SumProduct

NEWSLETTER #141 - August 2024

www.sumproduct.com | www.sumproduct.com/thought

And then there were still three!

As SumProduct heads into August, we have received the fantastic news that all three of our Excel Most Valuable Professionals (MVPs) are *still* MVPs. That's a total of 22 awards for the company now, which we think is a fantastic achievement, even if we do say so ourselves! Congratulations to all.

Of course, that's not our only news. There's the latest Gartner® Magic Quadrant™ announcement, checkboxes are now Generally Available in Excel, there are two new Excel translation functions (*"tlhorgh potlhwI' pov 'Iq"*) on their way, we introduce a new Python editor and the online Excel grid gets a makeover.

Plus the usual gang! We have the usual Beat the Boredom Challenge, Charts & Dashboards tips, Excel for Mac, Visual Basics, Power Pivot Principles, Power Query Pointers, the new Over to AI, Power BI Updates and more Excel Updates too. Our Keyboard Shortcuts put another **SHIFT** in, but we had to coerce the A to Z of Excel functions this month: they said **NOT NOW** but we insisted…

As always, happy reading and remember: stay safe, stay happy, stay healthy.

Liam Bastick, Managing Director, SumProduct

MVP Renewal 2024

SumProduct is pleased to announce that two of our Directors, **Liam Bastick** and **Tim Heng**, have been re-awarded Microsoft's Most Valuable Professional (MVP) award for Excel for 2024-25. Our third MVP, **Steve Kraynak**, did not need to bite his fingernails for this round as he was only recently awarded the honour (we think he just chewed on his toenails instead…).

This award recognises exceptional technical community leaders from around the world who voluntarily share their high-quality, real-world expertise with others. Microsoft MVPs are a highly select group of experts representing technology's best and brightest who share a deep commitment to community and a willingness to help others.

Worldwide, there are over 100 million participants in technical communities; of these participants, there are c.4,000 active Microsoft MVPs. In Excel, we believe there are approximately 130 that have received this award.

Microsoft's MVP Award evaluates technical expertise and voluntary community contributions for the past year, considering the quality, quantity and level of impact of contributions. It's a difficult award to attain and just as difficult to retain.

At SumProduct, you can rely on our experience and willingness to help - simply drop us a line at contact@sumproduct.com.

Latest Magic Quadrant for Analytics and Business Intelligence Platforms

From time to time, we do get accused of perhaps jumping too much onto the Microsoft bandwagon, but whilst we celebrate our success of MVP renewal, if other software serves our clients' needs better, we will always use that.

Therefore, it's interesting to view the latest Gartner® Magic Quadrant™, which still depicts Microsoft as furthest along both the Completeness of Vision and Ability to Execute spectra.

Gartner (June 2024)

Gartner

Microsoft has stated that it perceives its evolution of Power BI into Fabric and AI's interaction with Power BI as key factors in its success. This is the sixth year it has topped both of Gartner's scales. And that's why we remain on board.

Congratulations to Microsoft: we continue to advocate for many of their products, but as always, all of us here at SumProduct will keep a watching brief on the business intelligence and analytical future on behalf of all our (potential) clients.

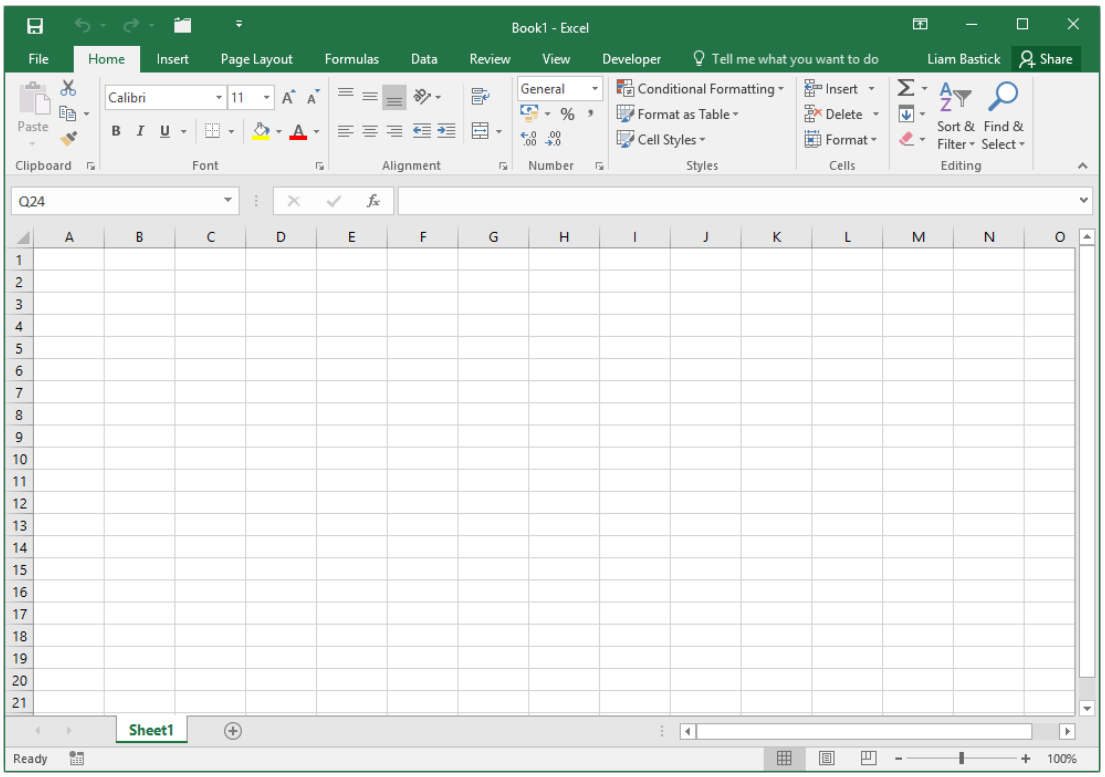
Laying Out a Financial Model

Most spreadsheets serve a purpose, typically to communicate a forecast, evaluate a project or undertake some other form of quantitative analysis. Note the verb: *communicate*. They are communicative tools first and foremost. However, that communication is frequently sullied by poor layouts, insufficient labels and inconsistencies which cause end users difficulties in understanding the model purpose and content. These issues can cause unnecessary extra work for the model builder, difficulty in comprehension for the decision maker and lead to conclusions based upon errors in logic and / or formulae which are not readily identifiable / visible.

Therefore, you should put some thought into designing your spreadsheets and not just the formulae, functions and formats you use. That's what I want to concentrate on today. You may recall from Best Practice Modelling, "Best Practice" should be considered a proper noun to reflect the idea that a good model has four key attributes I call **CRaFT**:

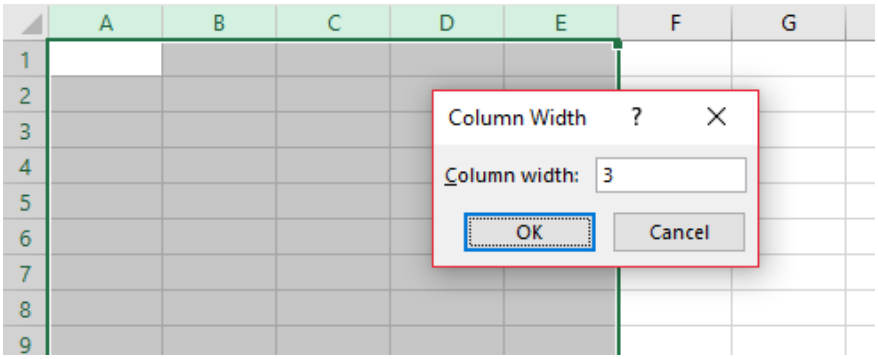
- Consistency;
- Robustness;
- Flexibility; *and*
- Transparency.

With this borne in mind, let me explain how I think when putting a worksheet together. To begin with, let's start with a blank worksheet:

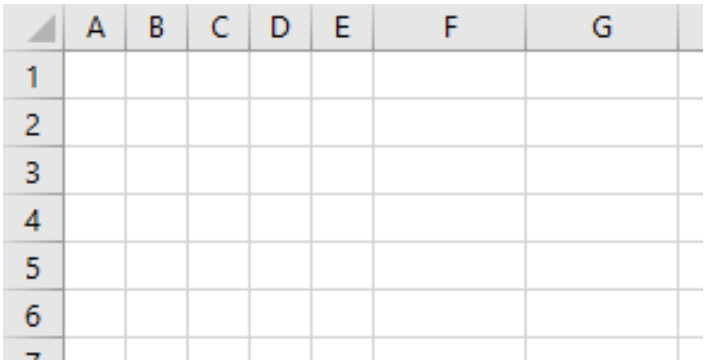


Building an effective communication tool by laying out an appropriate financial model is simple: it's all about designing and scoping. The problem is, we are all time poor in today's business environment with perpetual pressure on producing results more and more quickly. Getting a layout structure won't solve all of your problems but it's a start.

Let me show you how I develop this basic worksheet. Assuming this isn't a dashboard output page where column widths may be more critical, I tend to narrow the first few columns (highlight columns, then right-click and select 'Column width…' from the pop-up context menu):



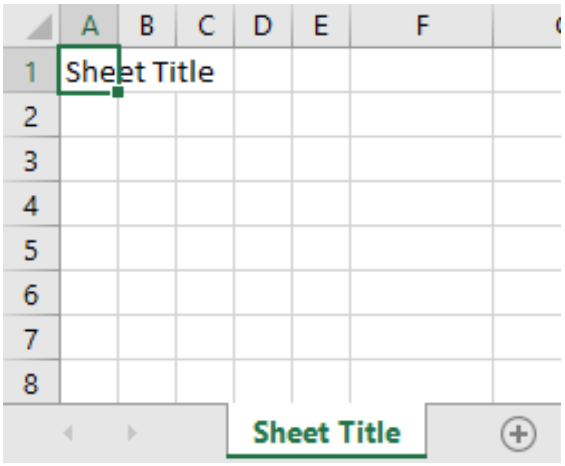
It may not be clear why I choose to do this, but read on (hey, I need to keep you on the edge of your seat!). I choose a width of 3 as this effectively makes the cells in these columns square.



You can elect to highlight a different number of columns and you can modify the width too. There are two key points to this:

1. Keep column **A** blank other than for the sheet headings (I will explain later)
2. Be consistent, both with the widths of the columns narrowed here and with other worksheets within the same workbook (again, I will explain soon).

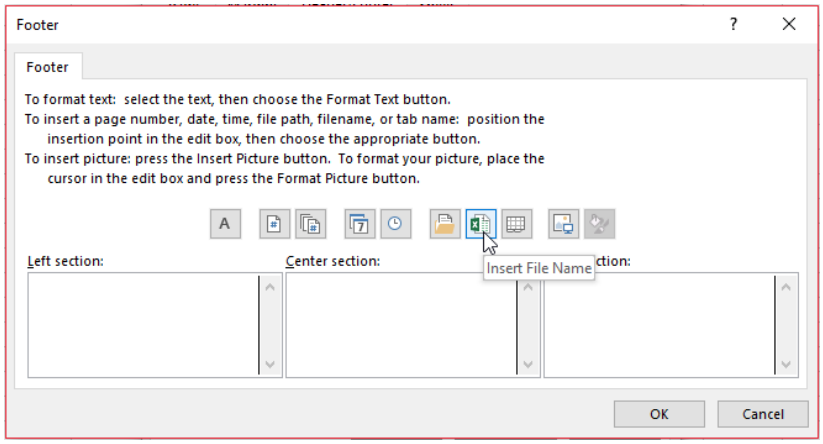
Next, let's put the sheet title in cell **A1**. This should be the same as the description in the sheet tab. For the purposes of this example, I am going to call it "Sheet Title" to emphasise the purpose of this placeholder, but without the quotation marks:



There are three reasons for this:

1. Given that sheet tab names cannot be infinitely long, sheet title is more succinct and easier for the end user to understand
2. Given that the sheet title appears on the worksheet, the name has to be written formally and cannot be an incomprehensible abbreviation, similar to many sheet tab names out there
3. This approach promotes consistency, one of the four key concepts of Best Practice modelling.

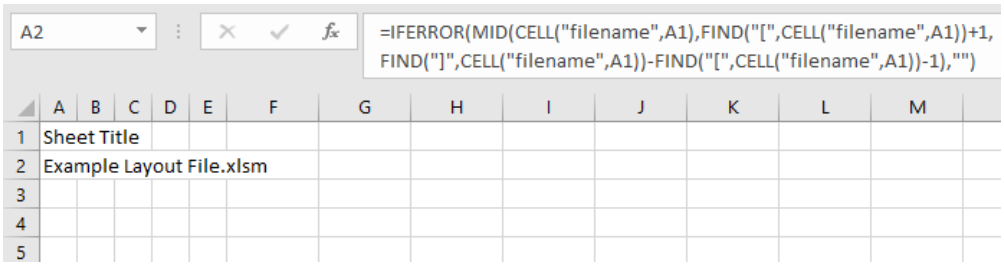
In cell **A2**, I will put the model name. This is important as often we only show extracts of a workbook in an appendix to a report or a PowerPoint presentation. Therefore, I place it here rather than in the header or footer of each worksheet instead (but you may do both):



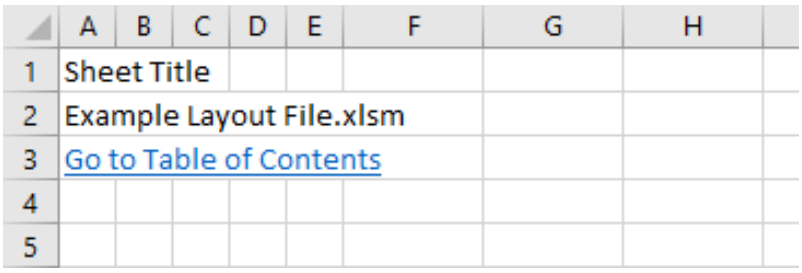
If you intend to add this to the header or footer, these attributes may be accessed by the keyboard shortcut **ALT + P + SP -> 'Header / Footer' tab -> 'Custom Footer…' button**. I acknowledge it's simpler than my alternative, but this filename will only display when the worksheet is printed. What if it is an image on a PowerPoint slide or, say, as Appendix 4 in a Word document? This is why I keep the model name front and centre on my worksheets.

There's a formula too:

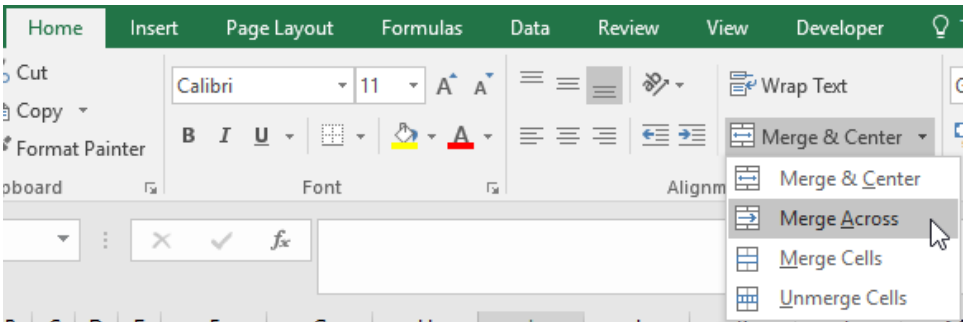
=IFERROR(MID(CELL("filename",A1),FIND("[",CELL("filename",A1))+1,FIND("]",CELL("filename",A1))-FIND("[",CELL("filename",A1))-1),"")



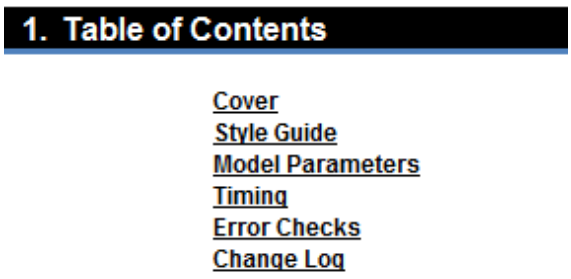
The next key item to position at the top of the sheet is a navigation aid. Today's workbooks can be quite complex with many worksheets. I recommend incorporating a central navigation page – a "Table of Contents" – which allows the end user to traverse the workbook quickly and easily.



It looks like I have added a hyperlink in cell **A3**, right? Not quite. I am a little craftier than that. Actually, I have highlighted cells **A3:F3** and then merged the cells using Excel's **Merge Across** functionality (**ALT + H + M + A**):

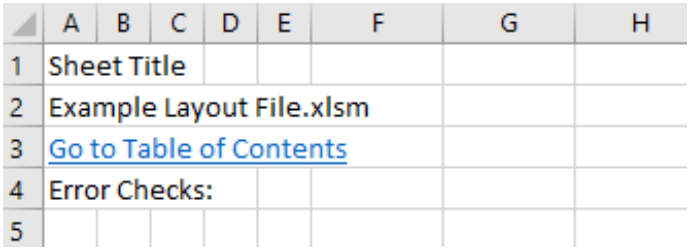


Hyperlinks may be created easily using the keyboard shortcut **CTRL + K** (or click the 'Link' button on the 'Insert' tab) to link to a place in your document. The intention is to set up a central Table of Contents worksheet where all of the hyperlinks to the other worksheets reside:

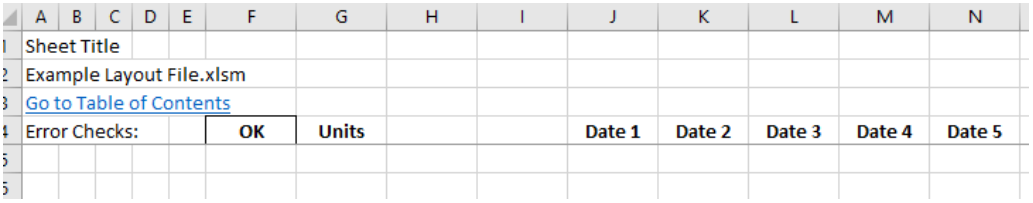


The hyperlink should link to cell **A1** *(say)* of that worksheet and that cell should have a range name such **HL_TOC**. A range name is essential in order to avoid a broken link should someone rename the destination worksheet and '**HL**' simply denotes that the cell is used as the destination for a **h**yper**l**ink. The reason cells **A3:F3** are merged is so that if the end user clicks anywhere in that range the hyperlink will activate; otherwise, the user will have to click on cell **A3** only for the hyperlink to work.

This brings us on nicely to cell **A4**:



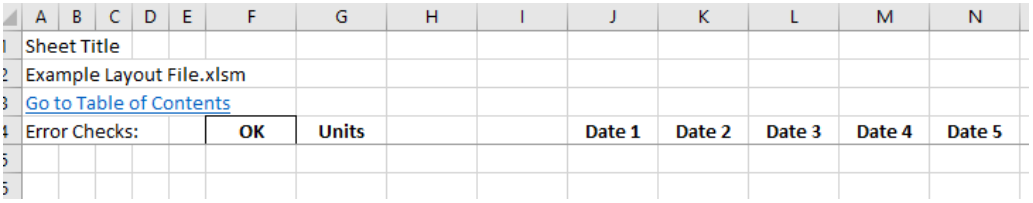
We can add a formula here to summarise all / any error checks present in the model. Typically, to conserve real estate on the worksheet, dates and other headings may share this row too:



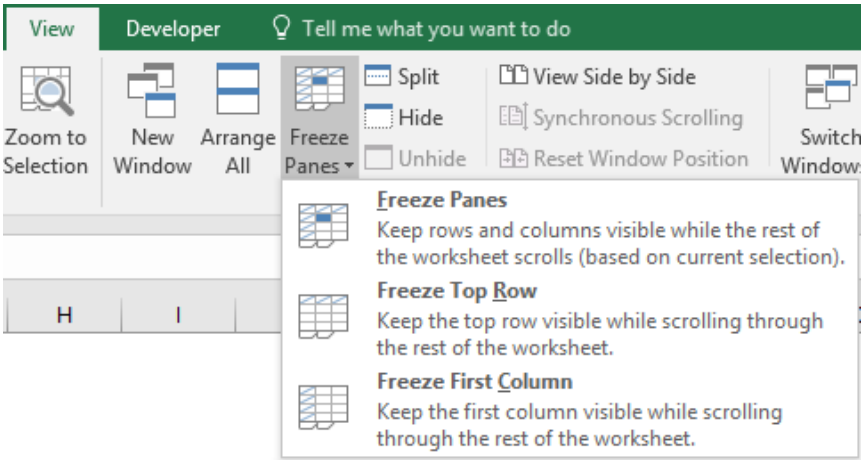
In my layout, I have made column **G** my **Units** column: down this column I shall put in all of my units so end users may distinguish between numerical fields. How often have you seen an output printed out and not known if it is in $, $'000, $m, kg or sliced tomatoes? This will make this issue a thing of the past. It should be noted that this column is not always required. For instance, on an outputs worksheet, you may simply state near the top of the sheet, "All outputs are displayed in $m unless stated otherwise".

Cells **J4:N4** contain the date headings. The dates should be periodic (*e.g.* monthly, quarterly, annually) and should always start and end in the same columns (and rows) on each forecast worksheet. That is not always possible: sometimes, you require some of you model to be annually forecast and other aspects monthly. Where different reporting periodicities are necessary, these inconsistent worksheets should be clearly delineated from other areas of the workbook.

You may have noticed as well that there is a line inserted in between rows 4 and 5 of our image:

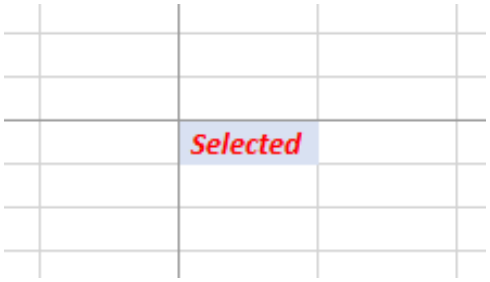


This is not a drawn line. This is called a frozen pane. Frozen panes break up the worksheet in to as many four pieces and allow parts of the worksheet to remain on view ("be frozen") whilst the reader scrolls down or across the worksheet. Located in the 'Window' grouping of the 'View' tab of the Ribbon, there are three ways to create a frozen pane:



- **Freeze top row:** Keeps the top row visible no matter how far down the spreadsheet you scroll
- **Freeze first column:** Keeps the first column visible no matter how far to the right you scroll the spreadsheet
- **Custom (Freeze Panes):** Creates a frozen locus at the intersection of the top row and the first column of the cell(s) selected.

That final option is a little confusing. Essentially the frozen panes are created as follows:



Frozen panes are created for the region the selection is in, the region directly above, the region to the immediate left and diagonally opposite the top left-hand corner of the selection. If the selection were in column **A**, there would only be two frozen panes: the rows immediately above and the remainder of the sheet. If the selection were in row 1, again, there would only be two frozen panes: the columns to the left and the remainder.

In our example, cell **A5** has been made the basis of the frozen pane, so that rows 1 to 4 will always be visible. This cell should be given a range name, *e.g.* **HL_Home**, as this is the cell hyperlinks to this sheet should link. This cell 'resets' the sheet when a frozen pane has been added (not cell **A1**) and makes the model easier to navigate consequently. This cell can always be identified by employing the keyboard shortcut **CTRL + HOME**.

At this point, let me revisit my unexplained narrowing of the first few columns. I would suggest headings should start in column **B**, not **A**, and then move out a column or two for sub headings and sub sub headings respectively. This causes a natural indentation. I then put data labels directly beneath sub sub headings:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Sheet Title													
2	Example Layout File.xlsm													
3	Go to Table of Contents													
4	Error Checks:					OK	Units			Date 1	Date 2	Date 3	Date 4	Date 5
5														
6		Main Heading												
7														
8			Sub Heading											
9														
10				Sub Sub Heading										
11				Label										
12				Label										
13				Label										
14				Label										
15				Label										
16														

I have called them "Headings" and "Sub Headings" *etc.* to make it clear, but this approach will become cumbersome quickly. Renaming the headings "Heading 1" and so is clearer. This also makes them consistent with pre-existing Style names *(hint, hint)*:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Sheet Title													
2	Example Layout File.xlsm													
3	Go to Table of Contents													
4	Error Checks:					OK	Units			Date 1	Date 2	Date 3	Date 4	Date 5
5														
6		Heading 1												
7														
8			Heading 2											
9														
10				Heading 3										
11				Label										
12				Label										
13				Label										
14				Label										
15				Label										
16														
17														
18		Heading 1												
19														
20			Heading 2											
21														
22				Heading 3										
23				Label										
24				Label										
25				Label										
26				Label										
27				Label										
28														

Aside from keeping column **A** clear, do you now see why I have narrowed columns **B**, **C** and **D** (I am keeping column **E** "just in case")? The narrowing of the columns effectively indents the headings and makes worksheets easier to read and navigate (especially if the gridlines, **ALT + W + VG**, are toggled off).

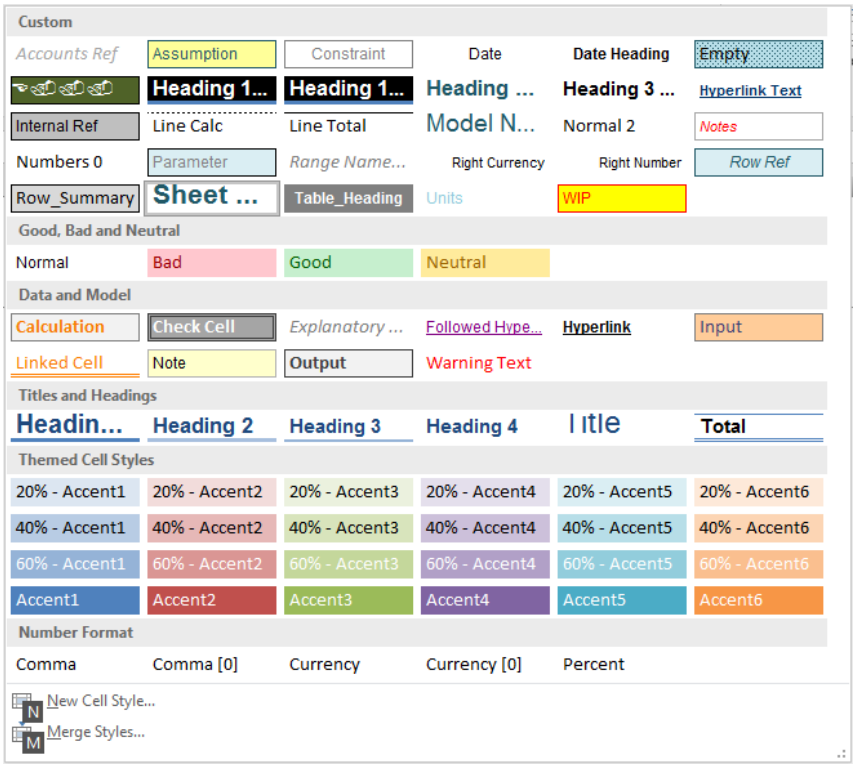
Take special note of the spacing: one blank row between headings; two lines between sections. That's my preference. You choose your own if you would prefer – just be consistent. It makes it very simple to copy sections and keep referencing if spacing is deliberate.

Blank columns **H** and **I** are in existence in case we have any calculations, inputs or referred values that do not refer to a particular time period. If they are not required, I tend to narrow the columns to a width of 1 (say), so that they are still there in case they are needed later.

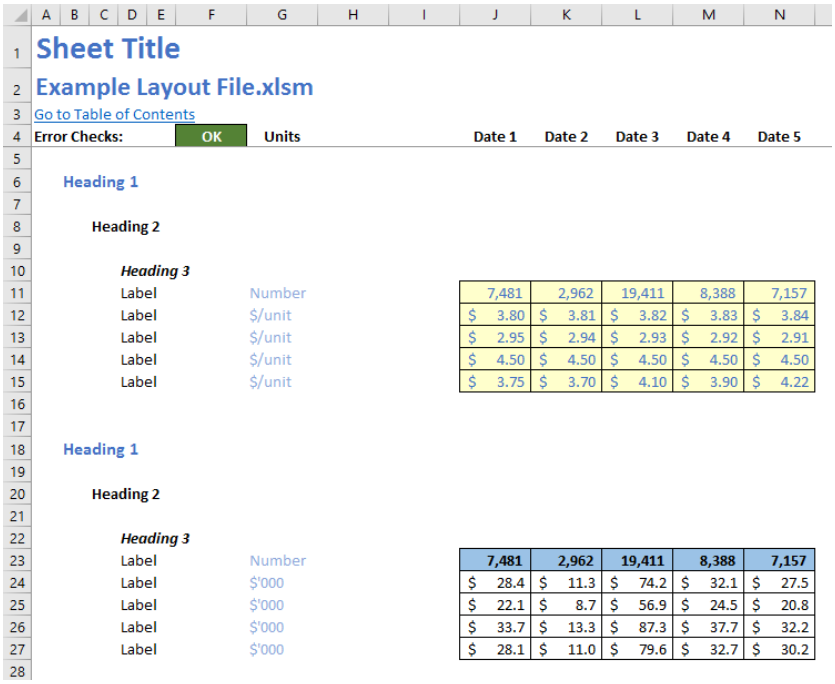
Adding labels, data and formulae:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Sheet Title													
2	Example Layout File.xlsm													
3	Go to Table of Contents													
4	Error Checks:					OK	Units			Date 1	Date 2	Date 3	Date 4	Date 5
5														
6		Heading 1												
7														
8			Heading 2											
9														
10				Heading 3										
11				Label			Number			7481	2962	19411	8388	7157
12				Label			$/unit			3.8	3.81	3.82	3.83	3.84
13				Label			$/unit			2.95	2.94	2.93	2.92	2.91
14				Label			$/unit			4.5	4.5	4.5	4.5	4.5
15				Label			$/unit			3.75	3.7	4.1	3.9	4.22
16														
17														
18		Heading 1												
19														
20			Heading 2											
21														
22				Heading 3										
23				Label			Number			7481	2962	19411	8388	7157
24				Label			$'000			28427.8	11285.22	74150.02	32126.04	27482.88
25				Label			$'000			22068.95	8708.28	56874.23	24492.96	20826.87
26				Label			$'000			33664.5	13329	87349.5	37746	32206.5
27				Label			$'000			28053.75	10959.4	79585.1	32713.2	30202.54
28														

It's starting to look more like a spreadsheet now. The next step is to incorporate Styles (**ALT + H + J**):



Very quickly, our spreadsheet is taking shape:



1	**Sheet Title**													
2	**Example Layout File.xlsm**													
3	Go to Table of Contents													
4	**Error Checks:**		**OK**		**Units**					**Date 1**	**Date 2**	**Date 3**	**Date 4**	**Date 5**
5														
6	**Heading 1**													
7														
8	**Heading 2**													
9														
10	***Heading 3***													
11	Label				Number					7,481	2,962	19,411	8,388	7,157
12	Label				$/unit					$ 3.80	$ 3.81	$ 3.82	$ 3.83	$ 3.84
13	Label				$/unit					$ 2.95	$ 2.94	$ 2.93	$ 2.92	$ 2.91
14	Label				$/unit					$ 4.50	$ 4.50	$ 4.50	$ 4.50	$ 4.50
15	Label				$/unit					$ 3.75	$ 3.70	$ 4.10	$ 3.90	$ 4.22
16														
17														
18	**Heading 1**													
19														
20	**Heading 2**													
21														
22	***Heading 3***													
23	Label				Number					**7,481**	**2,962**	**19,411**	**8,388**	**7,157**
24	Label				$'000					$ 28.4	$ 11.3	$ 74.2	$ 32.1	$ 27.5
25	Label				$'000					$ 22.1	$ 8.7	$ 56.9	$ 24.5	$ 20.8
26	Label				$'000					$ 33.7	$ 13.3	$ 87.3	$ 37.7	$ 32.2
27	Label				$'000					$ 28.1	$ 11.0	$ 79.6	$ 32.7	$ 30.2
28														

If I switch off gridlines on my spreadsheets, then the majority of my files appear to have a white background. There is more to this point than merely aesthetics. Adding a colour to the background of a spreadsheet can make a file significantly larger – unnecessarily.

The spacing is deliberate too. Not only does it look neater (remember, Excel 2007 onwards has 1,048,576 rows and 16,384 columns, *i.e.* it is 1,024 times larger than an Excel 2003 worksheet so there is plenty of room), but the space is functional too.

Want to navigate between the main headings in column **B**? Click on cell **B6**, go **CTRL + Down Arrow** and you will arrive at cell **B18**. Repeat this action and the next cell you will hit is cell **B1048576**, *i.e.* the very bottom of the spreadsheet because there is nothing else in this column.

Click on cell **D10** (**Heading 3**) and use the keyboard shortcut **CTRL + Down Arrow** to take you to cell **D15**, the final cell in the contiguous range. **CTRL + Up Arrow**, **CTRL + Right Arrow** and **CTRL + Left Arrow** will all perform similar actions. Need to highlight a range? Click on any cell within the range and **CTRL + A** will select the whole contiguous range. This makes the model easier for developer and user alike to navigate and manipulate.

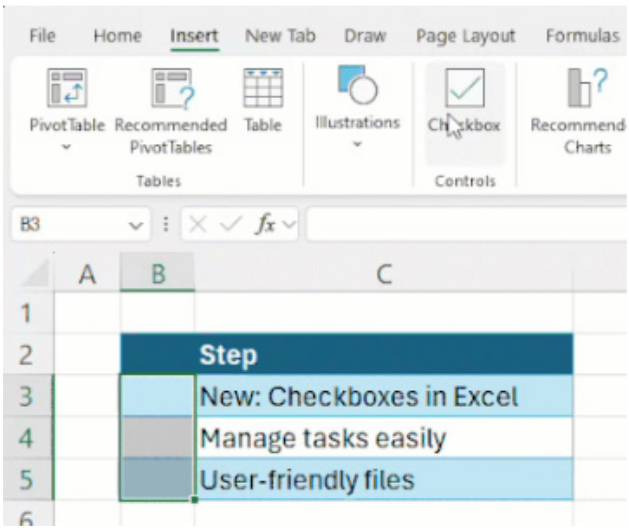
So why have I kept column **A** blank? The reason is to consider work in progress. How often have you started creating a spreadsheet only to be interrupted, have to go to a meeting, take a telephone call, go home or go to sleep? The point is, when we are interrupted we need to remember how far along we were. If you design a spreadsheet similar to the one discussed here, imagine you are interrupted without notice. Before you turn your attention to the disruption, whichever row you are working on, press the **HOME** key which will take you to column **A** of that row. Type anything in that cell, *e.g.* "w" for "work in progress" or "check" and so on. That's it.

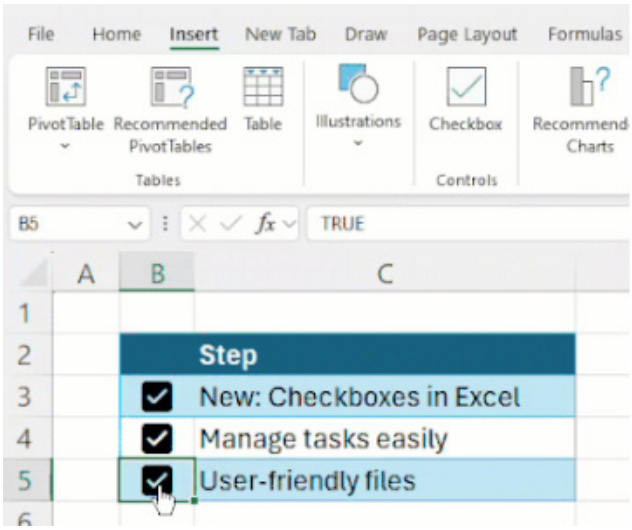
Word to the Wise

Whatever you decide to do, keep it consistent, make it transparent, ensure there are checks to protect the robustness and that inputs are clearly marked to aid flexibility. Any layout addressing these points will necessarily adhere to the **CRaFT** ideology.

Checkboxes in Excel

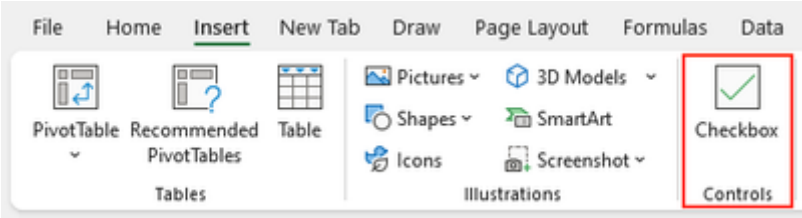
Checkboxes are finally released to production in Excel. These are useful for checklists (obviously!), managing tasks and visualising your data quickly. With just a few clicks, you can insert Checkboxes into any cell, making your spreadsheets more dynamic and user-friendly.





To insert Checkboxes:

- select the range where you want the Checkboxes
- select **Insert -> Checkbox**.



To check or uncheck a Checkbox:

- click on the checkbox
- Select one or more checkboxes and press **SPACE**.

To remove Checkboxes:

- press the **DELETE** key
- if any of the Checkboxes were checked, **DELETE** will first uncheck them. Press **DELETE** again to remove them.

These Checkboxes should be available to you all by the time you read this here!

- Windows and Mac Desktop: rollout began June 26
- on Web and Mobile (iPad, iOS & Android): coming soon.

Microsoft expects all users on Current Channel to have access by the end of July.

New Translation Functions in Excel

There are two new Preview functions coming to Excel. Be careful using these: their signature and results may change substantially before being broadly released, based upon feedback from those fortunate enough to be able to access them. Therefore, we strongly recommend you do not rely on these functions in important workbooks until they are Generally Available.

The two new translation functions out in Preview now are **TRANSLATE** and **DETECTLANGUAGE**.

Microsoft is introducing these two new functions to simplify and automate translations directly within your spreadsheet:

- **TRANSLATE():** a function that translates a text from one language to another
- **DETECTLANGUAGE():** a function that detects the language of the specified text.

TRANSLATE

TRANSLATE takes text you provide and translates it from one language to another using Microsoft Translation Services. Currently, there are 133 languages supported – including two variations of Klingon!!

At the time of writing, these are:

1. Afrikaans
2. Albanian
3. Amharic
4. Arabic
5. Armenian
6. Assamese
7. Azerbaijani (Latin)
8. Bangla
9. Bashkir
10. Basque
11. Bhojpuri
12. Bodo
13. Bosnian (Latin)
14. Bulgarian
15. Cantonese (Traditional)
16. Catalan
17. Chinese (Literary)
18. Chinese Simplified
19. Chinese Traditional
20. chiShona
21. Croatian
22. Czech
23. Danish
24. Dari
25. Divehi
26. Dogri
27. Dutch
28. English
29. Estonian
30. Faroese
31. Fijian
32. Filipino
33. Finnish
34. French
35. French (Canada)
36. Galician
37. Georgian
38. German
39. Greek
40. Gujarati
41. Haitian Creole
42. Hausa
43. Hebrew
44. Hindi
45. Hmong Daw (Latin)
46. Hungarian
47. Icelandic
48. Igbo
49. Indonesian

50. Inuinnaqtun
51. Inuktitut
52. Inuktitut (Latin)
53. Irish
54. Italian
55. Japanese
56. Kannada
57. Kashmiri
58. Kazakh
59. Khmer
60. Kinyarwanda
61. Klingon
62. Klingon (plqaD)
63. Konkani
64. Korean ko
65. Kurdish (Central)
66. Kurdish (Northern)
67. Kyrgyz (Cyrillic)
68. Lao
69. Latvian
70. Lithuanian
71. Lingala
72. Lower Sorbian
73. Luganda
74. Macedonian
75. Maithili
76. Malagasy
77. Malay (Latin)
78. Malayalam
79. Maltese
80. Maori
81. Marathi
82. Mongolian (Cyrillic)
83. Mongolian (Traditional)
84. Myanmar
85. Nepali
86. Norwegian
87. Nyanja
88. Odia
89. Pashto
90. Persian
91. Polish
92. Portuguese (Brazil)
93. Portuguese (Portugal)
94. Punjabi
95. Queretaro Otomi
96. Romanian
97. Rundi
98. Russian
99. Samoan (Latin)
100. Serbian (Cyrillic)
101. Serbian (Latin)
102. Sesotho
103. Sesotho sa Leboa
104. Setswana
105. Sindhi
106. Sinhala
107. Slovak
108. Slovenian
109. Somali (Arabic)
110. Spanish es
111. Swahili (Latin)
112. Swedish
113. Tahitian
114. Tamil
115. Tatar (Latin)
116. Telugu
117. Thai
118. Tibetan
119. Tigrinya
120. Tongan
121. Turkish
122. Turkmen (Latin)
123. Ukrainian
124. Upper Sorbian
125. Urdu
126. Uyghur (Arabic)
127. Uzbek (Latin)
128. Vietnamese
129. Welsh
130. Xhosa
131. Yoruba
132. Yucatec Maya
133. Zulu.

As mentioned above, the **TRANSLATE** function allows you to translate text from one language to another in Microsoft Excel by using Microsoft Translation Services. The full signature is:

TRANSLATE(text, [source_language], [target_language])

This function has the following arguments:

- **text:** the **text** to translate. This value should either be enclosed in quotation marks or be a reference to a cell containing the appropriate **text**
- **source_language (optional):** the language code of the source language (*e.g.* "en" for English or "es" for Spanish). If not specified, the language will be automatically detected based upon the **text** provided. Auto-detection is supported for most languages. It is recommended to specify the language if known, especially for shorter texts
- **target_language (optional):** the language code of the target language (*e.g.* "en" for English or "es" for Spanish). If not specified, the system language will be used as the target language.

The supported languages and their respective language codes are as follows:

Language	Language code
Afrikaans	af
Albanian	sq
Amharic	am
Arabic	ar
Armenian	hy
Assamese	as
Azerbaijani (Latin)	az
Bangla	bn
Bashkir	ba
Basque	eu
Bhojpuri	bho
Bodo	brx
Bosnian (Latin)	bs
Bulgarian	bg
Cantonese (Traditional)	yue
Catalan	ca
Chinese (Literary)	lzh
Chinese Simplified	zh-Hans
Chinese Traditional	zh-Hant
chiShona	sn
Croatian	hr
Czech	cs
Danish	da
Dari	prs
Divehi	dv
Dogri	doi
Dutch	nl
English	en
Estonian	et
Faroese	fo
Fijian	fj

Language	Language code
Filipino	fil
Finnish	fi
French	fr
French (Canada)	fr-ca
Galician	gl
Georgian	ka
German	de
Greek	el
Gujarati	gu
Haitian Creole	ht
Hausa	ha
Hebrew	he
Hindi	hi
Hmong Daw (Latin)	mww
Hungarian	hu
Icelandic	is
Igbo	ig
Indonesian	id
Inuinnaqtun	ikt
Inuktitut	iu
Inuktitut (Latin)	iu-Latn
Irish	ga
Italian	it
Japanese	ja
Kannada	kn
Kashmiri	ks
Kazakh	kk
Khmer	km
Kinyarwanda	rw
Klingon	tlh-Latn
Klingon (plqaD)	tlh-Piqd

Language	Language code
Konkani	gom
Korean	ko
Kurdish (Central)	ku
Kurdish (Northern)	kmr
Kyrgyz (Cyrillic)	ky
Lao	lo
Latvian	lv
Lithuanian	lt
Lingala	ln
Lower Sorbian	dsb
Luganda	lug
Macedonian	mk
Maithili	mai
Malagasy	mg
Malay (Latin)	ms
Malayalam	ml
Maltese	mt
Maori	mi
Marathi	mr
Mongolian (Cyrillic)	mn-Cyrl
Mongolian (Traditional)	mn-Mong
Myanmar	my
Nepali	ne
Norwegian	nb
Nyanja	nya
Odia	or
Pashto	ps
Persian	fa
Polish	pl
Portuguese (Brazil)	pt
Portuguese (Portugal)	pt-pt

Language	Language code
Punjabi	pa
Queretaro Otomi	otq
Romanian	ro
Rundi	run
Russian	ru
Samoan (Latin)	sm
Serbian (Cyrillic)	sr-Cyrl
Serbian (Latin)	sr-Latn
Sesotho	st
Sesotho sa Leboa	nso
Setswana	tn
Sindhi	sd
Sinhala	si
Slovak	sk
Slovenian	sl
Somali (Arabic)	so
Spanish	es
Swahili (Latin)	sw
Swedish	sv
Tahitian	ty
Tamil	ta
Tatar (Latin)	tt
Telugu	te
Thai	th
Tibetan	bo
Tigrinya	ti
Tongan	to
Turkish	tr
Turkmen (Latin)	tk
Ukrainian	uk
Upper Sorbian	hsb

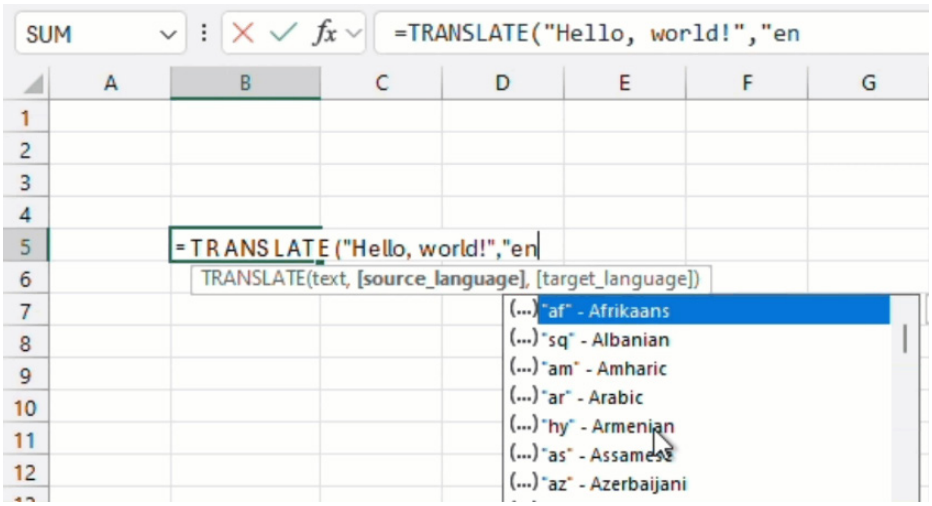
Language	Language code
Urdu	ur
Uyghur (Arabic)	ug
Uzbek (Latin)	uz
Vietnamese	vi
Welsh	cy
Xhosa	xh
Yoruba	yo
Yucatec Maya	yua
Zulu	zu

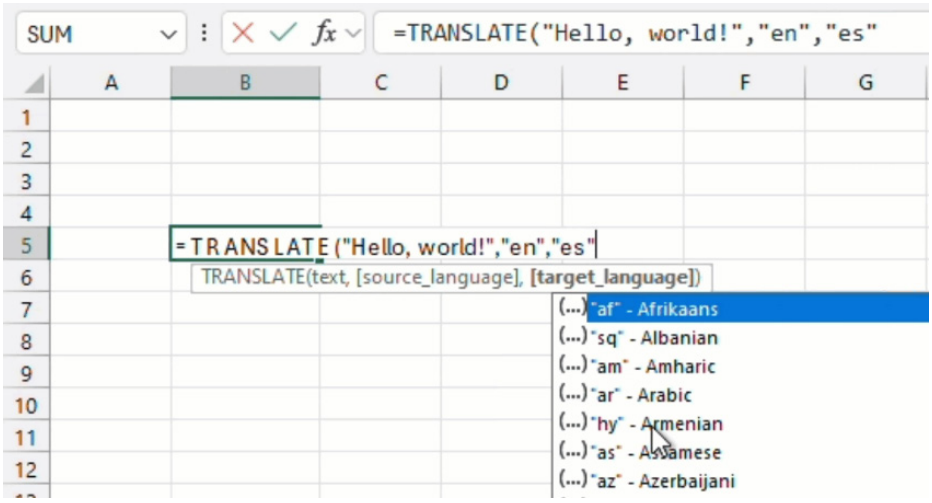
Suppose you have the following text in cell **A1**: "Hello, World!" and you want to translate it to Spanish. You can use the **TRANSLATE** function as follows:

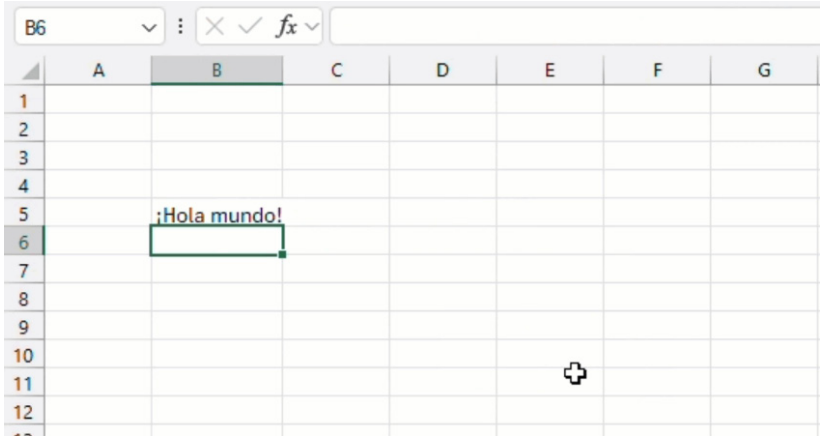
=TRANSLATE(A1, "en", "es")

In this example, the source language is English (en) and the target language is Spanish (es). The translated text, "Hola mundo!" will be displayed in the cell where you entered the formula.

Alternatively, you may just type the text in, *viz.*







Common errors include the following:

- **Text Too Long:** you have too many characters in a cell. Reduce your cell size and try again
- **Error in Value:** you have a non-text value in your cell. The function only accepts a text argument
- **Invalid Language:** you have entered an invalid language code or one not presently supported (see above)
- **Request Throttled:** you have exceeded your daily quota of the translation function (now that is interesting, but we are not quite sure what that means at the time of writing).

DETECTLANGUAGE

DETECTLANGAUGE detects the language of text you provide using the Microsoft Translation Services and returns the language code. The full signature is:

DETECTLANGUAGE(text)

The function has the following arguments:

- **text:** the **text** or reference to cells containing **text** to evaluate.

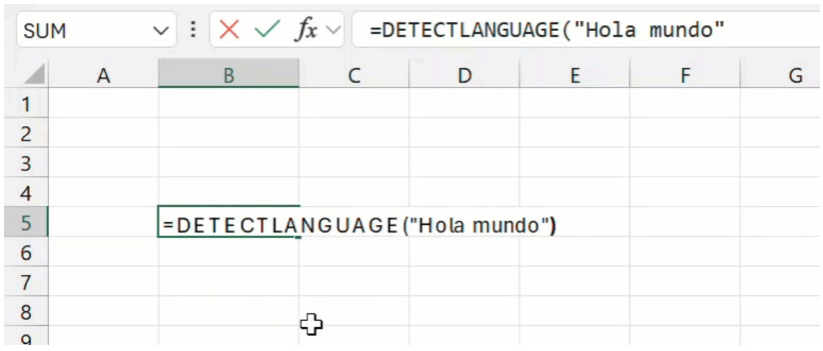
The supported languages and their respective language codes are as above.

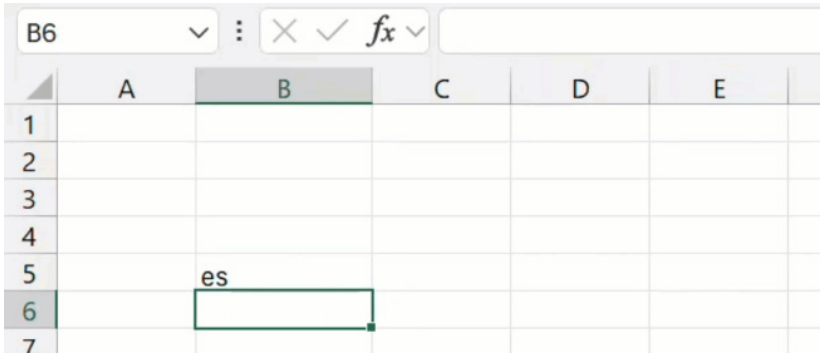
Suppose you have the following text in cell **A1**: "Hola mundo!" and you want to find out what the language of the text is. You can use the **DETECTLANGUAGE** function as follows:

=DETECTLANGUAGE(A1)

This will return the detected language for the text in cell **A1**. The language code "es" for Spanish will be displayed in the cell where you entered the formula.

Alternatively, you may just type the text in, *viz.*





Common errors include the following:

- **Text Too Long:** you have too many characters in a cell. Reduce your cell size and try again
- **Error in Value:** you have a non-text value in your cell. The function only accepts a text argument
- **Invalid Language:** you have entered an invalid language code or one not presently supported (see above)
- **Request Throttled:** you have exceeded your daily quota of the translation function.

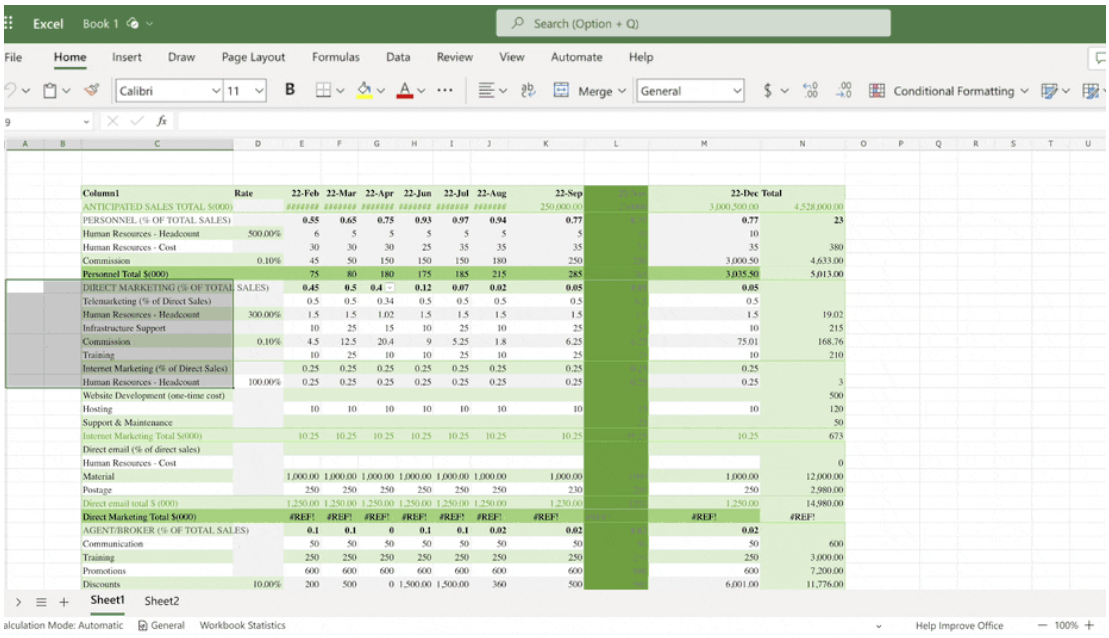
These functions are currently available to *some* Beta Channel users running:

- Windows: Version 2407 (Build 16.0.17808.20000) or later
- Mac: 16.87 (Build 24062430) or later.

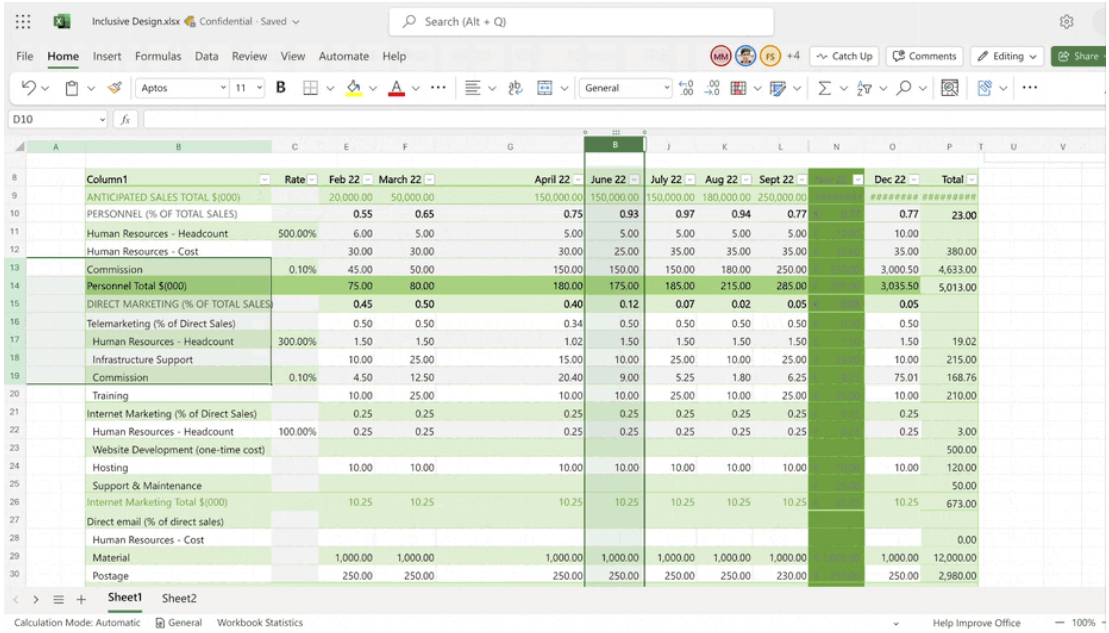
We say “some” as we haven’t access *yet*. ☹

Revamped Excel Grid

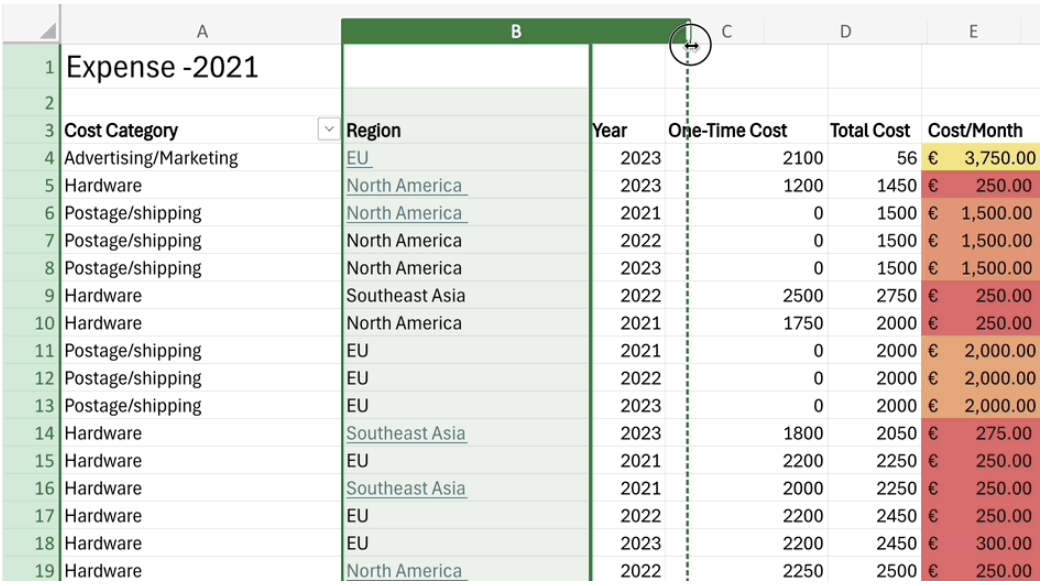
Not yet detailed in the Excel Updates *(see below)*, one update for web users provides versatile features designed to make your spreadsheet tasks simpler and more efficient. Essentially Excel for the web is transforming from this:



to this:



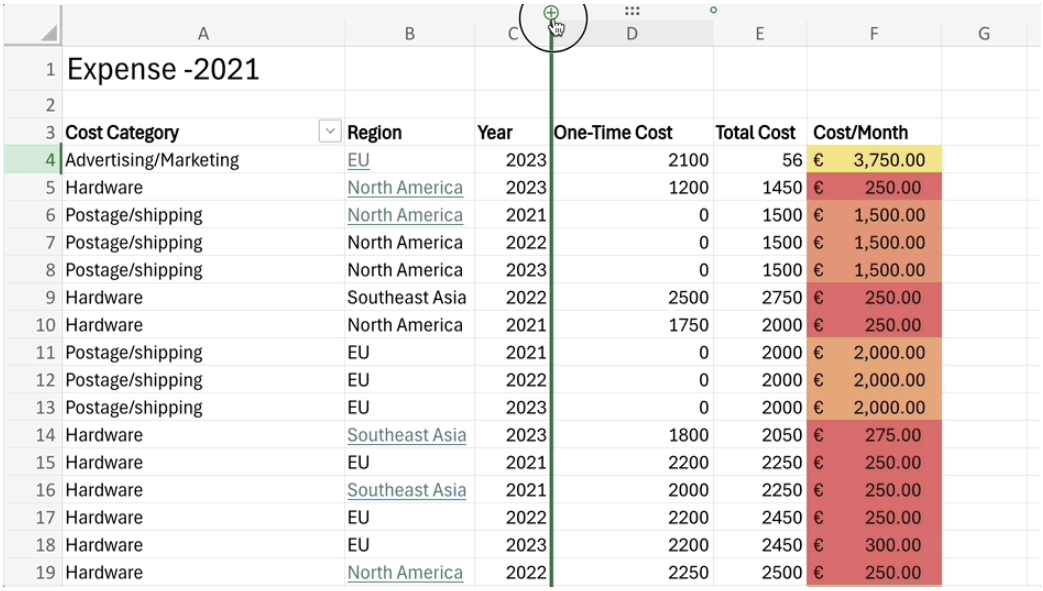
You may now quickly resize rows and columns for better data visibility and presentation. All you need to do is hover over the border of a row or column header, click and hold the handles, and then drag to resize.



Expense -2021

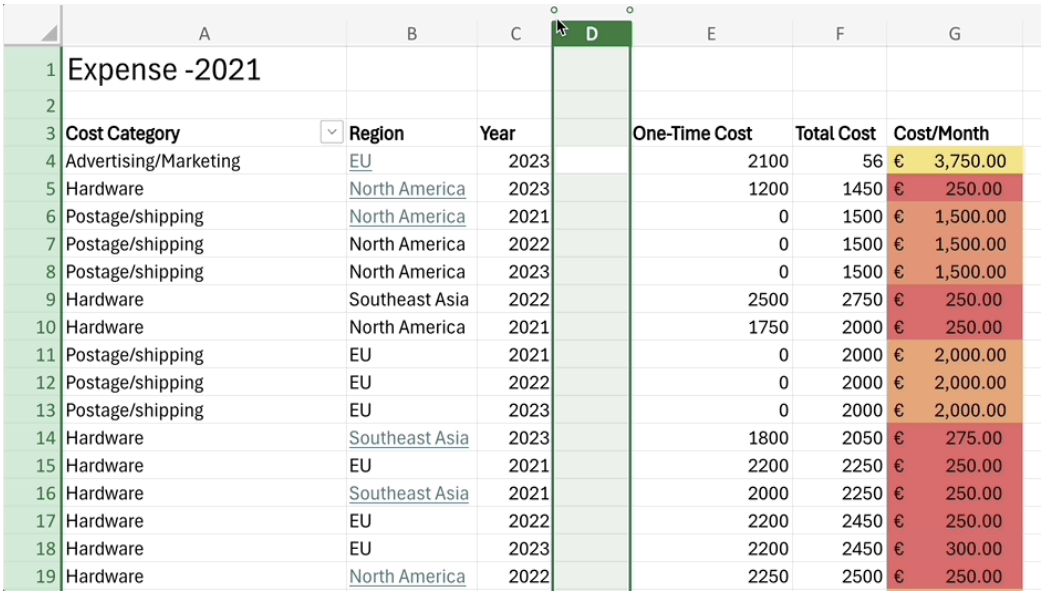
Cost Category	Region	Year	One-Time Cost	Total Cost	Cost/Month
Advertising/Marketing	EU	2023	2100	56	€ 3,750.00
Hardware	North America	2023	1200	1450	€ 250.00
Postage/shipping	North America	2021	0	1500	€ 1,500.00
Postage/shipping	North America	2022	0	1500	€ 1,500.00
Postage/shipping	North America	2023	0	1500	€ 1,500.00
Hardware	Southeast Asia	2022	2500	2750	€ 250.00
Hardware	North America	2021	1750	2000	€ 250.00
Postage/shipping	EU	2021	0	2000	€ 2,000.00
Postage/shipping	EU	2022	0	2000	€ 2,000.00
Postage/shipping	EU	2023	0	2000	€ 2,000.00
Hardware	Southeast Asia	2023	1800	2050	€ 275.00
Hardware	EU	2021	2200	2250	€ 250.00
Hardware	Southeast Asia	2021	2000	2250	€ 250.00
Hardware	EU	2022	2200	2450	€ 250.00
Hardware	EU	2023	2200	2450	€ 300.00
Hardware	North America	2022	2250	2500	€ 250.00

There is also a new simplified interface which makes adding rows, or columns to your spreadsheet straightforward. Just hover over the respective row or column header and then click on the small circles (convert to **+** on hover).



Expense -2021

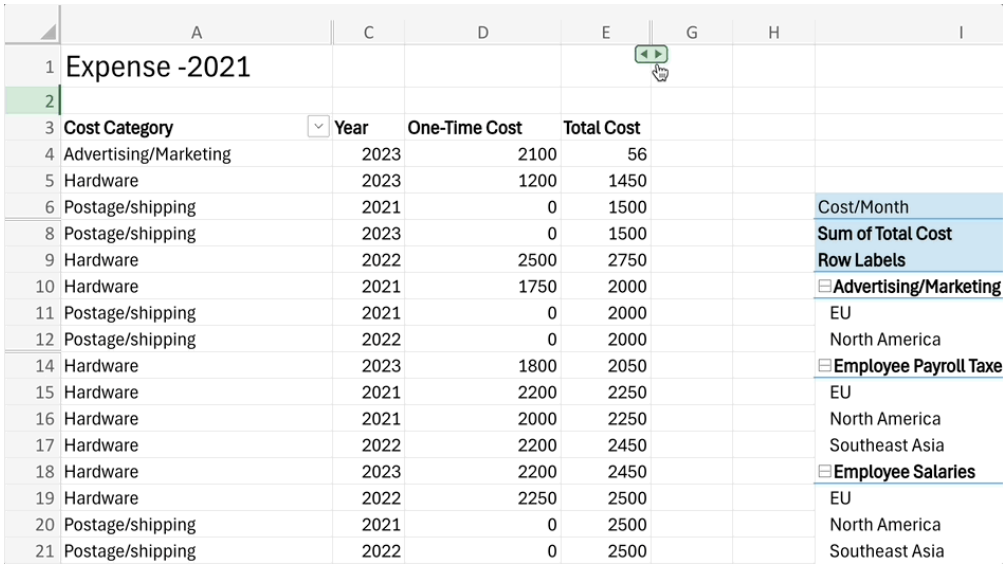
Cost Category	Region	Year	One-Time Cost	Total Cost	Cost/Month
Advertising/Marketing	EU	2023	2100	56	€ 3,750.00
Hardware	North America	2023	1200	1450	€ 250.00
Postage/shipping	North America	2021	0	1500	€ 1,500.00
Postage/shipping	North America	2022	0	1500	€ 1,500.00
Postage/shipping	North America	2023	0	1500	€ 1,500.00
Hardware	Southeast Asia	2022	2500	2750	€ 250.00
Hardware	North America	2021	1750	2000	€ 250.00
Postage/shipping	EU	2021	0	2000	€ 2,000.00
Postage/shipping	EU	2022	0	2000	€ 2,000.00
Postage/shipping	EU	2023	0	2000	€ 2,000.00
Hardware	Southeast Asia	2023	1800	2050	€ 275.00
Hardware	EU	2021	2200	2250	€ 250.00
Hardware	Southeast Asia	2021	2000	2250	€ 250.00
Hardware	EU	2022	2200	2450	€ 250.00
Hardware	EU	2023	2200	2450	€ 300.00
Hardware	North America	2022	2250	2500	€ 250.00



Expense -2021

Cost Category	Region	Year	One-Time Cost	Total Cost	Cost/Month
Advertising/Marketing	EU	2023	2100	56	€ 3,750.00
Hardware	North America	2023	1200	1450	€ 250.00
Postage/shipping	North America	2021	0	1500	€ 1,500.00
Postage/shipping	North America	2022	0	1500	€ 1,500.00
Postage/shipping	North America	2023	0	1500	€ 1,500.00
Hardware	Southeast Asia	2022	2500	2750	€ 250.00
Hardware	North America	2021	1750	2000	€ 250.00
Postage/shipping	EU	2021	0	2000	€ 2,000.00
Postage/shipping	EU	2022	0	2000	€ 2,000.00
Postage/shipping	EU	2023	0	2000	€ 2,000.00
Hardware	Southeast Asia	2023	1800	2050	€ 275.00
Hardware	EU	2021	2200	2250	€ 250.00
Hardware	Southeast Asia	2021	2000	2250	€ 250.00
Hardware	EU	2022	2200	2450	€ 250.00
Hardware	EU	2023	2200	2450	€ 300.00
Hardware	North America	2022	2250	2500	€ 250.00

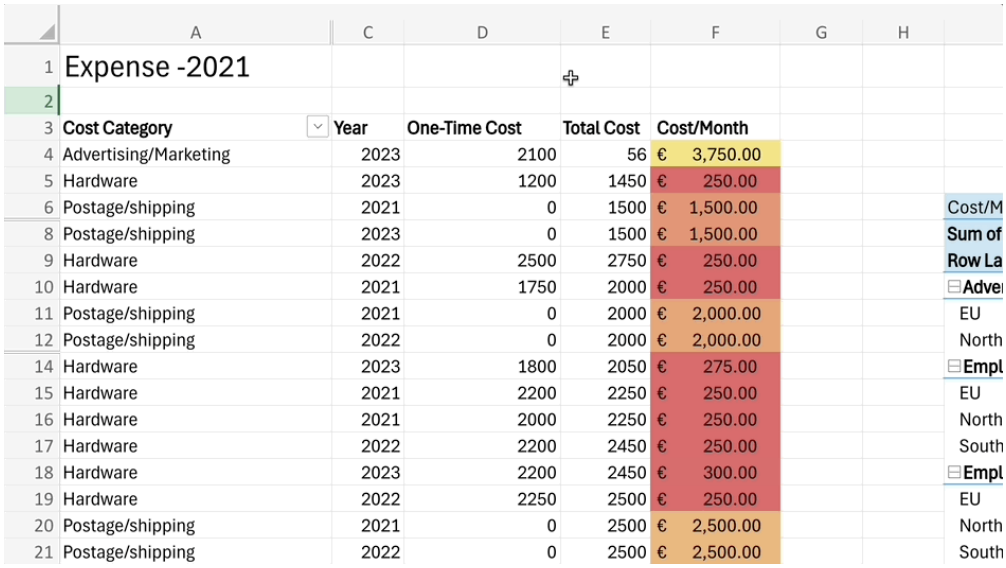
You may also show hidden rows or columns with one click and get a complete view of your data instantly. Just hover over the row or column header and then select the small arrows that appear.



Expense -2021

Cost Category	Year	One-Time Cost	Total Cost
Advertising/Marketing	2023	2100	56
Hardware	2023	1200	1450
Postage/shipping	2021	0	1500
Postage/shipping	2023	0	1500
Hardware	2022	2500	2750
Hardware	2021	1750	2000
Postage/shipping	2021	0	2000
Postage/shipping	2022	0	2000
Hardware	2023	1800	2050
Hardware	2021	2200	2250
Hardware	2021	2000	2250
Hardware	2022	2200	2450
Hardware	2023	2200	2450
Hardware	2022	2250	2500
Postage/shipping	2021	0	2500
Postage/shipping	2022	0	2500

Cost/Month
Sum of Total Cost
Row Labels
Advertising/Marketing
EU
North America
Employee Payroll Taxe
EU
North America
Southeast Asia
Employee Salaries
EU
North America
Southeast Asia

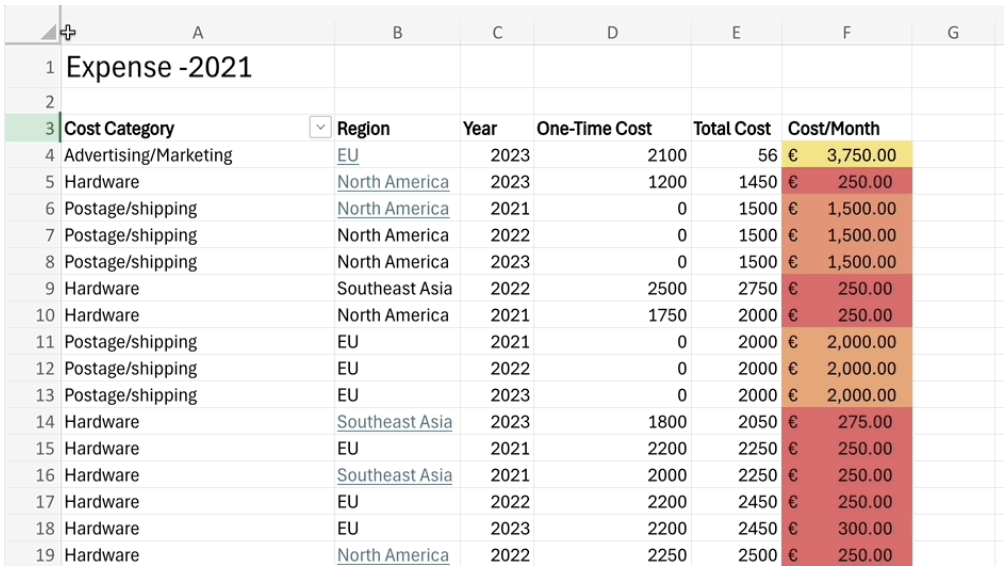


Expense -2021

Cost Category	Year	One-Time Cost	Total Cost	Cost/Month
Advertising/Marketing	2023	2100	56	€ 3,750.00
Hardware	2023	1200	1450	€ 250.00
Postage/shipping	2021	0	1500	€ 1,500.00
Postage/shipping	2023	0	1500	€ 1,500.00
Hardware	2022	2500	2750	€ 250.00
Hardware	2021	1750	2000	€ 250.00
Postage/shipping	2021	0	2000	€ 2,000.00
Postage/shipping	2022	0	2000	€ 2,000.00
Hardware	2023	1800	2050	€ 275.00
Hardware	2021	2200	2250	€ 250.00
Hardware	2021	2000	2250	€ 250.00
Hardware	2022	2200	2450	€ 250.00
Hardware	2023	2200	2450	€ 300.00
Hardware	2022	2250	2500	€ 250.00
Postage/shipping	2021	0	2500	€ 2,500.00
Postage/shipping	2022	0	2500	€ 2,500.00

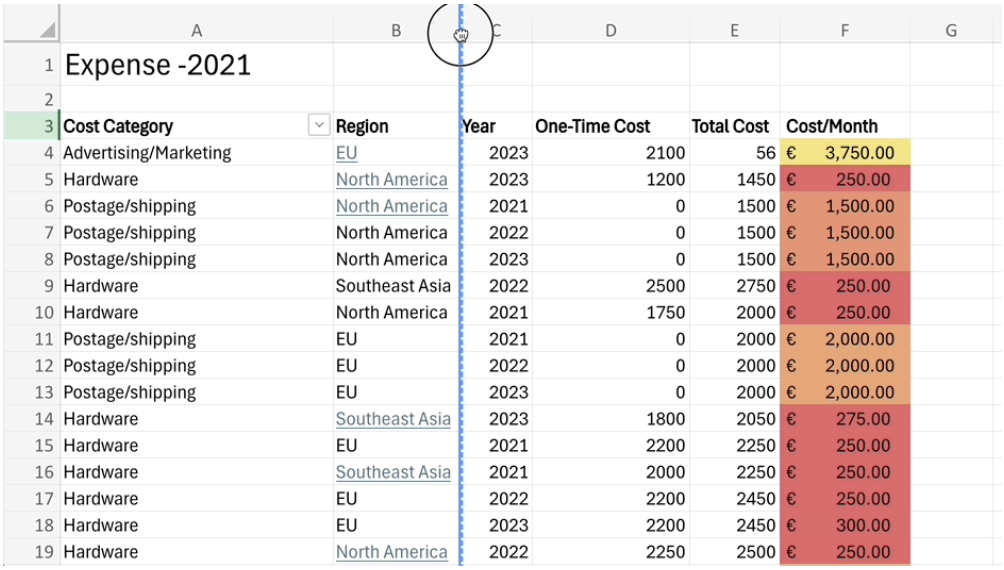
Cost/M
Sum of
Row La
Adve
EU
North
Empl
EU
North
South
Empl
EU
North
South

You may also keep important headers or columns visible as you scroll to ensure that important information stays visible, no matter how far you scroll down or across your spreadsheet. To do so, drag the handles in the top left corner of the headers and drag them to the desired position. To change existing freeze panes, just drag the freeze pane line.



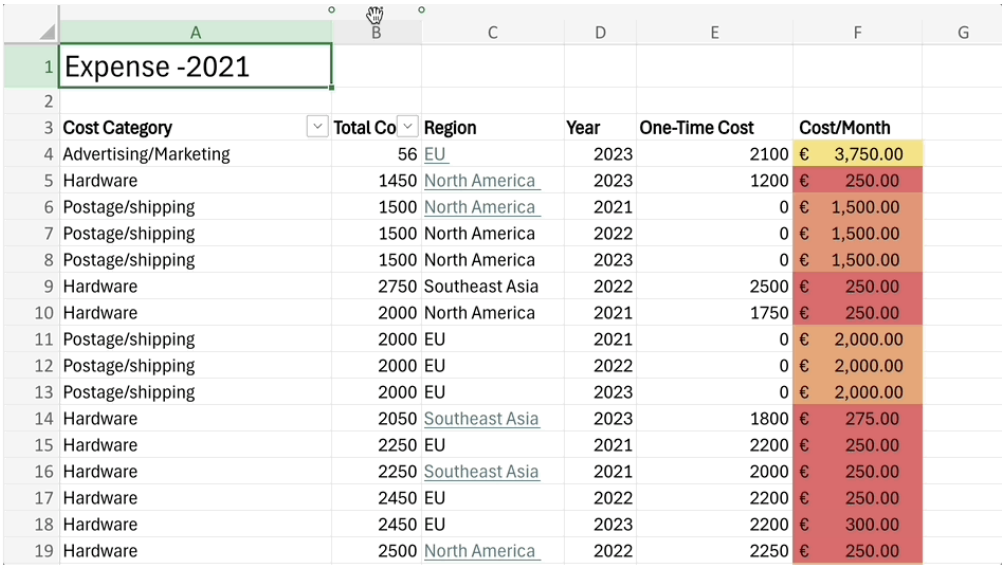
Expense -2021

Cost Category	Region	Year	One-Time Cost	Total Cost	Cost/Month
Advertising/Marketing	EU	2023	2100	56	€ 3,750.00
Hardware	North America	2023	1200	1450	€ 250.00
Postage/shipping	North America	2021	0	1500	€ 1,500.00
Postage/shipping	North America	2022	0	1500	€ 1,500.00
Postage/shipping	North America	2023	0	1500	€ 1,500.00
Hardware	Southeast Asia	2022	2500	2750	€ 250.00
Hardware	North America	2021	1750	2000	€ 250.00
Postage/shipping	EU	2021	0	2000	€ 2,000.00
Postage/shipping	EU	2022	0	2000	€ 2,000.00
Postage/shipping	EU	2023	0	2000	€ 2,000.00
Hardware	Southeast Asia	2023	1800	2050	€ 275.00
Hardware	EU	2021	2200	2250	€ 250.00
Hardware	Southeast Asia	2021	2000	2250	€ 250.00
Hardware	EU	2022	2200	2450	€ 250.00
Hardware	EU	2023	2200	2450	€ 300.00
Hardware	North America	2022	2250	2500	€ 250.00

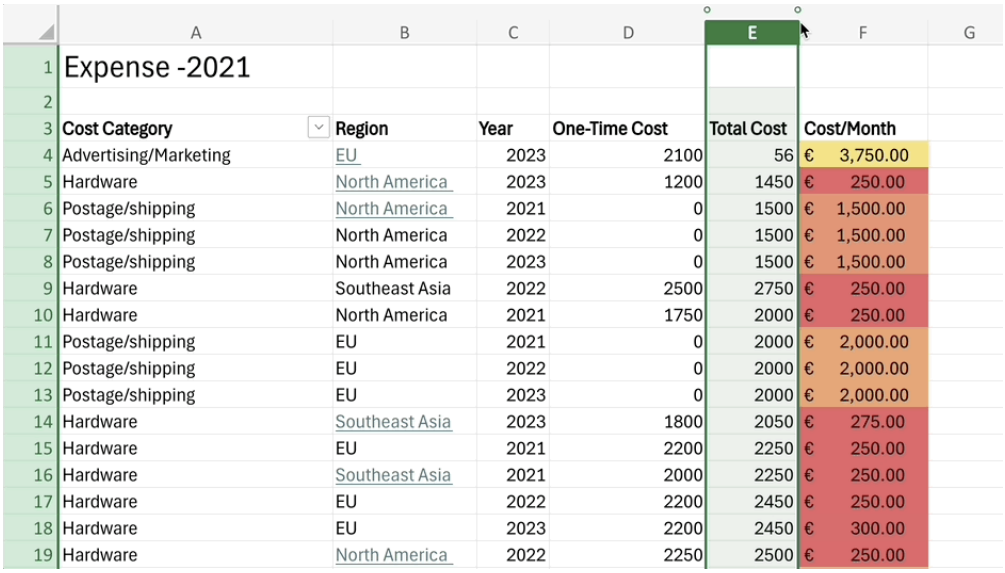


	A	B	C	D	E	F	G
1	Expense -2021						
2							
3	Cost Category	Region	Year	One-Time Cost	Total Cost	Cost/Month	
4	Advertising/Marketing	EU	2023	2100	56	€ 3,750.00	
5	Hardware	North America	2023	1200	1450	€ 250.00	
6	Postage/shipping	North America	2021	0	1500	€ 1,500.00	
7	Postage/shipping	North America	2022	0	1500	€ 1,500.00	
8	Postage/shipping	North America	2023	0	1500	€ 1,500.00	
9	Hardware	Southeast Asia	2022	2500	2750	€ 250.00	
10	Hardware	North America	2021	1750	2000	€ 250.00	
11	Postage/shipping	EU	2021	0	2000	€ 2,000.00	
12	Postage/shipping	EU	2022	0	2000	€ 2,000.00	
13	Postage/shipping	EU	2023	0	2000	€ 2,000.00	
14	Hardware	Southeast Asia	2023	1800	2050	€ 275.00	
15	Hardware	EU	2021	2200	2250	€ 250.00	
16	Hardware	Southeast Asia	2021	2000	2250	€ 250.00	
17	Hardware	EU	2022	2200	2450	€ 250.00	
18	Hardware	EU	2023	2200	2450	€ 300.00	
19	Hardware	North America	2022	2250	2500	€ 250.00	

Another simplified feature is the ability to rearrange elements in your worksheet with drag and drop, making data organisation rudimentary. To try the drag and drop feature, select any row or column, hold and drag when the cursor shows the hand icon, and then drop in any other row or column.

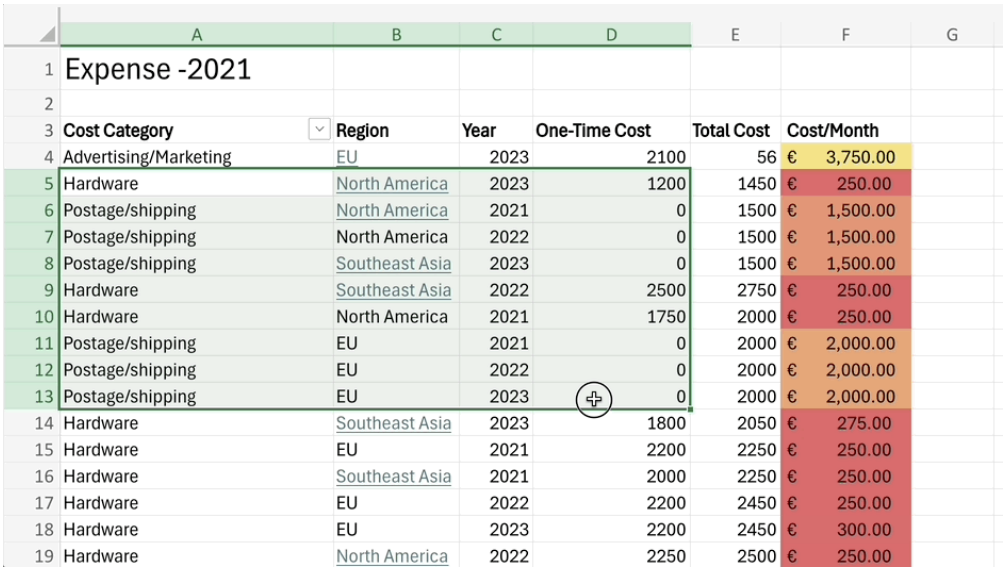


	A	B	C	D	E	F	G
1	Expense -2021						
2							
3	Cost Category	Total Co	Region	Year	One-Time Cost	Cost/Month	
4	Advertising/Marketing	56	EU	2023	2100	€ 3,750.00	
5	Hardware	1450	North America	2023	1200	€ 250.00	
6	Postage/shipping	1500	North America	2021	0	€ 1,500.00	
7	Postage/shipping	1500	North America	2022	0	€ 1,500.00	
8	Postage/shipping	1500	North America	2023	0	€ 1,500.00	
9	Hardware	2750	Southeast Asia	2022	2500	€ 250.00	
10	Hardware	2000	North America	2021	1750	€ 250.00	
11	Postage/shipping	2000	EU	2021	0	€ 2,000.00	
12	Postage/shipping	2000	EU	2022	0	€ 2,000.00	
13	Postage/shipping	2000	EU	2023	0	€ 2,000.00	
14	Hardware	2050	Southeast Asia	2023	1800	€ 275.00	
15	Hardware	2250	EU	2021	2200	€ 250.00	
16	Hardware	2250	Southeast Asia	2021	2000	€ 250.00	
17	Hardware	2450	EU	2022	2200	€ 250.00	
18	Hardware	2450	EU	2023	2200	€ 300.00	
19	Hardware	2500	North America	2022	2250	€ 250.00	



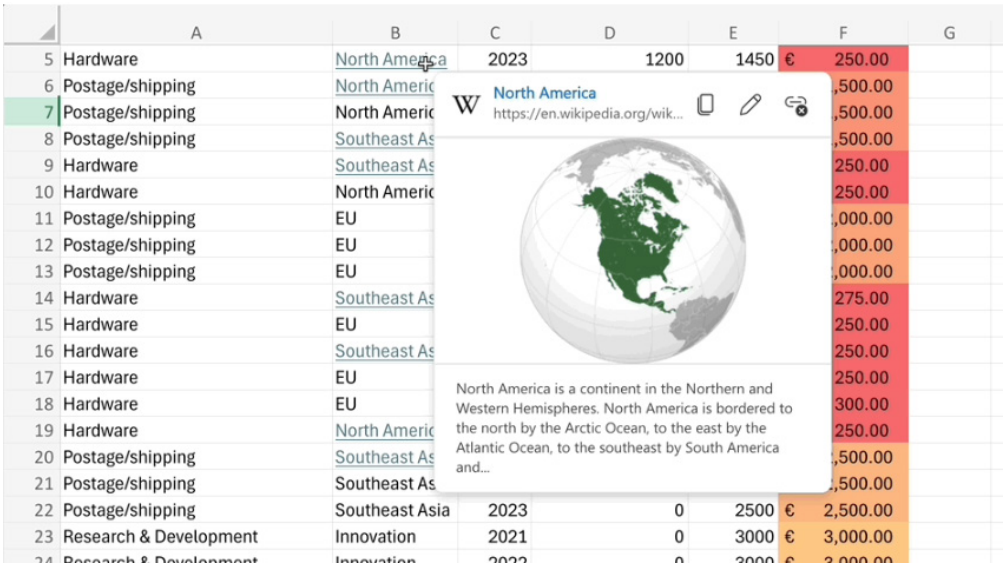
	A	B	C	D	E	F	G
1	Expense -2021						
2							
3	Cost Category	Region	Year	One-Time Cost	Total Cost	Cost/Month	
4	Advertising/Marketing	EU	2023	2100	56	€ 3,750.00	
5	Hardware	North America	2023	1200	1450	€ 250.00	
6	Postage/shipping	North America	2021	0	1500	€ 1,500.00	
7	Postage/shipping	North America	2022	0	1500	€ 1,500.00	
8	Postage/shipping	North America	2023	0	1500	€ 1,500.00	
9	Hardware	Southeast Asia	2022	2500	2750	€ 250.00	
10	Hardware	North America	2021	1750	2000	€ 250.00	
11	Postage/shipping	EU	2021	0	2000	€ 2,000.00	
12	Postage/shipping	EU	2022	0	2000	€ 2,000.00	
13	Postage/shipping	EU	2023	0	2000	€ 2,000.00	
14	Hardware	Southeast Asia	2023	1800	2050	€ 275.00	
15	Hardware	EU	2021	2200	2250	€ 250.00	
16	Hardware	Southeast Asia	2021	2000	2250	€ 250.00	
17	Hardware	EU	2022	2200	2450	€ 250.00	
18	Hardware	EU	2023	2200	2450	€ 300.00	
19	Hardware	North America	2022	2250	2500	€ 250.00	

You may also highlight important cells to emphasise critical information and improve readability. To use this feature, just select a row, column, range of cells or individual cell.



	A	B	C	D	E	F	G
1	Expense -2021						
2							
3	Cost Category	Region	Year	One-Time Cost	Total Cost	Cost/Month	
4	Advertising/Marketing	EU	2023	2100	56	€ 3,750.00	
5	Hardware	North America	2023	1200	1450	€ 250.00	
6	Postage/shipping	North America	2021	0	1500	€ 1,500.00	
7	Postage/shipping	North America	2022	0	1500	€ 1,500.00	
8	Postage/shipping	Southeast Asia	2023	0	1500	€ 1,500.00	
9	Hardware	Southeast Asia	2022	2500	2750	€ 250.00	
10	Hardware	North America	2021	1750	2000	€ 250.00	
11	Postage/shipping	EU	2021	0	2000	€ 2,000.00	
12	Postage/shipping	EU	2022	0	2000	€ 2,000.00	
13	Postage/shipping	EU	2023	0	2000	€ 2,000.00	
14	Hardware	Southeast Asia	2023	1800	2050	€ 275.00	
15	Hardware	EU	2021	2200	2250	€ 250.00	
16	Hardware	Southeast Asia	2021	2000	2250	€ 250.00	
17	Hardware	EU	2022	2200	2450	€ 250.00	
18	Hardware	EU	2023	2200	2450	€ 300.00	
19	Hardware	North America	2022	2250	2500	€ 250.00	

Another change is that when you now hover over web or internal links, you'll see improved hyperlink previews with options to copy, edit or remove the link. If a thumbnail of the linked page is available, you'll see that too, giving you a better idea of where the link will take you.



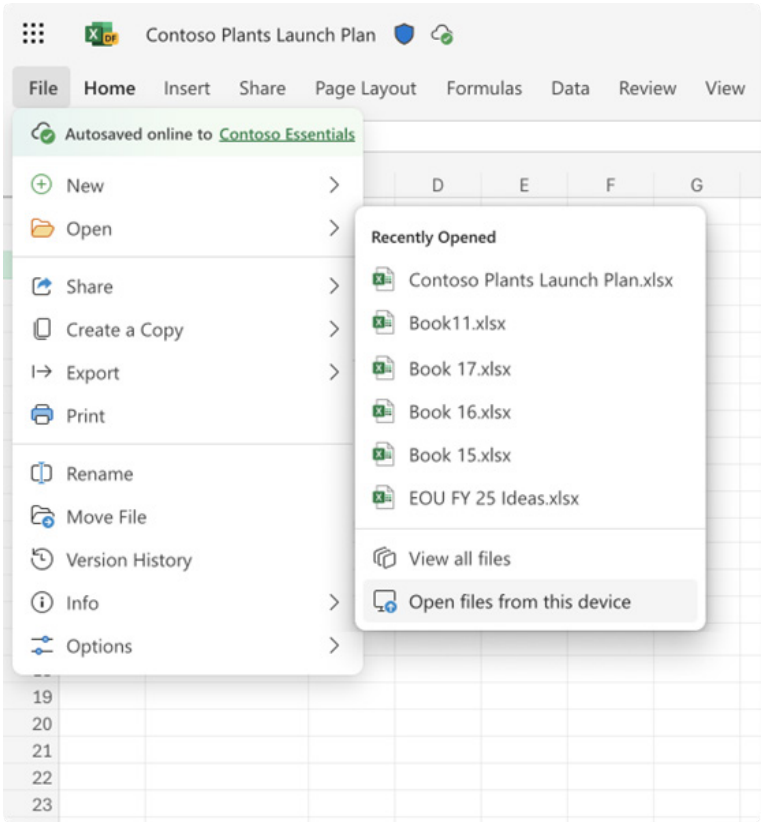
Customising your grid zoom is also easier now with the Status bar. You may now quickly edit zoom values to suit your preferences by either selecting a percentage or typing one in.

Renaming sheets has also been updated. Simply go to the sheet tab and double-click to rename it directly (just as you can on the desktop version): no more navigating through dialogs is required.

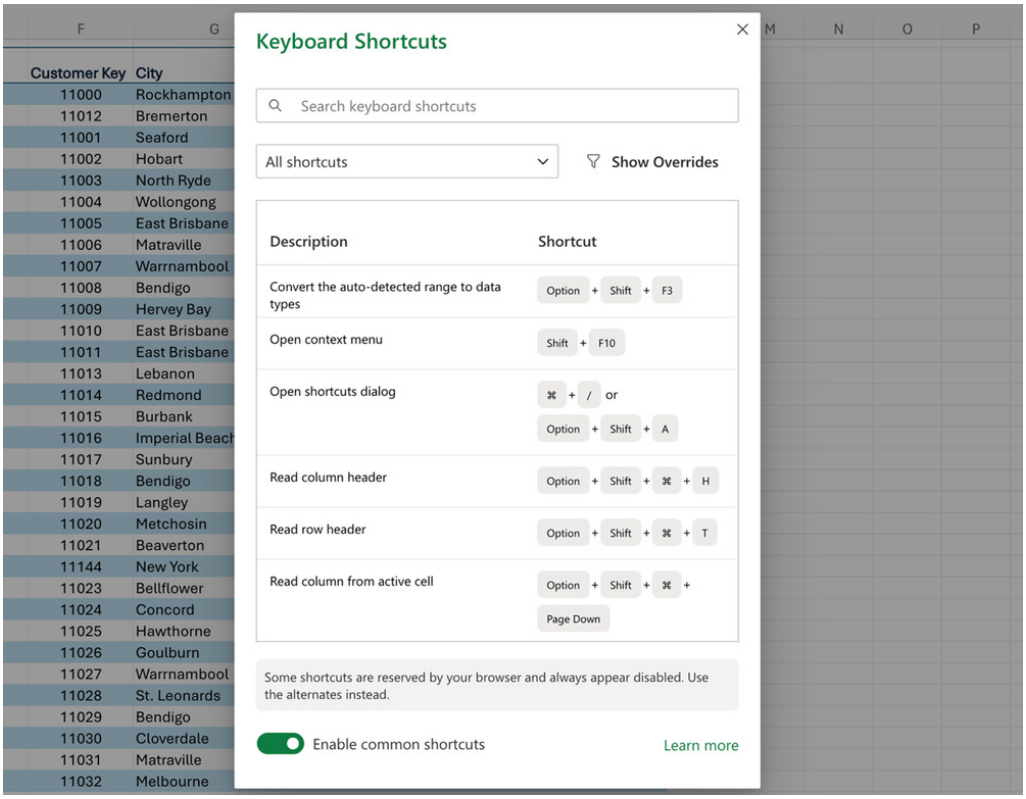
22	Pending	Male	Luke	Lal	11019	Langley
23	Pending	Male	Jordan	King	11020	Metchosin
24	Pending	Female	Destiny	Wilson	11021	Beaverton
25	Pending	Male	Edward	Hernandez	11144	New York
26	Verified	Male	Seth	Edwards	11023	Bellflower
27	Verified	Male	Russell	Xie	11024	Concord
28	Verified	Male	Alejandro	Beck	11025	Hawthorne
29	Pending	Male	Harold	Sai	11026	Goulburn
30	Verified	Male	Jessie	Zhao	11027	Warrnambool
31	Verified	Female	Jill	Jimenez	11028	St. Leonards
32	Verified	Male	Jimmy	Moreno	11029	Bendigo
33	Verified	Female	Bethany	Yuan	11030	Cloverdale
34	Verified	Female	Theresa	Ramos	11031	Matraville
35	Verified	Female	Denise	Stone	11032	Melbourne
36	Verified	Male	Jaime	Nath	11033	Milsons Point
37	Verified	Female	Ebony	Gonzalez	11034	North Sydney

Financial trends | 2024 Summary | 2023 Summary | Details | +

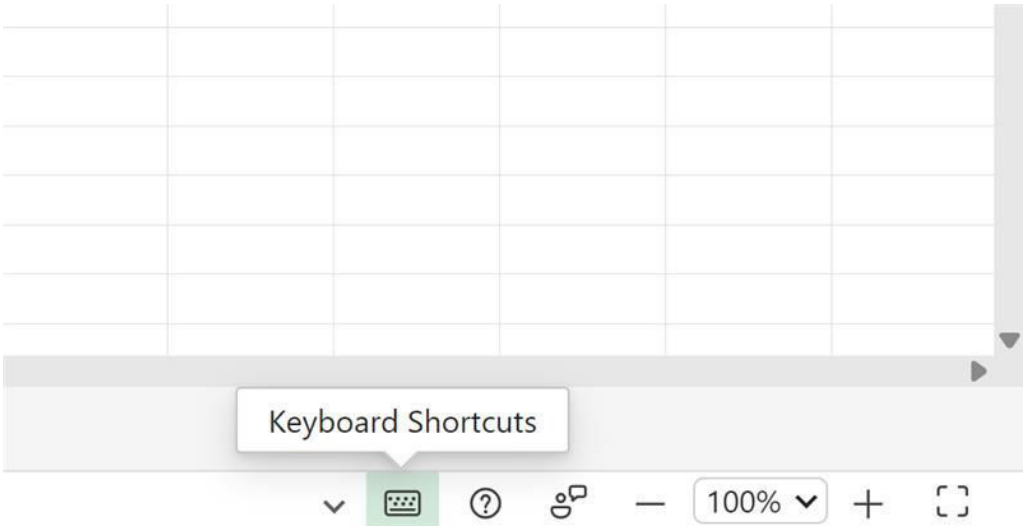
There is also a new 'Open files from this device' option in the File menu. This lets you upload a local file to your OneDrive and access it directly from your browser.



Next up, Microsoft has added support for accelerator keyboard shortcuts (**ALT + E**, **ALT + V**, **ALT + O**) plus over 120 legacy shortcuts. Additionally, the Keyboard Shortcuts dialog has been modified, with richer information and styling.



Finally (for now!), you may now quickly access shortcuts to Feedback, Help and Keyboard shortcuts straight from the Status bar.

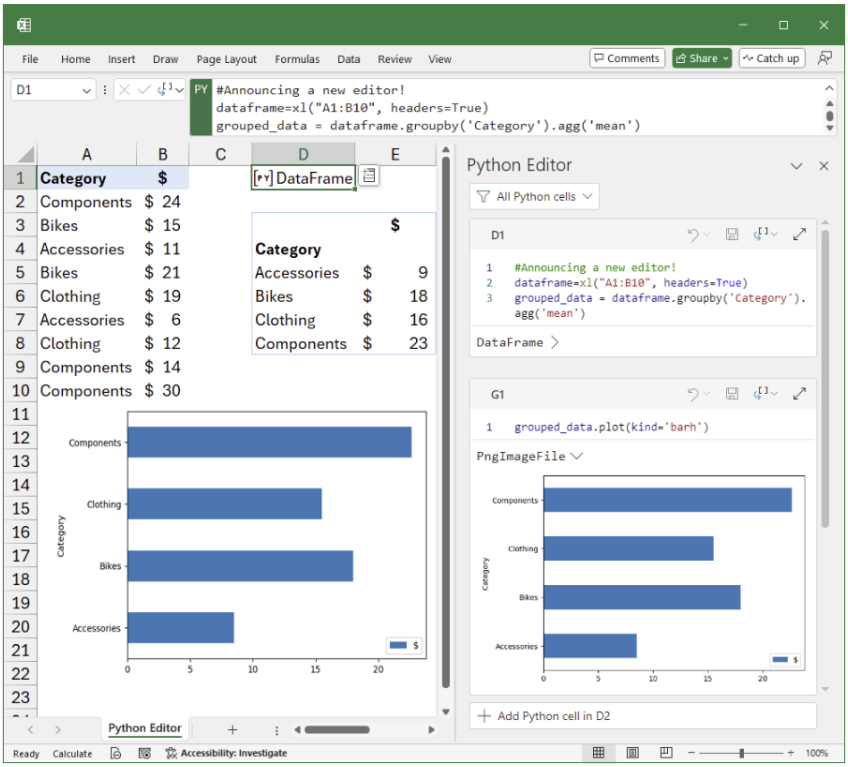


These features are currently rolling out to all Excel for the web users. I wouldn't say they are gamechangers, but they do modernise the look and feel of Excel online.

Python Editor in Insiders Beta

Almost a year ago, Microsoft shipped a new experiment to the Excel Labs add-in. This experiment allowed you to write and edit Python formulae in Excel using a dedicated code editor with similar capabilities as those available in Python notebook environments.

Now, Microsoft is making the editor available as an in-the-box option, and they have improved the appearance and usability of the Python Editor. You'll see a slightly different user interface that stays faithful to the original design, but there will be other changes shortly too.



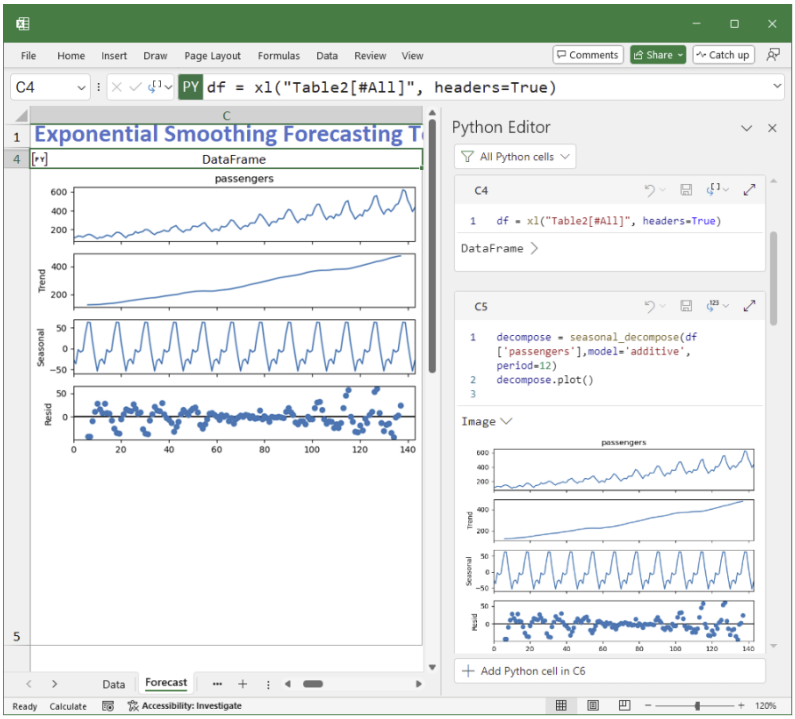
The Python Editor enhances the Python in Excel experience by providing a bigger editing space for writing larger code blocks as well as productivity features.

From the Python Editor, you can see a list of all the Python cells in your spreadsheet. Each cell is treated like that of a code cell in a Jupyter notebook. Within the Python Editor pane, you can edit the cells and run them. The output of the Python cell can be displayed natively in your Excel spreadsheet, whether it is plain text, numeric or even a visualisation or DataFrame.

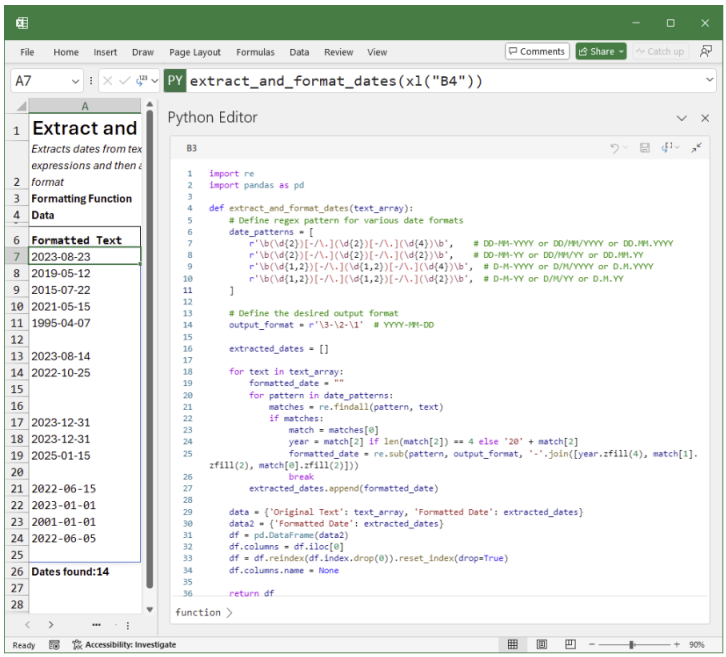
This new Python Editor is powered by the same technology that supports notebooks in other Microsoft products so you can use the full range of editor features like IntelliSense, colourisation and function help as you code in Excel.

The Python Editor offers several benefits when writing and editing Python formulae in Excel:

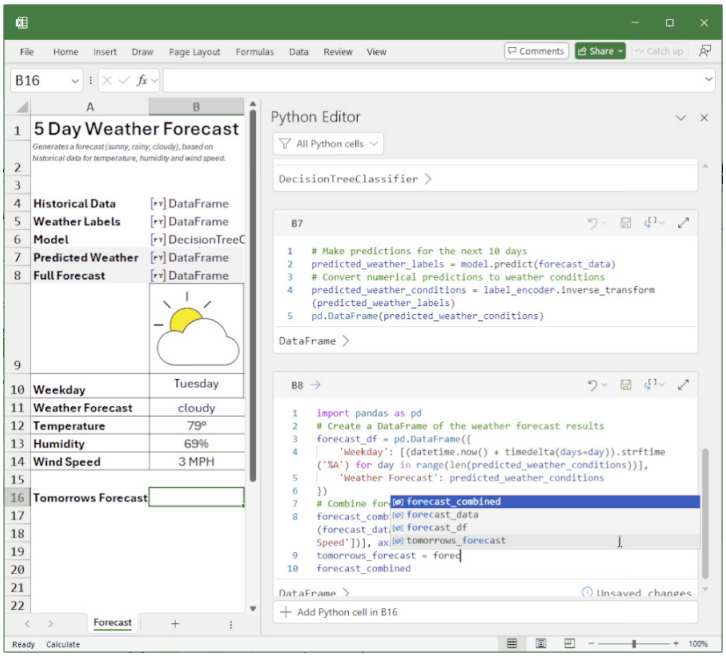
- **See the flow of code execution with results:** the Python Editor shows Python cells in execution order in addition to the output of each cell's code, which can help you more easily debug and understand your code. This is useful because, Python in Excel cells execute in row major order



- **Easily edit longer chunks of code:** the Python Editor provides a new way to create and edit your Python code, offering more immersion when working with longer scripts

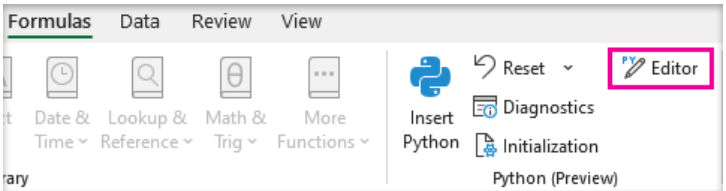


- **Gain flexibility with coding workflows:** the Python Editor allows you to write code, click elsewhere in the application, but not commit the code to Excel until you are happy with it. This gives you more flexibility and control over when and how you apply your code to your workbook. Additionally, when in manual recalculation mode, editing and committing from the Editor will calculate just that cell (like a notebook) providing a faster feedback loop



For the near term, there will still be two versions of the Editor, making it available wherever Python in Excel is available:

1. **Built into Excel:** if you are using Insiders Beta, Microsoft has now begun the rollout and you can try out the feature by going to the Ribbon and opening the 'Editor button in the 'Formulas' tab



2. **Excel Labs:** the Python Editor will remain in Excel Labs as Microsoft ships to additional audiences.

You need to have access to Python in Excel to make use of the Python Editor.

Beat the Boredom Challenge

With many of us currently "working from home" / quarantined, there are only so Zoom / Teams calls and virtual parties you can make before you reach your (data) limit. Perhaps they should measure data allowance in blood pressure millimetres of mercury (mmHg). To try and keep our readers engaged, we will continue to reproduce some of our popular ***Final Friday Fix*** *challenges from yesteryear in this and upcoming newsletters. One suggested solution may be found later in this newsletter. Here's this month's…*

There may be a time when you are using Excel that you wish to extract a name of a person from a churn of endless text strings that have all sorts of characters that you have never seen or used before. Indeed, the recently announced regular expression ("regex") functions may resolve this simply very soon. In the meantime, to replicate that experience, we give you a similar challenge here.

This month's challenge is to write a **formula** to extract the name of a person from any given text string. The result should look like similar to the following:

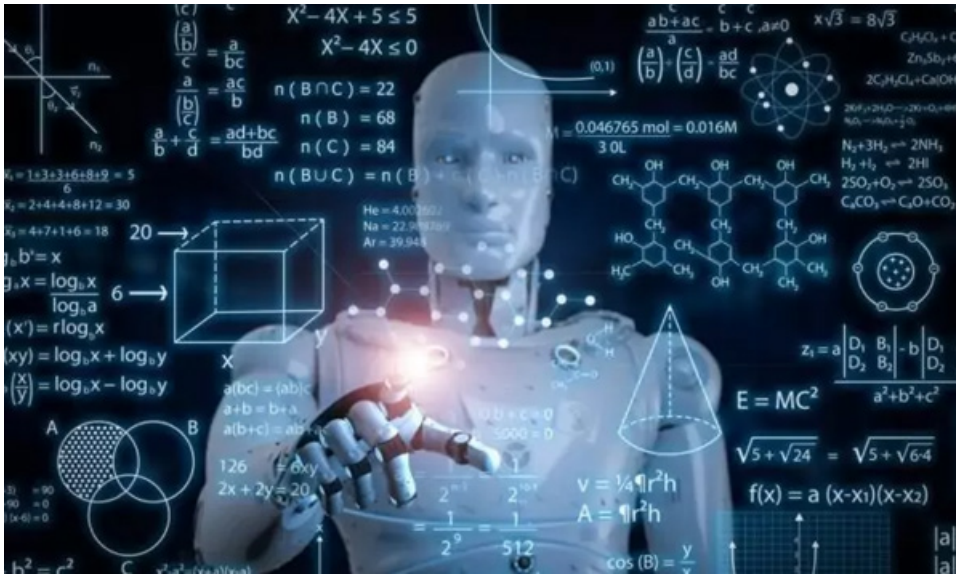
	Text	Solution
1.	Sara's Address	Sara
2.	Brian T's Job title	Brian T
3.	Karina's Date of Birth	Karina

As always, there are some requirements:

- the formula needs to be within just one column (no "helper" cells)
- this is a formula challenge; no Power Query / Get & Transform or VBA!
- the formula should be dynamic enough when a similar text string is added.

Sounds easy? Try it. One solution *just might* be found later in this newsletter – but no reading ahead!

Over to AI



Here at SumProduct we write many articles for many magazines and our authoring contracts have been revised recently to ensure Artificial Intelligence hasn't taken over our duties. We can assure everyone they have not. They'd never be able to make as many mistakes as we ltgsdgafjksw/.

But it did get us thinking: *what if we did let AI loose and write a regular article for the newsletter?*

So that's what we've done. See what you think. Is it better than the usual rubbish..?

This Month's Prompt: Write the Perfect Article for a Financial Modelling Newsletter

WHAT IS FINANCIAL MODELLING AND WHY IS IT IMPORTANT?

Financial modelling is the process of creating a representation of a real-world financial situation or scenario, such as the performance of a business, a project or an investment. Financial models are used for various purposes, such as budgeting, forecasting, valuation, decision making and risk analysis. Financial modelling is a vital skill for anyone who works in finance, accounting, investment banking, corporate development or consulting.

WHAT ARE THE LATEST TRENDS IN FINANCIAL MODELLING?

The field of financial modelling is constantly evolving and adapting to the changing needs and challenges of the business world. Some of the current trends that are shaping the future of financial modelling are:

- **Automation and artificial intelligence:** With the advancement of technology, many tasks and processes in financial modelling can be automated or enhanced by artificial intelligence. For example, AI can help with data collection, validation, analysis and visualisation, as well as generate insights and recommendations based on the model outputs. Automation and AI can also reduce human errors, save time and improve efficiency and accuracy.
- **Scenario analysis and stress testing:** In the wake of the COVID-19 pandemic and the resulting economic uncertainty, scenario analysis and stress testing have become more important than ever for financial modelling. Scenario analysis is the process of evaluating the impact of different assumptions and variables on the model outcomes, while stress testing is the process of assessing the resilience of the model under extreme or adverse conditions. Scenario analysis and stress testing can help financial modellers to identify and mitigate risks, as well as to explore opportunities and alternatives.
- **Cloud computing and collaboration:** Cloud computing is the delivery of computing services, such as data storage, processing and software, over the internet. Cloud computing enables financial modellers to access and share data and models from anywhere, anytime, and on any device. Cloud computing also offers scalability, security and cost-effectiveness. Collaboration is the ability to work together with other people on the same or related models. Collaboration can enhance the quality and reliability of the models, as well as foster innovation and creativity.

WHAT ARE THE BEST TIPS AND TOOLS FOR FINANCIAL MODELLING?

Financial modelling is both an art and a science, and it requires a combination of knowledge, skills and tools. Here are some of the best tips and tools for financial modelling:

Tip: Follow the best practices and standards for financial modelling. Some of the best practices and standards are: use clear and consistent assumptions, formulas, and formatting; document and explain your model logic and structure; avoid circular references and hard-coding; use error checks and validation; perform sensitivity and quality analysis.

Tool: Excel is the most widely used and versatile tool for financial modelling. Excel offers a range of features and functions that can help you create and manipulate financial models, such as formulas, charts, pivot tables, data analysis and macros. Excel also integrates well with other tools and platforms, such as Power BI, Python and VBA.

Tip: Learn from the experts and the community. Financial modelling is a dynamic and competitive field, and it is important to keep learning and improving your skills and knowledge. You can learn from the experts and the community by taking online courses, reading books and blogs, watching videos and podcasts, joining forums and groups, and attending events and workshops.

Tool: Financial Modelling software is a specialised software that is designed to facilitate and enhance the process and output of financial modelling. Financial modelling software can offer features and functions that are not available or limited in Excel, such as advanced modelling capabilities, data integration, scenario analysis, reporting and visualisation.

What do you think? Do you agree? (I am still trying to find those agreed best practices and standards for financial modelling…) Send us your comments and ideas for questions to contact@sumproduct.com.

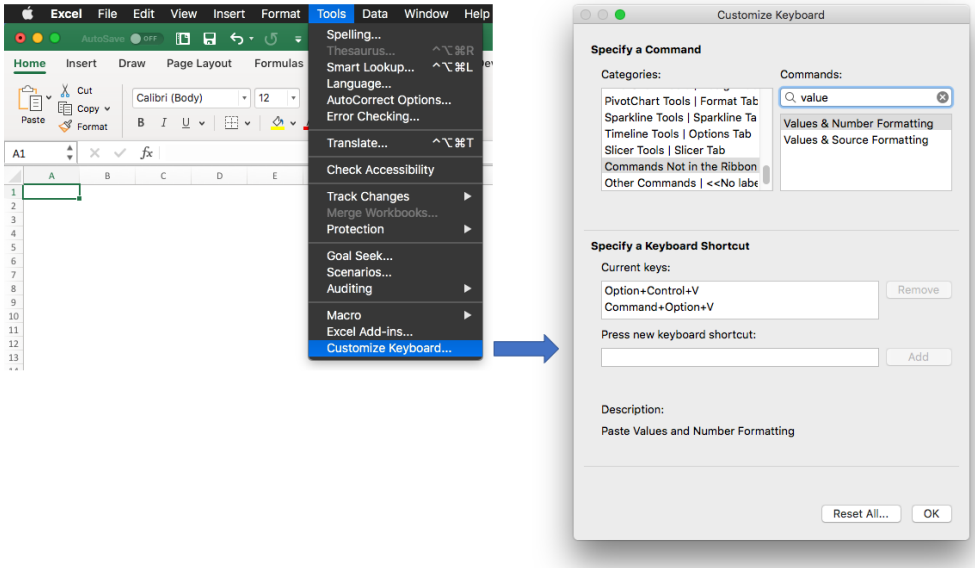
Excel for Mac

With Steve Kraynak joining the team, we thought we would exploit his knowledge and recant all about Microsoft Excel for Mac. Each month, we'll cover a different topic to help you understand how Excel for Mac is different than Excel for Windows. This month, we show how to customise your keyboard shortcuts using both Excel's built-in 'Customize Keyboard' feature or Mac's keyboard shortcut preferences. This is one of the things you can do with Excel for Mac that you can't do on Windows.

To create or change a keyboard shortcut in Excel for Mac, follow these steps:

- Go to the Tools menu and choose 'Customize Keyboard'
- Pick the category of command you're trying to find, and then search or browse for the command in the list
- Select a command, press a key combination, and see if it's used already. If it's already in use, you may want to pick a different combination
- Then just click the 'Add' button to assign the key combination to the selected command.

In the example below, we show the category called 'Commands Not in the Ribbon' with a search for commands with the word "value", which filters to the commands 'Values & Number Formatting' and Values & Source Formatting'.



You should note that some commands may not be available. Generally, only commands that appear in the 'Ribbon Customization' dialog can be found in the 'Customize Keyboard' dialog. For example, there is no command that allows you to set a cell format to use a particular font. Some keyboard shortcuts may be used by your Mac and you may not be able to use these key combinations, even though you can assign them in the dialog.

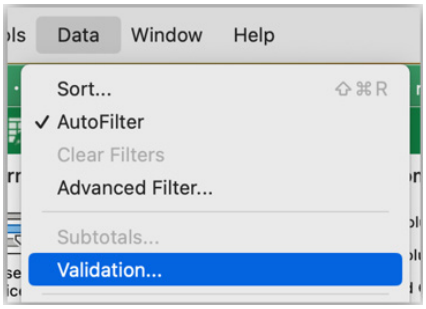
You can read more details here: Create a Custom Keyboard Shortcut.

macOS Keyboard Shortcuts

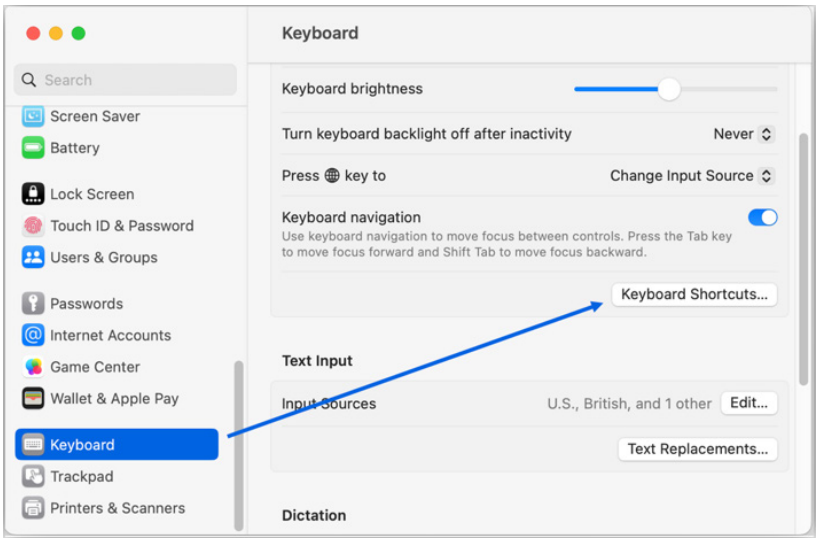
Another way to create a customised keyboard shortcut is to use the Mac Preferences. This allows you to set a keyboard shortcut for any app, but only for commands that appear in one of that app's menus.

Just follow the simple steps below:

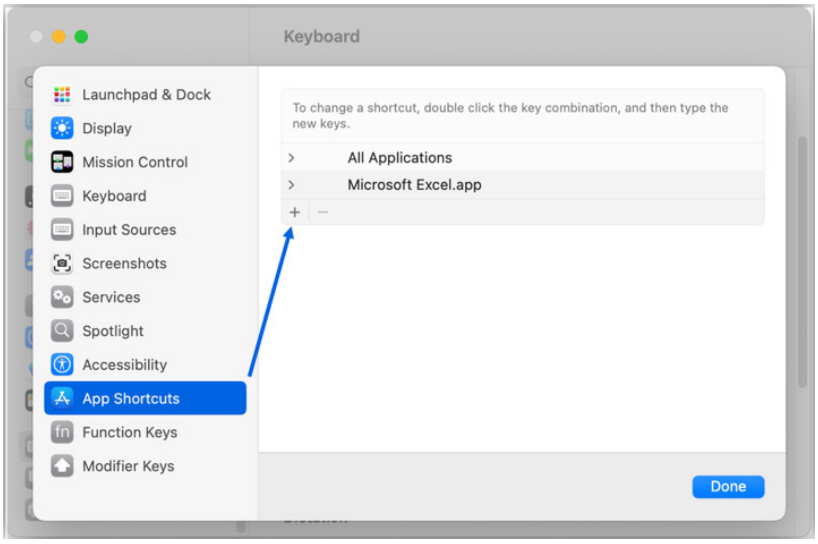
- Look for a menu command that you want to use a keyboard shortcut to activate. For example, you can go to the Data menu in Excel and you'll see 'Validation…', which will open the 'Data Validation' dialog:



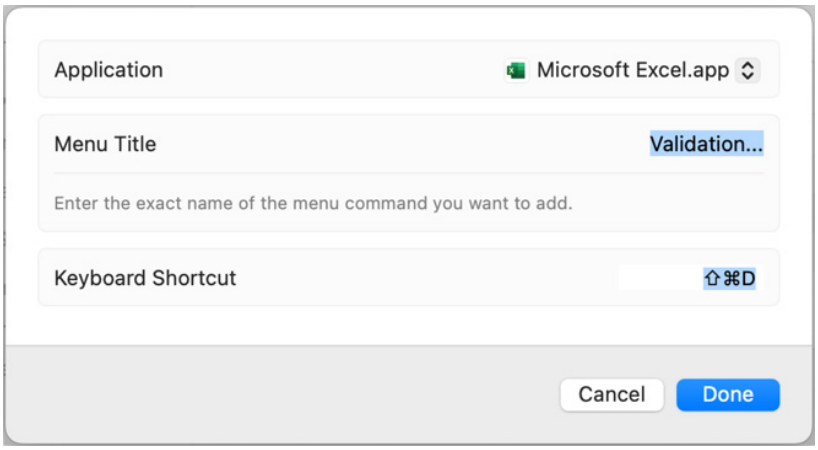
- Go to **macOS -> Preferences -> Keyboard -> Keyboard Shortcuts -> App Shortcuts**



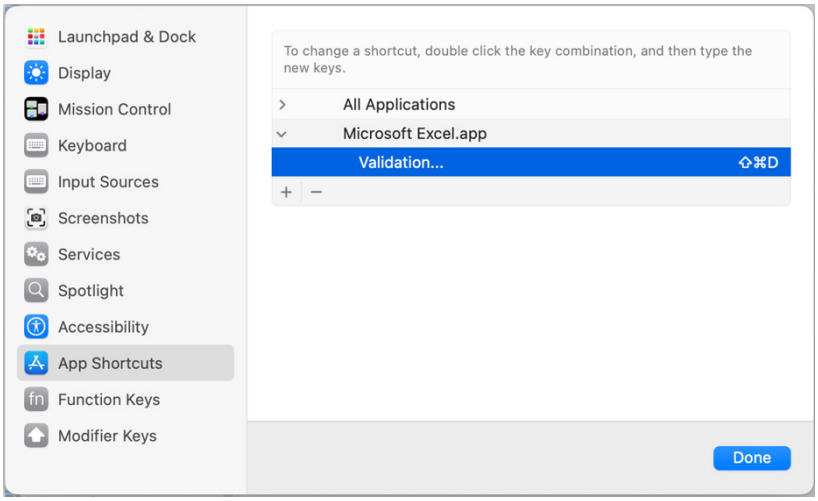
- Click the **+** button to add a new shortcut



- Select 'Microsoft Excel.app' from the list of applications



- In the Menu Title field, type the name of the menu item exactly as it appears in Excel. For example, type 'Validation…' to create a shortcut that will simulate opening the Data menu and pressing 'Validation…', which opens the 'Data Validation' dialog
- Press a key combination. It's a good idea to choose a key combination that's not already used to do something else
- Now go back to Excel and try out the new shortcut you just created!



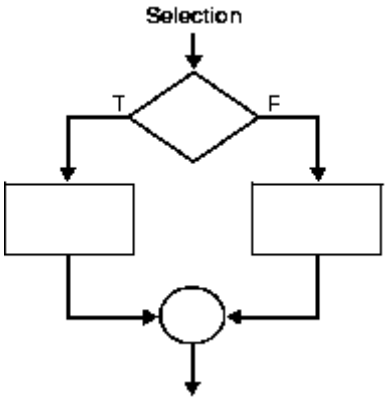
See Apple's help article for more information: Use macOS keyboard shortcuts - Apple Support

We'll continue next month…

Visual Basics

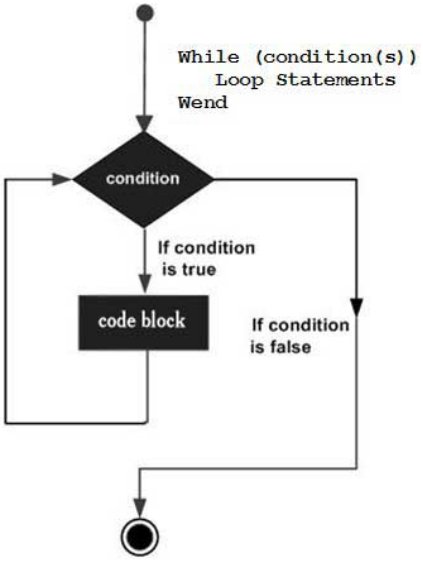
We thought we'd run an elementary series going through the rudiments of Visual Basic for Applications (VBA) as a springboard for newer users. This month, we take continue our discussion on control structures.

In a programming, a control structure determines the order in which statements are executed. The iteration control structure is used for repetitively executing a block of code multiple times.



The iteration structure executes a sequence of statements repeatedly if a condition holds true. There are three main types of loops in VBA:

1. While…Wend

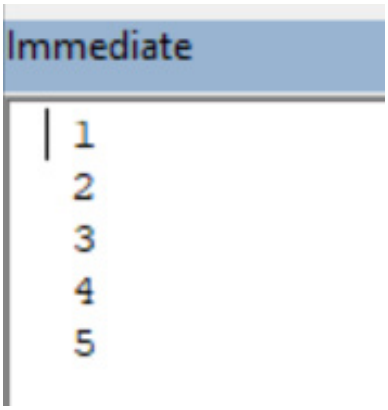


The **WHILE…WEND** loop executes a series of statements as long as a given condition is `True`. The syntax is very simple:

```
While condition
    [statements]
Wend
```

The condition must result in a Boolean value of **True** or **False**. **WHILE** tests the condition and if it is **True** then proceeds to execute the statements inside the loop.

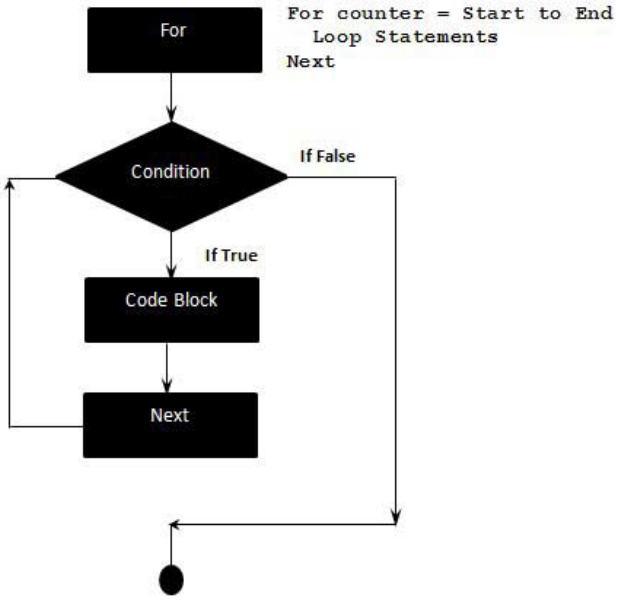
```
Sub WhileWend()
    Dim counter As Integer
    counter = 0
    While counter < 5
        counter = counter + 1
        Debug.Print counter
    Wend
End Sub
```



While loops are preferred when the number of iterations is unknown. For example, modelling how many days it takes to reach sales a target, or running through a worksheet column until it reaches an empty cell.

Notice how the condition is tested first – this means that the code will not run at all if the condition is not met. **WHILE…WEND** is a remnant from BASIC where VBA originated from. These are not as powerful as **DO…LOOP**.

2. FOR

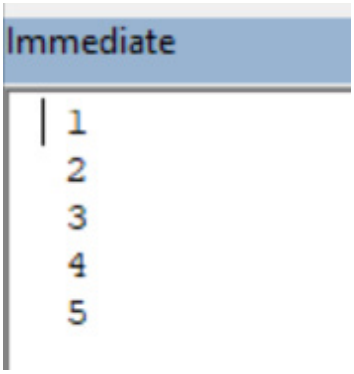


The **FOR…NEXT** loop uses a variable, which cycles through a series of values within a specified range and the statements inside the loop is then executed for each value.

```
For counter = start To end [ Step step ]
[ statements ]
Exit For
[ statements ]
Next [ counter ]
```

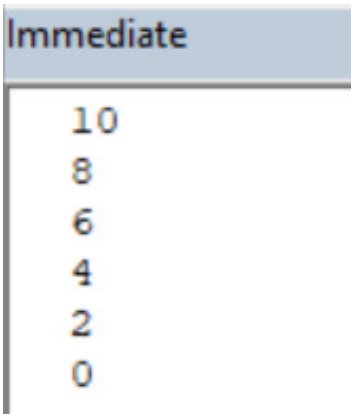
Here's a simple example:

```
Sub ForNext()
    Dim counter As Integer
    For counter = 1 To 5
        Debug.Print counter
    Next counter
End Sub
```



The **STEP** keyword allows the specification of how the counter changes. It defaults to an increment of 1, but it can be used for jumps and decrements.

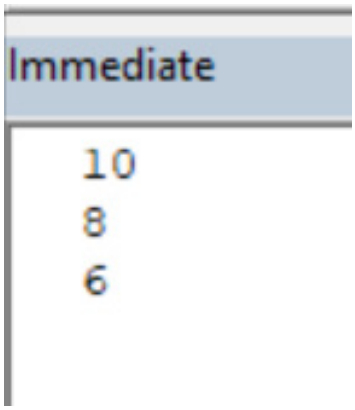
```
Sub ForNextStep()
    Dim counter As Integer
    For counter = 10 To 0 Step -2
        Debug.Print counter
    Next counter
End Sub
```



EXIT FOR

EXIT FOR statements may be placed anywhere in the loop as an alternate way to exit. This is often used after evaluating of some condition, for example **IF…THEN**, and then skips to the statements after the loop.

```
Sub ForNextExit()
    Dim counter As Integer
    For counter = 10 To 0 Step -2
        Debug.Print counter
        If counter = 6 Then
            Exit For
        End If
    Next counter
End Sub
```



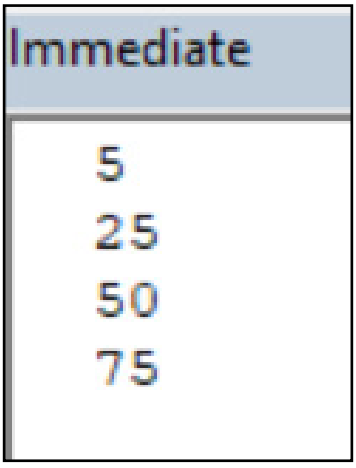
FOR EACH…NEXT

What if an action is needed to be performed to every object in a set?

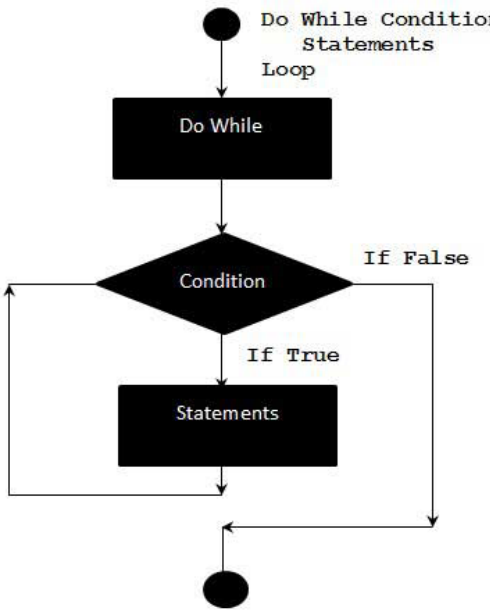
FOR EACH…NEXT loops are a great way to cycle through sets – like an array or a range. Sometimes the number of rows or columns is uncertain. It is relatively easy to count the number of objects and set the upper bound of the **FOR…NEXT** loop appropriately. However, using **FOR EACH…NEXT** more clearly illustrates that the instructions are happening to every object.

As an example:

```
Sub ForEach()

    Dim myNumbers() As Variant

    myNumbers = Array(1, 5, 10, 15)

    Dim aNumber As Variant

    For Each aNumber In myNumbers

        Debug.Print aNumber * 5

    Next

End Sub
```



3. DO



The final loops belong to the **DO** family. There are several members and examples are presented below.

DO…LOOP

DO…LOOP loops are considered the upgraded alternative to **WHILE WEND**. Let's have a look at how they work:

```
Do [{ While |Until } condition ]
        [ statements ]
        [ Exit Do ]
        [ statements]
Loop
```

How does the code change from **WHILE WEND** to **DO…LOOP**? Simply replace the **WHILE** with **DO WHILE** and **WEND** with **LOOP**. It's as easy as that!

```
Option Explicit

Sub JackAndJill()

Dim CurrentStepsTaken As Integer
CurrentStepsTaken = 0

Dim FallenDown As Boolean
FallenDown = False

While FallenDown = False

    CurrentStepsTaken = CurrentStepsTaken + 1

    If Rnd() <= 0.3 Then
        FallenDown = True
    End If

Wend

Debug.Print CurrentStepsTaken & " step(s) up the hill were taken."
Debug.Print "Jack fell down the hill!"
Debug.Print "Jill came tumbling after..."

End Sub
```

```
Option Explicit

Sub JackAndJillDo()

Dim CurrentStepsTaken As Integer
CurrentStepsTaken = 0

Dim FallenDown As Boolean
FallenDown = False

Do While FallenDown = False

    CurrentStepsTaken = CurrentStepsTaken + 1

    If Rnd() <= 0.3 Then
        FallenDown = True
    End If

Loop

Debug.Print CurrentStepsTaken & " step(s) up the hill were taken."
Debug.Print "Jack fell down the hill!"
Debug.Print "Jill came tumbling after..."

End Sub
```

DO…LOOP is superior to While Wend for several reasons:

- **WHILE WEND** has no ability to have an **EXIT**
- **WHILE WEND** loops check for the condition prior to running – but with **DO…LOOP** the condition can be checked at the end. This is useful if the code needs to be run at least once.

  This is done by simply moving the "**WHILE** [condition]" part of the **DO** statement next to **LOOP**. The syntax changes to:

```
Do
[ statements ]
[ Exit Do ]
[ statements]
Loop [{ While |Until } condition ]
```

```
Sub JackAndJillDo()

Dim CurrentStepsTaken As Integer
CurrentStepsTaken = 0

Dim FallenDown As Boolean
FallenDown = False

Do While FallenDown = False

    CurrentStepsTaken = CurrentStepsTaken + 1

    If Rnd() <= 0.3 Then
        FallenDown = True
    End If

Loop

Debug.Print CurrentStepsTaken & " step(s) up the hill were taken."
Debug.Print "Jack fell down the hill!"
Debug.Print "Jill came tumbling after..."

End Sub
```

```
Sub JackAndJillDoItAtLeastOnce()

Dim CurrentStepsTaken As Integer
CurrentStepsTaken = 0

Dim FallenDown As Boolean
FallenDown = False

Do

    CurrentStepsTaken = CurrentStepsTaken + 1

    If Rnd() <= 0.3 Then
        FallenDown = True
    End If

Loop While FallenDown = False

Debug.Print CurrentStepsTaken & " step(s) up the hill were taken."
Debug.Print "Jack fell down the hill!"
Debug.Print "Jill came tumbling after..."

End Sub
```

- The ability to replace **WHILE** with **UNTIL**: what effect does this achieve? This essentially reverses the value of the condition to be tested.

  WHILE executes the block of code when the condition is `True` and keeps executing that till the condition becomes `False`. Once the condition becomes `False`, the loop is terminated. However, if the condition tested is initially `False`, the condition must be tested as:
  `DO WHILE condition = FALSE`

  UNTIL does the opposite. It executes the block of code when the condition is `False` and keep executing that till the condition becomes `True`. Once the condition becomes `True`, the **UNTIL** loop is terminated.

  It should be noted that the `[condition]` is a Boolean value, the loop can then be adjusted with the starting statement:

  `DO UNTIL condition`

```
Sub JackAndJillDo()

Dim CurrentStepsTaken As Integer
CurrentStepsTaken = 0

Dim FallenDown As Boolean
FallenDown = False

Do While FallenDown = False

    CurrentStepsTaken = CurrentStepsTaken + 1

    If Rnd() <= 0.3 Then
        FallenDown = True
    End If

Loop

Debug.Print CurrentStepsTaken & " step(s) up the hill were taken."
Debug.Print "Jack fell down the hill!"
Debug.Print "Jill came tumbling after..."

End Sub
```

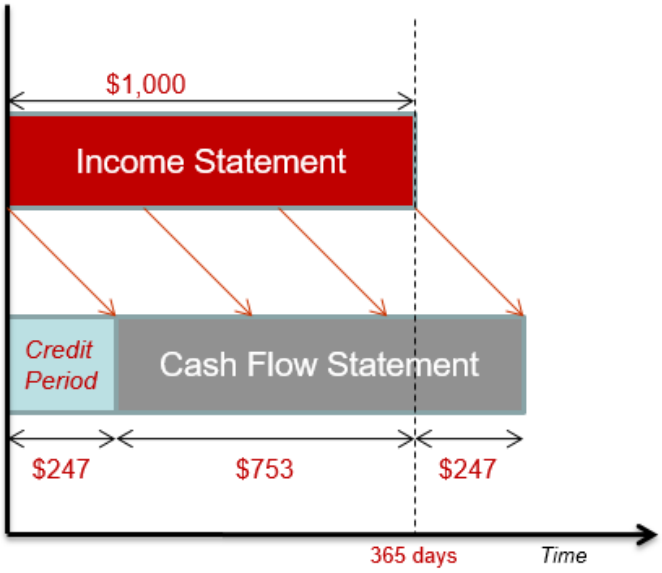
```
Sub JackAndJillDoUntil()

Dim CurrentStepsTaken As Integer
CurrentStepsTaken = 0

Dim FallenDown As Boolean
FallenDown = False

Do Until FallenDown

    CurrentStepsTaken = CurrentStepsTaken + 1

    If Rnd() <= 0.3 Then
        FallenDown = True
    End If

Loop

Debug.Print CurrentStepsTaken & " step(s) up the hill were taken."
Debug.Print "Jack fell down the hill!"
Debug.Print "Jill came tumbling after..."

End Sub
```

More next time.

Charts and Dashboards

It's time to chart our progress with an introductory series into the world of creating charts and dashboards in Excel. This month, we create a Working Capital Adjustment chart.

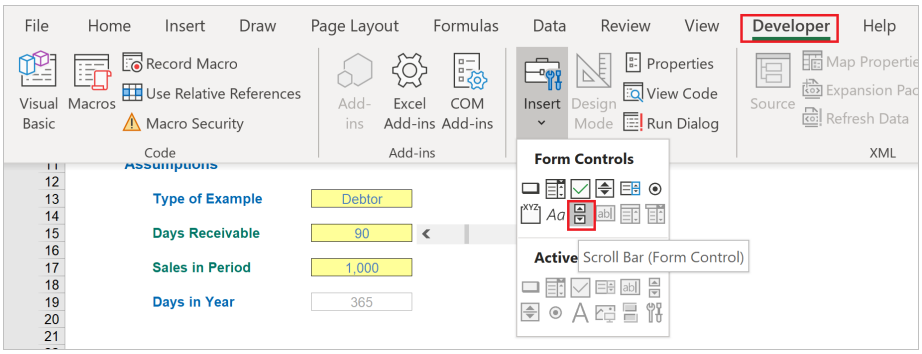
When modelling working capital adjustments (https://www.sumproduct.com/thought/working-capital-adjustments), a chart is useful to facilitate the presentation of cash flow figures against existing profit and loss projections.



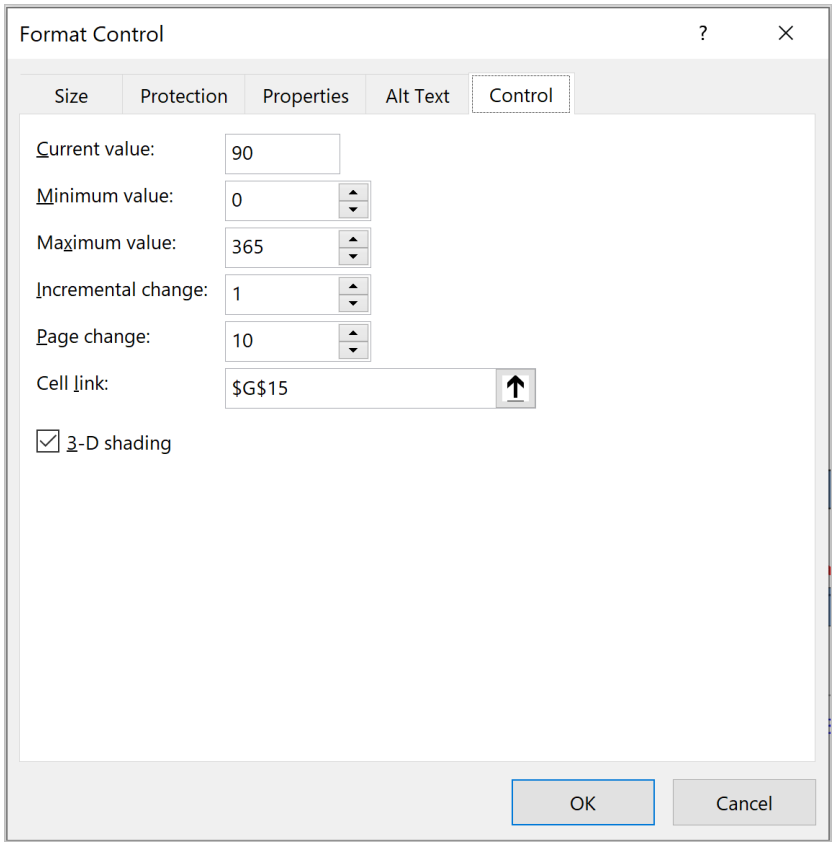
This time, we will see how we can create a Working Capital Adjustment chart, by using the example data *(below)*.

Simple Example			
This assumes that the number of days in the period exceeds the			
Assumptions			
Type of Example	Debtor		
Days Receivable	90		
Sales in Period	1,000		
Days in Year	365		
Control Account			
Opening Debtors	-	*Prior BS*	
Sales in Period	1,000	*IS*	
Cash Receipts	(753)	*CFS*	
Closing Debtors	247	*Current BS*	

To create the scroll bar, go to the Developer tab on the Ribbon (which you may need to install using **Tools -> Options -> Customize Ribbon** and then check the Developer tab in the 'Main Tabs' section). From Insert, choose 'Scroll Bar (Form Control)' and draw a scroll bar box next to the '**Days Receivable**' cell (holding the **ALT** button down makes the graphic "snap to grid"):



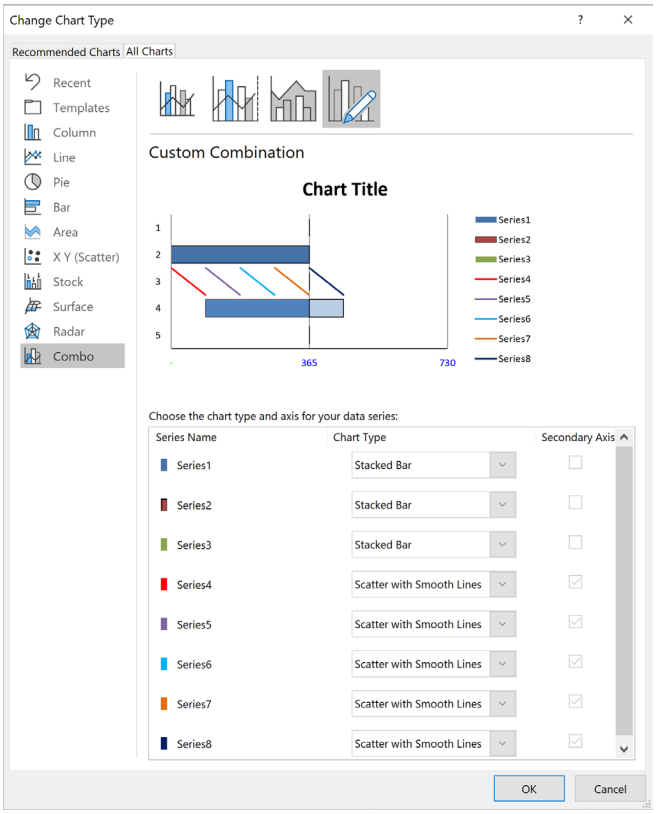
Then, right-click on the scroll bar box, choose 'Format Control' and link the '**Days Receivable**' cell, cell **G15**, whose value will adjust as we adjust the scroll bar.



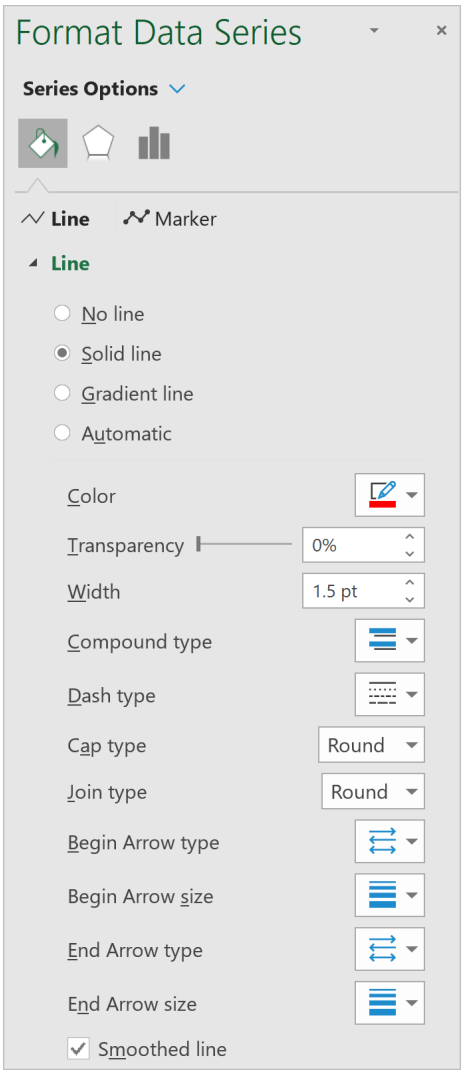
We need to prepare the data that can be used in the chart similar to below. The formulae for the calculated cells in columns **G** are noted down in column **I**.

	A	B	C	D	E	F	G	H	I	J
30										
31						**Chart Data**				
32										
33							365	0	=Days_in_Year	-
34						Sales in Period	365		=Days_in_Year	
35							365	0	=Days_in_Year	-
36						Cash Receipts	90	275	=IF(G26,G28/G26*Da	90
37							365	0	=Days_in_Year	-
38										
39							-	3		
40							90	2	=G36	
41										
42							91	3	=Days_in_Year/4	
43							181	2	=G40+G42	
44										
45							183	3	=Days_in_Year/2	
46							273	2	=G45+G40	
47										
48							274	3	=G42+G45	
49							364	2	=G48+G40	
50										
51							365	3	=Days_in_Year	
52							455	2	=G51+G40	
53										

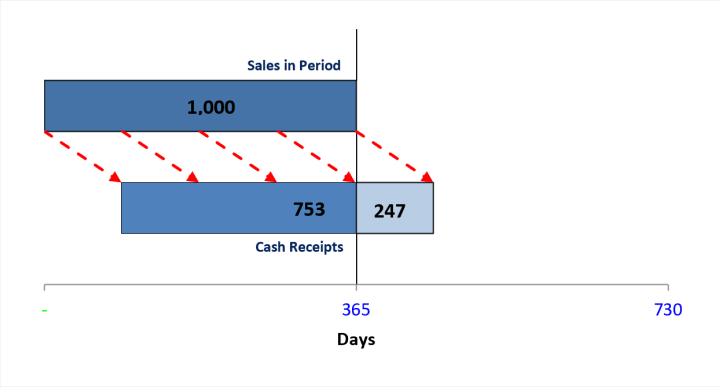
Next, highlight all the group of chart data cells in column **G**, **H** and **J** to create a Bar chart. Then from the 'Chart Design' tab on the Ribbon, choose 'Change Chart Type'. Here, keep the Series 1, 2 and 3 as 'Stacked Bar' and change Series 4 to 8 to 'Scatter with Smooth Line'.



Right-click on Series 4, choose ‘Format Data Series’, here change the Color, Dash type, Begin / End Arrow type, and repeat the same format settings for Series 5 to 7.



Finally, remove the chart title and add labels to enhance the chart information and the chart is done!



More next time.

Power Pivot Principles

*We continue our series on the Excel COM add-in, Power Pivot. This month, we show you how the **DATESINPERIOD** function works.*

The **DATESINPERIOD** Function is a time intelligence function, just like the **DATESYTD** and **DATEADD** functions. We have covered the **DATESYTD** and the **DATEADD** functions previously.

The **DATESINPERIOD** function uses the following syntax to operate:

DATESINPERIOD(dates, start_date, number_of_intervals, interval)

It should be noted that:

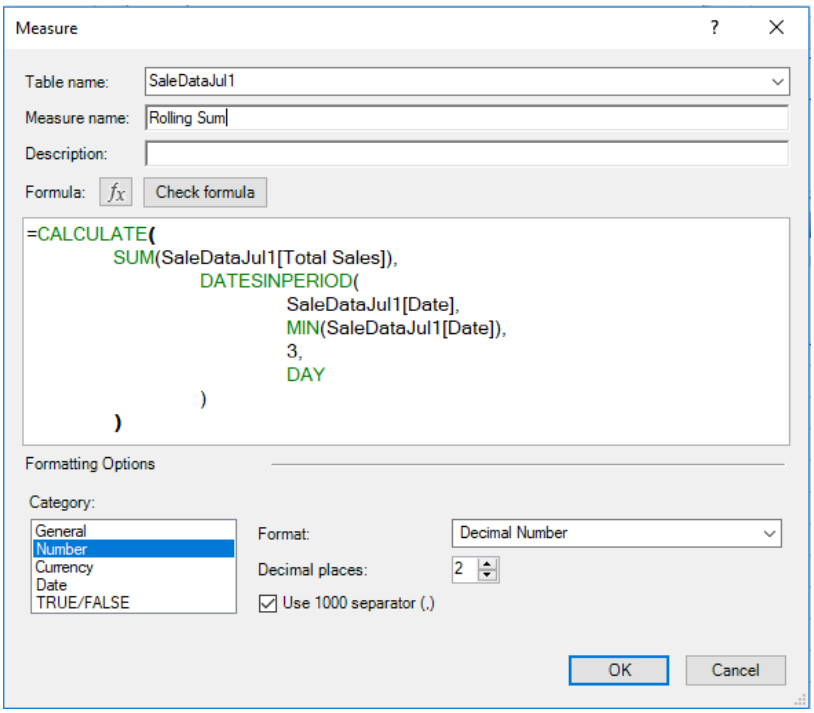
- This function returns with a table. This table will contain a column of dates that begins with the **start_date** and continues on with the specified **number_of_intervals**.
- The **dates** parameter has to be a column with dates.
- The **interval** parameter has to be one of four predefined inputs by PowerPivot: year, quarter, month, day.
- This function is commonly used together with the **CALCULATE** function.

Let's take a look at a simple example. Imagine we had the following sales data:

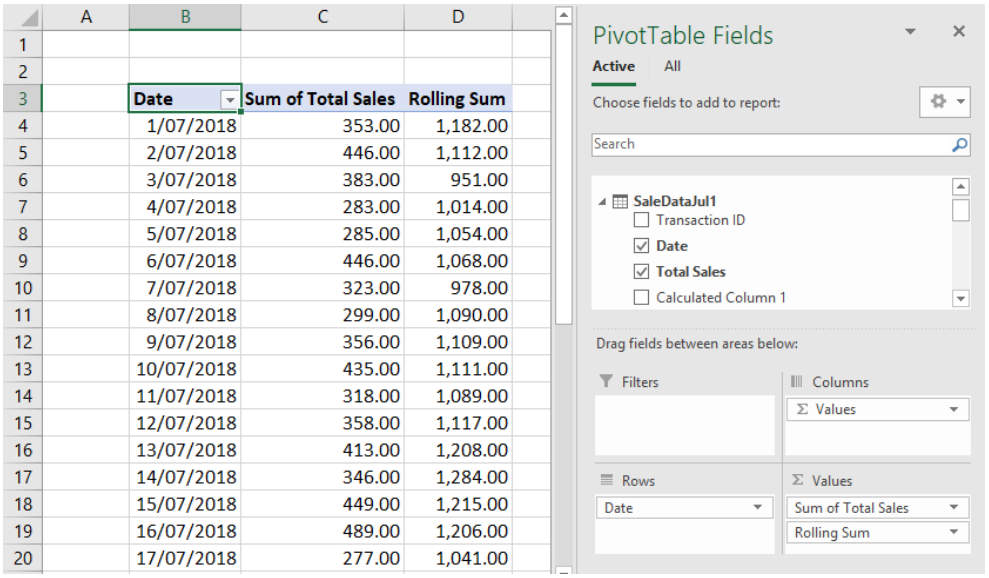
Date	Total Sales
1/07/2018	490.00
2/07/2018	475.30
3/07/2018	451.54
4/07/2018	465.08
5/07/2018	483.68
6/07/2018	498.19
7/07/2018	518.12
8/07/2018	492.22
9/07/2018	477.45
10/07/2018	472.68
11/07/2018	491.58
12/07/2018	506.33
13/07/2018	521.52
14/07/2018	537.17
15/07/2018	521.05
16/07/2018	531.47
17/07/2018	504.90
18/07/2018	515.00
19/07/2018	494.40
20/07/2018	504.28
21/07/2018	509.33
22/07/2018	529.70
23/07/2018	550.89
24/07/2018	523.34
25/07/2018	518.11
26/07/2018	523.29
27/07/2018	528.52
28/07/2018	517.95

In this example we want to create a rolling sum for every three days. We can use the following measure:

```
=CALCULATE(
        SUM(SaleDataJul1[Total Sales]),
                DATESINPERIOD(
                        SaleDataJul1[Date],
                        MIN(SaleDataJul1[Date]),
                        3,
                        DAY
                )
        )
```



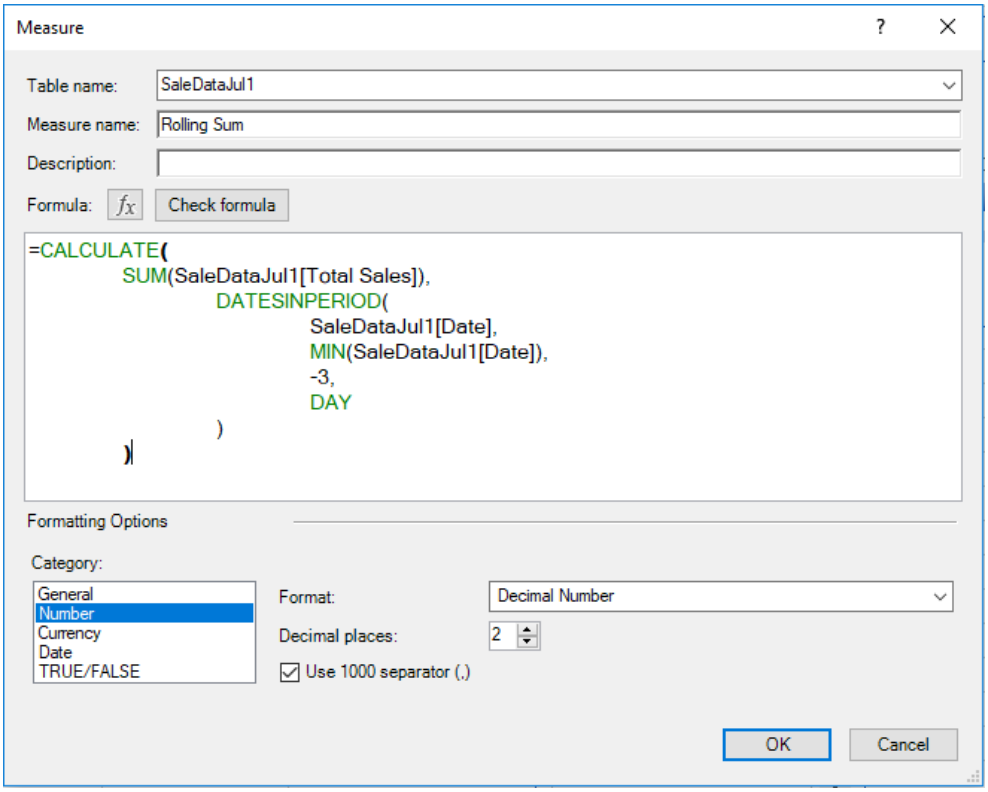
This would result in the following PivotTable:



	A	B	C	D
1				
2				
3		Date	Sum of Total Sales	Rolling Sum
4		1/07/2018	353.00	1,182.00
5		2/07/2018	446.00	1,112.00
6		3/07/2018	383.00	951.00
7		4/07/2018	283.00	1,014.00
8		5/07/2018	285.00	1,054.00
9		6/07/2018	446.00	1,068.00
10		7/07/2018	323.00	978.00
11		8/07/2018	299.00	1,090.00
12		9/07/2018	356.00	1,109.00
13		10/07/2018	435.00	1,111.00
14		11/07/2018	318.00	1,089.00
15		12/07/2018	358.00	1,117.00
16		13/07/2018	413.00	1,208.00
17		14/07/2018	346.00	1,284.00
18		15/07/2018	449.00	1,215.00
19		16/07/2018	489.00	1,206.00
20		17/07/2018	277.00	1,041.00

Strangely, the rolling sum seems to be adding up the future dates. This is because we entered '3' as the **number_of_intervals.** Therefore, it looks like positive intervals means that the measure will use future dates. Let's try '-3':

```
=CALCULATE(
        SUM(SaleDataJul1[Total Sales]),
                DATESINPERIOD(
                        SaleDataJul1[Date],
                        MIN(SaleDataJul1[Date]),
                        -3,
                        DAY
                )
        )
```

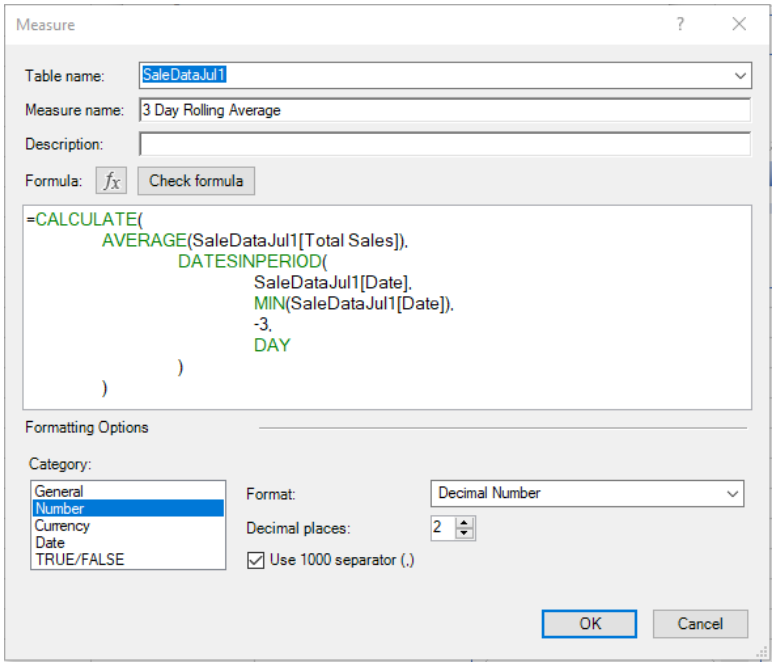


Dragging the new measure into our PivotTable:

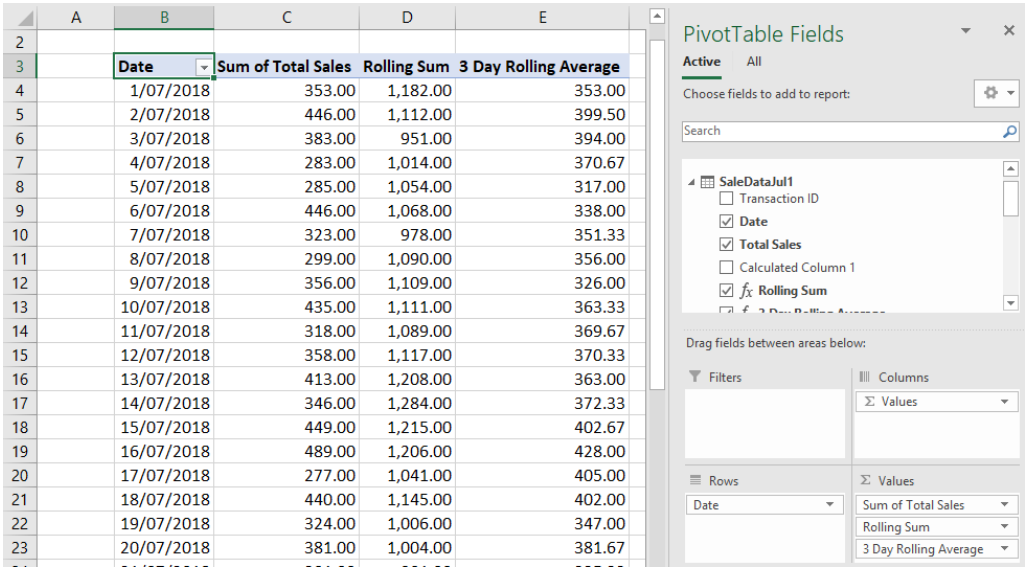
Date	Sum of Total Sales	Rolling Sum
1/07/2018	353.00	353.00
2/07/2018	446.00	799.00
3/07/2018	383.00	1,182.00
4/07/2018	283.00	1,112.00
5/07/2018	285.00	951.00
6/07/2018	446.00	1,014.00
7/07/2018	323.00	1,054.00
8/07/2018	299.00	1,068.00
9/07/2018	356.00	978.00
10/07/2018	435.00	1,090.00
11/07/2018	318.00	1,109.00
12/07/2018	358.00	1,111.00
13/07/2018	413.00	1,089.00
14/07/2018	346.00	1,117.00

Now the measure is adding up sales from the previous dates rather than the future dates. This is all well and good, but having the rolling sum isn't that useful. What if we changed the **SUM** function into an **AVERAGE** function instead?

```
=CALCULATE(
        AVERAGE(SaleDataJul1[Total Sales]),
                DATESINPERIOD(
                        SaleDataJul1[Date],
                        MIN(SaleDataJul1[Date]),
                        -3,
                        DAY
                )
        )
```



Adding this measure into our PivotTable:



Date	Sum of Total Sales	Rolling Sum	3 Day Rolling Average
1/07/2018	353.00	1,182.00	353.00
2/07/2018	446.00	1,112.00	399.50
3/07/2018	383.00	951.00	394.00
4/07/2018	283.00	1,014.00	370.67
5/07/2018	285.00	1,054.00	317.00
6/07/2018	446.00	1,068.00	338.00
7/07/2018	323.00	978.00	351.33
8/07/2018	299.00	1,090.00	356.00
9/07/2018	356.00	1,109.00	326.00
10/07/2018	435.00	1,111.00	363.33
11/07/2018	318.00	1,089.00	369.67
12/07/2018	358.00	1,117.00	370.33
13/07/2018	413.00	1,208.00	363.00
14/07/2018	346.00	1,284.00	372.33
15/07/2018	449.00	1,215.00	402.67
16/07/2018	489.00	1,206.00	428.00
17/07/2018	277.00	1,041.00	405.00
18/07/2018	440.00	1,145.00	402.00
19/07/2018	324.00	1,006.00	347.00
20/07/2018	381.00	1,004.00	381.67

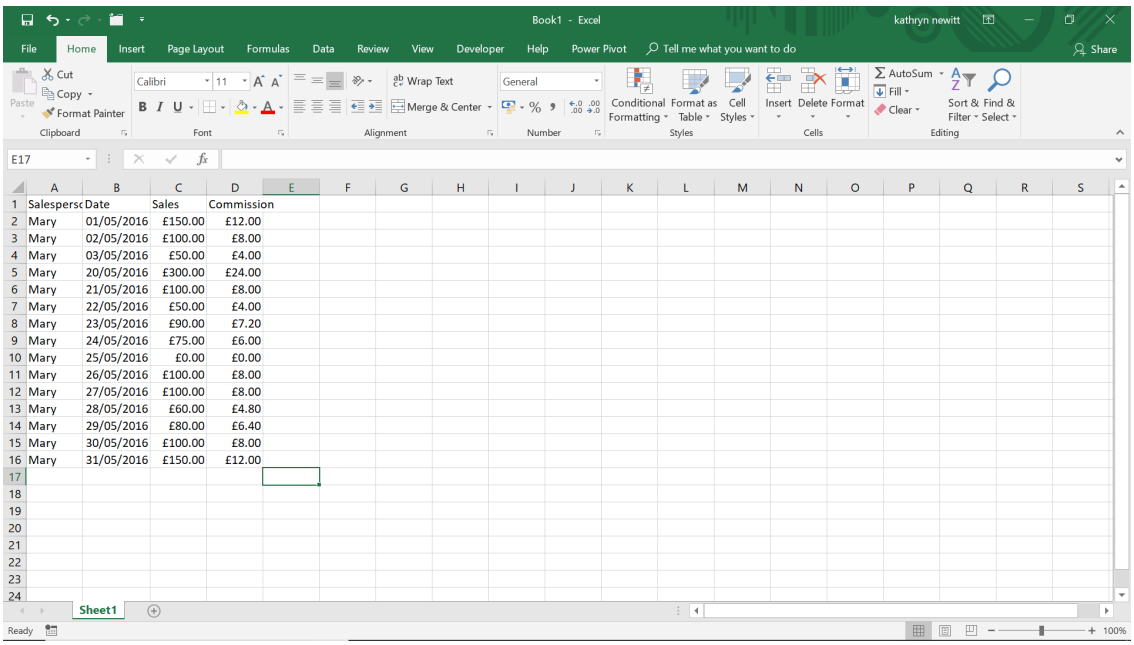
Thus, a rolling average measure where we can change the number of days / periods we want to include in the average.

That's it for this month; more next time.

Power Query Pointers

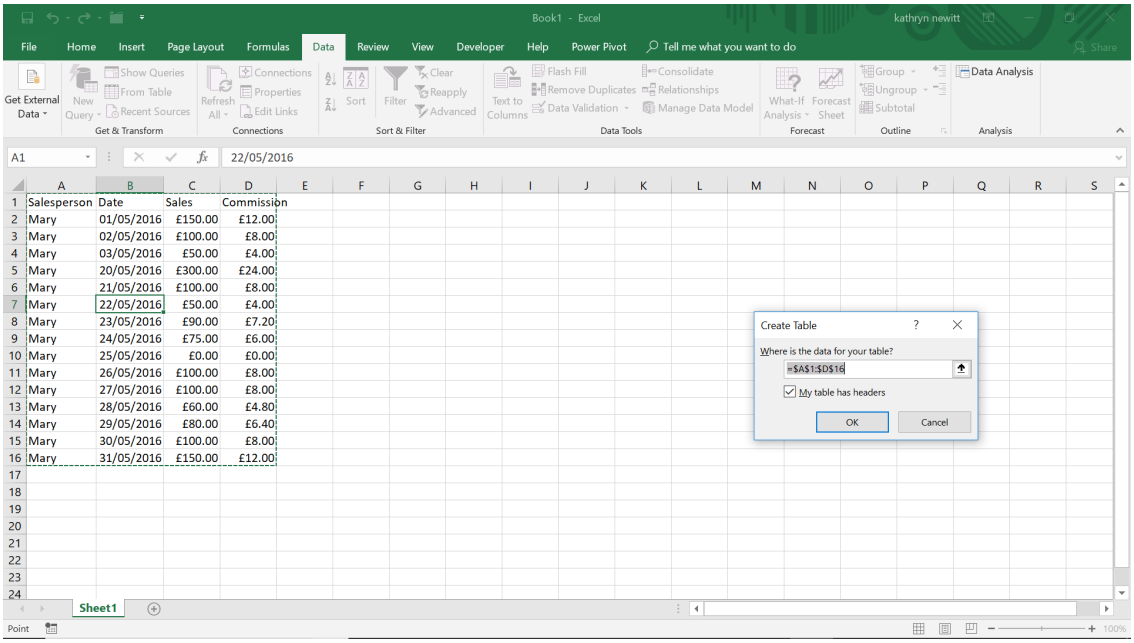
Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Office 365, Excel 2016 and 2019) from www.sumproduct.com/blog. If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month, we consider how to fill in rows when there is a gap in dates.

Let's have the following data from our imaginary salesperson, Mary.



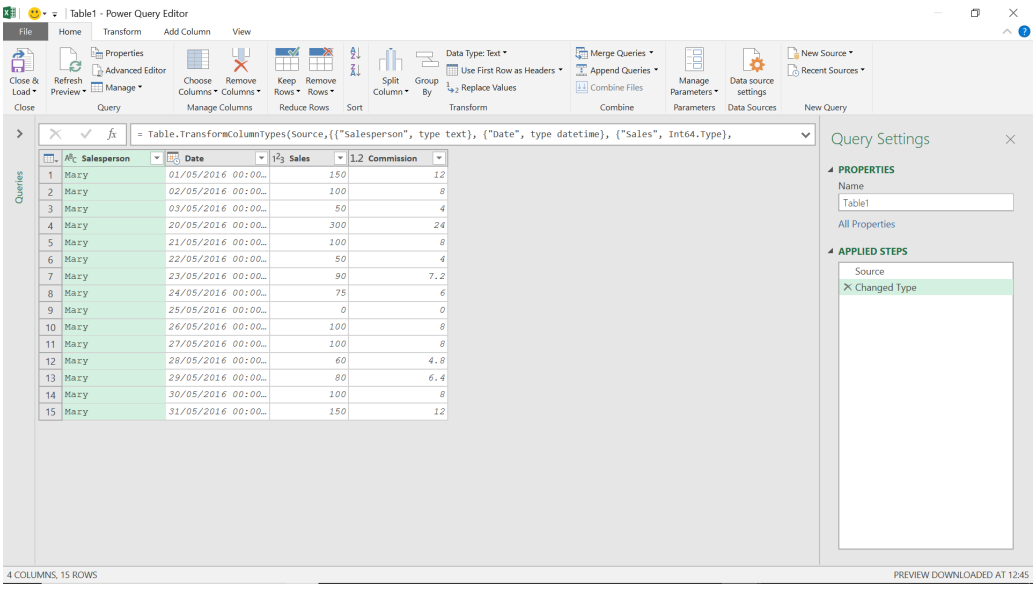
Salesperso	Date	Sales	Commission
Mary	01/05/2016	£150.00	£12.00
Mary	02/05/2016	£100.00	£8.00
Mary	03/05/2016	£50.00	£4.00
Mary	20/05/2016	£300.00	£24.00
Mary	21/05/2016	£100.00	£8.00
Mary	22/05/2016	£50.00	£4.00
Mary	23/05/2016	£90.00	£7.20
Mary	24/05/2016	£75.00	£6.00
Mary	25/05/2016	£0.00	£0.00
Mary	26/05/2016	£100.00	£8.00
Mary	27/05/2016	£100.00	£8.00
Mary	28/05/2016	£60.00	£4.80
Mary	29/05/2016	£80.00	£6.40
Mary	30/05/2016	£100.00	£8.00
Mary	31/05/2016	£150.00	£12.00

Mary has supplied us with her sales figures for May 2016 (she is a little late with her report!). We need to add these to existing data, but there is a problem. Mary was on holiday from May 4th to May 19th, so she hasn't created any data for these dates. We want to add the missing rows.

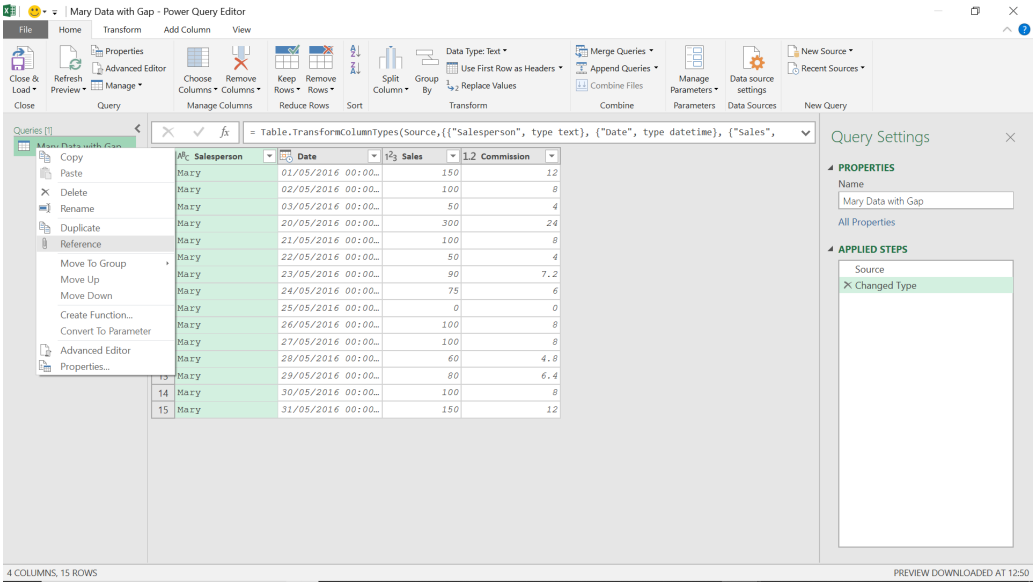


Salesperson	Date	Sales	Commission
Mary	01/05/2016	£150.00	£12.00
Mary	02/05/2016	£100.00	£8.00
Mary	03/05/2016	£50.00	£4.00
Mary	20/05/2016	£300.00	£24.00
Mary	21/05/2016	£100.00	£8.00
Mary	22/05/2016	£50.00	£4.00
Mary	23/05/2016	£90.00	£7.20
Mary	24/05/2016	£75.00	£6.00
Mary	25/05/2016	£0.00	£0.00
Mary	26/05/2016	£100.00	£8.00
Mary	27/05/2016	£100.00	£8.00
Mary	28/05/2016	£60.00	£4.80
Mary	29/05/2016	£80.00	£6.40
Mary	30/05/2016	£100.00	£8.00
Mary	31/05/2016	£150.00	£12.00

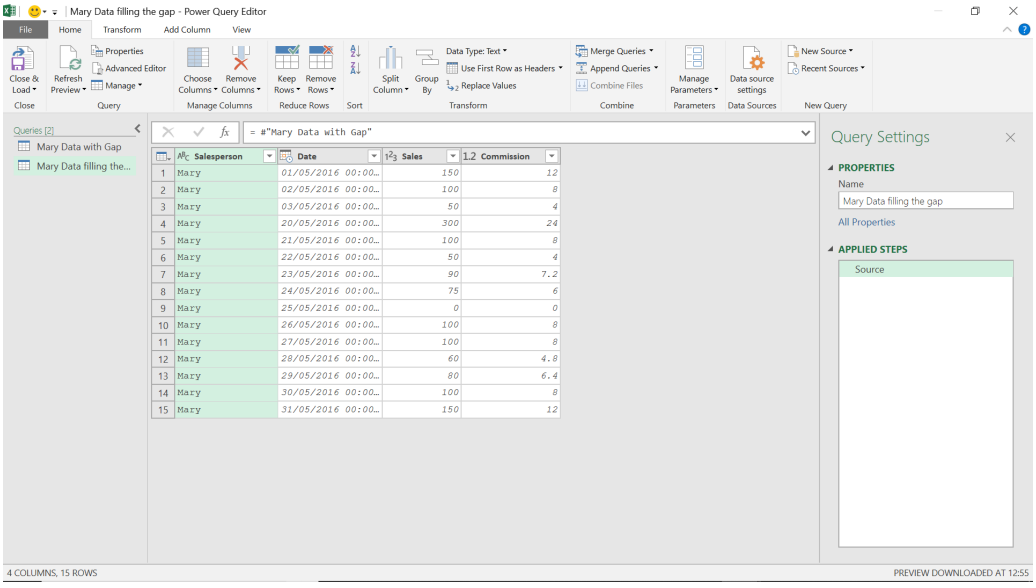
We begin by creating a query 'From Table' in the 'Get & Transform' section of the 'Data' tab. We are prompted to define the boundaries of our Table (and to check that headers exist) and we'll accept the defaults.



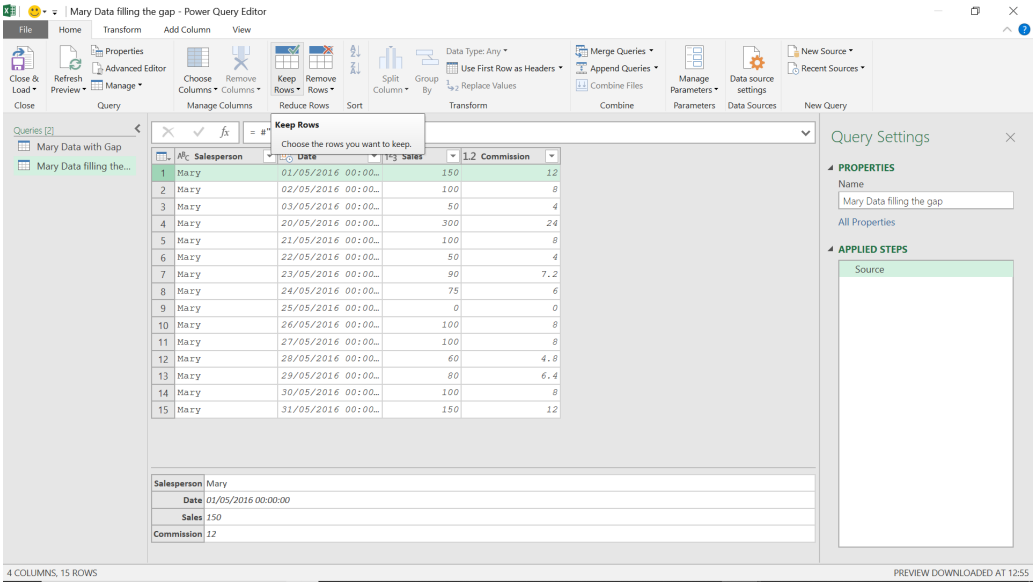
Now we need to add the missing rows.



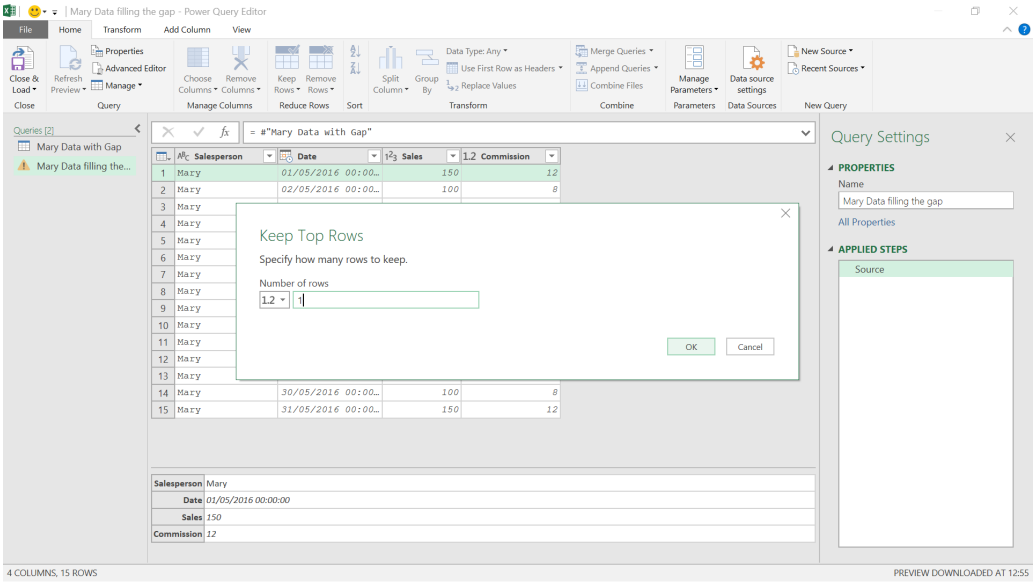
We will start by creating a copy of the existing query, and to do this, we will create a reference query. For more on the differences between reference and duplicate queries see Power Query: Cleanse, Tone and Upload.



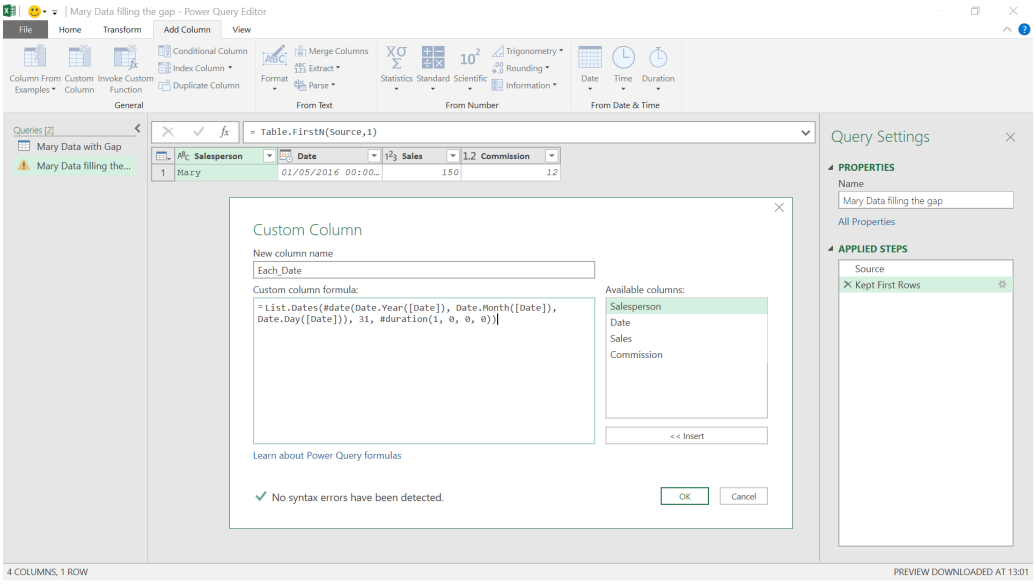
We now have two queries containing Mary's data; we are going to savage this query by only keeping one row!



We can do this by using the 'Keep Rows' option:

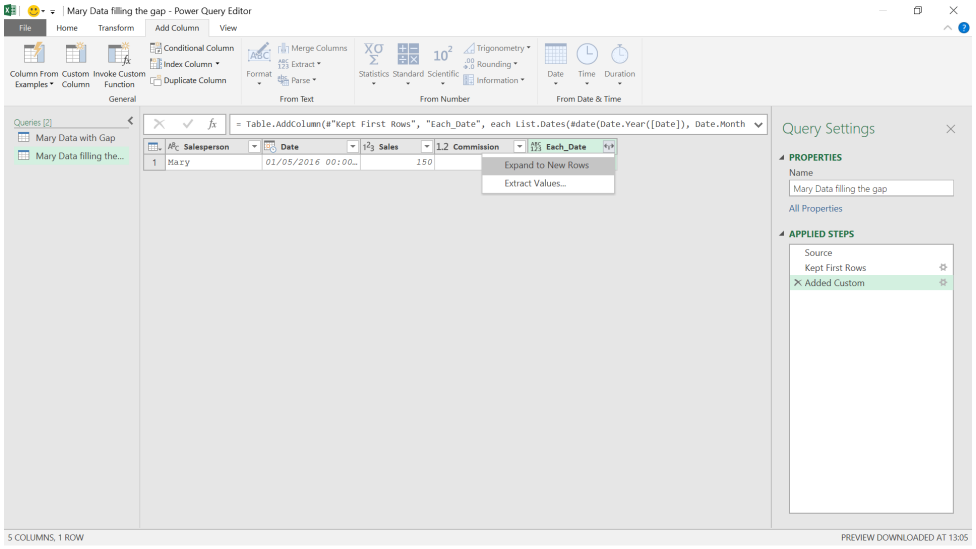


We choose to keep just the top row.

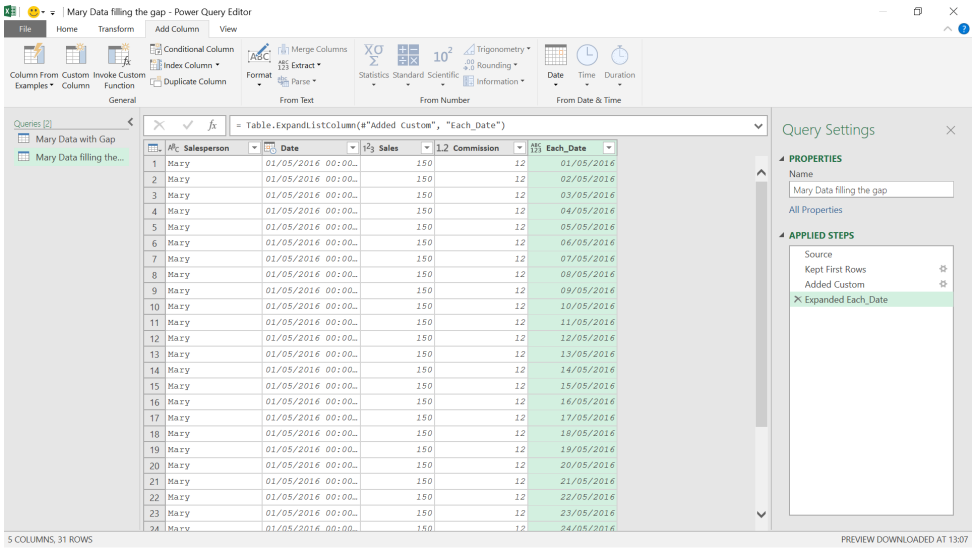


We create a new column by using 'Custom Column' from the 'Add Column' tab. We use this to create a list of dates from the date on the row for 31 days:

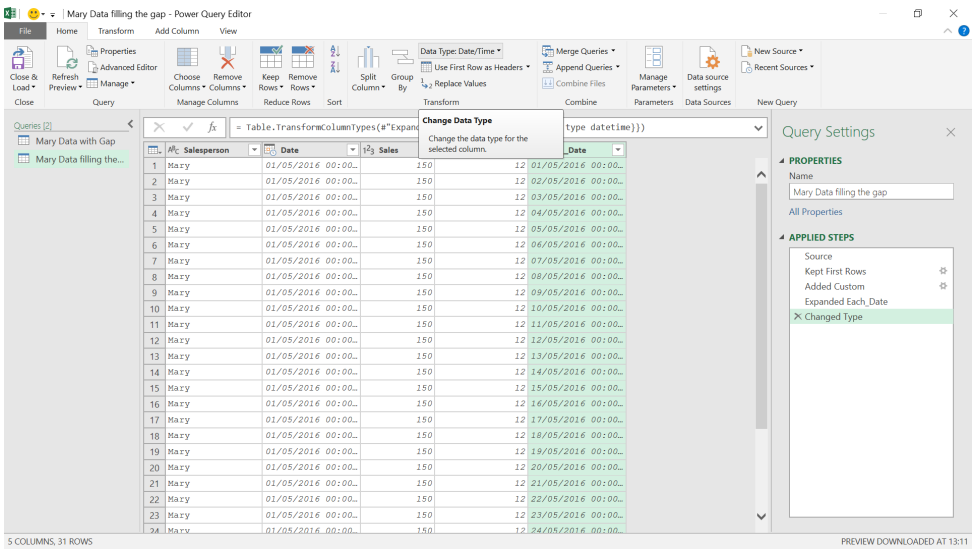
= List.Dates(#date(Date.Year([Date]), Date.Month([Date]), Date.Day([Date])), 31, #duration(1, 0, 0, 0))



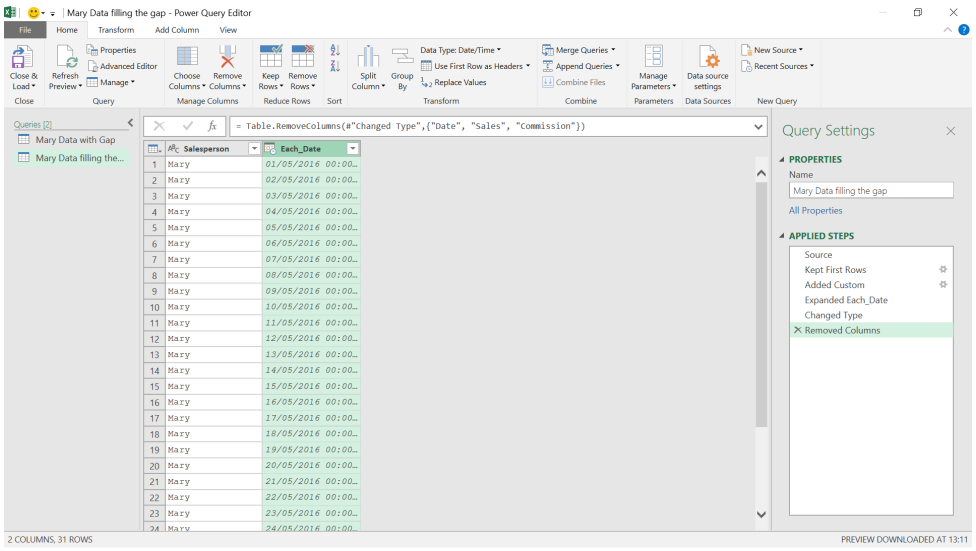
We expand the list in **Each_Date** to new rows.



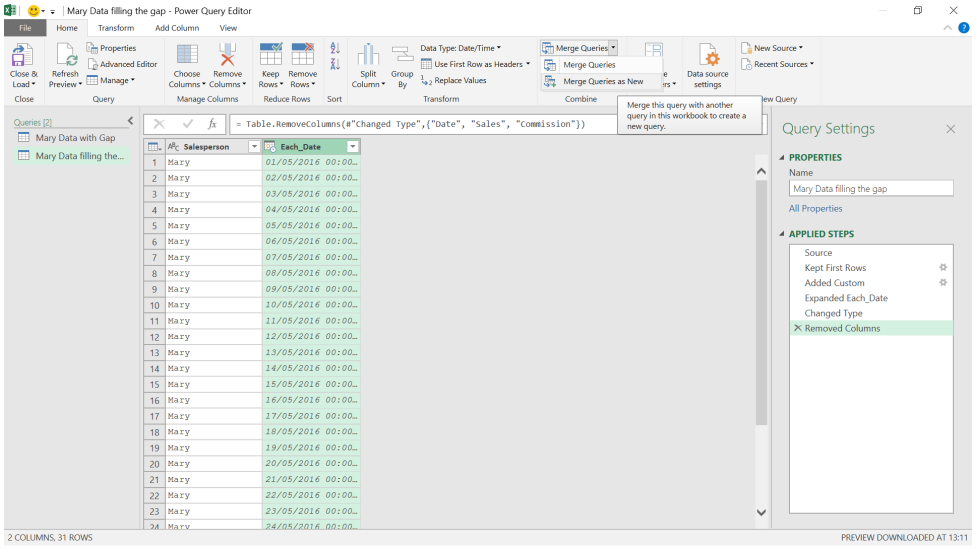
We now have a row for each date in the date range. Our new column doesn't look like a date though, so we need to change the data type ready for the next step.



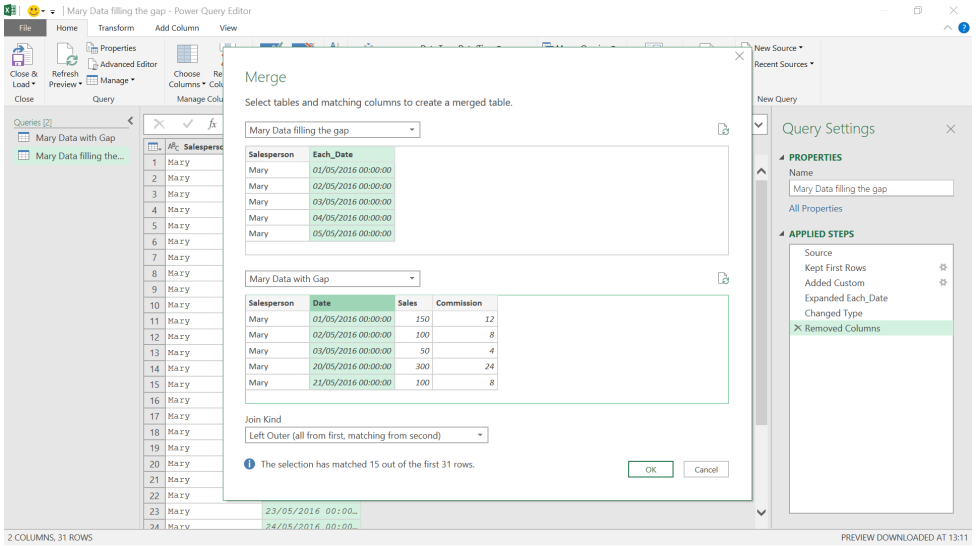
We want to simplify my query as we don't need the original **Date**, **Sales** and **Commission** columns – these will come from Mary's data.



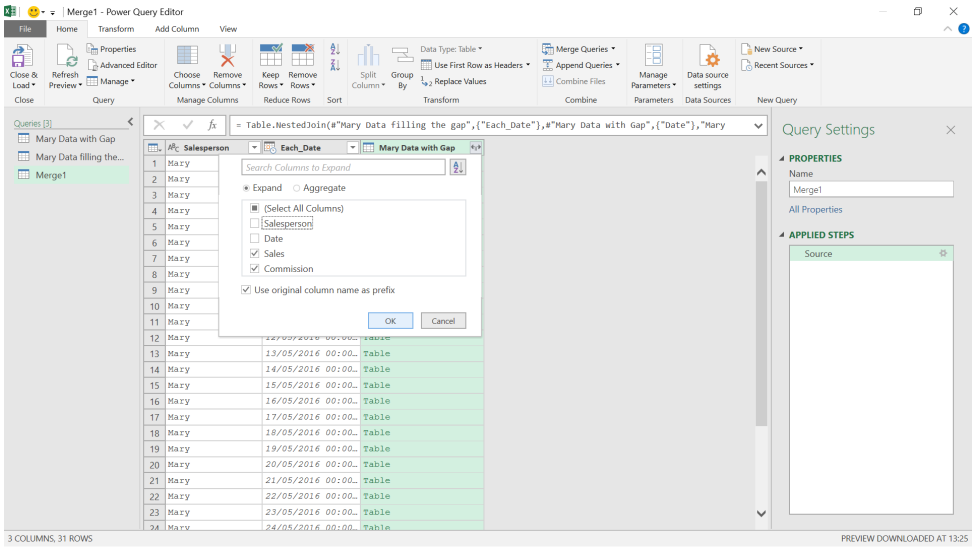
Now all we need to do is put Mary's data back in.



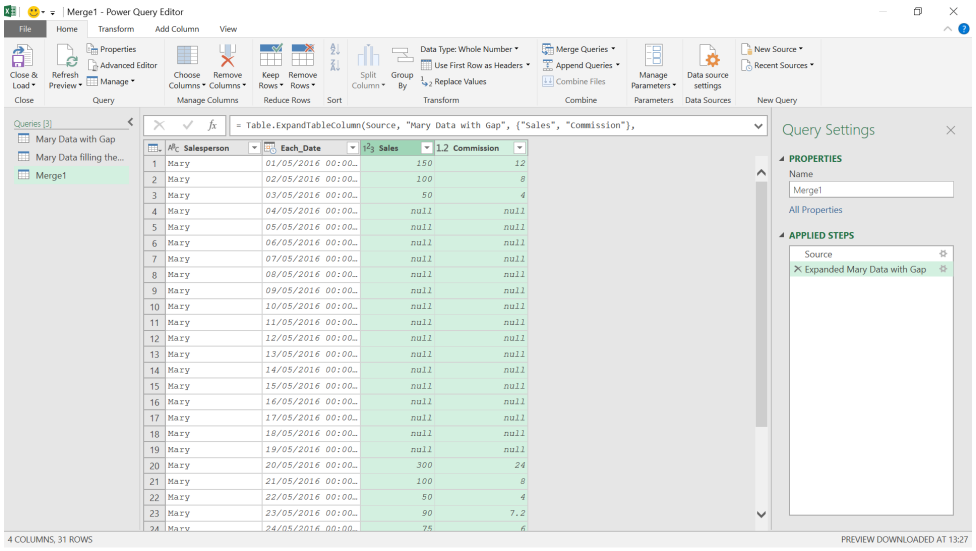
We choose 'Merge Queries as New' from the 'Merge Queries' section from the 'Home' tab.



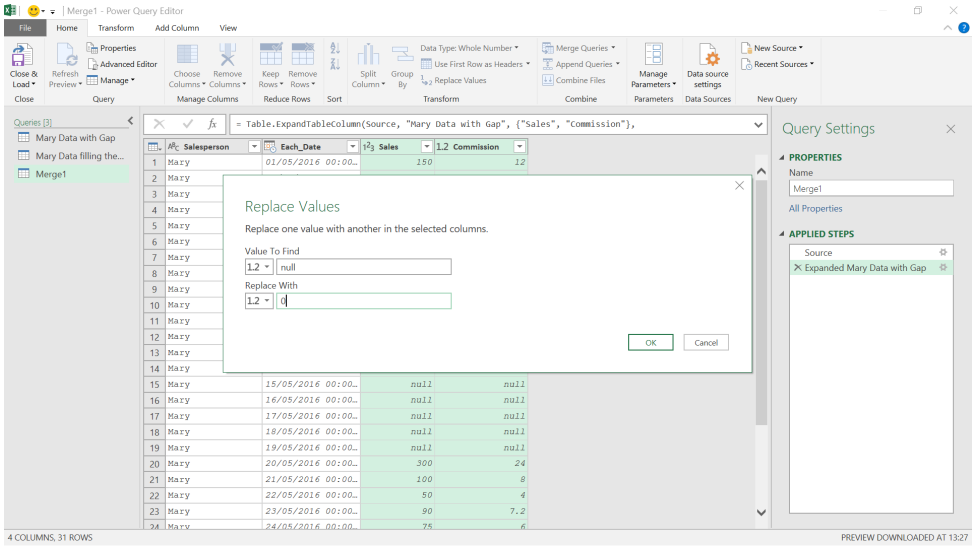
We want all the rows from our first query and matching rows (with the sales data) from the second query. We will use the 'Left Outer' join.



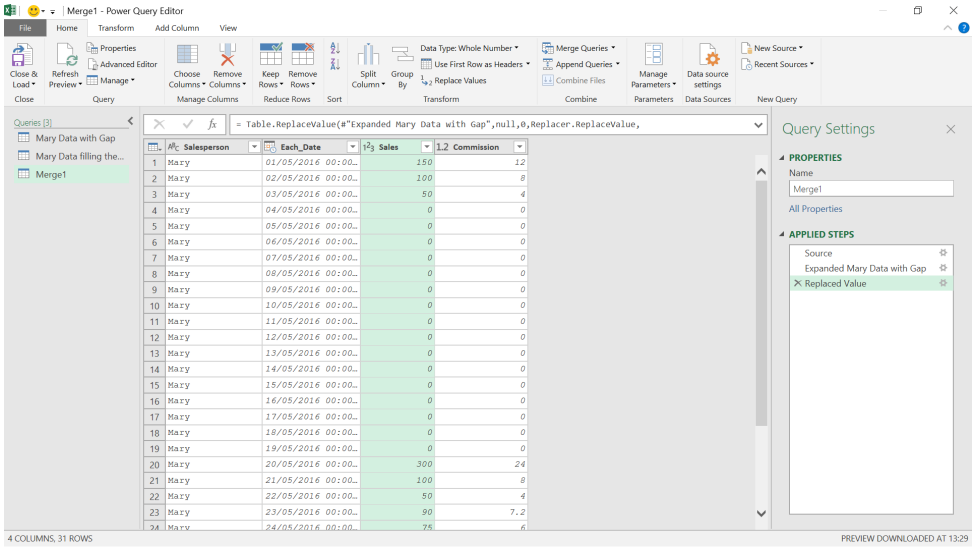
A new query 'Merge1' is created, and Mary's data is held in column **Mary Data with Gap** which contains a table. We choose to expand the **Sales** and **Commission** columns. We will (of course) uncheck the 'Use original column name as prefix' option.



We now have data for each date – but it needs to be tidied. We want zeroes instead of *null* values in the currency columns.



We may replace values to achieve this.



We now have entries for each date so that this data can be combined with other similar data.

Until next month.

Power BI Updates

Another month, another load of updates for Power BI. The latest round allows you to customise your reference layers in Azure Maps visual, DAX query view is now available in live connect and there is also an update to the Power BI enhanced report format (PBIR).

The full list of updates is as follows:

Reporting

- Customise your reference layers in the Azure Maps visual

Modelling

- Announcing the General Availability of the enhanced Row Level Security editor in Power BI Desktop
- DAX query view is now available in live connect
- Add or update multiple measures in DAX query view

Data Connectivity

- Certified connector updates

Service

- Storytelling in PowerPoint: new 'Export to PowerPoint' dialog

Developers

- New update for field parameter feature for custom visuals
- Power BI enhanced report format (PBIR) update

Visualisations

- New visuals in AppSource
- Linear Gauge by Powerviz
- Drill Down Map PRO by ZoomCharts
- PowerGantt Chart by Nova Silva
- Advanced Geospatial Analytics Made Simple with Icon Map Pro for Power BI

Paginated Reports

- Bind to Gateway API support for Paginated Reports
- Parameters, Header / Footer and much more in the web authoring experience for Paginated Reports in Preview

Power BI Report Server

- Power BI Report Server key in Fabric Capacities.

Let's look at each in turn.

Customise your reference layers in the Azure Maps visual

Recently, Microsoft has brought numerous improvements to the Azure Maps visual, including reference layer support for a variety of new data formats. This month sees the announcement of several more improvements to reference layers:

- CSV support
- new customisation options
- dynamic URL sources.

Firstly, the Azure Maps visual now supports CSV files as data sources for reference layers. Just as you can already use GeoJSON, Shapefiles, WKT and KML files, you may now upload a CSV file instead in the reference layer section of the Formatting pane.

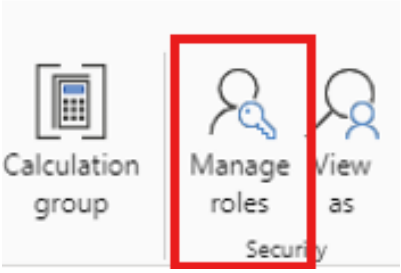
Secondly, you may also now format reference layer shapes from within the Formatting pane. Previously, Azure Maps required you to define the colour and width of points, lines and polygons from within your reference layer files. Otherwise, these shapes would be drawn on your maps with the default colours and formatting. This requirement brought additional complexity to working with your reference layers in Power BI, since the files needed more than just the data you intended to visualise. Now, these have been added as standard formatting settings to each type of object in the reference layers in the Formatting pane, so you can customise them directly from within Power BI.

Lastly, for those of you who need your reference layers to change with time or other data-bound conditions, you can now provide a dynamic

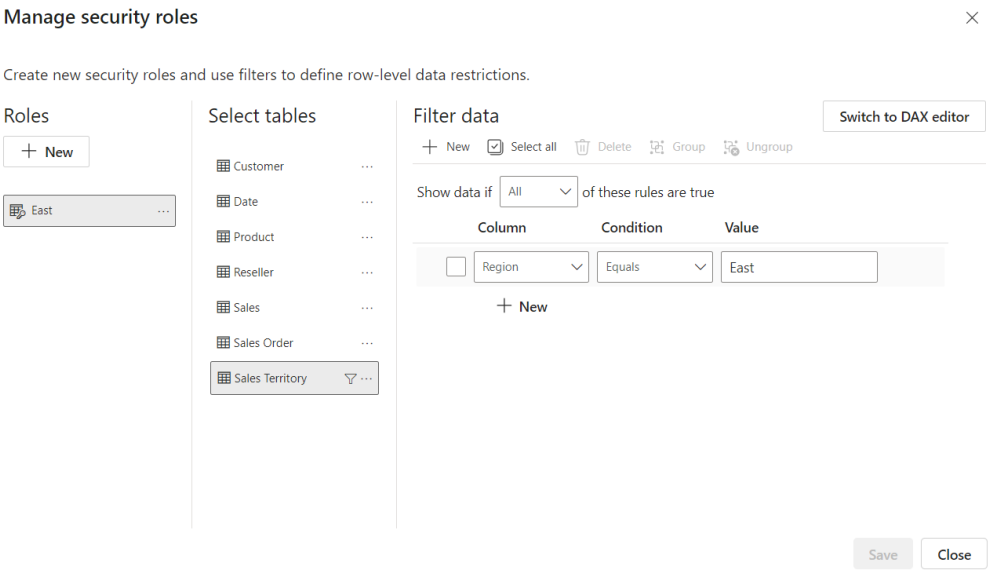
URL using conditional formatting. This allows you to set custom logic to determine the reference layer URL the Azure Maps visual will use. For example, you can load in different reference layers based upon the categories selected by a slicer, *e.g.* to visualise performance of different product lines over the same geography.

Announcing the General Availability of the enhanced Row Level Security editor in Power BI Desktop

This month also welcomes the General Availability of the enhanced Row Level Security editor in Power BI Desktop. With this editor, you can quickly and easily create row-level security roles and filters. Simply choose 'Manage roles' from the Ribbon to open the editor.

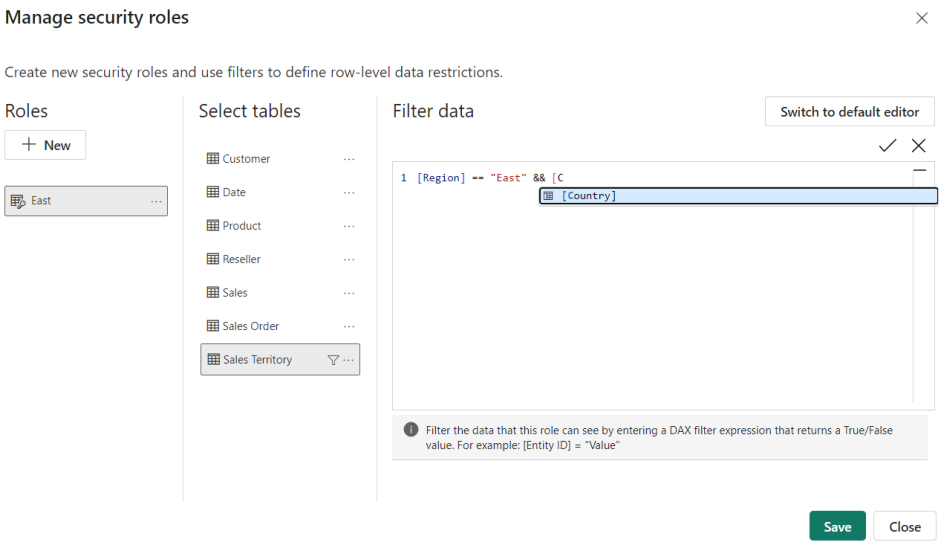


By default, this will open an easy-to-use drop-down interface for creating and editing security roles all without having to write any **DAX**.



If you prefer using **DAX** or need it for your filter definitions, you can switch to use the DAX editor to define your role. This is a rich editor complete with autocomplete for formulae (IntelliSense). It also allows you to easily verify the validity of your **DAX** expressions by selecting the check button and revert any changes by selecting the **X** button.

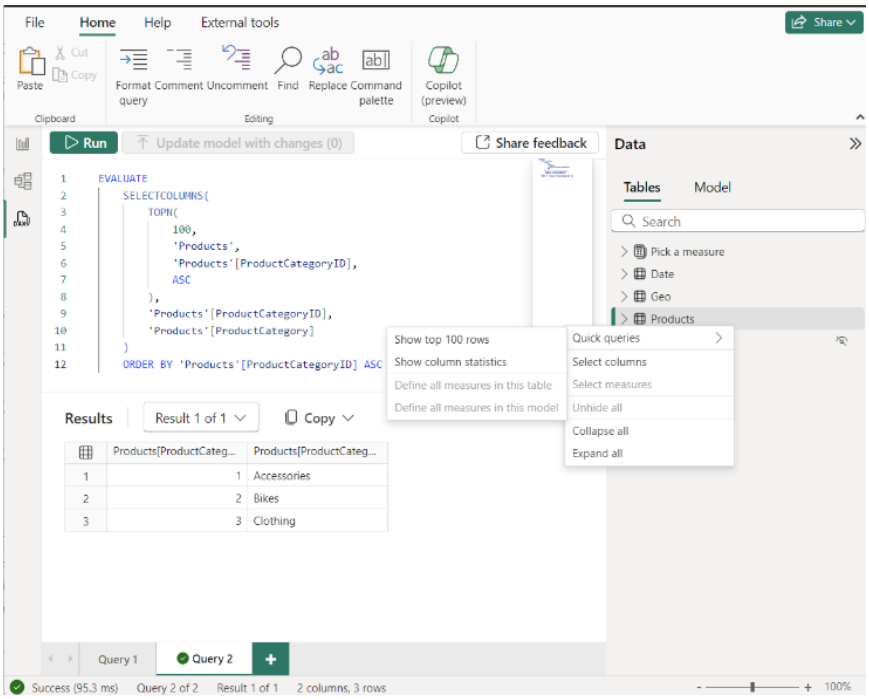
At any point you can also switch back to the default editor by selecting 'Switch to default editor'. All changes made in either editor persist when switching interfaces when possible, giving you maximum flexibility as you create your row-level security roles.



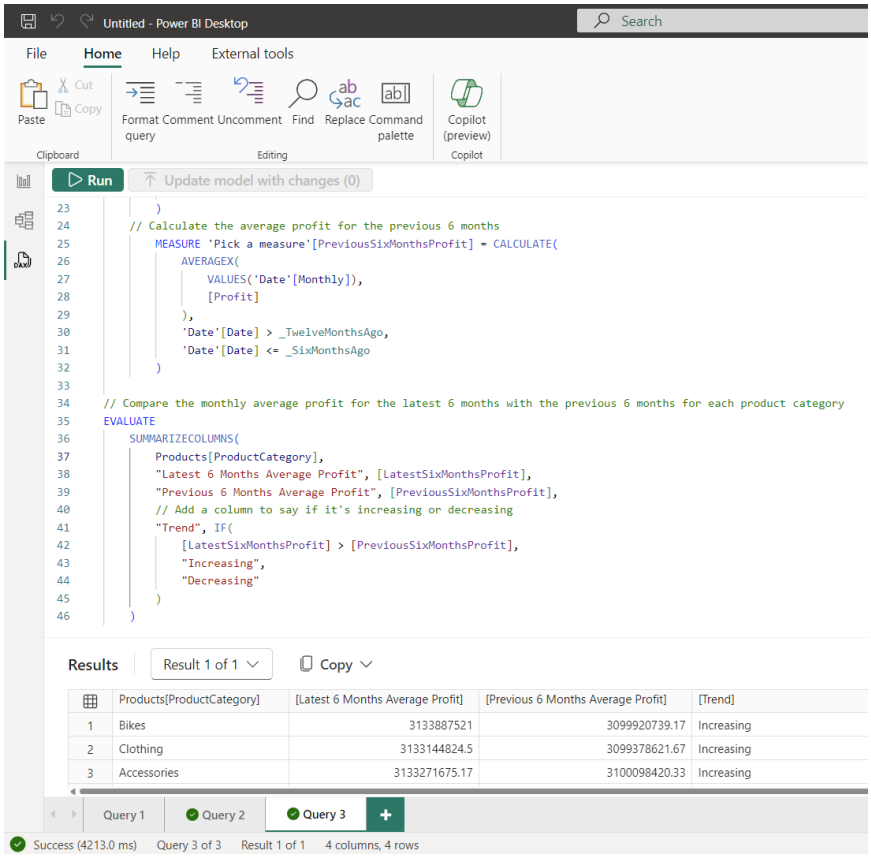
DAX query view is now available in live connect

There is now the ability to use DAX query view while live connected to a published semantic model. With this release you may write **DAX** queries with DAX query view when live connected to a published semantic model in Power BI Desktop.

This includes live connecting to Direct Lake semantic models created in Microsoft Fabric. You may live connect to your published Direct Lake, import, DirectQuery or composite semantic model in Desktop and use the DAX query view to quickly view data without having to create any visuals. Use quick queries to have a **DAX** query generated for you from any table, column or measure, and Copilot can help you as you write your **DAX** queries.



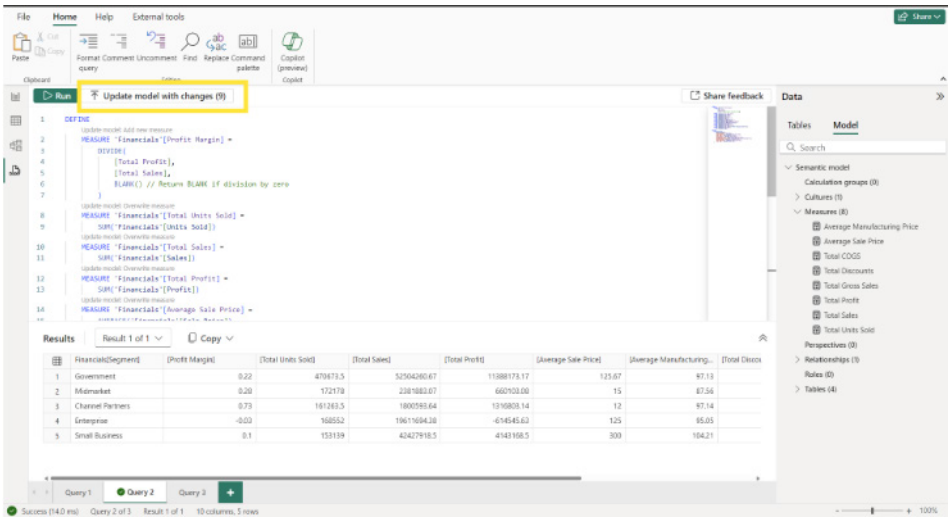
This can be helpful to further your analysis beyond report authoring even when you are using a published semantic model managed by someone else.



Add or update multiple measures in DAX query view

Another popular request for DAX query view is also now available, namely to add or update the model with multiple measure changes.

In a **DAX** query, you can use the **DEFINE** syntax to add a measure. These **DAX** query scoped measures are helpful for authoring **DAX** formulae and trying them out with different groups by columns before adding them to the model. In DAX query view, Microsoft has made it simpler to then add these measures to the model by clicking the text between the lines above the **DEFINE MEASURE**. **DAX** queries also allow you to define many measures at once, so it can be tedious to click each one. Now this task is very easy with the option to update the model with a single click for all measures.



This can be handy to quickly format all your **DAX** formulae at once:

- in DAX query view, right-click in the Data pane and choose **Quick queries -> Define all measures in this model**
- click the 'Format query' button in the Ribbon
- click 'Update model with changes' button.

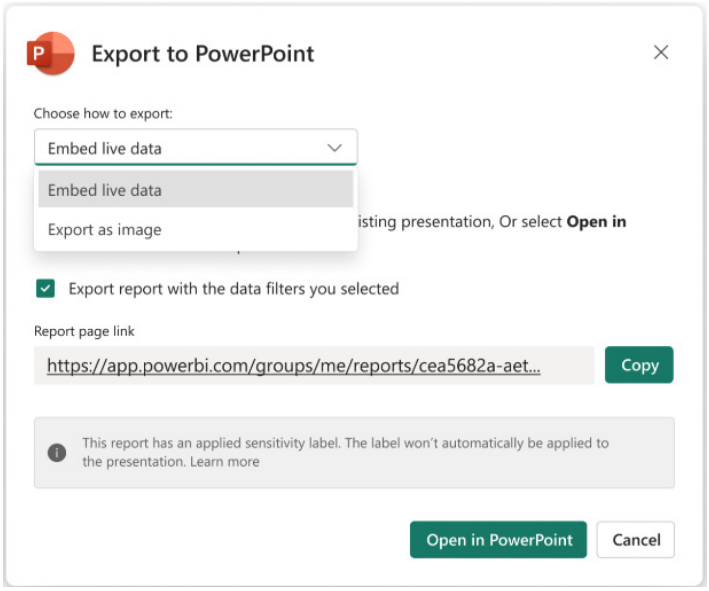
Certified connector updates

There are four new / updated connectors in this latest release:

1. BuildingConnected *(new)*
2. SingleStore *(updated)*
3. SmartSheet *(updated)*
4. Windsor *(new)*.

Storytelling in PowerPoint: new 'Export to PowerPoint' dialog

To reduce complexity and drive more clarity, Microsoft has now merged the two [2] options to export to PowerPoint into a single dialog. You may now choose between embedding live data using the Power BI add-in for PowerPoint and exporting the report as images from the same dialog.



New update for field parameter feature for custom visuals

sourceFieldParameters is a new property in **DataViewMetadataColumn** that identifies whether a query field results from a field parameter resolution. If a single field can originate from multiple field parameters, this property will list all the related field parameters. This new update is available with API v5.10.0.

Power BI enhanced report format (PBIR) update

The following previously announced limitations of the PBIR format have been resolved:

- cannot be exported to PPTX or PDF
- cannot be included in Subscriptions
- Mobile layouts aren't applied
- cannot be used in Power BI Embedded.

New visuals in AppSource

There are eight [8] new visuals this month:

1. Icon Map Pro
2. Multilevel Matrix Xerppa
3. Sankey Diagram
4. Smart Grid-Map
5. SPC visual
6. StackedTrends visual
7. Waffle Chart Maker
8. Waterfall chart.

Linear Gauge by Powerviz

Powerviz Linear Gauge is a visual that is used to display the progress against set targets on a linear scale, with an axis displaying a range of values or percentages. The Linear Gauge quickly conveys the status or progress of a task or value being measured.

Key features include:

- **Gauge styles:** four [4] different gauges including Linear, Bar in Bar, Cylinder, Thermometer and various customisation options
- **Templates:** select from pre-made templates or customise your own
- **Scale:** select an absolute or percentage scale, with a customisable minimum - maximum range
- **Targets:** set a custom target or apply a target using a value field
- **Data colours:** 30+ colour palettes available
- **Band:** 30+ colour palettes and customisation options
- **Labels: i**mprove readability with labels
- **Small Multiples:** divide visuals based upon fields
- **Ranking:** filter Top / Bottom **N** shows remaining as "Others".

Other features included are fill pattern, annotation, grid view, show condition amongst others.

Business use cases include sales performance tracking, project milestone monitoring and financial KPI analysis.

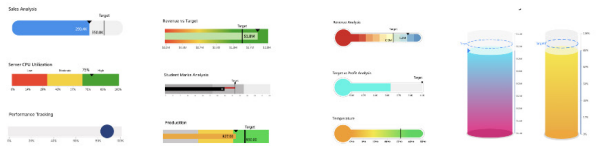
Linear Gauge

Powerviz
www.powerviz.ai

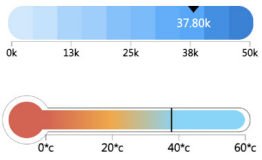
INTRODUCING LINEAR GAUGE BY POWERVIZ

An advanced visual to track the progress on a linear scale against the targets.

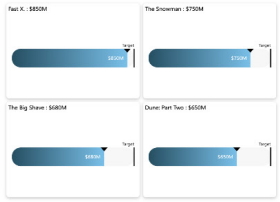
Four Different Gauge Style



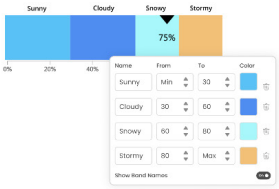
Linear | Bar In Bar | Thermometer | Cylinder



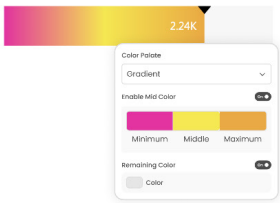
Scale



Small Multiple



Band Customization



Color Scheme

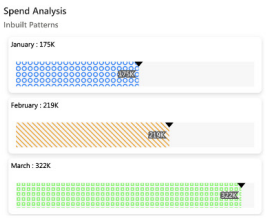
Linear Gauge

Powerviz
www.powerviz.ai

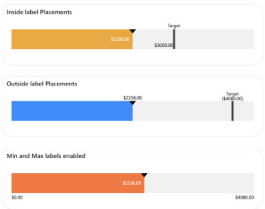
INTRODUCING LINEAR GAUGE BY POWERVIZ

Four in one gauge, Custom templates, Scale, Bands Fill patterns, Small multiples, 30+ Palettes, & More.

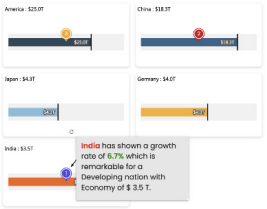
Conditional Formatting



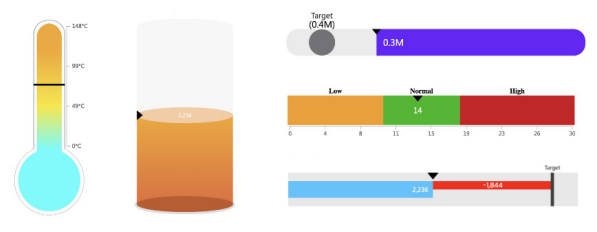
Fill Patterns



Labels



Annotation



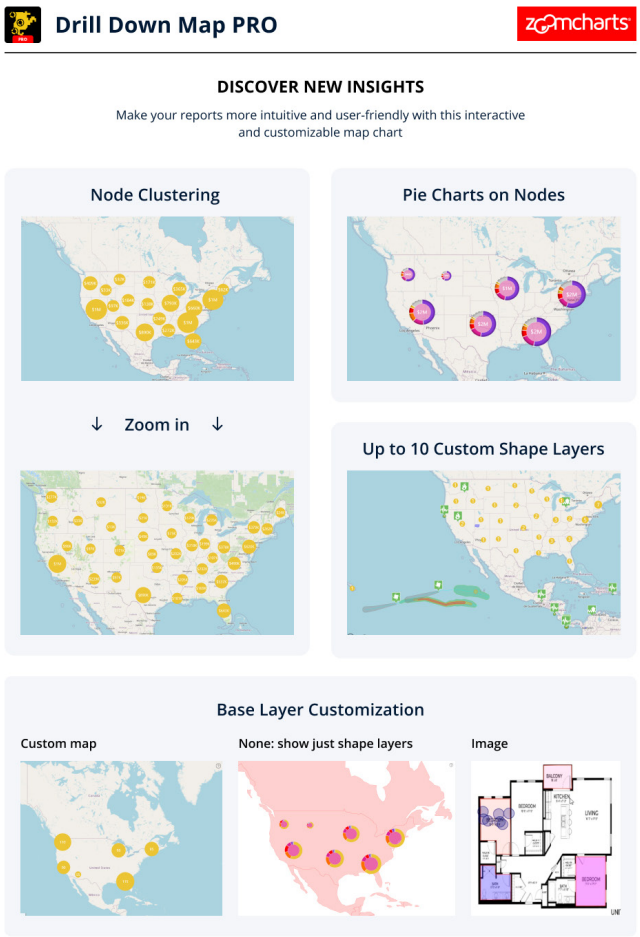
Drill Down Map PRO by ZoomCharts

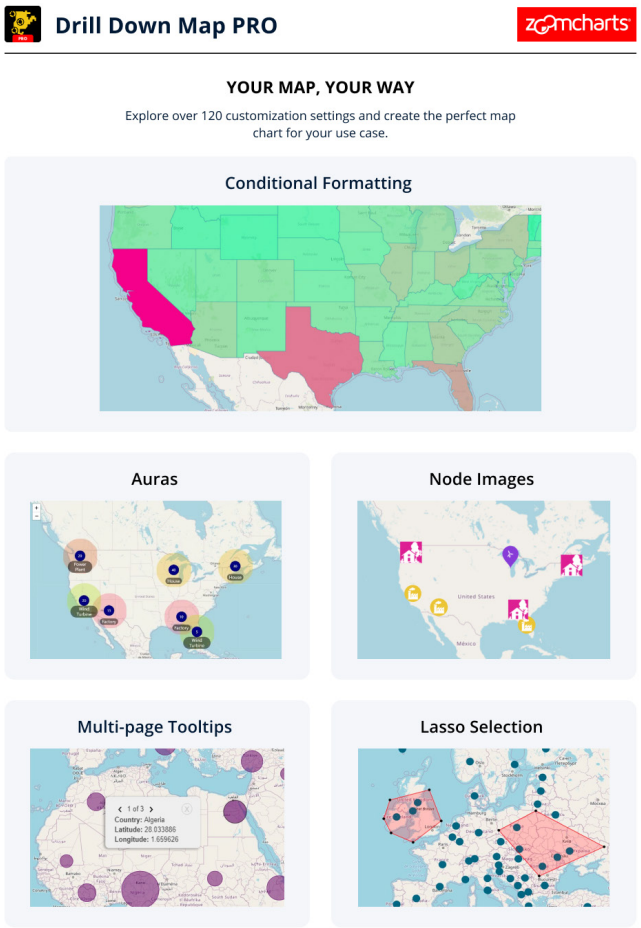
When visualising data with geographic coordinates, maps are usually the best way to go. Map charts are therefore becoming a more popular visualisation type in Power BI reports, and the Drill Down Map PRO custom visual by ZoomCharts expands on the capabilities of built-in Map charts.

Features include:

- **Node clustering:** multiple nearby nodes can create clusters and even display the values as Pie charts; simply zoom in to drill down
- **Base layer customisation:** choose between AzureMaps or any custom tileserver, use your own images as the base layer or else disable it entirely
- **Custom shape layers:** enable up to 10 individually customisable shape layers. You may use preset shapes or import your own KML / GeoJSON files
- **Conditional formatting:** automatically apply colour fill to each area by comparing their values against other shapes or by using each shape's own reference value
- **Others:** paginated ToolTips, custom ToolTip fields, auras, node images and a lasso tool.

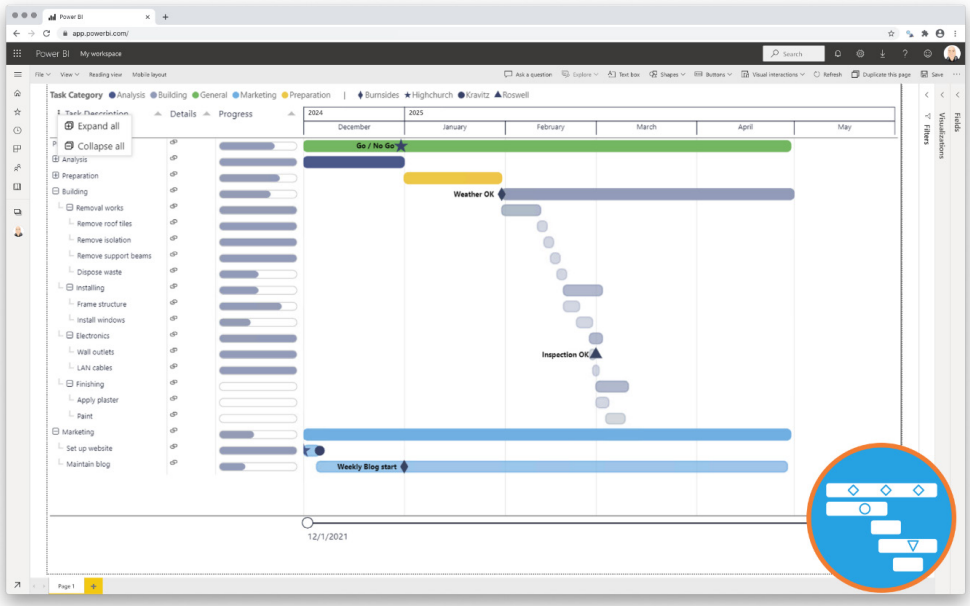
Drill Down Map PRO works well with other visuals by dynamically cross-filtering data, enabling you to build insightful and user-friendly Power BI reports.





PowerGantt Chart by Nova Silva

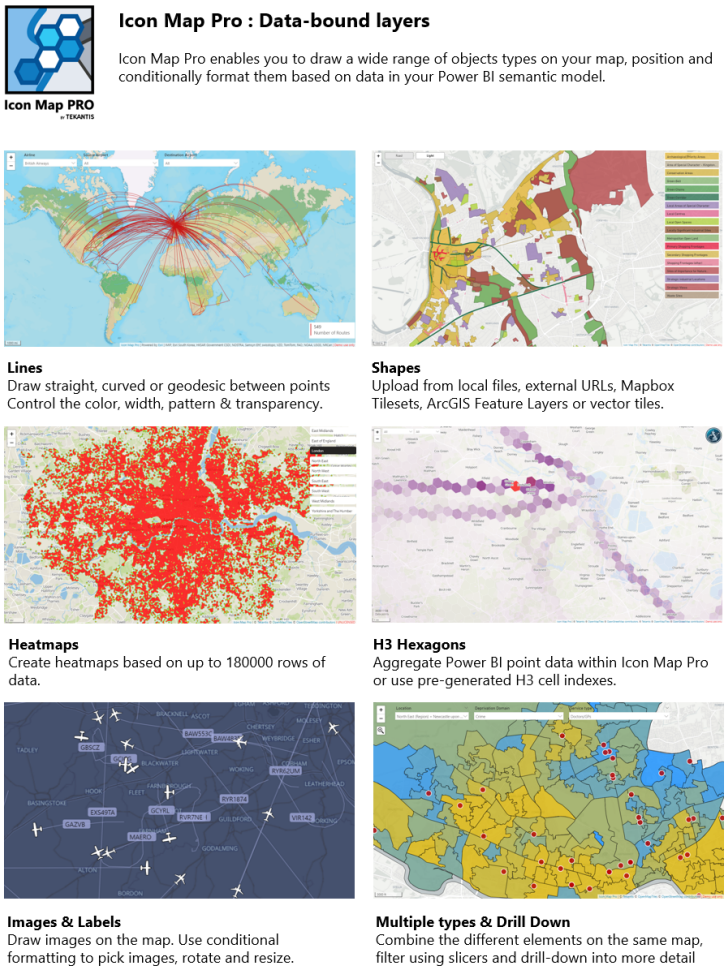
In the latest release of the PowerGantt Chart, several new features have been added. These include the ability to show incomplete tasks and display progress as a separate column. Links have also been included in the additional columns and their formatting options have been enhanced. You can now change the milestone shapes and wrap text in columns for better readability too.



Additionally, an option has been added to preset the zoom slider, expand and collapse all hierarchy elements, and add milestone labels. To further improve the experience, the display and formatting of vertical grid lines has also been enabled.

Advanced Geospatial Analytics Made Simple with Icon Map Pro for Power BI

Icon Map Pro, the new professional version of Icon Map, has been developed with an extensive set of new features and a simplified interface. This tool offers a robust solution for visualising and analysing geospatial data within Power BI. It is designed for data analysts, GIS specialists and business intelligence professionals, addressing the need for integration of geographic insights into Business Intelligence dashboards. Users can now transform complex geospatial data into actionable visuals, enhancing decision-making and strategic planning with the intuitive low / no-code Power BI interface.



Icon Map Pro : Backgrounds, Reference Layers & Interactivity

Icon Map Pro enables you to add additional context to your data, whether that's from the use of specific mapping services, additional reference layers or real-time context such as traffic and weather, all with the familiar experience of a Power BI visual supporting tooltips, cross-highlighting, drill down and drill through.

Map Backgrounds

Icon Map Pro comes with its **own world-wide background mapping service included**.
It is also possible to use third-party providers including Azure Maps, Google Maps, Mapbox, ArcGIS, MapTiler and Ordnance Survey with the appropriate subscription.
In addition, custom layers are supported from raster tiles and vector tiles.

Overlays and Reference Layers

Additional information can be overlayed over the base map. Examples include; Place names, Daylight Terminator, Shapes (from external URLs, uploaded files or ArcGIS Online feature layers), WMS layers, Traffic from Azure Maps and Google Maps, and Weather from Azure Maps and OpenWeatherMap.

Controls and Interactivity

Add controls to control the zoom, add metric, imperial or nautical scales, lasso map elements, measure distances or add custom attribution messages for compliance with custom layers.
Icon Map Pro supports cross filtering and highlighting other visuals, tooltip customisation, report page tooltips, drill down and drill through.

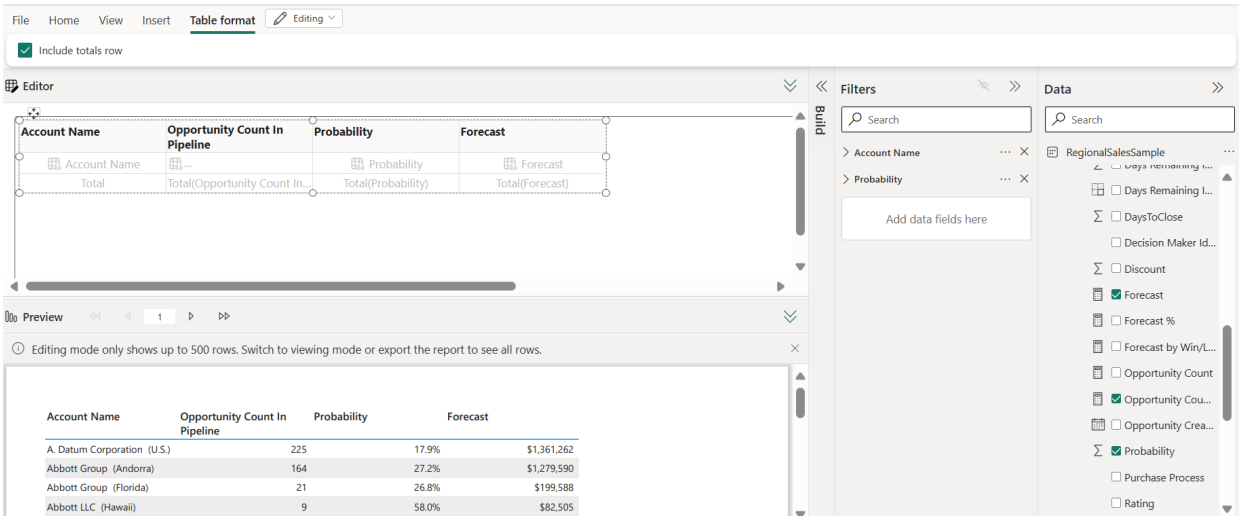
Bind to Gateway API support for Paginated Reports

You can now bind your paginated reports to gateways with a REST API. This will allow Paginated Reports to connect to on-premises gateways without requiring users to go to the user interface in the Power BI Service and bind the report to the specified gateway.

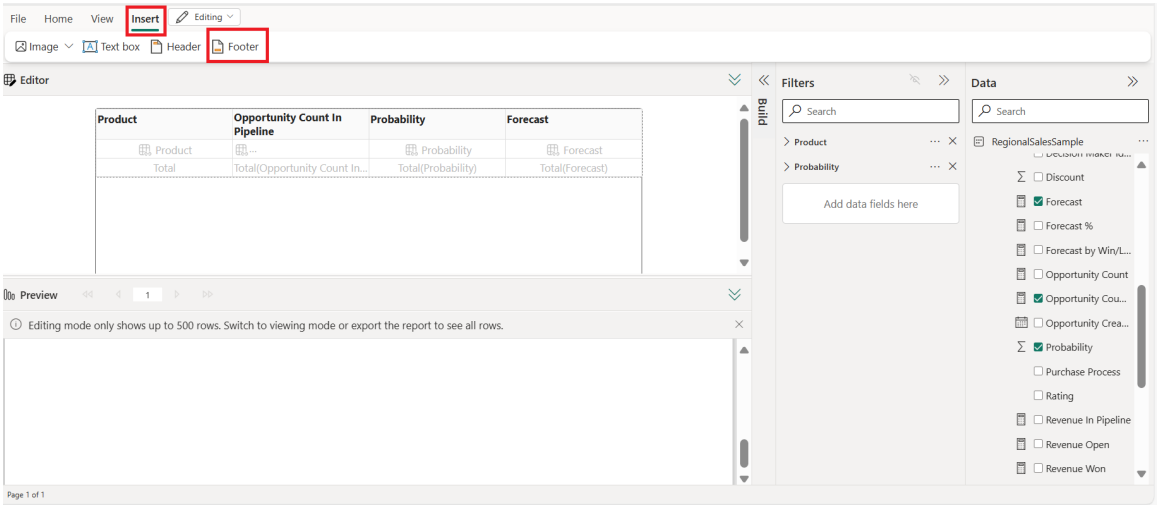
Parameters, Header / Footer and much more in the web authoring experience for Paginated Reports in Preview

A new experience to web author paginated reports has now been introduced. It's not just an update to the look and feel, but also it has introduced a host of new capabilities. You may now define parameters, headers, footers and page numbers in your web authored reports.

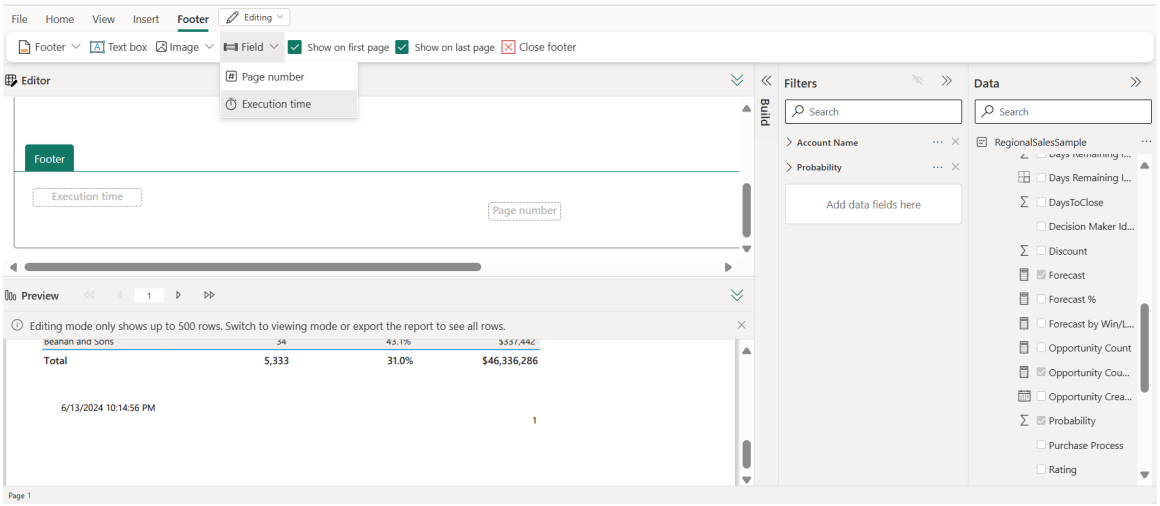
Once you select the fields, they will appear in the Editor along with a 'Preview' of the report with sample data. You can move the table in the Editor and the Preview will reflect the change as well.



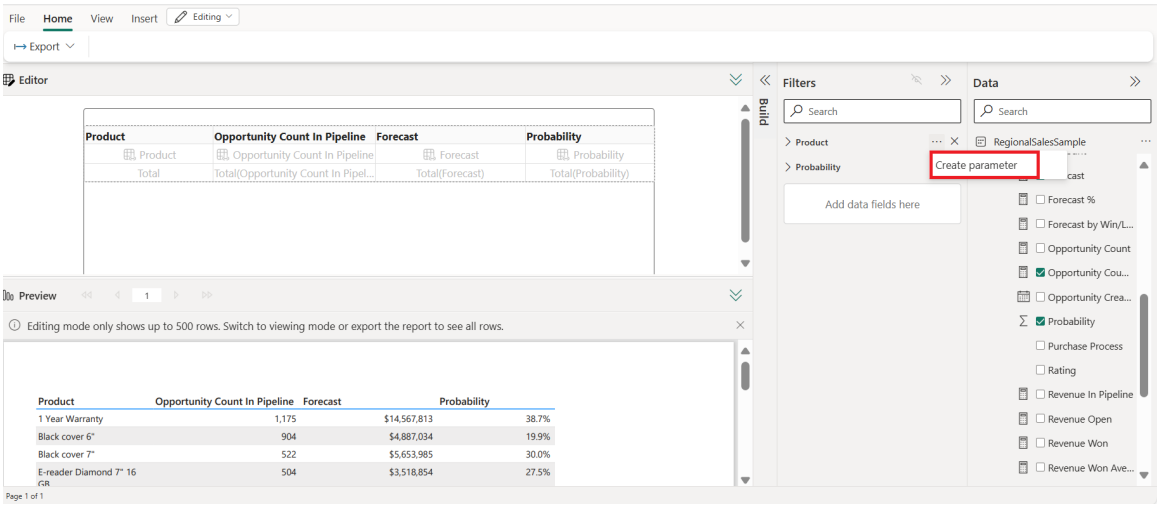
You can choose to add a header, footer, textbox or image. To add a footer, choose 'Insert' and select 'Footer'.



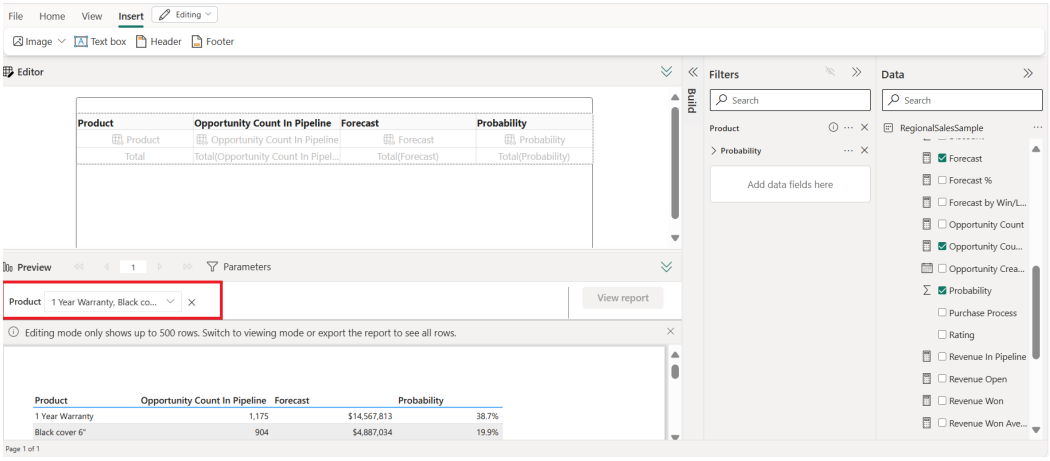
You can add a Text box, Image, page number and / or execution time. You may choose to display the footer and header on the first and last pages as well.



You can exit the footer by clicking out. Once you are in the body of the report, you can 'Create parameter'. By creating a parameter, you can create a report that requires the viewer of the report to enter one or more values to view the report.



When you 'Create parameter', you can see the parameter at the top of the 'Preview' portion of the screen. You may show / hide the parameter by clicking on the 'Parameters' on the Preview Ribbon.

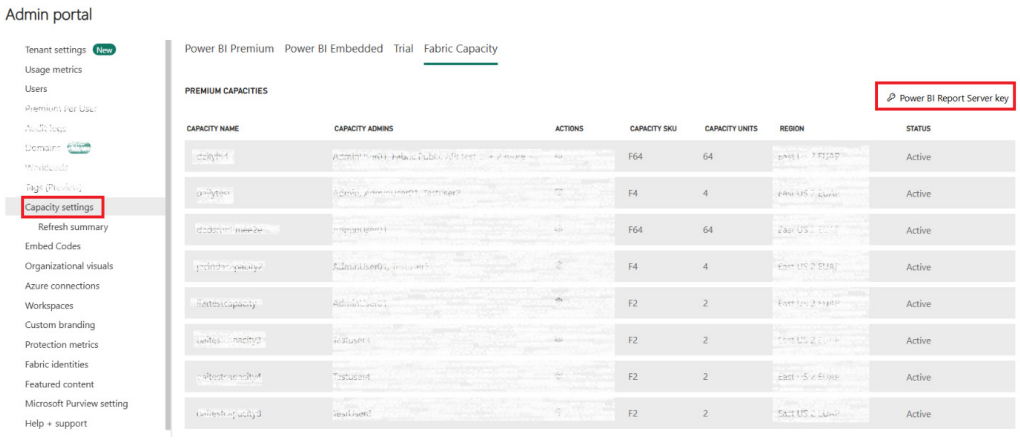


When you save the report, it is saved with the parameter defined and the viewer of the report must specify the parameters to view the report. You can now share the report with others.

This is a Preview feature and will not be available on Sovereign clouds until it is Generally Available. The update will be rolling out in the coming weeks.

Power BI Report Server key in Fabric Capacities

Power BI Report Server is now included with F64+ Reserved Instance purchases. It continues to be available with SQL Server Enterprise core licenses with software assurance. You can get the PBIRS key in the 'Fabric Capacity' tab under 'Capacity Settings' in the admin portal.



More next month.

New Features for Excel

This month sees you able to prompt Copilot in Excel to work with columns and formulae in Excel for web, Windows and Mac. But that's not all: in what appears to be a second successive "abridged" month, you are also now able to navigate and perform tasks more quickly with KeyTips in Excel for Mac (albeit Beta Channel users), plus view and open your most recently accessed Excel files from the Home screen on your iOS device (in Insiders).

We know we mention two other features elsewhere in this newsletter, but they aren't "formally announced" in the ***Excel Updates*** section of Microsoft's website, so it's a short but sweet summary this time out:

Excel for the web, Windows and Mac (but no cheese)

- Copilot in Excel: working with columns and formulae

Excel for Mac

- KeyTips (Insiders)

Excel for iOS

- Recent Files widgets (Insiders).

Let's get started.

Copilot in Excel: working with columns and formulae

Coming to Excel for the web, Excel for Windows and Excel for Mac, Copilot in Excel can now write additional formulae, including ones that work with text. The following example demonstrates Excel Copilot's ability to work with complex formulae with a dataset containing some noteworthy individuals.

Improvements made this month include:

- Organise data by splitting a single column into multiple columns
- Calculate age by extracting the years from a string of text.

Let's take a look.

ORGANISE DATA BY SPLITTING A SINGLE COLUMN INTO MULTIPLE COLUMNS

We have data that comes in this format:

Alan Turing (1912-1954, United Kingdom):

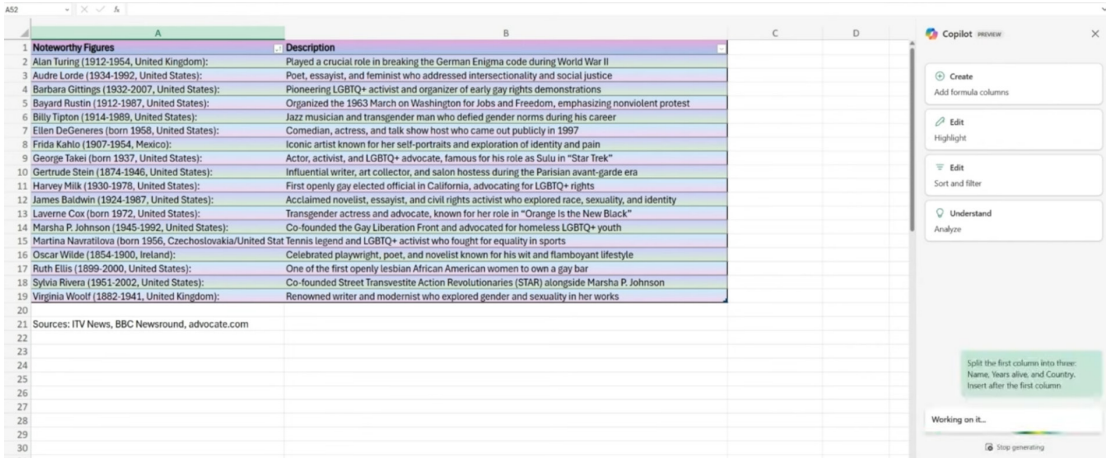
However, we would like to make the Table cleaner by splitting the data into multiple columns.

Alan Turing	1912-1954	United Kingdom

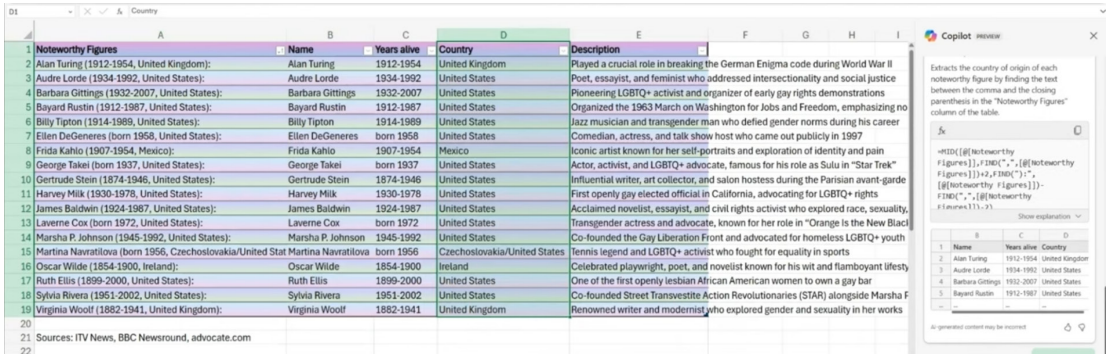
Here. the prompt used for Copilot in Excel was:

Split the first column into three: Name, Years alive, and Country. Insert after the first column.

Copilot is able to take natural language input, make an educated guess as to what part of the original column means 'Years alive' for example, and perform the task by generating a formula that works on the data supplied:



Et voila!



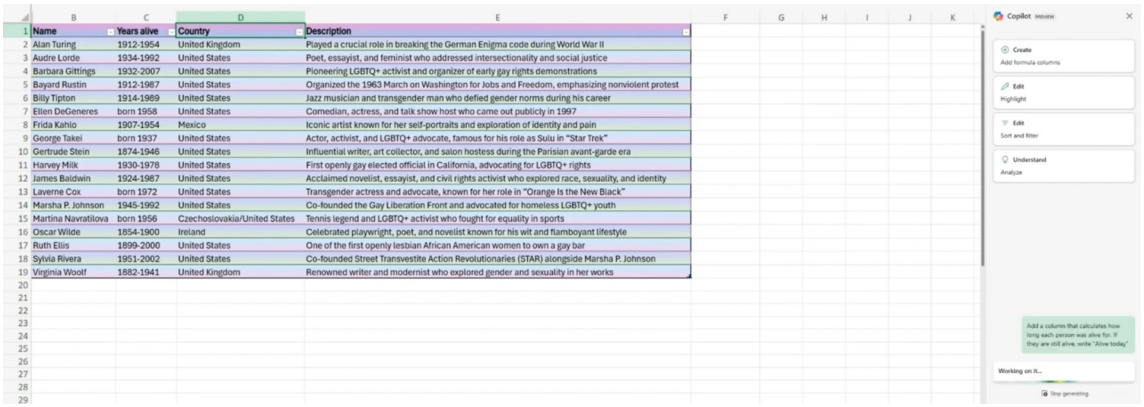
CALCULATE AGE BY EXTRACTING THE YEARS FROM A STRING OF TEXT

For this illustration, imagine you have the data as follows:

Alan Turing	1912-1954	United Kingdom
George Takei	born 1937	United States

In this instance, the intention is to get Copilot to add a column with the individual's age as follows:

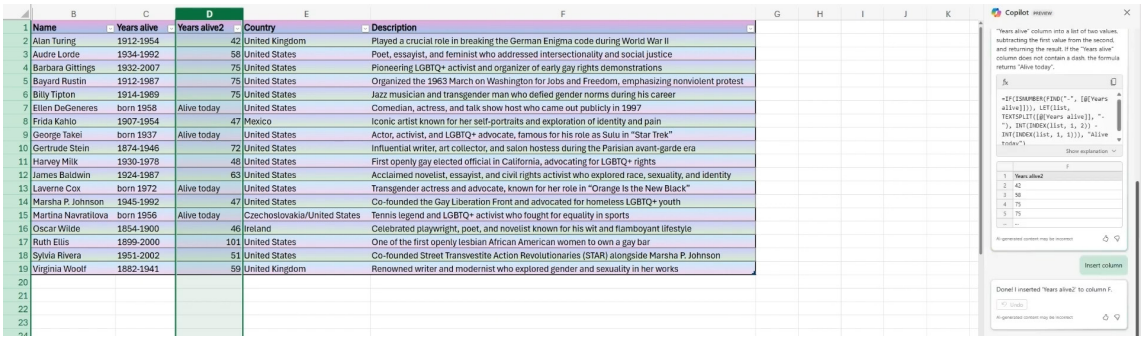
Alan Turing	1912-1954	42	United Kingdom
George Takei	born 1937	Alive today	United States



Name	Years alive	Country	Description
Alan Turing	1912-1954	United Kingdom	Played a crucial role in breaking the German Enigma code during World War II
Audre Lorde	1934-1992	United States	Poet, essayist, and feminist who addressed intersectionality and social justice
Barbara Gittings	1932-2007	United States	Pioneering LGBTQ+ activist and organizer of early gay rights demonstrations
Bayard Rustin	1912-1987	United States	Organized the 1963 March on Washington for Jobs and Freedom, emphasizing nonviolent protest
Billy Tipton	1914-1989	United States	Jazz musician and transgender man who defied gender norms during his career
Ellen DeGeneres	born 1958	United States	Comedian, actress, and talk show host who came out publicly in 1997
Frida Kahlo	1907-1954	Mexico	Iconic artist known for her self-portraits and exploration of identity and pain
George Takei	born 1937	United States	Actor, activist, and LGBTQ+ advocate, famous for his role as Sulu in "Star Trek"
Gertrude Stein	1874-1946	United States	Influential writer, art collector, and salon hostess during the Parisian avant-garde era
Harvey Milk	1930-1978	United States	First openly gay elected official in California, advocating for LGBTQ+ rights
James Baldwin	1924-1987	United States	Acclaimed novelist, essayist, and civil rights activist who explored race, sexuality, and identity
Laverne Cox	born 1972	United States	Transgender actress and advocate, known for her role in "Orange Is the New Black"
Marsha P. Johnson	1945-1992	United States	Co-founded the Gay Liberation Front and advocated for homeless LGBTQ+ youth
Martina Navratilova	born 1956	Czechoslovakia/United States	Tennis legend and LGBTQ+ activist who fought for equality in sports
Oscar Wilde	1854-1900	Ireland	Celebrated playwright, poet, and novelist known for his wit and flamboyant lifestyle
Ruth Ellis	1899-2000	United States	One of the first openly lesbian African American women to own a gay bar
Sylvia Rivera	1951-2002	United States	Co-founded Street Transvestite Action Revolutionaries (STAR) alongside Marsha P. Johnson
Virginia Woolf	1882-1941	United Kingdom	Renowned writer and modernist who explored gender and sexuality in her works

Here, the following prompt was used:

Add a column that calculates how long each person was alive for.
If they are still alive, write "Alive today".



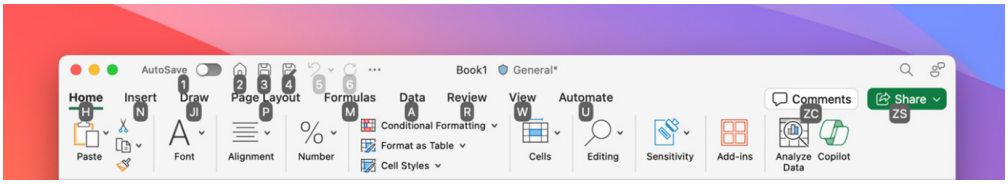
Name	Years alive	Years alive2	Country	Description
Alan Turing	1912-1954	42	United Kingdom	Played a crucial role in breaking the German Enigma code during World War II
Audre Lorde	1934-1992	58	United States	Poet, essayist, and feminist who addressed intersectionality and social justice
Barbara Gittings	1932-2007	75	United States	Pioneering LGBTQ+ activist and organizer of early gay rights demonstrations
Bayard Rustin	1912-1987	75	United States	Organized the 1963 March on Washington for Jobs and Freedom, emphasizing nonviolent protest
Billy Tipton	1914-1989	75	United States	Jazz musician and transgender man who defied gender norms during his career
Ellen DeGeneres	born 1958	Alive today	United States	Comedian, actress, and talk show host who came out publicly in 1997
Frida Kahlo	1907-1954	47	Mexico	Iconic artist known for her self-portraits and exploration of identity and pain
George Takei	born 1937	Alive today	United States	Actor, activist, and LGBTQ+ advocate, famous for his role as Sulu in "Star Trek"
Gertrude Stein	1874-1946	72	United States	Influential writer, art collector, and salon hostess during the Parisian avant-garde era
Harvey Milk	1930-1978	48	United States	First openly gay elected official in California, advocating for LGBTQ+ rights
James Baldwin	1924-1987	63	United States	Acclaimed novelist, essayist, and civil rights activist who explored race, sexuality, and identity
Laverne Cox	born 1972	Alive today	United States	Transgender actress and advocate, known for her role in "Orange Is the New Black"
Marsha P. Johnson	1945-1992	47	United States	Co-founded the Gay Liberation Front and advocated for homeless LGBTQ+ youth
Martina Navratilova	born 1956	Alive today	Czechoslovakia/United States	Tennis legend and LGBTQ+ activist who fought for equality in sports
Oscar Wilde	1854-1900	46	Ireland	Celebrated playwright, poet, and novelist known for his wit and flamboyant lifestyle
Ruth Ellis	1899-2000	101	United States	One of the first openly lesbian African American women to own a gay bar
Sylvia Rivera	1951-2002	51	United States	Co-founded Street Transvestite Action Revolutionaries (STAR) alongside Marsha P. Johnson
Virginia Woolf	1882-1941	59	United Kingdom	Renowned writer and modernist who explored gender and sexuality in her works

KeyTips

For Excel for Mac, KeyTips are keyboard shortcuts for items in the Ribbon menu in Microsoft applications. They provide a quick way to navigate and perform tasks with your keyboard: no clicking is required.

You might be familiar with KeyTips on a Windows PC, and they function the same way on a Mac.

To access them, press the Option (⌥) or **ALT** key. KeyTip shortcuts will appear over your Ribbon tabs, as well as the Quick Access Toolbar (QAT) and the buttons to the right of the Ribbon.



You may then find the command you wish to use and press the characters shown in the KeyTip over that command. Otherwise, to cancel an action and hide KeyTips, press the **ESC** or Option (⌥) keys.

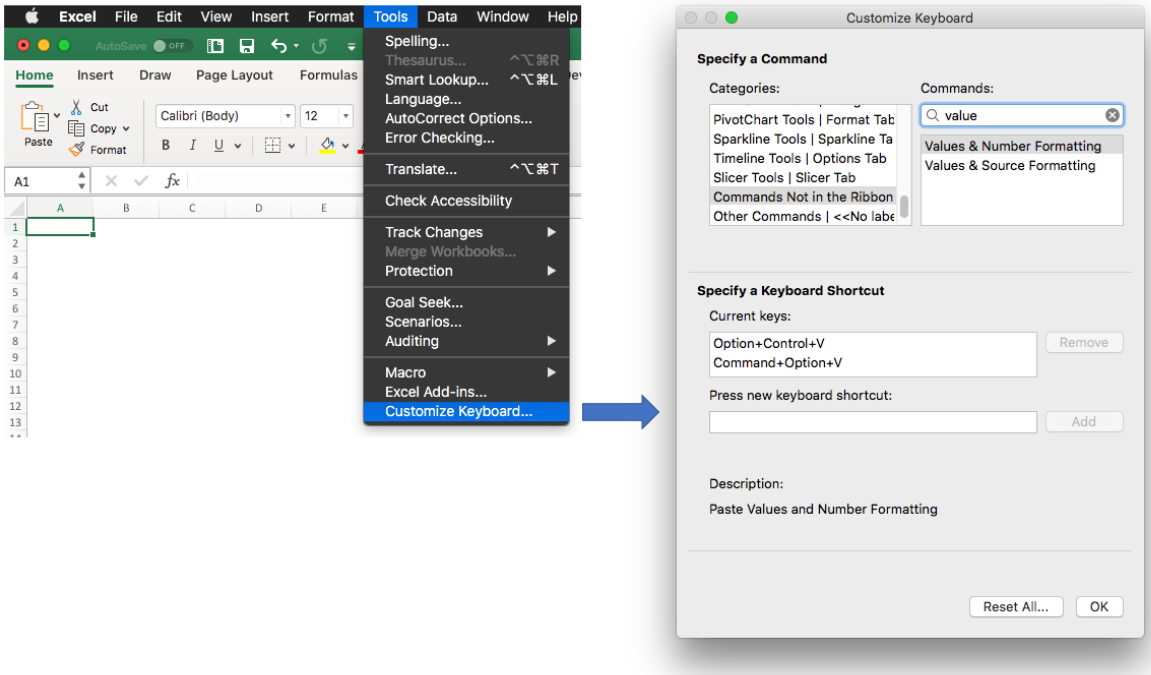
It should be noted that some KeyTips lead to additional KeyTips. For example, if the Home tab is active and you press **N**, the Insert tab appears along with KeyTips for items in that tab.

KeyTips are currently available only in English for Microsoft 365 Insiders but will be available in all languages upon general release. With regards to availability, KeyTips in Office for Mac are available to Beta Channel users of Word, Excel, PowerPoint and OneNote for Mac running Version 16.86 (Build 24052212) or later. The feature will become available to all users in the coming months.

Don't forget, you can create or change a keyboard shortcut in Excel by following these steps:

- Go to the Tools menu and choose 'Customize Keyboard'
- Pick the category of command you're trying to find, and then search or browse for the command in the list
- Select a command, press a key combination, and see if it's used already. If it's already in use, you may want to pick a different combination
- Then just click the 'Add' button to assign the key combination to the selected command.

In the example below, we show the category called 'Commands Not in the Ribbon' with a search for commands with the word "value", which filters to the commands 'Values & Number Formatting' and Values & Source Formatting'.



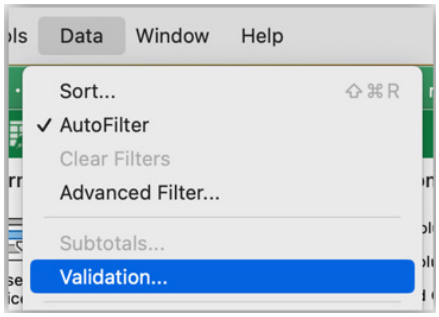
You should note that some commands may not be available. Generally, only commands that appear in the 'Ribbon Customization' dialog can be found in the 'Customize Keyboard' dialog. For example, there is no command that allows you to set a cell format to use a particular font. Some keyboard shortcuts may be used by your Mac and you may not be able to use these key combinations, even though you can assign them in the dialog.

You can read more details here: Create a Custom Keyboard Shortcut.

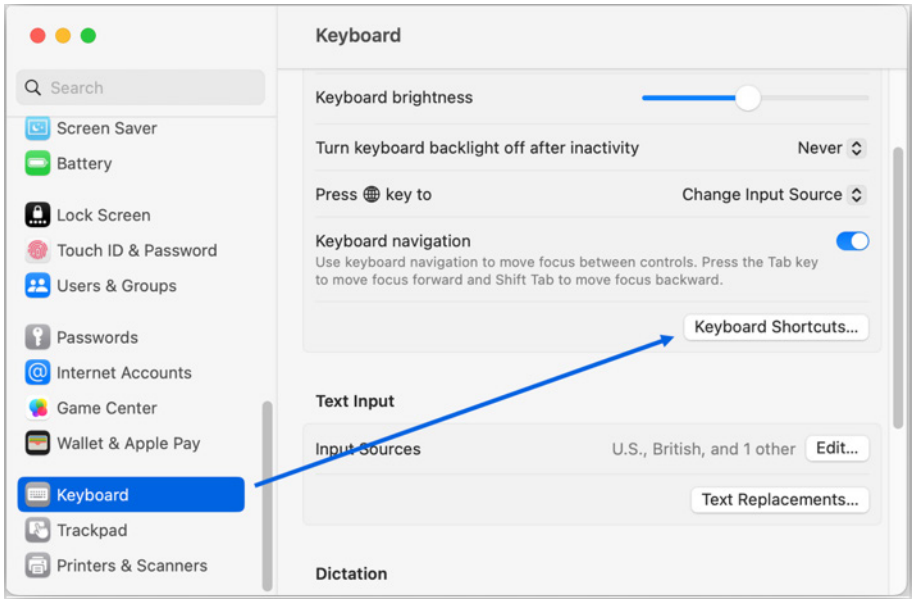
Another way to create a customised keyboard shortcut is to use the Mac Preferences. This allows you to set a keyboard shortcut for any app, but only for commands that appear in one of that app's menus.

Just follow the simple steps below:

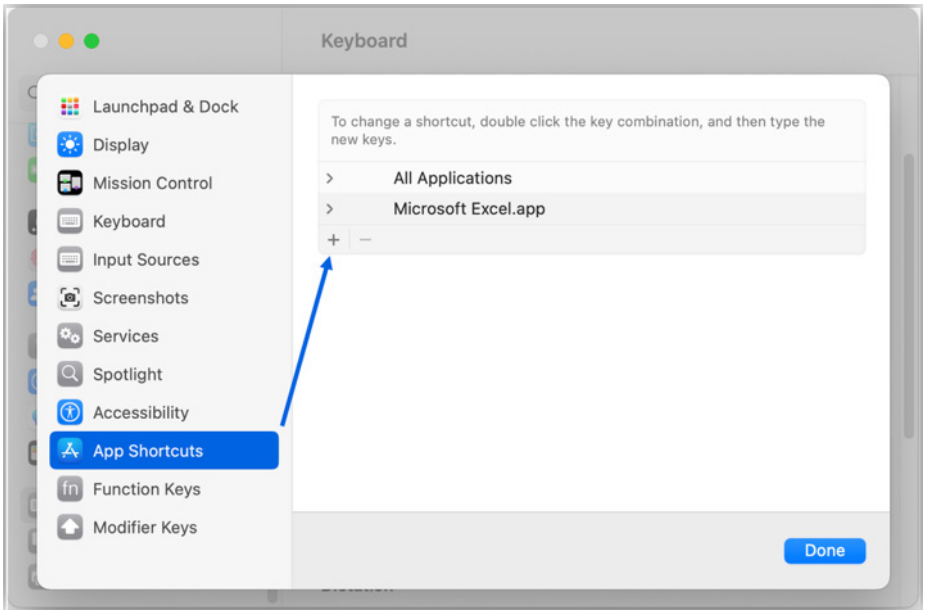
- Look for a menu command that you want to use a keyboard shortcut to activate. For example, you can go to the Data menu in Excel and you'll see 'Validation…', which will open the 'Data Validation' dialog:



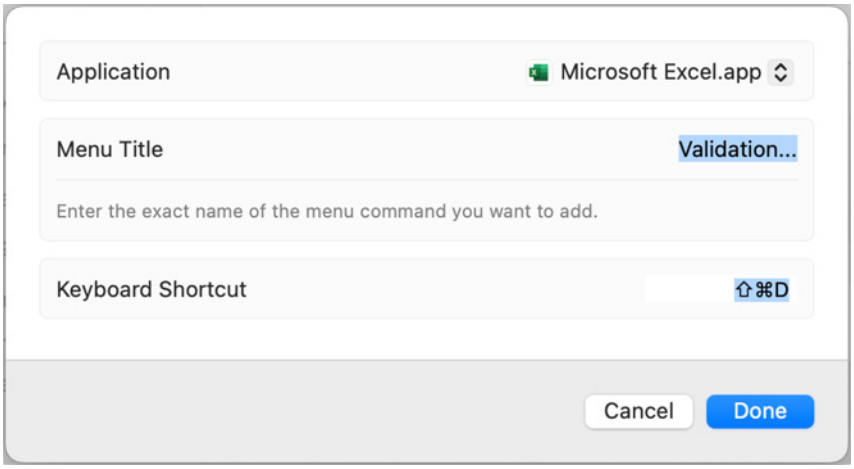
- Go to **macOS -> Preferences -> Keyboard -> Keyboard Shortcuts -> App Shortcuts**



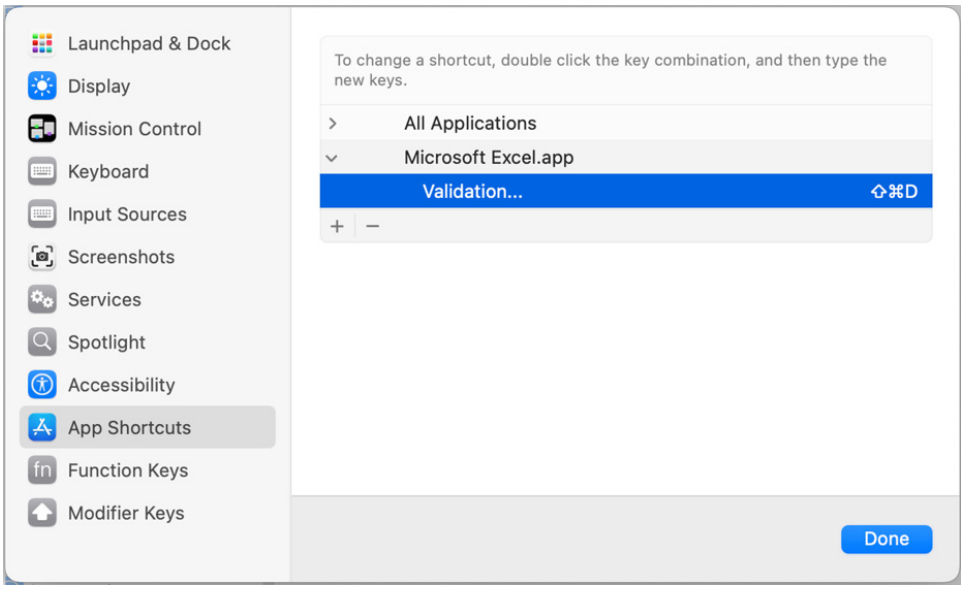
- Click the **+** button to add a new shortcut



Select 'Microsoft Excel.app' from the list of applications



- In the Menu Title field, type the name of the menu item exactly as it appears in Excel. For example, type 'Validation…' to create a shortcut that will simulate opening the Data menu and pressing 'Validation…', which opens the 'Data Validation' dialog
- Press a key combination. It's a good idea to choose a key combination that's not already used to do something else
- Now go back to Excel and try out the new shortcut you just created!



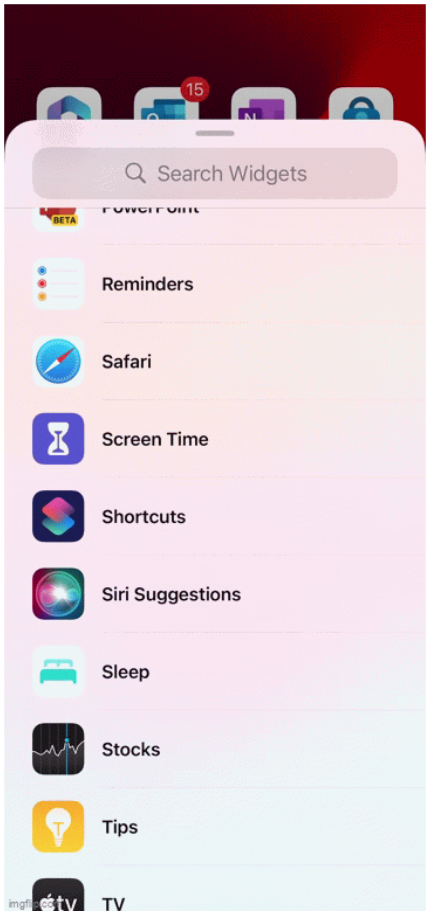
See Apple's help article for more information: Use macOS keyboard shortcuts - Apple Support.

Recent Files widgets

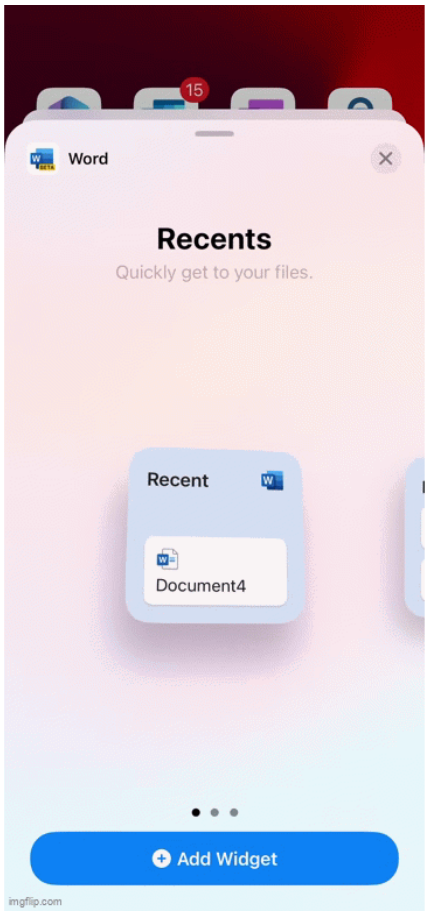
You can add Recent Files widgets for Word, Excel and PowerPoint directly to your iPhone or iPad home screen. The widgets allow you to both view and open your most recently accessed files in that app from the home screen on your device. It works as follows:

- On the device Home screen, press and hold an empty space and notice the **+** icon that appears at the top left corner of the screen

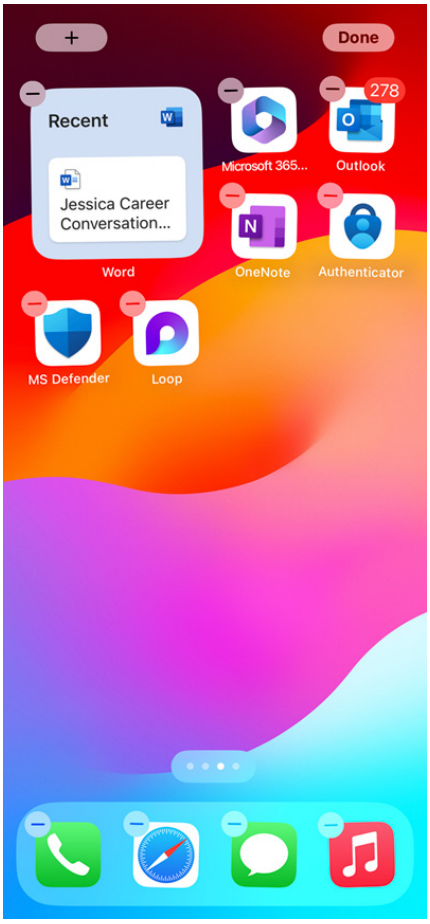
- Tap the + icon, scroll through or search the list of widgets, and tap to select the app widget you want



- A 'Recents' *(more great grammar – Ed.)* screen will appear. Scroll through the icons and pick the size of the widget you want, then tap 'Add Widget'



- Tap the widget to open it and then select the File card that you want; the file will open in the app.



The Recent Files widget offers sizes to choose from on the iPad, ranging from small to extra-large. There are three sizes to choose from on the iPhone: small, medium and large. If you would like to open the app's home page instead of a recent file, tap anywhere on the widget outside of the file cards.

This feature is rolling out to Microsoft 365 Insiders running iOS Version 2.85 (Build 24042818) or later.

Until next month.

The A to Z of Excel Functions: NOT

Liam, will you make a stupid pun here? Sorry, I'm a frayed knot.

NOT is one of Excel's logic functions and reverses the logic of its argument (oh no it doesn't, oh yes it does, … *[please stop – Ed.]*).

The syntax for **NOT** is as follows:

NOT(logical)

where:

- **logical:** the condition whose logic you wish to reverse.

It should be noted that:

- the arguments must evaluate to logical values, such as TRUE or FALSE, or the arguments must be arrays or references that contain logical values
- if an array or reference argument contains text or empty cells, those values are ignored
- blank cells are treated as FALSE
- all numerical values except zero [0] are treated as TRUE; zero is considered FALSE
- if the specified argument contains no logical values, the **NOT** function returns the *#VALUE!* error value.

Please see our example below:

	A	B	C
1	Condition 1		*blank cell*
2	Condition 2	0	
3	Condition 3	(4.7)	
4	Condition 4	TRUE	
5	Condition 5	text	
6			
7			
8	**Description**	**Results**	**Formula**
9	Condition 1	**TRUE**	***=NOT(B1)***
10	Condition 2	**TRUE**	***=NOT(B2)***
11	Condition 3	**FALSE**	***=NOT(B3)***
12	Condition 4	**FALSE**	***=NOT(B4)***
13	Condition 5	**#VALUE!**	***=NOT(B5)***

The A to Z of Excel Functions: NOW

Dates count from 1 January 1900, *e.g.* what is known as serial number one [1] is 1 January 1900, serial number two [2] is 2 January 1900, under the default Excel for Windows settings. The **NOW** function returns the serial number of the current date and time (the date is an integer at precisely midnight). If the cell format were General before the function was entered, Excel changes the cell format so that it matches the date and time format of your regional settings. You can change the date and time format for the cell by using the commands in the Number group of the Home tab on the Ribbon.

This function is useful when you need to display the current date and time on a worksheet or calculate a value based on the current date and time, and have that value updated each time you open the worksheet.

However, should the **NOW** function not update cell values when you expect it to, you might need to change the relevant settings that control when the workbook or worksheet recalculates. These settings may be changed in **File -> Options -> Formulas -> Calculation options**.

The **NOW** function has the following syntax:

NOW(number1, [number2], ...)

The **NOW** function takes no prisoners or arguments.

It should be noted that:

- Excel stores dates as sequential serial numbers so that they can be used in calculations. By default, January 1, 1900 is serial number 1, and January 1, 2008 is serial number 39448 because it is 39,447 days after January 1, 1900
- numbers to the right of the decimal point in the serial number represent the time; numbers to the left represent the date. For example, the serial number 0.5 represents the time 12:00 noon
- the results of the **NOW** function change only when the worksheet is calculated or when a macro that contains the function is run. It is not updated continuously.

Please see our final example for this month below:

	A	B	C
1	Description	Results	Formula
2	Returns the current dat and time	13-07-22 15:25	=NOW()
3	Returns the date and time 12 hours (0.5 days) ago	13-07-22 3:25	=NOW() - 0.5
4	Returns the time a week from now (seven days in the future)	20-07-22 15:25	=NOW() + 7
5	Type mismatch: numerical and text values may not be added	#VALUE!	=NOW() + "dog"

More Excel Functions next month.

Beat the Boredom Suggested Solution

The challenge this month was to extract a name from a text string that contained special characters.

The Challenge

There may be a time when you are using Excel that you wish to extract a name of a person from a churn of endless text strings that has all sorts of characters that you have never seen or used before. Therefore, this month's challenge was to write a **formula** to extract the name of a person from any text string. The result should look similar to the following:

	Text	Solution
1.	Sara's Address	Sara
2.	Brian T's Job title	Brian T
3.	Karina's Date of Birth	Karina

As always, there were some requirements:

- the formula needed to be within just one column (no "helper" cells)
- this was a formula challenge; no Power Query / Get & Transform or VBA!
- the formula should be dynamic enough when a similar text string was added.

Suggested Solution

Before we discuss the solution, there are several complicating factors here. Let's go through them.

Problem 1: The Unremovable White Space

When tackling this problem, we might rely on some functions like **TRIM** and **CLEAN** to clear the text. The **TRIM** function helps us strip extra white space from the text leaving only a single space between words and no space characters at the start or end of the text. The **CLEAN** function helps us remove all nonprintable characters from text. Thus, using the **TRIM** and **CLEAN** functions might help us remove all unwanted white spaces and nonprintable characters:

=CLEAN(TRIM(Text[@Text]))

In the formula above, **Text** is the name of the table. Therefore, **Text[@Text]** specifies one row of column **Text**, where the above formula is located. However, these two functions do not appear to work with these text strings no matter how many times these functions apply: the white space is still there. This is because there are some special invisible characters that the **TRIM** and **CLEAN** function cannot remove. Hence, using **TRIM** and **CLEAN** functions for this challenge will not solve our problem.

Problem 2: The Unfindable Apostrophe

The **FIND** function is quite useful here to address the challenge. It may help us look for the position of a character within the text string. Moreover, the target text we need to extract is between the white space and the apostrophe, so **FIND** can give us the location of those items which can, later on, be used to extract the target text.

However, you might run into the problem that the **FIND** function results in the *#VALUE!* error when you try to search for the apostrophe. This is because there is a "weird" apostrophe within the text string which is different from the normal apostrophe on the keyboard. Therefore, the **FIND** function must be tweaked to be able to search for that apostrophe in the text string.

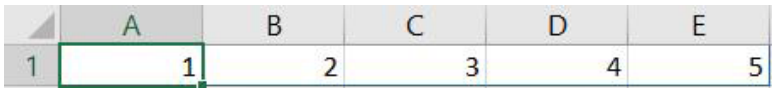
Brainstorming

To address the unremovable white space and unfindable apostrophe problems, we will need a quick inspection of the text string to fully understand it. Therefore, we will transform each letter of the text string into individual cells in the spreadsheet.

For Microsoft Excel 365 and online versions, we may use Dynamic Arrays with the following formula:

=MID(Text[@Text], SEQUENCE(1, LEN(Text[@Text])),1)

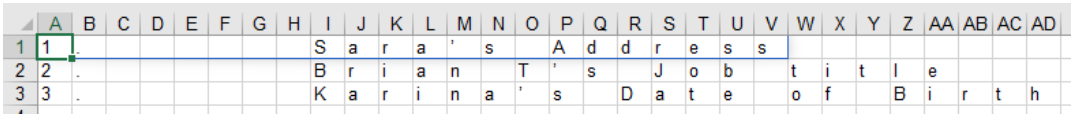
SEQUENCE(1, LEN(Text[@Text])) will help us create a horizontal list of the consecutive text string from one [1] to the last number which is equal to the length of the string. For example:



	A	B	C	D	E
1	1	2	3	4	5

The **MID** function will then extract each character of a string with the starting point one by one, equal to the number list created by **SEQUENCE** above.

After you copy down the formula, the result should look like this:



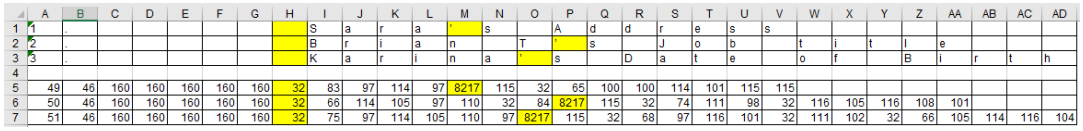
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1	1	.							S	a	r	a	'	s		A	d	d	r	e	s	s								
2	2	.							B	r	i	a	n		T	'	s		J	o	b		t	i	t	l	e			
3	3	.							K	a	r	i	n	a	'	s		D	a	t	e		o	f		B	i	r	t	h

We can see that there are many invisible characters and white space at the beginning of the strings.

Next, we will need to identify what those invisible characters are. We can use **UNICODE** and **UNICHAR** functions to our Dynamic Arrays above. **UNICODE** allows us to return numeric code for the first character in a text string. **UNICHAR** translates that code back into a character. (You can also use **CODE** and **CHAR** functions here, but we suggest using **UNICODE** and **UNICHAR** as some special characters are not in the database of **CODE** and **CHAR**). The formula to return numeric code is as follows:

=UNICODE(MID(Text[@Text], SEQUENCE(1, LEN(Text[@Text])),1))

The result is as follows:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1	1								S	a	r	a	'	s		A	d	d	r	e	s	s								
2	2								B	r	i	a	n		T	'	s		J	o	b		t	i	t	l	e			
3	3	.							K	a	r	i	n	a	'	s		D	a	t	e		o	f		B	i	r	t	h
4																														
5	49	46	160	160	160	160	160	32	83	97	114	97	8217	115	32	65	100	100	114	101	115	115								
6	50	46	160	160	160	160	160	32	66	114	105	97	110	32	84	8217	115	32	74	111	98	32	116	105	116	108	101			
7	51	46	160	160	160	160	160	32	75	97	114	105	110	97	8217	115	32	68	97	116	101	32	111	102	32	66	105	114	116	104

Upon inspection, we can see that the codes of these "invisible characters" are 160 and 32. Unicode 32 is our normal "Space" generated by pressing the spacebar on the keyboard, while 160 is the "No-Break Space", generated by pressing **ALT + 0160**.

For the normal "Apostrophe" that we use on the keyboard, they have the Unicode of 39, while the apostrophe used in the text string is the "Right Single Quotation Mark" which has the Unicode 8217.

Back to the Suggested Solution

As we can see here, all 3-text strings have 32 in front of the target name and 8217 at the end of the target name. So, we can write a **FIND** function to find their position of them.

This situation is perfect to extract the key data from the text string as **FIND** function will give out the position of the first appearance of the letter. So, if we use **FIND** function on the "No-Break Space" character that has Unicode 160, it will result in three [3] which is not a desirable starting position.

Hence, the starting position of the target text should be:

=FIND(UNICHAR(32), Text[@Text])+1

and the number of characters we extract is:

=FIND(UNICHAR(8217), Text[@Text String])-FIND(UNICHAR(32), Text[@Text])-1

From here we can use the **MID** function to extract the target text as follows:

=MID(Text[@Text], FIND(UNICHAR(32), Text[@Text]) +1, FIND(UNICHAR(8217), Text[@Text])-FIND(UNICHAR (32), Text[@Text])-1)

More next month.

Upcoming SumProduct Training Courses

Location	Course	Course Date	Local TIme	UTC	Duration
London UK	Power Pivot, Power Query and Power BI	8 August 2024 - 9 August 2024	09:00 - 17:00 BST	08 Aug 2024 08:00 UTC - 09 Aug 2024 16:00 UTC	2 Days
London UK	Financial Modelling	12 August 2024 - 13 August 2024	09:00 - 17:00 BST	12 Aug 2024 08:00 UTC - 13 Aug 2024 16:00 UTC	2 Days
Sydney Australia	ChatGPT	20 August 2024 - 21 August 2024	09:00 - 17:00 AEST	19 Aug 2024 23:00 UTC - 21 Aug 2024 07:00 UTC	2 Days
Melbourne Australia	Excel Tips and Tricks	28 August 2024	09:00 - 17:00 AEST	27 Aug 2024 23:00 UTC - 28 Aug 2024 07:00 UTC	1 Day
Sydney Australia	Excel Tips and Tricks	1 October 2024	09:00 - 17:00 AEST	31 Sep 2024 23:00 UTC - 01 Oct 2024 07:00 UTC	1 Day
Melbourne Australia	Financial Modelling	14 October 2024 - 15 October 2024	09:00 - 17:00 AEDT	13 Oct 2024 22:00 UTC - 15 Oct 2024 06:00 UTC	2 Days
London UK	ChatGPT	29 October 2024 - 30 October 2024	09:00 - 17:00 GMT	29 Oct 2024 09:00 UTC - 30 Oct 2024 17:00 UTC	2 Days
Melbourne Australia	Power Pivot, Power Query and Power BI	11 November 2024 - 12 November 2024	09:00 - 17:00 AEDT	10 Nov 2024 22:00 UTC - 12 Nov 2024 06:00 UTC	2 Days

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. We need to get a **SHIFT** on with those pesky function keys this month:

Keystroke	What it does
SHIFT + F1	What is... (Help)
SHIFT + F2	Insert / edit comment
SHIFT + F3	Function wizard
SHIFT + F4	Find next (from most recent Search)
SHIFT + F5	'Find' dialog
SHIFT + F6	Previous pane
SHIFT + F7	Thesaurus
SHIFT + F8	Add to Selection Mode
SHIFT + F9	Calculation sheet
SHIFT + F10	Activate context menus (right-click)
SHIFT + F11	Insert new worksheet
SHIFT + F12	Save

There are *c.*550 keyboard shortcuts in Excel. For a comprehensive list, please download our Excel file at http://www.sumproduct.com/thought/keyboard-shortcuts. Also, check out our new daily **Excel Tip of the Day** feature on the www.sumproduct.com homepage.

Our Services

We have undertaken a vast array of assignments over the years, including:

- **Business planning**
- **Building three-way integrated financial statement projections**
- **Independent expert reviews**
- **Key driver analysis**
- **Model reviews / audits for internal and external purposes**
- **M&A work**
- **Model scoping**
- **Power BI, Power Query & Power Pivot**
- **Project finance**
- **Real options analysis**
- **Refinancing / restructuring**
- **Strategic modelling**
- **Valuations**
- **Working capital management**

If you require modelling assistance of any kind, please do not hesitate to contact us at contact@sumproduct.com.

Link to Others

These newsletters are not intended to be closely guarded secrets. Please feel free to forward this newsletter to anyone you think might be interested in converting to "the SumProduct way".

If you have received a forwarded newsletter and would like to receive future editions automatically, please subscribe by completing our newsletter registration process found at the foot of any www.sumproduct.com web page.

Any Questions?

If you have any tips, comments or queries for future newsletters, we'd be delighted to hear from you. Please drop us a line at newsletter@sumproduct.com.

Training

SumProduct offers a wide range of training courses, aimed at finance professionals and budding Excel experts. Courses include Excel Tricks & Tips, Financial Modelling 101, Introduction to Forecasting and M&A Modelling.

Check out our more popular courses in our training brochure:

Drop us a line at training@sumproduct.com for a copy of the brochure or download it directly from www.sumproduct.com/training.

Sydney Address: SumProduct Pty Ltd, Suite 803, Level 8, 276 Pitt Street, Sydney NSW 2000
New York Address: SumProduct Pty Ltd, 48 Wall Street, New York, NY, USA 10005
London Address: SumProduct Pty Ltd, Office 7, 3537 Ludgate Hill, London, EC4M 7JN, UK
Melbourne Address: SumProduct Pty Ltd, Ground Floor, 470 St Kilda Road, Melbourne, VIC 3004
Registered Address: SumProduct Pty Ltd, Level 14, 440 Collins Street, Melbourne, VIC 3000

contact@sumproduct.com
www.sumproduct.com
+61 3 9020 2071