

Sum Product

NEWSLETTER #131 - October 2023

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



Full steam ahead for Q4 2023! The worlds of Excel, financial modelling, Power BI *et al.* continue to gain momentum unabated. And we continue to report!

It's Excel Virtually Global this month: will we see you there? Numerous speakers from all over the world continue to give their time for free, all in the name of charity. Make a donation and come drop by. We'd love to see you there.

We revisit our old friend the Internal Rate of Return this month, to accompany all the usual items. We've another Beat the Boredom Challenge, plus tips on Charts & Dashboards, Visual Basics, Power Pivot Principles, Power Query Pointers, Power BI Updates, Excel Updates, say **HI** our A to Z of Keyboard Shortcuts and there is even time to **MATCH** up what you know in this month's A to Z of Excel functions.

But we add a new regular too: our new US lead Steve Kraynak begins to wax lyrical on all that is **Excel for Mac** – to be honest, an area we haven't focused on particularly in the past. You can check it out below.

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

Liam Bastick, Managing Director, SumProduct



Excel Virtually Global 2023



Not long to go now... Just a reminder, this 40+ hour virtual conference, for charity, presents many Excel, Data Platform and PowerPoint MVPs, together with other acknowledged experts from Microsoft and / or around the globe to present, answer questions and demonstrate the future of Excel and how it will make your life easier, personally and professionally.

Topics will include all things Excel, financial modelling, dynamic arrays, Python in Excel, Fabric, Copilot, ChatGPT, Power Pivot, Power Query and Power BI. It's not all in English either!

Each session (including Q&A) will last no more than an hour and topics will cover all expertise levels, from novice to expert. Most presenters are well known in their spheres, and have written blogs, books and articles and / or present video sessions.

Some sessions will be live and some will be recorded so you may watch them later with downloads aplenty – there are no medals for staying up to watch the entire event live! That's just as well, as it will run from midnight GMT / UTC on Tuesday 9 October into some time on Thursday 11 October.

From your own favourite chair, bring a laptop, an inquisitive mind and your sense of humour. Be prepared to learn heaps. And remember, it's for charity – this year, there is no ticketing process: all we ask is an "honesty box" where you donate to your favourite charity.

For more details (*e.g.* times, speakers, sessions), please go to www.excelvirtuallyglobal.com. The links will go live nearer the time, so keep checking back as the program may change too.

Hopefully, we'll see you there!



Irrelevant IRR

We've had a few queries about problems with calculating IRR recently, so we thought it was time to revisit this old chestnut.

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:

$$\begin{aligned}
 (\$100) + PV(\$121) &= (\$100) + \$121 / (1 + 10\%)^2 \\
 &= (\$100) + \$100 \\
 NPV &= 0
 \end{aligned}$$

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.

Excel has two functions that estimate 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%
- **XIRR(values,dates,[guess])** returns the annual IRR (assuming a 365 day year) for a set of cashflows which may neither be sequential nor occur on a regular periodic basis. Again, there must be at least one positive and one negative value in the range. **XIRR** will then cycle through a similar iterative technique (this time up to 100 times) to try and find an answer which is accurate to within 0.000001%
- If the guess is not specified, Excel will assume that it is 10% (0.1) for both functions.

Both are fraught with problems in practice – allow me to provide several examples. However, before we discuss some of the problems with the functions, let me compare and contrast with a simple example:

IRR vs. XIRR									
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
XIRR Illustration									
		3/7/11	18/9/11	24/2/12	23/2/13	19/4/13	20/4/15	18/10/15	18/10/17
XIRR	2.77%	(\$50,000.00)	\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66	-	-
XIRR	0.00%	-	(\$50,000.00)	\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66	-
XIRR	1.32%	(\$50,000.00)	-	-	\$18,750.00	\$12,412.00	\$57.16	\$1,655.22	\$19,450.66

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.

The **XIRR** use the same three cashflow scenarios and calculates correctly for the first and third scenarios. They demonstrate what happens when cashflows are aperiodic. For example, in the first scenario, instead of an annual IRR of 21.54% (as calculated above) we get 2.77%, reflecting the irregular payment profile.

The second scenario should be different now too, as the timing of the cashflows is different. The rate cited is 0.00%. This means that with no discounting the NPV should be zero, *i.e.* the numbers should add up to zero. They do not: the sum is actually \$2,325.05 and there is a clue in the number formatting as to what is going on here.

Regular readers will know that I always format numbers so that values that are precisely zero are represented by a dash ("-"), whereas numbers that are approximately zero are shown as 0, or similar. Here, the actual value (in scientific notation) is **2.98 E-09** (which is 2.98×10^{-9} or 0.00000000298). Watch out for this value with **XIRR**: it occurs regularly and shows that Excel has been unable to calculate the value correctly, which may call for a (better) guess to be used. Microsoft claims that if the function cannot calculate a value within the tolerable limit then **#NUM!** is supposed to be returned, but this is often not the case:

	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	XIRR	IRR
Cashflow - Scenario 1	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$2,000.00	203.34%	203.66%
Cashflow - Scenario 2	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$1,800.00	152.15%	152.39%
Cashflow - Scenario 3	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$1,600.00	104.08%	104.23%
Cashflow - Scenario 4	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$1,400.00	59.47%	59.55%
Cashflow - Scenario 5	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$1,100.00	0.00%	0.00%
Cashflow - Scenario 6	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$1,000.00	-17.55%	-17.57%
Cashflow - Scenario 7	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$912.93	-31.78%	-31.82%
Cashflow - Scenario 8	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$900.00	-31.79%	#NUM!
Cashflow - Scenario 9	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$800.00	0.00%	#DIV/0!
Cashflow - Scenario 10	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$700.00	0.00%	#DIV/0!
Cashflow - Scenario 11	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$600.00	0.00%	#DIV/0!
Cashflow - Scenario 12	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$500.00	0.00%	#NUM!
Cashflow - Scenario 13	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$400.00	0.00%	#DIV/0!
Cashflow - Scenario 14	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$300.00	0.00%	#DIV/0!
Cashflow - Scenario 15	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$200.00	0.00%	#DIV/0!
Cashflow - Scenario 16	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$100.00	0.00%	#DIV/0!
Cashflow - Scenario 17	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$	#NUM!	#NUM!
Cashflow - Scenario 18	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100.00)	#NUM!	#DIV/0!
Cashflow - Scenario 19	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$200.00)	#NUM!	#DIV/0!
Cashflow - Scenario 20	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$1,200.00)	#NUM!	#DIV/0!
Cashflow - Scenario 21	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$1,300.00)	#NUM!	#NUM!
Cashflow - Scenario 22	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$1,400.00)	#NUM!	#NUM!
Cashflow - Scenario 23	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$1,500.00)	#NUM!	#DIV/0!
Cashflow - Scenario 24	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$1,600.00)	#NUM!	#DIV/0!
Cashflow - Scenario 25	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$1,600.00)	#NUM!	#DIV/0!

IRR and XIRR can give several erroneous results. In summary, never trust 2.98 E-09.

One of the reasons this second example has gone wrong here is because XIRR requires a non-zero first value (in fact there appears to be significant anecdotal evidence to suggest it prefers a negative number).

First Cashflow

As cited above, XIRR requires a non-zero first cashflow and that the first date in the series is the start date (further dates do not need to be in order but are supposed to occur after the start date). Consider the following example:

Dates	
1/1/13	\$5,000
1/1/14	(\$55,000)
1/1/15	\$10,000
1/1/16	\$60,000
XIRR	2.98023E-09

XIRR gives a value of 2.98 E-09 here, i.e. a wrong answer (whereas IRR gives 21.43%, which is correct). XIRR should work here, but it does not. Negating all of the cashflows (which will not affect the IRR) makes no difference.

However, if the series is rearranged:

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

XIRR now gives the right answer (when it shouldn't). Of course, IRR will not work as cashflows need to be in the correct order. There appears to be little logic however:

Dates	
1/1/16	\$60,000
1/1/15	\$10,000
1/1/14	(\$55,000)
1/1/13	\$5,000
XIRR	#NUM!

Changing dates around to make things work is not recommended. One tip many professional modellers employ is to use a very small negative number for the first period's value, viz.

Dates	
2/1/12	(\$0)
1/1/13	\$5,000
1/1/14	(\$55,000)
1/1/15	\$10,000
1/1/16	\$60,000
XIRR	21.43%
XNPV using XIRR	\$0
IRR	21.43%
XNPV using IRR	\$0

Here, a value of -0.000001 was used. The date 2 January 2012 was chosen so that all periods are of equal length (*i.e.* so IRR should equal XIRR). This appears to work much of the time and hence is used by many in the modelling fraternity to overcome known **XIRR** issues. Unfortunately, this will not always be the case.

Date Recognition

Aside from understanding the importance of ordering time series, dates present another problem. Many times, when we analyse data we download information from third party Management Information Systems. Often, this downloaded data provides dates in a text format. This may cause problems for **XIRR**, *viz.*

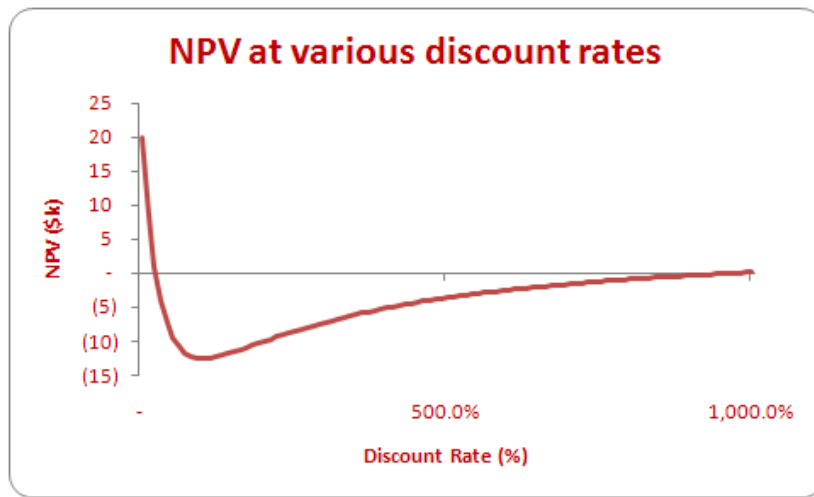
Date	Amt
3/3/2012	(\$3,000)
6/30/2012	\$1,000
9/30/2012	\$500
12/31/2012	\$2,000
XIRR	#VALUE!

What's gone wrong this time? Here, the dates have been downloaded as text strings and the **XIRR** function needs dates to be numerical. More often than not, this can be readily solved by applying the **DATEVALUE** function to these dates, although on occasion more painful (manual) approaches will have to be adopted.

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 **XIRR** function (albeit of the other solution 21.43%), **XIRR** and **IRR** return the two IRRs associated with this cashflow scenario.

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

Checking IRRs

What is a simple method for checking an IRR is correct? Consider the following example:

1/01/2013 12:00 AM	(\$100)
2/04/2013 6:00 AM	\$5
2/07/2013 12:00 PM	\$5
1/10/2013 6:00 PM	\$5
1/01/2014 12:00 AM	\$120
Duration	365
XIRR	37.643173%
IRR	37.635129%
XNPV using XIRR	0.000000
XNPV using IRR	0.005459

Here, I have split the calendar year up into four equal parts (hence taking into account the time of day). **XIRR** and **IRR** should give the same answer, allowing for the fact that theoretically the **XIRR** solution is calculated to within 0.000001% whereas **IRR** is 'only' calculated to within 0.00001%. Unsurprisingly, testing the values using **XNPV(rate,values,dates)** shows that **XIRR** gives a more accurate answer.

Rubbish.

Let's calculate the implied NPV long hand from first principles:

XIRR Rate					
XIRR Rate	37.643173%				
Quarterly Rate	8.315016%				
Time	0	1	2	3	4
Cashflows	(100)	5	5	5	120
Discount Factor	1.000	0.923	0.852	0.787	0.727
Present Value of Cashflows	(100)	5	4	4	87
NPV	<u>(0.005460)</u>				
IRR Rate					
IRR Rate	37.635129%				
Quarterly Rate	8.313434%				
Time	0	1	2	3	4
Cashflows	(100)	5	5	5	120
Discount Factor	1.000	0.923	0.852	0.787	0.727
Present Value of Cashflows	(100)	5	4	4	87
NPV	<u>0.000000</u>				

Notice that the **IRR** value actually gives the more accurate answer. This is because of the little known fact that **XIRR** always truncates dates and assumes cashflows occur at midnight. You might think this immaterial, but I provide this example to show that **XIRR** is not always as accurate as many think. Take care with **XNPV**: it follows a similar logic to **XIRR** and should not be used to verify **XIRR** values.

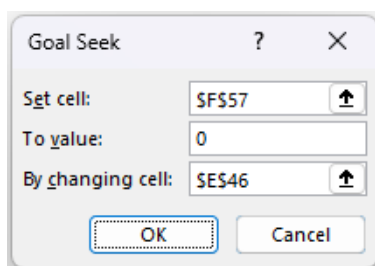
So What Would You Use Instead?

There are times when accuracy is paramount, e.g. bond issues involving large sums of money. Excel will only calculate **XIRR** on a maximum cycle of 100 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:

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
41																	
42					Falling Back on GOAL SEEK												
43																	
46				Rate Used													
47																	
48					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	
49				Days Since Start	-	31	59	90	120	151	181	212	243	273	304	334	
51				Cashflow - Scenario 7	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	(\$100)	\$913
52				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	
53				Present Value	(\$100)	(\$103)	(\$106)	(\$110)	(\$113)	(\$117)	(\$121)	(\$125)	(\$129)	(\$133)	(\$138)	(\$141)	
54				Net Present Value													\$0.000000
55																	
56																	
57																	
58																	

Here, an NPV calculation is constructed long hand with the discount rate an input in cell **E46**. Next, call up 'Goal Seek' (**ALT + T + G**):



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.

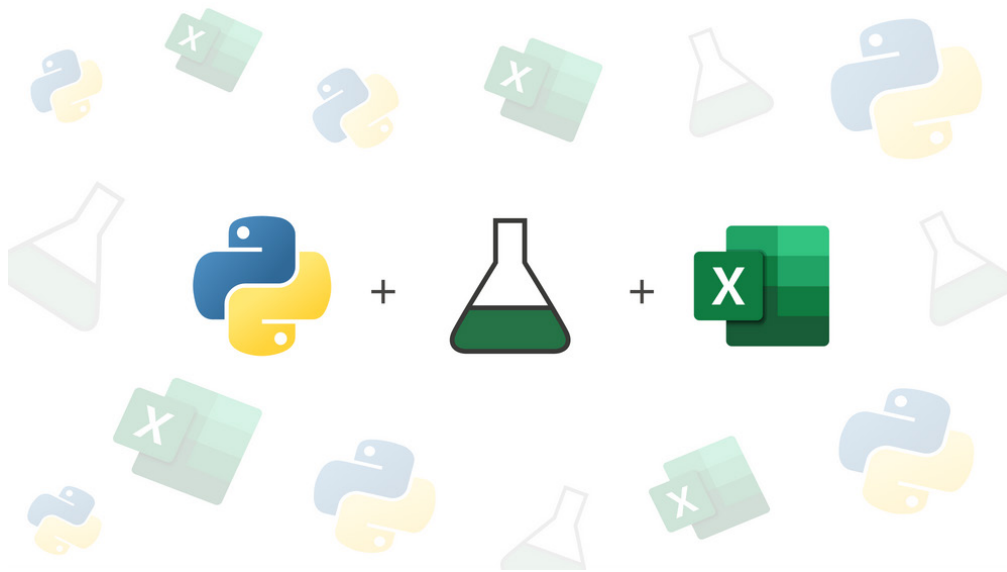
It may be a cumbersome method, but give me reliability over elegance any day.

Python Editor from Excel Labs



Hot on the tails of Python in Excel, the Python Editor is a new addition to Excel Labs that allows you to write and edit Python formulae in Excel using a dedicated code editor with similar capabilities as those available in Python notebook environments. It should be noted that this is an

experimental release to get early feedback for Microsoft, so it might all change. The Python Editor is designed to complement the Python in Excel experience with an editing surface optimised for more complex and longer Python code snippets.

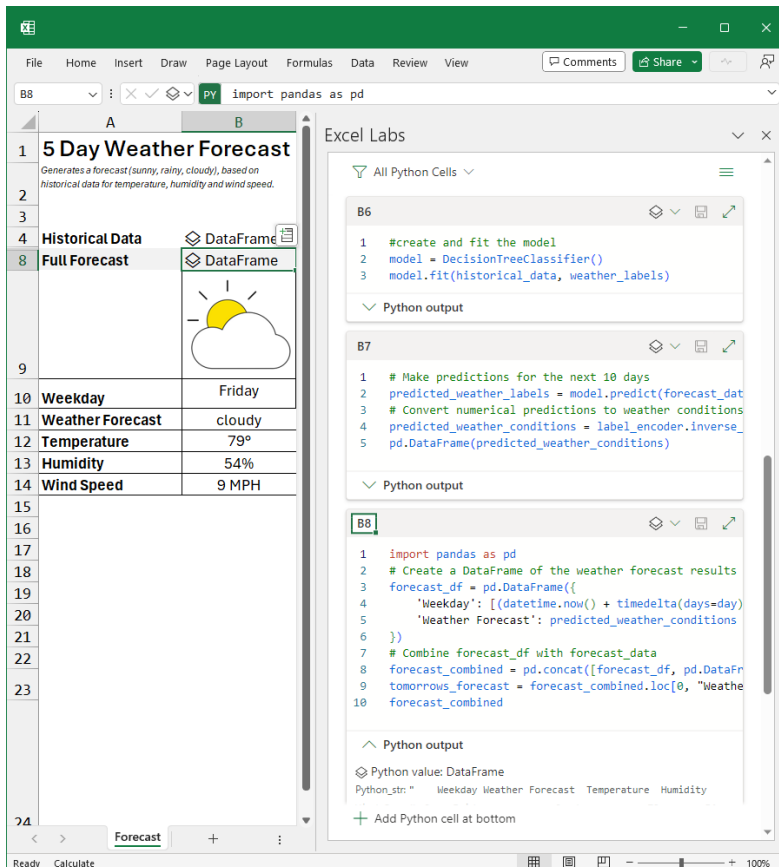


The Python Editor offers several benefits when writing and editing Python formulae in Excel.

View the flow of code execution with results

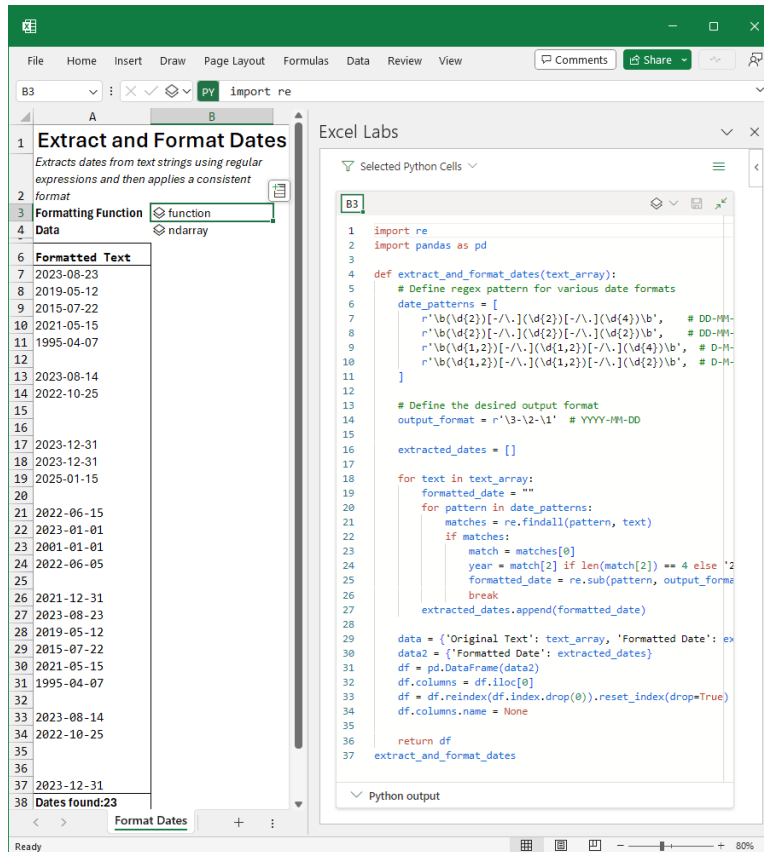
The Python Editor shows Python cells in execution order in addition to the output of each cell's code, which can help you debug and understand your code. This is useful because Python cells within the Excel grid

execute in row major order. This means that the cell calculations run across a row (from column A to column XFD), and then across each following row down the worksheet.



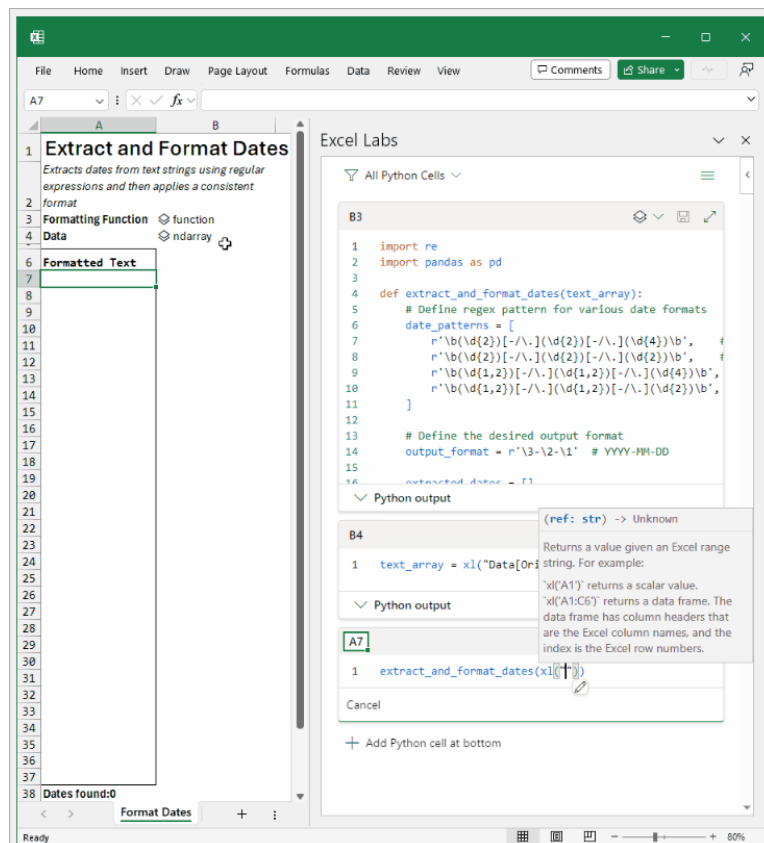
Built for longer sections of code

The Python Editor provides a new way to create and edit your Python code, offering more flexibility when working with longer scripts.



Standard Visual Studio code features

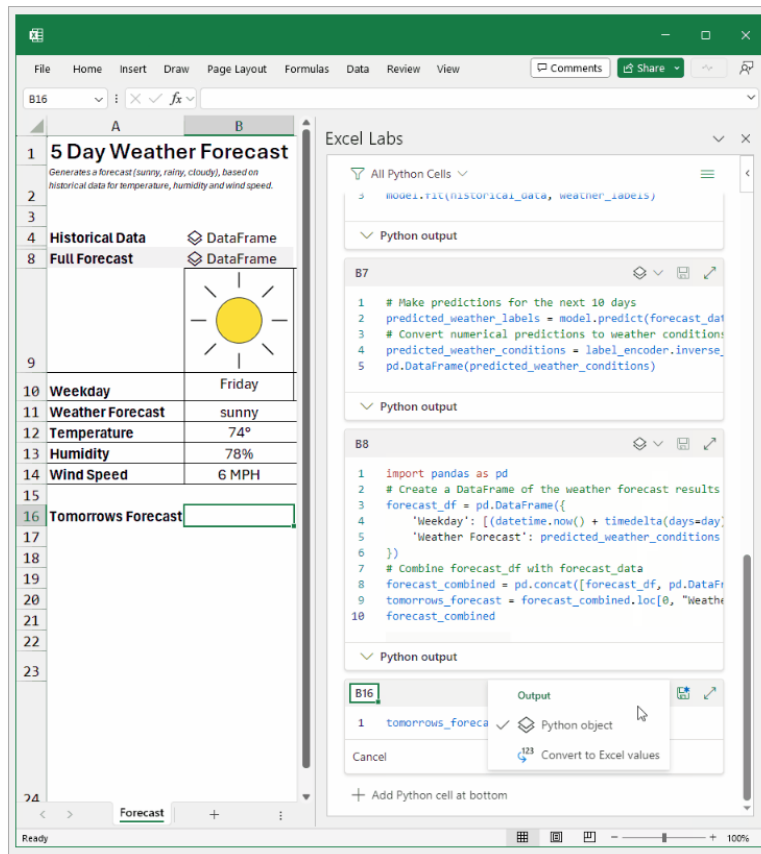
The Python Editor uses some of the same components that power Visual Studio Code and supports many of the features that Visual Studio Code offers for Python development, such as IntelliSense, formatting, code completion and syntax highlighting.



Flexibility for intermittent coding workflows

The Python Editor allows you to write some code, click elsewhere in the application, and not commit the code to Excel until you are happy with it. This gives you more flexibility and control over when and how you apply

your code to your workbook. Additionally, when in manual recalculation mode, editing and committing from the Editor will calculate just that cell (like a notebook) providing a faster feedback loop.



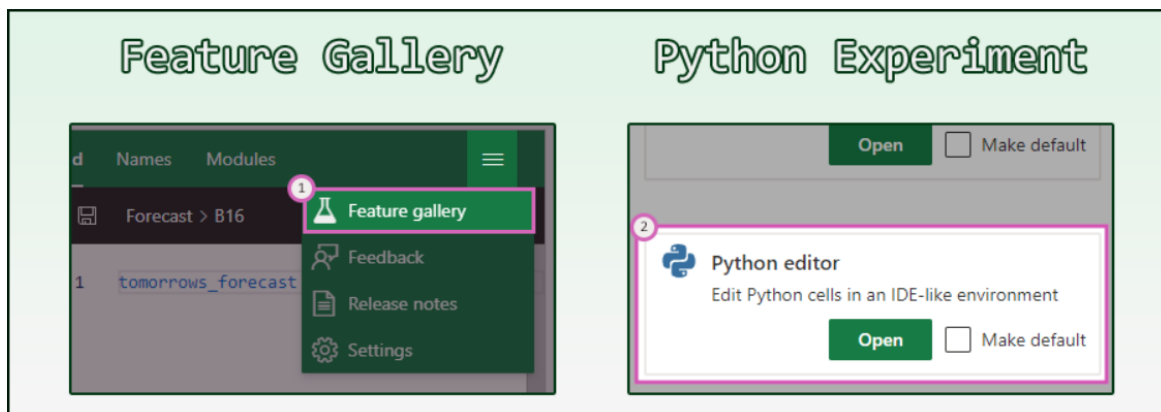
The Python Editor is intended to enhance your productivity and creativity with Python in Excel. You can use it to write and edit Python formulae that perform complex calculations, manipulate data, create charts, and more. You can also switch between the task pane editor, the cell editor and the Formula bar, depending upon your needs and preferences.

How to get the Python Editor

If you already have the Excel Labs add-in, you can update it to get the latest version that includes the Python Editor. If you don't have the Excel Labs add-in, you can install it from the Add-ins Store. To do this, go to the Insert tab, click on 'Get Add-ins', and then search for Excel Labs. Click on 'Add' to install the add-in.

Once you have the Excel Labs add-in, you need to launch the Python Editor. This may be done by:

- hitting the menu at the top right and selecting 'Feature Gallery'
- selecting the 'Python Editor experiment'.



Anyone may get Excel Labs. However, you need to have access to Python in Excel to make use of the Python Editor.

Beat the Boredom Challenge


With many of us currently “working from home” / quarantined, there are only so Zoom / Teams calls and virtual parties you can make before you reach your (data) limit. Perhaps they should measure data allowance in blood pressure millimetres of mercury (mmHg). To try and keep our

readers engaged, we will continue to reproduce some of our popular **Final Friday Fix** challenges from yesteryear in this and upcoming newsletters. One suggested solution may be found later in this newsletter. Here's this month's...

The issue this month may be common and useful for anyone who uses Excel to produce cumulative / running total reports frequently.

This month's challenge is to write a **formula** on a new column of the provided table to calculate the running total of column **Number** regardless of whether the Excel Table (*i.e.* created using **Insert -> Table** or **CTRL + T**) is filtered or not. The result should look like the column **Total** generated on the right (*below*):

Letter	Number
a	1
b	2
c	3
d	4
dd	(20)
e	5
f	6
g	7
h	8
i	9
j	10



Letter	Number	Total
a	1	1
c	3	4
d	4	8
dd	(20)	(12)
f	6	(6)
h	8	2
i	9	11
j	10	21

As always, there are some requirements:

- the formula needs to be within just the Total column (no “helper” cells)
- this is a formula challenge; no Power Query / Get & Transform or VBA!
- the formula still calculates the correct running total when the table is *filtered*.

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

Excel for Mac – A New Series

With Steve Kraynak joining the team, we thought we would exploit his knowledge and recant all about Microsoft Excel for Mac. Each month, we'll cover a different topic to help you understand how Excel for Mac is

different than Excel for Windows. But hey, let's get introductions out of the way first!



Microsoft describes their applications on Mac as “uniquely Office and optimized for Mac”. This is their way of saying that the apps are not exactly the same on Mac as they are on Windows, but it's intentional, because Mac is not the same as Windows. In this series, we'll dive into details about important differences, and we'll show how you can use Excel successfully even with those differences. There are some features that just don't exist in Excel for Mac, and we'll help with workarounds or other solutions where possible.

A bit of history

Did you know that the first version of Excel was released for Macintosh computers? Not only that, but it was available for Macintosh two years before it became available for Windows. That's right: in 1985, version 1.0 of Excel was released, but it was only compatible with Macintosh at the time. When released for Windows two years later, it was called version 2.0 to stay in step with the version on Macintosh.

As Windows became the dominant operating system in use throughout the world, making Excel great on Windows took priority, and understandably so. At some point, Microsoft even split off the development teams for their Mac apps into a separate business unit, call “Mac BU”, which later became known as APEX (the Apple Productivity Experience Group).

The goal of Mac BU was to make experiences in the apps that would meet the expectations of Mac users. It's understandable that there should be differences, given that the operating systems are quite different. For

example, Mac's “light-dismiss” behaviour for dialog boxes led to differences in Excel's Filter dialog between Mac and Windows. The key difference being that filter selections are applied immediately on Mac. Conversely, they aren't applied until you click OK or press **ENTER** on Windows. The Filter dialog on Mac now has an option for ‘auto apply’ or not.

It became typical for innovations to be developed and added to Excel for Windows, and then “ported” to Excel for Mac. Porting is a developer's way of copying code from one codebase to another, but it involves some work to adapt it so that code written to run on Windows will work on the Mac.

For these two reasons – having a separate business unit and the need to port code from Windows to Mac – the number of differences, including feature gaps, grew over the years. Some Mac-specific capabilities were added by the Mac BU / APEX team, and some features were either not ported completely or not ported at all.

Coming back together

Eventually, there were enough differences between the apps on Mac versus Windows that many customers weren't satisfied. Mac users weren't getting the same power and capability, and they let Microsoft know about it. Microsoft could have decided to port more features to Mac. However, one of the big innovations – co-authoring – would be very complex to port, particularly for Excel.

Co-authoring was a must-have for the Mac apps. Therefore, in 2017,

the decision was made to bring the codebases and business units back together. This was the best way to enable co-authoring in Excel for Mac. For the next year, the previously separate Excel teams came together and embarked on a momentous project to build the apps for both Windows and Mac from a single codebase. In simple terms, this meant having only a single engine, or core, for Excel, with would work on both operating systems. The user interface would be different, but not the engine.



In January 2018, version 16.x was released. This was built from the shared codebase, and Office for Mac and Windows have been running on shared code ever since. There are still differences, but the number of features missing on Mac has been reduced dramatically. Some are "earned" differences, meaning that they're intentional for the experience

on each operating system, and not due to the organisational structure of Microsoft. Other differences are due to the underlying capabilities of each operating system.

We'll continue next month...

Charts and Dashboards

It's time to chart our progress with an introductory series into the world of creating charts and dashboards in Excel. This month, we continue our review of dashboard design, with a look at how to implement conditional formatting in Tables.

Dashboards do not necessarily contain only charts. We can use a tabular report as the source document for a dashboard table or chart component, provided we limit the number of rows returned.

When using tables for charts or dashboards, it is important that we format the number components properly. You can read about [number formatting](#) and [conditional custom number formatting](#) as needed. Additionally, conditional formatting is the best and simplest choice to highlight data and assist visualisation, explore and analyse data, detect

critical issues, and identify patterns / trends. Conditional formatting (<https://www.sumproduct.com/thought/conditional-formatting>) is an Excel feature that does allow users to apply formats, such as colours, icons and data bars, to a cell or range of cells, and have that formatting change, depending upon the value of the cell or the value of a corresponding formula.

For example, we have an 'Order Volume' data table with order volumes greater than 25 displayed in a different colour:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
7																					
8																					
9																					
10																					
11																					
12																					
13																					
14																					
15																					
16																					
17																					

Order Volume	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Myer	25	19	19	20	23	16	16	20	23	19	12	25
Sephora	23	17	15	19	17	16	32	24	27	15	15	20
Target	21	12	19	17	19	13	26	23	13	12	16	28

In this table, the 'Sales Amount' points which lie in top 20% are highlighted in red, making them stand out.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
25																					
26																					
27																					
28																					
29																					
30																					
31																					

In this case, we have the 'Total Sales' in a particular month shown in a comparison with the 'Target sales' by means of data bars. The bars stretching to the right are blue in colour, indicating positive values, which means 'Total Sales' are above the target, and the bars stretching to the left are red in colour indicating negative values, which in contrast, means that 'Total Sales' do not meet the target.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
25																					
26																					
27																					
28																					
29																					
30																					
31																					
32																					
33																					

Color Scales (sic), like a heat map, are employed to show the value in a cell, relative to the values in the other cells in the selected range, by using the cell shading to display the difference in cell value. In the table below, we can quickly see the months with higher 'Total Sales' coloured in a darker shade:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
25																					
26																					
27																					
28																					
29																					
30																					
31																					
32																					
33																					

Icon Sets can also be used when describing the movement up, slightly up, slight down and down e.g. in 'Total Sales'.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
25																						
26																						
27																						
28																						
29																						
30																						
31																						
32																						
33																						

Conditional formatting options in Excel are not restricted to built-in options. We can define our own rules and format a range of cells satisfying a particular condition, like in the example below, where both highlighting rules are used to point out the highest sales in a month, and icons are used to get the movement in 'Total Sales'.

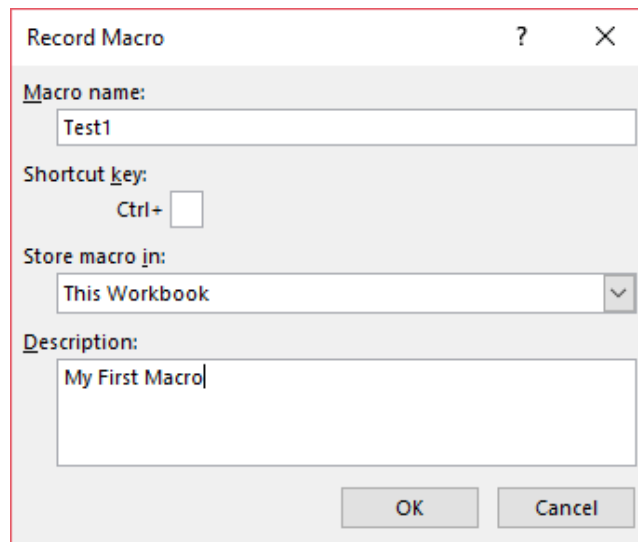
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
25																					
26																					
27																					
28																					
29																					
30																					
31																					
32																					
33																					

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 review how to record the macro.

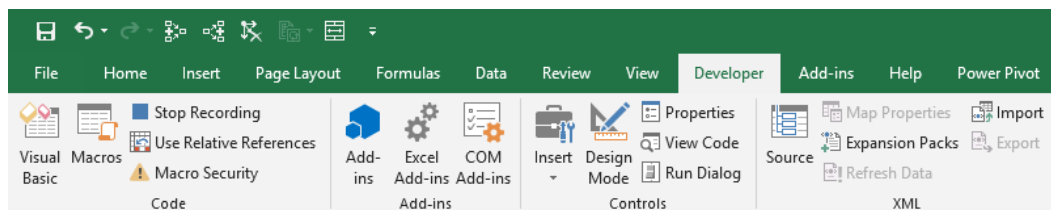
Last month, we talked through the four ways of recording a macro. Let's assume we've got that initiated. After pressing 'Record Macro', the 'Record Macro' dialog box will appear. Let's start by giving the macro a name and description and press 'OK':



After hitting 'OK' the recording mode will start in Excel. Looking back at the status bar in Excel, the icon next to 'Ready' has changed into a Stop button:



Pressing the Stop button will stop the recording of the macro. 'Stop Recording' will also replace the 'Record Macro' buttons on the Developer and View tabs, as will the keyboard shortcut **ALT + W + M + R**.



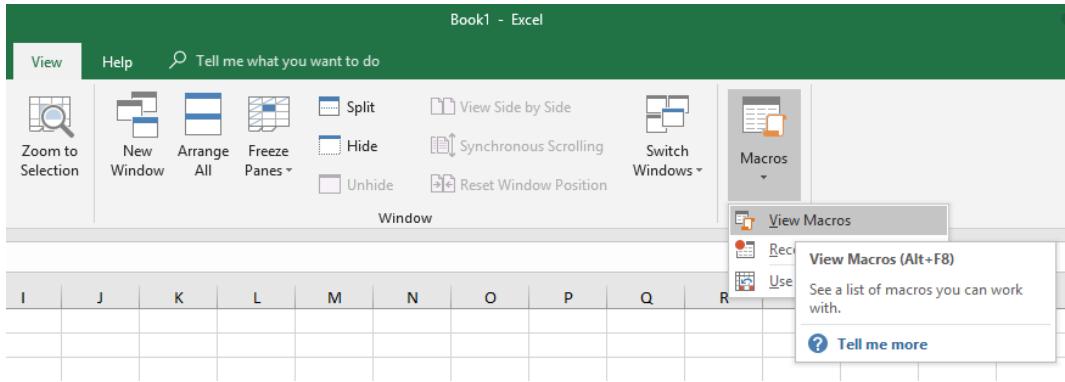
For our first example macro, let's perform the following actions

- Click on the 'Record Macro' button
- Fill cell **A5** red
- Copy cell **A5** to **B3**
- Type "colour" in cell **C6**
- Click on the 'Stop recording' button.

	A	B	C	D
1				
2				
3				
4				
5				
6			colour	
7				
8				

That's our first macro recorded!

Going back to the 'Macros' button in the Ribbon and select 'View Macros':



A list of all the macros in the workbook will come up including our macro that we have just recorded.

Continued next month.

Power Pivot Principles

We continue our series on the Excel COM add-in, Power Pivot. This month, we review the **MEDIANX** function.

Last month, we introduced the **MEDIAN** function, so this week we thought it would be appropriate to talk about the **MEDIANX** function. That's right: this one is 'X' rated. We detailed what 'median' means and how to use the **MEDIAN** function last time.

Just like the **MEDIAN** function, the **MEDIANX** function also returns the median value from a column of values in a data table. The 'X' at the end of the function name highlights the fact that **MEDIANX** works on a record by record basis, rather than in aggregation (similar to **SUMX**). Also, unlike the **MEDIAN** function, the **MEDIANX** function does not ignore blanks and it also has different inputs.

The **MEDIANX** function uses the following syntax:

MEDIANX(table, expression)

The **MEDIANX** requires a **table** input. This means that we can embed another function that returns a table, including the **FILTER** function, followed by an **expression** for the **MEDIANX** function to work.

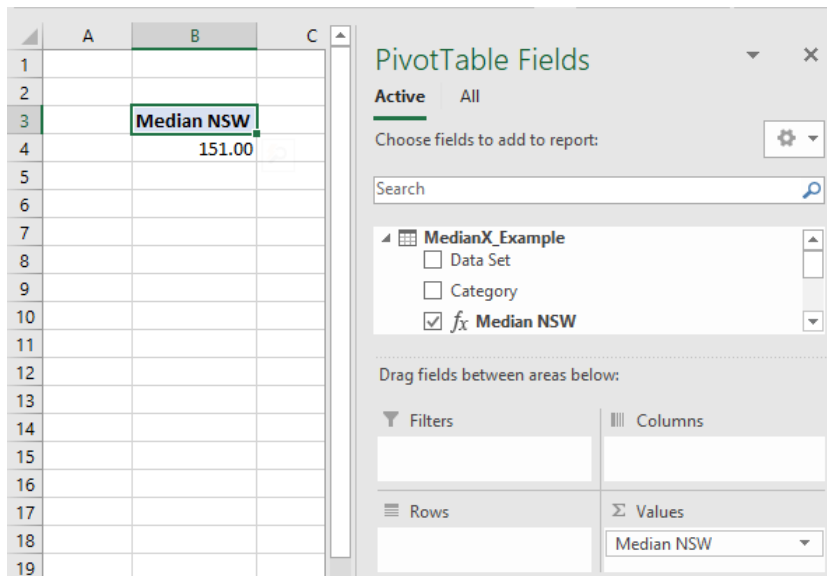
Let's take a look at an example. We have assigned Australian states to each value:

	A	B	C
1			
2		Data Set	Category
3		199	NSW
4		197	VIC
5		191	WA
6		181	TAS
7		179	WA
8		177	WA
9		172	NSW
10		168	NSW
11		155	NSW
12		147	NSW
13		127	NSW
14		105	WA
15		101	TAS
16		96	VIC
17		90	VIC
18		82	VIC
19		70	VIC
20		57	NSW
21		52	WA
22		46	VIC
23		42	VIC
24		35	VIC
25		33	NSW
26		25	TAS
27		19	VIC
28		8	TAS
29			

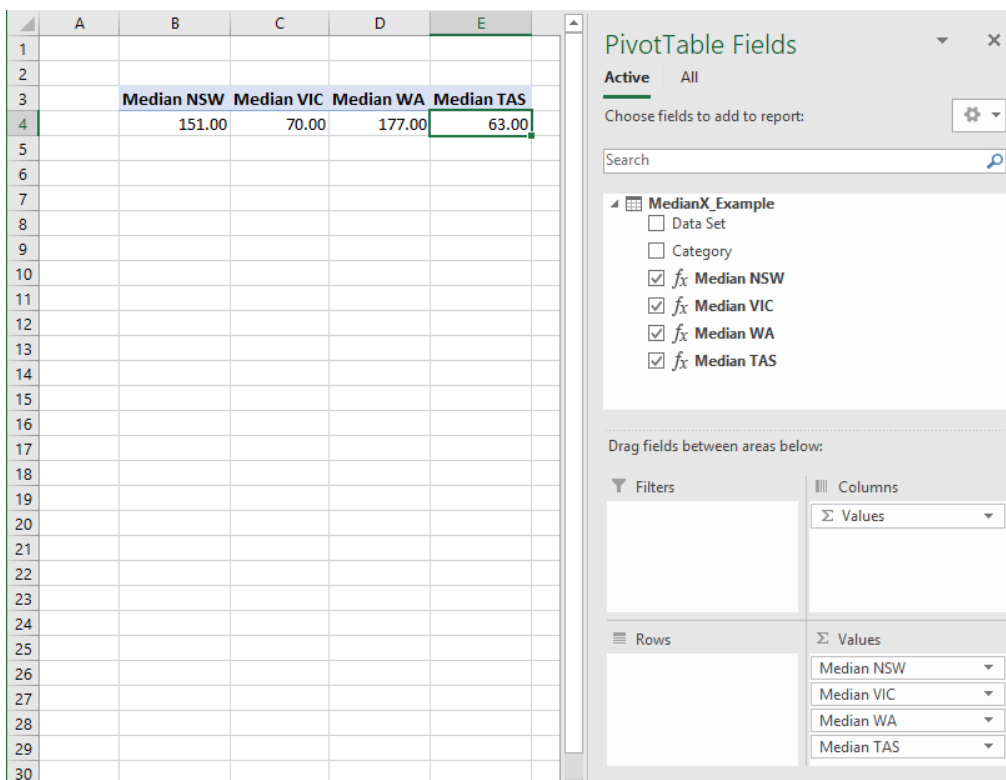
Let's create four measures that will calculate the median value for each state. For example, for NSW, the measure would be:

```
=MEDIANX(  
    FILTER(  
        MedianX_Example,[Category]="NSW"  
    )  
    ,MedianX_Example[Data Set]  
)
```

Looking at our PivotTable:



We can create measures for the rest of the states (these measures are similar, we just have to change the "NSW" text in the measure above to the new state), viz.



As we can see, the **MEDIANX** also sorts the dataset into ascending order before calculating the median value, so it will ignore any sorting that we apply to the column.

Let's see what happens when we add some blank values into our data set:

	A	B	C
1			
2		Data Set	Category
3		199	NSW
4		197	VIC
5		191	WA
6		181	TAS
7		179	WA
8		177	WA
9		172	NSW
10		168	NSW
11		155	NSW
12			NSW
13			NSW
14			NSW
15		147	NSW
16		127	NSW
17		105	WA
18		101	TAS
19		96	VIC
20		90	VIC
21		82	VIC
22		70	VIC
23		57	NSW
24		52	WA
25		46	VIC
26		42	VIC
27		35	VIC
28		33	NSW
29		25	TAS
30		19	VIC
31		8	TAS
32			

Refreshing the same PivotTable, the median for NSW has changed from 151 to 127, this is because the **MEDIANX** function **DOES NOT** ignore blank values.

The screenshot shows an Excel PivotTable with the following data:

	Median NSW	Median VIC	Median WA	Median TAS
	127.00	70.00	177.00	63.00

The PivotTable Fields task pane shows the following configuration:

- Active: All
- Choose fields to add to report:
 - Data Set
 - Category
 - f_x Median NSW
 - f_x Median VIC
 - f_x Median WA
 - f_x Median TAS
- Drag fields between areas below:
 - Filters: (Empty)
 - Columns: Σ Values
 - Rows: (Empty)
 - Values:
 - Σ Values
 - Median NSW
 - Median VIC
 - Median WA
 - Median TAS

It is important to keep that in mind when using the **MEDIANX**. It does not ignore blank values in the dataset. Therefore, be sure to clear out blank values in the dataset before calculating the median value.

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

Power Query Pointers

Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Office 365, Excel 2016 and 2019) from www.sumproduct.com/blog. If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month, we look at an example where we need to translate numbers to letters in multiple columns using the **M function List.ReplaceMatchingItems()**.

This month, we have the school examination results for some children who were lucky(!) enough to take their exams when the UK system was in the middle of changing from letters to numbers. It's confusing, not least because nine [9] is the new top grade, whereas intuitively you might expect one [1] to be the highest. We want to replace the numbers with letters so that we may see everything in the same format. The equivalent

grades are 9 = 'A*', 8 = 'A*/A', 7 = 'A', 6 = 'B', 5 = 'B/C', 4 = 'C', 3 = 'D/E', 2 = 'E/F', 1 = 'G'.

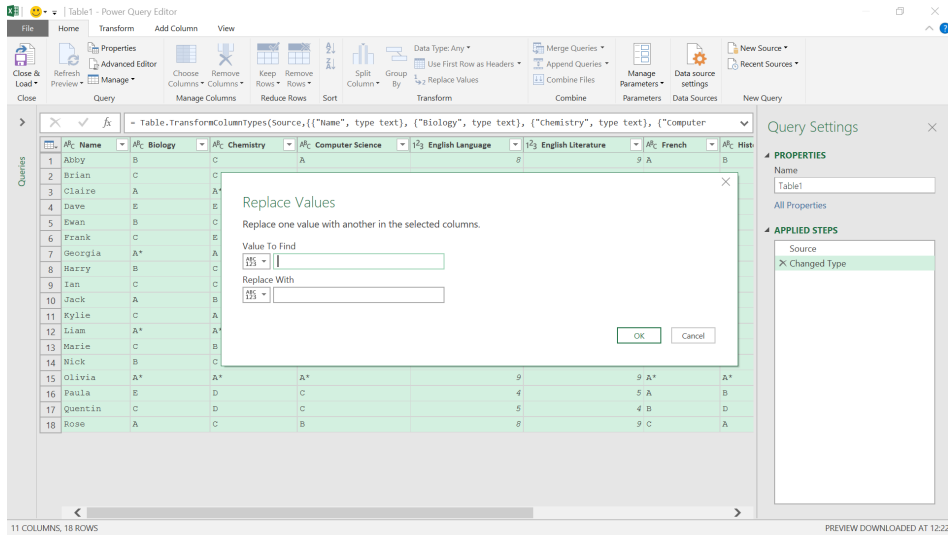
This week, we'll look at how to find the equivalent letter grade using the **M function List.ReplaceMatchingItems()**; next month, we'll use a different approach.

Name	Biology	Chemistry	Computer Science	English Language	English Literature	French	History	Mathematics	Physics	Spanish
Abby	B	C	A	8	9 A	B	7 B	A		
Brian	C	C	A	7	6 B	B	6 C	C		
Claire	A	A*	B	5	6 A	A	7 B	B		
Dave	E	E	E	2	2 E	E	3 E	E		
Ewan	B	C	B	7	8 A	C	8 C	C		
Frank	C	E	B	6	5 C	B	7 C	C		
Georgia	A*	A	A	9	8 B	A*	8 B	A*		
Harry	B	C	A	7	6 B	C	7 B	A		
Ian	C	C	D	6	6 A	B	6 C	B		
Jack	A	B	B	6	7 C	A	6 A	A		
Kylie	C	A	C	5	5 B	D	6 B	B		
Liam	A*	A*	A*	9	9 A*	A*	9 A*	A*		
Marie	C	B	A	7	8 B	A	8 B	D		
Nick	B	C	D	6	8 C	A	5 E	B		
Olivia	A*	A*	A*	9	9 A*	A*	9 A*	A*		
Paula	E	D	C	4	5 A	B	3 B	B		
Quentin	C	D	C	5	4 B	D	2 C	B		
Rose	A	C	B	8	9 C	A	8 B	A		

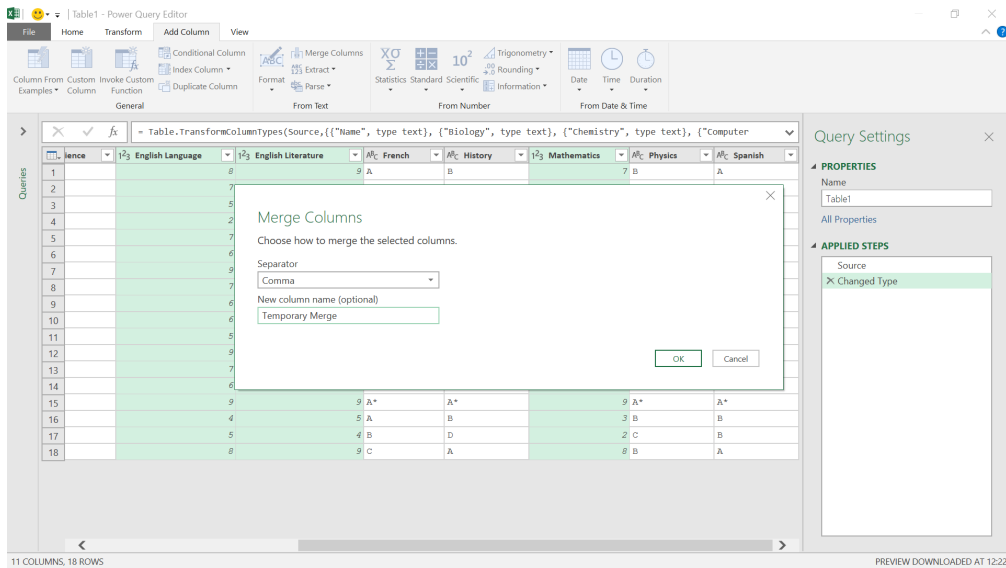
Let's begin by creating a query 'From Table' in the 'Get & Transform' section of the 'Data' tab.

Name	Biology	Chemistry	Computer Science	English Language	English Literature	French	History
Abby	B	C	A	8	9 A	B	7 B
Brian	C	C	A	7	6 B	B	6 C
Claire	A	A*	B	5	6 A	A	7 B
Dave	E	E	E	2	2 E	E	3 E
Ewan	B	C	B	7	8 A	C	8 C
Frank	C	E	B	6	5 C	B	7 C
Georgia	A*	A	A	9	8 B	A*	8 B
Harry	B	C	A	7	6 B	C	7 B
Ian	C	C	D	6	6 A	B	6 C
Jack	A	B	B	6	7 C	A	6 A
Kylie	C	A	C	5	5 B	D	6 B
Liam	A*	A*	A*	9	9 A*	A*	9 A*
Marie	C	B	A	7	8 B	A	8 B
Nick	B	C	D	6	8 C	A	5 E
Olivia	A*	A*	A*	9	9 A*	A*	9 A*
Paula	E	D	C	4	5 A	B	3 B
Quentin	C	D	C	5	4 B	D	2 C
Rose	A	C	B	8	9 C	A	8 B

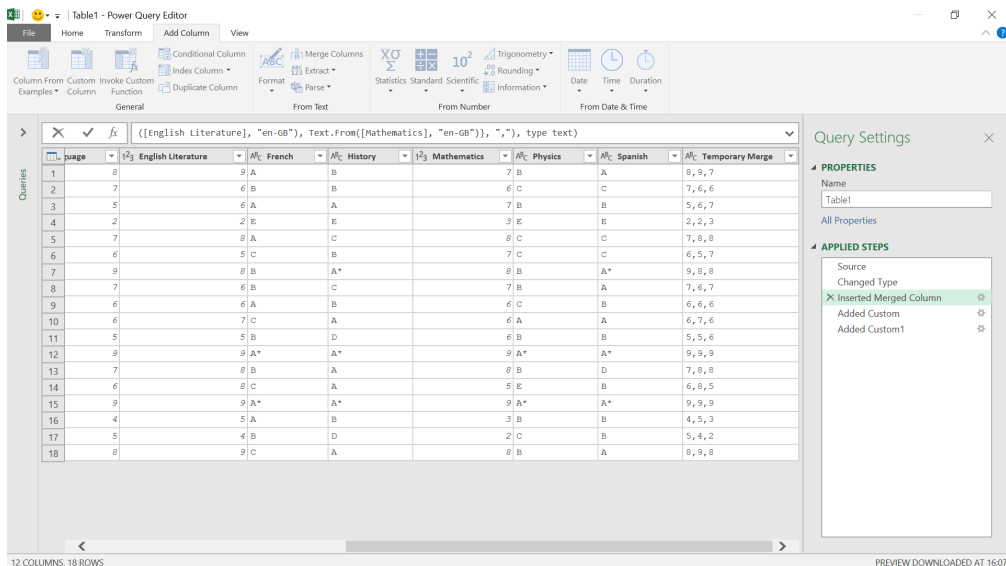
We could create a custom column for each of my numerically graded subjects, but we want to substitute all the columns at the same time. We could try replacing values in all of our columns:



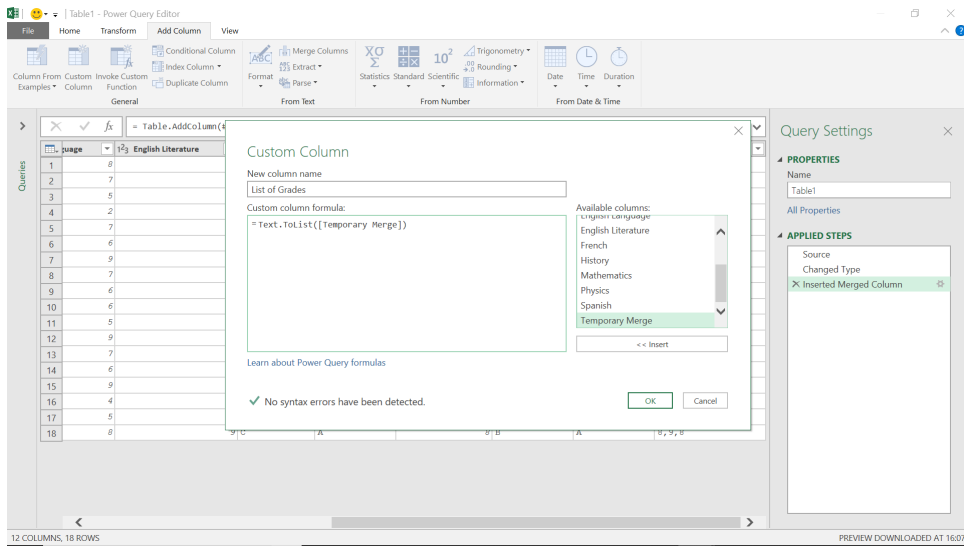
However, this would only replace one value at a time. We want to translate all the values at the same time for all the numerical columns. One way to do this is to use the **List()** M functionality. Let's begin by merging the columns we need to translate; **English Language, English Literature and Mathematics**.



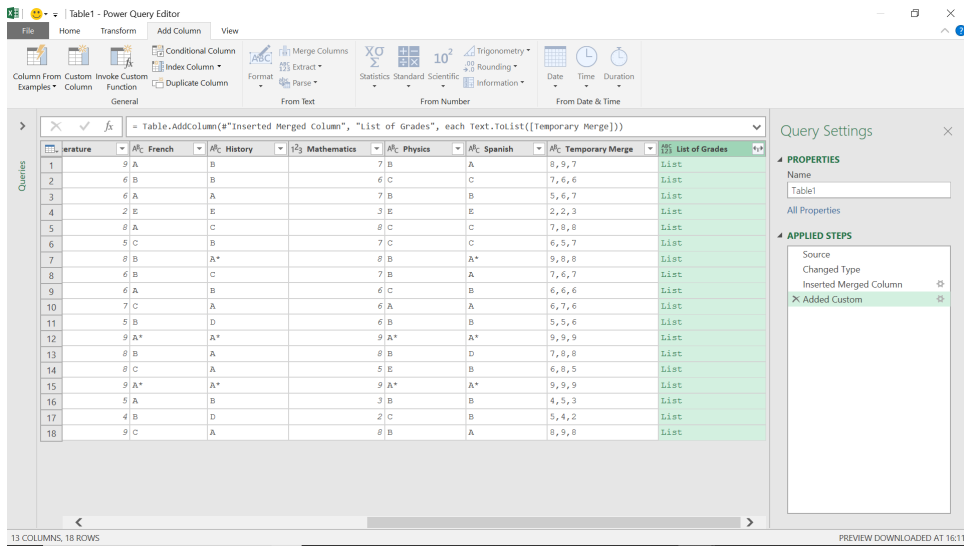
We will decide to use a comma delimiter to help convert our data back later.



Now, we may use **Text.ToList()** to convert our column to lists:



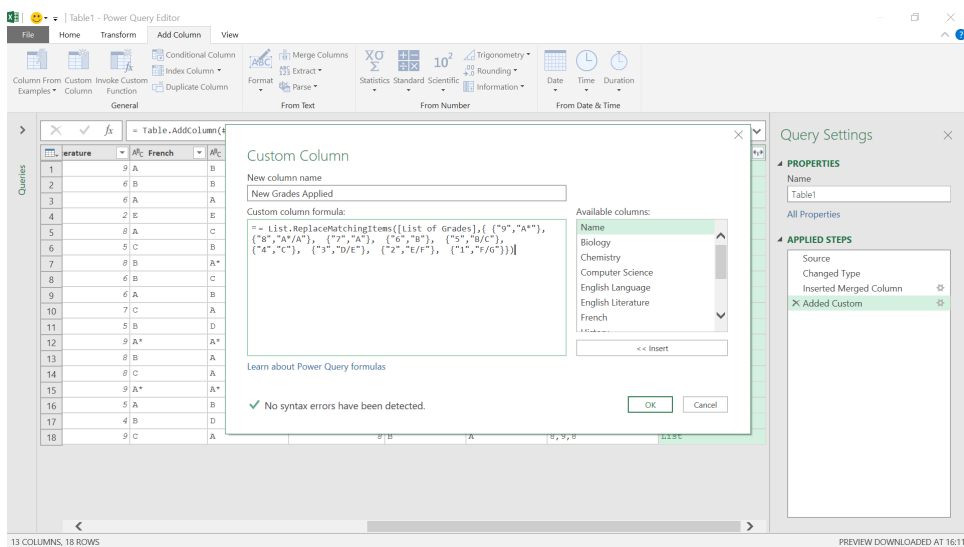
This provides a column of lists that we may manipulate.



We can now use a **List()** function to transform our data. The one we are going to use is

List.ReplaceMatchingItems(list as list, replacements as any, optional equationCriteria as any) as list

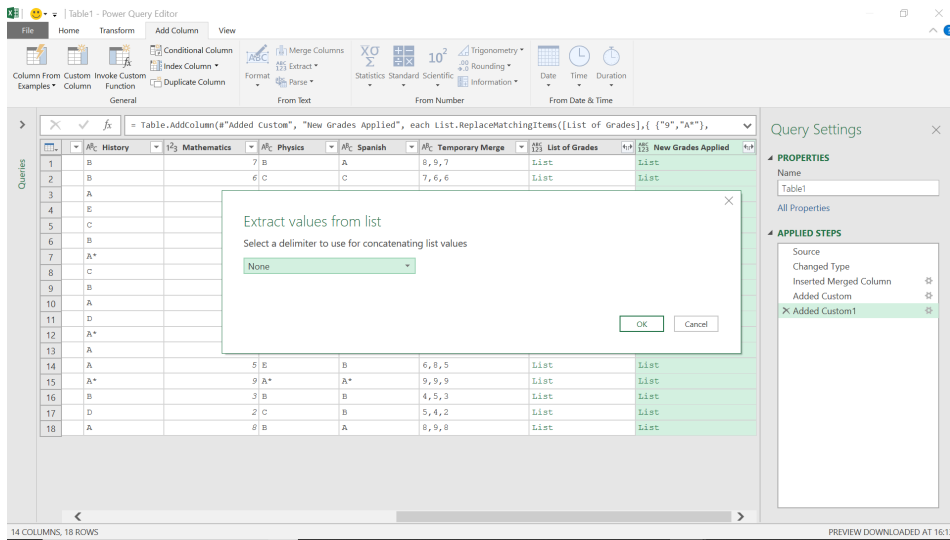
This replaces occurrences of existing values in the **list** with new values using the provided **equationCriteria**. Old and new values are provided by the **replacements** parameters.



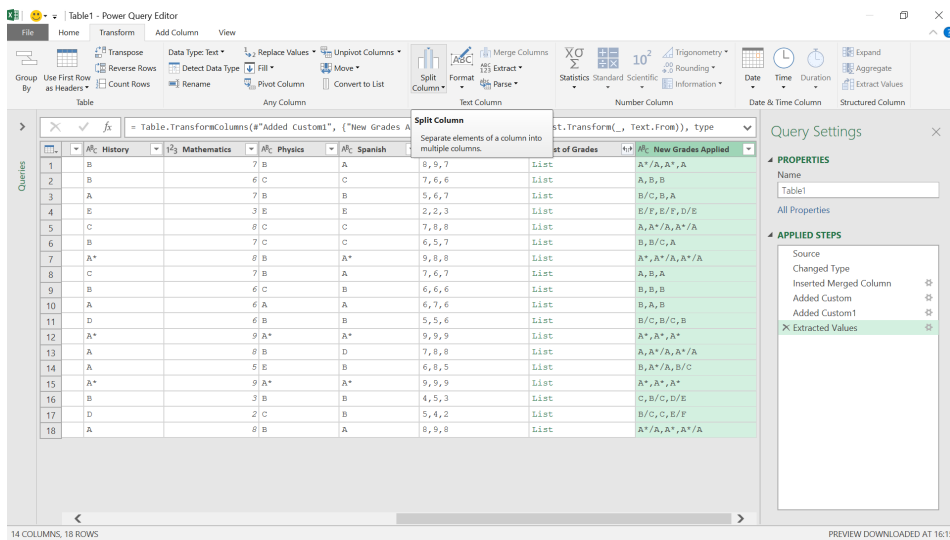
The M formula used is

```
= List.ReplaceMatchingItems([List of Grades],{"9","A*"}, {"8","A*/A"}, {"7","A"}, {"6","B"}, {"5","B/C"}, {"4","C"}, {"3","D/E"}, {"2","E/F"}, {"1","F/G"}])
```

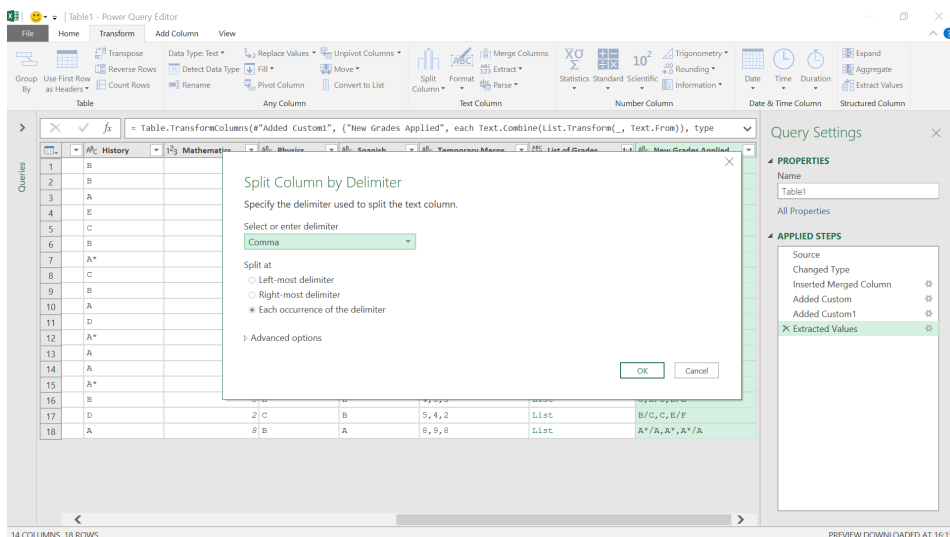
Now we will extract the list.



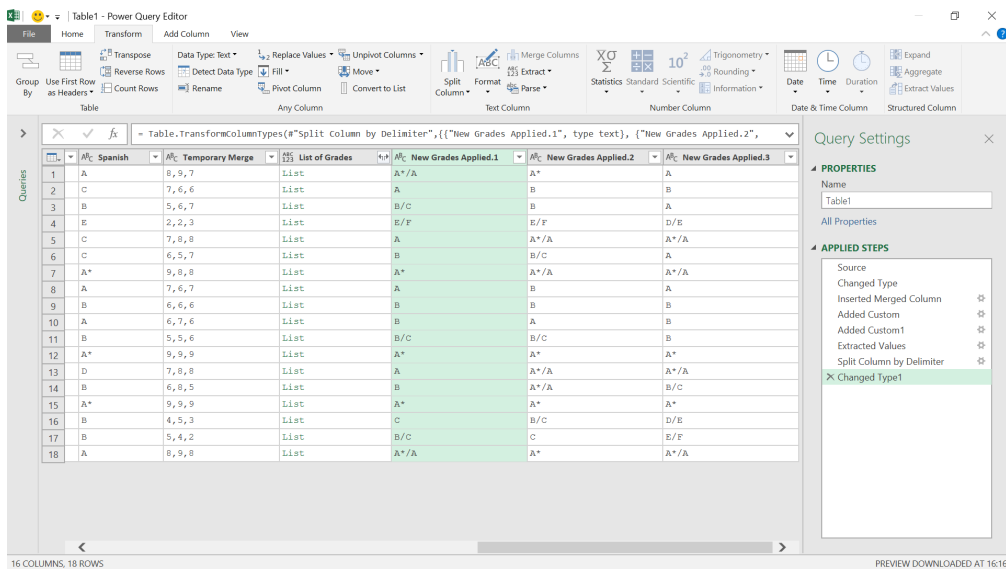
We could have chosen a delimiter here, but we already have a comma ready to split our text.



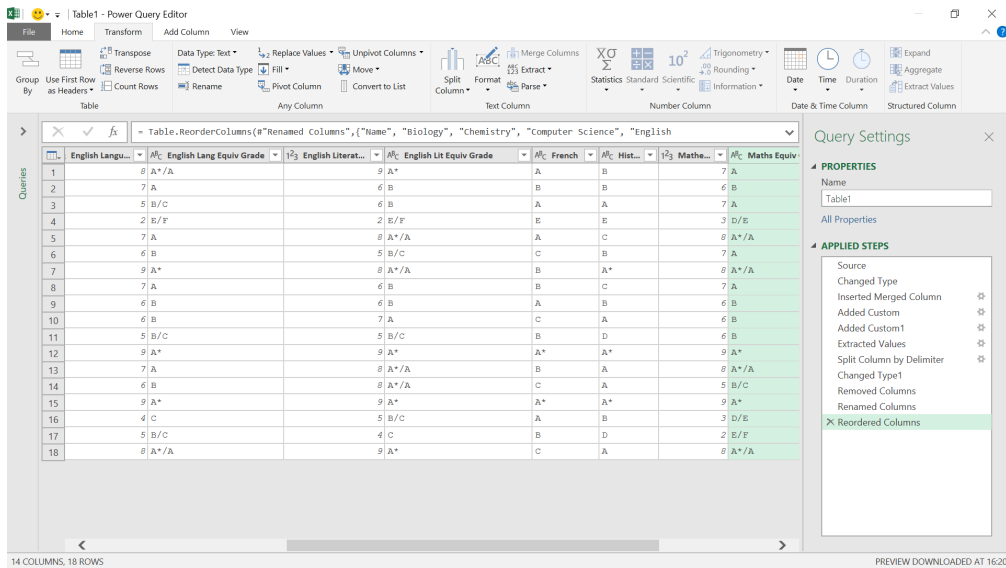
On the 'Transform' tab, we can choose to split my column.



We'll click 'OK' to create three [3] new columns.



We may now remove the other list and the temporary column used to merge the grades. We can also rename the new columns.



We may now see what the equivalent grade is for English and Mathematics. There is often more than one way to achieve a result and next newsletter we'll look at another method which allows for new numeric grades to be added (which is why the examination boards decided to have 9 as the top grade – they can add a grade 10 above this for even higher results!).

More anon.

Power BI Updates

This month's updates see modifications to editing the data models, row-level security, and also provides a reminder to update your .NET Framework version for Power BI Desktop. The full list is as follows:

General

- Upgrade Power BI Desktop .NET Framework to 4.7.2 or newer

Reporting

- Mobile layout interactive canvas
- Smart Narrative improvements

Modelling

- Edit your data model in Power BI Service – updates
- Edit linguistic relationships in Q&A setup
- **DAX** function changes to **MINX** and **MAXX**
- ‘Edit relationships’ in the Properties pane now Generally Available

Data connectivity

- SAP HANA (connector update)
- Emplifi Metrics (new connector)

Service

- Row-level security (RLS) test as role improvements

Mobile

- Supporting AAD Shared device mode (Preview)

Developers

- Git integration:
 - Paginated reports now supported with Git integration
 - Checkout and Switch Branches in a Workspace
 - New E2E CI / CD tutorial now available

Visualisations

- New visuals in AppSource
- Beeswarm chart
- Explore time-based data down to each millisecond with Drill Down TimeSeries PRO
- New updates for accoPLANNING (Release 57)
- Zebra BI charts on-visual settings
- Word Cloud by Powerviz
- Innofalls charts
- Hierarchy chart by MAQ Software
- Power BI JSON Report Theme Generator by BIBB.

Let’s look at each in turn.

Upgrade Power BI Desktop .NET Framework to 4.7.2 or newer

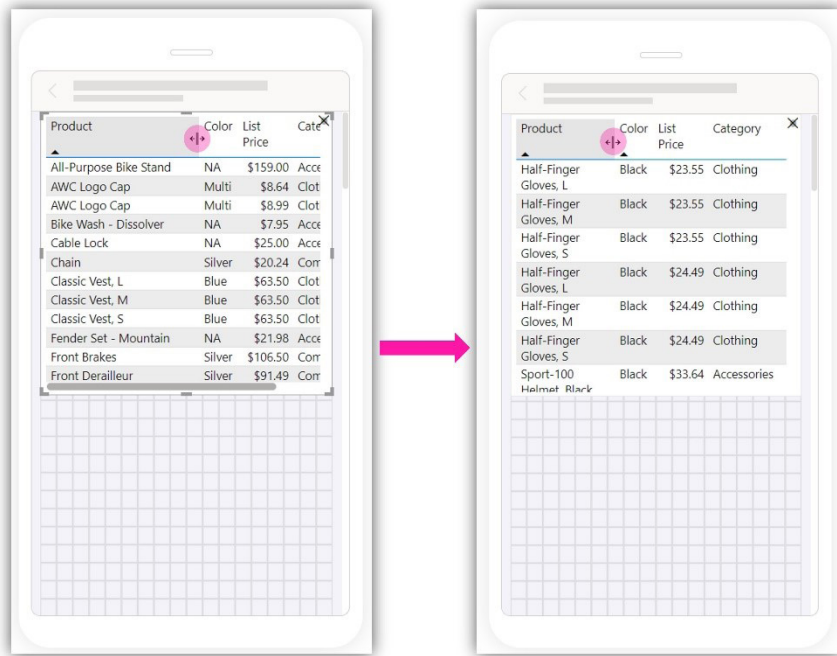
From October onwards, Power BI Desktop with .NET Framework 4.5 will no longer be supported. Microsoft state that you must upgrade to .NET Framework 4.7.2 or newer to avoid any support issues.

Mobile layout interactive canvas

The mobile canvas has now been made interactive. This new capability provides the ability to test how buttons, slicers and visuals will behave in the application before publishing the report.

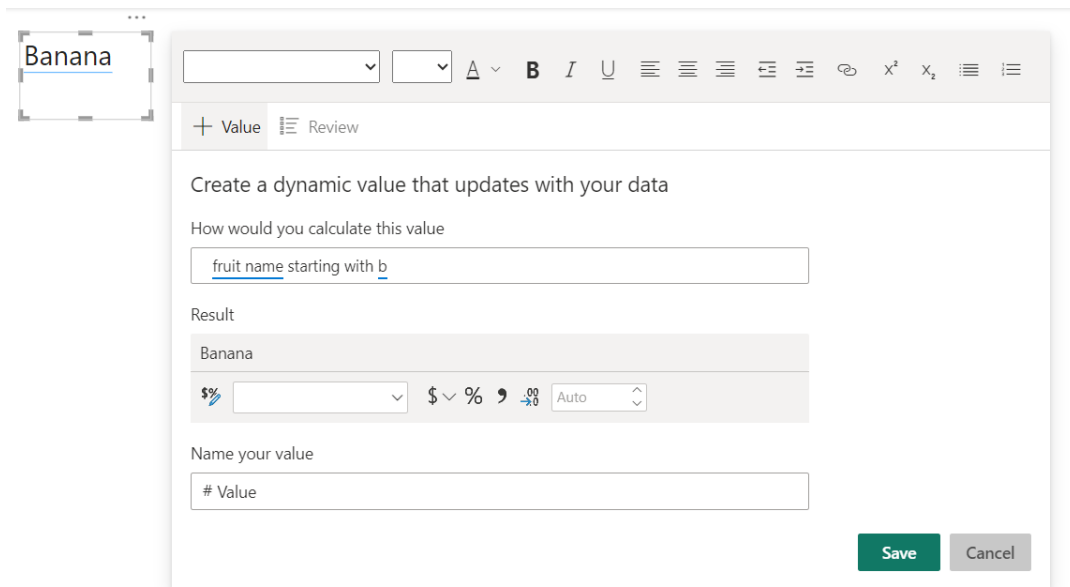


With this canvas interactivity, users may now interact with visuals directly and adjust Table and Matrix column headers to align accurately with mobile screens.



Smart Narrative improvements

The Smart Narrative visual can now display either text or numerical values when showing data for a column. Previously, this was limited to only numerical values.



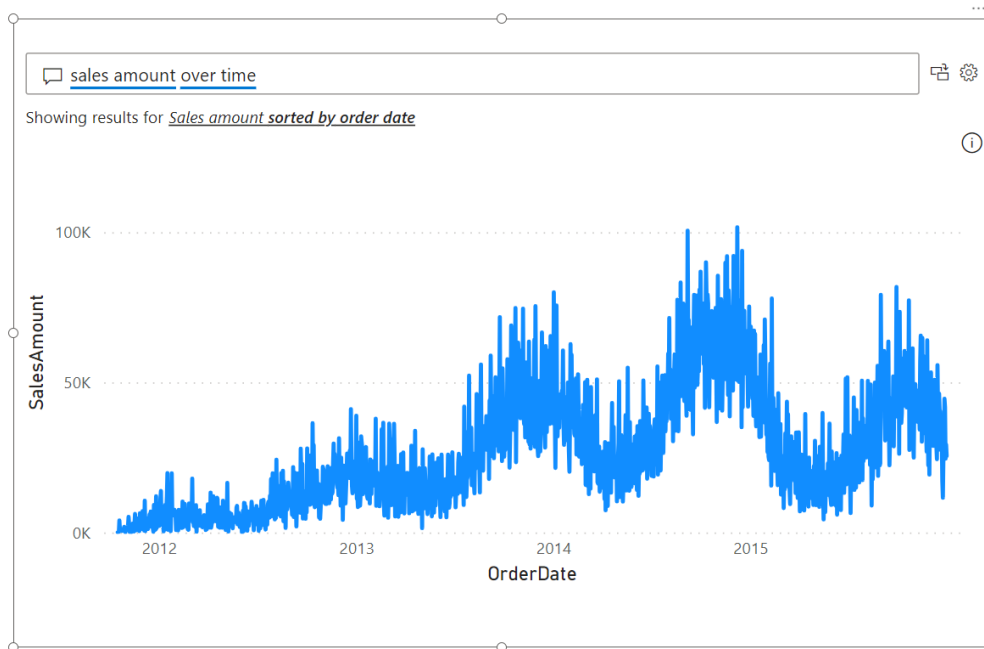
Edit your data model in Power BI Service – updates

The new data model editing in the Power BI Service feature was released to Preview six months ago. The layout limitations have now been improved since changes you make to layouts will now persist between Power BI Desktop and the Service, including:

- changes made to data model layouts in Power BI Desktop will now be incorporated into the Service upon uploading the .PBIX file
- similarly, changes made to layouts in the data model within the Service will now be incorporated in Power BI Desktop upon downloading the .PBIX file.

Edit linguistic relationships in Q&A setup

The Q&A visual is an effective way to help users further understand their data by asking questions and receiving answers in the form of visuals. It offers users a way to explore their data in ways not covered by the rest of the report without requiring deeper knowledge of their data model or report authoring.

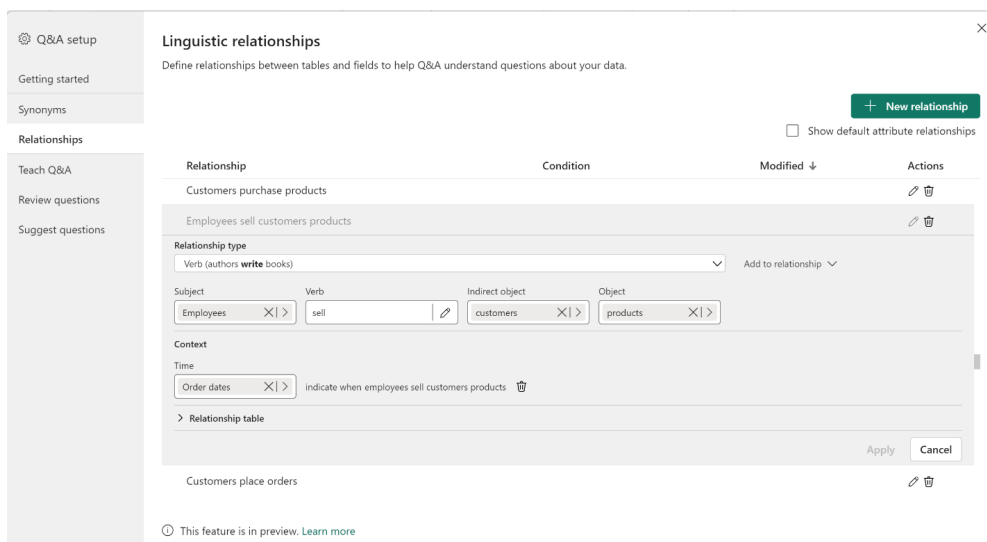


However, while the Q&A engine is good at answering precise questions about data, it may not be able to associate every word or phrase a user inputs with data in the model. For example, answering “what are our best consoles this year?” may require connecting the term “consoles” to the name “products” in the model, and understanding that the term best corresponds to the highest sales values. These terms are contextual. However, users could mean something completely different asking for “console” and “best” in other industries, organisations or even datasets.

To help authors ensure that the Q&A visual provides consistent and accurate answers based upon the unique language their report consumers actually use, Microsoft introduced Q&A setup tools with an emphasis on providing Q&A with synonyms for column and table names in the model. This way, authors can explicitly define console as referring to products and users will always receive the correct answers when they ask similar questions in the future.

However, synonyms (nouns) are only half of the picture. The other half of the terms (adjectives, verbs, prepositions, adverbs) can’t be defined with such straightforward mappings because they must be understood as a part of a phrase: they qualify other terms or relate other terms together. The word “best” in the above previous example is one; asking “who sold the most books” requires us to know that stores sell books, connecting stores to books.

There are many types of these linguistic relationships, so Microsoft has built a new tab entirely to help you create and manage linguistic relationships for your data. You can get into the Q&A setup menu using the gear icon on the Q&A visual or the Q&A setup option in the Modeling tab of the Ribbon, then selecting the new Relationships tab.



There, you’ll be able to define a variety of relationships, including verb, adjective, noun, preposition, and more. You should choose a type which fits the term you’re trying to define (e.g. “best” is an adjective), then follow the prompts to define what it means in the context of your data.

Microsoft states that their investment into Q&A does not stop with this. Natural language capabilities are increasingly driven by large language models. Therefore, there is value in the precision, consistency, and customisability of the Q&A engine. Further, defining synonyms and relationships can be a lengthy process just asking to be streamlined with the power of AI-generated suggestions. Watch this space.

DAX function changes to MINX and MAXX



This update adds an optional variant parameter to **MINX** and **MAXX** DAX functions. These functions ignore text and Boolean values when there are variants or mixed data types, such as text and numerical. Now, with the new optional variant parameter set to **TRUE**, the functions will consider the text values.

Here is an example of a variant measure. In the table below, the variant measure is ordered in ascending order:

```

1 Variant Measure =
2 SWITCH(
3     MIN('Table'[Column1]),
4     1, CURRENCY(1.1),
5     2, "A",
6     3, 3,
7     4, DATE(2023,1,1),
8     5, blank(),
9     TRUE()
10 )
    
```

Data Type	Variant Measure	MINX	MINX with Variant	MAXX	MAXX with Variant
Currency	\$1.10	\$1.10	\$1.10	\$1.10	\$1.10
Number	3	3	3	3	3
Date	1/1/2023	1/1/2023	1/1/2023	1/1/2023	1/1/2023
Text	A		A		A
Boolean	False				
Total	\$1.10	\$1.10	\$1.10	1/1/2023	A

MINX and **MAXX** without the variant parameter set, or set to **FALSE**, will ignore text and Boolean data types.

```

1 MAXX =
2 MAXX('Table',[Variant Measure])
    
```

Data Type	Variant Measure	MINX	MINX with Variant	MAXX	MAXX with Variant
Currency	\$1.10	\$1.10	\$1.10	\$1.10	\$1.10
Number	3	3	3	3	3
Date	1/1/2023	1/1/2023	1/1/2023	1/1/2023	1/1/2023
Text	A		A		A
Boolean	False				
Total	\$1.10	\$1.10	\$1.10	1/1/2023	A

MINX and **MAXX** with the variant parameter set to **TRUE** will now include text values. Boolean values are still ignored.

✕ ✓

1 MINX with Variant =
 2 | MINX('Table',[Variant Measure],TRUE)

Data Type

 Select all
 Boolean
 Currency
 Date
 Number
 Text

Data Type	Variant Measure	MINX	MINX with Variant	MAXX	MAXX with Variant
Currency	\$1.10	\$1.10	\$1.10	\$1.10	\$1.10
Number	3	3	3	3	3
Date	1/1/2023	1/1/2023	1/1/2023	1/1/2023	1/1/2023
Text	A		A		A
Boolean	False				
Total	\$1.10	\$1.10	\$1.10	1/1/2023	A

The default of the **MINX** and **MAXX** optional variant parameter is **FALSE**, so all existing **DAX** expressions using **MINX** and **MAXX** will not be impacted by this change. To allow text values to be included, you can update the **DAX** expression to include **TRUE** as the third parameter.

'Edit relationships' in the Properties pane now Generally Available

'Edit relationships' in the Properties pane is now Generally Available. The 'edit relationship' in the Properties pane has been available for public Preview since the October 2022 release. Now, you can click on any relationship line in the modelling section to edit your relationship in the Properties pane. This is available in the Model view of Power BI

Desktop and in 'web modelling' of the Power BI Service. You may use 'Edit relationships' without running queries to preview the data and only validate the relationship when you click apply changes. For those using DirectQuery storage mode, this will be especially welcome as these queries can take time and impact your data sources.

SAP HANA (connector update)

The update enhances the SAP HANA connector with the capability to consume HANA Calculation Views deployed in SAP Datasphere by taking into account SAP Datasphere's additional security concepts. This enables

consumption of Calculation Views in Datasphere and allows customers to connect to HANA Cloud views without the need for additional privileges on the `_SYS_BI` schema.

Emplifi Metrics (new connector)

With Emplifi Power BI Connector, you will be able to include social media data from the Emplifi Platform in your charts and graphs and combine them with other data you own. The Power BI Connector is a layer between Emplifi Public API and Power BI itself. It helps you work with

your data intuitively, directly in the Power BI tool. The majority of data and metrics available in the Emplifi Public API are also available in the connector.

Row-level security (RLS) test as role improvements

This month's updates see a new dropdown added, allowing you to choose and test any report connected to the dataset.

Viewing: Adventure Works DW 2020 (1) ▾
Now viewing as: Example role ▾

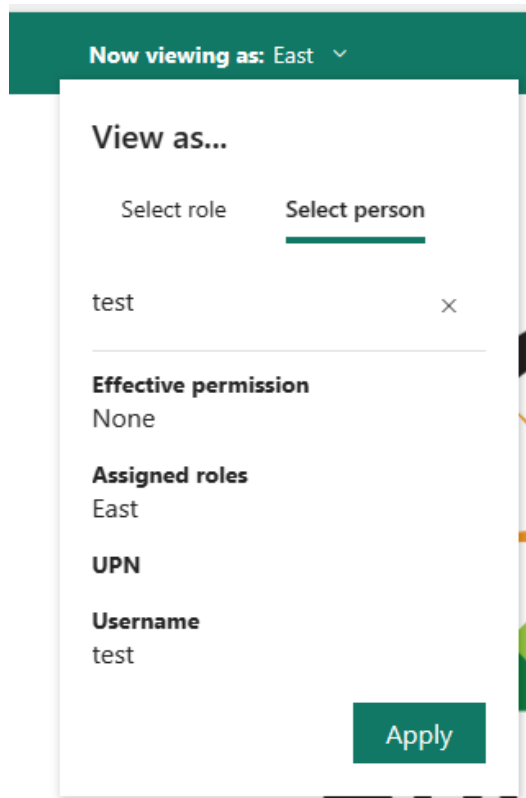
Select a report to view

Choose a report created from the dataset Adventure Works DW 2020

Report	Workspace
✓ Adventure Works DW 2020 (1)	My workspace
Example scratchpad report	My workspace

Confirm
Cancel

Also, you can now easily see important permissions details pertaining to a specific individual while role testing for that person.



Warning and error messages have also been improved – allegedly!

Supporting AAD Shared device mode (Preview)

This month, the Power BI mobile applications now add support in Azure active directory shared device mode. Shared Device Mode enables you to configure devices to be shared by a group of employees. This is very common for frontline workers, who do not use their personal device for work related tasks but are getting a work-related mobile device from a shared pool of devices for these tasks.

The Shared Device Mode enables single sign-on (SSO) and device-wide sign out for Microsoft Power BI and all other apps that support Shared Device Mode. Once a user signs in into a supported application, the user will be logged into any application compatible with Shared Device Mode as well.

Once a user signs out of a supported application, all other applications integrated with shared device mode sign out, to prevent unauthorised or unintended access by the next user.

Furthermore, when an employee picks a device from a pool at the start of their shift, they need only a single sign-in to one supported application (e.g. Power BI), and they're automatically signed in to all other applications that support shared device mode on the device, so the device and the apps are ready for work with the user's account. At the end of their shift, when they sign out of one application, they are signed out globally from all other applications that support shared device mode, having the device ready for the next employee.

Git integration

Several updates have been added here.

PAGINATED REPORTS NOW SUPPORTED WITH GIT INTEGRATION

Since git integration has been launched a few months ago, Microsoft has supported only two [2] items in Power BI: Power BI reports and Power BI datasets. But now there is a third item: Paginated reports

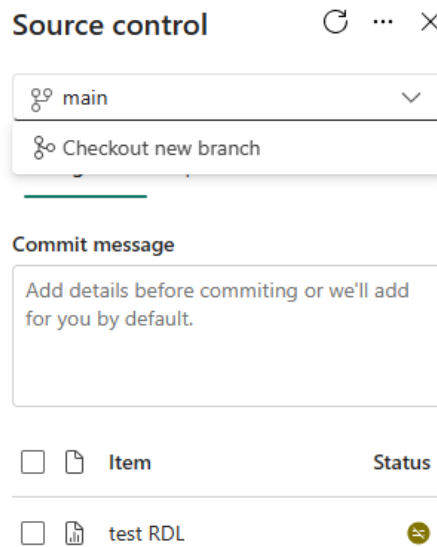
After connecting your workspace to Azure DevOps, you can commit your paginated reports and have them versioned in your repository. After

that, you can open the .rdl files directly from git in Power BI Report Builder, edit and push the changes into git. The workspace will identify the changes and will prompt developers to update the workspace with the changes to the paginated report.

CHECKOUT AND SWITCH BRANCHES IN A WORKSPACE

A workspace can connect to a single branch at a time. Once you've setup your separate workspace to work with git, there might be scenarios where you want to change just the connection of a branch. Microsoft has now added two [2] new features to help you achieve this much faster:

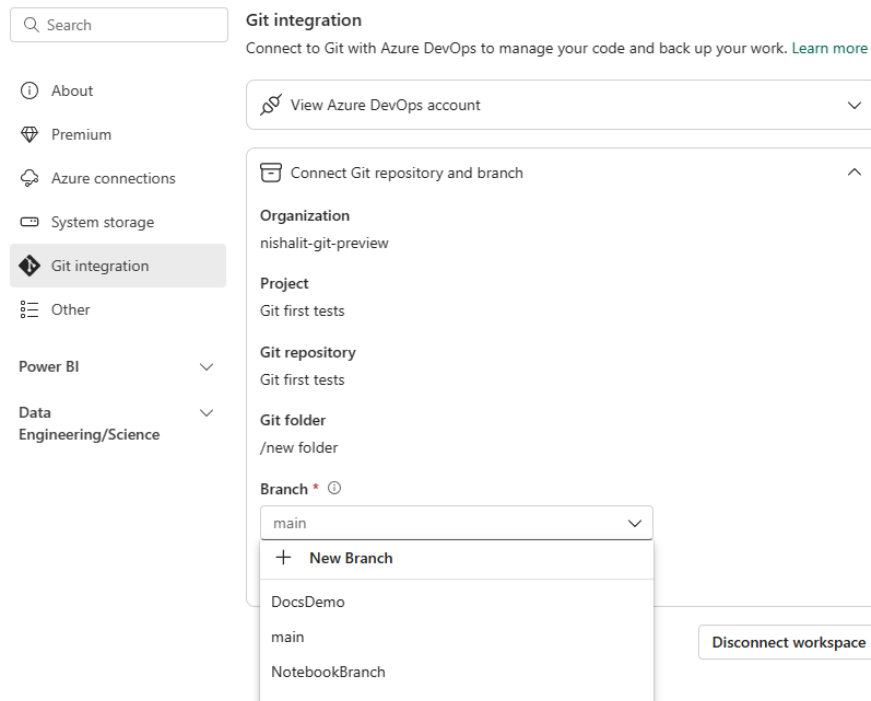
1. **checkout branch:** checkout allows you to easily create a new branch, based upon the current state of the workspace.



After choosing to checkout, you can create a new branch that will be connected to the WS, while the uncommitted changes are retained. This is useful in cases of conflicts, that allows you to commit your changes to a backup branch, and then manage the merge conflict in the git repo

2. **switch branch:** workspace administrators may decide to change just the connection of the branch very easily.

Workspace settings



📁 Connect Git repository and branch
⤴

Organization
nishalit-git-preview

Project
Git first tests

Git repository
Git first tests

Git folder
/new folder

Branch * ⓘ

NotebookBranch
▼

Switch and override
Cancel

Disconnect workspace

It is helpful in cases you want to start working on a new branch and wipe clean your WS content for that or when you are moving from your regular project to small bug fixes (assuming they are on the same content), that should happen on a different branch.

NEW E2E CI / CD TUTORIAL NOW AVAILABLE

Microsoft has stated that many users have been asking, since the release of git integration, how to work with deployment pipelines and git as part of a continuous integration / continuous delivery (CI / CD) process. These two [2] features can help build a compelling end-to-end (e2e) flow, when connected properly.

New visuals in AppSource

This month's new visuals include:

- Activity Gauge by Powerviz
- Aimplan Data Input Table
- Apex Milestone Trend
- Bar Chart with Export Data Feature
- Box Ploty by Devlup Funnels
- Bridger Visual by BI Samurai
- Change Chart
- Date Picker by Powerviz
- Dot Chart
- Funnel Chart by Powerviz
- Number lines by BI-Champ
- PackedBubbleChart
- Pro Circle Card by Devlup Funnels
- LeapLytics by Markdown Viewer
- Wordcloud by Powerviz.

Beeswarm chart

A Beeswarm chart shows the distribution of data along one axis or both axes whilst also showing individual points. It is like a one-dimensional or two-dimensional scatter plot but with closely-packed, non-overlapping points.

The visual is intuitive and easy to follow. Only one [1] field is mandatory, *i.e.* Bee Category. If you don't have any category, you can simply add a calculated column with any name and use it in the category field instead.

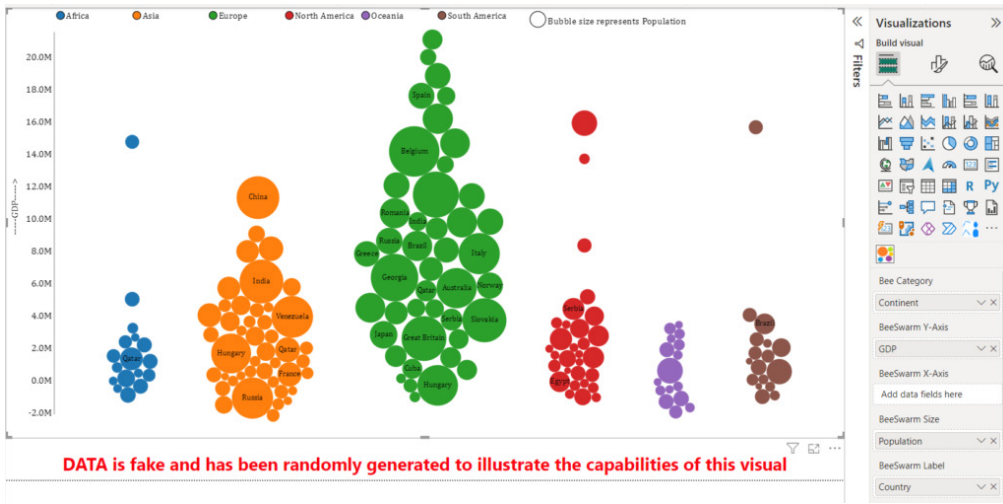
The Bee Size field is optional and is represented in the visual by bubble area or size. If this field is missing, then all bubbles will be of the same size.

The Bee Label field is also optional. It is used to add labels to each individual bubbles.

This chart may be used in three [3] ways as follows.

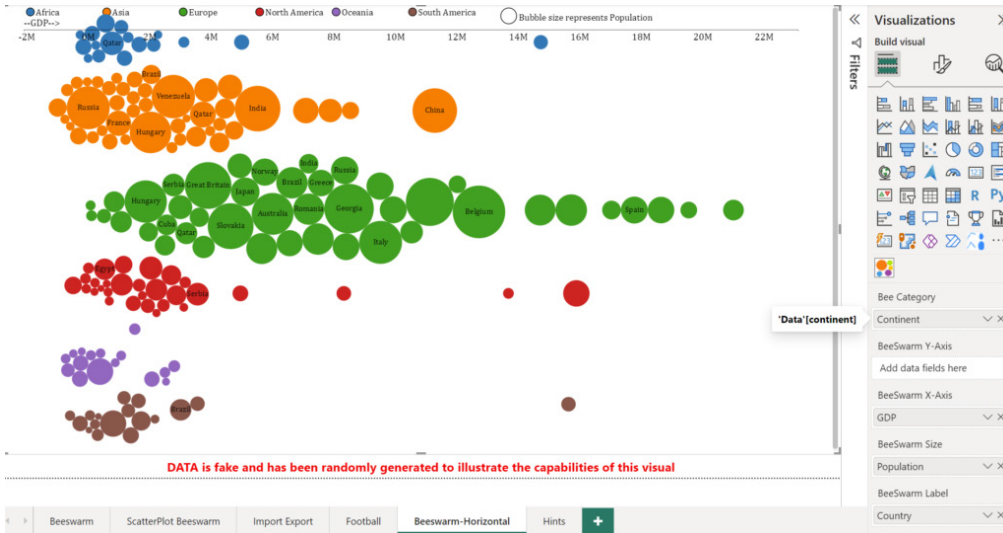
1. Y-AXIS ONLY

As shown in the image below, if you only use a column / field in the 'Beeswarm y-axis' section while keeping the 'Beeswarm x-axis' section empty, your data will be plotted on the y-axis, categorized according to the 'Bee Category' section.



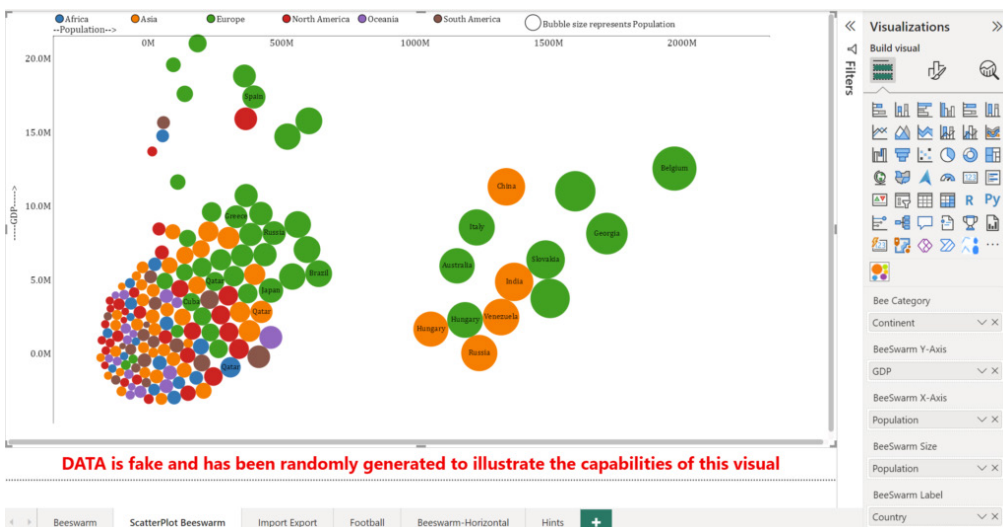
2. X-AXIS ONLY

Similarly, if you only use a column / field in the 'Beeswarm x-axis' section whilst keeping the 'Beeswarm y-axis' section empty, your data will be plotted on the x-axis categorised according to the 'Bee Category' section.

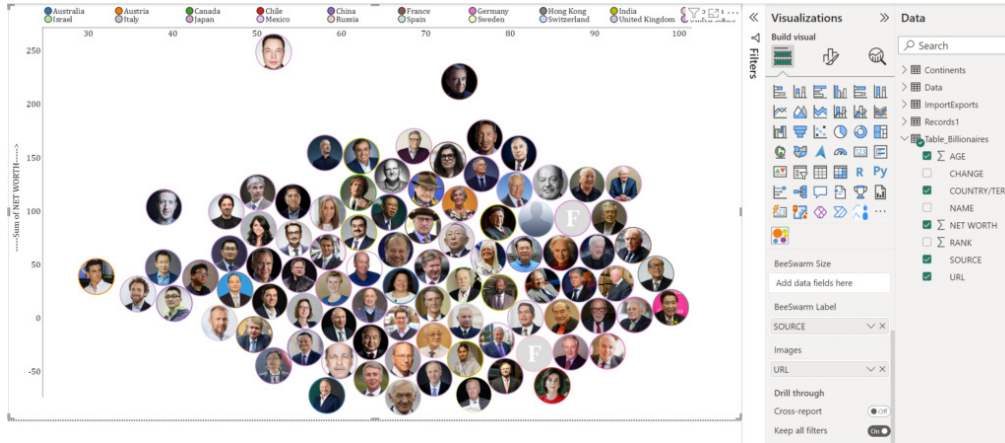


3. SCATTER PLOT (BOTH X AND Y AXES)

If you use columns / fields in both sections, *i.e.* the 'Beeswarm x-axis' section and the 'Beeswarm y-axis' section, the visual will work like a scatter plot with data points plotted on the x-axis and the y-axis.



You can show images inside bubbles as well using the 'Images' field. For example, the report below shows the top millionaires of the world with their net worth on the y-axis and their age on the x-axis.



Formatting options are available in the Format pane of the visual under 'Settings'.

1. **Bubble size:** as the name implies, this setting simple increases / decrease the bubble size
2. **Lower Upper and Left Space:** the Beeswarm chart produces non-colliding bubbles, *i.e.* circles that do not overlap.

This can often cause bubbles do go outside the chart area, especially when there is a concentration of data around minimum and maximum data values. This is where these settings are very useful to bring data inside the chart area. These settings increase or decrease the y-axis and the x-axis ranges to achieve this objective.



You can download the custom visual from AppSource – just like all of the following ones.

Explore time-based data down to each millisecond with Drill Down TimeSeries PRO

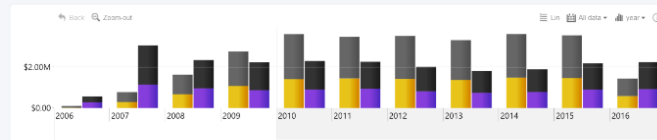
Drill Down TimeSeries PRO allows you to create time series charts that are easy to explore on any device. Using its interactions, users may drill down to months, days or hours by clicking on the chart. You can combine up to 12 series and choose between multiple charts, such as Line, Column and Area.



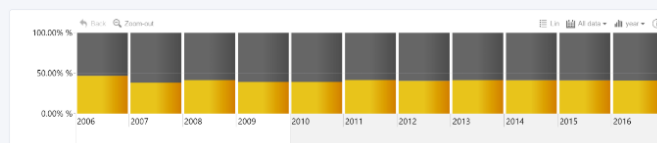
Normal Stack



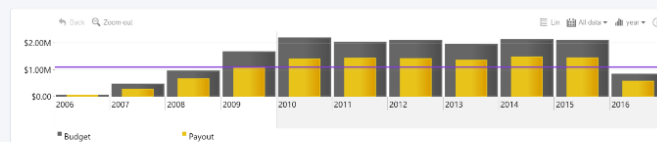
Normal Stack + Clustering



100% Proportional



Zero Based Stack



zoomcharts.com/powerbi

Main features include:

- **on-chart interactions:** pan, zoom and drill down
- **full customisation:** set intervals, minimum / maximum values, colours, fonts, stacking and clustering
- **static and dynamic thresholds:** set up to four [4] thresholds to demonstrate Key Performance Indicators (KPIs)
- **cross-chart filtering:** select data points on multiple charts instead of using slicers
- **touch device friendly:** explore your data anywhere.

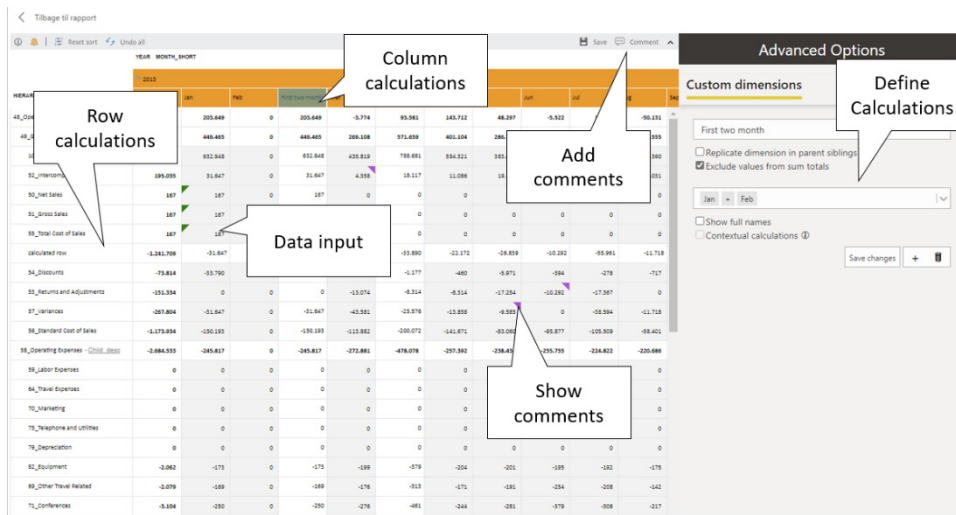
Popular use cases include:

- **Finance:** stock exchange indices, capital ratios, transaction volumes
- **Sales:** web traffic, audience reach, lead volume
- **IT:** network traffic, response times, syslog and error trends
- **Logistics:** inventory movement and turnover, loading time, fleet maintenance costs.

New updates for accoPLANNING (Release 57)

accoPLANNING for Power BI provides business users with writeback capabilities, advanced planning, forecasting, budgeting, project management and analysis solutions. There are new features with this release:

- **custom calculated rows and columns configuration.** This allows you to customise your own client-side calculations in your already established accoPLANNING table
- **end user Lock cells.** This gives the flexibility for the end user to lock cells and make sure these will not be affected by splashing or use as an approval indicator
- **hide rows and columns.** This allows users to hide rows and columns adding flexibility in making asymmetric column and row selections in the grid for better reporting or just for *ad hoc* purpose to get a better overview.

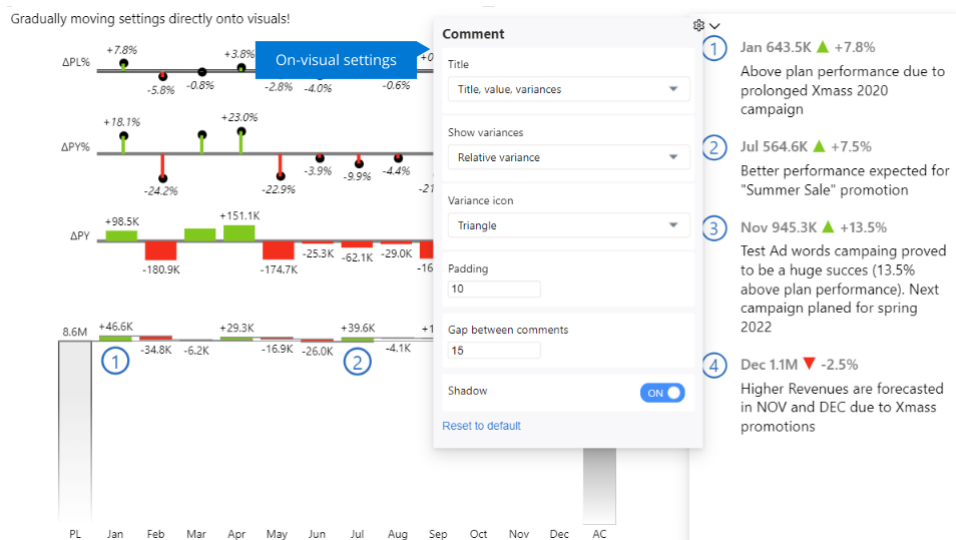


The latest update also includes a range of new features and improvements designed to increase productivity:

- navigation has undergone a major overhaul, with the most notable change being the revamped toolbar
- better API error message handling in the grid
- support for automatically expanding all rows / columns / both, eliminating the need to manually adjust the size of each cell.

Zebra BI charts on-visual settings

Zebra BI Charts has received some important updates. With the on-visual settings, you don't need to go to the Visualizations (*sic*) pane every time you want to make a change.



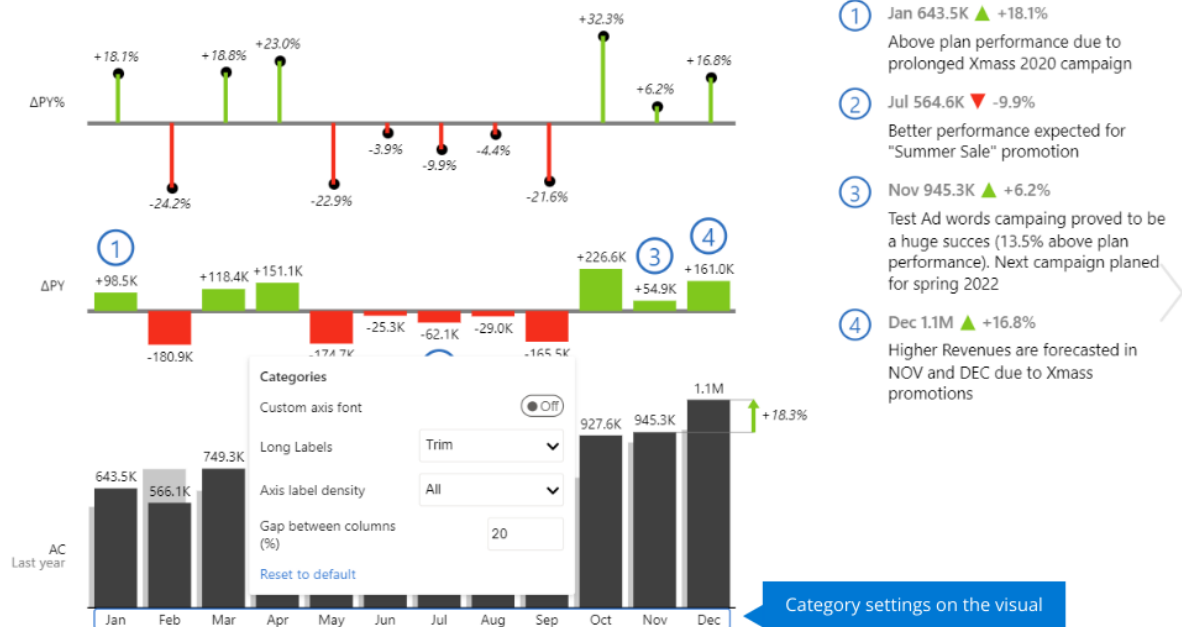
Comments settings can now be adjusted directly on the visual by simply clicking on the settings icon next to the comment box. You can adjust the title, which variances you display, the icon, gap between the comments, and padding.

Category settings let you adjust several settings by just clicking on the category area. You may customise the axis font (family and colour), trim / rotate long labels, adjust the label density and set the gap between the columns.

Legend settings come in handy when you want to rename the entries and don't want to search for this option in the Formatting pane. Adjust margins, use aliases in tooltips and switch comparisons.

Stacked charts settings on the visual let you adjust the Top N feature, set colour of the chart and display labels as percentages.

New on-visual settings



- ① Jan 643.5K ▲ +18.1%
Above plan performance due to prolonged Xmass 2020 campaign
- ② Jul 564.6K ▼ -9.9%
Better performance expected for "Summer Sale" promotion
- ③ Nov 945.3K ▲ +6.2%
Test Ad words campaign proved to be a huge succes (13.5% above plan performance). Next campaign planned for spring 2022
- ④ Dec 1.1M ▲ +16.8%
Higher Revenues are forecasted in NOV and DEC due to Xmass promotions

Word Cloud by Powerviz

There is a new Word Cloud by Powerviz, which is a visual which allows you to create high-quality and creative word art in Power BI.



Key features include:

- **word styling:** make your word clouds pop with personalised text styles. It offers font styling, direction and text editing features
- **colour options:** choose from 30+ colour palettes, including colour-blind safe options
- **shapes:** want to make a statement? You may create word clouds by choosing shapes from icons and images, or upload your own image
- **exclude:** easily remove unwanted words, symbols from the text to create a clean and focused word cloud
- **ranking:** filter out Top / Bottom N Words
- **conditional formatting:** easily spot words with dynamic rules.

Many other features included lasso / reverse lasso, grid view, show condition and accessibility support.

CREATE EYE-CATCHING & STUNNING WORD CLOUDS

CAPTURE EVERYONE'S ATTENTION WITH POWERFUL DESIGNS

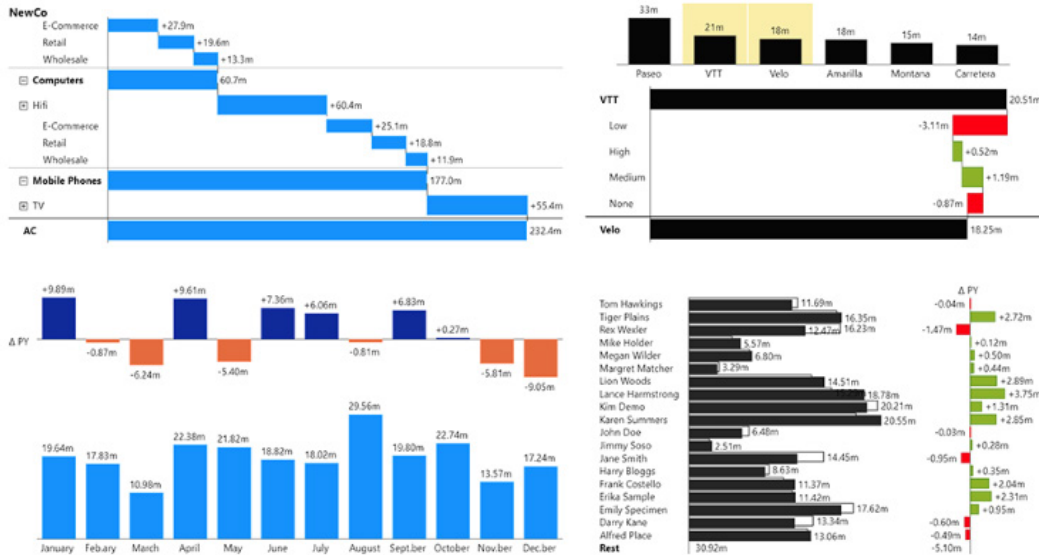


Business use cases include:

- **Marketing:** sentiment analysis and SEO keywords
- **Education:** brainstorming, improve engagement
- **Market Research:** opinion poll, open-ended survey
- **Presentation:** capture attention and easy communication.

Innofalls charts

Innofalls charts are a visual that offer a range of interactive features. The Bar and Waterfalls charts come with a drill-down history, enabling orientation and facilitating comparisons. All charts may be stacked or grouped into columns.



Waterfalls: includes running totals and delta waterfalls, enabling effective comparisons of measures or members. The waterfalls are expandable, drillable, stackable and offer intermediate sums.

Drill down history: you may keep track of your drill downs with visual feedback. You can change your drill downs retroactively and combine multiple drill downs for in-depth exploration.

New comparison features: you may make comparisons while viewing your report and incorporate deviation charts with a single click.

Interactivity: there are various interactive features such as drill, expand, undo / redo, dynamic comparisons and zoom for enhanced data exploration.

IBCS (International Bureau Communications Standards) formatting: the IBCS theme provides scenario formatting, equal scaling, highlights, deviations and more for consistent and standardised reports.

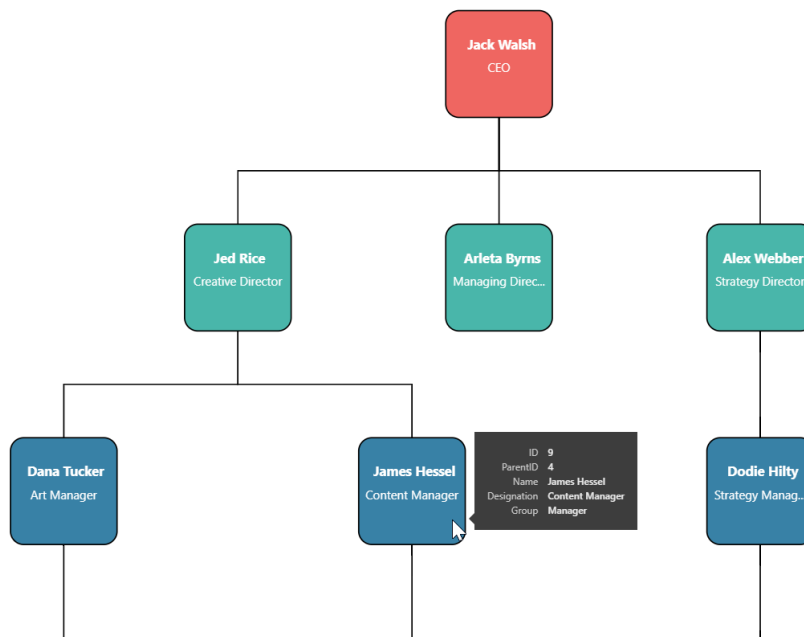
Animations: you can enhance data comprehension and reduce change blindness with animations. Promptly detect resorted or added data elements. It should be noted these charts are free for Power BI Desktop.

Hierarchy chart by MAQ Software

In today's data-driven environment, visualising hierarchical relationships can be key to business insights and decision-making. From sales structures to budget allocations, this Power BI-verified visual offers a customisable way to represent these structures.

Organization Chart

Departments: ● CEO ● Director ● Manager ● Executive ● Engineer ● Designer ● Writer ● Tester



Key business uses include:

- **Organisation / HR:** Organise human resources by department and hierarchy
- **Operations:** illustrate manufacturing processes, breaking down components or ingredients hierarchically
- **Finance:** show budget allocations by division or project, with colour-coded subcategories to highlight differences
- **Sales:** visualise sales structure, territory responsibilities and targets with group field indicators showing performance status
- **IT:** visualise IT assets by category and availability with detail.

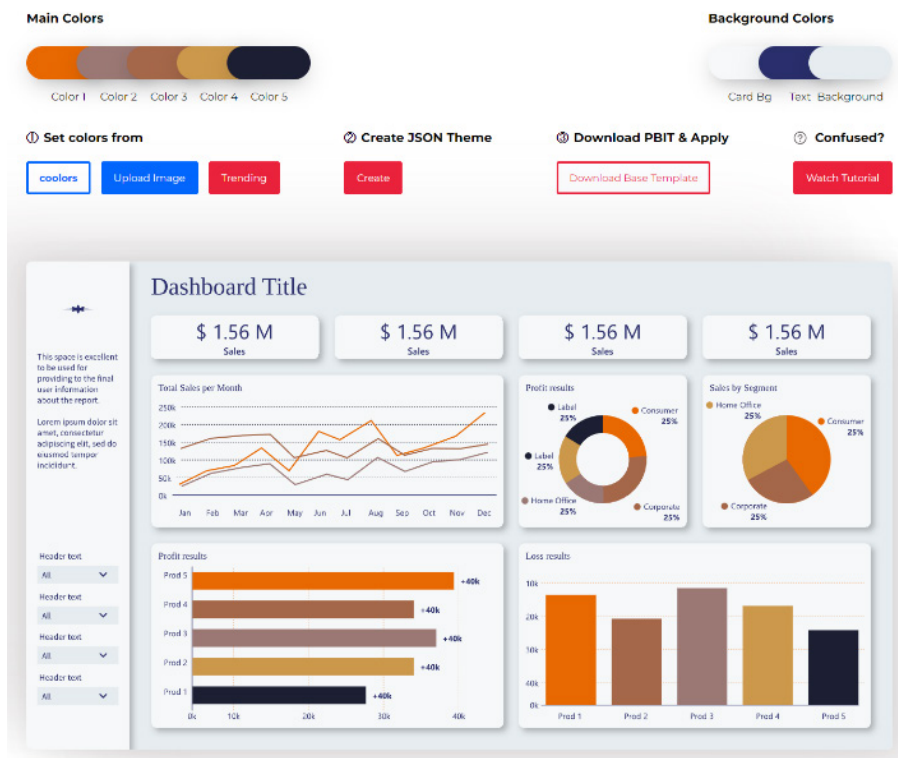
Key features include:

- colour-code cards using a legend
- adjust card (*i.e.* box) size, borders and corners
- customise the appearance of the links connecting each card
- interact with cards to control levels and cross-filter visuals
- zoom and reposition the visual with ease.

Power BI JSON Report Theme Generator by BIBB

BIBB recently released its take on the Power BI Theme Generator, aiming to streamline the theming process for users. This new tool offers a user-friendly interface, allowing for an easy selection of colours and

generation of JSON themes. Within BIBB's generator, users may choose colours in various ways, from manual selection to trending combinations and even importing from images or external sources like Coolors.



More next month!

New Features for Excel

Now, yes, some of these items we divulged in our last newsletter, but we repeat them here because we want to be completist. Now, yes, some of these items we divulged in our last newsletter, but we repeat them here because we want to be completist. Now, yes, some of these items we divulged in our last newsletter, but we repeat them here because we want to be completist.

However, there are other additions for this month's updates. You can check out the full list here:

Excel for the web

- Interactive PivotCharts
- Simplified Sharing across Microsoft 365
- Access and Explore Office Add-ins

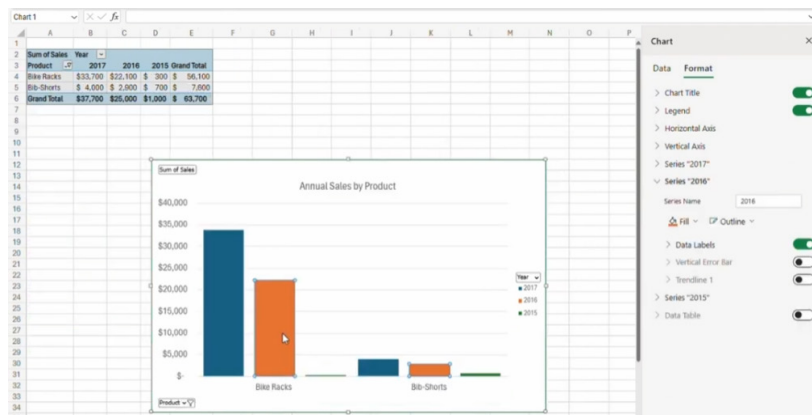
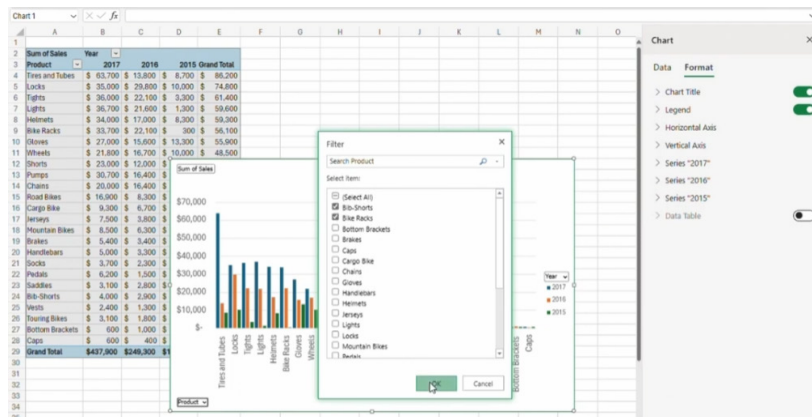
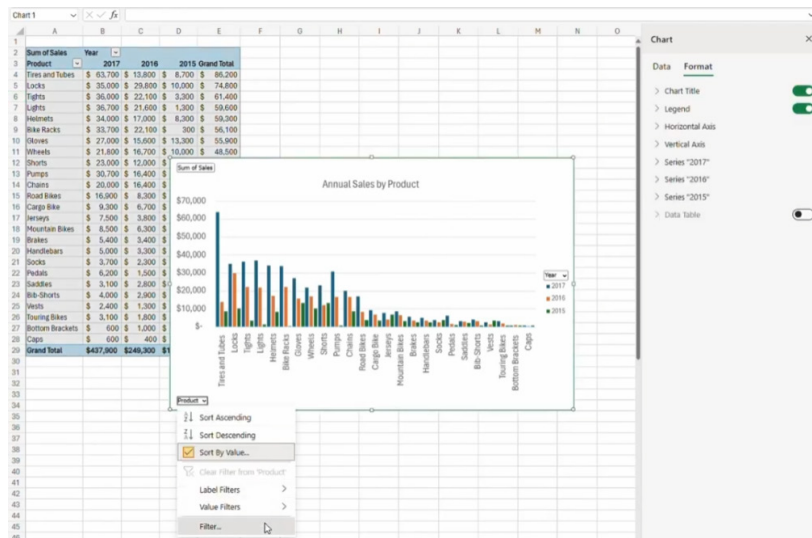
Excel for Windows

- Python in Excel (Insiders Beta)
- Stale Value Formatting (Insiders Beta)
- Monospaced Font in Formula bar (Insiders Beta)
- Power BI Connected Tables
- Improved Save As / Discard Dialog
- Simplified Sharing across Microsoft 365
- Access and Explore Office Add-Ins (Insiders).

Let's get started.

Interactive PivotCharts

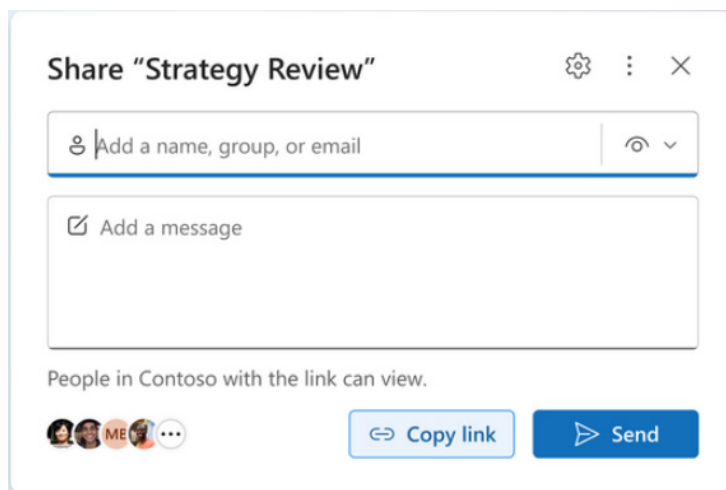
PivotCharts in Excel for the web are now interactive with an updated look and feel for buttons. Simply select chart elements to launch and navigate the 'Format' task pane to make changes.



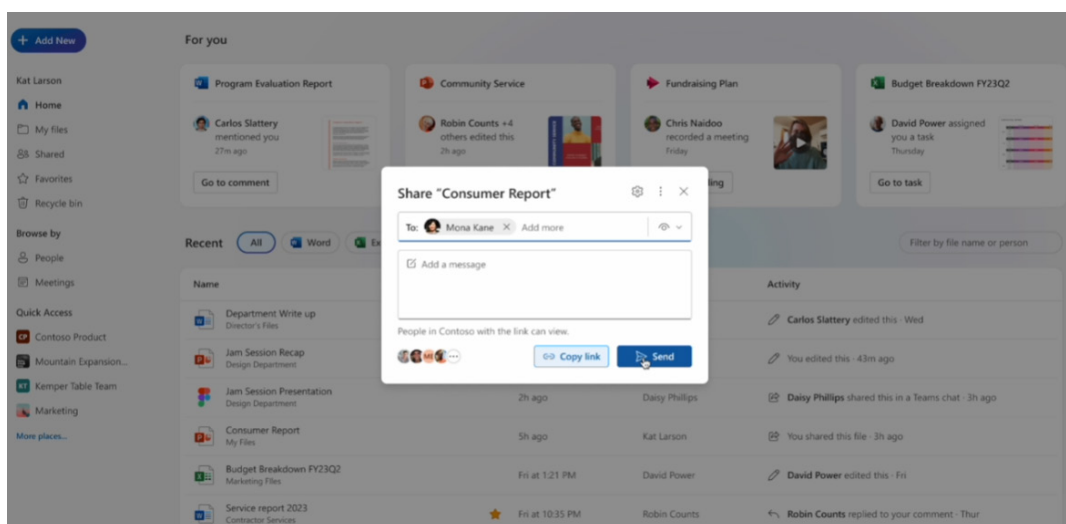
Simplified Sharing across Microsoft 365

Sharing and collaboration are key focal areas of Microsoft 365, empowering you to work with coworkers and partners on files and documents. Collaboration across Excel, PowerPoint, Teams, Outlook, OneDrive, SharePoint, Word and other applications maximises your efficiency and productivity at work.

Now in Excel for the web and Excel for Windows, when you click the 'Share' button, the 'Share Dialog' box appears which has now been improved. With this update, the 'Simplified Sharing' experience is available across more than 50 Microsoft 365 applications on all platforms.



As part of the 'Simplified Sharing' experience, the 'Share' dialog has been streamlined to be free of distractions.

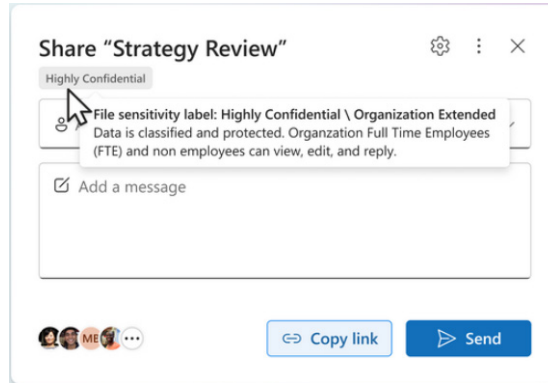


The following have now been simplified:

- **email a link:** when you want to email someone a link to a file, simply start typing a name, group or email and you'll be given suggestions based on your most frequent collaborators. You also have the option to add a brief message. Once you hit 'Send', OneDrive will craft an e-mail and send it to your chosen recipients
- **copy a link:** if you want to send your file in a different application, such as Teams or Outlook, you can always copy a link to your clipboard. Microsoft has made it easier to create and copy a link with a single click and see a quick confirmation. Once copied, you can paste it into a Microsoft Teams chat, document, or email to give others access to the file. If you want more granular control around how others can access your file, such as giving them edit, view, review permissions or if you want to block downloads, just click the gear icon for further customisation
- **share files externally:** you can also share links easily with people external to your organisation. The new streamlined experience asks for your confirmation before creating an external share link while respecting admin settings and external policies, making it a lot easier and faster than the previous experience.

It goes further though. This update is also making it easier to view and control who has access to shared files or folders and what permissions they have:

- **manage access:** at the bottom of the 'Share' dialog, you'll see the people who have access to the shared file. Clicking on the people's profile pictures or initials will take you to the 'Manage Access' dialog which shows a comprehensive list of people and groups who have access to the file or folder, along with their respective permissions. The 'Manage Access' dialog also lets you grant additional access, stop sharing or set controls for how recipients access the files and folders you've shared with them
- **links:** clicking on the 'Links' tab will surface a view of all the sharing links that have been created for the file or folder and who has access via each link. From the links view you can fine-tune who has access and what permissions they have



- **sharing sensitive files:** some organisations want to handle sensitive information with an extra layer of protection. Sensitivity Labels in Microsoft 365 allow you to classify files based on their level of confidentiality, enabling better control over access and distribution. This has now been integrated these labels into the 'Share' and 'Manage Access' dialogs. These labels, specific to each file, give you at-a-glance understanding of file sensitivity before sharing, helping to ensure security for your sensitive documents.

Access and Explore Office Add-ins

For Excel for the web and Excel for Windows, there is a new button on the Home tab you can use to find, install and access Office add-ins.

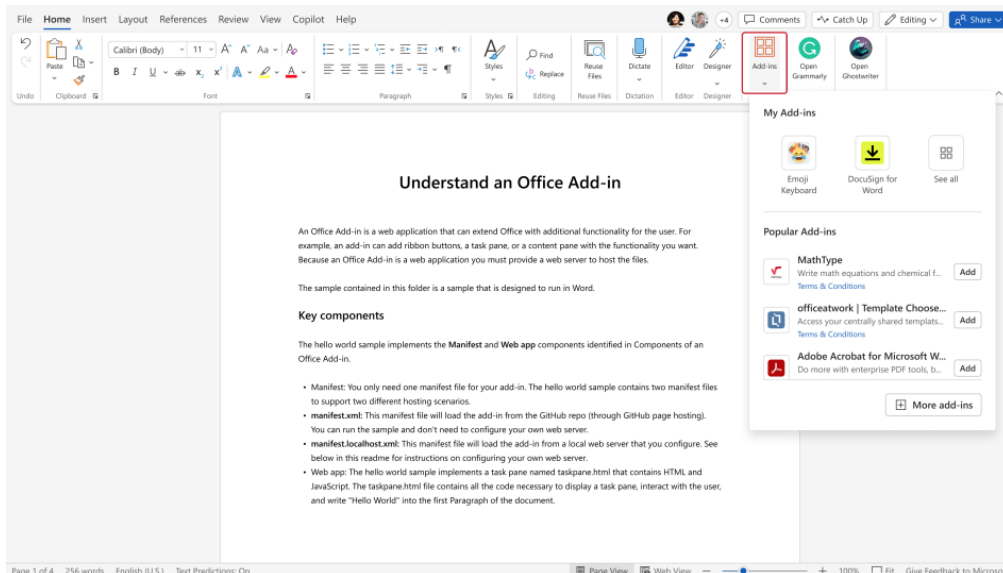
Office add-ins are software extensions that integrate third party services and add extra value to Microsoft Office applications such as Excel. For example, an Office add-in might allow you to insert external content

such as pictures, equations, and text into documents easily, or allow you to import or export data from external services to Excel and analyse data in an easy way. Most Office add-ins are developed by third-party developers and can be downloaded and installed from the Microsoft Office Store.

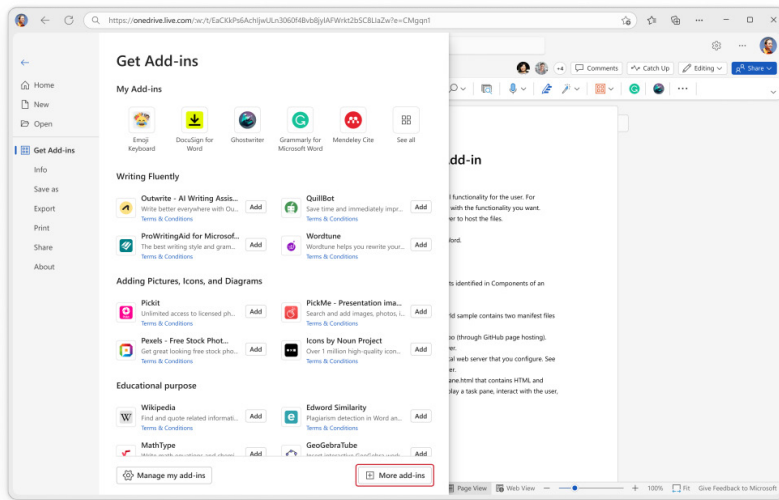
It works as follows:

- create or open an existing document, worksheet or presentation
- go to **Home** -> **Add-ins**.

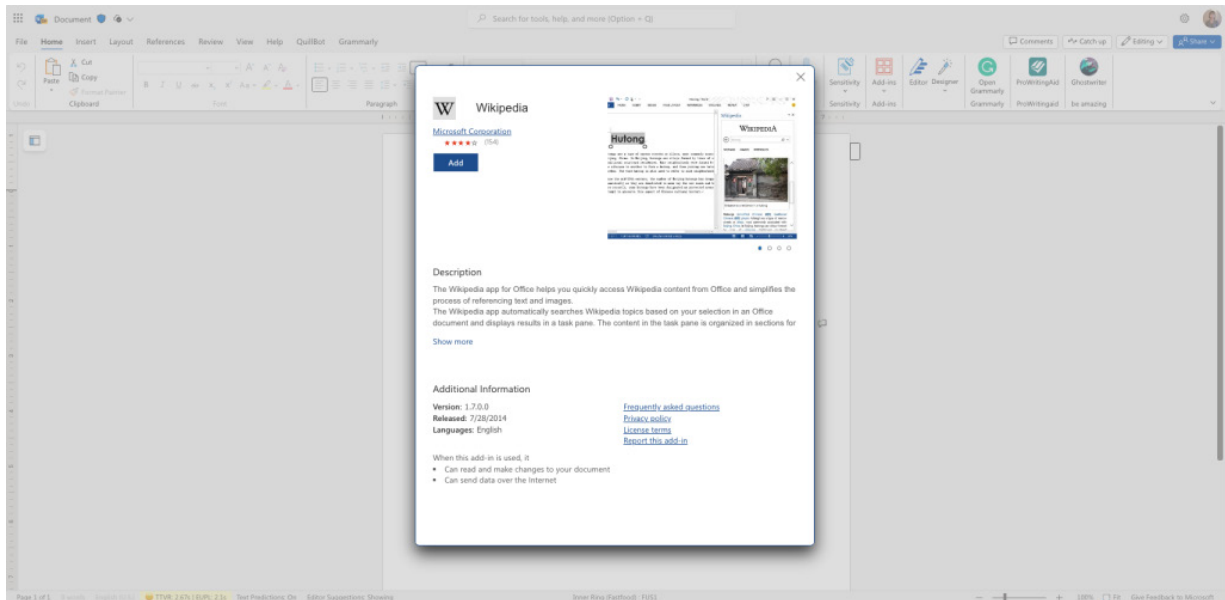
An illustration is shown for Word:



It should be noted that you can also select **File -> Get Add-ins**.

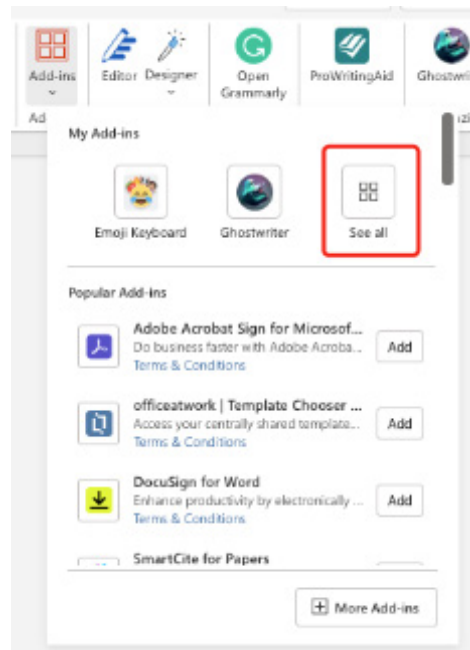


The add-ins you installed appear in the 'My add-ins' section. The 'Get Add-ins' screen appears with descriptions and star ratings from Microsoft 365 users.

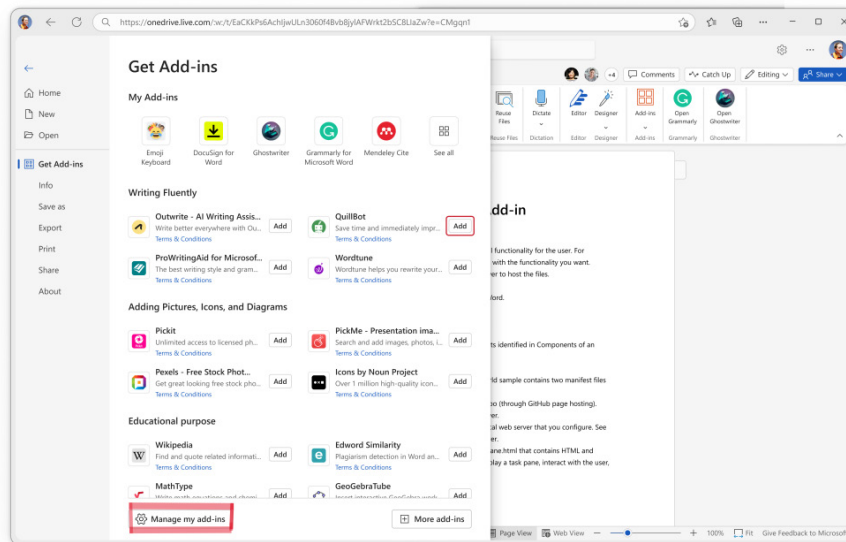


You can manage your installed add-ins in two ways:

1. select **File -> Get add-ins -> Manage my add-ins**
2. select **Home -> Add-ins -> See all**.



You may manage your add-ins by organising them into your preferred order, finding usage data to determine what other add-ins may be useful, and installing add-ins to enhance your device's security and performance.



This feature is available to all Word, Excel, and PowerPoint for web users, and to Excel, PowerPoint and Word for Windows users running Version 2301 (Build 16026.20146) or later.

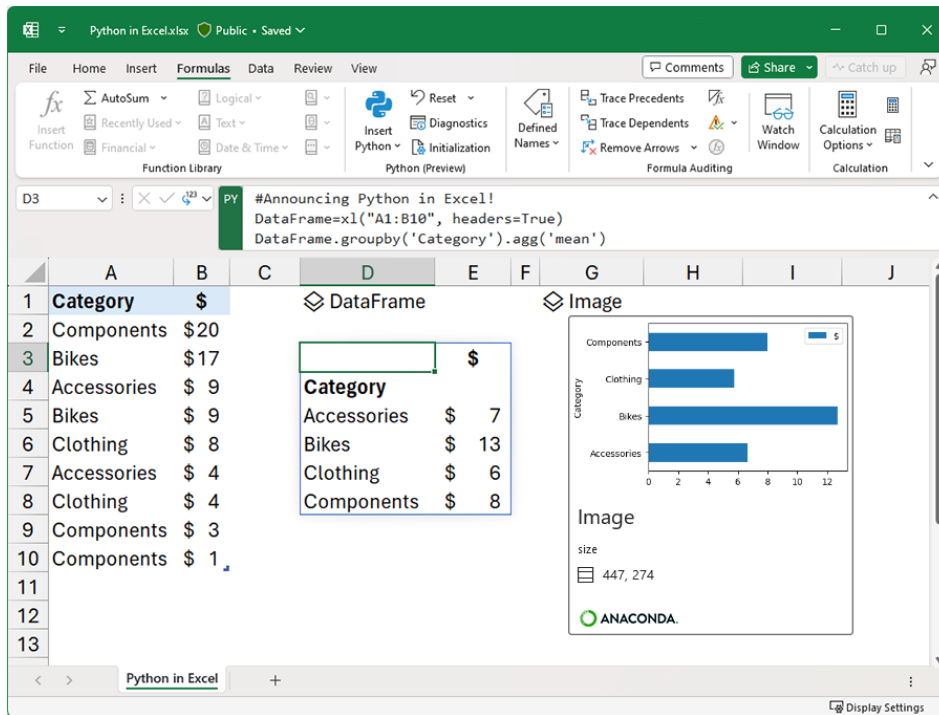
Python in Excel (Insiders Beta)



Yes, it's still a cobra...

This month's update for Excel for Windows (Insiders Beta) sees "a significant evolution" in the analytical capabilities available within Excel by releasing a Public Preview of Python in Excel. Python in Excel makes it possible to natively combine the programming language Python and

Excel analytics within the same workbook - with no setup required. With Python in Excel, you can type Python directly into a cell, the Python calculations run in the Microsoft Cloud, and your results are returned to the worksheet, including plots and visualisations.



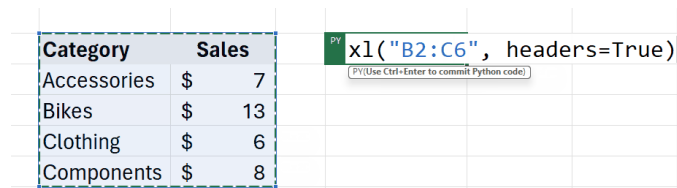
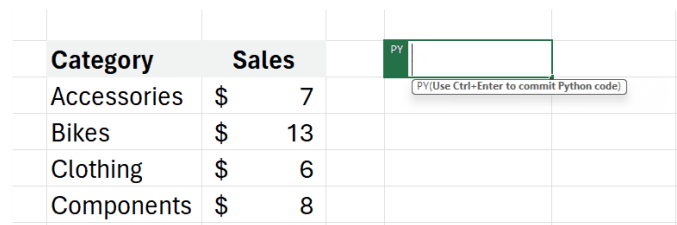
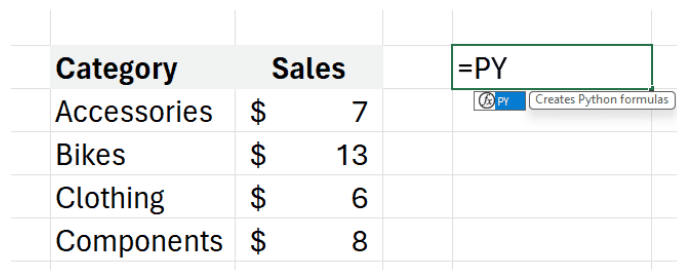
Given there are several languages out there that analysts use (e.g. R, SQL to name but two), it is not unreasonable to consider why is Python the first cab off the rank? There are several reasons mentioned by the Microsoft team. These are detailed below.

Built for analysts

Every day, millions of users around the world rely on familiar Excel tools such as formulas, charts and PivotTables to analyse and understand their data. As of this update, Python in Excel will also be natively integrated directly into the Excel grid. To get started simply use the new **PY** function

which allows you to input Python code directly into Excel cells.

For example, to create DataFrames with just a few simple clicks, it's no more difficult than this:



Category	Sales
Accessories	\$ 7
Bikes	\$ 13
Clothing	\$ 6
Components	\$ 8

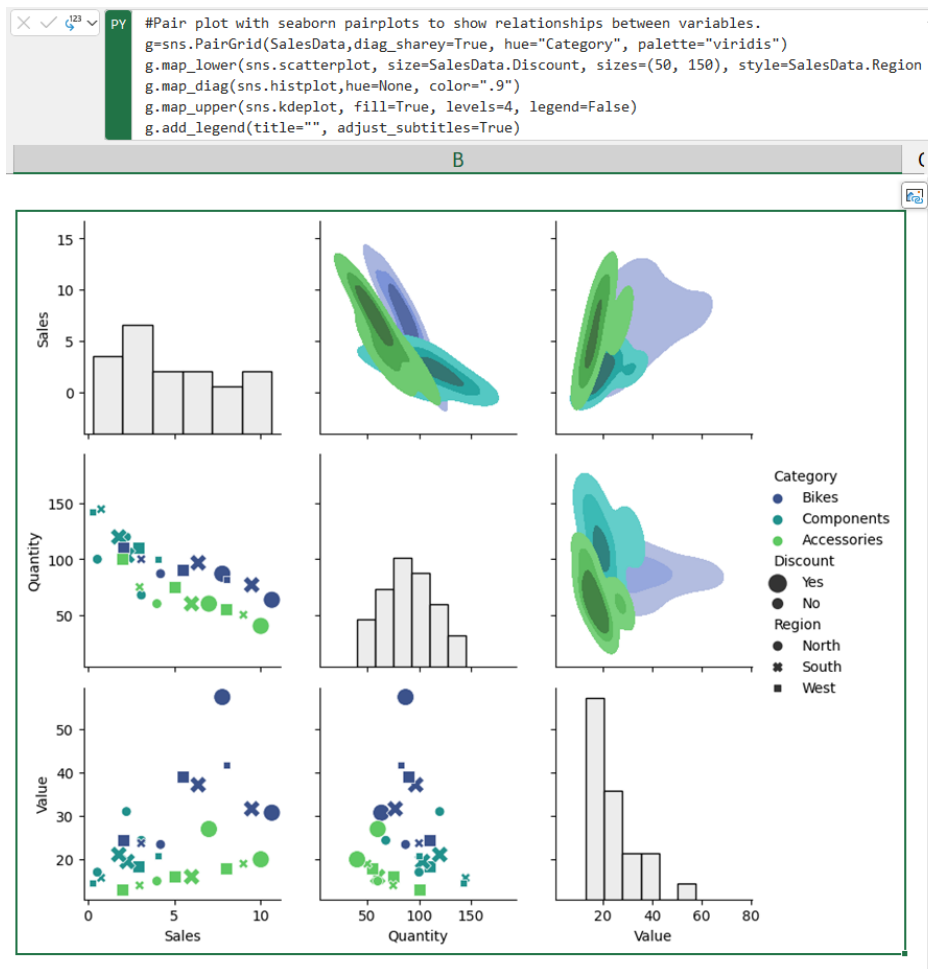
DataFrame

Category	Sales
Accessories	6.65
Bikes	12.7
Clothing	5.75
Components	8

ANACONDA

Excel users now have access to powerful analytics via Python for visualisations, cleaning data, machine learning, predictive analytics, and more. Users may now create end-to-end solutions that seamlessly combine Excel and Python all within Excel. Using Excel's built-in connectors and Power Query, users can easily bring external data into Python in Excel workflows. Python in Excel is compatible with the tools users know well, such as formulae, PivotTables and Excel charts.

An example of an advanced visualisation, for instance, might be the following. You can tap into the potential of well-known Python charting libraries like Matplotlib and seaborn to create a wide variety of charts, spanning from conventional bar graphs and line plots to specialised visualisations such as heatmaps, violin plots and swarm plots.



There is more though. You can consider the impact of machine learning, predictive analytics and forecasting. For examples, you may leverage the capabilities of Python libraries like scikit-learn and statsmodels to

apply popular machine learning, predictive analytics, and forecasting techniques such as regression analysis, time series modelling, etc. You might even throw in a **LAMBDA** along the way!

```

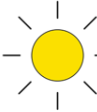
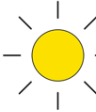


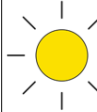
#create and fit the model
model = DecisionTreeClassifier()
model.fit(historical_data, weather_labels)

```

5 Day Weather Forecast

Generates a forecast (sunny, rainy, cloudy), based on historical data for temperature, humidity and wind speed.

- Historical Data DataFrame
- Weather Labels DataFrame
- Model DecisionTreeClassifier
- Full Forecast DataFrame

					
Weekday	Thursday	Friday	Saturday	Sunday	Monday
Weather Forecast	sunny	sunny	cloudy	rainy	sunny
Temperature	67°	65°	77°	60	74°
Humidity	54%	61%	77%	75	63%
Wind Speed	9 MPH	8 MPH	2 MPH	9	8 MPH

You can consider data cleansing too. You may make efficient use of advanced data cleaning techniques such as locating missing values, standardising formats, removing duplicates and employing techniques like regular expressions for pattern-based transformations.

B7 `extract_and_format_dates(x1("B5"))`

Extract and Format Dates

Extracts dates from text strings using regular expressions and then applies a consistent format

Formatting Function function

Data ndarray

7	Extracted Dates	#BUSY!
---	-----------------	--------

Dates found: #BUSY!

C23 `=IFERROR(COUNTA(CHOOSECOLS(B7#,2))-COUNTIF(B7#,""),-1,0)`

Extract and Format Dates

Extracts dates from text strings using regular expressions and then applies a consistent format

Formatting Function function

Data ndarray

7	Extracted Dates	Original Text	Formatted Date
8		Today's date is 23/08/2023	2023-08-23
9		I remember the date 12-05-19.	2019-05-12
10		Another date format: 22-07-15.	2015-07-22
11		Dates in different formats: 15.5.21	2021-05-15
12		A date in the middle: 07.04.1995	1995-04-07
13		Random text without dates.	
14		Date format test: 14-08-23	2023-08-14
15		New date: 25/10/2022	2022-10-25
16		No dates here.	
17		Strange format: 01-20000-01	
18		Yet another date: 31-12-2023	2023-12-31
19		Different format: 31.12.2023	2023-12-31
20		Mixed formats: 15/01/25	2025-01-15
21		An unusual date: 21.01.20001	
22		Trying DD.MM.YYYY: 15.06.2022	2022-06-15

Dates found: 11



Anaconda is a leading enterprise Python repository used by tens of millions of data practitioners worldwide. Python in Excel leverages Anaconda Distribution for Python running in Azure, which includes the

most popular Python libraries (e.g. pandas, Matplotlib, scikit-learn), and is securely built, tested and supported by Anaconda. Python provided by Anaconda supports a wide array of analytics with Python in Excel.

Runs securely on the Microsoft Cloud, with no setup required

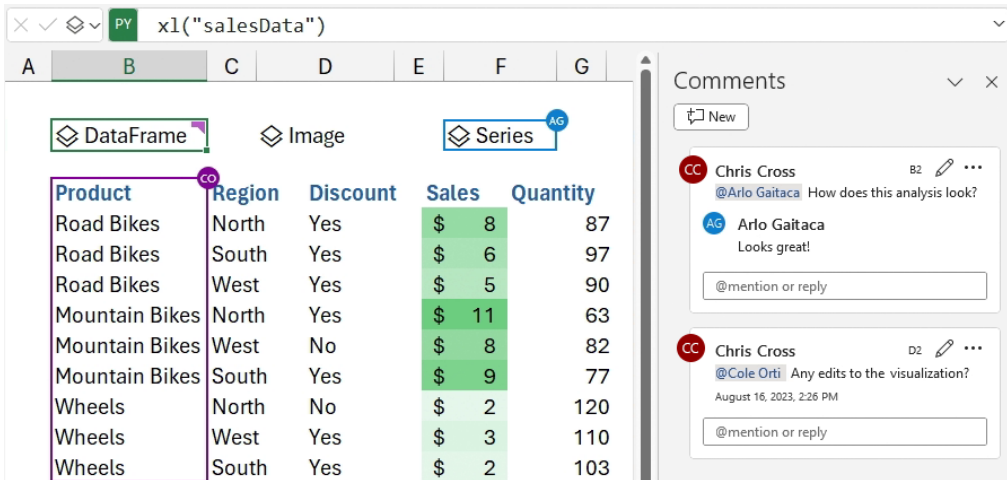
Python code used by Excel runs on the Microsoft Cloud with enterprise-level security as a compliant Microsoft 365 connected experience. The Python code runs in its own hypervisor isolated container using Azure Container Instances and secure, source-built packages from Anaconda through a secure software supply chain. Python in Excel keeps your data

private by preventing the Python code from knowing who you are, and opening workbooks from the internet in further isolation within their own separate containers. Data from your workbooks can only be sent via the built-in **xl()** Python function, and the output of the Python code can only be returned as the result of the **=PY()** Excel function.

Built for teams

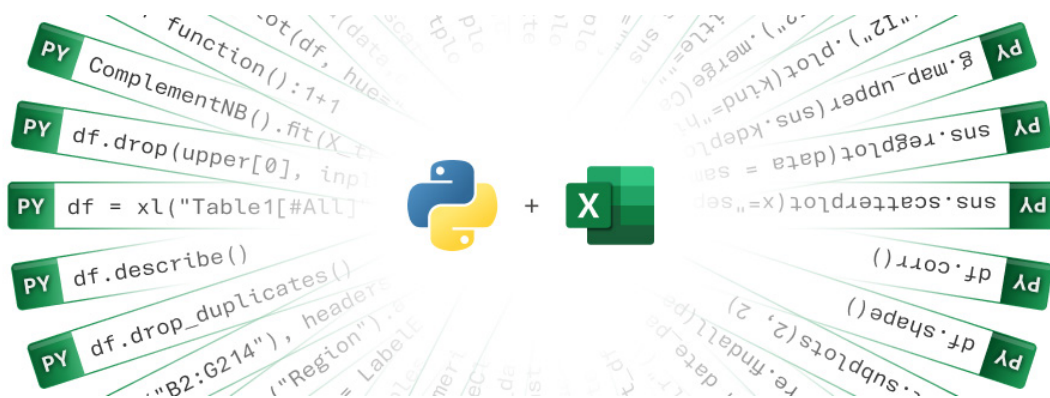
Users can share Python in Excel workbooks with confidence. Teammates can seamlessly interact with and refresh Python in Excel based analytics without needing to worry about installing additional tools, Python runtimes or managing libraries and dependencies. Users may share workbooks using collaboration tools such as Microsoft Teams and

Microsoft Outlook and work together seamlessly via comments, @ mentions, and co-authoring with colleagues as they normally would in Excel. Sensitivity labels applied to your workbooks containing Python will keep them compliant with your organisation's information protection policies too.



Python in Excel is currently available to users running Beta Channel on Windows. This feature will roll out to Excel for Windows first, starting with build 16.0.16818.20000, and then to the other platforms at a later

date. To use Python in Excel, join the Microsoft 365 Insider Program. Then, choose the Beta Channel Insider level to get the latest builds of the Excel application.



Stale Value Formatting (Insiders Beta)

In Excel for Windows (Insiders Beta), there is now a significant improvement to one of spreadsheeting's oldest features: Manual calculation mode.

Manual calculation mode suspends calculations, which means that the values in cells can become out of date. Microsoft refers to these as being **stale**. When you edit a cell in a spreadsheet, not all cells become

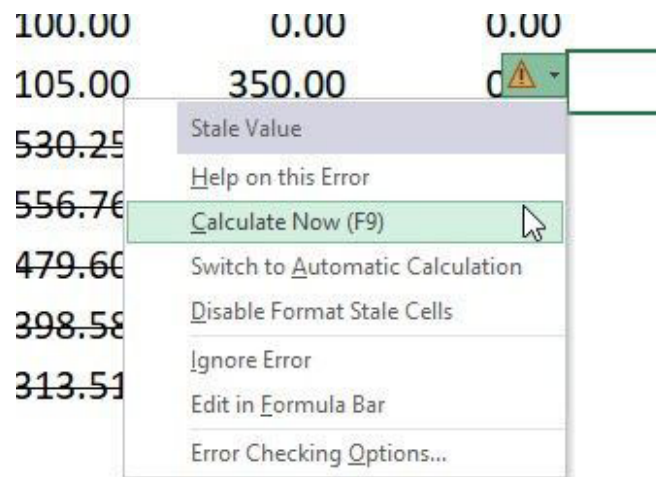
stale, just the ones that depend on your change. Excel tracks stale cells internally. Unfortunately, there has been no way to see which cells were pending calculation.

However, with this update, Excel can show these stale cells by **striking them out**. That way, you know not to rely on them until they are calculated.

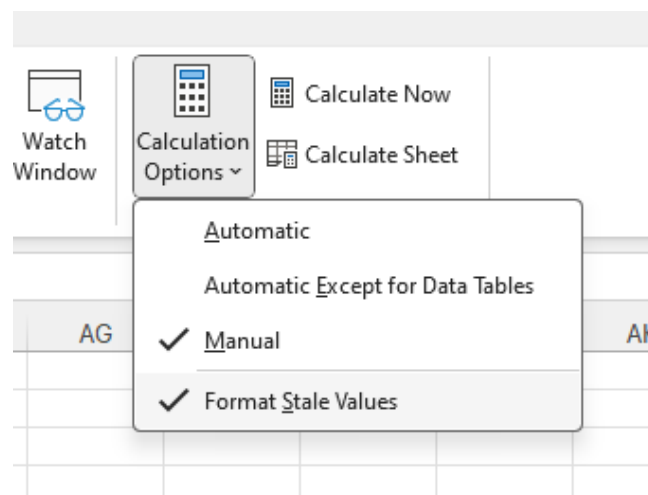
	A	B	C	D
1				
2		Updated:		10
3		Stale:		20
4				

Whilst stale formatting will most often be seen in Manual calculation mode, it is not limited to it. For example, you may get stale cells in Automatic calculation mode if you abort a long-running calculation by pressing **ESC**. Once the calculation resumes and completes, the stale formatting will be lifted.

When a cell containing a stale value is selected, Excel will show a warning icon. If you click on the icon, you will be provided with a menu with contextual actions, such as triggering calculation or switching to Automatic calculation mode.



'Stale Value Formatting' is an application setting that is on by default. If you prefer, you can turn it off via the 'Format Stale Values' checkbox located under the 'Calculation Options' menu on the Formulas tab.

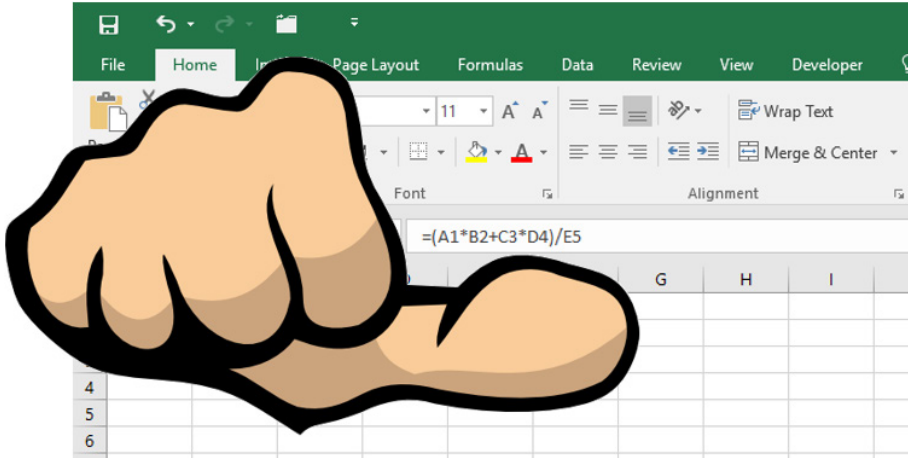


Stale Formatting is currently available to users running Beta Channel on Windows. This feature will roll out to Excel for Windows first and come to the other platforms at a later date.

Monospaced Font in Formula bar (Insiders Beta)

If you're like us, you probably have a lot of your working life with your head immersed in formulae. These pesky little critters are the backbone of any spreadsheet, and they allow you to perform calculations, analyse data, build financial models, forecast key variables and create dynamic

reports. However, no matter how experienced you (think you) are, writing and editing formulae can sometimes be challenging, especially if you have complex or nested expressions – which you shouldn't because you should follow the "Rule of Thumb", viz.



No formula should be longer than your thumb!

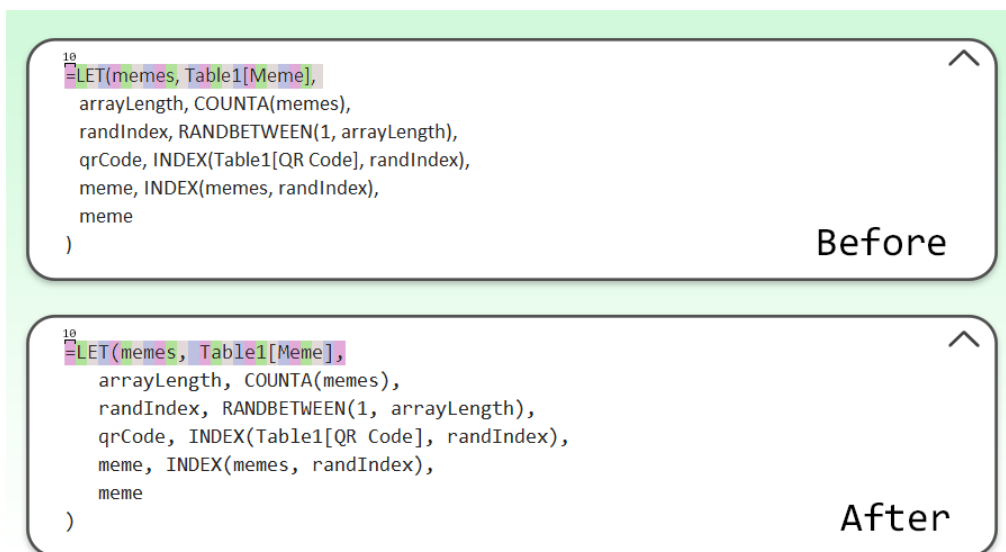
To this end, Microsoft has announced a change that should make your formula editing experience easier and more straightforward: the use of the Monospaced font in the Formula bar. The Monospaced font is a type of font where each character has the same width, regardless of

its shape or style. This means that your formulae will align and space evenly, making them easier to read and debug.

Monospaced font, also known as fixed-width or typewriter font, has several advantages over a proportional font. Here are some of the benefits of using monospaced font in the Formula bar:

- **Alignment and Readability:** Monospaced fonts significantly improve the readability of code by ensuring consistent and clear alignment. Each character occupies the same amount of space horizontally, which makes it easier to maintain an organised, structured and consistent layout. This is particularly helpful for indentation and presenting a clean image
- **Error Detection and Visual Parsing:** Monospaced fonts offer consistent alignment, making it easier to quickly spot errors, typos and discrepancies in code. This uniform spacing enhances visual parsing of complex structures, aiding in identifying patterns, nested levels and relationships within the logic. This is particularly valuable for understanding intricate nested or conditional statements and catching mistakes early in the authoring process
- **Cross-Platform Consistency:** Monospaced fonts offer streamlined copy-pasting of code from different sources while maintaining formatting. They also ensure a uniform appearance across various platforms and editors, ensuring consistent readability and structure.

To use the Monospaced font in the Formula bar, simply edit or author a formula in the Formula bar. You should notice a different rendering of your formula text. Here is an example of a before and after of some formula text:



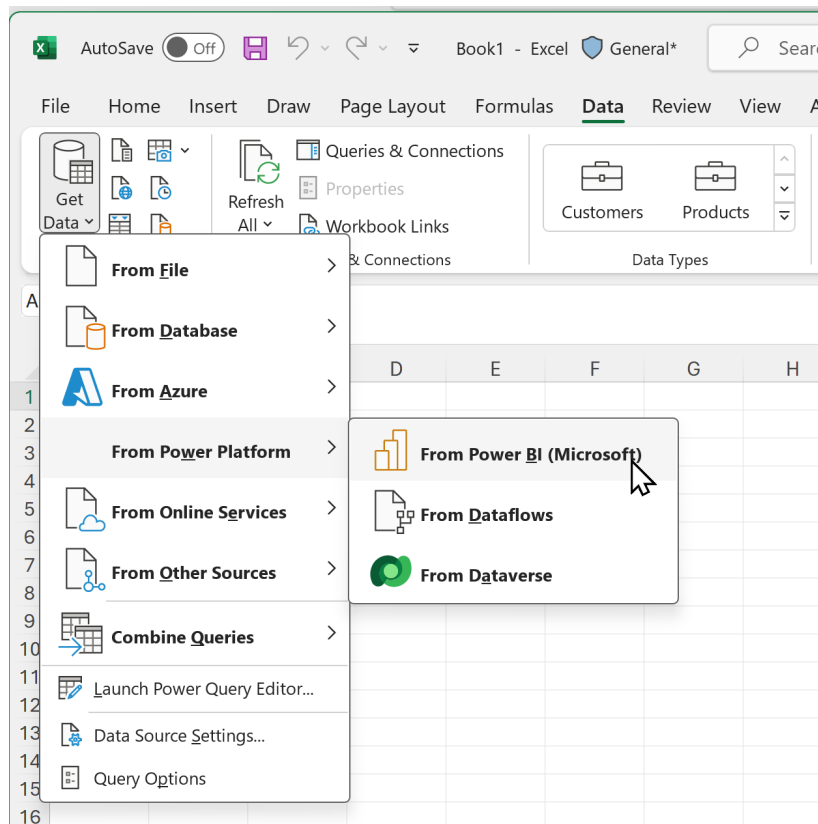
You can see how by the time we get to the first "m" in the formula, the letters are not consistently centred in 10px width boxes in the "Before" image.

Power BI Connected Tables

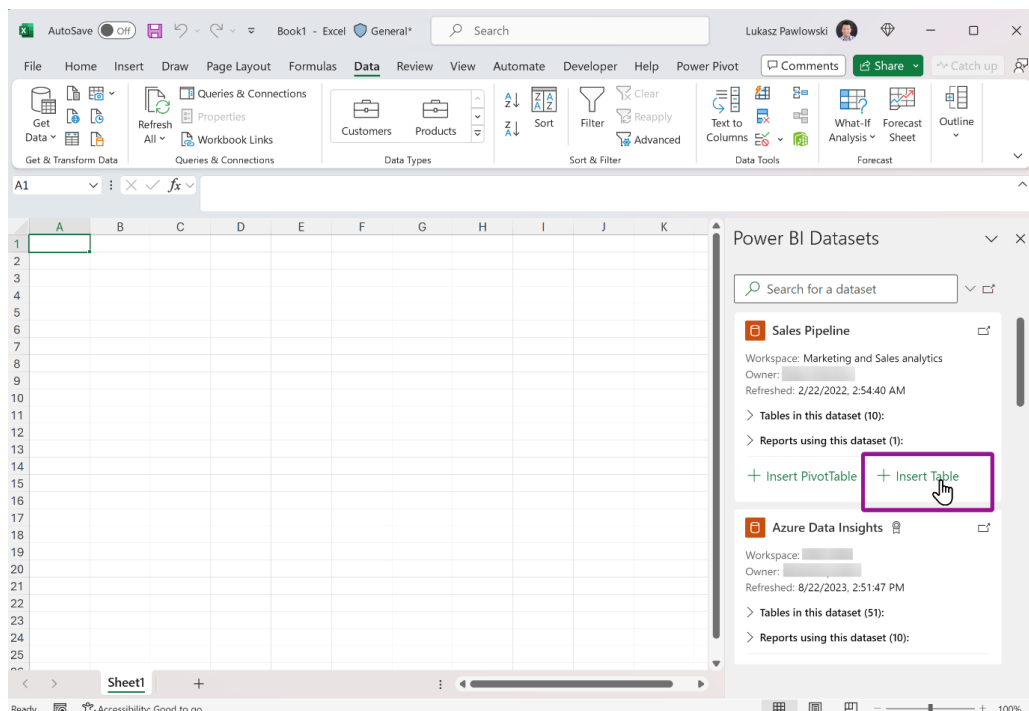
Also in Excel for Windows, Microsoft has announced new capabilities that make it easier for users to add Power BI data to their Excel workbooks. Earlier this year, Microsoft introduced 'Export with Live Connection' for interactive reports. This made it easy for users to create Connected Tables in Excel sheets. Now, it has been updated, with the new capabilities rolling out over the next few weeks.

The new 'Insert Table' option is rolling out for the 'Power BI Datasets' pane in Excel. It enables users to create a connected table directly in Excel. This streamlines the workflow of adding data and is friendlier to more Excel users.

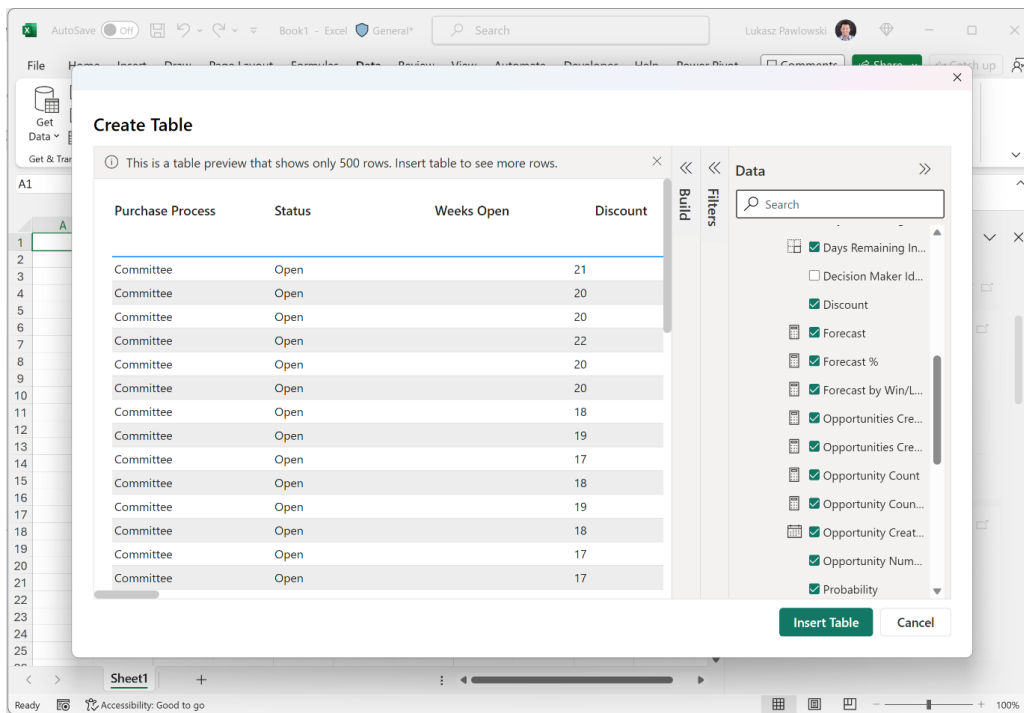
You may use the **Data Ribbon -> From Power Platform -> From Power BI (Microsoft)** option to launch the 'Power BI Datasets' side pane in Excel.



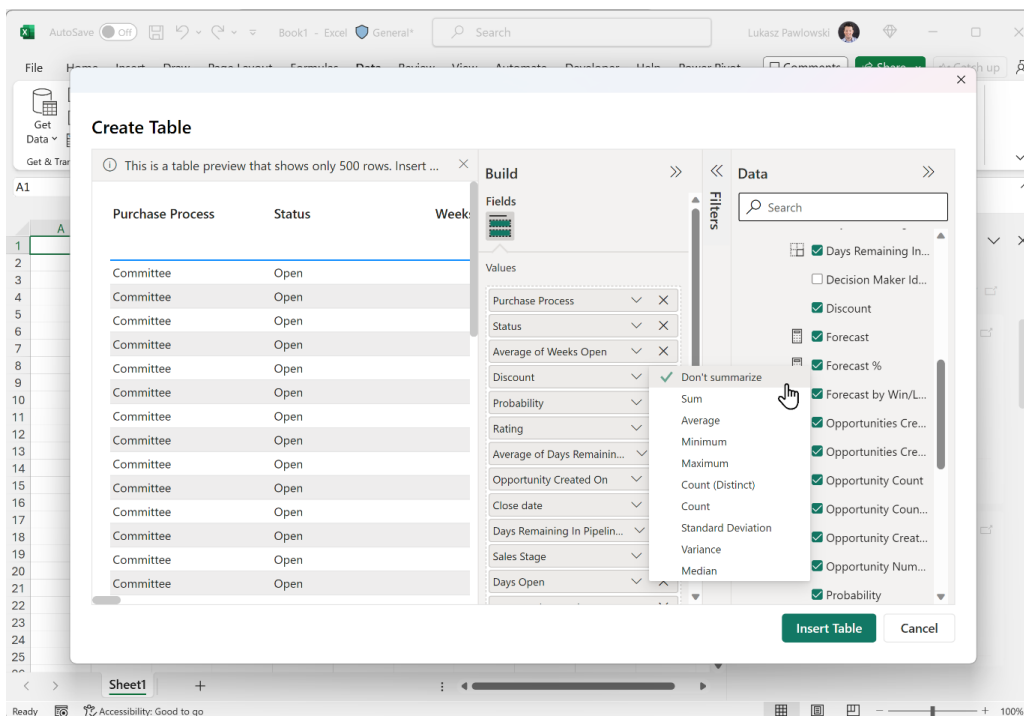
For each dataset, you'll see the new 'Insert Table' option. This option is initially available in Excel for Windows.



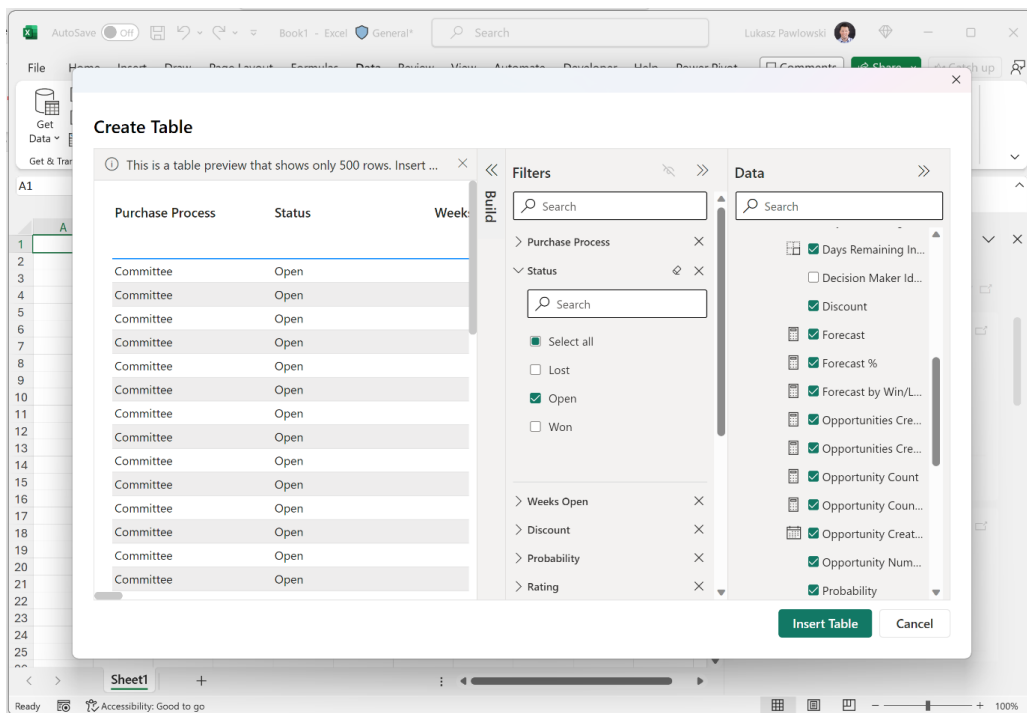
The new 'Create Table' dialog will open. You can use the Data pane to select the fields you want to add to your Excel workbook.



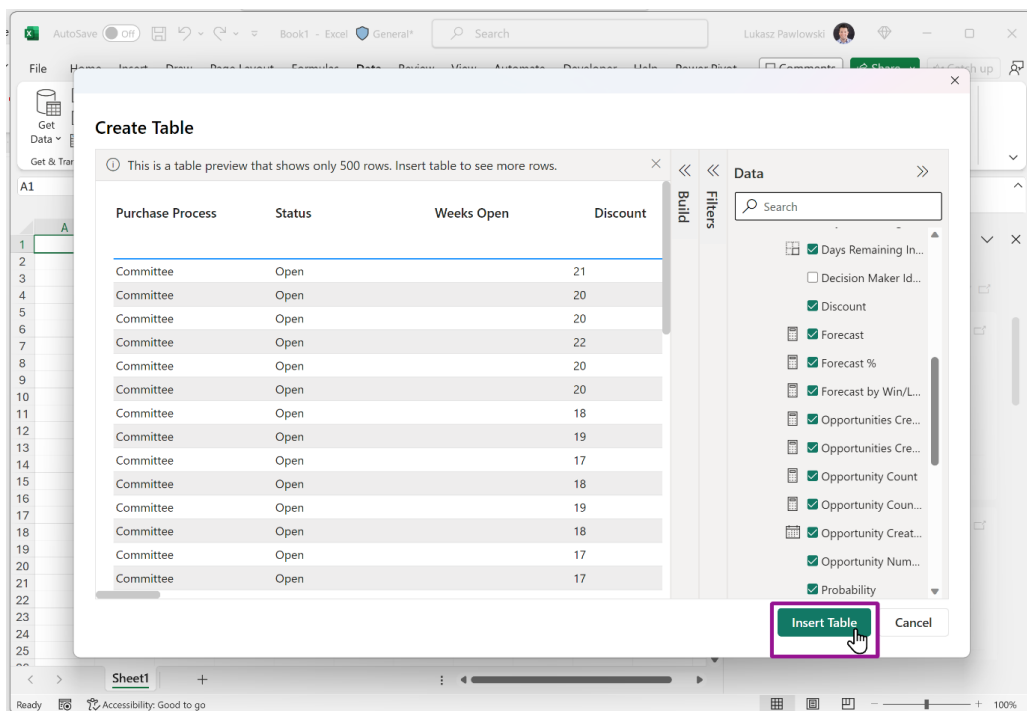
You may use the 'Build' pane to reorder, remove fields and change the aggregation for fields.



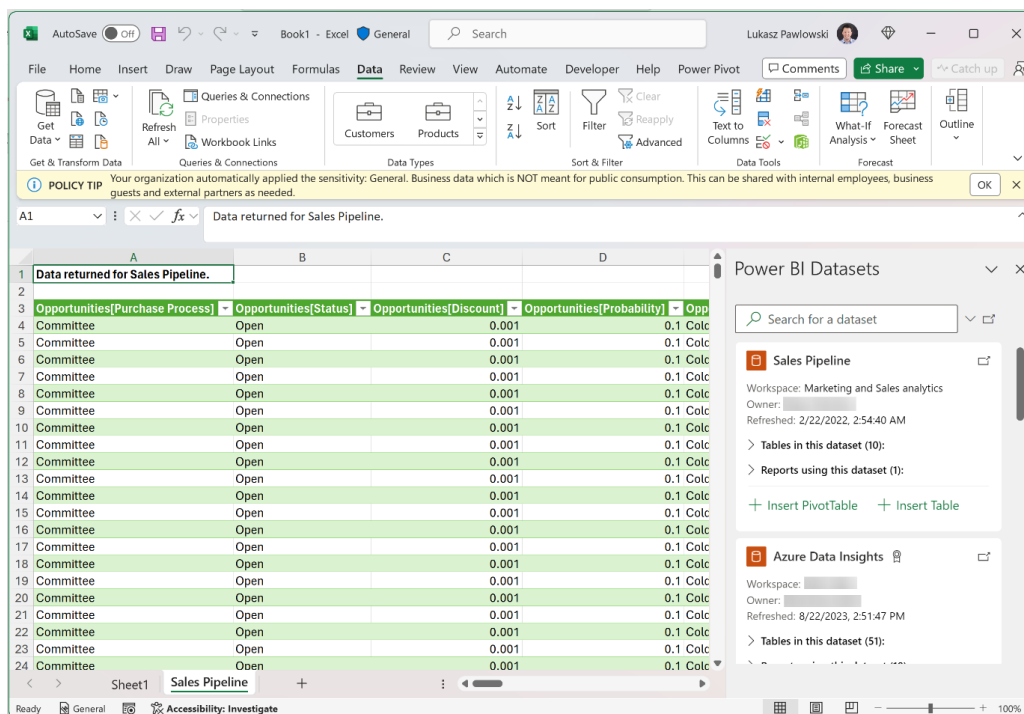
You may use the 'Filters' pane to do basic filtering for rows in your table.



When you're ready, press 'Insert Table' to add a Connected Table to your Excel sheet.



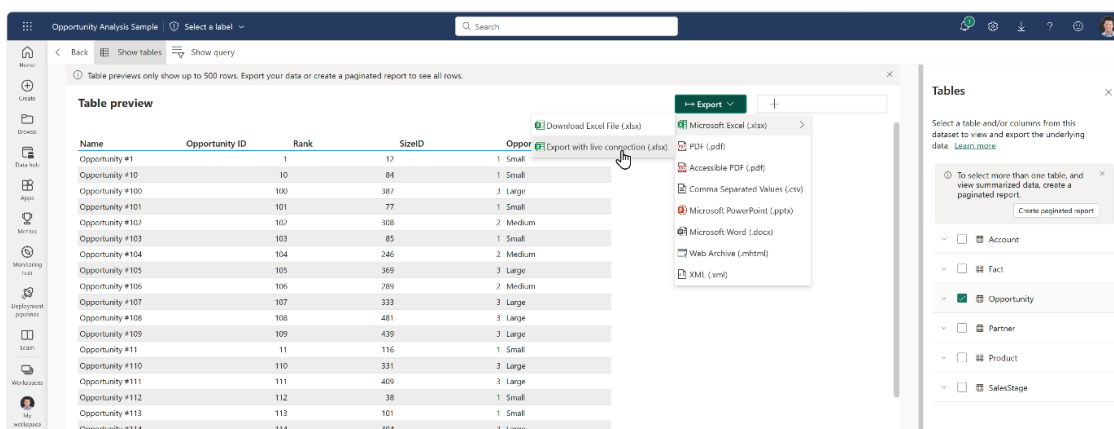
At this point, a Connected Table is inserted into your worksheet. You may refresh the data to get the latest data. Any data security like Row Level Security, Permissions and Sensitivity Labels will be enforced as well.



Just like with other Connected Table experiences, the workbook contains a Connection that you can review and modify through **Data Ribbon -> Queries and Connections -> Queries and Connections pane -> Connections Tab -> right-click -> Properties**. You may even change the query if you'd like using the 'Command Text' on the 'Definition' tab.

Microsoft is also starting to roll out a new 'Export with live connection (.xlsx)' option for the OneLake Data Hub experiences. When you navigate to a dataset in Power BI, you can use the Tables side pane to pick a table.

This will be added to the Export menu. This will create a connected Excel workbook that shows a Connected Table as above, or with the 'Export to Excel' capability when viewing an interactive report.



Whilst this is great for creating new queries, especially for those who don't know how to write **DAX**, there are several limitations to be aware of:

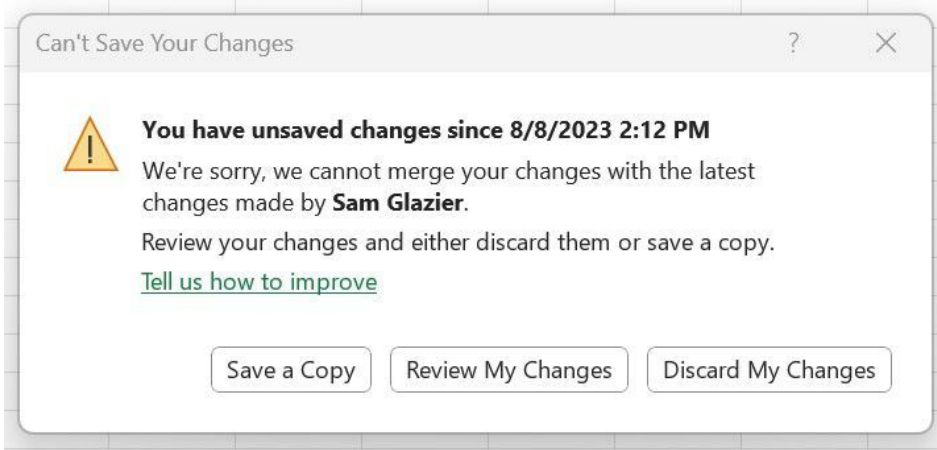
- the 'Insert Table' option is rolling out to Excel for Windows users incrementally. It won't be available immediately in Excel for the web
- the 'Export with Live connection (.xlsx)' option is rolling out for Power BI users in the Data Hub

- the column names in Excel can't be changed. If you refresh, the query will reset the column names to the original ones
- you can't edit the query in the user interface (UI). If you'd like to make changes, it's best to just create a new query with the settings you want.

As stated above, if you don't see these capabilities immediately, just give it some time since the roll-out will take several weeks to complete fully.

Improved Save As / Discard Dialog

Finally for this month's Excel for Windows updates, the 'Save As / Discard' dialog now includes additional information such as when you last saved your changes and who else has made changes:



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

Excel Features Availability

Page 1 of 4

Feature	Insider		Production				Web
	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	
Block XLI Add-Ins	Version 2302 (Build 16130.20128) or later						
PivotTables: Manual Sort of Rows & Columns			Already Supported	Already Supported	Already Supported	Already Supported	February 2023
Automatic Recalculation Optimization	Version 2208 (Build 15529.10000) or later	Version 16.64 (Build 22081401) or later					
Import Data from SQL Server Database		Version 16.68 (Build 22110801) or later					
Import Data from Additional Sources						Version 16.69 (Build 23010700) or later	
Power Query Editor						Version 16.69 (Build 23010700) or later	
IMAGE Function			Version 2211 (Build 15831.20190) or later	Version 2211 (Build 15831.20252) or later		Version 16.67 (Build 22102900) or later	December 2022
Check Formulas with Value Preview Tooltips	Version 2302 (Build 16116.20000) or later	Version 16.70 (Build 230116) or later					
Office Scripts	Version 2212 (Build 15922.20000) or later	Version 16.68 (Build 22120101) or later					
Automate Tasks with Power Automate tab			Version 2301 (Build 15793.10000) or later			Version 16.66 (Build 22092500) or later	
PivotTable Show Details to XLO							January 2023
Excel Live in Teams							December 2022
Formula Suggestions							December 2022*
Formula by Example							December 2022*
Suggested Links							December 2022
Add search bar in queries pane							December 2022

*Starting to roll out

Features Flyer: aka.ms/ExcelFeaturesFlyer

CC: Current Channel; MEC: Monthly Enterprise Channel; SA: Semi-Annual Enterprise Channel. All information is subject to change.

Excel Features Availability

Page 2 of 4

Feature	Insider		Production				Web
	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	
Add keyboard shortcut to open PQ editor			Version 2211 (Build 15730.31883) or later				
Create nested PQ data types	Version 2211 (Build 15928.10000) or later						
Add Get Data from Dynamic Arrays	Version 2105 (Build 14014.20002) or later						
Data from picture			Version 2210 (Build 15723) or later	Version 2210 (Build 15726.20262) or later		Version 16.38 or later	December 2022
Chart Data Foils							November 2022
Show Changes			Version 2209 (Build 15703.10000) or later			Version 16.66 (Build 22092500) or later	Already Supported
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
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)	Version 2208 (Build 15601.20456) or later	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			Version 2210 (Build 15726.20262) 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					

Features Flyer: aka.ms/ExcelFeaturesFlyer

CC: Current Channel; MEC: Monthly Enterprise Channel; SA: Semi-Annual Enterprise Channel. All information is subject to change.

Excel Features Availability

Page 3 of 4

Feature	Insider		Production				Web
	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	
New Excel functions			Version 2208 (Build 15427.20194) or later	Version 2208 (Build 15601.20230) 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
Add and edit rich text formatting			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Sort by color or icon from auto filter menu			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Edit files with legacy data connections			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Edit files with legacy Shared Workbook feature			Already Supported	Already Supported	Already Supported	Already Supported	August 2022
Delete chart elements							August 2022
Multiline formula bar							August 2022
Search within PivotTable Field List			Already Supported	Already Supported	Already Supported	Already Supported	July 2022
Get automatic data conversions	Version 2207 (Build 15427.20000) or later						
Natural Language Query Improvements			Version 2206 (Build 15330.20230) or later	Version 2205 (Build 15225.20354) or later		Version 16.63 (Build 22070801) or later	
Resize Conditional Formatting dialog box		Version 16.64 (Build 22070600) or later					
Sheet protection			Already Supported	Already Supported	Already Supported	Already Supported	June 2022

Features Flyer: aka.ms/ExcelFeaturesFlyer

CC Current Channel; MEC Monthly Enterprise Channel; SA Semi-Annual Enterprise Channel
All information is subject to change.

Excel Features Availability

Page 4 of 4

Feature	Insider		Production				Web
	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	
Semi-select for links creation			Already Supported	Already Supported	Already Supported	Already Supported	June 2022
Add "PivotTable Connections to Slicer settings pane"			Already Supported	Already Supported	Already Supported	Already Supported	June 2022
Import from local text, CSV, and XLSX files						Version 16.57 (22011100) or later	
Provide automatic all-text suggestions on charts and PivotCharts			Version 2205 (Build 15225.20288) or later	Version 2204 (Build 15126.20280) or later		Version 16.62 (22061100) or later	
Power Query refresh for selected data sources			Already Supported	Already Supported	Already Supported	Already Supported	May 2022
Changing source file for workbook links			Already Supported	Already Supported	Already Supported	Already Supported	May 2022
Improved Recommended PivotTable experience	Version 2204 (Build 15126.10000) or later						
Faster recalc on resource constrained devices		Version 16.62 (Build 22050804) or later	Version 2204 (Build 15126.20248) or later	Version 2204 (Build 15126.20280) or later			
Faster AutoFilter				Version 2204 (Build 15126.20248) or later	Version 2208 (Build 15601.20456) or later	Version 16.61 (22050700) or later	
Dataflow connector				Version 2203 (Build 15026.20248) or later			
Dataverse connector			Version 2204 (Build 15126.20178) or later				
Improved Find dialog and Find All						Version 16.60 (220410) or later	
LAMBDA: Helper Functions			Version 2202 (Build 14931.20120) or later	Version 2202 (Build 14931.20274) or later	Version 2208 (Build 15601.20456) or later	Version 16.56 (Build 211211) or later	Already Supported

Features Flyer: aka.ms/ExcelFeaturesFlyer

CC Current Channel; MEC Monthly Enterprise Channel; SA Semi-Annual Enterprise Channel
All information is subject to change.

More next month, we're sure.

The A to Z of Excel Functions: MAKEARRAY

What is an array?

Dimensions	Example	Terminology									
1	<table border="1"><tr><td>0</td><td>1</td><td>2</td></tr></table>	0	1	2	Vector						
0	1	2									
2	<table border="1"><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td></tr></table>	0	1	2	3	4	5	6	7	8	Matrix
0	1	2									
3	4	5									
6	7	8									
3	<table border="1"><tr><td>0</td><td>1</td><td>2</td></tr><tr><td>3</td><td>4</td><td>5</td></tr><tr><td>6</td><td>7</td><td>8</td></tr></table>	0	1	2	3	4	5	6	7	8	3D Array (3 rd order Tensor)
0	1	2									
3	4	5									
6	7	8									
N	<table border="1"><tr><td>...</td></tr><tr><td>...</td></tr></table>	ND Array							
...											
...											

MAKEARRAY returns a calculated array of a specified row and column size, by applying a **LAMBDA** function. This function is useful for situations where you wish to combine or transform arrays, as well as being useful for generating data. The syntax is as follows:

MAKEARRAY(rows, columns, lambda)

It has the following arguments:

- **rows**: this argument is required and represents the number of rows in the array (which must be greater than zero)
- **columns**: this argument is also required and represents the number of columns in the array (which again must be greater than zero)
- **lambda**: also necessary, this is the **LAMBDA** that is called to create the array. In particular, this lambda function must take two parameters, namely:
 - **row_index**: the index of the row (row number)
 - **column_index**: the index of the column (column number).

As an example, consider the following:

	A	B	C	D	E	F	G
1							
2		Number of rows		5			
3		Number of columns		3			
4							
5		Colours of Rainbow					
6		Red			Red	Orange	Violent
7		Orange			Orange	Red	Blue
8		Yellow			Red	Violent	Indigo
9		Green			Blue	Indigo	Violent
10		Blue			Green	Blue	Red
11		Indigo					
12		Violent					

Imagine, for reasons best known to myself, we want to generate an array of colours of the rainbow (albeit with the final colour, ahem, slightly amended). In the image above, I have specified the number of rows (cell **D2**) and the number of columns (cell **D3**) in my array, and listed the colours in cells **B6:B12** inclusive.

The formula in cell **E6** is given by

=MAKEARRAY(D2, D3, LAMBDA(row, column, INDEX(B6:B12, RANDBETWEEN(1, 7))))

The first two arguments in this formula are **D2** and **D3**, which refer to the number of rows and columns for the array to be generated respectively. The final argument of **MAKEARRAY** is the **LAMBDA**, which must take two parameters, corresponding to the value generated by **LAMBDA**, namely:

- **row**: the index of the row
- **column**: the index of the column.

The calculation thus uses the non-dynamic array function **RANDBETWEEN** to generate an integer between one [1] and seven [7] to select from the list of colours of the rainbow, stipulated in cells **B6:B12**. For example, if Excel generates the number 5, the value “Blue” will be chosen, etc.

Now it is true that existing functions could be used to achieve the same result, e.g.

=INDEX(B6:B12, RANDARRAY(D2, D3, 1, 7, TRUE))

This formula seems shorter and simpler, and indeed, may be the better option for this above illustration. But that is exactly what this is – a simple example. As more complex arrays need to be created, existing function counterparts may prove difficult, convoluted or impossible to construct – and this is precisely where **MAKEARRAY** and **LAMBDA** come in.

The A to Z of Excel Functions: MAP

As one of the more recent Excel functions, **MAP** takes the honour of becoming **Excel's 500th function** – as agreed by fellow Excel Most Valuable Professional (MVP) Bill Jelen and ourselves.



You should be aware that the **MAP** function, does not actually return a map!



Instead, it returns an array formed by mapping each value in the array(s) to a new value and applying a **LAMBDA** to create a new value accordingly. It has the following syntax:

MAP(array1, lambda or array2, [lambda or array3, ...])

where:

- **array1**: this is a required argument and represents the (first) array to be mapped
- **array2 and subsequent arrays**: these are optional arguments and represent additional arrays to be mapped
- **lambda**: this is a required argument which represents a **LAMBDA** which must be the final argument and must have a parameter for each array passed or another array to be mapped.

In short, **MAP** transforms values.

Let's consider my Melbourne average temperatures data (in degrees Celsius):

Month	Yr 2011	Yr 2012	Yr 2013	Yr 2014	Yr 2015	Yr 2016	Yr 2017	Yr 2018	Yr 2019	Yr 2020
January	25.0	17.5	18.0	21.0	18.4	14.6	24.6	20.6	16.2	25.3
February	14.7	18.4	23.3	20.7	17.3	24.2	24.5	23.8	17.0	14.7
March	21.2	15.4	14.4	19.7	15.9	21.9	14.4	18.2	14.2	22.1
April	13.1	13.0	19.8	13.9	17.2	18.5	12.2	11.4	16.9	11.4
May	11.6	11.5	13.6	15.6	10.0	10.5	14.8	15.3	14.0	12.4
June	7.6	10.4	6.9	10.0	9.3	13.3	13.5	11.1	7.8	10.7
July	9.0	11.2	6.8	12.5	12.0	12.5	12.1	9.4	11.1	6.7
August	8.2	7.5	11.5	11.8	10.7	9.3	10.7	14.4	6.7	13.2
September	14.2	10.3	14.7	17.3	9.4	14.9	8.1	10.2	15.8	14.4
October	15.7	15.4	14.0	13.9	16.9	13.3	18.2	11.2	17.8	18.8
November	22.0	11.6	21.6	21.4	16.9	11.5	18.6	11.2	11.8	15.0
December	13.2	20.0	20.1	18.7	14.0	20.5	13.8	15.4	20.2	18.8

If I want the year-on-year comparisons for each month where the average temperature is above 15 degrees Celsius, I can avoid using a "helper column" and instead use the formula

=FILTER(Temps, BYROW(Temps, LAMBDA(year, AVERAGE(year) > 15)))

Here, the **LAMBDA** function returns values where the average temperature for the year is above 15 degrees Celsius, using the **BYROW** function, viz.

January	25.0	17.5	18.0	21.0	18.4	14.6	24.6	20.6	16.2	25.3
February	14.7	18.4	23.3	20.7	17.3	24.2	24.5	23.8	17.0	14.7
March	21.2	15.4	14.4	19.7	15.9	21.9	14.4	18.2	14.2	22.1
October	15.7	15.4	14.0	13.9	16.9	13.3	18.2	11.2	17.8	18.8
November	22.0	11.6	21.6	21.4	16.9	11.5	18.6	11.2	11.8	15.0
December	13.2	20.0	20.1	18.7	14.0	20.5	13.8	15.4	20.2	18.8

The problem is, these temperatures have all been provided in Celsius, which those of you in the United States (amongst other countries) may not understand. If a US reader sees a temperature of 25 degrees, they will be breaking out the gloves, bobble hat and duffle coat, whereas us Aussies will be heading for the beach.

We need to convert – **transform** – this data to Fahrenheit, so our US colleagues may better understand. All I need to do is wrap the above formula in a **MAP** function:

**=MAP(FILTER(Temps, BYROW(Temps, LAMBDA(year, AVERAGE(year) > 15))),
LAMBDA(temperature, IF(ISNUMBER(temperature),
CONVERT(temperature, "C", "F"), temperature)))**

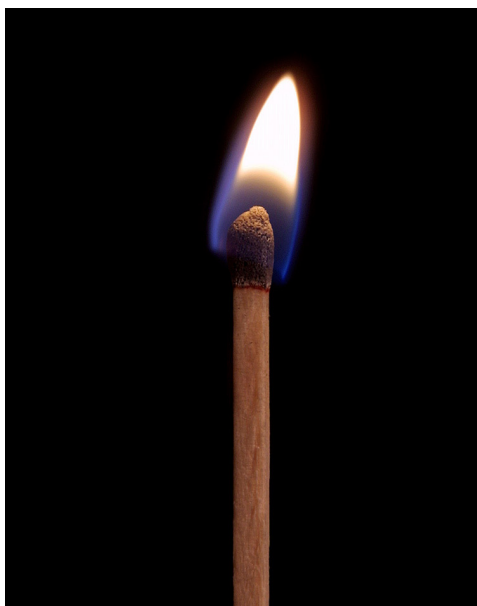
January	77.0	63.5	64.4	69.8	65.1	58.3	76.3	69.1	61.2	77.5
February	58.5	65.1	73.9	69.3	63.1	75.6	76.1	74.8	62.6	58.5
March	70.2	59.7	57.9	67.5	60.6	71.4	57.9	64.8	57.6	71.8
October	60.3	59.7	57.2	57.0	62.4	55.9	64.8	52.2	64.0	65.8
November	71.6	52.9	70.9	70.5	62.4	52.7	65.5	52.2	53.2	59.0
December	55.8	68.0	68.2	65.7	57.2	68.9	56.8	59.7	68.4	65.8

CONVERT(temperature, "C", "F") simply converts the variable **temperature** from degrees Celsius to degrees Fahrenheit. This is wrapped in an **IF(ISNUMBER())** check to ensure that we don't try to convert text values (as this would cause an error): the **IF** statement leaves the value of **temperature** "as is" in this instance, and **LAMBDA**

just wraps around all of this in order to declare the variable **temperature** "work", so that **MAP** may do its work.

It's true you could generate this result in stages, but the whole idea of these **LAMBDA** helper functions is to be able to create dynamic arrays in one fell swoop.

The A to Z of Excel Functions: MATCH



MATCH with its best friend **INDEX** – as a combination – are two of the most useful functions at a modeller’s disposal. They provide a versatile lookup in a way that **LOOKUP**, **HLOOKUP** and **VLOOKUP** simply cannot. The best way to illustrate this point is by means of an example.

Here is a common problem. Imagine you have built a financial model and your Balance Sheet – ahem! – contains misbalances. You need to fix it.

Now I am sure you have never had this mistake yourself, but you have “close friends” that have encountered this feast of fun: solving Balance Sheet errors can take a *long* while. One of the first things modellers will do is locate the first period (in ascending order) that has such an error, as identifying the issue in this period may often solve the problem for all periods.

Consider the following example:

The screenshot shows an Excel spreadsheet with the following data:

Month Ending	Jan-20	Feb-20	Mar-20	Apr-20	May-20	Jun-20	Jul-20	Aug-20	Sep-20	Oct-20	Nov-20	Dec-20
Month	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
Net Assets	100.0	110.0	120.0	130.0	137.8	150.0	160.0	170.0	179.9	190.0	200.0	210.0
Total Equity	100.0	110.0	120.0	130.0	140.0	150.0	160.0	170.0	180.0	190.0	200.1	210.0
Balance Check	-	-	-	-	1	-	-	-	1	-	1	-

The spreadsheet also includes an error message: "Error Message: First Error in May-20" and a section for Error Reporting Options with radio buttons for INDEX MATCH (selected), LOOKUP, and HLOOKUP.

This is a common modelling query. The usual suspects, **LOOKUP** and **HLOOKUP** / **VLOOKUP** do not work here:

LOOKUP(lookup_value, lookup_vector, [result_vector]) gives the wrong date as the balance checks are not in strict ascending order (*i.e.* ascending alphanumerically with no duplicates); *whilst*

HLOOKUP(lookup_value, table_array, row_index_number, [range_lookup]) gives **#VALUE!** since the first row must contain the data to be ‘looked up’, but the Balance Check is in row 13 in our example above, whereas the dates we need to return are in row 4 – hence we get a syntax error.

There is a solution, however: **INDEX MATCH**. They form a highly versatile tag team, but are worth introducing individually.

MATCH

MATCH(lookup_value, lookup_vector, [match_type]) returns the relative position of an item in an array that (approximately) matches a specified value. It is not case sensitive.

The third argument, **match_type**, does not have to be entered, but for many situations, I strongly recommend that it is specified. It allows one of three values:

- **match_type 1 [default if omitted]:** finds the largest value less than or equal to the **lookup_value** – but the **lookup_vector** must be in strict ascending order, limiting flexibility
- **match_type 0:** probably the most useful setting, **MATCH** will find the position of the first value that matches **lookup_value** exactly. The **lookup_array** can have data in any order and even allows duplicates
- **match_type -1:** finds the smallest value greater than or equal to the **lookup_value** – but the **lookup_array** must be in strict descending order, again limiting flexibility.

When using **MATCH**, if there is no (approximate) match, **#N/A** is returned (this may also occur if data is not correctly sorted depending upon **match_type**).

MATCH is fairly straightforward to use:

	E	F
12	1	a
13	2	b
14	3	a
15	4	c
16	5	b
17	6	d
18	7	d
19	8	e
20	9	f
21	10	a
22	11	c

In the figure above, **MATCH("d",F12:F22,0)** gives a value of 6, being the relative position of the first 'd' in the range. Note that having **match_type 0** here is important. The data contains duplicates and is not sorted alphanumerically. Consequently, using **match_type 1** and **-1** would give the wrong answer: 7 and **#N/A** respectively.

INDEX

Essentially, **INDEX(array, row_number, [column_number])** returns a value or the reference to a value from within a table or range (list). For example, **INDEX({7,8,9,10,11,12},3)** returns the third item in the list {7,8,9,10,11,12}, i.e. 9. This could have been a range: **INDEX(A1:A10,5)** gives the value in cell **A5**, etc.

INDEX can work in two dimensions as well (hence the **column_number** reference). Consider the following example:

		1	2	3	4	5	6	7
		F	G	H	I	J	K	L
1	11	1	2	3	4	5	6	7
2	12	8	9	10	11	12	13	14
3	13	15	16	17	18	19	20	21
4	14	22	23	24	25	26	27	28
5	15	29	30	31	32	33	34	35
6	16	36	37	38	39	40	41	42
7	17	43	44	45	46	47	48	49
8	18	50	51	52	53	54	55	56
9	19	57	58	59	60	61	62	63
10	20	64	65	66	67	68	69	70
11	21	71	72	73	74	75	76	77

INDEX(F11:L21,4,5) returns the value in the fourth row, fifth column of the table array **F11:L21** (clearly 26 in the above illustration).

INDEX MATCH

Whilst useful functions in their own right, combined they form a highly versatile partnership. Consider the original problem:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
1	Balance Sheet - Looking For Misbalances																						
2	INDEX MATCH Example																						
3																							
4	Month Ending		Jan-20	Feb-20	Mar-20	Apr-20	May-20	Jun-20	Jul-20	Aug-20	Sep-20	Oct-20	Nov-20	Dec-20									
5	Month		M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12									
6	Balance Sheet Summary																						
7																							
8	Net Assets		100.0	110.0	120.0	130.0	137.8	150.0	160.0	170.0	179.9	190.0	200.0	210.0									
9	Total Equity		100.0	110.0	120.0	130.0	140.0	150.0	160.0	170.0	180.0	190.0	200.1	210.0									
10	Balance Check:		-	-	-	-	1	-	-	-	1	-	1	-									
11	Error Message: First Error in May-20																						
12																							
13	Error Reporting Options																						
14																							
15	INDEX MATCH		May-20	✓																			
16	LOOKUP		Sep-20	✗																			
17	HLOOKUP		#VALUE!	✗																			
18																							
19																							
20																							
21																							
22																							
23																							

MATCH(1,J13:O13,0) equals 5, *i.e.* the first period the balance sheet does not balance in is Period 5. But we can do better than that. **INDEX(J4:O4,5)** equals May-20, so combining the two functions:

INDEX(J4:O4,MATCH(1,J13:O13,0))

equals May-20 in one step. This process of stepping out two calculations and then inserting one into another is often referred to as “staggered development”. No, this is not how you construct a financial model late in the evening after having the odd drink or two!

Do note how flexible this combination really is. We do not need to specify an order for the lookup range, we can have duplicates and the value to be returned does not have to be in a row / column below / to the right of the lookup range (indeed, it can be in another workbook never mind another worksheet!). With a little practice, the above technique can be extended to match items on a case sensitive basis, use multiple criteria and even ‘grade’.


More Excel Functions next month.

Beat the Boredom Suggested Solution

The issue this month may be common and useful for anyone who uses Excel to produce cumulative / running total reports frequently.

This month’s challenge was to write a **formula** on a new column of the provided table to calculate the running total of column **Number** regardless of whether the Excel Table (*i.e.* created using **Insert -> Table** or **CTRL + T**) is filtered or not. The result should look like the column **Total** generated on the right (*below*):

Letter	Number
a	1
b	2
c	3
d	4
dd	(20)
e	5
f	6
g	7
h	8
i	9
j	10



Letter	Number	Total
a	1	1
c	3	4
d	4	8
dd	(20)	(12)
f	6	(6)
h	8	2
i	9	11
j	10	21

As always, there were some requirements:

- the formula needed to be within just one column (no “helper” cells)
- this was a formula challenge; no Power Query / Get & Transform or VBA!
- the formula still calculated the correct running total when the table was *filtered*.

Suggested Solution

Normally, we would think of using **SUM** to sum up the range from the top row to current row to calculate the running total. However, the result will not be adjusted if the table is filtered, especially if the first row goes walkabout.

Hence, we think of using an Excel function that ignores any rows that are not included in the result of a filter. That function is **SUBTOTAL**.

Firstly, the current row of column Number can be easily identified using the Table reference as below.

[@Number]

You may recall the **@** symbol denotes the implicit intersection and literally means the value of the ‘**Number**’ field for the row you are on.

Secondly, we can easily select a range from the top row to current row and anchor the top row reference. However, it is not dynamic enough when a top row can be added or deleted. Therefore, we use **INDEX** to find the top cell of the ‘**Number**’ field as follows.

INDEX([Number],1)

As you may know, **INDEX** returns not only a value but also a cell reference. This means you can use it along with a colon “:” to refer to a range of cells. The range that we need for the running total is as follows:

INDEX([Number],1):[@Number]

Finally, we then use **SUBTOTAL** to generate the running total. **SUBTOTAL** ignores all filtered rows regardless of which function number in the first argument is used.

=SUBTOTAL(9,INDEX([Number],1):[@Number])

The number nine [9] represents the **SUM** function including hidden rows while number 109 works similarly but ignores hidden rows. These hidden rows need to be caused by either hiding or grouping (not filtering). Depending upon your purpose for the running total, you can choose which number to use.

Until next month.

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	22 - 24 Nov 2023	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	5 Dec 2023	09:00-17:00 AEDT	(-1 day) 22:00-17:00 GMT	1 Day
Online (Australia)	Financial Modelling	6 - 7 Dec 2023	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. We've started going through the alphabet actions. After being sorted for Es last month, we've whizzed through the alphabet to say HI...

Keystroke	What it does
<u>H</u> ELP	Calls Help
SHIFT + <u>H</u> ELP	What is..?
<u>H</u> OME	Select the first cell in the row
CTRL + <u>H</u> OME	Select cell to reset window (or pane)
CTRL + SHIFT + <u>H</u> OME	Extend selection to the cell that resets the window or pane
SHIFT + <u>H</u> OME	Extend selection to first cell in the row
<u>I</u> NSERT	Toggle Overwrite mode in Edit mode
CTRL + <u>I</u> NSERT	Copy
SHIFT + <u>I</u> NSERT	Paste

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