium übereinstimmen.

Die Funktion erlaubt (im Gegensatz zur WENN()-Funktion) eine Mustersuche mit den Platzhalterzeichen »?« und »*«. Das Fragezeichen steht für ein beliebiges Zeichen, der Stern für eine beliebige Anzahl (auch 0) beliebiger Zeichen. Mustersuchen müssen, ebenso wie Kriterien, die logische oder mathematische Symbole enthalten, in doppelte Anführungszeichen (") eingeschlossen werden.

Syntax

Definition

Argumente

Hintergrund

Mustersuchen arbeiten nur in Bereichen, die Texte enthalten. Eine Summation über Texte ist nicht möglich, daher muss die Funktion bei diesem Einsatz immer das dritte Argument *Summe_Bereich* aufweisen.

Tipp Wenn Sie bei der Angabe eines Kriteriums Vergleichsoperatoren mit einer Tabellenfunktion kombinieren wollen, z.B. »><« mit MITTELWERT(), dann wird die Funktion MITTELWERT() nicht als mathematische Funktion ausgewertet werden, weil sie wegen der Syntaxforderung nach doppelten Anführungszeichen von Excel als Text interpretiert wird:

```
=SUMMEWENN(Bereich;"<MITTELWERT(Ihr_Bereich)";Summe_Bereich)
```

Ihr Ziel können Sie dennoch erreichen, wenn Sie den Vergleichsoperator und die Tabellenfunktion mit dem Verkettungsoperator »&<« verbinden:

```
=SUMMEWENN(Bereich;"<"&MITTELWERT(Ihr_Bereich);Summe_Bereich)
```

Hinweis Excel stellt weitere Funktionen zur Verfügung, die es ermöglichen, Daten auf der Grundlage einer Bedingung zu analysieren. Um beispielsweise die Anzahl der Vorkommen einer Textzeichenfolge oder einer Zahl innerhalb eines Bereichs von Zellen zu zählen, verwenden Sie die Funktion ZÄHLENWENN() aus dem Bereich »Statistische Funktionen«, siehe Kapitel 11. Dort finden Sie auch die neuen Funktionen MITTELWERTWENN() und MITTELWERTWENNS().

Damit eine Formel einen Wert auf der Grundlage einer Bedingung zurückgibt, beispielsweise einen Umsatzbonus, der auf einer bestimmten Umsatzmenge beruht, verwenden Sie die Logikfunktion WENN(), siehe Kapitel 8.

Praxiseinsatz 1 Aufgabe ist es, aus einer Liste mit Umsätzen, diejenigen Umsätze zu addieren, die dem Außendienstmitarbeiter »Meier« zugeordnet sind.

	A	B	C	D
1	Name	Umsatz		
2	Meier	350,00 €		
3	Klotz	260,00 €		
4	Meier	300,00 €		
5	Peters	360,00 €		
6	Schmitz	240,00 €		
7				
8	Gesamtumsatz von:			
9	Meier	=SUMMEWENN(A2:A6;A9;B2:B6)		

Abbildung 15.39: Listensumme mit Filter

In die Zelle B9 wird folgende Formel eingegeben:

```
=SUMMEWENN(A2:A6;A9;B2:B6)
```

- ▶ Als Erstes wird der Bereich der Werte festgelegt, aus dem das Kriterium herausgefiltert werden soll (A2:A6).
- ▶ Anschließend wird das Suchkriterium, nach dem summiert wird, festgelegt (A9).
- ▶ Zum Schluss muss noch der *Summe_Bereich* bestimmt werden, in dem die zu addierenden Werte stehen (B2:B6).

Das Ergebnis ist die Summe aller Beträge von Meier: 650 €.

Fehlt eine Angabe für *Summe_Bereich*, werden die zu Bereich gehörenden Zellen addiert.

Eine Liste mit Firmenbezeichnungen (im Bereich *B12:B230*) und beigeordneten Zahlenwerten wie z.B. Aktienkurs, Umsatz, Mitarbeiterzahl o.Ä. (im Bereich *F12:F230*) soll nach Firmen durchsucht werden, deren Firmenbezeichnung den Begriff »solar« enthält. Die zugehörigen Werte in Spalte F sollen addiert werden. Eine geeignete Formel sieht so aus:

```
=SUMMEWENN(B12:B230;"*solar*"; F12:F230)
```

SUMME(), SUMMEWENNS() sowie MITTELWERTWENN(), MITTELWERTWENNS(), ZÄHLENWENN() (Statistische Funktionen)

Diese und weitere Beispiele finden Sie im Ordner *Ms5-235\Kap15* in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *SUMMEWENN*.

Praxiseinsatz 2

Siehe auch



SUMMEWENNS()



SUMIFS()

SUMMEWENNS(*Summe_Bereich*; *Kriterien_Bereich1*; *Kriterien1*; *Kriterien_Bereich2*; *Kriterien2*; ...)

Syntax

Diese Funktion addiert alle Zahlen in *Summe_Bereich*, die mit den angegebenen Suchkriterien in den Bereichen *Kriterien_Bereich1*, *Kriterien_Bereich2* usw. mit den *Kriterien1*, *Kriterien2* usw. übereinstimmen.

Definition

Summe_Bereich (erforderlich) ist der Zellbereich, in dem Sie Werte entsprechender Kriterien addieren möchten.

Argumente

Kriterien_Bereich1 (erforderlich), *Kriterien_Bereich2*, ... sind 1 bis 127 Zellbereiche, in denen Sie die Werte aus den zugeordneten *Kriterien1*, *Kriterien2*, ... suchen möchten.

Kriterien1 (erforderlich), *Kriterien2*, ... gibt die Kriterien in Form einer Zahl, eines Ausdrucks oder einer Zeichenfolge an. Diese Kriterien bestimmen, welche Zellen addiert werden. Zum Beispiel kann das Argument Kriterien als 32, "32", ">32", "Äpfel" oder als Funktion z.B. HEUTE() formuliert werden.

Die Funktion erlaubt (im Gegensatz zur WENN()-Funktion) eine Mustersuche mit den Platzhalterzeichen »?« und »*«. Das Fragezeichen steht für ein beliebiges Zeichen, der Stern für eine beliebige Anzahl (auch 0) beliebiger Zeichen. Mustersuchen müssen, ebenso wie Kriterien, die logische oder mathematische Symbole enthalten, in doppelte Anführungszeichen (") eingeschlossen werden.

Mustersuchen arbeiten nur in Bereichen, die Texte enthalten. Eine Summation über Texte ist nicht möglich.

Beachten Sie bitte auch den »Profitipp« bei der Funktion SUMMEWENN() auf Seite 781!

Bei Angabe mehrerer Kriterien werden die Kriterien mit einem logischen UND verknüpft, d.h. nur wenn **alle Kriterien** auf **einen Suchbegriff** zutreffen, wird sein zugehöriger Wert in *Summe_Bereich* in die Addition einbezogen.

Tipp

Auf diese Funktion (seit Excel 2007) hatte die Welt gewartet: Endlich war es möglich, in der SUMMEWENN()-Berechnung mehr als ein Kriterium anzugeben.

Hintergrund

SUMMEWENNS() ist also eine Erweiterung der SUMMEWENN()-Funktion und, wie diese, ein Instrument, um aus einer Auflistung von Werten bestimmte Werte herauszufiltern und diese zu addieren. Werte aus einem festgelegten Bereich werden nur dann addiert, wenn sie mit den Suchkriterien übereinstimmen.

Hinweis Excel stellt weitere Funktionen zur Verfügung, die es ermöglichen, Daten auf der Grundlage einer Bedingung zu analysieren. Um beispielsweise die Anzahl der Vorkommen einer Textzeichenfolge oder einer Zahl innerhalb eines Bereichs von Zellen zu zählen, verwenden Sie die Funktion ZÄHLENWENN() aus dem Bereich »Statistische Funktionen« (siehe Kapitel 11). Dort finden Sie auch die neuen Funktionen MITTELWERTWENN() und MITTELWERTWENNS().

Damit eine Formel einen oder zwei Werte auf der Grundlage einer Bedingung zurückgibt, beispielsweise einen Umsatzbonus, der auf einer bestimmten Umsatzmenge beruht, verwenden Sie die Logikfunktion WENN(), siehe Kapitel 8.

Praxiseinsatz Aufgabe ist es, aus einer Liste mit Umsätzen, diejenigen Umsätze zu addieren, die einer bestimmten Firma zugeordnet sind, für ein bestimmtes Produkt stehen und in einem bestimmten Monat getätigt wurden. Die Abbildung 15.40 zeigt die Lösung mit SUMMEWENNS().

	Firma	Produkt	Monat	Umsatz
3	Gartenbau OHG	Desktop PC	1	6.214,00 €
4	Seegesellschaft mbH	Desktop PC	6	47.570,00 €
5	Floyd AG	Desktop PC	6	5.723,00 €
6	Gartenbau OHG	Notebook	5	6.214,00 €
7	Maier & Sohn KG	Monitor	6	7.321,00 €
8	Zeitungsverlag	Notebook	5	1.533,00 €
9	Baum GmbH	Notebook	4	3.241,00 €
10	Seegesellschaft mbH	Notebook	4	2.145,00 €
11	TeGeKa mbH	Notebook	3	12.412,00 €
12	Floyd AG	Notebook	6	5.723,00 €

Gesamtumsatz:	5.723,00 €
der Firma:	Floyd AG
für das Produkt:	Desktop PC
im Monat:	6

Abbildung 15.40: Listensummen mit Mehrfachfilter bilden

In den Zellen I7 bis I9 stehen die Filterkriterien. In die Zelle I5 geben Sie folgende Formel ein: =SUMMEWENNS(E3:E45;B3:B45;I7;C3:C45;I8;D3:D45;I9)

- ▶ Als Erstes wird der *Summe_Bereich* bestimmt, in dem die zu addierenden Werte stehen (E3:E45). Die Liste ist länger als in Abbildung 15.40 dargestellt.
- ▶ Es folgt der erste Bereich, aus dem das erste Kriterium herausgefiltert werden soll (B3:B45).
- ▶ Anschließend wird das erste Suchkriterium, nach dem summiert wird, festgelegt (I7).
- ▶ Es geht weiter mit dem zweiten Kriterienbereich (C3:C45), dem zweiten Kriterium (I8) usw.

Die Floyd AG hat im Monat 6 mit dem Produkt »Desktop PC« 5.723 € umgesetzt.

Siehe auch SUMME(), SUMMEWENN() sowie MITTELWERTWENN(), MITTELWERTWENNS(), ZÄHLENWENN() (»Statistische Funktionen«)



Das Beispiel finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt SUMMEWENNS.

SUMMEX2MY2()



SUMMEX2MY2(*Matrix_x*; *Matrix_y*)

Die Funktion summiert für zusammengehörige Komponenten zweier Matrizen die Differenzen der Quadrate.

Matrix_x (erforderlich) ist die erste Matrix bzw. der erste Wertebereich.

Matrix_y (erforderlich) ist die zweite Matrix bzw. der zweite Wertebereich.

Als Argumente müssen in die Formel SUMMEX2MY2() Zahlen, Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 (null) enthalten, werden dagegen berücksichtigt.

Enthalten *Matrix_x* und *Matrix_y* unterschiedlich viele Werte, gibt SUMMEX2MY2() den Fehlerwert #NV zurück.

Die Gleichung für die Summe der aus Quadratzahlen gebildeten Differenzen lautet:

$$\sum(x^2 - y^2)$$

1. Aus zwei festgelegten Bereich mit Werten ...

Bereich A: 4, 5

Bereich B: 2, 3

2. werden zuerst für jeden Wert die Quadratzahlen ermittelt:

Bereich A: 16, 25

Bereich B: 4, 9

3. Anschließend werden die Quadratzahlen für jeden Bereich addiert und ...

Bereich A: 41

Bereich B: 13

4. Summen der Quadratzahlen voneinander subtrahiert:

41 – 13 = 28

	A	B
1	Bereich A:	Bereich B:
2	4	2
3	5	3
4		
5	Ergebnis	28
6		

Abbildung 15.41: Darstellung der Formel SUMMEX2MY2()

Diese Funktion benötigen Sie, wenn Sie statistische Prüfungen anstellen, in denen es um die Parallelität von Modellen, deren Richtigkeit oder Relevanz geht. Die Rede ist von Anwendungen der Varianzanalyse wie

- ▶ dem Entdecken von Variationen in Messserien oder
- ▶ der Analyse von linearen Modellen (Signifikanz von Parametern, Güte des Modells)

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

In diesen Verfahren werden die Quadratsummen gebildet und die Freiheitsgrade (F-Werte) für die Differenzen der Quadratsummen gebildet. Die Funktion SUMMEX2MY2() bildet einen guten Baustein innerhalb solcher Berechnungen. Lesen Sie mehr darüber unter Funktionen wie FTEST(), FVERT() und VARIANZ() bzw. VARIANZEN() in Kapitel 11 »Statistische Funktionen«.

Siehe auch SUMMENPRODUKT(), SUMMEX2PY2(), SUMMEXMY2()



Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe Kap15.xlsx auf dem Arbeitsblatt SummeXY.

SUMMEX2PY2() SUMX2PY2()

Syntax SUMMEX2PY2(*Matrix_x*; *Matrix_y*)

Definition Die Funktion summiert für zusammengehörige Komponenten zweier Matrizen die Summen der Quadrate. Die Gesamtsumme aus der Summe von Quadratzahlen ist ein Ausdruck, der häufig in statistischen Berechnungen verwendet wird.

Argumente *Matrix_x* ist die erste Matrix bzw. der erste Wertebereich.

Matrix_y ist die zweite Matrix bzw. der zweite Wertebereich.

Hintergrund Als Argumente der Formel SUMMEX2PY2() müssen Zahlen, Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Enthalten *Matrix_x* und *Matrix_y* unterschiedlich viele Werte, gibt SUMMEX2PY2() den Fehlerwert #NV zurück.

Die Gleichung für die Summe der aus Quadratzahlen gebildeten Differenzen lautet:

$$\sum (x^2 - y^2)$$

1. Aus zwei festgelegten Bereichen mit Werten ...

Bereich A: 4, 5

Bereich B: 2, 3

2. ... werden zuerst für jeden Wert die Quadratzahlen ermittelt:

Bereich A: 16, 25

Bereich B: 4, 9

3. Anschließend werden die Quadratzahlen für jeden Bereich addiert und ...

Bereich A: 41

Bereich B: 13

4. ... Summen der Quadratzahlen wieder miteinander addiert:

41 + 13 = 54

	A	B
1	Bereich A:	Bereich B:
2	4	2
3	5	3
4		
5	Ergebnis	54

Abbildung 15.42: Quadratsumme für zusammengehörige Komponenten von Matrizen

SUMMENPRODUKT(), SUMMEX2MY2(), SUMMEXMY2()

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `SummeXY`.

Siehe auch



SUMMEXMY2()



SUMXMY2()

SUMMEXMY2(*Matrix_x*; *Matrix_y*)

Die Funktion summiert für zusammengehörige Komponenten zweier Matrizen die quadrierten Differenzen.

Matrix_x ist die erste Matrix oder der erste Wertebereich.

Matrix_y ist die zweite Matrix oder der zweite Wertebereich.

Als Argumente müssen in die Formel SUMMEXMY2() Zahlen, Namen, Matrizen oder Bezüge angegeben werden, die Zahlen enthalten.

Enthält ein als Matrix oder Bezug angegebenes Argument Text, Wahrheitswerte oder leere Zellen, werden diese Werte ignoriert. Zellen, die den Wert 0 enthalten, werden dagegen berücksichtigt.

Enthalten *Matrix_x* und *Matrix_y* unterschiedlich viele Werte, gibt SUMMEXMY2() den Fehlerwert #NV zurück.

Die Gleichung für die Summe quadrierter Differenzen lautet:

$$\sum (x - y)^2$$

1. Aus zwei festgelegten Bereichen mit Werten ...

Bereich A: 4, 5

Bereich B: 2, 3

2. ... werden die jeweils gegenüberstehenden Werte voneinander abgezogen und ...

Erster Wert der Bereiche A und B: 4 – 2

Zweiter Wert der Bereiche A und B: 5 – 3

3. ... die Quadratzahlen der Ergebnisse addiert:

$$2^2 + 2^2 = 4 + 4 = 8$$

	A	B
1	Bereich A:	Bereich B:
2	4	2
3	5	3
4		
5	Ergebnis	8

Abbildung 15.43: Darstellung der Funktion SUMMEXMY2()

Syntax

Definition

Argumente

Hintergrund

Siehe auch SUMMENPRODUKT(), SUMMEX2MY2(), SUMMEX2PY2()



Das Beispiel finden Sie im Ordner `\\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `SummeXY`.

TAN() TAN()

Syntax TAN(*Zahl*)

Definition Die Funktion gibt den Tangens eines Winkels zurück.

Argumente *Zahl* (erforderlich) ist der Winkel im Bogenmaß, für den Sie den Tangenswert berechnen möchten.

Hintergrund In einem rechtwinkligen Dreieck ist das Verhältnis von Gegenkathete zur Ankathete abhängig vom Öffnungswinkel. Das Verhältnis der Gegenkathete eines Winkels zur Ankathete ist der Tangens (tan) eines Winkels.

$$\tan \alpha = \frac{a}{b}; \tan \beta = \frac{b}{a};$$

Die Funktion TAN() benötigt als Rechengrundlage das Bogenmaß. Liegt der Winkel im Gradmaß vor, müssen Sie ihn durch Multiplizieren mit $\text{PI}()/180$ oder der Funktion BOGENMASS() in das Bogenmaß überführen.

Bei der Darstellung der Abhängigkeit des Tangens vom Winkel α in einem Einheitskreis ($c = 1$) nimmt der Winkel α von 0° bis 90° zu, so nimmt der Tangens von 0 bis ∞ (Unendlich) zu.

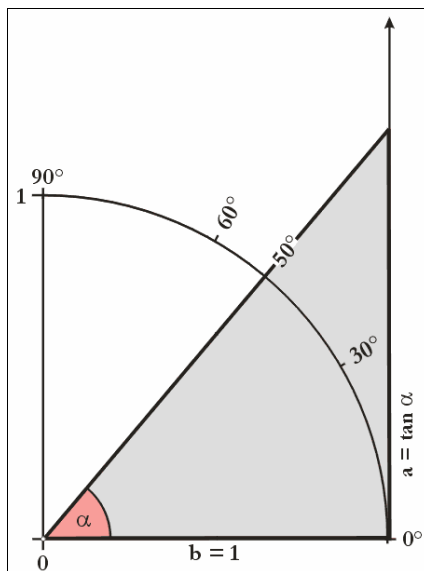


Abbildung 15.44: Der Tangens im Einheitskreis

Beim Übertrag der Werte in ein Koordinatensystem, mit den Werten des Winkels α auf der x-Achse und dem Tangens von α auf der y-Achse, ergeben sich folgende Kurven für den gesamten Kreis.

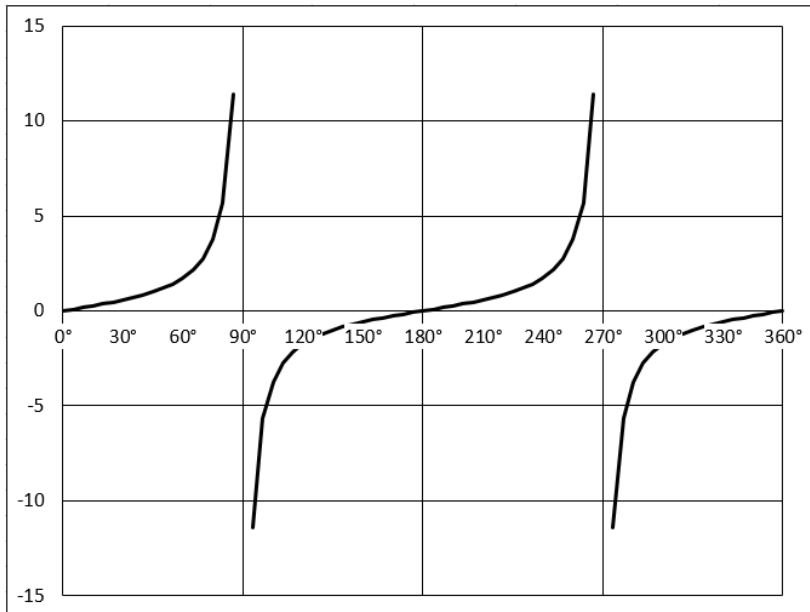


Abbildung 15.45: Tangenskurve

Zu den Funktionen Sinus und Kosinus steht der Tangens in der Beziehung:

$$\tan = \frac{\sin}{\cos}$$

Dem Tangens kommt eine ganz besondere Stellung zu, da er den Zusammenhang zwischen dem Steigungswinkel und dem Anstieg (der Steigung) einer Geraden ausdrückt.

Ist also beispielsweise der Steigungswinkel einer Straße 12° , so ist ihr Anstieg $\tan(12^\circ)$, was näherungsweise $0,21$ ist. Die Verkehrstafel, die die Steigung der Straße anzeigt, wird dann die Aufschrift »21%« tragen (was als »21 Meter Höhenunterschied pro 100 Meter laut Straßenkarte zurückgelegter Entfernung« gelesen werden kann).

Für eine senkrechte Gerade ist der Anstieg kein sinnvolles Konzept, was genau der Tatsache, dass $\tan(90^\circ)$ und $\tan(-90^\circ)$ nicht definiert sind, entspricht.

Die Höhe eines Schornsteins soll berechnet werden. Der Schornstein ist vom Betrachter aus 130 m entfernt und die Spitze erscheint unter dem Erhebungswinkel von $\alpha=24^\circ 50'$. Wie hoch ist der Schornstein, wenn die Augenhöhe $1,6$ m beträgt?

Praxiseinsatz

Die Formel für den Tangens lautet:

$$\tan() = \frac{\text{Gegenkathete}}{\text{Ankathete}}$$

Die Formel wird mit den vorhandenen Daten ausgefüllt,

$$\tan(24,5) = \frac{\text{Schornsteinhöhe}(h)}{130}$$

und für die zu berechnende Höhe umgestellt.

$$h = 130 \cdot \tan(24,5)$$

Das Ergebnis wird auf zwei Stellen hinter dem Komma gerundet und die 1,60 m für die Augenhöhe addiert.

$$h = 59,2444132 \approx 59,24 \text{ m} + 1,60 \text{ m (Augenhöhe)} = 60,84 \text{ m}$$

In die Bearbeitungsleiste von Excel wird folgende Formel eingegeben:

```
=RUNDEN(TAN(BOGENMASS(24,5))*130;2)+1,6
```

Damit Excel den Tangens des Winkels 19,5° berechnen kann, muss dieser zuerst in ein Bogenmaß umgerechnet werden. Anschließend wird der Tangens mit der Gegenkathete multipliziert. Das Ergebnis wird auf zwei Stellen hinter dem Komma gerundet und zum Schluss wird noch 1,6 für die Augenhöhe addiert

Siehe auch ARCTAN2(), ATAN(), PI()



Die Tabelle für die grafische Darstellung aus Abbildung 15.45 und die oben genannten Beispielwerte finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *TAN* sowie auf dem Arbeitsblatt *trigon* mit anderen Anwendungsbeispielen.

TANHYP() TANH()

Syntax TANHYP(*Zahl*)

Definition Gibt den hyperbolischen Tangens (Hyperbeltangens, Tangens Hyperbolicus, math. *tanh*) einer reellen Zahl zurück.

Argumente *Zahl* (erforderlich) ist eine beliebige reelle Zahl.

Hintergrund Der hyperbolische Tangens gehört zu den Hyperbelfunktionen, die ebenso wie die Kreisfunktionen (z.B. Sinus und Tangens) für alle reellen und komplexen Zahlen definiert sind (Microsoft Excel erlaubt aber nur reelle Argumente für diese Hyperbelfunktion.) Die Definition für den hyperbolischen Tangens lautet:

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{\sinh x}{\cosh x}$$

wobei die Ähnlichkeit zu den trigonometrischen Funktionen beim letzten Term auffällig ist. Seine grafische Darstellung zeigt die Abbildung 15.46.

Der hyperbolische Tangens wird in der Technik und in den Naturwissenschaften, vor allem in der Forschung und Entwicklung, häufig gebraucht. Populärwissenschaftliche Anwendungen sind nicht zu finden.

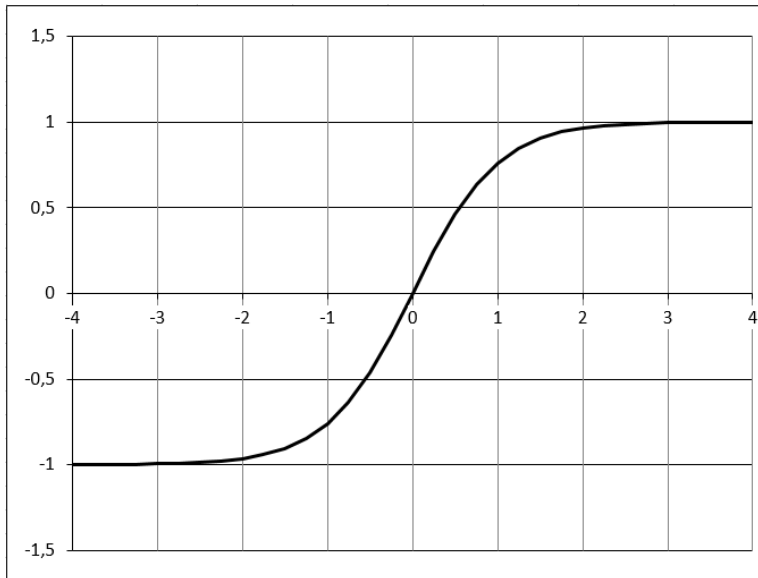


Abbildung 15.46: Der Hyperbeltangens als Grafik

Die Ausbreitungsgeschwindigkeit von Wasserwellen bedient sich des Hyperbeltangens. Die Formel für die Ausbreitungsgeschwindigkeit v von Wasserwellen lautet

Praxiseinsatz

$$v = \sqrt{\frac{g\lambda}{2\pi}} \tanh \frac{2\pi h}{\lambda}$$

mit der Erdbeschleunigung g [m/s²], der Wellenlänge λ [m] und der Wassertiefe h [m]. Wenn Sie diese Formel in einer Tabelle verwenden, können Sie die beiden folgenden Näherungsformeln für seichtes und tiefes Wasser verifizieren:

$$\text{Seichtes Wasser: } h \ll \lambda \rightarrow v_s = \sqrt{gh}$$

$$\text{Tiefes Wasser: } h \gg \lambda \rightarrow v_t = \sqrt{\frac{g\lambda}{2\pi}}$$

Hier weitere Beispiele, um die Funktion zu dokumentieren:

=TANHYP(0) ergibt 0.

=TANHYP(1) ergibt 0,761594156.

=TANHYP(-1) ergibt -0,76159416.

=TANHYP(10) ergibt 1.

=TANHYP(-10) ergibt -1.

ARCTANHYP(), COSHYP(), SINHYP()

Siehe auch

Die Tabelle für die grafische Darstellung und die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *TANHYP.xlsx* auf den Arbeitsblättern *Graph* und *Wasserwellen*.



TEILERGEBNIS() SUBTOTAL()

Syntax TEILERGEBNIS(*Funktion*; *Bezug1*; *Bezug2*; ...)

Definition Die Funktion gibt ein Teilergebnis über eine Liste oder Datenbank zurück. Grundsätzlich ist es einfacher, eine mit Teilergebnissen versehene Liste mithilfe des Befehls *DATEN/Teilergebnisse* zu erstellen. Nachdem eine solche mit Teilergebnissen versehene Liste erstellt wurde, können Sie diese mit der Funktion TEILERGEBNIS() bearbeiten.

Argumente *Funktion* (erforderlich) ist eine Zahl von 1 bis 11 (bezieht ausgeblendete Werte ein) oder von 101 bis 111 (ignoriert ausgeblendete Werte), die festlegt, welche Funktion bei der Berechnung des Teilergebnisses innerhalb einer Liste verwendet werden soll.

Tabelle 15.6
Codetabelle
der Funktion
TEILERGEBNIS()

Code für eingeblendet Werte	Code für ausgeblendet Werte	Funktion
1	101	MITTELWERT
2	102	ANZAHL
3	103	ANZAHL2
4	104	MAX
5	105	MIN
6	106	PRODUKT
7	107	STABW
8	108	STABWN
9	109	SUMME
10	110	VARIANZ
11	111	VARIANZEN

Bezug1; *Bezug2*; ... (mind. ein Bezugsbereich erforderlich) ist der Bereich oder Bezug, für den Teilergebnisse berechnet werden sollen.

Hintergrund Während bei einer »einfachen« Formel herausgefilterte Werte mit einbezogen werden, kann mit der Funktion TEILERGEBNIS() ein Bezug auf gefilterte Ergebnisse gebildet werden. Werden innerhalb der mit *Bezug1*; *Bezug2*; ... angegebenen Bereiche weitere Teilergebnisse (oder geschachtelte Teilergebnisse) berechnet, werden diese geschachtelten Teilergebnisse ignoriert, damit sie nicht mehrfach berücksichtigt werden.

Handelt es sich bei einem Bezug um einen 3D-Bezug, gibt TEILERGEBNIS() den Fehlerwert #WERT! zurück.

Durch das Einsetzen der Funktionskonstanten von 1 bis 11 bezieht die Funktion TEILERGEBNIS() die über *DATEN/Gliederung/Detail ausblenden* ausgeblendeten Datenwerte mit ein. Verwenden Sie diese Konstanten, wenn Sie das Teilergebnis von ausgeblendeten und nicht ausgeblendeten Zahlen in einer Liste berechnen möchten.

Bei den Funktionskonstanten von 101 bis 111 ignoriert die Funktion TEILERGEBNIS() Datenwerte, die über *DATEN/Gliederung/Detail ausblenden* ausgeblendet wurden. Verwenden Sie diese Konstanten, wenn das Teilergebnis von ausschließlich nicht ausgeblendeten Zahlen in einer Liste berechnet werden soll.

Die Funktion TEILERGEBNIS() ignoriert alle Zeilen, die nicht im Ergebnis eines Filters enthalten sind, unabhängig vom verwendeten Funktionswert.

Die Funktion TEILERGEBNIS() wurde für Datenspalten oder vertikale Bereiche konzipiert, jedoch nicht für Zeilen mit Daten oder horizontale Bereiche. Wenn Sie beispielsweise das Teilergebnis eines horizontalen Bereichs mithilfe der Funktionskonstanten 101 oder höher berechnen, wie z.B. =TEILERGEBNIS(109,B2:G2), hat das Ausblenden einer Spalte keine Auswirkungen auf das Teilergebnis. Wird hingegen eine Zeile im Teilergebnis eines vertikalen Bereichs ausgeblendet, wirkt sich das auf das Teilergebnis aus.

Mit der Funktion TEILERGEBNIS() können Sie z.B. für eine Umsatzliste Summen auf gefilterte Werte bilden:

Praxiseinsatz

	A	B	C	D	E
1	Bundesland	Firma	Umsatz	Umsatz Summe:	485.694,00 €
2	NRW	Michel AG	51.857,00 €		
3	Bayern	Hummer & Sohn GmbH	30.090,00 €		
4	Hamburg	Müller GmbH & Co. KG	85.164,00 €		
5	NRW	Lotus GmbH	84.843,00 €		
6	Bremen	Sahne KG	46.536,00 €		
7	Sachsen	Michel Delias	71.731,00 €		
8	Hamburg	Himmelsberger AG	51.560,00 €		
9	Berlin	Möchtegern AG	63.913,00 €		

Abbildung 15.47: Die Umsatzsumme in E1 richtet sich nach der Filterung der Liste

Die Formel für »Umsatz Summe« lautet: =TEILERGEBNIS(9;C2:C9)

Wenn Sie jetzt die Umsätze z.B. für die Region NRW ansehen wollen, brauchen sie nur den entsprechenden Filter in der Spalte A setzen und erhalten folgendes Ergebnis:

	A	B	C	D	E
1	Bundesland	Firma	Umsatz	Umsatz Summe:	136.700,00 €
2	NRW	Michel AG	51.857,00 €		
5	NRW	Lotus GmbH	84.843,00 €		

Abbildung 15.48: Umsatzliste mit »Summe« als Teilergebnis

ANZAHL(), ANZAHL2(), MAX(), MIN(), MITTELWERT(), STABW(), STDAN(), AGGREGAT(), VARIANZ() (alles »Statistische Funktionen«), PRODUKT(), SUMME()

Siehe auch

Ein Beispiel finden Sie im Ordner \\Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *TEILERGEBNIS*.



UNGERADE() ODD()

Syntax UNGERADE(*Zahl*)

Definition Die Funktion rundet eine Zahl auf die nächste ungerade ganze Zahl auf.

Argumente *Zahl* (erforderlich) ist der Wert, der aufgerundet werden soll.

Hintergrund Ungerade Zahlen sind alle Zahlen, die sich nicht ganzzahlig durch 2 teilen lassen. Die Funktion UNGERADE() rechnet nur mit numerischen Werten.

Unabhängig davon, welches Vorzeichen *Zahl* hat, wird ein Wert aufgerundet, wenn die Anpassung von null ausgehend erfolgt. Ist *Zahl* eine ungerade ganze Zahl, erfolgt keine Rundung.

Hier noch ein paar Beispiele, um die Funktion zu dokumentieren:

=UNGERADE(1,9) ergibt 3

=UNGERADE(-2,8) ergibt -3

=UNGERADE(18) ergibt 19

Siehe auch GERADE(), GANZZAHL(), ISTGERADE() (Informationsfunktionen), ISTUNGERADE() (Informationsfunktionen), OBERGRENZE(), OBERGRENZE.GENAU(), RUNDEN(), KÜRZEN(), UNTERGRENZE(), UNTERGRENZE.GENAU()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *UNGERADE*.

UNTERGRENZE() FLOOR()

Syntax UNTERGRENZE(*Zahl*; *Schritt*)

Definition Die Funktion rundet eine Zahl auf das kleinste Vielfache einer anderen Zahl ab.

Argumente *Zahl* (erforderlich) ist der Wert, den Sie runden möchten.

Schritt (erforderlich) ist das Vielfache, auf das Sie runden möchten.

Hintergrund Die Funktion UNTERGRENZE(), genauso wie die Funktion UNTERGRENZE.GENAU() (s.u.), sollen durch die einfacher einzusetzende Funktion UNTERGRENZE.MATHEMATIK() auf Seite 795 ersetzt werden. Lesen Sie bitte dort weitere Erklärungen.

UNTERGRENZE.GENAU() FLOOR.PRECISE()

Syntax UNTERGRENZE.GENAU(*Zahl*; *Schritt*)

Definition Rundet eine Zahl auf die nächste ganze Zahl oder das nächste Vielfache von *Schritt* auf. Die Zahl wird unabhängig vom Vorzeichen abgerundet. Wenn die Zahl oder der *Schritt* jedoch Null ist, wird Null zurückgegeben.

Argumente *Zahl* (erforderlich) ist der Wert, den Sie runden möchten.

Schritt (optional) ist das Vielfache, auf das Sie runden möchten.

Wenn *Schritt* ausgelassen wird, ist der Standardwert 1.

Die Funktion `UNTERGRENZE.GENAU()`, genauso wie die Funktion `UNTERGRENZE()` (s.o.), sollen durch die einfacher einzusetzende Funktion `UNTERGRENZE.MATHEMATIK()` auf Seite 795 ersetzt werden. Lesen Sie bitte dort weitere Erklärungen.

Hintergrund

UNTERGRENZE.MATHEMATIK()

**FLOOR.MATH()**

`UNTERGRENZE.MATHEMATIK` (*Zahl*; *Schritt*; *Modus*)

Rundet eine Zahl auf die nächste ganze Zahl oder auf das nächste Vielfache von *Schritt* ab.

Zahl (erforderlich) ist die Zahl, die gerundet werden soll.

Schritt (optional) ist der Wert für das Vielfache, auf das *Zahl* aufgerundet werden soll.

Modus (optional) steuert für *Zahl* $\neq 0$, ob *Zahl* in Richtung des größeren oder des kleineren Werts gerundet wird.

Diese Funktion gibt Ihnen die Möglichkeit, Zahlen auf eine bestimmte Intervallgrenze abzurunden, wobei Sie das Intervall angeben können. Diese Herangehensweise kommt besonders bei kaufmännischen Berechnungen vor. Natürlich wäre diese Art der Berechnung auch mit den anderen Runden-Funktionen in einer eigenen Formelkonstruktion möglich, aber `UNTERGRENZE()` stellt den einfachsten Weg dar.

Die Funktionen `UNTERGRENZE.GENAU()` aus Excel 2010 und `UNTERGRENZE()` wurden in Excel 2013 durch die neue Funktion `UNTERGRENZE.MATHEMATIK()` ersetzt, um das Verhalten im Fall von negativen Werten besser steuern zu können.

Die Funktion `UNTERGRENZE.GENAU()` ist weder in den Kategorien des Menübands noch in den Listen des Funktions-Assistenten noch aufgeführt. Sie können sie aber weiter nutzen; Sie müssen sie dann händisch mit komplettem Namen in die Formel eingeben.

Die Funktion `UNTERGRENZE()` können Sie ebenfalls weiter nutzen; sie befindet sich jetzt in der Kategorie *Kompatibilität*.

Bei der Funktion `UNTERGRENZE.MATHEMATIK()` kann über das Argument *Modus* die Rundungsrichtung geändert werden:

- ▶ Bei *Modus* = 0 (*FALSCH* oder weggelassen) bewegt sich das Rundungsergebnis zu kleineren Zahlen hin.
- ▶ Bei *Modus* = -1 (*WAHR* oder eine andere von 0 verschiedene Zahl) bewegt sich das Ergebnis zu kleineren Beträgen, also zu 0 hin.

Bei der Funktion `UNTERGRENZE.GENAU()` wird immer zu kleineren Beträgen (»abwärts«) gerundet.

Die Funktion `UNTERGRENZE()` arbeitet etwas uneinheitlich (siehe Tabelle 15.7).

Bei einem Fahrradhersteller kann ein Großhändler nur mindestens 5 Fahrräder bestellen. Immer wenn der Lagerbestand unter 0 fällt, soll bestellt werden. Dabei kann der Lagerbestand unter 0 fallen, wenn keine Räder mehr auf Lager sind und Bestellungen durch den Einzelhandel vorliegen. Angenommen der Lagerbestand sei -6, d.h. 6 Fahrräder sind von Einzel-

Syntax**Definition****Argumente****Hintergrund****Hinweis****Praxiseinsatz**

händlern bestellt. Damit ergibt sich für den Großhändler beim Hersteller eine Bestellmenge von $\text{UNTERGRENZE.MATHEMATIK}(-6;-5) = -10$ bzw. $\text{ABS}(-10) = 10$ Fahrräder. Die beiden Formeln kann man auch zusammenfassen:

$$=\text{ABS}(\text{UNTERGRENZE.MATHEMATIK}(-6;-5))$$

Hier noch ein paar Beispiele, um die Funktion zu dokumentieren:

$$=\text{UNTERGRENZE.MATHEMATIK}(2345,47;0,5) \text{ ergibt } 2.345,00 \text{ €.}$$

$$=\text{UNTERGRENZE.MATHEMATIK}(2345,67;0,5) \text{ ergibt } 2.345,50 \text{ €.}$$

Tabelle 15.7
Vergleich der drei Varianten der Funktion
UNTERGRENZE

Formel	Ergebnis
=UNTERGRENZE.MATHEMATIK(3,6;1)	3
=UNTERGRENZE.MATHEMATIK(3,6;-1)	3
=UNTERGRENZE.MATHEMATIK(-3,6;1)	-4
=UNTERGRENZE.MATHEMATIK(-3,6;-1)	-4
=UNTERGRENZE.MATHEMATIK(3,6;1;1)	3
=UNTERGRENZE.MATHEMATIK(3,6;-1;1)	3
=UNTERGRENZE.MATHEMATIK(-3,6;1;1)	-3
=UNTERGRENZE.MATHEMATIK(-3,6;-1;1)	-3
=UNTERGRENZE.MATHEMATIK(3,6;1;-1)	3
=UNTERGRENZE.MATHEMATIK(3,6;-1;-1)	3
=UNTERGRENZE.MATHEMATIK(-3,6;1;-1)	-3
=UNTERGRENZE.MATHEMATIK(-3,6;-1;-1)	-3
=UNTERGRENZE(3,6;1)	3
=UNTERGRENZE(3,6;-1)	#ZAHL!
=UNTERGRENZE(-3,6;1)	-4
=UNTERGRENZE(-3,6;-1)	-3
=UNTERGRENZE.GENAU(3,6;1)	3
=UNTERGRENZE.GENAU(3,6;-1)	3
=UNTERGRENZE.GENAU(-3,6;1)	-4
=UNTERGRENZE.GENAU(-3,6;-1)	-4

Siehe auch ABRUNDEN(), AUFRUNDEN(), GANZZAHL(), GERADE(), KÜRZEN(), OBERGRENZE(), OBERGRENZE.GENAU(), OBERGRENZE.MATHEMATIK(), RUNDEN(), UNGERADE(), UNTERGRENZE(), UNTERGRENZE.GENAU()



Eine Beispielreihe mit alternativen Prüfberechnungen finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `UNTERGRENZE`.

VORZEICHEN() SIGN()

VORZEICHEN(*Zahl*)

Die Funktion gibt das Vorzeichen einer Zahl zurück. Die Funktion gibt folgende Werte zurück:

- ▶ 1, wenn die Zahl positiv ist;
- ▶ 0 (null), wenn die Zahl 0 ist;
- ▶ -1, wenn die Zahl negativ ist.

Zahl (erforderlich) ist eine beliebige reelle Zahl.

In der Mathematik werden Zahlen, die größer als 0 (null) sind, als »Positive Zahlen« bezeichnet und mit einem »+« (Plus) gekennzeichnet. Zahlen kleiner als 0 bezeichnet man als »Negative Zahlen«. Vor negativen Zahlen steht immer ein »-« (Minus)-Zeichen. Das »+« Zeichen vor positiven Zahlen kann weggelassen werden. Die Zahl 0 (null) ist weder positiv noch negativ.

Durch die Formel VORZEICHEN() lassen sich beispielsweise alle Werte mit einem bestimmten Vorzeichen herausfiltern.

Aus einer Liste mit Gewinnen sollen die Niederlassungen herausgefiltert werden, die negative Beträge erwirtschaften. Zusätzlich soll die Summe der Verluste separat ausgewiesen werden.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

	A	B	C	D	E
1	Niederlassung	Gewinn	Vorzeichen		
2	Wien	723.672,00 €	1		
3	Bremen	-15.423,00 €	-1		
4	Hamburg	12.489,00 €	1		
5	Frankfurt	124.875,00 €	1		
6	Paris	-23.534,00 €	-1		
7	Essen	62.335,00 €	1		
8	München	-44.347,00 €	-1		
9	Berlin	235.612,00 €	1		
10					
11				Verluste	-83.304,00 €
12				Gewinn	1.158.983,00 €
13				Saldo	1.075.679,00 €

Abbildung 15.49: Gewinnaufstellung mit unterschiedlichen Vorzeichen

In der Spalte C fügen Sie die Formel VORZEICHEN() mit Bezug auf den »Gewinn« ein (siehe Abbildung 15.49).

Jetzt haben Sie die Möglichkeit, sich über die Filterfunktion in der Spalte C alle Gewinne mit negativem Vorzeichen herauszufiltern und separat zu betrachten:

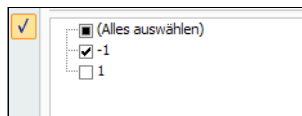


Abbildung 15.50: Filter von Werten mit negativem Vorzeichen

Um die Summe der Verluste auszuweisen, wird in das Feld *E11* folgende Formel eingegeben:

```
{=SUMME(WENN(VORZEICHEN(B2:B9)=-1;B2:B9))}
```

Für die Gewinne ist folgende Formel eingegeben worden:

```
{=SUMME(WENN(VORZEICHEN(B2:B9)=1;B2:B9))}
```

Beide Formeln müssen als Matrixformel eingegeben werden, d.h. Sie schließen die Formel-eingabe mit den Tasten `[Strg]` + `[↕]` + `[↵]` ab.

Siehe auch ABS()



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *VORZEICHEN*.

VRUNDEN() MROUND()

Syntax VRUNDEN(*Zahl*; *Vielfaches*)

Definition Die Funktion gibt eine auf das gewünschte Vielfache gerundete Zahl zurück.

Argumente *Zahl* (erforderlich) ist der Wert, der gerundet werden soll.

Vielfaches (erforderlich) ist das Vielfache, auf das Sie *Zahl* runden möchten.

Hintergrund Die Funktion VRUNDEN() ermöglicht, eine Zahl auf das Vielfache eines bestimmten Werts zu runden. VRUNDEN() rundet auf (weg von null), wenn der Rest der Division von Zahl durch Vielfaches größer oder gleich der Hälfte von Vielfaches ist. Ist die Zahl kleiner oder gleich der Hälfte des Vielfachen, wird abgerundet.

Praxiseinsatz Für die Auszeichnung im Geschäft sollen die Verkaufspreise alle einen Endwert haben, der sich durch 0,05 teilen lässt. Beispiele: 10,05 €, 11,10 €, 15,25 € oder 20,00 €. Für die Verkaufspreise 12,31 €, 0,32 € und 1,24 € sollen die Werte für die Auszeichnung ermittelt werden:

=VRUNDEN(12,31;0,05) ergibt einen Auszeichnungspreis von 12,30 €.

=VRUNDEN(0,32;0,05) ergibt einen Auszeichnungspreis von 0,30 €.

=VRUNDEN(1,24;0,05) ergibt einen Auszeichnungspreis von 1,25 €.

Siehe auch ABRUNDEN(), AUFRUNDEN(), GERADE(), KÜRZEN(), ABRUNDEN(), AUFRUNDEN(), OBERGRENZE.MATHEMATIK(), REST(), RUNDEN(), UNGERADE(), UNTERGRENZE.MATHEMATIK()



Die Beispiele finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *VRUNDEN*.

WURZEL()



WURZEL(*Zahl*)

Die Funktion gibt die Quadratwurzel von *Zahl* zurück.

Zahl (erforderlich) ist die Zahl, deren Quadratwurzel Sie berechnen möchten.

Die Wurzelfunktion ist neben dem Logarithmus eine Umkehrfunktion des Potenzierens.

$$a^x = b \Leftrightarrow \sqrt[x]{b} = a$$

Die Basis (*b*) wird als »Radix« und *x* als »Wurzelexponent« bezeichnet.

Ist der Wurzelexponent 2, spricht man von einer Quadratwurzel.

Die Funktion WURZEL() gibt nur die Quadratwurzel einer positiven *Zahl* wieder. Wenn *Zahl* negativ ist, gibt WURZEL() den Fehlerwert #ZAHLE! zurück.

Die Seiten eines rechteckigen Bauplatzes sind 19,5 m x 10,5 m. Sie möchten jetzt wissen, wie lang die Seite eines flächengleichen quadratischen Bauplatzes ist.

Sowohl die Formel

=WURZEL(19,5*10,5)

als auch

=WURZEL(PRODUKT(19,5;10,5))

ergeben 14,309088. Um das Ergebnis gleich auf zwei Stellen hinter dem Komma zu runden, können Sie zusätzlich noch die Formel RUNDEN() einsetzen:

=RUNDEN((WURZEL(PRODUKT(19,5;10,5)));2)

Die Formel löst sich wie folgt auf:

1. Bilden Sie das Produkt von 19,5 und 10,5:

=PRODUKT(19,5;10,5)

2. Ziehen Sie die Wurzel aus dem Produkt:

=WURZEL(PRODUKT(19,5;10,5))

3. Runden Sie das Ergebnis auf zwei Stellen hinter dem Komma:

=RUNDEN((WURZEL(PRODUKT(19,5;10,5)));2)

Das Ergebnis lautet 14,31 m.

WURZELPI()

Das Beispiel finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *WURZEL*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



WURZELPI()



SQRTPI()

Syntax WURZELPI(*Zahl*)

Definition Die Funktion gibt die Wurzel aus der mit Pi (π) multiplizierten Zahl zurück.

Argumente *Zahl* (erforderlich) ist die Zahl, mit der Pi multipliziert wird.

Hintergrund Die Funktion WURZELPI() berechnet die Wurzel einer mit Pi – Näheres zu Pi siehe Funktion PI() – multiplizierten positiven Zahl. Ist $Zahl < 0$, gibt WURZELPI() den Fehlerwert #ZAHL! zurück. Es gilt

$$WURZELPI(x) = WURZEL(PI() \cdot x) = \sqrt{\pi \cdot x}$$

Hier noch ein paar Beispiele, um die Funktion zu dokumentieren:

=WURZELPI(12) ergibt 6,13996025.

=WURZELPI(5) ergibt 3,9633273.

Siehe auch PI(), WURZEL()



Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *WURZELPI*.

ZUFALLSBEREICH()



RANDBETWEEN()

Syntax ZUFALLSBEREICH(*Untere_Zahl*; *Obere_Zahl*)

Definition Die Funktion gibt eine ganze Zufallszahl aus dem festgelegten Bereich zurück. Bei jeder Neuberechnung der jeweiligen Tabelle wird eine neue Zufallszahl ausgegeben.

Argumente *Untere_Zahl* (erforderlich) ist die kleinste ganze Zahl, die ZUFALLSBEREICH als Ergebnis zurückgeben kann.

Obere_Zahl (erforderlich) ist die größte ganze Zahl, die ZUFALLSBEREICH als Ergebnis zurückgeben kann.

Hintergrund Hin und wieder benötigt man zufällige Werte, sei es für Beispieldarstellungen oder in der Naturwissenschaft/Technik für das Simulieren bestimmter Prozesse.

Zum Generieren einer Datumzufallszahl muss der jeweilige Wert im Zahlenformat vorliegen.

Tipp Für den Fall, dass Sie mit ZUFALLSBEREICH() zwar eine Zufallszahl erzeugen möchten, aber nicht wünschen, dass sich deren Wert bei jeder Neuberechnung der entsprechenden Zelle ändert, können Sie auch =ZUFALLSBEREICH(*Untere_Zahl*; *Obere_Zahl*) in die Bearbeitungsleiste eingeben und anschließend die [F9]-Taste drücken, um die Formel in eine von nun an feste Zufallszahl zu ändern.

Größere Mengen an Zellen mit einer Zufallszahl-Formel können Sie auch kopieren und anschließend in denselben Bereich mit der rechten Maustaste über *Inhalte einfügen/Werte* in Zahlenwerte umwandeln.

Siehe das Beispiel zur Funktion ZUFALLSZAHL().

ZUFALLSZAHL()

Das Beispiel finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *ZUFALL*.

ZUFALLSZAHL() RAND()

ZUFALLSZAHL()

Die Funktion gibt eine Zufallszahl zwischen 0 und 1 mit bis zu 16 Stellen hinter dem Komma zurück.

Keine

Die Funktion ZUFALLSZAHL() gibt zufällige Werte zwischen 0 und 1 zurück, wobei die 0 (null) als zufälliger Wert erscheinen kann, die 1 aber nie.

Mit der folgenden Anweisung kann eine reelle Zufallszahl erzeugt werden, die zwischen a und b liegt:

$=\text{ZUFALLSZAHL}()*(b-a)+a$

Bei jeder Neuberechnung in der jeweiligen Tabelle wird eine neue Zufallszahl erzeugt.

Diese Funktion wird recht häufig verwendet, um eine Tabelle mit Testdaten zu füllen oder in der Naturwissenschaft/Technik für das Simulieren bestimmter Prozesse. Dafür wird die Formel in jede Zelle eingefügt.

Für den Fall, dass Sie mit ZUFALLSZAHL() zwar eine Zufallszahl erzeugen möchten, aber nicht wünschen, dass sich deren Wert bei jeder Neuberechnung der entsprechenden Zelle ändert, können Sie auch =ZUFALLSZAHL() in die Bearbeitungsleiste eingeben und anschließend die **[F9]**-Taste drücken, um die Formel in eine Zufallszahl zu ändern.

Größere Mengen an Zellen mit einer Zufallszahl-Formel können Sie auch kopieren und anschließend in denselben Bereich mit der rechten Maustaste über *Inhalte einfügen/Werte* in Zahlenwerte umwandeln.

Beispiele:

$=\text{ZUFALLSZAHL}()*(100-1)+1$ ergibt eine Zufallszahl zwischen 1 und 100

$=\text{ZUFALLSZAHL}()*(250-150)+150$ ergibt eine Zufallszahl zwischen 150 und 250

ZUFALLSBEREICH()

Die Beispiele finden Sie im Ordner \Ms5-235\Kap15 in der Arbeitsmappe *Kap15.xlsx* auf dem Arbeitsblatt *ZUFALL*.

Praxiseinsatz

Siehe auch



Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Tip

Siehe auch



ZWEIFAKULTÄT()



FACTDOUBLE()

Syntax ZWEIFAKULTÄT(*Zahl*)

Definition Die Funktion gibt die Fakultät zu Zahl mit Schrittlänge 2 zurück.

Argumente *Zahl* (erforderlich) ist der Wert, für den die Fakultät mit Schrittlänge 2 berechnet werden soll. Ist *Zahl* keine ganze Zahl, werden die Nachkommastellen abgeschnitten.

Hintergrund Die Fakultät $n!$ einer Zahl n ist eine häufig benutzte Funktion, deren Definition lautet:

$$n! = n(n-1)(n-2)\dots(3)(2)(1)$$

Davon abgeleitet ist die Funktion ZWEIFAKULTÄT(), die in der Zahlentheorie eine große Rolle spielt. Sie ist für gerade Zahlen so definiert:

$$n!! = n(n-2)(n-4)\dots(4)(2)$$

und für ungerade Zahlen folgendermaßen:

$$n!! = n(n-2)(n-4)\dots(3)(1)$$

Die Funktion ZWEIFAKULTÄT() rechnet auf der Grundlage von numerischen Werten.

Praxiseinsatz =ZWEIFAKULTÄT(5) ergibt 15 (=5·3·1).

=ZWEIFAKULTÄT(8) ergibt 384 (=8·6·4·2).

Siehe auch FAKULTÄT(), POLYNOMINAL()



Beispiele finden Sie im Ordner `\Ms5-235\Kap15` in der Arbeitsmappe `Kap15.xlsx` auf dem Arbeitsblatt `FAKULTÄT`.

Kapitel 16

Konstruktionsfunktionen

Umwandlungsfunktionen	806
Funktionen für Bitoperationen	824
Funktionen für komplexe Zahlen	828
Funktionen der höheren Mathematik	846
Sprungfunktionen	856



In diesem Kapitel werden die Funktionen der Kategorie »Konstruktion« behandelt, die Sie auf der Registerkarte *FORMELN* des Menübands im Dropdownmenü zum Befehl *Mehr Funktionen* erreichen. Nach einem Mausklick auf *Mehr Funktionen* werden Ihnen sechs weitere Funktionskategorien angeboten: an zweiter Stelle finden Sie *Konstruktion*. In der Excel-Hilfe werden sie wieder (oder noch) als »Technische Funktionen« bezeichnet.

Die Kategorie »Konstruktion« lässt sich in fünf Gruppen einteilen:

- ▶ Umwandlungsfunktionen
- ▶ Funktionen für Bitoperationen
- ▶ Funktionen für komplexe Zahlen
- ▶ Funktionen der höheren Mathematik
- ▶ Sprungfunktionen

Abweichend von der alphabetischen Liste im Befehlsregister sollen die Funktionen in diesem Kapitel entsprechend dieser Gruppeneinteilung behandelt werden.

Umwandlungsfunktionen

Die Umwandlungsfunktionen helfen Ihnen, Zahlen aus einem Zahlensystem in ein anderes umzurechnen, z.B. von dem in der elektronischen Datenverarbeitung gebräuchlichen Hexadezimalsystem in das uns vertraute Dezimalsystem.

Mit *UMWANDELN()* steht Ihnen eine Funktion zur Verfügung, mit der Sie physikalische Maße ineinander umwandeln können. Das ist vor allem hilfreich, wenn es sich um anglo-amerikanische Maße handelt. Wem sind schon beispielsweise die Umrechnungsregeln für die verschiedenen Temperaturskalen geläufig? Diese Funktionen zu verstehen, sollte nicht allzu schwer sein; es fehlt im Allgemeinen nur an der Übung im Umgang mit den verschiedenen Zahlen- und Maßsystemen.

Funktionen für Bitoperationen

Die meisten Excel-Funktionen behandeln Zahlen als Ganzes; einige können Teile von Zahlen bearbeiten, wie die Funktionen zum Runden von Zahlen beispielsweise die Nachkommastellen verändern können.

Mit den Funktionen dieser Gruppe lassen sich Zahlen auf Bitebene bearbeiten.

Funktionen für komplexe Zahlen

Grundsätzlich lässt Excel in Zellen, Formeln oder Funktionen nur reelle Zahlen zu. Wollen Sie jedoch Berechnungen mit imaginären oder komplexen Zahlen durchführen, muss dafür ein Umweg über die Funktionen dieser Gruppe beschritten werden. Hier finden Sie alle dafür notwendigen Funktionen und ihre Beschreibungen.

Funktionen der höheren Mathematik

Die vierte Gruppe beinhaltet Funktionen, die typischerweise im Ingenieurwesen eingesetzt werden. Der englische Name »engineering functions« für diese Kategorie ist daher treffend gewählt. Für den Einsatz dieser Funktionen sollten Sie idealerweise über Kenntnisse in der höheren Mathematik verfügen, und zwar aus den Sachgebieten Differenzialgleichungen, Fehlerrechnung und Wahrscheinlichkeitsrechnung.

Sprungfunktionen

Hier befinden sich zwei Funktionen, die nicht in die üblichen Gruppierungen passen. (Mathematiker nennen sie Sprungfunktionen, was wir deswegen auch als Gruppentitel nehmen.)

Hinweis

Die Funktion *ZWEIFAKULTÄT()* befand sich bis zur Version 2003 in der Kategorie *Technisch* (jetzt: *Konstruktion*). Ab Excel-Version 2007 ist sie in der Kategorie *Mathematik & Trigonometrie* untergebracht und wird deswegen in Kapitel 15 erklärt.

Alle Funktionen der Kategorie »Konstruktion«

Umwandlungsfunktionen

Funktion	Beschreibung
BININDEZ()	Wandelt eine Binär- in eine Dezimalzahl um
BININHEX()	Wandelt eine Binär- in eine Hexadezimalzahl um
BININOKT()	Wandelt eine Binär- in eine oktale Zahl um
DEZINBIN()	Wandelt eine Dezimal- in eine binäre Zahl um
DEZINHEX()	Wandelt eine Dezimal- in eine Hexadezimalzahl um
DEZINOKT()	Wandelt eine Dezimal- in eine oktale Zahl um
HEXINBIN()	Wandelt eine Hexadezimal- in eine binäre Zahl um
HEXINDEZ()	Wandelt eine Hexadezimal- in eine Dezimalzahl um
HEXINOKT()	Wandelt eine Hexadezimal- in eine oktale Zahl um
OKTINBIN()	Wandelt eine oktale in eine binäre Zahl um
OKTINDEZ()	Wandelt eine oktale in eine Dezimalzahl um
OKTINHEX()	Wandelt eine oktale in eine Hexadezimalzahl um
UMWANDELN()	Führt Umrechnungen zwischen verschiedenen Maßsystemen durch

Funktionen für Bitoperationen

Funktion	Beschreibung
BITODER()	Gibt ein bitweises »Oder« zweier Zahlen zurück
BITLVERSCHIEB()	Gibt eine Zahl zurück, die um Verschiebebetrag Bits nach links verschoben ist
BITRVERSCHIEB()	Gibt eine Zahl zurück, die um Verschiebebetrag Bits nach rechts verschoben ist
BITUND()	Gibt ein bitweises »Und« zweier Zahlen zurück
BITXODER()	Gibt ein bitweises »Ausschließliches Oder« zweier Zahlen zurück

Funktionen für komplexe Zahlen

Funktion	Beschreibung
IMABS()	Berechnet den Absolutwert einer komplexen Zahl
IMAGINÄRTEIL()	Berechnet den Imaginärteil y einer komplexen Zahl
IMAPOTENZ()	Potenziert eine komplexe Zahl
IMARGUMENT()	Berechnet das Argument (?) einer komplexen Zahl
IMCOS()	Berechnet den Kosinus einer komplexen Zahl
IMCOSEC	Gibt den Kosekans einer komplexen Zahl zurück
IMCOSECHYP	Gibt den hyperbolischen Kosekans einer komplexen Zahl zurück
IMCOSHYP	Gibt den hyperbolischen Kosinus einer komplexen Zahl zurück
IMCOT	Gibt den Kotangens einer komplexen Zahl zurück
IMDIV()	Berechnet den Quotienten zweier komplexer Zahlen
IMEXP()	Exponentialfunktion für komplexe Zahlen
IMKONJUGIERTE()	Berechnet die konjugiert komplexe Zahl zu einer komplexen Zahl

Tabelle 16.1

Die Konstruktionsfunktionen in der Übersicht

Alle Funktionen der Kategorie »Konstruktion«

IMLN()	Berechnet den natürlichen Logarithmus einer komplexen Zahl
IMLOG10()	Berechnet den dekadischen Logarithmus einer komplexen Zahl
IMLOG2()	Berechnet den binären Logarithmus einer komplexen Zahl
IMPRODUKT()	Berechnet das Produkt komplexer Zahlen
IMREALTEIL()	Liefert den Realteil einer komplexen Zahl
IMSEC	Gibt den Sekans einer komplexen Zahl zurück
IMSECHYP	Gibt den hyperbolischen Sekans einer komplexen Zahl zurück
IMSIN()	Berechnet den Sinus einer komplexen Zahl
IMSINHYP	Gibt den hyperbolischen Sinus einer komplexen Zahl zurück
IMTAN	Gibt den Tangens einer komplexen Zahl zurück
IMSUB()	Berechnet die Differenz zweier komplexer Zahlen
IMSUMME()	Berechnet die Summe komplexer Zahlen
IMWURZEL()	Berechnet die Quadratwurzel einer komplexen Zahl
KOMPLEXE()	Berechnet aus zwei Zahlen eine komplexe Zahl

Funktionen der höheren Mathematik

Funktion	Beschreibung
BESSELI()	Berechnet einen Wert der modifizierten Besselfunktion der ersten Art $I_n(x)$
BESSELJ()	Berechnet einen Wert der Besselfunktion der ersten Art $J_n(x)$
BESSELK()	Berechnet einen Wert der modifizierten Besselfunktion der zweiten Art $K_n(x)$
BESSELY()	Berechnet einen Wert der Besselfunktion der zweiten Art $Y_n(x)$
GAUSSFEHLER()	Berechnet die Werte des Gauß'schen Fehlerintegrals
GAUSSF.GENAU()	Berechnet die Werte des Gauß'schen Fehlerintegrals
GAUSSFKOMPL()	Berechnet die Werte des Komplements zur Gauß'schen Fehlerfunktion
GAUSSFKOMPL.GENAU()	Berechnet die Werte des Komplements zur Gauß'schen Fehlerfunktion

Sprungfunktionen

Funktion	Beschreibung
DELTA()	Überprüft zwei Werte auf Gleichheit
GGANZZAHL()	Überprüft, ob eine Zahl größer als ein gegebener Schwellenwert ist

Umwandlungsfunktionen

Zahlensysteme In dieser Gruppe steht eine Serie von Umwandlungsfunktionen, die Zahlen von einem Zahlensystem in ein anderes konvertieren.

Das gebräuchlichste und uns am besten vertraute Zahlensystem ist das Dezimalsystem, das mit zehn Ziffern arbeitet und sich von unseren zehn Fingern herleitet. In der englischen Sprache hat sich dieser Zusammenhang auch sprachlich erhalten: »digit« kann sowohl mit »Finger« als auch mit »Ziffer« übersetzt werden.

Mit der Entwicklung der Datenverarbeitung kam ein Bedarf an einem Zahlensystem auf, das den technischen Gegebenheiten der frühen Rechenmaschinen entgegen kam. So wird gern als Grund für die Einführung des Dualsystems angeführt, dass die Rechenmaschinen als Binärrechner nur mit zwei Zuständen (*ein* und *aus*) umgehen können. (Konrad Zuse baute übriges im Jahre 1941 den ersten programmierbaren Binärrechner, den Z3.)

Das eben Gesagte über die Einführung des Dualsystems in die Rechentechnik ist nur die halbe Wahrheit: Ebenso wichtig oder noch wichtiger ist, dass das Dualsystem viel einfachere Rechenregeln hat.

Hinweis

Mit wie vielen Mühen müssen unsere »Abc-Schützen« im Rechenunterricht die zahlreichen Regeln unseres lieb gewonnenen 10er-Systems erlernen! $3+4$, $3+5$, $4+7$ usw. – dazu noch das kleine und große Einmaleins! Das Alles in die damals noch mechanischen Rechenmaschinen einzubauen (zu »implementieren«) war ein Ding der Unmöglichkeit.

Das Dualsystem kommt im Grunde mit drei Rechenregeln aus, auf denen alle weiteren aufbauen:

$$\begin{aligned} 0 + 0 &= 0 \\ 0 + 1 &= 1 \text{ bzw. } 1 + 0 = 1 \\ 1 + 1 &= 10 \end{aligned}$$

So kommen zum vertrauten Dezimalsystem noch drei weitere in der Rechentechnik gebräuchliche, die unten näher beschrieben werden. Weil Excel diese vier Zahlensysteme beherrscht, benötigt man zwölf Umwandlungsfunktionen: vier Zahlensysteme können in jeweils die drei anderen konvertiert werden.

Die Zahlensysteme sind im Einzelnen:

- ▶ Dezimal (10)
- ▶ Dual (2)
- ▶ Oktal (8)
- ▶ Hexadezimal (16)

In den Klammern steht die Basis des jeweiligen Zahlensystems und damit die Anzahl der Ziffern, über die dieses Zahlensystem verfügt. Weil wir bei dem Hexadezimalsystem mehr Ziffern benötigen als uns zur Verfügung stehen, werden für die Darstellung der fehlenden Ziffern die Buchstaben A bis F genommen. (Das Hexadezimalsystem sollte nicht mit dem »Hexagesimalsystem« (eigentlich das »Sexagesimalsystem«) mit der Basis 60 verwechselt werden, das bei Winkeln, geografischen Längen und Breiten und bei der Zeitmessung angewendet wird und uns Zahlbegriffe aus alten Zeiten wie »Dutzend« oder »Schock« eingebracht hat.)

System	Basis	Ziffern
Dualsystem	2	0 und 1
Oktalsystem	8	0 ... 7
Dezimalsystem	10	0 ... 9
Hexadezimalsystem	16	0 ... 9, A ... F

Tabelle 16.2
Übersicht der
Zahlensysteme

Einige Funktionsnamen enthalten die Buchstabenfolge »BIN« wie beispielsweise die Funktion BININDEZ(). Der Name von BININDEZ() leitet sich vom Englischen her und lautet dort »bin2dec« und in voller Länge »BINary TO DECimal«. Das Wort »binary« bedeutet

»binär« und bildet in der IT-Branche mit »digit« eine feste Wendung (binary digit = Dualziffer), deren Abkürzung »bit« aus der heutigen Zeit nicht mehr wegzudenken ist.

Hinweis Im Gegensatz dazu wird der Begriff »binär« im Deutschen eher für physikalische Systeme mit zwei stabilen Zuständen wie z.B. einem elektrischen Schalter benutzt, sodass man hierzulande beim Zahlensystem mit 2 als Basis lieber vom dualen Zahlensystem oder kurz Dualsystem spricht (nach Gottfried Wilhelm Leibniz, der in seinem berühmten Neujahrsbrief vom 2. Januar 1679 Herzog Rudolf August von Wolfenbüttel erklärte, dass sich alle Zahlen mit nur zwei Ziffern ausdrücken lassen: der Eins und der Null¹). Daher hieße diese Funktion besser »DUALINDEZ()«. Das duale Zahlensystem ist ein Sonderfall aller binären Systeme.

Dualsystem Weil beim Dualsystem nur zwei Werte pro Zahlenstelle möglich sind, erreichen Dualzahlen schnell beträchtliche Längen, sind unübersichtlich und somit schwierig zu lesen. Daher fasst man

Oktalsystem ► entweder drei duale Ziffern zu einem Zahlensystem auf der Basis 8 (Oktalsystem)

Hexadezimalsystem ► oder vier duale Ziffern zu einem Zahlensystem auf der Basis 16 (Hexadezimalsystem)
zusammen.

Beispielsweise ist

2005 (dezimal) = 11111010101 (dual) = 3725 (oktal) = 7D5 (hexadezimal).

Es ist deutlich zu sehen, dass das Zahlensystem mit der größten Basis (hier: hexadezimal) mit der geringsten Anzahl an Ziffern auskommt.

Allgemein ist der Zusammenhang zwischen Wertebereich M , der mit einer Zahl bestimmter Länge n in einem Zahlensystem der Basis b ausgedrückt werden kann, und der Anzahl der Ziffern n :

$$M = B^n$$

Beispielsweise können wir mit acht Ziffern ($n = 8$) im Dualsystem ($B = 2$) maximal $M = 2^8 = 256$ Zahlen ausdrücken. Der Wertebereich erstreckt sich von 0 bis 255. Weil man auf diese Weise nur positive Zahlen darstellen kann, nennt man diese Zahlen »vorzeichenlos« oder auf Englisch »unsigned integer«².

Im Alltagsleben kennzeichnen wir negative Zahlen mit einem vorangestellten Minuszeichen. Im Computer mit seinem Binärsystem (= zwei physikalische Zustände) ist kein Platz für ein drittes Symbol, das Vorzeichen. Daher reserviert man zur Kennzeichnung negativer Zahlen die am weitesten links stehende Dualziffer, das höchstwertige Bit (MSB = Most Significant Bit). Negative Zahlen haben dort eine Eins, positive eine Null.

Einerkomplement Damit der Computer mit diesen Zahlen rechnen kann, muss die »Laufriichtung« der negativen Zahlen umgekehrt werden: Auch bei den Dualzahlen gilt, dass negative Zahlen umso kleiner werden, je größer ihr Absolutbetrag ist. Dies erreicht man, indem die einzelnen Ziffern (mit Ausnahme des Vorzeichenbits) invertiert werden. Jede Eins wird in eine Null verwandelt und umgekehrt. Somit hat man das »Einerkomplement« erhalten, das allerdings noch ein paar Nachteile aufweist, auf die hier nicht näher eingegangen werden soll – abgesehen davon, dass das Einerkomplement zwei Nullwerte aufweist: +0 und -0, was mathematisch, aber nicht technisch dasselbe ist.

¹ Renate Loosen, Franz Vonessen, G. W. Leibniz: Zwei Briefe über das binäre Zahlensystem und die chinesische Philosophie, Stuttgart, 1968, S. 19-23

² Das entspricht dem Datentyp *Byte* in VBA.

Zur Perfektionierung dieser Zahlendarstellung addiert man eine Eins hinzu, vernachlässigt einen eventuellen Überlauf und erhält die »Zweierkomplement-Darstellung«. Ihr Wertebereich erstreckt sich von -128 bis $+127$. Auf diese Zahlen kann der Computer die Rechenregeln für vorzeichenlose Zahlen unterschiedslos anwenden. Die Zahlen werden nur anders interpretiert. Sinngemäß gilt das Gesagte auch für größere Zahlen, lediglich die Anzahl der Ziffern steigt. An diese Zahlendarstellung³ halten sich auch die Dualzahlen in Excel, die maximal zehn Bits haben dürfen und deswegen den Wertebereich von -512 bis $+511$ (wegen $2^{10} = 1.024$) abdecken.

Zweierkomplement

BININDEZ() BIN2DEC()

BININDEZ(Zahl)

Die Funktion wandelt eine Dualzahl in eine Dezimalzahl um.

Zahl (erforderlich) ist die maximal 10stellige Dualzahl in Zweierkomplement-Darstellung (s.o.), die in eine Dezimalzahl umgewandelt werden soll.

Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

=BININDEZ(1110) ergibt 14

=BININDEZ(11111111) ergibt 511

=BININDEZ(111111111) ergibt -1

=BININDEZ(111111110) ergibt -2

Zahlreiche weitere Beispiele können Sie in der Tabelle in Abbildung 16.1 sehen, wobei zum Vergleich auch die Umrechnung in das Oktal- und Hexadezimalsystem aufgeführt ist.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

	A	B	C	D	E
1	Konvertierung von Dualzahlen in ein anderes Zahlensystem				
2					
3	Zahlensystem	dual	dezimal	oktal	hexadezimal
4	<i>Basis</i>	2	10	8	16
5		00000001	1	001	01
6		00001001	9	011	09
7		00001010	10	012	0A
8		00010011	19	023	13
9		00010100	20	024	14
10		01100011	99	143	63
11		01100100	100	144	64
12		11000111	199	307	C7
13		11001000	200	310	C8
14		11111111	255	377	FF

Abbildung 16.1: Mit dieser Excel-Tabelle können Sie Dualzahlen in die anderen Zahlensysteme umrechnen

BININHEX(), BININOKT(), DEZINBIN(), DEZINHEX(), DEZINOKT(), HEXINBIN(), HEXINDEZ(), HEXINOKT(), OKTINBIN(), OKTINDEZ(), OKTINHEX()

Siehe auch

Das Beispiel finden Sie im Ordner `Ms5-235\Kap16` in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *dual*.



³ So werden in VBA die Datentypen *Integer* und *Long* dargestellt.

BININHEX() BIN2HEX()

Syntax BININHEX(*Zahl*; *Stellen*)

Definition Die Funktion wandelt eine Dualzahl in eine Hexadezimalzahl um.

Argumente *Zahl* (erforderlich) ist die maximal 10stellige Dualzahl in Zweierkomplement-Darstellung (siehe Seite 809), die in eine Hexadezimalzahl umgewandelt werden soll. Bei negativen Werten von *Zahl* werden immer 10stellige Hexadezimalzahlen zurückgegeben.

Stellen (optional) gibt an, wie viele Ziffern der Hexadezimalzahl angezeigt werden sollen, und dient zur Anzeige von führenden Nullen im Ergebnis. Wenn das Argument *Stellen* fehlt, wird nur die erforderliche Anzahl an Ziffern angezeigt. Eventuell vorhandene Nachkommastellen werden beim Argument *Stellen* vernachlässigt.

Hintergrund Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

Achtung Das Ergebnis von BININHEX() ist eine Zeichenfolge, die Sie nicht ohne Weiteres für weitere Rechenoperationen verwenden dürfen. Falls erforderlich, können Sie Textfunktionen auf die so erhaltenen Hexadezimalzahlen anwenden.

Ein besonderes Augenmerk verdient die Tatsache, dass Formeln, die Bezüge auf die Hexadezimalzahlen mit Rechenoperatoren verknüpfen, keine Fehlermeldung produzieren, solange diese Zahlen nur Ziffern enthalten, die auch im Dezimalsystem vorkommen (also 0...9); sie werden dann als Dezimalzahlen interpretiert und die Formel liefert dann ein nicht beabsichtigtes Ergebnis.

Einen besonderen Fall zeigt das folgende Beispiel:

=BININHEX(111100100) ergibt *1E4* als hexadezimale Zahl. Wenn dieser Wert über einen Zellbezug in einer Formel verwendet wird, kann er als »1E+4«, also als eine gänzlich andere Zahl im wissenschaftlichen Format interpretiert werden. Weil keine Fehlermeldung in der Tabelle angezeigt wird, ist dieser Fehler schwer zu finden.

Praxiseinsatz =BININHEX(1110) ergibt die hexadezimale Zahl *E*

=BININHEX(1110;4) ergibt die hexadezimale Zahl *000E*

=BININHEX(11111111) ergibt *1FF*

=BININHEX(111111111) ergibt *FFFFFFF* (entspricht -1_{dezimal})

=BININHEX(111111111;4) ergibt ebenfalls *FFFFFFF* (entspricht -1_{dezimal})

Für weitere Beispiele sehen Sie bitte im Abschnitt »Praxiseinsatz« von BININDEZ() auf Seite 809 nach.

Siehe auch BININDEZ(), BININOKT(), DEZINBIN(), DEZINHEX(), DEZINOKT(), HEXINBIN(), HEXINDEZ(), HEXINOKT(), OKTINBIN(), OKTINDEZ(), OKTINHEX()



Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *dual*.

BININOKT() BIN2OCT()

BININOKT(*Zahl*; *Stellen*)

Die Funktion wandelt eine Dualzahl in eine Oktalzahl um.

Zahl (erforderlich) ist die maximal 10stellige Dualzahl in Zweierkomplement-Darstellung (siehe Seite 809), die in eine Oktalzahl umgewandelt werden soll. Bei negativen Werten von *Zahl* werden immer 10stellige Oktalzahlen zurückgegeben.

Stellen (optional) gibt an, wie viele Ziffern der Oktalzahl angezeigt werden sollen, und dient zur Anzeige von führenden Nullen im Ergebnis. Wenn das Argument *Stellen* fehlt, wird nur die erforderliche Anzahl an Ziffern angezeigt. Eventuell vorhandene Nachkommastellen werden beim Argument *Stellen* vernachlässigt.

Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

Die mit dieser Funktion erhaltenen Oktalzahlen sind Zeichenfolgen, die nicht ohne Weiteres wie gewöhnliche Zahlen in anderen Formeln weiterverwendet werden dürfen. Wird eine Oktalzahl in einer Formel mit Rechenoperatoren verwendet, wird sie als Dezimalzahl interpretiert. Wenn beispielsweise in den Zellen C13 und C14 die Oktalzahlen 320 bzw. 377 stehen, ergibt die Formel

=C13+C14

den Wert 697, was aber falsch ist. Offensichtlich wird das schon dadurch, dass die Zahl 697 keine gültige Oktalzahl ist, weil der Ziffervorrat des Oktalsystems nur von 0 bis 7 reicht und die Ziffer 9 darin nicht vorkommt. Richtig wäre die Oktalzahl 717.

Korrekt arbeitet die Formel

=DEZINOKT(OKTINDEZ(C13)+OKTINDEZ(C14))

Vor dem Addieren müssen die Oktalzahlen in Dezimalzahlen umgewandelt werden; zum Schluss findet die Umwandlung in die gewünschte Zahlendarstellung, dem Oktalsystem, statt.

=BININOKT(1110) ergibt 16

=BININOKT(1110;4) ergibt 0016

=BININOKT(111111111) ergibt 777

=BININOKT(1111111111) ergibt 7777777777 (-1_{dezimal})

Für weitere Beispiele sehen Sie bitte im Abschnitt »Praxiseinsatz« von BININDEZ() auf Seite 809 nach.

BININDEZ(), BININHEX(), DEZINBIN(), DEZINHEX(), DEZINOKT(), HEXINBIN(), HEXINDEZ(), HEXINOKT(), OKTINBIN(), OKTINDEZ(), OKTINHEX()

Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *dual*.

Syntax

Definition

Argumente

Hintergrund

Achtung

Praxiseinsatz

Siehe auch



DEZINBIN() DEC2BIN()

Syntax DEZINBIN(*Zahl*; *Stellen*)

Definition Die Funktion wandelt eine dezimale Zahl in eine Dualzahl (binäre Zahl) um.

Argumente *Zahl* (erforderlich) ist die Dezimalzahl, die in eine Dualzahl in Zweierkomplement-Darstellung (siehe Seite 809) umgewandelt werden soll. Die gültigen Eingabewerte für *Zahl* gehen von -512 bis +511. Wenn das Argument *Zahl* keine ganze Zahl ist, werden ihre Nachkommastellen ignoriert. Bei negativen Werten von *Zahl* werden immer 10stellige Dualzahlen zurückgegeben.

Stellen (optional) gibt an, wie viele Ziffern der Dualzahl angezeigt werden sollen und dient zur Anzeige von führenden Nullen im Ergebnis. Wenn das Argument *Stellen* fehlt, wird nur die erforderliche Anzahl an Ziffern angezeigt. Eventuell vorhandene Nachkommastellen werden beim Argument *Stellen* vernachlässigt.

Hintergrund Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

Achtung Excel-Funktionen betrachten die erhaltenen Dualzahlen als Text. Mit ihnen können daher ohne besondere Maßnahmen keine Rechenoperationen in anderen Formeln durchgeführt werden. Zur Weiterverwendung bieten sich nur Textfunktionen an.

Wenn Formeln mit Rechenoperatoren Bezüge auf diese Dualzahlen verarbeiten, gibt Excel keine Fehlermeldung aus, sondern interpretiert sie als Dezimalzahlen, was selbstverständlich zu falschen Ergebnissen führt.

Praxiseinsatz =DEZINBIN(14) ergibt 1110
 =DEZINBIN(341) ergibt 101010101
 =DEZINBIN(6;8) ergibt 00000110
 =DEZINBIN(-1) ergibt 111111111
 =DEZINBIN(-1;8) ergibt ebenfalls 111111111

Zahlreiche weitere Beispiele können Sie in der Tabelle in Abbildung 16.2 sehen, wobei zum Vergleich auch die Umrechnung in das Oktal- und Hexadezimalsystem aufgeführt ist.

	A	B	C	D	E
1	Konvertierung von Dezimalzahlen in ein anderes Zahlensystem				
2					
3	Zahlensystem	dezimal	dual	oktal	hexadezimal
4	<i>Basis</i>	10	2	8	16
5		1	00000001	001	01
6		9	00001001	011	09
7		10	00001010	012	0A
8		19	00010011	023	13
9		20	00010100	024	14
10		99	01100011	143	63
11		100	01100100	144	64
12		199	11000111	307	C7
13		200	11001000	310	C8
14		255	11111111	377	FF

Abbildung 16.2: Mit dieser Excel-Tabelle können Sie Dezimalzahlen in die anderen Zahlensysteme umrechnen

BININDEZ(), BININHEX(), BININOKT(), DEZINHEX(), DEZINOKT(), HEXINBIN(), HEXINDEZ(), HEXINOKT(), OKTINBIN(), OKTINDEZ(), OKTINHEX()

Siehe auch

Das Beispiel finden Sie im Ordner `Ms5-235\Kap16` in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *dezimal*.



DEZINHEX() DEC2HEX()

DEZINHEX(*Zahl*; *Stellen*)

Die Funktion wandelt eine dezimale Zahl in eine hexadezimale Zahl um.

Zahl (erforderlich) ist die Dezimalzahl, die in eine Hexadezimalzahl in Zweierkomplement-Darstellung (siehe Seite 809) umgewandelt werden soll. Die gültigen Eingabewerte für *Zahl* reichen von $-549.755.813.888$ bis $+549.755.813.887$. Wenn das Argument *Zahl* keine ganze Zahl ist, werden ihre Nachkommastellen vernachlässigt. Bei negativen Werten von *Zahl* werden immer 10stellige Hexadezimalzahlen zurückgegeben.

Stellen (optional) gibt an, wie viele Ziffern der Hexadezimalzahl angezeigt werden sollen, und dient zur Anzeige von führenden Nullen im Ergebnis. Wenn das Argument *Stellen* fehlt, wird nur die erforderliche Anzahl an Ziffern angezeigt. Eventuell vorhandene Nachkommastellen werden beim Argument *Stellen* vernachlässigt.

Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

Das Ergebnis von DEZINHEX() ist eine Zeichenfolge, die Sie nicht ohne besondere Maßnahmen für weitere Rechenoperationen nehmen dürfen. Falls erforderlich können Sie Textfunktionen auf die so erhaltenen Hexadezimalzahlen anwenden.

Ein besonderes Augenmerk verdient die Tatsache, dass Formeln, die Bezüge auf die Hexadezimalzahlen mit Rechenoperatoren verknüpfen, keine Fehlermeldung produzieren, solange diese Zahlen nur Ziffern enthalten, die auch im Dezimalsystem vorkommen (also 0...9). Sie werden dann als Dezimalzahlen interpretiert und die Formel liefert dann ein nicht beabsichtigtes Ergebnis.

Einen besonderen Fall illustriert das folgende Beispiel:

`=DEZINHEX(123456)` ergibt *1E240* als hexadezimale Zahl. Wenn dieser Wert über einen Zellbezug in einer Formel verwendet wird, kann er als »1E+240«, also als eine gänzlich andere Zahl im wissenschaftlichen Format interpretiert werden.

`=DEZINHEX(14)` ergibt *E*

`=DEZINHEX(31;4)` ergibt *001F*

`=DEZINHEX(341;4)` ergibt *0155*

`=DEZINHEX(-5)` ergibt *FFFFFFFB*

`=DEZINHEX(-5;4)` ergibt ebenfalls *FFFFFFFB*

Für weitere Beispiele sehen Sie bitte im Abschnitt »Praxiseinsatz« von DEZINBIN() auf Seite 812 nach.

Syntax

Definition

Argumente

Hintergrund

Achtung

Praxiseinsatz

Siehe auch BININDEZ(), BININHEX(), BININOKT(), DEZINBIN(), DEZINOKT(), HEXINBIN(), HEXINDEZ(), HEXINOKT(), OKTINBIN(), OKTINDEZ(), OKTINHEX()



Das Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *dezimal*.

DEZINOKT() DEC2OCT()

Syntax DEZINOKT(*Zahl*; *Stellen*)

Definition Die Funktion wandelt eine Dezimalzahl in eine Oktalzahl um.

Argumente *Zahl* (erforderlich) ist die Dezimalzahl, die in eine Oktalzahl in Zweierkomplement-Darstellung (siehe Seite 809) umgewandelt werden soll. Die gültigen Eingabewerte für *Zahl* reichen von $-536.870.912$ bis $+536.870.911$. Wenn das Argument *Zahl* keine ganze Zahl ist, werden ihre Nachkommastellen vernachlässigt. Bei negativen Werten von *Zahl* werden immer 10stellige Oktalzahlen zurückgegeben.

Stellen (optional) gibt an, wie viele Ziffern der Oktalzahl angezeigt werden sollen, und dient zur Anzeige von führenden Nullen im Ergebnis. Wenn das Argument *Stellen* fehlt, wird nur die erforderliche Anzahl an Ziffern angezeigt. Eventuell vorhandene Nachkommastellen werden beim Argument *Stellen* vernachlässigt.

Hintergrund Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

Achtung Die mit dieser Funktion erhaltenen Oktalzahlen sind Zeichenfolgen, die nicht ohne Weiteres wie gewöhnliche Zahlen in anderen Formeln weiterverwendet werden dürfen. Wird eine Oktalzahl in einer Formel mit Rechenoperatoren verwendet, wird sie als Dezimalzahl interpretiert. Wenn beispielsweise in den Zellen *C13* und *C14* die Oktalzahlen 320 bzw. 377 stehen, ergibt die Formel

`=C13+C14`

den Wert 697, was aber falsch ist. Offensichtlich wird das schon dadurch, dass die Zahl 697 keine gültige Oktalzahl ist, weil der Ziffernvorrat des Oktalsystems nur von 0 bis 7 reicht und die Ziffer 9 darin nicht vorkommt. Richtig wäre die Oktalzahl 717.

Korrekt arbeitet die Formel

`=DEZINOKT(OKTINDEZ(C13)+OKTINDEZ(C14))`

Vor dem Addieren müssen die Oktalzahlen in Dezimalzahlen umgewandelt werden; zum Schluss findet die Umwandlung in die gewünschte Zahlendarstellung, dem Oktalsystem, statt.

Praxiseinsatz `=DEZINOKT(14)` ergibt 16

`=DEZINOKT(341)` ergibt 525

`=DEZINOKT(341;4)` ergibt 0525

=DEZINOKT(-5) ergibt 777777773

=DEZINOKT(-5;4) ergibt ebenfalls 777777773

Für weitere Beispiele sehen Sie bitte im Abschnitt »Praxiseinsatz« von DEZINBIN() auf Seite 812 nach.

BININDEZ(), BININHEX(), BININOKT(), DEZINBIN(), DEZINOKT(), HEXINBIN(), HEXINDEZ(), HEXINOKT(), OKTINBIN(), OKTINDEZ(), OKTINHEX()

Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *dezimal*.

Siehe auch



HEXINBIN() HEX2BIN()

HEXINBIN(*Zahl*; *Stellen*)

Die Funktion wandelt eine Hexadezimalzahl in eine Dualzahl um.

Zahl (erforderlich) ist die maximal 10stellige Hexadezimalzahl, die in eine Dualzahl in Zweierkomplement-Darstellung (siehe Seite 809) umgewandelt werden soll. Die gültigen Eingabewerte für *Zahl* gehen von FFFFFFFE00 (= -512_{dezimal}) bis 1FF (= $+511_{\text{dezimal}}$). Bei negativen Werten von *Zahl* werden immer 10stellige Dualzahlen zurückgegeben.

Stellen (optional) gibt an, wie viele Ziffern der Dualzahl angezeigt werden sollen, und dient zur Anzeige von führenden Nullen im Ergebnis. Wenn das Argument *Stellen* fehlt, wird nur die erforderliche Anzahl an Ziffern angezeigt. Eventuell vorhandene Nachkommastellen werden beim Argument *Stellen* vernachlässigt.

Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

Excel-Funktionen geben die erhaltenen Dualzahlen als Text im Standard-Format/linksbündig aus. Mit ihnen können daher ohne besondere Maßnahmen keine Rechenoperationen in anderen Formeln durchgeführt werden. Zur Weiterverwendung bieten sich nur Textfunktionen an, obwohl das Ergebnis den Anschein einer Zahl hat.

Wenn Formeln mit Rechenoperatoren Bezüge auf diese Dualzahlen verarbeiten, gibt Excel keine Fehlermeldung aus, sondern interpretiert sie als Dezimalzahlen, was selbstverständlich zu falschen Ergebnissen führt.

=HEXINBIN("E") ergibt 1110

=HEXINBIN("E";6) ergibt 001110

=HEXINBIN("155") ergibt 101010101

=HEXINBIN("FFFFFFFF") ergibt 1111111111 (entspricht -1_{dezimal})

=HEXINBIN("FFFFFFFF";6) ergibt ebenfalls 1111111111 (entspricht -1_{dezimal})

Zahlreiche weitere Beispiele können Sie in der Tabelle in Abbildung 16.3 sehen, wobei zum Vergleich auch die Umrechnung in das Dezimal- und Hexadezimalsystem aufgeführt ist.

Syntax

Definition

Argumente

Hintergrund

Achtung

Praxiseinsatz

	A	B	C	D	E
1	Konvertierung von Hexadezimalzahlen in ein anderes Zahlensystem				
2					
3	Zahlensystem	hexadezimal	dual	dezimal	oktal
4	<i>Basis</i>	16	2	10	8
5		01	00000001	1	001
6		09	00001001	9	011
7		0A	00001010	10	012
8		13	00010011	19	023
9		14	00010100	20	024
10		63	01100011	99	143
11		64	01100100	100	144
12		C7	11000111	199	307
13		C8	11001000	200	310
14		FF	11111111	255	377

Abbildung 16.3: Mit dieser Excel-Tabelle können Sie Hexadezimalzahlen in die anderen Zahlensysteme umrechnen

Siehe auch BININDEZ(), BININHEX(), BININOKT(), DEZINBIN(), DEZINHEX(), DEZINOKT(), HEXINDEZ(), HEXINOKT(), OKTINBIN(), OKTINDEZ(), OKTINHEX()



Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *hexadez*.

HEXINDEZ() HEX2DEC()

Syntax HEXINDEZ(*Zahl*)

Definition Die Funktion wandelt eine hexadezimale Zahl in eine dezimale Zahl um.

Argumente *Zahl* (erforderlich) ist die maximal 10stellige Hexadezimalzahl in Zweierkomplement-Schreibweise (siehe Seite 809), die in eine Dezimalzahl umgewandelt werden soll. Die gültigen Eingabewerte für *Zahl* gehen von 800000000 bis 7FFFFFFF (d.h. von $-549.755.813.888_{\text{dezimal}}$ bis $+549.755.813.887_{\text{dezimal}}$).

Ein (optionales) Argument *Stellen* ist hier nicht notwendig, weil das Ziel, führende Nullen anzuzeigen, leicht mit der Zellenformatierung erreicht werden kann.

Hintergrund Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

Praxiseinsatz =HEXINDEZ("E") ergibt 14
 =HEXINDEZ("155") ergibt 341
 =HEXINDEZ("FFFFFFFF") ergibt -1

Für weitere Beispiele sehen Sie bitte im Abschnitt »Praxiseinsatz« von HEXINBIN() auf Seite 815 nach.

Siehe auch BININDEZ(), BININHEX(), BININOKT(), DEZINBIN(), DEZINHEX(), DEZINOKT(), HEXINBIN(), HEXINOKT(), OKTINBIN(), OKTINDEZ(), OKTINHEX()



Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *hexadez*.

HEXINOKT() HEX2OCT()

HEXINOKT(*Zahl*; *Stellen*)

Die Funktion wandelt eine Hexadezimalzahl in eine Oktalzahl um.

Zahl (erforderlich) ist die maximal 10stellige Hexadezimalzahl in Zweierkomplement-Schreibweise (siehe Seite 809), die in eine Oktalzahl umgewandelt werden soll. Die gültigen Eingabewerte für *Zahl* gehen von FFE000000 bis 1FFFFFFF.

Stellen (optional) gibt an, wie viele Ziffern angezeigt werden sollen, und dient zur Anzeige von führenden Nullen im Ergebnis. Wenn das Argument *Stellen* fehlt, wird nur die erforderliche Anzahl an Ziffern angezeigt. Eventuell vorhandene Nachkommastellen werden beim Argument *Stellen* vernachlässigt.

Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

Die mit dieser Funktion erhaltenen Oktalzahlen sind Zeichenfolgen, die nicht wie gewöhnliche Zahlen in anderen Formeln weiterverwendet werden dürfen. Wird eine Oktalzahl in einer Formel mit Rechenoperatoren verwendet, wird sie als Dezimalzahl interpretiert. Wenn beispielsweise in den Zellen C13 und C14 die Oktalzahlen 320 bzw. 377 stehen, ergibt die Formel

```
=C13+C14
```

den Wert 697, was aber falsch ist. Offensichtlich wird das schon dadurch, dass die Zahl 697 keine gültige Oktalzahl ist, weil der Ziffernvorrat des Oktalsystems nur von 0 bis 7 reicht und die Ziffer 9 darin nicht vorkommt. Korrekt ist die Oktalzahl 717.

Einwandfrei arbeitet die Formel

```
=DEZINOKT(OKTINDEZ(C13)+OKTINDEZ(C14))
```

```
=HEXINOKT("E") ergibt 16
```

```
=HEXINOKT("E";4) ergibt 0016
```

```
=HEXINOKT("155") ergibt 525
```

```
=HEXINOKT("155";4) ergibt 0525
```

```
=HEXINOKT("FFFFFFFF") ergibt 777777777 (entspricht -1dezimal)
```

Für weitere Beispiele sehen Sie bitte im Abschnitt »Praxiseinsatz« von HEXINBIN() auf Seite 815 nach.

BININDEZ(), BININHEX(), BININOKT(), DEZINBIN(), DEZINHEX(), DEZINOKT(), HEXINBIN(), HEXINDEZ(), OKTINBIN(), OKTINDEZ(), OKTINHEX()

Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *hexadez*.

Syntax

Definition

Argumente

Hintergrund

Achtung

Praxiseinsatz

Siehe auch



OKTINBIN() OCT2BIN()

Syntax OKTINBIN(*Zahl*; *Stellen*)

Definition Die Funktion wandelt eine Oktalzahl in eine Dualzahl (binäre Zahl) um.

Argumente *Zahl* (erforderlich) ist die maximal 10stellige Oktalzahl in Zweierkomplement-Darstellung (siehe Seite 809), die in eine Dualzahl umgewandelt werden soll. Bei negativen Werten von *Zahl* werden immer 10stellige Dualzahlen zurückgegeben.

Stellen (optional) gibt an, wie viele Ziffern der Dualzahl angezeigt werden sollen, und dient zur Anzeige von führenden Nullen im Ergebnis. Wenn das Argument *Stellen* fehlt, wird nur die erforderliche Anzahl an Ziffern angezeigt. Eventuell vorhandene Nachkommastellen werden beim Argument *Stellen* vernachlässigt.

Hintergrund Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

Achtung Excel-Funktionen geben die erhaltenen Dualzahlen als Text im Standardformat linksbündig aus. Mit ihnen können daher ohne besondere Maßnahmen keine Rechenoperationen in anderen Formeln durchgeführt werden. Zur Weiterverwendung bieten sich nur Textfunktionen an, obwohl das Ergebnis den Anschein einer Zahl hat.

Wenn Formeln mit Rechenoperatoren Bezüge auf diese Dualzahlen verarbeiten, gibt Excel keine Fehlermeldung aus, sondern interpretiert sie als Dezimalzahlen, was selbstverständlich zu falschen Ergebnissen führt.

Praxiseinsatz =OKTINBIN(16) ergibt 1110
 =OKTINBIN(16;6) ergibt 001110
 =OKTINBIN(525) ergibt 101010101
 =OKTINBIN(77777777) ergibt 111111111 (entspricht -1_{dezimal})

Zahlreiche weitere Beispiele können Sie in der Tabelle in Abbildung 16.4 sehen, wobei zum Vergleich auch die Umrechnung in das Dezimal- und Hexadezimalsystem aufgeführt ist.

	A	B	C	D	E
1	Konvertierung von Oktalzahlen in ein anderes Zahlensystem				
2					
3	Zahlensystem	oktal	dual	dezimal	hexadezimal
4	<i>Basis</i>	8	2	10	16
5		001	0000001	1	01
6		011	00001001	9	09
7		012	00001010	10	0A
8		023	00010011	19	13
9		024	00010100	20	14
10		143	01100011	99	63
11		144	01100100	100	64
12		307	11000111	199	C7
13		310	11001000	200	C8
14		377	11111111	255	FF

Abbildung 16.4: Mit dieser Excel-Tabelle können Sie Oktalzahlen in die anderen Zahlensysteme umrechnen

BININDEZ(), BININHEX(), BININOKT(), DEZINBIN(), DEZINHEX(), DEZINOKT(), HEXINBIN(), HEXINDEZ(), HEXINOKT(), OKTINDEZ(), OKTINHEX()

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *oktal*.

Siehe auch



OKTINDEZ() OCT2DEC()

OKTINDEZ(*Zahl*)

Diese Funktion wandelt eine Oktalzahl in eine Dezimalzahl um.

Zahl ist die oktale Zahl, die Sie in eine dezimale Zahl umwandeln wollen. Das Argument *Zahl* darf aus höchstens 10 Zeichen (30 Bits) bestehen. Das signifikante Bit von *Zahl* ist das Vorzeichenbit. (Das entspricht dem 30. Bit von rechts. Die anderen 29 Bits sind Betragsbits und stellen den Zahlenwert dar.) Negative Zahlen werden mit der Zweierkomplement-Schreibweise dargestellt.

Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

=OKTINDEZ(16) ergibt 14

=OKTINDEZ(525) ergibt 341

=OKTINDEZ(777777777) ergibt -1

Für weitere Beispiele sehen Sie bitte im Abschnitt »Praxiseinsatz« von OKTINBIN() auf Seite 818 nach.

BININDEZ(), BININHEX(), BININOKT(), DEZINBIN(), DEZINHEX(), DEZINOKT(), HEXINBIN(), HEXINDEZ(), HEXINOKT(), OKTINBIN(), OKTINHEX()

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *oktal*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



OKTINHEX() OCT2HEX()

OKTINHEX(*Zahl*; *Stellen*)

Diese Funktion wandelt eine Oktalzahl in eine Hexadezimalzahl um.

Zahl (erforderlich) ist die maximal 10stellige Oktalzahl in Zweierkomplement-Darstellung (siehe Seite 809), die in eine Hexadezimalzahl umgewandelt werden soll. Bei negativen Werten von *Zahl* werden immer 10stellige Hexadezimalzahlen zurückgegeben.

Stellen (optional) gibt an, wie viele Ziffern angezeigt werden sollen, und dient zur Anzeige von führenden Nullen im Ergebnis. Wenn das Argument *Stellen* fehlt, wird nur die erforderliche Anzahl an Ziffern angezeigt. Eventuell vorhandene Nachkommastellen werden beim Argument *Stellen* vernachlässigt.

Das Thema »Zahlensysteme« wurde ausführlich in der Einleitung zu diesen Funktionen erläutert (siehe Seite 805).

Syntax

Definition

Argumente

Hintergrund

Achtung Das Ergebnis von OKTINHEX() ist eine Zeichenfolge, die Sie nicht für weitere Rechenoperationen nehmen dürfen. Falls erforderlich, können Sie Textfunktionen auf die so erhaltenen Hexadezimalzahlen anwenden. Ein besonderes Augenmerk verdient die Tatsache, dass Formeln, die Bezüge auf die Hexadezimalzahlen mit Rechenoperatoren verknüpfen, keine Fehlermeldung produzieren, solange diese Zahlen nur Ziffern enthalten, die auch im Dezimalsystem vorkommen (also 0...9). Sie werden dann als Dezimalzahlen interpretiert und die Formel liefert anschließend ein nicht beabsichtigtes Ergebnis.

Einen Fall dieser Art veranschaulicht das folgende Beispiel:

=OKTINHEX(1750) ergibt 3E8 als hexadezimale Zahl. Wenn dieser Wert über einen Zellbezug in einer Formel verwendet wird, kann er als »3E+08«, also als eine gänzlich andere Zahl im wissenschaftlichen Format interpretiert werden.

Praxiseinsatz =OKTINHEX(16) ergibt E
 =OKTINHEX(16;2) ergibt 0E
 =OKTINHEX(525) ergibt 155
 =OKTINHEX(77777777) ergibt FFFFFFFF (entspricht -1_{dezimal})
 =OKTINHEX(77777777;4) ergibt ebenfalls FFFFFFFF (entspricht -1_{dezimal})

Für weitere Beispiele sehen Sie bitte im Abschnitt »Praxiseinsatz« von OKTINBIN() auf Seite 818 nach.

Siehe auch BININDEZ(), BININHEX(), BININOKT(), DEZINBIN(), DEZINHEX(), DEZINOKT(), HEXINBIN(), HEXINDEZ(), HEXINOKT(), OKTINBIN(), OKTINDEZ()



Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Konvert.xlsx* auf dem Arbeitsblatt *oktal*.

UMWANDELN() CONVERT()

Syntax UMWANDELN(*Zahl*; *Von_Maßeinheit*; *In_Maßeinheit*)

Definition Die Funktion führt Umrechnungen zwischen verschiedenen Maßsystemen durch.

Argumente *Zahl* (erforderlich) ist der Wert, der umgewandelt werden soll, in der Einheit *Von_Maßeinheit*.

Von_Maßeinheit (erforderlich) ist die Einheit der Ausgangszahl.

In_Maßeinheit (erforderlich) ist die Einheit, in die umgerechnet werden soll.

UMWANDELN() akzeptiert für *Von_Maßeinheit* und *In_Maßeinheit* die in den folgenden Tabellen aufgeführten Textwerte als Maßeinheiten, die in Anführungszeichen gesetzt werden müssen. Wenn diese Argumente Zellbezüge enthalten, müssen die Texte in den einbezogenen Zellen nicht in Anführungszeichen stehen, weil die in Excel implementierten Regeln diese Textwerte auch ohne Anführungszeichen als Texte kenntlich machen. Beachten Sie auf jeden Fall Groß- und Kleinschreibweise! Anderenfalls erhalten Sie die Fehlermeldung #NV.

Manche Maßeinheiten haben auch alternative Schreibweisen, die ebenfalls in den Tabellen angeführt sind.

Gewicht und Masse**Abkürzungen für die Maßeinheiten**

Gramm	g
»Stück«, Slug ^a	sg
Pfund (Handelsgewicht)	lbm
U (Atommasseeinheit)	u
Unze (Handelsgewicht)	ozm

^a »Slug« ist eine Masseneinheit im »British Gravitational System« (BG) und im »English Engineering System« (EE). Die deutsche Übersetzung »Stück« in der Hilfe zu Microsoft Excel führt etwas in die Irre.

Tabelle 16.3

(Handels-) Gewichte der Funktion UMWANDELN()

Entfernung**Abkürzungen für die Maßeinheiten**

Meter	m
Feste Meile, statute mile, engl. Meile	mi
Seemeile	Nmi
Zoll, inch	in
Fuß, foot	ft
Yard	yd
Ängström	ang
Pica (1/72 Zoll)	Pica

Tabelle 16.4

Längeneinheiten der Funktion UMWANDELN()

Zeit**Abkürzungen für die Maßeinheiten**

Jahr ^a	yr
Tag	day
Stunde	hr
Minute	mn
Sekunde	sec

^a Das Jahr wird mit 365,25 Tagen als Mittelwert über vier Jahre angenommen. Die wissenschaftlichen Jahre (astronomisch, siderisch, tropisch, »mittlere Sonne«, Kalenderjahr) werden nicht berücksichtigt.

Tabelle 16.5

Zeiteinheiten der Funktion UMWANDELN()

Druck**Abkürzungen für die Maßeinheiten**

Pascal	Pa, p
Atmosphäre (physikalisch)	atm, at
mm Quecksilber	mmHg

Tabelle 16.6

Druckeinheiten der Funktion UMWANDELN()

Kraft**Abkürzungen für die Maßeinheiten**

Newton	N
Dyn	dyn, dy
Pound-Kraft, pound force	lbf

Tabelle 16.7

Krafteinheiten der Funktion UMWANDELN()

Tabelle 16.8 Energie- einheiten der Funktion UMWANDELN()	Energie	Abkürzungen für die Maßeinheiten
	Joule	J
	Erg	e
	Thermodynamische Kalorie	c
	IT-Kalorie, int. Kalorie	cal
	Elektronenvolt	eV, ev
	Pferdestärke/Stunde	HPh, hh
	Wattstunde	Wh, wh
	Fuß-Pound, foot-poundals	flb
BTU	BTU, btu	
Tabelle 16.9 Leistungs- einheiten der Funktion UMWANDELN()	Leistung	Abkürzungen für die Maßeinheiten
	Pferdestärke, horsepower (UK)	HP, h
	Watt	W, w
Tabelle 16.10 Einheiten der magnetischen Feldstärke der Funktion UMWANDELN()	Magnetismus	Abkürzungen für die Maßeinheiten
	Tesla	T
	Gauß	ga
Tabelle 16.11 Temperatur- einheiten der Funktion UMWANDELN()	Temperatur	Abkürzungen für die Maßeinheiten
	Grad Celsius (°C)	C, cel
	Grad Fahrenheit (°F)	F, fah
	Kelvin (K)	K, kel
Tabelle 16.12 Flüssigkeits- maße der Funktion UMWANDELN()	Flüssigkeitsmaße	Abkürzungen für die Maßeinheiten
	Teelöffel, teaspoon (US)	tsp
	Esslöffel, tablespoon (US)	tbs
	Flüssigunze, Ounce (US)	oz
	Tasse, cups (US)	cup
	U.S. Pint (liq.)	pt, us_pt
	U.K. Pint	uk_pt
	Quart (US, liq.)	qt
	Gallone (US, liq.)	gal
	Liter	l, lt

Die folgenden abgekürzten Einheitenpräfixe können allen metrischen Angaben *Von_Maßeinheit* und *In_Maßeinheit* vorangestellt werden. Beachten Sie bitte, dass auch hier zwischen Groß- und Kleinschreibung unterschieden werden muss.

Präfix	Multiplikator	Abkürzung
exa	1E+18 (10^{18})	E
peta	1E+15 (10^{15})	P
tera	1E+12 (10^{12})	T
giga	1E+09 (10^9)	G
mega	1E+06 (10^6)	M
kilo	1E+03 (10^3)	k
hekto	1E+02 (10^2)	h
deka	1E+01 (10^1)	e
dezi	1E-01 (10^{-1})	d
zenti	1E-02 (10^{-2})	c
milli	1E-03 (10^{-3})	m
mikro	1E-06 (10^{-6})	u
nano	1E-09 (10^{-9})	n
pico	1E-12 (10^{-12})	p
femto	1E-15 (10^{-15})	f
atto	1E-18 (10^{-18})	a

Tabelle 16.13
Einheiten-
präfixe

Traditionell benutzt der angloamerikanische Sprachraum nicht-metrische Maße, also Maße, die nicht auf dem Gaußschen Maßsystem (CGS-System) beruhen. Heute wird im wissenschaftlichen, geschäftlichen und amtlichen Verkehr nur das Internationale Einheitensystem (SI-System) verwendet, das sich auch im angloamerikanischen Raum immer mehr durchsetzt. Die Umrechnung zwischen den teilweise exotisch anmutenden angloamerikanischen und den internationalen Einheiten fällt nicht immer leicht. Hier kann Ihnen die `UMWANDELN()`-Funktion viel Arbeit abnehmen.

Hintergrund

So kann `UMWANDELN()` eine metrische Längenangabe in Yards (und umgekehrt) ändern. Die umzuwandelnden Einheiten müssen allerdings aus derselben Kategorie stammen. So können Sie z.B. Zeitangaben natürlich nicht in Längenmaße umformen. In den oben stehenden Tabellen von Tabelle 16.3 bis Tabelle 16.12 sind die unterschiedlichsten Maßsysteme nach Kategorien geordnet. Die anschließende Tabelle 16.13 führt die Vergrößerungs- und Verkleinerungsfaktoren, die »Einheitenpräfixe« auf.

Bei Einheitenamen und -präfixen wird zwischen Groß- und Kleinschreibung unterschieden.

Wichtig

Zu jeder Möglichkeit dieser Funktion finden Sie in der Tabelle der Abbildung 16.5 ein Beispiel.

Praxiseinsatz

	A	B	C	D	E	F	G	H	I
1	UMWANDELN()								
2									
3	Gewicht und Masse				Energie				
4	1 Slug		14.593,90 g			1 erg		1,00E-07 J	
5	1 Pfund		453,59 g			1 Thermodynamische Kalorie		4,18 J	
6	1 Atommasseseinheit u		1,66E-24 g			1 IT-Kalorie		4,19 J	
7	1 Unze		28,35 g			1 Elektronenvolt (eV)		1,60E-19 J	
8									
9	Länge								
10	1 Feste Meile		1.609,34 m			1 Pferdestärke je Stunde		2.684.519,54 J	
11	1 Seemeile		1.852,00 m			1 Wattstunde (Wh)		3.600,00 J	
12	1 Zoll		0,0254 m			1 Fuß-Pound		1,356 J	
13	1 Fuß		0,305 m			1 BTU		1.055,06 J	
14	1 Yard		0,91 m			Leistung			
15	1 Ängström		1,00E-10 m			1 Pferdestärke		745,70 W	
16	1 Pica (1/72 Zoll)		0,353 mm			Magnetismus			
17									
18	Zeit								
19	1 Jahr		31.557.600 s			Temperatur			
20	1 Tag		86.400 s			37 Grad Celsius (°C)		310,15 K	
21	1 Stunde		3.600 s			98,6 Grad Fahrenheit (°F)		310,15 K	
22	1 Minute		60 s			Flüssigkeiten			
23									
24	Druck								
25	1 Atmosphäre		101.325,00 pa			1 Teelöffel		4,93 ml	
26	1 mm Quecksilber (Torr)		133,32 pa			1 Esslöffel		14,79 ml	
27									
28	Kraft								
29	1 Dyn		1,00E-05 N			1 Flüssigunze		29,57 ml	
30	1 Pound-Kraft		4,45 N			1 Tasse		236,59 ml	
31									
						1 U.S. Pint		473,18 ml	
						1 U.K. Pint		568,26 ml	
						1 Quart		946,35 ml	
						1 Gallone		3.785,41 ml	

Abbildung 16.5: Die Möglichkeiten der Funktion UMWANDELN() auf einen Blick



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe `Umwandeln.xlsx` auf dem Arbeitsblatt `Tabelle1`.

Funktionen für Bitoperationen

BITLVERSCHIEB() BITLSHIFT ()

Syntax BITLVERSCHIEB(*Zahl*; *Bits*)

Definition Die Funktion gibt eine Zahl zurück, deren Bits um eine bestimmte Anzahl von Stellen nach links verschoben wurden.

Argumente *Zahl* (erforderlich) ist eine Zahl, in deren Binärdarstellung die Bits nach links verschoben werden sollen. Sie muss eine ganze Zahl sein und darf nicht negativ sein. Ihr maximaler Betrag ist $2^{48}-1$.

Bits (erforderlich) ist eine ganze Zahl, die die Anzahl der Stellen vorgibt, um die die Bits in *Zahl* nach links verschoben werden soll. *Bits* darf nicht größer als 53 sein.

Ist *Bits* negativ, werden die Bits nach rechts verschoben. Somit gilt:

$$\text{BITLVERSCHIEB}(\text{Zahl}; -\text{Bits}) = \text{BITRVERSCHIEB}(\text{Zahl}; +\text{Bits})$$

Obwohl intern mit Dualzahlen gearbeitet wird, müssen *Zahl* und *Bits* in dezimaler Form eingegeben werden. Das Ergebnis dieser Funktion ist auch eine Dezimalzahl.

Wichtig

Das Bitmuster einer Zahl wird um die angegebene Anzahl nach links geschoben und die frei werdenden Stellen an ihrer rechten Seite werden mit Nullen aufgefüllt.

Hintergrund

Bei der Formel =BITLVERSCHIEB(6;3) wird das erste Argument in seine Dualzahl konvertiert: 6 → 110, dann werden die Bits um 3 Stellen nach links geschoben und rechts genauso viele Nullen angefügt → 110000. Durch Umwandlung dieser Bitfolge in eine Dezimalzahl erhält man das Ergebnis 48.

Das Nach-links-Verschieben um eine Stelle entspricht jeweils einer Multiplikation der Zahl mit 2; Ein Nach-links-Verschieben um 3 Stellen ergibt eine Multiplikation mit 8 (= 2 · 2 · 2).

Die Bitdarstellung von Zahlen wurde ausführlich unter dem Thema »Zahlensysteme« in der Einleitung zu den Umwandlungsfunktionen erläutert (siehe Seite 805).

=BITLVERSCHIEB(6;3) ergibt 48

=BITLVERSCHIEB(63;15) ergibt 2.064.384

BITODER(), BITRVERSCHIEB(), BITUND(), BITXODER()

Praxiseinsatz

Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Bitoperationen.xlsx* auf dem Arbeitsblatt *Praxiseinsatz*.

**Siehe auch**

BITODER() BITOR()

BITODER(*Zahl1*; *Zahl2*)

Syntax

Die Funktion gibt ein bitweises ODER (Disjunktion) zweier Dezimalzahlen als Dezimalzahl zurück.

Definition

Zahl1 (erforderlich) und *Zahl2* (erforderlich) sind die Zahlen, mit deren Binärdarstellung das bitweise ODER gebildet werden soll. *Zahl1* und *Zahl2* müssen positive ganze Zahlen sein. Ihr maximaler Wert darf $2^{48}-1$ sein.

Argumente

Obwohl intern mit Dualzahlen gearbeitet wird, müssen *Zahl1* und *Zahl2* in dezimaler Form eingegeben werden. Das Ergebnis dieser Funktion ist auch eine Dezimalzahl.

Wichtig

Zahl1 und *Zahl2* werden bei dieser Bitoperation Stelle für Stelle verglichen. Sobald wenigstens eine von beiden Zahlen an dieser Stelle ein gesetztes Bit (Wert gleich 1) hat, erhält der Ergebniswert an dieser Stelle ebenfalls eine Eins. Nur wenn keine der beiden Zahlen eine Eins aufweist, wird das Bit im Ergebnis nicht gesetzt (Wert gleich 0).

Hintergrund

Wenn beide Zahlen zufälligerweise an einer Stelle nicht gleichzeitig ein gesetztes Bit haben, dann läuft die BITODER-Operation auf eine Summation hinaus – aber nur unter dieser Bedingung, nicht generell!

Die Bitdarstellung von Zahlen wurde ausführlich unter dem Thema »Zahlensysteme« in der Einleitung zu den Umwandlungsfunktionen erläutert (siehe Seite 805).

=BITODER(63;15) ergibt 63

=BITODER(123;0) ergibt 123

=BITODER(4;2) ergibt 6

Praxiseinsatz

Siehe auch BITLVERSCHIEB(), BITRVERSCHIEB(), BITUND(), BITXODER()



Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Bitoperationen.xlsx* auf dem Arbeitsblatt *BITODER*.

BITUND() BITAND()

Syntax BITUND(*Zahl1*; *Zahl2*)

Definition Die Funktion gibt ein bitweises UND (Konjunktion) zweier Dezimalzahlen als Dezimalzahl zurück.

Argumente *Zahl1* (erforderlich) und *Zahl2* (erforderlich) sind die Zahlen, mit deren Binärdarstellung das bitweise UND gebildet werden soll. *Zahl1* und *Zahl2* müssen positive ganze Zahlen sein. Ihr maximaler Wert darf $2^{48}-1$ sein.

Wichtig Obwohl intern mit Dualzahlen gearbeitet wird, müssen *Zahl1* und *Zahl2* in dezimaler Form eingegeben werden. Das Ergebnis dieser Funktion ist auch eine Dezimalzahl.

Hintergrund Die Bitdarstellung von Zahlen wurde ausführlich unter dem Thema »Zahlensysteme« in der Einleitung zu den Umwandlungsfunktionen erläutert (siehe Seite 805).

Praxiseinsatz =BITUND(63;15) ergibt 15

=BITUND(63;0) ergibt 0

Siehe auch BITLVERSCHIEB(), BITODER(), BITRVERSCHIEB(), BITXODER()



Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Bitoperationen.xlsx* auf dem Arbeitsblatt *BITUND*.

BITRVERSCHIEB() BITRSHIFT()

Syntax BITRVERSCHIEB (*Zahl1*; *Zahl2*)

Definition Die Funktion gibt eine Zahl zurück, deren Bits um eine bestimmte Anzahl von Stellen nach rechts verschoben wurden.

Argumente *Zahl* (erforderlich) ist eine Zahl, in deren Binärdarstellung die Bits nach rechts verschoben werden sollen. Sie muss eine ganze Zahl sein und darf nicht negativ sein. Ihr maximaler Betrag ist $2^{48}-1$.

Bits (erforderlich) ist eine ganze Zahl, die die Anzahl der Stellen vorgibt, um die die Bits in *Zahl* nach rechts verschoben werden soll. *Bits* darf nicht größer als 53 sein.

Ist *Bits* negativ, werden die Bits nach links verschoben. Somit gilt:

$\text{BITRVERSCHIEB}(\text{Zahl}; -\text{Bits}) = \text{BITLVERSCHIEB}(\text{Zahl}; +\text{Bits})$

Wichtig Obwohl intern mit Dualzahlen gearbeitet wird, müssen *Zahl* und *Bits* in dezimaler Form eingegeben werden. Das Ergebnis dieser Funktion ist auch eine Dezimalzahl.

Das Bitmuster einer Zahl wird um die angegebene Anzahl nach rechts geschoben und die frei werdenden Stellen an ihrer linken Seite werden mit Nullen aufgefüllt.

Bei der Formel =BITVERSCHIEB(6;1) wird das erste Argument in seine Dualzahl konvertiert: 6 → 110, dann werden die Bits um 1 Stelle nach rechts geschoben und links genauso viele Nullen, hier also eine, angefügt → 011. Durch Umwandlung dieser Bitfolge in eine Dezimalzahl erhält man das Ergebnis 3.

Das Nach-rechts-Verschieben um eine Stelle entspricht jeweils einer Division der Zahl durch 2; ein Nach-rechts-Verschieben um 3 Stellen ergibt eine Division durch 8:

$$\frac{1}{2 \cdot 2 \cdot 2} = \frac{1}{8}$$

Eventuell entstehende Nachkommastellen werden verworfen.

Die Bitdarstellung von Zahlen wurde ausführlich unter dem Thema »Zahlensysteme« in der Einleitung zu den Umwandlungsfunktionen erläutert (siehe Seite 805).

=BITVERSCHIEB(6;1) ergibt 3

=BITVERSCHIEB(63;5) ergibt 1

=BITVERSCHIEB(64;5) ergibt 2

BITLVERSCHIEB(), BITODER(), BITUND(), BITXODER()

Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Bitoperationen.xlsx* auf dem Arbeitsblatt *Praxiseinsatz*.

BITXODER() BITXOR()

BITXODER(*Zahl1*; *Zahl2*)

Die Funktion gibt ein bitweises XODER (Exklusiv-Oder, ausschließliches Oder, Antivalenz) zweier Dezimalzahlen als Dezimalzahl zurück.

Zahl1 (erforderlich) und *Zahl2* (erforderlich) sind die Zahlen, mit deren Binärdarstellung das bitweise XODER gebildet werden soll. *Zahl1* und *Zahl2* müssen positive ganze Zahlen sein. Ihr maximaler Wert darf $2^{48}-1$ sein.

Obwohl intern mit Dualzahlen gearbeitet wird, müssen *Zahl1* und *Zahl2* in dezimaler Form eingegeben werden. Das Ergebnis dieser Funktion ist auch eine Dezimalzahl.

Bei der BITXODER-Funktion werden die Bits der beiden Argumente Stelle für Stelle miteinander verglichen. Sind ihre Bits verschieden, erhält das Ergebnis an dieser Stelle eine Eins, sind sie gleich, eine Null.

Diese Funktion wird vielfältig in der technischen Logik, insbesondere in der Datenverarbeitung, eingesetzt. Schwerpunkt dabei sind Kryptografie und Prüfsummenbildung für die Sicherheit von IT-Systemen.

Hintergrund

Praxiseinsatz

Siehe auch



Syntax

Definition

Argumente

Wichtig

Hintergrund

So führt das bitweise Exklusiv-Oder zweier Zahlen, zum Beispiel *Wert* und *Schlüssel*, zu einer neuen, verschlüsselten Zahl, die keinen Zusammenhang mit der *Zahl* Wert erkennen lässt. Ein erneutes Exklusiv-Oder mit dieser Zahl und der Zahl *Schlüssel* führt wieder zu *Wert*.

Hinweis Die Bitdarstellung von Zahlen wurde ausführlich unter dem Thema »Zahlensysteme« in der Einleitung zu den Umwandlungsfunktionen erläutert (siehe Seite 805).

Praxiseinsatz =BITXODER(63;15) ergibt 48

=BITXODER(48;15) ergibt 63

Siehe auch BITLVERSCHIEB(), BITODER(), BITUND(), BITRVERSCHIEB()



Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Bitoperationen.xlsx* auf dem Arbeitsblatt *BITXODER*.

Funktionen für komplexe Zahlen

Komplexe Zahlen Insgesamt 26 Funktionen betreffen das Thema »Komplexe Zahlen«. An dieser Stelle möchten wir Ihnen zu diesem Thema einige grundsätzliche Erläuterungen mit auf den Weg geben; bei den einzelnen Funktionen wird hierauf verwiesen.

Als Schüler haben wir gelernt, dass das Quadrat einer beliebigen Zahl immer positiv ist:

$$2^2 = 4$$

$$(-2)^2 = 4$$

Andersherum bedeutet dies, dass wir aus negativen Zahlen nicht die Wurzel ziehen können:

$$\sqrt{-4} = ?$$

Bei den vier Grundrechenarten gibt es für alle Zahlen eine Umkehrung: Addition « Subtraktion und Multiplikation « Division. Sollte das bei Potenzieren « Wurzelziehen nicht gelten? Mathematikern ließ dies keine Ruhe. Denn sie sehen die Forderung der »Vollständigkeit der Mathematik« als gegeben an. Was bedeutet, dass man zumindest theoretisch jedes mathematische Problem lösen können muss, – und sie fanden auch hier eine Lösung. Es zeigte sich sogar, dass diese Lösung auch für die Praxis von Bedeutung ist. Viele technische Berechnungen greifen darauf zurück.

Zur Lösung derartiger Probleme muss man den Zahlenstrahl für die reellen Zahlen – so nennt man in der Mathematik unsere »Alltagszahlen« –, der von $-\infty$ über Null bis $+\infty$ reicht, verlassen und den Zahlenraum, der oberhalb und unterhalb davon liegt, betreten.

Imaginärteil Ähnlich wie bei einem zweidimensionalen Koordinatensystem mit x- und y-Achse gibt es bei den komplexen Zahlen ein kartesisches Achsenkreuz aus Abszisse (x-Achse) und Ordinate (y-Achse). Der Zahlenstrahl der reellen Zahlen bildet die Abszisse; die Ordinate wird mit einer Eigenschaft versehen, die die oben dargestellten mathematischen Probleme in sich ver-

einigt und »imaginäre Einheit i « (oder auch » j «) genannt wird. Die y -Achse heißt nun »imaginär« und x -Achse »reell«. Die Zahlen, die sich in diesem Koordinatensystem (Gaußsche Ebene) darstellen lassen, nennt man »komplexe Zahlen«.

Damit erhält man als Darstellung einer komplexen Zahl z :

$$z = x + yi$$

mit x als Realteil (Strecke auf der x -Achse) und y als Imaginärteil (Strecke auf der y -Achse) der komplexen Zahl (Abbildung 16.6). Die Menge aller Zahlen $x + yi$ bildet den »Körper der komplexen Zahlen«, von denen jede einem geordneten Paar $(x;y)$ entspricht. Neben der Darstellung der komplexen Zahlen in der arithmetischen Form kann man sie auch mit Polarkoordinaten beschreiben – mit Betrag r und Winkel ϕ (die »goniometrische Form«):

$$z = r(\cos \phi + i \sin \phi)$$

Zu dieser Darstellungsform gehören die Excel-Funktionen `IMARGUMENT()` auf Seite 832, `IMCOS()` auf Seite 833 und `IMSIN()` auf Seite 841.

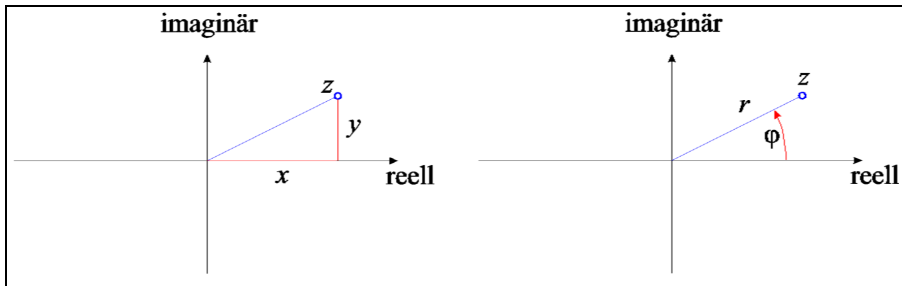


Abbildung 16.6: Grafische Darstellungen einer komplexen Zahl in der arithmetischen Form und in der goniometrischen Form

Die komplexen Zahlen bilden die Grundlage für die *Funktionentheorie*, in der unter vielen anderen die Besselfunktionen entwickelt wurden (siehe den Abschnitt »Hintergrund« zu `BESSELI()` u.a. auf Seite 847ff.).

Aus diesen Grundlagen der komplexen Zahlen ergibt sich ihre besondere Behandlung innerhalb eines Tabellenkalkulationsprogramms wie Excel: Komplexe Zahlen bestehen aus zwei Zahlen, dem Realteil und dem Imaginärteil (bzw. Betrag und Winkel). Tabellenkalkulationen arbeiten nun mit Zellen, in die nur jeweils genau ein Wert eingesetzt werden kann: ein Text, eine Datums-/Zeitangabe, ein Fehlerwert, ein Wahrheitswert, eine Formel oder eben eine Zahl. Excels wirksames Mittel zum Entkommen aus diesem Dilemma ist – wie schon bei den anderen »unmöglichen« Zahlen (Dual-, Oktal- und Hexadezimalzahlen) eingesetzt – die Darstellung der komplexen Zahl als Text, der beide Zahlen verkettet und nun in einer Zelle unterkommen kann.

Wichtig

Dabei wird allerdings der Nachteil erkaufte, dass die gängigen Rechenoperationen und die übrigen Excel-Funktionen darauf nicht angewandt werden können, weil sie »echte« Zahlen erfordern.

Aus der arithmetischen Form in Abbildung 16.6 links ist ersichtlich, wie der Ort der komplexen Zahl z durch den Realteil x und dem Imaginärteil y gegeben ist; der Abstand vom Ursprung ist damit aber noch nicht bekannt. Dieser Abstand entspricht der Länge r in der

goniometrischen Form (Abbildung 16.6 rechts). r wird auch das Modul oder der (Absolut-) Betrag $|z|$ der komplexen Zahl z genannt. Der Absolutwert $|z|$ einer komplexen Zahl wird wie folgt berechnet:

$$|z| = \sqrt{x^2 + y^2}$$

was genau die Funktion IMABS() leistet, die hier direkt folgt.

IMABS() IMABS()

Syntax IMABS(*Komplexe_Zahl*)

Definition Diese Funktion gibt den Absolutwert (Modul) einer komplexen Zahl zurück. Akzeptiert werden Zeichenfolgen der Form $x + yi$ oder $x + yj$.

Argumente *Komplexe_Zahl* (erforderlich) ist die komplexe Zahl, deren Absolutwert berechnet werden soll.

Hinweis Mit der Funktion KOMPLEXE() können Sie aus einer Zahl als Realteil und einer zweiten als Imaginärteil eine komplexe Zahl zusammensetzen.

Hintergrund Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

Praxiseinsatz =IMABS("3-4i") ergibt 5

Beispiel 1

Beispiel 2

In der Abbildung 16.7 werden sämtliche Funktionen, die das Thema »Komplexe Zahlen« berühren, mit jeweils einem Beispiel vorgeführt. In den Zellen mit blauem Hintergrund (C5, C8 und E8) befinden sich die Werte, die als Argumente in den verschiedenen Formeln benötigt werden.

	A	B	C	D	E
1	Funktionen mit komplexen Zahlen - als Argument und/oder Ergebnis				
2	a :	3	→ a + bi		
3	b :	-4			
4	Funktion mit Argumenten (wie sie in Spalte C eingesetzt wurde)		Unformatiertes Ergebnis		Formatiertes Ergebnis
5	=KOMPLEXE(B3;B3) :		3-4i		
6	=IMABS(C5) :		5		
7	=IMAGINARTEIL(C5) :		-4		
8	=IMPOTENZ(C5;E8) :		-7-24i		2
9	=IMARGUMENT(C5) :		-0,927295218		
10	=IMCOS(C5) :		-27,0349456030742+3,85115333481178i		-27,03+3,85i
11	=IMDIV(C5;C8) :		0,12+0,16i		
12	=IMEXP(C5) :		-13,1287830814622+15,200784463068i		-13,13+15,20i
13	=IMKONJUGIERTE(C5) :		3+4i		
14	=IMLN(C5) :		1,6094379124341-0,927295218001612i		1,61-0,93i
15	=IMLOG10(C5) :		0,698970004336019-0,402719196273373i		0,70-0,40i
16	=IMLOG2(C5) :		2,32192809488736-1,33780421245098i		2,32-1,34i
17	=IMPRODUKT(C5;C8) :		-117-44i		
18	=IMREALTEIL(C5) :		3		
19	=IMSIN(C5) :		3,85373803791938+27,0168132580039i		3,85+27,02i
20	=IMSUB(C5;C8) :		10+20i		
21	=IMSUMME(C5;C8) :		-4-28i		
22	=IMWURZEL(C5) :		2-i		

Abbildung 16.7: Übersicht über alle bei den Konstruktions- bzw. technischen Funktionen verfügbaren Funktionen mit Bezug auf komplexe Zahlen

Beachten Sie bitte, dass die Funktionsergebnisse teilweise durch sehr lange Zahlen wiedergegeben werden, wie Sie in Spalte C der Excel-Tabelle erkennen können. Wir können allerdings aus den oben erwähnten Gründen nicht die üblichen Formatierungsmöglichkeiten auf komplexe Zahlen anwenden.

In dieser Tabelle wird eine in VBA (Visual Basic für Applikationen) geschriebene Formatfunktion eingesetzt, die der Microsoft-Support bis vor einiger Zeit im Internet bereitgestellt hatte.

Um das mitgelieferte Beispiel nachvollziehen zu können, benötigen Sie die Datei aus dem Internet aber nicht – genauso wenig wie VBA-Kenntnisse. Diese Funktion ist schon in das Beispiel eingearbeitet. Zum Verwenden dieser Funktion müssen Sie möglicherweise die Makrosicherheit von Excel reduzieren, anderenfalls erhalten Sie in der Spalte E Fehlermeldungen. Das geht bei Excel 2013 mit folgenden Schritten (in Klammern finden Sie die Angaben für Excel 2010):

1. Im Menüband die Registerkarte *DATEI* öffnen
2. Schaltfläche *Optionen* anklicken
3. Kategorie *Trust Center (Sicherheitscenter)* auswählen
4. Schaltfläche *Einstellungen für das Trust Center (Sicherheitscenter)* betätigen
5. Kategorie *Makroeinstellungen (Einstellungen für Makros)* öffnen
6. Option *Alle Makros aktivieren*
7. Zwei Mal mit *OK* bestätigen

Wenn Sie die Funktionen, die Makros und VBA-Module bereitstellen, nicht mehr benötigen, können Sie die Makrosicherheit wieder in den vorherigen Zustand zurücksetzen.

IMAGINÄRTEIL(), IMARGUMENT(), IMEXP(), IMREALTEIL(), KOMPLEXE()

Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner *\Ms5-235\Kap16* in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

IMAGINÄRTEIL() IMAGINARY()

IMAGINÄRTEIL(*Komplexe_Zahl*)

Die Funktion gibt den Imaginärteil y einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt.

Mit der Funktion KOMPLEXE() können Sie aus einer Zahl als Realteil und einer zweiten als Imaginärteil eine komplexe Zahl bilden.

Komplexe_Zahl (erforderlich) ist die komplexe Zahl, deren Imaginärteil Sie ermitteln möchten.

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMAGINÄRTEIL("3-4i") ergibt -4

Dieses Beispiel finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMABS(), IMARGUMENT(), IMEXP(), IMREALTEIL(), KOMPLEXE()

Das Beispiel finden Sie im Ordner *\Ms5-235\Kap16* in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

Tipp

Siehe auch



Syntax

Definition

Hinweis

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



IMAPOTENZ() IMPOWER()

Syntax IMAPOTENZ(*Komplexe_Zahl*;Potenz)

Definition Die Funktion potenziert eine komplexe Zahl, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegen muss, mit einer ganzen Zahl.

Argumente *Komplexe_Zahl* (erforderlich) ist die komplexe Zahl, die Sie in eine Potenz erheben möchten.
Potenz (erforderlich) ist der Exponent, mit dem Sie die komplexe Zahl potenzieren möchten; das Argument *Potenz* kann eine ganze Zahl, eine rationale Zahl (ein Bruch) oder negativ sein.

Hinweis Mit der Funktion KOMPLEXE() können Sie aus einer Zahl als Realteil und einer zweiten als Imaginärteil eine komplexe Zahl bilden.

Hintergrund Eine Potenz einer komplexen Zahl wird nach dem Moivreschen Satz wie folgt berechnet:

$$(x + yi)^n = r^n \cdot e^{n\phi i} = [r \cdot (\cos \phi + i \cdot \sin \phi)]^n = r^n \cdot (\cos n\phi + i \cdot \sin n\phi)$$

wobei

$$r = \sqrt{x^2 + y^2}$$

und

$$\phi(x + yi) = \arctan \frac{y}{x}$$

mit

$$\phi \in (-\pi, \pi]$$

(halboffenes Intervall) gilt.

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

Praxiseinsatz =IMAPOTENZ("3-4i";2) ergibt $-7-24i$

Dieses Beispiel finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

Siehe auch IMEXP(), IMLN(), IMLOG10(), IMLOG2(), IMWURZEL()



Das Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Komplex.xlsxm* auf dem Arbeitsblatt *Komplex*.

IMARGUMENT() IMARGUMENT()

Syntax IMARGUMENT(*Komplexe_Zahl*)

Definition Die Funktion gibt das Argument (ϕ) einer komplexen Zahl zurück, einen Winkel im Bogenmaß für die Darstellung einer komplexen Zahl in der goniometrischen Form (siehe Abbildung 16.6 rechts auf Seite 829).

$$x + yi = |x + yi| \cdot (\cos \phi + i \cdot \sin \phi)$$

Komplexe_Zahl (erforderlich) ist die komplexe Zahl, für die Sie das Argument ϕ ermitteln möchten.

Argumente

Das Argument einer komplexen Zahl in der Form $x + yi$ wird auf folgende Art berechnet:

Hintergrund

$$\phi(x + yi) = \arctan \frac{y}{x}$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMARGUMENT("3-4i") ergibt $-0,927295218$ (das ist der Winkel ϕ im Bogenmaß.)

Praxiseinsatz

Dieses Beispiel finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMABS(), IMAGINÄRTEIL(), IMREALTEIL(), KOMPLEXE()

Siehe auch

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.



IMCOS() IMCOS()

IMCOS(*Komplexe_Zahl*)

Syntax

Die Funktion gibt den Kosinus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt.

Definition

Komplexe_Zahl (erforderlich) ist die komplexe Zahl, deren Kosinus Sie berechnen möchten.

Argumente

Für den Kosinus einer komplexen Zahl gilt

Hintergrund

$$\cos(x + yi) = \cos(x) \cdot \cosh(y) - i \cdot \sin(x) \cdot \sinh(y)$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

sinh und *cosh* sind die mathematischen Funktionen »Sinus Hyperbolicus« und »Cosinus Hyperbolicus«, über die Sie sich in Kapitel 15 »Mathematische und trigonometrische Funktionen« informieren können.

=IMCOS("3-4i") ergibt $-27,03+3,85i$

Praxiseinsatz

Dieses Beispiel (mit mehr Nachkommastellen) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMCOT(), IMSIN(), IMSEC(), IMTAN()

Siehe auch

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.



IMCOSHYP() IMCOSH()

Syntax IMCOSHYP(*Komplexe_Zahl*)

Definition Die Funktion gibt den hyperbolischen Kosinus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt.

Argumente *Komplexe_Zahl* (erforderlich) ist die komplexe Zahl, deren hyperbolischen Kosinus Sie berechnen möchten.

Hintergrund Für den hyperbolischen Kosinus einer komplexen Zahl gilt

$$\cosh(x + yi) = \cosh(x) \cdot \cos(y) + i \cdot \sinh(x) \cdot \sin(y)$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

sinh und *cosh* sind die mathematischen Funktionen »Sinus Hyperbolicus« und »Cosinus Hyperbolicus«, über die Sie sich in Kapitel 15 »Mathematische und trigonometrische Funktionen« informieren können.

Praxiseinsatz =IMCOSHYP("4-3i") ergibt $-27,03+3,85i$

Dieses Beispiel (mit mehr Nachkommastellen) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

Siehe auch IMCOS(), IMSIN(), IMSINHYP()



Das Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

IMCOSEC() IMCSC()

Syntax IMCOSEC(*Komplexe_Zahl*)

Definition Die Funktion gibt den Kosekans einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt.

Argumente *Komplexe_Zahl* (erforderlich) ist die komplexe Zahl, deren Kosekans Sie berechnen möchten.

Hintergrund Für den Kosekans einer komplexen Zahl gilt

$$\operatorname{cosec}(x + yi) = \frac{1}{\sin(x + yi)}$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

Praxiseinsatz =IMCOSEC("4+3i") ergibt $-0,0755+0,0649i$

=IMCOSEC("1-3i") ergibt $0,0838+0,0536i$

Dieses Beispiel (mit mehr Nachkommastellen) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMCOS(), IMSIN(), IMTAN()

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

Siehe auch



IMCOSECHYP() IMCSCH()

IMCOSECHYP(*Komplexe_Zahl*)

Die Funktion gibt den hyperbolischen Kosekans einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt.

Komplexe_Zahl (erforderlich) ist die komplexe Zahl, deren hyperbolischen Kosekans Sie berechnen möchten.

Für den hyperbolischen Kosekans einer komplexen Zahl gilt

$$\operatorname{cosech}(x + yi) = \frac{1}{\sinh(x + yi)} = \frac{2}{e^x - e^{-x}}$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMCOSECHYP("4+3i") ergibt $-0,0363-0,0052i$

=IMCOSECHYP("1-3i") ergibt $-0,8304+0,1554i$

Dieses Beispiel (mit mehr Nachkommastellen) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMCOS(), IMCOSEC(), IMCOSHYP(), IMSIN(), IMSINH()

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



IMCOT() IMCOT()

IMCOT(*Komplexe_Zahl*)

Die Funktion gibt den Kotangens einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt.

Komplexe_Zahl (erforderlich) ist die komplexe Zahl, deren Kotangens Sie berechnen möchten.

Für den Kotangens einer komplexen Zahl gilt

$$\cot(x + yi) = \frac{\cos(x + yi)}{\sin(x + yi)}$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMCOT("4+3i") ergibt $0,0049-0,9992i$

=IMCOT("1-3i") ergibt $0,0045+0,9979i$

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Dieses Beispiel (mit mehr Nachkommastellen) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

Siehe auch IMCOS(), IMSIN(), IMTAN()



Das Beispiel finden Sie im Ordner `Ms5-235\Kap16` in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

IMDIV() IMDIV()

Syntax `IMDIV(Komplexe_Zahl1;Komplexe_Zahl2)`

Definition Die Funktion gibt den Quotienten zweier komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden.

Argumente *Komplexe_Zahl1* (erforderlich) ist der komplexe Zähler oder Dividend.

Komplexe_Zahl2 (erforderlich) ist der komplexe Nenner oder Divisor.

Hintergrund Für den Quotienten z zweier komplexer Zahlen $z_1 = a + bi$ und $z_2 = c + di$ gilt

$$z = \frac{z_1}{z_2} = \frac{(a + bi)}{(c + di)} = \frac{ac + bd}{c^2 + d^2} + i \cdot \frac{bc - ad}{c^2 + d^2}$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

Praxiseinsatz `=IMDIV("3-4i";"-7-24i")` ergibt $0,12 + 0,16i$

Dieses Beispiel finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

Siehe auch IMPRODUKT(), IMSUB(), IMSUMME()



Beispiele finden Sie im Ordner `Ms5-235\Kap16` in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

IMEXP() IMEXP()

Syntax `IMEXP(Komplexe_Zahl)`

Definition Die Funktion gibt die algebraische Form einer komplexen Zahl z zurück, die als Ergebnis der Exponentialfunktion mit einem komplexen Exponenten erhalten wird:

$$z = e^{(x+yi)}$$

Sie ist damit das Gegenstück zur `EXP()`-Funktion, die mit einer reellen Zahl als Argument arbeitet: $z = e$

Die Umkehrfunktion zu `IMEXP()` ist `IMLN()` (siehe Seite 838).

Achtung Wenn Sie die `IMEXP()`- und die `IMLN()`-Funktion zum Vorwärts- und Rückwärtsrechnen einsetzen, werden Sie gelegentlich bemerken, dass der zurückgewonnene Wert nicht der ursprünglichen Zahl entspricht. So wird z.B. aus $(1+4i)$ mittels der `IMEXP()`-Funktion $(-1,77-2,057i)$; beim Logarithmieren dieser komplexen Zahl mit der `IMLN()`-Funktion erhalten Sie jedoch $(1-2,283i)$ und nicht wie erwartet die Ausgangszahl $(1+4i)$.

Dieses auf den ersten Blick sonderbare Verhalten beruht auf der Periodizität der Winkelfunktionen: Bei Winkeln von 0° bis 360° (oder von 0 bis 2π im Bogenmaß) durchläuft der Sinus Werte von Null über 1 zurück zu Null und weiter über -1 wieder zu Null. Bei größer werdenden Winkeln wird diese Folge immer wieder durchlaufen – periodisch mit Ergebnissen zwischen -1 und $+1$.

Damit sich die IMEXP()- und die IMLN()-Funktionen reziprok verhalten, muss der Imaginärteil y von *Komplexe_Zahl* bei IMEXP() im Bereich von $-\pi$ bis $+\pi$ liegen. Wenn y außerhalb dieses Bereichs liegt, erhalten Sie beim Anwenden der Umkehrfunktion unerwartete (aber korrekte) Ergebnisse.

Komplexe_Zahl (erforderlich) ist der Exponent der Exponentialfunktion, der als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben werden muss.

Für eine in exponentieller Schreibweise vorliegende komplexe Zahl gilt wegen der Eulerschen Formel:

$$e^{(x+yi)} = e^x \cdot e^{yi} = e^x \cdot (\cos y + i \cdot \sin y)$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMEXP("3-4i") ergibt $-13,13+15,20i$

Dieses Beispiel finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

EXP(), IMABS(), IMAGINÄRTEIL(), IMARGUMENT(), IMREALTEIL(), KOMPLEXE()

Beispiele finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Komplex.xlsx* auf dem Arbeitsblatt *Komplex*.

IMKONJUGIERTE() IMCONJUGATE()

IMKONJUGIERTE(*Komplexe_Zahl*)

Die Funktion gibt die konjugiert komplexe Zahl zu einer komplexen Zahl zurück, wobei die komplexe Zahl als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird.

Komplexe_Zahl (erforderlich) ist die komplexe Zahl, deren konjugiert komplexe Zahl Sie erzeugen möchten.

Für die konjugiert komplexe Zahl \bar{z} einer komplexen Zahl $z = x + yi$ gilt:

$$\bar{z} = x - yi$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMKONJUGIERTE("3-4i") ergibt $3+4i$

Dieses Beispiel finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

KOMPLEXE()

Beispiele finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Komplex.xlsx* auf dem Arbeitsblatt *Komplex*.

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



IMLN()  **IMLN()****Syntax** IMLN(*Komplexe_Zahl*)**Definition** Die Funktion gibt den natürlichen Logarithmus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird. Sie ist die für komplexe Zahlen zu verwendende Variante der reellen LN()-Funktion aus der Kategorie »Mathematik & Trigonometrie«.

Die IMLN()-Funktion stellt die Umkehrfunktion zu IMEXP() dar. Beachten Sie dazu bitte die Warnung bei IMEXP() auf Seite 836!

Argumente *Komplexe_Zahl* (erforderlich) ist die komplexe Zahl, deren natürlichen Logarithmus Sie berechnen möchten.**Hintergrund** Für den natürlichen Logarithmus einer komplexen Zahl gilt

$$\ln(x + yi) = \ln \sqrt{x^2 + y^2} + i \cdot \arctan \frac{y}{x}$$

mit

$$\phi \in (-\pi, \pi]$$

(halboffenes Intervall).

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 820 wichtige Grundlagen nachlesen.

Praxiseinsatz =IMLN("3-4i") ergibt 1,61-0,93i

Dieses Beispiel (mit mehr Dezimalstellen) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

Siehe auch IMAPOTENZ(), IMEXP(), IMLOG10(), IMLOG2(), IMWURZEL(), LN()Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.**IMLOG10()**  **IMLOG10()****Syntax** IMLOG10(*Komplexe_Zahl*)**Definition** Die Funktion gibt den Logarithmus einer komplexen Zahl zur Basis 10 zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird.

Der gewöhnliche Logarithmus einer komplexen Zahl kann wie folgt aus dem natürlichen Logarithmus berechnet werden:

$$\log_{10}(x + yi) = \log_{10} e \cdot \ln(x + yi)$$

Argumente *Komplexe_Zahl* ist die komplexe Zahl, deren gewöhnlichen (dekadischen) Logarithmus Sie berechnen möchten.**Hintergrund** Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMLOG10("3-4i") ergibt $0,70-0,40i$

Dieses Beispiel (mit höherer Genauigkeit) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMAPOTENZ(), IMEXP(), IMLN(), IMLOG2(), IMWURZEL(), LOG10()

Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

Praxiseinsatz

Siehe auch



IMLOG2() IMLOG2()

IMLOG2(*Komplexe_Zahl*)

Die Funktion gibt den Logarithmus einer komplexen Zahl zur Basis 2 zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird.

Komplexe_Zahl (erforderlich) ist die komplexe Zahl, deren Zweierlogarithmus Sie berechnen möchten.

Der Zweierlogarithmus einer komplexen Zahl wird wie folgt aus dem natürlichen Logarithmus berechnet:

$$\log_2(x + yi) = \log_2 e \cdot \ln(x + yi)$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMLOG2("3-4i") ergibt $2,32-1,34i$

Dieses Beispiel (mit größerer Genauigkeit) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMAPOTENZ(), IMEXP(), IMLN(), IMLOG10(), IMWURZEL(), LOG()

Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

Siehe auch



IMPRODUKT() IMPRODUKT()

IMPRODUKT(*Komplexe_Zahl1*; *Komplexe_Zahl2*; ...)

Die Funktion gibt das Produkt komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden.

Komplexe_Zahl1 (erforderlich), *Komplexe_Zahl2*, ... (optional) sind 1 bis 255 komplexe Zahlen, die multipliziert werden sollen.

Das Produkt zweier komplexer Zahlen $(a + bi)$ und $(c + di)$ wird wie folgt berechnet:

$$(a + bi) \cdot (c + di) = (ac - bd) + i \cdot (ad + bc)$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz =IMPRODUKT("3-4i";"-1-24i") ergibt $-99-68i$

Dieses Beispiel finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

Siehe auch IMDIV(), IMSUB(), IMSUMME(), PRODUKT()



Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

IMREALTEIL() IMREAL()

Syntax IMREALTEIL(*Komplexe_Zahl*)

Definition Die Funktion liefert den Realteil x einer komplexen Zahl, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt.

Argumente *Komplexe_Zahl* (erforderlich) ist die komplexe Zahl, zu der der Realteil ermittelt werden soll.

Hintergrund Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

Praxiseinsatz =IMREALTEIL("3-4i") ergibt 3

Dieses Beispiel finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

Siehe auch IMABS(), IMAGINÄRTEIL(), IMARGUMENT(), IMEXP(), KOMPLEXE()



Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

IMSEC() IMSEC()

Syntax IMSEC(*Komplexe_Zahl*)

Definition Die Funktion gibt den Sekans einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt.

Argumente *Komplexe_Zahl* (erforderlich) ist die komplexe Zahl, deren Sekans Sie berechnen möchten.

Hintergrund Für den Sekans einer komplexen Zahl gilt

$$\sec(x + yi) = \frac{1}{\cos(x + yi)}$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

Praxiseinsatz =IMSEC("4+3i") ergibt $-0,0653-0,0752i$

=IMSEC("1-3i") ergibt $0,0540-0,08375i$

Dieses Beispiel (mit mehr Nachkommastellen) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMCOS(), IMCOSEC(), IMSIN()

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

Siehe auch**IMSECHYP()**  **IMSECH()**

IMSECHYP(Komplexe_Zahl)

Die Funktion gibt den hyperbolischen Sekans einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt.

Komplexe_Zahl (erforderlich) ist die komplexe Zahl, deren hyperbolischen Sekans Sie berechnen möchten.

Für den hyperbolischen Sekans einer komplexen Zahl gilt

$$\operatorname{sech}(x + yi) = \frac{1}{\cosh(x + yi)} = \frac{2}{e^x + e^{-x}}$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMSECHYP("4+3i") ergibt $-0,0363-0,0052i$

=IMSECHYP("1-3i") ergibt $-0,64707+0,07027i$

Dieses Beispiel (mit mehr Nachkommastellen) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMCOSHYP(), IMCOSEC(), IMSINHYP()

Das Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

Syntax**Definition****Argumente****Hintergrund****Praxiseinsatz****Siehe auch****IMSIN()**  **IMSIN()**

IMSIN(Komplexe_Zahl)

Die Funktion gibt den Sinus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegen muss.

Komplexe_Zahl (erforderlich) ist die komplexe Zahl, deren Sinus zurückgegeben werden soll.

Der Sinus einer komplexen Zahl ist wie folgt definiert:

$$\sin(x + yi) = \sin(x) \cdot \cosh(y) - i \cdot \cos(x) \cdot \sinh(y)$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

Syntax**Definition****Argumente****Hintergrund**

\sinh und \cosh sind die mathematischen Funktionen »Sinus Hyperbolicus« und »Cosinus Hyperbolicus«, über die Sie sich in Kapitel 15 »Mathematische und trigonometrische Funktionen« informieren können.

Praxeinsatz =IMSIN("3-4i") ergibt 3,85+27,02i

=IMSIN("3+4i") ergibt 3,85-27,02i

Dieses Beispiel (mit mehr Dezimalstellen) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

Siehe auch IMCOS(), IMCOSHYP(), IMSINHYP()



Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

IMSINHYP() IMSINH()

Syntax IMSINHYP(*Komplexe_Zahl*)

Definition Die Funktion gibt den hyperbolischen Sinus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegen muss.

Argumente *Komplexe_Zahl* (erforderlich) ist die komplexe Zahl, deren hyperbolischer Sinus zurückgegeben werden soll.

Hintergrund Der hyperbolische Sinus einer komplexen Zahl ist wie folgt definiert:

$$\sinh(x + yi) = \sinh(x) \cdot \cos(y) + i \cdot \cosh(x) \cdot \sin(y)$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

\sinh und \cosh sind die mathematischen Funktionen »Sinus Hyperbolicus« und »Cosinus Hyperbolicus«, über die Sie sich in Kapitel 15 »Mathematische und trigonometrische Funktionen« informieren können.

Praxeinsatz =IMSINHYP("3-4i") ergibt -6,548+7,619i

=IMSINHYP("3+4i") ergibt -6,548-7,619i

Dieses Beispiel (mit mehr Dezimalstellen) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

Siehe auch IMCOS(), IMCOSHYP(), IMSIN()



Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.

IMSUB() IMSUB()

Syntax IMSUB(*Komplexe_Zahl1*; *Komplexe_Zahl2*)

Definition Die Funktion gibt die Differenz zweier komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ vorliegen müssen.

Komplexe_Zahl1 (erforderlich) ist die komplexe Zahl, von der **Komplexe_Zahl2** subtrahiert werden soll.

Argumente

Komplexe_Zahl2 (erforderlich) ist die komplexe Zahl, die von **Komplexe_Zahl1** subtrahiert werden soll.

Die Differenz zweier komplexer Zahlen wird berechnet, indem man die Differenz der Realteile und die Differenz der Imaginärteile bildet:

Hintergrund

$$(a + bi) - (c + di) = (a - c) + i \cdot (b - d)$$

D.h. die Realteile und Imaginärteile beider komplexer Zahlen werden getrennt subtrahiert.

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMSUB("3-4i";"-7-24i") ergibt 10+20i

Praxiseinsatz

Dieses Beispiel finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMDIV(), IMPRODUKT(), IMSUMME()

Siehe auch

Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.



IMSUMME() IMSUM()

IMSUMME(**Komplexe_Zahl1**;**Komplexe_Zahl2**;...)

Syntax

IMSUMME() gibt die Summe komplexer Zahlen zurück, die als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden.

Definition

Komplexe_Zahl1 (erforderlich), **Komplexe_Zahl2**,... (optional) sind 1 bis 255 komplexe Zahlen, die addiert werden.

Argumente

Die Summe zweier komplexer Zahlen wird berechnet, indem man die Realteile und die Imaginärteile addiert:

Hintergrund

$$(a + bi) + (c + di) = (a + c) + i \cdot (b + d)$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMSUMME("3-4i";"-7-24i") ergibt -4-28i

Praxiseinsatz

Dieses Beispiel finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMDIV(), IMPRODUKT(), IMSUB() sowie SUMME() aus der Kategorie »Mathematik & Trigonometrie«

Siehe auch

Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsm* auf dem Arbeitsblatt *Komplex*.



IMTAN() IMTAN()

Syntax IMTAN(*Komplexe_Zahl*)

Definition Die Funktion gibt den Tangens einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegen muss.

Argumente *Komplexe_Zahl* (erforderlich) ist die komplexe Zahl, deren Tangens zurückgegeben werden soll.

Hintergrund Der Tangens einer komplexen Zahl ist definiert:

$$\tan(x + yi) = \frac{\sin(x + yi)}{\cos(x + yi)} = \frac{\sin(2x)}{\cos(2x) + \cosh(2y)} + i \frac{\sinh(2y)}{\cos(2x) + \cosh(2y)}$$

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

\sinh und \cosh sind die mathematischen Funktionen »Sinus Hyperbolicus« und »Cosinus Hyperbolicus«, über die Sie sich in Kapitel 15 »Mathematische und trigonometrische Funktionen« informieren können.

Praxiseinsatz =IMTAN("3-4i") ergibt $-0,0001873-0,9994i$

=IMTAN("3+4i") ergibt $-0,0001873+0,9994i$

Dieses Beispiel (mit mehr Dezimalstellen) finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

Siehe auch IMCOS(), IMCOSHYP(), IMSIN(), IMSINHYP(), TAN()



Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsx* auf dem Arbeitsblatt *Komplex*.

IMWURZEL() IMSQRT()

Syntax IMWURZEL(*Komplexe_Zahl*)

Definition Diese Funktion gibt die Quadratwurzel einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ erwartet wird.

Argumente *Komplexe_Zahl* (erforderlich) ist die komplexe Zahl, deren Quadratwurzel ermittelt werden soll.

Hintergrund Die Quadratwurzel einer komplexen Zahl wird wie folgt berechnet:

$$\sqrt{x + yi} = \sqrt{r} \cdot \left(\cos \frac{\phi}{2} + i \cdot \sin \frac{\phi}{2} \right)$$

wobei

$$r = \sqrt{x^2 + y^2}$$

und

$$\phi = \arctan \frac{y}{x}$$

mit

$$\phi \in (-\pi, \pi]$$

(halboffenes Intervall) ist.

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=IMWURZEL("3-4i") ergibt $2-i$

Dieses Beispiel finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMAPOTENZ(), IMEXP(), IMLN(), IMLOG10(), IMLOG2() sowie WURZEL() aus der Kategorie »Mathematik & Trigonometrie«

Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsx* auf dem Arbeitsblatt *Komplex*.

Praxiseinsatz**Siehe auch**

KOMPLEXE() COMPLEX()

KOMPLEXE(*Realteil*; *Imaginärteil*; *Suffix*)

KOMPLEXE() verknüpft zwei reelle Zahlen zu einer komplexen Zahl; die erste Zahl bildet den Realteil der komplexen Zahl und die zweite den Imaginärteil ($x + yi$ oder $x + yj$).

Realteil (erforderlich) ist der Realteil der zu bildenden komplexen Zahl.

Imaginärteil (erforderlich) ist der Imaginärteil der zu bildenden komplexen Zahl.

Suffix (optional) ist der Buchstabe, der für die imaginäre Einheit der komplexen Zahl verwendet werden soll. Wenn das Argument *Suffix* fehlt, wird es als »i« angenommen.

Komplexe Zahlen, die in Tabellenfunktionen als Argument eingesetzt werden sollen, müssen immer in der algebraischen Form $a + bi$ vorliegen. Die imaginäre Einheit kann wahlweise i oder j sein und muss immer mit Kleinbuchstaben geschrieben werden; innerhalb der Argumentliste einer Funktion darf die Schreibweise der imaginären Einheit nicht wechseln.

Über komplexe Zahlen können Sie in der Einleitung zum Abschnitt »Funktionen für komplexe Zahlen« auf Seite 828 wichtige Grundlagen nachlesen.

=KOMPLEXE(3;-4) ergibt $3-4i$

=KOMPLEXE(3;-4;"j") ergibt $3-4j$

Diese Beispiele finden Sie mit den Beispielen der anderen komplexen Funktionen bei der zuerst aufgeführten Funktion der komplexen Zahlen auf Seite 830.

IMABS(), IMAGINÄRTEIL(), IMARGUMENT(), IMEXP(), IMREALTEIL()

Beispiele mit allen Funktionen zu dem Thema »Komplexe Zahlen« finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Komplex.xlsx* auf dem Arbeitsblatt *Komplex*.

Syntax**Definition****Argumente****Wichtig****Hintergrund****Praxiseinsatz****Siehe auch**

Funktionen der höheren Mathematik

Mathematik im Alltag

Geht es Ihnen auch so? Obwohl wir in der Schule und Berufsausbildung eine Menge gelernt haben, wenden wir das Gelernte auf unsere Beobachtungen im Alltag nicht an; das gilt vor allem in der Mathematik und den Naturwissenschaften. Wir haben das ja schon so oft gesehen.

Dennoch: Manchmal passiert es eben. Wie neulich – ein paar Jungen toben mit einem Seil herum. Unter ihnen Kevin, ein etwas jüngerer und schwächerer Junge, der aber seine Nachteile an Jahren und Körpergröße mit Pffiffigkeit wettzumachen versucht.

Kevin und ein anderer Junge drehen ein Seil herum, über das die anderen springen. Kevin versucht nun das Seil dem anderen wegzunehmen. Das geht aber nicht. Der andere Junge ist zu groß und zu stark. Als dieser sich jedoch ablenken lässt, reißt Kevin seine Hände kurz und kräftig nach oben und sofort wieder nach unten. Der Ruck wandert das Seil entlang und entreißt dem anderen das Seil.

Wie kommt der Ruck bei dem anderen Jungen an? Nun hat das Seil unsere Aufmerksamkeit. Zum Glück versucht Kevin seinen Erfolg zu wiederholen, als auf der anderen Seite das Seil wieder festgehalten wird. Noch mal ein Ruck und noch einer – aber sein Kompagnon passt jetzt auf. Ein Doppelruck. So können wir deutlich sehen, wie dieses Rucken mehrere Ausschläge nach oben und unten erzeugt, die das Seil entlanglaufen. Die Form dieser Ausschläge kommt uns sehr vertraut vor. Richtig – das sind Sinuskurven! Wir beobachten gerade ein mechanisches Modell für die Ausbreitung elektromagnetischer Wellen in einem elektrischen Leiter. Wenn Sie sich die Form von Sinuskurven noch einmal ansehen wollen, dann schauen Sie doch einfach in Kapitel 15 »Mathematische und trigonometrische Funktionen« bei der Funktion $\text{SIN}()$ hinein!

Kann uns Kevin noch mehr zeigen? Jetzt halten unsere jungen Darsteller das Seil mal ruhig. Also nichts Besonderes – oder doch? Welche Form hat eigentlich ein hängendes Seil? Die Antwort auf diese Frage können Sie ebenfalls in Kapitel 15 »Mathematische und trigonometrische Funktionen« finden, dieses Mal bei der Funktion $\text{COSHYP}()$. Sie können auch im Stichwortverzeichnis unter »Kettenlinie« nachschauen, so heißt diese Form nämlich.

Inzwischen hat Kevin das Seil mit dem einen Ende an einem Klettergerüst festgebunden und bewegt das lose Ende rhythmisch auf und ab. Sind das wieder Sinus-Kurven? Nein, sie werden zum befestigten Ende hin kleiner und auch die Form ist irgendwie anders. Das, was Kevin uns hier so anschaulich vor Augen führt, nennt der Techniker »gedämpfte Schwingungen«. Und wie kann man diese mathematisch beschreiben? »Mit Besselfunktionen« lautet die Antwort. Deren Beschreibung folgt hier. Zuvor sollten Sie sich aber die Diagramme zu den Funktionen $\text{BESSELJ}()$ und $\text{BESSELY}()$ auf Seite 849 und Seite 853 ansehen – wegen Kevin.

Bessel- funktionen

Die Besselfunktionen (oder Zylinderfunktionen) sind die bei der Integration der Besselschen Differenzialgleichung durch Reihenentwicklung gefundenen Lösungen, die bei manchen elektrotechnischen Problemen die Möglichkeit einer exakten (oder doch recht genauen) Berechnung bieten, z.B. bei der Stromverdrängung in Leitern. Ferner werden sie bei der Beschreibung von Schwingungen und den dazu gehörenden Dämpfungen, zur Berechnung von Planetenbahnen und Bahnstörungen bei Himmelskörpern, wo sie von Friedrich Wilhelm Bessel (Königsberger Astronom, 1784 bis 1846) eingeführt wurden, und zur Simulation von Pumpversuchen in der Hydrogeologie angewendet und sind in der Quantenmechanik,

der Satelliten-Geodäsie (Erdvermessung) und der Nachrichtentechnik von Wichtigkeit. Sie entstehen typischerweise als Lösungen der Wellengleichung, wobei zylindrische Randbedingungen zugrunde liegen.

Nach den Besselfunktionen folgen noch zwei Funktionen, mit denen man die Wahrscheinlichkeit berechnen kann, mit der ein bestimmter Wert innerhalb oder außerhalb einer Gaußverteilung liegt (siehe STANDNORMVERT() in Kapitel 11 »Statistische Funktionen«).

Fehlerintegrale

BESSELI() BESSELI()

BESSELI($z;n$)

Syntax

Die Funktion gibt die modifizierte Besselfunktion der ersten Art $I_n(z)$ zurück, die der für rein imaginäre Argumente ausgewerteten Besselfunktion J_n entspricht.

Definition

z (erforderlich) ist der Wert, für den die Funktion ausgewertet werden soll; x muss eine reelle Zahl sein und kann Werte von über ± 700 annehmen (der Wertebereich variiert mit der Ordnung n).

Argumente

n (erforderlich) ist die Ordnung der Besselfunktion und muss positiv und ganzzahlig sein. Ist n keine ganze Zahl, werden deren Nachkommastellen von der Funktion für die Berechnung ignoriert, also abgeschnitten und nicht gerundet.

$I_n(z)$ gehört wie auch $K_n(z)$ zu den modifizierten Besselfunktionen, die – im Gegensatz zu den normalen Besselfunktionen $J_n(z)$ und $Y_n(z)$, die oszillieren, – exponentiell ansteigen oder fallen. $I_n(z)$ ist die Lösung der leicht veränderten Besselschen Differenzialgleichung

Hintergrund

$$x^2 y'' + xy' - (x^2 + n^2) \cdot y = 0$$

bzw.

$$x^2 \cdot \frac{d^2}{dx^2} y + x \cdot \frac{d}{dx} y - (x^2 + n^2) \cdot y = 0$$

Diese modifizierte Besselfunktion der ersten Art $I_n(z)$ kann als das Ringintegral

$$I_n(z) = \frac{1}{2\pi i} \cdot \int_{\gamma} e^{\frac{z}{2}(t+\frac{1}{t})} t^{-n-1} dt$$

n -ter Ordnung beschrieben werden. Für eine reelle Zahl ν kann die Funktion mittels

$$I_\nu(z) = \left(\frac{1}{2}z\right)^\nu \sum_{k=0}^{\infty} \frac{\left(\frac{1}{4}z^2\right)^k}{k! \Gamma(\nu + k + 1)}$$

berechnet werden, wobei $\Gamma(y)$ die Gammafunktion ist.

Weil der Gebrauch der Besselfunktionen vertiefte Kenntnisse im jeweiligen Anwendungsgebiet erfordert, wird auf sehr spezielle Praxisbeispiele verzichtet und der praktische Einsatz anhand der mathematischen Eigenschaften und ihrer grafischen Darstellung erklärt.

Praxiseinsatz

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
1	Modifizierte Bessel-Funktion 1. Art							$I_n(x)$										
2																		
3	Ordnung	x	0,00	0,25	0,50	0,75	1,00	1,25	1,50	1,75	2,00	2,25	2,50	2,75	3,00	3,25	3,50	
4	0	$I_n(x)$	1,00	1,02	1,06	1,15	1,27	1,43	1,65	1,93	2,28	2,73	3,29	4,00	4,88	5,99	7,38	
5	1		0,00	0,13	0,26	0,40	0,57	0,76	0,98	1,26	1,59	2,00	2,52	3,16	3,95	4,95	6,21	
6	2		0,00	0,01	0,03	0,07	0,14	0,22	0,34	0,49	0,69	0,95	1,28	1,70	2,25	2,94	3,83	
7	3		0,00	0,00	0,00	0,01	0,02	0,04	0,08	0,13	0,21	0,32	0,47	0,68	0,96	1,33	1,83	
8	4		0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,03	0,05	0,09	0,14	0,21	0,33	0,48	0,70	

Abbildung 16.8: Ausschnitt aus der Excel-Tabelle *Bessel_1.xls* zur Berechnung der Funktionswerte für die Ordnungen 0 bis 4

Die Zelle mit dem Text »Ordnung« hat in der Beispieldatei zum Buch den Bezug A3. Im Bereich C4 bis Q8 stehen die Formeln mit den BESSELI()-Funktionen. Ihr erstes Argument, der x-Wert, befindet sich jeweils in der Zelle oberhalb in der Zeile 3 und in der Spalte A in derselben Zeile wurde das zweite Argument, die Ordnung der Funktion, eingetragen, die für alle Formeln in einer Zeile gleich ist. Damit muss die vollständige Formel in C4 beispielsweise lauten

=BESSELI(C\$3;\$A4)

Aus dem Bereich C3 bis Q8 lässt sich das in Abbildung 16.9 gezeigte Diagramm erstellen, das die modifizierte Besselfunktion der ersten Art grafisch darstellt. Gut ist der exponentielle Anstieg zu erkennen, der auch schon für kleine x-Werte große Beträge hervorruft.

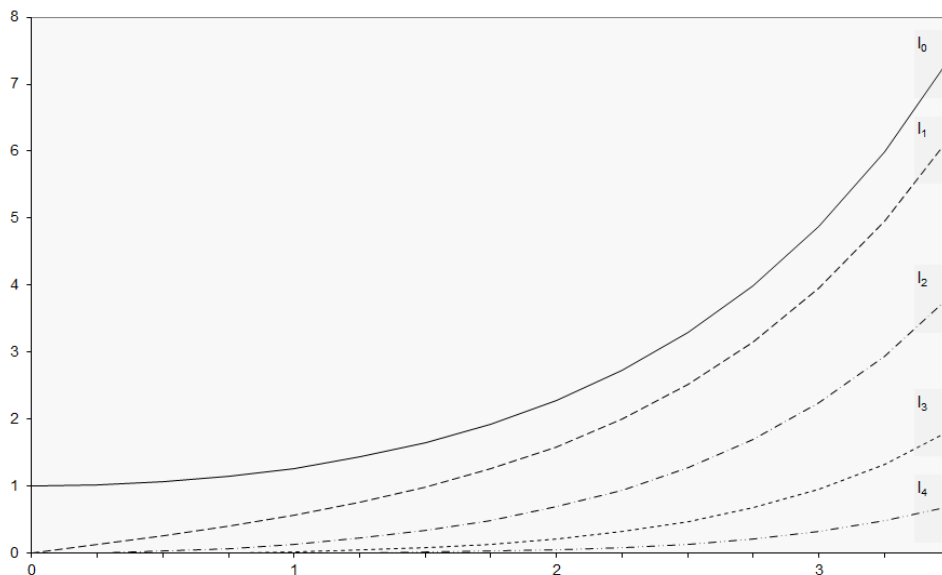


Abbildung 16.9: Grafische Darstellung der Besselfunktion der ersten Art $I_n(x)$ mittels der BESSELI()-Funktion

BESSELJ(), BESSELK(), BESSELY()

Dieses Beispiel finden Sie im Ordner `Ms5-235\Kap16` in der Arbeitsmappe `Bessel_I.xlsx` auf dem Arbeitsblatt `Bessel I`. Die grafische Darstellung befindet sich auf dem Arbeitsblatt `Diagramm`.

Siehe auch



BESSELJ() BESSELJ()

BESSELJ(*x;n*)

Die Funktion gibt die Besselfunktion der ersten Art $J_n(x)$ zurück.

x (erforderlich) ist der Wert, für den die Funktion ausgewertet werden soll; *x* muss eine reelle Zahl sein und kann Werte von unter $-1,34 \cdot 10^8$ bis über $+1,34 \cdot 10^8$ annehmen. (Der Wertebereich variiert mit der Ordnung *n*.)

n (erforderlich) ist die Ordnung der Besselfunktion und muss positiv sein. Ist *n* keine ganze Zahl, werden deren Nachkommastellen von der Funktion für die Berechnung ignoriert, also nicht gerundet.

Die Besselfunktion der ersten Art $J_n(x)$ ist die Lösung der Besselschen Differenzialgleichung

$$x^2 y'' + xy' + (x^2 - n^2) \cdot y = 0$$

bzw.

$$x^2 \cdot \frac{d^2}{dx^2} y + x \cdot \frac{d}{dx} y + (x^2 - n^2) \cdot y = 0$$

Für eine reelle Zahl *v* kann die Funktion mittels

$$J_\nu(z) = \sum_{k=0}^{\infty} \frac{(-1)^k z^{n+2k}}{k! \cdot 2^{n+2k} \cdot \Gamma(\nu + k + 1)}$$

berechnet werden, wobei $\Gamma(y)$ die Gamma-Funktion ist.

Wegen des sehr speziellen Charakters der Besselfunktionen wird auch hier, wie bei den anderen Besselfunktionen, nur auf deren grafische Darstellung eingegangen (siehe Seite 847).

In dieser Tabelle, die Sie ebenfalls finden können, werden die Funktionswerte für die BESSELJ()-Funktion mit den Ordnungen von 0 bis 4 berechnet und auf dem folgenden Diagrammblatt (siehe Abbildung 16.11) gezeichnet.

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	Bessel-Funktion 1. Art							J _n (x)								
2																
3	Ordnung	x	0,00	0,25	0,50	0,75	1,00	1,25	1,50	1,75	2,00	2,25	2,50	2,75	3,00	
4	0	J_n(x)	1,00	0,98	0,94	0,86	0,77	0,65	0,51	0,37	0,22	0,08	-0,05	-0,16	-0,26	
5	1		0,00	0,12	0,24	0,35	0,44	0,51	0,56	0,58	0,58	0,55	0,50	0,43	0,34	
6	2		0,00	0,01	0,03	0,07	0,11	0,17	0,23	0,29	0,35	0,40	0,45	0,47	0,49	
7	3		0,00	0,00	0,00	0,01	0,02	0,04	0,06	0,09	0,13	0,17	0,22	0,26	0,31	
8	4		0,00	0,00	0,00	0,00	0,00	0,01	0,01	0,02	0,03	0,05	0,07	0,10	0,13	

Abbildung 16.10: Berechnung der BESSELJ()-Funktion für die Ordnungen n = 0...4 (Ausschnitt)

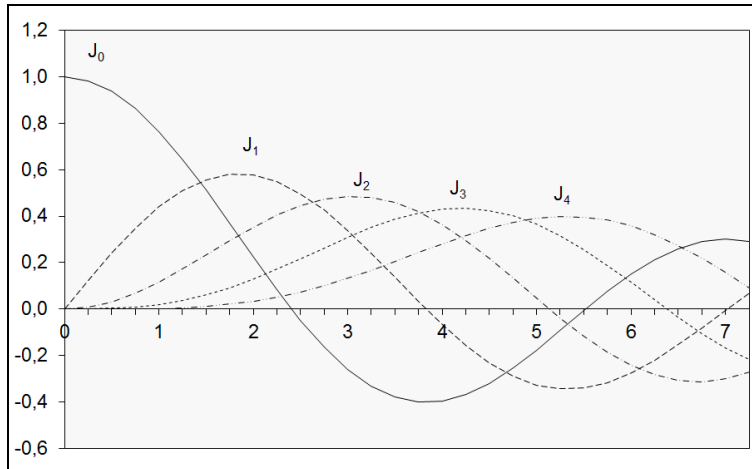


Abbildung 16.11: Grafische Darstellung der Besselfunktion der ersten Art $J_n(x)$ mittels der BESSELJ()-Funktion

Siehe auch BESSELI(), BESSELK(), BESSELY()



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe `Bessel_J.xlsx` auf dem Arbeitsblatt `Bessel J`; die dazugehörige Abbildung auf dem Arbeitsblatt `Diagramm`.

BESSELK() BESSELK()

Syntax BESSELK($x;n$)

Definition Die Funktion gibt die modifizierte Besselfunktion 2. Art $K_n(x)$ zurück.

Argumente x (erforderlich) ist der Wert, für den die Funktion ausgewertet werden soll; x muss eine reelle Zahl sein und kann Werte von $+1 \cdot 10^{-307}$ bis über $+700$ annehmen. Der genaue Wert für die Obergrenze von x ist von der Ordnung n abhängig (und umgekehrt), liegt aber weit oberhalb der in der Praxis vorkommenden Zahlen.

n (erforderlich) ist die Ordnung der Besselfunktion und muss eine positive und ganzzahlige Zahl sein, die größer als 100 sein kann. In der Praxis kommen in der Regel jedoch nur Ordnungen kleiner als 10 vor. Ist n keine ganze Zahl, werden deren Nachkommastellen von der Funktion für die Berechnung ignoriert, also abgeschnitten und nicht gerundet.

Hintergrund $K_n(x)$ entspricht den für rein imaginäre Argumente ausgewerteten Besselfunktionen J_n und Y_n . Sie wird gelegentlich »Basset-Funktion«, »Macdonaldsche Funktion« oder auch modifizierte Besselfunktion der 3. Art genannt. Sie ist die Lösung der modifizierten Besselschen Differenzialgleichung

$$x^2 y'' + xy' - (x^2 + n^2) \cdot y = 0$$

bzw.

$$x^2 \cdot \frac{d^2}{dx^2} y + x \cdot \frac{d}{dx} y - (x^2 + n^2) \cdot y = 0$$

Die modifizierte Besselfunktion zweiter Art $K_n(x)$ der Ordnung n kann in Termen der modifizierten Besselfunktion der ersten Art $I_n(x)$ ausgedrückt werden:

$$K_n(x) = \lim_{p \rightarrow n} \frac{I_{-p}(x) - I_p(x)}{\sin p\pi} \quad \text{mit } n = 0, 1, 2, \dots$$

Wegen des sehr speziellen Charakters der Besselfunktionen wird auch hier, wie bei der BESSELI()-Funktion, nur auf deren grafische Darstellung eingegangen (siehe Seite 847).

Praxiseinsatz

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
1	Modifizierte Bessel-Funktion 2. Art						$K_n(x)$									
2																
3	Ordnung	x	0,00	0,10	0,30	0,50	0,70	0,90	1,10	1,30	1,50	1,70	1,90	2,10	2,30	
4	0	$K_n(x)$	7,02	2,43	1,37	0,92	0,66	0,49	0,37	0,28	0,21	0,17	0,13	0,10	0,08	
5	1			9,85	3,06	1,66	1,05	0,72	0,51	0,37	0,28	0,21	0,16	0,12	0,09	
6	2				21,75	7,55	3,66	2,08	1,29	0,85	0,58	0,41	0,30	0,22	0,16	
7	3						21,97	9,96	5,21	2,99	1,83	1,18	0,78	0,54	0,38	
8	4									14,66	7,92	4,57	2,78	1,75	1,14	

Abbildung 16.12: Berechnung der BESSELK()-Funktion für die Ordnungen $n = 0 \dots 4$

Mit dieser Tabelle lassen sich die Funktionswerte der BESSELK()-Funktion für die Ordnungen $n = 0 \dots 4$ berechnen und grafisch darstellen. Bei den höheren Ordnungen werden die ersten Formeln weggelassen, weil ihre Funktionswerte für eine übersichtliche Grafik zu groß sind.

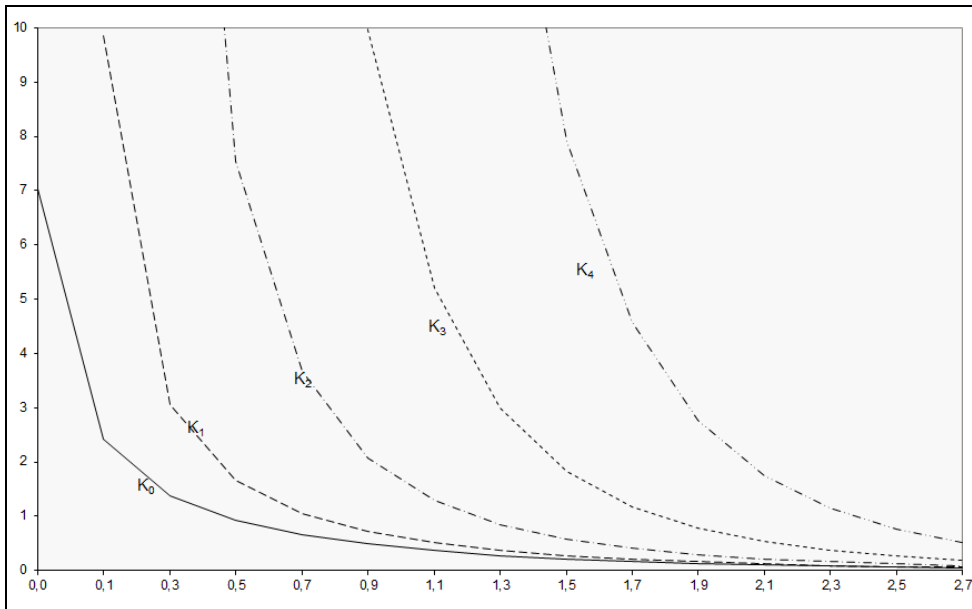


Abbildung 16.13: Grafische Darstellung der modifizierten Besselfunktion der ersten Art $K_n(x)$ mittels der BESSELK()-Funktion

Siehe auch BESSELI(), BESSELJ(), BESSELY()



Dieses Beispiel finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe `Bessel_K.xls/x` auf den Blättern `Bessel K` und `Diagramm`.

BESSELY() BESSELY()

Syntax BESSELY(*x*; *n*)

Definition Die Funktion gibt die Besselfunktion zweiter Art $Y_n(x)$ zurück.

Argumente *x* (erforderlich) ist der Wert, für den die Funktion ausgewertet werden soll; *x* muss eine reelle Zahl größer als 0 sein und kann Werte bis über $\pm 1,34 \cdot 10^8$ annehmen. (Der Wertebereich variiert etwas mit der Ordnung *n*.)

n (erforderlich) ist die Ordnung der Besselfunktion und muss positiv sein. Der Maximalwert von *n* hängt von *x* ab, deckt aber alle praxisrelevanten Bereiche ab. Wenn *n* keine ganze Zahl ist, werden deren Nachkommastellen von der Funktion für die Berechnung ignoriert, also nicht gerundet.

Hintergrund Die Besselfunktion zweiter Art $Y_n(x)$ der Ordnung *n*, die auch als »Webersche Funktion« oder »Neumannsche Funktion« bezeichnet wird, ist die Lösung der Besselschen Differenzialgleichung

$$x^2 y'' + xy' + (x^2 - n^2) \cdot y = 0$$

bzw.

$$x^2 \cdot \frac{d^2}{dx^2} y + x \cdot \frac{d}{dx} y + (x^2 - n^2) \cdot y = 0$$

Sie kann in Termen der Besselfunktion der ersten Art $J_n(x)$ ausgedrückt werden:

$$Y_n(x) = \lim_{p \rightarrow n} \frac{J_p(x) \cdot \cos p\pi - J_{-p}(x)}{\sin p\pi} \quad \text{mit } n = 0, 1, 2, \dots$$

Praxiseinsatz Wegen des sehr speziellen Charakters der Besselfunktionen wird auch hier, wie bei der BESSELI()-Funktion, nur auf deren grafische Darstellung eingegangen (siehe Seite 847).

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Bessel-Funktion 2. Art							Y _n (x)							
2															
3	Ordnung	x	0,25	0,50	0,75	1,00	1,25	1,50	1,75	2,00	2,25	2,50	2,75	3,00	3,25
4	0	Y _n (x)	-0,93	-0,44	-0,14	0,09	0,26	0,38	0,47	0,51	0,52	0,50	0,45	0,38	0,29
5	1				-1,04	-0,78	-0,58	-0,41	-0,25	-0,11	0,03	0,15	0,25	0,32	0,38
6	2							-0,93	-0,76	-0,62	-0,50	-0,38	-0,27	-0,16	-0,05
7	3										-0,91	-0,76	-0,64	-0,54	-0,45
8	4													-0,92	-0,77

Abbildung 16.14: Berechnung der BESSELY()-Funktion für die Ordnungen $n = 0 \dots 4$

Mit dieser Tabelle lassen sich die Funktionswerte der BESSELY()-Funktion für die Ordnungen $n = 0 \dots 4$ berechnen und grafisch darstellen. Bei den höheren Ordnungen werden die ersten Formeln weggelassen, weil ihre Funktionswerte für eine übersichtliche Grafik zu groß sind.

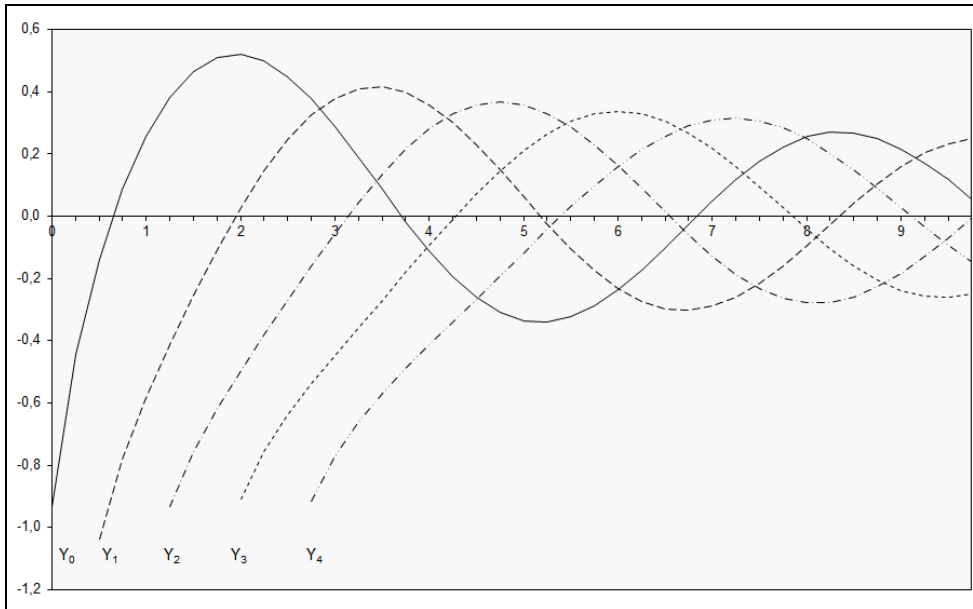


Abbildung 16.15: Grafische Darstellung der Besselfunktionen zweiter Art $Y_n(x)$

BESSELI(), BESSELJ(), BESSELK()

Dieses Beispiel und die Grafik finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe `Bessel_Y.xlsx` auf dem Arbeitsblatt `Bessel Y` bzw. `Diagramm`.

Siehe auch



GAUSSF.GENAU() bzw. GAUSSFEHLER() ERF.PRECISE() bzw. ERF()

GAUSSF.GENAU(*Untere_Grenze*)

GAUSSFEHLER(*Untere_Grenze*; *Obere_Grenze*)

Diese Funktionen geben die Werte des Wahrscheinlichkeitsintegrals (auch »Gaußsches Fehlerintegral«, »Fehler-Funktion« oder »Error-Funktion« genannt) zurück.

Für Excel 2010 wurde die Funktion GAUSSFEHLER() im Algorithmus verbessert. Zusätzlich bekam sie einen Zwilling, der dasselbe leistet, aber einen neuen, an Konventionen angepassten Namen erhielt: GAUSSF.GENAU(). Wer Abwärtskompatibilität benötigt, kann ohne Nachteil die »alte« Funktion verwenden – so wie die folgenden Erklärungen.

Untere_Grenze (erforderlich) ist die untere Grenze für die Integration in GAUSSF.GENAU() und GAUSSFEHLER().

Obere_Grenze (optional; nur bei GAUSSFEHLER()) ist die obere Grenze für die Integration in GAUSSFEHLER(). Wenn dieses Argument fehlt, integriert GAUSSFEHLER von 0 (Null) bis *Untere_Grenze*.

Syntax

Definition

Argumente

Hintergrund Die Gaußsche Fehlerfunktion $erf(z)$ erhält man durch Integration der Normalverteilung, der normalisierten Form der Gaußfunktion. Sie hat die Form:

$$erf(z) = \frac{2}{\sqrt{\pi}} \cdot \int_0^z e^{-t^2} dt$$

Bei der Excel-Funktion ist die untere Grenze zusätzlich variabel und lässt sich als erstes Argument eingeben, um alle Sonderfälle abdecken zu können. Mit den Integrationsgrenzen z_1 und z_2 lautet die Definition der »verallgemeinerten Fehlerfunktion« $erf(z_1, z_2)$:

$$erf(z_1, z_2) = \frac{2}{\sqrt{\pi}} \cdot \int_{z_1}^{z_2} e^{-t^2} dt$$

Praxiseinsatz =GAUSSF.GENAU(0,5) und =GAUSSFEHLER(0,5) ergeben jeweils 0,52049988
 =GAUSSFEHLER(0;0,5) ergibt 0,52049988
 =GAUSSFEHLER(0,5;1) ergibt 0,32220086

Als weitere Beispiele sehen Sie in der Abbildung 16.16 die Funktionswerte von GAUSSF.GENAU() mit dem Argument *Untere_Grenze* = 0...3 bzw. von GAUSSFEHLER() mit den Argumenten *Untere_Grenze* = 0 und *Obere_Grenze* = 0...3. Zum Vergleich sind daneben die Funktionswerte der Funktionen GAUSSFKOMPL.GENAU() bzw. GAUSSFKOMPL() im selben Intervall aufgeführt.

Aus dem dazugehörigen Diagramm (Abbildung 16.17) können Sie den Zusammenhang zwischen GAUSSF.GENAU() bzw. GAUSSFEHLER() und GAUSSFKOMPL.GENAU() bzw. GAUSSFKOMPL() grafisch dargestellt erkennen: Beide Funktionen summieren sich an jeder Stelle zum Wert 1.

	A	B	C	D
1	Gauss			
2				
3	x	GAUSSFEHLER	GAUSSFKOMPL	GAUSSFEHLER + GAUSSFKOMPL
4	0,00	0	1	1,00
5	0,25	0,27632639	0,72367361	1,00
6	0,50	0,52049988	0,47950012	1,00
7	0,75	0,71115563	0,28884437	1,00
8	1,00	0,84270079	0,15729921	1,00
9	1,25	0,92290013	0,07709987	1,00
10	1,50	0,96610515	0,03389485	1,00
11	1,75	0,98667167	0,01332833	1,00
12	2,00	0,99532227	0,00467773	1,00
13	2,25	0,99853728	0,00146272	1,00
14	2,50	0,99959305	0,00040695	1,00
15	2,75	0,99989938	0,00010062	1,00
16	3,00	0,99997791	0,00002209	1,00

Abbildung 16.16: In dieser Tabelle sehen Sie, wie die Ergebnisse der GAUSSF.GENAU()/GAUSSFEHLER()-Funktion (Spalte B) und der GAUSSFKOMPL.GENAU()/GAUSSFKOMPL()-Funktion (Spalte C) zusammen den Wert 1 (Spalte D) ergeben

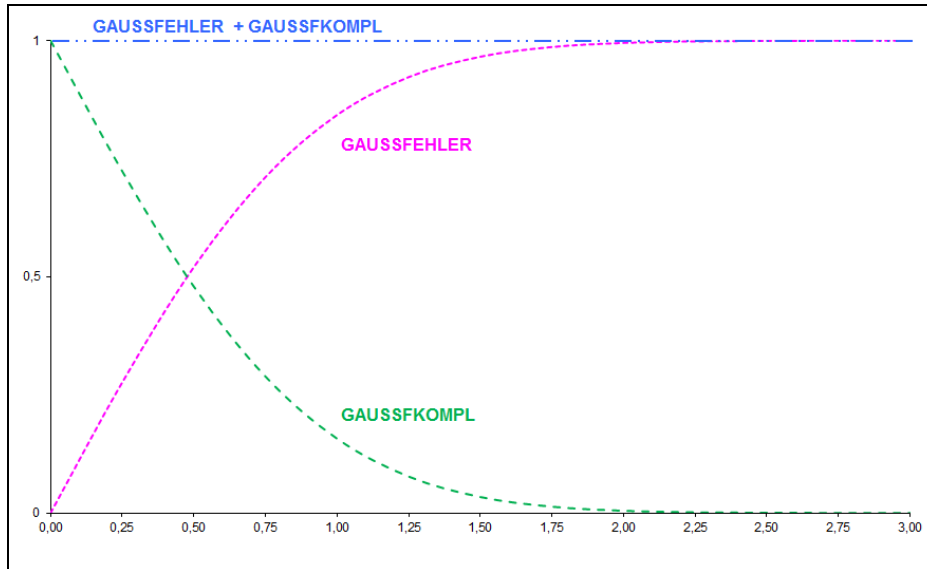


Abbildung 16.17: Dieses Diagramm führt Ihnen grafisch vor, wie sich die beiden Funktionen GAUSSF.GENAU()/GAUSSFEHLER() und GAUSSFKOMPL.GENAU()/GAUSSFKOMPL() ergänzen

GAUSSFKOMPL.GENAU(), GAUSSFKOMPL()

Das Beispiel und das Diagramm finden Sie im Ordner `\Ms5-235\Kap16` in der Arbeitsmappe *Gauss.xlsx* auf dem Arbeitsblatt *GaussDaten* bzw. *GAUSS*.

Siehe auch



GAUSSFKOMPL.GENAU() bzw. GAUSSFKOMPL()

 **ERFC.PRECISE() bzw. ERFC()**

GAUSSFKOMPL.GENAU(*Untere_Grenze*)

GAUSSFKOMPL(*Untere_Grenze*)

Diese Funktionen geben das Komplement zur Gaußschen Fehlerfunktion zurück.

Für Excel 2010 wurde die Funktion GAUSSFKOMPL() im Algorithmus verbessert. Zusätzlich bekam sie einen Zwilling, der dasselbe leistet, aber einen neuen, an Konventionen angepassten Namen erhielt: GAUSSFKOMPL.GENAU(). Wer Abwärtskompatibilität benötigt, kann ohne Nachteil die »alte« Funktion verwenden – so wie die folgenden Erklärungen.

Untere_Grenze (erforderlich) ist die untere Grenze für die Integration in GAUSSFKOMPL.GENAU() bzw. GAUSSFKOMPL(). Diese Funktionen kennen nur ein Argument.

Das Komplement $erfc(z)$ der Gaußschen Fehlerfunktion $erf(z)$ (siehe im Abschnitt »Hintergrund« bei der Funktion GAUSSFEHLER() auf Seite 854) ergibt sich aus folgendem Zusammenhang:

$$erfc(z) = 1 - erf(z) = \frac{2}{\sqrt{\pi}} \cdot \int_z^{\infty} e^{-t^2} dt$$

Syntax

Definition

Argumente

Hintergrund

Die obere Integrationsgrenze ist auf unendlich (∞) festgesetzt; $erfc(z)$ erfordert daher nur eine Variable z (das einzige Argument, das dann auch konsequenterweise *Untere_Grenze* genannt wird).

Praxiseinsatz Ein Beispiel finden Sie im Abschnitt »Praxiseinsatz« bei der Funktion GAUSSF.GENAU()/GAUSSFEHLER() auf Seite 854.

Siehe auch GAUSSF.GENAU(), GAUSSFEHLER()



Das Beispiel und das Diagramm finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe Gauss.xlsx auf dem Arbeitsblatt GaussDaten bzw. GAUSS.

Sprungfunktionen

Hier begegnen uns zwei mathematische Funktionen, die wie Schalter wirken. Anders als die uns vertrauten Funktionen, die – wie beispielsweise der Sinus – zu einem Ausgangswert uns einen von diesem Ausgangswert abhängigen Ergebniswert liefern und ohne Überraschungen auskommen⁴, kennen die beiden Sprungfunktionen nur zwei verschiedene Ergebniswerte: 0 und 1. Je nachdem, welchen Betrag der Ausgangswert besitzt, springen sie zwischen 0 und 1 hin und her. Dabei kennt die Funktion DELTA() nur einen Punkt, an dem sie sich für 1 entscheidet; ansonsten bleibt sie hartnäckig bei 0. Die Funktion GGANZZAHL() dagegen schaltet dauerhaft von 0 auf 1 um, wenn der Ausgangswert eine festzulegende Mindestgröße, eine Schwelle, erreicht und überschreitet.

Die Diracsche Deltafunktion $\delta(x)$ ist der mathematische Stammvater dieser Excel-Funktionen. Sie zählt wie die beiden Excel-Funktionen zu den Sprungfunktionen. Auch die Sprungfunktionen, wie sie in der Mathematik gelehrt werden, ändern an einer besonderen Stelle (oder auch mehreren) übergangslos, also sprunghaft, ihren Ergebniswert und sind an dieser Stelle also nicht stetig.

Der Definitionsbereich der Diracschen Deltafunktion reicht von $x = -|\infty$ bis $x = +|\infty$ und an der Stelle $x = 0$ wird sie unendlich (nicht 1!). Sie wird beispielsweise in der Mechanik und der Elektrizitätslehre zur Darstellung von Momentanimpulsen eingesetzt.

DELTA() DELTA()

Syntax DELTA(*Zahl1*; *Zahl2*)

Definition Die DELTA()-Funktion überprüft zwei Werte auf Gleichheit. Die Funktion gibt 1 zurück, wenn *Zahl1* und *Zahl2* exakt gleich sind; andernfalls gibt sie 0 (Null) zurück.

Argumente *Zahl1* (erforderlich) und *Zahl2* (optional) sind zwei beliebige reelle Zahlen. Wenn das Argument *Zahl2* fehlt, wird es als Null angenommen.

Hintergrund Wenn man bei dieser Funktion das zweite (optionale) Argument weglässt oder gleich Null setzt, hat man nahezu die direkte Entsprechung der Diracschen Deltafunktion: Für jeden Wert von *Zahl1* liefert die DELTA()-Funktion den Wert Null und für *Zahl1* gleich Null den Wert 1. Eigentlich müsste dieser letzte Wert nicht 1, sondern ∞ (unendlich) sein. Weil aber

⁴ Diese Eigenschaft wird in der Mathematik »stetig« genannt.

Rechnersysteme den Wert ∞ nicht darstellen können, haben die Programmierer dieser Funktion den sinnvollen Wert 1 als Ergebniswert festgelegt (siehe auch die Einleitung zum Abschnitt »Sprungfunktionen« auf Seite 856).

Sie leistet damit das Gleiche wie die WENN()-Funktion in der Schreibweise:

WENN(Zahl1=Zahl2;1;0)

Die Vorteile der DELTA()-Funktion gegenüber der WENN()-Funktion liegen in der einfacheren Syntax, der größeren Übersichtlichkeit und der höheren Ausführungsgeschwindigkeit.

Mit dieser Funktion können Sie eine Gruppe von Werten filtern. Beispielsweise können Sie durch Summieren über mehrere DELTA()-Funktionen berechnen, wie viele gleiche Zahlenpaare vorliegen.

=DELTA(A1;C3) ergibt 1,

wenn die Inhalte der Zellen A1 und C3 gleich sind, andernfalls 0.

Ein fertig verpacktes Sortiment soll aus fünf Teilen mit unterschiedlicher und zuweilen wechselnder Stückzahl bestehen. Dazu soll in übersichtlicher Form der Verpackungsfortschritt des Sortiments in einer Excel-Tabelle dargestellt werden. Die Übereinstimmung von Soll- und Ist-Wert lässt sich einfach mit der Delta-Funktion erledigen. In der Excel-Tabelle in Abbildung 16.18 erhält man in der Zelle D8 nur dann den Wert 5 (= der Anzahl der verschiedenen Teile im Sortiment), wenn alle Soll- und Ist-Werte identisch sind.

Eine ähnliche Lösung hätten Sie auch mit der Funktion GGANZZAHL() auf Seite 858 erreichen können. Der Nachteil wäre allerdings, dass eine Übererfüllung nicht festzustellen wäre, was die DELTA()-Funktion jedoch leisten kann. Wenn Sie in der in Abbildung 16.18 stehenden Tabelle beispielsweise in der Zelle C3 den Wert von 12 auf 13 ändern, wird auch das Ergebnis der DELTA()-Funktion in der Zelle D3 sogleich auf Null geändert.

	A	B	C	D
1	Sortiment A			
2		Sollwert	Istwert	Übereinstimmung
3	Teil 1	12	12	1
4	Teil 2	5	4	0
5	Teil 3	20	17	0
6	Teil 4	100	100	1
7	Teil 5	3	3	1
8				3

Abbildung 16.18: Die Übereinstimmung zweier Werte lässt sich mit der DELTA()-Funktion leicht ermitteln

Beim Betrachten dieses einfachen Beispiels könnte man auf die Idee kommen, den Einsatz der DELTA()-Funktion als überflüssig zu erachten, weil man beim Eingeben der Werte schon mit einem Blick erkennen kann, ob die Übereinstimmung erreicht ist. Dem ist entgegenzuhalten, dass in der beruflichen Praxis selten Dateneingabe und Auswertung örtlich und zeitlich zusammenfallen. So können die Werte in den Zellen von B3 bis C7 durchaus aus anderen Excel-Arbeitsmappen durch Verknüpfungen – häufig über ein Netzwerk – in diese Tabelle eingebunden werden. Derjenige, der die Daten eingibt, muss nicht zwangsläufig derjenige sein, der sie auch auswerten muss. Daher hebt die DELTA()-Funktion für den Nutzer der Tabelle das Entscheidende hervor.

Praxiseinsatz

Beispiel 1

Beispiel 2

Hinweis

Siehe auch GGANZZAHL() sowie aus der Kategorie »Textfunktionen« IDENTISCH()



Das Beispiel aus Abbildung 16.18 finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe *Delta.xlsx* auf dem Arbeitsblatt *Delta*.

GGANZZAHL() GESTEP()

Syntax GGANZZAHL(*Zahl*; *Schritt*)

Definition GGANZZAHL() gibt den Wert 1 zurück, wenn das Argument *Zahl* größer oder gleich dem Argument *Schritt* ist; andernfalls gibt sie 0 (Null) zurück.

Argumente *Zahl* (erforderlich) ist der Wert, der gegen *Schritt* geprüft werden soll.

Schritt (optional) steht für den Schwellenwert. Wenn Sie für *Schritt* keinen Wert angeben, arbeitet GGANZZAHL() mit 0 (Null).

Anders, als der Name vermuten lässt, dürfen beide Argumente auch Dezimalbrüche und müssen nicht ganze Zahlen sein.

Hintergrund Mit dieser Funktion können Sie eine Gruppe von Werten filtern. Beispielsweise können Sie durch Aufsummieren mehrerer GGANZZAHL()-Funktionen berechnen, wie viele Werte größer sind als ein Schwellenwert (siehe auch die Einleitung zum Abschnitt »Sprungfunktionen« auf Seite 856).

Sie leistet das Gleiche wie die WENN()-Funktion in der Schreibweise:

```
=WENN(Zahl>=Schritt;1;0)
```

Die Vorteile der GGANZZAHL()-Funktion gegenüber der WENN()-Funktion liegen in der einfacheren Syntax, der größeren Übersichtlichkeit und der höheren Ausführungsgeschwindigkeit.

Praxiseinsatz =GGANZZAHL(2;5) ergibt 0

Beispiel 1 =GGANZZAHL(5;5) ergibt 1

=GGANZZAHL(8;5) ergibt 1

Beispiel 2 Eine Schulklasse tritt bei einem Sportfest zum Weitsprung an. Der Lehrer will wissen, welche Sprünge ein gewisses Mindestmaß erreichen oder überschreiten. Im Bereich *E6:G16* stehen die Formeln, die mithilfe der GGANZZAHL()-Funktion die jeweilige Sprungweite im Bereich *B6:D16* mit dem vorgegebenen Mindestmaß von 3,80 m in Zelle *E5* (benannt als »minWeite«) vergleichen.

	A	B	C	D	E	F	G	H
1	Bundesjugendspiele 2010		Weitsprung					
2								
3	Klasse 9a	Weite			Mindest-		Von 3 Sprüngen über der Mindestweite	
4					weite			
5	Schüler/Schülerin	1. Sprung	2. Sprung	3. Sprung	3,80			
6	Arwenheim	3,78	3,95	3,80	0	1	1	2
7	Brabant	4,12	4,07	4,28	1	1	1	3
8	Cullman	4,09	3,75	3,79	1	0	0	1
9	Duisenberg	3,52	4,17	4,32	0	1	1	2
10	Ehrig	2,95	3,09	3,41	0	0	0	0
11	Flunkherr	4,89	4,88	4,87	1	1	1	3
12	Gieselmann	3,66	3,54	3,74	0	0	0	0
13	Hansen	3,43	3,39	3,41	0	0	0	0
14	Imain	4,52	4,31	4,78	1	1	1	3
15	Jolandarka	3,94	3,82	4,01	1	1	1	3
16	Kim	3,62	3,71	3,53	0	0	0	0
17					insgesamt:			17

Abbildung 16.19: Im Bereich E6:G16 zeigt uns die GGANZZAHL()-Funktion mit einer 1, welcher Sprung über der Mindestweite von 3,80 m (vorgegeben in Zelle E5) lag

DELTA() sowie aus der Kategorie »Textfunktionen« IDENTISCH()

Dieses Beispiel finden Sie im Ordner \Ms5-235\Kap16 in der Arbeitsmappe GGanzzahl.xlsx auf dem Arbeitsblatt Weitsprung.

Siehe auch



Kapitel 17

Webfunktionen

URLCODIEREN()	862
WEBDIENST()	863
XMLFILTERN()	865
Was ist XML?	867
Was sind Namespaces?	869
Was ist XPath?	871



In diesem Kapitel beschreiben wir Funktionen, die den Nutzern von Excel 2013 aufwärts vorbehalten sind, denn mit Excel 2013 kam eine neue Kategorie hinzu – die der Webfunktionen.

Microsoft Excel ist seit längerem in das SharePoint-System integriert. Mit den Versionen seit 2007 erfolgt die Integration immer tiefer. SharePoint verfügt inzwischen über die Excel-Services, die Arbeitsmappen im *.xlsx*-Format auch im Webbrowser anzeigen und bearbeiten lassen. Im SharePoint-System gibt es Listen und Daten, welche über Webdienste abgefragt bzw. gefüllt werden.

Hinweis Mehr zur Integration von Excel und Microsoft SharePoint finden Sie im Handbuch »Microsoft SharePoint 2010 für Anwender«, Microsoft Press, ISBN 978-3-86645-137-7.

Über die sogenannten Webdienste (Webservices) werden heute auch Daten im Web ausgetauscht. Viele Geschäftsprozesse kommunizieren über Webdienste, etwa bei der Übermittlung einer Kontoanfrage bei einer Bank und der Antwortliste durch den angefragten Server. Mit den neuen Webfunktionen kann nun auch Excel an diesem Datenverkehr teilnehmen.

Am Ende des Kapitels haben wir versucht, mit den Begriffsbestimmungen für die drei wichtigsten Begriffe in den folgenden Funktionsbeschreibungen einen Einstieg in das Thema »XML« zu geben.

Tabelle 17.1
Die neuen
Webfunktionen
in der Übersicht

Funktion	Beschreibung
URLCODIEREN()	Gibt eine URL-codierte Zeichenfolge zurück
WEBDIENST()	Empfängt (XML-) Daten vom angegebenen Webdienst
XMLFILTERN()	Extrahiert ausgewählte Daten aus einer XML-Datei

Neu in Excel
2013

URLCODIEREN()



ENCODEURL()

Syntax URLCODIEREN(*text*)

Definition Die Funktion gibt die im Argument *text* übergebene Zeichenfolge URL-codiert zurück.

Argumente *text* (erforderlich) ist die nach URL-Standard zu codierende Zeichenfolge.

Hintergrund Wenn Sie sich mit Themen wie »Webanalyse« oder »Daten aus dem Web« beschäftigen oder wenn Sie schon einmal eine Webadresse (Uniform Resource Locator, URL) mit Parametern aufgerufen haben, dann kennen Sie das Problem, dass eine URL und zusätzliche URL-Parameter nicht aus beliebigen Zeichen zusammengesetzt werden können. Man muss darauf achten, dass die Informationen in der URL auch immer URL-konform codiert sind, die URL also als gültiger »Quellenanzeiger« funktioniert.

Im Wesentlichen wandelt die Funktion URLCODIEREN() Sonderzeichen in gültige Zeichen um. So wird z.B. das Leerzeichen in die Zeichenfolge *%20* umgewandelt, weil das Leerzeichen als Ende der URL interpretiert wird. Zur Codierung werden nur bestimmte Zeichen des ASCII-Zeichensatzes verwendet.

Einzelheiten zum Standard, Aufbau und Verwendung einer URL finden Sie u.a. im Internet unter den Adressen http://de.wikipedia.org/wiki/Uniform_Resource_Locator und <http://tools.ietf.org/html/rfc1738>.

In einer zusammengesetzten URL soll ein Buchtitel als Parameter übergeben werden. Dazu muss der Titel URL-codiert vorliegen. Die Formel

=URLCODIEREN("Ötzi, der Mann im Eis")

ergibt die codierte Zeichenfolge %C3%96tzi%2C%20der%20Mann%20im%20Eis.

ASC(), CODE(), WEBDIENST(), XMLFILTERN()

Das gezeigte Beispiel und weitere finden Sie im Ordner \Ms5-235\Kap17 in der Arbeitsmappe *Webfunktionen.xlsx* auf dem Arbeitsblatt *URLCODIEREN*.

Hinweis

Praxiseinsatz

Siehe auch



WEBDIENST() WEBSERVICE()

WEBDIENST(*url*)

Die Funktion gibt die vom in *url* adressierten Webdienst gelieferten Daten im XML-Format aus.

url (erforderlich) ist die Adresse des Webdiensts, einschließlich notwendiger Parameter.

Die Übermittlung von Daten zwischen verschiedenen Anwendungen und Prozessen erfolgt in vielen Fällen über sogenannte Webdienste. Das Format, in dem die Daten übertragen werden, ist standardmäßig XML (siehe auch die Begriffsbestimmungen am Ende dieses Kapitels). Mit der Funktion WEBDIENST() können Sie aus einem Arbeitsblatt heraus Daten eines Webdiensts abfragen (GET-Funktion des Webdiensts) oder Daten an einen Webdienst schicken (PUT-Funktion des Webdiensts).

Bitte beachten Sie die folgenden Hinweise zum Einsatz der Funktion:

- ▶ Die Funktion unterstützt keine Authentifizierung, d.h. sie kann nur auf Webdienste zugreifen, die anonyme Abfragen unterstützen.
- ▶ Wenn die *url* mehr als die zulässigen 2.048 Zeichen für eine GET-Anforderung umfasst, gibt WEBDIENST() den Fehlerwert #WERT! zurück.
- ▶ Wenn die Daten eine Zeichenfolge enthalten, die ungültig ist oder mehr als die in Zellen zugelassenen 32.767 Zeichen enthält, gibt WEBDIENST() den Fehler #WERT! zurück.
- ▶ Wenn keine Daten zurückgegeben werden können, gibt WEBDIENST() den Fehlerwert #WERT! aus.
- ▶ Bei nicht unterstützten Webprotokollen, z.B. *ftp://* oder *file://*, gibt WEBDIENST() den Fehlerwert #WERT! zurück.

**Neu in Excel
2013**

Syntax

Definition

Argumente

Hintergrund

Praxeinsatz In einem Reiseantragsformular sollen aktuelle Wetterdaten für das Reiseziel angegeben werden. Dazu wird ein Webdienst für lokale Wetterinformationen angefragt. Die Formel

```
=WEBDIENST("http://weather.yahooapis.com/forecastrss?w=638242&u=c")
```

holt lokale Wetterinformationen für Berlin (w=638242) in Grad Celsius (u=c) vom Wetterservice des Internetdiensteanbieters Yahoo!. Die erhaltenen Wetterdaten liegen im XML-Format vor und die Details zu Temperatur, Wind usw. müssen aus der XML-Zeichenkette extrahiert werden (Abbildung 17.1).

```

F3
:
x ✓ f_x
=WEBDIENST("http://weather.yahooapis.com/forecastrss?w=638242&u=c")

E F
2 WEBDIENST()
3 <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
4 <rss version="2.0" xmlns:yweather="http://xml.weather.yahoo.com/ns/rss/1.0"
5 xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
6 <channel>
7 <title>Yahoo! Weather - Berlin, DE</title>
8 <link>http://us.rd.yahoo.com/dailynews/rss/weather/Berlin__DE/*http://weather.yahoo.com/forecast
9 /GMXX1273_c.html</link>
10 <description>Yahoo! Weather for Berlin, DE</description>
11 <language>en-us</language>
12 <lastBuildDate>Wed, 30 Jan 2013 9:20 pm CET</lastBuildDate>
13 <ttl>60</ttl>
14 <yweather:location city="Berlin" region="BE" country="Germany"/>
15 <yweather:units temperature="C" distance="km" pressure="mb" speed="km/h"/>
16 <yweather:wind chill="3" direction="260" speed="27.36" />
17 <yweather:atmosphere humidity="87" visibility="9.99" pressure="982.05" rising="0" />
18 <yweather:astronomy sunrise="7:52 am" sunset="4:47 pm"/>
19 <image>
20 <title>Yahoo! Weather</title>
21 <width>142</width>
22 <height>18</height>
23 <link>http://weather.yahoo.com</link>
24 <url>http://l.yimg.com/a/i/brand/purplelogo//uh/us/news-wea.gif</url>
25 </image>
26 <item>
27 <title>Conditions for Berlin, DE at 9:20 pm CET</title>
28 <geo:lat>52.52</geo:lat>
29 <geo:long>13.38</geo:long>
30 <link>http://us.rd.yahoo.com/dailynews/rss/weather/Berlin__DE/*http://weather.yahoo.com/forecast
31 /GMXX1273_c.html</link>
32 <pubDate>Wed, 30 Jan 2013 9:20 pm CET</pubDate>
33 <yweather:condition text="Partly Cloudy" code="29" temp="7" date="Wed, 30 Jan 2013 9:20 pm
  
```

Abbildung 17.1: Das Ergebnis der Abfrage mit WEBDIENST() liegt im XML-Format vor

Hinweis Die Einzelheiten, welche Daten der Webdienst liefert, ob und welche Parameter dieser hat, finden Sie immer in der Beschreibung des Webdiensts. In unserem Fall sind die Details unter der Internetadresse <http://developer.yahoo.com/weather> nachlesbar.

URLCODIEREN(), XMLFILTERN()

Die gezeigten Beispiele finden Sie im Ordner `\Ms5-235\Kap17` in der Arbeitsmappe *Webfunktionen.xlsx* auf dem Arbeitsblatt *Wetter*. Dazu gehört das Arbeitsblatt *Wetterhifstabelle*. Weitere Beispiele zu einem Webdienst von Wikipedia und zwei RSS-Feeds finden Sie auf entsprechend benannten Arbeitsblättern.

Siehe auch



XMLFILTERN()



FILTERXML()

XMLFILTERN(*xml*;*xpath*)

Die Funktion extrahiert die mit *xpath* adressierten Daten aus der *xml*-Zeichenfolge.

xml (erforderlich) ist eine Zeichenfolge im XML-Format.

xpath (erforderlich) ist der Pfad auf ein XML-Knotenelement bzw. dessen Attribut.

Die Funktion WEBDIENST() gibt die Antwort eines Webservice zurück, die in jedem Fall im XML-Format vorliegt. Der Zugriff auf ausgewählte (gefilterte) Informationen erfolgt über eine Abfrage mit XPath, das, wie der Name schon sagt, die Daten im XML adressiert. Lesen Sie hierzu bitte auch die Begriffsbestimmungen am Ende dieses Kapitels.

Beachten Sie auch diese Hinweise beim Einsatz der Funktion:

- ▶ Enthält das Argument *xml* kein gültiges XML, gibt XMLFILTERN() den Fehlerwert #WERT! zurück.
- ▶ Enthält das Argument *xml* einen Namespace mit einem ungültigen Präfix, gibt XMLFILTERN() den Fehlerwert #WERT! zurück.
- ▶ Enthält das Argument *xpath* einen ungültigen Ausdruck, gibt die Funktion ebenfalls #WERT! zurück.

In einem Reiseantragsformular sollen aktuelle Wetterdaten für das Reiseziel angegeben werden. Dazu wird ein Webdienst für lokale Wetterinformationen angefragt. Die Formel

```
=WEBDIENST("http://weather.yahooapis.com/forecastrss?w=638242&u=c")
```

holt lokale Wetterinformationen für Berlin (*w=638242*) in Grad Celsius (*u=c*) vom Wetterservice des Internetdiensteanbieters Yahoo!. Die erhaltenen Wetterdaten liegen im XML-Format vor und die Details zu Temperatur, Wind usw. müssen aus der XML-Zeichenkette extrahiert werden (Abbildung 17.1).

Dafür wird die Funktion XMLFILTERN() genutzt:

```
=XMLFILTERN($F3;"//yweather:astronomy/@sunrise")
```

ergibt für die in der Zelle *F3* befindliche WEBDIENST()-Formel mit den Wetterdaten für Berlin die Uhrzeit für den Sonnenaufgang.

```
=XMLFILTERN($F3;"//lastBuildDate")
```

ergibt das Datum und die Uhrzeit der letzten Aktualisierung für die Berliner Wetterdaten.

Neu in Excel 2013

Syntax

Definition

Argumente

Hintergrund

Praxiseinsatz

	A	B	C	D
1	Wetterabfrage			
2	Wetterdaten von Yahoo.com	Ort:	Berlin	
3		Temperatur in:	°C	
4	URL			
5	http://weather.yahooapis.com/forecstrss			
6				
7	XMLFILTERN()			
8	Dienst:	Yahoo! Weather for Berlin, DE		
9	Datum/Zeit:	Wed, 30 Jan 2013 9:20 pm CET		
10	Aktuelle Temperatur:	7°C	Nachts teilweise bewölkt	
11	Gefühlte Temperatur:	3°C		
12	Windstärke:	27,36 km/h	(7,6 m/sec)	
13	Wind aus Richtung:	Westesüdwest		
14	Luftdruck:	982,05 mb		
15	Sonnenaufgang:	07:52		
16	Sonnenuntergang:	16:47		

Abbildung 17.2: Die Detaildaten aus XML werden mit XMLFILTERN() anhand des XPATH-Ausdrucks ermittelt

Achtung **Vorsicht: Falle!**

Beim Ausprobieren der Funktion XMLFILTERN() sind wir auf eine Besonderheit im deutsch lokalisierten Excel 2013 gestoßen, an die die Excel-Programmierer bei Microsoft USA wohl nicht gedacht haben:

Die Abfrage der Windgeschwindigkeit in unserem Wetterbeispiel ergibt im Normalfall einen numerischen Wert im Textformat. Das Dezimaltrennzeichen ist bei Yahoo! typischerweise der Punkt, was kein Problem darstellt, da wir mit der Funktion ZAHLENWERT() bei Bedarf leicht in einen numerischen Wert unter Angabe des verwendeten Dezimaltrennzeichens umwandeln können.

Die Auflösung der Formel =XMLFILTERN(\$F\$3;"//yweather:wind/@speed") gibt z.B. die Windgeschwindigkeit "35.4" zurück (Abbildung 17.3):

	A	B	C	D	E	F
11	Gefühlte Temperatur:	1°C				<lastBuildDate>Thu, 31 Jan 2013 9:00 am CET</lastBuildDate>
12	Windstärke:	35,4 km/h	(9,8 m/sec)	35,4		<ttl>60</ttl>
13	Wind aus Richtung:	Westesüdwest		km/h		<yweather:location city="Kiel" region="SH" country="Germany"/>
14	Luftdruck:	1003,1 mb		1003,1		<yweather:units temperature="C" distance="km" pressure="mb" s
15	Sonnenaufgang:	08:07		mb		<yweather:wind chill="1" direction="260" speed="35.4" />
16	Sonnenuntergang:	16:56				<yweather:atmosphere humidity="81" visibility="15" pressure="1
						<yweather:astronomy sunrise="8:07 am" sunset="4:56 pm"/>

Abbildung 17.3: Die Formel XMLFILTERN() extrahiert die korrekte Windgeschwindigkeit aus dem XML

Dieselbe Formel gibt aber für einen anderen Ort keinen Text, sondern eine große Zahl zurück (Abbildung 17.4).

	A	B	C	D	E	F
11	Gefühlte Temperatur:	-3°C				<lastBuildDate>Thu, 31 Jan 2013 10:19 am EET</lastBuildDate>
12	Windstärke:	16,9 km/h	(4,7 m/sec)	41533		<ttl>60</ttl>
13	Wind aus Richtung:	Süd		km/h		<yweather:location city="Helsinki" region="" country="Finland"/>
14	Luftdruck:	982,05 mb		982,05		<yweather:units temperature="C" distance="km" pressure="mb" s
15	Sonnenaufgang:	08:37		mb		<yweather:wind chill="-3" direction="180" speed="16.09" />
16	Sonnenuntergang:	16:28				<yweather:atmosphere humidity="100" visibility="9.99" pressure
						<yweather:astronomy sunrise="8:37 am" sunset="4:28 pm"/>

Abbildung 17.4: Die Formel XMLFILTERN() extrahiert einen Wert, der keine Windgeschwindigkeit sein kann

Des Rätsels Lösung: Noch bevor Sie eingreifen können, hat Excel den XML-Inhalt "16.09" als Datum interpretiert, hier als 16.09.2013, unformatiert als Datumszahl 41533. Wenn die Regel lauten würde, dass in diesen Sonderfällen immer das Schema TAG.MONAT greifen würde, wäre es noch einfach, den Fehler abzufangen. Leider ist die Vielfalt der möglichen Fehler etwas größer: In anderen Fällen wird eine Angabe wie "12.87" als das Datum 01.12.1987, also 32112 ausgegeben. Hier haben Sie es also mit dem Interpretationsschema MONAT.JAHR zu tun!

Wie stellen Sie den Fehlerfall fest und vor allem: wie wandeln Sie die Zahl 41533 in den Wert 16,09 und die Zahl 32112 in den Wert 12,87 um? In unserer Beispielmappe haben wir versucht, dies mit einem verschachtelten WENN() zu lösen. Herausgekommen ist eine etwas komplexere Formel. Und das alles, weil der Datentyp der Rückgabe von XMLFILTERN() bei der Abfrage aus einer XML-**Zeichenfolge** nicht zwingend eine **Zeichenfolge** ist – zumindest im deutschen Excel 2013.

URLCODIEREN(), WEBDIENST()

Die gezeigten Beispiele und weitere finden Sie im Ordner \Ms5-235\Kap17 in der Arbeitsmappe *Webfunktionen.xlsx* auf allen Arbeitsblättern.

Für Interessenten versuchen wir zum Schluss, in den folgenden drei Abschnitten die Begriffe *XML*, *Namespace* und *XPath* zu erklären, um Ihnen einen Einstieg in das Thema, welches unmittelbar mit den in diesem Kapitel behandelten Funktionen zusammenhängt, zu geben.

Siehe auch



Was ist XML?

Vor ca. 40 Jahren wurden verschiedene Weiterentwicklungen der bis dahin für die Speicherung und Verarbeitung von Daten entwickelten generischen Auszeichnungssprachen betrieben, unter anderem auch bei IBM. Dort stellte man 1969 unter dem Namen »Generalized Markup Language« einen Sprachvorschlag zusammen, der in der Folgezeit auch kommerziell vermarktet wurde. Aus den GML-Aktivitäten bei IBM entwickelte sich die internationale Standardisierungsbewegung und mit ihr der Standard GML (SGML). Mit SGML wurde eine Sprache festgelegt, welche die Definition eigener Sprachen erlaubt; daher auch der Begriff »Metasprache«. SGML bietet keinen feststehenden problemspezifischen Sprachumfang an, sondern eine Anzahl unterschiedlicher struktureller Konstrukte zur Formulierung von Dokumentgrammatiken.

Mittels SGML definierte Tim Berners-Lee Mitte der 80er Jahre eine eigene Sprache zur vereinfachten Formulierung von Dokumenten, die er HyperText Markup Language (HTML) nannte. Hauptbeweggrund war der Versuch, den Dokumentenaustausch am Europäischen Kernforschungszentrum CERN rechnergestützt zu vereinfachen. Zur technischen Realisierung der Verknüpfung zwischen den Dokumenten mittels Anker und Links definiert er den »Uniform Resource Locator« (URL) – eine global eindeutige Adresse für beliebige Inhalte. Seine Aktivitäten sollten die Keimzelle des World Wide Web bilden. Das Kürzel »WWW« bildet heute den Beginn vieler Webadressen.

HTML-Dokumente sind alle durch einen ähnlichen Aufbau gekennzeichnet. Alle HTML-Dokumente gehören demselben Typ an, nämlich dem Dokumenttyp HTML. Letztlich ist das auch die Grundidee von XML: Man sorgt dafür, dass alle Dokumente in ihrem Aufbau definierten Grundmustern und -strukturen folgen. Mit Dokumenten, die diesen Regeln folgen, lässt sich mehr anfangen, als wenn jedes Dokument eigenen Regeln folgt: Es wird möglich, Programme zu schreiben, die diese Dokumente automatisch verarbeiten.

Der letztendlich vom W3C (World Wide Web Consortium) verabschiedete Vorschlag zur Extensible Markup Language (XML) bildet konzeptionell eine Untermenge der Sprachmöglichkeiten von SGML. Konsequenterweise ist jedes XML-Dokument auch ein gültiges SGML-Dokument. Die Abweichung zu SGML wird besonders aus den Entwicklungszielen für XML deutlich:

1. Einfache Nutzung im Internet

In Abkehr von der Hauptnutzung von SGML als Offlinedokumentationsformat ist XML für die primäre Nutzung im Internet vorgesehen.

2. Unterstützung eines breiten Anwendungsspektrums

Auch hier soll XML das Einsatzspektrum über die Hauptnutzung von SGML als Format der technischen Dokumentation hinaus erweitern.

3. SGML-Kompatibilität

XML bildet eine Untermenge des ISO-Standards SGML. Dadurch kann jedes XML-Dokument auch als gültiges SGML-Dokument interpretiert und durch die entsprechenden SGML-Werkzeuge verarbeitet werden.

4. Einfache Applikationsentwicklung

Ausdrücklich erklärtes Ziel ist eine (im Hinblick auf SGML) deutlich vereinfachte Entwicklung von XML-verarbeitenden Applikationen.

5. Minimierung optionaler Sprachmerkmale (idealerweise gleich Null)

Auch dies zielt auf eine vereinfachte Applikationsentwicklung sowie eine einfachere Benutzbarkeit durch den Anwender, indem die Komplexität der Grundstrukturen reduziert wird.

6. Lesbarkeit

Das entstehende Textformat soll für Menschen und Maschinen gleichermaßen lesbar und verständlich sein.

7. Kompakte Spezifikation

Die XML-Spezifikation soll deutlich weniger Umfang aufweisen als der SGML-Standard. Letztlich konnte die Seitenzahl von über 600 Seiten für die SGML-Spezifikation auf ca. 30 Seiten für XML reduziert werden.

8. Formaler und präziser Sprachentwurf

Um die schnelle Akzeptanz seitens der Anwender zu erreichen, erachteten die Mitglieder der XML-Arbeitsgruppe beim W3C die schnelle Verfügbarkeit von XML-Werkzeugen als essentiell. Aus diesem Grunde sollte der XML-Sprachentwurf möglichst leicht und eindeutig in XML-Werkzeuge zu implementieren sein.

9. Leichte Dokumenterstellung

Die Erstellung von korrekten XML-Dokumenten sollte idealerweise so einfach sein, dass hierfür keine speziellen Werkzeuge benötigt werden.

10. Nicht notwendigerweise knappes Markup

Kompaktheit und Effizienz hinsichtlich des Volumens eines XML-Dokuments war zu keinem Zeitpunkt eines der Hauptentwicklungsziele.

XML bietet somit eine Möglichkeit zum Beschreiben von strukturierten Daten. Im Gegensatz zu HTML-Tags, die in erster Linie zum Steuern der Anzeige und des Erscheinungsbilds von Daten dienen, werden mit XML-Tags die Struktur und die Datentypen der Daten selbst definiert. Es ist also eine Markierungssprache, die ein Format für die Beschreibung von Daten bereitstellt. Damit ermöglicht sie plattformübergreifend eine genaue Inhaltsdeklaration und sinnvolle Suchergebnisse. Außerdem lässt XML die Trennung der Darstellung von den Daten zu. In HTML wird der Browser mithilfe von Tags angewiesen, Daten fett oder kursiv anzuzeigen. In XML werden Tags »nur« zum Beschreiben von Daten verwendet, etwa Ort, Straße und PLZ. In XML werden Daten in einem Browser über XSL (Extensible Stylesheet Language) und CSS (Cascading Style Sheets) dargestellt. XML trennt die Daten von der Darstellung und dem Prozess, um durch Zuweisen unterschiedlicher Stylesheets und Anwendungen die Daten nach Belieben anzeigen und verarbeiten zu können.

XML verwendet eine Reihe von Tags zum Entwerfen von Datenelementen. Jedes Element kapselt einen Datenabschnitt, der entweder sehr einfach oder sehr komplex sein kann. Sie können eine unbegrenzt große Gruppe von XML-Tags definieren. So lassen sich beispielsweise XML-Tags zum Deklarieren von Daten aus einer Rechnung definieren, z.B. des Preises, der Mehrwertsteuer, der Rechnungsadresse usw. Da XML-Tags organisationsweit und über Organisationsgrenzen hinaus angewendet werden, können Daten aus allen möglichen Datenspeichern leichter ausgetauscht und bearbeitet werden.

XML ist einfach, plattformunabhängig und ein inzwischen weit verbreiteter Standard. Der Vorzug von XML gegenüber HTML besteht in der schon erwähnten Trennung der Benutzeroberfläche von den (strukturierten) Daten. Aufgrund dieser Trennung der Daten von der Darstellung können Daten aus verschiedenen Quellen integriert werden. Kundeninformationen, Bestellscheine, Vertriebszahlen, Zeiterfassungsdaten, Katalogdaten und andere Informationen können in XML konvertiert werden.

Mit XQUERY und XPATH verfügt XML über eine Abfragesprache. Lesen Sie mehr dazu im übernächsten Kasten.

Die W3C-Veröffentlichungen zum Thema XML finden Sie im Internet unter der Adresse <http://www.w3.org/TR/2002/WD-xag-20021003> und zum Thema XPATH unter <http://www.w3.org/TR/2011/REC-xpath-full-text-10-20110317>.

Was sind Namespaces?

Die breite Entwicklung immer neuer XML-Sprachen führt zwangsläufig zu Mehrfachentwicklungen für ähnliche oder identische Problemstellungen. Bei natürlich sprachlicher Benennung der Elemente kann es schnell zur Verwendung identischer Bezeichner in verschiedenen XML-Sprachen kommen. In der Folge der Verfügbarkeit verschiedenster XML-Sprachen für beliebige Anwendungsbereiche entsteht die Anforderung, existierende Sprachfragmente in eigene Sprachen zu integrieren, um so zeitraubenden und vielfach fehleranfälligen Mehrfachentwicklungen vorzubeugen. Jedoch stößt man bei diesen Integrationsszenarien auf die unter Umständen nicht mehr gegebene Elementeindeutigkeit aufgrund der Namensgebung.

Denken Sie z.B. daran, dass Ihnen zwei XML-Dokumente vorliegen. In beiden geht es um Kundendaten. Dabei wird das Element »name« in der ersten Datei so verwendet:

```
<name>Sara</name>
```

In der zweiten Datei dagegen steht:

```
<name>Sara Unverhau</name>
```

Auf den ersten Blick scheint das nicht kritisch zu sein. Dies wird es aber, wenn die Daten zusammengeführt werden sollen. Während in der zweiten Datei auch der Nachname im Element enthalten ist, gibt es in der ersten Datei mit Sicherheit ein eigenes Element dafür. Richtig kompliziert wird es, wenn jetzt noch ein drittes XML-Dokument hinzukommt, in dem das Element »name« für eine Produktbezeichnung verwendet wird:

```
<name>Handbuch</name>
```

Solange die betreffenden Dokumente in unterschiedlichen Anwendungswelten, d.h. Unternehmen o.Ä., verwendet werden, ist das nicht problematisch. Beim Zusammenführen von Dokumenten folgt das entstehende Zieldokument nicht mehr den Strukturierungsregeln eines der Ausgangsdokumente. Es entsteht eine neue Dokumentstruktur, deren Regeln nicht explizit dokumentiert sind.

Eine noch größere Herausforderung stellt die Zusammenfassung und Veröffentlichung von XML-Strukturen in sogenannten Schemabibliotheken oder Datenbanken dar. Hier werden zwar die Dokumente nicht vereinigt, jedoch offenbart sich die gleiche Anwendungsdomäne (z.B. Rechnungsverwaltung, Stücklisten oder Produktstrukturen) als problematisch, da sie die XML-Strukturen in direkte Konkurrenz treten lässt. In Zeiten immer stärker werdenden Flexibilisierungsdrucks erweist sich dies im Hinblick auf eine angestrebte Standardisierung als kontraproduktiv.

Zusammenfassend lassen sich die Gründe für die Einführung bzw. Benutzung von Namensräumen wie folgt darstellen:

- ▶ Wiederverwendung externer XML-Strukturen in eigenen Dokumenten
- ▶ Verringerung des Designaufwands und Nutzung bereits gesammelter Designerfahrung
- ▶ Zusammenführung verschiedener XML-codierter Inhalte

XML-Namensräume stellen eine Syntax zur Verfügung, um Element- und Attributnamen eindeutig zu qualifizieren und so Bedeutungsüberschneidungen durch gleichbenannte Elemente oder Attribute in zu unterscheidenden Anwendungen auszuschließen. Sie bilden damit die notwendige Voraussetzung zur freien dezentralen Entwicklung eigener Vokabulare, ohne die Möglichkeit einer späteren Zusammenführung zu verlieren.

Die Anfang 1999 veröffentlichte Spezifikation des W3C zu den XML-Namespaces können Sie unter dieser Adresse im Internet nachlesen: <http://www.w3c.org/TR/REC-xml-names>

Konzept der Namensräume

Wie schafft man es, eindeutige Bezeichner zu verwenden? Jegliche Namenskoordination würde eine zentrale Vergabestelle zur Registrierung der vergebenen Namen, die über die Eindeutigkeit wacht und Mehrfachnutzungen unterbindet, bedingen. Die Einführung einer solchen Stelle hätte jedoch einen unüberschaubaren Verwaltungsaufwand bedeutet, den das

W3C nicht zu leisten im Stande wäre. Man nehme nur das Vergabeverfahren von Einträgen des Internet Domain Name Systems (DNS), welches bereits dezentral durch die einzelnen nationalen Domänenregistrierungsstellen gehandhabt wird. Der dort anzutreffende Aufwand hätte sich für XML-Namensräume potenziert, legt man pro Domainadresse mehrere Namensräume zugrunde.

Dem Anspruch eines dezentralen, aber dennoch die Eindeutigkeit garantierenden Ansatzes genügt das Namensschema der Uniform Resource Identification (URI). Es kombiniert zentrale und dezentrale Elemente in der Handhabung und ermöglicht so, trotz Existenz und Pflege einer zentralen Registratur, die geforderte Flexibilität in der Anwendung. Der bekannteste Einsatz von URI-Namen ist der im World Wide Web allgegenwärtige Uniform Resource Locator (URL).

Die zentrale Komponente findet sich im Domänennamen verwirklicht. Er ist entweder durch die IP-Adresse (Beispiel: *212.112.255.51*) oder deren literaler Entsprechung (Beispiel: *www.mindbusiness.de*) gegeben. Unterhalb der Domänenebene kann durch deren Verwalter eine beliebige Strukturierung vorgenommen werden. Genau nach diesem Schema arbeiten Sie, bzw. in unserem Fall InfoPath, auch beim Erstellen von XML-Dokumenten.

Wie auch bei URLs ist das Schema (z.B. *http*) zwingend mit anzugeben. Trotz der Möglichkeit, XML-Namensräume durch URLs zu identifizieren, handelt es sich dabei nicht um die Bezeichnung einer Internetquelle. Die verwendete Zeichenkette dient ausschließlich der Benennung der im Namensraum versammelten XML-Elemente und -Attribute.

Was ist XPath?

XPath stellt den Kern einer Anfragesprache, zur Extraktion verschiedenster Informationen (Knotenmengen, Positionsangaben, Werte usw.) aus einem XML-Dokument, dar. Aufgrund des dadurch eröffneten Anwendungsgebiets finden sich Teile von XPath in der Transformationsprache XSLT wieder. Hier dienen sie zum Auffinden der zu transformierenden Dokumentstrukturen. Das haben Sie im vorangegangenen Abschnitt gesehen. Ohne dass es Ihnen vielleicht aufgefallen ist, haben Sie dort bereits einfache XPath-Konstrukte kennengelernt.

Folgende Punkte charakterisieren kurz die Sprache XPath:

- ▶ XPath ist eine Sprache, um Teile eines XML-Dokuments zu adressieren. Sie wird z.B. in XSLT, XPointer und XQuery verwendet.
- ▶ XPath kann Dokumentknoten unter Angabe verschiedener Kriterien selektieren und grundlegende Manipulationen an Zeichenketten, booleschen Werten und Knotenmengen durchführen.
- ▶ XPath unterstützt Mustererkennung auf Dokumentknoten (Testen gegen ein vorgegebenes Muster).
- ▶ XPath enthält eine einfache Funktionsbibliothek, welche durch benutzerdefinierte Funktionen erweitert werden kann.
- ▶ XPath hat eine kompakte, nicht an XML orientierte Syntax, ähnlich den Pfadangaben in Betriebssystemen.

Lokalisierungspfade

Wie bei Pfadangaben im Betriebssystem, zum Beispiel:

`C:\MSPress\Excel\Ms5-235\Kap17\Webfunktionen.xlsx`

greifen Sie mit dem sogenannten Lokalisierungspfad auf einen bestimmten Knoten zu, zum Beispiel: `/Reisekosten/Daten/Kostenelemente/Anzahl`

Der Unterschied zum Betriebssystempfad besteht darin, dass XPath durchaus mehrere gefundene XML-Elemente zurückgeben kann, während der obere Pfad immer nur zu einem Element führt – einem Ordner oder Datei.

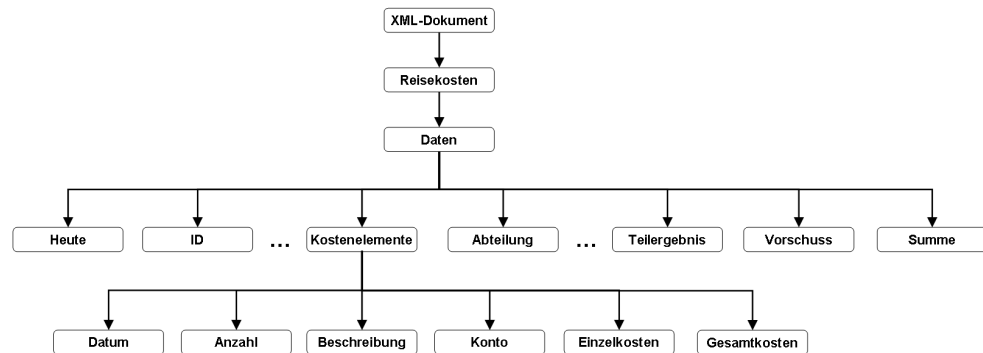


Abbildung 17.5: So sieht XPath ein XML-Dokument mit Reisekostenabrechnungsdaten

Bei der Bildung des Pfads hat das Tag `<xsl:template match="/">` eine besondere Bedeutung. Die Anweisung `match="/"` spricht das XML-Dokument an (nicht das Wurzelement *Reisekosten!*). Anders gesagt: Die Pfade in einem XSL-Dokument zur Ausgabe der Daten könnten durch die Anweisung `match="/Reisekosten/Daten"` verkürzt werden. Eine andere Variante könnte darin bestehen, die Adressen mit doppelten Schrägstrichen einzuleiten (`select="//Abteilung"`). Sie werden dadurch unabhängig von ihrer Ebene gefunden; in Dokumenten mit eindeutigen Elementnamen also eine Möglichkeit.

Der Weg über die Achsen

In der Natur der Baumstruktur, die XML-Dokumenten eigen ist, liegt die Tatsache, dass die Knoten verschiedene (familiäre) Beziehungen untereinander haben. So gibt es Knoten, die von anderen Knoten abhängen, und solche, die auf einer gemeinsamen Ebene liegen. Dafür hält XPath verschiedene Begriffe bereit, die aus der Verwandtschaftsterminologie stammen und als Achsen bezeichnet werden. XPath kennt insgesamt 13 Achsen, mit denen Knoten, d.h. Elemente und Attribute, abgefragt werden können.



Im Ordner `\Ms5.235\Kap17` der Beispieldateien zum Buch haben wir Ihnen eine PowerPoint-Folienserie mit Illustrationen zu den im Folgenden aufgeführten Navigationsachsen vorbereitet. Die Datei heißt `XPath.pptx`. Sie hilft Ihnen sicherlich, die in XPath verwendbaren Verwandtschaftsbeziehungen zu verstehen.

Achse	Erläuterungen	select="[Beispiel]"
<i>child</i>	Kinder: Alle unmittelbar untergeordneten Knoten (eine Ebene)	<code>child::*</code>
<i>parent</i>	Eltern: Alle unmittelbar übergeordneten Knoten (eine Ebene)	<code>parent::*</code>
<i>descendant</i>	Nachkömmlinge: Alle untergeordneten Knoten, d.h. auch solche, die mehr als eine Ebene darunter liegen	<code>descendant::*</code>
<i>ancestor</i>	Vorfahren: Alle übergeordnete Knoten, d.h. auch solche, die mehr als eine Ebene darüber liegen	<code>ancestor::*</code>
<i>following</i>	Nachfolgende Knoten: Knoten, die nach dem aktuellen Knoten im restlichen XML-Dokument folgen, unabhängig von der Hierarchie der Knoten	<code>following::*</code>
<i>preceding</i>	Vorherige Knoten: Knoten, die vor dem aktuellen Knoten im XML-Dokument stehen, unabhängig von der Hierarchie der Knoten	<code>preceding::*</code>
<i>following-sibling</i>	Nachfolgende Geschwisterknoten: Knoten, die nach dem aktuellen Knoten im restlichen XML-Dokument auf der gleichen Hierarchie-Ebene folgen	<code>following-sibling::*</code>
<i>preceding-sibling</i>	Vorherige Geschwisterknoten: Knoten, die vor dem aktuellen Knoten im XML-Dokument auf der gleichen Hierarchie-Ebene stehen	<code>preceding-sibling::*</code>
<i>attribute</i>	Attribut: Ermittelt Attributknoten des aktuellen Elements. Attribute erhalten eine Sonderstellung und werden nicht in das Schema der Begriffe <i>child</i> , <i>parent</i> , <i>descendant</i> und <i>ancestor</i> eingefügt.	<code>attribute::*</code> oder <code>@*</code>
<i>namespace</i>	Namensraum: Zugehöriger Namensraumknoten des aktuellen Elements	<code>namespace::*</code>
<i>self</i>	Der aktuelle Knoten selbst	<code>self::*</code>
<i>descending-or-self</i>	Nachkömmling oder der aktuelle Knoten selbst	<code>descending-or-self::*</code>
<i>ancestor-or-self</i>	Vorfahre oder der aktuelle Knoten selbst	<code>ancestor-or-self::*</code>

Tabelle 17.2
Die Verwandtschaftsbeziehungen in XPath

Dieser kurze Einblick in XPath sollte Ihnen die Idee und ein gewisses Verständnis für die bei der Transformation verwendeten Ausdrücke geben. Zumindest die Entwickler unter Ihnen werden damit in Zukunft mehr und mehr zu tun haben.

Die komplette Dokumentation zur Sprache XPath können Sie beim W3C im Internet unter der Adresse <http://www.w3c.org/TR/xpath> einsehen.

Teil D

Anhang

Anhang A	Funktionslisten alphabetisch	877
Anhang B	Funktionen nach Kategorie	937
Anhang C	Die Beispieldateien zum Buch	957

Anhang A

Funktionslisten alphabetisch

Funktionsliste Deutsch-Englisch

878

Funktionsliste Englisch-Deutsch

907

Prinzipiell sind die folgenden Funktionslisten alphabetisch sortiert. Die neuen bzw. umbenannten Funktionen in Excel 2010 und 2013 machen hier teilweise eine Ausnahme. Wir haben sie immer im Zusammenhang mit den alten Funktionen angeordnet, welche in 2010 bzw. 2013 in der Gruppe der *Kompatibilitäts*-Funktionen erscheinen. Es ergibt sich so nicht immer die korrekte alphabetische Sortierung. Wir ziehen es trotzdem vor, diese zusammengehörigen Funktionen auch untereinander aufzulisten.

Funktionsliste Deutsch-Englisch

ID	Deutsch	Englisch	Kategorie	Zweck
1	ABRUNDEN	ROUNDDOWN	Math. & Trigonom.	Rundet die Zahl auf <i>Anzahl_Stellen</i> ab
2	ABS	ABS	Math. & Trigonom.	Liefert den Absolutwert einer Zahl
3	ACHSENABSCHNITT	INTERCEPT	Statistik	Gibt den Schnittpunkt der Regressionsgeraden zurück
4	ADRESSE	ADDRESS	Matrix	Liefert einen Bezug auf eine Zelle einer Tabelle als Text
5	AGGREGAT (2010)	AGGREGATE	Math. & Trigonom.	Gibt ein Aggregat in einer Liste oder Datenbank zurück
6	AMORDEGRK	AMORDEGRC	Finanzmathematik	Liefert den für eine Abrechnungsperiode anzusetzenden Abschreibungsbetrag auf Basis des französischen Buchführungssystems
7	AMORLINEARK	AMORLINC	Finanzmathematik	Gibt den für eine Abrechnungsperiode anzusetzenden Abschreibungsbetrag auf Basis des französischen Buchführungssystems zurück
8	ANZAHL	COUNT	Statistik	Berechnet, wie viele Zahlen eine Liste von Argumenten enthält
9	ANZAHL2	COUNTA	Statistik	Berechnet, wie viele Werte eine Liste von Argumenten enthält
10	ANZAHLLEEREZELLEN	COUNTBLANK	Statistik	Zählt die leeren Zellen in einem Zellbereich
11	ARABISCH (2013)	ARABIC	Math. & Trigonom.	Wandelt eine römische Zahl in eine arabische Zahl um
12	ARBEITSTAG	WORKDAY	Datum & Zeit	Gibt die Datumsangabe als fortlaufenden Tag im Jahr zurück, vor oder nach einer bestimmten Anzahl von Arbeitstagen
13	ARBEITSTAG.INTL (2010)	WORKDAY.INTL	Datum & Zeit	Gibt unter Beachtung der Wochenenddefinition die Datumsangabe als fortlaufenden Tag im Jahr zurück, vor oder nach einer bestimmten Anzahl von Arbeitstagen ▶

ID	Deutsch	Englisch	Kategorie	Zweck
14	ARCCOS	ACOS	Math. & Trigonom.	Liefert den Arkuskosinus oder umgekehrten Kosinus einer Zahl
15	ARCCOSHYP	ACOSH	Math. & Trigonom.	Liefert den umgekehrten hyperbolischen Kosinus einer Zahl
16	ARCCOT (2013)	ACOT	Math. & Trigonom.	Gibt den Arkuskotangens einer Zahl zurück
17	ARCCOTHYP (2013)	ACOTH	Math. & Trigonom.	Gibt den hyperbolischen Arkuskotangens einer Zahl zurück
18	ARCSIN	ASIN	Math. & Trigonom.	Gibt den Arkussinus oder auch umgekehrten Sinus einer Zahl zurück
19	ARCSINHYP	ASINH	Math. & Trigonom.	Gibt den umgekehrten hyperbolischen Sinus einer Zahl zurück
20	ARCTAN	ATAN	Math. & Trigonom.	Gibt den Arkustangens oder auch umgekehrten Tangens einer Zahl zurück
21	ARCTAN2	ATAN2	Math. & Trigonom.	Gibt den Arkustangens oder auch umgekehrten Tangens ausgehend von einer x- und einer y-Koordinate zurück
22	ARCTANHYP	ATANH	Math. & Trigonom.	Gibt den umgekehrten hyperbolischen Tangens einer Zahl zurück
23	ASC	ASC	Text	Wandelt bei Sprachen mit einem Double-Byte-Zeichensatz (DBCS) Zeichen voller Breite (Double-Byte-Zeichen) in Zeichen halber Breite (Single-Byte-Zeichen) um
24	AUFGELZINS	ACCRINT	Finanzmathematik	Liefert die aufgelaufenen Zinsen (Stückzinsen) eines Wertpapiers mit periodischen Zinszahlungen
25	AUFGELZINSF	ACCRINTM	Finanzmathematik	Liefert die aufgelaufenen Zinsen (Stückzinsen) eines Wertpapiers, die bei Fälligkeit ausgezahlt werden
26	AUFRUNDEN	ROUNDUP	Math. & Trigonom.	Rundet die Zahl auf <i>Anzahl_Stellen</i> auf
27	AUSZAHLUNG	INTRATE	Finanzmathematik	Gibt den Auszahlungsbetrag eines voll investierten Wertpapiers am Fälligkeitstermin zurück
28	BAHTTEXT	BAHTTEXT	Text	Wandelt eine Zahl in Thai-Text um und fügt diesem das Suffix »Baht« hinzu
29	BASIS (2013)	BASE	Math. & Trigonom.	Konvertiert eine Zahl in eine Textdarstellung mit der angegebenen Basis

ID	Deutsch	Englisch	Kategorie	Zweck
30	BEREICH.VERSCHIEBEN	OFFSET	Matrix	Gibt einen Bezug zurück, der gegenüber dem angegebenen Bezug versetzt ist
31	BEREICHE	AREAS	Matrix	Gibt die Anzahl der innerhalb eines Bezuges aufgeführten Bereiche zurück
32	BESSELI	BESSELI	Technisch	Gibt die modifizierte Besselfunktion der ersten Art $I_n(z)$ zurück, die der für rein imaginäre Argumente ausgewerteten Besselfunktion J_n entspricht
33	BESSELJ	BESSELJ	Technisch	Gibt die Besselfunktion der ersten Art $J_n(x)$ zurück
34	BESSELK	BESSELK	Technisch	Gibt die modifizierte Besselfunktion 2. Art $K_n(x)$ zurück
35	BESSELY	BESSELY	Technisch	Gibt die Besselfunktion zweiter Art $Y_n(x)$ zurück, die auch als Webersche Funktion oder Neumannsche Funktion bezeichnet wird
36	BESTIMMTHEITSMASS	RSQ	Statistik	Gibt das Quadrat des Pearsonschen Korrelationskoeffizienten zurück, entsprechend den in <i>Y_Werte</i> und <i>X_Werte</i> abgelegten Datenpunkten
37	BETA.INV (2010)	BETA.INV	Statistik	Siehe BETAINV
38	BETA.VERT (2010)	BETA.DIST	Statistik	Siehe BETAVERT
39	BETAINV	BETAINV	Statistik (ab 2010 Kompatibilität)	Gibt das Quantil der angegebenen Betaverteilung zurück
40	BETAVERT	BETADIST	Statistik (ab 2010 Kompatibilität)	Gibt die Werte der kumulierten Betaverteilungsfunktion zurück
41	BININDEZ	BIN2DEC	Technisch	Wandelt eine binäre Zahl (Dualzahl) in eine dezimale Zahl um
42	BININHEX	BIN2HEX	Technisch	Wandelt eine binäre Zahl (Dualzahl) in eine hexadezimale Zahl um
43	BININOKT	BIN2OCT	Technisch	Wandelt eine Binärzahl (Dualzahl) in eine Oktalzahl um
44	BINOM.INV (2010)	BINOM.INV	Statistik	Gibt den kleinsten Wert, für den die kumulierten Wahrscheinlichkeiten der Binomialverteilung größer oder gleich einer Grenzwahrscheinlichkeit sind, zurück
45	BINOM.VERT (2010)	BINOM.DIST	Statistik	Siehe BINOMVERT
46	BINOM.VERT.BEREICH (2013)	BINOM.DIST.RANGE	Statistik	Gibt die Erfolgswahrscheinlichkeit eines Versuchsergebnisses als Binomialverteilung zurück ►

ID	Deutsch	Englisch	Kategorie	Zweck
47	BINOMVERT	BINOMDIST	Statistik (ab 2010 Kompatibilität)	Gibt Wahrscheinlichkeiten einer binomialverteilten Zufallsvariablen zurück
48	BITLVERSCHIEB (2013)	BITLSHIFT	Technisch	Gibt eine Zahl zurück, die um <i>Verschiebebetrag</i> Bits nach links verschoben ist
49	BITODER (2013)	BITOR	Technisch	Gibt ein bitweises »Oder« zweier Zahlen zurück
50	BITRVERSCHIEB (2013)	BITRSHIFT	Technisch	Gibt eine Zahl zurück, die um <i>Verschiebebetrag</i> Bits nach rechts verschoben ist
51	BITUND (2013)	BITAND	Technisch	Gibt ein bitweises »Und« zweier Zahlen zurück
52	BITXODER (2013)	BITXOR	Technisch	Gibt ein bitweises »Ausschließliches Oder« zweier Zahlen zurück
53	BLATT (2013)	SHEET	Information	Gibt die Nummer des Blatts zurück, welches durch den Bezug <i>Bereich</i> angesprochen wird
54	BLÄTTER (2013)	SHEETS	Information	Gibt die Anzahl der Blätter zurück, welche durch den Bezug <i>Bereich</i> angesprochen werden
55	BOGENMASS	RADIANS	Math. & Trigonom.	Wandelt Grad in Bogenmaß (Radiant) um
56	BRTEILJAHRE	YEARFRAC	Datum & Zeit	Wandelt die Anzahl der ganzen Tage zwischen Ausgangsdatum und Enddatum in Bruchteile von Jahren um
57	BW	PV	Finanzmathematik	Gibt den Barwert einer Investition zurück
58	CHIINV	CHIINV	Statistik (ab 2010 Kompatibilität)	Gibt Quantile der γ^2 -Verteilung zurück
59	CHIQU.INV (2010)	CHISQ.INV	Statistik	Gibt Quantile der linksseitigen γ^2 -Verteilung zurück
60	CHIQU.INV.RE (2010)	CHISQ.INV.RE	Statistik	Gibt Quantile der rechtsseitigen γ^2 -Verteilung zurück
61	CHITEST	CHITEST	Statistik (ab 2010 Kompatibilität)	Gibt die Teststatistik eines γ^2 -Unabhängigkeitstests zurück
62	CHIQU.TEST (2010)	CHISQ.TEST	Statistik	Siehe CHITEST
63	CHIVERT	CHIDIST	Statistik (ab 2010 Kompatibilität)	Gibt Werte der Verteilungsfunktion (1-Alpha) einer Chi-Quadratverteilten Zufallsgröße zurück ►

ID	Deutsch	Englisch	Kategorie	Zweck
64	CHIQU.VERT (2010)	CHISQ.DIST	Statistik	Gibt Werte der linksseitigen Verteilungsfunktion (1-Alpha) einer Chi-Quadrat-verteilten Zufallsgröße zurück
65	CHIQU.VERT.RE (2010)	CHISQ.DIST.RT	Statistik	Gibt Werte der rechtsseitigen Verteilungsfunktion (1-Alpha) einer Chi-Quadrat-verteilten Zufallsgröße zurück
66	CODE	CODE	Text	Gibt die Codezahl des ersten Zeichens in einem Text zurück
67	COS	COS	Math. & Trigonom.	Gibt den Kosinus einer Zahl zurück
68	COSEC (2013)	CSC	Math. & Trigonom.	Gibt den Kosekans eines Winkels zurück
69	COSECHYP (2013)	CSCH	Math. & Trigonom.	Gibt den hyperbolischen Kosekans eines Winkels zurück
70	COSHYP	COSH	Math. & Trigonom.	Gibt den hyperbolischen Kosinus einer Zahl zurück
71	COT (2013)	COT	Math. & Trigonom.	Gibt den Kotangens eines Winkels zurück
72	COTHYP (2013)	COTH	Math. & Trigonom.	Gibt den hyperbolischen Kotangens einer Zahl zurück
73	CUBEELEMENT	CUBEMEMBER	Cubefunktionen	Gibt für den angegebenen Dimensionspfad das Dimensionselement zurück
74	CUBEELEMENT-EIGENSCHAFT	CUBEMEMBER-PROPERTY	Cubefunktionen	Gibt für ein Dimensionselement die geforderte Eigenschaft, auch Attribut genannt, zurück
75	CUBEKPIELEMENT	CUBEKPIMEMBER	Cubefunktionen	Gibt die angeforderte Eigenschaft von Key Performance-Indikatoren eines Würfels zurück
76	CUBEMENGE	CUBESSET	Cubefunktionen	Gibt eine definierte Menge von Elementen eines Cubes zurück. Dieses sog. Set stellt einen Teilwürfel dar.
77	CUBEMENGENANZAHL	CUBESSETCOUNT	Cubefunktionen	Gibt die Anzahl der Elemente eines Sets zurück
78	CUBERANGELEMENT	CUBERANKED-MEMBER	Cubefunktionen	Gibt das per Index definierte Element eines Sets zurück
79	CUBEWERT	CUBEVALUE	Cubefunktionen	Gibt den Wert für eine gegebene Adresse aus einem Datenwürfel zurück
80	DATEDIF	DATEDIF	Datum & Zeit	Ermittelt die Differenz zwischen <i>Anfangs-</i> und <i>Enddatum</i> in Jahren, Monaten oder Tagen ▶

ID	Deutsch	Englisch	Kategorie	Zweck
81	DATUM	DATE	Datum & Zeit	Gibt die fortlaufende Zahl zurück, die ein bestimmtes Datum darstellt
82	DATWERT	DATEVALUE	Datum & Zeit	Wandelt ein als Text vorliegendes Datum in eine fortlaufende Zahl um
83	DBANZAHL	DCOUNT	Datenbank	Zählt die Anzahl von Zellen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
84	DBANZAHL2	DCOUNTA	Datenbank	Zählt die Anzahl von Zellen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
85	DBAUSZUG	DGET	Datenbank	Extrahiert einen einzelnen Wert in einer Spalte einer Liste oder Datenbank, der den angegebenen Bedingungen entspricht
86	DBMAX	DMAX	Datenbank	Liefert die größte Anzahl in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entspricht
87	DBMIN	DMIN	Datenbank	Liefert die kleinste Anzahl in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entspricht
88	DBMITTELWERT	DAVERAGE	Datenbank	Liefert den Mittelwert aus den Werten einer Listen- oder Datenbankspalte, die den von Ihnen angegebenen Bedingungen entsprechen
89	DBPRODUKT	DPRODUCT	Datenbank	Multipliziert die Werte in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
90	DBSTDABW	DSTDEV	Datenbank	Schätzt die Standardabweichung einer Grundgesamtheit ausgehend von den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
91	DBSTDABWN	DSTDEVP	Datenbank	Berechnet die Standardabweichung einer vollständigen Grundgesamtheit ausgehend von den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
92	DBSUMME	DSUM	Datenbank	Fügt die Zahlen in einer Spalte aus einer Liste oder Datenbank hinzu, die den angegebenen Bedingungen entsprechen ►

ID	Deutsch	Englisch	Kategorie	Zweck
93	DBVARIANZ	DVAR	Datenbank	Schätzt die Varianz einer Grundgesamtheit ausgehend von einer Stichprobe mit den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
94	DBVARIANZEN	DVARP	Datenbank	Berechnet die Varianz einer vollständigen Grundgesamtheit ausgehend von den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
95	DELTA	DELTA	Technisch	Überprüft, ob zwei Werte gleich sind
96	DEZIMAL (2013)	DECIMAL	Math. & Trigonom.	Konvertiert eine Textdarstellung einer Zahl mit einer angegebenen Basis in eine Dezimalzahl
97	DEZINBIN	DEC2BIN	Technisch	Wandelt eine dezimale Zahl in eine binäre Zahl (Dualzahl) um
98	DEZINHEX	DEC2HEX	Technisch	Wandelt eine dezimale Zahl in eine hexadezimale Zahl um
99	DEZINOKT	DEC2OCT	Technisch	Wandelt eine Dezimalzahl in eine Oktalzahl um
100	DIA	SYD	Finanzmathematik	Gibt die arithmetisch-degressive Abschreibung eines Wirtschaftsguts für eine bestimmte Periode zurück
101	DISAGIO	DISC	Finanzmathematik	Gibt den in Prozent ausgedrückten Abzinsungssatz eines Wertpapiers zurück
102	DM	DOLLAR	Text	Konvertiert eine Zahl in ein Textformat und ordnet ein Währungssymbol zu
103	DURATION	DURATION	Finanzmathematik	Gibt für einen angenommenen Nennwert von 100 € die Macauley-Dauer zurück
104	EDATUM	EDATE	Datum & Zeit	Gibt die fortlaufende Zahl des Datums zurück, das eine bestimmte Anzahl von Monaten vor bzw. nach dem angegebenen Datum (Ausgangsdatum) liegt
105	EFFEKTIV	EFFECT	Finanzmathematik	Gibt die jährliche Effektivverzinsung zurück, ausgehend von einer Nominalverzinsung sowie der jeweiligen Anzahl der Zinszahlungen pro Jahr



ID	Deutsch	Englisch	Kategorie	Zweck
106	ERSETZEN	REPLACE	Text	Ersetzt auf der Grundlage der Anzahl von Zeichen, die Sie angeben, einen Teil einer Textzeichenfolge durch eine andere Textzeichenfolge
107	ERSETZENB	REPLACEB	Text	Ersetzt auf der Grundlage der von Ihnen angegebenen Anzahl der Bytes einen Teil einer Textzeichenfolge durch eine andere Textzeichenfolge
108	EXP	EXP	Math. & Trigonom.	Potenziert die Basis e mit der als Argument angegebenen Zahl. Die Konstante e ist die Basis des natürlichen Logarithmus und hat den Wert $2,71828182845904$.
109	EXPON.VERT (2010)	EXPON.DIST	Statistik	Siehe EXPONVERT
110	EXPONVERT	EXPONDIST	Statistik (ab 2010 Kompatibilität)	Gibt Wahrscheinlichkeiten einer exponential verteilten Zufallsvariablen zurück
111	FAKULTÄT	FACT	Math. & Trigonom.	Gibt die Fakultät einer Zahl zurück. Die Fakultät einer Zahl wird aus $1*2*3*... * Zahl$ berechnet.
112	FALSCH	FALSE	Logik	Gibt den Wahrheitswert <i>FALSCH</i> zurück
113	FEHLER.TYP	ERROR.TYPE	Information	Gibt eine Zahl zurück, die einem der Fehlerwerte in Microsoft Excel entspricht, oder den Fehlerwert #NV, wenn kein Fehler vorhanden ist
114	FEST	FIXED	Text	Formatiert eine Zahl als Text mit einer festen Anzahl von Nachkommastellen
115	FINDEN	FIND	Text	Sucht eine Zeichenfolge (Suchtext) innerhalb einer anderen Zeichenfolge (Text) und gibt als Ergebnis die Nummer der Anfangsposition von Suchtext ab dem ersten Zeichen von Text zurück
116	FINDENB	FINDB	Text	Sucht eine Zeichenfolge (Suchtext) innerhalb einer anderen Zeichenfolge (Text) und gibt als Ergebnis die Nummer der Anfangsposition von Suchtext ab dem ersten Zeichen von Text auf der Grundlage der Anzahl von Bytes zurück, die jedes Zeichen verwendet
117	FINV	FINV	Statistik (ab 2010 Kompatibilität)	Gibt Quantile der F-Verteilung zurück

ID	Deutsch	Englisch	Kategorie	Zweck
118	F.INV (2010)	F.INV	Statistik	Gibt Quantile der linksseitigen F-Verteilung zurück
119	F.INV.RE (2010)	F.INV.RT	Statistik	Gibt Quantile der rechtsseitigen F-Verteilung zurück
120	FISHER	FISHER	Statistik	Gibt die Fisher-Transformation für x zurück
121	FISHERINV	FISHERINV	Statistik	Gibt die Umkehrung der Fisher-Transformation zurück
122	FORMELTEXT (2013)	FORMULATEXT	Matrix	Liefert die Formel, die im gegebenen Bereich eingetragen wurde, als Text
123	FTEST	FTEST	Statistik (ab 2010 Kompatibilität)	Gibt die Teststatistik eines F-Tests zurück
124	F.TEST (2010)	F.TEST	Statistik	Siehe FTEST
125	FVERT	FDIST	Statistik (ab 2010 Kompatibilität)	Gibt Werte der Verteilungsfunktion (1-Alpha) einer F-verteilten Zufallsvariablen zurück
126	F.VERT (2010)	F.DIST	Statistik	Gibt Werte der Verteilungsfunktion (1-Alpha) einer linksseitigen F-verteilten Zufallsvariablen zurück
127	F.VERT.RE (2010)	F.DIST.RT	Statistik	Gibt Werte der Verteilungsfunktion (1-Alpha) einer rechtsseitigen F-verteilten Zufallsvariablen zurück
128	GAMMA (2013)	GAMMA	Statistik	Gibt den Wert der Gammafunktion für eine bestimmte Zahl zurück
129	GAMMAINV	GAMMAINV	Statistik (ab 2010 Kompatibilität)	Gibt Quantile der Gammaverteilung zurück
130	GAMMA.INV (2010)	GAMMA.INV	Statistik	Siehe GAMMAINV
131	GAMMALN	GAMMALN	Statistik	Gibt den natürlichen Logarithmus der Gammafunktion zurück, $\Gamma(x)$
132	GAMMALN.GENAU (2010)	GAMMALN.PRECISE	Statistik	Siehe GAMMALN
133	GAMMAVERT	GAMMADIST	Statistik (ab 2010 Kompatibilität)	Gibt Wahrscheinlichkeiten einer gammaverteilten Zufallsvariablen zurück
134	GAMMA.VERT (2010)	GAMMA.DIST	Statistik	Siehe GAMMAVERT
135	GANZZAHL	INT	Math. & Trigonom.	Rundet eine Zahl auf die nächste kleinere ganze Zahl ab
136	GAUSS (2013)	GAUSS	Statistik	Gibt die Wahrscheinlichkeit, mit der eine standardnormal-verteilte Zufallsvariable zwischen dem Mittelwert der Grundgesamtheit und x Standardabweichungen vom Mittelwert liegt, zurück. ►

ID	Deutsch	Englisch	Kategorie	Zweck
137	GAUSSFEHLER	ERF	Technisch	Gibt die Gaußsche Fehlerfunktion zurück
138	GAUSSF.GENAU (2010)	ERF.PRECISE	Technisch	Siehe GAUSSFEHLER
139	GAUSSFKOMPL	ERFC	Technisch	Gibt das Komplement zur Gaußschen Fehlerfunktion zurück
140	GAUSSFKOMPL.GENAU (2010)	ERFC.PRECISE	Technisch	Siehe GAUSSFKOMPL
141	GDA	DDB	Finanzmathematik	Gibt die Abschreibung eines Anlageguts für einen angegebenen Zeitraum unter Verwendung der degressiven Doppelratenabschreibung oder eines anderen von Ihnen angegebenen Abschreibungsverfahrens zurück
142	GDA2	DB	Finanzmathematik	Gibt die geometrisch-degressive Abschreibung eines Wirtschaftsguts für eine bestimmte Periode zurück
143	GEOMITTEL	GEOMEAN	Statistik	Gibt das geometrische Mittel einer Menge positiver Zahlen zurück
144	GERADE	EVEN	Math. & Trigonom.	Rundet eine Zahl auf die nächste gerade ganze Zahl auf
145	GESTUTZTMITTEL	TRIMMEAN	Statistik	Gibt den Mittelwert einer Daten­gruppe zurück, ohne die Randwerte zu berücksichtigen
146	GGANZZAHL	GESTEP	Technisch	Gibt den Wert 1 zurück, wenn $Zahl \geq Schritt$ gilt; andernfalls gibt sie 0 (Null) zurück
147	GGT	GCD	Math. & Trigonom.	Gibt den größten gemeinsamen Teiler zurück
148	GLÄTTEN	TRIM	Text	Löscht Leerzeichen in Text, die nicht als jeweils einzelne zwischen Wörtern stehende Trennzeichen dienen
149	GRAD	DEGREES	Math. & Trigonom.	Wandelt ein Bogenmaß (Radiant) in Grad um
150	GROSS	UPPER	Text	Wandelt Text in Großbuchstaben um
151	GROSS2	PROPER	Text	Wandelt den ersten Buchstaben aller Wörter einer Zeichenfolge in Großbuchstaben und alle anderen Buchstaben in Kleinbuchstaben um
152	GTEST	ZTEST	Statistik (ab 2010 Kompatibilität)	Gibt den einseitigen Wahrscheinlichkeitswert für einen Gaußtest (Normalverteilung) zurück ►

ID	Deutsch	Englisch	Kategorie	Zweck
153	G.TEST (2010)	Z.TEST	Statistik	Gibt den einseitigen Wahrscheinlichkeitswert für einen Gaußtest (Normalverteilung) zurück
154	HARMITTEL	HARMEAN	Statistik	Gibt das harmonische Mittel einer Datenmenge zurück
155	HÄUFIGKEIT	FREQUENCY	Statistik	Gibt eine Häufigkeitsverteilung als einspaltige Matrix zurück
156	HEUTE	TODAY	Datum & Zeit	Gibt die fortlaufende Zahl des heutigen Datums zurück
157	HEXINBIN	HEX2BIN	Technisch	Wandelt eine hexadezimale Zahl in eine Binärzahl um
158	HEXINDEZ	HEX2DEC	Technisch	Wandelt eine hexadezimale Zahl in eine dezimale Zahl um
159	HEXINOKT	HEX2OCT	Technisch	Wandelt eine hexadezimale Zahl in eine Oktalzahl um
160	HYPERLINK	HYPERLINK	Matrix	Erstellt eine Verknüpfung oder einen Sprung, über die ein auf einem Netzwerkservers, in einem Intranet oder im Internet gespeichertes Dokument geöffnet wird
161	HYPGEOM.VERT (2010)	HYPGEOM.DIST	Statistik	Gibt Wahrscheinlichkeiten einer hypergeometrisch-verteilten Zufallsvariablen zurück
162	HYPGEOMVERT	HYPGEOMDIST	Statistik (ab 2010 Kompatibilität)	Gibt Wahrscheinlichkeiten einer hypergeometrisch-verteilten Zufallsvariablen zurück
163	IDENTISCH	EXACT	Text	Prüft, ob zwei Zeichenfolgen identisch sind
164	IKV	IRR	Finanzmathematik	Gibt den internen Zinsfuß einer Investition ohne Finanzierungskosten oder Reinvestitionsgewinne zurück
165	IMABS	IMABS	Technisch	Gibt den Absolutwert (Modul) einer komplexen Zahl zurück
166	IMAGINÄRTEIL	IMAGINARY	Technisch	Gibt den Imaginärteil einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt
167	IMAPOTENZ	IMPOWER	Technisch	Potenziiert eine komplexe Zahl, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt, mit einer ganzen Zahl
168	IMARGUMENT	IMARGUMENT	Technisch	Gibt das Argument (Theta) zurück, einen Winkel, der als Bogenmaß ausgedrückt wird



ID	Deutsch	Englisch	Kategorie	Zweck
169	IMCOS	IMCOS	Technisch	Gibt den Kosinus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt
170	IMCOSEC (2013)	IMCSC	Technisch	Gibt den Kosekans einer komplexen Zahl zurück
171	IMCOSECHYP (2013)	IMCSCH	Technisch	Gibt den hyperbolischen Kosekans einer komplexen Zahl zurück
172	IMCOSHYP (2013)	IMCOSH	Technisch	Gibt den hyperbolischen Kosinus einer komplexen Zahl zurück
173	IMCOT (2013)	IMCOT	Technisch	Gibt den Kotangens einer komplexen Zahl zurück
174	IMDIV	IMDIV	Technisch	Gibt den Quotienten zweier komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
175	IMEXP	IMEXP	Technisch	Gibt die algebraische Form einer in exponentieller Form vorliegenden komplexen Zahl zurück, wobei deren Exponent als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
176	IMKONJUGIERTE	IMCONJUGATE	Technisch	Gibt die konjugiert komplexe Zahl zu einer komplexen Zahl zurück, wobei die komplexe Zahl als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
177	IMLN	IMLN	Technisch	Gibt den natürlichen Logarithmus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
178	IMLOG10	IMLOG10	Technisch	Gibt den Logarithmus einer komplexen Zahl zur Basis 10 zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
179	IMLOG2	IMLOG2	Technisch	Gibt den Logarithmus einer komplexen Zahl zur Basis 2 zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
180	IMPRODUKT	IMPRODUCT	Technisch	Gibt das Produkt komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
181	IMREALTEIL	IMREAL	Technisch	Gibt den Realteil einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
182	IMSEC (2013)	IMSEC	Technisch	Gibt den Sekans einer komplexen Zahl zurück

ID	Deutsch	Englisch	Kategorie	Zweck
183	IMSECHYP (2013)	IMSECH	Technisch	Gibt den hyperbolischen Sekans einer komplexen Zahl zurück
184	IMSIN	IMSIN	Technisch	Diese Funktion gibt den Sinus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
185	IMSINHYP (2013)	IMSINH	Technisch	Gibt den hyperbolischen Sinus einer komplexen Zahl zurück
186	IMSUB	IMSUB	Technisch	Gibt die Differenz zweier komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
187	IMSUMME	IMSUM	Technisch	Gibt die Summe komplexer Zahlen zurück, die als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
188	IMTAN (2013)	IMTAN	Technisch	Gibt den Tangens einer komplexen Zahl zurück
189	IMWURZEL	IMSQRT	Technisch	Gibt die Quadratwurzel einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
190	INDEX	INDEX	Matrix	Verwendet einen Index, um aus einem Bezug oder einer Matrix einen Wert zu wählen
191	INDIREKT	INDIRECT	Matrix	Gibt den Bezug eines Textwerts zurück. Bezüge werden sofort ausgewertet, sodass die zu ihnen gehörenden Werte angezeigt werden.
192	INFO	INFO	Information	Gibt Informationen zu der aktuellen Betriebssystemumgebung zurück
193	ISOKALENDERWOCHE (2013)	ISOWEEKNUM	Datum & Zeit	Berechnet die Kalenderwochennummer nach ISO-Norm für einen Datumswert
194	ISPMT	ISPMT	Finanzmathematik	Berechnet die während eines bestimmten Zeitraums für eine Investition gezahlten Zinsen
195	ISTBEZUG	ISREF	Information	Gibt <i>WAHR</i> zurück, wenn der Wert ein gültiger Zellbezug ist, oder <i>FALSCH</i> , wenn der Wert ein ungültiger Zellbezug ist
196	ISTFEHL	ISERR	Information	Gibt <i>WAHR</i> zurück, wenn der Wert einen Fehler ergibt, oder <i>FALSCH</i> , wenn der Wert keinen Fehler ergibt



ID	Deutsch	Englisch	Kategorie	Zweck
197	ISTFEHLER	ISERROR	Information	Gibt <i>WAHR</i> zurück, wenn der Wert einen beliebigen Fehler ergibt, oder <i>FALSCH</i> , wenn der Wert keinen Fehler ergibt
198	ISTFORMEL (2013)	ISFORMULA	Information	Gibt <i>WAHR</i> zurück, wenn ein Bezug auf eine existierende Zelle verweist und sich dort eine Formel befindet
199	ISTGERADE	ISEVEN	Information	Gibt <i>WAHR</i> zurück, wenn die Zahl gerade ist, oder <i>FALSCH</i> , wenn die Zahl ungerade ist
200	ISTKTEXT	ISNONTEXT	Information	Gibt <i>WAHR</i> zurück, wenn der Wert kein Text ist, oder <i>FALSCH</i> , wenn der Wert ein Text ist
201	ISTLEER	ISBLANK	Information	Gibt <i>WAHR</i> zurück, wenn der Wert sich auf eine leere Zelle bezieht, oder <i>FALSCH</i> , wenn der Wert sich auf eine nicht leere Zelle bezieht
202	ISTLOG	ISLOGICAL	Information	Gibt <i>WAHR</i> zurück, wenn der Wert sich auf einen Wahrheitswert bezieht, oder <i>FALSCH</i> , wenn der Wert sich nicht auf einen Wahrheitswert bezieht
203	ISTNV	ISNA	Information	Gibt <i>WAHR</i> zurück, wenn der Wert sich auf den Fehlerwert <i>#NV</i> bezieht, oder <i>FALSCH</i> , wenn der Wert sich nicht auf den Fehlerwert <i>#NV</i> bezieht
204	ISTTEXT	ISTEXT	Information	Gibt <i>WAHR</i> zurück, wenn der Wert ein Text ist, oder <i>FALSCH</i> , wenn der Wert kein Text ist. Beachten Sie, dass diese Funktion <i>FALSCH</i> zurückgibt, wenn sich der Wert auf eine leere Zelle bezieht.
205	ISTUNGERADE	ISODD	Information	Gibt <i>WAHR</i> zurück, wenn die Zahl ungerade ist, oder <i>FALSCH</i> , wenn die Zahl gerade ist
206	ISTZAHL	ISNUMBER	Information	Gibt <i>WAHR</i> zurück, wenn der Wert eine Zahl (numerischer Wert) ist, oder <i>FALSCH</i> , wenn der Wert keine Zahl ist
207	JAHR	YEAR	Datum & Zeit	Wandelt eine fortlaufende Zahl in eine Jahreszahl um
208	JETZT	NOW	Datum & Zeit	Liefert die fortlaufende Zahl des aktuellen Datums und der aktuellen Uhrzeit ►

ID	Deutsch	Englisch	Kategorie	Zweck
209	KALENDERWOCHE	WEEKNUM	Datum & Zeit	Gibt eine Zahl zurück, die angibt, in welche Woche des dazugehörigen Jahres das angegebene Datum fällt
210	KAPZ	PPMT	Finanzmathematik	Gibt die Kapitalrückzahlung einer Investition für die angegebene Periode zurück
211	KGRÖSSTE	LARGE	Statistik	Gibt den k -größten Wert einer Datengruppe zurück
212	KGV	LCM	Math. & Trigonom.	Gibt das kleinste gemeinsame Vielfache der als Argumente angegebenen ganzen Zahlen zurück
213	KKLEINSTE	SMALL	Statistik	Gibt den k -kleinsten Wert einer Datengruppe zurück
214	KLEIN	LOWER	Text	Wandelt einen Text in Kleinbuchstaben um
215	KOMBINATIONEN	COMBIN	Math. & Trigonom.	Berechnet, wie viele Gruppen aus einer bestimmten Anzahl von Elementen gebildet werden können
216	KOMBINATIONEN2 (2013)	COMBINA	Math. & Trigonom.	Gibt die Anzahl der Kombinationen mit Wiederholung für eine angegebene Anzahl von Elementen zurück
217	KOMPLEXE	COMPLEX	Technisch	Wandelt den Real- und Imaginärteil in eine komplexe Zahl um ($x + yi$ oder $x + yj$)
218	KONFIDENZ	CONFIDENCE	Statistik (ab 2010 Kompatibilität)	Gibt einen Wert zurück, den Sie zum Erstellen eines Konfidenzintervalls für den Erwartungswert einer Zufallsvariablen verwenden können
219	KONFIDENZ.NORM (2010)	CONFIDENCE.NORM	Statistik	Siehe KONFIDENZ
220	KONFIDENZ.T (2010)	CONFIDENCE.T	Statistik	Gibt einen Wert zurück, den Sie zum Erstellen eines Konfidenzintervalls für den Erwartungswert einer t-verteilten Zufallsvariablen verwenden können
221	KORREL	CORREL	Statistik	Gibt den Korrelationskoeffizienten einer zweidimensionalen Zufallsgröße zurück, deren Werte in den Zellbereichen <i>Matrix1</i> und <i>Matrix2</i> stehen
222	KOVAR	COVAR	Statistik (ab 2010 Kompatibilität)	Diese Funktion gibt die Kovarianz zurück, den Mittelwert der für alle Datenpunktpaare gebildeten Produkte der Abweichungen
223	KOVARIANZ.P (2010)	COVARIANCE.P	Statistik	Siehe KOVAR ▶

ID	Deutsch	Englisch	Kategorie	Zweck
224	KOVARIANZ.T (2010)	COVARIANCE.S	Statistik	Diese Funktion gibt die Kovarianz einer Stichprobe zurück, den Mittelwert der für alle Datenpunktpaare gebildeten Produkte der Abweichungen
225	KRITBINOM	CRITBINOM	Statistik	Gibt den kleinsten Wert zurück, für den die kumulierten Wahrscheinlichkeiten der Binomialverteilung größer oder gleich einer Grenzwahrscheinlichkeit sind
226	KUMKAPITAL	CUMPRINC	Finanzmathematik	Berechnet die aufgelaufene Tilgung eines Darlehens, die zwischen zwei Perioden zu zahlen ist
227	KUMZINSZ	CUMIPMT	Finanzmathematik	Berechnet die kumulierten Zinsen, die zwischen zwei Perioden zu zahlen sind
228	KURS	PRICE	Finanzmathematik	Gibt den Kurs pro 100 € Nennwert eines Wertpapiers zurück, das periodisch Zinsen auszahlt
229	KURSDISAGIO	PRICEDISC	Finanzmathematik	Gibt den Kurs pro 100 € Nennwert eines unverzinslichen Wertpapiers zurück
230	KURSFÄLLIG	PRICEMAT	Finanzmathematik	Gibt den Kurs pro 100 € Nennwert eines Wertpapiers zurück, das Zinsen am Fälligkeitsdatum auszahlt
231	KURT	KURT	Statistik	Gibt die Kurtosis (Exzess) einer Datengruppe zurück
232	KÜRZEN	TRUNC	Math. & Trigonom.	Schneidet die Kommastellen der Zahl ab und gibt als Ergebnis eine ganze Zahl zurück
233	LÄNGE	LEN	Text	Gibt die Anzahl der Zeichen einer Zeichenfolge zurück
234	LIA	SLN	Finanzmathematik	Gibt die lineare Abschreibung eines Wirtschaftsguts pro Periode zurück
235	LINKS	LEFT	Text	Gibt auf der Grundlage der Anzahl von Zeichen, die Sie angeben, das oder die erste(n) Zeichen in einer Textzeichenfolge zurück
236	LN	LN	Math. & Trigonom.	Gibt den natürlichen Logarithmus einer Zahl zurück
237	LOG	LOG	Math. & Trigonom.	Gibt den Logarithmus einer Zahl zu der angegebenen Basis zurück
238	LOG10	LOG10	Math. & Trigonom.	Gibt den Logarithmus einer Zahl zur Basis 10 zurück

ID	Deutsch	Englisch	Kategorie	Zweck
239	LOGINV	LOGINV	Statistik (ab 2010 Kompatibilität)	Gibt Perzentile der Lognormalverteilung von x zurück, wobei $\ln(x)$ mit den Parametern <i>Mittelwert</i> und <i>Standabwn</i> normal verteilt ist
240	LOGNORM.INV (2010)	LOGNORM.INV	Statistik	Siehe LOGINV
241	LOGNORMVERT	LOGNORMDIST	Statistik (ab 2010 Kompatibilität)	Gibt Werte der Verteilungsfunktion einer lognormalverteilten Zufallsvariablen zurück, wobei $\ln(x)$ normalverteilt ist mit den Parametern <i>Mittelwert</i> und <i>Standabwn</i>
242	LOGNORM.VERT (2010)	LOGNORM.DIST	Statistik	Siehe LOGNORMVERT
243	MAX	MAX	Statistik	Gibt den größten Wert innerhalb einer Argumentliste zurück
244	MAXA	MAXA	Statistik	Gibt den größten Wert einer Liste von Argumenten zurück
245	MDET	MDETERM	Math. & Trigonom.	Gibt die Determinante einer Matrix zurück
246	MDURATION	MDURATION	Finanzmathematik	Gibt die modifizierte Macauley-Duration eines Wertpapiers mit einem angenommenen Nennwert von 100 € zurück
247	MEDIAN	MEDIAN	Statistik	Gibt den Median der angegebenen Zahlen zurück
248	MEINHEIT (2013)	MUNIT	Math. & Trigonom.	Gibt die Einheitsmatrix für die angegebene Größe zurück
249	MIN	MIN	Statistik	Gibt den kleinsten Wert innerhalb einer Argumentliste zurück
250	MINA	MINA	Statistik	Gibt den kleinsten Wert einer Liste von Argumenten zurück
251	MINUTE	MINUTE	Datum & Zeit	Wandelt eine fortlaufende Zahl in eine Minute um
252	MINV	MINVERSE	Math. & Trigonom.	Gibt die Inverse einer Matrix (die zu einer Matrix gehörende Kehrmatrix) zurück
253	MITTELABW	AVEDEV	Statistik	Gibt die durchschnittliche absolute Abweichung einer Reihe von Merkmalsausprägungen und ihrem Mittelwert zurück
254	MITTELWERT	AVERAGE	Statistik	Durch das arithmetische Mittel wird der Zahlenwert bestimmt, der sich bei gleichmäßiger Aufteilung der Summen auf die Werte ergibt
255	MITTELWERTA	AVERAGEA	Statistik	Berechnet den Mittelwert (arithmetisches Mittel) der Werte in der Liste der Argumente ▶

ID	Deutsch	Englisch	Kategorie	Zweck
256	MITTELWERTWENN	AVERAGEIF	Statistik	Durchschnittswert (arithmetisches Mittel) für alle Zellen in einem Bereich, die einem angegebenen Kriterium entsprechen
257	MITTELWERTWENNS	AVERAGEIFS	Statistik	Gibt den Durchschnittswert (arithmetisches Mittel) aller Zellen zurück, die mehreren Kriterien entsprechen
258	MMULT	MMULT	Math. & Trigonom.	Gibt das Produkt zweier Matrizen zurück
259	MODALWERT	MODE	Statistik (ab 2010 Kompatibilität)	Gibt den häufigsten Wert einer Datengruppe zurück
260	MODUS.EINF (2010)	MODE.SNGL	Statistik	Siehe MODALWERT
261	MODUS.VIELF (2010)	MODE.MULT	Statistik	Gibt ein vertikales Array der am häufigsten vorkommenden Werte in einem Array oder Datenbereich zurück
262	MONAT	MONTH	Datum & Zeit	Wandelt eine fortlaufende Zahl in einen Monat um
263	MONATSENDE	EOMONTH	Datum & Zeit	Gibt die fortlaufende Zahl des letzten Tages des Monats zurück, der eine bestimmte Anzahl von Monaten vor bzw. nach dem Ausgangsdatum liegt
264	MTRANS	TRANSPOSE	Matrix	Gibt die transponierte Matrix der angegebenen Matrix zurück
265	N	N	Information	Gibt den in eine Zahl umgewandelten Wert zurück
266	NBW	NPV	Finanzmathematik	Gibt den Nettobarwert (Kapitalwert) einer Investition auf Basis eines Abzinsungsfaktors für eine Reihe periodischer Zahlungen zurück
267	NEGBINOMVERT	NEGBINOMDIST	Statistik (ab 2010 Kompatibilität)	Gibt Wahrscheinlichkeiten einer negativbinomialverteilten Zufallsvariablen zurück
268	NEGBINOM.VERT (2010)	NEGBINOM.DIST	Statistik	Siehe NEGBINOMVERT
269	NETTOARBEITSTAGE	NETWORKDAYS	Datum & Zeit	Gibt die Anzahl der Arbeitstage in einem Zeitintervall zurück
270	NETTOARBEITSTAGE.INTL (2010)	NETWORK-DAYS.INTL	Datum & Zeit	Gibt die Anzahl der Arbeitstage in einem Zeitintervall unter Beachtung der Wochenend-Definition zurück
271	NICHT	NOT	Logik	Keht den Wert eines Arguments um

ID	Deutsch	Englisch	Kategorie	Zweck
272	NOMINAL	NOMINAL	Finanzmathematik	Gibt die jährliche Nominalverzinsung zurück, ausgehend vom effektiven Zinssatz sowie der Anzahl der Verzinsungsperioden innerhalb eines Jahres
273	NORMINV	NORMINV	Statistik (ab 2010 Kompatibilität)	Gibt Perzentile der Normalverteilung zurück
274	NORM.INV (2010)	NORM.INV	Statistik	Siehe NORMINV
275	NORM.S.INV (2010)	NORM.S.INV	Statistik	Gibt Perzentile der Standardnormalverteilung zurück
276	NORMVERT	NORMDIST	Statistik (ab 2010 Kompatibilität)	Gibt die Normalverteilung für den angegebenen Mittelwert und die angegebene Standardabweichung zurück
277	NORM.VERT (2010)	NORM.DIST	Statistik	Siehe NORMVERT
278	NORM.S.VERT (2010)	NORM.S.DIST	Statistik	Gibt Werte der Verteilungsfunktion einer standardnormalverteilten Zufallsvariablen zurück
279	NOTIERUNGBRU	DOLLARFR	Finanzmathematik	Wandelt eine Notierung, die als Dezimalzahl ausgedrückt wurde, in einen Dezimalbruch um
280	NOTIERUNGDEZ	DOLLARDE	Finanzmathematik	Wandelt eine Notierung, die als Dezimalbruch ausgedrückt wurde, in eine Dezimalzahl um
281	NV	NA	Information	Gibt den Fehlerwert #NV zurück
282	OBERGRENZE	CEILING	Math. & Trigonom.	Rundet eine Zahl betragsmäßig auf das kleinste Vielfache von Schritt auf
283	OBERGRENZE.GENAU (2010)	CEILING.PRECISE	Math. & Trigonom.	Siehe OBERGRENZE
284	OBERGRENZE.MATHEMATIK (2013)	CEILING.MATH	Math. & Trigonom.	Rundet eine Zahl auf die nächste Ganzzahl oder auf das kleinste Vielfache des angegebenen Schritts auf
285	ODER	OR	Logik	Gibt <i>WAHR</i> zurück, wenn ein Argument <i>WAHR</i> ist. Gibt <i>FALSCH</i> zurück, wenn alle Argumente <i>FALSCH</i> sind.
286	OKTINBIN	OCT2BIN	Technisch	Wandelt eine Oktalzahl in eine Binärzahl (Dualzahl) um
287	OKTINDEZ	OCT2DEC	Technisch	Wandelt eine Oktalzahl in eine Dezimalzahl um
288	OKTINHEX	OCT2HEX	Technisch	Wandelt eine Oktalzahl in eine Hexadezimalzahl um



ID	Deutsch	Englisch	Kategorie	Zweck
289	PDURATION	PDURATION	Finanzmathematik	Berechnet die Laufzeit einer Investition bei gegebenem Zinssatz sowie eingesetztem und erwartetem Kapital
290	PEARSON	PEARSON	Statistik	Gibt den Pearsonschen Korrelationskoeffizienten r zurück
291	PHI (2013)	PHI	Statistik	Gibt den Wert der Dichtefunktion für eine Standardnormalverteilung zurück
292	PHONETIC	PHONETIC	Text	Extrahiert die phonetischen (Furigana-) Zeichen aus einer Textzeichenfolge
293	PI	PI	Math. & Trigonom.	Gibt den Wert Pi zurück, die mathematische Konstante (3,14159265358979) mit einer Genauigkeit von 15 Stellen
294	PIVOTDATEN-ZUORDNEN	GETPIVOTDATA	Datenbank	Gibt Daten aus einem PivotTable-Bericht zurück
295	POISSON	POISSON	Statistik (ab 2010 Kompatibilität)	Gibt Wahrscheinlichkeiten einer poissonverteilten Zufallsvariablen zurück
296	POISSON.VERT (2010)	POISSON.DIST	Statistik	Siehe POISSON
297	POLYNOMIAL	MULTINOMINAL	Math. & Trigonom.	Gibt den Polynomkoeffizienten einer Gruppe von Zahlen zurück
298	POTENZ	POWER	Math. & Trigonom.	Gibt als Ergebnis eine potenzierte Zahl zurück
299	POTENZREIHE	SERIESSUM	Math. & Trigonom.	Gibt die Summe von Potenzen (zur Berechnung von Potenzreihen und dichotomen Wahrscheinlichkeiten) zurück
300	PRODUKT	PRODUCT	Math. & Trigonom.	Multipliziert die Argumente und gibt das Produkt zurück
301	QIKV	MIRR	Finanzmathematik	Gibt einen modifizierten internen Zinsfuß zurück, bei dem positive und negative Cashflows mit unterschiedlichen Zinssätzen finanziert werden
302	QUADRATSUMME	SUMSQ	Math. & Trigonom.	Summiert die quadrierten Argumente
303	QUANTIL	PERCENTILE	Statistik (ab 2010 Kompatibilität)	Gibt das Alpha-Quantil einer Gruppe von Daten zurück
304	QUANTIL.EXKL (2010)	PERCENTILE.EXC	Statistik	Gibt das k -Quantil einer Gruppe von Daten zurück, wobei k ausschließlich im Bereich 0 bis 1 liegt ▶

ID	Deutsch	Englisch	Kategorie	Zweck
305	QUANTIL.INKL (2010)	PERCENTILE.INC	Statistik	Gibt das k -Quantil einer Gruppe von Daten zurück, wobei k einschließlich im Bereich 0 bis 1 liegt
306	QUANTILSRANG	PERCENTRANK	Statistik (ab 2010 Kompatibilität)	Gibt den prozentualen Rang (Alpha) eines Werts zurück
307	QUANTILSRANG.EXKL (2010)	PERCENTRANK.EXC	Statistik	Gibt den prozentualen Rang (Alpha) eines Werts ausschließlich im Bereich 0 bis 1 zurück
308	QUANTILSRANG.INKL (2010)	PERCENTRANK.INC	Statistik	Gibt den prozentualen Rang (Alpha) eines Werts einschließlich im Bereich 0 bis 1 zurück
309	QUARTILE	QUARTILE	Statistik (ab 2010 Kompatibilität)	Gibt die Quartile der Datengruppe zurück
310	QUARTILE.EXKL (2010)	QUARTILE.EXC	Statistik	Gibt die Quartile der Datengruppe zurück, ausschließlich basierend auf Werten von 0 bis 1
311	QUARTILE.INKL (2010)	QUARTILE.INC	Statistik	Gibt die Quartile der Datengruppe zurück, einschließlich basierend auf Werten von 0 bis 1
312	QUOTIENT	QUOTIENT	Math. & Trigonom.	Gibt den ganzzahligen Anteil einer Division zurück
313	RANG	RANK	Statistik (ab 2010 Kompatibilität)	Gibt den Rang zurück, den eine Zahl innerhalb einer Liste von Zahlen einnimmt
314	RANG.GLEICH (2010)	RANK.EQ	Statistik	Siehe RANG
315	RANG.MITTELW (2010)	RANK.AVG	Statistik	Gibt die durchschnittliche Rangzahl für eine Zahl innerhalb einer Liste von Zahlen zurück
316	RECHTS	RIGHT	Text	Gibt das letzte oder die letzten Zeichen einer Textzeichenfolge auf der Grundlage der von Ihnen angegebenen Anzahl von Zeichen zurück
317	RENDITE	YIELD	Finanzmathematik	Gibt die Rendite eines Wertpapiers zurück, das periodisch Zinsen auszahlt
318	RENDITEDIS	YIELDDISC	Finanzmathematik	Gibt die jährliche Rendite eines unverzinslichen Wertpapiers zurück
319	RENDITEFÄLL	YIELDMAT	Finanzmathematik	Gibt die jährliche Rendite eines Wertpapiers zurück, das Zinsen am Fälligkeitsdatum auszahlt
320	REST	MOD	Math. & Trigonom.	Gibt den Rest einer Division zurück. Das Ergebnis hat dasselbe Vorzeichen wie Divisor. ►

ID	Deutsch	Englisch	Kategorie	Zweck
321	RGP	LINEST	Statistik	Berechnet die Statistik für eine Linie unter Verwendung der Methode der kleinsten Quadrate, zur Berechnung einer geraden Linie, die für die Daten am geeignetsten ist, und gibt eine Matrix zurück, die die Linie beschreibt
322	RKP	LOGEST	Statistik	In Regressionsanalysen berechnet diese Funktion eine Exponentialkurve, die möglichst gut an die von Ihnen bereitgestellten Daten angepasst ist, und liefert ein Wertarray, die diese Kurve beschreibt
323	RMZ	PMT	Finanzmathematik	Gibt die konstante Zahlung einer Annuität pro Periode zurück, wobei konstante Zahlungen und ein konstanter Zinssatz vorausgesetzt werden
324	RÖMISCH	ROMAN	Math. & Trigonom.	Wandelt eine arabische Zahl in eine römische Zahl als Text um
325	RTD	RTD	Matrix	Ruft Echtzeitdaten von einem Programm ab, das die COM-Automatisierung unterstützt
326	RUNDEN	ROUND	Math. & Trigonom.	Rundet eine Zahl auf eine bestimmte Anzahl von Dezimalstellen
327	SÄUBERN	CLEAN	Text	Löscht alle nicht druckbaren Zeichen aus einem Text
328	SCHÄTZER	FORECAST	Statistik	Gibt den Schätzwert für einen linearen Trend zurück
329	SCHIEFE	SKEW	Statistik	Gibt die Schiefe einer Verteilung zurück
330	SCHIEFE.P (2013)	SKEW.P	Statistik	Gibt die Schiefe einer Verteilung auf Basis einer Grundgesamtheit zurück
331	SEC (2013)	SEC	Math. & Trigonom.	Gibt den Sekans eines Winkels zurück
332	SECHYP (2013)	SECH	Math. & Trigonom.	Gibt den hyperbolischen Sekans eines Winkels zurück
333	SEKUNDE	SECOND	Datum & Zeit	Wandelt eine fortlaufende Zahl in eine Sekunde um
334	SIN	SIN	Math. & Trigonom.	Gibt den Sinus einer Zahl zurück
335	SINHYP	SINH	Math. & Trigonom.	Gibt den hyperbolischen Sinus einer Zahl zurück
336	SPALTE	COLUMN	Matrix	Gibt die Spaltennummer eines Bezugs zurück
337	SPALTEN	COLUMNS	Matrix	Gibt die Anzahl der Spalten einer Matrix oder eines Bezugs zurück ►

ID	Deutsch	Englisch	Kategorie	Zweck
338	SQL.REQUEST	SQL.REQUEST	Externe Funktionen	Stellt eine Verbindung mit einer externen Datenquelle her und führt eine Abfrage aus, die sich auf einem Arbeitsblatt befindet
339	STABW	STDEV	Statistik (ab 2010 Kompatibilität)	Schätzt die Standardabweichung ausgehend von einer Stichprobe
340	STABW.S (2010)	STDEV.S	Statistik	Siehe STABW
341	STABWA	STDEVA	Statistik	Schätzt die Standardabweichung ausgehend von einer Stichprobe
342	STABWN	STDEVP	Statistik (ab 2010 Kompatibilität)	Berechnet die Standardabweichung ausgehend von der Grundgesamtheit
343	STABW.N (2010)	STDEV.P	Statistik	Siehe STABWN
344	STABWNA	STDEVPA	Statistik	Berechnet die Standardabweichung ausgehend von einer als Argument angegebenen Grundgesamtheit, einschließlich Text und logischer Werte
345	STANDARDISIERUNG	STANDARDIZE	Statistik	Gibt den standardisierten Wert einer Verteilung zurück, die durch <i>Mittelwert</i> und <i>Standabwn</i> charakterisiert ist
346	STANDNORMINV	NORMSINV	Statistik (ab 2010 Kompatibilität)	Gibt Quantile der Standardnormalverteilung zurück
347	STANDNORMVERT	NORMSDIST	Statistik (ab 2010 Kompatibilität)	Gibt Werte der Verteilungsfunktion einer standardnormalverteilten Zufallsvariablen zurück
348	STEIGUNG	SLOPE	Statistik	Gibt die Steigung der Regressionsgeraden zurück, die an die in <i>Y_Werte</i> und <i>X_Werte</i> abgelegten Datenpunkte angepasst ist
349	STFEHLERYX	STEYX	Statistik	Gibt den Standardfehler der geschätzten <i>y</i> -Werte für alle <i>x</i> -Werte der Regression zurück
350	STUNDE	HOUR	Datum & Zeit	Gibt die Stunde einer Zeitangabe zurück
351	SUCHEN	SEARCH	Text	Gibt, beginnend mit <i>Erstes_Zeichen</i> , die Nummer des Zeichens zurück, an der das zu suchende Zeichen oder die zu suchende Textzeichenfolge erstmals gefunden wurde
352	SUMME	SUM	Math. & Trigonom.	Summiert die Argumente
353	SUMMENPRODUKT	SUMPRODUCT	Math. & Trigonom.	Multipliziert die einander entsprechenden Komponenten der angegebenen Matrizen miteinander und gibt die Summe dieser Produkte zurück



ID	Deutsch	Englisch	Kategorie	Zweck
354	SUMMEWENN	SUMIF	Math. & Trigonom.	Addiert Zahlen, die mit dem Suchkriterium übereinstimmen
355	SUMMEWENNS	SUMIFS	Math. & Trigonom.	Addiert Zahlen, die mit mehreren Suchkriterien übereinstimmen
356	SUMMEX2MY2	SUMX2MY2	Math. & Trigonom.	Summiert für zusammengehörige Komponenten zweier Matrizen die Differenzen der Quadrate
357	SUMMEX2PY2	SUMX2PY2	Math. & Trigonom.	Summiert für zusammengehörige Komponenten zweier Matrizen die Summen der Quadrate
358	SUMMEXMY2	SUMXMY2	Math. & Trigonom.	Summiert für zusammengehörige Komponenten zweier Matrizen die quadrierten Differenzen
359	SUMQUADABW	DEVSQ	Statistik	Gibt die Summe der quadrierten Abweichungen von Datenpunkten von deren Stichprobenmittelwert zurück
360	SVERWEIS	VLOOKUP	Matrix	Sucht in der am weitesten links gelegenen Spalte einer Tabelle nach einem Wert und gibt in der gleichen Zeile einen Wert aus einer von Ihnen angegebenen Spalte in der Tabelle zurück
361	T	T	Text	Wandelt die Argumente in Text um
362	TAG	DAY	Datum & Zeit	Gibt den Tag eines Datums als fortlaufende Zahl zurück
363	TAGE	DAYS	Datum & Zeit	Berechnet die Anzahl der Tage zwischen zwei Datumsangaben
364	TAGE360	DAYS360	Datum & Zeit	Berechnet, ausgehend von einem Jahr, das 360 Tage umfasst, die Anzahl der zwischen zwei Tagesdaten liegenden Tage
365	TAN	TAN	Math. & Trigonom.	Gibt den Tangens einer Zahl zurück.
366	TANHYP	TANH	Math. & Trigonom.	Gibt den hyperbolischen Tangens einer Zahl zurück
367	TBILLÄQUIV	TBILLEQ	Finanzmathematik	Rechnet die Verzinsung eines Schatzwechsels (Treasury Bill) in die für Anleihen übliche einfache jährliche Verzinsung um
368	TBILLKURS	TBILLPRICE	Finanzmathematik	Gibt den Kurs pro 100 € Nennwert eines Schatzwechsels (Treasury Bill) zurück
369	TBILLRENDITE	TBILLYIELD	Finanzmathematik	Gibt die Rendite eines Schatzwechsels (Treasury Bill) zurück ►

ID	Deutsch	Englisch	Kategorie	Zweck
370	TEIL	MID	Text	Gibt auf der Grundlage der angegebenen Anzahl von Zeichen eine bestimmte Anzahl von Zeichen einer Zeichenfolge ab der von Ihnen angegebenen Position zurück
371	TEILERGEBNIS	SUBTOTAL	Math. & Trigonom.	Gibt ein Teilergebnis in einer Liste oder Datenbank zurück
372	TEXT	TEXT	Text	Formatiert eine Zahl und wandelt sie in Text um
373	TINV	TINV	Statistik (ab 2010 Kompatibilität)	Gibt den <i>t</i> -Wert der <i>t</i> -Verteilung als eine Funktion der Wahrscheinlichkeit und der Freiheitsgrade zurück
374	T.INV (2010)	T.INV	Statistik	Siehe TINV
375	T.INV.2S (2010)	T.INV.2T	Statistik	Siehe TINV
376	TREND	TREND	Statistik	Liefert Werte, die sich aus einem linearen Trend ergeben
377	TTEST	TTEST	Statistik (ab 2010 Kompatibilität)	Gibt die Teststatistik eines Studentischen <i>t</i> -Tests zurück
378	T.TEST (2010)	T.TEST	Statistik	Siehe TTEST
379	TVERT	TDIST	Statistik (ab 2010 Kompatibilität)	Gibt Werte der Verteilungsfunktion (1-Alpha) einer (Student) <i>t</i> -verteilten Zufallsvariablen zurück
380	T.VERT (2010)	T.DIST	Statistik	Gibt die (Student) <i>t</i> -Verteilung der linken Endfläche zurück
381	T.VERT.2S (2010)	T.DIST.2T	Statistik	Gibt die (Student) <i>t</i> -Verteilung für zwei Endflächen zurück
382	T.VERT.RE (2010)	T.DIST.RT	Statistik	Gibt die (Student) <i>t</i> -Verteilung der rechten Endfläche zurück
383	TYP	TYPE	Information	Gibt eine Zahl zurück, die den Datentyp des angegebenen Werts anzeigt
384	UMWANDELN	CONVERT	Technisch	Wandelt eine Zahl von einem Maßsystem in ein anderes um
385	UND	AND	Logik	Gibt <i>WAHR</i> zurück, wenn alle Argumente <i>WAHR</i> sind. Sind die Aussagen eines oder mehrerer Argumente <i>FALSCH</i> , gibt diese Funktion den Wert <i>FALSCH</i> zurück.
386	UNGERADE	ODD	Math. & Trigonom.	Rundet eine Zahl auf die nächste ungerade ganze Zahl auf
387	UNICODE (2013)	UNICODE	Text	Gibt die Unicodezahl des ersten Zeichens eines Texts zurück
388	UNIZEICHEN (2013)	UNICHAR	Text	Gibt das Unicodezeichen für eine Codezahl zurück



ID	Deutsch	Englisch	Kategorie	Zweck
389	UNREGER.KURS	ODDFPRICE	Finanzmathematik	Liefert den Kurs pro 100 € Nennwert eines Wertpapiers mit einem unregelmäßigen ersten Zinstermin
390	UNREGER.REND	ODDFYIELD	Finanzmathematik	Gibt die Rendite eines Wertpapiers mit einem unregelmäßigen ersten Zinstermin zurück
391	UNREGLE.KURS	ODDLPRICE	Finanzmathematik	Gibt den Kurs pro 100 € Nennwert eines Wertpapiers mit einem unregelmäßigen letzten Zinstermin zurück
392	UNREGLE.REND	ODDLYIELD	Finanzmathematik	Gibt die Rendite eines Wertpapiers mit einem unregelmäßigen letzten Zinstermin zurück
393	UNTERGRENZE	FLOOR	Math. & Trigonom.	Rundet eine Zahl ab
394	UNTERGRENZE.GENAU (2010)	FLOOR.PRECISE	Math. & Trigonom.	Siehe UNTERGRENZE
395	UNTERGRENZE.MATHEMATIK (2013)	FLOOR.MATH	Math. & Trigonom.	Rundet eine Zahl auf die nächste Ganzzahl oder auf das kleinste Vielfache des angegebenen Schritts ab
396	URLCODIEREN (2013)	ENCODEURL	Web	Gibt eine URL-codierte Zeichenfolge zurück
397	VARIANZ	VAR	Statistik (ab 2010 Kompatibilität)	Schätzt die Varianz auf der Basis einer Stichprobe
398	VAR.S (2010)	VAR.S	Statistik	Siehe VARIANZ
399	VARIANZA	VARA	Statistik	Schätzt die Varianz auf der Basis einer Stichprobe
400	VARIANZEN	VARP	Statistik (ab 2010 Kompatibilität)	Berechnet die Varianz ausgehend von der Grundgesamtheit
401	VAR.P (2010)	VAR.P	Statistik	Siehe VARIANZEN
402	VARIANZENA	VARPA	Statistik	Berechnet die Varianz ausgehend von der Grundgesamtheit
403	VARIATION	GROWTH	Statistik	Liefert Werte, die sich aus einem exponentiellen Trend ergeben
404	VARIATIONEN	PERMUT	Statistik	Gibt die Anzahl der Möglichkeiten zurück, um k Elemente aus einer Menge von n Elementen ohne Zurücklegen zu ziehen
405	VARIATIONEN2 (2013)	PERMUTATIONA	Statistik	Gibt die Anzahl der Möglichkeiten – mit Wiederholungen – für eine angegebene Menge von Objekten zurück ▶

ID	Deutsch	Englisch	Kategorie	Zweck
406	VDB	VDB	Finanzmathematik	Gibt die degressive Doppelraten-Abschreibung eines Wirtschaftsguts für eine bestimmte Periode oder Teilperiode zurück
407	VERGLEICH	MATCH	Matrix	Sucht Werte innerhalb eines Bezugs oder einer Matrix
408	VERKETTEN	CONCATENATE	Text	Verknüpft mehrere Textzeichenfolgen zu einer Textzeichenfolge
409	VERWEIS	LOOKUP	Matrix	Durchsucht die Werte eines Vektors oder einer Matrix
410	VORZEICHEN	SIGN	Math. & Trigonom.	Gibt das Vorzeichen einer Zahl zurück
411	VRUNDEN	MROUND	Math. & Trigonom.	Gibt eine auf das gewünschte Vielfache gerundete Zahl zurück
412	WAHL	CHOOSE	Matrix	Verwendet Index, um einen Wert aus der Liste der Werteargumente zurückzugeben
413	WAHR	TRUE	Logik	Gibt den Wahrheitswert <i>WAHR</i> zurück
414	WAHRSCBEREICH	PROB	Statistik	Gibt die Wahrscheinlichkeit für ein von zwei Werten eingeschlossenes Intervall zurück
415	WEBDIENST (2013)	WEBSERVICE	Web	Empfängt (XML-) Daten vom angegebenen Webdienst
416	WECHSELN	SUBSTITUTE	Text	Ersetzt alten Text durch neuen Text in einer Zeichenfolge
417	WEIBULL	WEIBULL	Statistik (ab 2010 Kompatibilität)	Gibt Wahrscheinlichkeiten einer weibullverteilten Zufallsvariablen zurück
418	WEIBULL.VERT (2010)	WEIBULL.DIST	Statistik	Siehe WEIBULL
419	WENN	IF	Logik	Gibt eine Wahrheitsprüfung an, die durchgeführt werden soll
420	WENNFEHLER	IFERROR	Logik	Berechnet eine definierte Ausgabe im Fall, dass ein Ausdruck fehlerhaft ist, und gibt ansonsten das Ergebnis des Ausdrucks aus
421	WENNNV	IFNA	Logik	Berechnet eine definierte Ausgabe im Fall, dass ein Ausdruck zum Fehler #NV führt
422	WERT	VALUE	Text	Wandelt ein als Text angegebenes Argument in eine Zahl um
423	WIEDERHOLEN	REPT	Text	Wiederholt einen Text so oft wie angegeben ▶

ID	Deutsch	Englisch	Kategorie	Zweck
424	WOCHENTAG	WEEKDAY	Datum & Zeit	Wandelt eine fortlaufende Zahl in einen Wochentag um
425	WURZEL	SQRT	Math. & Trigonom.	Gibt die Quadratwurzel einer Zahl zurück
426	WURZELPI	SQRTPI	Math. & Trigonom.	Gibt die Wurzel aus der mit Pi multiplizierten Zahl zurück
427	WVERWEIS	HLOOKUP	Matrix	Sucht in der obersten Zeile einer Tabelle oder einer Matrix nach Werten und gibt dann in der gleichen Spalte einen Wert aus einer Zeile zurück, die Sie in der Tabelle oder Matrix angeben
428	XINTZINSFUSS	XIRR	Finanzmathematik	Gibt den internen Zinsfuß einer Reihe nicht periodisch anfallender Zahlungen zurück
429	XKAPITALWERT	XNPV	Finanzmathematik	Gibt den Nettobarwert (Kapitalwert) einer Reihe nicht periodisch anfallender Zahlungen zurück
430	XMLFILTERN (2013)	FILTERXML	Web	Extrahiert ausgewählte Daten aus einer XML-Datei
431	XODER (2013)	XOR	Logik	Gibt den Wahrheitswert <i>WAHR</i> nur dann zurück, wenn genau eines der Argumente <i>WAHR</i> ist
432	ZÄHLENWENN	COUNTIF	Statistik	Zählt die nicht leeren Zellen eines Bereichs, deren Inhalte mit einem Suchkriterium übereinstimmen
433	ZÄHLENWENNNS	COUNTIFS	Statistik	Die Zellen eines Bereichs, deren Inhalte mit mehreren Kriterien übereinstimmen, werden gezählt
434	ZAHLENWERT (2013)	NUMBERVALUE	Text	Konvertiert unabhängig vom Gebietsschema des PC einen (Zahl-) Text in einen numerischen Wert
435	ZEICHEN	CHAR	Text	Gibt das der Codezahl entsprechende Zeichen zurück
436	ZEILE	ROW	Matrix	Liefert die Zeilennummer eines Bezugs
437	ZEILEN	ROWS	Matrix	Gibt die Anzahl der Zeilen in einem Bezug oder einer Matrix zurück
438	ZEIT	TIME	Datum & Zeit	Gibt die Dezimalzahl einer bestimmten Uhrzeit zurück
439	ZEITWERT	TIMEVALUE	Datum & Zeit	Wandelt eine als Text vorliegende Zeitangabe in eine fortlaufende Zahl um

ID	Deutsch	Englisch	Kategorie	Zweck
440	ZELLE	CELL	Information	Gibt Informationen zur Formatierung, der Position oder dem Inhalt der Zelle links oben in einem Bezug zurück
441	ZINS	RATE	Finanzmathematik	Gibt den Zinssatz einer Annuität pro Periode zurück
442	ZINSSATZ	RECEIVED	Finanzmathematik	Gibt den Zinssatz eines voll investierten Wertpapiers zurück
443	ZINSTERMNZ	COUPNCD	Finanzmathematik	Gibt eine Zahl zurück, die den nächsten Zinstermin nach dem Abrechnungstermin angibt
444	ZINSTERMTAGE	COUPDAYS	Finanzmathematik	Gibt die Anzahl der Tage der Zinsperiode zurück, die den Abrechnungstermin einschließt
445	ZINSTERMTAGNZ	COUPDAYSNC	Finanzmathematik	Gibt die Anzahl der Tage vom Abrechnungstermin bis zum nächsten Zinstermin (Kupontermin) an
446	ZINSTERMTAGVA	COUPDAYBS	Finanzmathematik	Gibt die Anzahl der Tage vom Anfang des Zinstermins bis zum Abrechnungstermin zurück
447	ZINSTERMVZ	COUPPCD	Finanzmathematik	Gibt eine Zahl an, die die letzte Zinszahlung vor dem Abrechnungstermin repräsentiert
448	ZINSTERMZAHL	COUPNUM	Finanzmathematik	Gibt die Anzahl der zwischen dem Abrechnungsdatum und dem Fälligkeitsdatum zahlbaren Zinszahlungen an, und zwar aufgerundet zur nächsten ganzzahligen Zinszahlung
449	ZINSZ	IPMT	Finanzmathematik	Gibt die Zinszahlung einer Investition für die angegebene Periode, ausgehend von regelmäßigen, konstanten Zahlungen und einem konstanten Zinssatz zurück
450	ZSATZINVEST (2013)	RRI	Finanzmathematik	Berechnet den Zinssatz, zu welchem ein Kapital investiert werden muss, damit es einen bestimmten Zukunftswert besitzt
451	ZUFALLSBEREICH	RANDBETWEEN	Math. & Trigonom.	Gibt eine ganze Zufallszahl aus dem festgelegten Bereich zurück
452	ZUFALLSZAHL	RAND	Math. & Trigonom.	Gibt eine Zufallszahl zwischen 0 und 1 zurück
453	ZW	FV	Finanzmathematik	Gibt den zukünftigen Wert (Endwert) einer Investition zurück ►

ID	Deutsch	Englisch	Kategorie	Zweck
454	ZW2	FVSCCHEDULE	Finanzmathematik	Gibt den aufgezinsten Wert des Anfangskapitals für eine Reihe periodisch unterschiedlicher Zinssätze zurück
455	ZWEIFAKULTÄT	FACTDOUBLE	Math. & Trigonom.	Gibt die Fakultät zu Zahl mit der Schrittlänge 2 zurück
456	ZZR	NPER	Finanzmathematik	Gibt die Anzahl der Zahlungsperioden einer Investition zurück, die auf periodischen, gleichbleibenden Zahlungen sowie einem konstanten Zinssatz basiert

Funktionsliste Englisch-Deutsch

ID	Englisch	Deutsch	Kategorie	Zweck
1	ABS	ABS	Math. & Trigonom.	Liefert den Absolutwert einer Zahl
2	ACCRINT	AUFGELZINS	Financial	Liefert die aufgelaufenen Zinsen (Stückzinsen) eines Wertpapiers mit periodischen Zinszahlungen
3	ACCRINTM	AUFGELZINSF	Financial	Liefert die aufgelaufenen Zinsen (Stückzinsen) eines Wertpapiers, die bei Fälligkeit ausgezahlt werden
4	ACOS	ARCCOS	Math. & Trigonom.	Liefert den Arkuskosinus oder umgekehrten Kosinus einer Zahl
5	ACOSH	ARCCOSHYP	Math. & Trigonom.	Liefert den umgekehrten hyperbolischen Kosinus einer Zahl
6	ACOT	ARCCOT (2013)	Math. & Trigonom.	Gibt den Arkuskotangens einer Zahl zurück
7	ACOTH	ARCCOTHYP (2013)	Math. & Trigonom.	Gibt den hyperbolischen Arkuskotangens einer Zahl zurück
8	ADDRESS	ADRESSE	Lookup & Reference	Liefert einen Bezug auf eine Zelle einer Tabelle als Text
9	AGGREGATE	AGGREGAT (2010)	Math. & Trigonom.	Gibt ein Aggregat in einer Liste oder Datenbank zurück
10	AMORDEGRC	AMORDEGRK	Financial	Liefert den für eine Abrechnungsperiode anzusetzenden Abschreibungsbetrag auf Basis des französischen Buchführungssystems
11	AMORLINC	AMORLINEARK	Financial	Gibt den für eine Abrechnungsperiode anzusetzenden Abschreibungsbetrag auf Basis des französischen Buchführungssystems zurück

ID	Englisch	Deutsch	Kategorie	Zweck
12	AND	UND	Logical	Gibt <i>WAHR</i> zurück, wenn alle Argumente <i>WAHR</i> sind. Sind die Aussagen eines oder mehrerer Argumente <i>FALSCH</i> , gibt diese Funktion den Wert <i>FALSCH</i> zurück.
13	ARABIC	ARABISCH (2013)	Math. & Trigonom.	Wandelt eine römische Zahl in eine arabische Zahl um
14	AREAS	BEREICHE	Lookup & Reference	Gibt die Anzahl der innerhalb eines Bezugs aufgeführten Bereiche zurück
15	ASC	ASC	Text & Data	Wandelt bei Sprachen mit einem Double-Byte-Zeichensatz (DBCS) Zeichen voller Breite (Double-Byte-Zeichen) in Zeichen halber Breite (Single-Byte-Zeichen) um
16	ASIN	ARCSIN	Math. & Trigonom.	Gibt den Arkussinus oder auch umgekehrten Sinus einer Zahl zurück
17	ASINH	ARCSINHYP	Math. & Trigonom.	Gibt den umgekehrten hyperbolischen Sinus einer Zahl zurück
18	ATAN	ARCTAN	Math. & Trigonom.	Gibt den Arkustangens oder auch umgekehrten Tangens einer Zahl zurück
19	ATAN2	ARCTAN2	Math. & Trigonom.	Gibt den Arkustangens oder auch umgekehrten Tangens ausgehend von einer x - und einer y -Koordinate zurück
20	ATANH	ARCTANHYP	Math. & Trigonom.	Gibt den umgekehrten hyperbolischen Tangens einer Zahl zurück
21	AVEDEV	MITTELABW	Statistical	Gibt die durchschnittliche absolute Abweichung einer Reihe von Merkmalsausprägungen und ihrem Mittelwert zurück
22	AVERAGE	MITTELWERT	Statistical	Durch das arithmetische Mittel wird der Zahlenwert bestimmt, der sich bei gleichmäßiger Aufteilung der Summen auf die Werte ergibt
23	AVERAGEA	MITTELWERTA	Statistical	Berechnet den Mittelwert (arithmetisches Mittel) der Werte in der Liste der Argumente
24	AVERAGEIF	MITTELWERTWENN	Statistical	Durchschnittswert (arithmetisches Mittel) für alle Zellen in einem Bereich, die einem angegebenen Kriterium entsprechen
25	AVERAGEIFS	MITTELWERTWENNS	Statistical	Gibt den Durchschnittswert (arithmetisches Mittel) aller Zellen zurück, die mehreren Kriterien entsprechen



ID	Englisch	Deutsch	Kategorie	Zweck
26	BAHTTEXT	BAHTTEXT	Text & Data	Wandelt eine Zahl in Thai-Text um und fügt diesem das Suffix »Baht« hinzu
27	BASE	BASIS (2013)	Math. & Trigonom.	Konvertiert eine Zahl in eine Textdarstellung mit der angegebenen Basis
28	BESSELI	BESSELI	Engineering	Gibt die modifizierte Besselfunktion der ersten Art $I_n(z)$ zurück, die der für rein imaginäre Argumente ausgewerteten Besselfunktion J_n entspricht
29	BESSELJ	BESSELJ	Engineering	Gibt die Besselfunktion der ersten Art $J_n(x)$ zurück
30	BESSELK	BESSELK	Engineering	Gibt die modifizierte Besselfunktion 2. Art $K_n(x)$ zurück
31	BESSELY	BESSELY	Engineering	Gibt die Besselfunktion zweiter Art $Y_n(x)$ zurück, die auch als Webersche Funktion oder Neumannsche Funktion bezeichnet wird
32	BETA.DIST	BETA.VERT (2010)	Statistical	Siehe BETADIST
33	BETA.INV	BETA.INV (2010)	Statistical	Siehe BETAINV
34	BETADIST	BETAVERT	Statistical (ab 2010 Compatibility)	Gibt die Werte der kumulierten Betaverteilungsfunktion zurück
35	BETAINV	BETAINV	Statistical (ab 2010 Compatibility)	Gibt das Quantil der angegebenen Betaverteilung zurück
36	BIN2DEC	BININDEZ	Engineering	Wandelt eine binäre Zahl (Dualzahl) in eine dezimale Zahl um
37	BIN2HEX	BININHEX	Engineering	Wandelt eine binäre Zahl (Dualzahl) in eine hexadezimale Zahl um
38	BIN2OCT	BININOKT	Engineering	Wandelt eine Binärzahl (Dualzahl) in eine Oktalzahl um
39	BINOM.DIST	BINOM.VERT (2010)	Statistical	Siehe BINOMDIST
40	BINOM.DIST.RANGE	BINOM.VERT.BEREICH (2013)	Statistical	Gibt die Erfolgswahrscheinlichkeit eines Versuchsergebnisses als Binomialverteilung zurück
41	BINOM.INV	BINOM.INV (2010)	Statistical	Gibt den kleinsten Wert, für den die kumulierten Wahrscheinlichkeiten der Binomialverteilung größer oder gleich einer Grenzwahrscheinlichkeit sind, zurück
42	BINOMDIST	BINOMVERT	Statistical (ab 2010 Compatibility)	Gibt Wahrscheinlichkeiten einer binomialverteilten Zufallsvariablen zurück
43	BITAND	BITUND (2013)	Engineering	Gibt ein bitweises »Und« zweier Zahlen zurück

ID	Englisch	Deutsch	Kategorie	Zweck
44	BITLSHIFT	BITLVERSCHIEB (2013)	Engineering	Gibt eine Zahl zurück, die um Verschiebebetrag Bits nach links verschoben ist
45	BITOR	BITODER (2013)	Engineering	Gibt ein bitweises »Oder« zweier Zahlen zurück
46	BITRSHIFT	BITRVERSCHIEB (2013)	Engineering	Gibt eine Zahl zurück, die um <i>Verschiebebetrag</i> Bits nach rechts verschoben ist
47	BITXOR	BITXODER (2013)	Engineering	Gibt ein bitweises »Ausschließliches Oder« zweier Zahlen zurück
48	CEILING	OBERGRENZE	Math. & Trigonom.	Rundet eine Zahl betragsmäßig auf das kleinste Vielfache von Schritt auf
49	CEILING.MATH	OBERGRENZE.MATHEMATIK (2013)	Math. & Trigonom.	Rundet eine Zahl auf die nächste Ganzzahl oder auf das kleinste Vielfache des angegebenen Schritts auf
50	CEILING.PRECISE	OBERGRENZE.GENAU (2010)	Math. & Trigonom.	Siehe CEILING
51	CELL	ZELLE	Information	Gibt Informationen zur Formatierung, der Position oder dem Inhalt der Zelle links oben in einem Bezug zurück
52	CHAR	ZEICHEN	Text & Data	Gibt das der Codezahl entsprechende Zeichen zurück
53	CHIDIST	CHIVERT	Statistical (ab 2010 Compatibility)	Gibt Werte der Verteilungsfunktion (1-Alpha) einer Chi-Quadrat-verteilten Zufallsgröße zurück
54	CHIINV	CHIINV	Statistical (ab 2010 Compatibility)	Gibt Quantile der χ^2 -Verteilung zurück
55	CHISQ.DIST	CHIQU.VERT (2010)	Statistical	Gibt Werte der linksseitigen Verteilungsfunktion (1-Alpha) einer Chi-Quadrat-verteilten Zufallsgröße zurück
56	CHISQ.DIST.RT	CHIQU.VERT.RE (2010)	Statistical	Gibt Werte der rechtsseitigen Verteilungsfunktion (1-Alpha) einer Chi-Quadrat-verteilten Zufallsgröße zurück
57	CHISQ.INV	CHIQU.INV (2010)	Statistical	Gibt Quantile der linksseitigen χ^2 -Verteilung zurück
58	CHISQ.INV.RE	CHIQU.INV.RE (2010)	Statistical	Gibt Quantile der rechtsseitigen χ^2 -Verteilung zurück
59	CHISQ.TEST	CHIQU.TEST (2010)	Statistical	Siehe CHITEST
60	CHITEST	CHITEST	Statistical (ab 2010 Compatibility)	Gibt die Teststatistik eines χ^2 -Unabhängigkeitstests zurück ▶

ID	Englisch	Deutsch	Kategorie	Zweck
61	CHOOSE	WAHL	Lookup & Reference	Verwendet Index, um einen Wert aus der Liste der Werteargumente zurückzugeben
62	CLEAN	SÄUBERN	Text & Data	Löscht alle nicht druckbaren Zeichen aus einem Text
61	CODE	CODE	Text & Data	Gibt die Codezahl des ersten Zeichens in einem Text zurück
64	COLUMN	SPALTE	Lookup & Reference	Gibt die Spaltennummer eines Bezugs zurück
65	COLUMNS	SPALTEN	Lookup & Reference	Gibt die Anzahl der Spalten einer Matrix oder eines Bezugs zurück
66	COMBIN	KOMBINATIONEN	Math. & Trigonom.	Berechnet, wie viele Gruppen aus einer bestimmten Anzahl von Elementen gebildet werden können
67	COMBINA	KOMBINATIONEN2 (2013)	Math. & Trigonom.	Gibt die Anzahl der Kombinationen mit Wiederholung für eine angegebene Anzahl von Elementen zurück
68	COMPLEX	KOMPLEXE	Engineering	Wandelt den Real- und Imaginärteil in eine komplexe Zahl um ($x + yi$ oder $x + yj$)
69	CONCATENATE	VERKETTEN	Text & Data	Verknüpft mehrere Textzeichenfolgen zu einer Textzeichenfolge
70	CONFIDENCE	KONFIDENZ	Statistical (ab 2010 Compatibility)	Gibt einen Wert zurück, den Sie zum Erstellen eines Konfidenzintervalls für den Erwartungswert einer Zufallsvariablen verwenden können
71	CONFIDENCE.NORM	KONFIDENZ.NORM (2010)	Statistical	Siehe CONFIDENCE
72	CONFIDENCE.T	KONFIDENZ.T (2010)	Statistical	Gibt einen Wert zurück, den Sie zum Erstellen eines Konfidenzintervalls für den Erwartungswert einer t-verteilten Zufallsvariablen verwenden können
73	CONVERT	UMWANDELN	Engineering	Wandelt eine Zahl von einem Maßsystem in ein anderes um
74	CORREL	KORREL	Statistical	Gibt den Korrelationskoeffizienten einer zweidimensionalen Zufallsgröße zurück, deren Werte in den Zellbereichen <i>Matrix1</i> und <i>Matrix2</i> stehen
75	COS	COS	Math. & Trigonom.	Gibt den Kosinus einer Zahl zurück
76	COSH	COSHYP	Math. & Trigonom.	Gibt den hyperbolischen Kosinus einer Zahl zurück

ID	Englisch	Deutsch	Kategorie	Zweck
77	COT	COT (2013)	Math. & Trigonom.	Gibt den Kotangens eines Winkels zurück
78	COTH	COTHYP (2013)	Math. & Trigonom.	Gibt den hyperbolischen Kotangens einer Zahl zurück
79	COUNT	ANZAHL	Statistical	Berechnet, wie viele Zahlen eine Liste von Argumenten enthält
80	COUNTA	ANZAHL2	Statistical	Berechnet, wie viele Werte eine Liste von Argumenten enthält
81	COUNTBLANK	ANZAHLLEEREZELLEN	Statistical	Zählt die leeren Zellen in einem Zellbereich
82	COUNTIF	ZÄHLENWENN	Statistical	Zählt die nicht leeren Zellen eines Bereichs, deren Inhalte mit einem Suchkriterium übereinstimmen
83	COUNTIFS	ZÄHLENWENNNS	Statistical	Die Zellen eines Bereichs, deren Inhalte mit mehreren Kriterien übereinstimmen, werden gezählt
84	COUPDAYBS	ZINSTERMTAGVA	Financial	Gibt die Anzahl der Tage vom Anfang des Zinstermins bis zum Abrechnungstermin zurück
85	COUPDAYS	ZINSTERMTAGE	Financial	Gibt die Anzahl der Tage der Zinsperiode zurück, die den Abrechnungstermin einschließt
86	COUPDAYSNC	ZINSTERMTAGNZ	Financial	Gibt die Anzahl der Tage vom Abrechnungstermin bis zum nächsten Zinstermin (Kupontermin) an
87	COUPNCD	ZINSTERMNZ	Financial	Gibt eine Zahl zurück, die den nächsten Zinstermin nach dem Abrechnungstermin angibt
88	COUPNUM	ZINSTERMZAHL	Financial	Gibt die Anzahl der zwischen dem Abrechnungsdatum und dem Fälligkeitsdatum zahlbaren Zinszahlungen an, und zwar aufgerundet zur nächsten ganzzahligen Zinszahlung
89	COUPPCD	ZINSTERMVZ	Financial	Gibt eine Zahl an, die die letzte Zinszahlung vor dem Abrechnungstermin repräsentiert
90	COVAR	KOVAR	Statistical (ab 2010 Compatibility)	Diese Funktion gibt die Kovarianz zurück, den Mittelwert der für alle Datenpunktpaare gebildeten Produkte der Abweichungen
91	COVARIANCE.P	KOVARIANZ.P (2010)	Statistical	Siehe COVAR
92	COVARIANCE.S	KOVARIANZ.T (2010)	Statistical	Diese Funktion gibt die Kovarianz einer Stichprobe zurück, den Mittelwert der für alle Datenpunktpaare gebildeten Produkte der Abweichungen



ID	Englisch	Deutsch	Kategorie	Zweck
93	CRITBINOM	KRITBINOM	Statistical	Gibt den kleinsten Wert zurück, für den die kumulierten Wahrscheinlichkeiten der Binomialverteilung größer oder gleich einer Grenzwahrscheinlichkeit sind
94	CSC	COSEC (2013)	Math. & Trigonom.	Gibt den Kosekans eines Winkels zurück
95	CSCH	COSECHYP (2013)	Math. & Trigonom.	Gibt den hyperbolischen Kosekans eines Winkels zurück
96	CUBEKPIMEMBER	CUBEKPIELEMENT	Cube	Gibt die angeforderte Eigenschaft von Key Performance-Indikatoren eines Würfels zurück
97	CUBEMEMBER	CUBEELEMENT	Cube	Gibt für den angegebenen Dimensionspfad das Dimensionselement zurück
98	CUBEMEMBER-PROPERTY	CUBEELEMENT-EIGENSCHAFT	Cube	Gibt für ein Dimensionselement die geforderte Eigenschaft, auch Attribut genannt, zurück
99	CUBERANKED-MEMBER	CUBERANGELEMENT	Cube	Gibt das per Index definierte Element eines Sets zurück
100	CUBESET	CUBEMENGE	Cube	Gibt eine definierte Menge von Elementen eines Cubes zurück. Dieses sog. Set stellt einen Teilwürfel dar.
101	CUBESETCOUNT	CUBEMENGENANZAHL	Cube	Gibt die Anzahl der Elemente eines Sets zurück
102	CUBEVALUE	CUBEWERT	Cube	Gibt den Wert für eine gegebene Adresse aus einem Datenwürfel zurück
103	CUMIPMT	KUMZINSZ	Financial	Berechnet die kumulierten Zinsen, die zwischen zwei Perioden zu zahlen sind
104	CUMPRINC	KUMKAPITAL	Financial	Berechnet die aufgelaufene Tilgung eines Darlehens, die zwischen zwei Perioden zu zahlen ist
105	DATE	DATUM	Date & Time	Gibt die fortlaufende Zahl zurück, die ein bestimmtes Datum darstellt
106	DATEDIF	DATEDIF	Date & Time	Ermittelt die Differenz zwischen <i>Anfangs-</i> und <i>Enddatum</i> in Jahren, Monaten oder Tagen
107	DATEVALUE	DATWERT	Date & Time	Wandelt ein als Text vorliegendes Datum in eine fortlaufende Zahl um
108	DAVERAGE	DBMITTELWERT	Database	Liefert den Mittelwert aus den Werten einer Listen- oder Datenbankspalte, die den von Ihnen angegebenen Bedingungen entsprechen

ID	Englisch	Deutsch	Kategorie	Zweck
109	DAY	TAG	Date & Time	Gibt den Tag eines Datums als fortlaufende Zahl zurück
110	DAYS	TAGE	Date & Time	Berechnet die Anzahl der Tage zwischen zwei Datumsangaben
111	DAYS360	TAGE360	Date & Time	Berechnet, ausgehend von einem Jahr, das 360 Tage umfasst, die Anzahl der zwischen zwei Tagesdaten liegenden Tage
112	DB	GDA2	Financial	Gibt die geometrisch-degressive Abschreibung eines Wirtschaftsguts für eine bestimmte Periode zurück
113	DCOUNT	DBANZAHL	Database	Zählt die Anzahl von Zellen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
114	DCOUNTA	DBANZAHL2	Database	Zählt die Anzahl von Zellen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
115	DDB	GDA	Financial	Gibt die Abschreibung eines Anlageguts für einen angegebenen Zeitraum unter Verwendung der degressiven Doppelratenabschreibung oder eines anderen von Ihnen angegebenen Abschreibungsverfahrens zurück
116	DEC2BIN	DEZINBIN	Engineering	Wandelt eine dezimale Zahl in eine binäre Zahl (Dualzahl) um
117	DEC2HEX	DEZINHEX	Engineering	Wandelt eine dezimale Zahl in eine hexadezimale Zahl um
118	DEC2OCT	DEZINOKT	Engineering	Wandelt eine Dezimalzahl in eine Oktalzahl um
119	DECIMAL	DEZIMAL (2013)	Math. & Trigonom.	Konvertiert eine Textdarstellung einer Zahl mit einer angegebenen Basis in eine Dezimalzahl
120	DEGREES	GRAD	Math. & Trigonom.	Wandelt ein Bogenmaß (Radiant) in Grad um
121	DELTA	DELTA	Engineering	Überprüft, ob zwei Werte gleich sind
122	DEVSQ	SUMQUADABW	Statistical	Gibt die Summe der quadrierten Abweichungen von Datenpunkten von deren Stichprobenmittelwert zurück



ID	Englisch	Deutsch	Kategorie	Zweck
123	DGET	DBAUSZUG	Database	Extrahiert einen einzelnen Wert in einer Spalte einer Liste oder Datenbank, der den angegebenen Bedingungen entspricht
124	DISC	DISAGIO	Financial	Gibt den in Prozent ausgedrückten Abzinsungssatz eines Wertpapiers zurück
125	DMAX	DBMAX	Database	Liefert die größte Anzahl in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entspricht
126	DMIN	DBMIN	Database	Liefert die kleinste Anzahl in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entspricht
127	DOLLAR	DM	Text & Data	Konvertiert eine Zahl in ein Textformat und ordnet ein Währungssymbol zu
128	DOLLARDE	NOTIERUNGDEZ	Financial	Wandelt eine Notierung, die als Dezimalbruch ausgedrückt wurde, in eine Dezimalzahl um
129	DOLLARFR	NOTIERUNGBRU	Financial	Wandelt eine Notierung, die als Dezimalzahl ausgedrückt wurde, in einen Dezimalbruch um
130	DPRODUCT	DBPRODUKT	Database	Multipliziert die Werte in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
131	DSTDEV	DBSTDABW	Database	Schätzt die Standardabweichung einer Grundgesamtheit ausgehend von den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
132	DSTDEVP	DBSTDABWN	Database	Berechnet die Standardabweichung einer vollständigen Grundgesamtheit ausgehend von den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
133	DSUM	DBSUMME	Database	Fügt die Zahlen in einer Spalte aus einer Liste oder Datenbank hinzu, die den angegebenen Bedingungen entsprechen
134	DURATION	DURATION	Financial	Gibt für einen angenommenen Nennwert von 100 € die Macauley-Dauer zurück ▶

ID	Englisch	Deutsch	Kategorie	Zweck
135	DVAR	DBVARIANZ	Database	Schätzt die Varianz einer Grundgesamtheit ausgehend von einer Stichprobe mit den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
136	DVARP	DBVARIANZEN	Database	Berechnet die Varianz einer vollständigen Grundgesamtheit ausgehend von den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
137	EDATE	EDATUM	Date & Time	Gibt die fortlaufende Zahl des Datums zurück, das eine bestimmte Anzahl von Monaten vor bzw. nach dem angegebenen Datum (Ausgangsdatum) liegt
138	EFFECT	EFFEKTIV	Financial	Gibt die jährliche Effektivverzinsung zurück, ausgehend von einer Nominalverzinsung sowie der jeweiligen Anzahl der Zinszahlungen pro Jahr
139	ENCODEURL	URLCODIEREN (2013)	Web	Gibt eine URL-codierte Zeichenfolge zurück
140	EOMONTH	MONATSENDE	Date & Time	Gibt die fortlaufende Zahl des letzten Tages des Monats zurück, der eine bestimmte Anzahl von Monaten vor bzw. nach dem Ausgangsdatum liegt
141	ERF	GAUSSFEHLER	Engineering	Gibt die Gaußsche Fehlerfunktion zurück
142	ERF.PRECISE	GAUSSF.GENAU (2010)	Engineering	Siehe ERF
143	ERFC	GAUSSFKOMPL	Engineering	Gibt das Komplement zur Gaußschen Fehlerfunktion zurück
144	ERFC.PRECISE	GAUSSFKOMPL.GENAU (2010)	Engineering	Siehe ERFC
145	ERROR.TYPE	FEHLER.TYP	Information	Gibt eine Zahl zurück, die einem der Fehlerwerte in Microsoft Excel entspricht, oder den Fehlerwert #NV, wenn kein Fehler vorhanden ist
146	EVEN	GERADE	Math. & Trigonom.	Rundet eine Zahl auf die nächste gerade ganze Zahl auf
147	EXACT	IDENTISCH	Text & Data	Prüft, ob zwei Zeichenfolgen identisch sind



ID	Englisch	Deutsch	Kategorie	Zweck
148	EXP	EXP	Math. & Trigonom.	Potenziert die Basis e mit der als Argument angegebenen Zahl. Die Konstante e ist die Basis des natürlichen Logarithmus und hat den Wert $2,71828182845904$.
149	EXPON.DIST	EXPON.VERT (2010)	Statistical	Siehe EXPONDIST
150	EXPONDIST	EXPONVERT	Statistical (ab 2010 Compatibility)	Gibt Wahrscheinlichkeiten einer exponential verteilten Zufallsvariablen zurück
151	F.DIST	F.VERT (2010)	Statistical	Gibt Werte der Verteilungsfunktion (1-Alpha) einer linksseitigen F-verteilten Zufallsvariablen zurück
152	F.DIST.RT	F.VERT.RE (2010)	Statistical	Gibt Werte der Verteilungsfunktion (1-Alpha) einer rechtsseitigen F-verteilten Zufallsvariablen zurück
153	F.INV	F.INV (2010)	Statistical	Gibt Quantile der linksseitigen F-Verteilung zurück
154	F.INV.RT	F.INV.RE (2010)	Statistical	Gibt Quantile der rechtsseitigen F-Verteilung zurück
155	F.TEST	F.TEST (2010)	Statistical	Siehe FTEST
156	FACT	FAKULTÄT	Math. & Trigonom.	Gibt die Fakultät einer Zahl zurück. Die Fakultät einer Zahl wird aus $1*2*3*... * Zahl$ berechnet.
157	FACTDOUBLE	ZWEIFAKULTÄT	Math. & Trigonom.	Gibt die Fakultät zu Zahl mit der Schrittlänge 2 zurück
158	FALSE	FALSCH	Logical	Gibt den Wahrheitswert <i>FALSCH</i> zurück
159	FDIST	FVERT	Statistical (ab 2010 Compatibility)	Gibt Werte der Verteilungsfunktion (1-Alpha) einer F-verteilten Zufallsvariablen zurück
160	FILTERXML	XMLFILTERN (2013)	Web	Extrahiert ausgewählte Daten aus einer XML-Datei
161	FIND	FINDEN	Text & Data	Sucht eine Zeichenfolge (<i>Suchtext</i>) innerhalb einer anderen Zeichenfolge (<i>Text</i>) und gibt als Ergebnis die Nummer der Anfangsposition von <i>Suchtext</i> ab dem ersten Zeichen von <i>Text</i> zurück
162	FINDB	FINDENB	Text & Data	Sucht eine Zeichenfolge (<i>Suchtext</i>) innerhalb einer anderen Zeichenfolge (<i>Text</i>) und gibt als Ergebnis die Nummer der Anfangsposition von <i>Suchtext</i> ab dem ersten Zeichen von <i>Text</i> auf der Grundlage der Anzahl von Bytes zurück, die jedes Zeichen verwendet ►

ID	Englisch	Deutsch	Kategorie	Zweck
163	FINV	FINV	Statistical (ab 2010 Compatibility)	Gibt Quantile der F-Verteilung zurück
164	FISHER	FISHER	Statistical	Gibt die Fisher-Transformation für x zurück
165	FISHERINV	FISHERINV	Statistical	Gibt die Umkehrung der Fisher-Transformation zurück
166	FIXED	FEST	Text & Data	Formatiert eine Zahl als Text mit einer festen Anzahl von Nachkommastellen
167	FLOOR	UNTERGRENZE	Math. & Trigonom.	Rundet eine Zahl ab
168	FLOOR.MATH	UNTERGRENZE.MATHEMATIK (2013)	Math. & Trigonom.	Rundet eine Zahl auf die nächste Ganzzahl oder auf das kleinste Vielfache des angegebenen Schritts ab
169	FLOOR.PRECISE	UNTERGRENZE.GENAU (2010)	Math. & Trigonom.	Siehe FLOOR
170	FORECAST	SCHÄTZER	Statistical	Gibt den Schätzwert für einen linearen Trend zurück
171	FORMULATEXT	FORMELTEXT (2013)	Lookup & Reference	Liefert die Formel, die im gegebenen Bereich eingetragen wurde, als Text
172	FREQUENCY	HÄUFIGKEIT	Statistical	Gibt eine Häufigkeitsverteilung als einspaltige Matrix zurück
173	FTEST	FTEST	Statistical (ab 2010 Compatibility)	Gibt die Teststatistik eines F-Tests zurück
174	FV	ZW	Financial	Gibt den zukünftigen Wert (Endwert) einer Investition zurück
175	FVSCHEDULE	ZW2	Financial	Gibt den aufgezinsten Wert des Anfangskapitals für eine Reihe periodisch unterschiedlicher Zinssätze zurück
176	GAMMA	GAMMA (2013)	Statistical	Gibt den Wert der Gammafunktion für eine bestimmte Zahl zurück
177	GAMMA.DIST	GAMMA.VERT (2010)	Statistical	Siehe GAMMADIST
178	GAMMA.INV	GAMMA.INV (2010)	Statistical	Siehe GAMMAINV
179	GAMMADIST	GAMMAVERT	Statistical (ab 2010 Compatibility)	Gibt Wahrscheinlichkeiten einer gammaverteilten Zufallsvariablen zurück
180	GAMMAINV	GAMMAINV	Statistical (ab 2010 Compatibility)	Gibt Quantile der Gammaverteilung zurück
181	GAMMALN	GAMMALN	Statistical	Gibt den natürlichen Logarithmus der Gammafunktion zurück, $\Gamma(x)$
182	GAMMALN.PRECISE	GAMMALN.GENAU (2010)	Statistical	Siehe GAMMALN



ID	Englisch	Deutsch	Kategorie	Zweck
183	GAUSS	GAUSS (2013)	Statistical	Gibt die Wahrscheinlichkeit, mit der eine standardnormal-verteilte Zufallsvariable zwischen dem Mittelwert der Grundgesamtheit und x Standardabweichungen vom Mittelwert liegt, zurück.
184	GCD	GGT	Math. & Trigonom.	Gibt den größten gemeinsamen Teiler zurück
185	GEOMEAN	GEOMITTEL	Statistical	Gibt das geometrische Mittel einer Menge positiver Zahlen zurück
186	GESTEP	GGANZZAHL	Engineering	Gibt den Wert 1 zurück, wenn <i>Zahl</i> \geq <i>Schritt</i> gilt; andernfalls gibt sie 0 (Null) zurück
187	GETPIVOTDATA	PIVOTDATEN-ZUORDNEN	Database	Gibt Daten aus einem PivotTable-Bericht zurück
188	GROWTH	VARIATION	Statistical	Liefert Werte, die sich aus einem exponentiellen Trend ergeben
189	HARMEAN	HARMITTEL	Statistical	Gibt das harmonische Mittel einer Datenmenge zurück
190	HEX2BIN	HEXINBIN	Engineering	Wandelt eine hexadezimale Zahl in eine Binärzahl um
191	HEX2DEC	HEXINDEZ	Engineering	Wandelt eine hexadezimale Zahl in eine dezimale Zahl um
192	HEX2OCT	HEXINOKT	Engineering	Wandelt eine hexadezimale Zahl in eine Oktalzahl um
193	HLOOKUP	WVERWEIS	Lookup & Reference	Sucht in der obersten Zeile einer Tabelle oder einer Matrix nach Werten und gibt dann in der gleichen Spalte einen Wert aus einer Zeile zurück, die Sie in der Tabelle oder Matrix angeben
194	hour	STUNDE	Date & Time	Gibt die Stunde einer Zeitangabe zurück
195	HYPERLINK	HYPERLINK	Lookup & Reference	Erstellt eine Verknüpfung oder einen Sprung, über die ein auf einem Netzwerkserver, in einem Intranet oder im Internet gespeichertes Dokument geöffnet wird
196	HYPGEOM.DIST	HYPGEOM.VERT (2010)	Statistical	Gibt Wahrscheinlichkeiten einer hypergeometrisch-verteilten Zufallsvariablen zurück
197	HYPGEOMDIST	HYPGEOMVERT	Statistical (ab 2010 Compatibility)	Gibt Wahrscheinlichkeiten einer hypergeometrisch-verteilten Zufallsvariablen zurück
198	IF	WENN	Logical	Gibt eine Wahrheitsprüfung an, die durchgeführt werden soll ►

ID	Englisch	Deutsch	Kategorie	Zweck
199	IFERROR	WENNFEHLER	Logical	Berechnet eine definierte Ausgabe im Fall, dass ein Ausdruck fehlerhaft ist, und gibt ansonsten das Ergebnis des Ausdrucks aus
200	IFNA	WENNNV	Logical	Berechnet eine definierte Ausgabe im Fall, dass ein Ausdruck zum Fehler #NV führt
201	IMABS	IMABS	Engineering	Gibt den Absolutwert (Modul) einer komplexen Zahl zurück
202	IMAGINARY	IMAGINÄRTEIL	Engineering	Gibt den Imaginärteil einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt
203	IMARGUMENT	IMARGUMENT	Engineering	Gibt das Argument (Theta) zurück, einen Winkel, der als Bogenmaß ausgedrückt wird
204	IMCONJUGATE	IMKONJUGIERTE	Engineering	Gibt die konjugiert komplexe Zahl zu einer komplexen Zahl zurück, wobei die komplexe Zahl als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
205	IMCOS	IMCOS	Engineering	Gibt den Kosinus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt
206	IMCOSH	IMCOSHYP (2013)	Engineering	Gibt den hyperbolischen Kosinus einer komplexen Zahl zurück
207	IMCOT	IMCOT (2013)	Engineering	Gibt den Kotangens einer komplexen Zahl zurück
208	IMCSC	IMCOSEC (2013)	Engineering	Gibt den Kosekans einer komplexen Zahl zurück
209	IMCSCH	IMCOSECHYP (2013)	Engineering	Gibt den hyperbolischen Kosekans einer komplexen Zahl zurück
210	IMDIV	IMDIV	Engineering	Gibt den Quotienten zweier komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
211	IMEXP	IMEXP	Engineering	Gibt die algebraische Form einer in exponentieller Form vorliegenden komplexen Zahl zurück, wobei deren Exponent als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
212	IMLN	IMLN	Engineering	Gibt den natürlichen Logarithmus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird ►

ID	Englisch	Deutsch	Kategorie	Zweck
213	IMLOG10	IMLOG10	Engineering	Gibt den Logarithmus einer komplexen Zahl zur Basis 10 zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
214	IMLOG2	IMLOG2	Engineering	Gibt den Logarithmus einer komplexen Zahl zur Basis 2 zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
215	IMPOWER	IMAPOTENZ	Engineering	Potenziiert eine komplexe Zahl, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt, mit einer ganzen Zahl
216	IMPRODUCT	IMPRODUKT	Engineering	Gibt das Produkt komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
217	IMREAL	IMREALTEIL	Engineering	Gibt den Realteil einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
218	IMSEC	IMSEC (2013)	Engineering	Gibt den Sekans einer komplexen Zahl zurück
219	IMSECH	IMSECHYP (2013)	Engineering	Gibt den hyperbolischen Sekans einer komplexen Zahl zurück
220	IMSIN	IMSIN	Engineering	Diese Funktion gibt den Sinus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
221	IMSINH	IMSINHYP (2013)	Engineering	Gibt den hyperbolischen Sinus einer komplexen Zahl zurück
222	IMSQRT	IMWURZEL	Engineering	Gibt die Quadratwurzel einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
223	IMSUB	IMSUB	Engineering	Gibt die Differenz zweier komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
224	IMSUM	IMSUMME	Engineering	Gibt die Summe komplexer Zahlen zurück, die als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
225	IMTAN	IMTAN (2013)	Engineering	Gibt den Tangens einer komplexen Zahl zurück
226	INDEX	INDEX	Lookup & Reference	Verwendet einen Index, um aus einem Bezug oder einer Matrix einen Wert zu wählen

ID	Englisch	Deutsch	Kategorie	Zweck
227	INDIRECT	INDIREKT	Lookup & Reference	Gibt den Bezug eines Textwerts zurück. Bezüge werden sofort ausgewertet, sodass die zu ihnen gehörenden Werte angezeigt werden.
228	INFO	INFO	Information	Gibt Informationen zu der aktuellen Betriebssystemumgebung zurück
229	INT	GANZZAHL	Math. & Trigonom.	Rundet eine Zahl auf die nächste kleinere ganze Zahl ab
230	INTERCEPT	ACHSENABSCHNITT	Statistical	Gibt den Schnittpunkt der Regressionsgeraden zurück
231	INTRATE	AUSZAHLUNG	Financial	Gibt den Auszahlungsbetrag eines voll investierten Wertpapiers am Fälligkeitstermin zurück
232	IPMT	ZINSZ	Financial	Gibt die Zinszahlung einer Investition für die angegebene Periode, ausgehend von regelmäßigen, konstanten Zahlungen und einem konstanten Zinssatz zurück
233	IRR	IKV	Financial	Gibt den internen Zinsfuß einer Investition ohne Finanzierungskosten oder Reinvestitionsgewinne zurück
234	ISBLANK	ISTLEER	Information	Gibt <i>WAHR</i> zurück, wenn der Wert sich auf eine leere Zelle bezieht, oder <i>FALSCH</i> , wenn der Wert sich auf eine nicht leere Zelle bezieht
235	ISERR	ISTFEHL	Information	Gibt <i>WAHR</i> zurück, wenn der Wert einen Fehler ergibt, oder <i>FALSCH</i> , wenn der Wert keinen Fehler ergibt
236	ISERROR	ISTFEHLER	Information	Gibt <i>WAHR</i> zurück, wenn der Wert einen beliebigen Fehler ergibt, oder <i>FALSCH</i> , wenn der Wert keinen Fehler ergibt
237	ISEVEN	ISTGERADE	Information	Gibt <i>WAHR</i> zurück, wenn die Zahl gerade ist, oder <i>FALSCH</i> , wenn die Zahl ungerade ist
238	ISFORMULA	ISTFORMEL (2013)	Information	Gibt <i>WAHR</i> zurück, wenn ein Bezug auf eine existierende Zelle verweist und sich dort eine Formel befindet
239	ISLOGICAL	ISTLOG	Information	Gibt <i>WAHR</i> zurück, wenn der Wert sich auf einen Wahrheitswert bezieht, oder <i>FALSCH</i> , wenn der Wert sich nicht auf einen Wahrheitswert bezieht



ID	Englisch	Deutsch	Kategorie	Zweck
240	ISNA	ISTNV	Information	Gibt <i>WAHR</i> zurück, wenn der Wert sich auf den Fehlerwert <i>#NV</i> bezieht, oder <i>FALSCH</i> , wenn der Wert sich nicht auf den Fehlerwert <i>#NV</i> bezieht
241	ISNONTEXT	ISTKTEXT	Information	Gibt <i>WAHR</i> zurück, wenn der Wert kein Text ist, oder <i>FALSCH</i> , wenn der Wert ein Text ist
242	ISNUMBER	ISTZAHL	Information	Gibt <i>WAHR</i> zurück, wenn der Wert eine Zahl (numerischer Wert) ist, oder <i>FALSCH</i> , wenn der Wert keine Zahl ist
243	ISODD	ISTUNGERADE	Information	Gibt <i>WAHR</i> zurück, wenn die Zahl ungerade ist, oder <i>FALSCH</i> , wenn die Zahl gerade ist
244	ISOWEEKNUM	ISOKALENDERWOCHE (2013)	Date & Time	Berechnet die Kalenderwochennummer nach ISO-Norm für einen Datumswert
245	ISPMT	ISPMT	Financial	Berechnet die während eines bestimmten Zeitraums für eine Investition gezahlten Zinsen
246	ISREF	ISTBEZUG	Information	Gibt <i>WAHR</i> zurück, wenn der Wert ein gültiger Zellbezug ist, oder <i>FALSCH</i> , wenn der Wert ein ungültiger Zellbezug ist
247	ISTEXT	ISTTEXT	Information	Gibt <i>WAHR</i> zurück, wenn der Wert ein Text ist, oder <i>FALSCH</i> , wenn der Wert kein Text ist. Beachten Sie, dass diese Funktion <i>FALSCH</i> zurückgibt, wenn sich der Wert auf eine leere Zelle bezieht.
248	KURT	KURT	Statistical	Gibt die Kurtosis (Exzess) einer Datengruppe zurück
249	LARGE	KGRÖSSTE	Statistical	Gibt den <i>k</i> -größten Wert einer Datengruppe zurück
250	LCM	KGV	Math. & Trigonom.	Gibt das kleinste gemeinsame Vielfache der als Argumente angegebenen ganzen Zahlen zurück
251	LEFT	LINKS	Text & Data	Gibt auf der Grundlage der Anzahl von Zeichen, die Sie angeben, das oder die erste(n) Zeichen in einer Textzeichenfolge zurück
252	LEN	LÄNGE	Text & Data	Gibt die Anzahl der Zeichen einer Zeichenfolge zurück ►

ID	Englisch	Deutsch	Kategorie	Zweck
253	LINEST	RGP	Statistical	Berechnet die Statistik für eine Linie unter Verwendung der Methode der kleinsten Quadrate, zur Berechnung einer geraden Linie, die für die Daten am geeignetsten ist, und gibt eine Matrix zurück, die die Linie beschreibt
254	LN	LN	Math. & Trigonom.	Gibt den natürlichen Logarithmus einer Zahl zurück
255	LOG	LOG	Math. & Trigonom.	Gibt den Logarithmus einer Zahl zu der angegebenen Basis zurück
256	LOG10	LOG10	Math. & Trigonom.	Gibt den Logarithmus einer Zahl zur Basis 10 zurück
257	LOGEST	RKP	Statistical	In Regressionsanalysen berechnet diese Funktion eine Exponentialkurve, die möglichst gut an die von Ihnen bereitgestellten Daten angepasst ist, und liefert ein Wertarray, die diese Kurve beschreibt
258	LOGINV	LOGINV	Statistical (ab 2010 Compatibility)	Gibt Perzentile der Lognormalverteilung von x zurück, wobei $\ln(x)$ mit den Parametern <i>Mittelwert</i> und <i>Standabwn</i> normal verteilt ist
259	LOGNORM.DIST	LOGNORM.VERT (2010)	Statistical	Siehe LOGNORMDIST
260	LOGNORM.INV	LOGNORM.INV (2010)	Statistical	Siehe LOGINV
261	LOGNORMDIST	LOGNORMVERT	Statistical (ab 2010 Compatibility)	Gibt Werte der Verteilungsfunktion einer lognormalverteilten Zufallsvariablen zurück, wobei $\ln(x)$ normalverteilt ist mit den Parametern <i>Mittelwert</i> und <i>Standabwn</i>
262	LOOKUP	VERWEIS	Lookup & Reference	Durchsucht die Werte eines Vektors oder einer Matrix
263	LOWER	KLEIN	Text & Data	Wandelt einen Text in Kleinbuchstaben um
264	MATCH	VERGLEICH	Lookup & Reference	Sucht Werte innerhalb eines Bezugs oder einer Matrix
265	MAX	MAX	Statistical	Gibt den größten Wert innerhalb einer Argumentliste zurück
266	MAXA	MAXA	Statistical	Gibt den größten Wert einer Liste von Argumenten zurück
267	MDETERM	MDET	Math. & Trigonom.	Gibt die Determinante einer Matrix zurück
268	MDURATION	MDURATION	Financial	Gibt die modifizierte Macauley-Duration eines Wertpapiers mit einem angenommenen Nennwert von 100 € zurück



ID	Englisch	Deutsch	Kategorie	Zweck
269	MEDIAN	MEDIAN	Statistical	Gibt den Median der angegebenen Zahlen zurück
270	MID	TEIL	Text & Data	Gibt auf der Grundlage der angegebenen Anzahl von Zeichen eine bestimmte Anzahl von Zeichen einer Zeichenfolge ab der von Ihnen angegebenen Position zurück
271	MIN	MIN	Statistical	Gibt den kleinsten Wert innerhalb einer Argumentliste zurück
272	MINA	MINA	Statistical	Gibt den kleinsten Wert einer Liste von Argumenten zurück
273	MINUTE	MINUTE	Date & Time	Wandelt eine fortlaufende Zahl in eine Minute um
274	MINVERSE	MINV	Math. & Trigonom.	Gibt die Inverse einer Matrix (die zu einer Matrix gehörende Kehrmatrix) zurück
275	MIRR	QIKV	Financial	Gibt einen modifizierten internen Zinsfuß zurück, bei dem positive und negative Cashflows mit unterschiedlichen Zinssätzen finanziert werden
276	MMULT	MMULT	Math. & Trigonom.	Gibt das Produkt zweier Matrizen zurück
277	MOD	REST	Math. & Trigonom.	Gibt den Rest einer Division zurück. Das Ergebnis hat dasselbe Vorzeichen wie Divisor.
278	MODE	MODALWERT	Statistical (ab 2010 Compatibility)	Gibt den häufigsten Wert einer Datengruppe zurück
279	MODE.MULT	MODUS.VIELF (2010)	Statistical	Gibt ein vertikales Array der am häufigsten vorkommenden Werte in einem Array oder Datenbereich zurück
280	MODE.SNGL	MODUS.EINF (2010)	Statistical	Siehe MODE
281	MONTH	MONAT	Date & Time	Wandelt eine fortlaufende Zahl in einen Monat um
282	MROUND	VRUNDEN	Math. & Trigonom.	Gibt eine auf das gewünschte Vielfache gerundete Zahl zurück
283	MULTINOMINAL	POLYNOMIAL	Math. & Trigonom.	Gibt den Polynomkoeffizienten einer Gruppe von Zahlen zurück
284	MUNIT	MEINHEIT (2013)	Math. & Trigonom.	Gibt die Einheitsmatrix für die angegebene Größe zurück
285	N	N	Information	Gibt den in eine Zahl umgewandelten Wert zurück
286	NA	NV	Information	Gibt den Fehlerwert #NV zurück ►

ID	Englisch	Deutsch	Kategorie	Zweck
287	NEGBINOM.DIST	NEGBINOM.VERT (2010)	Statistical	Siehe NEGBINOMDIST
288	NEGBINOMDIST	NEGBINOMVERT	Statistical (ab 2010 Compatibility)	Gibt Wahrscheinlichkeiten einer negativbinomialverteilten Zufallsvariablen zurück
289	NETWORKDAYS	NETTOARBEITSTAGE	Date & Time	Gibt die Anzahl der Arbeitstage in einem Zeitintervall zurück
290	NETWORKDAYS.INTL	NETTOARBEITSTAGE.INTL (2010)	Date & Time	Gibt die Anzahl der Arbeitstage in einem Zeitintervall unter Beachtung der Wochenend-Definition zurück
291	NOMINAL	NOMINAL	Financial	Gibt die jährliche Nominalverzinsung zurück, ausgehend vom effektiven Zinssatz sowie der Anzahl der Verzinsungsperioden innerhalb eines Jahres
292	NORM.DIST	NORM.VERT (2010)	Statistical	Siehe NORMDIST
293	NORM.INV	NORM.INV (2010)	Statistical	Siehe NORMINV
294	NORM.S.DIST	NORM.S.VERT (2010)	Statistical	Gibt Werte der Verteilungsfunktion einer standardnormalverteilten Zufallsvariablen zurück
295	NORM.S.INV	NORM.S.INV (2010)	Statistical	Gibt Perzentile der Standardnormalverteilung zurück
296	NORMDIST	NORMVERT	Statistical (ab 2010 Compatibility)	Gibt die Normalverteilung für den angegebenen Mittelwert und die angegebene Standardabweichung zurück
297	NORMINV	NORMINV	Statistical (ab 2010 Compatibility)	Gibt Perzentile der Normalverteilung zurück
298	NORMSDIST	STANDNORMVERT	Statistical (ab 2010 Compatibility)	Gibt Werte der Verteilungsfunktion einer standardnormalverteilten Zufallsvariablen zurück
299	NORMSINV	STANDNORMINV	Statistical (ab 2010 Compatibility)	Gibt Quantile der Standardnormalverteilung zurück
300	NOT	NICHT	Logical	Keht den Wert eines Arguments um
301	NOW	JETZT	Date & Time	Liefert die fortlaufende Zahl des aktuellen Datums und der aktuellen Uhrzeit
302	NPER	ZZR	Financial	Gibt die Anzahl der Zahlungsperioden einer Investition zurück, die auf periodischen, gleichbleibenden Zahlungen sowie einem konstanten Zinssatz basiert ▶

ID	Englisch	Deutsch	Kategorie	Zweck
303	NPV	NBW	Financial	Gibt den Nettobarwert (Kapitalwert) einer Investition auf Basis eines Abzinsungsfaktors für eine Reihe periodischer Zahlungen zurück
304	NUMBERVALUE	ZAHLENWERT (2013)	Text & Data	Konvertiert unabhängig vom Gebietsschema des PC einen (Zahl-) Text in einen numerischen Wert
305	OCT2BIN	OKTINBIN	Engineering	Wandelt eine Oktalzahl in eine Binärzahl (Dualzahl) um
306	OCT2DEC	OKTINDEZ	Engineering	Wandelt eine Oktalzahl in eine Dezimalzahl um
307	OCT2HEX	OKTINHEX	Engineering	Wandelt eine Oktalzahl in eine Hexadezimalzahl um
308	ODD	UNGERADE	Math. & Trigonom.	Rundet eine Zahl auf die nächste ungerade ganze Zahl auf
309	ODDFPRICE	UNREGER.KURS	Financial	Liefert den Kurs pro 100 € Nennwert eines Wertpapiers mit einem unregelmäßigen ersten Zinstermin
310	ODDFYIELD	UNREGER.REND	Financial	Gibt die Rendite eines Wertpapiers mit einem unregelmäßigen ersten Zinstermin zurück
311	ODDLPRICE	UNREGLE.KURS	Financial	Gibt den Kurs pro 100 € Nennwert eines Wertpapiers mit einem unregelmäßigen letzten Zinstermin zurück
312	ODDYIELD	UNREGLE.REND	Financial	Gibt die Rendite eines Wertpapiers mit einem unregelmäßigen letzten Zinstermin zurück
313	OFFSET	BEREICH.VERSCHIEBEN	Lookup & Reference	Gibt einen Bezug zurück, der gegenüber dem angegebenen Bezug versetzt ist
314	OR	ODER	Logical	Gibt <i>WAHR</i> zurück, wenn ein Argument <i>WAHR</i> ist. Gibt <i>FALSCH</i> zurück, wenn alle Argumente <i>FALSCH</i> sind.
315	PDURATION	PDURATION	Financial	Berechnet die Laufzeit einer Investition bei gegebenem Zinssatz sowie eingesetztem und erwartetem Kapital
316	PEARSON	PEARSON	Statistical	Gibt den Pearsonschen Korrelationskoeffizienten r zurück
317	PERCENTILE	QUANTIL	Statistical (ab 2010 Compatibility)	Gibt das Alpha-Quantil einer Gruppe von Daten zurück
318	PERCENTILE.EXC	QUANTIL.EXKL (2010)	Statistical	Gibt das k -Quantil einer Gruppe von Daten zurück, wobei k ausschließlich im Bereich 0 bis 1 liegt ▶

ID	Englisch	Deutsch	Kategorie	Zweck
319	PERCENTILE.INC	QUANTIL.INKL (2010)	Statistical	Gibt das k -Quantil einer Gruppe von Daten zurück, wobei k einschließlich im Bereich 0 bis 1 liegt
320	PERCENTRANK	QUANTILSRANG	Statistical (ab 2010 Compatibility)	Gibt den prozentualen Rang (Alpha) eines Werts zurück
321	PERCENTRANK.EXC	QUANTILSRANG.EXKL (2010)	Statistical	Gibt den prozentualen Rang (Alpha) eines Werts ausschließlich im Bereich 0 bis 1 zurück
322	PERCENTRANK.INC	QUANTILSRANG.INKL (2010)	Statistical	Gibt den prozentualen Rang (Alpha) eines Werts einschließlich im Bereich 0 bis 1 zurück
323	PERMUT	VARIATIONEN	Statistical	Gibt die Anzahl der Möglichkeiten zurück, um k Elemente aus einer Menge von n Elementen ohne Zurücklegen zu ziehen
324	PERMUTATIONA	VARIATIONEN2 (2013)	Statistical	Gibt die Anzahl der Möglichkeiten – mit Wiederholungen – für eine angegebene Menge von Objekten zurück
325	PHI	PHI (2013)	Statistical	Gibt den Wert der Dichtefunktion für eine Standardnormalverteilung zurück
326	PHONETIC	PHONETIC	Text & Data	Extrahiert die phonetischen (Furigana-) Zeichen aus einer Textzeichenfolge
327	PI	PI	Math. & Trigonom.	Gibt den Wert Pi zurück, die mathematische Konstante (3,14159265358979) mit einer Genauigkeit von 15 Stellen
328	PMT	RMZ	Financial	Gibt die konstante Zahlung einer Annuität pro Periode zurück, wobei konstante Zahlungen und ein konstanter Zinssatz vorausgesetzt werden
329	POISSON	POISSON	Statistical (ab 2010 Compatibility)	Gibt Wahrscheinlichkeiten einer poissonverteilten Zufallsvariablen zurück
330	POISSON.DIST	POISSON.VERT (2010)	Statistical	Siehe POISSON
331	POWER	POTENZ	Math. & Trigonom.	Gibt als Ergebnis eine potenzierte Zahl zurück
332	PPMT	KAPZ	Financial	Gibt die Kapitalrückzahlung einer Investition für die angegebene Periode zurück
333	PRICE	KURS	Financial	Gibt den Kurs pro 100 € Nennwert eines Wertpapiers zurück, das periodisch Zinsen auszahlt ▶

ID	Englisch	Deutsch	Kategorie	Zweck
334	PRICEDISC	KURSDISAGIO	Financial	Gibt den Kurs pro 100 € Nennwert eines unverzinslichen Wertpapiers zurück
335	PRICEMAT	KURSFÄLLIG	Financial	Gibt den Kurs pro 100 € Nennwert eines Wertpapiers zurück, das Zinsen am Fälligkeitsdatum auszahlt
336	PROB	WAHRSCHEBEREICH	Statistical	Gibt die Wahrscheinlichkeit für ein von zwei Werten eingeschlossenes Intervall zurück
337	PRODUCT	PRODUKT	Math. & Trigonom.	Multipliziert die Argumente und gibt das Produkt zurück
338	PROPER	GROSS2	Text & Data	Wandelt den ersten Buchstaben aller Wörter einer Zeichenfolge in Großbuchstaben und alle anderen Buchstaben in Kleinbuchstaben um
339	PV	BW	Financial	Gibt den Barwert einer Investition zurück
340	QUARTILE	QUARTILE	Statistical (ab 2010 Compatibility)	Gibt die Quartile der Datengruppe zurück
341	QUARTILE.EXC	QUARTILE.EXKL (2010)	Statistical	Gibt die Quartile der Datengruppe zurück, ausschließlich basierend auf Werten von 0 bis 1
342	QUARTILE.INC	QUARTILE.INKL (2010)	Statistical	Gibt die Quartile der Datengruppe zurück, einschließlich basierend auf Werten von 0 bis 1
343	QUOTIENT	QUOTIENT	Math. & Trigonom.	Gibt den ganzzahligen Anteil einer Division zurück
344	RADIANS	BOGENMASS	Math. & Trigonom.	Wandelt Grad in Bogenmaß (Radiant) um
345	RAND	ZUFALLSZAHL	Math. & Trigonom.	Gibt eine Zufallszahl zwischen 0 und 1 zurück
346	RANDBETWEEN	ZUFALLSBEREICH	Math. & Trigonom.	Gibt eine ganze Zufallszahl aus dem festgelegten Bereich zurück
347	RANK	RANG	Statistical (ab 2010 Compatibility)	Gibt den Rang zurück, den eine Zahl innerhalb einer Liste von Zahlen einnimmt
348	RANK.AVG	RANG.MITTELW (2010)	Statistical	Gibt die durchschnittliche Rangzahl für eine Zahl innerhalb einer Liste von Zahlen zurück
349	RANK.EQ	RANG.GLEICH (2010)	Statistical	Siehe RANK
350	RATE	ZINS	Financial	Gibt den Zinssatz einer Annuität pro Periode zurück
351	RECEIVED	ZINSSATZ	Financial	Gibt den Zinssatz eines voll investierten Wertpapiers zurück ►

ID	Englisch	Deutsch	Kategorie	Zweck
352	REPLACE	ERSETZEN	Text & Data	Ersetzt auf der Grundlage der Anzahl von Zeichen, die Sie angeben, einen Teil einer Textzeichenfolge durch eine andere Textzeichenfolge
353	REPLACEB	ERSETZENB	Text & Data	Ersetzt auf der Grundlage der von Ihnen angegebenen Anzahl der Bytes einen Teil einer Textzeichenfolge durch eine andere Textzeichenfolge
354	REPT	WIEDERHOLEN	Text & Data	Wiederholt einen Text so oft wie angegeben
355	RIGHT	RECHTS	Text & Data	Gibt das letzte oder die letzten Zeichen einer Textzeichenfolge auf der Grundlage der von Ihnen angegebenen Anzahl von Zeichen zurück
356	ROMAN	RÖMISCH	Math. & Trigonom.	Wandelt eine arabische Zahl in eine römische Zahl als Text um
357	ROUND	RUNDEN	Math. & Trigonom.	Rundet eine Zahl auf eine bestimmte Anzahl von Dezimalstellen
358	ROUNDDOWN	ABRUNDEN	Math. & Trigonom.	Rundet die Zahl auf <i>Anzahl_Stellen</i> ab
359	ROUNDUP	AUFRUNDEN	Math. & Trigonom.	Rundet die Zahl auf <i>Anzahl_Stellen</i> auf
360	ROW	ZEILE	Lookup & Reference	Liefert die Zeilennummer eines Bezugs
361	ROWS	ZEILEN	Lookup & Reference	Gibt die Anzahl der Zeilen in einem Bezug oder einer Matrix zurück
362	RRI	ZSATZINVEST (2013)	Financial	Berechnet den Zinssatz, zu welchem ein Kapital investiert werden muss, damit es einen bestimmten Zukunftswert besitzt
363	RSQ	BESTIMMTHEITSMASS	Statistical	Gibt das Quadrat des Pearsonschen Korrelationskoeffizienten zurück, entsprechend den in <i>Y_Werte</i> und <i>X_Werte</i> abgelegten Datenpunkten
364	RTD	RTD	Lookup & Reference	Ruft Echtzeitdaten von einem Programm ab, das die COM-Automatisierung unterstützt
365	SEARCH	SUCHEN	Text & Data	Gibt, beginnend mit <i>Erstes_Zeichen</i> , die Nummer des Zeichens zurück, an der das zu suchende Zeichen oder die zu suchende Textzeichenfolge erstmals gefunden wurde ►

ID	Englisch	Deutsch	Kategorie	Zweck
366	SEC	SEC (2013)	Math. & Trigonom.	Gibt den Sekans eines Winkels zurück
367	SECH	SECHYP (2013)	Math. & Trigonom.	Gibt den hyperbolischen Sekans eines Winkels zurück
368	SECOND	SEKUNDE	Date & Time	Wandelt eine fortlaufende Zahl in eine Sekunde um
369	SERIESSUM	POTENZREIHE	Math. & Trigonom.	Gibt die Summe von Potenzen (zur Berechnung von Potenzreihen und dichotomen Wahrscheinlichkeiten) zurück
370	SHEET	BLATT (2013)	Information	Gibt die Nummer des Blatts zurück, welches durch den Bezug <i>Bereich</i> angesprochen wird
371	SHEETS	BLÄTTER (2013)	Information	Gibt die Anzahl der Blätter zurück, welche durch den Bezug <i>Bereich</i> angesprochen werden
372	SIGN	VORZEICHEN	Math. & Trigonom.	Gibt das Vorzeichen einer Zahl zurück
373	SIN	SIN	Math. & Trigonom.	Gibt den Sinus einer Zahl zurück
374	SINH	SINHYP	Math. & Trigonom.	Gibt den hyperbolischen Sinus einer Zahl zurück
375	SKEW	SCHIEFE	Statistical	Gibt die Schiefe einer Verteilung zurück
376	SKEW.P	SCHIEFE.P (2013)	Statistical	Gibt die Schiefe einer Verteilung auf Basis einer Grundgesamtheit zurück
377	SLN	LIA	Financial	Gibt die lineare Abschreibung eines Wirtschaftsguts pro Periode zurück
378	SLOPE	STEIGUNG	Statistical	Gibt die Steigung der Regressionsgeraden zurück, die an die in <i>Y_Werte</i> und <i>X_Werte</i> abgelegten Datenpunkte angepasst ist
379	SMALL	KKLEINSTE	Statistical	Gibt den k-kleinsten Wert einer Datengruppe zurück
380	SQL.REQUEST	SQL.REQUEST	Externe Funktionen	Stellt eine Verbindung mit einer externen Datenquelle her und führt eine Abfrage aus, die sich auf einem Arbeitsblatt befindet
381	SQRT	WURZEL	Math. & Trigonom.	Gibt die Quadratwurzel einer Zahl zurück
382	SQRTPI	WURZELPI	Math. & Trigonom.	Gibt die Wurzel aus der mit Pi multiplizierten Zahl zurück
383	STANDARDIZE	STANDARDISIERUNG	Statistical	Gibt den standardisierten Wert einer Verteilung zurück, die durch <i>Mittelwert</i> und <i>Standabwn</i> charakterisiert ist

ID	Englisch	Deutsch	Kategorie	Zweck
384	STDEV	STABW	Statistical (ab 2010 Compatibility)	Schätzt die Standardabweichung ausgehend von einer Stichprobe
385	STDEV.P	STABW.N (2010)	Statistical	Siehe STDEV.P
386	STDEV.S	STABW.S (2010)	Statistical	Siehe STDEV
387	STDEVA	STABWA	Statistical	Schätzt die Standardabweichung ausgehend von einer Stichprobe
388	STDEV.P	STABWN	Statistical (ab 2010 Compatibility)	Berechnet die Standardabweichung ausgehend von der Grundgesamtheit
389	STDEVPA	STABWNA	Statistical	Berechnet die Standardabweichung ausgehend von einer als Argumente angegebenen Grundgesamtheit, einschließlich Text und logischer Werte
390	STEYX	STFEHLERYX	Statistical	Gibt den Standardfehler der geschätzten y -Werte für alle x -Werte der Regression zurück
391	SUBSTITUTE	WECHSELN	Text & Data	Ersetzt alten Text durch neuen Text in einer Zeichenfolge
392	SUBTOTAL	TEILERGEBNIS	Math. & Trigonom.	Gibt ein Teilergebnis in einer Liste oder Datenbank zurück
393	SUM	SUMME	Math. & Trigonom.	Summiert die Argumente
394	SUMIF	SUMMEWENN	Math. & Trigonom.	Addiert Zahlen, die mit dem Suchkriterium übereinstimmen
395	SUMIFS	SUMMEWENNS	Math. & Trigonom.	Addiert Zahlen, die mit mehreren Suchkriterien übereinstimmen
396	SUMPRODUCT	SUMMENPRODUKT	Math. & Trigonom.	Multipliziert die einander entsprechenden Komponenten der angegebenen Matrizen miteinander und gibt die Summe dieser Produkte zurück
397	SUMSQ	QUADRATSUMME	Math. & Trigonom.	Summiert die quadrierten Argumente
398	SUMX2MY2	SUMMEX2MY2	Math. & Trigonom.	Summiert für zusammengehörige Komponenten zweier Matrizen die Differenzen der Quadrate
399	SUMX2PY2	SUMMEX2PY2	Math. & Trigonom.	Summiert für zusammengehörige Komponenten zweier Matrizen die Summen der Quadrate
400	SUMXMY2	SUMMEXMY2	Math. & Trigonom.	Summiert für zusammengehörige Komponenten zweier Matrizen die quadrierten Differenzen
401	SYD	DIA	Financial	Gibt die arithmetisch-degressive Abschreibung eines Wirtschaftsguts für eine bestimmte Periode zurück



ID	Englisch	Deutsch	Kategorie	Zweck
402	T	T	Text & Data	Wandelt die Argumente in Text um
403	T.DIST	T.VERT (2010)	Statistical	Gibt die (Student) t-Verteilung der linken Endfläche zurück
404	T.DIST.2T	T.VERT.2S (2010)	Statistical	Gibt die (Student) t-Verteilung für zwei Endflächen zurück
405	T.DIST.RT	T.VERT.RE (2010)	Statistical	Gibt die (Student) t-Verteilung der rechten Endfläche zurück
406	T.INV	T.INV (2010)	Statistical	Siehe TINV
407	T.INV.2T	T.INV.2S (2010)	Statistical	Siehe TINV
408	T.TEST	T.TEST (2010)	Statistical	Siehe TTEST
409	TAN	TAN	Math. & Trigonom.	Gibt den Tangens einer Zahl zurück
410	TANH	TANHYP	Math. & Trigonom.	Gibt den hyperbolischen Tangens einer Zahl zurück
411	TBILLEQ	TBILLÄQUIV	Financial	Rechnet die Verzinsung eines Schatzwechsels (Treasury Bill) in die für Anleihen übliche einfache jährliche Verzinsung um
412	TBILLPRICE	TBILLKURS	Financial	Gibt den Kurs pro 100 € Nennwert eines Schatzwechsels (Treasury Bill) zurück
413	TBILLYIELD	TBILLRENDITE	Financial	Gibt die Rendite eines Schatzwechsels (Treasury Bill) zurück
414	TDIST	TVERT	Statistical (ab 2010 Compatibility)	Gibt Werte der Verteilungsfunktion (1-Alpha) einer (Student) t-verteilten Zufallsvariablen zurück
415	TEXT	TEXT	Text & Data	Formatiert eine Zahl und wandelt sie in Text um
416	TIME	ZEIT	Date & Time	Gibt die Dezimalzahl einer bestimmten Uhrzeit zurück
417	TIMEVALUE	ZEITWERT	Date & Time	Wandelt eine als Text vorliegende Zeitangabe in eine fortlaufende Zahl um
418	TINV	TINV	Statistical (ab 2010 Compatibility)	Gibt den t-Wert der t-Verteilung als eine Funktion der Wahrscheinlichkeit und der Freiheitsgrade zurück
419	TODAY	HEUTE	Date & Time	Gibt die fortlaufende Zahl des heutigen Datums zurück
420	TRANSPOSE	MTRANS	Lookup & Reference	Gibt die transponierte Matrix der angegebenen Matrix zurück
421	TREND	TREND	Statistical	Liefert Werte, die sich aus einem linearen Trend ergeben ►

ID	Englisch	Deutsch	Kategorie	Zweck
422	TRIM	GLÄTTEN	Text & Data	Löscht Leerzeichen in <i>Text</i> , die nicht als jeweils einzelne zwischen Wörtern stehende Trennzeichen dienen
423	TRIMMEAN	GESTUTZTMITTEL	Statistical	Gibt den Mittelwert einer Datengruppe zurück, ohne die Randwerte zu berücksichtigen
424	TRUE	WAHR	Logical	Gibt den Wahrheitswert <i>WAHR</i> zurück
425	TRUNC	KÜRZEN	Math. & Trigonom.	Schneidet die Kommastellen der Zahl ab und gibt als Ergebnis eine ganze Zahl zurück
426	TTEST	TTEST	Statistical (ab 2010 Compatibility)	Gibt die Teststatistik eines Studentschen t-Tests zurück
427	TYPE	TYP	Information	Gibt eine Zahl zurück, die den Datentyp des angegebenen Werts anzeigt
428	UNICHAR	UNIZEICHEN (2013)	Text & Data	Gibt das Unicodezeichen für eine Codezahl zurück
429	UNICODE	UNICODE (2013)	Text & Data	Gibt die Unicodezahl des ersten Zeichens eines Texts zurück
430	UPPER	GROSS	Text & Data	Wandelt Text in Großbuchstaben um
431	VALUE	WERT	Text & Data	Wandelt ein als Text angegebenes Argument in eine Zahl um
432	VAR	VARIANZ	Statistical (ab 2010 Compatibility)	Schätzt die Varianz auf der Basis einer Stichprobe
433	VAR.P	VAR.P (2010)	Statistical	Siehe VARP
434	VAR.S	VAR.S (2010)	Statistical	Siehe VAR
435	VARA	VARIANZA	Statistical	Schätzt die Varianz auf der Basis einer Stichprobe
436	VARP	VARIANZEN	Statistical (ab 2010 Compatibility)	Berechnet die Varianz ausgehend von der Grundgesamtheit
437	VARPA	VARIANZENA	Statistical	Berechnet die Varianz ausgehend von der Grundgesamtheit
438	VDB	VDB	Financial	Gibt die degressive Doppelraten-Abschreibung eines Wirtschaftsguts für eine bestimmte Periode oder Teilperiode zurück
439	VLOOKUP	SVERWEIS	Lookup & Reference	Sucht in der am weitesten links gelegenen Spalte einer Tabelle nach einem Wert und gibt in der gleichen Zeile einen Wert aus einer von Ihnen angegebenen Spalte in der Tabelle zurück



ID	Englisch	Deutsch	Kategorie	Zweck
440	WEBSERVICE	WEBDIENST (2013)	Web	Empfängt (XML-) Daten vom angegebenen Webdienst
441	WEEKDAY	WOCHENTAG	Date & Time	Wandelt eine fortlaufende Zahl in einen Wochentag um
442	WEEKNUM	KALENDERWOCHE	Date & Time	Gibt eine Zahl zurück, die angibt, in welche Woche des dazugehörigen Jahres das angegebene Datum fällt
443	WEIBULL	WEIBULL	Statistical (ab 2010 Compatibility)	Gibt Wahrscheinlichkeiten einer weibullverteilten Zufallsvariablen zurück
444	WEIBULL.DIST	WEIBULL.VERT (2010)	Statistical	Siehe WEIBULL
445	WORKDAY	ARBEITSTAG	Date & Time	Gibt die Datumsangabe als fortlaufenden Tag im Jahr zurück, vor oder nach einer bestimmten Anzahl von Arbeitstagen
446	WORKDAY.INTL	ARBEITSTAG.INTL (2010)	Date & Time	Gibt unter Beachtung der Wochenenddefinition die Datumsangabe als fortlaufenden Tag im Jahr zurück, vor oder nach einer bestimmten Anzahl von Arbeitstagen
447	XIRR	XINTZINSFUSS	Financial	Gibt den internen Zinsfuß einer Reihe nicht periodisch anfallender Zahlungen zurück
448	XNPV	XKAPITALWERT	Financial	Gibt den Nettobarwert (Kapitalwert) einer Reihe nicht periodisch anfallender Zahlungen zurück
449	XOR	XODER (2013)	Logical	Gibt den Wahrheitswert <i>WAHR</i> nur dann zurück, wenn genau eines der Argumente <i>WAHR</i> ist
450	YEAR	JAHR	Date & Time	Wandelt eine fortlaufende Zahl in eine Jahreszahl um
451	YEARFRAC	BRTEILJAHRE	Date & Time	Wandelt die Anzahl der ganzen Tage zwischen Ausgangsdatum und Enddatum in Bruchteile von Jahren um
452	YIELD	RENDITE	Financial	Gibt die Rendite eines Wertpapiers zurück, das periodisch Zinsen auszahlt
453	YIELDDISC	RENDITEDIS	Financial	Gibt die jährliche Rendite eines unverzinslichen Wertpapiers zurück

ID	Englisch	Deutsch	Kategorie	Zweck
454	YIELDMAT	RENDITEFÄLL	Financial	Gibt die jährliche Rendite eines Wertpapiers zurück, das Zinsen am Fälligkeitsdatum auszahlt
455	Z.TEST	G.TEST (2010)	Statistical	Gibt den einseitigen Wahrscheinlichkeitswert für einen Gaußtest (Normalverteilung) zurück
456	ZTEST	GTEST	Statistical (ab 2010 Compatibility)	Gibt den einseitigen Wahrscheinlichkeitswert für einen Gaußtest (Normalverteilung) zurück

Anhang B

Funktionen nach Kategorie

Datums- und Zeitfunktionen	938
Text- und Datenfunktionen	939
Logische Funktionen	940
Nachschlage- und Verweisfunktionen	940
Informationsfunktionen	941
Statistische Funktionen	942
Datenbankfunktionen	947
Cubefunktionen	948
Finanzmathematische Funktionen	948
Mathematische und trigonometrische Funktionen	951
Konstruktionsfunktionen	954
Webfunktionen	956

Datums- und Zeitfunktionen

Funktion	Beschreibung
ARBEITSTAG	Gibt die Datumsangabe als fortlaufenden Tag im Jahr zurück, vor oder nach einer bestimmten Anzahl von Arbeitstagen
ARBEITSTAG.INTL (2010)	Gibt unter Beachtung der Wochenenddefinition die Datumsangabe als fortlaufenden Tag im Jahr zurück, vor oder nach einer bestimmten Anzahl von Arbeitstagen
BRTEILJAHRE	Wandelt die Anzahl der ganzen Tage zwischen Ausgangsdatum und Enddatum in Bruchteile von Jahren um
DATEDIF	Ermittelt die Differenz zwischen <i>Anfangs-</i> und <i>Enddatum</i> in Jahren, Monaten oder Tagen
DATUM	Gibt die fortlaufende Zahl zurück, die ein bestimmtes Datum darstellt
DATWERT	Wandelt ein als Text vorliegendes Datum in eine fortlaufende Zahl um
EDATUM	Gibt die fortlaufende Zahl des Datums zurück, das eine bestimmte Anzahl von Monaten vor bzw. nach dem angegebenen Datum (Ausgangsdatum) liegt
HEUTE	Gibt die fortlaufende Zahl des heutigen Datums zurück
ISOKALENDERWOCHE (2013)	Berechnet die Kalenderwochennummer nach ISO-Norm für einen Datumswert
JAHR	Wandelt eine fortlaufende Zahl in eine Jahreszahl um
JETZT	Liefert die fortlaufende Zahl des aktuellen Datums und der aktuellen Uhrzeit
KALENDERWOCHE	Gibt eine Zahl zurück, die angibt, in welche Woche des dazugehörigen Jahres das angegebene Datum fällt
MINUTE	Wandelt eine fortlaufende Zahl in eine Minute um
MONAT	Wandelt eine fortlaufende Zahl in einen Monat um
MONATSENDE	Gibt die fortlaufende Zahl des letzten Tages des Monats zurück, der eine bestimmte Anzahl von Monaten vor bzw. nach dem Ausgangsdatum liegt
NETTOARBEITSTAGE	Gibt die Anzahl der Arbeitstage in einem Zeitintervall zurück
NETTOARBEITSTAGE.INTL (2010)	Gibt die Anzahl der Arbeitstage in einem Zeitintervall unter Beachtung der Wochenend-Definition zurück
SEKUNDE	Wandelt eine fortlaufende Zahl in eine Sekunde um
STUNDE	Gibt die Stunde einer Zeitangabe zurück
TAG	Gibt den Tag eines Datums als fortlaufende Zahl zurück
TAGE (2013)	Berechnet die Anzahl der Tage zwischen zwei Datumsangaben
TAGE360	Berechnet, ausgehend von einem Jahr, das 360 Tage umfasst, die Anzahl der zwischen zwei Tagesdaten liegenden Tage
WOCHENTAG	Wandelt eine fortlaufende Zahl in einen Wochentag um
ZEIT	Gibt die Dezimalzahl einer bestimmten Uhrzeit zurück
ZEITWERT	Wandelt eine als Text vorliegende Zeitangabe in eine fortlaufende Zahl um

Text- und Datenfunktionen

Funktion	Beschreibung
ASC	Wandelt bei Sprachen mit einem Double-Byte-Zeichensatz (DBCS) Zeichen voller Breite (Double-Byte-Zeichen) in Zeichen halber Breite (Single-Byte-Zeichen) um
BAHTTEXT	Wandelt eine Zahl in Thai-Text um und fügt diesem das Suffix »Baht« hinzu
CODE	Gibt die Codezahl des ersten Zeichens in einem Text zurück
DM	Konvertiert eine Zahl in ein Textformat und ordnet ein Währungssymbol zu
ERSETZEN	Ersetzt auf der Grundlage der Anzahl von Zeichen, die Sie angeben, einen Teil einer Textzeichenfolge durch eine andere Textzeichenfolge
ERSETZENB	Ersetzt auf der Grundlage der von Ihnen angegebenen Anzahl der Bytes einen Teil einer Textzeichenfolge durch eine andere Textzeichenfolge
FEST	Formatiert eine Zahl als Text mit einer festen Anzahl von Nachkommastellen
FINDEN	Sucht eine Zeichenfolge (Suchtext) innerhalb einer anderen Zeichenfolge (Text) und gibt als Ergebnis die Nummer der Anfangsposition von Suchtext ab dem ersten Zeichen von Text zurück
FINDENB	Sucht eine Zeichenfolge (Suchtext) innerhalb einer anderen Zeichenfolge (Text) und gibt als Ergebnis die Nummer der Anfangsposition von Suchtext ab dem ersten Zeichen von Text auf der Grundlage der Anzahl von Bytes zurück, die jedes Zeichen verwendet
GLÄTTEN	Löscht Leerzeichen in Text, die nicht als jeweils einzelne zwischen Wörtern stehende Trennzeichen dienen
GROSS	Wandelt Text in Großbuchstaben um
GROSS2	Wandelt den ersten Buchstaben aller Wörter einer Zeichenfolge in Großbuchstaben und alle anderen Buchstaben in Kleinbuchstaben um
IDENTISCH	Prüft, ob zwei Zeichenfolgen identisch sind
KLEIN	Wandelt einen Text in Kleinbuchstaben um
LÄNGE	Gibt die Anzahl der Zeichen einer Zeichenfolge zurück
LINKS	Gibt auf der Grundlage der Anzahl von Zeichen, die Sie angeben, das oder die erste(n) Zeichen in einer Textzeichenfolge zurück
PHONETIC	Extrahiert die phonetischen (Furigana-) Zeichen aus einer Textzeichenfolge
RECHTS	Gibt das letzte oder die letzten Zeichen einer Textzeichenfolge auf der Grundlage der von Ihnen angegebenen Anzahl von Zeichen zurück
SÄUBERN	Löscht alle nicht druckbaren Zeichen aus einem Text
SUCHEN	Gibt, beginnend mit <i>Erstes_Zeichen</i> , die Nummer des Zeichens zurück, an der das zu suchende Zeichen oder die zu suchende Textzeichenfolge erstmals gefunden wurde
T	Wandelt die Argumente in Text um
TEIL	Gibt auf der Grundlage der angegebenen Anzahl von Zeichen eine bestimmte Anzahl von Zeichen einer Zeichenfolge ab der von Ihnen angegebenen Position zurück
TEXT	Formatiert eine Zahl und wandelt sie in Text um
UNICODE (2013)	Gibt die Unicodezahl des ersten Zeichens eines Texts zurück
UNIZEICHEN (2013)	Gibt das Unicodezeichen für eine Codezahl zurück

Funktion	Beschreibung
VERKETTEN	Verknüpft mehrere Textzeichenfolgen zu einer Textzeichenfolge
WECHSELN	Ersetzt alten Text durch neuen Text in einer Zeichenfolge
WERT	Wandelt ein als Text angegebenes Argument in eine Zahl um
WIEDERHOLEN	Wiederholt einen Text so oft wie angegeben
ZAHLENWERT (2013)	Konvertiert unabhängig vom Gebietsschema des PC einen (Zahl-)Text in einen numerischen Wert
ZEICHEN	Gibt das der Codezahl entsprechende Zeichen zurück

Logische Funktionen

Funktion	Beschreibung
FALSCH	Gibt den Wahrheitswert <i>FALSCH</i> zurück
NICHT	Kehrt den Wert eines Arguments um
ODER	Gibt <i>WAHR</i> zurück, wenn ein Argument <i>WAHR</i> ist. Gibt <i>FALSCH</i> zurück, wenn alle Argumente <i>FALSCH</i> sind.
UND	Gibt <i>WAHR</i> zurück, wenn alle Argumente <i>WAHR</i> sind. Sind die Aussagen eines oder mehrerer Argumente <i>FALSCH</i> , gibt diese Funktion den Wert <i>FALSCH</i> zurück.
WAHR	Gibt den Wahrheitswert <i>WAHR</i> zurück
WENN	Gibt eine Wahrheitsprüfung an, die durchgeführt werden soll
WENNFEHLER	Berechnet eine definierte Ausgabe im Fall, dass ein Ausdruck fehlerhaft ist, und gibt ansonsten das Ergebnis des Ausdrucks aus
WENNNV	Berechnet eine definierte Ausgabe im Fall, dass ein Ausdruck zum Fehler <i>#NV</i> führt
XODER (2013)	Gibt den Wahrheitswert <i>WAHR</i> nur dann zurück, wenn genau eines der Argumente <i>WAHR</i> ist

Nachschlage- und Verweisfunktionen

Funktion	Beschreibung
ADRESSE	Liefert einen Bezug auf eine Zelle einer Tabelle als Text
BEREICH.VERSCHIEBEN	Gibt einen Bezug zurück, der gegenüber dem angegebenen Bezug versetzt ist
BEREICHE	Gibt die Anzahl der innerhalb eines Bezugs aufgeführten Bereiche zurück
FORMELTEXT (2013)	Liefert die Formel, die im gegebenen Bereich eingetragen wurde, als Text
HYPERLINK	Erstellt eine Verknüpfung oder einen Sprung, über die ein auf einem Netzwerkservers, in einem Intranet oder im Internet gespeichertes Dokument geöffnet wird
INDEX	Verwendet einen Index, um aus einem Bezug oder einer Matrix einen Wert zu wählen



Funktion	Beschreibung
INDIREKT	Gibt den Bezug eines Textwerts zurück. Bezüge werden sofort ausgewertet, sodass die zu ihnen gehörenden Werte angezeigt werden.
MTRANS	Gibt die transponierte Matrix der angegebenen Matrix zurück
RTD	Ruft Echtzeitdaten von einem Programm ab, das die COM-Automatisierung unterstützt
SPALTE	Gibt die Spaltennummer eines Bezugs zurück
SPALTEN	Gibt die Anzahl der Spalten einer Matrix oder eines Bezugs zurück
SVERWEIS	Sucht in der am weitesten links gelegenen Spalte einer Tabelle nach einem Wert und gibt in der gleichen Zeile einen Wert aus einer von Ihnen angegebenen Spalte in der Tabelle zurück
VERGLEICH	Sucht Werte innerhalb eines Bezugs oder einer Matrix
VERWEIS	Durchsucht die Werte eines Vektors oder einer Matrix
WAHL	Verwendet Index, um einen Wert aus der Liste der Werteargumente zurückzugeben
WVERWEIS	Sucht in der obersten Zeile einer Tabelle oder einer Matrix nach Werten und gibt dann in der gleichen Spalte einen Wert aus einer Zeile zurück, die Sie in der Tabelle oder Matrix angeben
ZEILE	Liefert die Zeilennummer eines Bezugs
ZEILEN	Gibt die Anzahl der Zeilen in einem Bezug oder einer Matrix zurück

Informationsfunktionen

Funktion	Beschreibung
BLATT (2013)	Gibt die Nummer des Blatts zurück, welches durch den Bezug <i>Bereich</i> angesprochen wird
BLÄTTER (2013)	Gibt die Anzahl der Blätter zurück, welche durch den Bezug <i>Bereich</i> angesprochen werden
FEHLER.TYP	Gibt eine Zahl zurück, die einem der Fehlerwerte in Microsoft Excel entspricht, oder den Fehlerwert <i>#NV</i> , wenn kein Fehler vorhanden ist
INFO	Gibt Informationen zu der aktuellen Betriebssystemumgebung zurück
ISTBEZUG	Gibt <i>WAHR</i> zurück, wenn der Wert ein gültiger Zellbezug ist, oder <i>FALSCH</i> , wenn der Wert ein ungültiger Zellbezug ist
ISTFEHL	Gibt <i>WAHR</i> zurück, wenn der Wert einen Fehler ergibt, oder <i>FALSCH</i> , wenn der Wert keinen Fehler ergibt
ISTFEHLER	Gibt <i>WAHR</i> zurück, wenn der Wert einen beliebigen Fehler ergibt, oder <i>FALSCH</i> , wenn der Wert keinen Fehler ergibt
ISTFORMEL (2013)	Gibt <i>WAHR</i> zurück, wenn ein Bezug auf eine existierende Zelle verweist und sich dort eine Formel befindet
ISTGERADE	Gibt <i>WAHR</i> zurück, wenn die Zahl gerade ist, oder <i>FALSCH</i> , wenn die Zahl ungerade ist
ISTKTEXT	Gibt <i>WAHR</i> zurück, wenn der Wert kein Text ist, oder <i>FALSCH</i> , wenn der Wert ein Text ist
ISTLEER	Gibt <i>WAHR</i> zurück, wenn der Wert sich auf eine leere Zelle bezieht, oder <i>FALSCH</i> , wenn der Wert sich auf eine nicht leere Zelle bezieht

Funktion	Beschreibung
ISTLOG	Gibt <i>WAHR</i> zurück, wenn der Wert sich auf einen Wahrheitswert bezieht, oder <i>FALSCH</i> , wenn der Wert sich nicht auf einen Wahrheitswert bezieht
ISTNV	Gibt <i>WAHR</i> zurück, wenn der Wert sich auf den Fehlerwert #NV bezieht, oder <i>FALSCH</i> , wenn der Wert sich nicht auf den Fehlerwert #NV bezieht
ISTTEXT	Gibt <i>WAHR</i> zurück, wenn der Wert ein Text ist, oder <i>FALSCH</i> , wenn der Wert kein Text ist. Beachten Sie, dass diese Funktion <i>FALSCH</i> zurückgibt, wenn sich der Wert auf eine leere Zelle bezieht.
ISTUNGERADE	Gibt <i>WAHR</i> zurück, wenn die Zahl ungerade ist, oder <i>FALSCH</i> , wenn die Zahl gerade ist
ISTZAHL	Gibt <i>WAHR</i> zurück, wenn der Wert eine Zahl (numerischer Wert) ist, oder <i>FALSCH</i> , wenn der Wert keine Zahl ist
N	Gibt den in eine Zahl umgewandelten Wert zurück
NV	Gibt den Fehlerwert #NV zurück
TYP	Gibt eine Zahl zurück, die den Datentyp des angegebenen Werts anzeigt
ZELLE	Gibt Informationen zur Formatierung, der Position oder dem Inhalt der Zelle links oben in einem Bezug zurück

Statistische Funktionen

Funktion	Beschreibung
ACHSENABSCHNITT	Gibt den Schnittpunkt der Regressionsgeraden zurück
ANZAHL	Berechnet, wie viele Zahlen eine Liste von Argumenten enthält
ANZAHL2	Berechnet, wie viele Werte eine Liste von Argumenten enthält
ANZAHLLEEREZELLEN	Zählt die leeren Zellen in einem Zellbereich
BESTIMMTHEITSMASS	Gibt das Quadrat des Pearsonschen Korrelationskoeffizienten zurück, entsprechend den in <i>Y_Werte</i> und <i>X_Werte</i> abgelegten Datenpunkten
BETA.INV (2010)	Siehe BETAINV
BETA.VERT (2010)	Siehe BETAVERT
BETAINV	Gibt das Quantil der angegebenen Betaverteilung zurück
BETAVERT	Gibt die Werte der kumulierten Betaverteilungsfunktion zurück
BINOM.INV (2010)	Gibt den kleinsten Wert, für den die kumulierten Wahrscheinlichkeiten der Binomialverteilung größer oder gleich einer Grenzwahrscheinlichkeit sind, zurück
BINOM.VERT (2010)	Siehe BINOMVERT
BINOM.VERT.BEREICH (2013)	Gibt die Erfolgswahrscheinlichkeit eines Versuchsergebnisses als Binomialverteilung zurück
BINOMVERT	Gibt Wahrscheinlichkeiten einer binomialverteilten Zufallsvariablen zurück
CHIINV	Gibt Quantile der χ^2 -Verteilung zurück
CHIQU.INV (2010)	Gibt Quantile der linksseitigen χ^2 -Verteilung zurück
CHIQU.INV.RE (2010)	Gibt Quantile der rechtsseitigen χ^2 -Verteilung zurück



Funktion	Beschreibung
CHIQU.TEST (2010)	Siehe CHITEST
CHIQU.VERT (2010)	Gibt Werte der linksseitigen Verteilungsfunktion (1-Alpha) einer Chi-Quadrat-verteilten Zufallsgröße zurück
CHIQU.VERT.RE (2010)	Gibt Werte der rechtsseitigen Verteilungsfunktion (1-Alpha) einer Chi-Quadrat-verteilten Zufallsgröße zurück
CHITEST	Gibt die Teststatistik eines χ^2 -Unabhängigkeitstests zurück
CHIVERT	Gibt Werte der Verteilungsfunktion (1-Alpha) einer Chi-Quadrat-verteilten Zufallsgröße zurück
EXPON.VERT (2010)	Siehe EXPONVERT
EXPONVERT	Gibt Wahrscheinlichkeiten einer exponential verteilten Zufallsvariablen zurück
F.INV (2010)	Gibt Quantile der linksseitigen F-Verteilung zurück
F.INV.RE (2010)	Gibt Quantile der rechtsseitigen F-Verteilung zurück
F.TEST (2010)	Siehe FTEST
F.VERT (2010)	Gibt Werte der Verteilungsfunktion (1-Alpha) einer linksseitigen F-verteilten Zufallsvariablen zurück
F.VERT.RE (2010)	Gibt Werte der Verteilungsfunktion (1-Alpha) einer rechtsseitigen F-verteilten Zufallsvariablen zurück
FINV	Gibt Quantile der F-Verteilung zurück
FISHER	Gibt die Fisher-Transformation für x zurück
FISHERINV	Gibt die Umkehrung der Fisher-Transformation zurück
FTEST	Gibt die Teststatistik eines F-Tests zurück
FVERT	Gibt Werte der Verteilungsfunktion (1-Alpha) einer F-verteilten Zufallsvariablen zurück
G.TEST (2010)	Gibt den einseitigen Wahrscheinlichkeitswert für einen Gaußtest (Normalverteilung) zurück
GAMMA (2013)	Gibt den Wert der Gammafunktion für eine bestimmte Zahl zurück
GAMMA.INV (2010)	Siehe GAMMAINV
GAMMA.VERT (2010)	Siehe GAMMAVERT
GAMMAINV	Gibt Quantile der Gammaverteilung zurück
GAMMALN	Gibt den natürlichen Logarithmus der Gammafunktion zurück, $\Gamma(x)$
GAMMALN.GENAU (2010)	Siehe GAMMALN
GAMMAVERT	Gibt Wahrscheinlichkeiten einer gammaverteilten Zufallsvariablen zurück
GAUSS (2013)	Gibt die Wahrscheinlichkeit, mit der eine standardnormal-verteilte Zufallsvariable zwischen dem Mittelwert der Grundgesamtheit und x Standardabweichungen vom Mittelwert liegt, zurück
GEOMITTEL	Gibt das geometrische Mittel einer Menge positiver Zahlen zurück
GESTUTZTMITTEL	Gibt den Mittelwert einer Datengruppe zurück, ohne die Randwerte zu berücksichtigen
GTEST	Gibt den einseitigen Wahrscheinlichkeitswert für einen Gaußtest (Normalverteilung) zurück

Funktion	Beschreibung
HARMITTEL	Gibt das harmonische Mittel einer Datenmenge zurück
HÄUFIGKEIT	Gibt eine Häufigkeitsverteilung als einspaltige Matrix zurück
HYPGEOM.VERT (2010)	Gibt Wahrscheinlichkeiten einer hypergeometrisch-verteilten Zufallsvariablen zurück
HYPGEOMVERT	Gibt Wahrscheinlichkeiten einer hypergeometrisch-verteilten Zufallsvariablen zurück
KGRÖSSTE	Gibt den k -größten Wert einer Datengruppe zurück
KKLEINSTE	Gibt den k -kleinsten Wert einer Datengruppe zurück
KONFIDENZ	Gibt einen Wert zurück, den Sie zum Erstellen eines Konfidenzintervalls für den Erwartungswert einer Zufallsvariablen verwenden können
KONFIDENZ.NORM (2010)	Siehe KONFIDENZ
KONFIDENZ.T (2010)	Gibt einen Wert zurück, den Sie zum Erstellen eines Konfidenzintervalls für den Erwartungswert einer t-verteilten Zufallsvariablen verwenden können
KORREL	Gibt den Korrelationskoeffizienten einer zweidimensionalen Zufallsgröße zurück, deren Werte in den Zellbereichen <i>Matrix1</i> und <i>Matrix2</i> stehen
KOVAR	Diese Funktion gibt die Kovarianz zurück, den Mittelwert der für alle Datenpunktpaare gebildeten Produkte der Abweichungen
KOVARIANZ.P (2010)	Siehe KOVAR
KOVARIANZ.T (2010)	Diese Funktion gibt die Kovarianz einer Stichprobe zurück, den Mittelwert der für alle Datenpunktpaare gebildeten Produkte der Abweichungen
KRITBINOM	Gibt den kleinsten Wert zurück, für den die kumulierten Wahrscheinlichkeiten der Binomialverteilung größer oder gleich einer Grenzwahrscheinlichkeit sind
KURT	Gibt die Kurtosis (Exzess) einer Datengruppe zurück
LOGINV	Gibt Perzentile der Lognormalverteilung von x zurück, wobei $\ln(x)$ mit den Parametern <i>Mittelwert</i> und <i>Standabwn</i> normal verteilt ist
LOGNORM.INV (2010)	Siehe LOGINV
LOGNORM.VERT (2010)	Siehe LOGNORMVERT
LOGNORMVERT	Gibt Werte der Verteilungsfunktion einer lognormalverteilten Zufallsvariablen zurück, wobei $\ln(x)$ normalverteilt ist mit den Parametern <i>Mittelwert</i> und <i>Standabwn</i>
MAX	Gibt den größten Wert innerhalb einer Argumentliste zurück
MAXA	Gibt den größten Wert einer Liste von Argumenten zurück
MEDIAN	Gibt den Median der angegebenen Zahlen zurück
MIN	Gibt den kleinsten Wert innerhalb einer Argumentliste zurück
MINA	Gibt den kleinsten Wert einer Liste von Argumenten zurück
MITTELABW	Gibt die durchschnittliche absolute Abweichung einer Reihe von Merkmalsausprägungen und ihrem Mittelwert zurück
MITTELWERT	Durch das arithmetische Mittel wird der Zahlenwert bestimmt, der sich bei gleichmäßiger Aufteilung der Summen auf die Werte ergibt
MITTELWERTA	Berechnet den Mittelwert (arithmetisches Mittel) der Werte in der Liste der Argumente



Funktion	Beschreibung
MITTELWERTWENN	Durchschnittswert (arithmetisches Mittel) für alle Zellen in einem Bereich, die einem angegebenen Kriterium entsprechen
MITTELWERTWENNS	Gibt den Durchschnittswert (arithmetisches Mittel) aller Zellen zurück, die mehreren Kriterien entsprechen
MODALWERT	Gibt den häufigsten Wert einer Datengruppe zurück
MODUS.EINF (2010)	Siehe MODALWERT
MODUS.VIELF (2010)	Gibt ein vertikales Array der am häufigsten vorkommenden Werte in einem Array oder Datenbereich zurück
NEGBINOM.VERT (2010)	Siehe NEGBINOMVERT
NEGBINOMVERT	Gibt Wahrscheinlichkeiten einer negativbinomialverteilten Zufallsvariablen zurück
NORM.INV (2010)	Siehe NORMINV
NORM.S.INV (2010)	Gibt Perzentile der Standardnormalverteilung zurück
NORM.S.VERT (2010)	Gibt Werte der Verteilungsfunktion einer standardnormalverteilten Zufallsvariablen zurück
NORM.VERT (2010)	Siehe NORMVERT
NORMINV	Gibt Perzentile der Normalverteilung zurück
NORMVERT	Gibt die Normalverteilung für den angegebenen Mittelwert und die angegebene Standardabweichung zurück
PEARSON	Gibt den Pearsonschen Korrelationskoeffizienten r zurück
PHI (2013)	Gibt den Wert der Dichtefunktion für eine Standardnormalverteilung zurück
POISSON	Gibt Wahrscheinlichkeiten einer poissonverteilten Zufallsvariablen zurück
POISSON.VERT (2010)	Siehe POISSON
QUANTIL	Gibt das Alpha-Quantil einer Gruppe von Daten zurück
QUANTIL.EXKL (2010)	Gibt das k -Quantil einer Gruppe von Daten zurück, wobei k ausschließlich im Bereich 0 bis 1 liegt
QUANTIL.INKL (2010)	Gibt das k -Quantil einer Gruppe von Daten zurück, wobei k einschließlich im Bereich 0 bis 1 liegt
QUANTILSRANG	Gibt den prozentualen Rang (Alpha) eines Werts zurück
QUANTILSRANG.EXKL (2010)	Gibt den prozentualen Rang (Alpha) eines Werts ausschließlich im Bereich 0 bis 1 zurück
QUANTILSRANG.INKL (2010)	Gibt den prozentualen Rang (Alpha) eines Werts einschließlich im Bereich 0 bis 1 zurück
QUARTILE	Gibt die Quartile der Datengruppe zurück
QUARTILE.EXKL (2010)	Gibt die Quartile der Datengruppe zurück, ausschließlich basierend auf Werten von 0 bis 1
QUARTILE.INKL (2010)	Gibt die Quartile der Datengruppe zurück, einschließlich basierend auf Werten von 0 bis 1
RANG	Gibt den Rang zurück, den eine Zahl innerhalb einer Liste von Zahlen einnimmt ►

Funktion	Beschreibung
RANG.GLEICH (2010)	Siehe RANG
RANG.MITTELW (2010)	Gibt die durchschnittliche Rangzahl für eine Zahl innerhalb einer Liste von Zahlen zurück
RGP	Berechnet die Statistik für eine Linie unter Verwendung der Methode der kleinsten Quadrate, zur Berechnung einer geraden Linie, die für die Daten am geeignetsten ist, und gibt eine Matrix zurück, die die Linie beschreibt
RKP	In Regressionsanalysen berechnet diese Funktion eine Exponentialkurve, die möglichst gut an die von Ihnen bereitgestellten Daten angepasst ist, und liefert ein Wertarray, die diese Kurve beschreibt
SCHÄTZER	Gibt den Schätzwert für einen linearen Trend zurück
SCHIEFE	Gibt die Schiefe einer Verteilung zurück
SCHIEFE.P (2013)	Gibt die Schiefe einer Verteilung auf Basis einer Grundgesamtheit zurück
STABW	Schätzt die Standardabweichung ausgehend von einer Stichprobe
STABW.N (2010)	Siehe STABWN
STABW.S (2010)	Siehe STABW
STABWA	Schätzt die Standardabweichung ausgehend von einer Stichprobe
STABWN	Berechnet die Standardabweichung ausgehend von der Grundgesamtheit
STABWNA	Berechnet die Standardabweichung ausgehend von einer als Argumente angegebenen Grundgesamtheit, einschließlich Text und logischer Werte
STANDARDISIERUNG	Gibt den standardisierten Wert einer Verteilung zurück, die durch <i>Mittelwert</i> und <i>Standabwn</i> charakterisiert ist
STANDNORMINV	Gibt Quantile der Standardnormalverteilung zurück
STANDNORMVERT	Gibt Werte der Verteilungsfunktion einer standardnormalverteilten Zufallsvariablen zurück
STEIGUNG	Gibt die Steigung der Regressionsgeraden zurück, die an die in <i>Y_Werte</i> und <i>X_Werte</i> abgelegten Datenpunkte angepasst ist
STFEHLERYX	Gibt den Standardfehler der geschätzten y-Werte für alle x-Werte der Regression zurück
SUMQUADABW	Gibt die Summe der quadrierten Abweichungen von Datenpunkten von deren Stichprobenmittelwert zurück
T.INV (2010)	Siehe TINV
T.INV.2S (2010)	Siehe TINV
T.TEST (2010)	Siehe TTEST
T.VERT (2010)	Gibt die (Student) t-Verteilung der linken Endfläche zurück
T.VERT.2S (2010)	Gibt die (Student) t-Verteilung für zwei Endflächen zurück
T.VERT.RE (2010)	Gibt die (Student) t-Verteilung der rechten Endfläche zurück
TINV	Gibt den t-Wert der t-Verteilung als eine Funktion der Wahrscheinlichkeit und der Freiheitsgrade zurück
TREND	Liefert Werte, die sich aus einem linearen Trend ergeben
TTEST	Gibt die Teststatistik eines Studentschen t-Tests zurück



Funktion	Beschreibung
TVERT	Gibt Werte der Verteilungsfunktion (1-Alpha) einer (Student) t-verteiltern Zufallsvariablen zurück
VAR.P (2010)	Siehe VARIANZEN
VAR.S (2010)	Siehe VARIANZ
VARIANZ	Schätzt die Varianz auf der Basis einer Stichprobe
VARIANZA	Schätzt die Varianz auf der Basis einer Stichprobe
VARIANZEN	Berechnet die Varianz ausgehend von der Grundgesamtheit
VARIANZENA	Berechnet die Varianz ausgehend von der Grundgesamtheit
VARIATION	Liefert Werte, die sich aus einem exponentiellen Trend ergeben
VARIATIONEN	Gibt die Anzahl der Möglichkeiten zurück, um k Elemente aus einer Menge von n Elementen ohne Zurücklegen zu ziehen
VARIATIONEN2 (2013)	Gibt die Anzahl der Möglichkeiten – mit Wiederholungen – für eine angegebene Menge von Objekten zurück
WAHRSCBEREICH	Gibt die Wahrscheinlichkeit für ein von zwei Werten eingeschlossenes Intervall zurück
WEIBULL	Gibt Wahrscheinlichkeiten einer weibullverteilten Zufallsvariablen zurück
WEIBULL.VERT (2010)	Siehe WEIBULL
ZÄHLENWENN	Zählt die nicht leeren Zellen eines Bereichs, deren Inhalte mit einem Suchkriterium übereinstimmen
ZÄHLENWENNNS	Die Zellen eines Bereichs, deren Inhalte mit mehreren Kriterien übereinstimmen, werden gezählt

Datenbankfunktionen

Funktion	Beschreibung
DBANZAHL	Zählt die Anzahl von Zellen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
DBANZAHL2	Zählt die Anzahl von Zellen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
DBAUSZUG	Extrahiert einen einzelnen Wert in einer Spalte einer Liste oder Datenbank, der den angegebenen Bedingungen entspricht
DBMAX	Liefert die größte Anzahl in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entspricht
DBMIN	Liefert die kleinste Anzahl in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entspricht
DBMITTELWERT	Liefert den Mittelwert aus den Werten einer Listen- oder Datenbankspalte, die den von Ihnen angegebenen Bedingungen entsprechen
DBPRODUKT	Multipliziert die Werte in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen ►

Funktion	Beschreibung
DBSTDABW	Schätzt die Standardabweichung einer Grundgesamtheit ausgehend von den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
DBSTDABWN	Berechnet die Standardabweichung einer vollständigen Grundgesamtheit ausgehend von den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
DBSUMME	Fügt die Zahlen in einer Spalte aus einer Liste oder Datenbank hinzu, die den angegebenen Bedingungen entsprechen
DBVARIANZ	Schätzt die Varianz einer Grundgesamtheit ausgehend von einer Stichprobe mit den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
DBVARIANZEN	Berechnet die Varianz einer vollständigen Grundgesamtheit ausgehend von den Zahlen in einer Spalte einer Liste oder Datenbank, die den angegebenen Bedingungen entsprechen
PIVOTDATEN-ZUORDNEN	Gibt Daten aus einem PivotTable-Bericht zurück

Cubefunktionen

Funktion	Beschreibung
CUBEELEMENT	Gibt für den angegebenen Dimensionspfad das Dimensionselement zurück
CUBEELEMENT-EIGENSCHAFT	Gibt für ein Dimensionselement die geforderte Eigenschaft, auch Attribut genannt, zurück
CUBEKPIELEMENT	Gibt die angeforderte Eigenschaft von Key Performance-Indikatoren eines Würfels zurück
CUBEMENGE	Gibt eine definierte Menge von Elementen eines Cubes zurück. Dieses sog. Set stellt einen Teilwürfel dar.
CUBEMENGENANZAHL	Gibt die Anzahl der Elemente eines Sets zurück
CUBERANGELEMENT	Gibt das per Index definierte Element eines Sets zurück
CUBEWERT	Gibt den Wert für eine gegebene Adresse aus einem Datenwürfel zurück

Finanzmathematische Funktionen

Funktion	Beschreibung
AMORDEGRK	Liefert den für eine Abrechnungsperiode anzusetzenden Abschreibungsbetrag auf Basis des französischen Buchführungssystems
AMORLINEARK	Gibt den für eine Abrechnungsperiode anzusetzenden Abschreibungsbetrag auf Basis des französischen Buchführungssystems zurück
AUFGELZINS	Liefert die aufgelaufenen Zinsen (Stückzinsen) eines Wertpapiers mit periodischen Zinszahlungen



Funktion	Beschreibung
AUFGELZINSF	Liefert die aufgelaufenen Zinsen (Stückzinsen) eines Wertpapiers, die bei Fälligkeit ausgezahlt werden
AUSZAHLUNG	Gibt den Auszahlungsbetrag eines voll investierten Wertpapiers am Fälligkeitstermin zurück
BW	Gibt den Barwert einer Investition zurück
DIA	Gibt die arithmetisch-degressive Abschreibung eines Wirtschaftsguts für eine bestimmte Periode zurück
DISAGIO	Gibt den in Prozent ausgedrückten Abzinsungssatz eines Wertpapiers zurück
DURATION	Gibt für einen angenommenen Nennwert von 100 € die Macauley-Dauer zurück
EFFEKTIV	Gibt die jährliche Effektivverzinsung zurück, ausgehend von einer Nominalverzinsung sowie der jeweiligen Anzahl der Zinszahlungen pro Jahr
GDA	Gibt die Abschreibung eines Anlageguts für einen angegebenen Zeitraum unter Verwendung der degressiven Doppelratenabschreibung oder eines anderen von Ihnen angegebenen Abschreibungsverfahrens zurück
GDA2	Gibt die geometrisch-degressive Abschreibung eines Wirtschaftsguts für eine bestimmte Periode zurück
IKV	Gibt den internen Zinsfuß einer Investition ohne Finanzierungskosten oder Reinvestitionsgewinne zurück
ISPMT	Berechnet die während eines bestimmten Zeitraums für eine Investition gezahlten Zinsen
KAPZ	Gibt die Kapitalrückzahlung einer Investition für die angegebene Periode zurück
KUMKAPITAL	Berechnet die aufgelaufene Tilgung eines Darlehens, die zwischen zwei Perioden zu zahlen ist
KUMZINSZ	Berechnet die kumulierten Zinsen, die zwischen zwei Perioden zu zahlen sind
KURS	Gibt den Kurs pro 100 € Nennwert eines Wertpapiers zurück, das periodisch Zinsen auszahlt
KURSDISAGIO	Gibt den Kurs pro 100 € Nennwert eines unverzinslichen Wertpapiers zurück
KURSFÄLLIG	Gibt den Kurs pro 100 € Nennwert eines Wertpapiers zurück, das Zinsen am Fälligkeitsdatum auszahlt
LIA	Gibt die lineare Abschreibung eines Wirtschaftsguts pro Periode zurück
MDURATION	Gibt die modifizierte Macauley-Duration eines Wertpapiers mit einem angenommenen Nennwert von 100 € zurück
NBW	Gibt den Nettobarwert (Kapitalwert) einer Investition auf Basis eines Abzinsungsfaktors für eine Reihe periodischer Zahlungen zurück
NOMINAL	Gibt die jährliche Nominalverzinsung zurück, ausgehend vom effektiven Zinssatz sowie der Anzahl der Verzinsungsperioden innerhalb eines Jahres
NOTIERUNGBRU	Wandelt eine Notierung, die als Dezimalzahl ausgedrückt wurde, in einen Dezimalbruch um
NOTIERUNGDEZ	Wandelt eine Notierung, die als Dezimalbruch ausgedrückt wurde, in eine Dezimalzahl um
PDURATION	Berechnet die Laufzeit einer Investition bei gegebenem Zinssatz sowie eingesetztem und erwartetem Kapital

Funktion	Beschreibung
QIKV	Gibt einen modifizierten internen Zinsfuß zurück, bei dem positive und negative Cashflows mit unterschiedlichen Zinssätzen finanziert werden
RENDITE	Gibt die Rendite eines Wertpapiers zurück, das periodisch Zinsen auszahlt
RENDITEDIS	Gibt die jährliche Rendite eines unverzinslichen Wertpapiers zurück
RENDITEFÄLL	Gibt die jährliche Rendite eines Wertpapiers zurück, das Zinsen am Fälligkeitsdatum auszahlt
RMZ	Gibt die konstante Zahlung einer Annuität pro Periode zurück, wobei konstante Zahlungen und ein konstanter Zinssatz vorausgesetzt werden
TBILLÄQUIV	Rechnet die Verzinsung eines Schatzwechsels (Treasury Bill) in die für Anleihen übliche einfache jährliche Verzinsung um
TBILLKURS	Gibt den Kurs pro 100 € Nennwert eines Schatzwechsels (Treasury Bill) zurück
TBILLRENDITE	Gibt die Rendite eines Schatzwechsels (Treasury Bill) zurück
UNREGER.KURS	Liefert den Kurs pro 100 € Nennwert eines Wertpapiers mit einem unregelmäßigen ersten Zinstermin
UNREGER.REND	Gibt die Rendite eines Wertpapiers mit einem unregelmäßigen ersten Zinstermin zurück
UNREGLE.KURS	Gibt den Kurs pro 100 € Nennwert eines Wertpapiers mit einem unregelmäßigen letzten Zinstermin zurück
UNREGLE.REND	Gibt die Rendite eines Wertpapiers mit einem unregelmäßigen letzten Zinstermin zurück
VDB	Gibt die degressive Doppelratenabschreibung eines Wirtschaftsguts für eine bestimmte Periode oder Teilperiode zurück
XINTZINSFUSS	Gibt den internen Zinsfuß einer Reihe nicht periodisch anfallender Zahlungen zurück
XKAPITALWERT	Gibt den Nettobarwert (Kapitalwert) einer Reihe nicht periodisch anfallender Zahlungen zurück
ZINS	Gibt den Zinssatz einer Annuität pro Periode zurück
ZINSSATZ	Gibt den Zinssatz eines voll investierten Wertpapiers zurück
ZINSTERMNZ	Gibt eine Zahl zurück, die den nächsten Zinstermin nach dem Abrechnungstermin angibt
ZINSTERMTAGE	Gibt die Anzahl der Tage der Zinsperiode zurück, die den Abrechnungstermin einschließt
ZINSTERMTAGNZ	Gibt die Anzahl der Tage vom Abrechnungstermin bis zum nächsten Zinstermin (Kupontermin) an
ZINSTERMTAGVA	Gibt die Anzahl der Tage vom Anfang des Zinstermins bis zum Abrechnungstermin zurück
ZINSTERMVZ	Gibt eine Zahl an, die die letzte Zinszahlung vor dem Abrechnungstermin repräsentiert
ZINSTERMZAHL	Gibt die Anzahl der zwischen dem Abrechnungsdatum und dem Fälligkeitsdatum zahlbaren Zinszahlungen an, und zwar aufgerundet zur nächsten ganzzahligen Zinszahlung
ZINSZ	Gibt die Zinszahlung einer Investition für die angegebene Periode, ausgehend von regelmäßigen, konstanten Zahlungen und einem konstanten Zinssatz zurück
ZSATZINVEST (2013)	Berechnet den Zinssatz, zu welchem ein Kapital investiert werden muss, damit es einen bestimmten Zukunftswert besitzt ▶

Funktion	Beschreibung
ZW	Gibt den zukünftigen Wert (Endwert) einer Investition zurück
ZW2	Gibt den aufgezinnten Wert des Anfangskapitals für eine Reihe periodisch unterschiedlicher Zinssätze zurück
ZZR	Gibt die Anzahl der Zahlungsperioden einer Investition zurück, die auf periodischen, gleichbleibenden Zahlungen sowie einem konstanten Zinssatz basiert

Mathematische und trigonometrische Funktionen

Funktion	Beschreibung
ABRUNDEN	Rundet die Zahl auf <i>Anzahl_Stellen</i> ab
ABS	Liefert den Absolutwert einer Zahl
AGGREGAT (2010)	Gibt ein Aggregat in einer Liste oder Datenbank zurück
ARABISCH (2013)	Wandelt eine römische Zahl in eine arabische Zahl um
ARCCOS	Liefert den Arkuskosinus oder umgekehrten Kosinus einer Zahl
ARCCOSHYP	Liefert den umgekehrten hyperbolischen Kosinus einer Zahl
ARCCOT (2013)	Gibt den Arkuskotangens einer Zahl zurück
ARCCOTHYP (2013)	Gibt den hyperbolischen Arkuskotangens einer Zahl zurück
ARCSIN	Gibt den Arkussinus oder auch umgekehrten Sinus einer Zahl zurück
ARCSINHYP	Gibt den umgekehrten hyperbolischen Sinus einer Zahl zurück
ARCTAN	Gibt den Arkustangens oder auch umgekehrten Tangens einer Zahl zurück
ARCTAN2	Gibt den Arkustangens oder auch umgekehrten Tangens ausgehend von einer x - und einer y -Koordinate zurück
ARCTANHYP	Gibt den umgekehrten hyperbolischen Tangens einer Zahl zurück
AUFRUNDEN	Rundet die Zahl auf <i>Anzahl_Stellen</i> auf
BASIS (2013)	Konvertiert eine Zahl in eine Textdarstellung mit der angegebenen Basis
BOGENMASS	Wandelt Grad in Bogenmaß (Radiant) um
COS	Gibt den Kosinus einer Zahl zurück
COSEC (2013)	Gibt den Kosekans eines Winkels zurück
COSECHYP (2013)	Gibt den hyperbolischen Kosekans eines Winkels zurück
COSHYP	Gibt den hyperbolischen Kosinus einer Zahl zurück
COT (2013)	Gibt den Kotangens eines Winkels zurück
COTHYP (2013)	Gibt den hyperbolischen Kotangens einer Zahl zurück
DEZIMAL (2013)	Konvertiert eine Textdarstellung einer Zahl mit einer angegebenen Basis in eine Dezimalzahl
EXP	Potenziert die Basis e mit der als Argument angegebenen Zahl. Die Konstante e ist die Basis des natürlichen Logarithmus und hat den Wert $2,71828182845904$. ►

Funktion	Beschreibung
FAKULTÄT	Gibt die Fakultät einer Zahl zurück. Die Fakultät einer Zahl wird aus $1*2*3*... * Zahl$ berechnet.
GANZZAHL	Rundet eine Zahl auf die nächste kleinere ganze Zahl ab
GERADE	Rundet eine Zahl auf die nächste gerade ganze Zahl auf
GGT	Gibt den größten gemeinsamen Teiler zurück
GRAD	Wandelt ein Bogenmaß (Radiant) in Grad um
KGV	Gibt das kleinste gemeinsame Vielfache der als Argumente angegebenen ganzen Zahlen zurück
KOMBINATIONEN	Berechnet, wie viele Gruppen aus einer bestimmten Anzahl von Elementen gebildet werden können
KOMBINATIONEN2 (2013)	Gibt die Anzahl der Kombinationen mit Wiederholung für eine angegebene Anzahl von Elementen zurück
KÜRZEN	Schneidet die Kommastellen der Zahl ab und gibt als Ergebnis eine ganze Zahl zurück
LN	Gibt den natürlichen Logarithmus einer Zahl zurück
LOG	Gibt den Logarithmus einer Zahl zu der angegebenen Basis zurück
LOG10	Gibt den Logarithmus einer Zahl zur Basis 10 zurück
MDET	Gibt die Determinante einer Matrix zurück
MEINHEIT (2013)	Gibt die Einheitsmatrix für die angegebene Größe zurück
MINV	Gibt die Inverse einer Matrix (die zu einer Matrix gehörende Kehrmatrix) zurück
MMULT	Gibt das Produkt zweier Matrizen zurück
OBERGRENZE	Rundet eine Zahl betragsmäßig auf das kleinste Vielfache von Schritt auf
OBERGRENZE.GENAU (2010)	Siehe OBERGRENZE
OBERGRENZE.MATHEMATIK (2013)	Rundet eine Zahl auf die nächste Ganzzahl oder auf das kleinste Vielfache des angegebenen Schritts auf
PI	Gibt den Wert Pi zurück, die mathematische Konstante (3,14159265358979) mit einer Genauigkeit von 15 Stellen
POLYNOMIAL	Gibt den Polynomkoeffizienten einer Gruppe von Zahlen zurück
POTENZ	Gibt als Ergebnis eine potenzierte Zahl zurück
POTENZREIHE	Gibt die Summe von Potenzen (zur Berechnung von Potenzreihen und dichotomen Wahrscheinlichkeiten) zurück
PRODUKT	Multipliziert die Argumente und gibt das Produkt zurück
QUADRATSUMME	Summiert die quadrierten Argumente
QUOTIENT	Gibt den ganzzahligen Anteil einer Division zurück
REST	Gibt den Rest einer Division zurück. Das Ergebnis hat dasselbe Vorzeichen wie Divisor
RÖMISCH	Wandelt eine arabische Zahl in eine römische Zahl als Text um
RUNDEN	Rundet eine Zahl auf eine bestimmte Anzahl von Dezimalstellen
SEC (2013)	Gibt den Sekans eines Winkels zurück



Funktion	Beschreibung
SECHYP (2013)	Gibt den hyperbolischen Sekans eines Winkels zurück
SIN	Gibt den Sinus einer Zahl zurück
SINHYP	Gibt den hyperbolischen Sinus einer Zahl zurück
SUMME	Summiert die Argumente
SUMMENPRODUKT	Multipliziert die einander entsprechenden Komponenten der angegebenen Matrizen miteinander und gibt die Summe dieser Produkte zurück
SUMMEWENN	Addiert Zahlen, die mit dem Suchkriterium übereinstimmen
SUMMEWENNS	Addiert Zahlen, die mit mehreren Suchkriterien übereinstimmen
SUMMEX2MY2	Summiert für zusammengehörige Komponenten zweier Matrizen die Differenzen der Quadrate
SUMMEX2PY2	Summiert für zusammengehörige Komponenten zweier Matrizen die Summen der Quadrate
SUMMEXMY2	Summiert für zusammengehörige Komponenten zweier Matrizen die quadrierten Differenzen
TAN	Gibt den Tangens einer Zahl zurück
TANHYP	Gibt den hyperbolischen Tangens einer Zahl zurück
TEILERGEBNIS	Gibt ein Teilergebnis in einer Liste oder Datenbank zurück
UNGERADE	Rundet eine Zahl auf die nächste ungerade ganze Zahl auf
UNTERGRENZE	Rundet eine Zahl ab
UNTERGRENZE.GENAU (2010)	Siehe UNTERGRENZE
UNTERGRENZE.MATHEMATIK (2013)	Rundet eine Zahl auf die nächste Ganzzahl oder auf das kleinste Vielfache des angegebenen Schritts ab
VORZEICHEN	Gibt das Vorzeichen einer Zahl zurück
VRUNDEN	Gibt eine auf das gewünschte Vielfache gerundete Zahl zurück
WURZEL	Gibt die Quadratwurzel einer Zahl zurück
WURZELPI	Gibt die Wurzel aus der mit Pi multiplizierten Zahl zurück
ZUFALLSBEREICH	Gibt eine ganze Zufallszahl aus dem festgelegten Bereich zurück
ZUFALLSZAHL	Gibt eine Zufallszahl zwischen 0 und 1 zurück
ZWEIFAKULTÄT	Gibt die Fakultät zu Zahl mit der Schrittlänge 2 zurück

Konstruktionsfunktionen

Funktion	Beschreibung
BESSELI	Gibt die modifizierte Besselfunktion der ersten Art $I_n(z)$ zurück, die der für rein imaginäre Argumente ausgewerteten Besselfunktion J_n entspricht
BESSELJ	Gibt die Besselfunktion der ersten Art $J_n(x)$ zurück
BESSELY	Gibt die modifizierte Besselfunktion 2. Art $K_n(x)$ zurück
BESSELY	Gibt die Besselfunktion zweiter Art $Y_n(x)$ zurück, die auch als Webersche Funktion oder Neumannsche Funktion bezeichnet wird
BININDEZ	Wandelt eine binäre Zahl (Dualzahl) in eine dezimale Zahl um
BININHEX	Wandelt eine binäre Zahl (Dualzahl) in eine hexadezimale Zahl um
BININOKT	Wandelt eine Binärzahl (Dualzahl) in eine Oktalzahl um
BITVERSCHIEB (2013)	Gibt eine Zahl zurück, die um Verschiebebetrag Bits nach links verschoben ist
BITODER (2013)	Gibt ein bitweises »Oder« zweier Zahlen zurück
BITRVERSCHIEB (2013)	Gibt eine Zahl zurück, die um Verschiebebetrag Bits nach rechts verschoben ist
BITUND (2013)	Gibt ein bitweises »Und« zweier Zahlen zurück
BITXODER (2013)	Gibt ein bitweises »Ausschließliches Oder« zweier Zahlen zurück
DELTA	Überprüft, ob zwei Werte gleich sind
DEZINBIN	Wandelt eine dezimale Zahl in eine binäre Zahl (Dualzahl) um
DEZINHEX	Wandelt eine dezimale Zahl in eine hexadezimale Zahl um
DEZINOKT	Wandelt eine Dezimalzahl in eine Oktalzahl um
GAUSSFEHLER	Gibt die Gaußsche Fehlerfunktion zurück
GAUSSF.GENAU (2010)	Siehe GAUSSFEHLER
GAUSSFKOMPL	Gibt das Komplement zur Gaußschen Fehlerfunktion zurück
GAUSSFKOMPL.GENAU (2010)	Siehe GAUSSFKOMPL
GGANZZAHL	Gibt den Wert 1 zurück, wenn $Zahl \geq Schritt$ gilt; andernfalls gibt sie 0 (Null) zurück
HEXINBIN	Wandelt eine hexadezimale Zahl in eine Binärzahl um
HEXINDEZ	Wandelt eine hexadezimale Zahl in eine dezimale Zahl um
HEXINOKT	Wandelt eine hexadezimale Zahl in eine Oktalzahl um
IMABS	Gibt den Absolutwert (Modul) einer komplexen Zahl zurück
IMAGINÄRTEIL	Gibt den Imaginärteil einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt
IMAPOTENZ	Potenziiert eine komplexe Zahl, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt, mit einer ganzen Zahl
IMARGUMENT	Gibt das Argument (Theta) zurück, einen Winkel, der als Bogenmaß ausgedrückt wird
IMCOS	Gibt den Kosinus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ vorliegt



Funktion	Beschreibung
IMCOSEC (2013)	Gibt den Kosekans einer komplexen Zahl zurück
IMCOSECHYP (2013)	Gibt den hyperbolischen Kosekans einer komplexen Zahl zurück
IMCOSHYP (2013)	Gibt den hyperbolischen Kosinus einer komplexen Zahl zurück
IMCOT (2013)	Gibt den Kotangens einer komplexen Zahl zurück
IMDIV	Gibt den Quotienten zweier komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
IMEXP	Gibt die algebraische Form einer in exponentieller Form vorliegenden komplexen Zahl zurück, wobei deren Exponent als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
IMKONJUGIERTE	Gibt die konjugiert komplexe Zahl zu einer komplexen Zahl zurück, wobei die komplexe Zahl als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
IMLN	Gibt den natürlichen Logarithmus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
IMLOG10	Gibt den Logarithmus einer komplexen Zahl zur Basis 10 zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
IMLOG2	Gibt den Logarithmus einer komplexen Zahl zur Basis 2 zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
IMPRODUKT	Gibt das Produkt komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
IMREALTEIL	Gibt den Realteil einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
IMSEC (2013)	Gibt den Sekans einer komplexen Zahl zurück
IMSECHYP (2013)	Gibt den hyperbolischen Sekans einer komplexen Zahl zurück
IMSIN	Diese Funktion gibt den Sinus einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
IMSINHYP (2013)	Gibt den hyperbolischen Sinus einer komplexen Zahl zurück
IMSUB	Gibt die Differenz zweier komplexer Zahlen zurück, die beide als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
IMSUMME	Gibt die Summe komplexer Zahlen zurück, die als Zeichenfolgen der Form $x + yi$ oder $x + yj$ erwartet werden
IMTAN (2013)	Gibt den Tangens einer komplexen Zahl zurück
IMWURZEL	Gibt die Quadratwurzel einer komplexen Zahl zurück, die als Zeichenfolge der Form $x + yi$ oder $x + yj$ eingegeben wird
KOMPLEXE	Wandelt den Real- und Imaginärteil in eine komplexe Zahl um ($x + yi$ oder $x + yj$)
OKTINBIN	Wandelt eine Oktalzahl in eine Binärzahl (Dualzahl) um
OKTINDEZ	Wandelt eine Oktalzahl in eine Dezimalzahl um
OKTINHEX	Wandelt eine Oktalzahl in eine Hexadezimalzahl um
UMWANDELN	Wandelt eine Zahl von einem Maßsystem in ein anderes um

Webfunktionen

Funktion	Beschreibung
URLCODIEREN (2013)	Gibt eine URL-codierte Zeichenfolge zurück
WEBDIENST (2013)	Empfängt (XML-) Daten vom angegebenen Webservice
XMLFILTERN (2013)	Extrahiert ausgewählte Daten aus einer XML-Datei

Anhang C

**Die Beispieldateien
zum Buch**

Alle im Buch beschriebenen Beispiele finden Sie im Download-Ordner \Ms5-235 zu diesem Buch. In der folgenden Tabelle sind jeweils der Name und der Speicherort der Beispieldatei zusammen mit einer Kurzbeschreibung aufgelistet.

Hinweis Sie können die ZIP-Datei mit allen Beispieldateien im Internet von einer der beiden folgenden Adressen herunterladen:

www.microsoft-press.de/support/9783866452350

oder

<http://msp.oreilly.de/support/2314/773>

Beachten Sie auch die Hinweise zur Handhabung der Beispiel- und Übungsdateien im jeweiligen Kapitel.

Dateiname und Pfad	Beschreibung
\Kap01	
<i>Excel_Uebungsbeispiel.xlsx</i>	Im Kapitel behandelte Praxisbeispiele mit Excel 2013
<i>Excel_Uebungsbeispiel_leer.xlsx</i>	Übungsdaten für die Praxisbeispiele mit Excel 2013
<i>PowerPivot_Beispiel_bearbeitet.xlsx</i>	Praxisbeispiele mit dem PowerPivot-Add-In zur Datenanalyse
<i>PowerPivot_Beispiel_leer.xlsx</i>	Übungsdaten für die Praxisbeispiele mit dem PowerPivot-Add-In
<i>PowerView_Beispiel_bearbeitet.xlsx</i>	Praxisbeispiele mit dem PowerView-Add-In zur Datenanalyse
<i>PowerView_Beispiel_leer.xlsx</i>	Übungsdaten für die Praxisbeispiele mit dem PowerPivot-Add-In
<i>Olympische Spiele und Haiangriffe.xlsx</i>	Power View-Praxisbeispiel von Robert Bruckner, Microsoft
\Kap02	
<i>Kap02.xlsx</i>	Alle Beispiele zu den Themen aus Kapitel 2
\Kap04	
<i>Kap04_BF.xlsx</i>	Beispiele zur Anwendung von Funktionen in der bedingten Formatierung
<i>Kap04_DP.xlsx</i>	Beispiele zu Funktionen in der Datenprüfung
<i>Kap04_Namen.xlsx</i>	Beispiele zur Anwendung von Funktionen in Namen
\Kap05	
<i>Geometrie.xlsm</i>	Beispieldatei zur Programmierung benutzerdefinierter Funktionen
<i>Quersumme.xlsm</i>	Beispieldatei zur Programmierung benutzerdefinierter Funktionen
\Kap06	
<i>Datum_Zeit.xlsx</i>	Alle Beispiele zum Kapitel 6 – Datum- und Zeitfunktionen
\Kap07	
<i>Text_Daten.xlsx</i>	Alle Beispiele zum Kapitel 7 – Text- und Datenfunktionen
\Kap08	
<i>Wahrheitswerte.xlsx</i>	Alle Beispiele zum Kapitel 8 – Logikfunktionen



Dateiname und Pfad	Beschreibung
\Kap09	
<i>Adresse.xlsx</i>	Beispiele zur Funktion ADRESSE()
<i>Bereich.verschieben.xlsx</i>	Beispiele zur Funktion BEREICH.VERSCHIEBEN()
<i>Bezugsfunktionen.xlsx</i>	Beispiele zu den Such- und Bezugsfunktionen
<i>Test.docx</i>	Word-Dokument mit Textmarke zum Testen der Funktion HYPERLINK() aus <i>Bezugsfunktionen.xlsx</i>
<i>exampleRTD.dll</i>	Die Beispiel-DLL zum Testen der Funktion RTD()
<i>exampleRTD.xlsm</i>	Die Beispiel-Mappe (2007/2010) zum Testen der Funktion RTD()
<i>Verweise.xlsx</i>	Beispiele zu den Verweis-Funktionen
\Kap10	
<i>Informationen.xlsx</i>	Alle Beispiele zu Kapitel 10 – Informationsfunktionen
\Kap11	
<i>Mittelwert.xlsx</i>	Beispiele zu allen Mittelwert- und verwandten Funktionen
<i>Regression.xlsx</i>	Beispiele zur Regression und verwandten Funktionen
<i>Symmetrie.xlsx</i>	Beispiele zu den Funktionen KURTOSIS() und SCHIEFE()
<i>Varianz.xlsx</i>	Beispiele zu Varianzen, Standardabweichungen und verwandten Funktionen
<i>Wahrscheinlichkeit.xlsx</i>	Beispiele zu den Wahrscheinlichkeitsfunktionen
<i>WEBZUGRIFF.xlsx</i>	Beispiel Pivot-Auswertung von Webzugriffen
<i>Zählen.xlsx</i>	Beispiele zu Anzahlberechnungen und verwandten Funktionen
\Kap12	
<i>DBFUNKTION leer.xlsx</i>	Mit dieser Mappe können Sie alle Beispiele zu den DB-Funktionen selbst nachvollziehen
<i>DBFUNKTIONEN.xlsx</i>	Die fertigen Beispiele zu den DB-Funktionen
\Kap13	
<i>Cube Test.cub</i>	Offline-Cube-Datei, um die Cubefunktionen auch ohne Analysis Services testen zu können
<i>cubeTest.xlsx</i>	Mappe mit den Beispielanwendungen der Cubefunktionen
<i>offLine.odc</i>	Office-Datenverbindungsdatei für den Zugriff über Arbeitsmappen-Verbindungen
<i>offLine.oqy</i>	Office-Datenverbindungsdatei für den Zugriff über MS Query
\Kap14	
<i>Abschreibungsrechnung.xlsx</i>	Beispiele zu den Abschreibungsfunktionen
<i>Einfache Zinsrechnung.xlsx</i>	Beispiele zu Zinsberechnungen
<i>Investitionsrechnung.xlsx</i>	Beispiele zu Investitionsberechnungen
<i>Kursrechnung.xlsx</i>	Beispiele zu Börsenfunktionen
<i>Rentenrechnung.xlsx</i>	Beispiele zu Rentenfunktionen ▶

Dateiname und Pfad	Beschreibung
<i>Sonstiges.xlsx</i>	Beispiele zu sonstigen finanzmathematischen Funktionen
<i>Tilgungsrechnung.xlsx</i>	Beispiele zu Tilgungsfunktionen
<i>Zinseszinsrechnung.xlsx</i>	Beispiele zu Zinseszinsberechnungen
\Kap15	
<i>AGGREGAT.xlsx</i>	Beispiele zur Funktion AGGREGAT() – ab Excel 2010
<i>ARCCOSHYP.xlsx</i>	Beispiele zur Funktion ARCCOSHYP()
<i>COSHYP.xlsx</i>	Beispiele zur Funktion COSHYP()
<i>Kap15.xlsx</i>	Alle weiteren Beispiele zu Kapitel 15 für Excel 2007/2010 – Math. & Trigonometrie
<i>Kap15_XL2013.xlsx</i>	Alle weiteren Beispiele zu Kapitel 15 für Excel 2013 – Math. & Trigonometrie (Formeldokumentation mit FORMELTEXT())
<i>OBERUNTERGRENZEGENAU.xlsx</i>	Beispiele zur Abgrenzung von OBERGRENZE()/UNTERGRENZE() und OBERGRENZE.GENAU()/UNTERGRENZE.GENAU()
<i>SINHYP.xlsx</i>	Beispiele zur Funktion SINHYP()
<i>TANHYP.xlsx</i>	Beispiele zur Funktion TANHYP()
\Kap16	
<i>Bessel_I.xlsx</i>	Beispiele zur Funktion BESSELI()
<i>Bessel_J.xlsx</i>	Beispiele zur Funktion BESSELJ()
<i>Bessel_K.xlsx</i>	Beispiele zur Funktion BESSELK()
<i>Bessel_Y.xlsx</i>	Beispiele zur Funktion BESSELY()
<i>Bitoperationen.xlsx</i>	Beispiele zu den Funktionen für Bitoperationen
<i>Delta.xlsx</i>	Beispiele zur Funktion DELTA()
<i>Gauss.xlsx</i>	Beispiele zu den Funktionen GAUSSFEHLER() und GAUSSFKOMPL()
<i>GGanzzahl.xlsx</i>	Beispiele zur Funktion GGANZZAHL()
<i>Komplex.xlsm</i>	Beispiele zu Funktionen mit komplexen Zahlen – als Argument und/oder Ergebnis einschließlich VBA-Formatierungsmakro
<i>Konvert.xlsx</i>	Beispiele zu den Konvertierungsfunktionen verschiedener Zahlensysteme
<i>Umwandeln.xlsx</i>	Beispiele zur Funktion UMWANDELN()
\Kap17	
<i>Webfunktionen.xlsx</i>	Beispiele zu den neuen Webfunktionen in Excel 2013
<i>XPath.pptx</i>	Präsentation mit einem Einstieg in die Abfragesprache für XML-Dateien XPATH

Beispielverzeichnis

A

Abrechenbare Arbeitstage im Projektzeitraum 218, 220
Abschreibung des ersten Jahres 626
Abschreibungsbetrag für Wirtschaftsgut 625
Abschreibungsbetrag je Periode 685
Abschreibungsplan erstellen 640
Abweichungsanalyse 549
Addition aller Werte aus ungeraden Spalten 297
Aktuelles Datum und Uhrzeit anzeigen 213
Aktuelles Tagesdatum berechnen 211
Änderungen visualisieren 335
Anrufverhalten analysieren 378
Arithmetisch-degressive Abschreibung berechnen 634
Artikelnummern zerlegen 251
Auswahl auf Text testen 326
Auswahl auf Zahl testen 328
Auswertung von Medikamentenverträglichkeitstests 527
Außendienstbesuche analysieren 458

B

Beschäftigungszeit berechnen 206
Bestellungen analysieren 349
Blattname berechnen 140
Blocksatzformatierung berechnet 258
Brennleistung von Glühbirnen 463
Bundeswertpapiere beurteilen 638

D

Dateiname berechnen 139
Datentyp erkennen und korrekt verarbeiten 332
Datumsangaben verstehen 329
Datumstext in Datumswerte umwandeln 209
Definition des Areatangens 715

Drei niedrigste Besucherzahlen ermitteln 420
Durchschnittlichen Umsatz eines Produkts berechnen 582
Durchschnittsumsatz berechnen 450
Dynamische Diagramme anlegen 283
Dynamische Liste erstellen 282

E

Effektivverzinsung berechnen 639
Effektivverzinsung einer Versicherungsprämie berechnen 687
Effektivverzinsung eines Ratenkredits berechnen 687
Enddatum für Baustelleneinsatz ermitteln 210
Entgelte entsprechend der vereinbarten Tarifierhöhung berechnen 761
Erbschaft zusammenrechnen 746
Erfolgsanalyse von Webitemarketing 496, 522
Erfolgswahrscheinlichkeit berechnen 364, 461

F

Fahrplansuche durchführen 308
Fakturierte Umsätze zählen 574
Fälligkeitstermin einer Finanzierung ermitteln 217
Fehlende Einträge in Listen ergänzen 352
Fehleranzeige vermeiden 328
Fehlerwert #NV erkennen 330
Fehlerwerte farbig hervorheben 314
Fehlerwerte ignorieren 320, 325
Fehlerwerte per Makro auswerten 315
Flexible Fehleranzeige einrichten 318, 319
Fortlaufende Nummerierung erzeugen 309, 336
Freie Tage-Liste anlegen 148
Frühbucherrabatt oder Normalpreis? 290
Fußballturnier organisieren 750

G

Geburtstagsliste nach Tagen sortieren 222
 Gefilterte Rechnungen zählen 575
 Geometrisch-degressive Abschreibung berechnen 642
 Gerade Zeilen hervorheben 321
 Geringsten Umsatz ermitteln 444
 Gesamtdurchschnittsgeschwindigkeit errechnen 409
 Gesperrte Zellen visualisieren 336
 Gewinnchancen beim Lotto berechnen 416
 Gleichzeitiges Auftreten von Läufern 749
 Gleitzeitige zählen 563
 Großbuchstabe am Wortanfang 241
 Gültigkeitsprüfung für Eingaben 322
 Günstigsten Anbieter finden 303
 Guthaben am Ende des Jahres berechnen 639

H

Häufigkeitsverteilung ermitteln 505
 Höchsten Umsatz in einer Region ermitteln 579
 Hyperlink auf aktuelles Verzeichnis setzen 317
 Hypothekendarlehen berechnen 639

I

Identische Inhalte signalisieren 149
 Indirekte Adressierung verstehen 280
 Investitionen berechnen 633
 Investitions-Analyse durchführen 292
 ISMA-Rendite ermitteln 667

J

Jahreswerte aus Datumsspalte ermitteln 212

K

Kalenderwoche anzeigen 212, 215
 Kapitalwertmethode verstehen 633
 Key Performance-Indikator hinterlegen 609
 Kreditberater 158
 Kurs einer Anleihe berechnen 658

L

Länge der zu einer Brücke ansteigende Strecke berechnen 735
 Laufende Uhr im Blatt anzeigen 296

Laufzeit einer Finanzierung ermitteln 205
 Lautstärke in Dezibel berechnen 754
 Leere Zellen bei Mittelwertbildung korrekt verarbeiten 323
 Leerzeichen entfernen 239
 Leerzeichenprüfung 156
 Leistungsanalyse von Glühbirnen 517
 Letzte Zelle im Bereich finden 280
 Lieferdatum für Produkt und Kunde ermitteln 578
 Lineare Abschreibung berechnen 656
 Link in Word-Dokumente einfügen 287
 Link zu Office Online einfügen 287
 Listeneinträge zählen 352
 Listeninhalte mit Kategoriennummern versehen 233
 Listenlänge kontrollieren 285

M

Maximalen Wert aus Liste ermitteln 439
 Medikamentenverträglichkeit untersuchen 388
 Mehrspaltige Suche in Listen 302
 Minutenwerte in Zeitangaben 216
 Mischzins berechnen 664
 Mitarbeiteridentifikation mit dem Unternehmen analysieren 381
 Mittelwertstreuungen in Stichproben 509
 Monatsletzten berechnen 217
 Monatsletzten/-ersten berechnen 208
 Monatswerte aus Datumsspalte ermitteln 216
 Muster in Texten suchen 249

N

Nachname extrahieren 246
 Nachrichten verschlüsseln 275
 Neigungswinkel berechnen 720
 Normbestimmung von Matrizen 294
 Normzeiten berechnen 227
 Notierungen vergleichen 661

O

Onlinebestellungen analysieren 353, 384, 424

P

Parallele elektrische Leiter 722
 Pfad der Arbeitsmappe prüfen 139
 Preis bei Gläubigerwechsel 655
 Preis in Liste finden 290

Preiskalkulation durchführen 715
 Produktbestellungen analysieren 427, 430
 Produkte mit geringem Umsatz ermitteln 580
 Produktgruppe aus Artikelnummer bestimmen 251
 Prognose auf Basis erhobener Daten 551
 Prognose für Bestellungen 503
 Prüfsummen bilden 275

R

Radioaktiven Zerfall berechnen 743
 Rangliste der Monatsumsätze ermitteln 490
 Realzinsen berechnen 628
 Rendite berechnen 681, 684
 Rendite einer Bundesanleihe berechnen 672
 Rentenberechnung durchführen 633, 673, 691, 703, 706
 Rückstellung von Weihnachtsgeld 206

S

Sachinvestition beurteilen 643, 659
 Schaltalgebra verstehen 275
 Signifikanzniveau für kritischen Wert von Stichproben 540
 Skonto beurteilen 689
 Sonderzeichen als Funktionsausdruck 260
 Sonntage in Datumsspalte hervorheben 225
 Spalten automatisch beschriften 279
 Standardnormalverteilte Werte einer Produktlebensdauer berechnen 466
 Statistische Beurteilung von Ausfallfolgen 511, 515
 Stichprobenanalyse durchführen 368, 372, 375
 Streubreite von Daten analysieren 543
 Stückzinsen berechnen 629
 Summe der überdurchschnittlichen Umsätze ermitteln 588
 Summe für gefilterte Werte ermitteln 793

T

Testreihenvergleiche anstellen 538
 Text in einer Spalte suchen 241
 Text in Zahlen umwandeln 257, 259
 Textbegrenzung kontrollieren 244
 Texte durch andere ersetzen 235
 Texte und Berechnungen zu einem Satz verbinden 255
 Textlänge überprüfen 244
 Textvergleich durchführen 240, 243
 Textwerte erkennen 326
 Tilgungsanteil berechnen 645
 Tilgungsrechnung durchführen 633, 647, 648, 673, 692, 704, 707

Top 3-Umsätze ermitteln 418
 Top-Elemente kennzeichnen 155
 Trefferwahrscheinlichkeit ermitteln 361
 Trendberechnungen 531
 Trendlinie für Bestellungen ermitteln 520

U

Uhrzeittext in Uhrzeitwert umwandeln 228
 Umfrageanalyse durchführen 390
 Umlaute ersetzen 256
 Umsatzdurchschnitt berechnen 405
 Umsätze analysieren 480
 Umsätze klassifizieren 412
 Umsätze summieren mit Filter 782
 Umsätze summieren mit Mehrfachfilter 784
 Umsätze zählen 351
 Umsatzrangliste erstellen 612
 Umsatzrangliste und -streuung 483
 Umsatzschwächstes Jahr ermitteln 613
 Umsatzstärkste Filiale ermitteln 613
 Umsatzstatistik pro Kopf anfertigen 486
 Umsatzsteigerung analysieren 404
 Umsatzstreuung ausgewählter Produkte ermitteln 585, 586, 591, 592
 Umsatzwerte filtern 615
 Ungerade Zeilen hervorheben 327
 Unterscheidbare Gegenstände in einer Reihe anzuordnen 744
 Unterschiede hervorheben 152

V

Variationen berechnen 554
 Verteilung von Websiteclicks ermitteln 434
 Volle Stunden aus Zeiterfassungsdaten berechnen 221
 Vorhersage von Bestellungen 501
 Vorname extrahieren 245
 Vorschüssigen Zinssatz in nachschüssigen umwandeln 675, 677

W

Wachstums- oder Zerfallsprozessen 753
 Wahrheitswerte farbig darstellen 324
 Wahrheitswerte in Umgangssprache umwandeln 324
 Wahrscheinlichkeit berechnen 743
 Wahrscheinlichkeit der Erreichung einer Produktlebensdauer berechnen 471

Wahrscheinlichkeit fehlerhafter Verpackungen berechnen 365
 Wahrscheinlichkeit für Erfolg von Vorbeugemaßnahmen berechnen 368, 372, 375
 Wahrscheinlichkeit von Reifenschäden berechnen 478
 Websitezugriffe analysieren 348, 442, 446, 448
 Wechselrechnung durchführen 636, 654, 670
 Wechselsumme ermitteln 631
 Welches Produkt erzielt welchen Umsatz? 594
 Wert einer Bundesanleihe bestimmen 629
 Werte aus Trefferzeile ermitteln 300
 Werte in Text umwandeln 236
 Wertetyp erkennen und korrekt verarbeiten 332
 Wetterdaten ermitteln 864, 865
 Wie viel Besuche bis zum Vertragsabschluss? 458
 Winkel berechnen 728
 Wochenende hervorheben 145
 Wochentag auswählen 307
 Wortanfang mit Großbuchstabe 241

Z

Zahl in Thai-Text umwandeln 232
 Zahlen in Währungstext umwandeln 234
 Zahlungsziel berechnen 141, 202, 203
 Zahlungsziel in Rechnungsformular einfügen 252
 Zahlwerte erkennen 328
 Zeichenfolge in einer Spalte suchen 241
 Zeichenfolge suchen 238
 Zeichenfolgen analysieren 238
 Zeilenumbrüche aus Datenlisten entfernen 247
 Zielerreichungsanalyse bei Umsätzen 562, 565
 Zielerreichungswahrscheinlichkeit berechnen 558
 Zimmer mit möglichst großen quadratischen Platten auslegen 747
 Zimmer tapezieren 731
 Zimmerflächen berechnen 583
 Zinsen taggenau berechnen 224
 Zinsertrag berechnen 644
 Zinseszins berechnen 632, 691, 703, 706
 Zusammenhang zwischen Mailing und Websitezugriffe ermitteln 452, 455
 Zusammenhang zwischen Websiteclicks und Bestellungen ermitteln 520
 Zusammenhang zwischen Websitezugriffen und Bestellungen ermitteln 474

Stichwortverzeichnis

1900-Datumssystem 207
1904-Datumssystem 207

A

A1-Schreibweise 279
Abrechenbaren Arbeitstage 218, 220
ABRUNDEN() 714, 752
ABS() 715
Abschreibung 624, 626, 634,
640–641, 685
Abschreibungsprozentsatz 624, 626,
640–641, 685
Abschreibungsrechnung 620
Absoluter Bezug 90, 150, 600
Absolutwert 715, 805, 830
ACCRINT() 627
ACCRINTM() 629
Achsen 872
ACHSENABSCHNITT() 347
ActiveX-Steuerelemente 290
Addieren von Brüchen 749
Addieren von Werten aus einer
Datenbank 587
Add-In 172, 194
 erstellen 194
Add-Ins-Manager 194
Addition 83
ADDRESS() 279
Adresse der Zelle 333
ADRESSE() 279
AGGREGAT() 716
Aktuelle Uhrzeit 213
Aktueller Blattname 140
Aktuelles Array 97
Aktuelles Datum 210, 213
Aktuelles Datum und Uhrzeit 213
Aktuelles Verzeichnis ermitteln 316
Algebraische Form 836
Algorithmus 171
Alle Spuren entfernen 111–112
Alpha-Quantil 479
Als Werte kopieren 107
AMORDEGRC() 623
AMORDEGRK() 623
AMORLINC() 625
AMORLINEARK() 625
Analysis Services 602
Ancestor 873
Ancestor-or-self 873
AND() 267
Ankathete 719, 727, 734, 739, 788
Annuitäten 186
 Darlehen 646
 Tilgung 633, 645–646, 673, 691,
 700, 704, 707
 Zahlung 619
ANOVA-Berechnung 380
Ansichtsoptionen 111
Antivalenz 827
Anzahl der Arbeitstage 218–219
Anzahl der Spalten 298
Anzahl von Zellen 573
 nicht leere 575
ANZAHL() 349
ANZAHL2() 351
ANZAHLLEEREZELLEN() 167, 352
Application 192
Äquivalenzprinzip 632, 650
ARABISCH() 711, 718
Arabische Zahl 771
Arbeitsbereich 173
Arbeitsmappen 33
 Verbindung 604
ARBEITSTAG.INTL() 202
ARBEITSTAG() 201
Arbeitstage 218, 220
ARCCOS() 719
ARCCOSHYP() 721
ARCCOT() 722
ARCCOTHYP() 724
ARCSIN() 725
ARCSINHYP() 726
ARCTAN() 727, 747
ARCTAN2() 729
ARCTANHYP() 715, 730
Areefunktion 721
Areakosinus 726
AREAS() 284
Areasinus 726
Areatangens 726
Argumente 117–118, 120, 171, 179, 833
 anzeigen 130
 der DB()-Funktionen 569
 eingeben 124
 optionale 117, 172, 179–180
Arithmetische Form 829
Arithmetisches Mittel 447, 449
 aus Datenbank 581
Arkusfunktion 720
Arkuskosinus 719
Arkustangens 727, 729
Arkustangens2 729
ASC() 231
ASCII() 862
Assistent 122
Asymmetrie 503
Attribute 873
AUFGELZINS() 627
AUFGELZINSF() 629
Aufrunden 731
AUFRUNDEN() 731
Ausfüllen per Doppelklick 106
Ausfüllkästchen 105
Ausrichtung von Text 334
Austausch von Zeilen und Spalten 293
AUSZAHLUNG() 630
AutoKorrektur 131
 zur Formeleingabe nutzen 132

AutoVervollständigen 749
 AVEDEV() 445
 AVERAGE() 447
 AVERAGEA() 449
 AVERAGEIF() 451
 AVERAGEIFS() 454

B

BAHTTEXT() 232
 Barwert 631
 BASIC() 170
 Basis 754
 Basis 10 754
 Basis e 753
 BASIS() 732, 742
 Basset-Funktion 850
 Baustelleneinsatz
 Ende berechnen 210
 Bearbeitenmodus 150
 Bearbeitungsmodus 110
 Bedingte Formatierung 144, 150, 158,
 263, 314, 336
 mit Formel 225
 Problembhebung 163
 suchen 162
 Teilergebnisse formatieren 161
 Tipps 163
 Bedingungen 181
 logische 179, 181
 ODER-verknüpft 570
 UND-verknüpft 570
 Befehlscode 174
 Befehlszeile 296
 Benutzerdefinierte Funktion 315, 332
 Berechnung der Steuererklärung 745
 Berechnungsoptionen 109
 automatische 109
 manuelle 109
 BEREICH.VERSCHIEBEN() 142, 281,
 301, 304
 BEREICHE() 284
 Bereichsbezug
 Ganze Spalten oder Zeilen 87
 Regeln 87
 Schreibweisen 87
 Bereichsoperator 86, 126
 Berners-Lee, Tim 867
 Bernoulli-Experiment 363
 Beschäftigung
 Monate berechnen 206
 Zeit berechnen 206

Bessel, Friedrich Wilhelm 846
 Besselfunktionen
 modifizierte 806, 847, 850
 BESSELI() 847
 BESSELJ() 829, 849
 BESSELK() 850
 Besselsche Differenzialgleichung
 846–847, 849
 BESSELY() 852
 BESTIMMTHEITSMASS() 352
 Bestimmtheitsmaß 353
 BETA.DIST() 358
 BETA.INV() 355, 359
 BETA.VERT() 358
 BETADIST() 358
 Betafunktion 356
 BETAINV() 355
 BETAVERT() 358
 Beta-Verteilungskurve 358
 Bezeichner 176
 Beziehungen erstellen 55
 Bezug 279, 289
 absoluter 279, 600
 absoluter und relativer 150
 Anzahl der Bereiche 284
 dynamischer 281
 gemischter 159
 relativer 279
 Bezugsart 279, 318
 Bezugsoperatoren 86
 BIN2DEC() 809–810, 812
 BIN2OCT() 811
 Binärdarstellung 808, 824, 826–827
 Binary Digit 808
 BININDEZ() 809–811
 BININHEX() 810
 BININOKT() 811
 BINOM.DIST.RANGE() 365
 BINOM.DIST() 362
 BINOM.INV() 360
 BINOM.VERT.BEREICH() 365
 BINOM.VERT() 361–362, 365
 BINOMDIST() 362
 Binomialkoeffizienten 764
 Binomialverteilung 361, 363, 476, 764
 kleinster Wert 360
 negative 460
 BINOMVERT() 362
 Bit 808
 BITAND() 826
 BITLSHIFT() 824
 BITLVERSCHIEB() 824

Bitmuster 825, 827
 BITODER() 825
 Bitoperation 824–825
 BITOR() 825
 BITRSHIFT() 826
 BITRVERSCHIEB() 824, 826
 BITUND() 826
 BITXODER() 827
 BITXOR() 827
 Blatt schützen 108
 BLATT() 313
 BLÄTTER() 313
 Blattname berechnen 140
 Blattschutz 108
 Blitzvorschau 37
 BOGENMASS() 733
 Bogenmaß 719, 747, 776, 788
 Breite der Zelle 333
 Briggs, Henry 753
 BRTEILJAHRE() 204
 Bundesanleihe 629, 672
 Bundesobligation 664
 Bundesschatzbrief 691
 BW() 631

C

Cascading Style Sheets 869
 CELL() 333
 CGS-System 823
 CHAR() 253, 260
 CHIDIST() 374, 376
 CHIINV() 367
 Child 873
 CHIQU.INV.RE() 367
 CHIQU.TEST() 368, 370
 CHIQU.VERT.RE() 374
 CHIQU.VERT() 376
 Chi-Quadrat 368
 Chi-Quadrat-Test 370–371
 CHISQ.DIST() 376
 CHISQ.INV.RT() 367
 CHISQ.INV() 369
 CHISQ.TEST() 370
 CHISQ.VERT.RT() 374
 CHITEST() 370
 CHIVERT() 374
 CHOOSE() 306
 CLEAN() 247
 CODE() 232, 260
 Codepage 231
 Codepoint 253

- Codezahl 260
 COLUMN() 297
 COLUMNS() 298
 COM+-Komponenten 296
 COMPLEX() 845
 CONCATENATE() 254
 CONFIDENCE.NORM() 421
 CONFIDENCE.T() 425
 CONVERT() 820
 CORREL() 426
 COS() 734
 COSEC() 736
 COSECHYP() 737
 Cosh 721
 COSHYP() 738
 COT() 739
 Coth 721, 724, 740
 COTHYP() 740
 COUNT() 349
 COUNTA() 351
 COUNTBLANK() 352
 COUNTIF() 562
 COUNTIFS() 564
 COUPDAYBS() 697
 COUPDAYS() 695
 COUPDAYSNC() 696
 COUPNCD() 694
 COUPNUM() 699
 COUPPCD() 698
 COVAR() 428
 COVARIANCE.P() 431
 COVARIANCE.S() 432
 Cramersche Regel 757
 CRITBINOM() 360
 Csc 736
 Csch 737
 CSS 869
 Cube 602
 CUBELEMENT() 604
 CUBELEMENTEIGENSCHAFT() 607
 CUBEKPIELEMENT() 608
 CUBEKPIMEMBER() 608
 CUBEMEMBER() 604
 CUBEMEMBERPROPERTY() 607
 CUBEMENGE() 610
 CUBEMENGENANZAHL() 612
 CUBERANGELEMENT() 612
 CUBERANKEDMEMBER() 612
 CUBESET() 610
 CUBESETCOUNT() 612
 CUBEVALUE() 614
 CUBEWERT() 614
 CUMIPMT() 647
 CUMPRINC() 646
- ## D
- DATE() 207
 DATEDIF() 205
 Dateienzahl 316
 Dateiname 333
 berechnen 139
 Datenanalyse 594
 Datenbank 792
 Aufbau 568
 Datenbankfunktionen
 Arbeiten mit Datenbanken
 und -sätzen 570
 Argumente 569, 577
 Überblick 572
 Verwendung von
 Steuerelementen 572
 Datenprüfung 158, 163, 165
 DATENREIHE() 283
 Datenschnitte 43
 Datentyp 176, 330, 869
 Double 176, 181
 Variant 176
 Datenüberprüfung 322
 Datenverbindungsdatei 602
 DATEVALUE() 208
 DATUM() 207, 210, 217
 Datumsangaben 329
 Datumsdifferenz 204–205, 209,
 217–219, 222–223
 Datumseingaben 198
 Datumssystem 198, 207
 Datumstext 208
 Datumswerte
 anzeigen 198
 eingeben 198
 speichern 198
 DATWERT() 208
 DAVERAGE() 581
 DAY() 222
 DAYS() 222
 DAYS360() 223
 DB() 641
 DBANZAHL() 573
 DBANZAHL2() 575
 DBAUSZUG() 577
 DBMAX() 578
 DBMIN() 580
 DBMITTELWERT() 581
 DBPRODUKT() 582
 DBSTDABW() 584
 DBSTDABWN() 586
 DBSUMME() 283, 587
 DBVARIANZ() 590
 DBVARIANZEN() 591
 DCOUNT() 573
 DCOUNTA() 575
 DDB() 640
 DEC2HEX() 813
 DEC2OCT() 814
 Dekadischer Logarithmus 754
 Deklaration 180
 DELTA() 856
 Descendant 873
 Descending-or-Self 873
 Detektiv 112
 Determinante 755, 757
 DEVSQ() 524
 Dezil 479
 DEZIMAL() 742
 Dezimalsystem 732, 742, 806–807
 Dezimalzahl 814, 819
 DEZINBIN() 732, 812–813, 815
 DEZINHEX() 732, 813
 DEZINOKT() 732, 814
 DGET() 577
 DIA() 634
 Diagrammansicht 57
 Differential- und
 Integralrechnung 743
 Differenzen 785
 der Quadrate 785
 Dimension 603
 DIN-Norm 1355 211
 Diracsche Delta-Funktion 856
 Direkte Zellbearbeitung 99
 Disagio 669, 674, 692
 DISAGIO() 635
 DISC() 635
 Disjunktion 825
 Diskont 635, 669, 692
 Diversifikation 292
 Division 83, 747, 768
 DLL-Registrierung 296
 DM() 234
 DMAX() 578
 DMIN() 580
 DNS 871
 Dokumentknoten 871
 DOLLAR() 234

DOLLARDE() 661
 DOLLARFR() 660
 DOLLARTEXT() 232
 Dollarzeichen 91
 Domain Name Systems (DNS) 871
 Doppelklick 106
 Doppelpunkt 86
 Doppelte Einträge verhindern 165
 Double-Byte
 Zeichensatz 231
 DPRODUCT() 582
 DSTDEV() 584
 DSTDEVP() 586
 DSUM() 587
 Dualsystem 807–808
 Dualzahl 812, 815, 818
 Duration 636, 657
 DURATION() 636
 Durchschnittliche Abweichung 445
 DVAR() 590
 DVARP() 591
 Dynamischer Bezug 281

E

Echtzeitdaten 295
 EDATE() 209
 EDATUM() 209
 EFFECT() 638
 EFFEKTIV() 638
 Effektiver Jahreszinssatz 639
 Effektivzins 643
 Effektivzinssatz 638, 660
 Eigenschaftenfenster 173
 Einerkomplement 808
 Einfache Varianzanalyse 380
 Einfache Zinsrechnung 619
 Einfachregression 347
 Eingabebereich 598
 Eingabekontrollen 163
 Einheitenpräfix 823
 Einheitensystem 823
 Einheitsmatrix 756
 Einzelne Werte aus Datenbank 578
 Elementeigenschaften 607
 Elementfolge 750–751
 Else 182–183
 Empfohlene Diagramme 38
 Empirische Varianz 590
 EN 28601 211
 ENCODEURL() 862
 End If 182

Endwert 702, 704
 Engineering Functions 804
 Entwicklertools 172
 EOMONTH() 217
 ERF.PRECISE() 853
 ERF() 853
 ERFC.PRECISE() 855
 ERFC() 855
 Ergebnis in Grad umgerechnen 747
 ERROR.TYPE() 314
 Error-Funktion 853
 ERSETZEN() 235, 256
 Erwartungswert 590
 Euklidischer Algorithmus 747
 Euler, Leonhard 743, 837
 Eulersche Formel 837
 Eulersche Identität 763
 Eulersche Zahl 743, 753
 Europäische Union 211
 EXACT() 241
 Excel-VBA-Projekte 173
 Excel-Version ermitteln 316
 Exklusiv-Oder 827
 EXP() 743, 753
 EXPON.DIST() 376
 EXPON.VERT() 376
 EXPONDIST() 376
 Exponent 764
 Exponentialfunktion 377, 753
 Exponentialverteilung 559
 Exponentielle Regression 494, 498
 Exponentieller Trend 549
 EXPONVERT() 376
 Extensible Markup Language (XML) 868
 Extensible Stylesheet Language 869
 Externe Bezüge 112

F

F.DIST.RT() 389
 F.DIST() 392
 F.INV.RE() 379, 382
 F.INV() 382
 F.TEST() 387
 F.VERT.RE() 389
 F.VERT() 392
 FAKULTÄT() 744
 Fälligkeitstermin 217
 FALSCH() 262
 FALSE() 262
 Farbe der Zelle 333
 FDIST() 389
 Fehler behandeln 269
 FEHLER.TYP() 314, 319
 Fehlerfunktion 853
 verallgemeinerte 854
 Fehlermeldung 112
 Fehlerspur 112
 Fehlersuche 132
 Fehlerüberprüfung 113
 Fehlerwert 314, 318–319, 325, 331
 #NAME? 193
 #NV 325, 329
 #WERT! 186
 Feiertage 201, 203, 218–219
 Feldinhalte 568
 Feldnamen 568
 FEST() 236
 Filterbedingungen 569
 Filterung
 nach Jahren 212
 nach Monaten 216
 nach Tagen 222
 FILTERXML() 865
 Finanzinvestition 643, 659
 Finanzmathematik
 (Grundaufgaben) 632
 FIND() 237
 FINDEN() 139, 157, 237
 FINV() 379
 FISHER() 383
 FISHERINV() 387
 Fisher-Transformation 383
 umgekehrt 387
 FIXED() 236
 FlächeKrSeg() 172, 186–187
 Following 873
 Following-sibling 873
 FORECAST() 500
 Format der Zelle 333
 Formatanweisungen für Datum
 und Zeit 199
 Formatierung, bedingte 263
 Formel
 Analyse 110
 Ansicht 97, 113
 Auswertung 113–114
 bearbeiten 130
 Bearbeitungsmodus 110
 Berechnungsoptionen 109
 Bezugszellen hervorheben 110
 durch Ergebnis ersetzen 107
 Eingabe 89
 Eingabe in mehrere Blätter 98

- Formel (*Fortsetzung*)
- Eingabe in mehrere Zellen 98
 - Fehlersuche 132
 - kopieren 103
 - mit Ausfüllen-Funktion kopieren 105
 - per Maus kopieren 103
 - per Maus verschieben 102
 - per Zwischenablage kopieren 103
 - per Zwischenablage verschieben 101
 - schützen 108
 - Teilberechnungen 132
 - überwachen 110
 - verschieben 101
- Formeln
- verstecken 108
- Formeln und Namen 285
- Formelüberwachung 110
- Nachfolger 111
 - Spurpfeil 111–112
 - Symbole 112
 - Überwachungsfenster 114
 - Vorgänger 111
- Formelzellen markieren 99
- Formular
- Steuerelement 290
 - Symbolleiste 290, 598
- FORMULATEXT() 285
- Fragezeichen 248
- Freie Tage 148, 201, 203, 218–219
- Freigabe 49
- Freiheitsgrade 786
- FREQUENCY() 411
- FTEST() 387
- Funktion 171
- Argumente 117–118, 120
 - Argumentübergabe 118
 - Assistent 122
 - benutzerdefinierte 171
 - Definition 117
 - einfügen 123
 - im Funktions-Assistent eintragen 193
 - in Namen 138
 - manuelle Eingabe 121
 - Offlinehilfe 126
 - Optionale Argumente 117
 - Syntax 117
 - Teilberechnungen 132
 - Trigonometrie 720
 - verschachtelt 118, 128
 - zyklometrisch 720
- Funktions-Assistent 122, 176, 193
- Funktionsdeklaration 175
- Funktionsname 171, 175
- Funktionspalette 128
- Funktions-QuickInfo 121
- Furigana 246
- Fußzeile 138
- FV() 702
- FVERT() 389
- F-verteilten Zufallsvariable 390
- F-Verteilung 380
- FVSCHEDULE() 704
- ## G
- G.TEST() 406
- GAMMA.DIST() 398
- GAMMA.VERT() 398
- GAMMA() 393
- GAMMADIST() 398
- Gamma-Funktion 849
- GAMMAINV() 394
- GAMMALN.GENAU() 397
- GAMMALN() 396
- Gammaverteilung 393–394, 399
- Logarithmus 396
- GAMMVERT() 398
- Ganze Zahl 745
- GANZZAHL() 745, 769
- GAUSS() 401
- GAUSSF.GENAU() 853
- GAUSSFEHLER() 853, 855–856
- GAUSSFKOMPL.GENAU() 855
- GAUSSFKOMPL() 854–855
- Gauß, Carl Friedrich
- Gaußsche Ebene 829
 - Gaußsche Fehlerfunktion 806, 853, 855
 - Gaußsches Eliminationsverfahren 757
 - Gaußstest 407
- GDA() 640
- GDA2() 641
- Gefilterte Ergebnisse 792
- Gegenhypothese 371
- Gegenkathete 725, 727, 739, 773, 776, 788
- Gehe zu 143
- Aktuelles Aray 97
- Gemischte Bezüge 92, 159
- Generalized Markup Language (GML) 867
- GEOMEAN() 403
- Geometrisch-degressive Abschreibung 641
- Geometrisches Mittel 403
- GEOMITTEL() 403
- Gerade Zahl 321, 327
- GERADE() 745
- Gesperrt 108
- Gesperrte Zellen 336
- GESTEP() 858
- Gestutztes Mittel 405
- GESTUTZTMITTEL() 405
- GETPIVOTDATA() 593
- Gewicht 821
- GGANZZAHL() 857–858
- GGT() 746
- GLÄTTEN() 238
- Gleichheitszeichen 269
- GML (Generalized Markup Language) 867
- Goniometrische Form 829, 832
- GRAD() 719, 747
- GROSS() 239
- GROSS2() 240
- Groß-/Kleinschreibweise 237, 241
- Großbuchstaben 239
- Größer als 269
- Größter Wert 418, 439–440
- aus Datenbank 578
- GROWTH() 549
- Grundaufgaben, finanzmathematische 632
- Grundgesamtheit 584, 586, 591
- Gruppen 750–751
- Gruppierung aufheben 99
- Gruppierungsmodus 98
- GTEST() 406
- Gültiger Zellbezug 317
- Gültigkeitsprüfung 158, 163, 165, 293
- Gültigkeitsregel 293
- ## H
- HARMEAN() 408
- HARMITTEL() 408
- Harmonisches Mittel 409
- HÄUFIGKEIT() 411
- Hauptdiagonale 756
- HEUTE() 210, 213
- HEX2BIN() 815
- HEX2DEC() 816
- HEX2OCT() 817

- Hexadezimalsystem 807–808
 Hexadezimalzahl 810, 813, 816–817, 820
 Hexagesimalsystem 807
 HEXINBIN() 815–817
 HEXINDEZ() 816
 HEXINOKT() 817
 Hiragana 245
 HLOOKUP() 307
 HOUR() 221
 HTML (Hypertext Markup Language) 867
 Hyperbelfunktion 721, 724, 738, 741, 775, 778, 790
 Hyperbelkosekans 737
 Hyperbelkosinus 739
 Hyperbelkotangens 740
 Hyperbelsekans 775
 Hyperbelsinus 778
 Hyperbeltangens 790
 Hyperbolicus 778
 Hyperbolische Funktion 726
 Hyperbolische Kosinus 739
 Hyperbolischen Kosinus 738
 Hyperbolischen Kotangens 724
 Hyperbolischen Sinus 778
 Hyperbolischen Tangens 790
 Hyperbolischer Kosekans 737, 835
 Hyperbolischer Kosinus 721, 834
 Hyperbolischer Kotangens 740
 Hyperbolischer Sekans 775, 841
 Hyperbolischer Tangens 730, 741
 Hypergeometrische Verteilung 416
 HYPERLINK() 286
 Hypertext Markup Language (HTML) 867
 HYPGEOM.DIST() 415
 HYPGEOM.VERT() 415
 HYPGEOMDIST() 415
 HYPGEOMVERT() 415
 Hypotenuse 719, 725, 734, 773, 776
- I**
 IDENTISCH() 241
 Identische Werte hervorheben 149
 If
 Abfragen 172
 Anweisung 181–182
 Block 183–184
 IF() 269
 IFERROR() 272
 IFNA() 273
 IKV() 270, 642
 IMABS() 830
 Imaginäre Einheit 829
 Imaginärteil 829–831
 IMAGINÄRTEIL() 831
 IMAGINARY() 831
 IMAPOTENZ() 832
 IMARGUMENT() 832
 IMCONJUGATE() 837
 IMCOS() 833
 IMCOSEC() 834
 IMCOSECHYP() 835
 IMCOSH() 834
 IMCOSHYP() 741, 834
 IMCOT() 835
 IMCSC() 834
 IMCSCH() 835
 IMDIV() 836
 IMEXP() 836
 IMKONJUGIERTE() 837
 IMLN() 836, 838
 IMLOG10() 838
 IMLOG2() 839
 IMPOWER() 832
 IMPRODUCT() 839
 IMPRODUKT() 839
 IMREAL() 840
 IMREALTEIL() 840
 IMSEC() 840
 IMSECH() 841
 IMSECHYP() 841
 IMSIN() 841
 IMSINH() 842
 IMSINHYP() 741, 778, 842
 IMSQRT() 844
 IMSUB() 842
 IMSUM() 843
 IMSUMME() 843
 IMTAN() 844
 IMWURZEL() 844
 In Formeln konvertieren 603
 Index 306
 INDEX() 287, 289, 301, 598
 INDIRECT() 292
 INDIREKT() 140, 153, 156, 292
 INFO() 316
 Informationen
 über Zellen 333
 zu Excel 316
 zum Betriebssystem 316
 Informationstyp 205
 Inhalt der Zelle 333
 Inhalte einfügen 107–108, 257
 INTERCEPT() 347
 Internationales Einheitensystem 823
 Interne Kapitalverzinsung 642, 663
 interne Kapitalverzinsung 270
 Interner Zinssatz 642, 686
 Intervall 761, 795
 Grenze 761, 795
 Inverse 757
 inverse Hyperbelfunktion 726
 inverse Hyperbelfunktionen 730
 Investitionsrechnung 620, 633, 642, 686, 692
 IPMT() 700
 IRR() 642
 ISBLANK() 322
 ISERR() 318
 ISERROR() 319
 ISEVEN() 321
 ISFORMULA() 320
 ISLOGICAL() 324
 ISMA-Methode 650, 666
 IsMissing() 181, 183
 ISNA() 325
 ISNONTEXT() 321
 ISNUMBER() 327
 ISO 10646 253
 ISO 8601 211
 ISO.OBERGRENZE() 748
 ISODD() 327
 ISOKALENDERWOCHE() 211
 ISOWEEKNUM() 211
 ISPMT() 644
 ISREF() 317
 ISTBEZUG() 317
 ISTEXT() 325
 ISTFEHL() 318
 ISTFEHLER() 318–319
 ISTFORMEL() 320
 ISTGERADE() 321
 ISTKTEXT() 321
 ISTLEER() 322, 330
 ISTLOG() 324
 ISTNV() 325, 330
 ISTTEXT() 325
 ISTUNGERADE() 327
 ISTZAHL() 147–148, 327

J

JAHR() 209, 212, 217
 Jahreszahl
 aus einem Datum 208
 ermitteln 212
 JETZT() 210, 213

K

KALENDERWOCHE() 214
 Kalkulationszinssatz 642, 658
 Kanji 245
 Kapitalbindungsdauer 636
 Kapitalverzinsung 270
 Kapitalwert 642, 658, 688
 Kapitalwertmethode 620, 633
 KAPZ() 645
 Kehrwertfunktion 739, 773, 775
 Kennzahl 602
 Key Performance-Indikator (KPI) 608
 KGRÖSSTE() 156, 418
 KGV() 749
 KLEINSTE() 156, 420
 Klammern 118, 175
 Eingabehilfen 84
 Rechenreihenfolge 83
 KLEIN() 242
 Kleinbuchstaben 242
 Kleiner als 269
 Kleinster Wert 420, 443–444
 aus Datenbank 580
 Kleinstes gemeinsames Vielfaches 749
 Kleinstes Vielfaches 760, 794
 Kombinationen
 mit Wiederholungen 751
 ohne Wiederholung 750
 KOMBINATIONEN() 750
 KOMBINATIONEN2() 751
 Kombinationsfeld 290, 598
 Kombinatorik 553, 555
 Kompatibilität 44, 795
 2003 in 2013 47
 2013 in 2003 45
 KOMPLEXE() 845
 KONFIDENZ.NORM() 421
 KONFIDENZ.T() 425
 KONFIDENZ() 421
 Konfidenzintervall 421–422
 Konjugierte komplexe Zahl 805, 837
 Konjunktion 826
 Konstanten 89

Konstruktion 804
 Konsumentenkredite 687
 Koordinaten 279
 Kopfzeile 138
 Kopieren
 als Wert 107
 Dividieren 108
 Formeln 103
 per Ausfüllen-Funktion 105
 per Maus 103
 per Zwischenablage 103
 KORREL() 426
 Korrelationsanalyse 426
 Korrelationskoeffizient 426, 473
 Kosekans 736, 834
 Kosekans Hyperbolicus 737
 Kosinus 719, 773, 833–834
 Kosinus Hyperbolicus 775
 Kosinus hyperbolicus 738, 741
 Kotangens 739, 835
 Kotangens Hyperbolicus 740
 KOVAR() 428
 Kovarianz 428
 KOVARIANZ.P() 431
 KOVARIANZ.S() 432
 KPI (Key Performance-Indikator) 608
 Kreis 733
 Kreisbewegung 763
 Kreisbogen 733
 Kreisfläche() 171–172, 177–179, 190, 193
 Kreisfläche1() 172, 190
 Kreisfunktion 738, 741
 Kreisumfang 733
 Kreiszahl 763
 Kreuztabelle 303
 KRITBINOM() 360
 Kriterienbereich 283, 569
 Kritischer Wert 368
 KUMKAPITAL() 646
 Kumulierte Zinsen 647
 KUMZINSZ() 647
 Kuponzinssatz 627
 Kurs 620, 628, 649, 654, 666, 677, 681, 683
 KURS() 649
 KURSDISAGIO() 653
 KURSFÄLLIG() 654
 Kursrechnung 620, 633
 KURT() 433
 Kurtosis 433, 504
 Kürzen von monetären Beträgen 752
 KÜRZEN() 752

L

Ländereinstellungen 234
 LÄNGE() 163, 243
 LARGE() 418
 Laufzeit 217
 einer Finanzierung 205
 Leere Zeichenfolge 325
 Leere Zelle 322, 575
 Leerer Text 575
 Leerschritt 86
 Leerstring 183
 Leerzeichen 570
 hervorheben 156
 löschen 238
 LEFT() 244
 Leibniz, Gottfried 756, 808
 LEN() 243
 Len() 190
 Letzter Tag im Monat 217
 LIA() 656
 Lineare Abschreibung 625, 656
 Lineare Regression 494, 522
 Linearer Trend 500, 528
 LINEST() 492
 Link 286
 LINKS() 139, 156, 244
 Listen 792
 variabler Länge 282
 LN() 752
 LOG() 753
 LOG10() 753–754
 Logarithmensystem 753
 Logarithmus 753, 764, 838
 dekadischer 806, 838
 natürlicher 838
 LOGEST() 497
 Logikfunktionen 262
 LOGINV() 436
 Logische Funktionen 262
 Logische Verknüpfungen 263
 LOGNORM.DIST() 437
 LOGNORM.INV() 436
 LOGNORM.VERT() 437
 Lognormalverteilung 437
 Umkehrfunktion 436
 LOGNORMDIST() 437
 LOGNORMVERT() 437
 Logogramme 245
 Lokalisierungspfade 872

Look 30
 LOOKUP() 304
 Loop siehe Schleife
 LOWER() 242

M

Macaulay-Duration 636
 Macdonaldsche Funktion 850
 Makro 170
 Befehl 170
 eingebaute Tabellenfunktionen nutzen 191
 Sicherheit 831
 Markieren
 alle Formelzellen 99
 per Maus 101
 per Tasten 100
 Maßsystem 805, 820
 Gaußsches 823
 Match 872
 MATCH() 301
 Matrix 287, 293, 299, 307, 331, 755–757, 780
 durchsuchen 304
 Formel 266, 288
 Formel markieren 97
 Formelbearbeitung 96
 Formeleingabe 96, 241, 258, 270, 281, 293, 297, 302, 306, 309, 331
 Formeln 96
 Matrixfunktion 756
 Matrixkonstante 202, 218–219, 287–288, 298, 569
 Beispiel 305
 Matrixversion 304
 Matrizen 786
 Mausmodus 42
 MAX() 156, 439
 MAXA() 440
 MDET() 755, 757
 MDURATION() 657
 MDX 605
 Measure 602
 Median 479, 485
 MEDIAN() 441
 Mehrwertsteuer berechnen 326
 MEINHEIT() 756
 Menge 610

Menüband 30
 anheften 30
 Look 30
 Methode des internen Zinssatzes 620, 642
 MID() 190, 250
 MIN() 156, 443
 MINA() 444
 Minderungsprozesse 403
 MINUTE() 215
 Minutenzahl 215
 MINV() 756–757, 760
 MIRR() 663
 MITTELABW() 445
 MITTELWERT() 447
 MITTELWERTA() 449
 MITTELWERTWENN() 451
 MITTELWERTWENNS() 454
 Mittlere Abweichung 446
 MMULT() 759
 MODALWERT() 457
 MODE.MULT() 459
 MODE.SINGLE() 457
 MODE() 457
 Modified Duration 657, 662
 Modul 173
 Modulo-Berechnungsmethode 769
 MODUS.EINF() 457, 460
 MODUS.VIELF() 459
 Moivre, Abraham de 832
 MONAT() 209, 216–217
 MONATSENDE() 217
 Monaterster 208
 Monatsletzter 208, 217
 Monatszahl 216
 aus einem Datum 208
 MONTH() 216
 Most Significant Bit 808
 MS Query 603
 MTRANS() 293
 Multidimensionale Ausdrücke (MDX) 605
 Multiplikation 83, 767
 Multiplizieren 764
 Multiplizierte Werte in Datenbank 582
 Mustererkennung 871

N

N() 328
 NA() 329
 Nachfolger 111

Namen 147
 aus Auswahl erstellen 155
 definieren 285
 dynamische 142
 mit Funktionen 138
 Namensräume 870
 Namespace 865, 873
 Natürlicher Logarithmus 743, 752
 NBW() 658
 Negative Korrelation 473
 Negative Zahlen 797
 NEGBINOM.DIST() 460
 NEGBINOM.VERT() 460
 NEGBINOMDIST() 460
 NEGBINOMVERT() 460
 NETTOARBEITSTAGE.INTL() 219
 NETTOARBEITSTAGE() 218
 Nettobarwert 658
 NETWORKDAYS.INTL() 219
 NETWORKDAYS() 218
 Neuberechnung erzwingen 316
 Neuheiten 32
 Arbeitsmappen 33
 Blitzvorschau 37
 Datenschnitte 43
 Empfohlene Diagramme 38
 PivotTable 51
 Power View 69
 Schnellanalyse 34
 Sparklines 35
 Veröffentlichen 47
 Vorlagen 32
 Neumannsche Funktion 852
 Nicht druckbare Zeichen 247
 NICHT() 264
 Nicht-polyadisch 742
 NOMINAL() 660
 Nominalzinssatz 628, 638, 660, 672
 NORM.DIST() 468
 NORM.INV() 462
 NORM.S.DIST() 466
 NORM.S.INV() 464
 NORM.S.VERT() 466, 472
 NORM.VERT() 433, 468
 Normalverteilung 469, 517, 534, 854
 standardisiert 462
 NORMDIST() 468
 NORMINV() 462
 NORMSDIST() 466
 NORMSINV() 464
 NORMVERT() 468
 NOT() 264

NOTIERUNGBRU() 660
 NOTIERUNGDEZ() 661
 NOW() 213
 NPER() 705
 NPV() 658
 Nullhypothese 371
 NUMBERVALUE() 259
 Numerische Werte 327
 in Text umwandeln 236, 250, 252
 in Währungstext umwandeln 234
 in Zahlwort umwandeln 232
 NV() 329

O

Oberfläche einer Kugel 763
 OBERGRENZE.GENAU() 748, 760
 OBERGRENZE.MATHEMATIK()
 748, 760–761
 OBERGRENZE() 760
 Objektorientierte
 Programmierung 191
 OCT2BIN() 818
 OCT2DEC() 819
 OCT2HEX() 819
 ODDFPRICE() 677
 ODDFYIELD() 680
 ODDLPRICE() 681
 ODDLYIELD() 683
 ODER() 146, 158, 265, 825
 ODER-Bedingung 570, 588
 Office-Website 287
 Offlinecube 602
 Offlinehilfe 126
 Öffnungswinkel 734, 773, 776
 OFFSET() 281
 Oktalsystem 808
 Oktalzahl 811, 814, 817–819
 OKTINBIN() 818–820
 OKTINDEZ() 819
 OKTINHEX() 819
 OLAP-Cube 602
 OLAP-PivotTable 593
 Operanden 82
 Operatoren 82, 571
 & 254
 arithmetische 83
 Bereichsoperator 86
 Bezugsoperatoren 86
 Doppelpunkt 86
 Leerschritt 86, 88
 Prioritäten 84

Rangfolge 84
 Schnittmengenoperator 88
 Semikolon 86, 88
 Textoperator 85
 Verbindungsoperator 88
 Vergleichsoperator 84
 Optional 180
 Optionale Argumente 180

P

Parameter 117
 Parent 873
 PC-Systemzeit 198
 PDURATION() 662
 PEARSON() 472
 Pearsonscher
 Korrelationskoeffizient 352, 472
 PERCENTILE.EXC() 481
 PERCENTILE.INC() 481
 PERCENTILE() 479
 PERCENTRANK.EXC() 483
 PERCENTRANK.INC() 484
 PERCENTRANK() 482
 Periodenzinssatz 644
 PERMUT() 553
 Permutation 750–751
 PERMUTATIONA() 555
 Pfad der aktuellen Mappe 139
 Pfeile entfernen 111–112
 PHI() 472
 PHONETIC() 245
 Pi 800
 Pi() 191–192
 PIVOTDATENZUORDNEN() 593, 606
 PivotTable 51, 593
 Beziehungen erstellen 55
 Empfehlungen 67
 Neue Funktionen 52
 PowerPivot Add-In aktivieren 53
 Tabellen verknüpfen 55
 Timeline 65
 und OLAP-Cubes 603
 Platzhalter 255, 294, 571
 Platzhalterzeichen 237, 248
 PMT() 672
 POISSON.DIST() 476
 POISSON.VERT() 476
 POISSON() 476
 Poisson-verteilte Zufallsvariable 476
 Polarkoordinaten 829
 Polyadisch 742

POLYNOMIAL() 763
 Polynomkoeffizient 763
 Polynomverteilung 764
 Positive Korrelation 473
 Positive Zahlen 797
 POTENZ() 764
 Potenzierung 83, 743
 POTENZREIHE() 765
 Power View 69
 Definition 69
 in Excel 2013 72
 Neuigkeiten 70
 Robert Bruckner 77
 Spannendes Beispiel 77
 PowerPivot-Add-In aktivieren 53
 PowerPivotTable
 Diagrammansicht 57
 PPMT() 645
 Präfix 334
 Preceding 873
 Preceding-sibling 873
 Preisangabenverordnung 639, 687
 PRICE() 649
 PRICEDISC() 653
 PRICEMAT() 654
 Primfaktorenzerlegung 746, 749
 PROB() 557
 Produkt zweier Matrizen 759
 PRODUKT() 767
 Projekteinsatz 218, 220
 Projekt-Explorer 173
 PROPER() 240
 Prozedur-Assistent 175
 Prozedurkopf 176
 Prozent 83
 Prozentualer Rang 482
 Public 175
 PV() 631

Q

QIKV() 663
 QUADRATESUMME() 767
 Quadratische Diagonalmatrix 756
 Quadratische Matrix 755–756
 Quadratwurzel 799
 einer komplexen Zahl 844
 Quadrierte Abweichungen 524
 Quadrierte Differenzen 787
 Qualifizierte interne
 Kapitalverzinsung 663
 Quantil 485

QUANTIL.EXKL() 481
 QUANTIL.INKL() 481
 QUANTIL() 479
 QUANTILSRANG() 482
 Quartil 479
 QUARTILE.EXKL() 488
 QUARTILE.INKL() 488
 QUARTILE() 485
 QUARTILSRANG.EXKL() 483
 QUARTILSRANG.INKL() 484
 Quersumme() 172, 187, 189
 QuickInfo 102, 112, 121
 QUOTIENT() 768

R

Rabattstaffel 272
 Radiant 733
 RANG.GLEICH() 490
 RANG.MITTELW() 491
 RANG() 489
 Rangfolge 83
 RANK.AVG() 491
 RANK.EQ() 490
 RANK() 489
 RATE() 690
 Ratenkreditgeschäfte 687
 Ratenzahlung 619
 Realteil 829–831
 RECEIVED() 630, 692
 Rechenreihenfolge 83
 RECHTS() 157, 246
 Rechtwinkeliges Dreieck 734
 Regelmäßige Zahlungen 705
 Regeln verwalten 150
 Registrierung einer DLL-Datei 296
 Registry 296
 Regression 524
 Regressionsanalyse 347, 493, 497, 519
 Regressionsgerade 519
 Regressionskenngrößen 493
 Reihenfolge beim Zieleinlauf 744
 Relative Bezüge 90, 104, 150
 Rendite 620, 643, 665, 669, 671, 680, 694
 RENDITE() 665
 RENDITEDIS() 669
 RENDITEFÄLL() 671
 Rente 672
 Berechnung 619, 633, 673, 691,
 703, 706
 nachsüssige 619
 vorschüssige 619

REPLACE() 235
 Representational State Transfer 31
 REPT() 257
 Rest 746
 einer Division 769
 REST() 31, 154, 769
 Restschuld 646
 RGP() 492
 RIGHT() 246
 RKP() 497
 RMZ() 157, 186, 672
 RÖMISCH() 711, 771
 ROW() 309
 ROWS() 310
 RRI() 701
 RSQ() 352
 RSS-Feeds 865
 RTD() 295
 Rückgabewert 176
 Rückstellung von Weihnachtsgeld 206
 RUNDEN() 714, 731, 745, 772

S

Sachinvestition 643, 659
 Satz (Set, Menge) 610
 SÄUBERN() 247
 Schaltjahre 200
 Schatzanweisungen 674
 SCHÄTZER() 500
 SCHIEFE.P() 506
 SCHIEFE() 503
 Schleife 172, 188
 Schleifenfuß 188–189
 Schleifenkopf 188–189
 Schleifenrumpf 189
 Schlüsselwort 176
 Schnellanalyse 34
 Ergebnisse 36
 Sparklines 35
 Schnittmenge 88
 Operator 88
 Schrittlänge 802
 Schutz 108, 334
 SEARCH() 248
 Sec 773
 SEC() 773
 Sech 775
 SECHYIP() 775
 SECOND() 220
 Seemeilen 733
 Sekans 773, 840
 Sekans Hyperbolicus 775
 SEKUNDE() 220
 Sekundenzahl 220
 Self 873
 Seltene Ereignisse 476
 Semikolon 86
 Sexagesimalsystem 807
 SGML 867
 SHEET() 313
 SHEETS() 313
 Signifikanzniveau 538
 SIN() 776
 Single-Byte
 Zeichensatz 231
 Sinh 721, 724, 778
 SINHYP() 778
 Sinus 841–842
 Sinus Hyperbolicus 737, 778
 Sinuswert 725
 SI-System 823
 Skalarprodukt 294
 SKEW.P() 506
 SKEW() 503
 Skonto 689
 SkyDrive 48
 SLN() 656
 SLOPE() 519
 SMALL() 420
 Sollzinssatz 638
 Solver 294
 Sortierreihenfolge 611
 Sortierung
 nach Jahren 212
 nach Monaten 216
 nach Tagen 222
 SPALTE() 153, 297
 SPALTEN() 298
 Spaltenindex 287, 299
 Spaltennummer 297, 334
 Sparklines 35
 Spezialfilter 303
 Sprungfunktionen 856
 Spur
 zum Fehler 111, 113
 zum Nachfolger 111–112
 zum Vorgänger 111–112
 Spurpfeil 111–112
 SQL Server 2005 602
 STABW.N() 513
 STABW.S() 508
 STABW() 508
 STABWA() 511
 STABWN() 513

- STABWNA() 514
 Standardabweichung 508, 511,
 513–514, 584, 586
 Standardfehler 521
 STANDARDISIERUNG() 516
 STANDARDIZE() 516
 Standardnormalverteilung 407, 464,
 466, 534
 Standardwerte 172
 Standardzahlenformat 176, 198
 STANDNORMINV() 464
 STANDNORMVERT() 466
 STDEV.P() 513
 STDEV.S() 508
 STDEV() 508
 STDEVA() 511
 STDEVPA() 513
 STDEVPA() 514
 STEIGUNG() 519
 Sternchen 248
 STEYX() 521
 STFEHLERYX() 521
 Stichprobe 590
 Stichprobenvarianz 590
 Stichtagsberechnung 206
 Str() 190
 Streubreite 590
 Streudiagramm 474
 Streumaße 584
 Streuung 292
 Stückzinsen 627–629, 649, 672
 STUNDE() 221
 Stundenzahl 221
 SUBSTITUTE() 255
 Subtraktion 83
 SUCHEN() 237, 248
 SUMME() 779
 SUMMENPRODUKT() 780
 Summenzeichen 779
 SUMMEWENN() 781
 SUMMEWENNS() 783
 SUMMEX2MY2() 785
 SUMMEX2PY2() 786
 SUMMEXMY2() 787
 SUMQUADABW() 524
 SVERWEIS() 291, 299
 SYD() 634
 Symbole
 Formelüberwachung 112
 Symbolleiste
 Formular 598
 Syntax 117
 Systemzeit des PC 198
- T**
 T.DIST.2T() 539
 T.DIST.RT() 541
 T.DIST() 541
 T.INV.2S() 526
 T.INV() 528
 T.TEST() 533
 T.VERT.2S() 539
 T.VERT.RE() 541
 T.VERT() 541
 T() 250
 Tabellen verknüpfen 55
 Tabellenfunktion 191
 integrierte 190
 Tablet-Gesten 42
 TAG() 209, 222
 TAGE() 222
 TAGE360() 223
 Tagesdatum berechnen 211
 Tags 869
 Tagzahl 222
 in einem Datum 208
 TAN() 788
 Tangens 727, 844
 Tangens Hyperbolicus 790
 Tanh 721, 724, 741
 TANHYP() 790
 Tasten
 Ausfüllen 105
 Blatt berechnen 110
 Eingabe von Matrixformeln 96
 Formelansicht 97
 Formel-Bearbeitungsmodus
 99, 110
 Gehe zu 99
 In mehrere Zellen eingeben 98
 Kopieren 107
 Markieren 100
 Tausendertrennzeichen 236
 TBILLÄQUIV() 674
 TBILLE() 674
 TBILLKURS() 675
 TBILLPRICE() 675
 TBILLRENDITE() 676
 TBILLYIELD() 676
 TDIST() 539
 TEIL() 139, 250
 Teilberechnungen 132
 TEILERGEBNIS() 792
 Teilergebnisse 718
 formatieren 161
 Teilzeichenfolge 250
 Template 872
 Text
 suchen 301
 trimmen 239
 TEXT() 146, 225, 252
 Textbausteine 138
 Textformat 252
 Textoperator 85
 Textvariable 183, 190
 Textverkettingsoperator & 254
 Thai-Text 232
 Then 181–183
 Tilde 248
 Tilgungsanteil 700
 Tilgungsrechnung 619, 633, 673, 691,
 704, 707
 TIME() 227
 Timeline 65
 Zeitachsentools 66
 TIMEVALUE() 228
 TINV() 526
 TODAY() 210
 Top-Elemente hervorheben 155
 Transformation skaliertes
 Zahlenwerte 517
 Transponieren 294
 TRANSPOSE() 293
 Treasury Bill 674
 TREND() 528
 Trendanalysen 500
 Trendlinie 531
 Trigonometrische Funktion 739, 773
 TRIM() 238
 TRIMMEAN() 405
 Trimmen von Text 239
 TRUE() 268
 t-Test 535
 einseitig 536
 Typen 535
 zweiseitig 537
 TTEST() 533
 TVERT() 539
 t-Verteilung 526, 534, 539
 t-Wert 526
 TYP() 330
 TYPE() 330

U

Überwachungsfenster 113
 Uhrzeitwerte 199
 Umkehrfunktion 720, 723–724, 726, 731, 838
 Umrechnungskurs 108
 UMWANDELN() 804, 820
 Umwandlungsfunktion 732, 806
 UND() 155, 157–158, 267, 826
 UND-Bedingungen 570
 Ungerade Zahl 321, 327
 UNGERADE() 794
 Ungleich 269
 Unicode 231
 UNICODE() 253–254
 Uniform Resource Identification (URI) 871
 Uniform Resource Locator (URL) 862, 867, 871
 Universal Character Set 253
 UNIZEICHEN() 253
 UNREGER.KURS() 677
 UNREGER.REND() 680
 UNREGLE.KURS() 681
 UNREGLE.REND() 683
 Unsigned integer 808
 UNTERGRENZE.GENAU() 794
 UNTERGRENZE.MATHEMATIK() 794–795
 UNTERGRENZE() 794
 UPPER() 239
 URI siehe Uniform Resource Identification
 URL siehe Uniform Resource Locator
 URLCODIEREN() 862

V

Val() 190
 VALUE() 256
 VAR.P() 546
 VAR.S() 542
 VAR() 542
 VARA() 544
 Variable 172
 Variant 176, 181
 Varianz 509, 542, 544, 546–547, 584, 590–591
 inferentielle 542
 VARIANZ() 542
 VARIANZA() 544

VARIANZEN() 546
 VARIANZENA() 547
 Variation 750–751
 VARIATION() 549
 VARIATIONEN() 553, 751
 VARIATIONEN2() 555
 Variationskoeffizienten 509
 VARP() 546
 VARPA() 547
 VBA 170–171, 174, 188, 191–192, 296, 831
 Code 187
 Editor 194
 Entwicklungsumgebung 170, 175, 191
 Funktion 172, 187, 190
 Funktionen 172
 Modul 175
 Projekt 191
 Prozeduren 171
 VBAProject 194
 VDB() 684
 Vektor 294
 Vektorversion 304
 Verbindungsoperator 88
 Vereinigungsoperator 88, 126
 VERGLEICH() 147–148, 301, 309
 Vergleichsoperatoren 84, 262, 269
 Vergleichstyp 301
 VERKETTEN() 254
 Verknüpfen von Zeichenfolgen 254
 Verknüpfte Zelle 290
 Vermeidung doppelter Einträge 165
 Veröffentlichen 47
 Freigabe 49
 SkyDrive 48
 Verschieben
 Formeln 101
 per Maus 102
 per Zwischenablage 101
 Versicherungsprämie 687
 Version von Excel ermitteln 316
 VERWEIS() 304
 Verzinsung
 nachsüssige 619, 670
 unterjährliche 619
 vorschüssige 619, 630, 635, 653, 670, 693
 Zinseszinsseffekt 619
 Verzweigung 181
 Vielfaches 798
 Viereckfläche() 172, 179, 185

Visual Basic-Editor 170, 175, 180
 VLOOKUP() 299
 Vollkreis 719, 727, 747
 Volumen eines Zylinders 763
 Vorgänger 111
 Vorlagen 32
 VORZEICHEN() 797
 Vorzeichenlos 808
 VRUNDEN() 798

W

W3C siehe World Wide Web Consortium
 Wachstum, exponentielles 743
 Wachstumsprozesse 377, 403
 WAHL() 306
 WAHR() 268
 Wahrheitswert 84, 262–263, 267, 274, 324, 327, 331
 suchen 301
 WAHRSCHEBEREICH() 557
 Wahrscheinlichkeitsbereich 557
 Wahrscheinlichkeitsverteilung 356
 Währungssymbol 234
 Webdienst 31
 WEBDIENST() 863
 Webersche Funktion 852
 WEBSERVICE() 863
 Wechsel 631, 636, 654
 WECHSELN() 235, 255
 Wechselrechnung 631, 636, 654, 670
 WEEKDAY() 225
 WEEKNUM() 214
 WEIBULL.DIST() 559
 WEIBULL.VERT() 559
 WEIBULL() 559
 Weibull-Verteilung 559
 Weihnachtsgeld 206
 WENN() 269, 291, 312, 857–858
 WENNFEHLER() 272, 717
 WENNNV() 273
 WERT() 256
 Werte
 multiplizieren in Datenbank 582
 suchen 307
 Wertpapier, festverzinsliches 627, 629, 636, 649, 665
 Wetterdaten 864–865
 Weterservice 864–865
 WIEDERHOLEN() 257
 Wikipedia 865

Windows-Registry 296
 Winkel
 berechnen 726
 Funktion 733
 Wochenende 219
 Wochennummer 211, 214
 WOCHENTAG() 145, 225
 Wochentage 214
 Wochentagsnummer 225
 WORKDAY.INTL() 202
 WORKDAY() 201
 WorksheetFunction 191
 World Wide Web Consortium 868
 WURZEL() 799
 Wurzelement 872
 WURZELPI() 800
 WVERWEIS() 307

X

X/Y-Koordinate 729
 XINTZINSFUSS() 686
 XIRR() 686
 XKAPITALWERT() 688
 XML 863, 867
 Datentypen 869
 Entwicklungsziele 868
 Namensräume 870
 Namespace 865
 Struktur 869
 Tags 869
 XMLFILTERN() 865
 XNPV() 688
 XODER() 274, 827
 XOR() 274
 XPath 865, 871
 Achsen 872
 Ancestor 873
 Ancestor-or-self 873
 Attribute 873
 Child 873
 Descendant 873
 Descending-or-self 873
 Dokumentknoten 871
 Following 873
 Following-sibling 873
 Lokalisierungspfade 872
 Match 872
 Mustererkennung 871
 Namespace 873
 Parent 873
 Preceding 873

 Preceding-sibling 873
 Self 873
 Template 872
 XPATH() 869
 XPointer 871
 Xquery 871
 XQUERY() 869
 XSL 869
 XSLT 871

Y

YEAR() 212
 YEARFRAC() 204
 YIELD() 665
 YIELDDISC() 669
 YIELDMAT() 671

Z

Z.TEST() 406
 Z1S1-Bezugsart 318
 Z1S1-Schreibweise 279
 Zahl
 Dezimaldarstellung 742
 Zahlen
 addieren in Datenbank 587
 imaginäre 804
 komplexe 804–805, 828, 832
 reelle 804
 suchen 301
 Zahlenformat 234, 252
 für Datum und Zeit 199
 Standard 198
 Zahlensystem 807
 ZÄHLENWENN() 166, 562
 ZÄHLENWENNS() 564
 ZÄHLENWERT() 259
 Zahlungsziel berechnen 202–203
 Zahlwort 232
 Zehnerlogarithmen 753
 Zehnerlogarithmus 754
 Zeichen, nicht druckbare 247
 ZEICHEN() 156, 232, 247, 254, 260
 Zeichenfolge
 Bezug 279
 ersetzen 235, 255
 finden 237
 in Bezug umwandeln 292
 in Datumswert umwandeln 208

 in Großbuchstaben
 umwandeln 239
 in Kleinbuchstaben
 umwandeln 242
 in numerischen Wert
 umwandeln 256, 259
 in Zahl umwandeln 328
 in Zeitwert umwandeln 228
 Länge ermitteln 243
 suchen 248
 Teil extrahieren 250
 vergleichen 241
 verketteten 254
 von links abschneiden 244
 von rechts abschneiden 246
 wiederholen 257
 Zeichenkette 321, 327
 Bezug 279
 in Bezug umwandeln 292
 Zeichensatz 260, 862
 Double-Byte 231
 Single-Byte 231
 Unicode 231
 Zeigemethode 89
 Zeigen-Modus 150
 ZEILE() 153–154, 309
 ZEILEN() 310
 Zeilenindex 287, 307
 Zeilennummer 309, 334
 ZEIT() 227
 Zeitachsentools 66
 ZEITWERT() 228
 Zellbezüge 279
 \$-Zeichen 91
 absolute 90
 anzeigen 99
 Bezugsart ändern 93
 Dollarzeichen 91
 eingeben 126
 externe 112
 gemischte 92
 relative 90, 104
 ZELLE() 138, 333
 Zellen formatieren 108
 Zelleninformationen 333
 Zellenüberwachungsfenster 113
 Zellverknüpfung 598
 Zentralwert 442
 Zerfallsprozesse 377
 Ziffernvorrat 732, 742
 ZINS() 690
 Zinsanteil 700

- Zinsbindung 647
- Zinsen 224
- Zinseszinsrechnung 619, 632, 691, 703
- Zinsfuß 619
- Zinssatz 619, 690, 692
- ZINSSATZ() 636, 692
- Zinstage 204, 224, 695–697
- Zinstermin 694, 698–699
- ZINSTERMNZ() 694
- ZINSTERMTAGE() 695
- ZINSTERMTAGNZ() 696
- ZINSTERMTAGVA() 697
- ZINSTERMVZ() 698
- ZINSTERMZAHL() 699
- ZINSZ() 700
- Zirkelbezug 94, 113, 336
- Zirkelverweis 94
- ZSATZINVEST() 701
- ZTEST() 406
- z-Transformation 384
- Zufällige Werte 801
- ZUFALLSBEREICH 800
- Zufallsexperiment 362
- Zufallsvariable 590
- Zufallszahl 800
- ZUFALLSZAHL() 801
- Zukunftswert 702, 704
- ZW() 702
- ZW2() 704
- Zweierkomplement 809, 816–817, 819
 - Darstellung 810, 813–815, 818
- Zweierlogarithmus 839
- ZWEIFAKULTÄT() 802, 804
- z-Wert 465
- Zwischenablage 294
 - Inhalte einfügen 257
- Zylinderfunktionen 846
- ZZR() 705