

Microsoft Excel 2019

Formulas and Functions

Paul McFedries



Sample files
on the web

FREE SAMPLE CHAPTER

SHARE WITH OTHERS



Microsoft Excel 2019 Formulas and Functions

Paul McFedries

Microsoft Excel 2019 Formulas and Functions

Published with the authorization of Microsoft Corporation by:

Pearson Education, Inc.

Copyright © 2019 by Pearson Education, Inc.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, request forms, and the appropriate contacts within the Pearson Education Global Rights & Permissions Department, please visit www.pearsoned.com/permissions/. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Nor is any liability assumed for damages resulting from the use of the information contained herein.

ISBN-13: 978-1-5093-0619-0

ISBN-10: 1-5093-0619-6

Library of Congress Control Number: 2019930661

1 19

Trademarks

Microsoft and the trademarks listed at <http://www.microsoft.com> on the “Trademarks” webpage are trademarks of the Microsoft group of companies. All other marks are property of their respective owners.

Warning and Disclaimer

Every effort has been made to make this book as complete and as accurate as possible, but no warranty or fitness is implied. The information provided is on an “as is” basis. The author, the publisher, and Microsoft Corporation shall have neither liability nor responsibility to any person or entity with respect to any loss or damages arising from the information contained in this book.

Special Sales

For information about buying this title in bulk quantities, or for special sales opportunities (which may include electronic versions; custom cover designs; and content particular to your business, training goals, marketing focus, or branding interests), please contact our corporate sales department at corpsales@pearsoned.com or (800) 382-3419.

For government sales inquiries, please contact governmentsales@pearsoned.com.

For questions about sales outside the U.S., please contact intlcs@pearson.com.

Editor-in-Chief: Brett Bartow

Executive Editor: Loretta Yates

Sponsoring Editor: Charvi Arora

Managing Editor: Sandra Schroeder

Senior Project Editor: Tracey Croom

Project Editor: Charlotte Kughen

Indexer: Cheryl Lenser

Proofreader: Gill Editorial Services

Technical Editor: Bob Umlas

Publishing Coordinator: Cindy Teeters

Cover Designer: Twist Creative, Seattle

Compositor: Bronkella Publishing LLC

Graphics: TJ Graham Art

To Karen and Chase

Contents at a Glance

	<i>Introduction</i>	<i>xxiii</i>
Part I	Mastering Excel formulas	1
CHAPTER 1	Building basic formulas	3
CHAPTER 2	Creating advanced formulas	25
CHAPTER 3	Troubleshooting formulas	49
Part II	Harnessing the power of functions	67
CHAPTER 4	Understanding functions	69
CHAPTER 5	Working with text functions	79
CHAPTER 6	Working with logical and information functions	105
CHAPTER 7	Working with lookup functions	135
CHAPTER 8	Working with date and time functions	153
CHAPTER 9	Working with math functions	183
Part III	Building business formulas	205
CHAPTER 10	Implementing basic business formulas	207
CHAPTER 11	Building descriptive statistical formulas	229
CHAPTER 12	Building inferential statistical formulas	245
CHAPTER 13	Applying regression to track trends and make forecasts	265
CHAPTER 14	Building loan formulas	305
CHAPTER 15	Working with investment formulas	325
CHAPTER 16	Building discount formulas	339
Part IV	Building business models	359
CHAPTER 17	Analyzing data with tables	361
CHAPTER 18	Analyzing data with PivotTables	381
CHAPTER 19	Using Excel's business modeling tools	399
CHAPTER 20	Solving complex problems with Solver	421
	<i>Index</i>	<i>443</i>

Contents

	<i>Introduction</i>	<i>xxiii</i>
Part I	Mastering Excel formulas	1
Chapter 1	Building basic formulas	3
	Understanding formula basics	3
	Formula limits in Excel 2019	4
	Entering and editing formulas	4
	Using arithmetic formulas	5
	Using comparison formulas	6
	Using text formulas	6
	Using reference formulas	6
	Understanding operator precedence	7
	The order of precedence	7
	Controlling the order of precedence	8
	Controlling worksheet calculation	9
	Copying and moving formulas	11
	Understanding relative reference format	12
	Understanding absolute reference format	13
	Copying a formula without adjusting relative references	14
	Displaying worksheet formulas	15
	Displaying all worksheet formulas	15
	Displaying a cell's formula by using FORMULATEXT()	15
	Converting a formula to a value	15
	Working with range names in formulas	16
	Pasting a name into a formula	16
	Applying names to formulas	17
	Naming formulas	20
	Working with links in formulas	21
	Understanding external references	22
	Updating links	22
	Changing the link source	23

Chapter 2	Creating advanced formulas	25
	Working with arrays	25
	Using array formulas.....	26
	Using array constants.....	30
	Functions that use or return arrays.....	31
	Using iteration and circular references.....	32
	Consolidating multisheet data	34
	Consolidating by position.....	35
	Consolidating by category.....	38
	Applying data-validation rules to cells.....	40
	Using dialog box controls on a worksheet.....	42
	Displaying the Developer tab.....	42
	Using the form controls.....	43
	Adding a control to a worksheet.....	43
	Linking a control to a cell value.....	43
	Understanding the worksheet controls	44
Chapter 3	Troubleshooting formulas	49
	Understanding Excel's error values	50
	#CALC!.....	50
	#DIV/0!	50
	#FIELD!	51
	#N/A	51
	#NAME?.....	51
	Avoiding #NAME? errors when deleting range names.....	52
	#NULL!.....	53
	#NUM!.....	53
	#REF!.....	53
	#SPILL!	54
	#UNKNOWN!	54
	#VALUE!	54
	Fixing other formula errors.....	55
	Missing or mismatched parentheses.....	55
	Erroneous formula results.....	56
	Fixing circular references.....	56

Handling formula errors with IFERROR()	57
Using the formula error checker	58
Choosing an error action	58
Setting error checker options	59
Auditing a worksheet	62
Understanding auditing	62
Tracing cell precedents	63
Tracing cell dependents	63
Tracing cell errors	64
Removing tracer arrows	64
Evaluating formulas	64
Watching cell values	65
Part II Harnessing the power of functions	67
Chapter 4 Understanding functions	69
About Excel's functions	70
The structure of a function	70
Typing a function into a formula	73
Using the Insert Function feature	74
Loading the Analysis ToolPak	76
Chapter 5 Working with text functions	79
Excel's text functions	79
Working with characters and codes	80
The CHAR() function	81
The CODE() function	83
Converting text	84
The LOWER() function	84
The UPPER() function	84
The PROPER() function	84
The NUMBERVALUE() function	85
Formatting text	85
The DOLLAR() function	86
The FIXED() function	87

The TEXT() function.....	87
Displaying when a workbook was last updated.....	89
Manipulating text.....	89
Removing unwanted characters from a string.....	89
The TEXTJOIN() function: Concatenating text with a delimiter.....	91
The REPT() function: Repeating a character or string.....	92
Extracting a substring.....	94
Converting text to sentence case.....	95
A date-conversion formula.....	95
Searching for substrings.....	97
The FIND() and SEARCH() functions.....	97
Extracting a first name or last name.....	97
Extracting first name, last name, and middle initial.....	99
Determining the column letter.....	99
Substituting one substring for another.....	100
The REPLACE() function.....	100
The SUBSTITUTE() function.....	101
Removing a character from a string.....	101
Removing two different characters from a string.....	102
Removing line feeds.....	103
Chapter 6 Working with logical and information functions	105
Adding intelligence with logical functions.....	105
Using the IF() function.....	106
Performing multiple logical tests.....	109
Combining logical functions with arrays.....	116
Getting data with information functions.....	124
The CELL() function.....	125
The ERROR.TYPE() function.....	128
The INFO() function.....	129
The SHEET() and SHEETS() functions.....	130
The IS functions.....	130
Chapter 7 Working with lookup functions	135
Taking a look at Excel's lookup functions.....	135
Understanding lookup tables.....	136

The CHOOSE() function	137
Determining the name of the day of the week.....	137
Determining the month of the fiscal year	138
Calculating weighted questionnaire results	139
Integrating CHOOSE() and worksheet option buttons.....	139
Looking up values in tables	140
The VLOOKUP() function	140
The HLOOKUP() function	141
Returning a customer discount rate with a range lookup.....	142
Returning a tax rate with a range lookup.....	143
Finding exact matches	144
Advanced lookup operations.....	146
The MATCH() and INDEX() functions.....	146
Chapter 8 Working with date and time functions	153
How Excel deals with dates and times.....	153
Entering dates and times.....	154
Excel and two-digit years	155
Using Excel's date functions	156
Returning a date.....	157
Returning parts of a date.....	159
Calculating the difference between two dates.....	169
Using Excel's time functions.....	172
Returning a time.....	173
Returning parts of a time.....	174
Calculating the difference between two times.....	176
Chapter 9 Working with math functions	183
Excel's math and trig functions	183
Understanding Excel's rounding functions.....	186
The ROUND() function.....	186
The MROUND() function	187
The ROUNDDOWN() and ROUNDUP() functions	188
The CEILING.MATH() and FLOOR.MATH() functions	188
Calculating Easter dates.....	189
The EVEN() and ODD() functions	189

The INT() and TRUNC() functions	190
Using rounding to prevent calculation errors.....	190
Summing values.....	192
The SUM() function	192
The SUMIF() function.....	193
Summing only the positive or negative values in a range.....	193
The SUMIFS() function	193
Calculating cumulative totals.....	194
The MOD() function.....	195
A better formula for time differences	195
Summing every <i>n</i> th row	196
Determining whether a year is a leap year.....	196
Creating ledger shading	197
Generating random numbers	198
The RAND() function.....	199
The RANDBETWEEN() function	201
The RANDARRAY() function.....	202
Creating increasing random numbers with the SEQUENCE() function.....	202
Part III Building business formulas	205
Chapter 10 Implementing basic business formulas	207
Pricing formulas.....	207
Price markups	207
Price discounts	209
Break-even point	211
Financial formulas	213
Sales ratios	214
Cost of goods sold.....	215
Gross margin.....	216
Net margin.....	216
Fixed-asset ratios.....	217
Inventory formulas	218
Inventory ratios.....	218
Inventory management formulas.....	220

Liquidity formulas	221
Accounts receivable ratios	222
Accounts payable ratios	223
Working capital	224
Liquidity ratios	226
Liquidity index	227
Chapter 11 Building descriptive statistical formulas	229
Understanding descriptive statistics	229
Counting items	230
The COUNT() function	230
The COUNTA() function	231
The COUNTBLANK() function	231
The COUNTIF() function	232
The COUNTIFS() function	232
Calculating averages	232
The AVERAGE() function	233
The AVERAGEIF() function	233
The AVERAGEIFS() function	234
The MEDIAN() function	234
The MODE() function	234
Calculating the weighted mean	235
Calculating extreme values	236
The MAX() and MIN() functions	236
The LARGE() and SMALL() functions	237
Performing calculations on the top <i>k</i> values	238
Performing calculations on the bottom <i>k</i> values	238
Working with rank and percentile	238
Calculating rank	239
Calculating percentile	239
Calculating measures of variation	240
Calculating the range	240
Calculating the variance	240
Calculating the standard deviation	241
Working with frequency distributions	242

Chapter 12 Building inferential statistical formulas	245
Understanding inferential statistics	245
Sampling data	246
Extracting a periodic sample	247
Extracting a random sample	249
Determining whether two variables are related	249
Calculating covariance	250
Calculating correlation	251
Working with probability distributions	252
Calculating probability	253
Discrete probability distributions	255
Understanding the normal distribution and the NORM.DIST() function	257
Determining confidence intervals	261
Hypothesis testing	263
Chapter 13 Applying regression to track trends and make forecasts	265
Choosing a regression method	266
Using simple regression on linear data	266
Analyzing trends using best-fit lines	266
Making forecasts	274
Using simple regression on nonlinear data	288
Working with an exponential trend	288
Working with a logarithmic trend	293
Working with a power trend	296
Using polynomial regression analysis	299
Using multiple regression analysis	303
Chapter 14 Building loan formulas	305
Understanding the time value of money	305
Calculating a loan payment	306
Loan payment analysis	307
Working with a balloon loan	308
Calculating interest costs, part I	309

Calculating the principal and interest	309
Calculating interest costs, part II	310
Calculating cumulative principal and interest	311
Building a loan amortization schedule	312
Building a fixed-rate amortization schedule	312
Building a dynamic amortization schedule	313
Calculating the term of a loan	315
Calculating the interest rate required for a loan	317
Calculating how much you can borrow	318
Chapter 15 Working with investment formulas	325
Working with interest rates	325
Understanding compound interest	325
Nominal versus effective interest	326
Converting between the nominal rate and the effective rate	327
Calculating the future value	328
The future value of a lump sum	328
The future value of a series of deposits	329
The future value of a lump sum plus deposits	329
Working toward an investment goal	330
Calculating the required interest rate	330
Calculating the required number of periods	331
Calculating the required regular deposit	332
Calculating the required initial deposit	333
Calculating the future value with varying interest rates	334
Chapter 16 Building discount formulas	339
Calculating the present value	339
Taking inflation into account	340
Calculating present value using PV()	341
Income investing versus purchasing a rental property	342
Buying versus leasing	343
Discounting cash flows	344
Calculating the net present value	345
Calculating net present value using NPV()	346

Net present value with varying cash flows.....	347
Net present value with nonperiodic cash flows.....	348
Calculating the payback period.....	349
Simple undiscounted payback period.....	349
Exact undiscounted payback point.....	350
Discounted payback period.....	351
Calculating the internal rate of return.....	352
Using the IRR() function.....	352
Calculating the internal rate of return for nonperiodic cash flows.....	353
Calculating multiple internal rates of return.....	353
Part IV Building business models	359
Chapter 17 Analyzing data with tables	361
Sorting a table.....	361
Sorting on part of a field.....	362
Sorting without articles.....	363
Sorting table data into an array, part I: The SORT() function.....	364
Sorting table data into an array, part II: The SORTBY() function.....	364
Filtering table data.....	365
Using complex criteria to filter a table.....	366
Entering computed criteria.....	369
Filtering table data with the FILTER() function.....	370
UNIQUE().....	371
Referencing tables in formulas.....	371
Using table specifiers.....	371
Entering table formulas.....	373
Excel's table functions.....	375
About table functions.....	375
Using DAVERAGE().....	377
Using DGET().....	377

Chapter 18 Analyzing data with PivotTables	381
Working with PivotTable subtotals	381
Hiding PivotTable grand totals	382
Hiding PivotTable subtotals	382
Customizing the subtotal calculation	382
Changing the value field summary calculation	383
Using a difference summary calculation	383
Toggling the difference calculation with VBA	385
Using a percentage summary calculation	385
Using a running total summary calculation	388
Using an index summary calculation	389
Creating custom PivotTable calculations	391
Creating a calculated field	393
Creating a calculated item	394
Using PivotTable results in a worksheet formula	395
Chapter 19 Using Excel's business modeling tools	399
Using what-if analysis	399
Setting up a one-input data table	400
Adding more formulas to the input table	402
Setting up a two-input data table	403
Editing a data table	405
Working with Goal Seek	405
How does Goal Seek work?	406
Running Goal Seek	406
Optimizing product margin	408
A note about Goal Seek's approximations	409
Performing a break-even analysis	410
Solving algebraic equations	411
Working with scenarios	412
Understanding scenarios	413
Setting up your worksheet for scenarios	414
Adding a scenario	414
Displaying a scenario	415
Editing a scenario	416

Merging scenarios	417
Generating a summary report	417
Deleting a scenario	419
Chapter 20 Solving complex problems with Solver	421
Some background on Solver	421
The advantages of Solver	422
When do you use Solver?	422
Loading Solver	423
Using Solver	423
Adding constraints	426
Saving a solution as a scenario	428
Setting other Solver options	428
Selecting the method Solver uses	429
Controlling how Solver works	429
Working with Solver models	433
Making sense of Solver's messages	434
Displaying Solver's reports	437
The Answer report	437
The Sensitivity report	439
The Limits report	441
<i>Index</i>	443

Acknowledgments

Substitute damn every time you're inclined to write very; your editor will delete it and the writing will be just as it should be.

—Mark Twain

I didn't follow Mark Twain's advice in this book (the word *very* appears throughout), but if my writing still appears "just as it should be," then it's because of the keen minds and sharp linguistic eyes of the editors at Pearson Education. Near the front of the book you'll find a long list of the hard-working professionals whose fingers made it into this particular paper pie. However, there are a few folks I worked with directly, so I'd like to single them out for extra credit. A big, heaping helping of thanks goes out to executive editor Loretta Yates, project editor Charlotte Kughen, and technical editor Bob Umlas.

About the author

Paul McFedries is an Excel expert and full-time technical writer. Paul has been authoring computer books since 1991 and has more than 95 books to his credit, which combined have sold more than 4 million copies worldwide. His titles include the Que Publishing books *My Office 2016*, *Windows 10 In Depth* (with coauthor Brian Knittel), and *PCs for Grownups*, as well as the Sams Publishing book *Windows 7 Unleashed*. Paul is also the proprietor of Word Spy (www.wordspy.com), a website devoted to *lexpionage*, the sleuthing of new words and phrases that have entered the English language. Please drop by Paul's personal website at mcfedries.com or follow Paul on Twitter, at twitter.com/paulmcf and twitter.com/wordspy.

For companion files, you can visit the book's page at mcfedries.com, <https://mcfedries.com/books/book.php?title=excel-2019-formulas-and-functions>, or at the Microsoft Press Store, MicrosoftPressStore.com/Excel2019FormulasFunctions/downloads.

Troubleshooting formulas

In this chapter, you will:

- Decipher the formula error values that Excel displays
- Learn how to fix a few other common formula error types
- Handle formula errors gracefully with the **IFERROR()** function
- Diagnose formula problems using Excel's Formula Error Checker
- Learn how to use Excel's worksheet auditing tools

Despite your best efforts, the odd error might appear in your formulas from time to time. Such errors can be mathematical (for example, dividing by zero), or Excel might simply be incapable of interpreting the formula. In the latter case, problems can be caught while you're entering the formula. For example, if you try to enter a formula that has unbalanced parentheses, Excel doesn't accept the entry, and it displays an error message. Other errors are more insidious. For example, your formula might *appear* to be working—that is, it might return a value—but the result might be incorrect because the data is flawed or because your formula has referenced the wrong cell or range.

Whatever the error and whatever the cause, formula woes need to be worked out because you or someone else in your company is likely depending on your models to produce accurate results. Don't fall into the trap of thinking that *your* spreadsheets are problem free. A recent University of Hawaii study found that 50% of spreadsheets contained errors that led to "significant miscalculations." And the more complex the model, the greater the chance that errors can creep in. A KPMG study from a few years ago found that a staggering 90% of spreadsheets used for tax calculations contained errors.

The good news is that fixing formula flaws need not be drudgery. With a bit of know-how and Excel's top-notch troubleshooting tools, sniffing out and repairing model maladies isn't hard. This chapter tells you everything you need to know.



Tip If you try to enter an incorrect formula, Excel won't let you do anything else until you either fix the problem or cancel the operation (which means you lose your formula). If the formula is complex, you might not be able to see the problem right away. Instead of deleting all your work, place an apostrophe (') at the beginning of the formula to convert it to text. This way, you can save your work while you try to figure out the problem.

Understanding Excel's error values

When you enter or edit a formula or change one of the formula's input values, Excel might show an error value as the formula result. Excel has eleven different error values: **#CALC!**, **#DIV/0!**, **#FIELD!**, **#N/A**, **#NAME?**, **#NULL!**, **#NUM!**, **#REF!**, **#SPILL!**, **#UNKNOWN!**, and **#VALUE!**. The next few sections give you a detailed look at these error values and offer suggestions for dealing with them.

#CALC!

The new **#CALC!** error appears when Excel's calculation engine comes across a formula or expression that it doesn't support. Examples include one of the following:

- An array nested inside another array
- An array that includes one or more range references
- An array that contains no items

#DIV/0!

The **#DIV/0!** error almost always means that the cell's formula is trying to divide by zero, a mathematical no-no. The cause is usually a reference to a cell that either is blank or contains the value **0**. Check the cell's precedents (the cells that are directly or indirectly referenced in the formula; **see** "Auditing a worksheet," later in this chapter) to look for possible culprits. You'll also see **#DIV/0!** if you enter an inappropriate argument in some functions. **MOD()**, for example, returns **#DIV/0!** if the second argument is **0**.

That Excel treats blank cells as the value **0** can pose problems in a worksheet that requires the user to fill in the data. If your formula requires division by one of the temporarily blank cells, it will show **#DIV/0!** as the result, possibly confusing the user. You can get around this by telling Excel not to perform the calculation if the cell used as the divisor is **0**. This is done with the **IF()** worksheet function, which I discuss in detail in Chapter 6, "Working with logical and information functions." (For an even better way to deal with potential formula errors, **see** "Handling formula errors with **IFERROR()**," later in this chapter.) For example, consider the following formula, which uses named cells to calculate gross margin:

```
=GrossProfit / Sales
```

To prevent the **#DIV/0!** error from appearing if the Sales cell is blank (or 0), you'd modify the formula as follows:

```
=IF(Sales = 0, "", GrossProfit / Sales)
```

If the value of the Sales cell is **0**, the formula returns the empty string; otherwise, it performs the calculation.

#FIELD!

The new **#FIELD!** error value tells you that your formula is referencing an invalid linked data type. This can happen in three ways:

- Your formula includes a data-type reference to a field that isn't found in a linked data type. For example, if cell A1 contains a Stocks data type and your formula references **A1.[52-week high]**, you'll get a **#FIELD!** error because the **52-week high** field doesn't exist. In this example, you can solve the problem by removing the dash: **A1.[52 week high]**.
- Your formula includes a data-type reference (for example, **A1.Price** for a Stocks data type) on a cell that doesn't use a linked data type.
- Your formula includes a data-type reference (for example, **A1.Description** for a Stocks data type) on a cell that doesn't have any data for the referenced field.

#N/A

The **#N/A** error value is short for *not available*, and it means that the formula couldn't return a legitimate result. You usually see **#N/A** when you use an inappropriate argument (or when you omit a required argument) in a function. **HLOOKUP()** and **VLOOKUP()**, for example (see Chapter 7, "Working with lookup functions"), return **#N/A** if the lookup value is smaller than the first value in the lookup range.

To solve the problem, first check the formula's input cells to see whether any of them are displaying the **#N/A** error. If so, that's why your formula is returning the same error; the problem actually lies in the input cell. When you've found where the error originates, examine the formula's operands to look for inappropriate data types. In particular, check the arguments used in each function to ensure that they make sense for the function and that no required arguments are missing.



Note It's common in spreadsheet work to purposely generate an **#N/A!** error to show that a particular cell value isn't currently available. For example, you might be waiting for budget figures from one or more divisions or for the final numbers from month- or year-end. In such a case, you enter **=NA()** into the cell. You fix this "problem" by replacing the **NA()** function with the appropriate data when it arrives.

#NAME?

The **#NAME?** error appears when Excel doesn't recognize a name you used in a formula or when it interprets text within the formula as an undefined name. This means that the **#NAME?** error pops up in a wide variety of circumstances:

- You spelled a range name incorrectly.
- You used a range name that you haven't yet defined.

- You spelled a function name incorrectly.
- You used a function that's part of an uninstalled add-in.
- You used a string value without surrounding it with quotation marks.
- You entered a range reference and accidentally omitted the colon.
- You entered a reference to a range on another worksheet and didn't enclose the sheet name in single quotation marks.



Tip When entering function names and defined names, use all lowercase letters. If Excel recognizes a name, it converts the function to all uppercase and the defined name to its original case. If no conversion occurs, you know that you misspelled the name, you haven't defined it yet, or you're using a function from an add-in that isn't loaded.

Remember that you also can use these commands to enter functions and names safely:

Formulas > Insert Function (or select Shift+F3); **Formulas > Use In Formula**; or **Formulas > Use In Formula > Paste Names** (or select F3).

These are mostly syntax errors, so fixing them means double-checking your formula and correcting range name or function name misspellings, or inserting missing quotation marks or colons. Also, be sure to define any range names you use and to install the appropriate add-in modules for functions you use.

Avoiding #NAME? errors when deleting range names

If you've used a range name in a formula and then you delete that name, Excel generates the #NAME? error. Wouldn't it be better if Excel just converted the name to its appropriate cell reference in each formula? Possibly, but there is an advantage to Excel's seemingly inconvenient approach. By generating an error, Excel enables you to catch range names that you delete by accident. Because Excel leaves the names in the formula, you can recover by redefining the original range name.

Redefining the original range name becomes problematic if you can't remember the appropriate range coordinates. This is why it's always a good idea to paste a list of range names and their references into each of your worksheets (select **Formulas > Use In Formula > Paste Names > Paste List**).

If you don't need this safety net, you can force Excel to convert deleted range names into their cell references. Here are the steps to follow:

1. Select **File > Options** to display the Excel Options dialog box.
2. Select **Advanced**.
3. In the **Lotus Compatibility Settings For** section, use the list to select the worksheet you want to use.

4. Select the **Transition Formula Entry** check box.
5. Select **OK**.

Excel now treats your formula entries the same way Lotus 1-2-3 did way back in days of yore. Specifically, in formulas that use a deleted range name, the name automatically gets converted to its appropriate range reference. As an added bonus, Excel also performs the following automatic conversions:

- If you enter a range reference in a formula, the reference gets converted to a range name (provided that a name exists, of course).
- If you define a name for a range, Excel converts any existing range references into the new name.



Caution The treatment of formulas in the Lotus 1-2-3 manner only applies to formulas that you create *after* you select the Transition Formula Entry check box.

#NULL!

Excel displays the **#NULL!** error in a very specific case: when you use the intersection operator (a space) on two ranges that have no cells in common. For example, because the ranges A1:B2 and C3:D4 have no common cells, the following formula returns the **#NULL!** error:

```
=SUM(A1:B2 C3:D4)
```

Check your range coordinates to ensure that they're accurate. Also, check to see if one of the ranges has been moved, causing the two ranges in your formula to no longer intersect.

#NUM!

The **#NUM!** error means there's a problem with a number in a formula. This almost always means that you entered an invalid argument in a math or trig function. For example, perhaps you entered a negative number as the argument for the **SQRT()** or **LOG()** function. Check the formula's input cells—particularly those that are used as arguments for mathematical functions—to make sure the values are appropriate.

The **#NUM!** error also appears if you're using iteration (or a function that uses iteration; see Chapter 2, "Creating advanced formulas"), and Excel can't calculate a result. There could be no solution to the problem, or you might need to adjust the iteration parameters.

#REF!

The **#REF!** error means that a formula contains an invalid cell reference, which is usually caused by one of the following actions:

- You deleted a cell to which the formula refers. You need to add the cell back in or adjust the formula reference.

- You cut a cell and then pasted it into a cell used by the formula. You need to undo the cut and then paste the cell elsewhere. (Note that it's okay to *copy* a cell and paste it onto a cell used by the formula.)
- Your formula references a nonexistent cell address, such as B0. This can happen if you cut or copy a formula that uses relative references and paste it in such a way that the invalid cell address is created. For example, suppose that your formula references cell B1. If you cut or copy the cell containing the formula and paste it one row higher, the reference to B1 becomes invalid because Excel can't move the cell reference up one row.

#SPILL!

You see the new **#SPILL!** error when a dynamic array formula or function that needs to spill data into one or more adjacent cells encounters a non-empty cell in the spill range. (See Chapter 2 to learn more about spill ranges.) You have two ways to handle a **#SPILL!** error:

- Move the formula that needs the spill range to a new location where the spill range is empty.
- Move the existing data out of the spill range.

In both cases, Excel removes the **#SPILL!** error and spills the formula result in the new spill range.

#UNKNOWN!

If you see the new **#UNKNOWN!** error, it means your formula or expression is referencing a data type that isn't supported in your version of Excel. This most often means that the reference is to a data type that was added to a later version of Excel, so the only way to resolve the error is to upgrade to a version of Excel that supports the data type.

#VALUE!

When Excel generates a **#VALUE!** error, it means you've used an inappropriate argument in a function. This is most often caused by using the wrong data type. For example, you might have entered or referenced a string value instead of a numeric value. Similarly, you might have used a range reference in a function argument that requires a single cell or value. Excel also generates this error if you use a value that's larger or smaller than Excel can handle. In all these cases, you solve the problem by double-checking your function arguments to find and edit the inappropriate arguments.



Note Excel can work with values between $-1E-307$ and $1E+307$.

Fixing other formula errors

Not all formula errors generate one of Excel's seven error values. Instead, you might see a warning dialog box from Excel (for example, if you try to enter a function without including a required argument). Or, you might not see any indication that something is wrong. To help you in these situations, the following sections cover some of the most common formula errors.

Missing or mismatched parentheses

If you miss a parenthesis when typing a formula, or if you place a parenthesis in the wrong location, Excel usually displays a dialog box like the one shown in Figure 3-1 when you attempt to confirm the formula. If the edited formula is what you want, select Yes to have Excel enter the corrected formula automatically; if the edited formula is not correct, select No and edit the formula by hand.

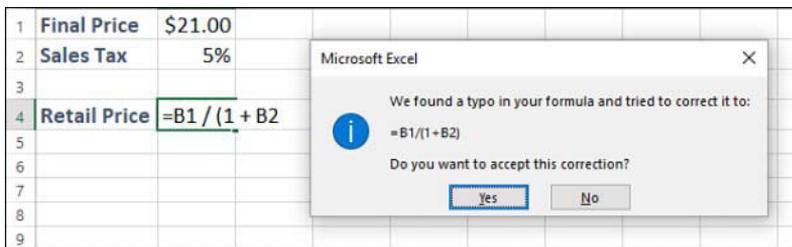


FIGURE 3-1 If you miss a parenthesis, Excel attempts to fix the problem and displays this dialog box to ask if you want to accept the correction.



Caution Excel doesn't always fix missing parentheses correctly. It tends to add the missing parenthesis to the end of the formula, which is often not what you want. Therefore, always check Excel's proposed solution carefully before accepting it.

To help you avoid missing or mismatched parentheses, Excel provides two visual clues in the formula itself when you're editing it:

- The first clue occurs when you type a right parenthesis. Excel temporarily bolds both the right parenthesis and its corresponding left parenthesis. If you type what you think is the last right parenthesis but Excel doesn't bold the first left parenthesis, your parentheses are unbalanced.
- The second clue occurs when you use the left and right arrow keys to navigate a formula. When you cross over a parenthesis, Excel bolds the other parenthesis in the pair.

Erroneous formula results

If a formula produces no warnings or error values, the result might still be in error. If the result of a formula is incorrect, here are a few techniques that can help you understand and fix the problem:

- **Calculate complex formulas one term at a time.** In the formula bar, select the expression you want to calculate and then select F9. Excel converts the expression into its value. Make sure that you select the Esc key when you're done to avoid entering the formula with just the calculated values.
- **Evaluate the formula.** You can step through the various parts of a formula, a technique that I describe later in this chapter (see "Evaluating formulas").
- **Break up long or complex formulas.** One of the most problematic aspects of formula troubleshooting is making sense out of long formulas. The previous techniques can help (by enabling you to evaluate parts of the formula), but it's usually best to keep your formulas as short as you can at first. When you get things working properly, you often can combine formulas for a more efficient model.
- **Recalculate all formulas.** A formula might display the wrong result because other formulas on which it depends need to be recalculated. This is particularly true if one or more of those formulas use custom VBA functions. Select Shift+F9 to recalculate all worksheet formulas.
- **Pay attention to operator precedence.** As explained in Chapter 1, "Building basic formulas," Excel's operator precedence means that certain operations are performed before others. An erroneous formula result could, therefore, be caused by Excel's precedence order. To control precedence, use parentheses.
- **Watch out for nonblank "blank" cells.** A cell might appear to be blank but actually contain data or even a formula. For example, some users "clear" a cell by selecting the spacebar, and Excel then treats the cell as nonblank. Similarly, some formulas return an empty string instead of a value. (For example, see the **IF()** function formula earlier in this chapter for avoiding the **#DIV/0!** error.)
- **Watch unseen values.** In a large model, your formula could be using cells that you can't see because they're offscreen or on another sheet. Excel's Watch Window enables you to keep an eye on the current value of one or more cells. To learn about the Watch Window, see "Watching cell values," later in this chapter.

Fixing circular references

A *circular reference* occurs when a formula refers to its own cell. This can happen in one of two ways:

- **Directly:** The formula explicitly references its own cell. For example, a circular reference would result if the following formula were entered into cell A1:

```
=A1+A2
```

- **Indirectly:** The formula references a cell or function that, in turn, references the formula's cell. For example, suppose that cell A1 contains the following formula:
`=A5*10`
- A circular reference would result if cell A5 referred to cell A1, as in this example:
`=SUM(A1:D1)`

When Excel detects a circular reference, it displays the dialog box shown in Figure 3-2.

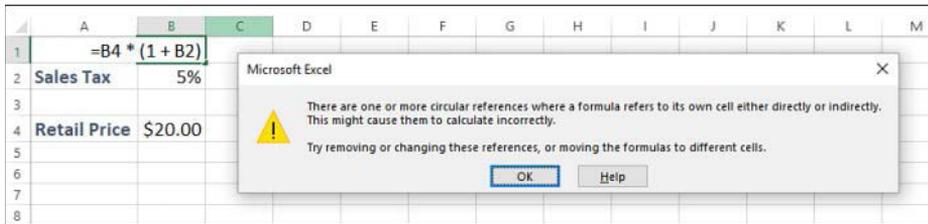


FIGURE 3-2 If you attempt to enter a formula that contains a circular reference, Excel displays this dialog box.

When you select OK, Excel displays *tracer* arrows that connect the cells involved in the circular reference. (Tracers are discussed in detail later in this chapter; see “Auditing a worksheet.”) Knowing which cells are involved enables you to correct the formula in one of them to solve the problem.

Handling formula errors with IFERROR()

Earlier you saw how to use the **IF()** function to avoid a **#DIV/0!** error by testing the value of the formula divisor to see if it equals 0. This works fine if you can anticipate the specific type of error the user might make. However, in many instances you can't know the exact nature of the error in advance. For example, the simple formula `=GrossProfit/Sales` would generate a **#DIV/0!** error if **Sales** equals 0. However, it would generate a **#NAME?** error if the name **GrossProfit** or the name **Sales** no longer exists, or it would generate a **#REF!** error if the cells associated with one or both of **GrossProfit** and **Sales** were deleted.

If you want to handle errors gracefully in your worksheets, it's often best to assume that *any* error can occur. Fortunately, that doesn't mean you have to construct complex tests using deeply nested **IF()** functions that check for every error type (**#DIV/0!**, **#N/A**, and so on). Instead, Excel enables you to use a simple test for *any* error by offering the **IFERROR()** function:

`IFERROR(value, value_if_error)`

`value` The expression that might generate an error

`value_if_error` The value to return if `value` returns an error

If the *value* expression doesn't generate an error, **IFERROR()** returns the expression result; otherwise, it returns *value_if_error* (which might be the null string or an error message). Here's an example:

```
=IFERROR(GrossProfit / Sales, "")
```



Note If you want to handle the specific case of the #N/A error, use the **IFNA(value, value_if_na)** function. Here, *value* is the expression that you're testing for the #N/A error, and *value_if_na* is the value to return if *value* returns the #N/A error.

Using the formula error checker

If you use Microsoft Word, you're probably familiar with the double blue line that appears under words and phrases that the grammar checker has flagged as being incorrect. The grammar checker operates by using a set of rules that determine correct grammar and syntax. As you type, the grammar checker operates in the background, constantly monitoring your writing. If something you write goes against one of the grammar checker's rules, the wavy line appears to let you know there's a problem.

Excel has a similar feature: the formula error checker. Like the grammar checker, the formula error checker uses a set of rules to determine correctness, and it operates in the background to monitor your formulas. If it detects something amiss, it displays an *error indicator*—a green triangle—in the upper-left corner of the cell containing the formula, as shown in Figure 3-3.

Error indicator

	A	B	C	D	E
1	Final Price	\$21.00	\$26.25	\$31.50	
2	Sales Tax	5%	5%	5%	
3					
4	Retail Price	\$20.00	\$25.00	\$30.00	
5					

FIGURE 3-3 If Excel's formula error checker detects a problem, it displays a green triangle in the upper-left corner of the formula's cell.

Choosing an error action

When you select the cell with the formula error, Excel displays a formula error icon beside the cell. If you hover your mouse pointer over the icon, a pop-up message describes the error, as shown in Figure 3-4. The formula error icon drop-down menu contains the following actions:

- **Corrective action:** This is a command (the name of which depends on the type of error) that Excel believes either will fix the problem or help you troubleshoot the error. In Figure 3-4, for

example, Excel is reporting that the formula in cell C3 differs from its neighboring formulas. (In the formula bar, the expression in the parentheses should be **1+C2** instead of **1-C2**.) In this case, the corrective action command in the formula error icon is Copy Formula From Left. Or, if Excel can't suggest a solution, it might show the command Show Calculation Steps, which runs the Evaluate Formula feature. See "Evaluating formulas," later in this chapter.

- **Help On This Error:** Select this option to get information on the error via the Excel Help system.
- **Ignore Error:** Select this option to leave the formula as is.
- **Edit In Formula Bar:** Select this option to display the formula in Edit mode in the formula bar. You can then fix the problem by editing the formula.
- **Error-Checking Options:** Select this option to display the Formulas tab of the Excel Options dialog box (discussed next).

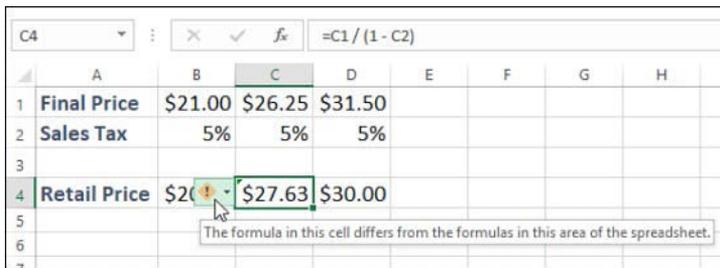


FIGURE 3-4 Select the cell containing the error and then move the mouse pointer over the formula error icon to see a description of the error.

Setting error checker options

Like Word's grammar checker, Excel's formula error checker has a number of options that control how it works and which errors it flags. To see these options, you have two choices:

- Select **File > Options > Formulas** to open the Excel Options dialog box and display the **Formulas** tab.
- Select **Error-Checking Options** in the formula error icon's drop-down menu (as described in the previous section).

Either way, the options appear in the Error Checking and Error Checking Rules sections in the **Formulas** tab, as shown in Figure 3-5.

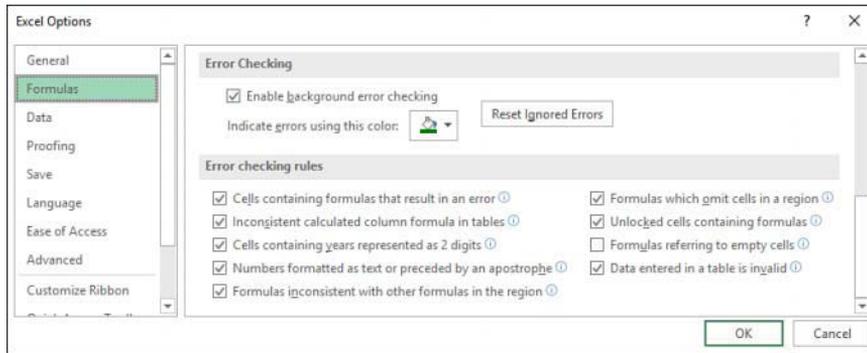


FIGURE 3-5 In the Formulas tab, the Error Checking and Error Checking Rules sections contain the options that govern the workings of the formula error checker.

Here's a rundown of the available options:

- **Enable Background Error Checking:** This check box toggles the formula error checker's background operation on and off. If you turn off the background checking, you can run a check at any time by choosing **Formulas > Error Checking**.
- **Indicate Errors Using This Color:** Use this color palette to select the color of the error indicator.
- **Reset Ignored Errors:** If you've ignored one or more errors, you can redisplay the error indicators by selecting this button.
- **Cells Containing Formulas That Result In An Error:** When this check box is selected, the formula error checker flags formulas that evaluate to **#DIV/0!**, **#NAME?**, or any of the other error values discussed earlier.
- **Inconsistent Calculated Column Formula In Tables:** When this check box is selected, Excel examines the formulas in a table's calculated column and flags any cell that contains a formula with a different structure than the other cells in the column. The formula error icon for this error includes the command **Restore To Calculated Column Formula**, which enables you to update the formula so that it's consistent with the rest of the column.
- **Cells Containing Years Represented As 2 Digits:** When this check box is selected, the formula error checker flags formulas that contain date text strings in which the year contains only two digits (a possibly ambiguous situation because the string could refer to a date in either the 1900s or the 2000s). In such a case, the list of options supplied in the formula error icon contains two commands—**Convert XX To 19XX** and **Convert XX To 20XX**—that enable you to convert the two-digit year to a four-digit year.
- **Numbers Formatted As Text Or Preceded By An Apostrophe:** When this check box is selected, the formula error checker flags cells that contain a number that is either formatted as text or preceded by an apostrophe. In such a case, the list of options supplied in the formula error icon contains the **Convert To Number** command to convert the text to its numeric equivalent.

- **Formulas Inconsistent With Other Formulas In The Region:** When this check box is selected, the formula error checker flags formulas that are structured differently from similar formulas in the surrounding area. In such a case, the list of options supplied in the formula error icon contains a command such as **Copy Formula From Left** to make the formula consistent with the surrounding cells.
- **Formulas Which Omit Cells In A Region:** When this check box is selected, the formula error checker flags formulas that omit cells that are adjacent to a range referenced in the formula. For example, suppose that the formula is **=AVERAGE(C4:C21)**, where **C4:C21** is a range of numeric values. If cell C3 also contains a numeric value, the formula error checker flags the formula to alert you to the possibility that you missed including cell C3 in the formula. Figure 3-6 shows this example. In such a case, the list of options supplied in the formula error icon will contain the command **Update Formula To Include Cells** to adjust the formula automatically.

Sales Rep	2017 Sales	2018 Sales
Nancy Freehafer	\$996,336	\$960,492
Andrew Cencini	\$606,731	\$577,983
Jan Kotas	\$622,781	\$967,580
Mariya Sergienko	\$765,327	\$771,399
Steven Thorpe	\$863,589	\$827,213
Michael Neipper	\$795,518	\$669,394
Robert Zare	\$722,740	\$626,945
Laura Giussani	\$992,059	\$574,472
Anne Hellung-Larsen	\$659,380	\$827,932
Maria Anders	\$509,623	\$569,609
Thomas Hardy	\$987,777	\$558,601
Hanna Moos	\$685,091	\$692,182
Victoria Ashworth	\$540,484	\$693,762
Patricio Simpson	\$650,733	\$823,034
Elizabeth Brown	\$509,863	\$511,569
Ann Devon	\$503,699	\$975,455
Paolo Accorti	\$630,263	\$599,514
Carlos Hernández	\$779,722	\$596,353
Yoshi Latimer	\$592,802	\$652,171
TOTAL	\$13,414,518	\$13,475,660
AVERAGE	\$689,899	

FIGURE 3-6 The formula error checker can flag formulas that omit cells that are adjacent to a range referenced by the formula. In this case, the formula in C23 should include cell C3.

- **Unlocked Cells Containing Formulas:** When this check box is selected, the formula error checker flags formulas that reside in unlocked cells. This isn't an error so much as a warning that other people could tamper with the formula even after you have protected the sheet. In such a case, the list of options supplied in the formula error icon will contain the command **Lock Cell** to lock the cell and prevent users from changing the formula after you protect the sheet.
- **Formulas Referring To Empty Cells:** When this check box is selected, the formula error checker flags formulas that reference empty cells. In such a case, the list of options supplied in the formula error icon will contain the command **Trace Empty Cell** to enable you to find the

empty cell. (At this point, you can either enter data into the cell or adjust the formula so that it doesn't reference the cell.)

- **Data Entered In A Table Is Invalid:** When this check box is selected, the formula error checker flags cells that violate a table's data-validation rules. This can happen if you set up a data-validation rule with only a Warning or Information style, in which case the user can still opt to enter the invalid data. In such cases, the formula error checker will flag the cells that contain invalid data. The formula error icon list includes the **Display Type Information** command, which shows the data-validation rule that the cell data violates.

Auditing a worksheet

As you've seen, some formula errors result from referencing other cells that contain errors or inappropriate values. The first step in troubleshooting these kinds of formula problems is to determine which cell (or group of cells) is causing an error. This is straightforward if the formula references only a single cell, but it gets progressively more difficult as the number of references increases. (Another complicating factor is the use of range names because it won't be obvious which range each name is referencing.)

To determine which cells are wreaking havoc on your formulas, you can use Excel's auditing features to visualize and trace a formula's input values and error sources.

Understanding auditing

Excel's formula-auditing features operate by creating *tracers*—arrows that literally point out the cells involved in a formula. You can use tracers to find three kinds of cells:

- **Precedents:** These are cells that are directly or indirectly referenced in a formula. For example, suppose that cell B4 contains the formula `=B2`; then B2 is a direct precedent of B4. Now suppose that cell B2 contains the formula `=A2/2`; this makes A2 a direct precedent of B2, but it's also an *indirect* precedent of cell B4.
- **Dependents:** These are cells that are directly or indirectly referenced by a formula in another cell. In the preceding example, cell B2 is a direct dependent of A2, and B4 is an indirect dependent of A2.
- **Errors:** These are cells that contain an error value and are directly or indirectly referenced in a formula (and therefore cause the same error to appear in the formula).

Figure 3-7 shows a worksheet with three examples of tracer arrows:

- Cell B4 contains the formula `=B2`, and B2 contains `=A2/2`. The arrows (they're blue onscreen) point out the precedents (direct and indirect) of B4.

- Cell D4 contains the formula =D2, and D2 contains =D1/0. The latter produces the #DIV/0! error. Therefore, the same error appears in cell D4. The arrow (it's red onscreen) is pointing out the source of the error.
- Cell G4 contains the formula =Sheet5!A1. Excel displays the dashed arrow with the worksheet icon whenever the precedent or dependent exists on a different worksheet.

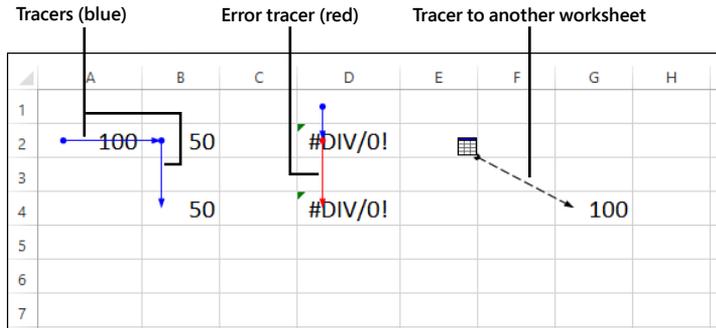


FIGURE 3-7 This worksheet demonstrates the three types of tracer arrows.

Tracing cell precedents

To trace cell precedents, follow these steps:

1. Select the cell that contains the formula whose precedents you want to trace.
2. Select **Formulas > Trace Precedents**. Excel adds a tracer arrow to each direct precedent.
3. Keep repeating step 2 to see more levels of precedents.



Tip You also can trace precedents by double-clicking the cell, provided that you turn off in-cell editing. You do this by choosing **File > Options > Advanced** and then deselecting the **Allow Editing Directly In Cells** check box. Now when you double-click a cell, Excel selects the formula's precedents.

Tracing cell dependents

Here are the steps to follow to trace cell dependents:

1. Select the cell whose dependents you want to trace.
2. Select **Formulas > Trace Dependents**. Excel adds a tracer arrow to each direct dependent.
3. Keep repeating step 2 to see more levels of dependents.

Tracing cell errors

To trace cell errors, follow these steps:

1. Select the cell that contains the error you want to trace.
2. Select **Formulas > Error Checking > Trace Error**. Excel adds a tracer arrow to each cell that produced the error.

Removing tracer arrows

To remove the tracer arrows, you have three choices:

- To remove all the tracer arrows, select **Formulas > Remove Arrows**.
- To remove precedent arrows one level at a time, select **Formulas**, drop down the **Remove Arrows** list, and select **Remove Precedent Arrows**.
- To remove dependent arrows one level at a time, select **Formulas**, drop down the **Remove Arrows** list, and select **Remove Dependent Arrows**.

Evaluating formulas

Earlier, you learned that you can troubleshoot a wonky formula by evaluating parts of it. You do this by selecting the part of the formula you want to evaluate and then selecting F9. This works fine, but it can be tedious in a long or complex formula, and there's always a danger that you might accidentally confirm a partially evaluated formula and lose your work.

A better solution is to use Excel's Evaluate Formula feature. It does the same thing as the F9 technique, but it's easier and safer. Here's how it works:

1. Select the cell that contains the formula you want to evaluate.
2. Select **Formulas > Evaluate Formula**. Excel displays the Evaluate Formula dialog box.
3. The current term in the formula is underlined in the Evaluation box. At each step, you select from one or more of the following buttons:
 - **Evaluate:** Display the current value of the underlined term.
 - **Step In:** Display the first dependent of the underlined term. If that dependent also has a dependent, select this button again to see it (see Figure 3-8).
 - **Step Out:** Hide a dependent and evaluate its precedent.
4. Repeat step 3 until you've completed your evaluation.
5. Select Close.

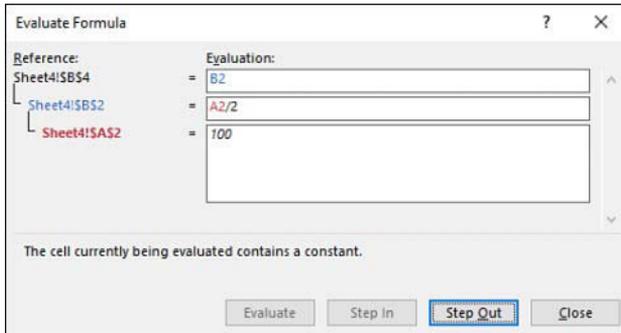


FIGURE 3-8 With the Evaluate Formula feature, you can “step into” the formula to display its dependent cells.

Watching cell values

In the precedent tracer example shown in Figure 3-7, the formula in cell G4 refers to a cell in another worksheet, which is represented in the trace by a worksheet icon. In other words, you can’t see the formula cell and the precedent cell at the same time. This could also happen if the precedent existed on another workbook or even elsewhere on the same sheet if you’re working with a large model.

This is a problem because there’s no easy way to determine the current contents or value of the unseen precedent. If you’re having a problem, troubleshooting requires that you track down the far-off precedent to see if it might be the culprit. That’s bad enough with a single unseen cell, but what if your formula refers to 5 or 10 such cells? And what if those cells are scattered in different worksheets and workbooks?

This level of hassle—not at all uncommon in the spreadsheet world—was no doubt the inspiration behind an elegant solution: the Watch Window. This window enables you to keep tabs on both the value and the formula in any cell in any worksheet in any open workbook. Here’s how you set up a watch:

1. Switch to the workbook that contains the cell or cells you want to watch.
2. Select **Formulas > Watch Window**. Excel displays the Watch Window.
3. Select **Add Watch**. Excel displays the Add Watch dialog box.
4. Either select the cell you want to watch or type in a reference formula for the cell (for example, =A1). Note that you can select a range to add multiple cells to the Watch Window.
5. Select Add. Excel adds the cell or cells to the Watch Window, as shown in Figure 3-9.

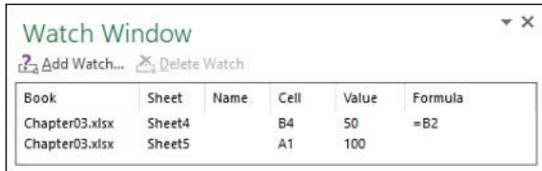


FIGURE 3-9 Use the Watch Window to keep an eye on the values and formulas of unseen cells that reside in other worksheets or workbooks.

When you no longer need a watch, you should remove it to avoid cluttering the Watch Window. To remove a watch, select **Formulas > Watch Window** to open the Watch Window, select the watch, and then select **Delete Watch**.

Index

Symbols

- & (ampersand) operator, 6
- { } (braces), in array formulas, 27
- , (commas), arguments and, 72
- \$ (dollar sign), in cell references, 13
- () (parentheses), in order of precedence, 8-9
- (' ') (single quotation marks), in external references, 22
- #CALC! error value, 50
- #DIV/0! error value, 50
- #FIELD! error value, 51
- #N/A! error value, 51, 58
- #NAME? error value, 51-53
- #NULL! error value, 53
- #NUM! error value, 53, 353
- #REF! error value, 53-54
- #SPILL! error value, 28, 54, 202
- #UNKNOWN! error value, 54
- #VALUE! error value, 54, 248

A

- ABS() function, 183
- absolute reference format, 13
 - ignoring when applying range names, 18
- account numbers
 - generating, 96, 102-103
 - looking up, 144-145
- accounts payable ratios, 223
 - accounts payable turnover, 223
 - days purchases in accounts payable, 223-224
 - sales to accounts payable, 224
- accounts payable turnover, 223
- accounts receivable aging worksheet, building, 122-124
- accounts receivable ratios, 222
 - accounts receivable turnover, 222
 - average AR duration, 222-223
- accounts receivable turnover, 222
- acid-test ratio, 226-227
- ACOS() function, 185
- ACOSH() function, 185
- ACOT() function, 185
- ACOTH() function, 186
- adding. *See also* inserting
 - constraints in Solver, 426-428
 - every nth row with MOD() function, 196
 - formulas to data tables, 402-403
 - months to dates, 163
 - with EDATE() function, 163
 - with EOMONTH() function, 164
 - scenarios
 - to Quick Access Toolbar, 416
 - to worksheets, 414-415
 - times, 176
 - values, 192
 - with SUM() function, 192-195
 - with SUMIF() function, 193
 - with SUMIFS() function, 193-194

add-ins

- add-ins, loading Solver, 423
- ADDRESS() function, 122
- advertising, analyzing versus sales, 274
- ages, calculating, 169-170
- AGGREGATE() function, 183
- aging overdue invoices, 123-124
- aging worksheet (accounts receivable), building, 122-124
- algebraic equations, solving, 411-412
- allowing principal paydowns, 321-323
- amortization schedules, building, 312-315, 320-321
- ampersand (&) operator, 6
- Analysis ToolPak
 - loading, 76-77
 - Regression tool, 273
- analyzing
 - data. *See* Goal Seek; scenarios; what-if analysis
 - table of defects, 378-379
- anchored cell references, 13
- AND() function, 106, 111-113
 - aging overdue invoices, 124
 - applying conditions across ranges, 116
 - looking up list values, 118
- ANSI character set, 80
 - generating, 81-82
- ANSI codes, 80
 - CHAR() function, 81-83
 - CODE() function, 83
- Answer report (Solver), 437-439
- applying range names to formulas, 17-20
- approximations in Goal Seek, 409-410
- AP. *See* accounts payable ratios
- ARABIC() function, 183
- arguments
 - commas and, 72
 - defined, 71
 - forms allowed, 72
 - in Function Arguments dialog box, 75-76
 - syntax, 71
- arithmetic formulas, operators for, 5-6
- array constants, 30-31
- array formulas
 - array constants in, 30-31
 - correspondences in, 28
 - defined, 25
 - dynamic, 27-28
 - entering, 26-27
 - on multiple ranges, 29
- arrays
 - creating, 25
 - defined, 25
 - functions using/returning, 31-32
 - logical functions combined with, 116-121
 - resizing, 27
 - selecting, 27
 - sorting tables into
 - with SORTBY() function, 364-365
 - with SORT() function, 364
- AR. *See* accounts receivable ratios
- articles, sorting without, 363-364
- ASIN() function, 186
- ASINH() function, 186
- ATAN() function, 186
- ATAN2() function, 186
- ATANH() function, 186
- auditing worksheets, 62-63
 - dependent tracing, 63
 - error cell tracing, 64
 - evaluating formulas, 64-65
 - precedent tracing, 63
 - removing tracer arrows, 64
 - setting Watch Windows, 65-66
- AutoComplete feature for functions, 73-74
- automatic recalculation, 10
- average AR duration, 222-223
- average deviation, 240
- average difference, 240

AVERAGE() function, 233
 AVERAGEA() function, 233
 AVERAGEIF() function, 233
 AVERAGEIFS() function, 234
 averages, calculating, 232
 with AVERAGE() function, 233
 with AVERAGEA() function, 233
 with AVERAGEIF() function, 233
 with AVERAGEIFS() function, 234
 with MEDIAN() function, 234
 with MODE() function, 234
 with SUMPRODUCT() function, 235-236

B

BAHTTEXT() function, 79
 balloon loan payments, calculating, 308
 BASE() function, 183
 bell curves, 257-258. *See also* normal distribution
 Bernoulli sampling process, 255
 best-fit trendlines, 266
 analyzing sales versus advertising, 274
 calculating
 with LINEST() function, 271-273
 with TREND() function, 270-271
 plotting, 267-268
 R2 for, 269-270
 regression equation for, 268-269
 billable time, rounding, 191-192
 bin interval, 242
 BINOM.DIST() function, 255
 binomial distributions, 255
 bins, 242
 birthday, returning, 164
 blanks, counting in ranges, 131-132
 Boolean logic. *See* logical functions
 borrowed amount, calculating, 318-319
 braces ({}), in array formulas, 27
 break-even analysis
 with Goal Seek, 410-411
 with Solver, 423-426
 break-even point, 211-213
 building. *See also* creating; entering
 investment schedules, 335-337
 loan amortization schedules, 312-315, 320-321
 built-in functions. *See* functions
 business forecasting. *See* regression analysis
 business formulas
 financial, 213
 cost of goods sold, 215
 fixed-asset ratios, 217-218
 gross margin, 216
 net margin, 216
 sales ratios, 214-215
 inventory, 218
 inventory management, 220-221
 inventory ratios, 218-220
 liquidity, 221
 accounts payable ratios, 223-224
 accounts receivable ratios, 222-223
 liquidity index, 227-228
 liquidity ratios, 226-227
 working capital, 224-225
 pricing, 207
 break-even point, 211-213
 discounts, 209-211
 markups, 207-209
 business modeling tools. *See* Goal Seek; scenarios; what-if analysis
 buying equipment, leasing versus, 343-344

C

#CALC! error value, 50
 calculated fields
 creating, 393-394
 defined, 391

calculated items

- calculated items
 - creating, 394-395
 - defined, 392
- calculating
 - accounts payable turnover, 223
 - accounts receivable turnover, 222
 - acid-test ratio, 226-227
 - ages, 169-170
 - average AR duration, 222-223
 - averages, 232
 - with AVERAGE() function, 233
 - with AVERAGEA() function, 233
 - with AVERAGEIF() function, 233
 - with AVERAGEIFS() function, 234
 - with MEDIAN() function, 234
 - with MODE() function, 234
 - with SUMPRODUCT() function, 235-236
 - best-fit trendlines
 - with LINEST() function, 271-273
 - with TREND() function, 270-271
 - borrowed amount, 318-319
 - break-even point, 211-213
 - cash ratio, 227
 - confidence intervals, 261-262
 - correlation, 251-252
 - cost of goods sold, 215
 - cost price, 209
 - covariance, 250-251
 - cumulative principal and interest, 311-312
 - cumulative totals, 194-195
 - current ratio, 226
 - days purchases in accounts payable, 223-224
 - deposits for investment goals, 332-333
 - difference between dates, 169-170
 - with DAYS() function, 170
 - with DAYS360() function, 171-172
 - with NETWORKDAYS() function, 170-171
 - with YEARFRAC() function, 172
 - differences between times, 176-177
 - with MOD() function, 195
 - discount amount, 210
 - discount rate, 210-211
 - exponential trend values, 290-291
 - extreme values, 236
 - with LARGE() and SMALL() functions, 237-238
 - with MAX() and MIN() functions, 236
 - fixed assets to short-term debt, 218
 - fixed-asset turnover, 217
 - frequency distributions, 242-244
 - future value, 328-330
 - for investment goals, 334
 - gross margin, 216
 - holiday dates, 167-168
 - Easter Sunday, 189
 - initial deposit for investment goals, 333
 - interest costs, 309-310
 - interest rate, 317-318
 - for investment goals, 330-331
 - internal rate of return, 352
 - with IRR() function, 352-353
 - with MIRR() function, 353-354
 - with XIRR() function, 353
 - inventory reorder level, 221
 - inventory safety stock, 220-221
 - inventory turnover, 219
 - inventory turnover rate, 219
 - Julian dates, 169
 - liquidity index, 227-228
 - list price, 211
 - loan payments, 306-312
 - logarithmic trend values, 295-296
 - markup amount, 208
 - markup rate, 208-209
 - net margin, 216
 - net present value, 345-346
 - with NPV() function, 346-348
 - with XNPV() function, 348-349
 - net price, 209-210

- payback period, 349
 - discounted, 351
 - exact undiscounted, 350-351
 - simple undiscounted, 349-350
- percentile, 239, 259
- periods for investment goals, 331-332
- PivotTable subtotals, 382-383
- polynomial regression values, 301-302
- power trend values, 298-299
- present value, 339-340
 - inflation, 340-341
 - with PV() function, 341-344
- preventing calculation errors, 190-191
- principal and interest, 309-310
- probability, 253-255
- range, 240
- rank, 239
- return on fixed assets, 217-218
- sales per employee, 215
- sales to accounts payable, 224
- sales to current assets, 214
- sales to inventory, 220
- sales to short-term debt, 214
- selling price, 208
- shape of curve, 260-261
- standard deviation, 241-242
- standard scores, 258-259
- term of loan, 315-317
- tiered payments, 110
- value field summaries, 383
 - difference calculations, 383-385
 - index calculations, 389-391
 - percentage calculations, 385-387
 - running total calculations, 388-389
- variance, 240-241
- weighted questionnaire results, 139
- working capital, 224-225
- working capital turnover, 225
- calculation of formulas, controlling, 9-10
- carriage returns, removing with CLEAN() function, 90
- cash flows, discounting, 344-345
 - calculating net present value, 345-349
- cash-flow analysis, book publishing example, 354-358. *See also* discount formulas
 - net cash flows, 356-357
 - operating costs and sales, 355-356
 - per-unit constants, 355
- cash ratio, 227
- catalog price, 209
- categories, consolidating by, 38-39
- categorizing by values, 112-113
- causation, 252
- CEILING.MATH() function, 183, 188-189
- cell errors. *See* error cells
- CELL() function, 99-100, 125-127
- cell references. *See also* referencing
 - absolute format, 13
 - ignoring when applying range names, 18
 - changing format of, 14
 - circular, troubleshooting, 56-57
 - converting range names to, 52
 - external, 22
 - mixed format, 14
 - in range names, 13
 - relative format, 12-14
 - ignoring when applying range names, 18
 - troubleshooting, 53-54
- cell values, watching, 65-66
- cells
 - linking dialog box controls to, 43
 - padding, 92
- changing
 - cell reference format, 14
 - cells
 - in Goal Seek, 406
 - in scenarios, 414
 - in Solver, 422
 - link source, 23

changing

- value field summary calculations, 383
 - difference calculations, 383-385
 - index calculations, 389-391
 - percentage calculations, 385-387
 - running total calculations, 388-389
- CHAR() function, 79-83, 200
- character codes, 80-81
 - CHAR() function, 81-83
 - CODE() function, 83
- characters
 - counting with SUBSTITUTE() function, 102
 - removing extra, 89
 - with CLEAN() function, 90
 - with TRIM() function, 89-90
 - removing from strings, 101-102
 - repeating with REPT() function, 92-93
- charts, building text-based, 92-93
- check boxes, inserting, 46
- CHOOSE() function, 136-137
 - day of week, determining, 137
 - disadvantages of, 140
 - month of fiscal year, determining, 138-139
 - option buttons, integration with, 139-140
 - weighted questionnaire results, calculating, 139
- choosing error actions, 58-59
- circular references, 32-34
 - troubleshooting, 56-57
- CLEAN() function, 79, 90, 103
- client workbooks, 21
- CODE() function, 79, 83
- coefficient of determination (R²), 269-270
- COGS (cost of goods sold), 215
- COLUMN() function, 31, 99-100
- column letters, extracting, 99-100
- column names in range names, 18-20
- columns
 - creating ledger shading, 198
 - lookup column, specifying, 148-149
- COLUMNS() function, 31
- COMBIN() function, 184
- COMBINA() function, 184
- combo boxes
 - inserting, 46-47
 - looking up values, 148
- commas (,), arguments and, 72
- comparing scenarios, 417-419
- comparison formulas, operators for, 6
- complex criteria, filtering with, 366
 - entering compound criteria, 368-369
 - filtering with criteria range, 367-368
 - setting up criteria range, 366-367
- compound computed criteria, 369
- compound criteria, entering, 368-369
- compound interest, 325
- computed criteria, 366
 - filtering with, 369-370
- CONCAT() function, 80
- concatenating text with delimiter, 91
- conditional formatting, 114-115
 - creating ledger shading, 197
- conditions
 - applying across ranges, 116
 - operations on cells meeting, 117-118
- confidence intervals, calculating, 261-262
- confidence levels, 262
- CONFIDENCE.NORM() function, 262
- CONFIDENCE.T() function, 262
- consolidating multisheet data, 34-35
 - by category, 38-39
 - by position, 35-38
- constants, array, 30-31
- constraint cells in Solver, 422
- constraints
 - adding in Solver, 426-428
 - non-negative constraints in Solver, 428
- continuous compounding, 329
- continuous random variables, 246
- contribution margin, 211

- controls. *See* dialog box controls
- converting
 - between nominal/effective interest rates, 327, 335
 - dates with text functions, 95
 - degrees/radians, 185
 - formulas
 - to text, 49
 - to values, 15-16
 - range names to cell references, 52
 - strings
 - to dates, 158
 - to times, 174
 - text, 84
 - with LOWER() function, 84
 - with NUMBERVALUE() function, 85
 - with PROPER() function, 84
 - to sentence case, 95
 - with UPPER() function, 84
- copying formulas, 11-12
 - with absolute references, 13
 - with mixed references, 14
 - with relative references, 12-14
- CORREL() function, 251-252, 270-271
- correlation, calculating, 251-252
- correspondences in array formulas, 28
- COS() function, 186
- COSH() function, 186
- cost of goods sold (COGS), 215
- cost price, 209
- costs, types of, 211
- COT() function, 186
- COTH() function, 186
- COUNT() function, 230
- COUNTA() function, 231
- COUNTBLANK() function, 131, 231
- COUNTIF() function, 232, 254
- COUNTIFS() function, 232
- counting
 - blanks in ranges, 131-132
 - characters with SUBSTITUTE() function, 102
 - errors in ranges, 132
 - items, 230
 - with COUNT() function, 230
 - with COUNTA() function, 231
 - with COUNTBLANK() function, 231
 - with COUNTIF() function, 232
 - with COUNTIFS() function, 232
 - non-numeric values in ranges, 132
 - occurrences in ranges, 119-120
- covariance, calculating, 250-251
- COVARIANCE.P() function, 250-251
- COVARIANCE.S() function, 250-251
- creating
 - arrays, 25
 - custom functions, 70
 - custom PivotTable calculations, 391
 - calculated fields, 393-394
 - calculated items, 394-395
 - restrictions on, 392
- criteria range
 - filtering with, 367-368
 - setting up, 366-367
- CSC() function, 186
- CSCH() function, 186
- CUMIPMT() function, 311-312
- CUMPRINC() function, 311-312
- cumulative principal and interest, calculating, 311-312
- cumulative totals, calculating, 194-195
- current assets, 224
 - sales to, 214
- current date, returning, 157
- current liabilities, 224
- current ratio, 226
- current time, returning, 173
- custom functions, creating, 70
- customizing PivotTable subtotal calculations, 382-383

custom PivotTable calculations

- custom PivotTable calculations, creating, 391
 - calculated fields, 393-394
 - calculated items, 394-395
 - restrictions on, 392

D

- data analysis tools. *See* Goal Seek; scenarios; what-if analysis
- data bars, 94
- data tables
 - adding formulas to, 402-403
 - editing, 405
 - one-input setup, 400-402
 - recalculating, 10, 404
 - two-input setup, 403-405
- data types, referencing, 51, 54
- data validation, 40-42
- DATE() function, 95, 156, 158, 161
 - returning birthday, 164
 - returning future dates, 161
 - returning last day of month, 164
- date functions
 - DATE(), 158, 161, 164
 - DATEVALUE(), 158
 - DAY(), 159
 - DAYS(), 170
 - DAYS360(), 171-172
 - EDATE(), 163
 - EOMONTH(), 164
 - ISOWEEKNUM(), 160
 - list of, 156-157
 - MONTH(), 159
 - NETWORKDAYS(), 170-171
 - NETWORKDAYS.INTL(), 171
 - TODAY(), 157
 - WEEKDAY(), 137, 160, 165-167
 - WEEKNUM(), 160
 - WORKDAY(), 161
 - WORKDAY.INTL(), 162-163
 - YEAR(), 159
 - YEARFRAC(), 172
- dates
 - adding months to, 163
 - with EDATE() function, 163
 - with EOMONTH() function, 164
 - calculating difference between, 169-170
 - with DAYS() function, 170
 - with DAYS360() function, 171-172
 - with NETWORKDAYS() function, 170-171
 - with YEARFRAC() function, 172
 - calculating
 - Easter Sunday, 189
 - holidays, 167-168
 - Julian dates, 169
 - converting
 - with text functions, 95
 - strings to, 158
 - entering, 154-155
 - formatting, 88, 154-155
 - returning, 157
 - with DATE() function, 158
 - with DATEVALUE() function, 158
 - future dates, 161
 - with TODAY() function, 157
 - returning parts of, 159
 - with DAY() function, 159
 - with ISOWEEKNUM() function, 160
 - with MONTH() function, 159
 - with WEEKDAY() function, 160
 - with WEEKNUM() function, 160
 - with WORKDAY() function, 161
 - with WORKDAY.INTL() function, 162-163
 - with YEAR() function, 159
 - serial numbers for, 153-154
 - two-digit years, 155-156
- DATEVALUE() function, 156, 158
- DAVERAGE() function, 376-377
- DAY() function, 156, 159

- day of week, determining, 137
- DAYS() function, 156, 170
- DAYS360() function, 156, 171-172
- days purchases in accounts payable, 223-224
- DCOUNT() function, 376
- DCOUNTA() function, 376
- DECIMAL() function, 184
- degree symbol, inserting, 89
- degrees, converting to/from radians, 185
- DEGREES() function, 185-186
- deleting. *See* removing
- delimiters, concatenating text with, 91
- dependents
 - defined, 62
 - tracing, 63
- dependent variables, 246, 265
- dependent workbooks, 21
- deposits, calculating
 - future value, 329
 - for investment goals, 332-333
- descriptive statistics, 229-230
 - averages, 232
 - with AVERAGE() function, 233
 - with AVERAGEA() function, 233
 - with AVERAGEIF() function, 233
 - with AVERAGEIFS() function, 234
 - with MEDIAN() function, 234
 - with MODE() function, 234
 - with SUMPRODUCT() function, 235-236
 - counting items, 230
 - with COUNT() function, 230
 - with COUNTA() function, 231
 - with COUNTBLANK() function, 231
 - with COUNTIF() function, 232
 - with COUNTIFS() function, 232
 - extreme values, 236
 - with LARGE() and SMALL() functions, 237-238
 - with MAX() and MIN() functions, 236
 - frequency distributions, 242-244
 - percentile, 239
 - range, 240
 - rank, 239
 - standard deviation, 241-242
 - variance, 240-241
 - deseasoning data, 286
 - destination ranges, 35
 - Developer tab, displaying, 42
 - DGET() function, 376-378
 - dialog box controls
 - check boxes, 46
 - combo boxes, 46-47
 - group boxes, 44
 - inserting, 43
 - linking to cells, 43
 - list boxes, 46-47
 - option buttons, 45
 - scroll bars, 47-48
 - spin boxes, 47-48
 - difference between dates, calculating, 169-170
 - with DAYS() function, 170
 - with DAYS360() function, 171-172
 - with NETWORKDAYS() function, 170-171
 - with YEARFRAC() function, 172
 - differences between times, calculating, 176-177
 - with MOD() function, 195
 - difference summary calculations, 383-385
 - direct costs, 215
 - direct relationships, 250
 - disabling data table recalculation, 404
 - discontinuities, 429
 - discount amount, 210
 - discounted payback period, calculating, 351
 - discount factor, 340
 - discount formulas. *See also* cash-flow analysis
 - discounting cash flows, 344-345
 - net present value, 345-349

discount formulas

- internal rate of return, 352
 - with IRR() function, 352-353
 - with MIRR() function, 353-354
 - with XIRR() function, 353
- payback period, 349
 - discounted, 351
 - exact undiscounted, 350-351
 - simple undiscounted, 349-350
- present value, 339-340
 - inflation, 340-341
 - with PV() function, 341-344
- relationship to present value/future value/
interest earned, 339
- discount functions
 - IRR(), 352-353
 - MIRR(), 353-354
 - NPV(), 346-348
 - PV(), 341-344
 - XIRR(), 353
 - XNPV(), 348-349
- discount rate, 210-211
 - returning, 142-143
- discounts (on price), 209
 - discount amount, 210
 - discount rate, 210-211
 - list price, 211
 - net price, 209-210
- discrete probability distributions
 - binomial, 255
 - hypergeometric, 256
 - Poisson, 256-257
- discrete random variables, 246
- displaying
 - Developer tab, 42
 - formulas, 15
 - last updated date, 89
 - scenarios, 415-416
- #DIV/0! error value, 50
- dividing by zero errors, 50
 - avoiding, 109
- DMAX() function, 376
- DMIN() function, 376
- DOLLAR() function, 80, 86
- dollar sign (\$), in cell references, 13
- dot follower effect, 92
- DPRODUCT() function, 376
- drop-down lists, in-cell lists and exact-match
lookups, 145-146
- DSTDEV() function, 376
- DSTDEVP() function, 376
- DSUM() function, 375-376
- due dates, avoiding weekends, 122-123
- DVAR() function, 376
- DVARP() function, 376
- dynamic amortization schedules, building,
313-315
- dynamic array formulas, 27-28
- dynamic array functions, list of, 32

E

- Easter Sunday, calculating, 168, 189
- EDATE() function, 156, 163
- editing
 - constraints, 427
 - data tables, 405
 - formulas, 5
 - scenarios, 416-417
- Edit mode, 5
- EFFECT() function, 327, 335
- effective interest, 326-327
 - converting to/from nominal interest, 327,
335
- employee time sheets, building, 177-181
- employees, sales per, 215
- empty cells, counting, 231
- engines in Solver, 429

- entering
 - array formulas, 26-27
 - compound criteria, 368-369
 - computed criteria, 369-370
 - dates/times, 154-155
 - dynamic array formulas, 27-28
 - formulas, 4-5
 - functions into formulas, 73-74
 - table formulas, 373-374
 - table functions, 376
 - Enter mode, 4
 - EOMONTH() function, 156, 164
 - error actions, choosing, 58-59
 - error cells
 - defined, 62
 - tracing, 64
 - error checker (for formulas), 58
 - choosing error action, 58-59
 - setting options, 59-62
 - error indicator (green triangle), 58
 - error messages for parentheses in formulas, 9
 - ERROR.TYPE() function, 125, 128-129
 - error values, 50
 - #CALC!, 50
 - #DIV/0!, 50
 - #FIELD!, 51
 - #N/A!, 51, 58
 - #NAME?, 51-53
 - #NULL!, 53
 - #NUM!, 53
 - #REF!, 53-54
 - #SPILL!, 54
 - #UNKNOWN!, 54
 - #VALUE!, 54
 - errors. *See also* troubleshooting formulas
 - counting in ranges, 132
 - ignoring in ranges, 132-133
 - preventing with rounding, 190-191
 - evaluating formulas, 64-65
 - EVEN() function, 184, 189
 - even rows, adding, 196
 - Evolutionary engine, 429, 432
 - EXACT() function, 80
 - exact-match lookups, 144-146
 - exact undiscounted payback point, calculating, 350-351
 - EXP() function, 184, 288, 299
 - exponential trends, 288-289
 - calculating values, 290-291
 - with GROWTH() function, 291-292
 - with LOGEST() function, 292-293
 - plotting, 289-290
 - exponentiation operator, 5
 - extending forecasts
 - with fill handle, 276
 - with Series command, 276-277
 - external references, 22
 - extracting
 - column letters, 99-100
 - names, 97-99
 - periodic samples, 247-248
 - random samples, 249
 - substrings, 94
 - with LEFT() function, 94
 - with MID() function, 94
 - with RIGHT() function, 94
 - extreme values, calculating, 236
 - with LARGE() and SMALL() functions, 237-238
 - with MAX() and MIN() functions, 236
- ## F
- FACT() function, 184
 - FACTDOUBLE() function, 184
 - FALSE results, handling in IF() function, 108
 - FALSE() function, 106
 - feasible solutions, 431

#FIELD! error value

- #FIELD! error value, 51
- field names, 361
- field values, 361
- fields
 - defined, 361
 - hiding, 362
 - partial fields, sorting on, 362-363
- FIELDVALUE() function, 136
- fill handle, extending forecasts, 276
- FILTER() function, 32, 370
- filtering tables, 365-366
 - with complex criteria, 366-369
 - with computed criteria, 369-370
 - with FILTER() function, 370
 - with UNIQUE() function, 371
- financial formulas, 213. *See also* business formulas; discount formulas; investment formulas; loan formulas
 - compound interest, 325
 - continuous compounding, 329
 - converting between nominal/effective interest rates, 327, 335
 - cost of goods sold, 215
 - effective interest, 326-327
 - fixed-asset ratios, 217
 - fixed assets to short-term debt, 218
 - fixed-asset turnover, 217
 - return on fixed assets, 217-218
 - gross margin, 216
 - net margin, 216
 - nominal interest, 326-327
 - sales ratios, 214
 - sales per employee, 215
 - sales to current assets, 214
 - sales to short-term debt, 214
 - simple interest, 325
 - time value of money, 305-306
- FIND() function, 80, 97-99
- finding
 - leap years, 196-197
 - row numbers of list values, 120-121
- fiscal year, determining month of, 138-139
- FISHER() function, 229
- fixed assets, 217
- fixed assets to short-term debt, 218
- fixed-asset ratios, 217
 - fixed assets to short-term debt, 218
 - fixed-asset turnover, 217
 - return on fixed assets, 217-218
- fixed-asset turnover, 217
- fixed costs, 211
- FIXED() function, 80, 87
- fixed-rate amortization schedules, building, 312-313
- FLOOR.MATH() function, 184, 188-189
- FORECAST() function, 279
- forecasts, 274. *See also* exponential trends; logarithmic trends; power trends; regression analysis
 - extending
 - with fill handle, 276
 - with Series command, 276-277
 - plotting, 275
 - regression equation for, 277-278
 - seasonal sales model case study, 280-288
 - with FORECAST() function, 279
 - with LINEST() function, 279
 - with TREND() function, 278
- Forecast Sheet, 267
- formatting
 - conditional formatting, 114-115
 - creating ledger shading, 197
 - dates/times, 154-155
 - numbers, restoring formatting, 388
 - text, 85
 - with DOLLAR() function, 86
 - with FIXED() function, 87
 - with TEXT() function, 87-89
- form controls
 - check boxes, 46
 - combo boxes, 46-47

- group boxes, 44
- inserting, 43
- linking to cells, 43
- list boxes, 46-47
- option buttons, 45
- scroll bars, 47-48
- spin boxes, 47-48
- formulas
 - adding to data tables, 402-403
 - arithmetic, operators for, 5-6
 - array
 - array constants in, 30-31
 - correspondences in, 28
 - defined, 25
 - dynamic, 27-28
 - entering, 26-27
 - on multiple ranges, 29
 - circular references, 32-34
 - comparison, operators for, 6
 - computed criteria, entering, 369-370
 - conditional formatting with, 114-115
 - converting
 - to text, 49
 - to values, 15-16
 - copying/moving, 11-12
 - with absolute references, 13
 - with mixed references, 14
 - with relative references, 12-14
 - discount
 - calculating internal rate of return, 352-354
 - calculating payback period, 349-351
 - calculating present value, 339-344
 - discounting cash flows, 344-349
 - relationship to present value/future value/interest earned, 339
 - displaying, 15
 - editing, 5
 - entering, 4-5
 - evaluating, 64-65
 - financial, 213
 - compound interest, 325
 - continuous compounding, 329
 - converting between nominal/effective interest rates, 327, 335
 - cost of goods sold, 215
 - effective interest, 326-327
 - fixed-asset ratios, 217-218
 - gross margin, 216
 - net margin, 216
 - nominal interest, 326-327
 - sales ratios, 214-215
 - simple interest, 325
 - time value of money, 305-306
 - inventory, 218
 - inventory management, 220-221
 - inventory ratios, 218-220
 - investment
 - building investment schedule, 335-337
 - calculating future value, 328-330
 - investment goals, 330-334
 - iteration, 32-34
 - limits on, 4
 - line breaks in, 3
 - links in, 21
 - changing source, 23
 - external references, 22
 - updating, 22-23
 - liquidity, 221
 - accounts payable ratios, 223-224
 - accounts receivable ratios, 222-223
 - liquidity index, 227-228
 - liquidity ratios, 226-227
 - working capital, 224-225
 - loan
 - allowing principal paydowns, 321-323
 - borrowed amount, 318-319
 - building loan amortization schedules, 312-315, 320-321

formulas

- cumulative principal and interest, 311-312
- interest costs, 309-310
- interest rate, 317-318
- loan payments, 306-312
- principal and interest, 309-310
- term of loan, 315-317
- naming, 20
- nesting, limits on, 4
- operator precedence, 7-9
- PivotTable results in, 395-397
- pricing, 207
 - break-even point, 211-213
 - discounts, 209-211
 - markups, 207-209
- range names in, 16
 - applying new, 17-20
 - pasting, 16-17
- recalculation, controlling, 9-10
- reference, operators for, 6
- referencing tables in, 371
 - entering table formulas, 373-374
 - table specifiers, 371-373
- structure of, 3
- table, entering, 373-374
- in table records, sorting, 362
- text, operators for, 6
- troubleshooting. *See* troubleshooting
 - formulas
 - typing functions into, 73-74
- FORMULATEXT() function, 8, 15
- fractional powers, raising numbers to, 9
- frequency distributions, calculating, 242-244
- FREQUENCY() function, 243-244, 253
- front loading interest costs, 309
- functions
 - ADDRESS(), 122
 - in Analysis TookPak, 76
 - AutoComplete feature, 73-74
 - COLUMNS(), 31
 - creating custom, 70
 - date functions
 - DATE(), 95, 158, 161, 164
 - DATEVALUE(), 158
 - DAY(), 159
 - DAYS(), 170
 - DAYS360(), 171-172
 - EDATE(), 163
 - EOMONTH(), 164
 - ISOWEEKNUM(), 160
 - list of, 156-157
 - MONTH(), 159
 - NETWORKDAYS(), 124, 170-171
 - NETWORKDAYS.INTL(), 171
 - TODAY(), 157
 - WEEKDAY(), 122, 137, 160, 165-167
 - WEEKNUM(), 160
 - WORKDAY(), 123, 161
 - WORKDAY.INTL(), 162-163
 - YEAR(), 159
 - YEARFRAC(), 172
 - defined, 70
 - discount functions
 - IRR(), 352-353
 - MIRR(), 353-354
 - NPV(), 346-348
 - PV(), 341-344
 - XIRR(), 353
 - XNPV(), 348-349
 - FORMULATEXT(), 8, 15
 - IFERROR(), 57-58
 - IFNA(), 58, 143
 - information functions
 - CELL(), 125-127
 - ERROR.TYPE(), 128-129
 - INFO(), 70, 129-130
 - ISxxx(), 130, 132-133
 - list of, 125
 - SHEET(), 130
 - SHEETS(), 130
 - Insert Function feature, 74-76

- investment functions
 - EFFECT(), 327, 335
 - FV(), 71-72, 328-330
 - FVSCHEDULE(), 334
 - NOMINAL(), 327, 335
 - NPER(), 331-332
 - PMT(), 332-333
 - PV(), 333
 - RATE(), 330-331
- loan functions
 - CUMIPMT(), 311-312
 - CUMPRINC(), 311-312
 - FV(), 317
 - INDIRECT(), 310
 - IPMT(), 309-310
 - NPER(), 315-316
 - PMT(), 70, 307-308
 - PPMT(), 309-310
 - PV(), 318-319
 - RATE(), 317-318
- logical functions
 - AND(), 111-113, 124
 - IF(), 50, 70, 106-124, 143
 - IFS(), 110-111
 - list of, 106
 - OR(), 113-114
 - XOR(), 114
- lookup functions, 135
 - CHOOSE(), 137-140
 - HLOOKUP(), 31, 141-146
 - INDEX(), 31, 47, 147-151
 - list of, 136
 - LOOKUP(), 31
 - MATCH(), 31, 146-151
 - VLOOKUP(), 32, 140-146
- math functions
 - CEILING.MATH(), 188-189
 - EVEN(), 189
 - FLOOR.MATH(), 188-189
 - INT(), 190-191
 - list of, 183-185
 - MOD(), 195-198
 - MROUND(), 187-188, 191-192
 - ODD(), 189
 - RAND(), 199-201
 - RANDARRAY(), 32, 202
 - RANDBETWEEN(), 201, 249
 - ROUND(), 186-187
 - ROUNDDOWN(), 188
 - ROUNDUP(), 188
 - SEQUENCE(), 32, 202-203
 - SUM(), 70, 192, 194-195
 - SUMIF(), 193
 - SUMIFS(), 193-194
 - TRUNC(), 190
- MDETERM(), 31
- MINVERSE(), 31
- MMULT(), 31
- names, troubleshooting, 51-52
- nesting, 73
- regression analysis
 - EXP(), 288-299
 - FORECAST(), 279
 - GROWTH(), 31, 291-292
 - INTERCEPT(), 273
 - LINEST(), 31, 271-273, 279, 295, 302, 304
 - LN(), 293-295, 298
 - LOGEST(), 31, 292-293
 - RSQ(), 273
 - SLOPE(), 273
 - TREND(), 32, 270-271, 278, 301-304
- ROW(), 31, 82, 121
- ROWS(), 31
- SINGLE(), 32
- statistical functions
 - AVERAGE(), 233
 - AVERAGEA(), 233
 - AVERAGEIF(), 233
 - AVERAGEIFS(), 234
 - BINOM.DIST(), 255
 - CONFIDENCE.NORM(), 262
 - CONFIDENCE.T(), 262

functions

CORREL(), 251-252, 270-271
COUNT(), 230
COUNTA(), 231
COUNTBLANK(), 131, 231
COUNTIF(), 232, 254
COUNTIFS(), 232
COVARIANCE.P(), 250-251
COVARIANCE.S(), 250-251
FISHER(), 229
FREQUENCY(), 243-244, 253
HYPGEOM.DIST(), 256
KURT(), 261
LARGE(), 237-238
MAX(), 115, 121, 236, 240
MAXA(), 237
MEDIAN(), 234
MIN(), 115, 121, 236, 240
MINA(), 237
MODE(), 234
MODE.MULT(), 235
MODE.SNGL(), 235
NORM.DIST(), 259
OFFSET(), 246-249
PEARSON(), 229
PERCENTILE.EXC(), 239
PERCENTILE.INC(), 239
PHI(), 229
POISSON.DIST(), 256
PROB(), 254
RANK.AVG(), 239
RANK.EQ(), 239
SKEW(), 260
SMALL(), 237-238
STDEVA(), 242
STDEV.P(), 242
STDEVPA(), 242
STDEV.S(), 242
SUMPRODUCT(), 31, 235-236
T.TEST(), 264
VARA(), 241
VAR.P(), 241
VARPA(), 241
VAR.S(), 241
Z.TEST(), 263

structure of, 70-72

summary functions

- changing, 383
- difference calculations, 383-385
- index calculations, 389-391
- list of, 383
- percentage calculations, 385-387
- running total calculations, 388-389

table functions

- advantages of, 375
- analyzing table of defects, 378-379
- DAVERAGE(), 377
- DGET(), 377-378
- DSUM(), 375
- entering, 376
- FILTER(), 32, 370
- GETPIVOTDATA(), 391, 395-397
- list of, 376
- SORT(), 32, 364
- SORTBY(), 32, 364-365
- syntax of, 375-376
- TABLE(), 405
- UNIQUE(), 32, 371

text functions

- CELL(), 99-100
- CHAR(), 81-83, 200
- CLEAN(), 90, 103
- CODE(), 83
- COLUMN(), 31, 99-100
- DOLLAR(), 86
- FIND(), 97-99
- FIXED(), 87
- LEFT(), 94
- LEN(), 90
- list of, 79-80
- LOWER(), 84
- MID(), 94
- NUMBERVALUE(), 85
- PROPER(), 84

REPLACE(), 100-101
 REPT(), 92-93
 RIGHT(), 94
 SEARCH(), 97-99
 SUBSTITUTE(), 101-103
 TEXT(), 87-89
 TEXTJOIN(), 91
 TRIM(), 89-90
 UNICHAR(), 81
 UPPER(), 84
 time functions
 HOUR(), 174
 list of, 173
 MINUTE(), 175
 NOW(), 71, 89, 173
 SECOND(), 175
 TIME(), 174-176
 TIMEVALUE(), 174
 TRANSPOSE(), 32
 trigonometric functions, list of, 185-186
 types of, 69
 typing into formulas, 73-74
 using/returning arrays, 31-32
 future dates, returning, 161
 future times, returning, 175-176
 future value, 306
 calculating, 328-330
 for investment goals, 334
 relationship to discounting, 339
 FV() function, 71-72, 317, 328-330
 FVSCHEDULE() function, 334

G

GCD() function, 184
 generating
 account numbers, 96, 102-103
 ANSI character set, 81-82
 random letters, 200

random numbers, 198
 with RAND() function, 199-201
 with RANDARRAY() function, 202
 with RANDBETWEEN() function, 201
 with SEQUENCE() function, 202-203
 series of letters, 82-83
 Solver reports, 437
 Answer report, 437-439
 Limits report, 441
 Sensitivity report, 439-440
 summary reports for scenarios, 417-419
 GETPIVOTDATA() function, 136, 391, 395-397
 Goal Seek, 405
 approximations in, 409-410
 break-even analysis, 410-411
 iterative methods, 406
 optimizing product margin, 408-409
 pausing, 407
 resuming, 407
 solving algebraic equations, 411-412
 steps in, 407-408
 worksheet setup, 406
 Good Friday, calculating, 168
 grand totals
 of calculated fields, 394
 in PivotTables, 381
 hiding, 382
 GRG Nonlinear engine, 429, 431-432
 gross margin, 216
 group boxes, inserting, 44
 GROWTH() function, 31, 291-292

H

hiding
 fields, 362
 PivotTable grand totals, 382
 PivotTable subtotals, 382
 histograms, 93

HLOOKUP() function

HLOOKUP() function, 31, 136, 141-142
 exact-match lookups, 144-146
 range lookups, 142-144
holidays, calculating, 167-168, 189
HOUR() function, 173-174
hurdle rate, 346
HYPGEOM.DIST() function, 256
hypergeometric distributions, 256
hypothesis testing, 263-264

I

IF() function, 70, 106, 143
 aging overdue invoices, 123-124
 applying conditions across ranges, 116
 avoiding
 #DIV/0 errors, 50
 division by zero, 109
 weekend due dates, 122-123
 combining with arrays, 116-121
 conditional formatting with, 114-115
 counting occurrences in ranges, 119-120
 finding row numbers in lists, 120-121
 handling FALSE results, 108
 multiple logical tests with, 109-115
 nesting, 109-110
 operations on cells that meet condition,
 117-118
 simple version, 106-107
IFERROR() function, 57-58, 106
IFNA() function, 58, 106, 143
IFS() function, 106, 110-111
ignoring errors in ranges, 132-133
in-cell drop-down lists, exact-match lookups
 and, 145-146
income investing, rental properties versus,
 342-343
incorrect formula results, troubleshooting, 56
independent variables, 246, 265
index summary calculations, 389-391
INDEX() function, 31, 47, 136, 147
 list boxes, looking up values, 148
 lookup column, specifying, 148-149
 multiple-column lookups, 150-151
 row-and-column lookups, 149-150
INDIRECT() function, 310
inferential statistics, 245-246
 confidence intervals, 261-262
 correlation, 251-252
 covariance, 250-251
 hypothesis testing, 263-264
 normal distribution, 257-258
 percentiles, 259
 shape of curve, 260-261
 standard scores, 258-259
 probability distributions, 252
 binomial, 255
 calculating, 253-255
 hypergeometric, 256
 Poisson, 256-257
 sampling data, 246-247
 periodic samples, 247-248
 random samples, 249
inflation, 340-341
INFO() function, 70, 125, 129-130
information functions
 CELL(), 125-127
 ERROR.TYPE(), 128-129
 INFO(), 129-130
 ISxxx(), 130, 132-133
 list of, 125
 SHEET(), 130
 SHEETS(), 130
initial deposit, calculating for investment goals,
 333
input modes, 4-5
Insert Function feature, 74-76
inserting
 degree symbol, 89
 dialog box controls, 43
 check boxes, 46
 combo boxes, 46-47

- group boxes, 44
 - list boxes, 46-47
 - option buttons, 45
 - scroll bars, 47-48
 - spin boxes, 47-48
- functions with Insert Function feature, 74-76
- integrating CHOOSE() function with option buttons, 139-140
- intelligent, defined, 105
- INTERCEPT() function, 273
- interest costs
 - calculating, 309-310
 - front loading, 309
- interest earned, relationship to discounting, 339
- interest rates
 - calculating, 317-318
 - for investment goals, 330-331
 - compound interest, 325
 - converting between nominal/effective rates, 327, 335
 - effective interest, 326-327
 - nominal interest, 326-327
 - simple interest, 325
- internal rate of return, calculating, 352
 - with IRR() function, 352-353
 - with MIRR() function, 353-354
 - with XIRR() function, 353
- INT() function, 184, 190-191
- inventory, 218
- inventory formulas, 218
 - inventory management, 220
 - inventory reorder level, 221
 - inventory safety stock, 220-221
 - inventory ratios, 218
 - inventory turnover, 219
 - inventory turnover rate, 219
 - sales to inventory, 220
- inventory management formulas, 220
 - inventory reorder level, 221
 - inventory safety stock, 220-221
- inventory ratios, 218
 - inventory turnover, 219
 - inventory turnover rate, 219
 - sales to inventory, 220
- inventory reorder level, 221
- inventory safety stock, 220-221
- inventory turnover, 219
- inventory turnover rate, 219
- inverse relationships, 250
- investment formulas. *See also* discount formulas
 - building investment schedule, 335-337
 - calculating future value, 328-330
 - investment goals, 330
 - deposits, 332-333
 - future value with varying interest rates, 334
 - initial deposit, 333
 - interest rate, 330-331
 - periods, 331-332
- investment functions
 - EFFECT(), 327, 335
 - FV(), 328-330
 - FVSCHEDULE(), 334
 - NOMINAL(), 327, 335
 - NPER(), 331-332
 - PMT(), 332-333
 - PV(), 333
 - RATE(), 330-331
- investment goals, 330
 - deposits, 332-333
 - future value with varying interest rates, 334
 - initial deposit, 333
 - interest rate, 330-331
 - periods, 331-332
- investment schedules, building, 335-337
- IPMT() function, 309-310

IRR() function

- IRR() function, 352-353
- ISBLANK() function, 125, 131-132
- ISERR() function, 125
- ISERROR() function, 125, 132-133
- ISEVEN() function, 125
- ISFORMULA() function, 125
- ISLOGICAL() function, 125
- ISNA() function, 125
- ISNONTEXT() function, 125
- ISNUMBER() function, 125, 132
- ISODD() function, 125
- ISOWEEKNUM() function, 156, 160
- ISREF() function, 125
- ISTEXT() function, 125
- ISxxx() functions, 130, 132-133
- iteration formulas, 32-34
- iterative methods
 - in Goal Seek, 406
 - in Solver, 422

J-K-L

- Julian dates, calculating, 169
- KURT() function, 261
- kurtosis, 261
- LARGE() function, 237-238
- last day of month, returning
 - with DATE() function, 164
 - with EOMONTH() function, 164
- last updated date, displaying, 89
- LCM() function, 184
- leap years, finding, 196-197
- leasing equipment, buying versus, 343-344
- ledger shading, creating, 197-198
- LEFT() function, 80, 94
- LEN() function, 80, 90

- letters, generating
 - random, 200
 - series of, 82-83
- limits on formulas, 4
- Limits report (Solver), 441
- linear data, simple regression on, 266
 - best-fit trendlines, 266-274
 - forecasts, 274-279
 - seasonal sales model case study, 280-288
- linear models, 429
- linear programming, 421
- line breaks in formulas, 3
- line feeds, removing
 - with CLEAN() function, 90
 - with SUBSTITUTE() function, 103
- line of best fit, 266
 - analyzing sales versus advertising, 274
 - calculating
 - with LINEST() function, 271-273
 - with TREND() function, 270-271
 - plotting, 267-268
 - R² for, 269-270
 - regression equation for, 268-269
- LINEST() function, 31, 271-273, 279, 295, 302, 304
- linked data types, referencing, 51
- linking
 - dialog box controls to cells, 43
 - workbooks, 21
- links in formulas, 21
 - changing source, 23
 - external references, 22
 - updating, 22-23
- liquid assets, 221
- liquidity, 221
- liquidity formulas, 221
 - accounts payable ratios, 223
 - accounts payable turnover, 223
 - days purchases in accounts payable, 223-224
 - sales to accounts payable, 224

- accounts receivable ratios, 222
 - accounts receivable turnover, 222
 - average AR duration, 222-223
- liquidity index, 227-228
- liquidity ratios, 226
 - acid-test ratio, 226-227
 - cash ratio, 227
 - current ratio, 226
 - working capital, 224-225
- liquidity index, 227-228
- liquidity ratios, 226
 - acid-test ratio, 226-227
 - cash ratio, 227
 - current ratio, 226
- list boxes
 - inserting, 46-47
 - looking up values, 148
- list price, 209, 211
- list values, looking up, 118-119
- lists, finding row numbers of values, 120-121
- LN() function, 184, 293-295, 298
- loading
 - Analysis TookPak, 76-77
 - Solver, 423
 - Solver models, 433
- loan amortization schedules, building, 312-315, 320-321
- loan analysis, 307-308
- loan formulas
 - allowing principal paydowns, 321-323
 - borrowed amount, 318-319
 - building loan amortization schedules, 312-315, 320-321
 - cumulative principal and interest, 311-312
 - interest costs, 309-310
 - interest rate, 317-318
 - loan payments, 306-312
 - principal and interest, 309-310
 - term of loan, 315-317
- loan functions
 - CUMIPMT(), 311-312
 - CUMPRINC(), 311-312
 - FV(), 317
 - INDIRECT(), 310
 - IPMT(), 309-310
 - NPER(), 315-316
 - PMT(), 307-308
 - PPMT(), 309-310
 - PV(), 318-319
 - RATE(), 317-318
- loan payments, calculating, 306-312
- LOG() function, 184
- LOG10() function, 184
- logarithmic trends, 293-294
 - calculating values, 295-296
 - plotting, 294-295
- LOGEST() function, 31, 292-293
- logical functions
 - AND(), 111-113, 124
 - IF(), 106-124
 - IFS(), 110-111
 - list of, 106
 - OR(), 113-114
 - XOR(), 114
- lookup column, specifying, 148-149
- lookup functions, 135
 - CHOOSE(), 137-140
 - HLOOKUP(), 141-146
 - INDEX(), 147-151
 - list of, 136
 - MATCH(), 146-151
 - VLOOKUP(), 140-146
- lookup tables, 136-137
- LOOKUP() function, 31, 136
- LOWER() function, 80, 84
- lump sum future value, calculating, 328

M

- manual recalculation, 10
- manufacturer's suggested retail price (MSRP), 209
- manufacturing contribution margin, 212
- markup amount, 208
- markup rate, 208-209
- markups, 207
 - cost price, 209
 - markup amount, 208
 - markup rate, 208-209
 - selling price, 208
- MATCH() function, 31, 136, 146-147
 - list boxes, looking up values, 148
 - lookup column, specifying, 148-149
 - multiple-column lookups, 150-151
 - row-and-column lookups, 149-150
- math functions
 - CEILING.MATH(), 188-189
 - EVEN(), 189
 - FLOOR.MATH(), 188-189
 - INT(), 190-191
 - list of, 183-185
 - MOD(), 195-198
 - MROUND(), 187-188, 191-192
 - ODD(), 189
 - RAND(), 199-201
 - RANDARRAY(), 202
 - RANDBETWEEN(), 201
 - ROUND(), 186-187
 - ROUNDDOWN(), 188
 - ROUNDUP(), 188
 - SEQUENCE(), 202-203
 - SUM(), 192, 194-195
 - SUMIF(), 193
 - SUMIFS(), 193-194
 - TRUNC(), 190
- MAX() function, 115, 121, 236, 240
- MAXA() function, 237
- MDETERM() function, 31, 184
- mean, calculating, 233-234
- measures of association, 249-250
- measures of central tendency, 240
- measures of dispersion, 240
- measures of location, 240
- measures of variation, 240
- median, calculating, 234
- MEDIAN() function, 234
- Memorial Day, calculating, 168
- merging scenarios, 417
- messages in Solver, 434-435
- MID() function, 80, 94
- MIN() function, 115, 121, 236, 240
- MINA() function, 237
- MINUTE() function, 173, 175
- MINVERSE() function, 31, 184
- MIRR() function, 353-354
- missing/mismatched parentheses, troubleshooting, 55
- mixed reference format, 14
- MMULT() function, 31, 184
- MOD() function, 184-195
 - adding every *n*th row, 196
 - calculating time differences, 195
 - creating ledger shading, 197-198
 - finding leap years, 196-197
- mode, calculating, 234
- MODE() function, 234
- MODE.MULT() function, 235
- MODE.SNGL() function, 235
- models (Solver), saving, 433-434
- money, time value of, 305-306
- month of fiscal year, determining, 138-139
- MONTH() function, 157, 159
- months
 - adding to dates, 163
 - with EDATE() function, 163
 - with EOMONTH() function, 164

- returning last day
 - with DATE() function, 164
 - with EOMONTH() function, 164
 - returning *n*th occurrence of weekday, 165-167
 - mortgages
 - principal paydowns, allowing, 321-323
 - variable-rate amortization schedules, building, 320-321
 - moving formulas, 11-12
 - with absolute references, 13
 - with mixed references, 14
 - with relative references, 12-14
 - MROUND() function, 184, 187-188, 191-192
 - MSRP (manufacturer's suggested retail price), 209
 - MULTINOMIAL() function, 184
 - multiple-column lookups, 150-151
 - multiple internal rates of return, calculating, 353-354
 - multiple logical tests, performing, 109-115
 - multiple ranges, array formulas on, 29
 - multiple regression, 266, 303-304
 - multisheet data, consolidating, 34-35
 - by category, 38-39
 - by position, 35-38
 - multithreaded calculation, 10
 - MUNIT() function, 184
- N**
- N() function, 125
 - #N/A! error value, 51, 58
 - NA() function, 125
 - #NAME? error value, 51-53
 - named ranges. *See* range names
 - names
 - extracting, 97-99
 - of functions, troubleshooting, 51-52
 - naming formulas, 20
 - negative values, adding in range, 193
 - nesting
 - formulas, limits on, 4
 - functions, 73
 - IF() functions, 109-110
 - parentheses, 9
 - SUBSTITUTE() functions, 102
 - net cash flows, 356-357
 - net margin, 216
 - net present value, calculating, 345-346
 - with NPV() function, 346-348
 - with XNPV() function, 348-349
 - net price, 209-210
 - NETWORKDAYS() function, 124, 157, 170-171
 - NETWORKDAYS.INTL() function, 157, 171
 - NOMINAL() function, 327, 335
 - nominal interest, 326-327
 - converting to/from effective interest, 327, 335
 - nonlinear data, 266
 - simple regression on, 288
 - exponential trends, 288-293
 - logarithmic trends, 293-296
 - power trends, 296-299
 - nonlinear programming, 421
 - non-negative constraints in Solver, 428
 - non-numeric values, counting in ranges, 132
 - nonperiodic cash flows
 - calculating internal rate of return, 353
 - net present value with, 348-349
 - normal distribution, 257-258
 - percentiles, 259
 - shape of curve, 260-261
 - standard scores, 258-259
 - NORM.DIST() function, 259
 - NOT() function, 106
 - NOW() function, 71, 89, 173
 - NPER() function, 315-316, 331-332
 - NPV() function, 346-348

#NULL! error value

- #NULL! error value, 53
- null hypothesis, 263
- #NUM! error value, 53, 353
- number formatting, restoring, 388
- numbers, formatting as text. *See* text, formatting
- NUMBERVALUE() function, 80, 85
- numeric formatting symbols, 87

O

- objective cells in Solver, 422
- ODD() function, 184, 189
- odd rows, adding, 196
- OFFSET() function, 246-247
 - extracting periodic samples, 247-248
 - extracting random samples, 249
- one-input data tables, setup, 400-402
- operating costs, 355-356
- operations research, 421
- operators
 - arithmetic, 5-6
 - comparison, 6
 - for constraints, 427
 - order of precedence, 7-9
 - reference, 6
 - text, 6
- optimizing product margin, 408-409
- option buttons
 - inserting, 45
 - integrating CHOOSE() function with, 139-140
- options for error checker, setting, 59-62
- order of precedence, 7-9
- OR() function, 106, 113-114
 - looking up list values, 118-119
- overdue invoices, aging, 123-124

P

- padding cells, 92
- parentheses ()
 - missing/mismatched, troubleshooting, 55
 - in order of precedence, 8-9
- partial fields, sorting on, 362-363
- parts of dates, returning, 159
 - with DAY() function, 159
 - with ISOWEEKNUM() function, 160
 - with MONTH() function, 159
 - with WEEKDAY() function, 160
 - with WEEKNUM() function, 160
 - with WORKDAY() function, 161
 - with WORKDAY.INTL() function, 162-163
 - with YEAR() function, 159
- parts of times, returning, 174
 - with HOUR() function, 174
 - with MINUTE() function, 175
 - with SECOND() function, 175
- pasting range names into formulas, 16-17
- pausing Goal Seek, 407
- payables. *See* accounts payable ratios
- payback period, calculating, 349
 - discounted, 351
 - exact undiscounted, 350-351
 - simple undiscounted, 349-350
- payments
 - defined, 306
 - loan payments, calculating, 306-312
 - mortgage payments, allowing principal paydowns, 321-323
- PEARSON() function, 229
- percentage summary calculations, 385-387
- percentile
 - calculating, 239, 259
 - defined, 238
- PERCENTILE.EXC() function, 239
- PERCENTILE.INC() function, 239

- periodic samples
 - defined, 246
 - extracting, 247-248
- periods
 - calculating for investment goals, 331-332
 - defined, 306
- per-unit constants, 355
- PHI() function, 229
- PI() function, 184
- PivotTables, 381
 - creating custom calculations, 391
 - calculated fields, 393-394
 - calculated items, 394-395
 - restrictions on, 392
 - grand totals, 381
 - hiding, 382
 - resetting to normal, 389
 - subtotals, 381-382
 - customizing calculation, 382-383
 - hiding, 382
 - value field summary calculations, 383
 - difference calculations, 383-385
 - index calculations, 389-391
 - percentage calculations, 385-387
 - running total calculations, 388-389
 - in worksheet formulas, 395-397
- plotting
 - best-fit trendlines, 267-268
 - exponential trends, 289-290
 - forecasts, 275
 - logarithmic trends, 294-295
 - polynomial regression, 300-301
 - power trends, 297-298
- PMT() function, 70, 307-308, 332-333
- Point mode, 4
- Poisson distributions, 256-257
- POISSON.DIST() function, 256
- polynomial regression, 266, 299-300
 - calculating values, 301-302
 - plotting, 300-301
- position, consolidating by, 35-38
- positive values, adding in range, 193
- power trends, 296-297
 - calculating values, 298-299
 - plotting, 297-298
- POWER() function, 184
- PP&E (property, plant, and equipment), 217
- PPMT() function, 309-310
- precedence. *See* order of precedence
- precedents
 - defined, 62
 - tracing, 63
- present value, 306
 - calculating, 339-340
 - inflation, 340-341
 - with PV() function, 341-344
 - relationship to discounting, 339
- price points, setting, 191
- pricing formulas, 207
 - break-even point, 211-213
 - discounts, 209
 - discount amount, 210
 - discount rate, 210-211
 - list price, 211
 - net price, 209-210
 - markups, 207
 - cost price, 209
 - markup amount, 208
 - markup rate, 208-209
 - selling price, 208
- principal and interest, calculating, 309-310
- principal paydowns, allowing, 321-323
- probability distributions, 252
 - binomial, 255
 - calculating, 253-255
 - hypergeometric, 256
 - Poisson, 256-257
- PROB() function, 254
- product margin, optimizing, 408-409
- PRODUCT() function, 184

projections

projections. *See* forecasts
PROPER() function, 80, 84
property, plant, and equipment (PP&E), 217
p-value, 263
PV() function, 318-319, 333, 341-344

Q–R

Quick Access Toolbar, adding scenarios to, 416
quick ratio, 226-227
QUOTIENT() function, 184

R2 (in regression analysis), 269-270
radians, converting to/from degrees, 185
RADIANS() function, 185-186
RAND() function, 184, 199-201
RANDARRAY() function, 32, 184, 202
RANDBETWEEN() function, 184, 201, 249
random letters, generating, 200
random numbers, generating, 198
 with RAND() function, 199-201
 with RANDARRAY() function, 202
 with RANDBETWEEN() function, 201
 with SEQUENCE() function, 202-203
random samples
 defined, 246
 extracting, 249
random variables, 246
range, calculating, 240
range lookups, 142-144
range names
 cell references in, 13
 converting to cell references, 52
 deleting, 52-53
 in formulas, 16
 applying new, 17-20
 pasting, 16-17
 troubleshooting, 51-53

ranges. *See also* array formulas; arrays
 adding only positive/negative values, 193
 applying conditions across, 116
 copying/moving, 11-12
 with absolute references, 13
 with mixed references, 14
 with relative references, 12-14
 counting
 blanks in, 131-132
 cells in, 232
 errors in, 132
 non-numeric values, 132
 occurrences in, 119-120
 ignoring errors, 132-133
 multiple ranges, array formulas on, 29
 recalculation, 10
rank
 calculating, 239
 defined, 238
RANK.AVG() function, 239
RANK.EQ() function, 239
RATE() function, 317-318, 330-331
recalculating data tables, 404
recalculation of formulas, controlling, 9-10
receivables. *See* accounts receivable ratios
records, defined, 361
#REF! error value, 53-54
reference formulas, operators for, 6
references. *See* cell references
referencing. *See also* cell references
 data types, troubleshooting, 54
 linked data types, troubleshooting, 51
 tables, 371
 entering table formulas, 373-374
 table specifiers, 371-373
regression analysis, 265
 methods of, 266
 multiple regression, 303-304
 polynomial regression, 299-300
 calculating values, 301-302
 plotting, 300-301

- simple regression on linear data, 266
 - best-fit trendlines, 266-274
 - forecasts, 274-279
 - seasonal sales model case study, 280-288
- simple regression on nonlinear data, 288
 - exponential trends, 288-293
 - logarithmic trends, 293-296
 - power trends, 296-299
- regression equation, 268-269
 - for forecasts, 277-278
- Regression tool in Analysis ToolPak, 273
- relative reference format, 12-14
 - ignoring when applying range names, 18
- removing
 - carriage returns with CLEAN() function, 90
 - characters from strings, 101-102
 - constraints, 427
 - extra string characters, 89
 - with CLEAN() function, 90
 - with TRIM() function, 89-90
 - line feeds
 - with CLEAN() function, 90
 - with SUBSTITUTE() function, 103
 - range names, 52-53
 - scenarios, 419
 - tracer arrows, 64
 - Watch Windows, 66
- rental properties, income investing versus, 342-343
- reorder level, 221
- repeating text with REPT() function, 92-93
- REPLACE() function, 80, 100-101
- reports (Solver), generating, 437
 - Answer report, 437-439
 - Limits report, 441
 - Sensitivity report, 439-440
- REPT() function, 80, 92-93
- reseasoning data, 287
- resetting PivotTables to normal, 389
- resizing arrays, 27
- restoring number formatting, 388
- results of formulas, troubleshooting, 56
- resuming Goal Seek, 407
- retail price, 209
- returning
 - arrays, 31-32
 - birthday, 164
 - dates, 157
 - with DATE() function, 158
 - with DATEVALUE() function, 158
 - with TODAY() function, 157
 - discount rates, 142-143
 - future dates, 161
 - future times, 175-176
 - last day of month
 - with DATE() function, 164
 - with EOMONTH() function, 164
 - n*th occurrence of weekday in month, 165-167
 - parts of dates, 159
 - with DAY() function, 159
 - with ISOWEEKNUM() function, 160
 - with MONTH() function, 159
 - with WEEKDAY() function, 160
 - with WEEKNUM() function, 160
 - with WORKDAY() function, 161
 - with WORKDAY.INTL() function, 162-163
 - with YEAR() function, 159
 - parts of times, 174
 - with HOUR() function, 174
 - with MINUTE() function, 175
 - with SECOND() function, 175
 - tax rates, 143-144
 - times, 173
 - with NOW() function, 173
 - with TIME() function, 174
 - with TIMEVALUE() function, 174

return on fixed assets

- return on fixed assets, 217-218
- Ribbon, displaying Developer tab, 42
- RIGHT() function, 80, 94
- risk premium, 306
- ROMAN() function, 184
- ROUND() function, 184, 186-187
- ROUNDDOWN() function, 185, 188
- rounding functions
 - for billable time, 191-192
 - CEILING.MATH(), 188-189
 - EVEN(), 189
 - FLOOR.MATH(), 188-189
 - INT(), 190-191
 - list of, 186
 - MROUND(), 187-188, 191-192
 - ODD(), 189
 - preventing calculation errors, 190-191
 - ROUND(), 186-187
 - ROUNDDOWN(), 188
 - ROUNDUP(), 188
 - setting price points, 191
 - TRUNC(), 190
- ROUNDUP() function, 185, 188
- row-and-column lookups, 149-150
- ROW() function, 31, 82, 121
- row names in range names, 18-20
- row numbers, finding for list values, 120-121
- rows
 - adding every *n*th row with MOD() function, 196
 - creating ledger shading, 197-198
- ROWS() function, 31
- RSQ() function, 273
- RTD() function, 136
- rule of addition, 253
- rule of multiplication, 253
- running total summary calculations, 388-389

S

- saddle points, 257
- safety stock, 220-221
- sales
 - analyzing versus advertising, 274
 - book publishing example, 355-356
 - seasonal sales model case study, 280-288
- sales mix, 212
- sales per employee, 215
- sales ratios, 214
 - sales per employee, 215
 - sales to current assets, 214
 - sales to short-term debt, 214
- sales to accounts payable, 224
- sales to current assets, 214
- sales to inventory, 220
- sales to short-term debt, 214
- sampling data, 246-247
 - extracting periodic samples, 247-248
 - extracting random samples, 249
- saving
 - Solver models, 433-434
 - Solver solutions as scenarios, 428
- scenarios, 412-414
 - adding
 - to Quick Access Toolbar, 416
 - to worksheets, 414-415
 - comparing, 417-419
 - deleting, 419
 - displaying, 415-416
 - editing, 416-417
 - merging, 417
 - saving Solver solutions as, 428
 - worksheet setup, 414
- scroll bars, inserting, 47-48
- SEARCH() function, 80, 97-99
- searching for substrings
 - with COLUMN() and CELL() functions, 99-100
 - with FIND() and SEARCH() functions, 97-99

- seasonal indexes, 284
- seasonal sales model case study, 280-288
- seasonally adjusting data, 286
- SEC() function, 186
- SECH() function, 186
- SECOND() function, 173, 175
- second-order polynomial regression equation, 299
- selecting arrays, 27
- selling price, 208
- Sensitivity report (Solver), 439-440
- sentence case, converting text to, 95
- SEQUENCE() function, 32, 185, 202-203
- serial numbers for dates/times, 153-154
- Series command, extending forecasts, 276-277
- series of deposits, calculating future value, 329
- series of letters, generating, 82-83
- SERIESSUM() function, 185
- server workbooks, 21
- shape of curve, calculating, 260-261
- SHEET() function, 125, 130
- SHEETS() function, 125, 130
- short-term debt
 - fixed assets to, 218
 - sales to, 214
- SIGN() function, 185
- significance levels, 262
- simple interest, 325
- simple regression, 266
 - on linear data, 266
 - best-fit trendlines, 266-274
 - forecasts, 274-279
 - seasonal sales model case study, 280-288
 - on nonlinear data, 288
 - exponential trends, 288-293
 - logarithmic trends, 293-296
 - power trends, 296-299
- simple undiscounted payback period, calculating, 349-350
- Simplex LP engine, 429
- SIN() function, 186
- SINGLE() function, 32
- single quotation marks (' ') in external references, 22
- SINH() function, 186
- SKEW() function, 260
- slack (in Solver Answer report), 438
- SLOPE() function, 273
- SMALL() function, 237-238
- smooth models, 429
- Solver, 421
 - adding constraints, 426-428
 - advantages of, 422
 - break-even analysis, 423-426
 - engines in, 429
 - generating reports, 437
 - Answer report, 437-439
 - Limits report, 441
 - Sensitivity report, 439-440
 - loading, 423
 - messages in, 434-435
 - non-negative constraints, 428
 - options, 429-432
 - saving
 - models, 433-434
 - solutions as scenarios, 428
 - solving transportation problem, 435-437
 - when to use, 422-423
- solving
 - algebraic equations, 411-412
 - transportation problem, 435-437
- SORT() function, 32, 364
- SORTBY() function, 32, 364-365
- sorting
 - tables, 361
 - without articles, 363-364
 - with formulas in, 362
 - on partial fields, 362-363

sorting

- with SORT() function, 364
- with SORTBY() function, 364-365
- values randomly, 200-201
- source (of links), changing, 23
- source ranges, 35
- source workbooks, 21
- specifiers. *See* table specifiers
- #SPILL! error value, 28, 54, 202
- spill ranges, troubleshooting, 54
- spills, 27
- spin boxes, inserting, 47-48
- spread, calculating, 240
- SQRT() function, 185
- SQRTPI() function, 185
- standard deviation, calculating, 241-242
- standard normal distribution, 257-258
- standard scores, calculating, 258-259
- statistical functions. *See also* table functions
 - AVERAGE(), 233
 - AVERAGEA(), 233
 - AVERAGEIF(), 233
 - AVERAGEIFS(), 234
 - BINOM.DIST(), 255
 - CONFIDENCE.NORM(), 262
 - CONFIDENCE.T(), 262
 - CORREL(), 251-252
 - COUNT(), 230
 - COUNTA(), 231
 - COUNTBLANK(), 231
 - COUNTIF(), 232, 254
 - COUNTIFS(), 232
 - COVARIANCE.P(), 250-251
 - COVARIANCE.S(), 250-251
 - FISHER(), 229
 - FREQUENCY(), 243-244, 253
 - HYPGEOM.DIST(), 256
 - KURT(), 261
 - LARGE(), 237-238
 - MAX(), 236, 240
 - MAXA(), 237
 - MEDIAN(), 234
 - MIN(), 236, 240
 - MINA(), 237
 - MODE(), 234
 - MODE.MULT(), 235
 - MODE.SNGL(), 235
 - NORM.DIST(), 259
 - OFFSET(), 246-249
 - PEARSON(), 229
 - PERCENTILE.EXC(), 239
 - PERCENTILE.INC(), 239
 - PHI(), 229
 - POISSON.DIST(), 256
 - PROB(), 254
 - RANK.AVG(), 239
 - RANK.EQ(), 239
 - SKEW(), 260
 - SMALL(), 237-238
 - STDEVA(), 242
 - STDEV.P(), 242
 - STDEVPA(), 242
 - STDEV.S(), 242
 - SUMPRODUCT(), 235-236
 - T.TEST(), 264
 - VARA(), 241
 - VAR.P(), 241
 - VARPA(), 241
 - VAR.S(), 241
 - Z.TEST(), 263
- statistics
 - descriptive, 229-230
 - averages, 232-236
 - counting items, 230-232
 - extreme values, 236-238
 - frequency distributions, 242-244
 - percentile, 239
 - range, 240
 - rank, 239
 - standard deviation, 241-242
 - variance, 240-241

- inferential, 245-246
 - confidence intervals, 261-262
 - correlation, 251-252
 - covariance, 250-251
 - hypothesis testing, 263-264
 - normal distribution, 257-261
 - probability distributions, 252-257
 - sampling data, 246-249
- status (in Solver Answer report), 438
- STDEVA() function, 242
- STDEVPA() function, 242
- STDEV.P() function, 242
- STDEV.S() function, 242
- strings
 - concatenating with delimiter, 91
 - converting, 84
 - to dates, 158
 - with LOWER() function, 84
 - with NUMBERVALUE() function, 85
 - with PROPER() function, 84
 - to times, 174
 - with UPPER() function, 84
 - counting characters in, 102
 - defined, 79
 - formatting, 85
 - with DOLLAR() function, 86
 - with FIXED() function, 87
 - with TEXT() function, 87-89
 - removing characters from, 101-102
 - removing extra characters, 89
 - with CLEAN() function, 90
 - with TRIM() function, 89-90
 - repeating with REPT() function, 92-93
 - substrings
 - extracting, 94
 - extracting names, 97-99
 - substituting, 100-102
- structured referencing of tables, 371
 - entering table formulas, 373-374
 - table specifiers, 371-373
- subproblems, 431
- SUBSTITUTE() function, 80, 101-103
- substituting substrings
 - with REPLACE() function, 100-101
 - with SUBSTITUTE() function, 101-102
- substrings
 - extracting, 94
 - with LEFT() function, 94
 - with MID() function, 94
 - with RIGHT() function, 94
 - searching for
 - with COLUMN() and CELL() functions, 99-100
 - with FIND() and SEARCH() functions, 97-99
 - substituting
 - with REPLACE() function, 100-101
 - with SUBSTITUTE() function, 101-102
- SUBTOTAL() function, 185
- subtotals in PivotTables, 381-382
 - customizing calculation, 382-383
 - hiding, 382
- SUM() function, 70, 185, 192, 194-195
 - counting occurrences in ranges, 119-120
 - operations on cells that meet condition, 117-118
- SUMIF() function, 185, 193
- SUMIFS() function, 185, 193-194
- summary functions
 - changing, 383
 - difference calculations, 383-385
 - index calculations, 389-391
 - list of, 383
 - percentage calculations, 385-387
 - running total calculations, 388-389
- summary reports, generating for scenarios, 417-419
- summary statistics. *See* descriptive statistics
- summing. *See* adding
- SUMPRODUCT() function, 31, 185, 235-236
- SUMSQ() function, 185

SUMX2MY2() function

SUMX2MY2() function, 185

SUMX2PY2() function, 185

SUMXMY2() function, 185

syntax (of functions), 71

T

T() function, 80

T.TEST() function, 264

table formulas, entering, 373-374

table functions

- advantages of, 375

- analyzing table of defects, 378-379

- DAVERAGE(), 377

- DGET(), 377-378

- DSUM(), 375

- entering, 376

- FILTER(), 370

- list of, 376

- SORT(), 364

- SORTBY(), 364-365

- syntax of, 375-376

- UNIQUE(), 371

table of defects, analyzing, 378-379

table ranges, 361

table specifiers, 371-373

TABLE() function, 405

tables. *See also* data tables; PivotTables

- defined, 361

- filtering, 365-366

 - with complex criteria, 366-369

 - with computed criteria, 369-370

 - with FILTER() function, 370

 - with UNIQUE() function, 371

- lookup tables, 136-137

- sorting, 361

 - without articles, 363-364

 - with formulas in, 362

 - on partial fields, 362-363

 - with SORT() function, 364

 - with SORTBY() function, 364-365

- structured referencing, 371

 - entering table formulas, 373-374

 - table specifiers, 371-373

TAN() function, 186

TANH() function, 186

target cells in Solver, 422

tax rates, returning, 143-144

term of loan, calculating, 315-317

text

- character codes, 80-81

 - CHAR() function, 81-83

 - CODE() function, 83

- concatenating with delimiter, 91

- converting, 84

 - formulas to, 49

 - with LOWER() function, 84

 - with NUMBERVALUE() function, 85

 - with PROPER() function, 84

 - to sentence case, 95

 - with UPPER() function, 84

- defined, 79

- formatting, 85

 - with DOLLAR() function, 86

 - with FIXED() function, 87

 - with TEXT() function, 87-89

- repeating with REPT() function, 92-93

text formulas, operators for, 6

text functions

- CELL(), 99-100

- CHAR(), 81-83

- CLEAN(), 90, 103

- CODE(), 83

- COLUMN(), 99-100

- DOLLAR(), 86

- FIND(), 97-99

- FIXED(), 87

- LEFT(), 94

- LEN(), 90

- list of, 79-80
- LOWER(), 84
- MID(), 94
- NUMBERVALUE(), 85
- PROPER(), 84
- REPLACE(), 100-101
- REPT(), 92-93
- RIGHT(), 94
- SEARCH(), 97-99
- SUBSTITUTE(), 101-103
- TEXT(), 87-89
- TEXTJOIN(), 91
- TRIM(), 89-90
- UNICHAR(), 81
- UPPER(), 84
- text values. *See* strings
- text-based charts, building, 92-93
- TEXT() function, 80, 87-89
- TEXTJOIN() function, 80, 91
- third-order polynomial regression equation, 300
- threads, 10
- tiered payments, calculating, 110
- time factor, 314
- time formatting symbols, 88
- time functions
 - HOUR(), 174
 - list of, 173
 - MINUTE(), 175
 - NOW(), 173
 - SECOND(), 175
 - TIME(), 174-176
 - TIMEVALUE(), 174
- time sheets, building, 177-181
- time value of money, 305-306
- TIME() function, 173-174
 - returning future times, 175-176
- times
 - adding, 176
 - calculating differences between, 176-177
 - with MOD() function, 195
 - converting strings to, 174
 - entering, 154-155
 - formatting, 154-155
 - returning, 173
 - with NOW() function, 173
 - with TIME() function, 174
 - with TIMEVALUE() function, 174
 - returning future, 175-176
 - returning parts of, 174
 - with HOUR() function, 174
 - with MINUTE() function, 175
 - with SECOND() function, 175
 - serial numbers for, 153-154
- TIMEVALUE() function, 173-174
- TODAY() function, 157
- tooggling difference calculations, 385
- totals, calculating cumulative, 194-195
- tracer arrows, 62-63
 - circular references, troubleshooting, 57
 - removing, 64
- tracing
 - dependents, 63
 - error cells, 64
 - precedents, 63
- transportation problem, solving, 435-437
- TRANSPOSE() function, 32
- TREND() function, 32, 270-271, 278, 301-304
- trendlines, 266
 - analyzing sales versus advertising, 274
 - calculating
 - with LINEST() function, 271-273
 - with TREND() function, 270-271
 - exponential trends, 288-289
 - calculating values, 290-291
 - with GROWTH() function, 291-292
 - with LOGEST() function, 292-293
 - plotting, 289-290
 - for forecasts, 274
 - extending with fill handle, 276
 - extending with Series command, 276-277

trendlines

- with FORECAST() function, 279
- with LINEST() function, 279
- plotting, 275
- regression equation for, 277-278
- seasonal sales model case study, 280-288
 - with TREND() function, 278
- logarithmic trends, 293-294
 - calculating values, 295-296
 - plotting, 294-295
- plotting, 267-268
- polynomial regression, 299-300
 - calculating values, 301-302
 - plotting, 300-301
- power trends, 296-297
 - calculating values, 298-299
 - plotting, 297-298
- R2 for, 269-270
- regression equation for, 268-269
- trigonometric functions, list of, 185-186
- TRIM() function, 80, 89-90
- troubleshooting formulas, 49
 - auditing worksheets, 62-63
 - dependent tracing, 63
 - error cell tracing, 64
 - evaluating formulas, 64-65
 - precedent tracing, 63
 - removing tracer arrows, 64
 - setting Watch Windows, 65-66
- circular references, 56-57
- converting to text, 49
- error checker, 58
 - choosing error action, 58-59
 - setting options, 59-62
- error values, 50
 - #CALC!, 50
 - #DIV/0!, 50
 - #FIELD!, 51
 - #N/A!, 51, 58
 - #NAME?, 51-53
 - #NULL!, 53
 - #NUM!, 53
 - #REF!, 53-54
 - #SPILL!, 54
 - #UNKNOWN!, 54
 - #VALUE!, 54
- IFERROR() function, 57-58
- incorrect results, 56
 - missing/mismatched parentheses, 9, 55
- TRUE() function, 106
- TRUNC() function, 185, 190
- two-digit years, 155-156
- two-input data tables, setup, 403-405
- TYPE() function, 125
- typing functions into formulas, 73-74

U

- undiscounted payback period, calculating, 349-351
- UNICHAR() function, 80-81
- UNICODE() function, 80
- UNIQUE() function, 32, 371
- #UNKNOWN! error value, 54
- unknowns in Solver, 422
- updating links, 22-23
- UPPER() function, 80, 84

V

- validating data, 40-42
- value field summary calculations in PivotTables, 383
 - difference calculations, 383-385
 - index calculations, 389-391
 - percentage calculations, 385-387
 - running total calculations, 388-389

- #VALUE! error value, 54, 248
- VALUE() function, 80
- values
 - adding, 192
 - with SUM() function, 192, 194-195
 - with SUMIF() function, 193
 - with SUMIFS() function, 193-194
 - categorizing by, 112-113
 - converting formulas to, 15-16
 - sorting randomly, 200-201
- VARA() function, 241
- variable cells in Solver, 422
- variable costs, 211
- variable interest rates, calculating future value
 - for investment goals, 334
- variable-rate mortgage amortization
 - schedules, building, 320-321
- variables
 - defined, 246
 - measures of association, 249-250
 - types of, 246
- variance, calculating, 240-241
- VARPA() function, 241
- VAR.P() function, 241
- VAR.S() function, 241
- varying cash flows, net present value with, 347-348
- viewing
 - current input mode, 5
 - formulas, 15
- VLOOKUP() function, 32, 136, 140-141
 - exact-match lookups, 144-146
 - range lookups, 142-144
- weekdays, returning *n*th occurrence in month, 165-167
- weekend due dates, avoiding, 122-123
- WEEKNUM() function, 157, 160
- weighted averages, 390
- weighted mean, calculating, 235-236
- weighted questionnaire results, calculating, 139
- what-if analysis, 399-400
 - data tables
 - adding formulas to, 402-403
 - editing, 405
 - one-input setup, 400-402
 - two-input setup, 403-405
- wildcard characters in MATCH() function, 147
- workbooks
 - displaying last updated date, 89
 - linking, 21
- WORKDAY() function, 123, 157, 161
- WORKDAY.INTL() function, 162-163
- WORKDAY.INTO() function, 157
- working capital, 224-225
- working capital ratio, 226
- working capital turnover, 225
- worksheets
 - account receivable aging worksheet,
 - building, 122-124
 - adding scenarios to, 414-415
 - auditing, 62-63
 - dependent tracing, 63
 - error cell tracing, 64
 - evaluating formulas, 64-65
 - precedent tracing, 63
 - removing tracer arrows, 64
 - setting Watch Windows, 65-66
 - dialog box controls
 - check boxes, 46
 - combo boxes, 46-47
 - group boxes, 44
 - inserting, 43
 - linking to cells, 43

W

- watching cell values, 65-66
- Watch Windows, setting, 65-66
- WEEKDAY() function, 122, 137, 157, 160, 165-167

worksheets

- list boxes, 46-47
- option buttons, 45
- scroll bars, 47-48
- spin boxes, 47-48
- time sheets, building, 177-181

X

- XIRR() function, 353
- XNPV() function, 348-349
- XOR() function, 106, 114
- x-values, 265

Y

- YEARFRAC() function, 157, 172
- YEAR() function, 157, 159
- years
 - finding leap years, 196-197
 - two-digit, 155-156
- y-values, 265

Z

- zero, dividing by, 50, 109
- Z-scores, calculating, 258-259
- Z.TEST() function, 263