



INFORMATION
TECHNOLOGY
FACULTY

Spreadsheet competency framework

A structure for classifying spreadsheet ability in finance professionals



Contents

1	Preface	2
2	About this framework	3
3	The framework specification	5
4	Explanatory notes to the framework	8
5	Twenty principles for good spreadsheet practice	19

ISBN 978-1-78363-665-5

Copyright © ICAEW 2016

All rights reserved.

If you want to reproduce or redistribute any of the material in this publication in any form, you should first get the permission of ICAEW in writing.

1 Preface

'Proficient with Microsoft Excel' – how many times have we all seen such a line on a CV, or in a job description? What does this sentence mean, and how can we know that what one person understands by it will match what another expects?

Spreadsheets are everywhere in modern business, from the smallest organisations to the largest and most complex. They are pervasive particularly among finance users, but notoriously prone to error. Getting the right spreadsheet specialist is vital to making sure that time and cost aren't wasted – but identifying who knows what isn't easy.

It is thinking about these questions that has spurred the ICAEW's Excel Community Advisory Committee to create this framework.

Spreadsheet skills are often learned ad hoc – almost two-thirds of Excel Community users are self-taught – and many users are unaware of their own true competency. Novices are generally overconfident; experts tend to sell themselves short. Getting the right person in the right role with the right skills is no small challenge, but it's one that we are looking to tackle with this framework.

Following on from our previous publication *Twenty principles for good spreadsheet practice*, you will find in the following pages a simple system of four levels that defines a common standard for classifying spreadsheet users and their skills. By putting such a system out to the market, our intention is to create a common language for everyone that needs to communicate clearly about spreadsheet ability.

I would like to thank the committee for their hard work creating this framework, and to encourage you to adopt the framework in your own business.



Michael Izza
Chief Executive Officer, ICAEW

2 About this framework

Spreadsheet risk is an area that affects many businesses, and more than most of them realise. Although eye-catching headlines about high-profile failures such as the misawarded *West Coast Mainline franchise* are the most commonly seen, the real cost for most businesses lies in under-trained staff, leading to constant time leakage through inefficient spreadsheet package use. ICAEW previously published *Twenty principles for good spreadsheet practice* to provide the most essential guidelines for efficient and safe spreadsheet use. Since publishing that guide, ICAEW has identified another area where guidance is needed: laying out exactly how different spreadsheet users' knowledge should be streamed into a simple system, and what skills are necessary for what roles.

This guide provides a common structure for discussing spreadsheet ability. It standardises a series of skillsets, called levels, which are useful for classifying different degrees of spreadsheet expertise.

For recruiters and businesses, this provides a simple way of understanding the difference in ability between different spreadsheet users, both those internal to the organisation and among potential new hires. It also lays out general expectations of what level of knowledge is appropriate for a range of common business roles that touch on the finance function. For individual spreadsheet users, the framework can be used both to provide a meaningful description of their own ability on a CV, and also to help direct the user towards potential learning topics for the future.

The levels are based on a broad description of a collection of skills. Naturally, different people will have encountered different tasks in their working life, and will have different sets of experience to one another. Learning to use spreadsheets is not a smooth progression from one level to another; in reality, some users may be expert in some specific areas but completely ignorant in others. However, in general a user will be familiar with most of the content of the level that best represents their current knowledge.

On the other hand, not all job roles are necessarily best served by the 'highest' level user available. Most roles will not have advanced spreadsheet use as a core requirement, and the level structure is designed to reflect that. Additional knowledge may be useful, or it may be simply unnecessary for the needs of that particular role.

Furthermore, while the levels are designed with a finance function in mind, their content is largely applicable to **any** person that uses spreadsheets in their job.

In brief, the levels are as follows.

Basic users will carry out **data entry** tasks in spreadsheets, and will have only the most fundamental knowledge necessary to be able to interact with a spreadsheet package.

General users are those with a moderate level of spreadsheet experience. Finance professionals who use spreadsheets on a regular basis should aim to reach this level in order to be at their most efficient when performing their duties. The primary interaction with spreadsheets for these individuals is to **modify** spreadsheets, rather than create sophisticated spreadsheets from scratch.

Creators are those with a greater degree of expertise and sophistication. Creators use spreadsheets as a primary element of their role, and need to consider how to **create and manage** spreadsheets of a greater degree of complexity. The variety of skills appropriate to the 'creator' level is considerably broader than for a 'general user', and some degree of specialisation is to be expected.

Developers represent the truly expert spreadsheet users, who are familiar with most of the core functionality of spreadsheet packages, and are able to **develop** high-complexity spreadsheets in a multi-user environment. Developers will frequently be highly specialised, with exceptional knowledge in specific areas. They will normally be able to self-teach practically any spreadsheet topic if they do not know it already.

The following provides a fictitious example of a company that employs Excel spreadsheet users of the different levels.

Example: Boxes Co

Boxes Co is a medium-sized shipping and logistics company with 100 employees in its central warehouse and office. These are mostly admin and finance staff, but also include warehouse operations as well as HR and other functions. Boxes Co uses Microsoft Excel for all its spreadsheet requirements.

Basic users

Ayesha works in Boxes Co's warehouse. She has only an occasional need to use spreadsheets, by updating goods received tracking sheets from a remote terminal in the warehouse office.

William is an HR administrator, and updates Boxes Co's logs of daily attendance and holiday numbers in an Excel template.

Both Ayesha and William learnt how to use their respective spreadsheets through on-the-job training.

General users

Sally works in Boxes Co's accounting department as an accounts receivable clerk. She frequently uses Excel to track and summarise the progress of outstanding debts, mostly by using templates but also adding her own formulas where appropriate.

Xu is a secretary for the team that organises Boxes Co's drivers. He fills in jobs, driver availability, and other information in a master team schedule workbook and coordinates HR and payroll information based on the drivers' submitted timesheets.

Sally and Xu both attended a half-day basic Excel training course as part of their induction process.

Creators

Cathryn works in a small team of management accountants and is responsible for producing reports for the management team. She produces many accounting spreadsheets both for her own use and for review by management. Cathryn also helps make some of the simpler templates that are used by the other teams within Boxes Co.

Sandeep is a payroll administrator, has more Excel knowledge than the other payroll administrators, and is responsible for most of the detailed Excel data manipulation that the team does. This includes using Excel to manipulate and summarise the data provided by Boxes Co's external payroll bureau and checking it against the payroll input data that Boxes Co holds.

Cathryn and Sandeep's job specifications include an expectation of appropriate spreadsheet skills, and their interviews included questions about their Excel knowledge.

Developer

Boxes Co employs an expert financial modeller, Helen. Helen is responsible for creating the detailed forecasting models used in the management's long-term planning, as well as creating models that are used by the logistics team to plan when they need to hire temporary extra drivers during periods of peak demand. She also creates automation macros and VBA for some of her co-workers' most time-intensive regular tasks.

Helen was hired on the basis of her financial modelling and Excel knowledge and experience.

Naturally, not everyone has the spreadsheet knowledge necessary to achieve the basic user level as described below. Our recommendation is that individuals below the basic user level should not be in a position to access an organisation's spreadsheets, as they are unlikely to use them safely and effectively. Furthermore, spreadsheets created for basic users should be made by creators or higher-level users, to ensure that they are designed with a clear enough purpose and strong enough guidance and protection to ensure that an inexperienced user will not go too far off the intended track.

There is currently no formal assessment process for the spreadsheet competency framework. This document is intended only as a structure to organise the very wide and inconsistent degrees of spreadsheet ability that exist in the marketplace, often hidden behind a throwaway CV line such as 'proficient with Microsoft Excel'. By creating a simple structure for assessing spreadsheet ability, the framework introduces the language for discussing what different jobs require, and what abilities different people have.

The framework as a whole should be applied with a degree of common sense. For example, there may be specific tools that a certain role requires, which are above and beyond the general level of spreadsheet knowledge which that role would otherwise need – this then provides a sensible case for deviation from the framework.

3 The framework specification

● – Core item. A user at this level must have knowledge of all of these items

icaew.com/spreadsheetcompetency

● – Beneficial item. A user at this level should have knowledge of a significant proportion of these items

This is not intended to be a complete inventory of spreadsheet skills. Please refer to the notes in Section 4 for further explanation and commentary.

	Basic user	General user	Creator	Developer
Design and best practice				
Twenty principles for good spreadsheet practice – see inside back cover		●	●	●
File naming and version control		●	●	●
Label data, sheets, ranges			●	●
Layout design			●	●
Documentation			●	●
- What and how to document				●
- Explain complex calculations				●
Application of modelling standards				●
Create flexible, extensible spreadsheets				●
Reviewing and team working				
Audit formulas			●	●
Detect inconsistent formulas			●	●
Resolve issues			●	●
High-level review of a spreadsheet			●	●
Basic skills				
Access and save files	●	●	●	●
Read and enter data	●	●	●	●
Set up and printing	●	●	●	●
Efficiency of use				
Shortcuts:				
- Navigation shortcuts	●	●	●	●
- Copy and paste shortcuts	●	●	●	●
- Additional shortcuts			●	●
Find and replace	●	●	●	●
Named cells:				
- Awareness		●	●	●
- Find a cell from the name		●	●	●
- Create named cells and ranges			●	●
Group sheets			●	●
Go To / Go To Special			●	●
Fixed references with \$		●	●	●

	Basic user	General user	Creator	Developer
Manage file size / calculation speed				●
Set up and manage multi-user spreadsheets				●
Formulas				
Basic arithmetic	●	●	●	●
Arithmetic formulas		●	●	●
Logical operators		●	●	●
Logical formulas		●	●	●
Lookup formulas		●	●	●
Rounding approaches		●	●	●
Text formulas		●	●	●
Financial formulas			●	●
Date formulas			●	●
Other complex formulas			●	●
Formatting				
Apply regular cell formatting		●	●	●
Custom number formats		●	●	●
Hiding, grouping, merging		●	●	●
Conditional formatting		●	●	●
Charting				
Link between chart and data		●	●	●
Pre-set charts		●	●	●
Protection and errors				
Default error messages:				
- Recognise the varieties		●	●	●
- Trace to source		●	●	●
- Resolve			●	●
- Build error-resistant formulas			●	●
Manual vs automatic calculation		●	●	●
Worksheet protection		●	●	●
Know limits of Excel protection			●	●
Build error checking formulas			●	●
Editable ranges			●	●
Data validation			●	●
Data analysis				
Filters				
- Apply filters to data			●	●
- Use pre-set filters			●	●
- Use rule-based filters			●	●
Excel tables:				
- Use data stored in an Excel table		●	●	●
- Use Excel tables to manage data			●	●
- Write structured formulas				●
PivotTables				
- Read and understand a PivotTable		●	●	●
- Insert and build simple PivotTables			●	●
- Build complex PivotTables/PowerPivot				●

	Basic user	General user	Creator	Developer
Use Power BI tools				●
Manipulate and cleanse data				○
Array formulas and Boolean logic				●
Scenarios and sensitivity analysis				●
Macros and automation				
Macros:				
- Find and run an installed macro			●	○
- Record and playback a macro				○
- Adjust a recorded macro				●
- Write custom VBA				●
Run a Goal Seek			●	●
Run a complex Solver scenario				●
Development and problem solving				
Break down and research problems			●	○
Trace errors in spreadsheet they build			○	○
Understand when to move past spreadsheets				●
Design to mitigate common problems				●

4 Explanatory notes to the framework

The title of this document refers only to ‘spreadsheets’, but we recognise that Microsoft Excel is the most commonly used spreadsheet package. As such, the specific examples in this section refer to Microsoft Excel features and functions. However, other spreadsheet packages are broadly similar and the framework applies equally to them with only minor changes in terminology.

1 – Twenty Principles for good spreadsheet practice

General user and higher

There are many aspects of good practice that should concern Excel users of all levels. These are summarised in the *Twenty principles for good spreadsheet practice*.

General users should be aware of the concept of spreadsheet risk, and know the major sins – eg, hard-coding of constants as explained in principle 14.

Capability Framework.xlsx - Excel				
FILE	ICAEW	HOME	INSERT	
11	VAT rate	20%	VAT rate	25%
12				
13				
14	Net	Vat	Net	Vat
15	100	20	100	=\\$A15*\$E\$11
16	200	40	200	=\\$A16*\$E\$11
17	300	60	300	=\\$A17*\$E\$11
18	400	80	100	=\\$A18*\$E\$11
19	500	100	125	=\\$A19*\$E\$11
20	600	120	150	=\\$A20*\$E\$11
21	700	140	175	=\\$A21*\$E\$11
22	800	160	200	=\\$A22*\$E\$11
23	900	180	225	=\\$A23*\$E\$11
24	1000	200	250	=\\$A24*\$E\$11
25				

Principle 14. Never embed in a formula anything that might change or need to be changed.

Creators should be familiar with the 20 principles and understand the importance of applying good practice to reduce risk. Developers should always follow best practice, and design spreadsheets that encourage their users in turn to do the same.

A summary of the *Twenty principles for good spreadsheet practice* can be found on the inside back cover of this publication. A full version of the principles can be obtained from icaew.com/excel20principles

2 – Application of modelling standards

Developer

While there are many modelling standards, no particular one is singled out here. A developer should be able to take any given modelling standard and apply it to any suitable project.

3 – Navigation shortcuts

All levels

Navigation shortcuts allow the user to move quickly and easily around the spreadsheet. They save time and reduce error. A basic user should know the PgUp/PgDn keys and the use of arrow keys, Tab, and Enter. A general user should also know how to use Ctrl and/or Shift with the above shortcuts.

4 – Copy and paste shortcuts

All levels

After data entry, copying and pasting is the most common task in Excel. Shortcuts greatly increase efficiency. A basic user should know Ctrl + c and Ctrl + v, and how to use paste values. A general user should know how to use paste special, insert shortcuts, fill down/right, and the fill handle.

A screenshot of Microsoft Excel showing a spreadsheet with data in columns A-E and rows 6-26. Cell D11 contains 'VAT rate' and '20%'. The context menu is open over cell D20, with the 'Copy Cells' option selected. Other options include 'Fill Formatting Only', 'Fill Without Formatting', and 'Flash Fill'. The formula bar shows the formula =A15*VATRate.

Cells can be copied to adjacent cells by dragging the fill handle in the bottom right-hand corner of the active cell or by using copy and paste commands and shortcuts. In both cases an option button will provide alternatives for how the operation is performed. Understanding the different options is important in order to avoid overwriting existing cells with inappropriate content. Cell contents can also be copied to adjacent cells by selecting cells from the active cell downwards or rightwards and using the Ctrl + d or Ctrl + r keyboard shortcut respectively.

5 – Additional shortcuts

Creator and higher

Additional shortcuts at the creator level may include the function keys and reviewing shortcuts such as ctrl + [and].

A screenshot of Microsoft Excel showing a spreadsheet with data in columns A-C and rows 7-16. Cell A15 contains '100'. A red arrow points to the cell B15, which contains '20'. The formula bar shows the formula =A15*VATRate. A tooltip indicates 'Control+[to select precedent cells'.

A screenshot of Microsoft Excel showing a spreadsheet with data in columns A-C and rows 7-16. Cell B15 contains '20'. A red arrow points to the cell A15, which contains '100'. The formula bar shows the formula =\$A15*VATRate. A tooltip indicates 'Control+] to select dependent cells'.

The [shortcut selects the cells that the active cell refers to (precedent cells). The] shortcut selects cells that refer to the active cell (dependent cells). Either shortcut can be used multiple times to trace successive levels of precedent and dependent cells. Precedents can be traced through to different worksheets and workbooks, dependents only as far as the active sheet. Note that the formulas ribbon tab, formula auditing group, and trace dependents command will additionally show dependent cells in other worksheets and other, open, workbooks.

6 – Find and replace

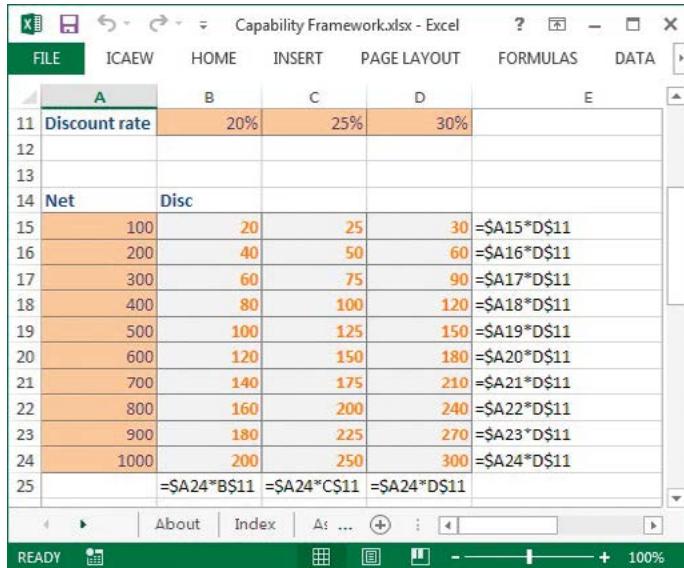
All levels

This tool is a quick and easy way to make sweeping changes to a workbook. A basic user should be able to use find to discover relevant data, and do a simple replace. A general user should be able to refine the area of operation of a replace operation, and replace formats or formula parts as well as cell content.

7 – Fixed references with \$

General user and higher

Fixed references are a timesaving device for creating multiple similar formulas. A general user should have the skills to read and follow a formula which contains \$ references, and create fixed references eg, \$A\$1. A creator should be able to use fixed references in any situation, including mixed reference such as A\$1 or \$A1.



A screenshot of Microsoft Excel showing a table titled "Capability Framework.xlsx - Excel". The table has columns labeled A, B, C, D, and E. Row 11 contains values 20%, 25%, and 30% in columns B, C, and D respectively. Row 14 is a header row with "Net" in A and "Disc" in B. Rows 15 through 24 show various calculations, such as =\\$A15*D\$11, =\\$A16*D\$11, etc., demonstrating the use of fixed references (\$A\$1) in formulas.

	A	B	C	D	E
11	Discount rate	20%	25%	30%	
12					
13					
14	Net	Disc			
15	100	20	25	30	=\\$A15*D\$11
16	200	40	50	60	=\\$A16*D\$11
17	300	60	75	90	=\\$A17*D\$11
18	400	80	100	120	=\\$A18*D\$11
19	500	100	125	150	=\\$A19*D\$11
20	600	120	150	180	=\\$A20*D\$11
21	700	140	175	210	=\\$A21*D\$11
22	800	160	200	240	=\\$A22*D\$11
23	900	180	225	270	=\\$A23*D\$11
24	1000	200	250	300	=\\$A24*D\$11
25		=\\$A24*B\$11	=\\$A24*C\$11	=\\$A24*D\$11	

Being able to use fixed and partially fixed references not only improves productivity by requiring the entry of fewer individual formulas, it also greatly improves reliability and ease of review by reducing the number of different formulas that need to be checked, and by making it easier to identify inconsistent formulas.

8 – Set up and manage multi-user spreadsheets

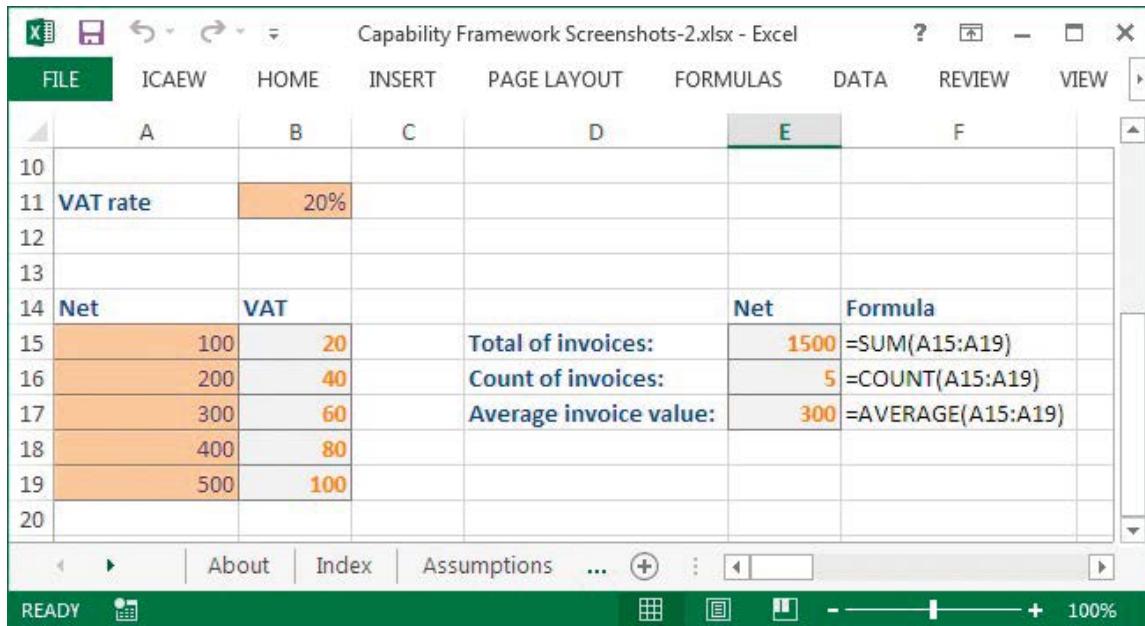
Developer

This may include shared workbooks, SharePoint, cloud systems and so on.

9 – Arithmetic formulas

General user and higher

This includes SUM, COUNT, and AVERAGE.



A screenshot of Microsoft Excel showing a table titled "Capability Framework Screenshots-2.xlsx - Excel". The table has columns labeled A, B, C, D, E, and F. Row 11 contains "VAT rate" in A and "20%" in B. Row 14 is a header row with "Net" in A and "VAT" in B. Rows 15 through 19 show calculations: Total of invoices: 1500 (=SUM(A15:A19)), Count of invoices: 5 (=COUNT(A15:A19)), and Average invoice value: 300 (=AVERAGE(A15:A19)).

	A	B	C	D	E	F
10						
11	VAT rate	20%				
12						
13						
14	Net	VAT			Net	Formula
15	100	20		Total of invoices:	1500	=SUM(A15:A19)
16	200	40		Count of invoices:	5	=COUNT(A15:A19)
17	300	60		Average invoice value:	300	=AVERAGE(A15:A19)
18	400	80				
19	500	100				
20						

These functions are most commonly used with a single cell range as an argument, but each can accept up to 255 arguments. Although seemingly simple to use, care needs to be taken to ensure that the right range is selected when first set up and that the formula is checked when additional cells are added in cells adjacent to the original range.

10 – Logical operators

General user and higher

This includes $>$, $<$, $=$, and combinations of these.

Value	Value	Result	Formula	Description
1	1	TRUE	=A13=B13	Equal
1	1	FALSE	=A14>B14	Greater than
1	1	FALSE	=A15<B15	Less than
1	1	FALSE	=A16<>B16	Not equal
1	1	TRUE	=A17>=B17	Greater than or equal
1	1	TRUE	=A18<=B18	Less than or equal

Although the logical operators are often used as an argument of an Excel function such as IF, they can also be used as a simple 'statement' that will return a value of TRUE or FALSE.

11 – Logical formulas

General user and higher

This includes IF, SUMIF(S), and COUNTIF(S). A general user should be able to follow what one of these formulas does in practice and check its operation manually. A creator should be comfortable writing their own formula from scratch.

Function	Product	Quality	Result	Formula
COUNTIF()		A	4	=COUNTIF(Stock[Quality],C11)
COUNTIFS()	Carrot	A	2	=COUNTIFS(Stock[Product],B12,Stock[Quality],C12)
SUMIF()		A	250	=SUMIF(Stock[Quality],C13,Stock[Quantity])
SUMIFS()	Carrot	A	120	=SUMIFS(Stock[Quantity],Stock[Product],B14,Stock[Quality],C14)

Product	Qualit	Quantit	A	Formula
Parsnip	A	50	50	=IF([@Quality]=Stock[[#Headers],[A]],[@Quantity],0)
Carrot	A	20	20	=IF([@Quality]=Stock[[#Headers],[A]],[@Quantity],0)
Parsnip	B	30	0	=IF([@Quality]=Stock[[#Headers],[A]],[@Quantity],0)
Turnip	A	80	80	=IF([@Quality]=Stock[[#Headers],[A]],[@Quantity],0)
Carrot	A	100	100	=IF([@Quality]=Stock[[#Headers],[A]],[@Quantity],0)
Carrot	C	70	0	=IF([@Quality]=Stock[[#Headers],[A]],[@Quantity],0)

The IF function returns its second argument if its first argument is TRUE and the third argument if it is FALSE.

COUNTIF and SUMIF compare a range of cells against a single criterion value and return the count or sum of cells that match the criterion. SUMIF can include a third argument that specifies a different range of cells to sum.

COUNTIFS and SUMIFS were introduced in Excel 2007 and are similar in operation to COUNTIF and SUMIF but, instead of a single criterion, they can each use up to 127 pairs of criteria ranges and values.

12 – Lookup formulas

General user and higher

This includes VLOOKUP, HLOOKUP, and INDEX/MATCH. A general user should be able to follow what one of these formulas does in practice and check its operation manually. A creator should be comfortable writing their own formula from scratch.

Capability Framework.xlsx - Excel					
FILE	ICAEW	HOME	INSERT	PAGE LAYOUT	FORMULAS
11	Product	Quantit	Product	Quantity	Formula
12	Aubergine	30	Carrot	30	=VLOOKUP(D12,UnsortedTable,2)
13	Turnip	80			
14	Carrot	100			
15					
16	Product	Quantit	Product	Quantity	Formula
17	Aubergine	30	Carrot	100	=VLOOKUP(D17,SortedTable,2)
18	Carrot	100			
19	Turnip	80			
20					
21	Product	Quantit	Product	Quantity	Formula
22	Aubergine	30	Carrot	100	=VLOOKUP(D22,UnsortedTable2,2,FALSE)
23	Turnip	80			
24	Carrot	100			
25					
26	Product	Quantit	Product	Quantity	Formula
27	Aubergine	30	Carrot	100	=INDEX(Table4[#All],MATCH(D27,Table4[[#All],[Product]],0),MATCH(E26,Table4[#Headers],0))
28	Turnip	80			
29	Carrot	100			
30					

Lookup formulas are particularly easy to get wrong, usually through failure to appreciate the importance of the argument that controls the kind of match being performed. Approximate matches with VLOOKUP, HLOOKUP and MATCH will only work if the first column of the table being referred to is sorted correctly.

INDEX and MATCH can be used as an alternative to VLOOKUP and HLOOKUP and offer more flexibility as well as the ability to find a match at an intersection of a row and a column.

13 – Rounding approaches

General user and higher

This includes ROUND(/UP/DOWN), MROUND/FLOOR/CEILING, and presentational rounding with formats. A general user should be able to follow what one of these formulas does in practice and check its operation manually. A creator should be comfortable writing their own formula from scratch.

Capability Framework.xlsx - Excel					
FILE	ICAEW	HOME	INSERT	PAGE LAYOUT	FORMULAS
11	Value	Rounded	Formula	Actual value	
12	1490.387568	1490.39	=ROUND(A12,2)	1490.390000	
13	1490.387568	1490	=ROUND(A13,0)	1490.000000	
14	1490.387568	1000	=ROUND(A14,-3)	1000.000000	
15	1490.387568	1491	=ROUNDUP(A15,0)	1491.000000	
16	1490.387568	1490	=ROUNDDOWN(A16,0)	1490.000000	
17	1490.387568	1490	=ROUNDDOWN(A17,0)	1490.000000	
18	1490.387568	1490.49	=MROUND(A18,0.5)-0.01	1490.490000	
19	1490.387568	1490.99	=MROUND(A19,0.5)-0.01	1490.990000	
20	1490.387568	1490.49	=CEILING(A20,0.5)-0.01	1490.490000	
21	1490.387568	1489.99	=FLOOR(A21,0.5)-0.01	1489.990000	
22					
23	Value	Formatted	Format code	Actual value	
24	1490.387568	1,490.39	,##0.00	1490.387568	
25	1490.387568	1,490	,##0	1490.387568	
26	1490.387568	1	,###,	1490.387568	

Although number formatting will change the way numbers are displayed, as the examples show, formatting does not change the precision used in dependent calculations. The various rounding functions do change the precision of dependent calculations. ROUND, ROUNDUP and ROUNDDOWN round to specified numbers of decimal places. Negative numbers of decimal places can be used. -3 will round to thousands for example. MROUND, CEILING and FLOOR round to a specified multiple. For example, the nearest multiple of 0.50.

14 – Text formulas

General user and higher

This includes &, LEN, LEFT/RIGHT/MID, FIND/SEARCH, and SUBSTITUTE. A general user should be able to follow what one of these formulas does in practice and check its operation manually. A creator should be comfortable writing their own formula from scratch.

Capability Framework.xlsx - Excel						
	A	B	C	D	E	F
11		Search for position of -	Number of characters in cell	Start of value (from right)	Value (as text)	Value
12	Sales-10000	6	11	5	10000	10000
13	Cost of sales-2000	14	18	4	2000	2000
14	Admin expenses-3100	15	19	4	3100	3100
15						
16	Formula	=SEARCH("-",A14)	=LEN(A14)	=C14-D14	=RIGHT(A14,D14)	=VALUE(D14)
17						
18	LEFT(), MID(), RIGHT()					
19	001-900-650	001	900	650		
20	Formula	=LEFT(A19,3)	=MID(A19,5,3)	=RIGHT(A19,3)		
21						
22	SEARCH() v FIND()	Character to locate	Result	Formula	Notes	
23	Cost of sales 2000	S	3	=SEARCH(B23,A23)	SEARCH() is not case sensitive	
24	Cost of sales-2000	S	#VALUE!	=FIND(B24,A24)	FIND() is case sensitive	
25	Cost of sales-2000	S	9	=SEARCH(B25,A25,5)	An optional argument allows SEARCH() and	
26	Cost of sales-2000	S	#VALUE!	=FIND(B26,A26,5)	FIND() to start from a particular position.	
27						
28	SUBSTITUTE()	Replace	With	Result	Formula	
29	Cost of sales 2000	-	£	Cost of sales £2000	=SUBSTITUTE(A29,B29,C29)	
30						
31	Concatenation			Result	Formula	
32	Using the & operator	Printed on	21/06/2016	Printed on 21 June 2016	=B32 & " " & TEXT(C32,"d mmmm yyyy")	
33	CONCATENATE()	Printed on	21/06/2016	Printed on 21 June 2016	=CONCATENATE(B33," ",TEXT(C33,"dd mmmm yyyy"))	
34						

Text functions can be used to extract part of a text value by position or by finding the position of a delimiter character using SEARCH – not case sensitive – or FIND – case sensitive.

SUBSTITUTE can replace one or a string of characters with one or a string of other characters. The & operator or CONCATENATE function can be used to join text in separate cells together. The TEXT function formats values according to a format code.

15 – Financial formulas

Creator and higher

This includes NPV, XNPV, IRR, XIRR, and other similar functions. A creator should be able to follow what one of these formulas does in practice and check its operation manually. A developer should be comfortable writing their own formula from scratch.

Capability Framework.xlsx - Excel				
FILE	ICAEW	HOME	INSERT	
	A	B	C	
11	Rate	4%		
12				
13	Item	Value	Date	
14	Installation	-16000	19/03/2016	
15	Generation income	3200	31/03/2017	
16	Generation income	3200	31/03/2018	
17	Generation income	3200	31/03/2019	
18	Generation income	3200	31/03/2020	
19	Generation income	3200	31/03/2021	
20	Generation income	3200	31/03/2022	
21	Generation income	3200	31/03/2023	
22	Maintenance	-5000	30/06/2023	
23	Generation income	3200	31/03/2024	
24				
25				
26	Net present value	1,758.05	=XNPV(Rate,CashFlows[Value],CashFlows[Date])	
27	Internal rate of return	7.0169%	=XIRR(CashFlows[Value],CashFlows[Date])	
28	Check NPV =0 at IRR	0.00015137	=XNPV(B27,CashFlows[Value],CashFlows[Date])	

16 – Date formulas

Creator and higher

This includes TODAY, NETWORKDAYS, WORKDAY, EOMONTH, EDATE, YEARFRAC, and DAY/MONTH/YEAR, as well as date arithmetic. A creator should be able to follow what one of these formulas does in practice and check its operation manually. A developer should be comfortable writing their own formula from scratch.

Capability Framework.xlsx - Excel				
FILE	ICAEW	HOME	INSERT	
	A	B	C	
12	Start date/time	End date/days	Result	Formula
13	21/06/2016 11:38			=NOW()
14	21/06/2016			=TODAY()
15	21/06/2016	3	30/09/2016	=EOMONTH(A15,B15)
16	21/06/2016	3	21/09/2016	=EDATE(A16,B16)
17	21/06/2016	20	19/07/2016	=WORKDAY(A17,B17,Holidays[Holiday dates])
18	21/06/2016	20	19/07/2016	=WORKDAY.INTL(A18,B18,1,Holidays[Holiday dates])
19	21/06/2016	30/06/2016	8	=NETWORKDAYS(A19,B19,Holidays[Holiday dates])
20	21/06/2016	30/06/2016	8	=NETWORKDAYS.INTL(A20,B20,1,Holidays[Holiday dates])
21	21/06/2016		21	=DAY(A21)
22	21/06/2016		6	=MONTH(A22)
23	21/06/2016		2016	=YEAR(A23)
24	21/06/2016	30/06/2016	0.02459016	=YEARFRAC(A24,B24,1)
25				
26	Holiday dates			
27	02/05/2016			
28	30/05/2016			
29	29/08/2016			
30	26/12/2016			
31	27/12/2016			
32	13/06/2016			
33	14/06/2016			
34	15/06/2016			
35	16/06/2016			
36	17/06/2016			

XNPV and XIRR are more flexible than NPV and IRR allowing cash flows to be entered at specified dates rather than being assumed to take place at the end of a set of regular periods.

Many of the financial functions depend on their arguments being entered in a specific way. For example, PV, PMT, IPMT and PPMT all require rates and numbers of payments to be entered using consistent time units. If you are calculating on the basis of monthly payments, rates and numbers of payments must be entered as monthly values also.

The .INTL forms of WORKDAY and NETWORKDAYS include an additional argument to specify the weekend pattern to be used.

17 – Other complex formulas

Creator and higher

This level is broad, but may include INDIRECT, OFFSET, SUBTOTAL, and the like. A creator should be able to follow what one of these formulas does in practice and check its operation manually. A developer should be comfortable writing their own formula from scratch.

A screenshot of Microsoft Excel showing a trace dependency dialog box. The dialog box is titled "Microsoft Excel" and contains the message: "The Trace Dependents command found no formulas that refer to the active cell." It has an "OK" button. In the background, there is a spreadsheet with data for vegetables across different months. The cell D13 (containing "36,000") is highlighted with a red border. A yellow arrow points from the text "36,000" in cell D13 down to the "OK" button of the dialog box. The dialog box is semi-transparent, allowing the underlying spreadsheet data to be seen.

	A	B	C	D	E	F	G	H	I	J
12		31/01/2016	29/02/2016	31/03/2016	30/04/2016	31/05/2016	30/06/2016			
13	Parsnips	50,000	40,000	36,000	48,000	30,000	25,000			
14	Turnips	40,000	32,000	27,000	40,000	28,000	21,000			
15	Carrots	18,000	24,000	17,000	35,000	40,000	29,000			
16										
17		31/03/2016								
18	Parsnips	36,000	=OFFSET(
19	Turnips	27,000	=OFFSET(
20	Carrots	17,000	=OFFSET(
21										
22										

Although more complex Excel functions and formulas can be useful in certain circumstances some do create issues for clarity, calculation speed and auditability. For example, OFFSET makes it easy to choose the results for a particular month, but the Formula Auditing Trace Precedents and Trace Dependents tools cannot trace the cells that the OFFSET function ultimately refers to.

18 – Custom number formats

General user and higher

A general user should be aware that number formats can differ, and that a cell's value may be different to its appearance. A creator should be able to make simple custom formats for themselves.

A screenshot of Microsoft Excel showing a table with custom number formats. The table has columns labeled "Employee", "Total £", and "Total £'000". The "Total £'000" column uses a custom format where numbers are preceded by a red parentheses. The table data includes names like Andrew Fuller, Anne Dodsworth, Janet Leverling, etc., and their corresponding total values.

	A	B	C	D	E
10	Employee	Total £	Total £'000	(formatted)	(formatted)
11	Andrew Fuller				
12	Anne Dodsworth	-89229.71	(89,230)	(89.2)	
13	Janet Leverling	36275.07	36,275	36.3	
14	Laura Callahan	-39675.48	(39,675)	(39.7)	
15	Margaret Peacock	66353.08	66,353	66.4	
16	Michael Suyama	-92624.62	(92,625)	(92.6)	
17	Nancy Davolio	25569.81	25,570	25.6	
18	Robert King	-41969.53	(41,970)	(42.0)	
19	Steven Buchanan	-97745.5	(97,746)	(97.7)	

Using the most appropriate number format can make a significant difference to the clarity and impact of a spreadsheet.

19 – Conditional formatting

Creator and higher

Conditional formatting is a deep tool that allows for a lot of customisation of cell appearance and automation of formatting. A creator should know how conditional formatting works and be able to apply the pre-set conditional formats to data. A developer should be comfortable reviewing and editing conditioning formatting rules, and making custom formats with formulas.

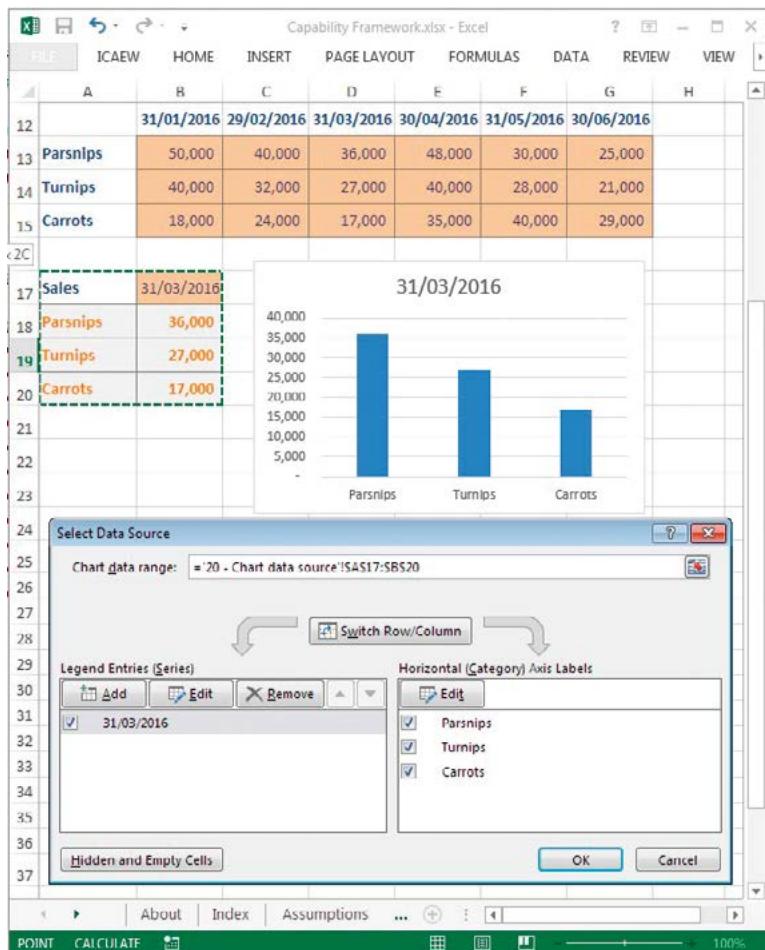
The screenshot shows a Microsoft Excel window with a PivotTable. The PivotTable has columns labeled 'Row Labels', 'Total £', '(formatted)', and 'Total - Data bars'. The 'Total - Data bars' column contains red data bars representing values from -89,229.71 to -97,745.5. A context menu is open over the last row of the table, showing options like 'Format Selection', 'Format Cells', and 'Format Rule'. The 'Format Rule' option is selected, opening the 'Edit Formatting Rule' dialog box. In the dialog box, the 'Apply Rule To:' field contains the formula '=16 - Number formatting!\$E\$11'. The 'Select a Rule Type:' section is expanded, showing five options: 'Format all cells based on their values' (selected), 'Format only cells that contain', 'Format only top or bottom ranked values', 'Format only values that are above or below average', and 'Use a formula to determine which cells to format'. The 'Format all cells based on their values:' section is also expanded, showing settings for 'Format Style' (Data Bar), 'Type' (Automatic), 'Value' (Automatic), and 'Bar Appearance' (Fill: Solid Fill, Color: black, Border: No Border, Color: black). There is a 'Preview' button at the bottom right of the dialog box.

Conditional formatting can be applied to fields in a PivotTable as well as to selected cells. The Manage Rules, Edit Rules option provides a wide range of options for customising how each type of conditional format works.

20 – Link between chart and data

General user and higher

A general user should be able to identify the data that feeds in to a given chart. A creator should be able to modify the data to include new items, or to edit what data is used.

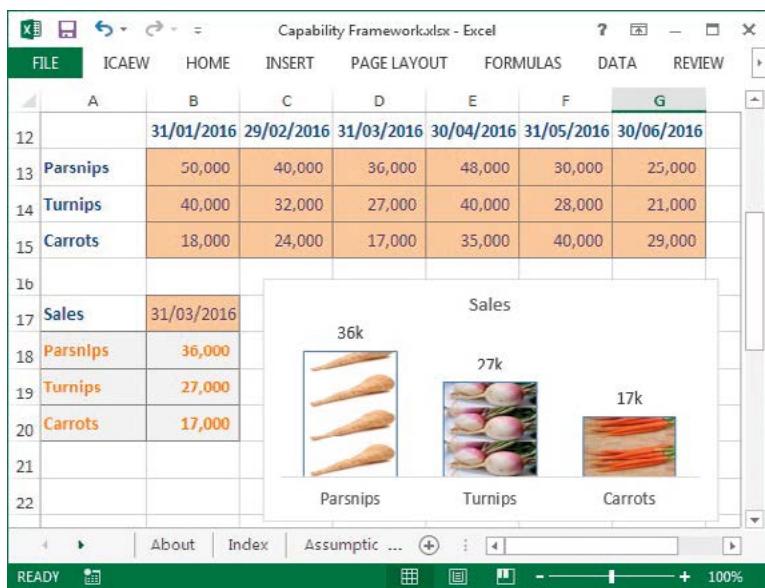


Where a chart is on the same sheet as the data and is based on a simple cell range, selecting the chart will identify the cells used for the data area, legend and axis. The Chart Tools, Design Ribbon tab, Data group includes a Select Data command.

21 – Pre-set charts

General user and higher

A general user should be comfortable selecting an appropriate pre-set chart and attaching it to data. A creator should be able to adjust pre-set charts as required, and make changes to the visual elements thereof.



Although most spreadsheet applications allow considerable control over every aspect of a chart, sometimes the simplest charts are the most effective.

22 – Build error-resistant formulas

Creator and higher

This may involve the IFERROR function, but should also include less broad approaches. For example, using an IF function to catch divide-by-zero errors, without also masking other genuine errors.

The screenshot shows a Microsoft Excel spreadsheet titled "Competency Framework.xlsx [Read-Only] - Excel". The data is organized into two main sections: "Expense" (rows 12-18) and "Lookup" (rows 19-26). The "Lookup" section contains various formulas to handle errors from VLOOKUP and VLKUP functions. A red box highlights the range B22:B23, which contains the formula =IF(ISERROR(VLKUP(A22,A14:B17,2,FALSE)),0,VLKUP(A22,A14:B17,2,FALSE)). This formula uses ISERROR to check if the VLKUP function returns an error, and if so, it returns 0 instead of the error message. Below this, another row (B23) shows the formula =IF(ISNA(VLKUP(A23,A12:B16,2,FALSE)),0,VLKUP(A23,A12:B16,2,FALSE)), where ISNA is used to check for a #NAME? error. The "Version if after 2003" column indicates that the first formula is valid for 2003 and later versions, while the second is specific to 2007 and later. The "2007" and "2013" columns show alternative formulas using IFERROR and IFNA respectively, which directly return 0 for errors without checking the error type.

Competency Framework.xlsx [Read-Only] - Excel											
FILE	ICAEW	HOME	INSERT	PAGE LAYOUT	FORMULAS	DATA	REVIEW	VIEW	DEVELOPER	ADD-INS	INQUIRE
A B C D E F G H I J K L											
12	Expense	Total									
13	Rental income	5000									
14	Rates	3000									
15	Bank charges	1200									
16	Bank interest	1100									
17	Accountancy	500									
18											
19	Lookup	Result	Formula	Version if after 2003							
20	Rent	#N/A	=VLOOKUP(A20,A13:B17,2,FALSE)								
21	Rent	0	=IF(ISERROR(VLOOKUP(A21,A13:B17,2,FALSE)),0,VLOOKUP(A21,A13:B17,2,FALSE))								
22	Rental income	0	=IF(ISERROR(VLKUP(A22,A14:B17,2,FALSE)),0,VLKUP(A22,A14:B17,2,FALSE))								
23	Rental income	#NAME?	=IF(ISNA(VLKUP(A23,A12:B16,2,FALSE)),0,VLKUP(A23,A12:B16,2,FALSE))								
24	Rent	0	=IF(ISNA(VLOOKUP(A24,A13:B17,2,FALSE)),0,VLOOKUP(A24,A13:B17,2,FALSE))								
25	Rent	0	=IFERROR(VLOOKUP(A25,A13:B17,2,FALSE),0)	2007							
26	Rent	0	=IFNA(VLOOKUP(A26,A13:B17,2,FALSE),0)	2013							
< > ... 16 - Number formatting 17 - Conditional formatting 18 - 20 Principles 20 - Char ... + 4			COUNT: 6	grid	row	col	-	+	100%		
READY											

Using the most specific check possible helps avoid an unintended error being masked by the expected error. In this example the VLOOKUP function has been spelt incorrectly in B22 and B23. The general ISERROR function recognises this as an error and the formula (incorrectly) returns 0 rather than an error message. The use of the more specific ISNA function doesn't mask the #NAME? error but does return 0 if the VLOOKUP function fails to find a match.

The more recent IFERROR and IFNA functions avoid the need to enter the formula to be checked twice.

23 – Use pre-set filters

All levels

A basic user should be able to select and unselect check boxes on filters, and use the menus to sort data. General users should be able to use the larger selection of built-in filters, eg, 'Top 10'.

The screenshot shows a Microsoft Excel spreadsheet titled "Capability Framework.xlsx - Excel". The data is organized into columns A, B, and C. Column A contains items like "Generation inco", and column C contains dates from "31/03/2017" to "31/03/2020". A filter menu is open over column B, specifically for the "Value" dropdown. The menu includes options like "Sort Smallest to Largest", "Sort Largest to Smallest", "Sort by Color", "Clear Filter From 'Value'", and "Filter by Color". A secondary filter menu for "Number Filters" is open over column B, showing "Search" and a list of numerical filters: "(Select All)", "-16000", "-5000", and "3200". The "Top 10..." option is checked, indicating that the user is setting up a top 10 filter for the "Value" column.

Capability Framework.xlsx - Excel					
FILE	ICAEW	HOME	INSERT	PAGE LAYOUT	
13	A	B	C	D	E
14	Item	Value	Date		
15	Generation inco	Sort Smallest to Largest	31/03/2017		
16	Generation inco	Sort Largest to Smallest	31/03/2018		
17	Generation inco	Sort by Color	31/03/2019		
18	Generation inco	Clear Filter From "Value"	31/03/2020		
19	Generation inco	Filter by Color			
20	Generation inco	Number Filters	Equals...		
21	Generation inco		Does Not Equal...		
22	Generation inco		Greater Than...		
23	Generation inco		Greater Than Or Equal To...		
24			Less Than...		
25			Less Than Or Equal To...		
26			Between...		
27					
28					
29					
30					
31					
32					
33					
About Index Assumptions ...					
READY					

Top 10 filters can be set to include any number of top or bottom items, by number or by percent.

5 Twenty principles for good spreadsheet practice

Below is a summary of the *Twenty principles for good spreadsheet practice*. For more details, please see icaew.com/excel20principles

1. Determine what role spreadsheets play in your business, and plan your spreadsheet standards and processes accordingly.
2. Adopt a standard for your organisation and stick to it.
3. Ensure that everyone involved in the creation or use of spreadsheets has an appropriate level of knowledge and competence.
4. Work collaboratively, share ownership, peer review.
5. Before starting, satisfy yourself that a spreadsheet is the appropriate tool for the job.
6. Identify the audience. If a spreadsheet is intended to be understood and used by others, the design should facilitate this.
7. Include an ‘about’ or ‘welcome’ sheet to document the spreadsheet.
8. Design for longevity.
9. Focus on the required outputs.
10. Separate and clearly identify inputs, workings and outputs.
11. Be consistent in structure.
12. Be consistent in the use of formulas.
13. Keep formulas short and simple.
14. Never embed in a formula anything that might change or need to be changed.
15. Perform a calculation once and then refer back to that calculation.
16. Avoid using advanced features where simpler features could achieve the same result.
17. Have a system of backup and version control, which should be applied consistently within an organisation.
18. Rigorously test the workbook.
19. Build in checks, controls and alerts from the outset and during the course of spreadsheet design.
20. Protect parts of the workbook that are not supposed to be changed by users.

Spreadsheet competency framework is the result of debate among members of the IT Faculty’s Excel Community Advisory Committee. The members of the Excel Community Advisory Committee are as follows:

Roland Brook, Smith & Williamson

John Tennent, Corporate Edge

Adrian Maconick, Finsbury Solutions

Mike Copeman, Mike Copeman

Alastair Hynd, RSM

Paul Wakefield, Paul Wakefield

Alex Carse, Numeritas

Rishi Sapra, KPMG

Angela Collins, EuSpRIG

Rob Bayliss, Grant Thornton

Chris Littlewood, Filtered

Sanjay Magecha, Financial Visibility

Daniel Emkes, Harrow School

Simon Hurst, The Knowledge Base

Glen Feechan, needaspreadsheet.com

Tom Brichieri-Colombi, Mazars

Excel is one of the most popular end-user tools in the accountant's portfolio. Spreadsheets enable us to quickly and flexibly perform analysis that otherwise would be difficult or time-consuming; however, there is a tendency to place undue trust in them. ICAEW's Excel Community provides a 'one-stop shop' for accountants who want to use Excel better and understand and minimise spreadsheet risk.

For more information about the Excel Community, please visit icaew.com/excel

For more information about the IT Faculty, please visit icaew.com/itfac

ICAEW is a world leading professional membership organisation that promotes, develops and supports over 145,000 chartered accountants worldwide. We provide qualifications and professional development, share our knowledge, insight and technical expertise, and protect the quality and integrity of the accountancy and finance profession.

As leaders in accountancy, finance and business our members have the knowledge, skills and commitment to maintain the highest professional standards and integrity. Together we contribute to the success of individuals, organisations, communities and economies around the world.

Because of us, people can do business with confidence.

ICAEW is a founder member of Chartered Accountants Worldwide and the Global Accounting Alliance.

www.charteredaccountantsworldwide.com

www.globalaccountingalliance.com



ICAEW
Chartered Accountants' Hall Moorgate Place London EC2R 6EA UK

T +44 (0)20 7920 8481

E itfac@icaew.com

icaew.com/itfac

[linkedin.com – find ICAEW](#)

[twitter.com/icaew_excel](#)

[facebook.com/icaew](#)