

A

Excel 2007 Features
and Functions**INTRODUCTION**

The main features and functions of Excel 2007 are introduced here. Go through this appendix to start learning the simple technical material. To effectively and instinctively apply them, you have to practice modeling. The notes and exercises in the main text of this book are specifically provided for that purpose. As you do the modeling, remember to periodically review the list below and identify items you should know but are still unclear. Be sure to check out **Help/Contents** in Excel and **Help** in the **VB Editor**. They contain more information than most Excel and **VBA Macro** reference books.

Elementary

Files

- Default directory and some file management, Properties

- New, Open/Close

- Save, Save as, *.xlsx, *.xlsm, *.xltx, *.xltm, *.xlam; *.xls, *.xlt, *.xla, *.csv, *.htm, *.pdf

Rows, Columns, and Sheets

- Insert/Move/Hide row/Column/Sheet, Change column width, Change row height

- Rename Sheets (Tab name, color)

- Cells and Ranges, Comments

- Formulas (+-*/^, calculation order of $A1 + 2*B1/C1^2$)

- Editing formulas, Formula auditing

- Number formats (general, fixed, scientific (E), \$, and %), Date/Time formats

- Font (size, type, color), Alignment, Border, Patterns, Protection (locked, hidden)

- Buttons (bold, italic, underline, 0.00, \$, and %)

- Character strings and string arithmetic (“=A”&B6&“B”)

Commands

- Cut/Copy/Paste, Absolute/Relative/Mixed referencing (A1, \$A\$1, \$A1, and A\$1)

- PasteSpecial (Formats, Values, Transpose, Operations), Fill, Undo

Options File (password) protection, Macro Security
 Page layout: Gridlines, Row and Column Headers, Formulas, Page Break
 Drawing: Draw, Group, Order, Rotate/Flip, Auto-shapes, Arrows, Shadow, 3-D

Intermediate

Functions: Function wizard, SUM, COUNT, AVERAGE, MAX, MIN
 Functions: IF, SUMIF, SUMIFS, COUNTIF, COUNTIFS, SUMPRODUCT
 Data: Goal Seek, Scenarios, DataTable
 Charts: Types (Bar, Line, XY), Formats, Series (add, delete), Legends, Axis, Titles
 View: Windows, Menus, Toolbars, New (Multiple) windows, Freeze pane, Split, Custom views
 Customization: Office Button/ExcelOptions
 Page layout: Set/Clear Print Area, Print Preview, Header/Footer, Fit to Page
 Options: Formula calculation (Automatic, Manual, Iteration)

Advanced

Data: Sort, Validation, Form, Filter, PivotTable
 Functions: LOOKUP, VLOOKUP, HLOOKUP, MATCH, INDEX, INDIRECT, CHOOSE, OFFSET, ADDRESS
 Functions: NORMDIST, NORMSDIST, NORMINV, NORMSINV, EXP, LN, BINOMDIST, CRITBINOM, LINEST
 Functions: Array function and formula, FREQUENCY, RAND, RANDBETWEEN
 Functions: DSUM, DCOUNT, DAVERAGE, DMAX, DMIN

 Add-ins: Analysis Toolpak, Solver
 Form: Button, Group, Option, Check, Spinner, Slider, Control
 Macros: Sub and Function procedures, User-defined functions (CellFormula, ShowFormula)
 Protection: Worksheet, Workbook, File Sharing, VBA Project
 Developer: Visual Basic Editor (Alt+F11), VB Help, Object Browser, Project Reference

FURTHER REFERENCES

- Leong, 2014. Excel Web Resources
 - <http://isotope.unisim.edu.sg/users/tyleong/SpreadsheetModeling.htm#Excel>
 - <http://dl.dropboxusercontent.com/u/19228704/SpreadsheetModeling.htm#Excel>
- Microsoft Office Developer site
 - <http://msdn.microsoft.com/en-us/office>
- Walkenbach and Banfield, 2007. *Excel 2007 for Dummies: Quick Reference*, Wiley.
- Winston, 2007. *Microsoft Excel 2007: Data Analysis and Business Modeling*, Microsoft Press.

EXCEL 2007 PRIMER

Knowing the basic features and functions of the spreadsheet well is paramount before you can ever be competent in spreadsheet modeling. A short review is done in this appendix to help first-time users learn it faster, and experienced users learn it better. This cannot replace the many excellent books on the market that deal directly with Excel 2007, which you may want to refer to after you are through with Appendix A. *Note 1* through *Note 4* are essentially for first-time users. If you have used Excel for some time now, you may skip these notes and proceed straight to *Note 5*. Users of spreadsheet applications other than Excel will find similar features in their applications and should be able to draw parallel inferences.

NOTES

Note 1: SPREADSHEET WINDOW

The Excel spreadsheet file is designed to be like a book and is thus referred to as a **Workbook**. In any workbook, there are several pages, each denoted as a **Worksheet**. By default, a new workbook contains three worksheets (aptly named *Sheet1*, *Sheet2*, and *Sheet3*). You can easily insert or delete worksheets, which is similar to adding pages to or deleting pages from an exercise book. These worksheets can also be renamed and their tabs colored by you. Right-click on the worksheet tab. You will see, in the side menu, the options to do this and more.

There are rectangular spaces on each worksheet, each referred to as a **Cell**, into which you can enter values or formulas. These values can be text, numbers, or logical values (i.e., TRUE or FALSE). The mathematical formulas that you write into the cells are used to compute any desired results. This is the most formidable part of spreadsheets since their formulas can be made to link data and values from all other cells.

Cells, arranged in a grid, are identified by their row and column positions. There are two alternative ways to identify a cell: the default **A1** or the **R1C1** reference styles. For example, D2 refers to the cell in row 2 of column D and equivalently R2C4 refers to the same since column D is the 4th column. Alphabet column labels go from A to Z, then AA to AZ, BA to BZ, and so on to XFD. These correspond to columns 1 through 16,384.

Trivia Fact

Q: How many rows and columns are there in an Excel worksheet?

A: Each Excel 2007 worksheet has 1,048,576 rows and 16,384 columns. Excel 2003 and older versions have significantly fewer with 65,536 rows and 256 columns. The numbers look rather odd because they are numbers expressed in powers of 2. There are actually few situations where these limits are ever reached and the workbook would usually work rather slowly when the spreadsheets get to be so large.

To enter a formula into a cell, begin with the = sign. For example, to compute $1 + 2$, just type =1+2. Once you hit the return key, the result will immediately appear in the cell as the value 3. Note that Excel does not by default show the formula expression in the cell itself, but rather the result of the formula. You can view in the **Formula** bar the formula for the cell where the cursor resides. Alternatively, you can change the worksheet option to view formulas in all the cells (instead of values) of the sheet. This can be done by keying **Ctrl + `** (typically located just left of the 1 key in the upper-left corner of your keyboard). To switch back to the default option, key **Ctrl + `** again. If this does not work, change your keyboard settings and restart Excel, Windows, or both.

Quick Tip

The usual mathematical operators, namely multiply (*), divide (/), add (+), subtract (-), and power of (^), are applicable in Excel. Do note that symbols used in Excel for multiplication and division are not \times and \div . The order of calculation still follows the usual mathematical *BODMAS* (bracket, of, divide, multiply, add, subtract) convention. Only round brackets () can be used to group and alter the calculation order. Other types of brackets are not permitted in Excel. Also, brackets do not automatically indicate multiplication and so the multiply (*) operator has to be inserted between brackets where applicable.

Of course, Excel's computing ability is more than just adding two numbers. It has a large set of functions, and together with its (absolute, relative, and mixed) cell referencing system, you can construct complex mathematical and other kinds of formulations in a small collection of cells and massively replicate them with little effort to other cells to complete your model. In fact, you should only be working with **Cell References**, and not data values, in formulas. This will be further illustrated and discussed in *Note 2* and *Note 3*.

Note 2: COMPUTATION

Let us try to compute the average, maximum, and minimum of stock prices for company XYZ for all the months in a year, as given in *Figure A-1*.

	A	B	C
1			
2			
3		Month	XYZ
4		1	4.53
5		2	5.62
6		3	3.42
7		4	7.89
8		5	5.78
9		6	6.43
10		7	5.88
11		8	4.67
12		9	3.55
13		10	3.89
14		11	4.12
15		12	6.57
16		Average	5.20
17		Maximum	7.89
18		Minimum	3.42
19			

Figure A-1

Stock Prices for
Company XYZ

We will compute the average stock price in cell C16. You would have learned in high school that the average stock price is equal to the sum of all the 12 months' stock prices divided by 12. If you still remember this, your math teacher will certainly be proud of you. To make it easier for you, Excel provides the **AVERAGE** function to do the same arithmetic calculations. Now, enter into cell C16 the formula `=AVERAGE(C4:C15)` and key **Enter**. The result is exactly what we want!

Quick Tip 1

While entering the formula, you can select cell C4, hold down the mouse's left button and drag it to cell C15 instead of typing the cell reference C4:C15 in the formula. This is quicker and less prone to error. Indeed! Excel has such a useful function! Yes, and it has about 250 or more. To view the other functions, you can either go to Excel **Help**, or simply click on the **fx** button in the **Formula** bar.

Now, continue to compute the maximum and minimum stock prices in cells C17 and C18, respectively. What Excel functions should you use? You have guessed it, the **MAX** and **MIN** functions. Enter into cells C17 and C18, the formulas `=MAX(C4:C15)` and `=MIN(C4:C15)` to obtain the results.

Let us imagine that we want to analyze the average, maximum, and minimum prices for five different stocks, as shown in *Figure A-2*. We do not need to type the same functions five times. Instead, we can perform a **Fill** operation. To begin, we can just select the three cells with formulas, namely C16 to C18, as a group, bring the cursor to the lower-right corner of the group, and drag it to the next four columns (i.e., to column G). Magically, the average, maximum, and minimum stock prices for the next four stocks will be computed.

Figure A-2

Stock Prices
for Five Stock
Counters

	A	B	C	D	E	F	G
1							
2			Stock Prices				
3		Month	XYZ	Stock 2	Stock 3	Stock 4	Stock 5
4		1	4.53	6.65	4.33	3.11	8.97
5		2	5.62	7.56	4.21	3.22	9.01
6		3	3.42	5.34	3.99	3.45	9.12
7		4	7.89	5.46	3.89	3.67	8.99
8		5	5.78	6.98	4.34	3.57	8.76
9		6	6.43	5.89	4.65	3.45	9.15
10		7	5.88	6.33	4.58	3.37	9.24
11		8	4.67	6.21	4.66	3.21	9.06
12		9	3.55	5.99	3.98	3.09	8.95
13		10	3.89	6.23	4.12	2.99	8.88
14		11	4.12	6.34	4.23	3.12	9.17
15		12	6.57	6.01	4.55	3.45	8.94
16		Average	5.20				
17		Maximum	7.89				
18		Minimum	3.42				
19							

Fill, performing **Copy** and **Paste** operations, will automatically duplicate the formulas. It will copy the first set of selected cells and paste them over cells in other locations. In our example above, the formulas in the destination cells take their inputs not from the first set of stock prices, but the stock prices of cells directly above them. This phenomenon is known as **Relative Referencing**. The Fill operation can also be performed on cells that contain data values instead of formulas like what we have demonstrated so far. This will be discussed later.

Quick Tip 2

Dragging the selection one column to the right will cause an increment in the column letters in the formulas by one letter. Similarly, dragging the selection one row down will increment the row numbers in the formulas by one.

What if you do not wish the column letters or row numbers to be changed when performing a Fill operation or Copy and Paste operations? Well, you can switch off the relative referencing to use **Absolute Referencing** or **Mixed Referencing** instead (see next section). **Cell Referencing** is one of the most, if not the most, useful features in electronic spreadsheets. It is also one of the most difficult to understand. Most people find it confusing when they are first introduced to it. However, you will benefit tremendously from it if you make an effort to understand it. Once you have learned it, you will never go back to the old way. Let us move on to *Note 3*.

Note 3: CELL REFERENCING

Cell referencing, that does not permit column or row, or both to change when the cell formulas are copied elsewhere, is done using the \$ sign. To disallow column change, just prefix \$ to the column letter in the cell reference; to disallow row change, do the same to the row number in the cell reference. You can also do both concurrently. That is, =\$A\$1 will stay as =\$A\$1 when the cell formula is copied to any other cell in the worksheet. A cell with =\$A1 will only have its row number changed in the destination cell formula when copied to another cell in a different row; a cell with =A\$1 will have its column changed in the destination cell formula when copied to another cell in a different column. Let us try to understand this concept further by working through the following example. Prepare a multiplication table as shown in *Figure A-3*. How can we apply the **Fill** operation to compute the results effectively?

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2			1	2	3	4	5	6	7	8	9	10	11	12	
3		3													
4															

Figure A-3

Multiplication
Table for the
Number 3

In cell C3, type the formula `=B3*C2`. If you simply drag this cell selection to the cell to the right of C3, the results will be incorrect. This is because B3 will be changed to C3 in the formula in cell D3, to D3 in cell E3, and so on. To keep B3 as it is, modify the formula to `=$B3*C2`. Now, drag this cell selection to the right again. You will see that \$B3 remains as \$B3 in all cells. The \$ before B indicates that column letter B must not be changed in a **Fill** operation.

Quick Tip 1

`A1` refers to **absolute referencing** where the same value in cell A1 will be applied after a **Fill** operation, regardless of dragging across rows or across columns, because the \$ sign preceding column letter A restricts increments in the column letter, while the \$ sign preceding row number 1 restricts increments in the row number.

`A$1` refers to **mixed referencing**, where only the column letter is restricted from increments. `A$1` is the other mixed referencing where only the row number is restricted from increments.

The **F4** key in the top row on your computer keyboard is a toggle key to facilitate changing the cell reference to the different forms (`A1`, `A1`, `A$1`, `$A1`). Just hit the key after you have typed a cell reference to cycle through the set of cell reference forms.

As a further challenge, let us prepare a bigger multiplication table in *Figure A-4*. Can you figure out where the \$ signs should be added in the formula?

Figure A-4

Multiplication
Table for
1 to 12

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2			1	2	3	4	5	6	7	8	9	10	11	12	
3		1													
4		2													
5		3													
6		4													
7		5													
8		6													
9		7													
10		8													
11		9													
12		10													
13		11													
14		12													
15															

In this multiplication table, enter into cell C3 the formula `=$B3*C2`. The \$ sign preceding column letter B ensures that the reference to column B remains static, while the \$ sign preceding the row number 2 ensures that the reference to row 2 remains static. Try it out. If it works, expand the multiplication table to 50 by 50 or even 200 by 200 to see if it is still okay.

Quick Tip 2

All cell formulas should only contain cell references, operators, and functions. There should not be any numerical, text, or logical value in them. Formulas therefore show the relationships between cells, which in turn can contain values or formulas. This makes updating the input data values explicit and clearer to the user.

To see a visual representation of the cell relationships, click on the cell of interest and then click the cell formula in the **Formula** bar. The cell references in it will then become color coded, and boxes whose colors match those used in the spreadsheet will frame the corresponding cells referenced in the formula. You can even modify the cell formula by dragging or changing the shape of these boxes.

An alternative approach is to use **Formula auditing**. This is found in **Formula** tab. Click **Trace Precedents** or **Trace Dependents** to visually show arrows linking the cells. Also try **Show Formulas**. This works the same as **Ctrl + `**.

Hopefully, examples used in this note have helped to illustrate the usefulness of cell referencing and clarified its correct use. We are now ready to learn more advanced features of Excel. As you model and analyze more complex problems, you will need to learn these additional spreadsheet features to be proficient and effective. In the following few notes, we will cover some of the more useful advanced features.

Note 4: FORMATTING

The appearance of a cell can be altered by changing its row height or column width. Rows or columns can be moved, deleted, or inserted to alter the arrangement of cells in any spreadsheet model. The cell references in the affected formula will automatically change without altering the underlying logic in the cell formulas. Most of the formatting actions can be done intuitively using the options available in the **Home** tab as shown in *Figure A-5a*. The options available include changing the font type, style, size, alignment, cell shading, and borders.

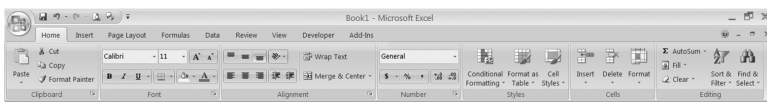


Figure A-5a

Formatting
Options in the
Home tab

The value in a cell is automatically formatted according to the way you entered it. If you enter 0.3501, then it would appear as such with the same number of decimal places. If the intended entry is 35.01%, it may be better to enter as such. The decimal value entered can of course be formatted to be a percentage by clicking the % icon in

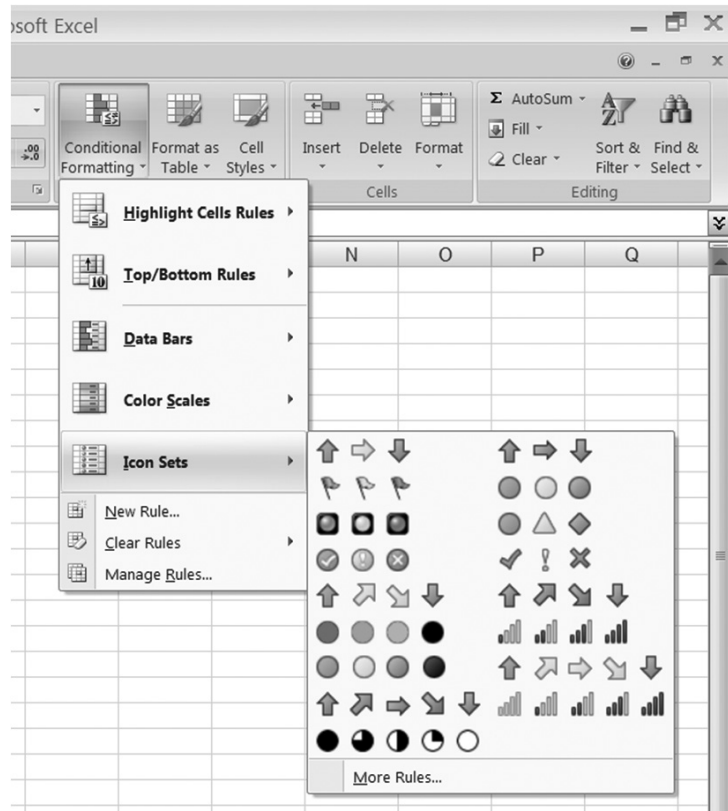
the menu. Other icons allow you to change the number of decimal places displayed, add commas, or add currency prefix. You can also right-click on a cell and then select **Format Cells** to open the dialog.

No matter how a value is reformatted, the value stored in the cell remains unchanged. In particular, a cell with the value 35.01% displayed is actually stored as 0.3501 and will be computed as 0.3501 and not 35.01 in formulas. There is therefore no need to divide by 100 to get the proportion value. You, however, should exercise caution when reducing the number of decimal places displayed as it may lead to calculations being misinterpreted. As a convention, it would be good to keep the number of decimal places the same for all values of the same variable and possibly across similar variables in the whole spreadsheet.

The format of a cell may be changed conditionally. The condition may be the state of value in the cell to be formatted or based on a logical formula. The logical formulas topic is discussed in *Note 10*. Simple **Conditional Formatting**, in the form of **Data bars**, **Color scales**, and **Icon sets** (e.g., traffic lights and directional arrows, shown in *Figure A-5b*), are available to make it easier to sense cells values, picking up the relevant ones especially when they are among many in large tables.

Figure A-5b

Simple
Conditional
Formatting



More complex Conditional Formatting will require the use of logical formulas. In the Excel **Home** tab, select **Conditional Formatting/New Rule**. Select **Format only cells that contain** to change the format based on the cell values or select **Use a formula to determine which cells to format** to specify the desired logical formula, as shown in *Figure A-6*. In Excel 2007, the maximum number of rules is 64 and you can specify whether the checking should stop or continue if the rule is TRUE.

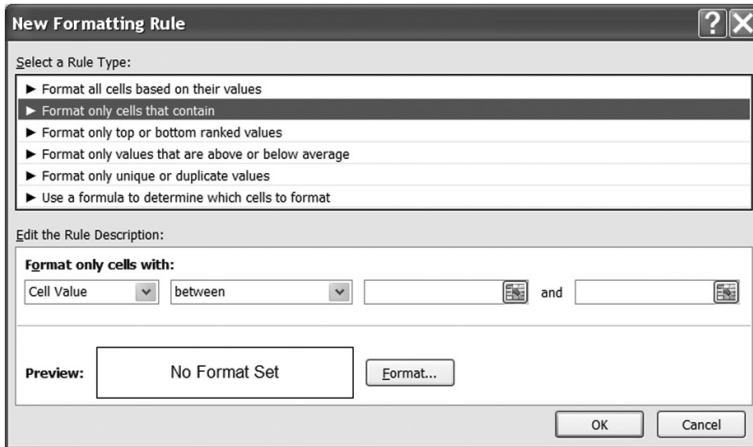
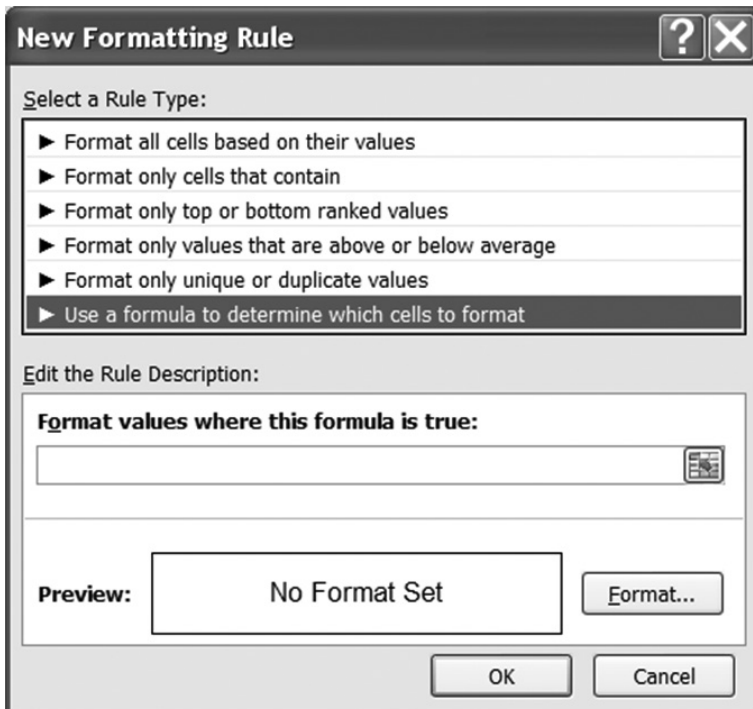


Figure A-6

Complex
Conditional
Formatting



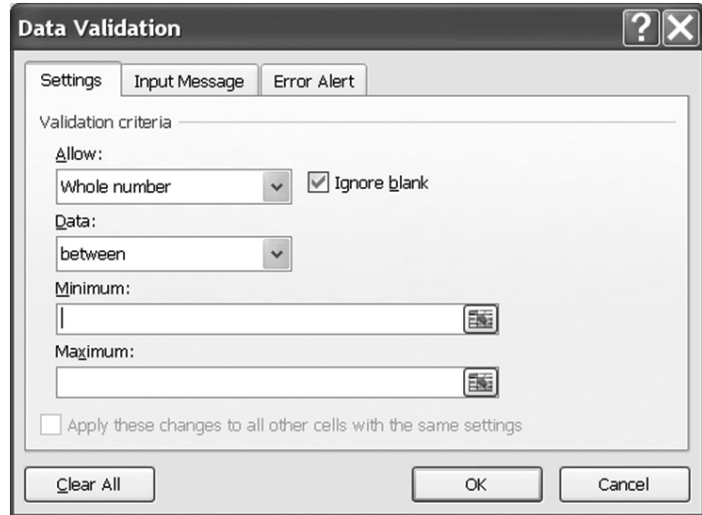
Note 5: DATA VALIDATION AND LOOKUP**Data Validation**

Whereas cells with formulas may be protected, input cells need to be available for users to enter the required values. These cells can therefore be subject to entry mistakes. To offer some assurance that the input cells are correctly used, you can apply Data Validation to check the inputs as they are entered. Data validation checks the variable type and value range for applied cells.

To set the validation for a cell, select the cell and then select **Data/Data Validation/Data Validation**. In the dialog as shown in *Figure A-7*, select the variable type in **Allow**, and then the conditions in **Data**. Enter also the lower and upper limits, where applicable, of the values.

Figure A-7

Data Validation



Ever think of allowing users to enter inputs by letting them choose from a list of permitted values? Excel's Data Validation also allows you to prepare a list of given values to be assigned to a cell. Once assigned, a drop-down list will appear to allow the user to make the selection. This helps to minimize mistakes in data entry.

Let us construct a drop-down list to allow a user to select *gender (female or male)*.

1. Select the cell where the drop-down list is to appear.
2. From the **Data** menu, select **Validation**.
3. In the **Settings** panel, in the **Allow** drop-down list, select **List**.
4. In the **Settings** panel, in the **Source** input area, enter *Female, Male*.

Quick Tip 1

For more flexibility, you can put the source input values as a group in a separate vertical or horizontal cell range, and in step 4 above, set the cell range reference or make the selection instead of entering the values.

For more complex data validation, you can use the **Custom** option in **Allow** of the **Data Validation** dialog to set logical formulas to govern the cell's data validation. For example, you may want to validate that a particular data value in a column of a table has not been entered in the rows preceding. To be able to do this, you must first learn logical formulas and the **COUNTIF** function. These topics are covered in *Note 10*.

Table Lookup

One of the most useful features of Excel is its ability to find the corresponding result for a lookup value in a table. For example, at a game stall, if you have shot three or fewer targets, you win yourself a bronze medal, and if you have shot four to seven targets, you win a silver medal, and finally, if you have shot eight to ten targets, you win a gold medal. The stall owner needs to look up the number of targets shot in a table to determine what kind of medal to award. This example seems trivial because there are only three categories of results. Imagine having to manage a large number of categories, with dynamically changing lookup values. The values in the table themselves may even be dynamically changing. The table lookup feature will prove to be extremely helpful then. Here are some lookup functions:

LOOKUP(*lookupValue*, *lookupRange*, *valueRange*)

- Returns the value in *valueRange* that has the same relative position that *lookupValue* has in *lookupRange*. The values in *lookupRange* must be in ascending order. If **LOOKUP** cannot find *lookupValue* in *lookupRange*, it matches the largest value in *lookupRange* that is less than *lookupValue*.
- For example, **LOOKUP**(6, A1:A6, B5:B10) with (3, 5, 7, 9, 11, 12) in A1:A6 and (2, 4, 6, 8, 10, 12) in B5:B10 would return the value of 4, as the value 5 in A1:A6, the largest value less than or equal to 6, is in the second position, and the second value in B5:B10 is 4.
- **LOOKUP**(**RAND**(), *lookupRange*, *mid-binValueRange*) returns a random value, according to the distribution of the values listed in the table defined by the mid-bin value and lookup ranges. This is one of the ways you can generate random values from a given distribution frequency table. Refer to *Note 6* for comments on the **RAND** function and other ways to generate random values.
- **LOOKUP** is best applied when the *lookupRange* and *valueRange* are not residing in the same table or are not of the same (row or column) orientation.

VLOOKUP(*lookupValue*, *tableArray*, *columnIndexNum*, [*rangeLookup*])

- Searches for *lookupValue* in the left-most column of *tableArray*, and returns the value in the same relative row position in the column identified by *columnIndexNum*. The *columnIndexNum* is a positive integer and *rangeLookup* is a logical value. The square brackets around *rangeLookup* indicate that it is an optional input. This square bracket convention applies to all other functions as well.
- When *rangeLookup* is TRUE (or omitted), the function will look for an approximate match for *lookupValue* in the left-most column, similar to the way the LOOKUP function works. For approximate match lookup, the values in the left-most column of *tableArray* must be in ascending order.
- However, when the *rangeLookup* is FALSE, an exact match for *lookupValue* in the left-most column is searched, and returns #N/A if none is found. The first matched position is returned if there are multiple possible matches.
- VLOOKUP is best applied when the *lookupValue* is compared with values in the left-most (i.e., first) column of *tableArray*.

MATCH(*lookupValue*, *lookupRange*, [*matchType*])

- Returns the relative position that the *lookupValue* is in *lookupRange*.
- If *matchType* is 0, MATCH looks for an exact value of *lookupValue* in *lookupRange* and returns #N/A if none is found. The first matched position is returned if there are multiple possible matches.
- If *matchType* is 1 or omitted, MATCH looks for the largest value less than or equal to the *lookupValue*. The *lookupArray* must first be in ascending order: ...-1, 0, 1, ..., A-Z, FALSE, TRUE.
- If *matchType* is -1, MATCH looks for the smallest value greater than or equal to the *lookupValue*. The *lookupArray* must first be in descending order. Only MATCH can deal with the approximate matches of a descending sorted *lookupRange*.

INDEX(*tableArray*, *rowIndex*, [*columnIndex*])

- Returns the value found in the relative *rowIndex* and *columnIndex* positions in *tableArray*.
- When *tableArray* is a single column of values, then *columnIndex* is not needed.
- However, when *tableArray* is a multiple-row and multiple-column matrix, then both *rowIndex* and *columnIndex* are needed to identify the value.

Quick Tip 2

The clever use of MATCH within INDEX allows us to identify values using dynamically changing *rowIndex* and *columnIndex*, where the *rowIndex* and *columnIndex* are results obtained from using MATCH twice.

Note 6: RANDOM NUMBERS AND RESAMPLING**Random Numbers**

Random numbers are useful, especially in Monte-Carlo simulations, where variations of input values are needed. Two functions are available in Excel to generate random numbers, one for continuous values and the other for discrete values.

RAND() returns with equal probability a random value in [0,1), i.e., between 0 and 1, inclusive of 0 but not of 1. The brackets, with nothing inside, are needed when you enter the function into a cell. This function can be used in formulas to generate random values of other probability distributions and interval values. For example, to generate a random real value uniformly distributed between 5 and 9, we can use $=5+RAND()*(9-5)$. So when **RAND** is 0, it returns 5; when **RAND** is almost 1, it returns a number that is almost 9 and when **RAND** is a value between 0 and 1, it generates the interpolated value between 5 and 9 with uniform probability density.

RANDBETWEEN(loNum, hiNum) returns with equal probability a random integer in $\{loNum, loNum + 1, \dots, hiNum\}$, i.e., integers between *loNum* and *hiNum*, $loNum < hiNum$ being integers. For example, **RANDBETWEEN**(1, 6) can be used to simulate the toss of a die. This function is only available when **Analysis ToolPak** (a standard **Add-in** in Excel) has been activated. You do this by selecting **Office/Excel Options/Add-ins/Excel Add-ins/Go**. Alternatively, without invoking Analysis ToolPak, formula $=INT(RAND()*(hiNum - loNum + 1)) + loNum$ would also give the same result. That is, you can also simulate the toss of a die using $INT(RAND()*6) + 1$.

Quick Tip

Calculation Options is found in the **Formulas** tab. When the Calculation option is (by default) set to **Automatic**, any recomputation or data entry into the worksheet will cause the random function to generate a new random number. To restrict this, especially when it is rather disruptive during model construction, set the Calculation option to **Manual**, and key **F9** whenever you wish to regenerate a new random value.

Resampling Data

Data resampling is important when we want to generate random data points from a handful of sample data points. Excel has some useful functions that can be used to perform data resampling. (Refer to the *Data Simulation* exercise and the *Resampling* tool in Chapter 6 for more details.)

SMALL(*valueRange*, *k*)

- Returns the *k*th smallest value, with *k* being an integer, in the data set specified by *valueRange*.
- For example, SMALL(A1:A20,1) returns the smallest value in A1:A20 (same result as MIN(A1:A20)), SMALL(A1:A20,2) returns the second smallest value, SMALL(A1:A20,3) returns the third smallest value, and so on. When used against a column of running serial numbers 1, 2, 3, ..., it can be used to sort a set of numbers in ascending order.
- SMALL(*valueRange*, RANDBETWEEN(1, *n*)) resamples with equal probability the *n* sample data values stored in *valueRange*.

LARGE(*valueRange*, *k*)

- Returns the *k*th largest value, with *k* being an integer, in the data set specified by *valueRange*.
- For example, LARGE(A1:A20,1) returns the largest value in A1:A20 (same result as MAX(A1:A20)), LARGE(A1:A20,2) returns the second largest value, LARGE(A1:A20,3) returns the third largest value, and so on. When used against a column of running serial numbers 1, 2, 3, ..., it can be used to sort a set of numbers in descending order.
- LARGE(*valueRange*, RANDBETWEEN(1, *n*)) also resamples with equal probability the *n* sample data values stored in *valueRange*.

PERCENTILE(*valueRange*, *k*)

- Returns the *k*th fractile, with *k* being a real number between 0 and 1, from among the data in *valueRange*.
- For example, PERCENTILE(A1:A5,0.5) with (91, 33, 52, 45, 67) in cell range A1:A5 returns the 50th percentile value of 52. PERCENTILE(*valueRange*, RAND()) works like SMALL(*valueRange*, RANDBETWEEN(1, *n*)), except now PERCENTILE also interpolates when the fractile required does not coincide with one of the given sample data points and it returns a continuous value.
- By using PERCENTILE to resample data points, you may end up having a resulting data point which is not one of the given sample data points, due to interpolation. When used with the INT function, it is even applicable for resampling discrete data.

Note 7: AUTOMATIC TABULATION

DataTable is an Excel feature that automatically generates a table of results of a spreadsheet model by replacing up to two of its variables with many given sets of input values for them. This can be done for a single input variable (in 1-dimensional DataTable) with one or more output variables, or for two input variables (in 2-dimensional DataTable) with only one output variable.

This operation is particularly useful when the model comprises more than just a formula in a single cell.

We will illustrate, as shown in *Figure A-8*, the construction of a 1-dimensional Data Table to compute the values of two output variables Y and Z using different values of input variable X. We arbitrarily define $Y = 2X$ and $Z = 3X + 2$.

1. Prepare a column containing the candidate values 3, 6, ..., 30 for input variable X in cells B5:B14.
2. Enter output formula $Y = 2X$ into the cell C4 as $=2*B2$. Note that the formula for output variables must reside in the row just above the first candidate input value for X, with the first output variable taking the first column to the right of the input value column, the second output variable in the second column, and so on. In addition, these output formulas must refer as input variable X a cell outside of the table. In this example, this cell is B2.
3. Enter the second formula $Z = 3X + 2$ into the cell D4 as $=3*B2+2$. This second formula resides to the right of the first formula. Repeat the step for additional formulas (if any).
4. Select the whole table in B4:D14 and then activate the computation by selecting **Data/What-If Analysis/Data Table**.
5. A dialog will prompt you to enter the **Row Input Cell** and **Column Input Cell**. Since this is a 1-dimensional Data Table constructed with input X values in a column, you can ignore the Row Input Cell and enter B2 as the Column Input Cell.
6. Results for the output variables will be computed automatically and placed into the table.

	A	B	C	D	E
1					
2		2			
3					
4			=2*B2	=3*B2+2	
5		3			
6		6			
7		9			
8		12			
9		15			
10		18			
11		21			
12		24			
13		27			
14		30			
15					

Figure A-8

One-dimensional
Data Table

The set-up of a 2-dimensional DataTable that computes the values of output variable $W = X1 * X2$ for given values of two input variables $X1$ and $X2$, as shown in *Figure A-9*, is illustrated as follows:

1. Prepare a column containing the candidate values 3, 6, ..., 30 for the first input variable $X1$ in cells C6:C15.
2. Prepare a row containing the candidate values 2, 4, ..., 10 for the second variable $X2$ in cells D5:H5.
3. Enter output formula $W = X1 * X2$ into the cell C5 as $=C2*C3$. Note that for a 2-dimensional DataTable, only one output variable can be computed. Its formula should reside at the upper left-hand corner, where the candidate values of $X1$ and $X2$ intersect. As in the 1-dimensional DataTable, this formula must refer to two cells outside of the table as $X1$ and $X2$. In this example, the two cells for $X1$ and $X2$ are C2 and C3, respectively.
4. Select the whole table in C5:H15 and activate the computation by selecting **Data/What-If Analysis/Data Table** from the main menu.
5. In the dialog, you will be prompted to enter the **Row Input Cell** and **Column Input Cell**. Enter C2 as the Column Input Cell and C3 as the Row Input Cell; both refer to variables $X1$ and $X2$, respectively, in the formula.
6. Results for output variable W will be computed automatically and placed in the corresponding positions in the table.

Figure A-9

Two-dimensional
DataTable

	A	B	C	D	E	F	G	H
1								
2		X1	1					
3		X2	2					
4								
5			=C2*C3	2	4	6	8	10
6			3					
7			6					
8			9					
9			12					
10			15					
11			18					
12			21					
13			24					
14			27					
15			30					
16								

Quick Tip

DataTable can be used to collect replication results in *Monte-Carlo* simulations. To do this, simply use the row (and column) input value (or values in combination) as the replication counter and use any arbitrary empty cell(s) to be the input cell(s).

Tool 30 (TableSolve) in Chapter 9 provides macros that emulate DataTable operations and extend upon them. They allow you to do multiple operation computations that concurrently involve more than two input variables and more than one output variable. Also, the *TableSolve* subroutines can be set to call any subroutine of your choice with every

change in data values in its operations. This is particularly useful when the computation involves any form of search or optimization.

Note 8: CHART

Charts are visual representations of data points that can in most cases display results more effectively than tables. Some of the commonly used charts in spreadsheet include bar chart, pie chart, and XY scatter plot. Excel 2007 provides a series of corresponding icon buttons in the **Insert** tab for you to select from. Let us construct a column chart for a stock investment portfolio given in *Figure A-10*.

	A	B	C
1			
2		Type	Percentage Investment
3		Stock 1	20%
4		Stock 2	25%
5		Stock 3	10%
6		Stock 4	5%
7		Stock 5	40%
8			

Figure A-10

Stock Investment
Portfolio Data

Select the data to plot (inclusive of table headers if any) and then click on the **Insert** tab to access the **Charts** group and work from there, as shown in *Figure A-11*. If you are plotting all the data in the table, you can simply put your cursor anywhere in the table. The whole table will be automatically selected as you (following the steps above) select the chart type. Once constructed, you can modify your chart using the tools available in the additional **Design** and **Layout** tabs, which only appear after a chart is selected.

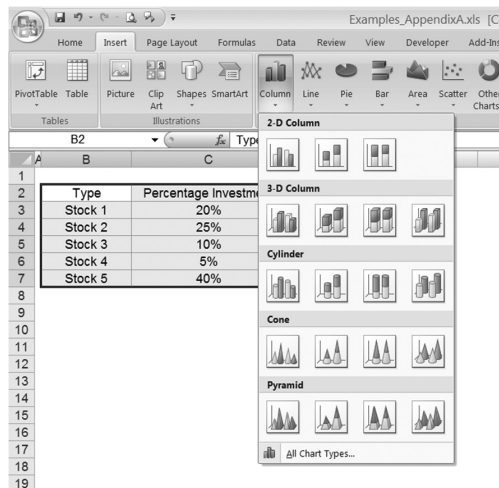


Figure A-11

Charts in Excel

In **Scatter** charts, you can right-click on the data markers or plotted line to **Add Trendline**. In the dialog for this, there are various options available, including setting the trend type, forecast, equation, R-square and format.

Note 9: DATE, TIME, AND TEXT

Date and Time

There are two date systems in Excel, namely the default **1900 date system** and the **1904 date system**. In the default date system, 1 January 1900 is day 1, and 2 January 1900 is day 2, and so on. To change to the alternative date system for a workbook, go to **Office/Excel Options/Advanced/When calculating this workbook**.

Now, why does Excel need to have an alternative date system? There is an error in the 1900 date system because Excel erroneously assumes 29 February 1900 exists but 1900, unlike other years that are divisible by 4, is not a leap year! Therefore, all dates after 29 February 1900 are in fact assigned an integer one larger than they should be. Calculations using dates that straddle 29 February 1900 will be off by a day. If you will not be doing computations involving dates before 1 March 1900, this mistake will not affect you. The 1904 date system corrects for this mistake by starting instead with 1 January 1904 as day 0. This system is however not applicable when you use dates before 1 January 1904.

There are several functions in Excel that allow us to manage dates and days:

- **TODAY()** returns the current date.
- **YEAR(serialNumber)** returns the year corresponding to the *serialNumber*.
 - Wrong: YEAR(14-Jan-05)
 - OK: YEAR("14-Jan-05")
 - OK: YEAR(B15) where B15 has the value 14-Jan-05
 - OK: YEAR(39014) = 2006
- **MONTH(serialNumber)** returns the month and **DAY(serialNumber)** returns the day corresponding to the serial number.
 - MONTH(39014) = 10, DAY(39014) = 24 since 39014 is 24 October 2006 in Excel.
- **DATE(year, month, day)** returns the serial number.

QuickTip

When dealing with subtraction of dates, the minus sign is easily mistaken as the dash used in dates. So, please take note of the correct and wrong ways to denote dates subtraction.

- WRONG: 14-Jan-05 – 23-Sep-04
- OK: "14-Jan-05" – "23-Sep-04" = 113
- OK: DATE(2005,1,14) – DATE(2004,9,23) = 113

Time is also managed in a serial number format, except time is stored as the fractional part of the serial number, that is, the digits to the right of the decimal point. A 24-hour day will make up a whole number 1, so part of a day, say 12 hours, will be 0.5. Let us look at some of the functions that can help us manage time.

- **NOW()** returns the current date and time.
- For example, if now is 8:00AM 25 December 2005, then the value of NOW() is 38711.3333333 where
 - 38711 is the day, which is 25 December 2005, and
 - .3333333 is the time of the day, which is 1/3 of the day.
- So, 8:00AM 26 Dec 2005 is 38712.3333333.
- **TIME(hour, minute, second)** returns the serial number to the right of the decimal point in the 0.##### format.
- **HOUR(serialNumber)**, **MINUTE(serialNumber)**, and **SECOND(serialNumber)** return respectively the hour, minute, and second of *serialNumber*.
 - HOUR(39461.847) = 20
 - MINUTE(39461.847) = 19
 - SECOND(39461.847) = 41

Text Management

Other than values and formulas, Excel can also manage texts (or commonly known as strings, denoting in short a string of characters). Important text functions include **CONCATENATE**, **LEN**, **FIND**, **LEFT**, and **RIGHT**.

CONCATENATE(string1, string2, ...)

- Combines end-to-end text strings *string1*, *string2*, and others. The alternative approach is to use the concatenation operator **&**.
- Therefore, "ABC" & "DEF" will yield "ABCDEF". This can be done to text or cells with values. For example, ">" & E20 yields ">5" when cell E20 contains the value 5. In this context, Excel implicitly converts number into text before completing the concatenation.
- This operation is helpful for specifying the criteria in functions such as **COUNTIF**, **COUNTIFS**, **SUMIF**, and **SUMIFS**. These functions are discussed in the last section of *Note 10*.

LEN(string)

- LEN returns the number of characters in text *string*.

FIND(findString, string, [startNumChar])

- FIND searches for text *findString* in text *string*, starting from the *startNumChar* position in *string*. Integer *startNumChar* is assumed to be 1 if it is not specified. If successful, it returns

the position of the first character of *findString* in text *string*, counting from the left. If there are multiple occurrences of *findString* in *string*, then the first occurrence will be the one found.

- For example, FIND("o", "Flower shop") returns 3 and FIND("er", "Flower shop") returns 5.

LEFT(*string*, [*numChar*])

- LEFT returns the left-most *numChar* characters in text *string*. Integer *numChar* is assumed to be 1 if it is not specified

RIGHT(*string*, [*numChar*])

- RIGHT returns the right-most *numChar* characters in text *string*. Integer *numChar* is assumed to be 1 if it is not specified.

All these text functions can be creatively applied to do many interesting things in your spreadsheets. It can, for example, split a full name into first name and last name, or extract postal codes from full addresses.

Note 10: LOGICAL FORMULA AND FUNCTION

A logical formula is one that returns a TRUE or FALSE value. For example, a cell with formula =(B4=C5) will return a TRUE when the result contained in cell B4 is the same as that of C5 and a FALSE otherwise. In logical formulas, several operators other than the equal operator (=) can be used. These include:

- < less than
- <= less than or equal to
- > greater than
- >= greater than or equal to
- < > not equal to

There are also logical functions such as **AND**, **OR**, **NOT**, and **IF**. AND and OR can combine the results of more than one logical test to depict more complex evaluations. AND returns a TRUE if and only if all logical tests within it are TRUE; OR returns a TRUE when at least one logical test it evaluates is TRUE; NOT transforms a FALSE to a TRUE, and a TRUE to a FALSE. The results of evaluation using AND, OR, and NOT with two arguments can be summarized as follows:

Function	Result
AND(TRUE, TRUE)	TRUE
AND(TRUE, FALSE)	FALSE
AND(FALSE, TRUE)	FALSE
AND(FALSE, FALSE)	FALSE

Function	Result
OR(TRUE, TRUE)	TRUE
OR(TRUE, FALSE)	TRUE
OR(FALSE, TRUE)	TRUE
OR(FALSE, FALSE)	FALSE

Function	Result
NOT(TRUE)	FALSE
NOT(FALSE)	TRUE

Logical formulas can be used in conjunction with arithmetic computations. In such cases, TRUE will be evaluated as a 1 and FALSE as a 0. Therefore, formula $=(B4=C5)*1$ will return a 1 when the result contained in cell B4 is the same as that of C5 and a 0 otherwise. Also, the multiplication of two logical tests would be equivalent to an AND operation. That is, $=(B4=C5)*(D5>F2)$ would give the same conclusion as $AND(B4=C5, D5>F2)$, except that the former will return a 0,1 result and the latter a TRUE, FALSE result.

IF(*logicalTest*, *resultIfTrue*, *resultIfFalse*)

The IF function is designed to evaluate a *logicalTest* condition, that should return a TRUE or FALSE. If *logicalTest* is TRUE, then *resultIfTrue* will be displayed; if FALSE, then *resultIfFalse* will be displayed. The *logicalTest* condition would be as in a logical formula, except without the = sign prefix. Both *resultIfTrue* and *resultIfFalse* can be values, cell references, or formulas. For example, to test if the age of a person who is over 18 to enter a club is $IF(age>18, "Welcome", "You are underage!")$.

Quick Tip 1

An IF function with another IF function as its *resultIfTrue* or *resultIfFalse* is known as a nested IF function. Excel 2007 allows up to 64 levels of nesting.

A more complex example would be a club that requires female members to be over 18 years of age while male members to be over 21. The IF function for this will be $IF(OR(AND(age>18, gender="Female"),AND(age>21,gender="Male")), "Welcome", "You are underage!")$.

Quick Tip 2

MAX(*value*, 0) and **MIN**(*value*, 0) are shorter and more elegant substitutes to $IF(value > 0, value, 0)$ and $IF(value < 0, value, 0)$, respectively. So indirectly, MAX and MIN, though classified by Excel as statistical functions, are also logical functions.

IFERROR(*value*, *valueIfError*)

Function IFERROR checks value against the error types #N/A, #VALUE!, #REF!, #DIV/0!, #NUM!, #NAME?, or #NULL!. If *value* is not in error, then *value* is returned. Otherwise, *valueIfError* is returned. It functions as an error trap to correct equations such as X/Y when Y can be 0. This function is not available in Excel 2003 and older versions. So do not use it if your workbook is to be shared with users of older versions.

Logical Count and Sum

The **COUNT** function counts the number of cells in the specified cell range that contains numerical values, while **COUNTA** counts the number of cells that contains numerical

or text values. When you need to count the number of cells that satisfies a certain evaluation criterion or a set of criteria, use respectively the **COUNTIF** and **COUNTIFS** functions. Similarly, there are **SUM**, **SUMIF**, and also **SUMIFS** functions. The **SUMIFS** and **COUNTIFS** functions are not available in Excel 2003 and older versions.

COUNTIF(*range, criteria*)

- Counts the number of values in the specified cell range that satisfies the criteria.
- The criteria is a simple logical expression expressed (in quotes) as a text. For example, **COUNTIF**(A1:A20, ">5") counts the number of values that is greater than 5 in cells A1 through A20.
- Now using the **&** concatenation operator and setting the value in cell E20 to 5, **COUNTIF**(A1:A20, ">"&E20) becomes equivalent to **COUNTIF**(A1:A20, ">5"). The advantage of the former is that the value in cell E20 can be arbitrarily changed to suit your purpose. Excel 2007 has a new **COUNTIFS**(*range1, criteria1, range2, criteria2, ...*) function, which is even more powerful.

SUMIF(*criteriaRange, criteria, [sumRange]*)

- Sums the number of values in *sumRange* for all corresponding values in *criteriaRange* that satisfy the criteria. *sumRange* may be omitted if it is the same as the *criteriaRange*.
- The criteria is a simple logical expression expressed (in quotes) as a text. For example, **SUMIF**(A1:A20, ">5") sums the number of values that is greater than 5 in cells A1 through A20.
- Now using the **&** concatenation operator and setting the value in cell E20 to 5, **SUMIF**(A1:A20, ">"&E20, B1:B20) becomes equivalent to **SUMIF**(A1:A20, ">5", B1:B20). The advantage of the former is that the value in cell E20 can be arbitrarily changed to suit your purpose. Excel 2007 has a new **SUMIFS**(*sumRange, criteriaRange1, criteria1, criteriaRange2, criteria2, ...*) function, which is even more powerful.

There are also now new in Excel 2007 the **AVERAGEIF** and **AVERAGEIFS** functions. Their structure and use are similar to those in **COUNTIF** and **COUNTIFS**.

Note 11: DATA MANAGEMENT

Data List

A **data list** is a list of records, with each record occupying a row and its attributes distributed over the columns. Each column corresponds to a data field, which is a variable in a record. *Figure A-12* shows an example of a data list of club membership information.

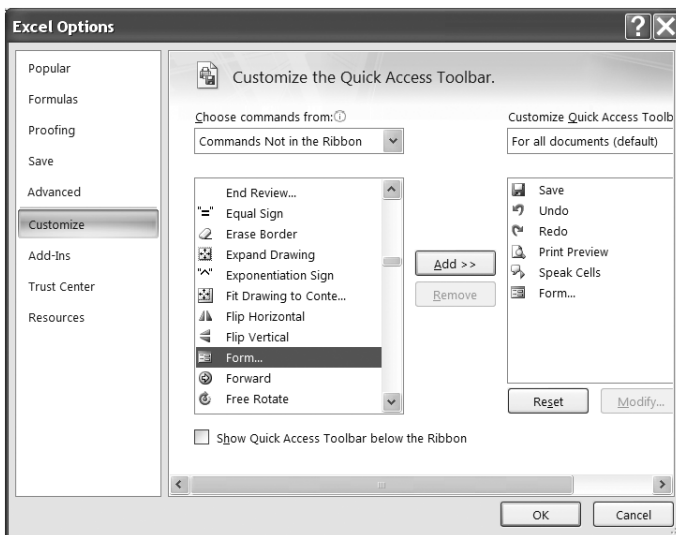
Excel 2003 has provided a quick way for users to enter and edit data in a data list. This **Data Form** feature is also available in Excel 2007 but you have to customize it into your **Quick Access Toolbar** to use it. The **Quick Access Toolbar** is the short toolbar next to the

Figure A-12

Example of a
Data List

	A	B	C	D	E	F	G
1							
2		Membership Number	Name	Address	Gender	Age	Date Joined
3		M00001	John Chew	56 Maryland Street	Male	35	1 Jan 2005
4		M00002	Mary Green	89 Futon Street	Female	33	25 Mar 2005
5		M00003	Michelle Pipe	90 Kronnen Street	Female	29	5 May 2005
6							

Office Button. To do the customization, click the **Office Button/Excel Options/Customize** sequence of steps to reveal the dialog as shown in *Figure A-13a*. Select **Commands Not in the Ribbon** in **Choose commands from** and then scroll down the left selection panel to find **Form**. Click the **Add** button to include the Form icon to the **Quick Access Toolbar**.

**Figure A-13a**

Customizing the
Quick Access
Toolbar

Follow the steps below to create the table in *Figure A-12*.

1. Enter the field labels as table headers into cells B2:G2.
2. Select the field labels in cells B2:G2. An entry form dialog will appear, as shown in *Figure A-13b*.
3. Select from the menu **Data/Form** and an entry form dialog will appear.
4. To add a new record
 - a. Click **New** to clear all current data in the input boxes.
 - b. Enter your information into the form.
 - c. Click **New** again to append your input to the data list, and continue with more new records, or click **Close** to append the data and end.
5. To edit an existing record
 - a. Locate the record using **Find Next** and **Find Previous**.
 - b. Edit the information and click **Close** to update.

Figure A-13b

Data Form for
Club Membership
Information

Membership Number	Name	Address	Gender	Age	Date Joined
M00001	John Chew	56 Maryland Street	Male	35	1 Jan 2005
M00002	Mary Green	89 Futon Street	Female	33	25 Mar 2005
M00003	Michelle Pipe	90 Kronnen Street	Female	29	5 May 2005

After experimenting with Data Form, you may notice its several limitations. Here are some we have found:

- No option for adding a drop-down selection list to simplify data entry. This would be particularly useful for fields like *Gender*.
- Does not validate input values. The *Age* field requires an integer input and so a quick check to ensure only valid values within a reasonable range will help ensure data integrity.
- Fixed input box width. A field like *Address* needs a longer input box, whereas *Postal Code* should have a shorter one.

Sort

To sort values in ascending order in a **data list**, put your cursor in a cell in the column of interest inside the table and click the $\frac{A}{Z}\downarrow$ icon. Similarly, you can click on $\frac{Z}{A}\downarrow$ to sort in descending order. A quick way to properly sort a table with data entry that has ties in values in the fields is to move your cursor to the relevant columns to sort and each time click on the sort ascending or sort descending icon. The order in which the columns are sorted is important. You should start with the least significant column, then move progressively to the most significant one.

Alternatively, select all the data you want to **sort** and click on the icon to the right of the $\frac{A}{Z}\downarrow$ and $\frac{Z}{A}\downarrow$ icons to call up the Sort dialog and work from there. If you have instead put your cursor anywhere in the data list before you do so, the whole cell range that contains the data will be also automatically selected and a dialog will appear, as shown in *Figure A-14*. If you indicate that your data list has a header row, the **Sort by** selections will be made according to the header labels. Otherwise, the worksheet column labels will

be used instead. Again, you can choose whether to sort each attribute in ascending or descending order. Unlike Excel 2003 where sorting can be performed by cell values only, Excel 2007 in addition allows sorting to be done on cell color, font color and cell icon.

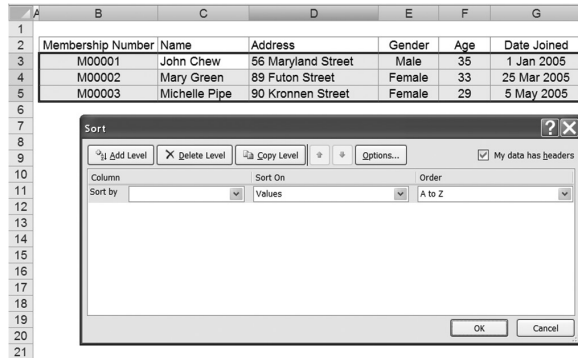


Figure A-14

Data Sort

AutoFilter

AutoFilter allows us to filter a large number of records in a data list for simple user-selected criteria. This feature is particularly useful when the number of records has grown so large that it is no longer possible to view all records at the same time without excessive scrolling. For example, a club with 2,000 members wants to organize a ladies' event and therefore needs to view the data of female members within an age range to work out a reasonable invitation list.

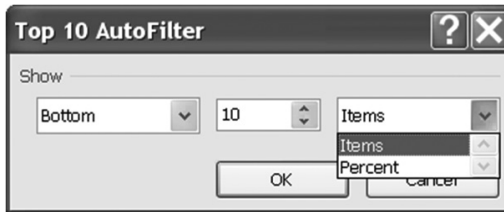
To set up the AutoFilter for *Figure A-12*, follow the steps below.

- Select any cell within the data list.
- Select **Data/Filter/AutoFilter** to activate it. With this, drop-down tabs will be added to the field labels in the data list header for you to select the various filtering operations for each data field.

It will be a good idea to increase the height of the header row and format the cells there with the text alignment set to top so that the drop-down arrows provided by **AutoFilter** will not hide the field names in the header. The filtering operations available are:

- **Sort ascending** – sorts the records so that the data in the selected field are in ascending order.
- **Sort descending** – sorts the records so that the data in the selected field are in descending order.
- **All** – displays all records.
- **Top 10 ...** – displays for the selected field the top or bottom *n* items or percent of the records. (See *Figure A-15* for the dialog.)
- **Custom ...** (See the discussion below.)
- Individual possible entries – show all records of the select entry value in the data field.

Figure A-15
Top 10 AutoFilter



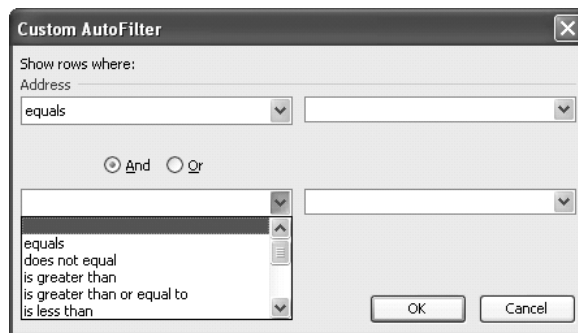
Custom AutoFilter

- Allows you to create a customized filter with up to two criteria, which can be combined by either an **AND** or **OR** logical operator (see *Figure A-16*).
- The criteria include:

<input type="radio"/> Equals	<input type="radio"/> Begins with
<input type="radio"/> Does not equal	<input type="radio"/> Does not begin with
<input type="radio"/> Is greater than	<input type="radio"/> Ends with
<input type="radio"/> Is greater than or equal to	<input type="radio"/> Does not end with
<input type="radio"/> Is less than	<input type="radio"/> Contains
<input type="radio"/> Is less than or equal to	<input type="radio"/> Does not contain

- Allows you to use wildcards such as ? and * to represent any single character or a string of characters, respectively.
- For example, we may want to filter for members whose ages ranged between 30 and 39. You can set the filter for criteria as “**Is greater than or equal to**” 30 **AND** “**Is less than**” 40.

Figure A-16
Custom AutoFilter



Advanced Filter

Consider now that the club would like to remind its members who have joined for at least one year to collect a special gift. The filtering criteria to set up could be to compare

their *Date Joined* with a date that is one year ago from today. If their *Date Joined* is less than or equal to this date (which means it is earlier than or equal to this date), then the member would have joined for at least one year. As this threshold date (one year ago from today) is a value that depends on the date today, you cannot use **Custom Autofilter**. You will have to use **Advanced Filter**.

In this example as shown in *Figure A-17*, you can set up the spreadsheet model as follows:

1. The criteria range is defined in rows 4 and 5, where row 4 contains the selected field labels as a header and row 5 contains the criteria values for the corresponding fields. The field label to use in the filtering is *Date Joined*, and the criteria value is set up as a formula in cell G5 as `= "<=" & TEXT(G2,"d MMM yyyy")`, where cell G2 contains the date for one year ago from today.
2. When the advanced filter is activated, by selecting **Data/Filter/Advanced Filter**, the dialog prompts for:
 - **List Range** – the data list range, here defined in B8:G20.
 - **Criteria Range** – the criteria range, here defined in B4:G5.
 - **Action** – the user has the choice to filter the list in its original place by hiding records that do not match the filtering criteria, or to copy the filtered results to another location.
 - **Copy to** – the destination location, if the user selects to copy the filtered results to another location.
 - **Unique records only** – the user checks this check box to hide duplicate records.

	A	B	C	D	E	F	G	H
1								
2		Today		27 Jan 2008		1 year from Today		27 Jan 2009
3								
4		Membership Number	Name	Address	Gender	Age	Date Joined	
5					Male		<= 27 Jan 2007	
6					Female			
7								
8		Membership Number	Name	Address	Gender	Age	Date Joined	
9		M00001	John Chew	56 Maryland Street	Male	35	1 Jan 2005	
10		M00002	Mary Green	89 Futon Street	Female	33	25 Mar 2005	
11		M00003	Michelle Pipe	90 Kronnen Street	Female	29	5 May 2005	
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								
25								

Advanced Filter [?] [X]

Action

Filter the list, in-place

Copy to another location

List range: [F5]

Criteria range: [F5]

Copy to: [F5]

Unique records only

OK Cancel

Figure A-17

Advanced Filter

Here are some rules to follow when using Advanced Filter:

1. There must be at least one empty row separating the criteria range and data list range.
2. There must be at least one field header in the criteria range that matches a field header in the data list range.
3. Criteria field labels can be arranged in any order in their header and they need not be directly above the corresponding field label in the data list header.
4. An empty criteria field value implies no match restriction for the field.
5. All field entries in the criteria range are combined within a row by the **AND** operator, and criteria of multiple rows are combined across rows by the **OR** operator. For example, in the criteria table in cells B4:G6 in *Figure A-17*, we have two filtering criteria rows, where the first criterion row of male *Gender* member **AND** *Date Joined* earlier than or at 27 January 2007 is **OR** to the second criterion of any member who is of the female *Gender*. All records that match the second set of criteria will be appended to records that match the first set of criteria.
6. Do not include an empty criteria row unless your intent is for Advanced Filter to show all records in the data list.

Database Functions

Filtering data in a **data list** hides some of the data. When you use functions such as **COUNT**, **SUM**, **AVERAGE**, and **STDEV**, the statistics they provide include all the values, including those that have been filtered out.

Like **COUNTIF** and **SUMIF** types of functions, there are also **DAVERAGE**, **DMAX** and other database functions that allow you to compute the results of the data under the specified field (i.e., the header label or equivalent, the column position in the table) that complies with the criteria you give, as is done in **Advanced Filter**. In all these functions, the data in the table are all visible to the user. To filter and then compute the results for the filtered data, you have to use the **SUBTOTAL** function.

SUBTOTAL(*functionNum*, *range1*, *range2*, ...)

Depending on the value you specify for *functionNum*, **SUBTOTAL** returns a statistics of the values in cells *range1*, *range2*, etc. These ranges have to be columns and you can have up to 254 ranges. If the cells are in a data list where the data may be filtered, the statistics you collect can be made to either include or ignore the hidden values as given by *functionNum*. *functionNum* constants ranging from number 1 to 11 will include hidden values and from 101 to 111 will ignore hidden values, as follows:

Function Number (includes hidden values)	Function Number (ignores hidden values)	Function
1	101	AVERAGE
2	102	COUNT
3	103	COUNTA
4	104	MAX
5	105	MIN
6	106	PRODUCT
7	107	STDEV
8	108	STDEVP
9	109	SUM
10	110	VAR
11	111	VARP

Rows or columns you hide using the **Row Hide** or **Column Hide** format commands are not considered by SUBTOTAL as hidden.

Pivot Table

PivotTable and **PivotChart** help us to create table and chart reports with automatic computation of certain results of interest, such as sum and average, and organize the results according to different attributes. The attributes can be in rows, columns, or pages. Consider the sales performance of a small flower shop for three months as shown in *Figure A-18*. There, sales figure for different product and customer types are given for the months of January to March. This information would be better organized if January, February, and March are themselves headers. This argument extends to using the individual product types as headers and customer types as headers. The only data entries in the example are really the sales figures. This would be a 3-dimensional table. PivotTable can take the given data to automatically create a table such as the one shown in *Figure A-20*.

To create a PivotTable, put your cursor anywhere in the data list and then select **Insert/PivotTable** or **Insert/PivotChart**:

1. Select **PivotTable** from the drop-down list.
2. Select **Table/Range** in the dialog as shown in *Figure A-19a* and enter the cell reference for the data. If your cursor is in the data list before you proceed with step 1, then it

Figure A-18Data for
PivotTable

	A	B	C	D	E
1					
2		Month	Product Type	Customer Type	Sales
3		January	Flowers	Walk-in	\$1,000
4		January	Flowers	Corporate	\$2,000
5		January	Gifts	Walk-in	\$800
6		January	Gifts	Corporate	\$500
7		February	Flowers	Walk-in	\$1,500
8		February	Flowers	Corporate	\$3,000
9		February	Gifts	Walk-in	\$1,000
10		February	Gifts	Corporate	\$900
11		March	Flowers	Walk-in	\$1,200
12		March	Flowers	Corporate	\$2,300
13		March	Gifts	Walk-in	\$2,000
14		March	Gifts	Corporate	\$1,100

will be automatically selected. You can alternatively choose an external data source and then specify the connection details.

3. A layout template will then be shown, according to your selection, on either the worksheet or a new worksheet with the **PivotTable Field List** on the extreme right, as shown in *Figure A-19b*.
 - The field labels from the data source are listed as buttons on the top of the field list.
 - Drag these labels, one at a time, into the Report Filter, Row, Column, or Values areas according to your preference. In the example shown in *Figure A-19c*, we put *Month* in **Report Filter**, *Product* in **Row**, *Customer* in **Column**, and *Sales* in **Values**.
 - Excel by default selects **Sum of Sales** as the method to compute the *Sales* values in the Values area. Click on the arrow next to **Sum of Sales** and then select **Value Field Settings** to show another dialog for you to change the computation method. The options available include **Count**, **Average**, **Max**, and **Min**, as shown in *Figure A-19d*. Click **OK** when done.
 - If you prefer, you can change the PivotTable to the classic layout used in Excel 2003. This will allow you to drag and drop the fields directly into the PivotTable itself and also allow more direct changes to the data computation method. To do this, right-click on any cell in the PivotTable and select **PivotTable Options**. In the dialog as shown in *Figure A-19e*, click the **Display** tab and then check **Classic PivotTable layout**. The PivotTable created is shown in *Figure A-20*.
 - To add a **PivotChart** after a PivotTable is done, put your cursor anywhere in the data list and then select **Options/PivotChart** from the menu. The PivotChart created is shown in *Figure A-21*.

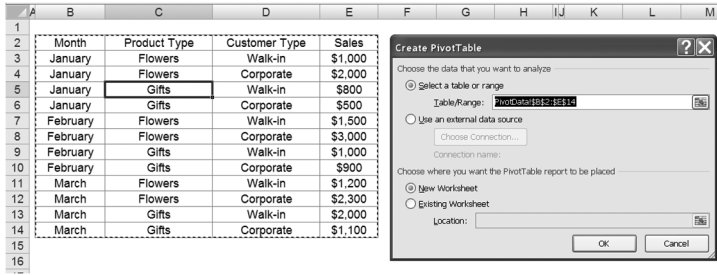


Figure A-19a
Create PivotTable

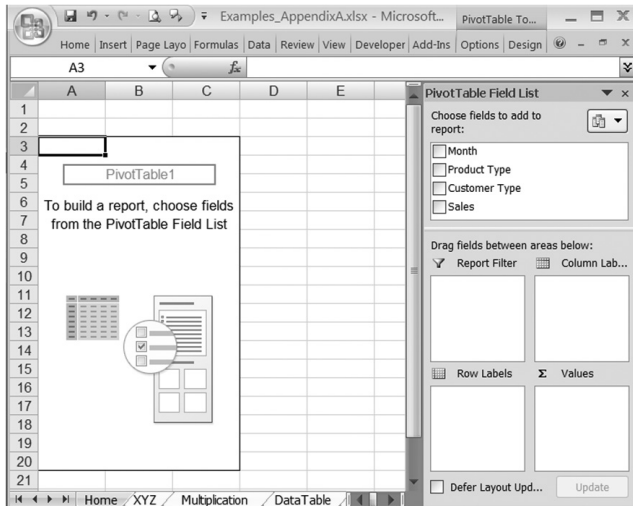


Figure A-19b
PivotTable Field List

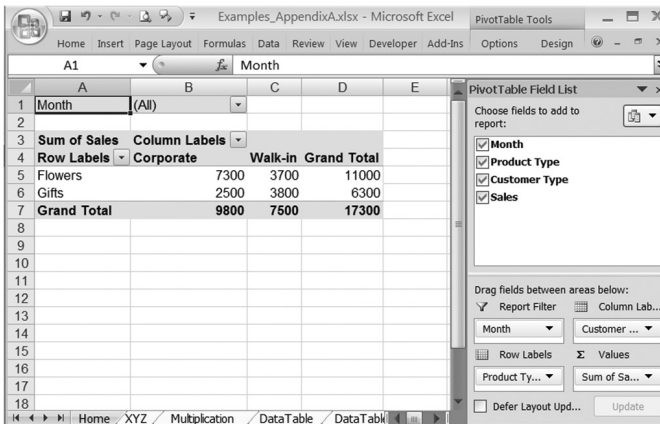


Figure A-19c
Layout for PivotTable

Figure A-19d

Changing the
Data Field
Computation

**Figure A-19e**

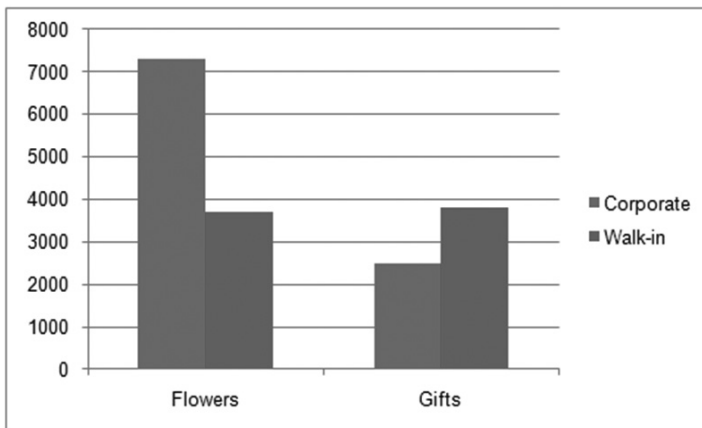
Changing to
Classic Layout



	A	B	C	D
1	Month	(All)		
2				
3	Sum of Sales	Customer Type		
4	Product Type	Corporate	Walk-in	Grand Total
5	Flowers	7300	3700	11000
6	Gifts	2500	3800	6300
7	Grand Total	9800	7500	17300

Figure A-20

PivotTable

**Figure A-21**

PivotChart

Put your cursor anywhere in your PivotTable and right-click your mouse. The pop-up side menu gives you the options to refresh the data, change refresh intervals, set up column and row totals, and others. You can try working with PivotTables in the *Grand Grocery* exercise in Chapter 8.

Note 12: TARGET AND OPTIMIZATION

Goal Seek

We are all too familiar with computing outputs for given inputs. **Goal Seek** is a spreadsheet operation that allows us to work backwards to find the value of a single input variable that gives the desired value for an output variable. The input variable cell contains only a value, while the output variable cell contains a formula. This

inverse operation is done in the background by Excel using iterative calculations, much like what one would do by trial and error but smarter. You therefore do not have to change the formulas in the cells to achieve this. You can access the operation by selecting from the main menu **Data/What-If Analysis/Goal Seek**.

Consider the same flower shop incurring a fixed cost of \$100 and a variable cost of \$5 for each flower bouquet sold. Selling price for each bouquet is \$8. How many bouquets must this shop sell to break even? To answer this question, let us first list the key relationships among the variables:

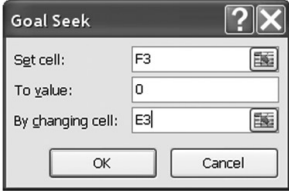
- Profit = Total Revenue – Total Cost
- Total Revenue = Unit Price * Number of Units Sold
- Total Cost = Fixed Cost + (Unit Cost * Number of Units Sold)
- To break even, Total Revenue = Total Cost or Profit = 0

As shown in *Figure A-22*, to find the breakeven price, you can use Goal Seek to set the profit in cell F3 to a target value of 0, by changing the number of bouquets sold in cell E3. Goal Seek will compute and leave the breakeven price in cell E3. We could have in this case easily manipulated the equations to determine the solution directly. However, there will be many other cases where this is not so easy. Even when it is, maintaining two sets of formulas, one in the forward direction and another in reverse, is troublesome.

It is important to note that the underlying Goal Seek algorithm may not converge in its search. Another complication is that there may be more than one solution for the problem posed. The input variable initial value therefore needs to be set appropriately so that the search may end within a reasonable time and find the desired result. To set limits on the search, in case it does not converge fast enough if at all, click **Office/Excel Options/Formulas**. In the **Calculation** panel, set the maximum iterations and maximum change parameter values. If the number of iterations taken by Goal Seek exceeds the specified maximum iterations, the search ends without a solution.

The algorithm ends successfully when the difference between the results of two consecutive iterations is less than the specified maximum change. The algorithm will converge faster when a larger maximum change parameter value is used, at the expense of lower precision in the solution value. You do not normally need to change the default values. These parameters are also applicable to **Iterative Calculations**, which is a topic explored in the *Charity Donation* exercise in Chapter 4.

	A	B	C	D	E	F
1						
2		Fixed Cost	Variable Cost	Unit Price	Units Sold	Profit
3		\$100	\$5	\$8	10	-\$70
4						
5						
6						
7						
8						
9						
10						
11						
12						



The Goal Seek dialog box is open, showing the following settings:

- Set cell: F3
- To value: 0
- By changing cell: E3

Buttons for OK and Cancel are visible at the bottom.

Figure A-22

Goal Seek

Solver

Solver is available as a standard **Add-in** in Excel. It can do the operation of **Goal Seek** and more. First, you can use it to search for the values of many input variables that give the desired value for an output variable. This is an improvement over Goal Seek that permits only one input variable. Next, Solver allows you to find the maximum or minimum value of an output variable (commonly referred to as the *objective function*) subject to the constraints you specify. These constraints can include lower and upper limits on individual input values or their combinations (e.g., budget constraints). Solver only becomes available in Excel after you activate it as an add-in. You do this by checking **Solver Add-in** option in **Office/Excel Options/Add-ins/Excel Add-ins/Go**.

Solver uses mathematical programming methods to determine the solution. These include linear programming, nonlinear programming, integer programming, and combinatorial optimization. In all such optimizations, there are five possible result outcomes:

- The solution found is the unique optimal solution.
- The solution found is one of the many local optimal solutions.
- The solution found is a point of inflection, i.e., neither a minima nor a maxima.
- The minimum (maximum) solution found is $-\infty$ ($+\infty$).
- There are no feasible solutions.

Consider a simple problem faced by a machine shop, as shown in *Figure A-23*. This machine shop is planning to acquire two types of machines. Machine type 1 costs \$100 and has a production capacity of 20 units, while machine type 2 costs \$150 and has a production capacity of 30 units. If the budget for acquiring these machines is \$30,000 and the ratio of

machine type 1 to machine type 2 must be less than or equal to 0.75, how many of each machine type should the machine shop purchase in order to maximize its total production capacity?

Figure A-23

Solver

	A	B	C	D	E	F	G	H	I	J
1										
2		Objective Function	80							
3										
4		Parameters								
5		P1	20							
6		P2	30							
7		C1	\$100							
8		C2	\$150							
9										
10		Decision Variables								
11		X1	1							
12		X2	2							
13										
14		Constraints								
15		LHS	Sign	RHS						
16		0.50	<=	0.75						Ratio of Type 1 and Type 2 equipment <= 0.75.
17		\$400	<=	\$30,000						Total investment <= Budget of \$30,000.
18										

To solve this problem using Solver, we can set up the Excel model as follows:

1. Objective function
 - This equation defines the objective of the problem, which is to be minimized or maximized.
 - In this example, cell C2 with the formula $=C5*C11+C6*C12$ has the *objective function*, computing the total production capacity for the given number of machines. This cell value is to be maximized.
2. Decision variables
 - These input variables of the model can be binary, integer, or real values.
 - In this example, they are variables X1 and X2 in cells C11 and C12, for the number of machine type 1 and type 2, respectively.
3. Constraints
 - These equations specify their left-hand side to be \leq , \geq , or $=$ to their right-hand side. The right-hand side is usually a resource limit value.
 - Before you can enter constraints into Solver, you need to set up the left-hand-side and right-hand-side cells to store their values or formulas.
 - In the above example, there are two constraints. The first is the ratio of machine constraint, where the left-hand-side cell B16 has the formula $=C11/C12$ and the right-hand-side cell D16 has the value 0.75.

- The second is the budget constraint, where the left-hand-side cell B17 has the formula $=C7*C11+C8*C12$ and the right-hand-side cell D17 has the value \$30,000.

Once the model is set up, click **Data/Analysis/Solver**, to activate the **Solver Parameters** dialog as shown in *Figure A-24* and make the following selections:

1. Set **Target Cell** to C2.
2. Select **Max** as the objective criterion.
3. Enter **By Changing Cells** as C11:C12.
4. Add constraint B16:B17 \leq D16:D17.
5. Click **Solve**.

If the problem is well defined, the initial values of the decision variables in **By Changing Cells** are reasonable, and the number of decision variables and constraints are not exceedingly large, an optimal solution will soon be found.

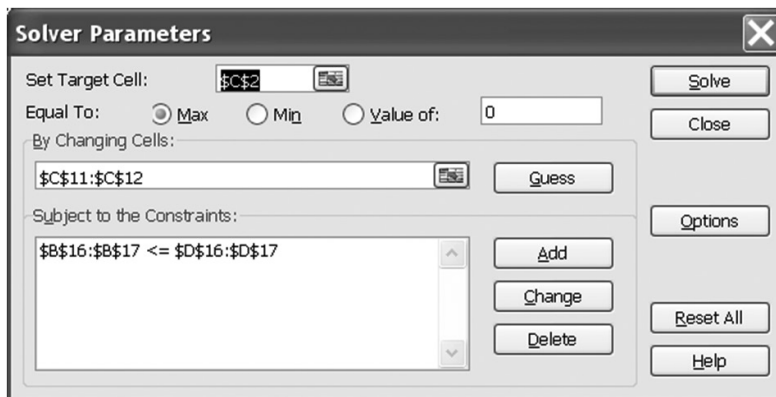


Figure A-24

Solver
Parameters
Dialog

The termination parameters for the number of iterations and maximum change can be found by clicking **Options** in the **Solver Parameters** dialog. You do not normally need to change their default values.

Note 13: DATA ANALYSIS

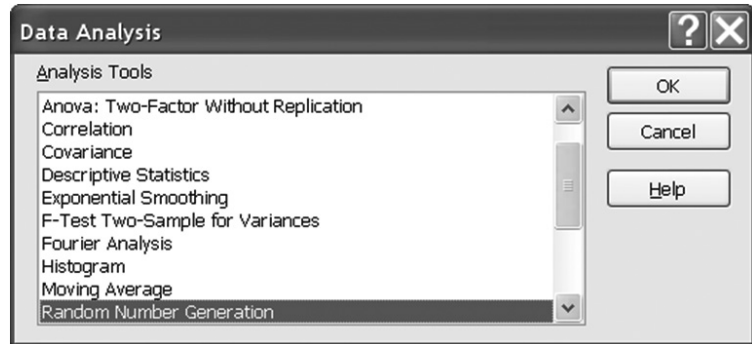
Analysis Toolpak

Analysis Toolpak is a standard **Add-in** in Excel that supports statistical data analysis. The analysis tools available as shown in *Figure A-25* include random number generation, frequency tabulation, test of hypothesis, and analysis of variance. Analysis Toolpak only becomes available in Excel after you activate it as an add-in. You do

this by selecting **Analysis Toolpak** in **Office/Excel Options/Add-ins/Excel Add-ins/Go**. To access it, select **Data/Data Analysis**. The dialog that appears will be as shown in *Figure A-25*.

Figure A-25

Data Analysis
Dialog



These tools are however computer programs that will take data from your worksheet for their computation and then post the required results back into the worksheet as values. As such, they are static tools and do not work interactively with the rest of the spreadsheet in a dynamic manner. Updates in cell values from changes in data inputs and formula recalculations will not be automatically recomputed by these analysis tools. To compute more dynamic results, you need to use the equivalent Excel functions when they exist. More functions have been added by Excel in future revisions or you may search for add-ins that can provide these functions.

Statistical Functions

The functions in Excel that currently support statistical analysis include **AVERAGE**, **STDEV**, **MIN**, **MAX**, **PERCENTILE**, **RANK**, **INTERCEPT**, **SLOPE**, **TREND**, **GROWTH**, **RSQ**, **NORMDIST**, **EXPONDIST**, **FDIST**, **CORREL**, **CHITEST**, **CONFIDENCE**, **ZTEST**, **TTEST**, **FTEST**, and **LINEST**. You can call up these functions by clicking the *fx* button in the **Formula** bar and select the **Statistical** category. You can refer to **Help** in Excel to examine in detail how they work.

The most complex function in this list is probably **LINEST**. Array function **LINEST**(*knownY's*, *knownX's*, [*const*], [*stats*]) returns the statistics that describe the *least squares method* for fitting a straight line to the given data set comprising *knownY's* and *knownX's*. It is called an array function because it returns results, not in a single cell but, in a range of cells. If the optional variable *const* is **TRUE** or omitted, the intercept value is computed. Otherwise, the intercept value is assumed to be 0. If the optional *stats* is **TRUE**, four additional rows of statistics are computed. If **FALSE** or omitted, then only the coefficients and the intercept of the fitted straight line are computed.

The example in *Figure A-26* illustrates the use of **LINEST**. The formula for range H3:K7 is `{=LINEST(E3:E14,B3:D14,,TRUE)}`. To key in the formula, first select all the cells that are to display the results. If *stats* is **FALSE** or omitted, this means one row and number of *X*'s plus 1 (for the *Y*) columns are to be selected. If *stats* is **TRUE**, this means 5 rows and number of *X*'s plus 1 (for the *Y*) columns are to be selected. Key the formula `=LINEST(E3:E14,B3:D14,,TRUE)`, without the { } brackets and then key **Shift + Ctrl + Enter**. Brackets { } are provided automatically by Excel to indicate that it is an array function (that is, with Shift + Ctrl + Enter applied).

It is interesting to note that the **DataTable** operation is depicted by Excel as an array function. If you put your cursor in any result cell in a **DataTable**, you will find displayed in the **Formula** bar the formula `{=TABLE(rowInputCell, colInputCell)}`, with the distinctive { } brackets. It is however not possible to invoke the **DataTable** as a function, just as it can be done for **LINEST**. The only way to call a **DataTable** is to use the Excel menu and enter inputs into the **DataTable** dialog.

	A	B	C	D	E	F	G	H	I	J	K	L
1												
2		X1	X2	X3	Y							
3		1	1	1	\$1,000							
4		1	1	2	\$2,000							
5		1	2	1	\$800							
6		1	2	2	\$500							
7		2	1	1	\$1,500							
8		2	1	2	\$3,000							
9		2	2	1	\$1,000							
10		2	2	2	\$900							
11		3	1	1	\$1,200							
12		3	1	2	\$2,300							
13		3	2	1	\$2,000							
14		3	2	2	\$1,100							
15												

	Coeff3	Coeff2	Coeff1	Intercept	
Slope =	383.33	-783.33	287.50	1466.67	= Intercept
Std Error =	353.92	353.92	216.73	884.80	= Std Error of Intercept
R ² =	0.4947	613.01	#N/A	#N/A	= Std Error of MSFT
F stats =	2.6105	8	#N/A	#N/A	= Degree of Freedom
SS _{res} =	3E+06	3E+06	#N/A	#N/A	= SSE = Residual sum of errors

Figure A-26
LINEST Statistics

Array Formulas

Array formulas do not have to involve array functions. They all have the distinctive { } brackets, which arise after you type in the formula with **Shift + Ctrl + Enter**, instead of the usual **Enter** key. Many interesting computations can be done as array formulas and they usually give spreadsheet models that are more compact. If you understand array formulas, such models may in fact be easier to comprehend. *Figure A-27* shows two ways for computing the errors of forecasting variable *Y* using a linearly fitted equation given by the coefficients and intercept values from *Figure A-26*.

Cell G17 gives the average absolute deviations, comparing each value in cells E3:E14 with the corresponding forecast in G3:G14. Similarly, G18 gives the maximum absolute deviation. The same results can be found in cells G21 and G22. The formula for cell G21 is `{=AVERAGE(ABS(E3:E14-G3:G14))}` and for cell G22 is `{=MAX(ABS(E3:E14-G3:G14))}`. These formulas do not reference the values in cells H3:H14 and therefore do not need them to be computed if these array formulas are used instead of those in cells G17 and G18.

Even though both G21 and G22 are single cells, array formulas have to be used because of the calculations involved. Let us examine these formulas in detail. The first computation step in these formulas is $\$E\$3:\$E\$14-\$G\$3:\$G\14 , which means $\$E\$3-\$G\3 , $\$E\$4-\$G\4 , ..., and $\$E\$14-\$G\14 . These 12 values form an array, on which we next apply the **ABS** function to find absolute differences. The results are collectively still an array. The functions **AVERAGE** or **MAX** are then applied on this array. In general, array formulas involve array subcomputation results or array functions and can, but do not necessarily need to, have array results.

Figure A-27

Array Formulas

	A	B	C	D	E	F	G	H	I
1									
2		X1	X2	X3	Y		Forecast	Absolute Deviation	
3		1	1	1	\$1,000		\$1,354	\$354	
4		1	1	2	\$2,000		\$1,738	\$263	
5		1	2	1	\$800		\$571	\$229	
6		1	2	2	\$500		\$954	\$454	
7		2	1	1	\$1,500		\$1,642	\$142	
8		2	1	2	\$3,000		\$2,025	\$975	
9		2	2	1	\$1,000		\$858	\$142	
10		2	2	2	\$900		\$1,242	\$342	
11		3	1	1	\$1,200		\$1,929	\$729	
12		3	1	2	\$2,300		\$2,313	\$13	
13		3	2	1	\$2,000		\$1,146	\$854	
14		3	2	2	\$1,100		\$1,529	\$429	
15									
16									
17							Mean Absolute Deviation	\$410	=AVERAGE(\$H\$3:\$H\$14)
18							Maximum Absolute Deviation	\$975	=MAX(\$H\$3:\$H\$14)
19									
20									
21									
22									
23									

NOTE 14: EXCEL 2007 VS EXCEL 2003

In case you are still using Excel 2003 or older versions of Excel and considering migrating to Excel 2007, here is a quick summary comparing Excel 2007 to Excel 2003. First, the worksheet in Excel 2007 is now much larger with 1,048,576 rows and 16,384 columns (A to XFD). It can handle more than the 1 GB memory limit in Excel 2003, constrained only by the memory allocated by your computer operating system.

Other than larger memory, some features are also extended. For example, cells can now have **Conditional Formatting** beyond the previous three conditions. There are many more new and easy-to-use features like **Data Bars**, **Color Scales**, and **Icon Sets**. There is also now a **Rules Manager** that allows you to manage the conditional formatting of the cells in the worksheet all at once. The **IF** function can now be nested up to 64 levels, much more than the 7-level limit in Excel 2003 and earlier versions. **COUNTIF**, which allows only one criteria range and criteria pair, has an extended

version called **COUNTIFS**. This new function can cater up to 127 criteria ranges and criteria pairs. There are also the **SUMIFS**, **AVERAGEIF**, **AVERAGEIFS**, and other new functions.

The main difference between the Excel versions is the menu system, which has been reorganized. The **Office Button** in the extreme top-left corner now provides access to **New**, **Open**, **Save**, **Print**, and other file management tools. Author's name and organization information are entered in **Prepare** under **Properties**. Note that **F1** is still the **Help** key. **Help** can also be accessed by clicking the ? icon in the top-right corner of the menu. To the right of **Office Button** is the **Quick Access** toolbar, where you can put the icons of the tools you frequently use but are not available in the main menu. Placed there by default are **Save**, **Undo**, and **Redo**.

Instead of vertical drop-down menus to the main menu items as in Excel 2003, Excel 2007 has made the main menu items into tabs in the **Ribbon**, which organize the tools in groups horizontally. The **Home** and **Page Layout** tabs, the first and third tabs, cover most of the cell and worksheet formatting work. In the bottom-right corners of these groups is a tiny diagonal down-right arrow icon. Clicking this icon will bring up the Excel 2007 **Format Cells** dialog, which looks exactly like the one in Excel 2003.

The second **Insert** tab has the **PivotTable**, **Pictures**, **Charts**, **Hyperlinks**, and **Text** groups of tools, all largely related graphics features. **Formula** is a whole menu item (tab) by itself, focusing mostly on cell calculation features. Listed here are also the **Define Names**, **Formula Auditing**, and **Calculation** tools. Data access from external sources and **Data Sort**, **Filter**, **Validation**, and **Analysis** tools are in the **Data** tab.

In the **Data Tools** group of the **Data** tab, there is a **What-If Analysis** button. **Scenario Manager**, **Goal Seek**, and **Data Table** now reside somewhat obscurely here. **Solver** is now grouped with **Data Analysis Tools** in the **Analysis** group, the last in this tab. This group is only present if it had first been added. To add **Solver**, click the **Office Button**, then **Excel Options** (in the bottom-right corner) and **Add-ins** option group. In the **Manage** box, select **Excel Add-ins** and click **Go**. Check **Solver**. You might as well at the same time check **Analysis Toolpak**, and then click **OK**.

Comments and **Protection** features are now under the **Review** tab. **Custom Views**, worksheet gridlines and header options, window split, freeze and zoom, and access to macros are now in the **View** tab. The last tab, the **Add-Ins** tab, has property tools. Less regular features are no longer in toolbars and have to be added singly. For example, to use text-to-speech commands, you need to get to **Quick Access Toolbar**. To do this, click the **Customize Quick Access Toolbar** icon, which is just next to **Quick Access Toolbar**. Click **More Commands**. In the **Choose** commands from list, select **All Commands**. Scroll down to find **Speak Cells**, click **Add**, and click **OK**.

Many of the option settings previously found in **Tools/Options** of the main menu in Excel 2003 are now more remotely placed in **Office Button**. The settings there have to be

initialized to your preference, at least once after installing Excel. To do this in Excel 2007, click **Office Button**, and then click **Excel Options** in the bottom-right corner of the pop-up panel. In the dialog, click **Popular**. You will want to ensure the **Show Developer tab** option is checked. This makes the macro recording and forms features more accessible. Click **OK**.

Now, review other option groups carefully, one at a time. Here are some more suggestions. In **Formula**, set workbook calculation to **Automatic** and disable **Iterative Calculation**. In **Proofing**, set the language dictionary to your preferred language. In **Save**, set your preferred **Auto Recovery** time intervals and file locations. Next in **Trust**, click **Trust Center Settings** and then **Macro Settings**. Select the second option of disabling all macros with notifications.

Finally, you can open Excel 2003 files (named with extensions **xls**, **xlt**, and **xla**) in Excel 2007. After working on them, Excel 2007 will by default still save them in their original format. A new workbook started in Excel 2007 will on the other hand by default be saved in the Excel 2007 format (named with extensions **xlsx**, **xlsm**, **xltx**, **xltm**, and **xlam**). It can still be **Save As** into a file of one of the earlier formats. You may encounter some incompatibility difficulties working in Excel 2007 when the workbooks are in Excel 2003 format. A particular one to note is that when you use Excel 2007 features that do not exist in Excel 2003, the changes may not be saved. Therefore, you will soon want to convert your current Excel files to the new Excel 2007 format.

A special point to note in the file format migration is that while all **xls** files can be **Saved As** to become **xlsx** files, those with macros will have their macros stripped away. To retain the codes, you will have to convert workbooks with macros into **xlsm** files, with **m** for macro. Similarly, Excel template files with macros should be saved as **xltm** files. By definition, all Excel **Add-in** files have macros and thus they have to be **xlam** files. This thus explains the larger collection of new extensions in Excel 2007.

Note 15: MIGRATING TO LIBREOFFICE CALC

Calc is the spreadsheet in LibreOffice (LO). LibreOffice branched from OpenOffice.org (OOo; now renamed OpenOffice or OO in short), and thus retains most of its features and functionalities. It has a user interface practically the same as that of Excel 2003:

Here are a few minor differences between Excel and LO Calc 4:

- **Align, Group, Flip**, and other graphics manipulation tools found in Excel 2003's **Drawing Toolbar** are in the **Format** menu. **Snap to grid** is found in **Tools/Options/LibreOffice Calc/Grid**.
- **Conditional Formatting** in Calc is similar to Excel 2003's. Calc uses **Formatting Styles** and you need to create the style if you are not using the standard ones.

- Arguments in Excel functions are separated by commas; the ones in OO Calc 3 are separated by semi-colons. LO Calc 4 allows you to set the separator, which by default is now a comma.
- **Formula Auditing** is called **Detective** in Calc.
- **PivotTable** in Excel is called **DataPilot** in OO Calc3. In LO Calc 4, it has the same name as in Excel.
- **Solver** is directly available in Calc, but the tool is possibly not as rich or mature as the **add-in** in Excel.
- The equivalent **macro** language in Calc is LibreOffice **Basic** (updated from **StarBasic and OOO Basic**) that, like **Visual Basic** (VB) used in Excel, is derived from the same underlying **BASIC** programming language. It however uses a different object library to access spreadsheet components. **VBA** codes are ported over automatically when Excel files are opened as LO Basic codes in Calc.
- LibreOffice will insert an additional `Option VBASupport 1` line at the start of each module. This enables your Excel VBA codes to run in the LO Basic environment. However, not all the spreadsheet features can be manipulated in LO Basic yet. The good news is that **Solver** can be called in LO Basic to automate model optimization.
- To give a sense of the differences in the macro codes, here is an example of the same program written in the two languages:

```
Sub VBA_Example()  
    Range("B3").Select  
    Range("B3") = "Hello"  
End Sub
```

```
Sub LOBasic_Example()  
    oSheet = ThisComponent.CurrentController.ActiveSheet  
    oCell = oSheet.getCellRangeByName("B3")  
    ThisComponent.CurrentController.select(oCell)  
    oCell.String = "Hello"  
End Sub
```