

Advanced Excel Functions and Procedures

The purpose of this chapter is to review certain Excel functions and procedures used in the text. These include mathematical, statistical and lookup functions from Excel's extensive range of functions, as well as much-used procedures such as setting up Data Tables and displaying results in XY charts. Also included are methods of summarising data sets, conducting regression analyses, and accessing Excel's Goal Seek and Solver. The objective is to clarify and ensure that this material causes the reader no difficulty. The advanced Excel user may wish to skim the content or use the chapter for further reference as and when required. To make the various topics more entertaining and more interactive, a workbook AMFEXCEL.xls includes the examples discussed in the text and allows the reader to check his or her proficiency.

2.1 ACCESSING FUNCTIONS IN EXCEL

Excel provides many worksheet functions, which are essentially calculation routines that have been coded up. They are useful for simplifying calculations performed in the spreadsheet, and also for combining into VBA macros and user-defined functions (topics covered in Chapters 3 and 4).

The Paste Function button (labelled *fx*) on the standard toolbar gives access to them. (It was previously known as the function wizard.) As Figure 2.1 shows, functions are grouped into different categories: mathematical, statistical, logical, lookup and reference, etc.

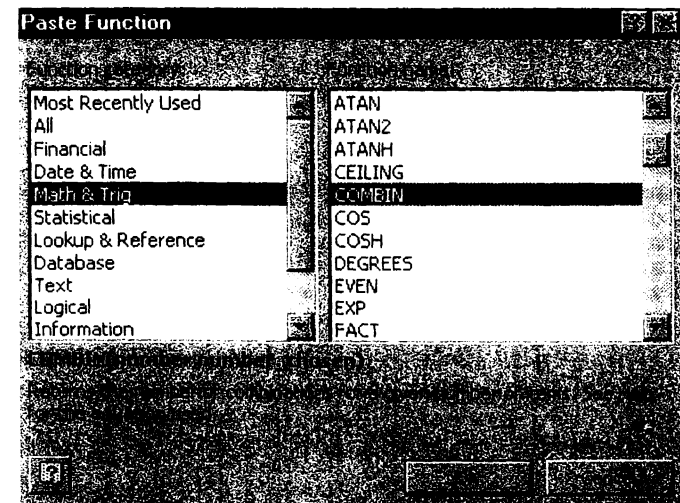


Figure 2.1 Paste Function dialog box showing the COMBIN function in the Math category

Here the Math & Trig function COMBIN has been selected, which produces a brief description of the function's inputs and outputs. For a fuller description, press the Help button (labelled ?).

On clicking OK, the Formula palette appears providing slots for entering the appropriate inputs, as in Figure 2.2. The required inputs can be keyed into the slots (as here) or 'selected' by referencing cells in the spreadsheet (by clicking the buttons to collapse the Formula palette). Note that the palette can be dragged away from its standard position. Clicking the OK button on the palette or the tick on the Edit line enters the formula in the spreadsheet.

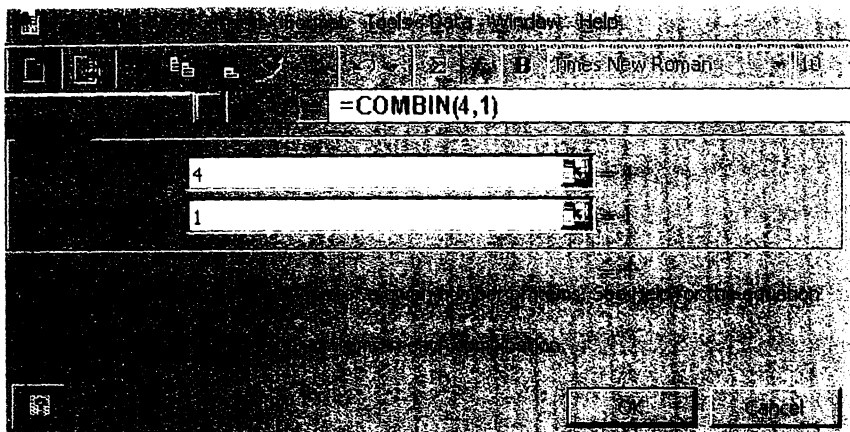


Figure 2.2 Building the COMBIN function in the Formula palette

As well as the Formula palette with inputs for function COMBIN, Figure 2.2 shows the construction of the cell formula on the Edit line, with the Paste Function button depressed (in action). Notice also the Paste Name button (labelled =ab) which facilitates pasting of named cells into the formula. (Attaching names to ranges and referencing cell ranges by names is reviewed in section 2.10.)

As well as all Excel functions, the Paste Function button also provides access to the user-defined category of functions which are described in Chapter 4.

Having discussed how to access the functions, in the following sections we describe some specific mathematical and statistical functions.

2.2 MATHEMATICAL FUNCTIONS

Within the Math & Trig category, we make use of the EXP(x), LN(x), SQRT(x), RAND(), FACT(x) and COMBIN(number, number_chosen) functions.

EXP(x) returns values of the exponential function, $\exp(x)$ or e^x . For example:

- EXP(1) returns value of e (2.7183 when formatted to four decimal places)
- EXP(2) returns value of e^2 (7.3891 to four decimal places)
- EXP(-1) returns value of $1/e$ or e^{-1} (0.36788 to five decimal places)

In finance calculations, cash flows occurring at different time periods are converted into future (or present) values by applying compounding (or discounting) factors. With continuous compounding at rate r , the compounding factor for one year is $\exp(r)$, and the equivalent annual interest rate r_a , if compounding were done on an annual basis, is given by the expression:

$$r_a = \exp(r) - 1$$

Continuous compounding and the use of the EXP function is illustrated further in section 2.7.1 on Data Tables.

LN(x) returns the natural logarithm of value x . Note that x must be positive, otherwise the function returns #NUM! for numeric overflow. For example:

- LN(0.36788) returns value -1
- LN(2.7183) returns value 1
- LN(7.3891) returns value 2
- LN(-4) returns value #NUM!

In finance, we frequently work with (natural) log returns, applying the LN function to transform the returns data into log returns.

SQRT(x) returns the square root of value x . Clearly, x must be positive, otherwise the function returns #NUM! for numeric overflow.

RAND() generates a uniformly distributed random number greater than or equal to zero and less than one. It changes each time the spreadsheet recalculates. We can use RAND() to introduce probabilistic variability into Monte Carlo simulation of option values.

FACT(number) returns the factorial of the number, which equals $1*2*3*...*$ number. For example:

- FACT(6) returns the value 720

COMBIN(number, number_chosen) returns the number of combinations (subsets of size 'number_chosen') that can be made up from a 'number' of items. The subsets can be in any internal order. For example, if a share moves either 'up' or 'down' at four discrete times, the number of sequences with three ups (and one down) is:

$$\text{COMBIN}(4,1) = 4 \text{ or equally } \text{COMBIN}(4,3) = 4$$

that is the four sequences 'up-up-up-down', 'up-up-down-up', 'up-down-up-up' and 'down-up-up-up'. In statistical parlance, COMBIN(4, 3) is the number of combinations of three items selected from four and is usually denoted as ${}_4C_3$ (or in general, ${}_nC_r$).

Excel has functions to transpose matrices, to multiply matrices and to invert square matrices. The relevant functions are:

- TRANSPOSE(array) which returns the transpose of an array
- MMULT(array1, array2) which returns the matrix product of two arrays
- MINVERSE(array) which returns the matrix inverse of an array

These fall in the same Math category. Since some readers may need an introduction to matrices before examining the functions, this material has been placed at the end of the chapter (see section 2.13).

2.3 STATISTICAL FUNCTIONS

Excel has several individual functions for quickly summarising the features of a data set (an 'array' in Excel terminology). These include AVERAGE(array) which returns the mean, STDEV(array) for the standard deviation, MAX(array) and MIN(array) which we assume are familiar to the reader.

To obtain the distribution of a moderate sized data set, there are some useful functions that deserve to be better known. For example, the QUARTILE function produces the individual quartile values on the basis of the percentiles of the data set and the FREQUENCY function returns the whole frequency distribution of the data set after grouping.

Excel also provides functions for a range of different theoretical probability distributions, in particular those for the normal distribution: NORMSDIST and NORMSINV for the standard normal with zero mean and standard deviation one; NORMDIST and NORMINV for any normal distribution.

Other useful functions in the statistical category are those for two variables, which provide many individual quantities used in regression and correlation analysis. For example:

- INTERCEPT(known_y's, known_x's)
- SLOPE(known_y's, known_x's)
- RSQ(known_y's, known_x's)
- STEYX(known_y's, known_x's)
- CORREL(array1, array2)
- COVAR(array1, array2)

There is also a little known array function, LINEST(known_y's, known_x's), which returns the essential regression statistics in array form. Most of these functions are examined in more detail in section 2.11 on regression. Their performance is compared and contrasted with the regression output from the Data Analysis Regression procedure.

In the next section, we explain how to use the FREQUENCY, QUARTILE and various normal functions via examples in the Frequency and SNorm sheets of the AMFEXCEL workbook.

2.3.1 Using the Frequency Function

FREQUENCY(data_array, bins_array) counts how often values in a data set occur within specified intervals (or 'bins'), and then returns these frequencies in a vertical array. The bins_array is the set of intervals into which the values are grouped. Since the function returns output in the form of an array, it is necessary to mark out a range of cells in the spreadsheet to receive the output *before* entering the function.

We explain how to use FREQUENCY with an example set out in the Frequency sheet of the AMFEXCEL workbook. As shown in Figure 2.3, monthly returns and log returns (using the LN function) in columns D10:D71 and E10:E70 have been summarised in rows 4 to 7. Suppose the aim is to get the frequency distribution of the log returns (E10:E71), i.e. the so-called 'data_array'. The objective might be to check that these returns are approximately normally distributed. First, we have to decide on intervals (or bins) for grouping the data. Inspection of the maximum and minimum log returns suggests about 10 to 12 intervals in the range -0.16 to $+0.20$. The 'interval' values, which have

been entered in range G5:G14, act as upper limits when the log returns are grouped into the so-called 'bins'.

	A	B	C	D	E	F	G	H	I	J
2	Returns for months 1 - 62									
3	Summary Statistics:			Returns	Ln Returns	Frequency Distribution:				
4		Mean		1.78%	0.014		interval	freq	%freq	%cum freq
5		St Dev		8.09%	0.080		-0.16			
6		Max		21.23%	0.193		-0.12			
7		Min		-14.21%	-0.153		-0.08			
8							-0.04			
9	Month			Returns	Ln Returns		0.00			
10	Feb-92	1		7.06%	0.0682		0.04			
11	Mar-92	2		-11.54%	-0.1226		0.08			
12	Apr-92	3		7.77%	0.0748		0.12			
13	May-92	4		10.66%	0.1013		0.16			
14	Jun-92	5		-11.72%	-0.1247		0.20			
15	Jul-92	6		-8.26%	-0.0862					
16	Aug-92	7		-2.89%	-0.0293					
17	Sep-92	8		9.93%	0.0947		Total			
18	Oct-92	9		12.65%	0.1191					

Figure 2.3 Layout for calculating the frequency distribution of log returns data

To enter the FREQUENCY function correctly, select the range H5:H15. Then start by typing = and clicking on the Paste Function button (labelled *fx*) to complete the function syntax:

=FREQUENCY(E10:E71,G5:G14)

After adding the last bracket ')', with the cursor on Excel's Edit line, enter the function by holding down the **Ctrl** then the **Shift** then the **Enter** keys. (You need to use three fingers, otherwise it will not work. If this fails, keep the output range of cells 'selected', press the Edit key (F2), edit the formula if necessary, then press **Ctrl+Shift+Enter** once again.)

You should now see the function enclosed in curly brackets {} in the cells, and the frequencies array in cells G5:G15. The results are in Figure 2.4. Use the SUM function in cell H17 to check that the frequencies sum to 62.

Interpreting the results, we can see that there were no log returns below -0.16 , six values in the range -0.16 to -0.12 and no values above 0.20. (The bottom cell in the FREQUENCY array, G15, contains any values above the bins' upper limit, 0.20.)

Since the FREQUENCY function has array output, individual cells cannot be changed. If a different number of intervals is required, the current array must be deleted and the function entered again.

It helps to convert the frequencies into percentage frequencies (relative to the size of the data set of 62 values) and then to calculate cumulated percentage frequencies as shown in columns I and J in Figure 2.4. The percentage frequency and cumulative percentage frequency formulas can be examined in the Frequency sheet.

	A	B	C	D	E	F	G	H	I	J
2	Returns for months 1 - 62									
3	Summary Statistics:			Returns	Ln Returns	Frequency Distribution:				
4		Mean	1.78%	0.0145	interval	freq	%freq	%cum freq		
5		St Dev	8.09%	0.0802	-0.16	0	0%	0%		
6		Max	21.23%	0.1925	-0.12	6	10%	10%		
7		Min	-14.21%	-0.1533	-0.08	4	6%	16%		
8					-0.04	7	11%	27%		
9	Month		Returns	Ln Returns		0.00	10	16%	44%	
10	Feb-92	1	7.06%	0.0682	0.04	5	8%	52%		
11	Mar-92	2	-11.54%	-0.1226	0.08	16	26%	77%		
12	Apr-92	3	7.77%	0.0748	0.12	9	15%	92%		
13	May-92	4	10.66%	0.1013	0.16	4	6%	98%		
14	Jun-92	5	-11.72%	-0.1247	0.20	1	2%	100%		
15	Jul-92	6	-8.26%	-0.0862		0	0%	100%		
16	Aug-92	7	-2.89%	-0.0293						
17	Sep-92	8	9.93%	0.0947	Total	62				

Figure 2.4 Frequency distribution of log returns with % frequency and cumulative distributions

The best way to display the percentage cumulative frequencies is an XY chart with data points connected by a smooth line with no markers. To produce a chart like that in Figure 2.5, select ranges G5:G14 and J5:J14 as the source data. Note that, to select non-contiguous ranges, select the first range, then hold down the Ctrl key whilst selecting the second and subsequent ranges.

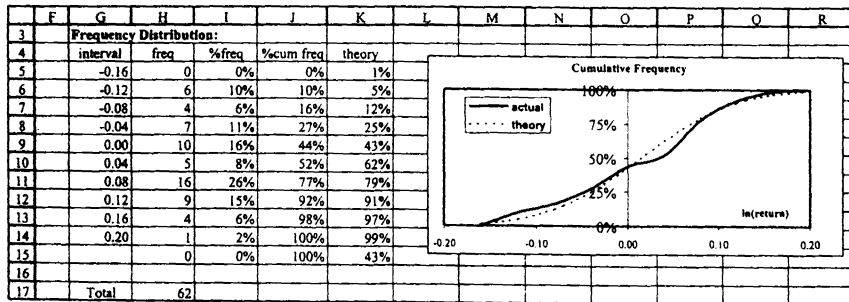


Figure 2.5 Chart of cumulative % frequencies (actual and strictly normal data)

For normally distributed log returns, the cumulative distribution should be sigmoid in shape (as indicated by the dashed line). The actual log returns data shows some departure from normality, possibly due to skewness.

2.3.2 Using the Quartile Function

QUARTILE(array, quart) returns the quartile of a data set. The second input 'quart' is an integer that determines which quartile is returned: if 0, the minimum value of the array; if 1, the first quartile (i.e. the 25th percentile of the array); if 2, the median value (50th percentile); if 3, the third quartile (75th percentile); if 4, the maximum value.

The quartiles provide a quick and relatively easy way to get the cumulative distribution of a data set. For example in cell H22 in Figure 2.6, the entry:

QUARTILE(E10:E71,G22)

where G22 contains the integer value 1, returns the first quartile. The value displayed in the cell is -0.043, which is the log return value below which 25% of the values in the data set fall. The second quartile, 0.028, is the median and the third quartile, 0.075, is the value below which 75% of the values fall. Figure 2.6 also shows an XY chart of the range H21:I25 with the data points marked. The cumulative curve based on just five data points can be seen to be quite close to the more accurate version in Figure 2.5.

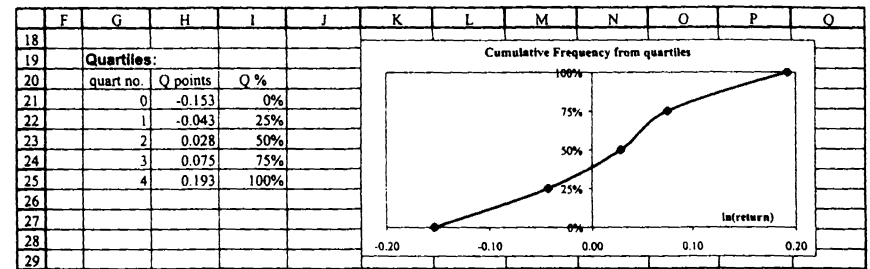


Figure 2.6 Quartiles for the log returns data in the Frequency sheet

The QUARTILE function is used in section 3.5 to illustrate array handling in VBA. A related function, PERCENTILE(array, k) which returns the kth percentile of a data set, is used to illustrate coding an array function in section 4.7.

2.3.3 Using Excel's Normal Functions

Of the statistical functions related to normal distribution, their names all start with the four letters NORM, and some include an S to indicate that the standard normal distribution is assumed.

NORMSDIST(z) returns the cumulative distribution function for the standard normal distribution. NORMSINV(probability) returns values of z for specified probabilities.

The rather more versatile NORMDIST(x, mean, standard_dev, cumulative) applies to any normal distribution. If the 'cumulative' input parameter = 1 (or TRUE), it returns values for the cumulative distribution function; if 'cumulative' input = 0 (or FALSE), it returns the probability density function.

Figure 2.7 shows the Norm sheet, with entries for the probability density and for the left-hand tail probability in cells C5 and D5 respectively. Both these formulas use the general NORMDIST function with mean and standard deviation inputs set to 0 and 1 respectively. In C5, the last input ('cumulative') takes value 0 for the probability density and in D5 takes value 1 for the left-hand tail probability.

The ordinate values corresponding to left-hand tail probabilities can be obtained from the NORMINV function as shown in cell F5.

To familiarise yourself with these functions, copy the formulas down and examine the results.

In the last section, we obtained a cumulative percentage frequency distribution for log returns. One check on normality is to use the NORMDIST function with the observed mean and standard deviation to calculate theoretical percentage frequencies. This has been done in column K of the Frequency sheet. The resulting frequencies are shown in the Figure 2.5 chart, superimposed on the distribution of actual returns. Some departures from normality can be seen.

	A	B	C	D	E	F	G
2	Excel Normal Functions for N(0, 1)						
3							
4			PDF	CDF		Inv(Normal)	
5		-4.00	0.0001	0.0000		-4.00	
6		-3.00					
7		-2.00				=NORMINV(D5,0,1)	
8		-1.00				=NORMDIST(B5,0,1,1)	
9		0.00				=NORMDIST(B5,0,1,0)	
10		1.00					
11		2.00					
12		3.00					
13		4.00					
14							

Figure 2.7 Excel's general normal distribution functions in the SNorm sheet

Excel provides an excellent range of functions for data summary, and for modelling various theoretical distributions. We make considerable use of them in both the Equity and the Options parts of the text.

2.4 LOOKUP FUNCTIONS

In tables of related information, 'lookups' allow items of information to be extracted on the basis of different lookup inputs. For example, in Figure 2.8 we illustrate the use of the VLOOKUP function which for a given volatility value 'looks up' the Black-Scholes call value from a table of volatilities and related call values. (We shall cover the background theory in Chapter 11 on the Black-Scholes formula.)

In general the function:

VLOOKUP(lookup_value, table_array, col_index_num, range_lookup)

searches for a value in the leftmost column of a table (table_array), and then returns a value in the same row from a column you specify (with col_index_num). By default, the first column of the table must be in ascending order (which implies that range_lookup = 1 (or TRUE)). In fact, if this is the case, the last input parameter can be ignored.

Lookup examples are in the LookUp sheet. To check your understanding, use the VLOOKUP function to decide the commission to be paid on different sales amounts, given the commission rates table in cell range F5:G7. Then scroll down to the Black-Scholes Call Value Lookup Table, illustrated in Figure 2.8.

The lookup_value (for volatility) is in C17 (20%), the table array is F17:G27, with volatilities in ascending order and call values in column 2 of the table array. So the

formula in cell D18:

=VLOOKUP(C17,F17:G27,2)

returns a call value of 9.73 for the 20% volatility.

	A	B	C	D	E	F	G	H
15	Black-Scholes Call Value Lookup Table							
16						Volatility	BS Call Value	
17		Volatility	20%			15%	8.63	
18		VLOOKUP				16%	8.84	
19						17%	9.05	
20						18%	9.27	
21		call value	9.73			19%	9.50	
22		MATCH				20%	9.73	
23						21%	9.96	
24						22%	10.19	
25		row	6			23%	10.43	
26		column	2			24%	10.67	
27		INDEX				25%	10.91	
28								

Figure 2.8 Layout for looking up call values given volatility in the LookUp sheet

The lookup_value is matched approximately (or exactly) against values in the first column of the table, a row selected on the basis of match and the entry in the specified column returned. Try experimenting with different volatility values such as 20.5%, 21.5% in cell C17 to see how the lookup function works.

The range_lookup input is a logical value (TRUE or FALSE) which specifies whether you want the function to return exact matches or approximate ones. If TRUE or omitted, an approximate match is returned. If no exact match is found, the next largest value (less than the lookup_value) is returned. If FALSE, then VLOOKUP will find an exact match or return the error value #NA.

There is a related HLOOKUP function that works horizontally, searching for matches across the top row of a table and reading off values from the specified row of the table.

MATCH and INDEX are other lookup functions, also illustrated in Figure 2.8. The function MATCH(lookup_value, lookup_array, match_type) returns the relative position of an item in a single column (or row) array that matches a specified value in a specified order (match_type). Note that the function returns a position within the array, rather than the value itself.

If the match_type input is 0, the function returns the position of an exact match, whatever the array order. If the match_type input is 1, the position of an approximate match is returned, assuming the array is in ascending order. Otherwise, with match_type = -1, the function returns an approximate match assuming that the array is in descending order.

In Figure 2.8, the call values in column G are in ascending order. To find the position in the array that matches value 9.73, the formula in D22 is:

=MATCH(C21,G17:G27,1)

which returns the position 6 in the array G17:G27.

The function INDEX(array, row_num, column_num) returns a value from within an array, the row number and column number having been specified. Thus the row and column numbers in cells C25 and C26 ensure that the INDEX expression in Figure 2.9 returns the value in the sixth row of the second column of the array F17:G27.

	A	B	C	D	E	F	G	H
15	Black-Scholes Call Value Lookup Table				=VLOOKUP(C17,F17:G27,2)			
16						Volatility	BS Call Value	
17		Volatility	20%			15%	8.63	
18		VLOOKUP		9.73		16%	8.84	
19						17%	9.05	
20						18%	9.27	
21		call value	9.73			19%	9.50	
22		MATCH		6		20%	9.73	
23						21%	9.96	
24					=MATCH(C21,G17:G27,1)	22%	10.19	
25		row		6		23%	10.43	
26		column		2		24%	10.67	
27		INDEX		9.73		25%	10.91	
28								
29					=INDEX(F17:G27,C25,C26)			

Figure 2.9 Function formulas and results in the LookUp sheet

If the array is a single column (or a single row), the unnecessary col_num (or row_num) input is left blank. You can experiment with the way INDEX works on such arrays by varying the inputs into the formula in cell D27.

We make use of VLOOKUP, MATCH and INDEX in the Equities part of the book.

2.5 OTHER FUNCTIONS

When developing spreadsheet formulas, as far as possible we try to develop 'general' formulas whose syntax takes care of related but different cases. For example, the cash flow in any year for any of the bonds shown in Figure 2.10 could be zero, could be a coupon payment or could be a principal plus coupon payment.

	A	B	C	D	E	F	G	H	I
2	Bond Cashflows								
3		Bond	Type 1	Type 2	Type 3	Type 4	Type 5		
4		Price	100.0	98.0	95.5	101.0	102.1		
5		Coupon	5	4	3	5	6		
6		Maturity	1	2	2	3	3		
7									
8	Cash Flows for bonds		Type 1	Type 2	Type 3	Type 4	Type 5		
9		Initial Cost	100.0	98.0	95.5	101.0	102.1		
10	Receipts:								
11	Year	1	105						
12	Year	2							
13	Year	3							
14									

Figure 2.10 A general formula with mixed addressing and nested IF functions in Bonds sheet

The IF function gives different outputs for each of two conditions, and a nested IF statement can be constructed to give three outputs (or even more different outputs if

further levels of nesting are constructed). The cash flow formula in cell C11 with one level of nesting:

$$=IF(\$B11 < C\$6, C\$5, IF(\$B11 = C\$6, 100 + C\$5, 0))$$

produces the cash flows for each bond type and in each year when copied through the range C11:H13.

For the type 1 bond, the cash flow depends on the particular year (cell B11) and the bond maturity (C6). If the year is prior to maturity (B11 < C6), the cash flow is a coupon payment C5; if maturity has just been reached (B11 = C6), the cash flow is principal plus coupon (100 + C5); otherwise (B11 > C6), the cash flow is zero. The nested IF takes care of the cash flows when the bond is at (or beyond) maturity, and the first condition in the outer IF takes care of the coupon payments.

The formula is written with 'mixed addressing' to ensure that when copied its cell references change appropriately. We write C\$6 and C\$5 to ensure that when copied down column C, rows 5 and 6 are always referenced for the relevant maturity and premium. However \$B11 will change to \$B12 and \$B13 for the different years. We write \$B11, so that when the formula is copied to column D, column B is still accessed for the year, but C\$5 and C\$6 change to D\$5 and D\$6.

The additional thought required to produce this general formula is more than repaid in terms of the time saved when replicating the results for a large model.

2.6 AUDITING TOOLS

With cell formulas of any complexity, it helps to have the Auditing buttons to hand, i.e. on a visible toolbar. One way to achieve this from the menubar is via View then Toolbars then Customise. With the Customise dialog box on screen as shown in Figure 2.11, tick the Auditing toolbar, which should then appear.

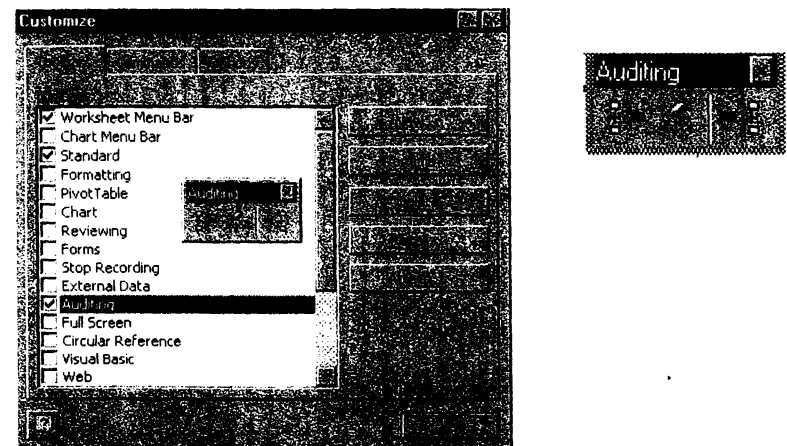


Figure 2.11 Accessing the Auditing toolbar with main buttons

The crucial buttons are those shown in Figure 2.11, namely from left to right, Trace Precedents, Remove All Arrows and Trace Dependents.

Returning to the spreadsheet, select cell C11 and click the Trace Precedents button to show the cells whose values feed into cell C11 as shown in Figure 2.12. (It also shows the cells feeding into F13.) Click Remove All Arrows to clear the lines.

	A	B	C	D	E	F	G
2	Bond Cashflows						
3		Bond	Type 1	Type 2	Type 3	Type 4	Type 5
4		Price	100.0	98.0	95.5	101.0	102.1
5		Coupon	5	4	3	5	6
6		Maturity	1	2	2	3	3
7							
8	Cash Flows for bonds						
9		Initial Cost	100.0	98.0	95.5	101.0	102.1
10	Receipts:						
11	Year	1	105	4	3	5	6
12	Year	2	0	104	103	5	6
13	Year	3	0	0	0	105	106

Figure 2.12 Illustration of Trace Precedents used on the Bonds sheet

The Customise dialog box is where you can tailor toolbars to your liking. If you click the Commands tab, and choose the appropriate category, you can drag tools from the Commands listbox by selecting and dragging buttons onto your toolbars. Conversely, you can remove buttons from your toolbars by selecting and dragging (effectively returning them to the toolbox).

2.7 DATA TABLES

Data Tables help in carrying out a sequence of recalculations of a formula cell without the need to re-enter or copy the formula. The AMFEXCEL workbook contains several examples of Data Tables. We use the calculation of compounding and discounting factors in sheet CompoundDTab to illustrate Data Tables with one input variable, and also with two input variables. A further sheet called BSDTab contains other examples on the use of Data Tables for consolidation.

2.7.1 Setting Up Data Tables with One Input

Figure 2.13 shows the compounding factor for continuous compounding at rate 5% for a period of one year (in cell C10). The equivalent discount factor at rate 5% for one year is in cell D10. The cell formulas for these compounding factors are also shown.

Suppose we want a table of compounding and discounting factors for different time periods, say $t = 1, 2$, up to 10 years. To do this via a Data Table, an appropriate layout is first set up, as shown in row 16 and below.

The formula(s) for recalculation are in the top row of the table (row 16). Thus in D16, the cell formula is simply =C10, which links the cell to the underlying formula in

cell C10. Similarly, in E16, the cell formula is =C11. The required list of times for input value t is in column C starting on the row immediately below the formula row. Notice that cell C16 at the intersection of the formula row and the column of values is left blank. The so-called Table Range for the example in Figure 2.13 is C16:E26.

	A	B	C	D	E	F	G
2	Continuous Compounding						
3	Inputs:						
4	Initial value (a)		1				
5	Interest rate-cont (r)		5.0%				
6	Interest rate-p.a.	$r_a = \exp(r) - 1$	5.1%				
7	Time (t)		1				
8	Outputs:						
9	Compound factor t yrs		1.051				=C\$4*EXP(C\$5*C\$7)
10	Discount factor t yrs		0.951				=C\$4*EXP(-C\$5*C\$7)
11							
12							
13				Enter formula(s) for output			
14							
15				compound	discount		
16				1.051	0.951		
17				1			
18	Enter numbers						
19	for input variable t		2				
20			3				
21			4				
22			5				
23			6				
24			7				
25			8				
26			9				
27			10				

Figure 2.13 Layout for Data Table with one input variable in CompoundDTab sheet

Now the spreadsheet is ready for the Data Table calculations, so simply:

- Select the Table range, that is cell range C16:E26
- From the main menu, choose Data then Table
In dialog box, specify: Column input cell as cell C7
then click OK

The results are shown in Figure 2.14, the cells in the table having been formatted to improve legibility. The table cells display as values but actually contain array formulas. These values are *dynamic*, which means they will be re-evaluated if another assumption value, such as the rate r , changes or if individual values of t are changed. Confirm this by changing the interest rate in cell C5 to 6% and watching the cells re-evaluate. To continue, remember to change the interest rate back to 5%.

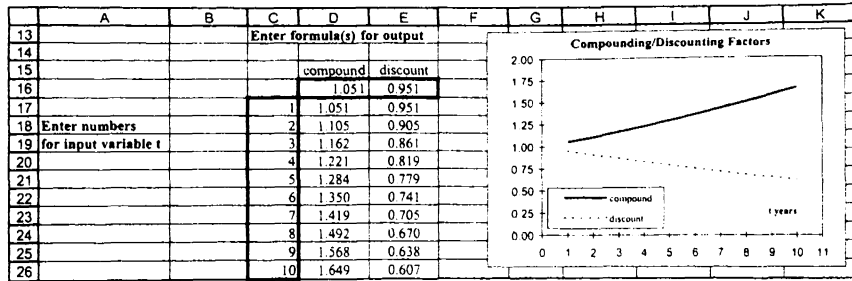


Figure 2.14 Data Table with different compounding and discounting factors for different periods

2.7.2 Setting Up Data Tables with Two Inputs

Suppose we wish to calculate discounting factors (from the formula in cell C11) for different rates of interest as well as different time periods. Once again, the appropriate layout for two inputs must be set up before invoking the Data Table procedure. One possible layout is shown in Figure 2.15.

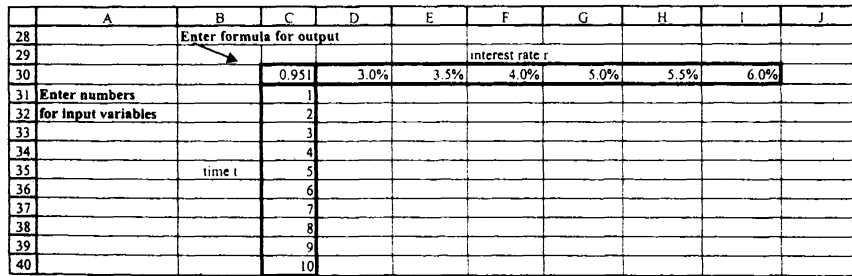


Figure 2.15 Layout for Data Table with two input variables in CompoundDTab sheet

Here the Table area is set up to be range C30:I40. Down the first column are the values for the period t (column input variable) for which discount factors are required (col input). Across row 30 are five values for the interest rate r (row input). The top left-hand cell of the table (C30) contains the formula to be recalculated for all the combinations of interest rate and time. The formula in C30 is =C11, which links to the discount factor formula.

The steps to get the Data Table data are:

- Select the Table range i.e. C30:I40
- From the menu, choose Data then Table
 - In dialog box, specify: Column input cell as cell C7
 - Row input cell as cell C5
 - then click OK

The results are displayed in Figure 2.16. Check that the values for the 5% rate tally with those previously calculated in Figure 2.14.

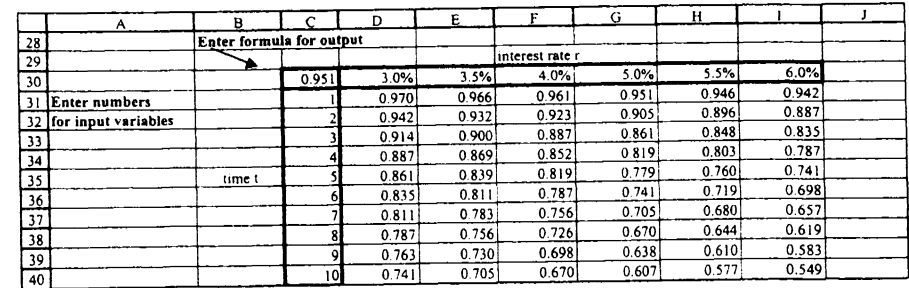


Figure 2.16 Data Table with discount factors for different interest rates and periods

Data Tables are extremely useful for doing a range of what-ifs painlessly. The good news is that the tables automatically adapt to changes in the model. The bad news is that if a sheet contains many Data Tables, their continual recalculation can slow down the speed with which the sheet adapts to other entries or modifications. For this reason, it is possible to switch off automatic recalculation for tables.

Note the following points in setting up Data Tables:

- At present, Excel requires the Data Table input cells to be in the same sheet as the table.
- Data Table cells contain formulas of the array type, e.g. the entries in the table cells take the form {=TABLE(C5,C7)} where C5 and C7 are the input cells. Since these are terms of an array, you cannot edit a single formula in a table.
- To rebuild or expand a Data Table, select all cells containing the {=TABLE()} formula with Edit then Clear All or press Delete.
- Changing the input values and other assumption values causes Data Tables to re-evaluate unless the default calculation method is deliberately changed.

In large models for which a long time is spent in recalculation after every entry in the spreadsheet, the automatic recalculation of Data Tables can be switched off if necessary. To do this:

- Choose Tools then Options
- Select the Calculation tab then choose Automatic except Tables

When automatic recalculation is switched off, pressing the F9 key forces a recalculation of all tables.

If you are familiar with the Black-Scholes formula for option valuation, you may wish to consolidate your knowledge of Data Tables by setting up the three tables suggested in the BSDTab sheet. This is equivalent to examining the sensitivity of the Black-Scholes call value to the current share price S , and to various other inputs.

2.8 XY CHARTS

Excel provides many types of charts, but for mathematical, scientific and financial purposes, the XY (Scatter) chart is preferable. Where unambiguous, we refer to this

type simply as an XY chart. The important point is that the XY chart has both the X and Y axes numerically scaled. In all other two-axis chart types (including the Line chart), only the vertical axis is numerically scaled, the horizontal X axis being for labels.

Creating an XY chart is handled by the Chart Wizard which proceeds through four steps referred to as Chart Type, Source Data, Chart Options and Chart Location. Given that we shall almost always be creating XY charts, embedded in the spreadsheet, the most important of the four steps is the second, Source Data. The steps are discussed for the results of the Data Table with one input described in section 2.7.1 and illustrated in Figure 2.14. The Data Table results to be charted are in range C17:E26 of the Compound-Tab sheet, column C containing the x values against which columns D and E are to be plotted. Having selected the data to be charted, the steps are:

1. Click the Chart Wizard button on the main toolbar. (It looks like a small bar chart.) In the Step 1 dialog box (in Figure 2.17), choose the 'Chart Type'; here XY (Scatter), subtype smoothed Line without markers. Click the button 'to view sample'. If OK, continue by clicking the Next button.

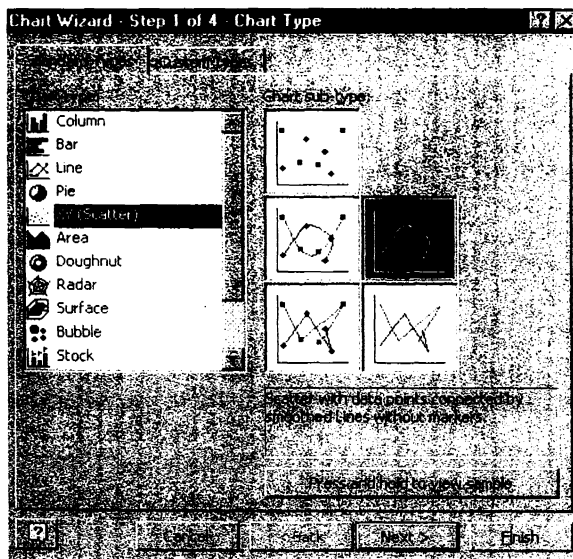


Figure 2.17 Dialog box for specifying the Chart Type

2. In the Step 2 dialog box, check that the correct 'Source Data' is specified on the Data Range sheet, noting that Excel will interpret this block as three column series. Click the Series tab on which the X and Y values are specified for Series1. Click in the Name box and add a name for the currently activated series, either by selecting a spreadsheet cell or by typing in a name (compound). Figure 2.18 shows Series2 being re-named 'discount', the entry in cell E15. Click on the Next button to continue.

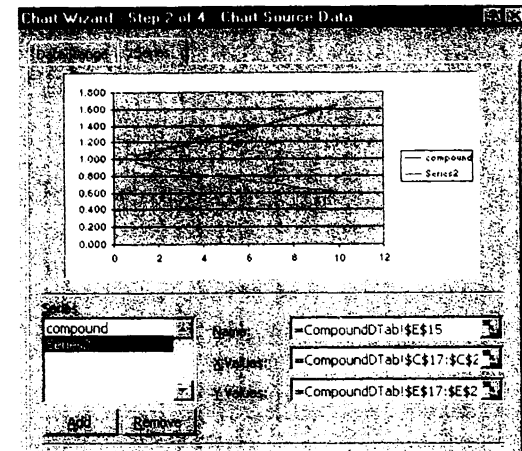


Figure 2.18 Dialog box for specifying the X and Y values in the Source Data

3. In the Step 3 dialog box, set up the 'Chart Options' such as titles, gridlines, appearance or not of legends, etc. For our chart, it is sufficient to add titles and switch off gridlines, as in Figure 2.19. Continue by clicking the Next button.

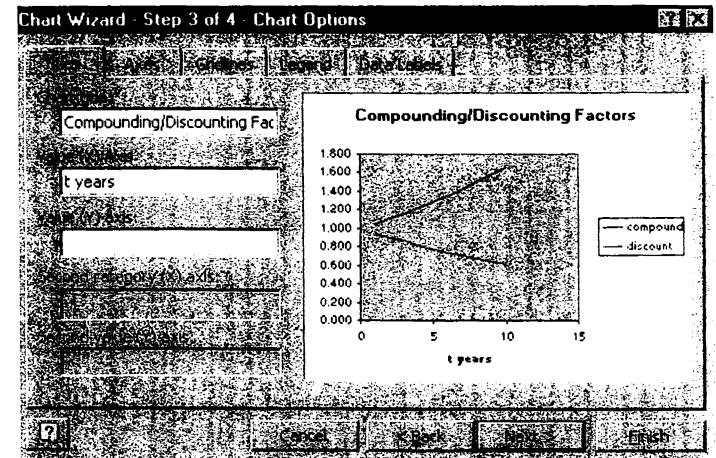


Figure 2.19 Dialog box for specifying the Chart Options

4. In the Step 4 dialog box, decide the 'Chart Location'. A chart can be created either as an object on the spreadsheet (an embedded chart) or on a separate chart sheet. Often it is preferable to have an embedded chart to see the effect in the chart of data changes, as is the case shown here in Figure 2.20.

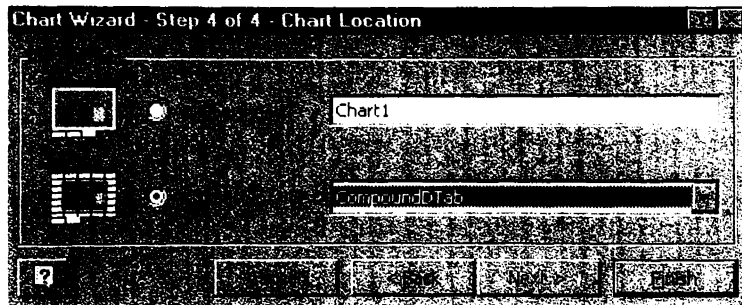


Figure 2.20 Dialog box for specifying the Chart Location

For a more professional look, the raw chart shown in Figure 2.21 needs certain cosmetic work to improve its appearance, the main changes being to Format the Plot area to a plain background (None), to work on the axes via Format Axis, similarly the Titles and Legends. Frequently the scales on axes, the font size for titles, etc. and size of chart area need changing. The plots of the data series may need to show the individual points (which can be achieved by activating the data series and reformatting). By modifying the raw chart, a more finished version such as that shown as part of Figure 2.14 can be achieved.

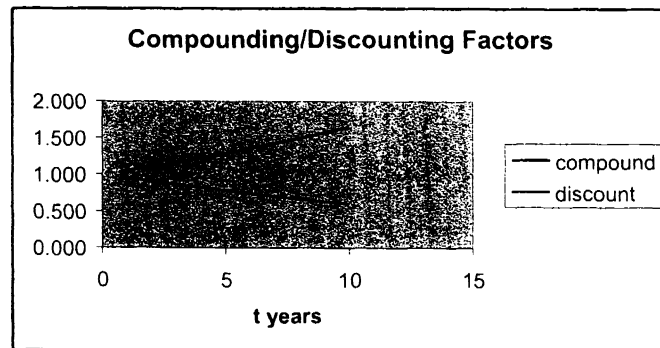


Figure 2.21 Raw chart produced by the Chart Wizard

2.9 ACCESS TO DATA ANALYSIS AND SOLVER

Excel has some additional modules which are available if a full installation of the package has occurred but may be missing if a space-saving installation was done. We use both Solver and the Analysis ToolPak regression routine, so it is worth checking on the Tools menu that both are available. See Figure 2.22 which includes both Solver and Data

Analysis options. If either is missing from the Tools menu, click on the Add-Ins option (also on the Tools menu). Check that Analysis ToolPak, Analysis ToolPak-VBA and Solver are ticked in the Add-Ins dialog box as shown in Figure 2.22, then click OK. As a result, you should see both Solver and Data Analysis on the Tools menu.

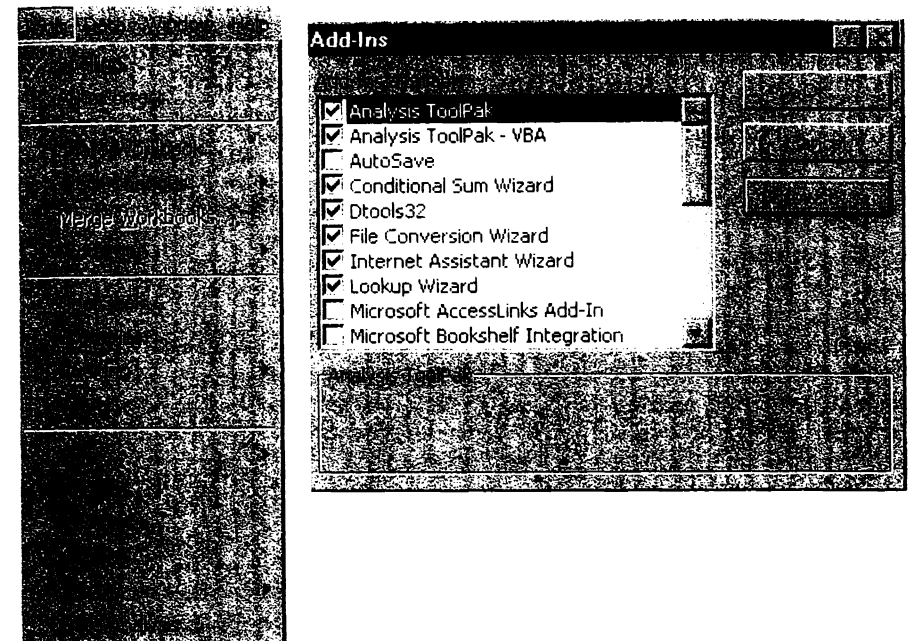


Figure 2.22 Tools menu with Data Analysis and Solver available, also the Add-Ins dialog box

Using Solver is best described in the context of an optimisation problem, so this is postponed until section 6.5 of the Equities part of the book where it is used to find optimum portfolio weights.

Before tackling regression analysis using functions and via the Analysis ToolPak, we briefly look at range names, which are useful for selecting and referencing large ranges of data.

2.10 USING RANGE NAMES

Figure 2.23 shows the top part of returns data for ShareA and Index in the Beta sheet of AMFEXCEL. Our purpose in the next section is to regress the Share returns on the Index returns to see if there is a relationship. To specify the calculation routines, it helps to have 'names' for the cell ranges containing the data, say ShareA for the data range B5:B64 and Index for the index returns in C5:C65.

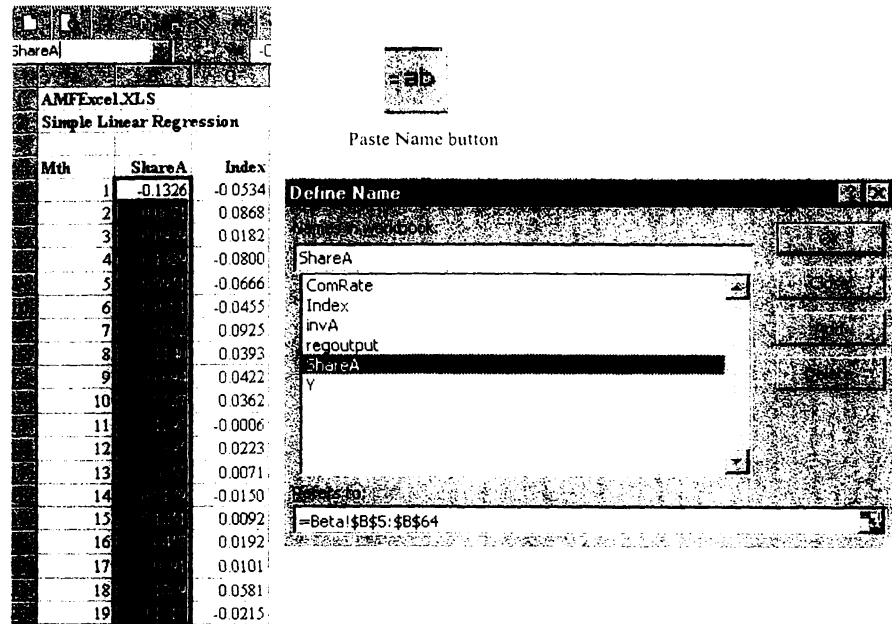


Figure 2.23 Data range with name, the Paste Name button and the Define Name dialog box

If the range to be named is selected, the desired name can be entered in the 'namebox' shown to the left just below the main menu bar, as illustrated in Figure 2.23. Thereafter the name ShareA is attached to range B5:B64 in the Beta sheet. Alternatively, the selected range can be named by choosing Insert then Name then Define, adding the name ShareA in the dialog box that appears (also in Figure 2.23). Thereafter, the returns range can be selected or referenced by name, for example by choosing the name ShareA from the namebox or in functions by using the Paste Name button.

2.11 REGRESSION

We assume that the reader is broadly familiar with simple (two-variable) regression analysis, which we shall employ in the Equities part of the book. Our purpose here is to outline how Excel can be applied to carry out the necessary computations. In fact, there are several ways of doing regression with Excel, the main division being between using Excel functions and applying the regression routine from the Analysis ToolPak. We illustrate these two main alternatives for simple regression using the Share and Index returns data in the Beta sheet of the workbook, first using the Excel functions and second via the Data Analysis Regression procedure. If you are unfamiliar with these functions and procedures, the Beta sheet provides a suitable layout for experimentation.

Excel provides individual functions for the most frequently required regression statistics. Figure 2.24 shows the intercept and slope functions that make up the regression

equation and two measures of fit, namely R-squared and the residual standard deviation (labelled STEYX for 'standard error of Y given X'). Since the data ranges have been named, their names can be 'pasted in' to the Function palette using the Paste Name button. The functions are dynamic in that they change if the data (in ranges ShareA and Index) alters.

	E	F	G	H	I
29	Excel regression functions				
30					
31	INTERCEPT	-0.0013	=INTERCEPT(ShareA,Index)		
32	SLOPE	1.5065	=SLOPE(ShareA,Index)		
33	RSQ	0.4755	=RSQ(ShareA,Index)		
34	STEYX	0.0595	=STEYX(ShareA,Index)		

Figure 2.24 Excel functions for regression in the Beta sheet

As well as the individual functions, there is also the LINEST array function, which takes as inputs the two columns of returns and outputs the essential regression quantities in array form. Note that it is important to select a suitably sized range for the output before entering the LINEST syntax. Usually for simple regression a cell range of five rows by two columns is required. Figure 2.25 shows the function LINEST(ShareA,Index,,1) being specified in the Function palette, the output range of F40:G44 having been selected. The Const input is left blank and the Stats input ensures that the full array of statistical results is returned.

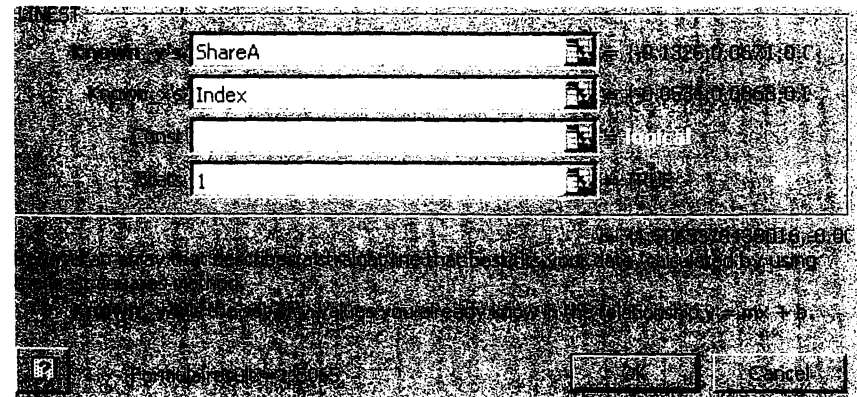


Figure 2.25 Building the LINEST array function in the formula palette

The array output is shown (with annotation) in Figure 2.26. In addition to the slope and intercept (here labelled Beta and Alpha respectively), LINEST also provides the standard errors of estimates and a rudimentary analysis of variance (or ANOVA).

	E	F	G	H	I
38	Output from Linest (Remember Ctrl+Shift+Enter)				
39					
40	Beta	1.5065	-0.0013	Alpha	
41	Beta (SE)	0.2077	0.0078	Alpha (SE)	
42	RSQ	0.4755	0.0595	STEYX	
43	F	52.5873	58.0000	N-2	
44	Regression SS	0.1860	0.2051	Residual SS	
45					

Figure 2.26 Array output from the LINEST function used on the returns data in the Beta sheet

Next, we examine the regression routine contained in the Analysis ToolPak. To set this up, choose Tools, then Data Analysis, then Regression, to see the dialog box shown in Figure 2.27. Once again the Y and X ranges can be referenced by selection or if named, by 'pasting in' the names. It is usually more convenient to specify a top left-hand cell for the output range in the data sheet rather than accept the default (of results dumped in a new sheet).

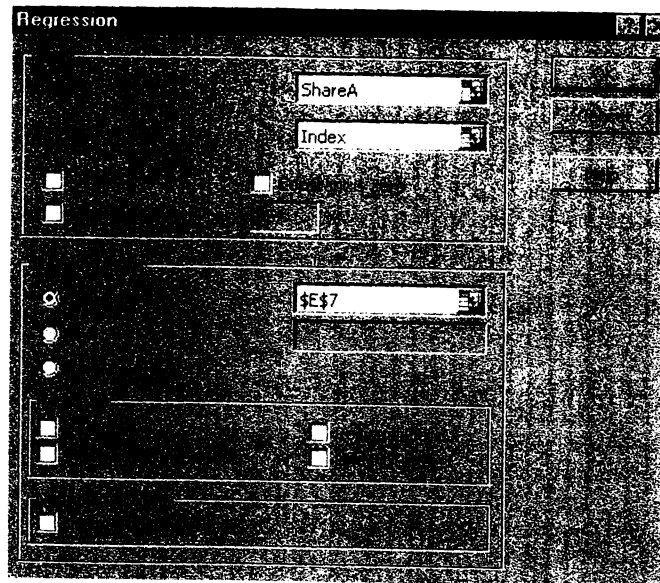


Figure 2.27 Specifying the Analysis ToolPak regression calculations and output option

The results are shown in Figure 2.28, the intercept and slope of the regression equation being in cells F23 and F24. The R-squared and Standard Error (the residual standard deviation) in cells F12 and F13 are measures of fit. The output is static, that is, it is a dump of numbers no longer connected to the original data. If any of the input data changes, the analysis does not adjust, but has to be manually recalculated.

	E	F	G	H	I	J	K
7	SUMMARY OUTPUT						
8							
9	Regression Statistics						
10	Multiple R	0.6896					
11	R Square	0.4755					
12	Adjusted R Square	0.4665					
13	Standard Error	0.0595					
14	Observations	60					
15							
16	ANOVA						
17		df	SS	MS	F	Sig F	
18	Regression	1	0.1860	0.1860	52.5873	1.11E-09	
19	Residual	58	0.2051	0.0035			
20	Total	59	0.3911				
21							
22		Coefficients	Std Error	t Stat	P-value	Lower 95%	Upper 95%
23	Intercept	-0.0013	0.0078	-0.164	0.870	-0.017	0.014
24	X Variable 1	1.5065	0.2077	7.252	0.000	1.091	1.922

Figure 2.28 Results from Analysis ToolPak regression routine shown in the Beta sheet

To illustrate the dynamic nature of functions in contrast, suppose that the index for month 4 was subsequently corrected to value 0.08 (not -0.08 as currently in cell C8). If the entry is changed, the functions in F31:F34 and F40:G44 update instantaneously, so that for example, RSQ drops from 0.4755 to 0.3260. To update the Analysis ToolPak results, the regression must be run again.

The static nature of the output from the Analysis ToolPak routines makes them less attractive for calculation purposes than the parallel Excel functions. Typically, the routines are written in the old XLM macro language from Excel version 4. More importantly, they cannot easily be incorporated into VBA user-defined functions, in contrast to most of the Excel functions.

2.12 GOAL SEEK

Goal Seek is another of Excel's static procedures. This tool produces a solution that matches a formula cell to a numerical target. For example, in Figure 2.29 there is a discrepancy between the market price (in G8) and the Black-Scholes call value (G4) for an option on a share. (The Black-Scholes value depends on the volatility of the share, and always contains an estimate of future volatility.) Suppose we want to know the size of the volatility that will match the two, or equivalently make the difference between the two (in G9) zero. This is an ideal problem for Goal Seek to solve.

To use Goal Seek on the BSDTab sheet, choose Tools then Goal Seek. Fill in the problem specification as shown in Figure 2.30 and click on OK. The solution produced by Goal Seek is shown in Figure 2.31, namely the volatility needs to be 23% for the Black-Scholes formula value to match the market price.

	A	B	C	D	E	F	G
2	Black-Scholes Option Values						
3							
4	Share price (S)		100.00			Black-Scholes Call Value	9.73
5	Exercise price (X)		95.00			via user-defined function	
6	Int rate-cont (r)		8.00%				
7	Dividend yield (q)		3.00%				
8	Option life (T, years)		0.50		Market value		10.43
9	Volatility (σ)		20.00%		Difference		0.70
10							

Figure 2.29 BS Call value formula and market price in BSDTab sheet

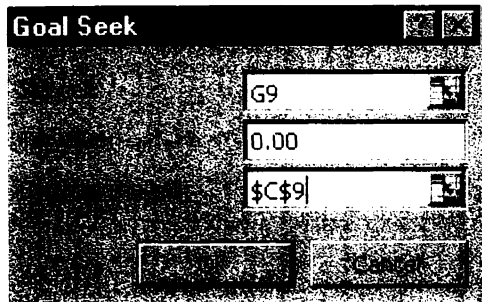


Figure 2.30 Goal Seek settings to find volatility (C9) that matches BS value and market price

	A	B	C	D	E	F	G
2	Black-Scholes Option Values						
3							
4	Share price (S)		100.00			Black-Scholes Call Value	10.43
5	Exercise price (X)		95.00			via user-defined function	
6	Int rate-cont (r)		8.00%				
7	Dividend yield (q)		3.00%				
8	Option life (T, years)		0.50		Market value		10.43
9	Volatility (σ)		23.00%		Difference		0.00
10							

Figure 2.31 Volatility value produced by Goal Seek making difference (in G9) zero

Goal Seek starts with an initial 'guess', then uses an iterative algorithm to get closer and closer to the solution. Effectively the initial value of the 'changing cell'—here volatility of 20%, is the initial guess. In finding a solution, Goal Seek allows you to vary one input cell, whereas Solver (which is introduced in Chapter 6) allows you to change several input cells.

To consolidate, use Goal Seek in the BSDTab sheet to find the volatility implied by a market price of 9.

2.13 MATRIX ALGEBRA AND RELATED FUNCTIONS

Matrix notation is widely used in algebra, as it provides a compact form of expression for systems of similarly structured equations. Operations with matrices look surprisingly like ordinary algebraic operations, but in particular multiplication of matrices is more complex. Excel contains useful matrix functions, in the Math category, which require a little background knowledge of how matrices behave to get full benefit from them. The following sections explain matrix notation, and describe the operations of transposing, adding, multiplying and inverting matrices. The examples illustrating these operations are in the MatDef sheet of the AMFEXCEL workbook. If you are conversant with matrices, you may wish to jump straight to the summary of matrix functions (section 2.13.7).

2.13.1 Introduction to Matrices

In algebra, rectangular arrays of numbers are referred to as matrices. A single column matrix is usually called a column vector; similarly a single row matrix is called a row vector. In Excel, rectangular blocks of cells are called arrays. All the following blocks of numbers can be considered as matrices:

$$\begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad | \quad 6 \quad 7 \quad | \quad \begin{bmatrix} -3 & 2 & 7 \\ 2 & 20 & 19 \\ 7 & 9 & 21 \end{bmatrix} \quad | \quad \begin{bmatrix} -3 & 2 & 7 \\ 2 & 20 & 19 \\ 7 & 9 & 21 \\ 0 & 13 & 3 \end{bmatrix}$$

where the brackets $| |$ are merely notational. Calling these matrices \mathbf{x} , \mathbf{y} , \mathbf{A} , and \mathbf{B} respectively, \mathbf{x} is a column vector and \mathbf{y} a row vector. Matrix \mathbf{A} has three rows and three columns and hence is a square matrix. \mathbf{B} is not square since it has four rows and three columns, i.e. \mathbf{B} is a 4 by 3 matrix. The numbers of rows, r , and of columns, c , give the dimensions of a matrix sometimes written as $(r \times c)$. For example, if:

$$\mathbf{x} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} \quad \text{and} \quad \mathbf{y} = | 6 \quad 7 |$$

then \mathbf{x} has dimensions (2×1) whereas \mathbf{y} has dimensions (1×2) .

2.13.2 Transposing a Matrix

Transposition of a matrix converts rows into columns (and vice versa). Clearly the transpose of column vector \mathbf{x} will be a row vector, denoted as \mathbf{x}^T . The spreadsheet extract in Figure 2.32 shows the transposes of column vector \mathbf{x} and row vector \mathbf{y} .

The TRANSPOSE function applied to the cells of an array returns its transpose. For example, the transpose of the 2 by 1 vector \mathbf{x} in cells C4:C5 will have dimensions (1×2) . To use the TRANSPOSE function, select the cell range I4:J4 and key in the formula:

$$=TRANSPOSE(C4:C5)$$

finishing with Ctrl+Shift+Enter pressed simultaneously. The result is shown in Figure 2.32.

	A	B	C	D	E	F	G	H	I	J	K
1	Array Manipulation:										
2	Transposes										
3	array:	dim				array:	dim				
4	x	(2x1)	2			x ^T	(1x2)	2	4		
5			4								
6											
7	y	(1x2)	6	7		y ^T	(2x1)	6			
8								7			
9	Array Multiplication										
10	xy	(2x2)	12	14		(xy) ^T	(2x2)	12	24		
11			24	28				14	28		
12											
13	yx	(1x1)	40			(yx) ^T	(1x1)	40			

Figure 2.32 Matrix operations illustrated in the MatDef sheet

2.13.3 Adding Matrices

Adding two matrices involves adding their corresponding entries. For this to make sense, the arrays being added must have the same dimensions. Whereas x and y cannot be added, x and y^T do have the same dimensions, 2 by 1, and therefore they can be added, the result being:

$$x + y^T = \begin{bmatrix} 2 \\ 4 \end{bmatrix} + \begin{bmatrix} 6 \\ 7 \end{bmatrix} = \begin{bmatrix} 8 \\ 11 \end{bmatrix} = z \text{ say}$$

To multiply vector y by 10 say, every entry of y is multiplied by 10. Thus:

$$10y = 10 * \begin{bmatrix} 6 & 7 \end{bmatrix} = \begin{bmatrix} 60 & 70 \end{bmatrix}$$

This is comparable to adding y to itself 10 times.

2.13.4 Multiplying Matrices

For two matrices to be multiplied they have to have a common dimension, that is, the number of columns for one must equal the number of rows for the other. The shorthand expression for this is 'dimensional conformity'. For the product xy the columns of x must match the rows of y , (2x1) times (1x2), resulting in a (2x2) matrix as output.

In Figure 2.32, the product xy in cells C10:D11 has elements calculated from:

$$\begin{bmatrix} 2 \\ 4 \end{bmatrix} \begin{bmatrix} 6 & 7 \end{bmatrix} = \begin{bmatrix} 2 * 6 & 2 * 7 \\ 4 * 6 & 4 * 7 \end{bmatrix} = \begin{bmatrix} 12 & 14 \\ 24 & 28 \end{bmatrix}$$

i.e. the row 1, column 1 element of product xy comes from multiplying the individual elements of row 1 of x by the elements of column 1 of y , etc.

In contrast, the product yx has dimensions (1x2) times (2x1), that is (1x1), i.e. it consists of a single element. Looking at product yx in cell C13, this element is computed as:

$$\begin{bmatrix} 6 & 7 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 6 * 2 + 7 * 4 \end{bmatrix} = \begin{bmatrix} 40 \end{bmatrix}$$

These results demonstrate that for matrices, xy is not the same as yx . The order of multiplication is critical.

The MMULT array function returns the product of two matrices, called array1 and array2. So to get the elements of the (2x2) matrix product xy , select the 2 by 2 cell range, C10:D11 and key in or use the Paste Function button and build the expression in the Formula palette:

$$=MMULT(C4:C5,C7:D7)$$

remembering to enter it with Ctrl+Shift+Enter.

If ranges C4:C5 and C7:D7 are named x and y respectively, then the formula to be keyed in simplifies to:

$$=MMULT(x,y)$$

Consider two more arrays:

$$C = \begin{bmatrix} 12 & 4 \\ 3 & 13 \end{bmatrix} \text{ and } D = \begin{bmatrix} 16 & 19 & -2 \\ 5 & 12 & 14 \end{bmatrix}$$

The dimensions of C and D are (2x2) and (2x3) respectively, so since the number of columns in C is matched by the number of rows in D , the product CD can be obtained, its dimensions being (2x3). So:

$$CD = \begin{bmatrix} (12 * 16 + 4 * 5) & (12 * 19 + 4 * 12) & (-12 * 2 + 4 * 14) \\ (3 * 16 + 13 * 5) & (3 * 19 + 13 * 12) & (-3 * 2 + 13 * 14) \end{bmatrix} = \begin{bmatrix} 212 & 276 & 32 \\ 113 & 213 & 176 \end{bmatrix}$$

However, the product DC cannot be formed because of incompatible dimensions (the number of columns in D does not equal the number of rows in C). In general, the multiplication of matrices is not commutative, so that usually $CD \neq DC$, as in this case.

If C and D are the names of the 2 by 2 and the 2 by 3 arrays respectively, then the cell formula:

$$=MMULT(C,D)$$

will produce the elements of the 2 by 3 product array.

2.13.5 Matrix Inversion

A square matrix I with ones for all its diagonal entries and zeros for all its off-diagonal elements is called an identity matrix. Thus:

$$I = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix} \text{ is an identity matrix}$$

Suppose D is the (2x3) matrix used above, and I is the (2x2) identity matrix, then:

$$ID = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 16 & 19 & -2 \\ 5 & 12 & 14 \end{bmatrix} = \begin{bmatrix} 16 & 19 & -2 \\ 5 & 12 & 14 \end{bmatrix} = D$$

Multiplying any matrix by an identity matrix of appropriate dimension has no effect on the original matrix (and is therefore similar to multiplying by one).

Now suppose \mathbf{A} is a square matrix of dimension n , that is an n by n matrix. Then, the square matrix \mathbf{A}^{-1} (also of dimension n) is called the inverse of \mathbf{A} if:

$$\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$$

For example, if:

$$\mathbf{A} = \begin{pmatrix} -3 & 2 & 7 \\ 2 & 20 & 19 \\ 7 & 9 & 21 \end{pmatrix} \text{ then } \mathbf{A}^{-1} = \begin{pmatrix} -0.175 & -0.015 & 0.072 \\ -0.064 & 0.079 & -0.050 \\ 0.086 & -0.029 & 0.045 \end{pmatrix}$$

and

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Finding the inverse of a matrix can be a lot of work. Fortunately, the MINVERSE function does this for us. For example, to get the inverse of matrix \mathbf{A} shown in the spreadsheet extract in Figure 2.33, select the 3 by 3 cell range I17:K19 and enter the array formula:

$$=\text{MINVERSE}(C17:E19)$$

You can check that the result is the inverse of \mathbf{A} by performing the matrix multiplication $\mathbf{A}\mathbf{A}^{-1}$.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
16	array: dim						array: dim										
17	A (3x3)	-3	2	7			A ⁻¹ (3x3)	-0.175	-0.015	0.072		AA ⁻¹	1.00	0.00	0.00		
18		2	20	19				-0.064	0.079	-0.050			0.00	1.00	0.00		
19		7	9	21				0.086	-0.029	0.045			0.00	0.00	1.00		
20																	
21	b (3x1)	20					x = A ⁻¹ b (3x1)	-3.4355									
22		-5						-1.6772									
23		0						1.8640									
24																	

Figure 2.33 Matrix inversion shown in the MatDef sheet

2.13.6 Solving Systems of Simultaneous Linear Equations

One use for the inverse of a matrix is in solving a set of equations such as the following:

$$-3x_1 + 2x_2 + 7x_3 = 20$$

$$2x_1 + 20x_2 + 19x_3 = -5$$

$$7x_1 + 9x_2 + 21x_3 = 0$$

These can be written in matrix notation as $\mathbf{Ax} = \mathbf{b}$ where:

$$\mathbf{A} = \begin{pmatrix} -3 & 2 & 7 \\ 2 & 20 & 19 \\ 7 & 9 & 21 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 20 \\ -5 \\ 0 \end{pmatrix} \quad \text{and } \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

The solution is given by premultiplying both sides of the equation by the inverse of \mathbf{A} :

$$\mathbf{A}^{-1}\mathbf{Ax} = \mathbf{A}^{-1}\mathbf{b}, \text{ so } \mathbf{Ix} = \mathbf{A}^{-1}\mathbf{b} \text{ i.e. } \mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$$

In Figure 2.33, the solution vector \mathbf{x} is obtained from the matrix multiplication function in cell range I21:I23 in the form:

$$=\text{MMULT}(I17:K19,C21:C23)$$

Not every system of linear equations has a solution, and in special cases there may be many solutions. The set $\mathbf{Ax} = \mathbf{b}$ has a unique solution only if the matrix \mathbf{A} is square and has an inverse \mathbf{A}^{-1} . In general, the solution is given by $\mathbf{x} = \mathbf{A}^{-1}\mathbf{b}$.

2.13.7 Summary of Excel's Matrix Functions

In summary, Excel has functions to transpose matrices, to multiply matrices and to invert square matrices. The relevant functions are these:

TRANSPOSE(array)	returns the transpose of an array
MMULT(array1, array2)	returns the matrix product of two arrays
MINVERSE(array)	returns the matrix inverse of an array

Because these functions produce arrays as outputs, the size of the resulting array must be assessed in advance. Having 'selected' an appropriately sized range of cells, the formula is keyed in (or obtained via the Paste Function button and built in the Formula palette). It is entered in the selected cell range with the combination of keys Ctrl+Shift+Enter (instead of simply Enter). If this fails, keep the output range of cells 'selected', press the Edit key (F2), edit the formula if necessary, then press Ctrl+Shift+Enter again.

To consolidate, try the matrix exercises in the sheet MatExs.

We make extensive use of the matrix functions in the Equities part of the book, both for calculations in the spreadsheet and as part of VBA user-defined functions.

SUMMARY

Excel has an extensive range of functions and procedures. These include mathematical, statistical and lookup functions, as well as much-used procedures such as setting up Data Tables and displaying results in XY charts.

Access to the functions is handled through the Paste Function button and the function inputs specified on the Formula palette. The use of range names simplifies the specification of cell ranges, especially when the ranges are sizeable. Range names can be used on the Formula palette.

Facilities on the Auditing toolbar, in particular the Trace Precedents, Trace Dependents and Remove Arrows buttons are invaluable in examining formula cells.

It helps to be familiar with the range of Excel functions because they can easily be incorporated into user-defined functions, economising on the amount of VBA code that has to be written.

Care is required in using array functions. It helps to decide in advance the size of the cell range appropriate for the array results. Then having selected the correct cell range, the formula is entered with the keystroke combination Ctrl+Shift+Enter.