Sum Product

NEWSLETTER #121 - December 2022

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

Eight conferences, 17 MVP awards, one Lifetime Achievement Award, six books, 1,284 articles, 837 keyboard shortcuts, 117 readers' questions and a whopping 2936 pages – we turn 10!!

I am feeling **TEN**-acious this month – except the time seems to have flown by, rather than us "limpet" to here... Sorry, I **TEN**d to make bad jokes on such anniversaries – it's best you pay at**TEN**tion... *(please stop – Ed.)*

Sadly, by our publishing deadline day, no Power BI updates were announced, but there is still more than enough to keep you occupied in this landmark birthday issue. We get all reflective with Best Practice considerations, show how to avoid circularity on interest calculations and revisit some readers' questions.

But then we also add in the staple diet of another Beat the Boredom Challenge, together with Charts & Dashboards, Visual Basics, Power Pivot Principles, Power Query Pointers and Excel Updates. You can find out why the first three letters of "irrelevant" are **IRR** in the A to Z of Excel functions series and we finish off the celebrations with yet more **CTRL + SHIFT** keyboard shortcuts.

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

Liam Bastick, Managing Director, SumProduct

Happy Birthday Dear Newsletter!

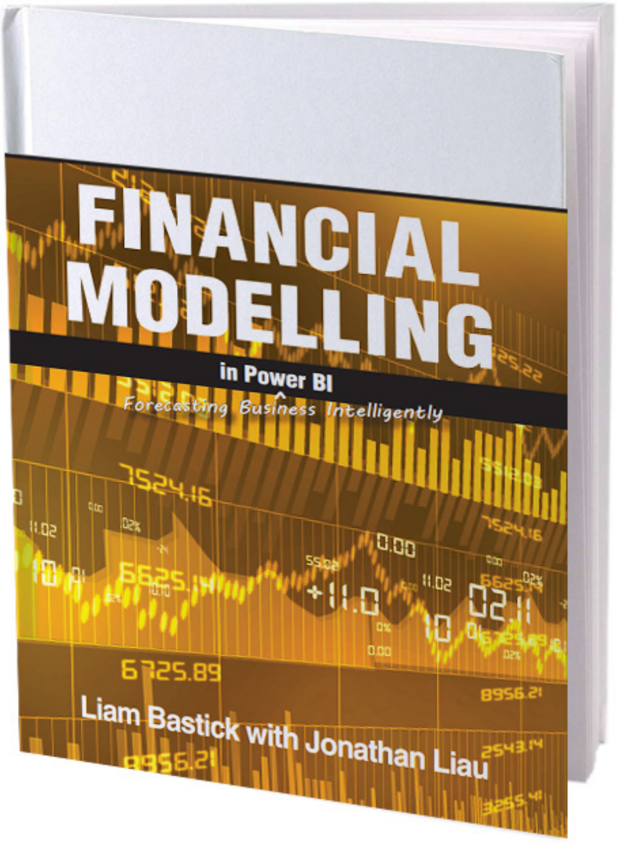
We turned 10! It won't be long now before this newsletter becomes a moody teenager...

A lot has happened in the past 10 years to both the newsletter and to all of us here at SumProduct.

Arguably Our Proudest Achievement

So what's been our crowning glory in the past 10 years? You may or may not be familiar with the first two books in this series, *An Introduction to Financial Modelling* and *Continuing Financial Modelling*, but we thought we peaked with *Financial Modelling in Power BI* (third time lucky, I suppose!). From covering how to model in Excel, we moved on to all things visual.

Power BI – Microsoft's self-service business intelligence and data visualisation tool – has become both more versatile and more popular as time goes by. Today's analysts need to use the best, most powerful and flexible analytical software for their work. More and more of us are realising Power BI fits the bill. We would have been foolish to stand still.


FINANCIAL MODELLING
in Power BI
Forecasting Business Intelligently
Liam Bastick with Jonathan Liau

This recently released book addresses the way forward for financial modelling. Many working in banking and finance are creating their financial models in Excel and then importing them into Power BI for graphical interpretation and further analysis. That's not what we do here: we went and built the model *in Power BI*.

If you can build the calculations for financial statements in Power BI, you can produce statements by product, by customer, by geography, by … Get the picture? The limitation will be restricted to the granularity of the underlying data. If you build the statements elsewhere and import them in, how do you propose drilling down to that level of detail?

Newsflash: you can't.

Whilst there is nothing wrong with building financial models in Excel or else download them from some third party management information system, this represents the next step. It's evolution. You can carry on building interactive spreadsheets if you wish. But this will take you to the next level in a thought provoking manner, and give you an advantage over your peers when it comes to financial modelling, analysis and forecasting.

This isn't meant to be a sales pitch (hopefully, you know us better than this now). We're very excited to release this ground-breaking book. You can order your copy now from Amazon or directly from Holy Macro books.

One Thing You Should Know…

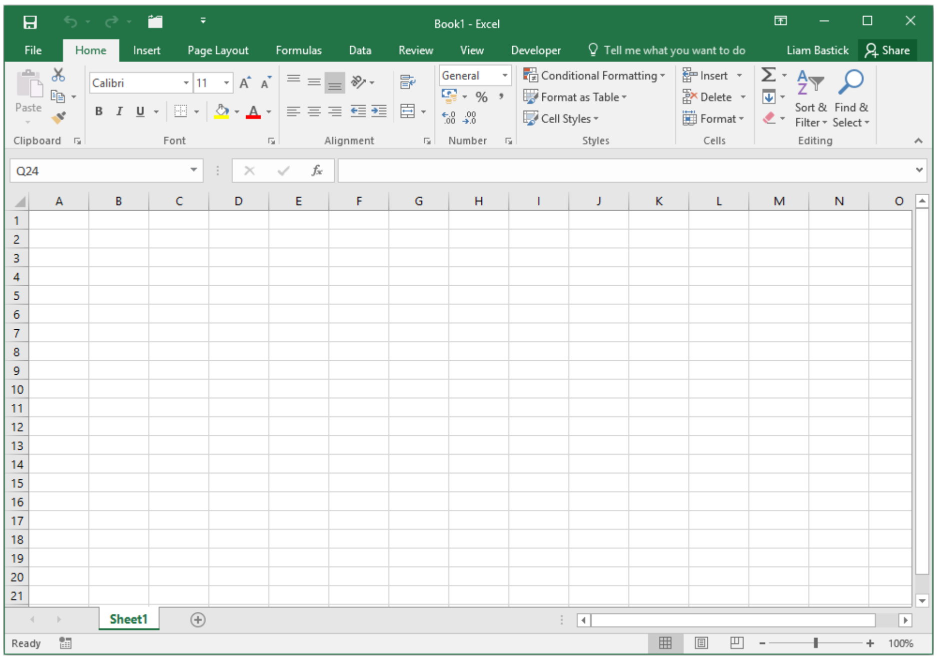
In the 121 newsletters we've put together in the past 10 years, there is certainly one point to get across – **CraFT**. We have written about it from time to time in these weighty tomes, but now seems as good a time as ever to revisit the "Best Practice" principles of building a spreadsheet.

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. "Best Practice" should be considered a proper noun to reflect the idea that a good model has four key attributes we call **CRaFT**:

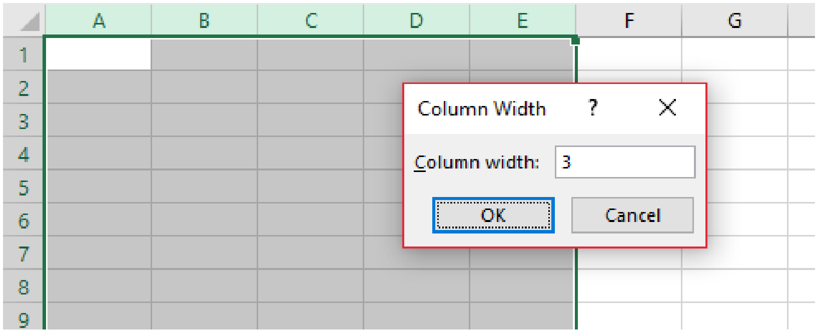
- **C**onsistency
- **R**obustness
- **F**lexibility
- **T**ransparency.

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



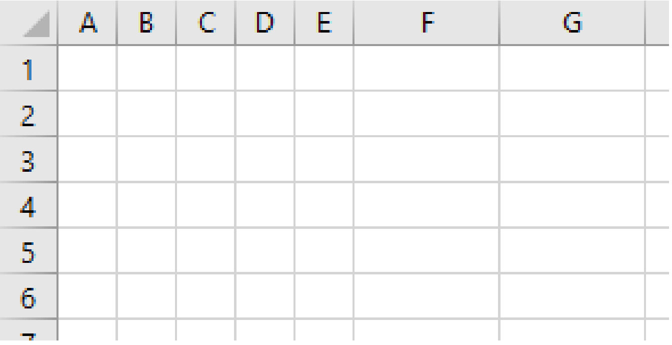
Building an effective communication tool by laying out an appropriate financial model is simple: it's all about designing and scoping. The problem is, we are all time poor in today's business environment with perpetual pressure on producing results more and more quickly. Getting a layout structure won't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

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



Column width: 3

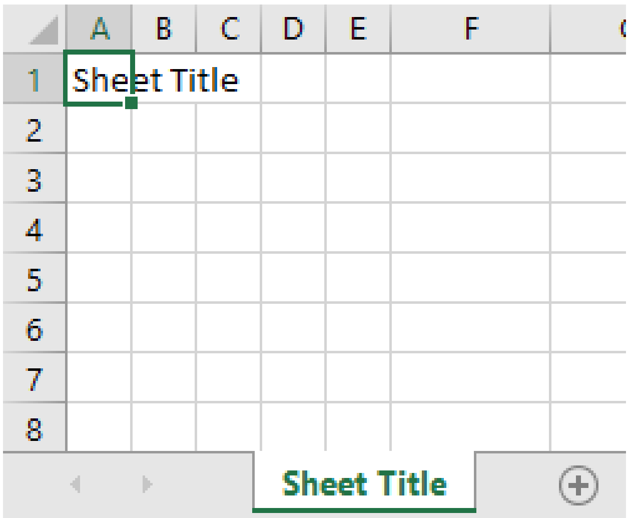
It may not be clear why we choose to do this, but read on (hey, I need to keep you on the edge of your seat!). We'll choose a width of 3 as this effectively makes the cells in these columns square.



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 (explained later)
2. Be consistent, both with the widths of the columns narrowed here and with other worksheets within the same workbook (again, to be explained shortly).

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:

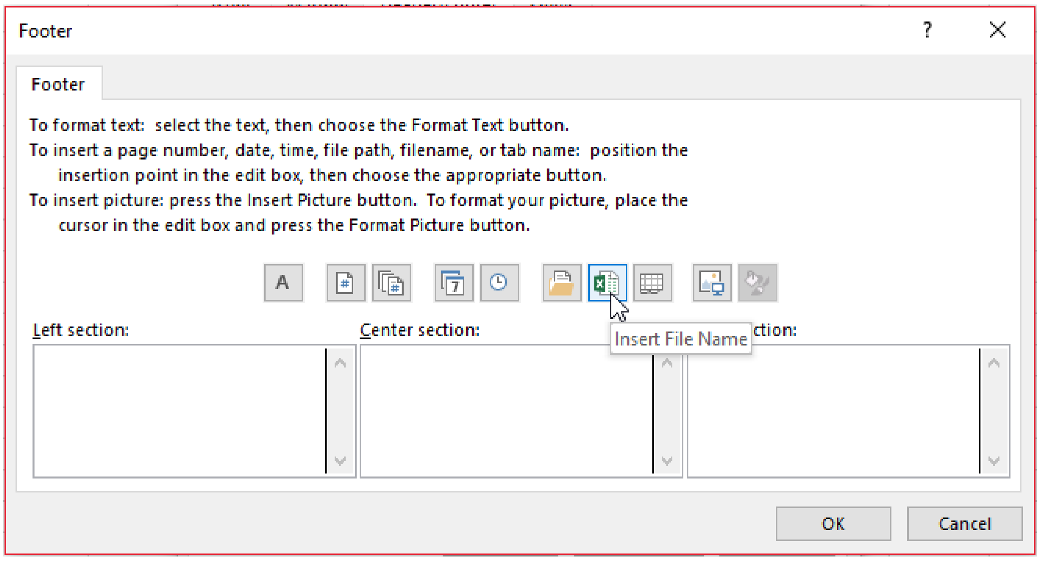


	A	B	C	D	E	F
1	Sheet Title					
2						
3						
4						
5						
6						
7						
8						

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**, we 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, we'll place it here rather than in the header or footer of each worksheet instead (but you may do both):



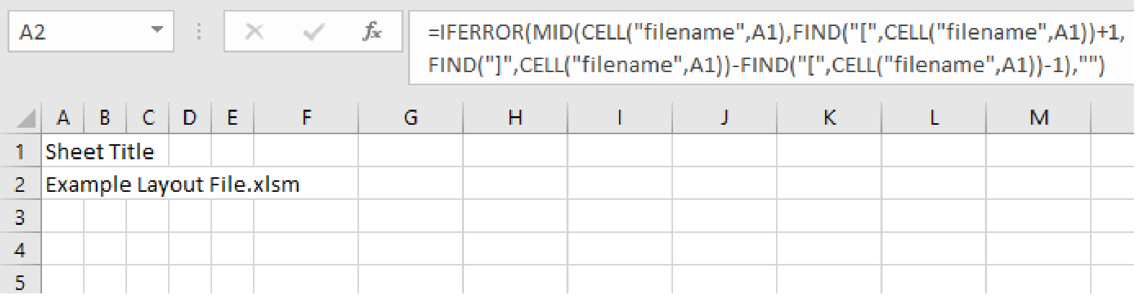
Footer

To format text: select the text, then choose the Format Text button.
To insert a page number, date, time, file path, filename, or tab name: position the insertion point in the edit box, then choose the appropriate button.
To insert picture: press the Insert Picture button. To format your picture, place the cursor in the edit box and press the Format Picture button.

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 we 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),"")

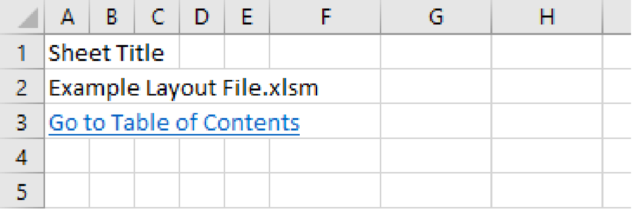


A2

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

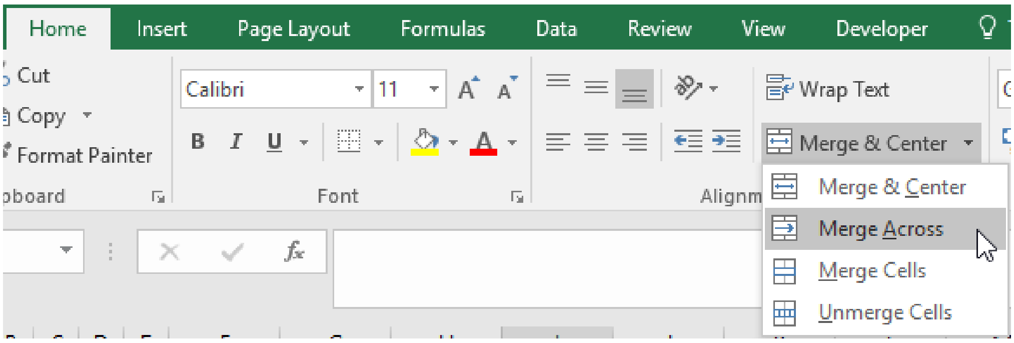
	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Sheet Title												
2	Example Layout File.xlsm												
3													
4													
5													

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. We recommend incorporating a central navigation page – a "Table of Contents" – which allows the end user to traverse the workbook quickly and easily.

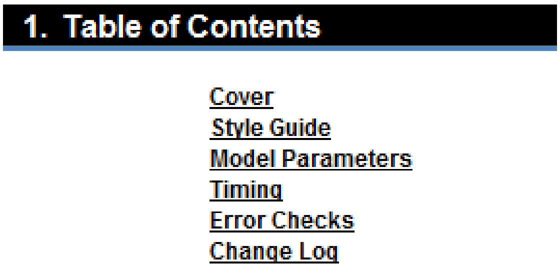


	A	B	C	D	E	F	G	H
1	Sheet Title							
2	Example Layout File.xlsm							
3	Go to Table of Contents							
4								
5								

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:

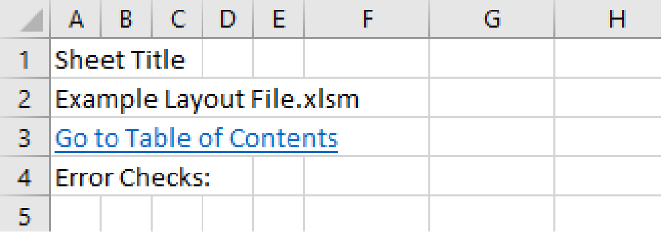


1. Table of Contents

Cover
Style Guide
Model Parameters
Timing
Error Checks
Change Log

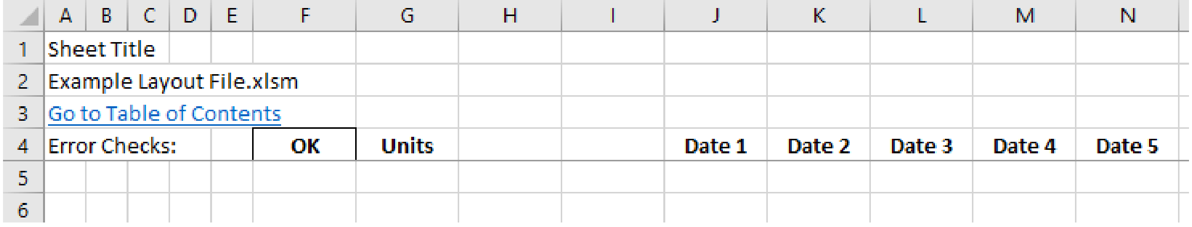
The hyperlink should link to cell **A1** *(say)* of that worksheet and that cell should have a range name such **HL_TOC**. A range name is essential in order to avoid a broken link should someone rename the destination worksheet and '**HL**' simply denotes that the cell is used as the destination for 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**:



	A	B	C	D	E	F	G	H
1	Sheet Title							
2	Example Layout File.xlsm							
3	Go to Table of Contents							
4	Error Checks:							
5								

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:

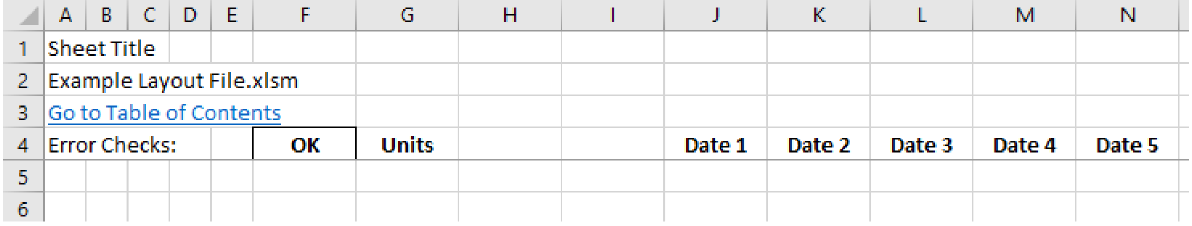


	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Sheet Title													
2	Example Layout File.xlsm													
3	Go to Table of Contents													
4	Error Checks:					**OK**	**Units**			**Date 1**	**Date 2**	**Date 3**	**Date 4**	**Date 5**
5														
6														

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".

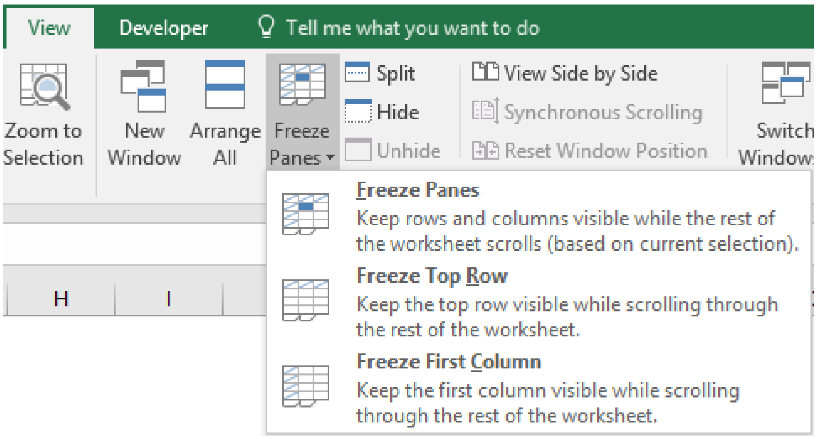
Cells **J4:N4** contain the date headings. The dates should be periodic (*e.g.* monthly, quarterly, annually) and should always start and end in the same columns (and rows) on each forecast worksheet. That is not always possible: sometimes, you require some of you model to be annually forecast and other 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:



	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														

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:



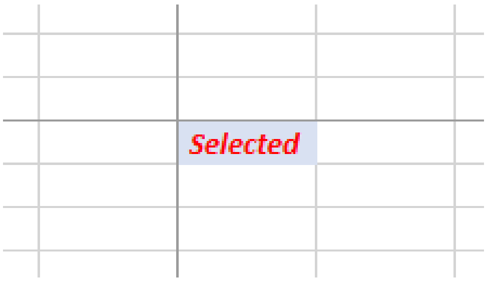
Freeze Panes
Keep rows and columns visible while the rest of the worksheet scrolls (based on current selection).

Freeze Top Row
Keep the top row visible while scrolling through the rest of the worksheet.

Freeze First Column
Keep the first column visible while scrolling through the rest of the worksheet.

- **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.

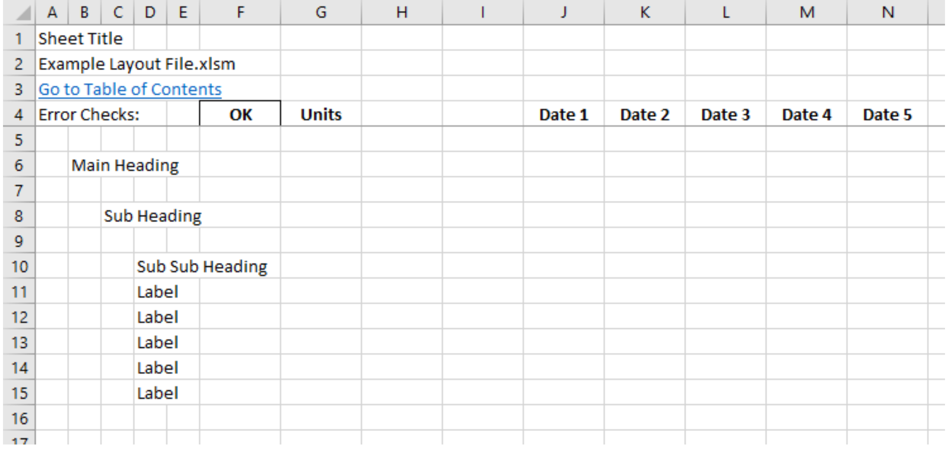
That final option is a little confusing. Essentially the frozen panes are created as follows:



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.

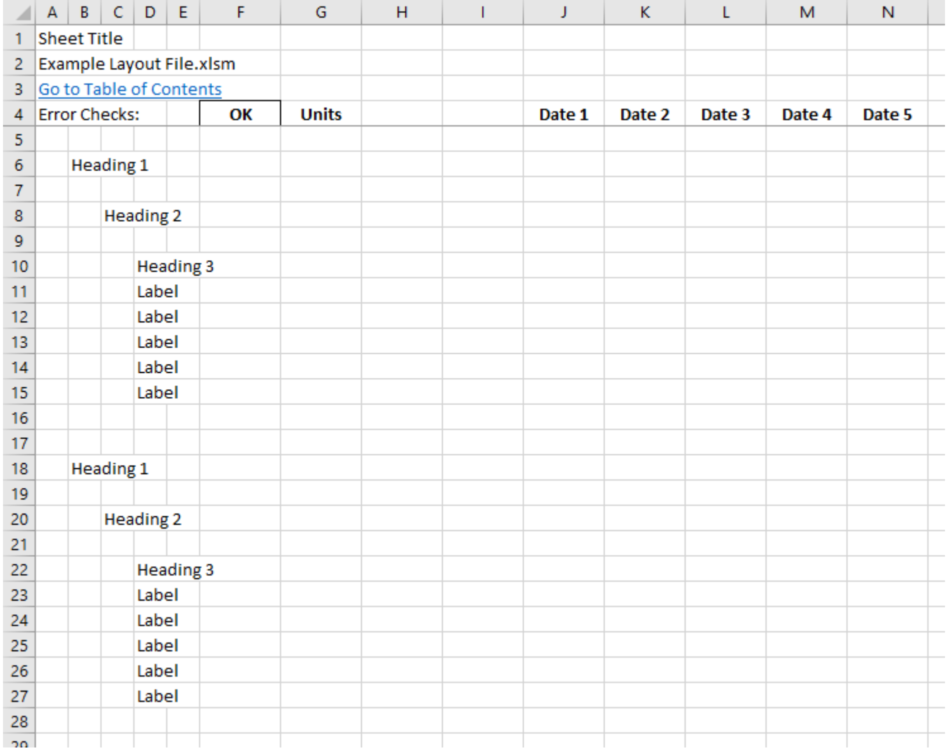
In our example, cell **A5** has been made the basis of the frozen pane, so that rows 1 to 4 will always be visible. This cell should be given a range name, *e.g.* **HL_Home**, as this is the cell hyperlinks to this sheet should link. This cell 'resets' the sheet when a frozen pane has been added (not 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's revisit my unexplained narrowing of the first few columns. Here at SumProduct, we 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. We 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														

These have been labelled "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)*:



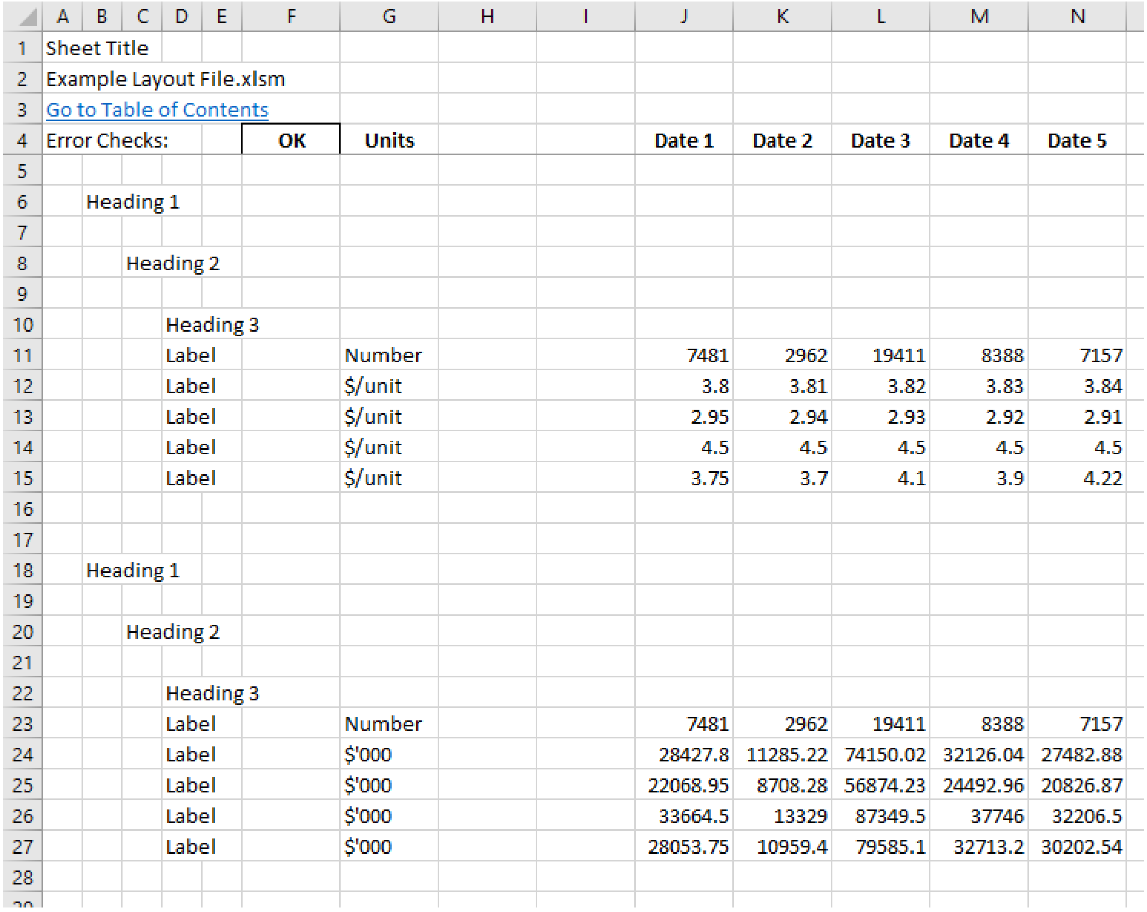
	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Sheet Title													
2	Example Layout File.xlsm													
3	Go to Table of Contents													
4	Error Checks:					**OK**	**Units**			**Date 1**	**Date 2**	**Date 3**	**Date 4**	**Date 5**
5														
6		Heading 1												
7														
8			Heading 2											
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 we have narrowed columns **B**, **C** and **D** (we are 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 our 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.

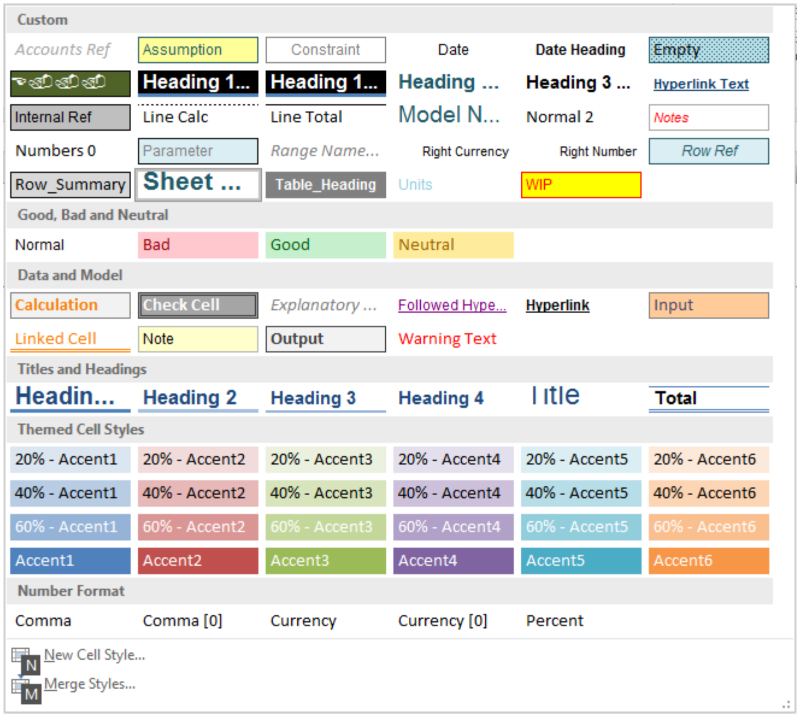
Blank columns **H** and **I** are in existence in case we have any calculations, inputs or referred values that do not refer to a particular time period. If they are not required, we tend to narrow the columns to a width of 1 (say), so that they are still there in case they are needed later.

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**):



Custom

Accounts Ref | Assumption | Constraint | Date | Date Heading | Empty

Heading 1... | Heading 1... | Heading ... | Heading 3 ... | Hyperlink Text

Internal Ref | Line Calc | Line Total | Model N... | Normal 2 | Notes

Numbers 0 | Parameter | Range Name... | Right Currency | Right Number | Row Ref

Row_Summary | Sheet ... | Table_Heading | Units | WIP

Good, Bad and Neutral

Normal | Bad | Good | Neutral

Data and Model

Calculation | Check Cell | Explanatory ... | Followed Hype... | Hyperlink | Input

Linked Cell | Note | Output | Warning Text

Titles and Headings

Headin... | Heading 2 | Heading 3 | Heading 4 | Title | Total

Themed Cell Styles

20% - Accent1 | 20% - Accent2 | 20% - Accent3 | 20% - Accent4 | 20% - Accent5 | 20% - Accent6

40% - Accent1 | 40% - Accent2 | 40% - Accent3 | 40% - Accent4 | 40% - Accent5 | 40% - Accent6

60% - Accent1 | 60% - Accent2 | 60% - Accent3 | 60% - Accent4 | 60% - Accent5 | 60% - Accent6

Accent1 | Accent2 | Accent3 | Accent4 | Accent5 | Accent6

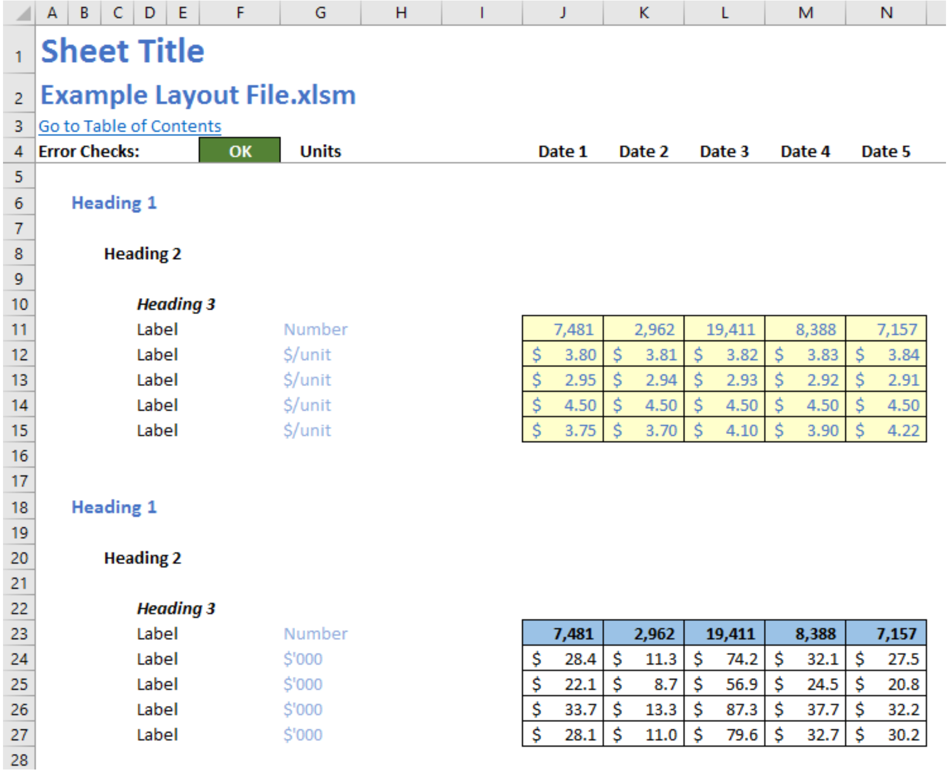
Number Format

Comma | Comma [0] | Currency | Currency [0] | Percent

N New Cell Style...

M Merge Styles...

Very quickly, our spreadsheet is taking shape:



	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			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 we kept column **A** blank? The reason is to take into account 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.

To conclude, 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.

Blast from the Past: Avoiding Circularity When Modelling Circularity

Back in the early days of the SumProduct website, this article was one of the most popular topics and is still requested by many. It is still one of our top-ranked articles ever. So where better to reproduce it than in our 10th birthday newsletter!

When a formula refers back to its own cell, either directly or indirectly, it is called a circular reference. Microsoft Excel cannot automatically calculate all open workbooks when one or more of them contains a circular reference and usually will put zero as the default value in the cell(s) instead.

You can remove a circular reference, or you can have Excel calculate each cell involved in the circular reference using the 'calculate iterations' feature. This requires you enabling iterations:

- go to Excel Options (**ALT + T + O**)
- choose 'Formulas' from the list in the left-hand column
- in the first section, 'Calculation options', ensure the check box 'Enable iterative calculation' is checked
- amend the 'Maximum Number of Iterations' (maximum is still 32,767) as required
- amend the 'Maximum Change' as required (the smaller the number, the longer it will take for Excel to calculate the answer)
- click 'OK'.

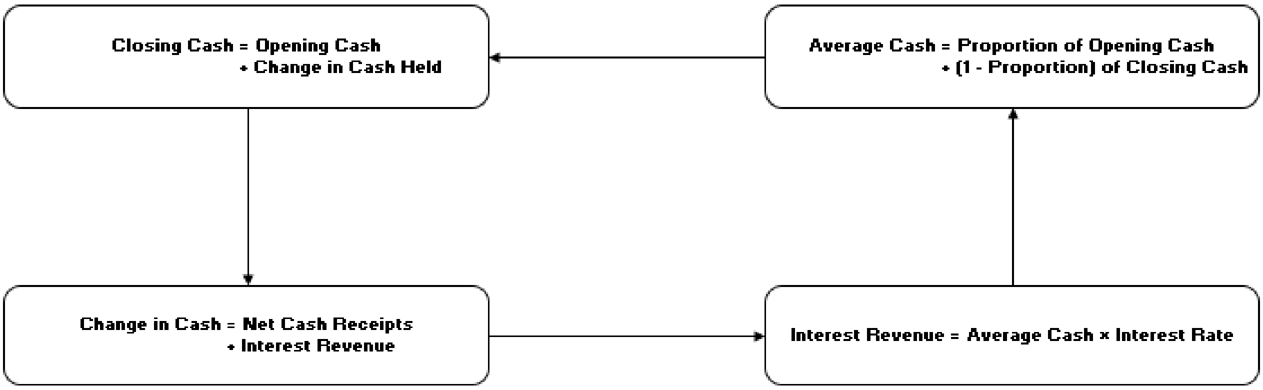
Various problems may arise with circular references:

- many users will agree that circular arguments can cause the Excel file to become unstable and even crash
- when solved, Excel may give one solution when there may be several, only one of which is correct in the given circumstances. If the problem is not fully understood, the danger is an incorrect solution may be accepted
- if Excel stops calculating after a given number of iterations / when the difference between iterations becomes miniscule, users expect the resulting values to be a solution when it may not be (you must always verify that the value reported provides the result required).

Therefore, circular arguments are not recommended, although it is conceded on occasion there may be no alternative viable solution.

Calculating interest without circularity

In a financial model, it is commonplace to have to calculate interest. For this illustration, let's assume we are calculating interest received on the business's average cash balance for certain periods of time (it could just as simply be interest paid on a debt balance, etc.). This gives rise to a perceived circular logic:


Closing Cash = Opening Cash
+ Change in Cash Held
Average Cash = Proportion of Opening Cash
+ (1 - Proportion) of Closing Cash
Change in Cash = Net Cash Receipts
+ Interest Revenue
Interest Revenue = Average Cash × Interest Rate

This problem can be solved algebraically in, er, a relatively straightforward manner without creating circularities – and is therefore our recommended approach.

In a newsletter, we wouldn't normally publish the following, but the derivation of the formula has proved to be one of our most popular web pages (see http://www.sumproduct.com/thought/interest-received). Therefore, we apologise for the following mathematical assault (for those not interested in the derivation, simply skip to the end) – unfortunately, Excel modelling sometimes boils down to solving simultaneous equations!

Let:

OB	=	opening cash balance for the period
CB	=	closing cash balance for the period
M	=	non-interest cash movement for the period
I	=	interest cash movement for the period
r	=	interest rate
t	=	tax rate (it is assumed this cannot equal 100%)
x	=	proportion into the period that the non-interest cash movements are assumed to occur, e.g. • If x = 0%, this means the movement occurred at the start of the period • If x = 100%, this means that the movement occurred at the end of the period • If x = 50%, this means that the movement occurred midway through the period

So,

$$\mathbf{CB} = \mathbf{OB} + \mathbf{M}(1-\mathbf{t}) + \mathbf{I}(1-\mathbf{t}) \qquad \text{and}$$

$$\mathbf{I}(1-\mathbf{t}) = (\mathbf{x}.\mathbf{OB} + (1-\mathbf{x}).\mathbf{CB}).\mathbf{r}.(1-\mathbf{t}) \qquad \text{so (as } \mathbf{t} \neq 100\%)$$

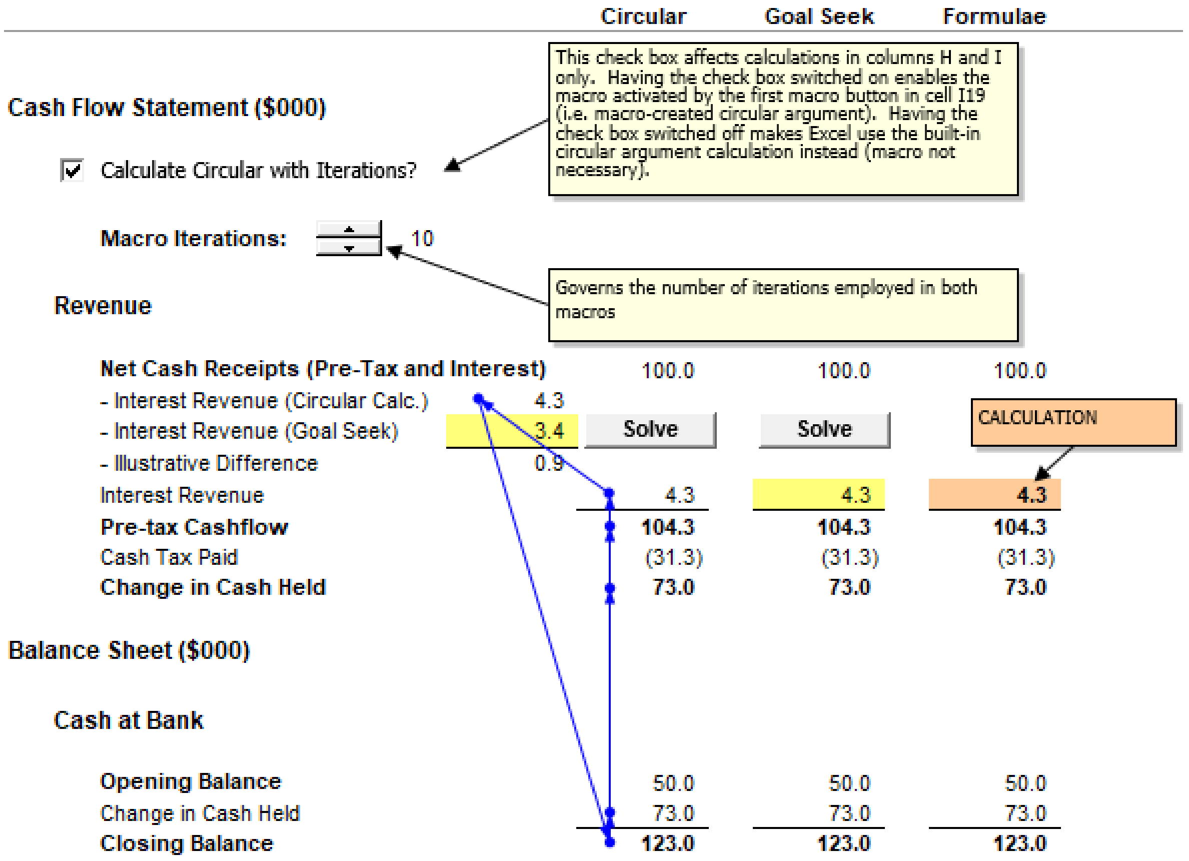
$$\begin{aligned}\mathbf{I} &= (\mathbf{x}.\mathbf{OB} + (1-\mathbf{x}).\mathbf{CB}).\mathbf{r} \\ &= (\mathbf{x}.\mathbf{OB} + (1-\mathbf{x}).(\mathbf{OB} + \mathbf{M}(1-\mathbf{t}) + \mathbf{I}(1-\mathbf{t}))).\mathbf{r} \\ &= \mathbf{OB}.\mathbf{r} + (1-\mathbf{x}).\mathbf{M}.(1-\mathbf{t}).\mathbf{r} + (1-\mathbf{x}).\mathbf{I}.(1-\mathbf{t}).\mathbf{r}\end{aligned}$$

Therefore,

$$\mathbf{I}.(\,1-(1-\mathbf{x}\,).(\,1-\mathbf{t}\,).\mathbf{r}\,) = \mathbf{OB}.\mathbf{r} + (\,1-\mathbf{x}\,).\mathbf{M}.(\,1-\mathbf{t}\,).\mathbf{r}$$

$$\Leftrightarrow \quad \mathbf{I} = \frac{\mathbf{OB}.\mathbf{r} + (\,1-\mathbf{x}\,).\mathbf{M}.(\,1-\mathbf{t}\,).\mathbf{r}}{(\,1-(1-\mathbf{x}\,).(\,1-\mathbf{t}\,).\mathbf{r}\,)}$$

Hence, we can calculate interest from this final equation and have no circular references or goal seek. Please see http://www.sumproduct.com/thought/interest-received for an example Excel file that illustrates this technique:



Cash Flow Statement ($000)

☑ Calculate Circular with Iterations?

This check box affects calculations in columns H and I only. Having the check box switched on enables the macro activated by the first macro button in cell I19 (i.e. macro-created circular argument). Having the check box switched off makes Excel use the built-in circular argument calculation instead (macro not necessary).

Macro Iterations: 10

Governs the number of iterations employed in both macros

Revenue

		Circular	Goal Seek	Formulae
Net Cash Receipts (Pre-Tax and Interest)		100.0	100.0	100.0
- Interest Revenue (Circular Calc.)	4.3			
- Interest Revenue (Goal Seek)	3.4	Solve	Solve	CALCULATION
- Illustrative Difference	0.9			
Interest Revenue		4.3	4.3	**4.3**
Pre-tax Cashflow		**104.3**	**104.3**	**104.3**
Cash Tax Paid		(31.3)	(31.3)	(31.3)
Change in Cash Held		**73.0**	**73.0**	**73.0**

Balance Sheet ($000)

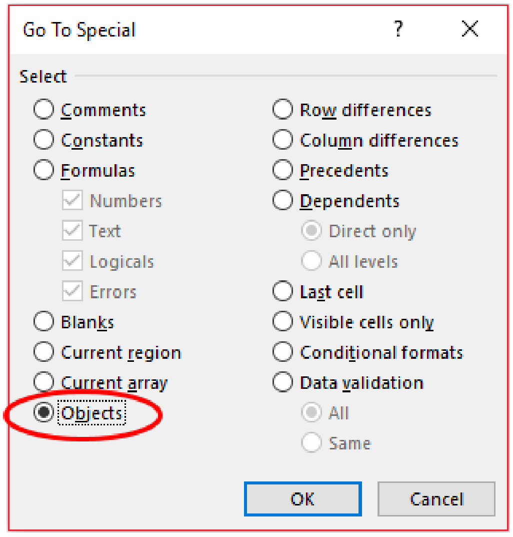
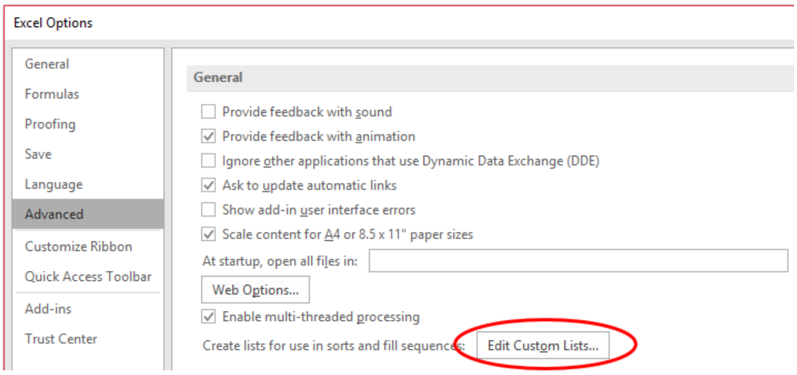
Cash at Bank

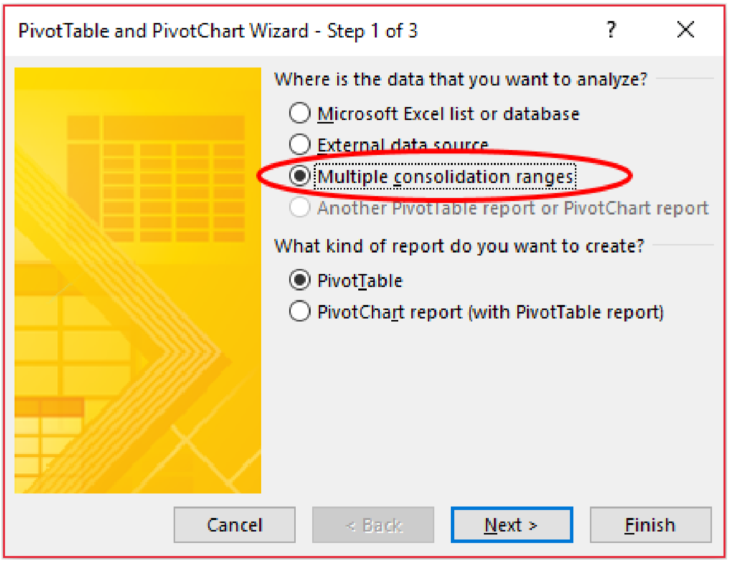
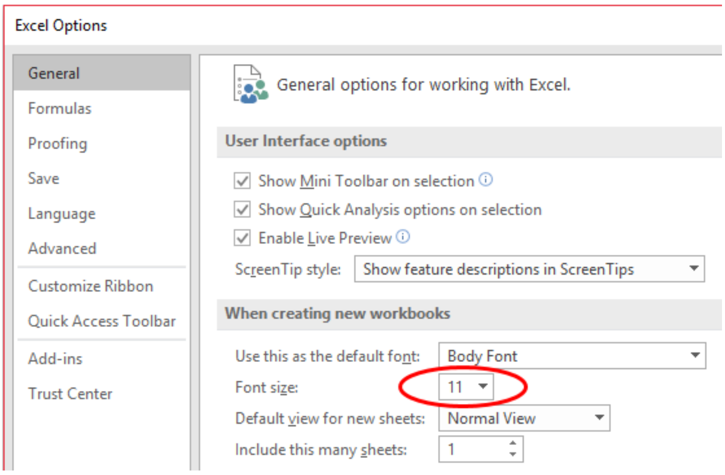
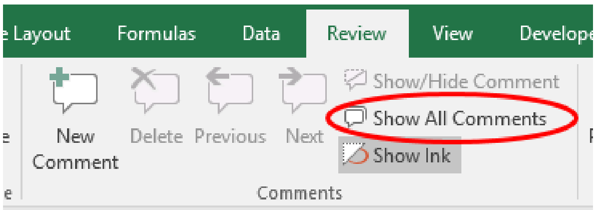
	Circular	Goal Seek	Formulae
Opening Balance	50.0	50.0	50.0
Change in Cash Held	73.0	73.0	73.0
Closing Balance	**123.0**	**123.0**	**123.0**

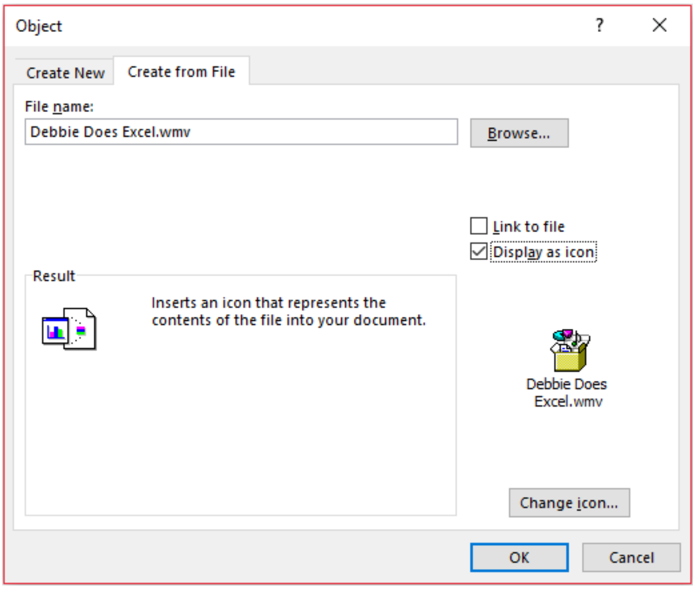
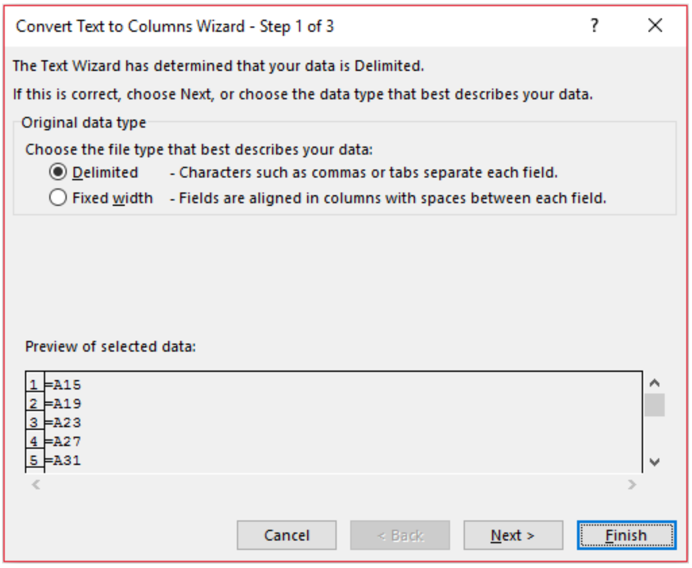
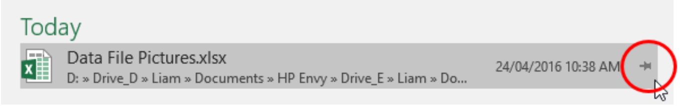
Any Questions..?

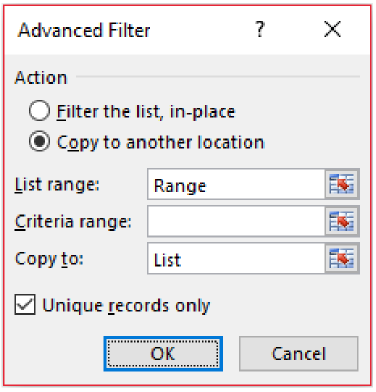
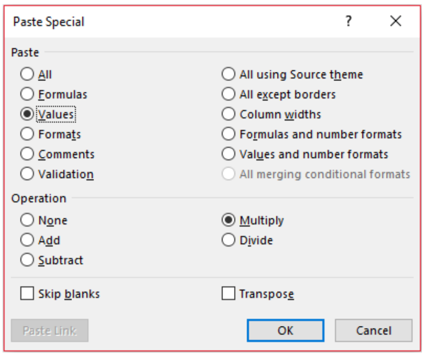
Where would we be without our readers and their questions? Whilst our *specific* replies are a little *sporadic*, we deal with common questions through each and every newsletter. Below are some of those we specifically provided short form responses too for this special edition newsletter.

Question	Answer
What is the maximum number of functions you nest in an Excel formula?	Excel 2003 and earlier allowed seven nesting levels; Excel 2007 and later allow 64. We will only allow you two. After this you will be taken outside and shot. Nesting functions usually occur when the modeller is unaware there is a better function / formula to use.
How do you round a number to the nearest odd or even number?	Use the formulae **=ODD(ROUND(Number,0))** and **=EVEN(Round(Number,0))**

Is it possible to select all objects in a sheet simultaneously?	Yes: - Press the shortcut key **F5** to open the 'Go To' dialog box - Click on the 'Special…' button - Select 'Objects' radio button - Click 'OK' - All objects will get selected  ○ Comments ○ Constants ○ Formulas ☑ Numbers ☑ Text ☑ Logicals ☑ Errors ○ Blanks ○ Current region ○ Current array ● Objects ○ Row differences ○ Column differences ○ Precedents ○ Dependents ● Direct only ○ All levels ○ Last cell ○ Visible cells only ○ Conditional formats ○ Data validation ● All ○ Same
I notice if I enter 'January' in one cell and then 'February' in the next cell and drag down (or across) the remaining months autofill. Is it possible to do this with common lists used at my company?	Absolutely! Go to 'Excel Options' (**ALT + T + O**) and select 'Advanced' from the column menu. Scroll down the main pane to 'General' and click on the 'Edit Custom Lists…' button. Type in your list or import it from an Excel range and click the 'Add' button. Done!  Excel Options General Formulas Proofing Save Language Advanced Customize Ribbon Quick Access Toolbar Add-ins Trust Center General □ Provide feedback with sound ☑ Provide feedback with animation □ Ignore other applications that use Dynamic Data Exchange (DDE) ☑ Ask to update automatic links □ Show add-in user interface errors ☑ Scale content for A4 or 8.5 x 11" paper sizes At startup, open all files in: ☑ Enable multi-threaded processing Create lists for use in sorts and fill sequences:
I cannot find where something is on the Ribbon – or even if it is there. How can I find it and make it easier to find going forward?	Microsoft has developed a range of interactive guides such as those that can be found at https://www.microsoft.com/en-us/download/search.aspx?q=interactive+menu+to+ribbon+guide. They are ok, but another useful resource often forgotten about is the Quick Access Toolbar. Right-click on the Quick Access Toolbar in Excel and then choose 'Customize Quick Access Toolbar…' from the shortcut menu (or else select 'Excel Options, **ALT + T + O** and then 'Quick Access Toolbar' from the columnar menu). Select 'All Commands' from the 'Choose commands from:' drop down menu. All commands will be listed alphabetically in the pane below the drop down box. Seek out what you are looking for, highlight the command and press the 'Add' button. It will now be on your Quick Access toolbar for easy future access.
Is it possible to create a PivotTable from more than one source without using Power Pivot?	There are ways and means to wriggle around this problem. It's easy if the data is all in multiple ranges of the same workbook (simply use the hidden 'wizard', **ALT + D + P**) and select 'Multiple consolidation ranges'. However, it is simpler to use Power Pivot if you are using Excel 2010 or later.

	 PivotTable and PivotChart Wizard - Step 1 of 3 Where is the data that you want to analyze? ○ Microsoft Excel list or database ○ External data source ● Multiple consolidation ranges ○ Another PivotTable report or PivotChart report What kind of report do you want to create? ● PivotTable ○ PivotChart report (with PivotTable report) Drop us a line at training@sumproduct.com for more details on Power Pivot.
I have accidentally created an add-in by saving one of my Excel files as an "add-in" type. Going to **Tools-> Add-Ins**, I cannot seem to remove it. How can it be deleted?	Easy. Move the offending file to a different location temporarily whilst Excel is closed. You may get an error message about this add-in upon opening Excel. Then, open Excel and open the 'Add-ins' dialog box. Check on the 'phantom' add-in. Excel will say it cannot find it and ask whether you wish to delete it from the list. Click 'Yes' and then move your Excel file back to its original location. All done.
How can I make the text on my Formula bar larger?	Go to 'Excel Options' (**ALT + T + O**), then select 'General' from the columnar menu. In the 'When creating new workbooks' section, change the font size. You will need to restart Excel for the effects to take place.  Excel Options General Formulas Proofing Save Language Advanced Customize Ribbon Quick Access Toolbar Add-ins Trust Center General options for working with Excel. User Interface options ☑ Show Mini Toolbar on selection ☑ Show Quick Analysis options on selection ☑ Enable Live Preview ScreenTip style: Show feature descriptions in ScreenTips When creating new workbooks Use this as the default font: Body Font Font size: 11 Default view for new sheets: Normal View Include this many sheets: 1
How can I make all comments in the cells on a particular worksheet appear simultaneously?	This appears to be a little-known feature in Excel for some reason. Go to the 'Review' tab on the Ribbon and in the 'Comments' grouping select 'Show All Comments'.  Layout Formulas Data Review View Develope New Comment Delete Previous Next Show/Hide Comment Show All Comments Show Ink Comments

Can you embed videos in Excel?	This is an unusual query, but yes, you can although we are not sure we recommend it. Go the 'Insert' tab on the Ribbon and then click on 'Object' in the 'Text' grouping (obviously!). Click on the 'Create from File' tab in the 'Object' dialog box and browse to the video. Check the 'Display as icon' box (checking 'Link to file' links rather than embeds the video) and then click 'OK' (you can change the icon by pressing the 'Change icon…' button).  Object ? × Create New | Create from File File name: Debbie Does Excel.wmv Browse... ☐ Link to file ☑ Display as icon Result Inserts an icon that represents the contents of the file into your document. Debbie Does Excel.wmv Change icon... OK | Cancel Do note this make the Excel file very large!!
I need to refer to every fourth row in a block of cells, *e.g.* cell **A1** has the formula **=A15**, cell **A2** has the formula **=A19**, cell **A3** has the formula **=A23** and so on. Is there a quick way to create these formulae without having to write out the calculation in each row?	Yes – and this technique works for any pattern where an equal number of rows is omitted each time. Type an apostrophe in front of the formula in cell **A1**, so that it becomes **'=A15**. Do the same for **A2**, *i.e.* the formula becomes **'=A19**. Now copy these formulae down. You will get **'=A23**, **'=A27** and so on. Now highlight all of these formulae and on the 'Data' tab, click on the 'Text to Columns' icon in the 'Data Tools' grouping. When the dialog box appears, just click on the 'Finish' button. It's as simple as that!  Convert Text to Columns Wizard - Step 1 of 3 ? × The Text Wizard has determined that your data is Delimited. If this is correct, choose Next, or choose the data type that best describes your data. Original data type Choose the file type that best describes your data: ● Delimited - Characters such as commas or tabs separate each field. ○ Fixed width - Fields are aligned in columns with spaces between each field. Preview of selected data: 1 =A15 2 =A19 3 =A23 4 =A27 5 =A31 Cancel | < Back | Next > | Finish
I use certain files all of the time, although they may not appear in my recent list. Is there a quick way to access them without having to go File->Open and browse to the appropriate directory?	Yes. Open the file in Excel and then return to what is known as the 'backstage' area by clicking on the file tab. You will now see your file in the list of recently opened files. You will see a 'pin' to the right of each file recently opened. Click on this and it will 'pin' this file so that it is always visible when you are selecting a file to open.  Today Data File Pictures.xlsx D: » Drive_D » Liam » Documents » HP Envy » Drive_E » Liam » Do... 24/04/2016 10:38 AM

How can I find the first item in a list that is a **case sensitive** match for what I am searching for?	Use the formula **{=MATCH(TRUE,EXACT(Criterion,List),0)}** This is entered as an array formula (**CTRL + SHIFT + ENTER**) – so braces are not manually entered. Please see https://www.sumproduct.com/thought/index-match for further information.
How can I count the number of unique items in an Excel list if I don't have access to Excel 365 / dynamic arrays?	Use the formula **=SUMPRODUCT((List<>"")/COUNTIF(List,List&""))** to count the number of unique items in the **List**. For more information, please see https://www.sumproduct.com/thought/sumproduct-squared.
How can I create a list of the unique entries from a list in Excel which may contain duplicates when I don't have access to Excel 365 / dynamic arrays?	This is when you should use the 'Advanced Filter' in Excel, located in the 'Sort & Filter' grouping of the 'Data' tab. Depending upon whether your list has a heading or not, the 'Advanced Filter' may prompt you to elect whether the list contains a heading (to be ignored). Select the **Range** and the **Location** where the unique list should be created. Then, complete the dialog box as follows:  Advanced Filter ? × Action ○ Filter the list, in-place ● Copy to another location List range: Range Criteria range: Copy to: List ☑ Unique records only OK Cancel
How can I get numbers in millions to appear in millions, numbers in thousands in thousands and others in units?	Use custom number formatting. Click on the cells to be formatted and then use the keyboard shortcut **CTRL + 1** to Format Cells. Select 'Custom' on the 'Number' tab and use syntax such as **[>=1000000]#,##0,,"m";[>=1000]#,##0,"k";#,##0** For more information on number formatting, please refer to https://www.sumproduct.com/thought/number-formatting.
How can I express a decimal as a fraction in Excel?	Use the formula **=TEXT(Decimal,"? ?/?")**
What is the quickest way to negate a set of input data?	Of course, you can always fall back on the formula **=-Cell_Reference**, but if you just want all of the inputs to be negated in one go, simply type -1 into a cell and then copy it. Then, highlight the data you wish to negate and Paste Special as Values (**ALT + E + S + V**) using the Multiply operator. Simple!  Paste Special ? × Paste ○ All ○ All using Source theme ○ Formulas ○ All except borders ● Values ○ Column widths ○ Formats ○ Formulas and number formats ○ Comments ○ Values and number formats ○ Validation ○ All merging conditional formats Operation ○ None ● Multiply ○ Add ○ Divide ○ Subtract □ Skip blanks □ Transpose Paste Link OK Cancel

How do I find the third largest value in a set of data?	Use the **LARGE** function: **=LARGE(Data, n)** returns the **n**th largest value in the **Data**. **SMALL** works similarly but finds the **n**th smallest.
How do I find the sum of the first 10 items in a list even if the list may have rows inserted or deleted (*i.e.* it always sums the first 10)?	Use the formula **=TEXT(Decimal,"? ?/?")**
What is the quickest way to negate a set of input data?	Assume the range to be summed initially is the selection **A1:A10**. The formula **=INDIRECT("A1:A10")** will always sum this range regardless of how many rows are added or deleted. For more information on the **INDIRECT** function, please see https://www.sumproduct.com/thought/being-direct-about-indirect.

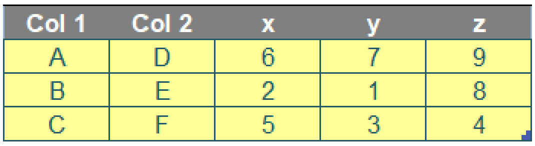
Beat the Boredom Challenge

*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…*

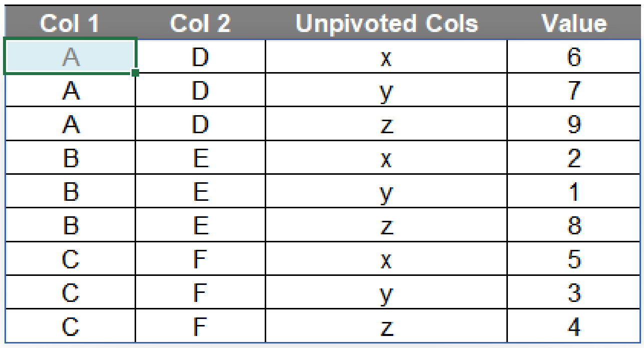
Sometimes, when you work with pivoted data that has a structure similar to a PivotTable, it is difficult to look up a value based on multiple column and row criteria. To make it simpler, we usually unpivot the data. Most of the time, you may choose to use the 'Unpivot Columns' feature in Power Query (please see https://www.sumproduct.com/blog/article/power-query-blogs/power-query-6-painless-unpivoting?L=0 for more details).

This challenge is designed to make you think outside the box to find another way using only Excel formulae.

This month's challenge is to write a **formula in one cell** using dynamic arrays (please see https://www.sumproduct.com/thought/getting-arrays-spilling-the-beans-on-seven-new-functions?L=0 for more details) that will spill a range of cells to unpivot only the last three [3] columns (*i.e.* **x**, **y** and **z**) of an array. Assuming the example on the left *(below)*, the result should look like the array generated on the right:



Col 1	Col 2	x	y	z
A	D	6	7	9
B	E	2	1	8
C	F	5	3	4



Col 1	Col 2	Unpivoted Cols	Value
A	D	x	6
A	D	y	7
A	D	z	9
B	E	x	2
B	E	y	1
B	E	z	8
C	F	x	5
C	F	y	3
C	F	z	4

As always, there are some requirements:

- the formula needs to be in just one cell (no "helper" cells)
- this is a formula challenge; no Power Query / Get & Transform or VBA!
- the formula needs to be flexible, so that if we adjust the number of rows and / or columns of the input table, the formula should still work
- obviously, the numbers of rows / columns of the output table cannot exceed the row / column limitations of Excel.

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

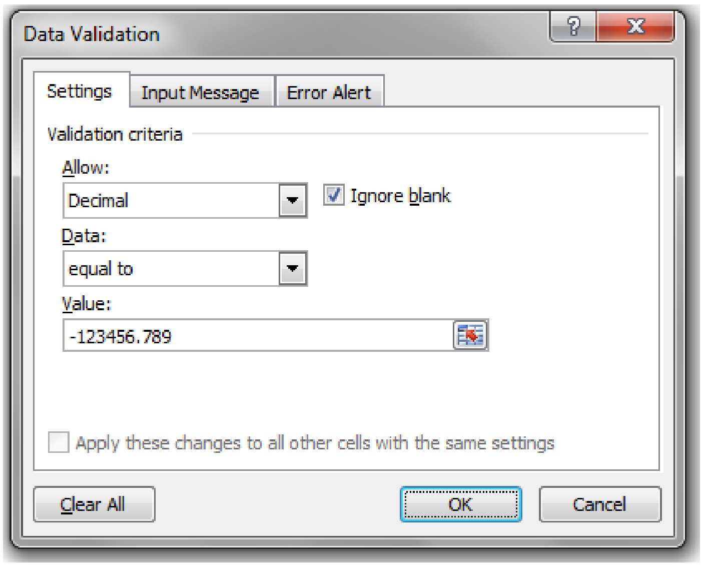
Our Top Ranked Tip

This one may have gone into legend these days, but after a challenge laid down way back in 2015, this was voted SumProduct's (Liam Bastick's??) Top Tip. Use this one at your peril – used in the workplace, it may spell the end of your career… We thought we'd keep the nostalgia by retaining the screen grabs from the time!

We have some twisted individuals reading this newsletter! By far and away the most popular 'trick' (in all sense of the word!) was this monster first divulged in our very first newsletter by our own resident MVP **Liam Bastick**. We have elected not to name all the people who suggested this – partly to save printing costs and partly to protect the guilty. You know who you are!!

Data Validation is a useful way to control what end users can type into a worksheet cell (see http://www.sumproduct.com/thought/data-validation). You can use this functionality to play a trick. Please use this at your own risk: if you get fired, you will get no sympathy here and we will disown all knowledge: this article will self-destruct in five seconds...

If someone is unfortunate to leave a spreadsheet unprotected, simply highlight the whole worksheet and then activate Data Validation (**ALT + D + L**). In the 'Settings' tab, select settings similar to the following (the aim is to pick a number the user won't use):



Data Validation

Validation criteria

Allow: Decimal

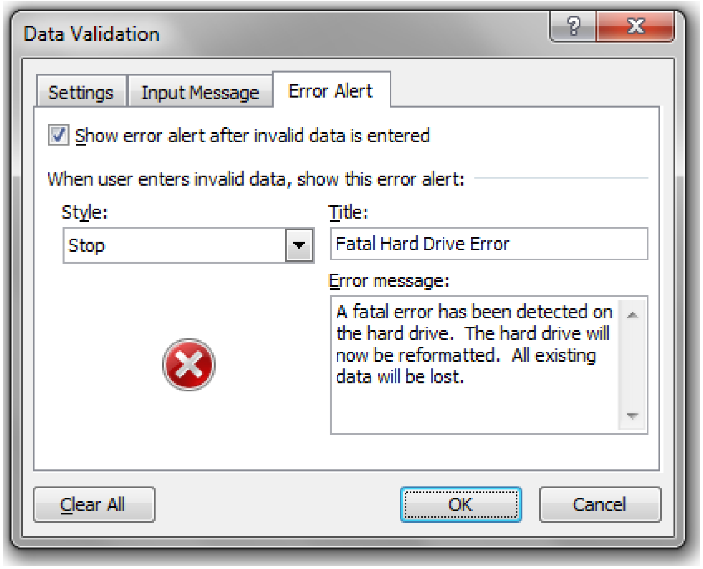
☑ Ignore blank

Data: equal to

Value: -123456.789

☐ Apply these changes to all other cells with the same settings

Then, select the 'Error Alert' tab:



Data Validation

☑ Show error alert after invalid data is entered

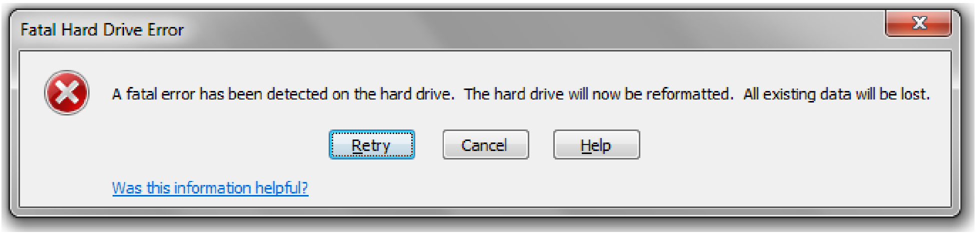
When user enters invalid data, show this error alert:

Style: Stop

Title: Fatal Hard Drive Error

Error message: A fatal error has been detected on the hard drive. The hard drive will now be reformatted. All existing data will be lost.

If you are feeling particularly horrible, add "Press any key to continue" to this message for maximum effect. Now, de-select the range and wait for your victim to use the worksheet. As soon as they type an invalid entry, they will be greeted with the following error alert:



Fatal Hard Drive Error

A fatal error has been detected on the hard drive. The hard drive will now be reformatted. All existing data will be lost.

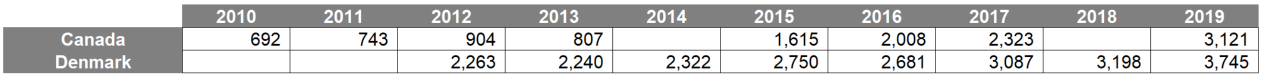
Who says spreadsheets can't be fun..?

Charts and Dashboards

It's time to chart our progress with an introductory series into the world of creating charts and dashboards in Excel. After looking at how to highlight data last month, this time we look at how to hide data instead.

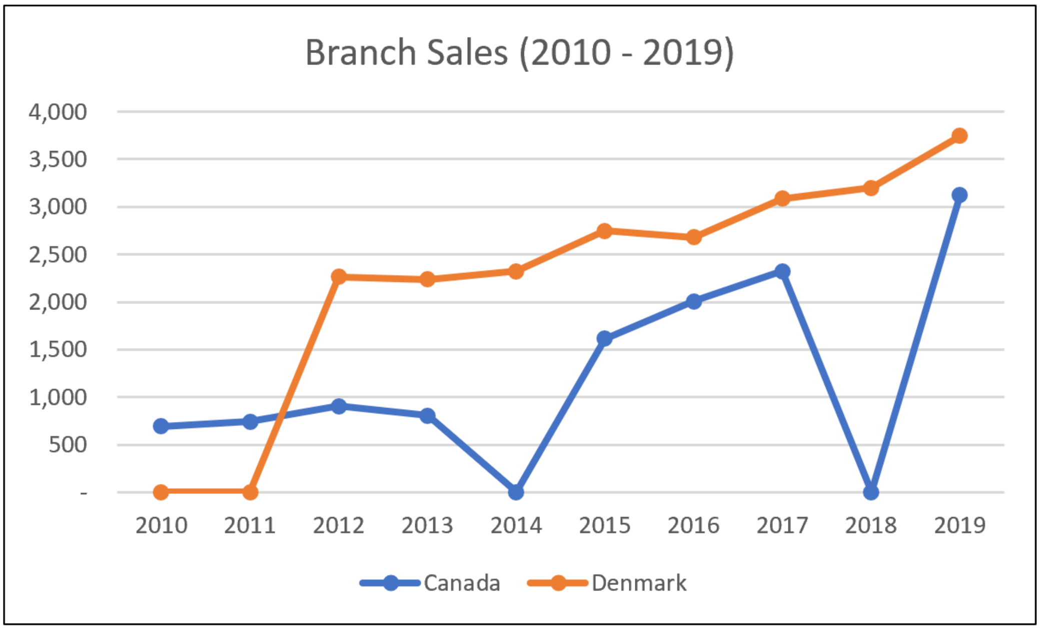
When creating charts with time series, you should not skip periods: if data only exists for seven [7] months of the year, your graph might still be best placed to show all 12 months, so as to remove any chance of the information being misinterpreted. If the reader sees seven columns on a graph, they may not notice that five [5] months of zero results are missing. When creating charts measuring values, typically these would go on the vertical axis and you should always show zero [0].

For example, imagine we have to chart selected data, but my data series is somewhat incomplete:

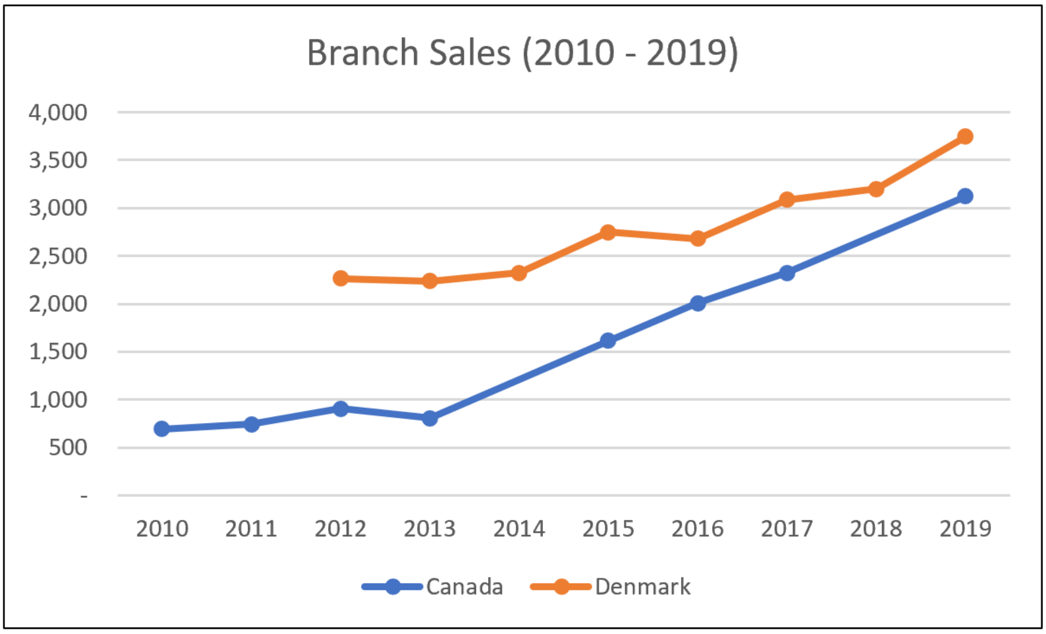


	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019
Canada	692	743	904	807		1,615	2,008	2,323		3,121
Denmark			2,263	2,240	2,322	2,750	2,681	3,087	3,198	3,745

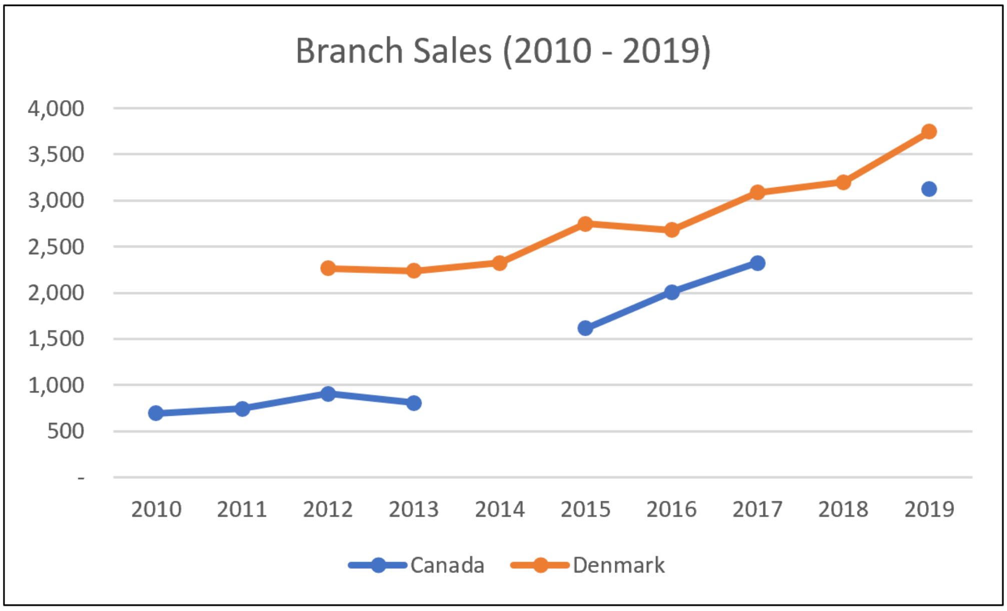
If you have ever created a Line chart, have you ever witnessed the following?



In this instance, Excel has assumed the missing values are zero [0]. Alternatively, you may see the following:



Here, Excel has drawn a line in place of the missing data points. However, we actually want the chart to display like this:



The trick is to review the settings regarding how Excel displays hidden and empty cells. To get there, right-click on any part of the chart, then select ‘Select Data’, and then select the ‘Hidden and Empty Cells’ option, *viz.*

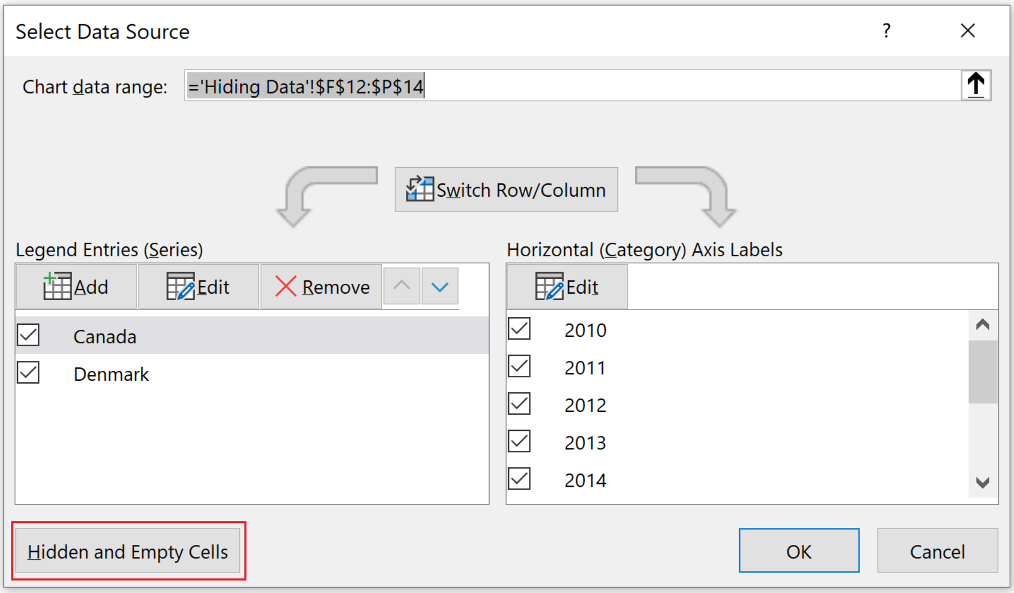
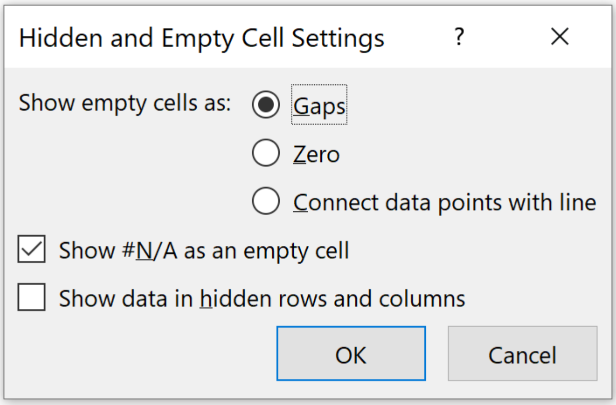

Select Data Source

Chart data range: ='Hiding Data'!$F$12:$P$14

☑ Canada
☑ Denmark

☑ 2010
☑ 2011
☑ 2012
☑ 2013
☑ 2014

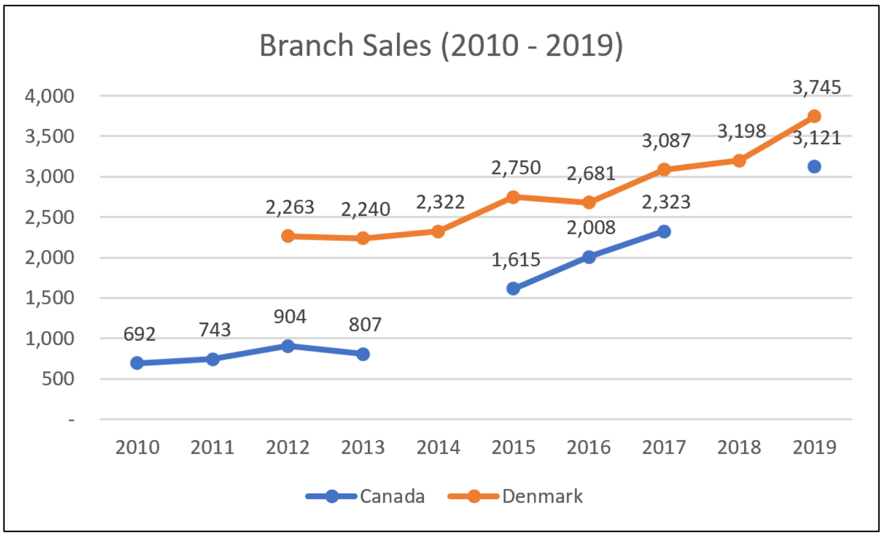
The ‘Hidden and Empty Cell Settings’ dialog will appear:


Hidden and Empty Cell Settings

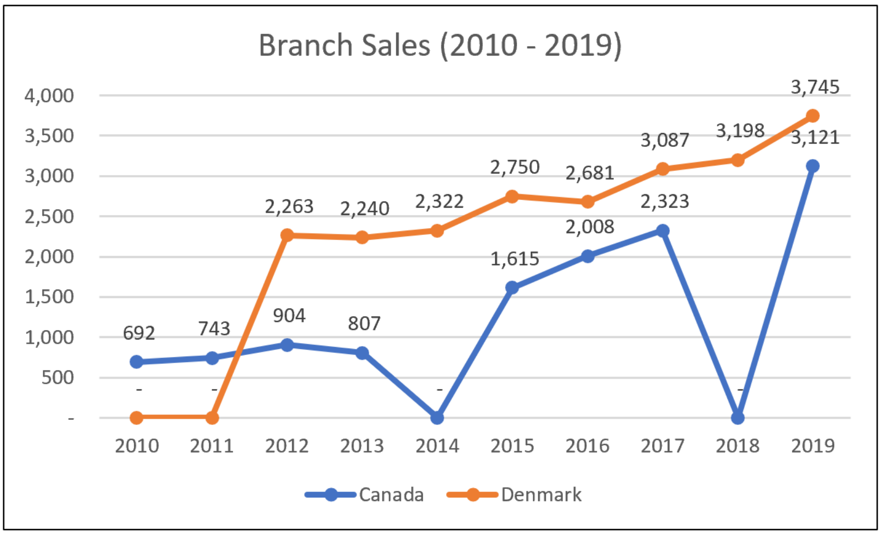
Show empty cells as: ● Gaps
○ Zero
○ Connect data points with line

☑ Show #N/A as an empty cell
□ Show data in hidden rows and columns

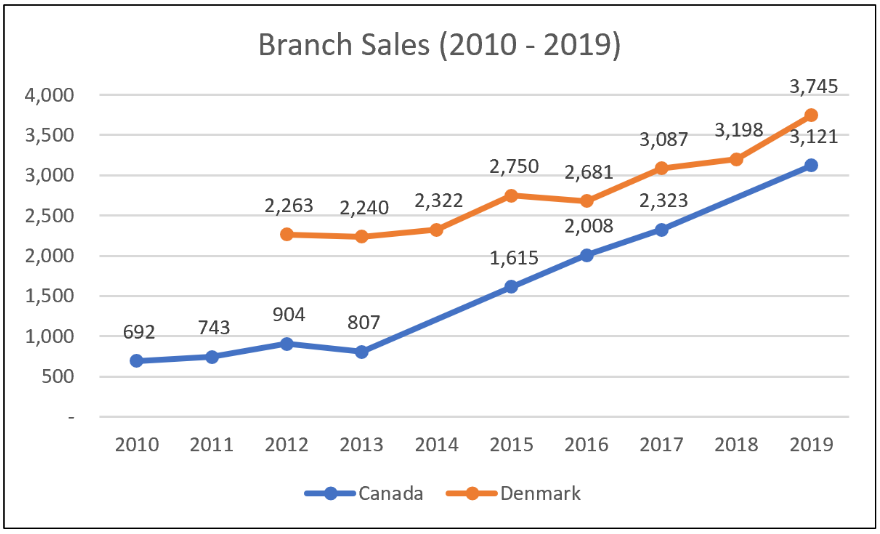
This chart is the result of setting 'Show empty cells as: Gaps':



This chart is the result of setting 'Show empty cells as: Zero':



This chart is the result of setting 'Show empty cells as: Connect data points with line':



This helps end users be better able to understand the charts presented – including alerting them to what data may be missing, *etc.*

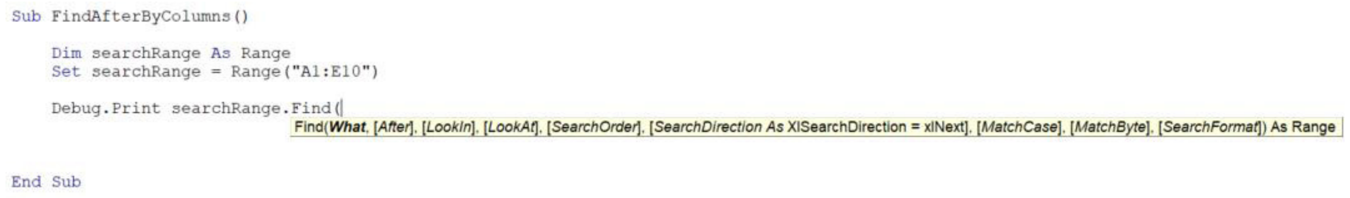
More next month…

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 look VBA up and down…

In last month's newsletter, we set the **After** parameter for the **Find** method to distinguish which cell we wanted to start from. However, we noted that it defaulted to go across the row. It may be set in order to go down the columns first using the SearchOrder parameter. This is where we need to look at how VBA handles parameters.

When we type the **Find** method and open the brackets, we will see the ToolTip for the parameter list, identical to when we use functions in Excel formulae.



```
Sub FindAfterByColumns()

    Dim searchRange As Range
    Set searchRange = Range("A1:E10")

    Debug.Print searchRange.Find(
                    Find(What, [After], [LookIn], [LookAt], [SearchOrder], [SearchDirection As XlSearchDirection = xlNext], [MatchCase], [MatchByte], [SearchFormat]) As Range

End Sub
```

The bolded parameter is the one we are currently entering. The parameters in square brackets are the Optional parameters. We could skip to **SearchOrder** using:

```
searchRange.Find("up", Range("C5"), , , xlByColumns)
```

However, this isn't a good approach. It is very easy (and we will admit to having done this) to put the wrong number of commas in and have the subroutine fail at that point in the code.

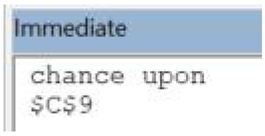
Instead, VBA allows us to assign values to specific parameters directly. This is performed by naming the parameter followed by ":=" then the value. In the previous example, that line is better written as:

```
searchRange.Find("up", After:=Range("C5"), SearchOrder:=xlByColumns)
```

With that borne in mind, our full subroutine will look like this:

```
Sub FindAfterByColumns()
    Dim searchRange As Range
    Set searchRange = Range("A1:E10")
    Dim foundrange As Range
    Set foundrange = searchRange.Find("up", After:=Range("C5"), SearchOrder:=xlByColumns)

    If foundrange Is Nothing Then
        Debug.Print "not found!"
    Else
        Debug.Print foundrange
        Debug.Print foundrange.Address
    End If
End Sub
```

This results in the following:



```
chance upon
$C$9
```

Next newsletter, we will pinpoint exactly how to dig up our column!

Until then.

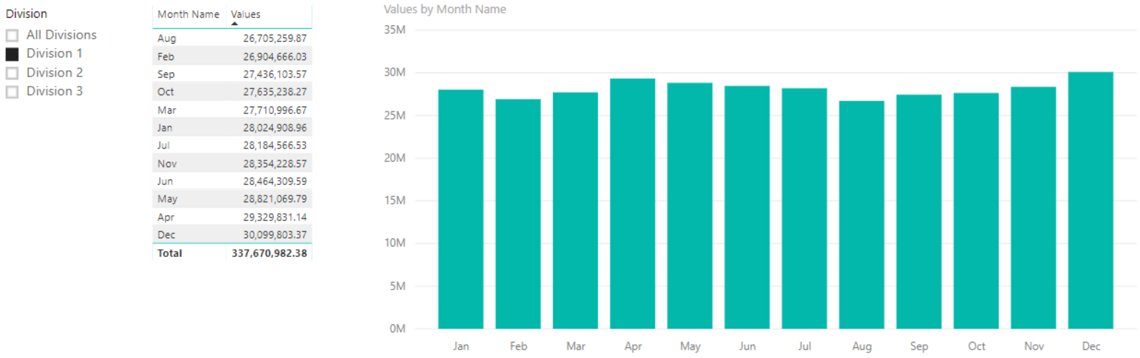
Power Pivot Principles

We continue our series on the Excel COM add-in, Power Pivot. This month, we discuss how to use a combination of disconnected tables and variables to produce unique slicers in Power BI and then consider a similar dynamic toggle for Power Pivot.

This month, we are going to do a bit of a crossover between Power Pivot's Data Analysis eXpressions (DAX) language and Power BI. As you may know, Power BI also uses DAX to create measures and process data just like Power Pivot. The reason why we will be using Power BI initially is because the **SELECTEDVALUE** function detailed below just isn't available in Power Pivot's DAX library – yet.

We are going to create a slicer that is connected to a disconnected table (see https://www.sumproduct.com/blog/article/power-pivot-principles/ppp-disconnected-tables-and-exchange-rates for further details).

Whilst the slicer will have a direct effect on the disconnected table, the table has no link to all of the other tables currently in our dataset. However, through using variables, the **SELECTEDVALUE** and the **SWITCH** function, we will be able to manipulate the visualisations on our dashboard, *viz.*

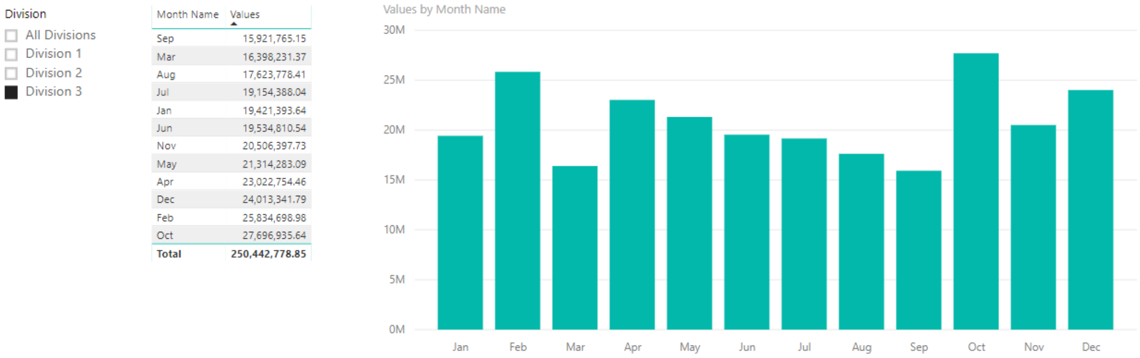


Division

☐ All Divisions
■ Division 1
☐ Division 2
☐ Division 3

Month Name	Values
Aug	26,705,259.87
Feb	26,904,666.03
Sep	27,436,103.57
Oct	27,635,238.27
Mar	27,710,996.67
Jan	28,024,908.96
Jul	28,184,566.53
Nov	28,354,228.57
Jun	28,464,309.59
May	28,821,069.79
Apr	29,329,831.14
Dec	30,099,803.37
Total	**337,670,982.38**

compared with

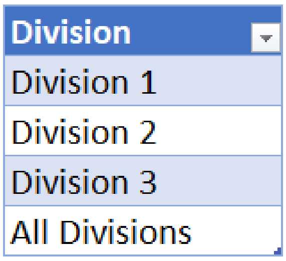


Division

☐ All Divisions
☐ Division 1
☐ Division 2
■ Division 3

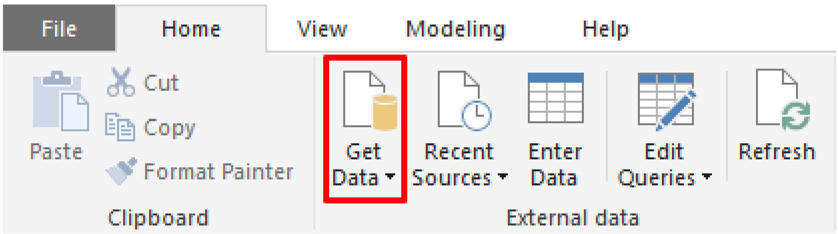
Month Name	Values
Sep	15,921,765.15
Mar	16,398,231.37
Aug	17,623,778.41
Jul	19,154,388.04
Jan	19,421,393.64
Jun	19,534,810.54
Nov	20,506,397.73
May	21,314,283.09
Apr	23,022,754.46
Dec	24,013,341.79
Feb	25,834,698.98
Oct	27,696,935.64
Total	**250,442,778.85**

To begin (assuming we already have all of our data loaded into Power BI and created measures to calculate the sales for Divisions 1, 2, 3, and All Divisions), we need to create a disconnected table with our four [4] categories. We can upload a pre-built table from Excel to detail this categorisation:

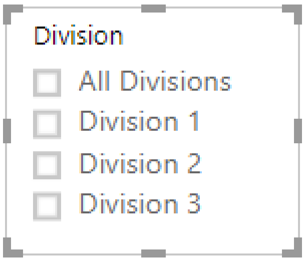


Division
Division 1
Division 2
Division 3
All Divisions

Alternatively, we can create a table directly in Power BI using the 'Enter Data' button on the Ribbon (but do not link this table to any other table in the dataset):



The next step is to create a slicer with the ‘Division’ table:



Division

□ All Divisions

□ Division 1

□ Division 2

□ Division 3

Now on to the measure. If you need to catch up on what variables are in DAX (employed below), we have an article (https://www.sumproduct.com/blog/article/power-pivot-principles/ppp-variables-in-dax) that you may read.

We will use the DAX expression:

```
Values =
        VAR DivisionNo =
        SELECTEDVALUE(‘Division Type’[Division], [All Divisions])

RETURN


        SWITCH(TRUE(),
                DivisionNo = "Division 1", [Division 1],
                DivisionNo = "Division 2", [Division 2],
                DivisionNo = "Division 3", [Division 3],
                [All Divisions])
```

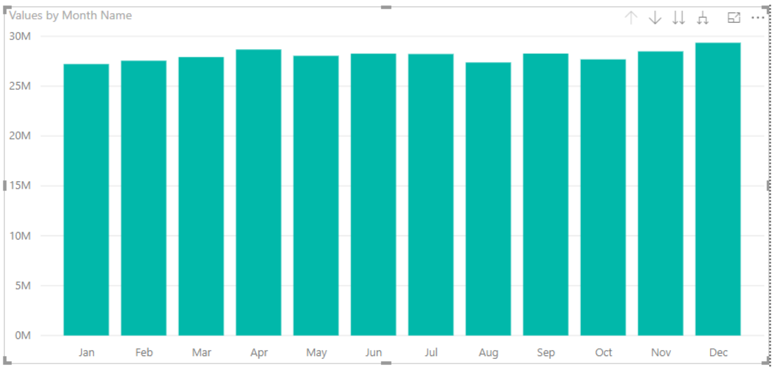
We use the **SELECTEDVALUE** function to declare the value that will be used by the variable ‘DivisionNo’. The **[All Divisions]** measure in the **SELECTEDVALUE** formula specifies the function to return (with the sum of all divisions) should none be selected.

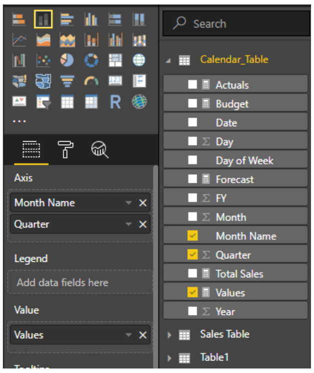
```
VAR DivisionNo =

        SELECTEDVALUE(‘Division Type’[Division], [All Divisions])
```

Then, we shall utilise DivisionNo’s value to call one of three measures that calculates the actual, budgeted or forecasted sales. Distinguishing between the three measures is achieved using the **SWITCH** function:

```
RETURN


SWITCH(TRUE(),
        DivisionNo = "Division 1", [Division 1],
        DivisionNo = "Division 2", [Division 2],
        DivisionNo = "Division 3", [Division 3],
        [All Divisions])
```

The final step is to insert the Values field into any visualisations where we want to be able to switch between actual, budgeted or forecast sales.





Axis

Month Name

Quarter

Legend

Add data fields here

Value

Values

Calendar_Table

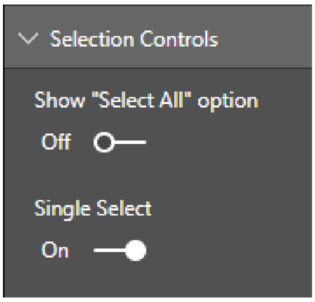
- □ Actuals
- □ Budget
- □ Date
- □ Σ Day
- □ Day of Week
- □ Forecast
- □ Σ FY
- □ Σ Month
- ☑ Month Name
- ☑ Σ Quarter
- □ Total Sales
- ☑ Values
- □ Σ Year

Sales Table

Table1

We can now toggle freely between the three divisions that we have specified and have the visualisations updated accordingly.

It should be noted that we do not need to worry about our users selecting two or three options in the slicer. We've deliberately left 'Single Select' on which does not allow for multiple option selections in the slicer.



Selection Controls

Show "Select All" option: Off

Single Select: On

So far, so good. Now that we've detailed a method using variables and disconnected tables to create a dynamic toggle to display our data in Power BI, let's refer to Power Pivot – given this is what this article is supposed to be about!

We are now going to revisit the dynamic toggle from above and construct it with Power Pivot in Excel. As aforementioned, the **SELECTEDVALUE** function is not in Power Pivot's DAX library. Therefore, we have to use a combination of functions to replicate the formula in Power Pivot.

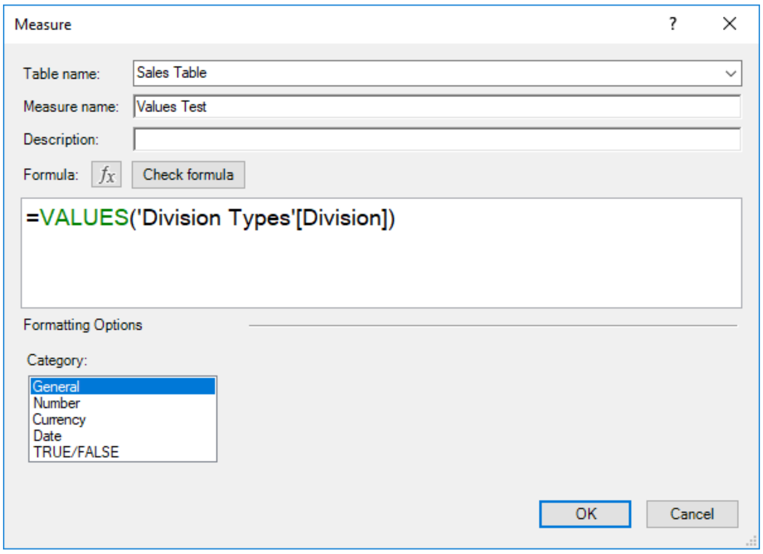
To replicate the **SELECTEDVALUE** result we have to use a combination of the **IF**, **HASONEVALUE** and **VALUES** functions in the following manner:

```
IF(HASONEVALUE('Division Types'[Division]),
        VALUES('Division Types'[Division]),[All Divisions])
```

The **HASONEVALUE** function will return with TRUE when the column name has been filtered down to one distinct value, otherwise it will return with FALSE. This is a useful function when you want a different formula for subtotals and totals, for example.

The **VALUES** function returns with a single text value from the '**Division Types**' column. This function will not function properly if the column is not filtered down to a single value. Therefore, we need to employ the use of the **HASONEVALUE** function and the **IF** function as an error trap for the measure.

If we create the following measure with just the **VALUES** function:



Measure

Table name: Sales Table

Measure name: Values Test

Description:

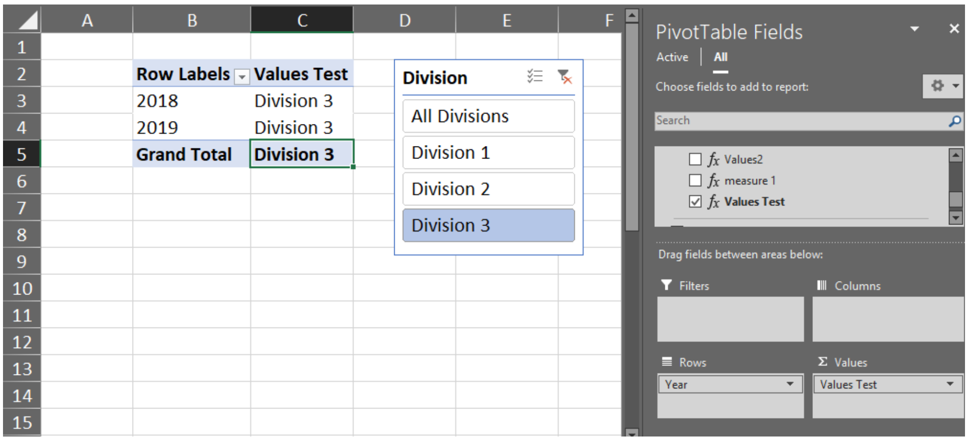
Formula:

```
=VALUES('Division Types'[Division])
```

Formatting Options

Category:
- General
- Number
- Currency
- Date
- TRUE/FALSE

The PivotTable will return with the single column value from '**Division Type**':



Row Labels	Values Test
2018	Division 3
2019	Division 3
Grand Total	Division 3

Division
- All Divisions
- Division 1
- Division 2
- Division 3

PivotTable Fields

Choose fields to add to report:

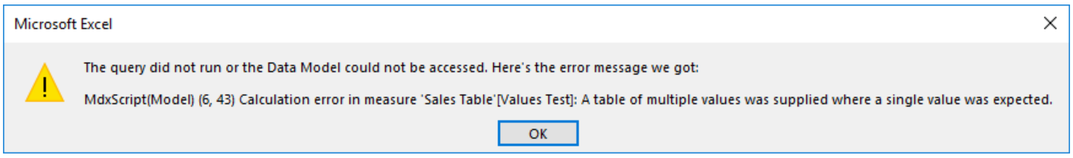
- ☐ Values2
- ☐ measure 1
- ☑ Values Test

Drag fields between areas below:

Rows: Year

Values: Values Test

Working in Excel, the **VALUES** function will return the following error message when there is no filter on the slicer:



Microsoft Excel

The query did not run or the Data Model could not be accessed. Here's the error message we got:

MdxScript(Model) (6, 43) Calculation error in measure 'Sales Table'[Values Test]: A table of multiple values was supplied where a single value was expected.

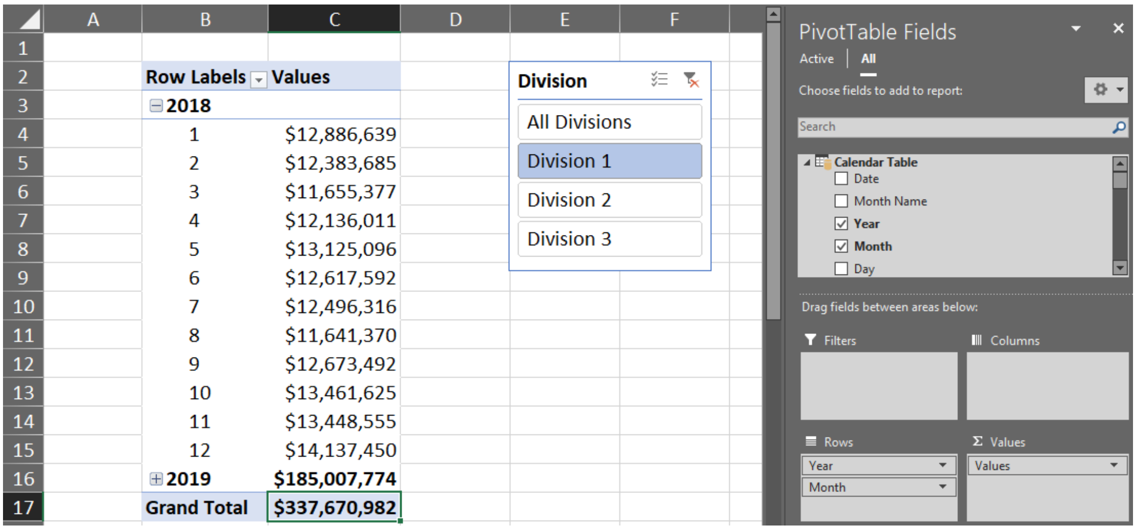
Moving on to the Power Pivot solution, we replace the **SELECTEDVALUE** segment of code from earlier:

```
=VAR DivisionNo =
        SELECTEDVALUE(‘Division Type’[Division], [All Divisions])

RETURN

SWITCH(TRUE(),
        DivisionNo = "Division 1", [Division 1 Sales],
        DivisionNo = "Division 2", [Division 2 Sales],
        DivisionNo = "Division 3", [Division 3 Sales],
        [All Divisions])
```

with the first piece of code detailed above to yield:

```
=VAR DivisionNo =
        IF(HASONEVALUE('Division Types'[Division]),
                VALUES('Division Types'[Division]),[Division 1 Sales]
        )

RETURN

SWITCH(TRUE(),
        DivisionNo = "Division 1", [Division 1 Sales],
        DivisionNo = "Division 2", [Division 2 Sales],
        DivisionNo = "Division 3", [Division 3 Sales],
        [All Divisions])
```

Assuming that we have imported all of the data into our data model and created the relevant relationships between our tables and slicers, we can construct the following PivotTable:



Row Labels	Values
⊟2018	
1	$12,886,639
2	$12,383,685
3	$11,655,377
4	$12,136,011
5	$13,125,096
6	$12,617,592
7	$12,496,316
8	$11,641,370
9	$12,673,492
10	$13,461,625
11	$13,448,555
12	$14,137,450
⊞2019	$185,007,774
Grand Total	$337,670,982

Division

- All Divisions
- Division 1
- Division 2
- Division 3

PivotTable Fields

Active | All

Choose fields to add to report:

Search

Calendar Table
- □ Date
- □ Month Name
- ☑ Year
- ☑ Month
- □ Day

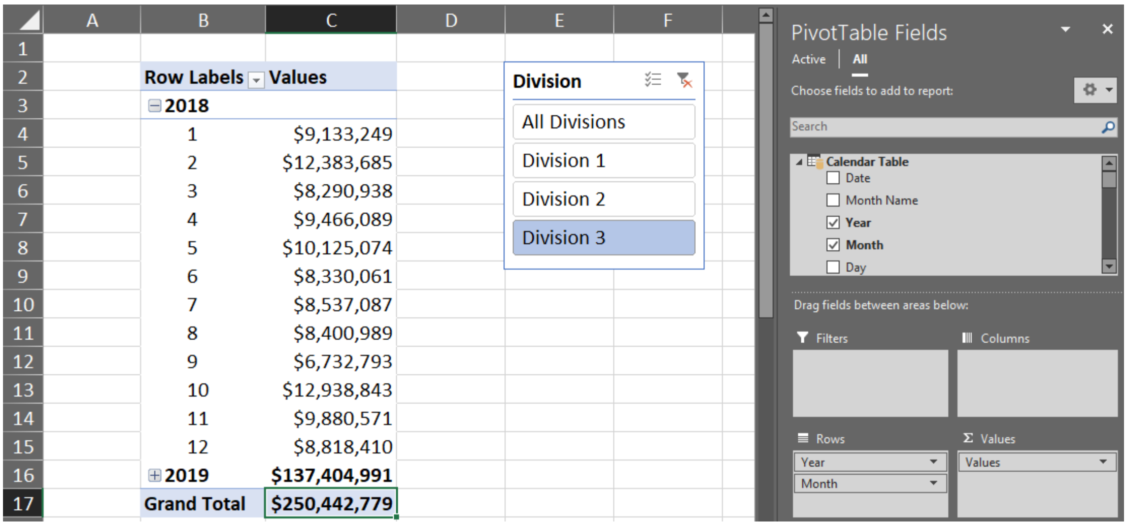
Drag fields between areas below:

Filters | Columns

Rows: Year, Month

Values: Values

Changing the selection on the slicer will yield the following:



Row Labels	Values
⊟2018	
1	$9,133,249
2	$12,383,685
3	$8,290,938
4	$9,466,089
5	$10,125,074
6	$8,330,061
7	$8,537,087
8	$8,400,989
9	$6,732,793
10	$12,938,843
11	$9,880,571
12	$8,818,410
⊞2019	$137,404,991
Grand Total	$250,442,779

Division: All Divisions, Division 1, Division 2, Division 3 (selected)

PivotTable Fields

Calendar Table
- □ Date
- □ Month Name
- ☑ Year
- ☑ Month
- □ Day

Rows: Year, Month

Values: Values

Active
All
Choose fields to add to report:
Search
Drag fields between areas below:
Filters
Columns
Rows
Values

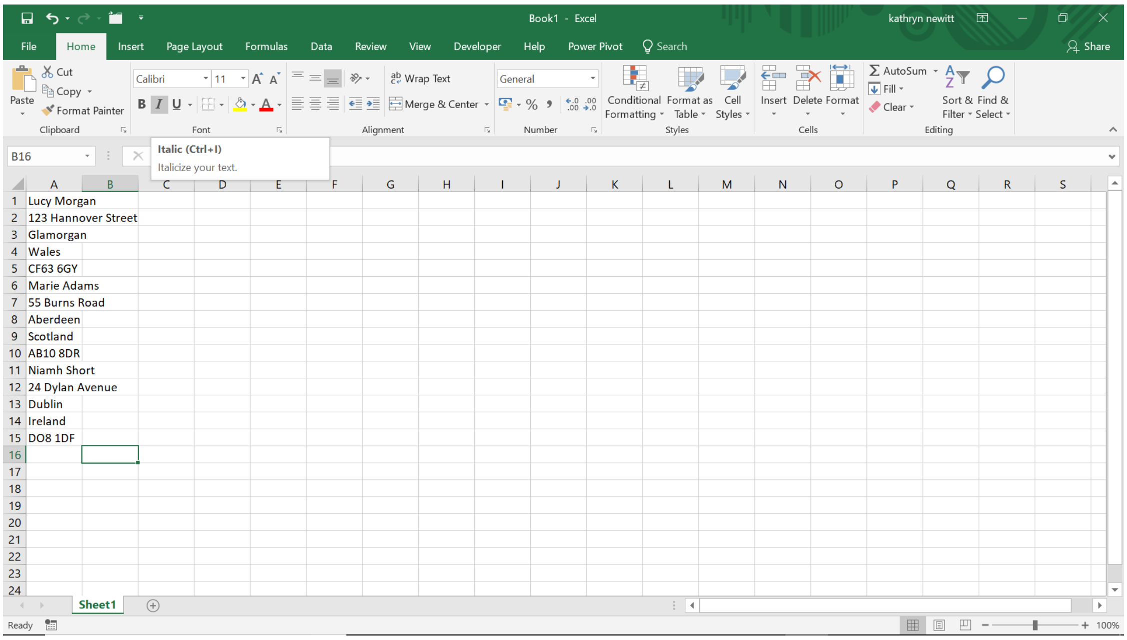
We can now dynamically toggle between divisions in our PivotTable in Excel.

More *Power Pivot Principles* next month.

Power Query Pointers

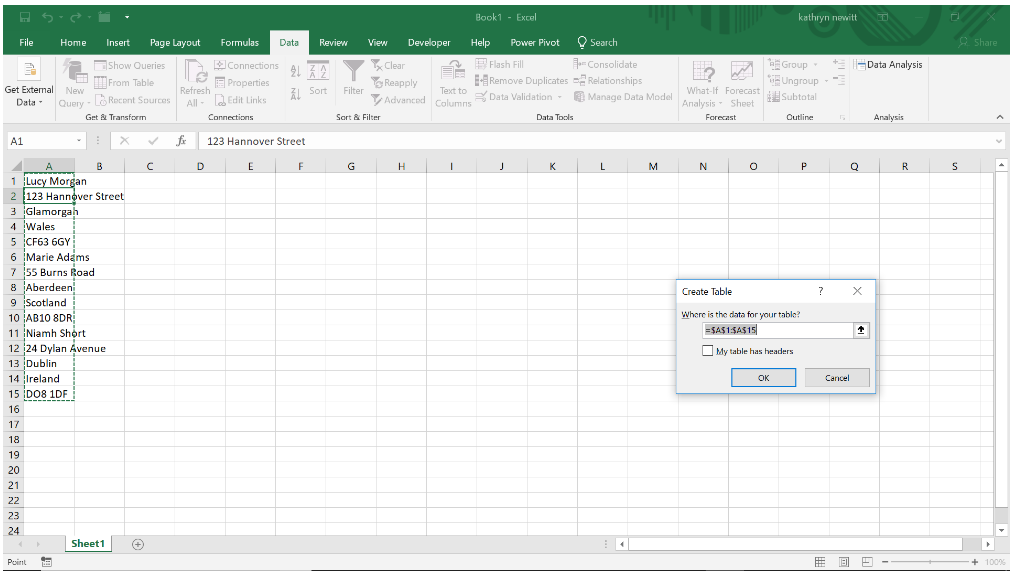
Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Office 365, Excel 2016 and 2019) from www.sumproduct.com/blog. If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month, we look at an example where data comes in one Excel column and needs to be converted into a table.

Not all data arrives in Excel in nicely organised tables. John the imaginary salesperson has sent in some new contacts, which he has copied to a worksheet:



	A
1	Lucy Morgan
2	123 Hannover Street
3	Glamorgan
4	Wales
5	CF63 6GY
6	Marie Adams
7	55 Burns Road
8	Aberdeen
9	Scotland
10	AB10 8DR
11	Niamh Short
12	24 Dylan Avenue
13	Dublin
14	Ireland
15	DO8 1DF

We have a list of names and addresses in a column. We would like them to be in a table, where we may extract the name, address, country and post code (AKA zip code). Let's start by creating a new query from my data using the 'From Table' option in the 'Get & Transform' section on the Data tab.



A1 | 123 Hannover Street

	A
1	Lucy Morgan
2	123 Hannover Street
3	Glamorgan
4	Wales
5	CF63 6GY
6	Marie Adams
7	55 Burns Road
8	Aberdeen
9	Scotland
10	AB10 8DR
11	Niamh Short
12	24 Dylan Avenue
13	Dublin
14	Ireland
15	DO8 1DF

Create Table

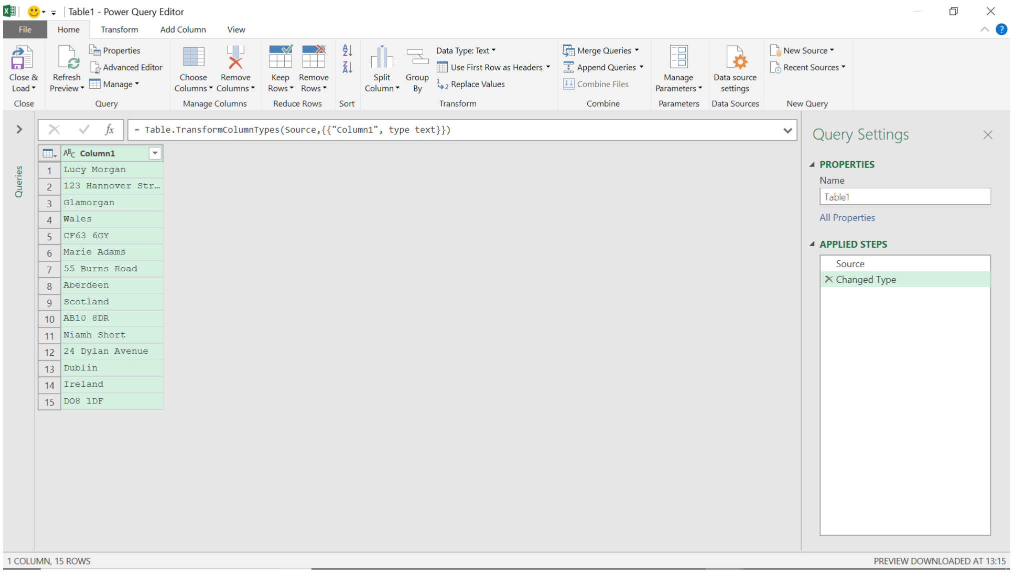
Where is the data for your table?

=$A$1:$A$15

□ My table has headers

OK | Cancel

Power Query confirms where our table is, and whether it has headers (no, thanks to John, it doesn't!). The default looks fine so let's click 'OK'.



= Table.TransformColumnTypes(Source,{{"Column1", type text}})

	Column1
1	Lucy Morgan
2	123 Hannover Str…
3	Glamorgan
4	Wales
5	CF63 6GY
6	Marie Adams
7	55 Burns Road
8	Aberdeen
9	Scotland
10	AB10 8DR
11	Niamh Short
12	24 Dylan Avenue
13	Dublin
14	Ireland
15	DO8 1DF

Query Settings

PROPERTIES

Name

Table1

All Properties

APPLIED STEPS

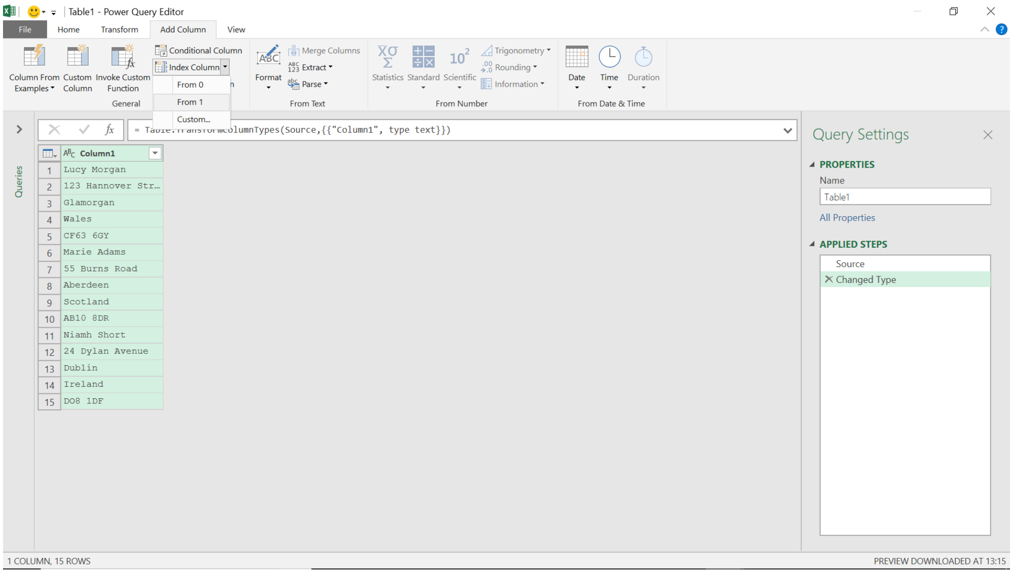
Source

Changed Type

1 COLUMN, 15 ROWS

PREVIEW DOWNLOADED AT 13:15

Our data is extracted, and now we may set about transforming it into a useful table. Since the data is grouped into five [5] rows for each address (and in this case we may rely upon this being consistent because it comes from a database which ensures this), it would be useful to count which row we are on. We can do this by creating an index column. In the 'Add Column' tab, let's choose 'Index Column' in the 'General' section, and start my column from one [1].



= Table.TransformColumnTypes(Source,{{"Column1", type text}})

	Column1
1	Lucy Morgan
2	123 Hannover Str…
3	Glamorgan
4	Wales
5	CF63 6GY
6	Marie Adams
7	55 Burns Road
8	Aberdeen
9	Scotland
10	AB10 8DR
11	Niamh Short
12	24 Dylan Avenue
13	Dublin
14	Ireland
15	DO8 1DF

Query Settings

PROPERTIES

Name

Table1

All Properties

APPLIED STEPS

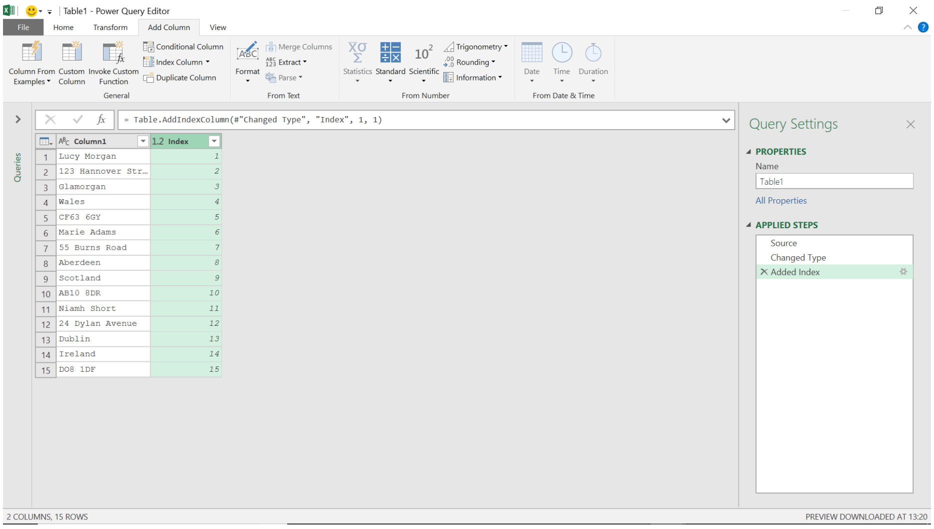
Source

Changed Type

1 COLUMN, 15 ROWS

PREVIEW DOWNLOADED AT 13:15

Having created this column, we now have some mathematical possibilities.



```
= Table.AddIndexColumn(#"Changed Type", "Index", 1, 1)
```

	Column1	Index
1	Lucy Morgan	1
2	123 Hannover Str...	2
3	Glamorgan	3
4	Wales	4
5	CF63 6GY	5
6	Marie Adams	6
7	55 Burns Road	7
8	Aberdeen	8
9	Scotland	9
10	AB10 8DR	10
11	Niamh Short	11
12	24 Dylan Avenue	12
13	Dublin	13
14	Ireland	14
15	D08 1DF	15

Name: Table1

APPLIED STEPS:
- Source
- Changed Type
- Added Index

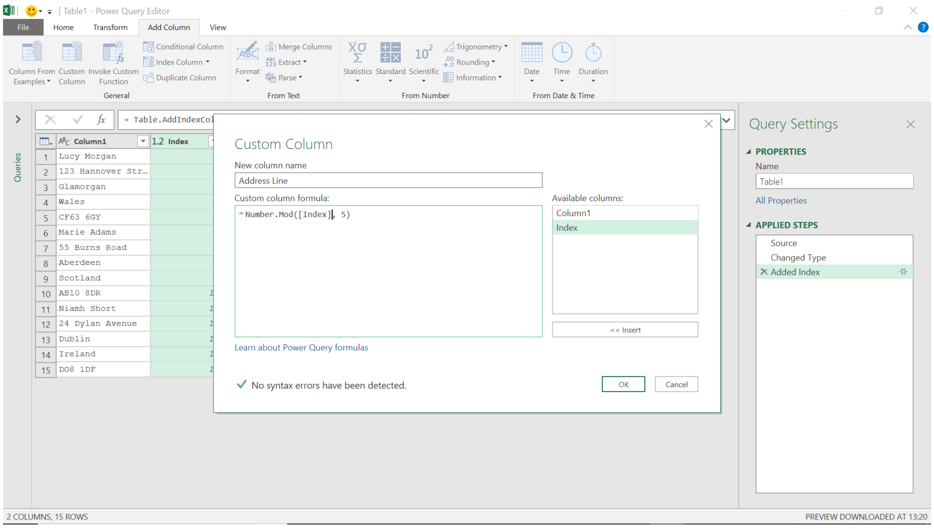
2 COLUMNS, 15 ROWS

PREVIEW DOWNLOADED AT 13:20

We're going to use the function **Number.Mod()** to determine where each address starts. **Number.Mod()** divides one number by another number and gives the remainder. This is similar to the **MOD** function in Excel (please see https://www.sumproduct.com/thought/a-modicum-of-mod for more details):

Number.Mod(number as nullable number**, divisor** as nullable number**, optional precision** as nullable number**)** as nullable number

It's much clearer to see it in practice. To see where each address starts, let's divide the index by five [5] and look at the remainder. Here, we'll create a new 'Custom Column' *viz.*



Custom Column

New column name: Address Line

Custom column formula:

```
= Number.Mod([Index], 5)
```

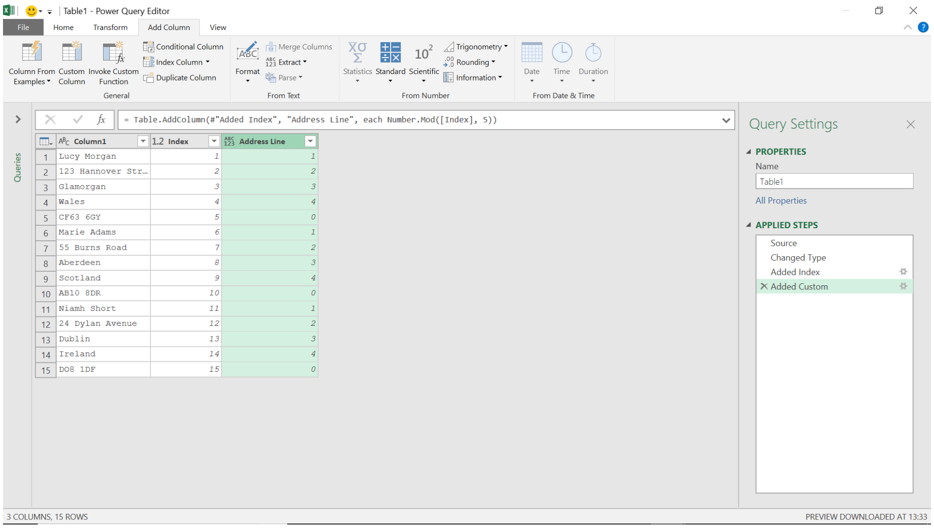
Available columns:
- Column1
- Index

✓ No syntax errors have been detected.

2 COLUMNS, 15 ROWS

PREVIEW DOWNLOADED AT 13:20

Click 'OK' to create a new column.



```
= Table.AddColumn(#"Added Index", "Address Line", each Number.Mod([Index], 5))
```

	Column1	Index	Address Line
1	Lucy Morgan	1	1
2	123 Hannover Str...	2	2
3	Glamorgan	3	3
4	Wales	4	4
5	CF63 6GY	5	0
6	Marie Adams	6	1
7	55 Burns Road	7	2
8	Aberdeen	8	3
9	Scotland	9	4
10	AB10 8DR	10	0
11	Niamh Short	11	1
12	24 Dylan Avenue	12	2
13	Dublin	13	3
14	Ireland	14	4
15	D08 1DF	15	0

Name: Table1

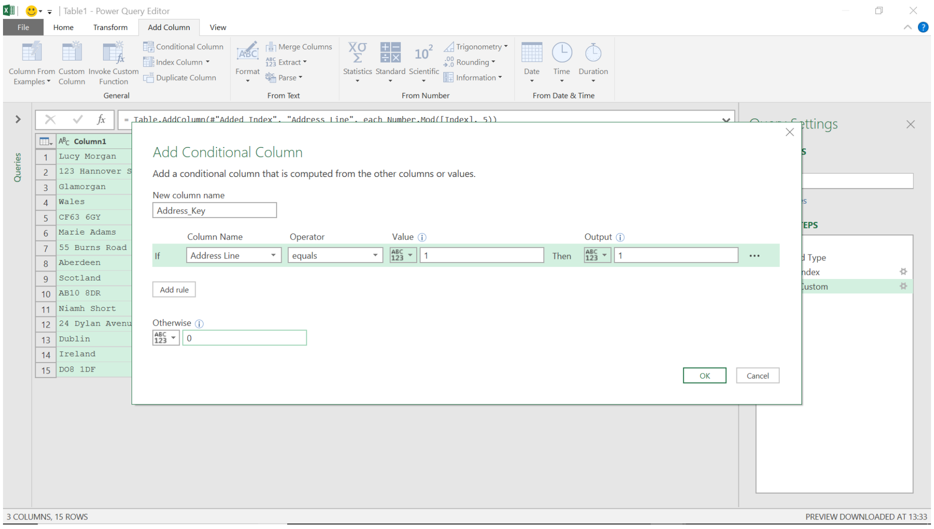
APPLIED STEPS:
- Source
- Changed Type
- Added Index
- Added Custom

3 COLUMNS, 15 ROWS

PREVIEW DOWNLOADED AT 13:33

The aim is to indicate which address lines belong together, by giving them the same value. We're going to do this with a running total (for more details on how to create a running total, please see https://www.sumproduct.com/blog/article/power-query-blogs/power-query-one-route-to-a-running-total).

The first step is to only count at the beginning of each address, and to do this we create another column, which this time is a 'Conditional Column':



```
= Table.AddColumn(#"Added Index", "Address Line", each Number.Mod([Index], 5))
```

Add Conditional Column

Add a conditional column that is computed from the other columns or values.

New column name: Address_Key

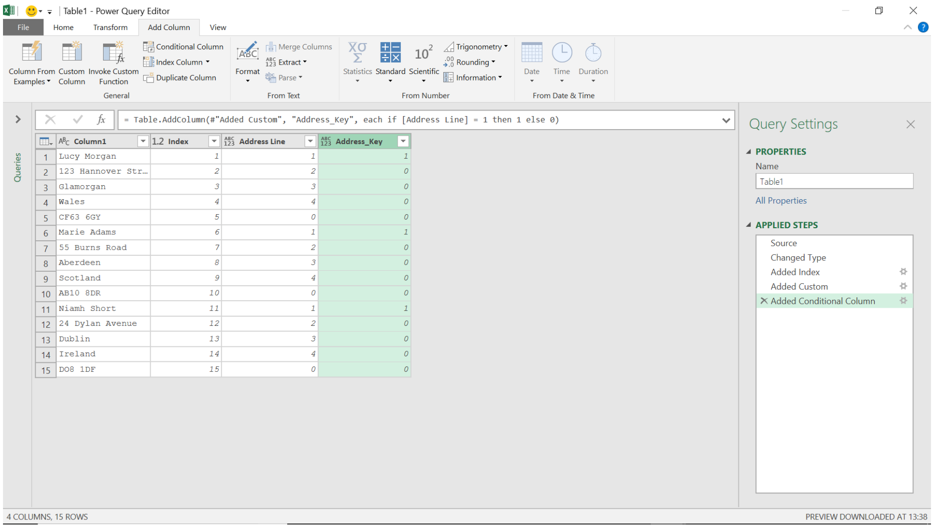
	Column Name	Operator	Value		Output
If	Address Line	equals	1	Then	1

Add rule

Otherwise: 0

OK Cancel

This column will be one [1] for the first line and zero [0] for the rest.



```
= Table.AddColumn(#"Added Custom", "Address_Key", each if [Address Line] = 1 then 1 else 0)
```

	Column1	Index	Address Line	Address_Key
1	Lucy Morgan	1	1	1
2	123 Hannover Str...	2	2	0
3	Glamorgan	3	3	0
4	Wales	4	4	0
5	CF63 6GY	5	0	0
6	Marie Adams	6	1	1
7	55 Burns Road	7	2	0
8	Aberdeen	8	3	0
9	Scotland	9	4	0
10	AB10 8DR	10	0	0
11	Niamh Short	11	1	1
12	24 Dylan Avenue	12	2	0
13	Dublin	13	3	0
14	Ireland	14	4	0
15	DO8 1DF	15	0	0

Query Settings

PROPERTIES

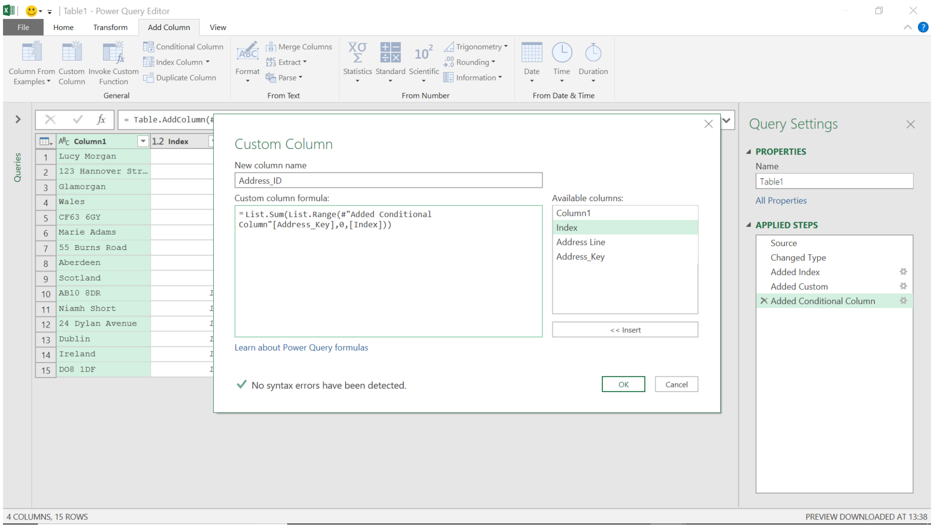
Name: Table1

All Properties

APPLIED STEPS

- Source
- Changed Type
- Added Index
- Added Custom
- Added Conditional Column

We can now create another 'Custom Column' for our running total.



Custom Column

New column name: Address_ID

Custom column formula:

```
= List.Sum(List.Range(#"Added Conditional
Column"[Address_Key],0,[Index]))
```

Available columns:

- Column1
- Index
- Address Line
- Address_Key

<< Insert

Learn about Power Query formulas

✓ No syntax errors have been detected.

OK Cancel

Query Settings

PROPERTIES

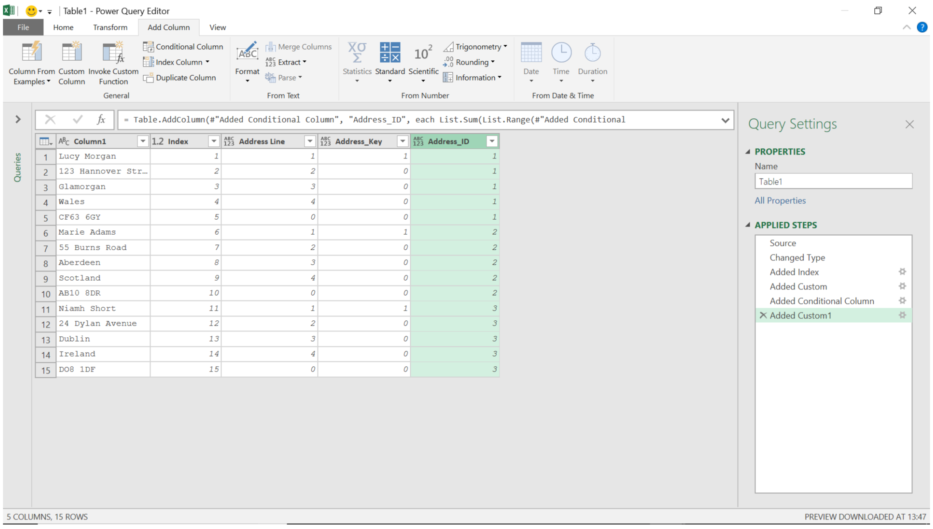
Name: Table1

All Properties

APPLIED STEPS

- Source
- Changed Type
- Added Index
- Added Custom
- Added Conditional Column

This provides us with the same value for each line belonging to the same address.



```
= Table.AddColumn(#"Added Conditional Column", "Address_ID", each List.Sum(List.Range(#"Added Conditional
```

	Column1	Index	Address Line	Address_Key	Address_ID
1	Lucy Morgan	1	1	1	1
2	123 Hannover Str…	2	2	0	1
3	Glamorgan	3	3	0	1
4	Wales	4	4	0	1
5	CF63 6GY	5	0	0	1
6	Marie Adams	6	1	1	2
7	55 Burns Road	7	2	0	2
8	Aberdeen	8	3	0	2
9	Scotland	9	4	0	2
10	AB10 8DR	10	0	0	2
11	Niamh Short	11	1	1	3
12	24 Dylan Avenue	12	2	0	3
13	Dublin	13	3	0	3
14	Ireland	14	4	0	3
15	DO8 1DF	15	0	0	3

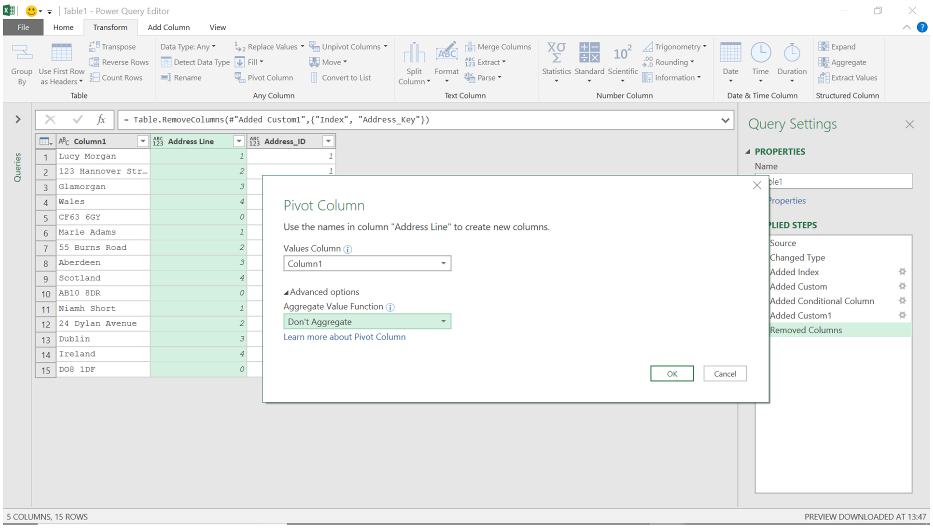
Name: Table1

Applied Steps:
- Source
- Changed Type
- Added Index
- Added Custom
- Added Conditional Column
- Added Custom1

5 COLUMNS, 15 ROWS
PREVIEW DOWNLOADED AT 13:47

Now we have a way to identify each address, we’re ready to pivot the data. We no longer need the **Index** or **Address_Key** columns, so we may remove these first.

On the Transform tab, select **Address_Line** and choose to pivot our data. The values will be in **Column1** and we do not wish to aggregate them.



```
= Table.RemoveColumns(#"Added Custom1",{"Index", "Address_Key"})
```

	Column1	Address Line	Address_ID
1	Lucy Morgan	1	1
2	123 Hannover Str…	2	1
3	Glamorgan	3	
4	Wales	4	
5	CF63 6GY	0	
6	Marie Adams	1	
7	55 Burns Road	2	
8	Aberdeen	3	
9	Scotland	4	
10	AB10 8DR	0	
11	Niamh Short	1	
12	24 Dylan Avenue	2	
13	Dublin	3	
14	Ireland	4	
15	DO8 1DF	0	

Pivot Column

Use the names in column "Address Line" to create new columns.

Values Column: Column1

Advanced options

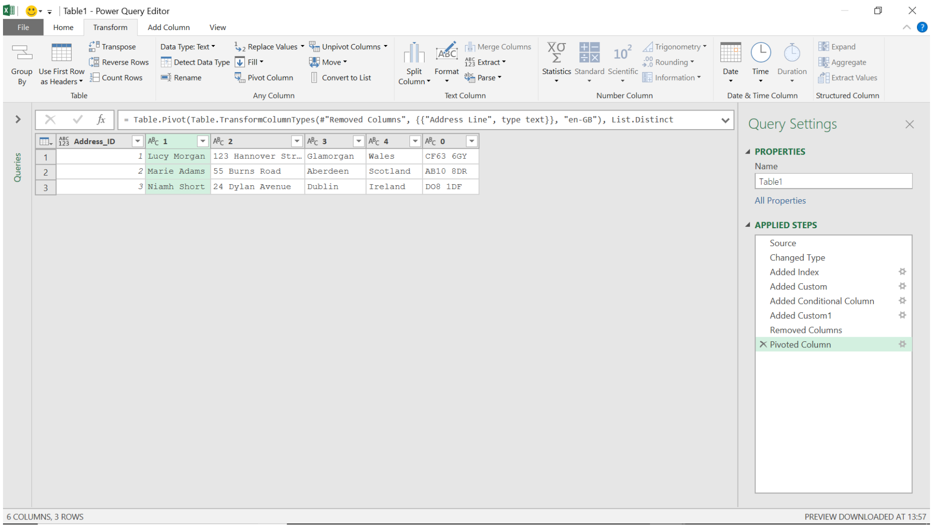
Aggregate Value Function: Don't Aggregate

Name: ble1

Applied Steps:
- Source
- Changed Type
- Added Index
- Added Custom
- Added Conditional Column
- Added Custom1
- Removed Columns

5 COLUMNS, 15 ROWS
PREVIEW DOWNLOADED AT 13:47

Clicl ‘OK’ to see the data.



```
= Table.Pivot(Table.TransformColumnTypes(#"Removed Columns", {{"Address Line", type text}}, "en-GB"), List.Distinct
```

	Address_ID	1	2	3	4	0
1	1	Lucy Morgan	123 Hannover Str…	Glamorgan	Wales	CF63 6GY
2	2	Marie Adams	55 Burns Road	Aberdeen	Scotland	AB10 8DR
3	3	Niamh Short	24 Dylan Avenue	Dublin	Ireland	DO8 1DF

Name: Table1

Applied Steps:
- Source
- Changed Type
- Added Index
- Added Custom
- Added Conditional Column
- Added Custom1
- Removed Columns
- Pivoted Column

6 COLUMNS, 3 ROWS
PREVIEW DOWNLOADED AT 13:57

This is looking much better. Our **Address_ID** column is populated correctly, so we just need to rename our other columns.

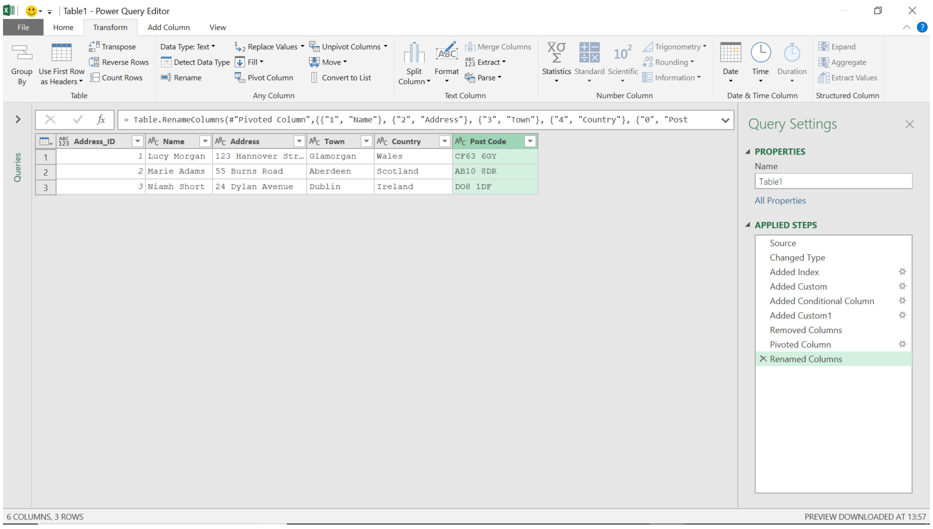


Table1 - Power Query Editor

```
= Table.RenameColumns(#"Pivoted Column",{{"1", "Name"}, {"2", "Address"}, {"3", "Town"}, {"4", "Country"}, {"0", "Post
```

	Address_ID	Name	Address	Town	Country	Post Code
1	1	Lucy Morgan	123 Hannover Str...	Glamorgan	Wales	CF63 6GY
2	2	Marie Adams	55 Burns Road	Aberdeen	Scotland	AB10 8DR
3	3	Niamh Short	24 Dylan Avenue	Dublin	Ireland	DO8 1DF

Query Settings

PROPERTIES

Name: Table1

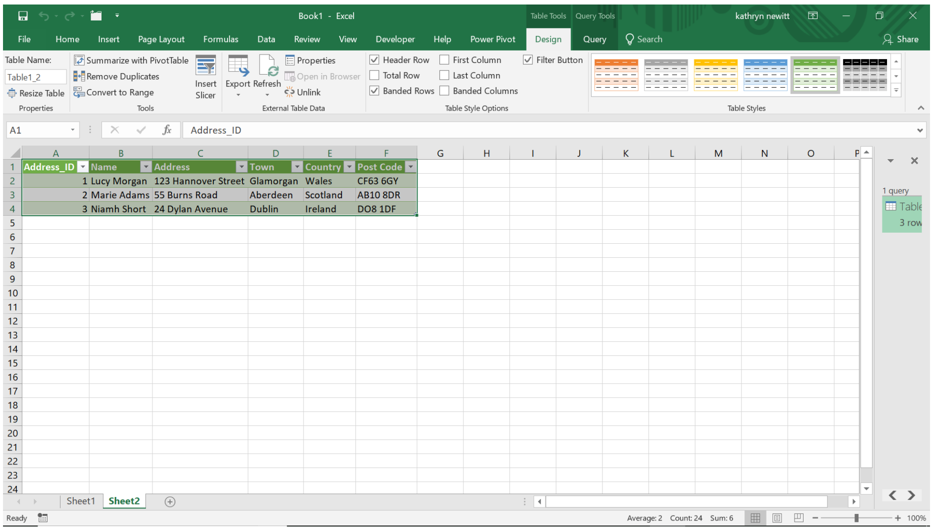
APPLIED STEPS

- Source
- Changed Type
- Added Index
- Added Custom
- Added Conditional Column
- Added Custom1
- Removed Columns
- Pivoted Column
- Renamed Columns

6 COLUMNS, 3 ROWS

PREVIEW DOWNLOADED AT 13:57

This data is now ready to 'Close & Load' to Excel from the File tab. If John uses the same Excel worksheet and adds more addresses (or updates any existing ones) then they may be refreshed and updated in the table too.



Book1 - Excel

Table Name: Table1_2

A1: Address_ID

	A	B	C	D	E	F
1	Address_ID	Name	Address	Town	Country	Post Code
2	1	Lucy Morgan	123 Hannover Street	Glamorgan	Wales	CF63 6GY
3	2	Marie Adams	55 Burns Road	Aberdeen	Scotland	AB10 8DR
4	3	Niamh Short	24 Dylan Avenue	Dublin	Ireland	DO8 1DF

Average: 2 Count: 24 Sum: 6

More next month.

Power BI Updates

Frustratingly for our 10th anniversary edition, no Power BI updates were announced in time for our publishing deadline. That's a bit of a damp squib, I know, but I am sure normal service will be resumed shortly.

New Features for Excel

The latest updates see the Navigation pane is now Generally Available for Excel for Windows, and for Windows and Mac Insiders, Automate Tasks with Office Scripts enables you to automate repetitive tasks in your Excel work. There are other additions / improvements too, with the full list as follows:

Excel for the web

- Quickly 'Find the Command' you need

Excel for Windows

- Automate Tasks with Office Scripts (Insiders Beta)
- Navigation pane
- Quickly 'Find the Command' you need
- New DAX Functions in Excel Data Models and Power Pivot (Insiders Beta)
- New Paste Options when using keyboard shortcuts (Insiders Beta)

Excel for Mac

- Automate Tasks with Office Scripts (Insiders Beta).

Let's plough through.

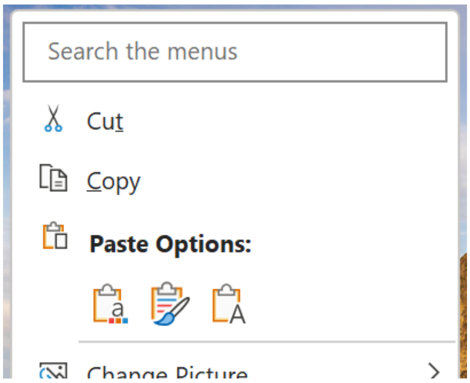
Quickly 'Find the Command' you need

In Excel for the web and Excel for Windows (on an Insiders Preview basis), it has just become easier to find that feature or function (*i.e.* "command") you need using the shortcut menu. This should save you time for other stuff and allow you to focus on the key problems at hand.

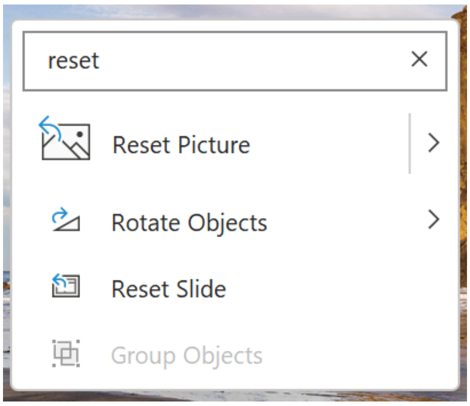
It is ever so simple to use:

- Right-click an area or object in your document, worksheet or presentation (alternatively, you may use the menu key on your keyboard or press **SHIFT + F10** to display the shortcut menu too)

Type the first few letters of the command you are looking for in the Search box at the top of the shortcut menu



- Click the command you want, or use the arrow keys to select it, and then press **ENTER**.


reset

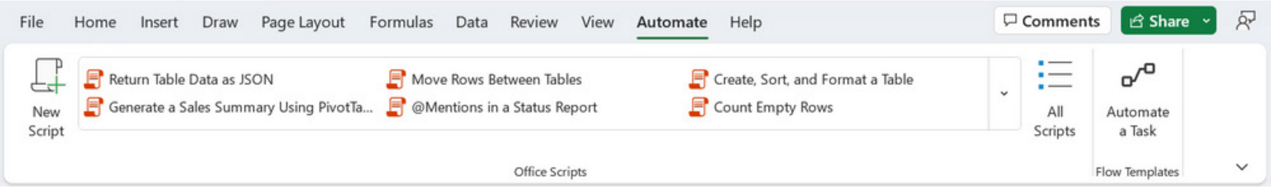
Erm, that's it. This feature is currently available on the following platforms and in the following applications:

- **Web:** Excel, PowerPoint and Word
- **Windows:** Excel, PowerPoint and Word for Office Insiders running Beta Channel and Current Channel (Preview) Version 2206 (Build 15331.20010) or later.

It may take a short while to roll out so don't panic if you don't see it immediately.

Automate Tasks with Office Scripts (Insiders Preview)

Office Scripts enables you to automate repetitive tasks in your spreadsheet workflows and automate repetitive tasks in your spreadsheet workflow and automate repetitive tasks in your spreadsheet workflow…

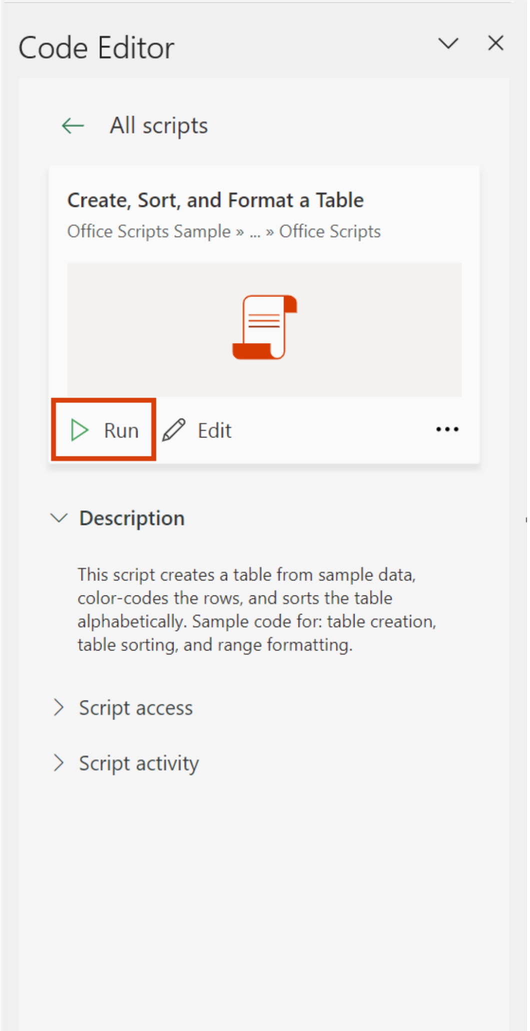


These scripts help you save time by increasing efficiency and reducing errors in your workflow, and you can even schedule your scripts so that your automations can run even while you're away.

Up until now, you could only create these helpful scripts in Excel on the web. With this update, you can now create and modify scripts in Excel for Windows and Excel for Mac (in Insiders Beta) using the Code Editor, as well as view and run scripts in the 'All Scripts' task pane. This work represents the first stage of bringing Office Scripts across all Excel platforms, but we don't imagine VBA will be quaking in its boots just yet.

To view and run scripts:

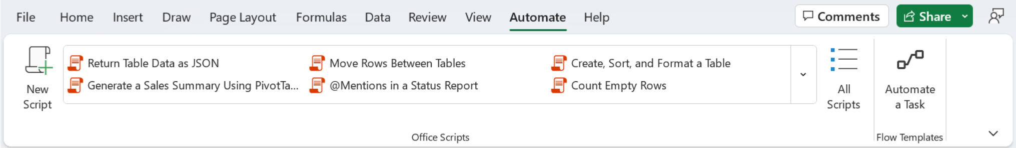
- open any workbook in Excel for Windows or for Mac and select the Automate tab
- to run an existing script, select one in the gallery or in the 'All Scripts' task pane
- in the 'All Scripts' task pane, select any script in the gallery or click the 'All Scripts' button to view more detail
- to run the script you've selected, click the Run button on the script's detail page.


Code Editor
All scripts
Create, Sort, and Format a Table
Office Scripts Sample » ... » Office Scripts
Run
Edit
Description
Script access
Script activity

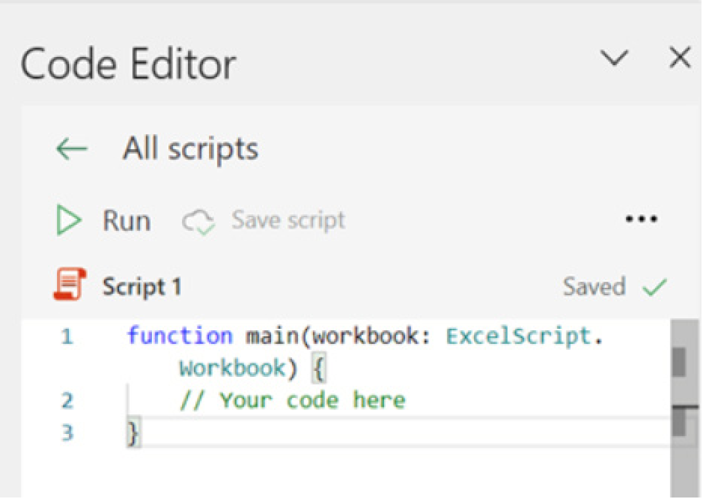
This script creates a table from sample data, color-codes the rows, and sorts the table alphabetically. Sample code for: table creation, table sorting, and range formatting.

To create and modify scripts:

- open any workbook in Excel for Windows or for Mac and navigate to the Automate tab. You should note that all the scripts in your workbook are available as well as various Microsoft samples



- you may start making your own scripts by selecting the ‘New Script’ button



```
function main(workbook: ExcelScript.Workbook) {
    // Your code here
}
```

- to modify an existing script, select Edit on the script’s details page or select the pencil icon by hovering over any script in the ‘All Scripts’ task pane.

As this is new, there are some known issues / restrictions:

- the ‘Action Recorder’, which records the actions you take on a worksheet and saves them into a script for you, remains a web-only feature at this time (that is a shame unfortunately)
- Office Scripts in Excel for Mac does not support scripts saved to SharePoint at this time
- the Ribbon buttons do not work when logged in with a non-work / school account, or when completely logged out. To fix this issue, switch to a work or school account as your primary identity (there is no need to restart Excel).

To use Office Scripts in Excel for Windows or Excel for Mac, you must have the following:

- a commercial (E3 / E5) licence for Microsoft 365
- for Windows users only: Microsoft Edge WebView2 installed.

This feature is available to Office Insiders running:

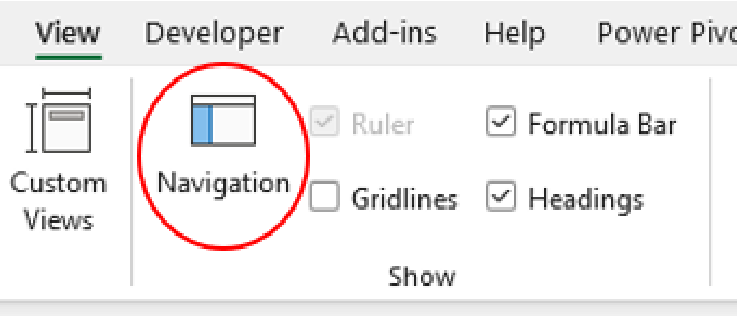
- **Windows:** version 2209 (Build 15726.20000) or later
- **Mac:** version 16.66 (Build 22092601) or later.

Navigation pane

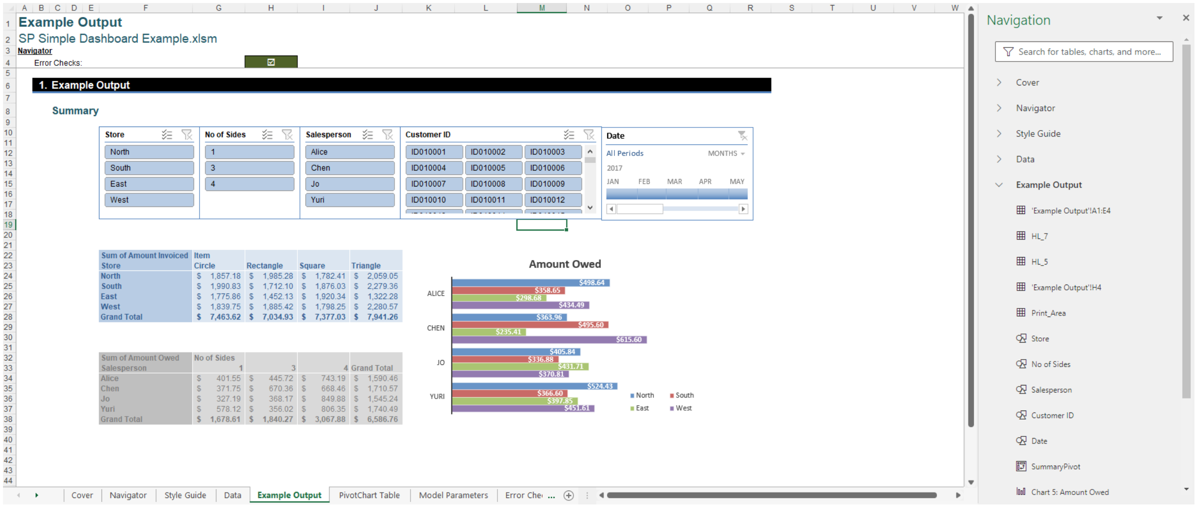
The Navigation pane in Excel is an easy way to understand a workbook's layout, see what elements exist within the workbook and navigate directly to those elements. Whether you're a new user getting familiar with Excel or an experienced user trying to navigate a large workbook, the Navigation pane may prove useful.

With this pane, you can find and access elements such as tables, charts, PivotTables and images within your workbook. Navigation will no longer be such a “pane”. Once you've opened the Navigation pane, it displays on the right side of the Excel window. The Navigation pane also makes it easier for those with visual impairments to access all parts of the workbook. It can improve how tools such as screen readers interpret your workbook.

Simply click on the ‘Navigation’ button in the ‘Show’ grouping on the View tab of the Ribbon, *viz.*



You can then navigate through the workbook with ease:



Example Output

SP Simple Dashboard Example.xlsm

Navigator

Error Checks: ☑

1. Example Output

Summary

Sum of Amount Invoiced	Item			
Store	Circle	Rectangle	Square	Triangle
North	$ 1,857.18	$ 1,985.28	$ 1,782.41	$ 2,059.05
South	$ 1,990.83	$ 1,712.10	$ 1,876.03	$ 2,279.36
East	$ 1,775.86	$ 1,452.13	$ 1,920.34	$ 1,322.28
West	$ 1,839.75	$ 1,885.42	$ 1,798.25	$ 2,280.57
Grand Total	$ 7,463.62	$ 7,034.93	$ 7,377.03	$ 7,941.26

Sum of Amount Owed	No of Sides			
Salesperson	1	3	4	Grand Total
Alice	$ 401.55	$ 445.72	$ 743.19	$ 1,590.46
Chen	$ 371.75	$ 670.36	$ 668.46	$ 1,710.57
Jo	$ 327.19	$ 368.17	$ 849.88	$ 1,545.24
Yuri	$ 578.12	$ 356.02	$ 806.35	$ 1,740.49
Grand Total	$ 1,678.61	$ 1,840.27	$ 3,067.88	$ 6,586.76

This new feature is part of Microsoft's grand scheme to make their products inclusive and accessible to everyone, including people with disabilities. It's clear that the Navigation pane will assist in:

- ensuring Excel works seamlessly with assistive technology
- building inclusive experiences for people with disabilities
- supporting authors in creating accessible content.

New DAX Functions in Excel Data Models and Power Pivot (Insiders Beta)

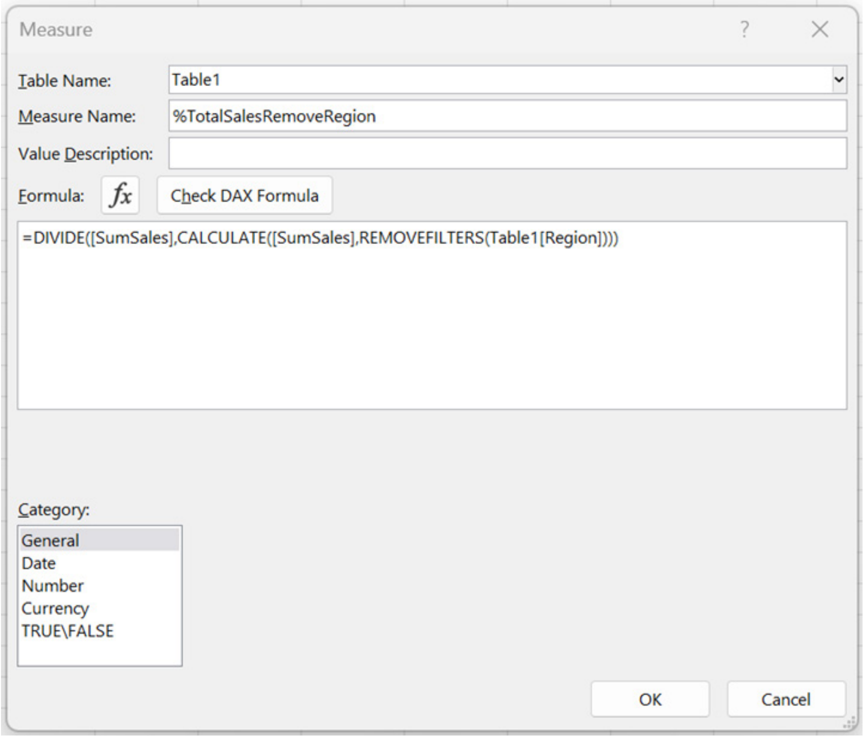


Table Name: Table1

Measure Name: %TotalSalesRemoveRegion

Value Description:

Formula:

```
=DIVIDE([SumSales],CALCULATE([SumSales],REMOVEFILTERS(Table1[Region])))
```

Category: General

Excel's Data Model and Power Pivot are tools that help you analyse data across multiple tables and support a powerful expression language called Data Analysis eXpressions (DAX). Using the features of the Data Model, you can quickly relate various data sets to create PivotTables and Pivot Charts, using a similar technology to that employed by Power BI.

As of these updates, you may now use 92 new DAX functions when creating measures in Power Pivot. This provides you with new capabilities to analyse, visualise and report on your data. At the time of writing, there are presently 353 DAX functions available, but a significant number were not recognised in Power Pivot. Adding these functions into the Excel realm brings Power Pivot closer to parity with its counterpart features in Power BI.

The new functions are as follows:

1. **ACCRINT**
2. **ACCRINTM**
3. **ALLCROSSFILTERED**
4. **AMORDEGRC**
5. **AMORLINC**
6. **BITAND**
7. **BITLSHIFT**
8. **BITOR**
9. **BITRSHIFT**
10. **BITXOR**
11. **COALESCE**
12. **COLUMNSTATISTICS**
13. **COMBINEVALUES**
14. **CONTAINSSTRING**
15. **CONTAINSSTRINGEXACT**
16. **CONVERT**
17. **COUPDAYBS**
18. **COUPDAYS**
19. **COUPDAYSNC**
20. **COUPNCD**
21. **COUPNUM**
22. **COUPPCD**
23. **CUMIPMT**
24. **CUMPRINC**
25. **DB**
26. **DDB**
27. **DISC**
28. **DISTINCTCOUNTNOBLANK**
29. **DOLLARDE**
30. **DOLLARFR**
31. **DURATION**
32. **EFFECT**
33. **ERROR**
34. **FIRSTNONBLANKVALUE**
35. **FV**
36. **GENERATESERIES**
37. **IF.EAGER**
38. **INTRATE**
39. **IPMT**
40. **ISAFTER**
41. **ISINSCOPE**
42. **ISPMT**
43. **ISSELECTEDMEASURE**
44. **LASTNONBLANKVALUE**
45. **MDURATION**
46. **NAMEOF**
47. **NOMINAL**
48. **NONVISUAL**
49. **NORM.DIST**
50. **NORM.INV**
51. **NORM.S.DIST**
52. **NORM.S.INV**
53. **NPER**
54. **ODDFPRICE**
55. **ODDFYIELD**
56. **ODDLPRICE**
57. **ODDLYIELD**
58. **PDURATION**
59. **PMT**
60. **PPMT**
61. **PRICE**
62. **PRICEDISC**
63. **PRICEMAT**
64. **PV**
65. **QUARTER**
66. **RATE**
67. **RECEIVED**
68. **REMOVEFILTERS**
69. **RRI**
70. **SELECTEDMEASURE**
71. **SELECTEDMEASUREFORMATSTRING**
72. **SELECTEDMEASURENAME**
73. **SELECTEDVALUE**
74. **SLN**
75. **SYD**
76. **T.DIST**
77. **T.DIST.2T**
78. **T.DIST.RT**
79. **T.INV**
80. **T.INV.2T**
81. **TBILLEQ**
82. **TBILLPRICE**
83. **TBILLYIELD**
84. **TREATAS**
85. **UNICHAR**
86. **USEROBJECTID**
87. **UTCNOW**
88. **UTCTODAY**
89. **VDB**
90. **YIELD**
91. **YIELDDISC**
92. **YIELDMAT.**

It is recognised elsewhere in this newsletter **SELECTEDVALUE** is mentioned (see ***Power Pivot Principles*** for more details) – good to see our advice will be out of date *very* soon!!

Presently, this feature is available to Beta Channel users running Version 2208 (Build 15504.10000) or later. It should be noted that Data Models may only be created in Excel for Windows and Power Pivot is a COM add-in and therefore must be installed.

New Paste Options when using keyboard shortcuts (Insiders Beta)

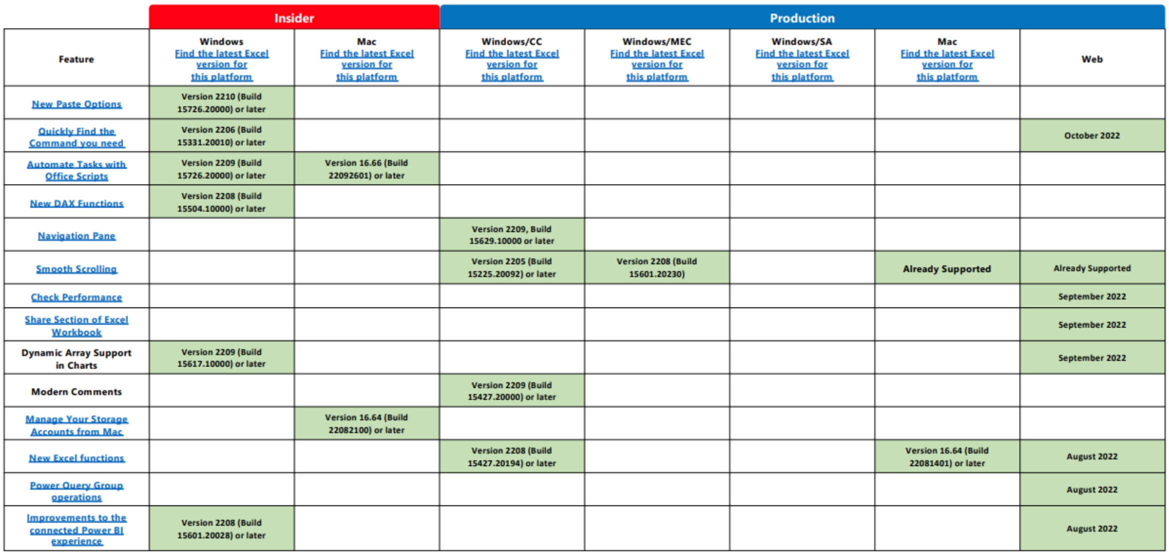
It's common to copy text from a web page, an email or another document and then paste it into Excel. Often, your goal is to match the format that's already in place in the cell or range where you're pasting. Now you can simply copy the content that you want to paste into an existing worksheet and press **CTRL + SHIFT + V**.

A lot of the keyboard shortcut fraternity have become very excited by this one (I must confess this has passed me by a little). It's not to say you couldn't do this previously, it's just it took more steps, *e.g.*

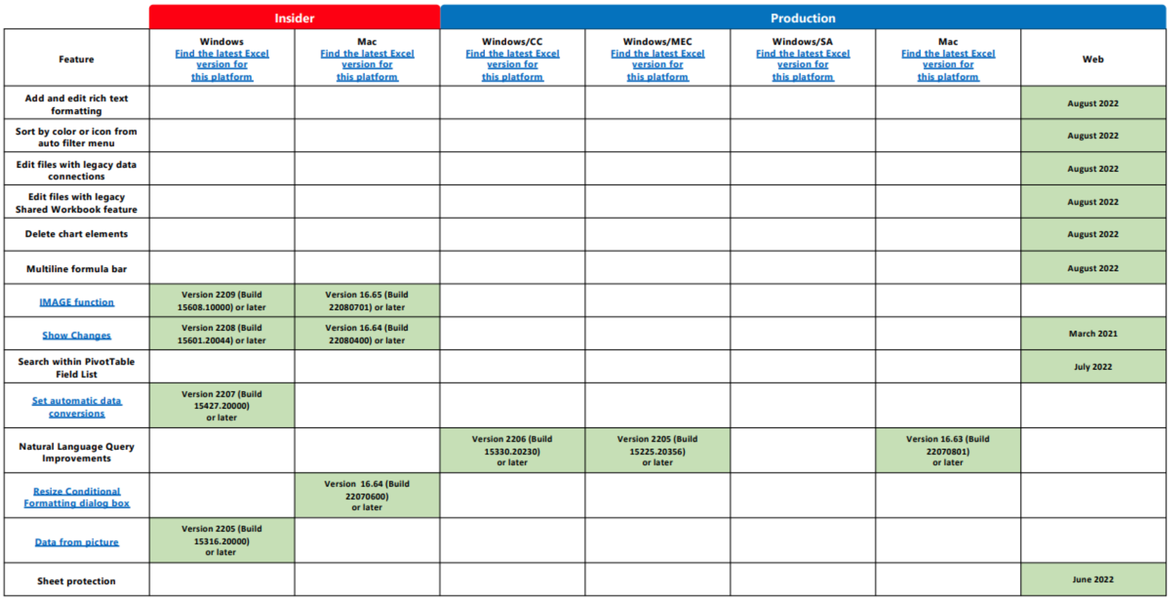
- Paste the copied content by pressing **CTRL + V**, and then select 'Paste Values' or 'Match Destination Formatting' in the 'Paste Options' dialog box
- Select 'Paste Values' or 'Match Destination Formatting' in the Paste drop-down menu in the Ribbon (press **ALT + H + V** to open the menu with the keyboard)
- Press **CTRL + ALT + V** to open the Paste Special dialog, choose 'Values' or 'Text', and then press **ENTER**.

Goodness me, those long winter nights will just fly by now.

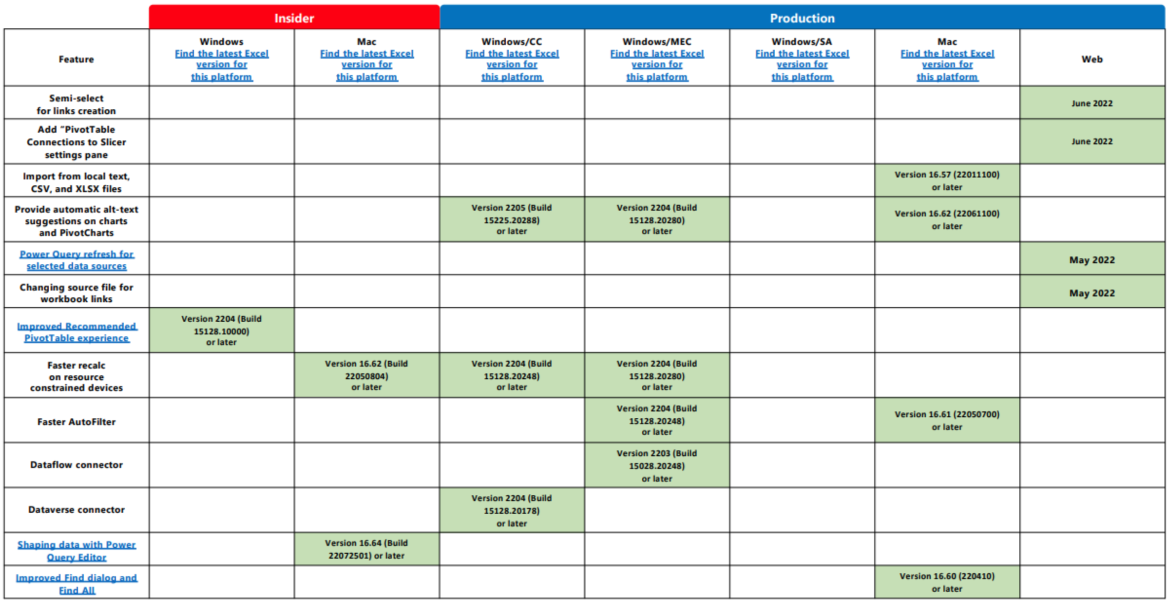
The updated version of the grid with all the new features is fast becoming too complicated to show here. Nonetheless, you can find the interactive links at aka.ms/ExcelFeaturesFlyer.



	Insider		Production				
Feature	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Web
New Paste Options	Version 2210 (Build 15726.20000) or later						
Quickly Find the Command you need	Version 2206 (Build 15331.20010) or later						October 2022
Automate Tasks with Office Scripts	Version 2209 (Build 15726.20000) or later	Version 16.66 (Build 22092601) or later					
New DAX Functions	Version 2208 (Build 15504.10000) or later						
Navigation Pane			Version 2209, Build 15629.10000 or later				
Smooth Scrolling			Version 2205 (Build 15225.20092) or later	Version 2208 (Build 15601.20230)		Already Supported	Already Supported
Check Performance							September 2022
Share Section of Excel Workbook							September 2022
Dynamic Array Support in Charts	Version 2209 (Build 15617.10000) or later						September 2022
Modern Comments			Version 2209 (Build 15427.20000) or later				
Manage Your Storage Accounts from Mac		Version 16.64 (Build 22082100) or later					
New Excel functions			Version 2208 (Build 15427.20194) or later			Version 16.64 (Build 22081401) or later	August 2022
Power Query Group operations							August 2022
Improvements to the connected Power BI experience	Version 2208 (Build 15601.20028) or later						August 2022



	Insider		Production				
Feature	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Web
Add and edit rich text formatting							August 2022
Sort by color or icon from auto filter menu							August 2022
Edit files with legacy data connections							August 2022
Edit files with legacy Shared Workbook feature							August 2022
Delete chart elements							August 2022
Multiline formula bar							August 2022
IMAGE function	Version 2209 (Build 15608.10000) or later	Version 16.65 (Build 22080701) or later					
Show Changes	Version 2208 (Build 15601.20044) or later	Version 16.64 (Build 22080400) or later					March 2021
Search within PivotTable Field List							July 2022
Set automatic data conversions	Version 2207 (Build 15427.20000) or later						
Natural Language Query Improvements			Version 2206 (Build 15330.20230) or later	Version 2205 (Build 15225.20356) or later		Version 16.63 (Build 22070801) or later	
Resize Conditional Formatting dialog box		Version 16.64 (Build 22070600) or later					
Data from picture	Version 2205 (Build 15316.20000) or later						
Sheet protection							June 2022



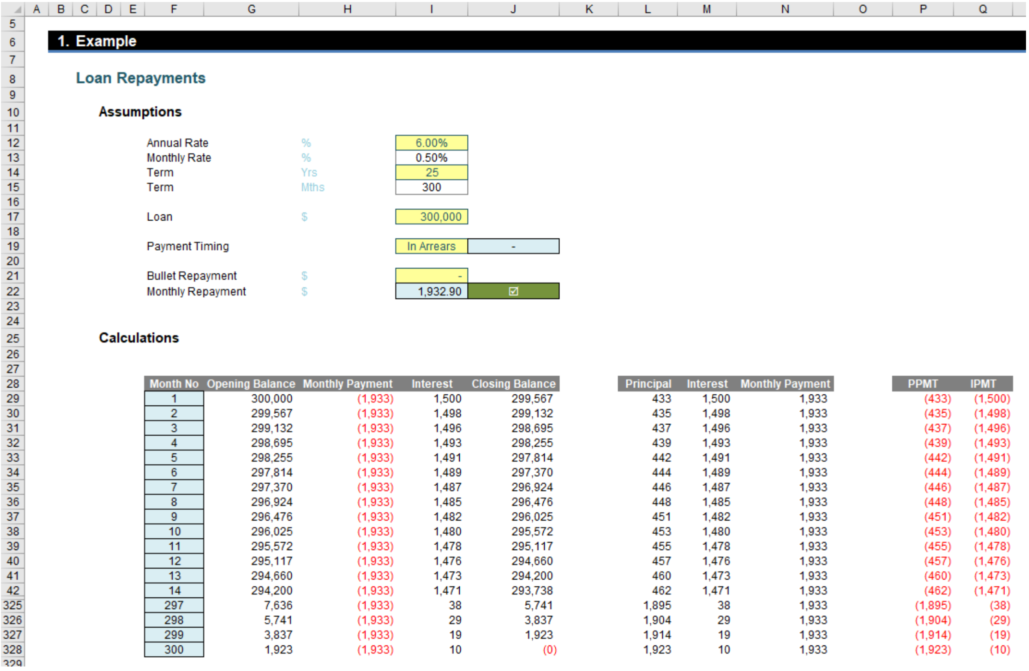
	Insider		Production				
Feature	Windows Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Windows/CC Find the latest Excel version for this platform	Windows/MEC Find the latest Excel version for this platform	Windows/SA Find the latest Excel version for this platform	Mac Find the latest Excel version for this platform	Web
Semi-select for links creation							June 2022
Add "PivotTable Connections to Slicer settings pane							June 2022
Import from local text, CSV, and XLSX files						Version 16.57 (22011100) or later	
Provide automatic alt-text suggestions on charts and PivotCharts			Version 2205 (Build 15225.20288) or later	Version 2204 (Build 15128.20280) or later		Version 16.62 (22061100) or later	
Power Query refresh for selected data sources							May 2022
Changing source file for workbook links							May 2022
Improved Recommended PivotTable experience	Version 2204 (Build 15128.10000) or later						
Faster recalc on resource constrained devices		Version 16.62 (Build 22050804) or later	Version 2204 (Build 15128.20248) or later	Version 2204 (Build 15128.20280) or later			
Faster AutoFilter				Version 2204 (Build 15128.20248) or later		Version 16.61 (22050700) or later	
Dataflow connector				Version 2203 (Build 15028.20248) or later			
Dataverse connector			Version 2204 (Build 15128.20178) or later				
Shaping data with Power Query Editor		Version 16.64 (Build 22072501) or later					
Improved Find dialog and Find All						Version 16.60 (220410) or later	

More next month, we're sure.

The A to Z of Excel Functions: IPMT

Imagine I were to borrow $300,000 over 25 years at an interest rate of 6% p.a. Assuming no final amount to pay (*i.e.* no bullet repayment) and payments were made monthly at the end of each month ("in arrears"), interest would accrue over the month at 6%/12 = 0.50% per month (since there would be no compounding of interest monthly as it would be paid each month and I will simplify that all months are of equal length).

Using Goal Seek, the **PMT** function or algebraic methods, I could soon determine the monthly payment would be $1,932.90:



1. Example

Loan Repayments

Assumptions

Annual Rate	%	6.00%	
Monthly Rate	%	0.50%	
Term	Yrs	25	
Term	Mths	300	
Loan	$	300,000	
Payment Timing		In Arrears	-
Bullet Repayment	$	-	
Monthly Repayment	$	1,932.90	☑

Calculations

Month No	Opening Balance	Monthly Payment	Interest	Closing Balance	Principal	Interest	Monthly Payment	PPMT	IPMT
1	300,000	(1,933)	1,500	299,567	433	1,500	1,933	(433)	(1,500)
2	299,567	(1,933)	1,498	299,132	435	1,498	1,933	(435)	(1,498)
3	299,132	(1,933)	1,496	298,695	437	1,496	1,933	(437)	(1,496)
4	298,695	(1,933)	1,493	298,255	439	1,493	1,933	(439)	(1,493)
5	298,255	(1,933)	1,491	297,814	442	1,491	1,933	(442)	(1,491)
6	297,814	(1,933)	1,489	297,370	444	1,489	1,933	(444)	(1,489)
7	297,370	(1,933)	1,487	296,924	446	1,487	1,933	(446)	(1,487)
8	296,924	(1,933)	1,485	296,476	448	1,485	1,933	(448)	(1,485)
9	296,476	(1,933)	1,482	296,025	451	1,482	1,933	(451)	(1,482)
10	296,025	(1,933)	1,480	295,572	453	1,480	1,933	(453)	(1,480)
11	295,572	(1,933)	1,478	295,117	455	1,478	1,933	(455)	(1,478)
12	295,117	(1,933)	1,476	294,660	457	1,476	1,933	(457)	(1,476)
13	294,660	(1,933)	1,473	294,200	460	1,473	1,933	(460)	(1,473)
14	294,200	(1,933)	1,471	293,738	462	1,471	1,933	(462)	(1,471)
297	7,636	(1,933)	38	5,741	1,895	38	1,933	(1,895)	(38)
298	5,741	(1,933)	29	3,837	1,904	29	1,933	(1,904)	(29)
299	3,837	(1,933)	19	1,923	1,914	19	1,933	(1,914)	(19)
300	1,923	(1,933)	10	(0)	1,923	10	1,933	(1,923)	(10)

You can see that over the 300 months the outstanding balance reduces to zero from an initial loan of $300,000. The monthly payments (column **H**) remain constant, but the interest reduces as it calculates the opening balance (for payments in arrears, *i.e.* the repayment is not included) multiplied by the monthly interest rate, *e.g.* for cell **I29**, interest is calculated as

=G29*$I$13

As long as you calculate this table, the formulae are simple. But what if you don't want to have to generate this time every time you wanted to know the interest element of the monthly payment for a given month? You may use the **IPMT** function, which will give the same solution, but be negative instead. This is because Excel's financial functions distinguish between cash inflows (positive) and outflows (negative).

It employs the following syntax to operate:

IPMT(rate, per, nper, pv, [fv], [type])

The **IPMT** function has the following arguments:

- **rate:** this is required and represents the constant interest rate for the loan
- **per:** this is required, and specified the period to be considered, between periods 1 and **nper**
- **nper:** this is also required and denotes the total number of payments for the loan
- **pv:** also necessary, this is the present value, or the total amount that a series of future payments is worth now, also known as the principal (*i.e.* what you are borrowing)
- **fv:** this is the first of two optional arguments. This is the future value, or a cash balance you want to attain, after the last payment is made.
- If **fv** is omitted, it is assumed to be zero (0), *i.e.* the future value of a loan is nil
- **type:** this final argument is also optional. This the number zero (0) or one (1) and indicates when payments are due:

type	Payments due
0 or omitted	At the end of the period
1	At the beginning of the period

It should be further noted that:

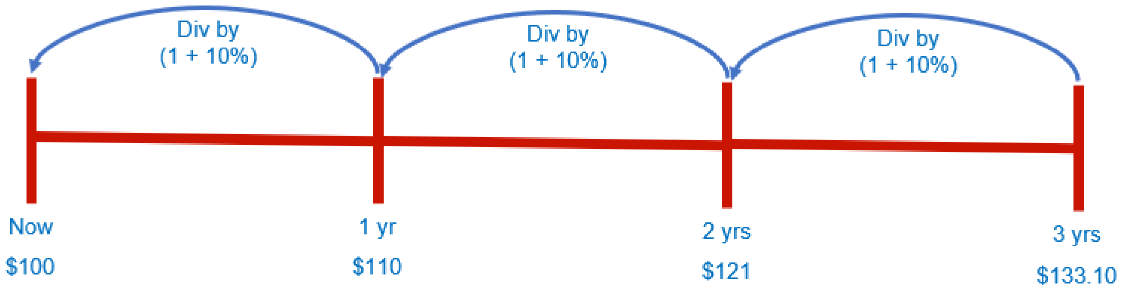
- the interest payment returned by **IPMT** relates to interest but considers no effect from taxes, reserve payments or other fees sometimes associated with loans
- make sure that you are consistent about the units you use for specifying **rate** and **nper**. If you make monthly payments on a four-year loan at an annual interest rate of 12%, use 12%/12 for **rate** and 4*12 for **nper**. If you make annual payments on the same loan, use 12% for the **rate** and 4 for **nper**.

The A to Z of Excel Functions: IRR

There are many techniques employed to value an asset, a project, a business, a shareholding, and so on. However, one is arguably more common than the rest these days – Net Present Value (NPV) using discounted cash flows.

A discounted cash flow (DCF) is a straightforward computation:

- Let's assume inflation is running at 10% (and we will assume this is after tax as we all earn our wages after tax and increases in spending affect this after-tax wage)
- Something that costs $100 this year will cost 10% more next year, *i.e.* $110
- Something that costs $110 next year will cost 10% more the year after, *i.e.* $121
- Something that costs $121 in that year will cost 10% more the following year, *i.e.* $133.10
- However, they are all worth the equivalent of $100 now (as we "discount" these *future values* back to their *present values*).



Note that all of these valuations are for a *point* of time not a *period*. This is a common mistake in modelling. We have to understand when we assume the cash flows will occur. The three most common assumptions are at the start, the middle or the end of the period in question. This assumption will obviously vary the overall valuation as a consequence.

Valuations include both cash inflows and cash outflows. Adding up all these positive and negative present values, provides a netted off total: the Net Present Value (NPV). The aim is to generate a positive return (a positive NPV) for a given rate of discounting, known as the discount rate.

The **Internal Rate of Return** (IRR) is the name given to the discount rate that makes the Net Present Value (NPV) of a range of cashflows zero. For example, if I invest \$100 now and receive \$121 back in two years' time this would give me an annual IRR of 10% since:

$$(\$100) + PV(\$121) = (\$100) + \$121 / (1 + 10\%)^2$$

$$= (\$100) + \$100$$

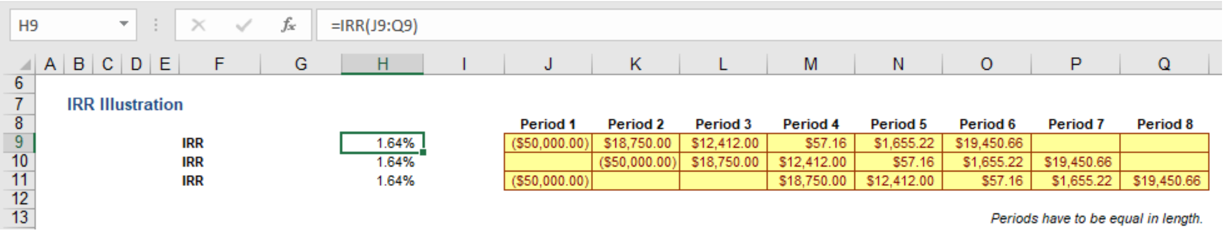
$$NPV \quad = 0$$

It is often used to calculate cost of capital hurdle rates, bond rates, discount rates implicit in leases and Compound Annual Growth Rates, amongst other tasks.

For cashflows with equal periodicity, Excel has a function that will estimate the IRR:

- **IRR(values,** [guess]**)** returns the periodic IRR for a set of sequential cashflows occurring on a regular periodic basis. There must be at least one positive and one negative value in the range. **IRR** will then cycle through an iterative technique (up to 20 times) to try and find an answer which is accurate to within 0.00001%
- If the guess is not specified, Excel will assume that it is 10% (0.1).

Both are fraught with problems in practice and the associated Excel file provides several documented examples. However, before we discuss some of the problems with the functions, let me compare and contrast with a simple example:



IRR Illustration

		Period 1	Period 2	Period 3	Period 4	Period 5	Period 6	Period 7	Period 8
IRR	1.64%	(\$50,000.00)	\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66		
IRR	1.64%		(\$50,000.00)	\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66	
IRR	1.64%	(\$50,000.00)			\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66

Periods have to be equal in length.

In the illustration above, we consider three slightly different cashflows. The first one has the old faithful "hockey stick" projections of an outflow followed by five successive cash inflows. In our example, this would give us a periodic IRR of 1.64%. If the periods were months and we wished to convert this to annual rate then, using the compounding formula, this would be:

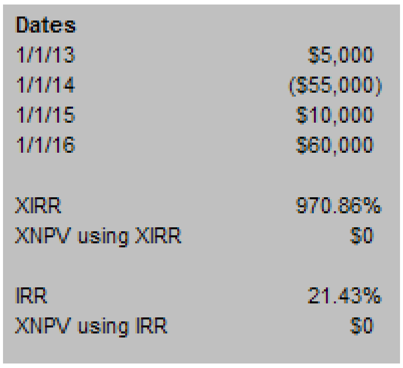
$$(1 + 1.64\%)^{12} -1 = 21.54\%.$$

The second example for IRR gives the same rate – which is correct. Start dates for IRR are irrelevant: the only thing that matters is how long after the initial cashflow subsequent flows occur.

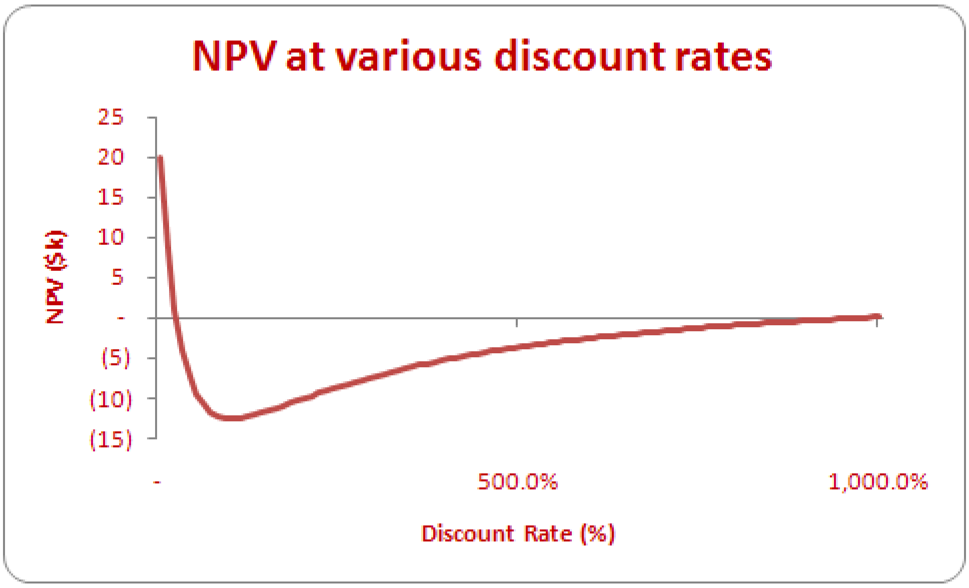
Unfortunately, the third scenario is not right and demonstrates a weakness in the IRR function. Here, the outflow still occurs in the first period, but then there is a two-month delay before the inflows are received. Clearly, this will impact both the NPV and the IRR. The IRR is unaffected however – *because blanks are ignored by the **IRR** function*. To get Excel to calculate the IRR correctly here, zeroes should be entered into the two blank cells.

More than One Solution

There can be more than one IRR. Every time a cashflow series changes sign (*i.e.* goes from positive to negative or vice versa) there is potentially another solution. Consider the following:



Dates	
1/1/13	\$5,000
1/1/14	(\$55,000)
1/1/15	\$10,000
1/1/16	\$60,000
XIRR	970.86%
XNPV using XIRR	\$0
IRR	21.43%
XNPV using IRR	\$0



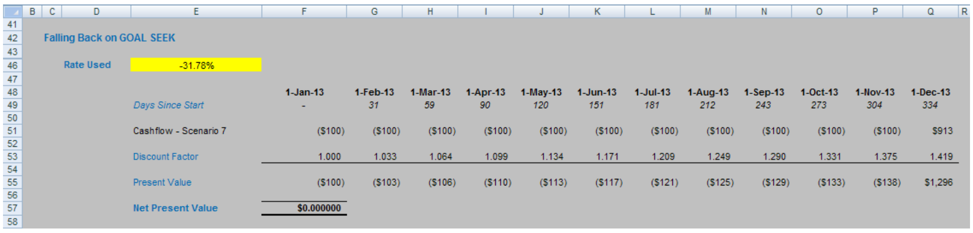
Here, prompted by a guess in the sister **XIRR** function (albeit of the other solution 21.43%), **XIRR** and **IRR** return two different IRRs associated with this cashflow scenario.

It is important to not only check that an IRR gives an NPV of zero [0] but that it is also the correct one in the circumstances.

So What Would You Use Instead?

There are times when accuracy is paramount, *e.g.* bond issues involving large sums of money. Excel will only calculate **IRR** on a maximum cycle of 20 times. Goal Seek, on the other hand, can calculate on a cycle of up to 32,767 iterations (this may be changed in Excel Options, **ALT + T + O**) with greater accuracy.

Therefore, when it is business critical, I always use Goal Seek to calculate IRRs. It is a very simple approach and uses the longhand checking approach described above. For example:



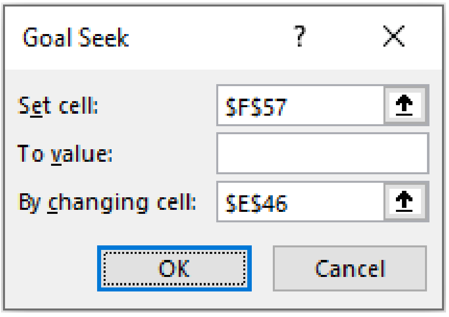
Falling Back on GOAL SEEK

Rate Used -31.78%

	F	G	H	I	J	K	L	M	N	O	P	Q
	1-Jan-13	1-Feb-13	1-Mar-13	1-Apr-13	1-May-13	1-Jun-13	1-Jul-13	1-Aug-13	1-Sep-13	1-Oct-13	1-Nov-13	1-Dec-13
Days Since Start	-	*31*	*59*	*90*	*120*	*151*	*181*	*212*	*243*	*273*	*304*	*334*
Cashflow - Scenario 7	($100)	($100)	($100)	($100)	($100)	($100)	($100)	($100)	($100)	($100)	($100)	$913
Discount Factor	1.000	1.033	1.064	1.099	1.134	1.171	1.209	1.249	1.290	1.331	1.375	1.419
Present Value	($100)	($103)	($106)	($110)	($113)	($117)	($121)	($125)	($129)	($133)	($138)	$1,296
Net Present Value	**$0.000000**											

Here, an NPV calculation is constructed long hand with the discount rate an input in cell **E46** (this is similar to an example in the attached Excel file).

Next, call up Goal Seek (**ALT + T + G**):



Set cell: $F$57

To value:

By changing cell: $E$46

The NPV (cell **F57** here) can be set to zero by Goal Seek, by changing the rate in cell **E46**. Once you have clicked 'OK' in the dialog box, the algorithm will cycle through to a solution where possible with the NPV displayed at the same time to confirm that the value obtained is indeed the IRR.

It may be a cumbersome method, but give me reliability over elegance any day.

More Excel Functions next month.

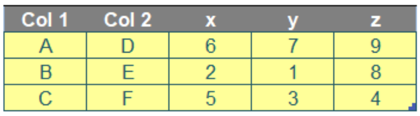
Beat the Boredom Suggested Solution

The challenge this month was to spill a range of cells that unpivot some last columns of an array. Easy, yes?

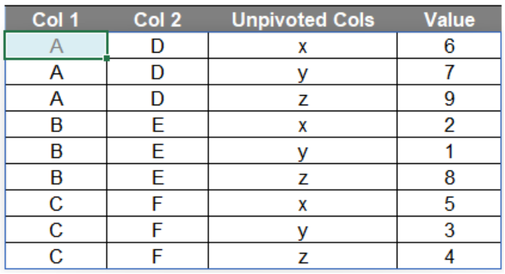
When you work with pivoted data that has a structure similar to a PivotTable, it is sometimes difficult to look up a value based on multiple column and row criteria. To make it simpler, you might unpivot the data. Most of the time, you may choose to use the useful 'Unpivot Columns' feature in Power Query, but this will require refreshing when the data is modified.

This challenge was designed to make you think outside the box to find another way using only Excel formulae, so that the results refresh automatically.

This month's challenge was to write a **formula in one cell** using dynamic arrays (see, we're nice – we gave you a hint!) that would spill a range of cells to unpivot only the last three [3] columns (*i.e.* **x**, **y** and **z**) of an array. The result should look like the array generated on the right based on the inputs adjacent and to the left *(below)*:



Col 1	Col 2	x	y	z
A	D	6	7	9
B	E	2	1	8
C	F	5	3	4



Col 1	Col 2	Unpivoted Cols	Value
A	D	x	6
A	D	y	7
A	D	z	9
B	E	x	2
B	E	y	1
B	E	z	8
C	F	x	5
C	F	y	3
C	F	z	4

As always, there were some requirements:

- the formula needed to be in just one cell (no "helper" cells)
- this was a formula challenge; no Power Query / Get & Transform or VBA!
- the formula needed to be flexible, so that if we adjusted the number of rows and / or columns of the input table, the formula should still work
- obviously, the numbers of rows / columns of the output table could not exceed the row / column limitations of Excel.

Suggested Solution

Before explaining our solution, we will clarify how we came up with it first. Firstly, inputs of the formula need to include:

- the **Data** table in the question
- the number of columns that will not be unpivoted, which is two [2] (*i.e.* **Col 1** and **Col 2**). We name it as **ColstoKeep**.

Therefore, the number of columns to unpivot is three [3], which is calculated as below. We name this number as **UCols**.

=COLUMNS(Data[#All]) – ColstoKeep

Secondly, we need to consider some features (*e.g*. numbers of rows and columns) of the output array. After we unpivot the table, they should be calculated as below:

- number of rows: nine [9]

=(ROWS(Data[#All]) - 1) * UCols

- number of columns: four [4]. The output table will include the first two [2] columns of initial table and two [2] additional columns for the old row headers (*i.e.* **x**, **y** and **z**) and values (*i.e.* numbers in **Data** table in this case).

=ColstoKeep + 2

Thirdly, to create a Dynamic Range for the output, we need the help of the **INDEX** (please refer to https://www.sumproduct.com/blog/article/a-to-z-of-excel-functions/the-index-function) and **SEQUENCE** (please refer to https://www.sumproduct.com/thought/creating-a-calendar-with-dynamic-arrays) functions.

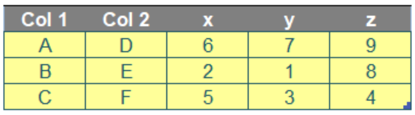
The row and column index numbers of output need to be created by **SEQUENCE** as follows. We will call them **RowID** and **ColID**.

- **RowID**:

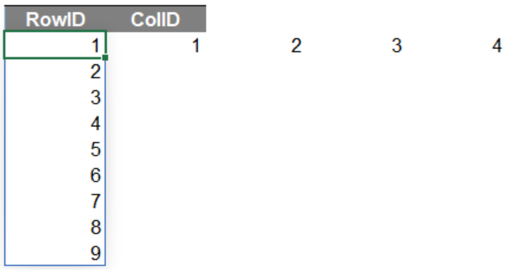
=SEQUENCE((ROWS(Data[#All]) - 1) * UCols)

- **ColID**:

=SEQUENCE(1, ColstoKeep + 2)



Col 1	Col 2	x	y	z
A	D	6	7	9
B	E	2	1	8
C	F	5	3	4

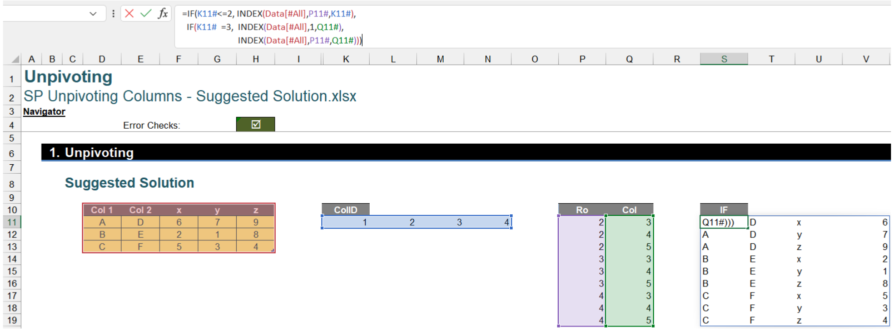


RowID	ColID			
1	1	2	3	4
2				
3				
4				
5				
6				
7				
8				
9				

Finally, the trick of this challenge is to use **ColID** with an **IF** statement (see https://www.sumproduct.com/blog/article/a-to-z-of-excel-functions/the-if-function for further details) *(below)* as a connector for three different **INDEX** functions, *i.e.*

"*If* **ColID** is less than or equal to **ColstoKeep**, *then* get the first two [2] columns of **Data**,
else if **ColID** is equal to **ColstoKeep** + 1, *then* get the Row Header of unpivoted columns of **Data**,
else get the Values of **Data**."

The result is as follows:



```
=IF(K11#<=2, INDEX(Data[#All],P11#,K11#),
 IF(K11# =3,  INDEX(Data[#All],1,Q11#),
              INDEX(Data[#All],P11#,Q11#)))
```

Unpivoting

SP Unpivoting Columns - Suggested Solution.xlsx

Navigator

Error Checks: ☑

1. Unpivoting

Suggested Solution

Col 1	Col 2	x	y	z
A	D	6	7	9
B	E	2	1	8
C	F	5	3	4

ColID			
1	2	3	4

Ro	Col
2	3
2	4
2	5
3	3
3	4
3	5
4	3
4	4
4	5

IF			
Q11#)))	D	x	6
A	D	y	7
A	D	z	9
B	E	x	2
B	E	y	1
B	E	z	8
C	F	x	5
C	F	y	3
C	F	z	4

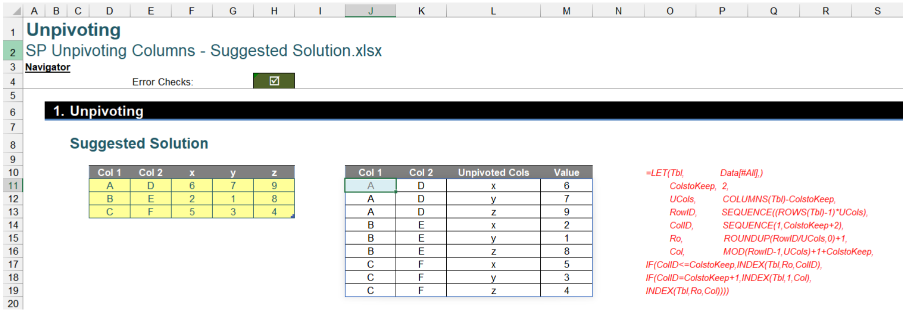
You may wonder why the challenge only allows a formula cell when there are several working steps above. Our solution is a combination of all described steps above within a **LET** formula (see https://www.sumproduct.com/blog/article/a-to-z-of-excel-functions/the-let-function) as follows:

=LET(Tbl, Data[#All],
ColstoKeep, 2,
UCols, COLUMNS(Tbl)-ColstoKeep,
RowID, SEQUENCE((ROWS(Tbl)-1)*UCols),
ColID, SEQUENCE(1,ColstoKeep+2),
Ro, ROUNDUP(RowID/UCols,0)+1,
Col, MOD(RowID-1,UCols)+1+ColstoKeep,
IF(ColID<=ColstoKeep,INDEX(Tbl,Ro,ColID), IF(ColID=ColstoKeep+1,INDEX(Tbl,1,Col), INDEX(Tbl,Ro,Col))))

There are seven [7] variables:

- **Tbl** is an input table to unpivot
- **ColstoKeep** is the number of first columns you do not want to unpivot
- **UCols** is the number of unpivoted columns
- **RowID** and **ColID** are row and column indices of the output table
- **Ro** and **Col** are initial row and column positions of Values in the input table.

Then, the final part of the formula is the calculation to unpivot the last three [3] columns, *viz.*



Unpivoting

SP Unpivoting Columns - Suggested Solution.xlsx

Navigator

Error Checks: ☑

1. Unpivoting

Suggested Solution

Col 1	Col 2	x	y	z
A	D	6	7	9
B	E	2	1	8
C	F	5	3	4

Col 1	Col 2	Unpivoted Cols	Value
A	D	x	6
A	D	y	7
A	D	z	9
B	E	x	2
B	E	y	1
B	E	z	8
C	F	x	5
C	F	y	3
C	F	z	4

```
=LET(Tbl,        Data[#All],)
     ColstoKeep, 2,
     UCols,      COLUMNS(Tbl)-ColstoKeep,
     RowID,      SEQUENCE((ROWS(Tbl)-1)*UCols),
     ColID,      SEQUENCE(1,ColstoKeep+2),
     Ro,         ROUNDUP(RowID/UCols,0)+1,
     Col,        MOD(RowID-1,UCols)+1+ColstoKeep,
IF(ColID<=ColstoKeep,INDEX(Tbl,Ro,ColID),
IF(ColID=ColstoKeep+1,INDEX(Tbl,1,Col),
INDEX(Tbl,Ro,Col))))
```

Although it is a long and complex formula, you can apply it to your input table by only replacing the values for **Tbl** and **ColstoKeep**.

Until next time.

Upcoming SumProduct Training Courses - COVID-19 update

Due to the COVID-19 pandemic that is currently spreading around the globe, we are suspending our in-person courses until further notice. However, to accommodate the new working-from-home dynamic, we are switching our public and in-house courses to an online delivery stream, presented via Microsoft Teams, with a live presenter running through the same course material, downloadable workbooks to complete the hands-on exercises during the training session, and a recording of the sessions for your use within 1 month for you to refer back to in the event of technical difficulties. To assist with the pacing and flow of the course, we will also have a moderator who will help answer questions during the course.

If you're still not sure how this will work, please contact us at training@sumproduct.com and we'll be happy to walk you through the process.

Location	Course	Date	Date	Duration	Duration
Online (Australia)	Power Pivot, Power Query and Power BI	7 - 9 Dec 2022	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	14 Dec 2022	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	1 Day
Online (Australia)	Financial Modelling	15 - 16 Dec 2022	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	2 Days

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This month, we look again at the **CTRL** and **SHIFT** keys, but this time combined with various special characters that Excel uses:

Keystroke	What it does
CTRL + SHIFT + Down Arrow	Extend selection down to last cell in area down
CTRL + SHIFT + END	Extend selection to the last cell in the sheet
CTRL + SHIFT + ENTER	Fill value in edited cell into all cells and do not move
CTRL + SHIFT + ESC	Task Manager (Windows – overriding an Excel command)
CTRL + SHIFT + EXECUTE	Fill value in edited cell into all cells and do not move
CTRL + SHIFT + HOME	Extend selection to the cell that resets the window or the pane
CTRL + SHIFT + Left Arrow	Extend selection down to last cell in area left
CTRL + SHIFT + Page Down	Extend selection one sheet down (sheet tab to the right)
CTRL + SHIFT + Page Up	Extend selection one sheet up (sheet tab to the left)
CTRL + SHIFT + RETURN	Fill value in edited cell into all cells and do not move
CTRL + SHIFT + Right Arrow	Extend selection down to last cell in area right
CTRL + SHIFT + SPACE	Select contiguous area
CTRL + SHIFT + TAB	Previous Window
CTRL + SHIFT + Up Arrow	Extend selection up to end of region

There are *c.*550 keyboard shortcuts in Excel. For a comprehensive list, please download our Excel file at 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.

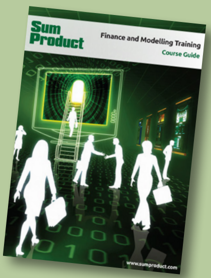
Any Questions?

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