

Sum Product

NEWSLETTER #132 - November 2023

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



Copilot support finally comes to Excel – albeit for Python in Excel initially, but hey, it's a start! We divulge more later in the weighty tomes of our latest newsletter novella.

Excel Virtually Global has been and gone: did we see you there? Numerous speakers from all over the world continue to give their time for free, all in the name of charity. It was our biggest event yet in terms of attendances. If you missed it, don't worry – selected highlights are available.

This month we look at left aligning chart labels, a surprisingly common request in recent months – we thought it was high time we addressed it! And then there are the regulars: Beat the Boredom Challenge, plus more tips on Charts & Dashboards, our new Excel for Mac article, plus Visual Basics, Power Pivot Principles, Power Query Pointers, Power BI Updates and Excel Updates. We also look at **L** and **M** in our A to Z of Keyboard Shortcuts and there is even time to **MAX** out on this month's A to Z of Excel functions.

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

Liam Bastick, Managing Director, SumProduct



Excel Virtually Global 2023 Been and Gone



We came, we saw, we exited stage right. SumProduct wants to thank the 40+ presenters who generally gave their time to this charity event – and to the 3,500+ attendees who took time to tune into one or more sessions.

The presenters waived their magic and engaged the audience. If you missed it, it's still possible to watch many of the sessions. Simply go to www.sumproduct.com/news/article/excel-virtually-global-2023-in-case-you-missed-it for further details.

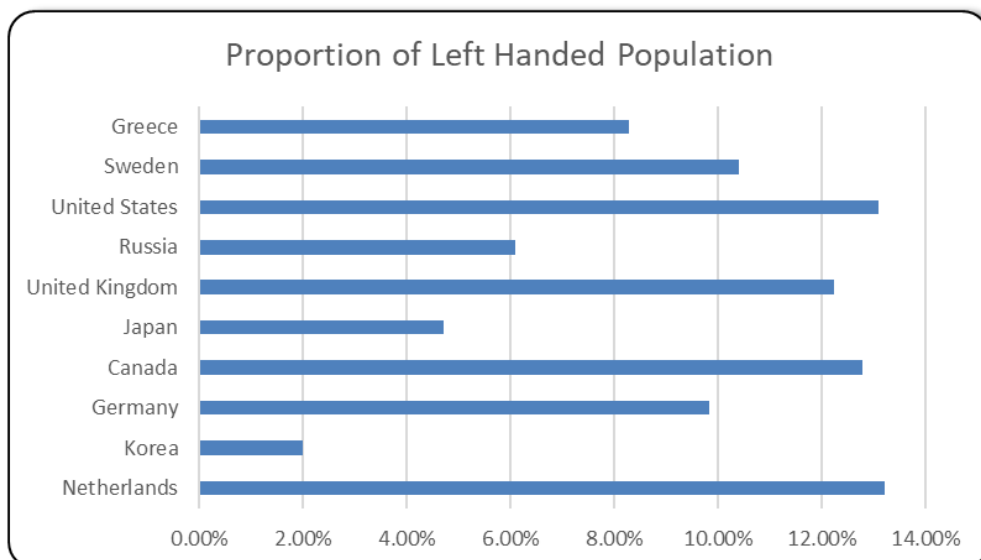
Please don't forget, this event was for charity. We'd love you to donate to a worthy cause of your choice and just let us know by filling in the form using the QR code:



bit.ly/EVG2023Charity

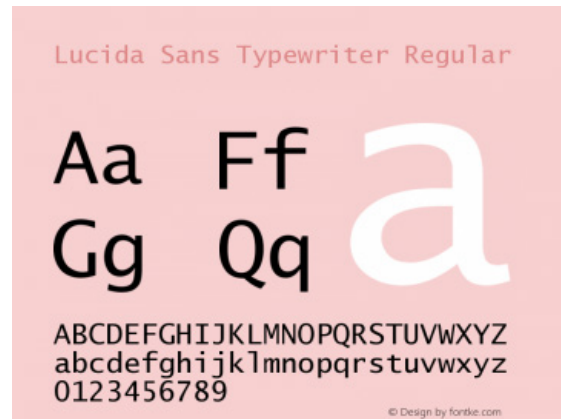
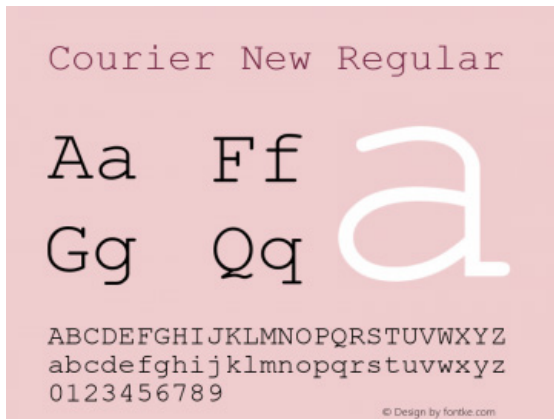
Left Aligning Chart Labels

Every now and again, we get several readers / clients ask the same question as if they have coordinated the “attack”. This month's article addresses one such question: how do you shift text alignment to the left for the vertical text labels of a bar chart - which default to the right.



The hack used to align the text to the left is using monospace type fonts. Monospace type fonts are fonts where letters and characters each occupy the same amount of horizontal space. Think of the good

old typewriter which uses monospace fonts and old-fashioned typeset print media. There are two monospaced fonts that come default with Windows / Office: Courier New and Lucida Sans Typewriter.

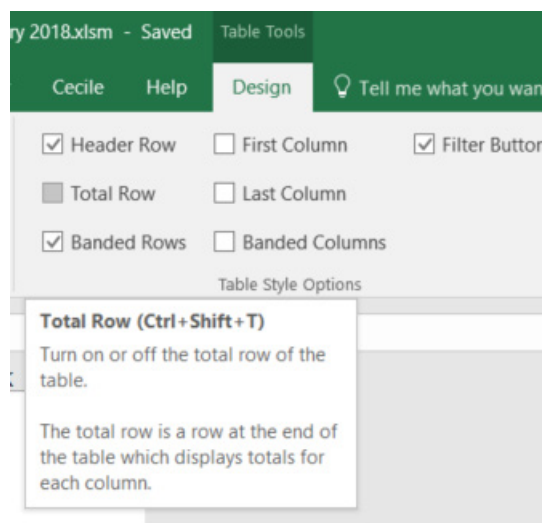


What this means is that spaces can be appended to the text to generate spaces as filler within the label to give the illusion of being left aligned.

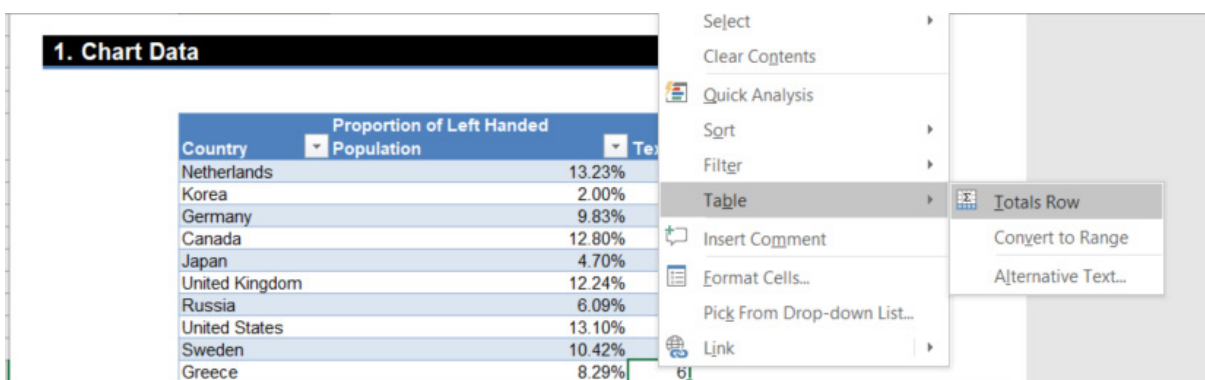
How many spaces would be needed to be added to the end? Well, let's look at the **longest** text label and use that as our base. The number of characters of a table can be found with the **LEN** function. Notice how the data is stored in a table. Let's add a column to the right with the following formula:

=LEN([@Country])

In order to find the maximum, let's put in a totals row. The total row can be added by using the 'Table Style Options' subgroup of Design tab under 'Table Tools' in the Ribbon:



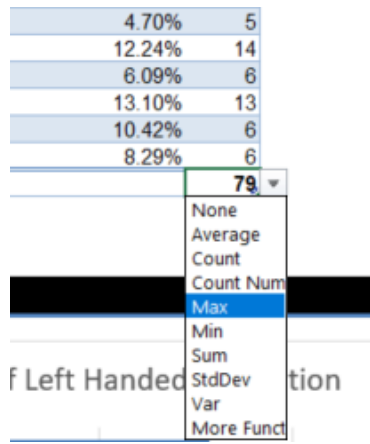
or alternatively in the contextual menu upon right-clicking a cell in the Table:



That will make the Totals Row – but this calculates the **SUM** of the Table:

Country	Proportion of Left Handed Population	Tex
Netherlands	13.23%	11
Korea	2.00%	5
Germany	9.83%	7
Canada	12.80%	6
Japan	4.70%	5
United Kingdom	12.24%	14
Russia	6.09%	6
United States	13.10%	13
Sweden	10.42%	6
Greece	8.29%	6
Total		79

Notice that there is a drop-down menu on right of the **Total** cell.



From there, **MAX** may be selected and it will change the formula in the bar to be:

=SUBTOTAL(104,[Text Length])

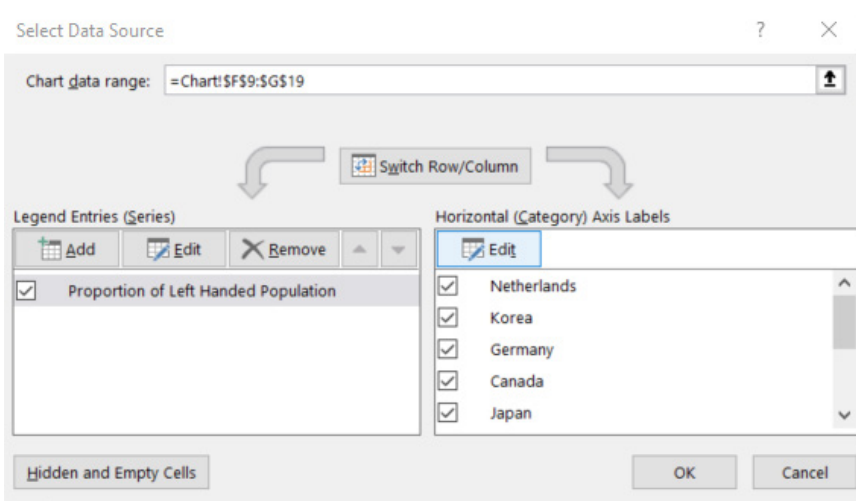
Don't forget to change the row label to "**Max**" instead of "**Total**"! Now, the table looks like this:

Country	Proportion of Left Handed Population	Tex
Netherlands	13.23%	11
Korea	2.00%	5
Germany	9.83%	7
Canada	12.80%	6
Japan	4.70%	5
United Kingdom	12.24%	14
Russia	6.09%	6
United States	13.10%	13
Sweden	10.42%	6
Greece	8.29%	6
Total		14

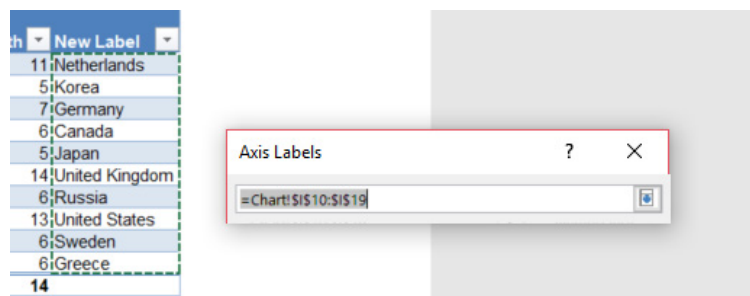
So the number of spaces required to fill in the rest of the label would be the maximum of the length LESS the actual length of the current label. The spaces can then be generated using the **REPT** function with the following formula:

=[@Country]&REPT(" ",Tbl_LeftHandedness[#[Totals],[Text Length]]-[@Text Length])

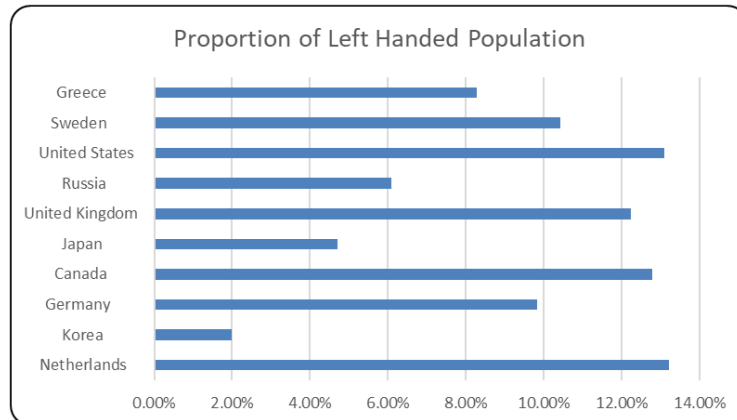
The next step is to point the labels of the chart to the new column. Go to 'Select Data Source' (found on the right click menu of the chart or the Data subgroup of the Design tab under the Chart Tools tab in the Ribbon).



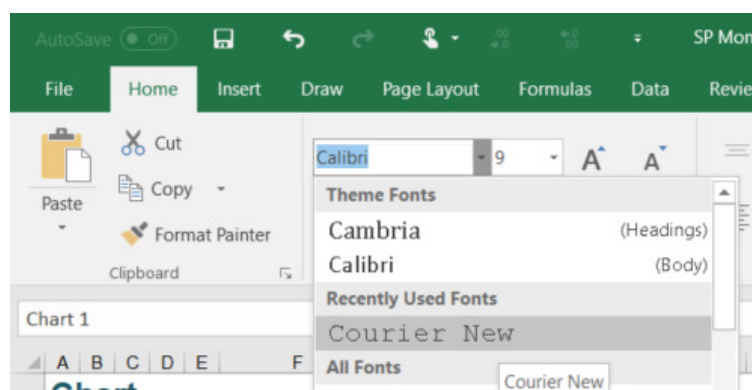
Click the Edit button under 'Horizontal (Category) Axis Labels' and select the **New Label** column.



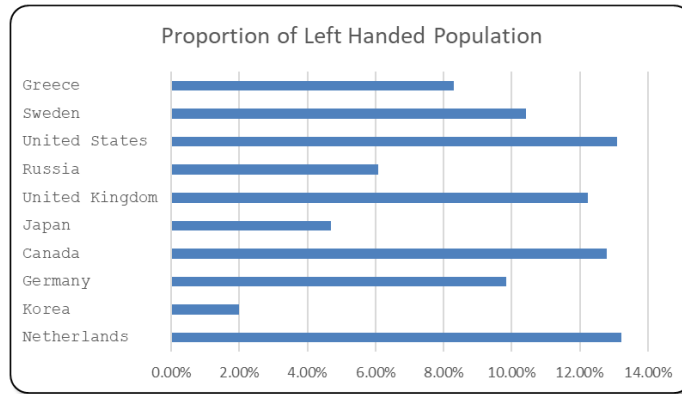
The chart will now be updated. Let's see how it looks.



Notice how the labels appear strangely offset? That's because the font is not spaced equally between each letter! Change the font to one of the monospaced fonts above by clicking on the axis labels and selecting it in the Font subgroup of the Home tab in the Ribbon.



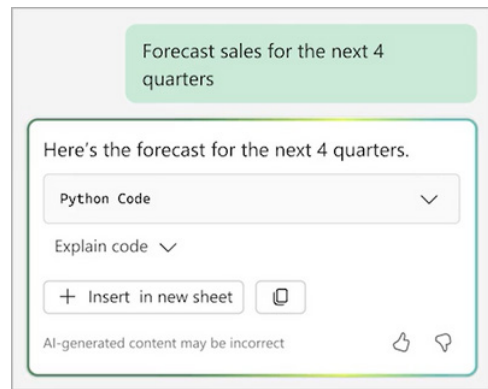
The chart will then be updated. Let's again see how it looks.



Hope that tickled the left side of the brain!

Copilot Support for Python in Excel

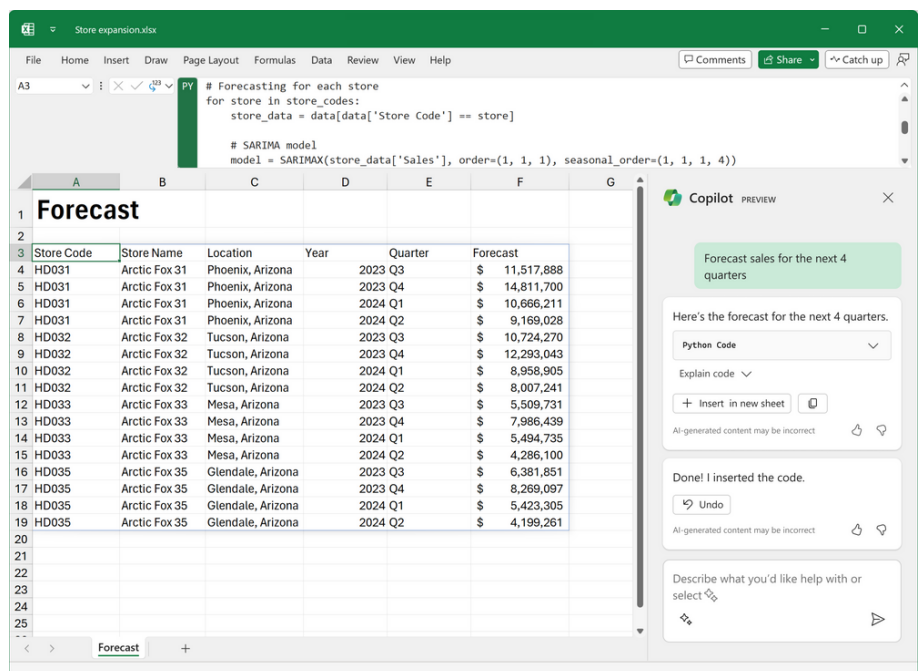
They are all coming together like buses. After the recent announcements of Python in Excel, Copilot joins the gang too. For those (like me) who know little about programming in Python, Copilot allows you to leverage the artificial intelligence (AI) to exploit Python in Excel.



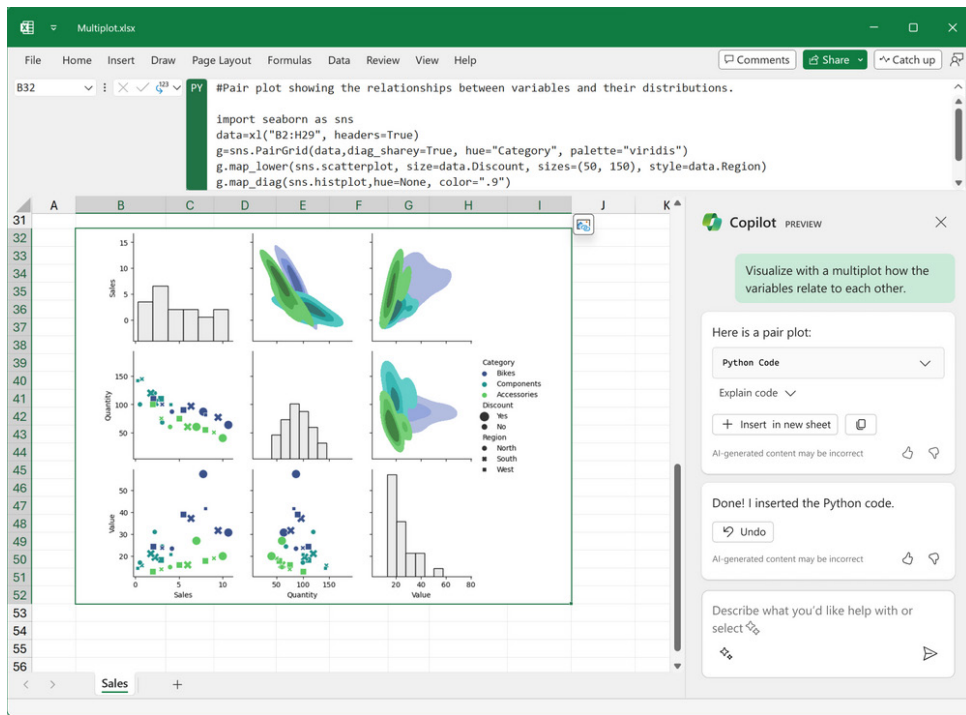
In addition to helping you write formulae, format your data, and perform data analysis, Copilot in Excel will help you analyse and explore your data using Python code. All you have to do is use everyday language ("Natural Language") to describe what you want to do with your data, and Copilot will generate and insert working Python code in the Excel grid for you. Perhaps it can do the washing and ironing too.

Whether you are new to Python or an experienced user, Copilot enables you to achieve more with your data in Excel by accessing advanced analytics. You can use Copilot as a learning tool, to unlock productivity, and as creative inspiration. Here are some of the benefits of using Copilot and Python in Excel:

- **Use natural language for data analysis with Python.** Simply describe the analysis you want to undertake, and Copilot will generate and insert Python code for you. For example, you can type, "Forecast sales for the next 4 quarters" and Copilot will produce a Python cell with the code and the forecast



- Access popular libraries and visualisations.** You don't need to know Python to get started. Copilot leverages the power and flexibility of Python and its popular libraries, such as pandas, matplotlib, and scikit-learn to name a few. You can use Copilot to perform advanced analytics such as forecasting, clustering, optimisation, causal models, statistical tests, classification, sampling, and more. You can also create a variety of charts that are not available in Excel alone, such as boxplots, network graphs and pairplots



- Break down complex problems iteratively.** Use Copilot as your sounding board and guide to perform your data analysis in an iterative fashion. Copilot remembers the context of your previous queries and results, allowing you to continue the conversation with additional analyses, follow-up questions and ideas

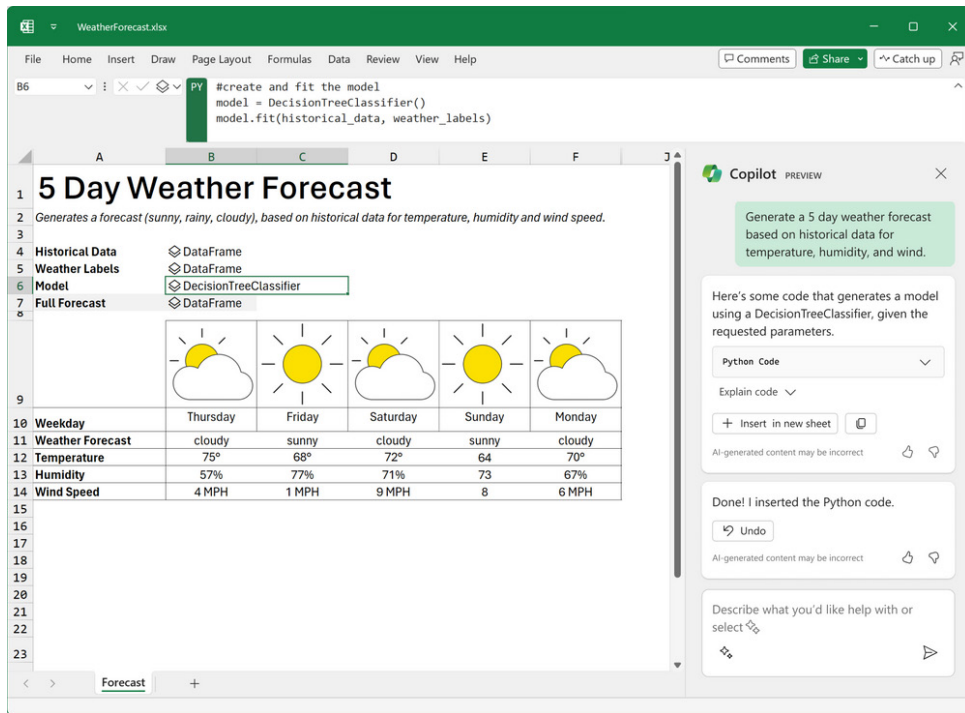
The screenshot shows an Excel spreadsheet with a Python code block in cell C7. The code is as follows:

```
# Define regex pattern for various date formats
date_patterns = [
    r"^\b(\d{2})[-./](\d{2})[-./](\d{4})\b'", # DD-MM-YYYY or DD/MM/YYYY or DD.MM.YYYY
    r"^\b(\d{2})[-./](\d{2})[-./](\d{2})\b'", # DD-MM-YY or DD/MM/YY or DD.MM.YY
    r"^\b(\d{1,2})[-./](\d{1,2})[-./](\d{4})\b'", # D-M-YYYY or D/M/YYYY or D.M.YYYY
]
```

The resulting table, titled "Extract and Format Dates", shows the original text and the formatted dates. The table is as follows:

Original Text	Formatted Text
Today's date is 23/08/2023	2023-08-23
I remember the date 12-05-19.	2019-05-12
Another date format: 22-07-15.	2015-07-22
Dates in different formats: 15.5.21	2021-05-15
A date in the middle: 07.04.1995	1995-04-07
Random text without dates.	
Date format test: 14-08-23	2023-08-14
New date: 25/10/2022	2022-10-25
No dates here.	
Strange format: 01-20000-01	
Yet another date: 31-12-2023	2023-12-31
Different format: 31.12.2023	2023-12-31
Mixed formats: 15/01/25	2025-01-15
An unusual date: 21.01.20001	

The Copilot interface on the right shows a preview of the table and a text box that says: "Extract and apply consistent formatting to the list of dates." Below this, there is a section titled "Let's extract the dates using a regular expression and then format them consistently using the following code." with a "Python Code" dropdown and an "Insert in new sheet" button.



Copilot adheres to Microsoft’s AI principles and Responsible AI Standard, and it follows their commitment to data security and privacy. Python in Excel runs in a secure container on the Microsoft Cloud with enterprise-level security as a compliant Microsoft 365 connected experience.

Security Concerns

Many were concerned when Power BI Service was introduced regarding privacy and security. Microsoft has learned from that feedback and has been proactive in explaining security protocols this time.

Python code runs on “hypervisor isolated containers built on Azure Container Instances” (I am glad I cleared that up). The container has Python and a curated set of secured libraries provided by Anaconda. The environment uses the Anaconda Distribution for Python, which contains source-built Python and libraries, provided directly from Anaconda. With this borne in mind, it should be noted that the Python code:

- does not have access to your computer, devices or account
- does not have network access
- does not have access to a user token
- can access data through the references via the built-in `xl()` function as part of a Python formula. This means that Python formulae have access to read cell values within the workbook, based upon the cell reference, or values from external data sources, through the Power Query connection name
- returns output to your workbooks through the `=PY()` Excel function, which displays the result of the Python code in the cell where the function is entered. Python functions cannot return other object types like macros, VBA code or other formulae
- doesn’t have access to other properties in the workbook, such as formulae, charts, PivotTables, macros or VBA code.

Python in Excel follows the same security policies as Excel when it comes to opening workbooks from the internet or untrusted sources. If you open a workbook that contains Python code from the internet, Excel Protected View won’t run Python formulae in the workbook. If a workbook is opened with Microsoft Defender Application Guard, Python formulae do not run by default. In addition, Python in Excel runs the

Python formulae in the untrusted workbook within its own dedicated hypervisor isolated container, helping prevent potential interaction or interference with other Python code running from other open workbooks.

Python in Excel uses containers to run Python code on Azure. Containers are isolated and secure environments that can run applications and services without affecting the rest of the system. To help ensure the security and reliability of the containers, regular updates and patches are applied to them.

Python and the libraries are updated in the following ways to help keep your data secure and your numerical results consistent:

- patches are applied to the underlying operating system that the Azure Container Instance runs on. This helps ensure that the container is protected from vulnerabilities and exploits that may affect your data. This is done automatically with no input required from you
- once Python in Excel becomes Generally Available (it is presently only available in the Beta Channel Insider level), periodic updates of the Python environment will be released, with Python and the libraries provided by Anaconda. These versions include the latest security fixes and enhancements from the Python and Anaconda communities. These releases will be announced through blog posts and documentation. This will be done automatically with no setup required, and Microsoft will default to the latest environment. Existing workbooks will still calculate against the version of the environments the workbook was created on, and users will be prompted to upgrade if there is a newer version. This approach helps ensure that your numerical results stay consistent and allows you and your organisation to stay up to date.

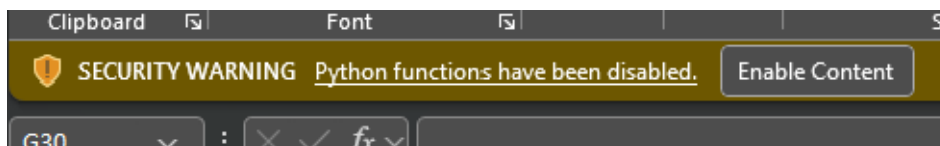
It should also be noted that the Microsoft privacy policy – the same one you accept for other Microsoft products such as Excel, Word and PowerPoint – still applies.

Furthermore, it is possible to update the registry to toggle security warnings for Python in Excel. The following commands show how to update the registry to change security warning settings for Python in Excel. You can run the commands from an elevated command prompt on a Windows device.

- Use this command to disable all security warnings. This is the default setting for Python in Excel:

```
reg add HKCU\software\policies\microsoft\office\16.0\excel\security /v PythonFunctionWarnings /t REG_DWORD /d 0 /f
```

- Use this command to enable a security prompt when opening a workbook that contains a Python formula. It enables a Security Warning notice in the Excel business bar



```
reg add HKCU\software\policies\microsoft\office\16.0\excel\security /v PythonFunctionWarnings /t REG_DWORD /d 1 /f
```

- Use this command to disable all Python functions from running. Python functions will return the error *#BLOCKED!*

```
reg add HKCU\software\policies\microsoft\office\16.0\excel\security /v PythonFunctionWarnings /t REG_DWORD /d 2 /f
```

Microsoft will continue to monitor and adjust Python in Excel to keep data safe. This may involve future changes to the user experience and registry settings.

Availability

Copilot support for Python in Excel will be available in Preview later this year.

To use Copilot and Python in Excel, you will need access to Python in Excel and Microsoft 365 Copilot. It will initially be available in Excel for Windows in English.

To use Python in Excel, you will need to join the Microsoft 365 Insider Program. You must choose the Beta Channel Insider level to get the latest builds of the Excel application. After you've installed the latest Insider build of Excel, open a blank workbook and take the following steps:

- select the Formulas tab in the Ribbon
- select 'Insert Python'
- in the dialog that appears, select the 'Try preview' button.

You may also enable the Python in Excel preview by entering **=PY** into an Excel cell and then choosing **PY** from the function AutoComplete menu. Selecting the **PY** function will trigger a dialog that allows you to enable the Preview.

The prerequisites for Microsoft 365 Copilot are a little more stringent. Before you can access Copilot, you must meet these requirements:

- the following applications must be deployed for your users, which will integrate with Microsoft 365 Copilot and other applications:

- Excel
- Exchange
- OneDrive
- Outlook
- PowerPoint
- SharePoint
- Teams
- Word

- To get started with the implementation process, you will need to deploy the application in the usual way for Microsoft 365 Apps.
 - **OneDrive Account:** you need to have a OneDrive account for several features within Microsoft 365 Copilot, such as saving and sharing your files
 - **New Outlook for Windows:** for integration of Microsoft 365 Copilot with Outlook, you are required to use the new Outlook for Windows, currently in Preview. You can switch to Outlook Mobile to access the new Outlook experience
 - **Microsoft Teams:** to use Microsoft 365 Copilot with Microsoft Teams, you must use the Teams desktop client or web client. You can download the desktop client or sign into the web app at <https://teams.microsoft.com>. Both the current and the new version of Teams are supported
 - **Microsoft Loop:** to use Copilot in Microsoft Loop, you must have Loop enabled for your tenant.

For Enterprise subscribers wishing to use Microsoft 365 Apps, users must be on the Current Channel to access Copilot.

Microsoft previously stated that support for the Monthly Enterprise Channel was available for Microsoft 365 Copilot. As Microsoft continues to make frequent product updates and enhancements during the early access program, the time between updates in the Monthly Enterprise Channel limits Microsoft's ability to provide an optimum Copilot experience on desktop clients. Going forward, users in the early access program must be on the Current Channel to receive Copilot updates when they become available. It is expected support will be available in the Monthly Enterprise Channel in the future.

You can manage Microsoft 365 Copilot licenses from the **Microsoft 365 admin center**. You can assign licenses to individual users or to groups of users, as well as reassign licenses to other users. To access license management in the Microsoft 365 admin center, go to **Billing -> Licenses**.

You may also assign licenses in bulk to groups of users through the Azure admin center or assign licenses to users with PowerShell.

Simple if you're an expert!?

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...*

A positive integer is said to be prime if it has precisely only two distinct positive integers that divided it (i.e. one [1] and itself). This is why one [1] is not a prime number, as it requires two [2] distinct divisors.

With the above borne in mind, using Excel, can you create a formula, without using a user-defined function or VBA, that will determine whether a number is prime?

Example

Number in Question

Number

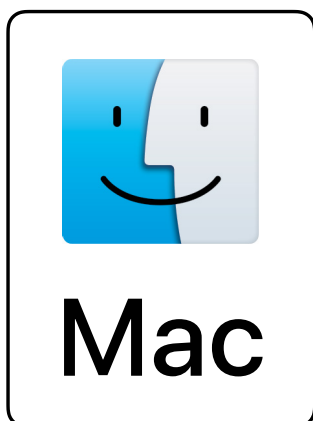
Result

Sound easy? Some of you might think this is trivial and have a solution in seconds. We're going to bet if you are one of those readers, you are probably wrong – but more on that later in this newsletter. No reading ahead!

Excel for Mac

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

Microsoft Excel for Mac, we'll show you some important PivotTable options that don't show up in the Field Settings dialog on Mac, even though they are available on Windows, and we'll show how to work around that limitation.



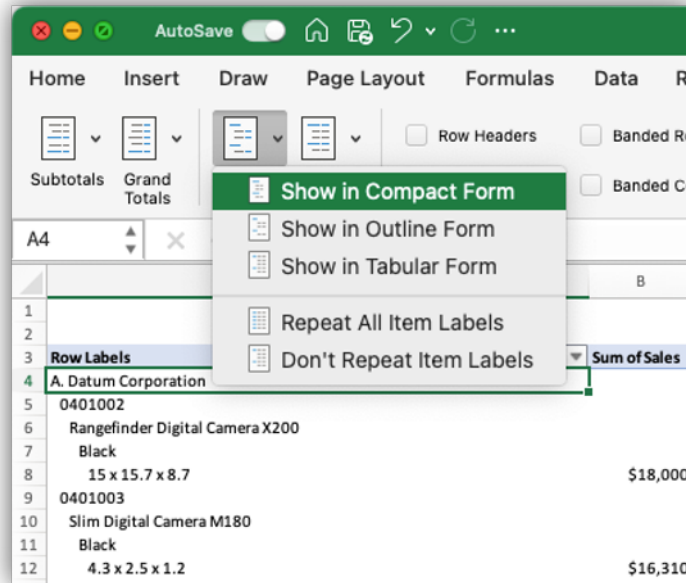
PivotTables in Excel have three [3] popular layout options:

1. Compact (default)
2. Tabular
3. Outline.

The differences between the layout options are described below. But did you know that you can mix and match these within the same PivotTable? It's useful when you have some fields that work well in one layout, but other fields that work better in a different layout.

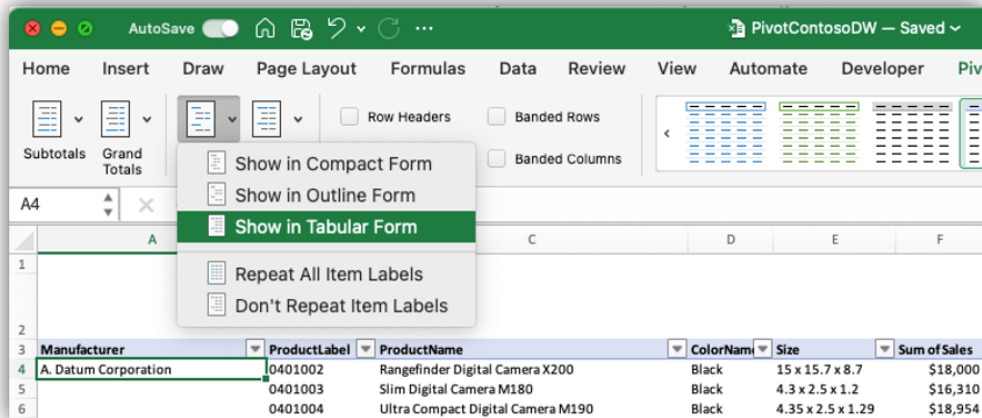
To begin, we need to explain why you'd want to use more than one layout option in a PivotTable, and then we'll explain how this pertains to Excel for Mac. Let's say your data has several fields that all describe the products you're selling, and you want to find the sum of sales for each. For example, you have fields like manufacturer, product number, name, colour and size, and you've added them all to the 'rows' area of your PivotTable.

The default layout is the Compact form, which puts each of those fields on a different row, as in the example below:



We can see the manufacturer, product number, name, colour and size on different lines. Compact layout keeps the PivotTable narrow by putting all the fields on the Rows section of the PivotTable into one column. Each field is slightly indented from its parent field. It keeps the table narrow, because they're all in a single column, but it uses four [4] rows for each 'Sum of sales' value. We need it to be easier to read have a neater appearance with less wasted space.

Tabular layout is better for this data, since several fields are used to describe the product for which you're getting the 'Sum of sales'. It makes sense to keep them together on a row. However, you may have other fields such as a manufacturer or category that apply to many products. These would be great to have in Outline form, so they may take up less space. In the Tabular layout example below, "A. Datum Corporation" is on the same line as the product information, but you might want to have it on the line above and move the other fields underneath it to save space.



When a standard layout doesn't meet your needs, you can combine layouts. It's simple to do if you're using Excel for Windows. If you have a Mac, it's not as easy, but we're going to show you how to make it easy.

Let's see how easy this is on Windows. You would just pick the standard layout for the entire PivotTable that gets closest to what you want, and then choose a different layout for specific fields that you want to show differently:

1. Go to the 'PivotTable Design' tab of the Ribbon -> Report Layout and choose the one that makes sense for most of your fields
2. Select the field that should have a different layout and open the 'Field Settings' dialog. You can either select a cell with that field in the PivotTable and choose the 'Field Settings' button on the Ribbon, or you can open the context menu for the field in your PivotTable fields pane
3. On the 'Layout & Print' tab of the dialog, choose the layout you like best
4. Click OK.

In the example shown here from Windows, the Tabular form has been set for the PivotTable, but then we've changed the settings for the Manufacturer field to have Outline form selected and 'Display labels from the next field in the same column'. This put the Manufacturer field

into Compact form, so "A. Datum Corporation" appears on its own line above the products. The rest of the fields stay in Tabular form. It saves space, is easier to read and looks neater.

Row Labels	ProductName	ColorName	Size	Sum of Sales
A. Datum Corporation				
0401002	Rangefinder Digital Camera X200	Black	15 x 15.7 x 8.7	\$18,000
0401003	Slim Digital Camera M180	Black	4.3 x 2.5 x 1.2	\$16,310
0401004	Field Settings			1.29
0401005	Source Name: Manufacturer			.2
0401006	Custom Name: Manufacturer			.2
0401007	Subtotals & Filters			.2
0401008	Layout & Print			.2
0401009	Layout			.2
0401010	<input checked="" type="radio"/> Show item labels in outline form <input checked="" type="checkbox"/> Display labels from the next field in the same column (compact form) <input type="checkbox"/> Display subtotals at the top of each group <input type="radio"/> Show item labels in tabular form			.4
0401011				.4
0401012				.4
0401013				.4
0401014				\$30,908

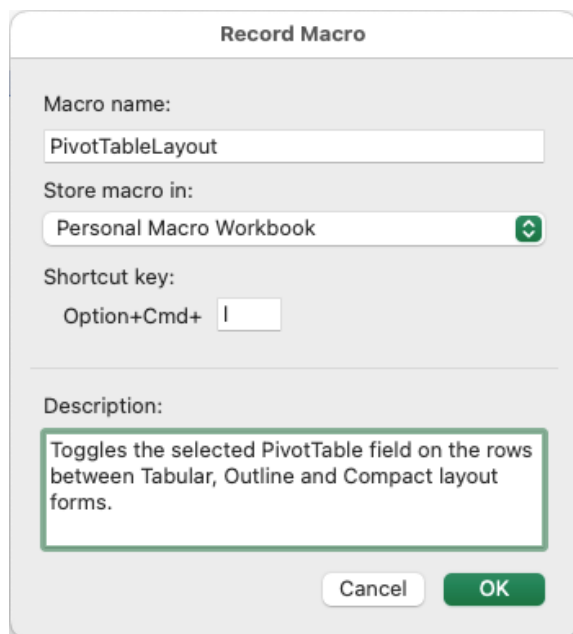
It's different on Mac.

Unfortunately, the 'Layout & Print' tab of the 'Field settings' dialog isn't available in Excel for Mac (as of September 2023), so how can we set a different layout for the Manufacturer field? This is a good example of when we may need to use a macro. The nice thing about this case, is

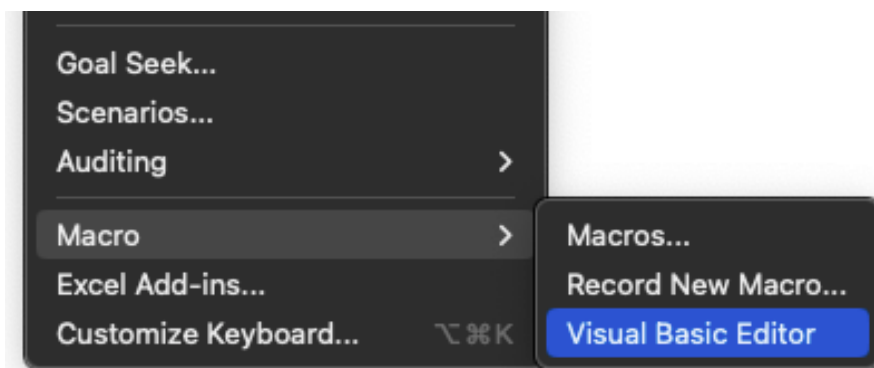
that you don't need to add the macro to the workbook. It can just be a "personal" macro, so you won't need to save your file as XLSM and you won't get a macro warning when you open it. It will just be saved to your [Personal Macro Workbook](#).

There are a few ways to create a new macro, but we'll just describe one way here.

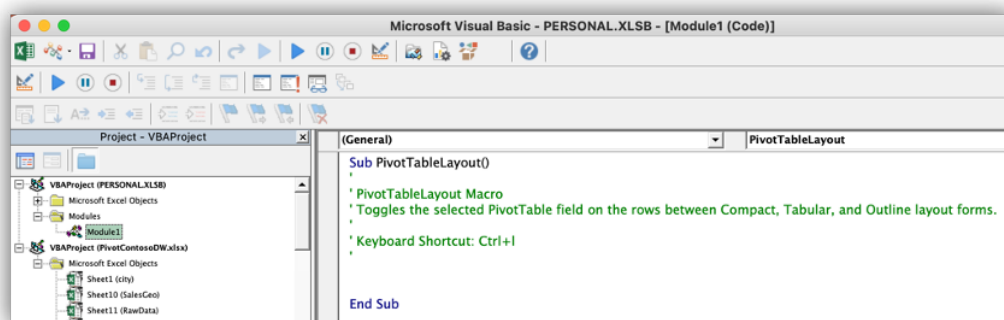
1. Record a new macro and in the 'Macro Name' field, type "PivotTableLayout". Choose to store it in 'Personal Macro Workbook'. If you want to assign a shortcut, type the desired letter in the Shortcut key field. Here we used I (lowercase L), so the shortcut will be **Opt + Cmd + L** to run the macro. Type a description:



2. Press OK
3. Now press the 'Stop Recording' button on the status bar near the bottom left of the Excel window
4. Go to **Tools** menu -> **Macro** -> **Visual Basic Editor**



5. In the VBE window, select and double-click on the new module that was created when you recorded the macro. If this is your first macro, it will be called 'Module1', otherwise it may have a different name



6. Copy the code from the code box below in this article
7. Go back to the Visual Basic Editor, place your cursor in the line above 'End Sub' and add a new line. Then paste the code you copied.

```

'If the user doesn't select a PivotTable cell there will be an error getting the PivotTable
name,
'so resume the next command which will give the user a message about it.

On Error Resume Next

Dim ActivePT As String
Dim ActivePTField As String

ActivePT = ActiveCell.PivotCell.Parent.Name
'If the active cell is not in a PivotTable, tell the user to try again.
If ActivePT = "" Then
MsgBox "Select a cell in the rows area of a PivotTable and try again."
Exit Sub
End If
  
```

```

'if the active cell isnot in the Rows section of the PivotTable tell the user to try again.
If ActiveCell.PivotField.Orientation = xlRowField Then
    ActivePTField = ActiveCell.PivotField.Name
Else
    MsgBox "Select a cell in the rows area and try again."
End If

'If the field is in Tabular form, switch it to Outline (Outline with CompactRow=False)
If (ActiveSheet.PivotTables(ActivePT).PivotFields(ActivePTField).LayoutForm = xlTabular) Then
    With ActiveSheet.PivotTables(ActivePT).PivotFields(ActivePTField)
        .LayoutForm = xlOutline
        .LayoutCompactRow = False
    End With
    'Exit the sub, since the next step would change the layout again.
    Exit Sub
End If

'If the field is in Outline form, switch it to Compact form (CompactRow=True)
If (ActiveSheet.PivotTables(ActivePT).PivotFields(ActivePTField).LayoutForm = xlOutline) And
-
(ActiveSheet.PivotTables(ActivePT).PivotFields(ActivePTField).LayoutCompactRow = False)
Then
    'Switch to compact form by setting CompactRow = True
    ActiveSheet.PivotTables(ActivePT).PivotFields(ActivePTField).LayoutCompactRow = True
Else
    'Else, it was already in Compact form, so switch back to Tabular form
    ActiveSheet.PivotTables(ActivePT).PivotFields(ActivePTField).LayoutForm = xlTabular
End If

```

Congratulations!

Now you can set your PivotTables to use a combination of layout options that works best for your data. To try your new macro, just go to a PivotTable that has at least two [2] fields in the Rows area. Select a cell in the rows area and run the macro. Hopefully, the 'Layout & Print' options will be added to the 'Field Settings' dialog on Mac, but until then, this macro will be handy for Mac users.

Here's an example of what you'll see in your PivotTable:

1. Let's say you chose Tabular layout for the PivotTable, since that works well for the ProductLabel, ProductName, ColorName, and Size fields. The PivotTable would look like this:

3	Row Labels	ProductLabel	ProductName	ColorName	Size	Sum of Sales
4	A. Datum Corporation	0401002	Rangefinder Digital Camera X200	Black	15 x 15.7 x 8.7	\$18,000
5		0401003	Slim Digital Camera M180	Black	4.3 x 2.5 x 1.2	\$16,310
6		0401004	Ultra Compact Digital Camera M190	Black	4.35 x 2.5 x 1.29	\$18,954
7		0401005	All in One Digital Camera M200	Black	4.2 x 2.3 x 1.2	\$13,743
8		0401006	Interchangeable lens Non-SLR Digital Camera X250	Black	3.4 x 0.9 x 2.2	\$44,381
9		0401007	Super-zoom Digital Camera X300	Black	3.4 x 1.1 x 2.2	\$18,096
10		0401008	Consumer Digital Camera E100	Black	3.5 x 0.9 x 2.2	\$45,944

- Run the macro once to switch the first field, Manufacturer, from Tabular to Outline layout. Notice the manufacturer field is now on a line above the other fields:

3	Row Labels	ProductLabel	ProductName	ColorName	Size	Sum of Sales
4	A. Datum Corporation					
5		0401002	Rangefinder Digital Camera X200	Black	15 x 15.7 x 8.7	\$18,000
6		0401003	Slim Digital Camera M180	Black	4.3 x 2.5 x 1.2	\$16,310
7		0401004	Ultra Compact Digital Camera M190	Black	4.35 x 2.5 x 1.29	\$18,954
8		0401005	All in One Digital Camera M200	Black	4.2 x 2.3 x 1.2	\$13,743
9		0401006	Interchangeable lens Non-SLR Digital Camera X250	Black	3.4 x 0.9 x 2.2	\$44,381
10		0401007	Super-zoom Digital Camera X300	Black	3.4 x 1.1 x 2.2	\$18,096
11		0401008	Consumer Digital Camera E100	Black	3.5 x 0.9 x 2.2	\$45,944

- Run the macro again to switch the Manufacturer field to Compact layout, with the next field appearing below the manufacturer name in the same column.

More 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, and look at sparklines.

A sparkline is a tiny chart in a worksheet cell that provides a visual representation of data. Sparklines are used to show trends in a series of values, such as seasonal increases or decreases, economic cycles or to highlight maximum and minimum values.

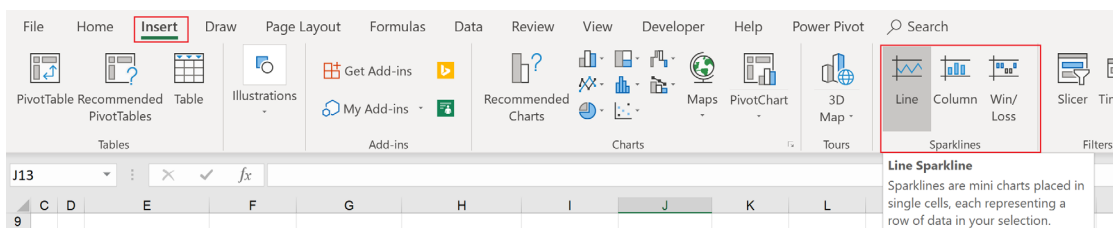
There are three types of sparklines:

- Line:** the Line sparkline summarises the data in small Line chart in a single cell
- Column:** the Column sparkline creates a miniaturised column chart of the data in a single cell
- Win / Loss:** the Win / Loss sparkline creates a miniaturised column chart that highlights positive and negative values in a single cell.

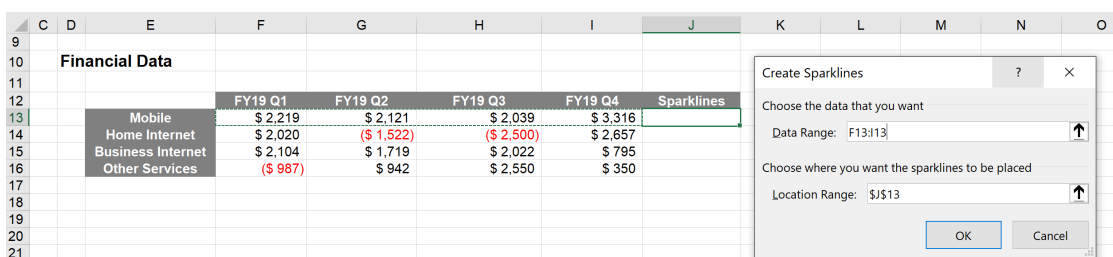
To illustrate sparklines, consider we have financial data for a store for the financial year 2019:

	C	D	E	F	G	H	I	J
9								
10			Financial Data					
11								
12				FY19 Q1	FY19 Q2	FY19 Q3	FY19 Q4	Sparklines
13			Mobile	\$ 2,219	\$ 2,121	\$ 2,039	\$ 3,316	
14			Home Internet	\$ 2,020	(\$ 1,522)	(\$ 2,500)	\$ 2,657	
15			Business Internet	\$ 2,104	\$ 1,719	\$ 2,022	\$ 795	
16			Other Services	(\$ 987)	\$ 942	\$ 2,550	\$ 350	
17								

To insert a sparkline in cell J13, we select cell J13, navigate to the Insert tab on the Ribbon, and select the Line option from the Sparklines group:



The 'Location Range' is where we want the Sparkline to appear. This has already been filled out for us, as cell J13. We just have to select the 'Data Range', which will be cells F13:I13, and then click OK.



We can then drag the cell down as if we're copying a formula:

	C	D	E	F	G	H	I	J
9								
10			Financial Data					
11								
12				FY19 Q1	FY19 Q2	FY19 Q3	FY19 Q4	Sparklines
13			Mobile	\$ 2,219	\$ 2,121	\$ 2,039	\$ 3,316	
14			Home Internet	\$ 2,020	(\$ 1,522)	(\$ 2,500)	\$ 2,657	
15			Business Internet	\$ 2,104	\$ 1,719	\$ 2,022	\$ 795	
16			Other Services	(\$ 987)	\$ 942	\$ 2,550	\$ 350	
17								

The process for creating Column and Win / Loss Sparklines is similar; it is just a matter of selecting a different option from the Sparklines group in the Ribbon.

	C	D	E	F	G	H	I	J
9								
10			Financial Data					
11								
12				FY19 Q1	FY19 Q2	FY19 Q3	FY19 Q4	Sparklines
13			Mobile	\$ 2,219	\$ 2,121	\$ 2,039	\$ 3,316	
14			Home Internet	\$ 2,020	(\$ 1,522)	(\$ 2,500)	\$ 2,657	
15			Business Internet	\$ 2,104	\$ 1,719	\$ 2,022	\$ 795	
16			Other Services	(\$ 987)	\$ 942	\$ 2,550	\$ 350	
17								
18								
19				FY19 Q1	FY19 Q2	FY19 Q3	FY19 Q4	Column
20			Mobile	\$ 2,219	\$ 2,121	\$ 2,039	\$ 3,316	
21			Home Internet	\$ 2,020	(\$ 1,522)	(\$ 2,500)	\$ 2,657	
22			Business Internet	\$ 2,104	\$ 1,719	\$ 2,022	\$ 795	
23			Other Services	(\$ 987)	\$ 942	\$ 2,550	\$ 350	
24								
25								
26				FY19 Q1	FY19 Q2	FY19 Q3	FY19 Q4	Win/Loss
27			Mobile	\$ 2,219	\$ 2,121	\$ 2,039	\$ 3,316	
28			Home Internet	\$ 2,020	(\$ 1,522)	(\$ 2,500)	\$ 2,657	
29			Business Internet	\$ 2,104	\$ 1,719	\$ 2,022	\$ 795	
30			Other Services	(\$ 987)	\$ 942	\$ 2,550	\$ 350	
31								

Sparklines provide a brief overview of your data, however as they are not as precise as charts, we only recommend they be used for users to “glance” at.

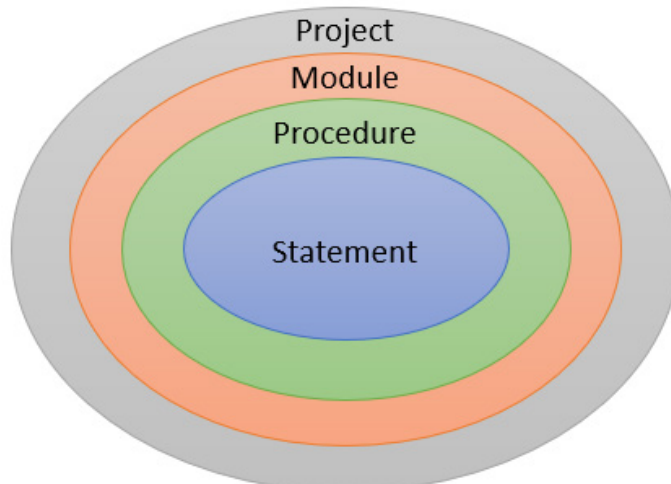
More next month...

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 revisit the principle of VBA.

Short and sweet this month – neither quality befits me.

Macros are organised and stored in the following way in VBA. The following common items are the principal components we need to know.



- A **statement** is a single command or action in a macro – that is, it's a line of code. For example, **Range("B3").Select** is a statement. If we think of VBA as a language, think of a statement as a sentence
- A **procedure** is, essentially, another way of referring to a macro, although there are other types of procedures as well, such as **functions**. A function is a procedure that returns a result
- A **module** is a collection of code. Think of a module as a code document. A module can contain several procedures. Like documents, modules can be saved as files, copied and shared
- A **project** is the collection of all modules and related VBA objects in our document, template or add-in. A project might have one or several modules, as well as other elements such as UserForms (dialog boxes).

Continued next month.

Power Pivot Principles

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

While working with data, it would be reasonable to assume that we have had add data to a data table based on values in a single column. Let's jump into our example, we will look at the following data:

	A	B	C
1			
2		Price ▾	Product Type ▾
3		28.00	3
4		51.00	5
5		88.00	4
6		23.00	2
7		95.00	2
8		91.00	3
9		65.00	5
10		90.00	2
11		62.00	4
12		68.00	1
13		54.00	3
14		72.00	3
15			

We have a simple Table with the product 'Price' and 'Product Type'. However, the numbers do not really tell us much about the product type. This is why we like to match these numbers to their corresponding text values. In this case:

	A	B	C
1			
2		Product Type ▾	Product Name ▾
3		1	Broccoli
4		2	Kale
5		3	Chicken Breast
6		4	Rice
7		5	Olive Oil
8			

Great, how do we bring in the 'Product Name' field into the first table?

One solution would be to use a nested IF formula to create a calculated column:

```
=IF([Product Type]=1,"Broccoli",  
  IF([Product Type]=2,"Kale",  
    IF([Product Type]=3,"Chicken Breast",  
      IF([Product Type]=4,"Rice",  
        IF([Product Type]=5,"Olive Oil", "")  
      )  
    )  
  )  
)
```

Rather nasty isn't it? Nasty sometimes works:

The screenshot shows an Excel spreadsheet with a table and a formula bar. The formula bar displays the nested IF formula: `=IF([Product Type]=1,"Broccoli", IF([Product Type]=2,"Kale", IF([Product Type]=3,"Chicken Breast", IF([Product Type]=4,"Rice", IF([Product Type]=5,"Olive Oil", "")))))`. The table below has columns for Price, Product Type, and Nested IF. The data in the table is as follows:

	Price	Product Type	Nested IF	Add Column
1	28	3	Chicken Breast	
2	51	5	Olive Oil	
3	88	4	Rice	
4	23	2	Kale	
5	95	2	Kale	
6	91	3	Chicken Breast	
7	65	5	Olive Oil	
8	90	2	Kale	
9	62	4	Rice	
10	68	1	Broccoli	
11	54	3	Chicken Breast	

Thinking down the line, what if we had to add more product types to our table and consequently more IFs to our nested IF formula?

There is an alternative for a scenario like this, introducing the SWITCH function. The SWITCH function requires the following syntax to operate:

SWITCH(expression, value, result [, value, result ..., else])

The arguments for SWITCH are as follows:

- the **expression** term requires a single scalar value that can be evaluated multiple times for each row or column
- the **value** term is the constant value that will be used to match the results of the **expression**
- the **result** is the scalar value that will be returned in the column if the **value** matches the **expression**
- the **else** term is the expression that will be returned in the column when no matches are found.

We can write the following **DAX** code to create a calculated column:

```
=SWITCH(
    [Product Type],
    1,"Broccoli",
    2,"Kale",
    3,"Chicken Breast",
    4,"Rice",
    5,"Olive Oil",
    ""
)
```

	Price	Product Type	Nested IF	SWITCH
1	28		3 Chicken Breast	Chicken Breast
2	51		5 Olive Oil	Olive Oil
3	88		4 Rice	Rice
4	23		2 Kale	Kale
5	95		2 Kale	Kale
6	91		3 Chicken Breast	Chicken Breast
7	65		5 Olive Oil	Olive Oil
8	90		2 Kale	Kale
9	62		4 Rice	Rice
10	68		1 Broccoli	Broccoli
11	54		3 Chicken Breast	Chicken Breast
12	72		3 Chicken Breast	Chicken Breast

There we have it: a **DAX** formula that is much easier to understand and readily expandable.

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

Power Query Pointers

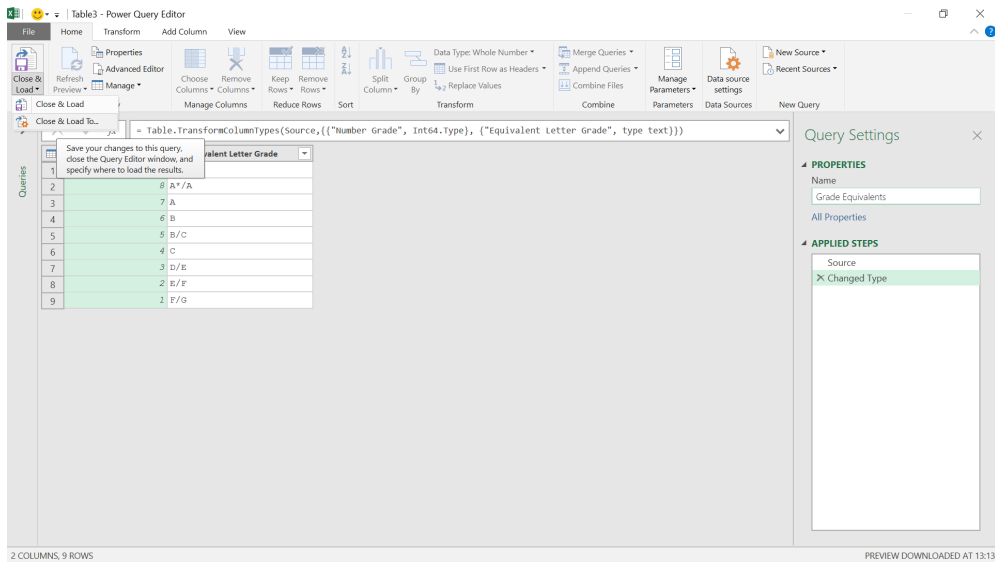
Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Office 365, Excel 2016 and 2019) from www.sumproduct.com/blog. If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month, we look at last month's problem again, but this time, we'll translate numbers to letters in multiple columns by joining tables.

We (still) have the exam results for some children who were lucky(!) enough to take their exams when the system was in the middle of changing from letters to numbers. We want to replace the numbers with

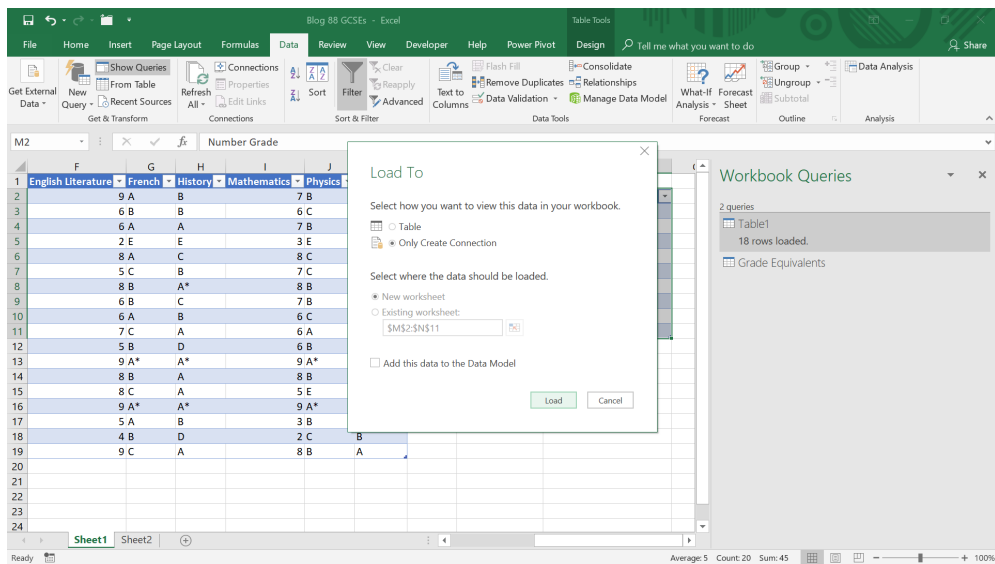
text so that we may see everything in the same format – this time we have two tables – we have created a table of the current numeric grades and their letter equivalent:

Name	Biology	Chemistry	Computer Science	English Language	English Literature	French	History	Mathematics	Physics	Spanish
1 Abby	B	C	A	8	9A	B	7B	A		
2 Brian	C	C	A	7	6B	B	6C	C		Number Grade Equivalent Letter G
3 Claire	A	A*	B	5	6A	A	7B	B		9A*
4 Dave	E	E	E	2	2E	E	3E	E		8A*/A
5 Ewan	B	C	B	7	8A	C	8C	C		7A
6 Frank	C	E	B	6	5C	B	7C	C		6B
7 Georgia	A*	A	A	9	8B	A*	8B	A*		5B/C
8 Harry	B	C	A	7	6B	C	7B	A		4 C
9 Ian	C	C	D	6	6A	B	6C	B		3 D/E
10 Jack	A	B	B	6	7C	A	6A	A		2 E/F
11 Kylie	C	A	C	5	5B	D	6B	B		1 F/G
12 Liam	A*	A*	A*	9	9A*	A*	9A*	A*		
13 Marie	C	B	A	7	8B	A	8B	D		
14 Nick	B	C	D	6	8C	A	9E	B		
15 Olivia	A*	A*	A*	9	9A*	A*	9A*	A*		
16 Paula	E	D	C	4	5A	B	3B	B		
17 Quentin	C	D	C	5	4B	D	2C	B		
18 Rose	A	C	B	8	9C	A	8B	A		

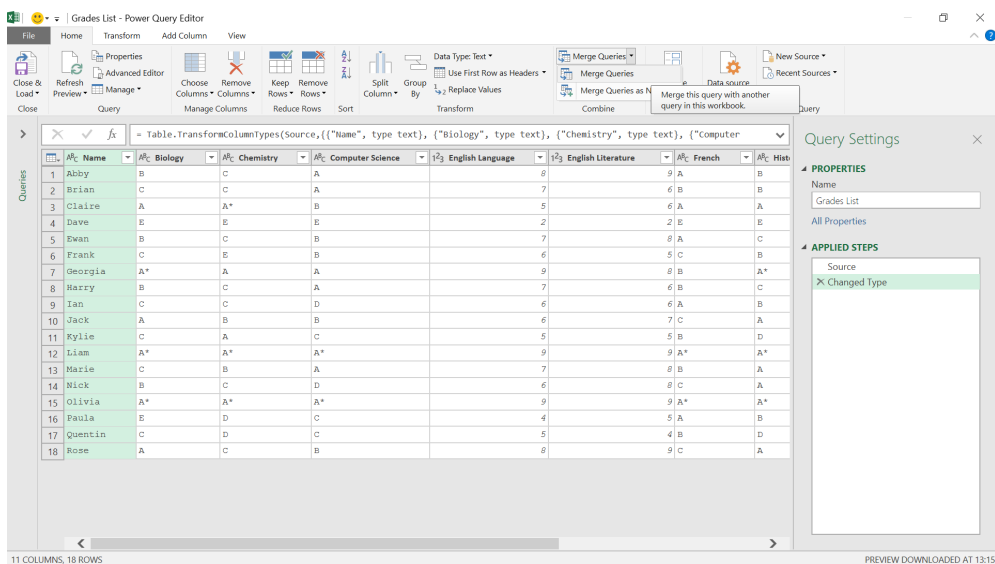
Let's create a query 'From Table' in the 'Get & Transform' section of the 'Data' tab. We'll begin by creating a query for the smaller numerical and letter equivalents table.



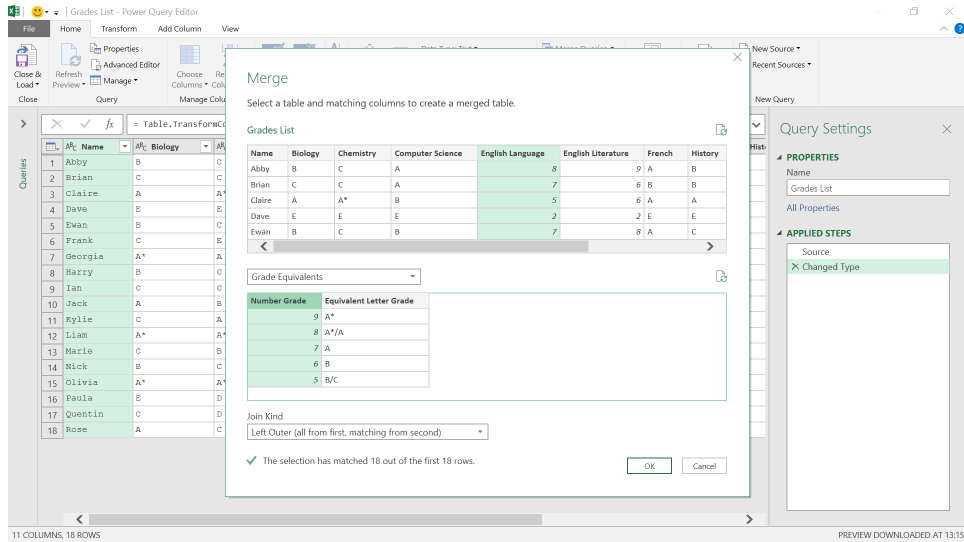
On the 'Close & Load' dropdown let's choose to 'Close & Load To...' so that we may create a 'Connection Only' query for this table, viz.



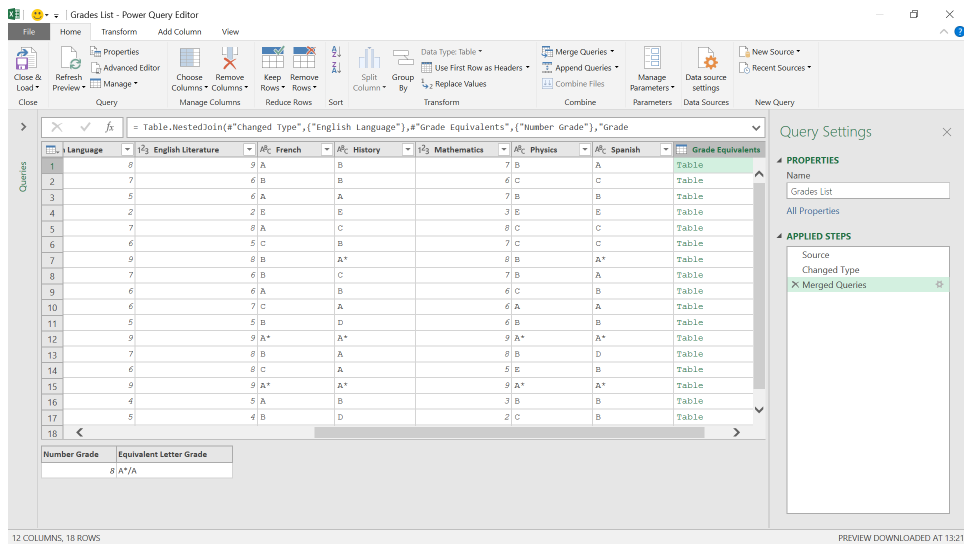
We can now create a query for our list of grades.



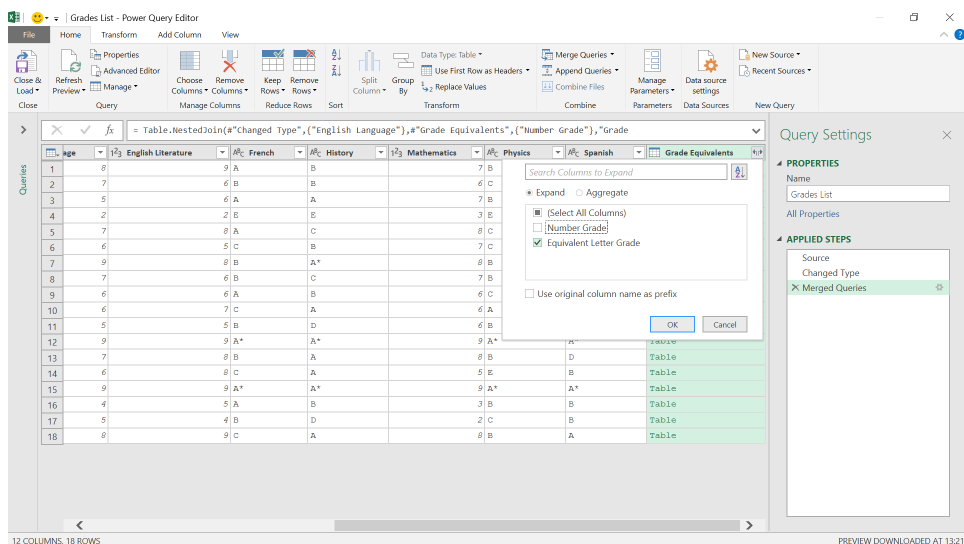
From within my 'Grades List' query, we can choose to 'Merge Queries' from the 'Combine' section of the 'Home' tab.



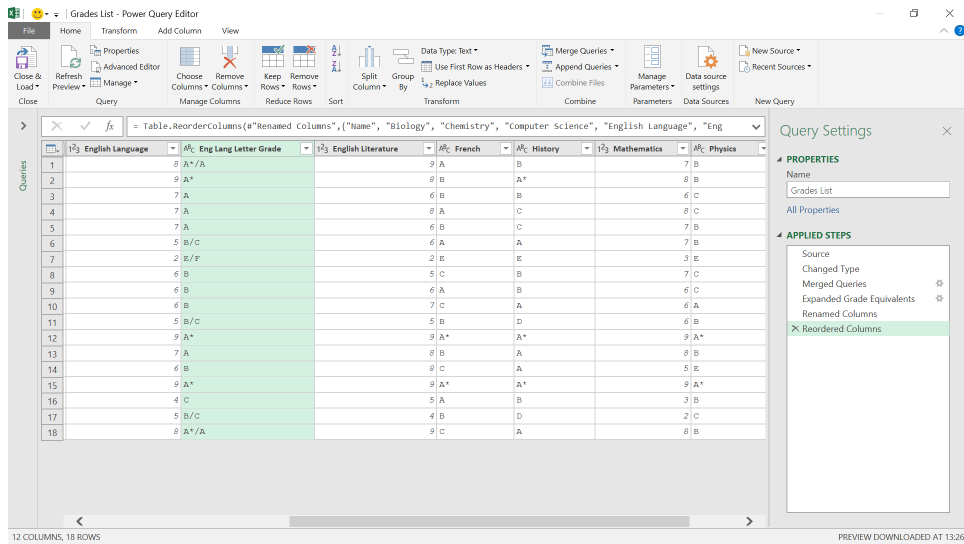
We have chosen a 'Left Outer' join as we want to keep my original 'Grades List' query and add the letter equivalents. We join the **English Language** and the **Number Grade** columns.



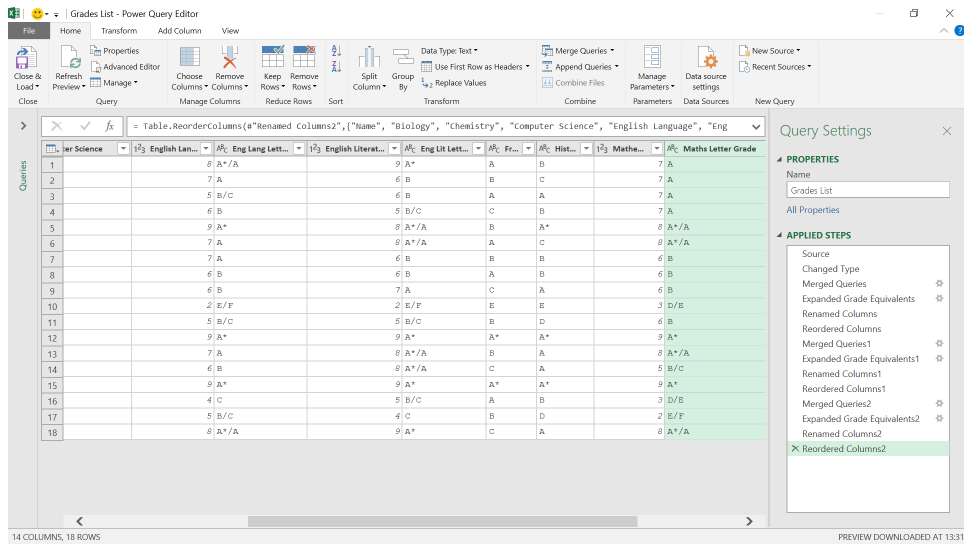
The merge has given us the letter equivalents for our **English Language** column: the first entry shows that the first grade in the table 8 corresponds to A*/A. We can expand the **Grade Equivalents** column.



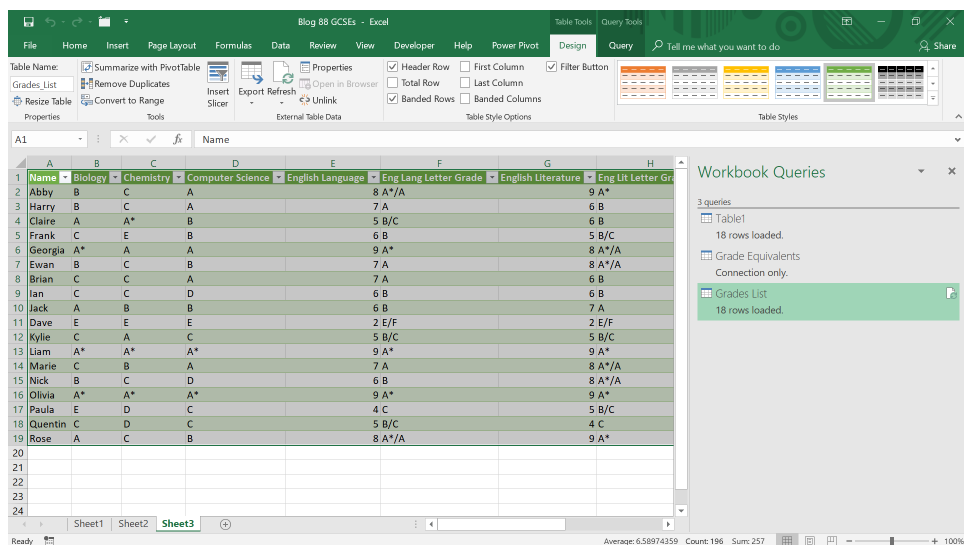
We only want the letter grade, so we expand the column and rename the letter grade to show that it corresponds to the English Language column.



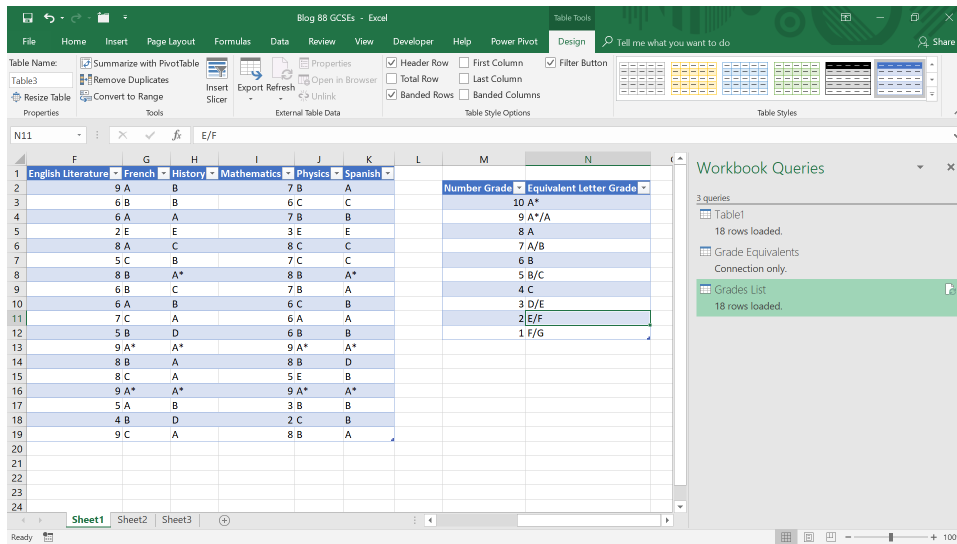
Now we have the Eng Lang Letter Grade column, we repeat the merge process for the English Language and Mathematics columns.



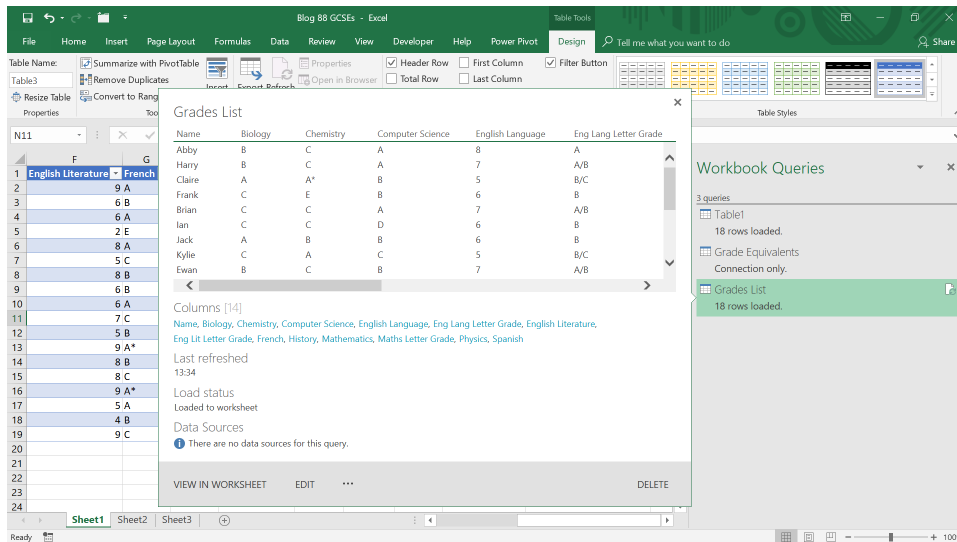
We have all the equivalent grades. Let's 'Close & Load' the query to a new worksheet.



We can also handle the issue of the examination board adding a level 10 to the grades and altering the boundaries.



If we update, our 'Grades List' query is refreshed, and we see that the grades equivalents have changed for 9, 8 and 7.



There is yet another way to approach this task, and we'll look at another method in next month's newsletter which combines replacing data with using the equivalents table.

More anon.

Power BI Updates

This month's updates see OneDrive and SharePoint integration, as well as improvements to On-object Interaction, and de-duplication rules for composite models. Perhaps the Power BI team will also create a dictionary for all these new words they are inventing too...

The full list is as follows:

Reporting

- Power BI Desktop OneDrive and SharePoint integration
- On-object Interaction updates in Preview:
 - Date Hierarchy on data flyout
 - Placeholder text for direct text editing
 - Ribbon and Funnel charts now support On-object formatting
- Power BI Home in Desktop
- De-duplication rules for composite models on Power BI Datasets and Analysis Services

Modelling

- Edit your data model in Power BI Service – updates
- Model explorer public Preview with calculation group authoring and creating relationships in the Properties pane

Data connectivity

- Snowflake (connector update)
- Planview OKR (new connector)
- BitSight Security Ratings (connector update)
- Starburst Enterprise (connector update)

Service

- One Lake data hub: 'Quick access' to your data:
 - Explorer pane: Quick access
 - Favo(u)rite items

Deployment pipelines:

- Create a customised deployment pipeline of two [2] to 10 stages
- Deployment pipeline multi-tasking
- New entry point to the 'Deployment pipelines' main page

Visualisations

- New visuals in AppSource
- valQ Plan
- Date Picker by Powerviz
- Drill Down Network PRO: show categorical relationships
- TMap 2.1
- Inforiver Enterprise brings writeback to Fabric Lakehouse and Warehouse.

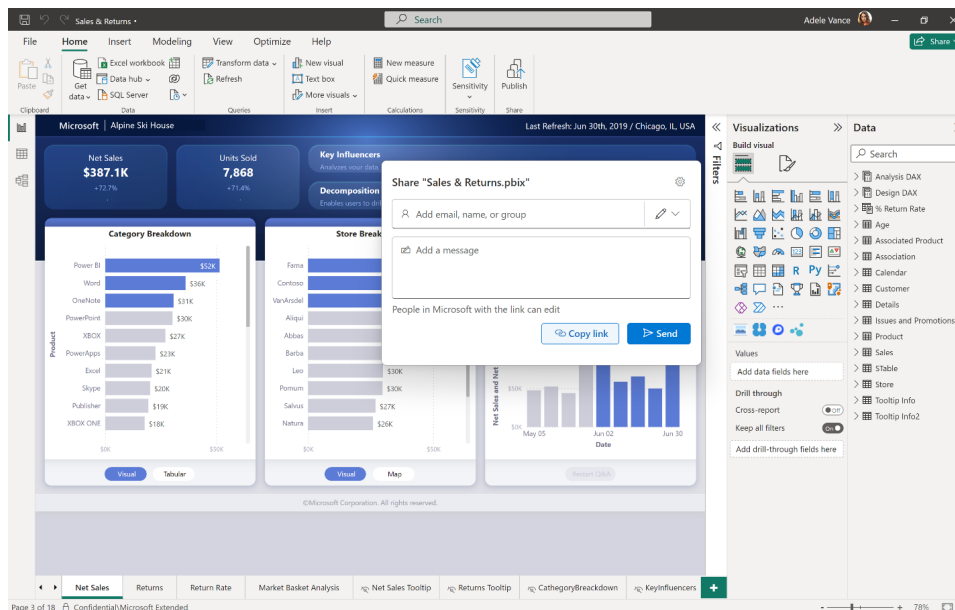
Let's look at each in turn.

Power BI Desktop OneDrive and SharePoint integration

OneDrive and SharePoint integration are both now easier in Power BI Desktop. Whilst the ability to open, save and share reports in OneDrive and SharePoint was released to Preview in May, the capabilities are now improved and switched on by default.

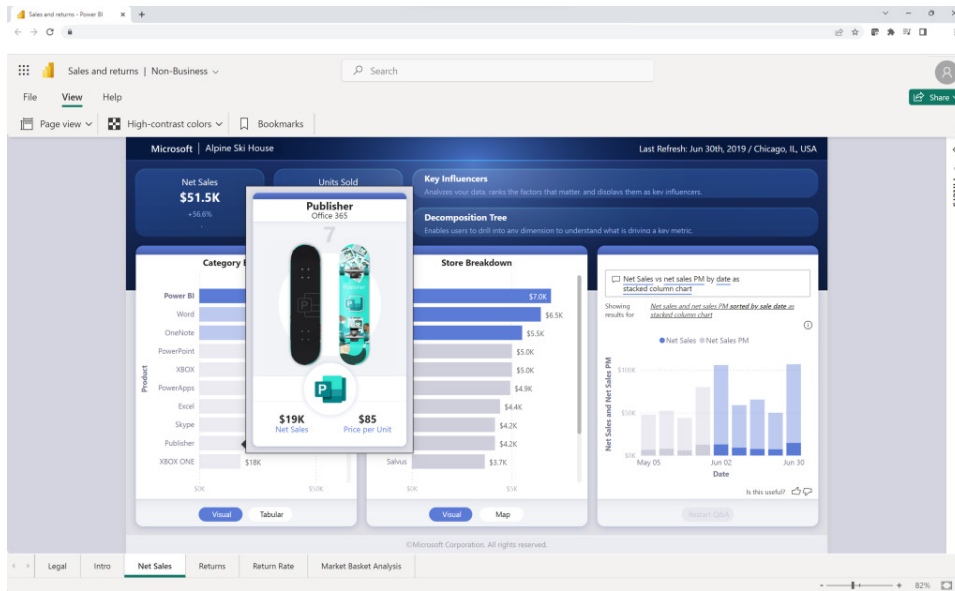
You can:

- open reports stored in OneDrive and SharePoint through the File menu
- save files directly to OneDrive and SharePoint
- share reports stored in OneDrive and SharePoint directly from Power BI Desktop.



These are important changes because many Power BI authors use OneDrive and SharePoint to collaborate on their reports before publishing through the Power BI Service. These new updates will streamline their workflow. *Ad hoc* reporting is made easier and simpler, and new users starting out will be comfortable with the familiar Office interface.

The new Power BI Desktop features are complemented by the ability to view Power BI reports stored in OneDrive and SharePoint directly in your browser. Previously, viewing a report stored in OneDrive or SharePoint required downloading both the Power BI file and Power BI Desktop. The new capability allows users to interact with their reports in seconds.



If you prefer not to have some of these features available in your organisation, they may be turned off in the Fabric admin portal.

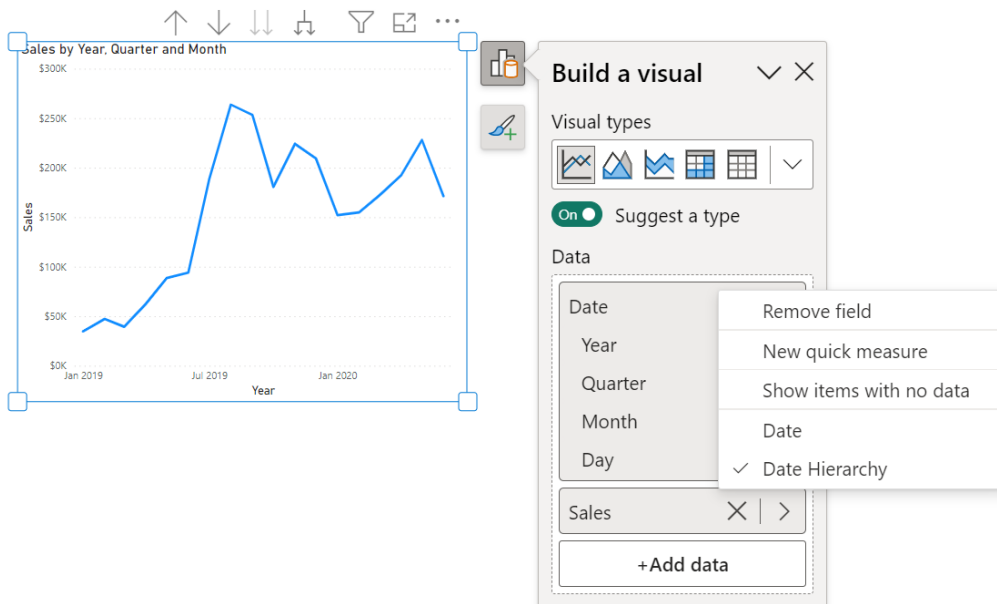
On-object Interaction updates in Preview

There are three [3] noteworthy updates this month.

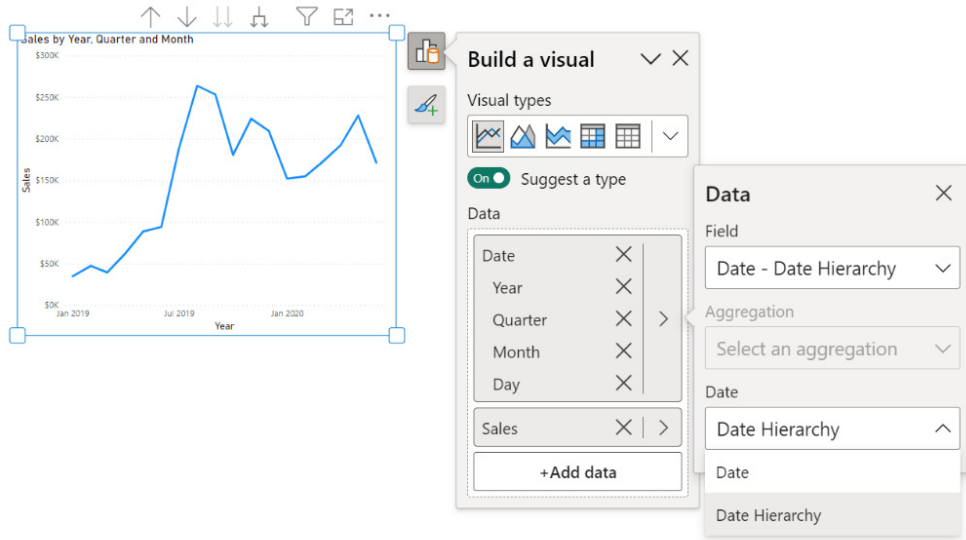
DATE HIERARCHY ON DATA FLYOUT

When working with dates, you may choose to swap from a hierarchy to the “raw” date field. This is still available when right clicking directly on the date field. For greater discoverability, Microsoft has now added the ability to switch between date hierarchy and raw date on the data flyout as well:

1. Swap using right-click



2. Swap using data flyout.



PLACEHOLDER TEXT FOR DIRECT TEXT EDITING

Previously, when turning on a text element such as a title for Tables, there was no reserved space to begin typing in characters directly on-object. You had to go to the Format pane to add a title. Now, when

turning on a text element that does not have an "auto" value, Power BI shows a placeholder for users to begin typing directly on the visual.

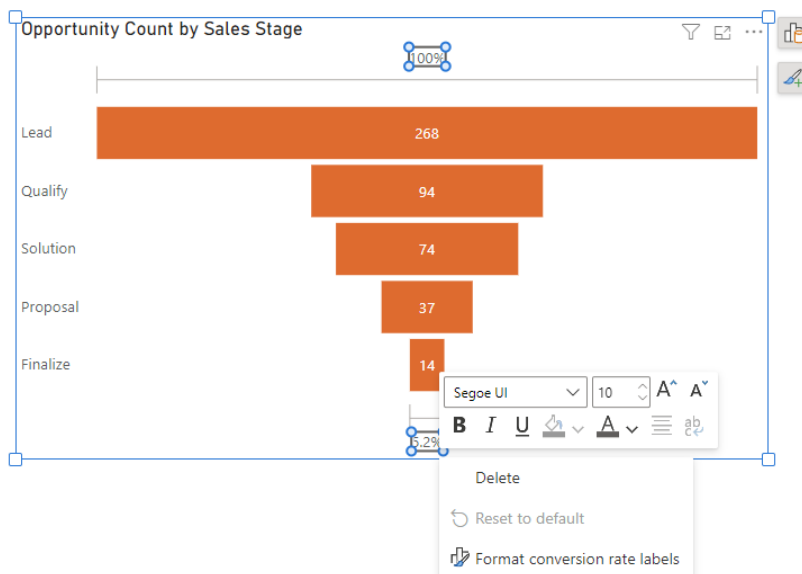
The image shows a table visual with a placeholder title "Add visual title" at the top. The table has three columns: "Category", "Sales", and "Order Quantity". The data is as follows:

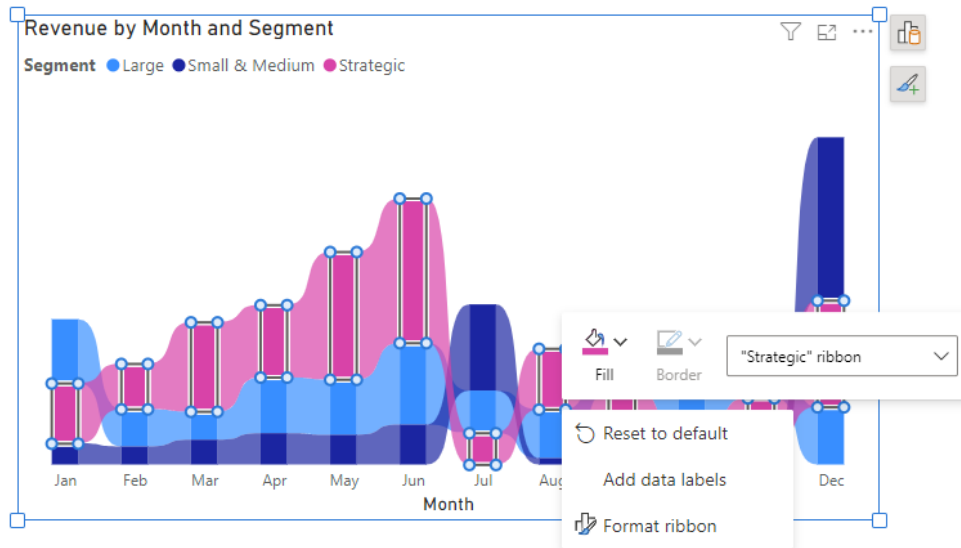
Category	Sales	Order Quantity
Clothing	\$1,597,649.96	54,539
Accessories	\$1,159,087.17	55,721
Total	\$2,756,737.13	110,260

When using direct text edit, you'll also see placeholders appear if you accidentally delete all the characters, but the text element is still ON. This again, reserves the space for you to be able to come back and a

text value using on-object. Placeholders only appear when the visual(s) are selected while editing: de-selecting the visual will remove all placeholders so you can preview what will be published.

RIBBON AND FUNNEL CHARTS NOW SUPPORT ON-OBJECT FORMATTING





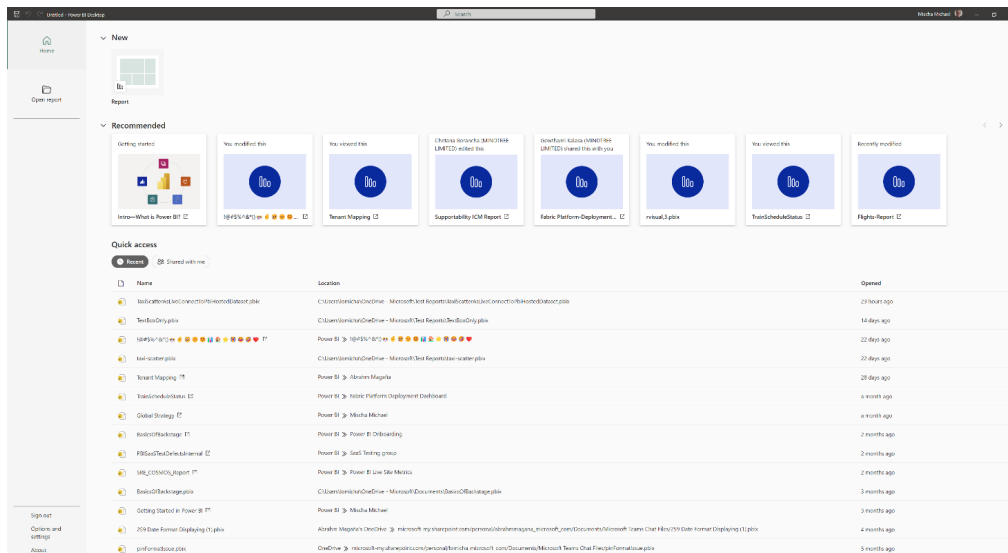
Power BI Home in Desktop

Power BI Home is a redesigned home to provide a familiar and centralised “home” for all your Power BI activities within the Power BI Desktop application. Microsoft’s stated aim is to enhance productivity and make it easier to discover and consume content.

With Power BI Home, you no longer need to navigate through multiple menus or tabs to access your files and reports. Therefore, this interface serves as a hub, similar to other Office products, where you can manage your reports, all from a single location.

This provides the capability to:

- initiate a new report directly from the new home screen
- access reports from recommendations that Power BI has curated
- locate your most recent reports through the ‘Quick Access’ lists.



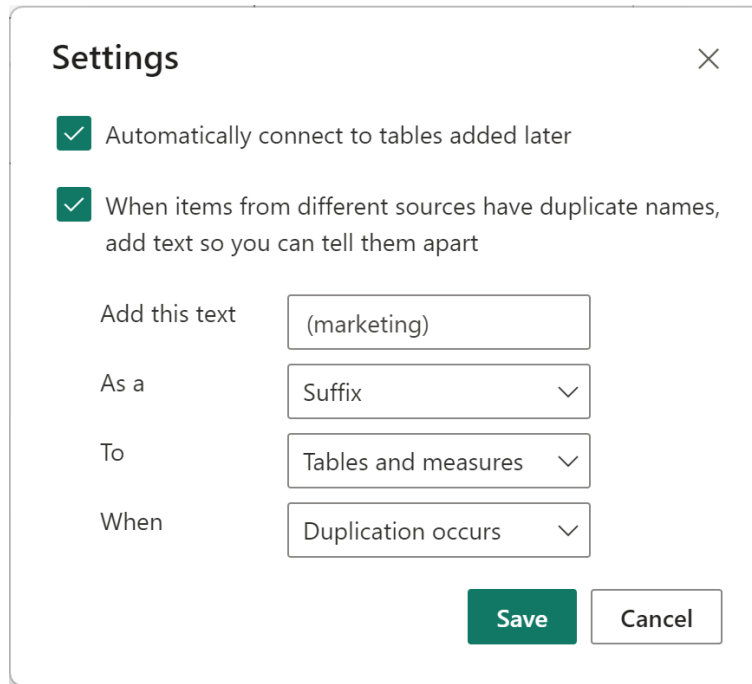
De-duplication rules for composite models on Power BI Datasets and Analysis Services

Tables and measures in your model must have unique names. If you use composite models on Power BI Datasets and Analysis Services, it’s easy to get into a situation where tables and measure names have duplicate names and are not unique. Up to this point, when that happened one of the tables or measures would be renamed for you. For example, if you created a composite model from two sources and both sources defined a table called ‘Customers’, one of the tables would be renamed ‘Customer 2’. This resulted in confusing situations as it was not clear which source the ‘Customer 2’ table came from. The same applies to measures: if you had

two sources that both contained a measure called ‘Total Sales’ one would be renamed ‘Total Sales 2’ in the composite model.

This month, Microsoft is handing (some) control back. You can now apply a name disambiguation rule to a source in a composite model when you anticipate name conflicts with tables or measures from another source. You can set up a text to be added as a prefix or suffix to table names, measure names or both. Additionally, you can choose to add that text only when a duplication occurs or if you prefer to have it added all the time.

Going back to the example above, let's say that one of the sources you are combining is about marketing and the other is about sales. You can now set up a de-duplication rule on the source connection so the 'Customer' table from the marketing source is named 'Customer (marketing)':



The 'Settings' dialog box contains the following options:

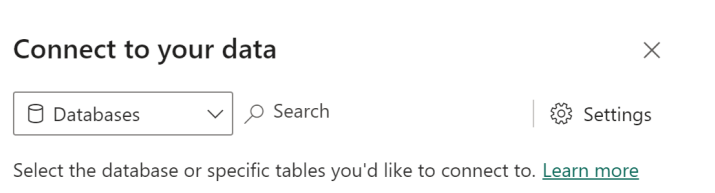
- Automatically connect to tables added later
- When items from different sources have duplicate names, add text so you can tell them apart

Below these options are four fields:

- 'Add this text' with a text input field containing '(marketing)'
- 'As a' with a dropdown menu set to 'Suffix'
- 'To' with a dropdown menu set to 'Tables and measures'
- 'When' with a dropdown menu set to 'Duplication occurs'

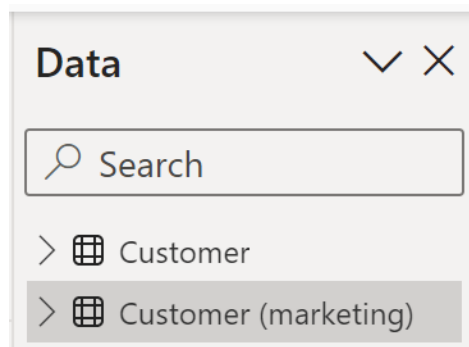
At the bottom right are 'Save' and 'Cancel' buttons.

You will find these options under Settings in the dialog that shows when you set up the composite model connection to a Power BI dataset or Analysis Services model:



The 'Connect to your data' dialog box features a search bar with a 'Databases' dropdown and a 'Search' icon. To the right is a 'Settings' gear icon. Below the search bar is the text: 'Select the database or specific tables you'd like to connect to. [Learn more](#)'

After you make the connections and set up the de-duplication rule, your Field list will show both 'Customer' and 'Customer (marketing)' according to the de-duplication rule you set up:



The 'Data' field list shows a search bar and two entries:

- > [Table Icon] Customer
- > [Table Icon] Customer (marketing)

The 'Customer (marketing)' entry is highlighted.

It should be noted that you can now:

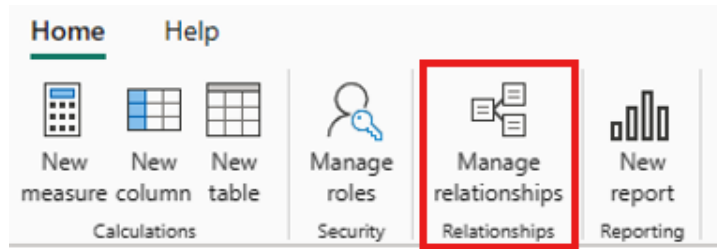
- specify whether you want the text to be added to the table or measure name as a prefix or a suffix
- apply the de-duplication rule to tables, measures or both
- choose to apply the de-duplication rule only when a name conflict occurs or apply it all the time. The default is to apply the rule only when duplication occurs. In the example above, any table or measure from the marketing source that does not have a duplicate in the sales source will not receive a name change.

Finally, it would be remiss if we didn't mention that if you do not specify a de-duplication rule or the de-duplication rules you specified do not resolve the name conflict, the standard deduplication actions will still be applied.

Edit your data model in Power BI Service – updates

The new data model editing feature in Power BI Service was released to Preview back in April. Since then, Microsoft has been busy reacting to customer feedback and revising the experience. Below are the improvements made to the 'Manage relationships' dialog.

This update makes it easier to view and edit all the relationships within your data model in the Service. In the Home tab, simply select the 'Manage relationships' button.



This will open the revamped 'Manage relationships' dialog, which provides a comprehensive view of all your relationships, along with their key properties, in one convenient location. From here, you can then choose to create new relationships or edit an existing relationship.

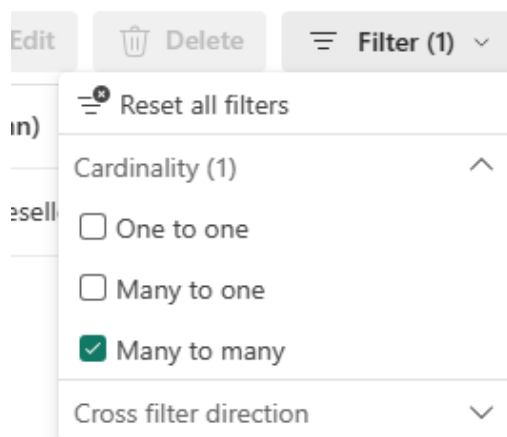
Manage relationships

+ New relationship Edit Delete Filter

From: table (column)	Relationship	To: table (column)	Status
<input checked="" type="checkbox"/> Sales (SalesTerritoryKey)	* —> 1	Sales Territory (SalesTerritoryK...	Active ...
<input type="checkbox"/> Sales (ProductKey)	* —> 1	Product (ProductKey)	Active ...
<input type="checkbox"/> Sales (SalesOrderLineKey)	1 <—> 1	Sales Order (SalesOrderLineKey)	Active ...
<input type="checkbox"/> Sales (OrderDateKey)	* —> 1	Date (DateKey)	Active ...
<input type="checkbox"/> Sales (DueDateKey)	* —> 1	Date (DateKey)	Inactive ...
<input type="checkbox"/> Sales (ShipDateKey)	* —> 1	Date (DateKey)	Inactive ...
<input type="checkbox"/> Sales (ResellerKey)	* —> *	Reseller (ResellerKey)	Active ...
<input type="checkbox"/> Sales (ProductKey)	* —> 1	Customer (CustomerKey)	Active ...

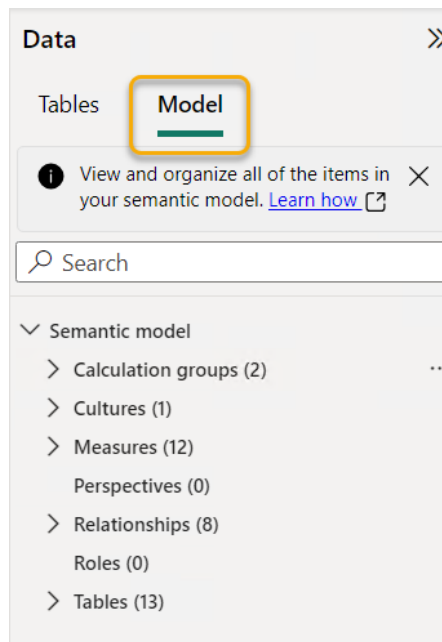
Close

Additionally, you have the option to filter and focus on specific relationships in your model based upon cardinality and cross filter direction.



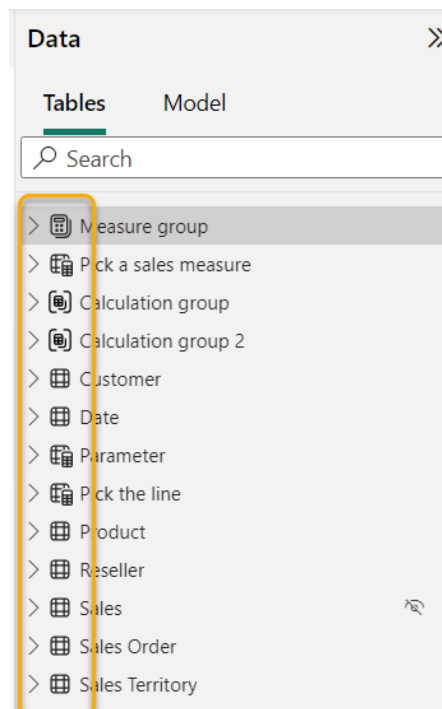
Model explorer public Preview with calculation group authoring and creating relationships in the Properties pane

The Model explorer is now available for public Preview in the Model view. You can see all your dataset semantic modelling objects in one place and easily navigate between them.



An additional Properties pane now shows for the dataset semantic model.

Furthermore, new icons are also showing on the Data pane: additional user interface (UI) changes will continue through December for the Data and Properties panes.

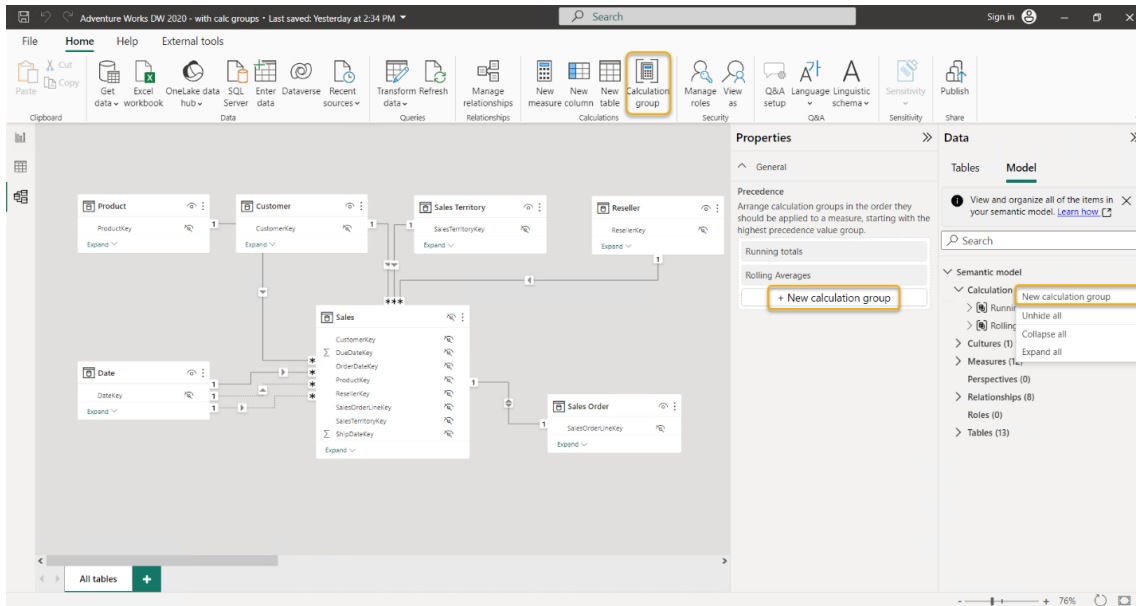


You should also note that measure groups will always show at the top, followed by calculation groups, and then finally the other tables in the model.

From the Model explorer, not only can you now see the calculation groups with their calculation items, but you can also create and edit them in Desktop. Calculation groups is a powerful feature that allows you to apply dynamic calculations to your existing measures. For example, you

can create a calculation group that applies time intelligence functions as calculation items, such as year-to-date, quarter-to-date or month-to-date, to any measure in your model. To author these in Desktop, simply go to the Model view and click on the new 'Model' tab of the Data pane. If you click on the Calculation groups node, you have three options to create a new one:

1. Ribbon button
2. Context menu
3. Properties pane.



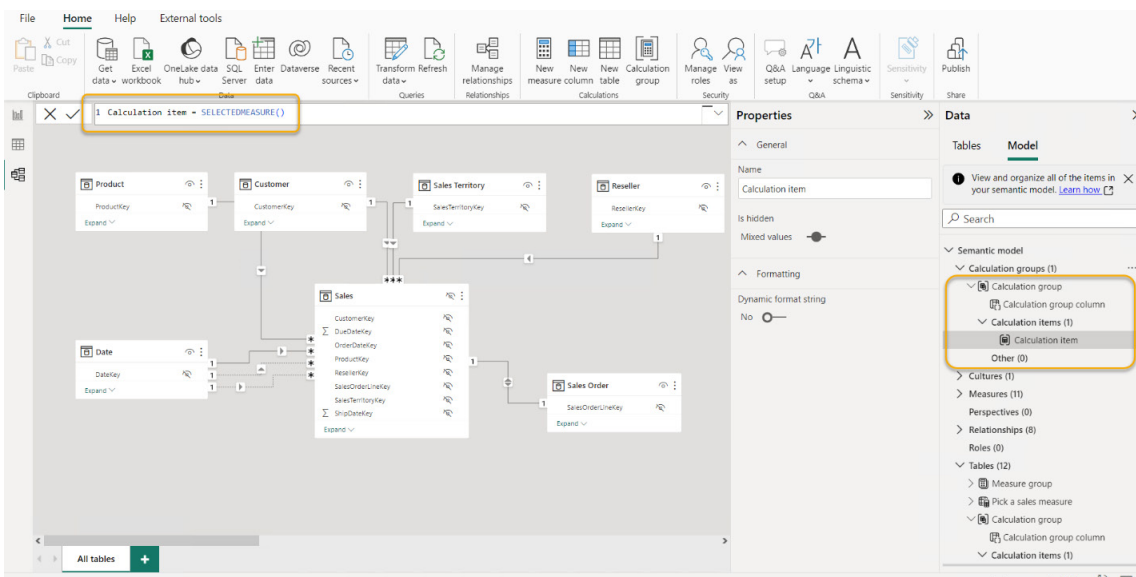
After clicking 'New calculation group', if you do not have the model property discourage implicit measures turned off, you will be told about this setting to create the calculation group, viz.

This change will discourage implicit measures ✕

When you add a calculation group, implicit measures will be discouraged in this model. This means you'll need to create explicit measures to aggregate data columns.

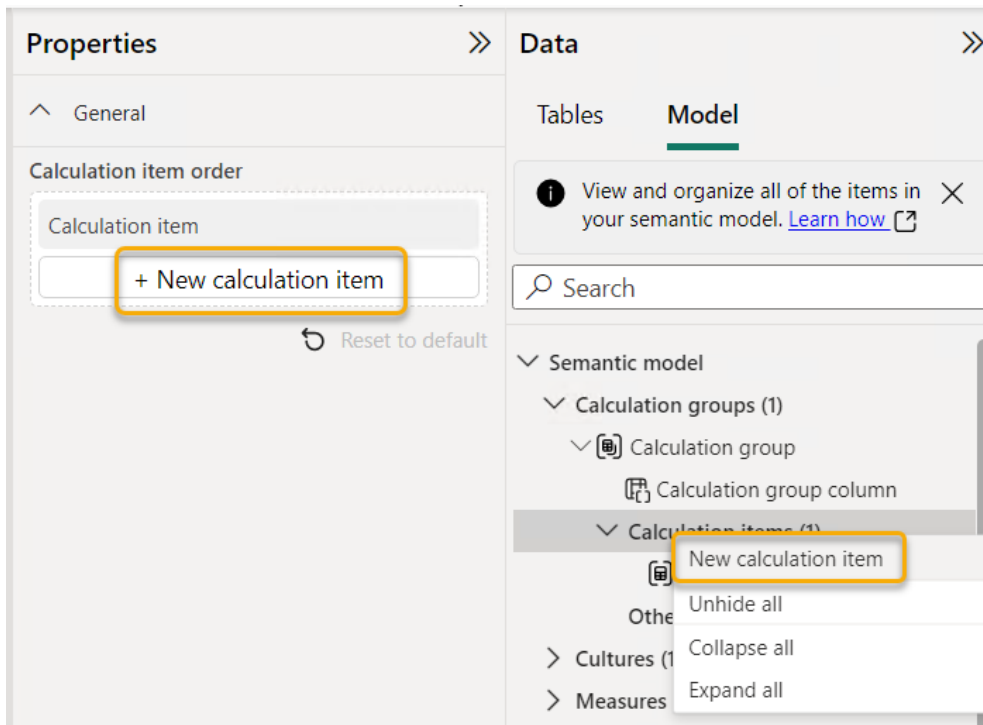


Once turned on, the calculation group is created, and you are landed directly in the first calculation item to define the **DAX** expression you want to apply to your existing measures.

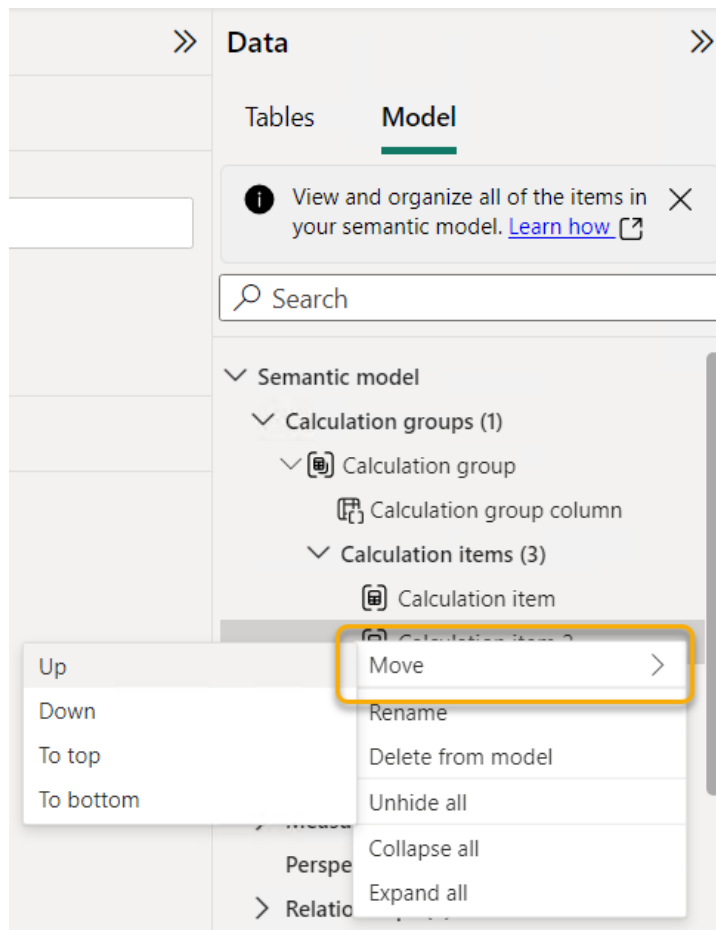


You can alter this **DAX** expression in the **DAX** Formula bar. Optionally, you may also add a dynamic format string to the calculation item from the Properties pane.

New calculation items can be created from the context menu (right-click) of the 'Calculation items' node or the calculation group, and in the Properties pane of the 'Calculation items' node.

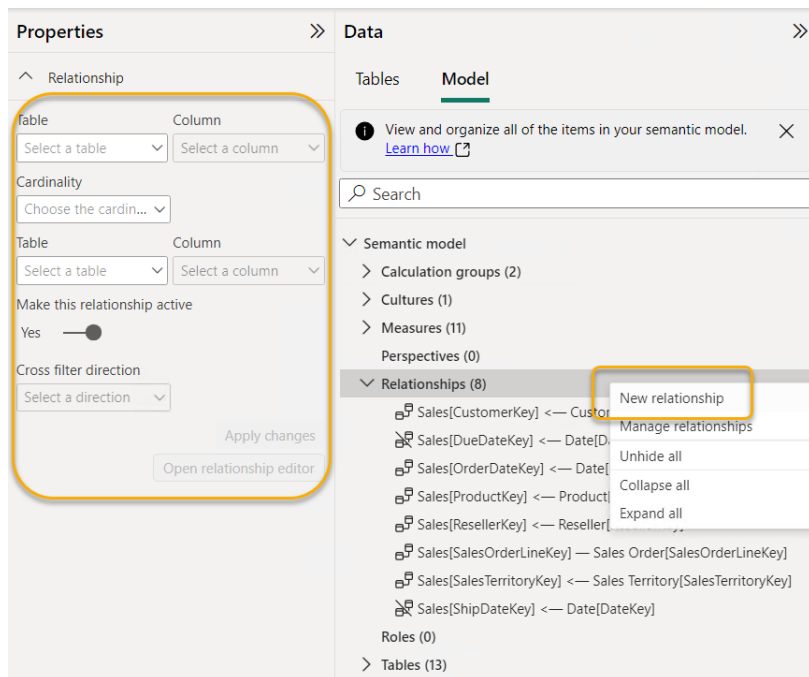


Controlling the order of the calculation items may be defined in the 'Calculation items' node Properties pane or using the context menu (right-click) of the calculation items themselves.

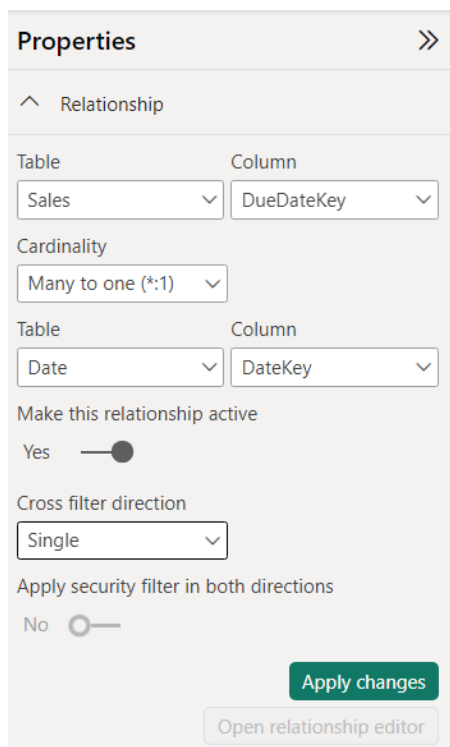


Another option also now available from the Model explorer is the ability to create relationships in the Properties pane. Just like 'Edit relationships' in the Properties pane, this allows you to simply change table and columns

without previewing data or validating until you click 'Apply change'. This experience is now available to create relationships too: simply choose a new relationship from the Model explorer 'Relationships' node.



This will show an empty relationship Properties pane to fill out and then apply when ready.



The Model explorer does show you other modelling features that do not yet have authoring paths in Power BI Desktop. These features are perspectives and cultures. These will still need to be authored outside of Desktop throughout XMLA write external tools or through XMLA directly.

These features are available in the latest version of Power BI Desktop. To use them, switch on the Model explorer public Preview switch: go to **File -> Options and settings -> Options -> Preview features** in the **GLOBAL** section.

Options



GLOBAL

- Data Load
- Power Query Editor
- DirectQuery
- R scripting
- Python scripting
- Security
- Privacy
- Regional Settings
- Updates
- Usage Data
- Diagnostics
- Preview features**
- Save and Recover
- Report settings

CURRENT FILE

- Data Load
- Regional Settings
- Privacy
- Auto recovery
- Published dataset settings
- Query reduction
- Report settings

Preview features

The following features are available for you to try in this release. Preview features might change or be removed in future releases.

- Shape map visual [Learn more](#)
- Spanish language support for Q&A [Learn more](#)
- Q&A for live connected Analysis Services databases [Learn more](#)
- Connect to external datasets shared with me [Learn more](#) | [Share feedback](#)
- Modern visual tooltips [Learn more](#) | [Share feedback](#)
- Sparklines [Learn more](#)
- Metrics visual [Learn more](#)
- Quick measure suggestions [Learn more](#) | [Share feedback](#)
- Field parameters [Learn more](#)
- Enhanced row-level security editor [Learn more](#)
- On-object interaction [Learn more](#) | [Share feedback](#)
- Power BI Home in Desktop [Learn more](#) | [Share feedback](#)
- Enable setting sensitivity label on exported PDF [Learn more](#)
- Dynamic format string for measures [Learn more](#)
- Save to OneDrive and SharePoint [Learn more](#)
 - Share to OneDrive and SharePoint [Learn more](#)
- Power BI Project (.pbip) save option [Learn more](#)
- New card visual [Learn more](#)
- Button slicer visual
- New Power Query experience
- Model explorer and Calculation group authoring [Learn more](#)

OK **Cancel**

Snowflake (connector update)

The Snowflake connector has been updated to support better implementation of 'LIMIT 1' queries, resulting in performance improvements.

Planview OKR (new connector)

Planview Objectives and Key Results (OKRs) are an outcome-driven framework adopted by organisations who want to define key organisational goals and track progress toward achieving them. Defining OKRs enables organisations to answer questions such as "Where do we want to go?" (objectives) and "How will we measure our efforts to get there?" (key results).

OKRs may be created at different levels of an organisational structure, such as enterprise, portfolio, program or team, and are connected using parent / child relationships. Linking organisational and team goals in a hierarchical way aligns work delivery to company strategy and provides a single line of sight into value delivered by the organisation.

BitSight Security Ratings (connector update)

The BitSight Security Ratings connector has been updated with minor bug fixes.

Starburst Enterprise (connector update)

The Starburst Enterprise connector has been updated. Here is what has happened:

- support for Native Queries (Custom SQL) has been added. Users can write their own SQL queries against Starburst clusters in both Import and Direct Query modes
- Starburst ODBC Driver version 2.0.15.1021 or later is now required
- introduced 'Lazy Evaluation of Metadata' for all connection modes, which improves performance when initially displaying the catalog tree
- added optional Connection string (non-credential properties) in the Advanced Section to allow using other ODBC connection parameters
- changed data source display from Starburst Enterprise to a value based on the connection details (**{ "Host": "sep.example.com", "Port": "443" }**) to allow distinction between multiple Starburst Galaxy or clusters connected as separate data sources
- fixed an issue with port 8443 for OAuth 2.0 authentication
- fixed query folding with timestamp columns.

One Lake data hub: 'Quick access' to your data

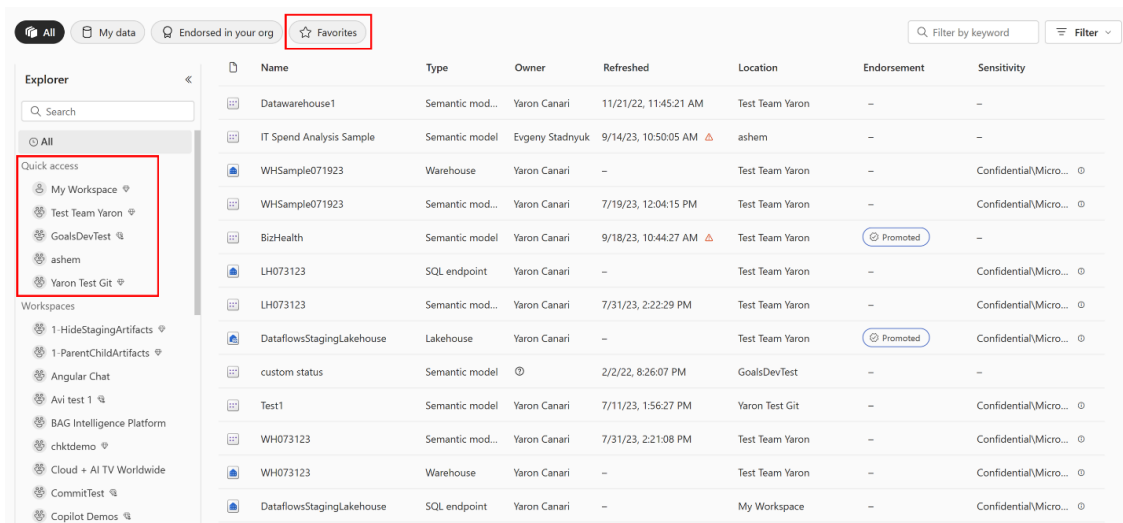
OneLake data hub is the central location for users to find and reuse existing organisational data. It allows users to browse through their data and discover insights that can help them make better decisions. This month sees some new features that added to enhance the OneLake data hub discovery experience.

EXPLORER PANE: QUICK ACCESS

The Explorer pane enables users to navigate through the workspaces hierarchy and scope the data items to a specific workspace. With the recent enhancements, Microsoft has added a 'Quick access' section at the top of the Explorer pane. The 'Quick access' section contains the active workspace, pinned workspaces and recently used workspaces.

FAVO(U)RITE ITEMS

Another new feature is the 'favorite items' (*sic*). You may now filter to view items that were marked previously as favourite in Power BI. Users can also 'favorite' (*extremely sic!*) items from within the data hub experience and they will also show up across all the experiences including Home and Browse. This feature helps you to keep track of the data items that are most important.

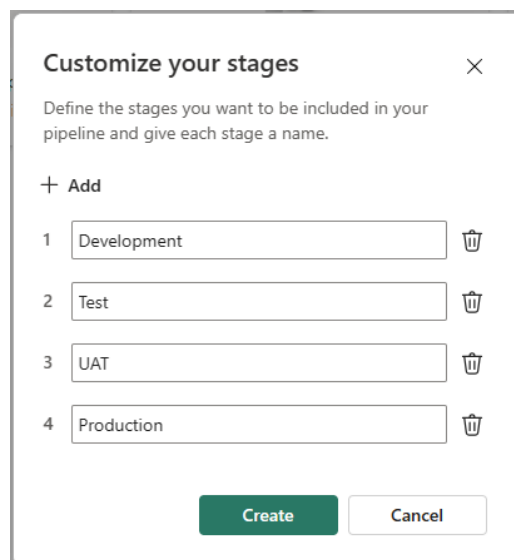


Deployment pipelines

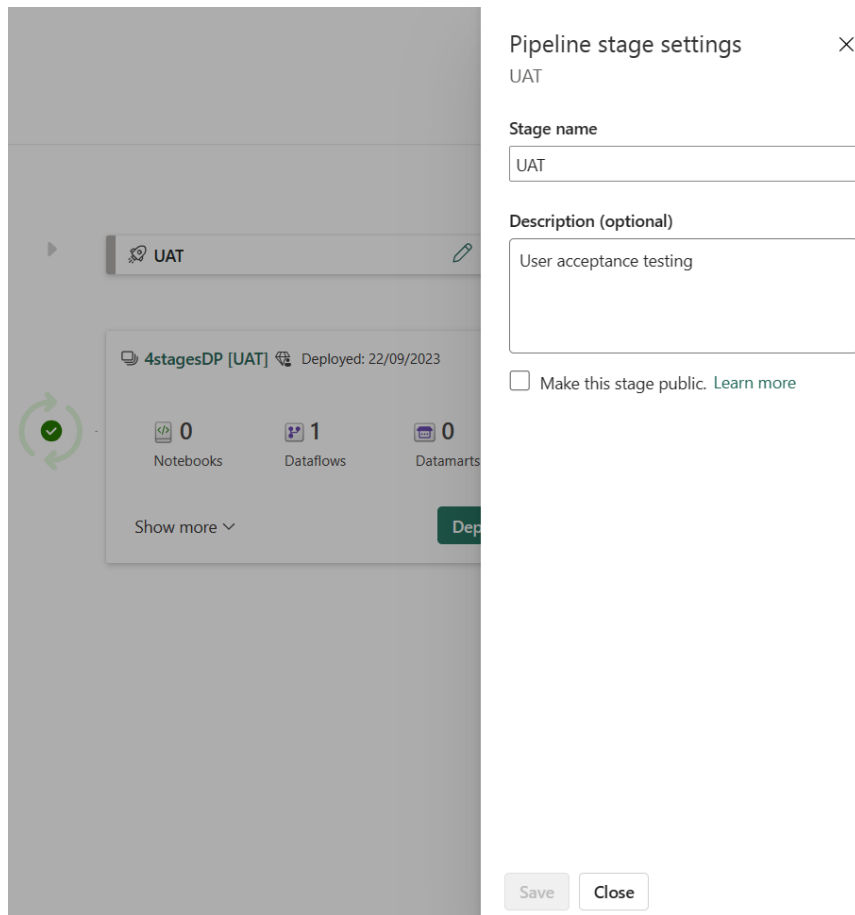
A customised pipeline has finally come to Power BI.

CREATE A CUSTOMISED DEPLOYMENT PIPELINE OF TWO [2] TO 10 STAGES

After a long wait it is finally here: a customised deployment pipeline of two [2] to 10 stages. Starting back in October, you are now able to decide how many stages your new deployment pipeline will have before it is created, within a new step added to the creation process:



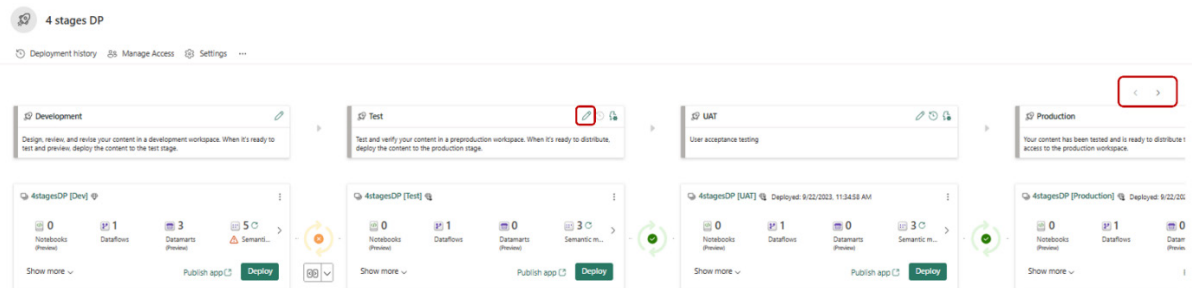
As shown above, you can set your own name to each stage, although these names may be changed any time later through the new stage settings pane (entry point is located next to the stage name):



Be advised though: the number of stages cannot be changed after the deployment pipeline's creation. With that, customisation of existing deployment pipelines is not possible and to transition an existing deployment pipeline to a two [2] or 4+ stages pipeline, a new one would

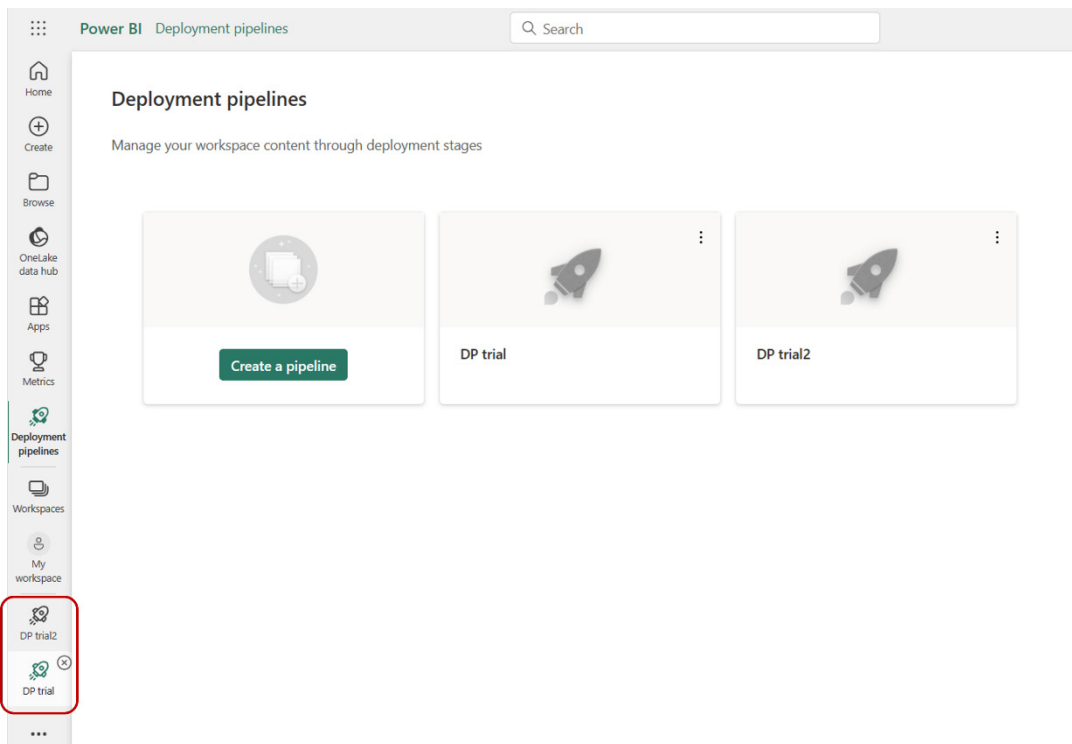
have to be created manually, so then the workspaces are unassigned from the existing deployment pipeline and re-assigned to the new one.

To navigate between the stages, use the arrows on the top-right corner:



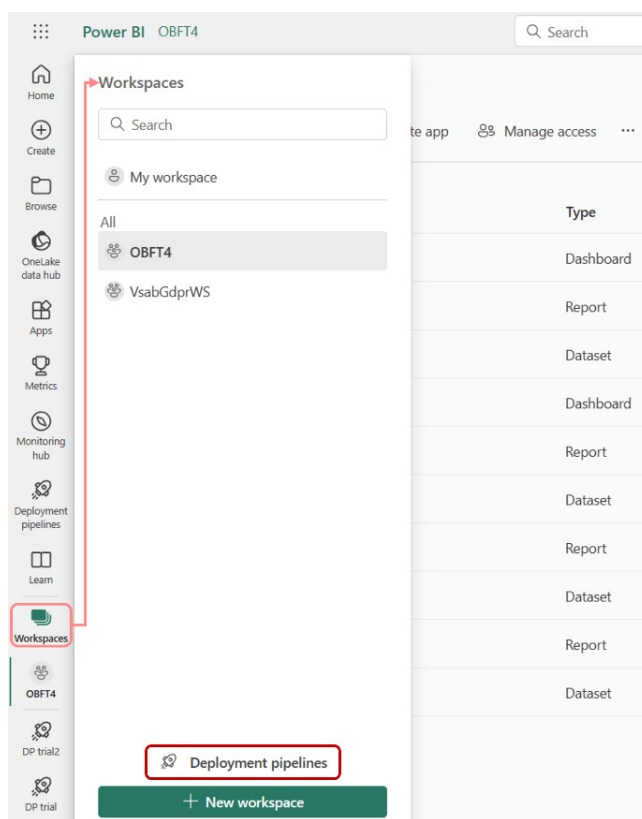
DEPLOYMENT PIPELINE MULTI-TASKING

'Deployment pipelines' joined the platform's new feature of multi-tasking, which opens a new tab on the left navigation bar whenever you open an item or a deployment pipeline. That means no more ending up with 10 explorer tabs of the same DP / workspaces anymore!



NEW ENTRY POINT TO THE 'DEPLOYMENT PIPELINES' MAIN PAGE

A new entry point to Deployment pipelines was also added. It is located at the bottom of the workspaces list and leads to the 'Deployment pipelines' main page:



New visuals in AppSource

Just the one this month: Step Filter.

valQ Plan

valQ Plan is a newer and re-architected Power BI-certified edition of valQ with numerous transformational updates and a simplified no-code experience.

valQ helps you to build complex business plans, what-if simulations, create

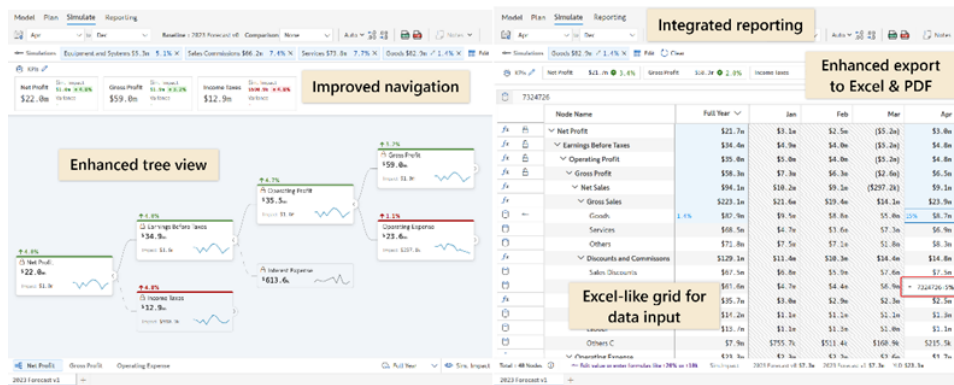
valQ Plan

- **niche use cases** such as Scenario Planning, Strategic Planning or Value Drive Trees and What-If Simulations
- **typical planning use cases** such as financial planning and analysis (FP&A), three-statement financial modelling and Supply Chain Demand Planning etc.
- **Excel-like grid input experience** for your rolling forecasts with no-code auto forecasting algorithms and goal seek
- **integrated reporting:** one-click IBCS templates for variance reporting, presentation mode with live charts and tables, enhanced export to Excel and PDF
- **intuitive modelling:** significantly enhanced formula editor experience, dynamic templating and more
- **time intelligence:** supports more than 12 periods allowing you to create 90-day rolling forecasts, 52-week forecasts and even 10-year strategic plans.

valQ Plan may be purchased directly from Microsoft AppSource.

and compare budgets, forecasts and scenarios, all within Microsoft Power BI. It can be used as your standalone and integrated business planning software tool or used in conjunction with your existing planning platforms.

Feature highlights include:



No-code user experience Reduced complexity Integrated reporting Time intelligence

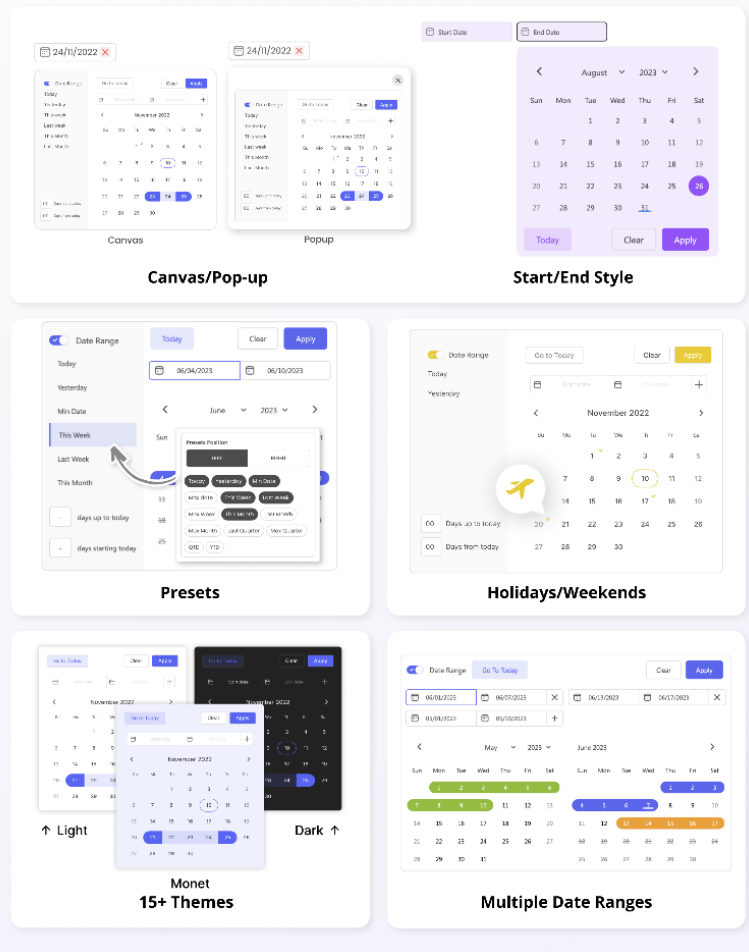
Niche use cases	Value driver modeling	Scenario planning	Strategic planning
Common use cases	Financial planning & analysis	3-statement financial modeling	Re-forecasting & variance analysis
	Supply chain optimization	Inventory management	Weekly/monthly sales reporting
			FP&A Operations Sales & marketing Mining
Planning & forecasting	Simulations & scenarios	Reporting	Modeling
<ul style="list-style-type: none"> ✓ Advanced time series forecasting methods ✓ Rolling forecasts ✓ Goal seek ✓ Plan allocation with quick actions 	<ul style="list-style-type: none"> ✓ Enhanced tree & table views ✓ Scenario creation flow ✓ Split screen – Tree & KPI detail ✓ Editing nodes in runtime 	<ul style="list-style-type: none"> ✓ All IBCS table templates ✓ Presentation mode ✓ Simplified filtering ✓ Enhanced export to Excel & PDF 	<ul style="list-style-type: none"> ✓ New model grid with better formula editor and bulk editor ✓ New formulas ✓ Dynamic templating ✓ Data preview & error validations

Date Picker by Powerviz

This is a new date slicer for Power BI. The 'Date Picker' visual comes with a modern calendar view along with Presets, Pop-up mode, Default Selection, Themes, and more.

- **Display Mode:** choose between Pop-up and Canvas modes
- **Presets:** many commonly used presets like Today, Last Week, YTD, MTD or else create your own preset using field
- **Default Selection:** enforce the selection of the desired period when a user opens the report without any custom DAX
- **Filter Type:** choose between Range and Start / End types
- **Multiple Date Ranges:** flexibility to select multiple date ranges
- **Themes:** 15+ pre-built themes with full customisation
- **Holidays and Weekends:** multiple formatting options
- **Import / Export JSON:** build templates and share your designs.

MOST ADVANCED DATE SLICER FOR POWER BI
WITH CALENDAR VIEW, DEFAULT SELECTION, PRESETS, THEMES AND FULL CUSTOMIZATION



Drill Down Network PRO: show categorical relationships

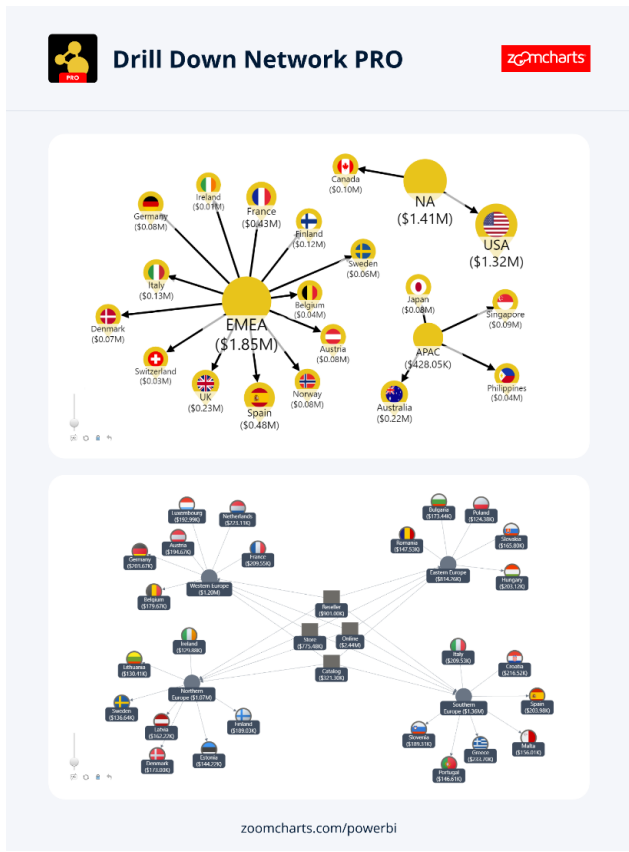
Drill Down Network PRO by Zoom Charts is designed for visualising categorical data and automatically detecting relationships based upon category structure. You may use your existing category-based data with few adjustments, and quickly create an interactive chart that makes relationships between each data category easy to read.

Main features include:

- **Cross-chart filtering:** eliminate slicers by selecting data points directly on the charts
- **Category-based customisation:** apply image to node and choose its type, shape, colour and font
- **Link styling:** configure 'from' and 'to' decorations and show link value
- **Display up to nine [9] data categories**
- **Touch-input device friendly:** explore data on any device.

Popular use cases include:

- **Accounting and Finance:** show cost attribution
- **Human Resources:** analyse salary data by department
- **Production:** map production volumes by product or factory
- **Sales & Marketing:** visualise marketing campaigns.



ZoomCharts Drill Down Visuals are known for interactive drilldowns, smooth animations and various customisation options. They support interactions, selections, custom and native ToolTips, filtering, bookmarks and context menus.

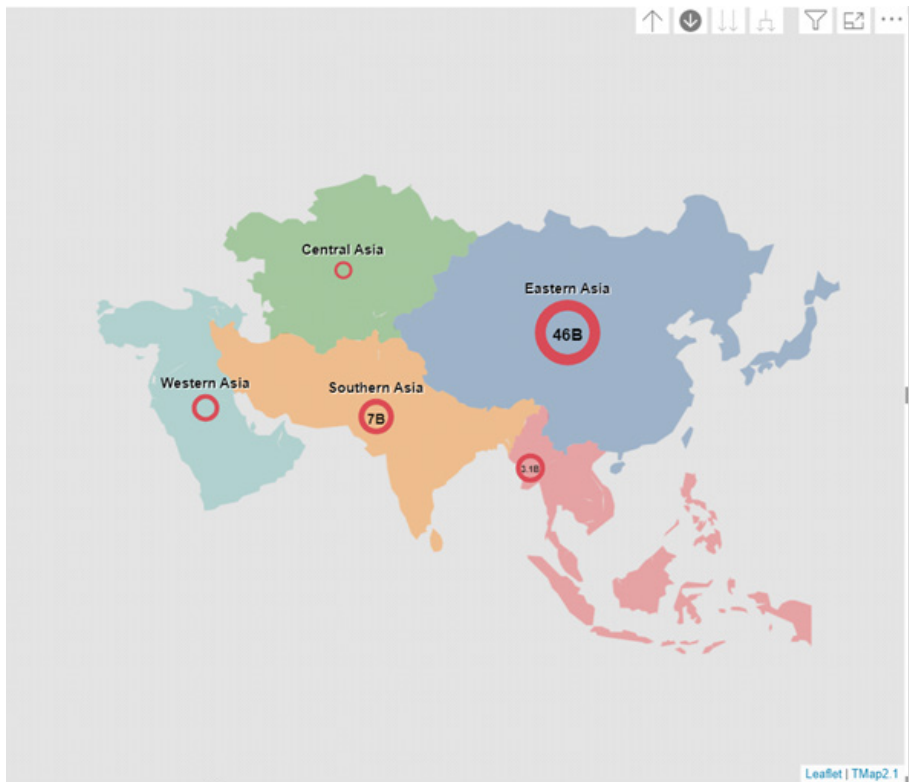
TMap 2.1

The newly released TMap 2.1 has added new features for drilling down Donut Map, Choropleth Map, Bar Chart Map, Pie Chart Map and Stacked Bar Chart Map by geo-region's names.

Drilling down maps by geo-region's names doesn't need to prepare a hierarchy of pre-built polygon layers and will save data professional's

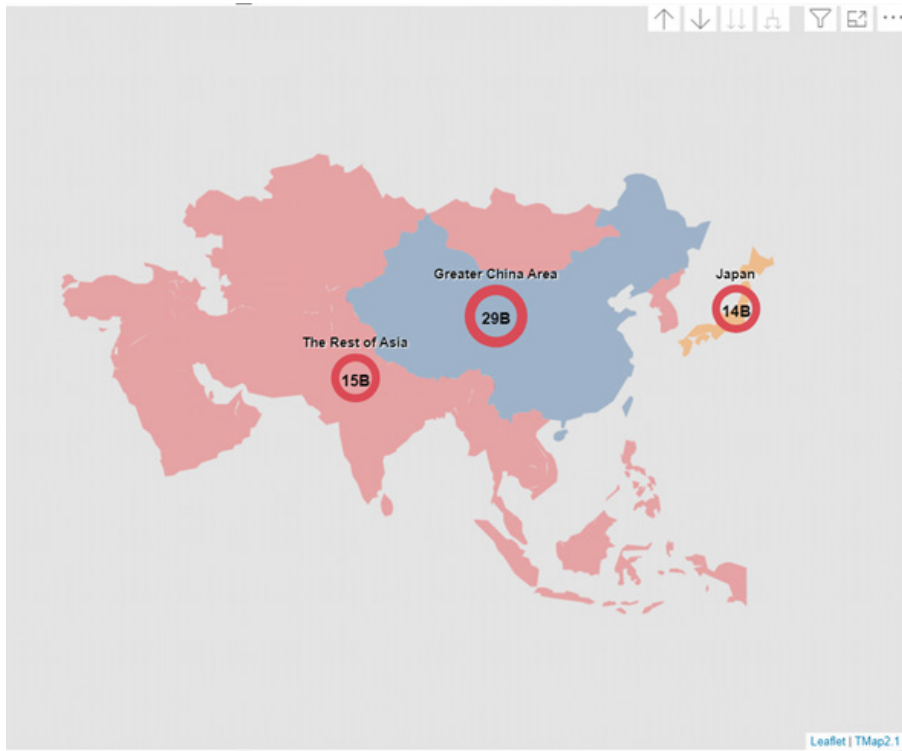
time and cost to extract deeper insights from the data on different levels of geographic regions.

For example, here is a drilldown donut map for company A's sale in Asia by subregions:



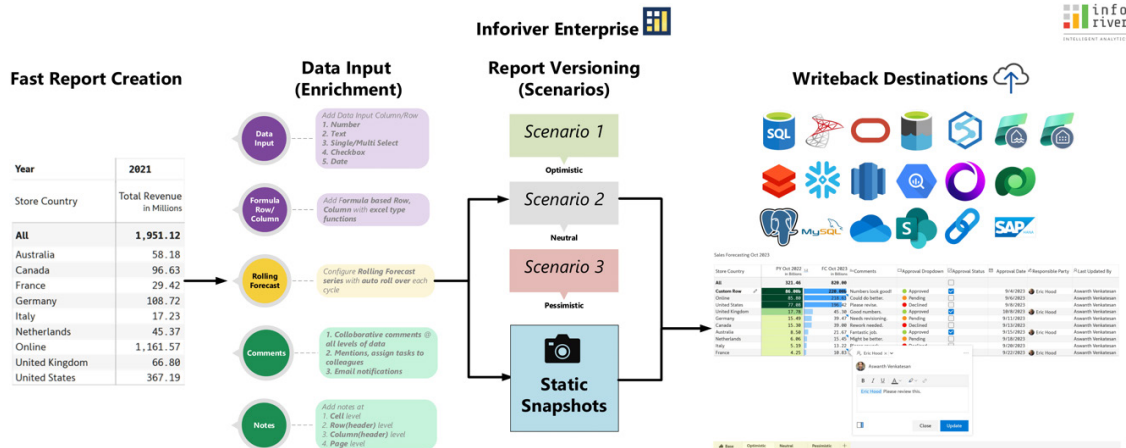
Realigning geo-regions will become much easier because you only need to change geo-region's name in the lowest level and the polygon layer for upper level will be automatically generated during the drilling down process.

As another illustration, here is a drilldown donut map for company A's sale in Asia by realigned subregions:



Inforiver Enterprise brings writeback to Fabric Lakehouse and Warehouse

Inforiver now supports three new data writeback destinations from Power BI: Microsoft Fabric Lakehouse, Fabric Datawarehouse, and Dataverse, along with other popular alternatives.



The latest Inforiver Enterprise release supports writeback to Fabric for these use cases:

- create business projections by entering / editing directly in Power BI reports
- create and manage rolling forecasts by blending actual and forecast data series
- create dynamic simulations at the enterprise, business unit or department level and roll up projections (to the top) or distribute projections (to granular dimensions) based upon chosen allocation / distribution methods
- create multiple scenarios, each containing its own set of simulations
- write back data, including comments and threaded conversations
- create periodic report snapshots
- support multiple data input types and writeback from multiple Power BI users in reading mode.

Inforiver supports writeback in the following deployment configurations:

- Azure SAAS service managed by Inforiver *or*
- managed by customers in their private tenant.

More next month!

New Features for Excel

This month sees Microsoft share that Python in Excel is now available to all Enterprise, Business, Education, Family and Personal users running Beta Channel on Windows, starting with build 16.0.16818.20000 – and then to the other platforms at a later date. Also, the Python Editor from Excel Labs and Copilot support for Python in Excel are formally announced too. I feel almost AutoComplete...

You can check out the full list here:

Excel for Windows

- Python in Excel (Insiders Beta)
- Python Editor from Excel Labs (Insiders Beta)
- Copilot Support for Python in Excel (available in Preview later this year)
- Data Validation AutoComplete

Excel for Mac

- Data Validation AutoComplete (Insiders).

Let's get started.

Python in Excel (Insiders Beta)

How many months running are we going to write about this in this newsletter!?



They keep plugging it, we'll keep showing it...

Python in Excel makes it possible to natively combine 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. This feature is now available to all Enterprise, Education, Family and Personal users running the Beta channel on Windows.

The screenshot shows the Excel interface with a Python script in the formula bar: `#Announcing Python in Excel! DataFrame=x1("A1:B10", headers=True) DataFrame.groupby('Category').agg('mean')`. The worksheet displays the following data:

Category	Price
Components	\$20
Bikes	\$17
Accessories	\$9
Bikes	\$9
Clothing	\$8
Accessories	\$4
Clothing	\$4
Components	\$3
Components	\$1

The Python output shows a DataFrame with the following mean values:

Category	Price
Accessories	7
Bikes	13
Clothing	6
Components	8

A horizontal bar chart titled 'Image' displays the mean price for each category, with the x-axis representing price (0 to 12) and the y-axis representing categories. The bars show: Components (8), Clothing (6), Bikes (13), and Accessories (7). The chart also includes a legend for the price unit (\$) and a size of 447,274. The ANACONDA logo is visible at the bottom of the chart area.

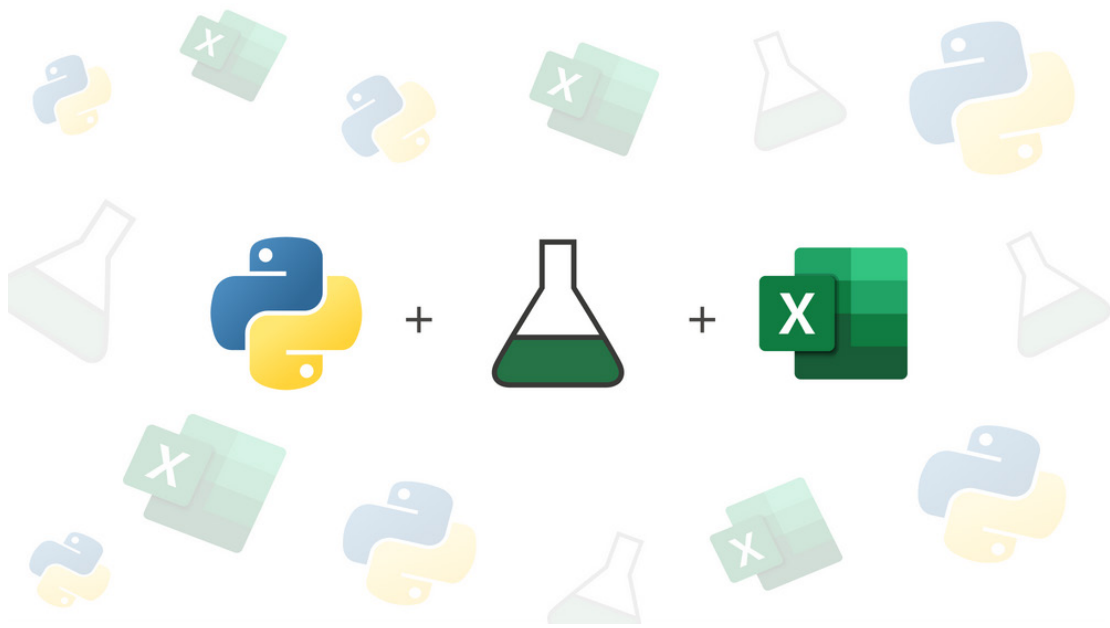
Python Editor from Excel Labs (Insiders Beta)

Another item we mentioned in last month's newsletter, but in case you missed it, we reproduce it here.



Hot on the tails of Python in Excel, the Python Editor is a new addition to Excel Labs (for Excel for Windows) 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

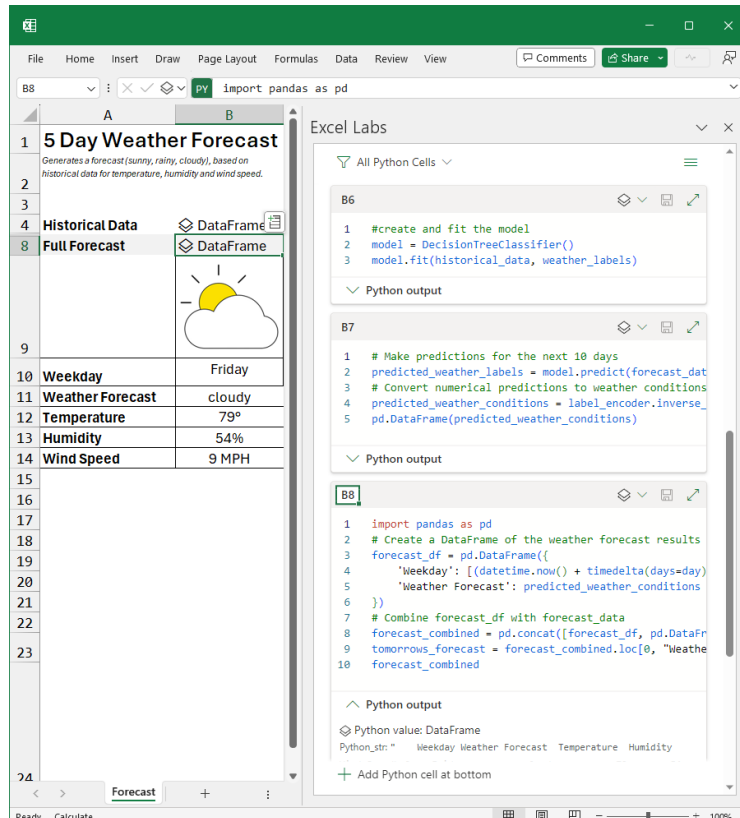
that this is an experimental release to get early feedback for Microsoft, so it might all change. The Python Editor is designed to complement the Python in Excel experience with an editing surface optimised for more complex and longer Python code snippets.



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

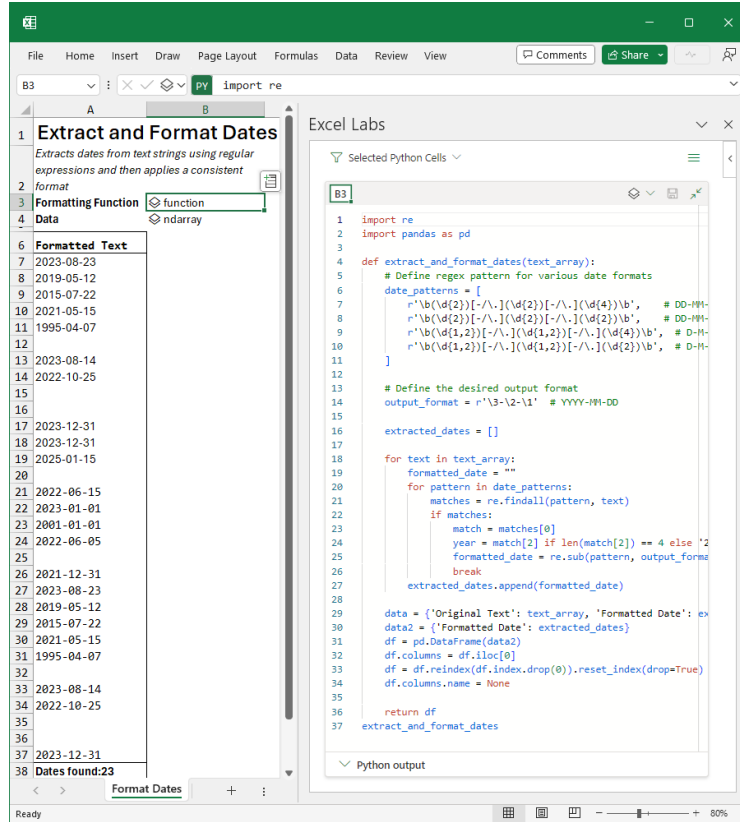
VIEW THE FLOW OF CODE EXECUTION WITH RESULTS

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.



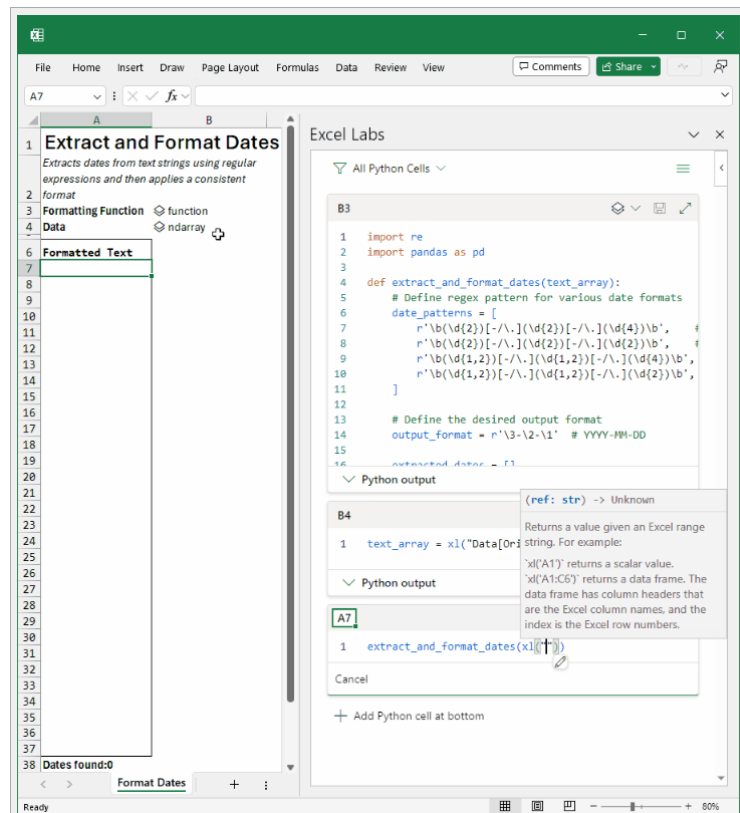
BUILT FOR LONGER SECTIONS OF CODE

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

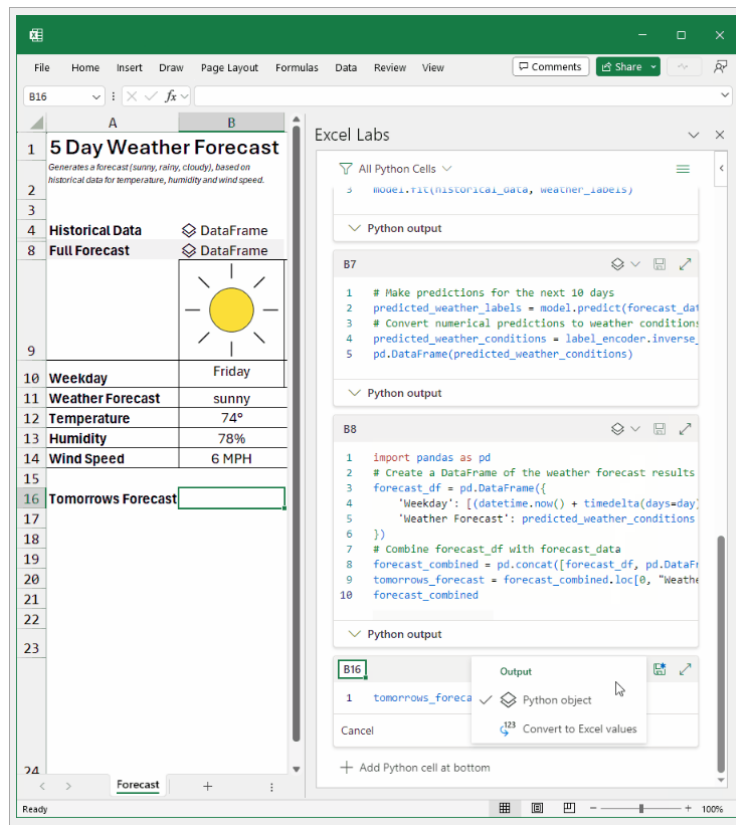
The Python Editor uses some of the same components that power Visual Studio Code and supports many of the features that Visual Studio Code offers for Python development, such as IntelliSense, formatting, code completion and syntax highlighting.



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

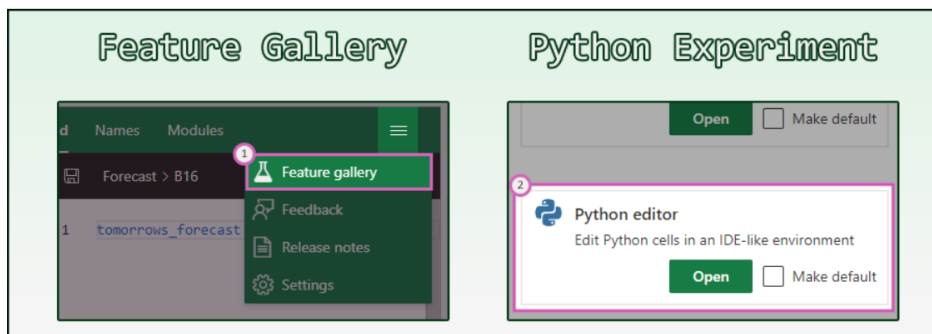
more. You can also switch between the task pane editor, the cell editor and the Formula bar, depending upon your needs and preferences.

HOW TO GET THE PYTHON EDITOR

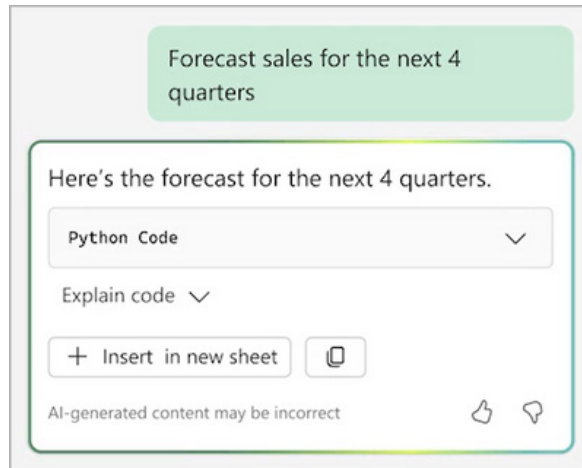
If you already have the Excel Labs add-in, you can update it to get the latest version that includes the Python Editor. If you don't have the Excel Labs add-in, you can install it from the Add-ins Store. To do this, go to the Insert tab, click on 'Get Add-ins', and then search for Excel Labs. Click on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



As mentioned earlier in this newsletter (in quite some detail!), Copilot in Excel for Windows will help you analyse and explore your data in a new way, by writing Python code. Just use everyday language to describe what you want to do with your data, and Copilot will generate and insert working Python code in the Excel grid for you.

To use Copilot and Python in Excel, you will need access to Python in Excel and Microsoft 365 Copilot. It will initially be available in Excel for Windows in English.

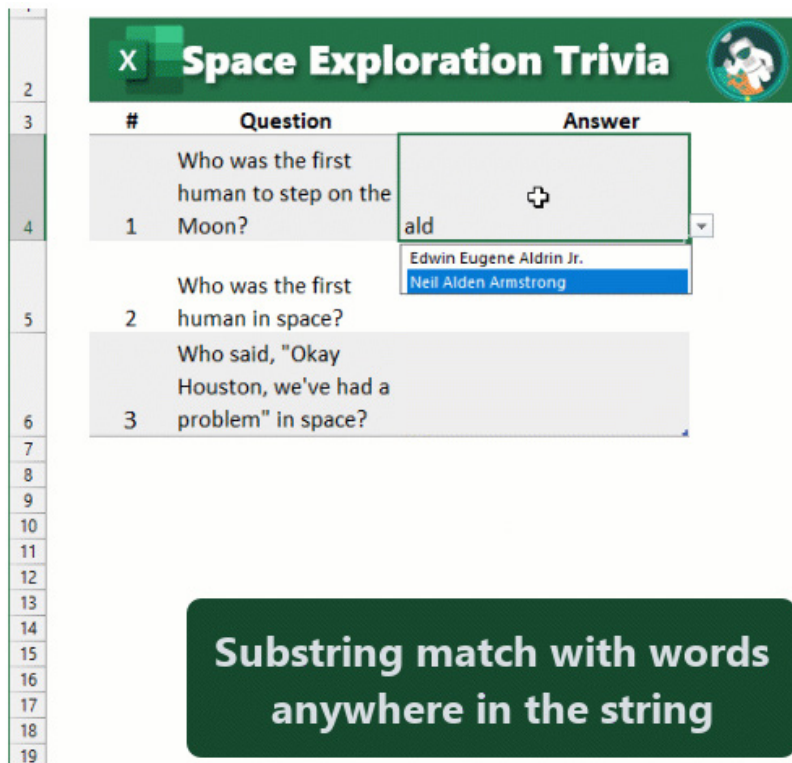
Data Validation AutoComplete

Coming to Excel for Windows and the Insiders version of Excel for Mac, data validation AutoComplete will make many users' lives easier.

Entering data in Excel can be a slow and time-consuming task. Dropdown lists are one way to speed up this process, but scrolling through long lists can be like watching paint dry. By enabling AutoComplete for dropdown lists, we've made it possible for Excel to automatically compare the text you type in a cell against all items in the list, then only display the items that match.

The feature's algorithm can match text from anywhere in the list item—the start, the middle, or the end of the text string. The algorithm also automatically excludes blank items from the validation list.

With AutoComplete, you can fill forms and other types of content faster and more efficiently. You'll spend less time scrolling through lists, dealing with data validation errors or writing complex code to handle this task.



If you want to try AutoComplete for yourself, it's simple:

- open a worksheet in Excel and create a dropdown list (**ALT + D + L** or **Data -> Data Validation -> Data Validation...**)
- enter text in a cell that matches one of the list items. Notice how the list updates as you type to show matching entries
- choose the entry you want in the list or press **ENTER** to fill the cell.

The AutoComplete for dropdown lists feature is rolling out to Microsoft 365 users and Insiders running:

- **Android:** Build 16.0.16816.10000 or later
- **iOS:** Version 2.77 (Build 23090704) or later
- **Mac:** Version 16.77 (Build 23091003) or later.

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 XL 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 15229.10000) 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 Formula 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 15703.10000) 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*
Formulas 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 15730.31863) 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			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
Set automatic data conversions	Version 2207 (Build 15427.20000) or later						
Natural Language Query Improvements			Version 2206 (Build 15330.20230) or later	Version 2205 (Build 15225.20356) or later		Version 16.63 (Build 22070801) or later	
Resize Conditional Formatting dialog box		Version 16.64 (Build 22070400) 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 alt-text suggestions on charts and PivotCharts			Version 2205 (Build 15225.20288) or later	Version 2204 (Build 15128.20280) or later		Version 16.62 (22061100) or later	
Power Query refresh for selected data sources			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 15128.10000) or later						
Faster recalc on resource constrained devices		Version 16.62 (Build 22050804) or later	Version 2204 (Build 15128.20248) or later	Version 2204 (Build 15128.20280) or later			
Faster AutoFilter				Version 2204 (Build 15128.20248) or later	Version 2208 (Build 15601.20456) or later	Version 16.61 (22050700) or later	
Dataflow connector				Version 2203 (Build 15028.20248) or later			
Dataverse connector			Version 2204 (Build 15128.20178) or later				
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: MAX



MAX (and **MIN**) may be familiar to many Excel users, but it is still probably worth performing a recap. **MAX** does exactly what it says on the tin – it calculates the maximum value in a range:

	D	E	F	G	H	I	J
11							
12			Values		MAX		
13			497		497	=MAX(F13:F27)	
14			29				
15			180				
16			115				
17			318				
18			85				
19			474				
20			292				
21			436				
22			56				
23			301				
24			400				
25			255				
26			157				
27			293				
28							

Modellers frequently make mistakes with **MAX**:

	E	F	G	H	I	J	K	L	M	N	O
33											
34		Values		MAX							
35		(390)		(390)							
36		(194)		-							
37		(247)		#VALUE!							
38											

Doesn't treat blanks as zeros
=MAX(F36,J) - does treat blanks in formula as zeros
=MAX(F37,"") - empty strings are treated as text; MAX cannot evaluate text strings

These examples highlight that you should be careful with using blanks both in the ranges and in the formulae themselves. Regular readers will notice I often omit zeros in my formulae but with **MAX** (and **MIN**) I will often take care to explicitly use zeros.

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

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

The **MAX** function has the following argument(s):

- **number1, number2, ...:** **number1** is required, but subsequent numbers are optional
- you may have up to 255 arguments in order to find the maximum value.

It should be noted that:

- arguments may be numbers, or names, arrays or references that contain numbers
- logical values and text representations of numbers that you type directly into the list of arguments are counted
if an argument is an array or reference, only numbers in that array or reference are used; empty cells, logical values or text in the array or reference are ignored
- if the arguments contain no numbers, **MAX** returns zero (0)
- arguments that are error values or text that cannot be translated into numbers cause errors
- if you want to include logical values and text representations of numbers in a reference as part of the calculation, use the **MAXA** function instead.

There is a highly relevant finance example combining **MAX** and **MIN** functions (**MIN** is the similar function that returns the *minimum* value in a range), namely calculating the maximum dividend allowable for a particular period.

Dividends may only be paid out of what are known as distributable reserves (this is a bit of an oxymoron as dividends are also known as **distributions**). Revaluation reserves, share premium accounts, capital

redemption reserves are all non-distributable. Essentially, dividends may only be paid out of the current year's Net Profit After Tax (NPAT) and the aggregation of all previous year's profits after past distributions, Retained Earnings. Dividends may not make the Balance Sheet's Total Equity become negative. This shows insolvency and this sort of distribution is illegal in most territories. Given non-distributable reserves may not become negative allow me to concentrate simply on NPAT and Retained Earnings here.

To derive the maximum dividend allowable, let me consider some scenarios. Let's imagine the following scenario:

Retained Earnings	100
NPAT	40
Maximum Dividend	140

It isn't rocket science that if Retained Earnings and NPAT are both positive, then the maximum dividend allowed is the sum of the two.

Retained Earnings	100
NPAT	(40)
Maximum Dividend	60

If NPAT is negative, but Retained Earnings are positive and exceed the NPAT figure, then the maximum dividend allowed is the net of the two figures. Should the net be negative, no dividend is allowed.

Retained Earnings	(100)
NPAT	40
Maximum Dividend	40

Here is the one that often surprises people. If Retained Earnings is negative but NPAT is positive, regardless of whether the net is positive or negative, the maximum dividend allowed is the NPAT amount. This may seem incomprehensible upon first thought, but it is typically dependent upon two conditions:

- The company's auditors must sign off on it. This is to ensure the company is still seen to be a going concern (*i.e.* it can still continue to operate and trade its way out of any short-term difficulties).
- The shareholders must vote for it. Almost as hilarious as when Members of Parliament solemnly vote for their 50% pay rise each year.

If you think about it some more, this makes sense. Remember, dividends cannot be paid if the company is insolvent. The auditors check to see whether the company can "afford" it for other reasons. But if you don't allow this scenario how would anyone ever attract share capital for a start-up company? A new company may have to provide for certain factors which may never come to fruition. A large non-current

asset may have to be written off as not fit for purpose if a company's strategy changes without any cash consequence. Is it acceptable that shareholders have to wait 10 years for the Retained Earnings losses to be covered even if the business is hugely profitable in the meantime? No, and this is precisely why this is the rule in many territories.

The next scenario is more obvious:

Retained Earnings	(100)
NPAT	(40)
Maximum Dividend	-

With both a negative NPAT and Retained Earnings, there is no leeway now. These scenarios seem to suggest the following formula:

$$=MAX(NPAT + Retained Earnings, NPAT, 0)$$

This allows for the above scenarios. The check to ensure that the value is non-negative (*i.e.* the inclusion of zero in the **MAX** formula) is so that shareholders do not get asked to pay a dividend to the company. I can't imagine that would go down too well.

We are not done yet though. Let's go back to the penultimate scenario but now consider the cash position as well:

Retained Earnings	(100)
NPAT	40
Cash Available	30
Maximum Dividend	30

Here, the Cash Available is the total amount of cash available to pay the dividend. Technically, this includes any cash reserves built up over time, but many companies only consider the cash position for the period the dividend relates to (this is the scenario I shall be modelling in the case study shortly). This seems to suggest the formula:

$$=MIN(MAX(NPAT + Retained Earnings, NPAT, 0), Cash Available)$$

Let me just check with slightly revised numbers:

Retained Earnings	(100)
NPAT	40
Cash Available	(30)
Maximum Dividend	(30)

In this scenario, the company is overdrawn. Oops. Here, this company is going to be asking for money again from its shareholders. Not a good idea. This leads to the slightly revised – and finally correct – formula:

$$\text{Maximum Dividend} = MAX(MIN(MAX(NPAT + Retained Earnings, NPAT), Cash Available), 0)$$

	E	F	G	H	I	J	K	L	M
69									
70		Retained Earnings			(100)				
71		NPAT			40				
72									
73		Cash Available			(30)				
74									
75		Maximum Dividend			-		=MAX(MIN(MAX(I70+I71,I71),I73),)		
76									

Ah yes, the wonderful **MAX(MIN(MAX))** formula. It may not be the prettiest formula in the world, but the point is, it gives the right number.

The A to Z of Excel Functions: MAXA



The **MAXA** function returns the largest value in a list of arguments. It has the following syntax:

$$\text{MAXA}(\text{value1}, [\text{value2}], \dots)$$

where:

- **value1** is required and represents the first number argument for which you want to find the largest value
- **value2, ...** are optional arguments. These are numerical arguments 2 to 255 for which you want to find the largest value.

It should be noted that:

- arguments may be any of the following: numbers; names, arrays or references that contain numbers; text representations of numbers;
- logical values, such as TRUE and FALSE, in a reference
- logical values and text representations of numbers that you type directly into the list of arguments are counted
- if an argument is an array or reference, only values in that array or reference are used. Empty cells and text values in the array or reference are ignored
- arguments that are error values or text that cannot be translated into numbers cause errors
- arguments that contain TRUE evaluate as 1; arguments that contain text or FALSE evaluate as zero (0)
- if the arguments contain no values, **MAXA** returns zero (0)
- if you do not want to include logical values and text representations of numbers in a reference as part of the calculation, use the **MAX** function instead.

Please consider the following example:

	A	B	C
1	Data		
2	FALSE		
3	0.2		
4	0.4		
5	0.6		
6	0.8		
7	TRUE		
8			
9			
10	Formula	Description	Result
11	<code>=MAXA(A2,A3:A6,A7)</code>	The largest number in the range A2:A7. Because TRUE evaluates to 1, this is the largest value.	1.0
12			

The A to Z of Excel Functions: MAXIFS



The **MAXIFS** function returns the maximum value among cells specified by a given set of conditions or criteria. It has the following syntax:

MAXIFS(max_range, criterion_range1, criterion1, [criterion_range2, criterion2], ...)

where:

- **max_range** is the actual range of cells in which the maximum is to be determined
- **criterion_range1** is the set of cells to evaluate with the criterion specified
- **criterion1** is the criterion in the form of a number, expression or text that defines which cells will be evaluated as a maximum
- **criterion_range2** (onwards) and **criterion2** (onwards) are the additional ranges and their associated criteria. 126 range / criterion pairs may be specified. All ranges must have the same dimensions otherwise the function returns an **#VALUE!** error.

It should be noted that:

- the size and shape of the **max_range** and **criteria_rangeN** arguments must be the same, otherwise these functions return the **#VALUE!** error.

As an example:

H38

Product	Sales Person	Business Unit	Amount
Guns	Al	North	100
Drugs	Bri	North	200
Drugs	Jo	South	300
Guns	Jo	East	400
Guns	Al	North	500
Roses	Al	West	600
Guns	Jo	East	700
Guns	Jo	South	800
Drugs	Al	North	900
Drugs	Bri	North	1,000
Roses	Al	South	1,100
Roses	Bri	South	1,200
Roses	Jo	North	1,300
Guns	Bri	South	1,400
Roses	Bri	East	1,500
Drugs	Al	West	1,600
Drugs	Al	East	1,700
Guns	Bri	East	1,800
Roses	Bri	South	1,900

Product: Guns
 Sales Person: Al
 Business Unit: North
 Amount: 500

This example *is* preferable to its standard Excel counterpart:

{=MAX(IF(G13:G31=H34,IF(H13:H31=H35,IF(I13:I31=H36,J13:J31)))}}

Array formulae (see www.sumproduct.com/thought/array-of-light.html for more information) are cumbersome and not readily understood, which is why **MAXIFS** may be a highly viable alternative.

More Excel Functions next month.

Beat the Boredom Suggested Solution

Earlier in the newsletter, we stated that a positive integer (whole number) is said to be **prime** if only two different numbers divide it: one [1] and itself. Therefore, the number 1 isn't prime (it has only one divisor). The challenge laid down was to ask, can you create a formula, without using a user-defined function or VBA, that will determine whether a number is prime?

Example

Number in Question

Number

Result

Let me call the number to evaluate, er, **Number**. Some of you will have headed straight to the internet, to the home of your favourite search engine and located one of two very popular solutions, either

{=IF(Number=2,"Prime",IF(AND(MOD(Number,ROW(INDIRECT("2:"&ROUNDUP(SQRT(Number),0))))<>0),"Prime","Not Prime"))}

or

{=IF(Number=2,"Prime",IF(AND(MOD(Number,ROW(OFFSET(\$A\$2,,,ROUNDUP(SQRT(Number),0)-1)))<>0),"Prime","Not Prime"))}

They have four things in common:

1. They are both array functions (*i.e.* entered using **CTRL + SHIFT + ENTER** – you do not try and type the braces ({ and }) in
2. These solutions are cited all over the internet, including on esteemed websites like Microsoft's own *Excel Blog*
3. They are both fairly incomprehensible upon initial inspection
4. Neither actually *work*.

Yup, read point 4 again. They're wrong. They don't work in all situations – and yet they seem to be universally accepted. And that's the point of today's blog. When we don't know how to do something, you head for your favourite search engine, hunt out the answer – and if you are wiser, check four or five sites to ensure they come to the same conclusion before accepting that answer.

These formulae would pass that sniff test. That's my point. But they are still wrong.

To show they don't work try these examples:

- apparently, 1.7 is prime
- with the first formula, the value 0 (zero) gives #REF!
- with the second formula, the value 1 gives #REF!
- negative numbers give #NUM!

Nobody tests anything anymore. We are all download drones. And this scares the living daylights out of me. People might try numbers like 2, 17, 59, etc. and see it seems to work and conclude everything is fine. But just because something is on page 1 of a well-known search engine page doesn't make it true.

Here's our attempt:

```
{=IF(AND(MOD(IF(OR(MOD(Number,1),Number<0),,Number-1*(Number<=2)),ROW(OFFSET($A$2,,,ROUNDUP(SQRT(ABS(Number))),0)-1*(Number>1))))<>0),"Prime","Not Prime")}
```

That's right – it's even worse than the other two. In the next section, I'll explain how this formula works, but feel free to stop reading at this juncture. Showing off some horrendous calculation is not the objective of this article. It's realising that you should be careful when reading articles on how to solve things in Excel. Yes, that includes SumProduct blogs too!

How the Formula Works

You don't have to read this! If you are interested though, the formula

```
{=IF(AND(MOD(IF(OR(MOD(Number,1),Number<0),,Number-1*(Number<=2)),ROW(OFFSET($A$2,,,ROUNDUP(SQRT(ABS(Number))),0)-1*(Number>1))))<>0),"Prime","Not Prime")}
```

needs to be picked apart. However, before I do that, I need to show you a quick way to check for a prime. Consider the divisors of 60:

1 x 60	60 x 1
2 x 30	30 x 2
3 x 20	20 x 3
4 x 15	15 x 4
5 x 12	12 x 5
6 x 10	10 x 6

Do you see the symmetry? Once you get to the middle – the square root – the factorisation reverses. Since the square root of 60 is slightly less than eight [8], I only need to test to see if the numbers 1 to 8 divide 60. Given 1 and n will always divide n , I only need to test the numbers two [2] to $\text{ROUNDUP}(\text{SQRT}(n),0)$ (the square root of n rounded up to the next whole number), which is eight here. This saves a lot of needless calculations!

This explains the element highlighted:

```
{=IF(AND(MOD(IF(OR(MOD(Number,1),Number<0),,Number-1*(Number<=2)),ROW(OFFSET($A$2,,,ROUNDUP(SQRT(ABS(Number))),0)-1*(Number>1))))<>0),"Prime","Not Prime")}
```

The function $\text{OFFSET}(\text{Reference}, \text{Rows}, \text{Columns}, [\text{Height}], [\text{Width}])$ has been discussed previously (see the article [Onset of Offset](#) for further details).

The formula $\text{OFFSET}(\$A\$2,,, \text{ROUNDUP}(\text{SQRT}(\text{Number}),0)-1)$ gives a range of **A2** to **A8** if **Number** were 60 (1 has to be subtracted otherwise with a height of 8, you would get the range **A2:A9**). Putting $\text{ROW}()$ around this expression is a crafty way to convert to the numbers 2 to 8 inclusive. This is why cell **A2** must be a cell reference on row 2.

The next key element is $\text{MOD}(\text{Number}, \text{Divisor})$. Again, I have discussed MOD before (please see [A Modicum of MOD](#) for further details): this provides the remainder after dividing **Number** by **Divisor**. So, if I divide **Number** by each **Divisor** for the numbers 2 to $\text{ROUNDUP}(\text{SQRT}(\text{Number}),0)$ – which would be 2 to 8 if **Number** were 60 – and any division gives me zero, that would mean **Number** is divisible by **Divisor**, i.e. the number is not prime. If no division gives zero, it has to be prime so

```
{= IF(AND(MOD(Number,ROW(OFFSET($A$2,,,ROUNDUP(SQRT(Number),0)-1)))<>0),"Prime","Not Prime")}
```

gives the appropriate prime / not prime identification in certain circumstances. This formula is both a variation of my proposed one and the second one identified as wrong earlier. Do note it is encapsulated in an **AND** function and is in an array formula. This forces Excel to perform all of the calculations from 2 to $\text{ROUNDUP}(\text{SQRT}(\text{Number}),0)$ in one cell. That's a useful way of using an array formula (for more on arrays, please see [Array of Light](#)).

The rest of my formula,

```
{=IF(AND(MOD(IF(OR(MOD(Number,1),Number<0),,Number-1*(Number<=2)),ROW(OFFSET($A$2,,,ROUNDUP(SQRT(ABS(Number))),0)-1*(Number>1))))<>0),"Prime","Not Prime")}
```

- is made up of tweaks to ensure:
- non-integers are not prime
- 0 is neither prime nor an error
- 1 is neither prime nor an error

My formula goes wrong eventually. 9×10^{200} is clearly not prime (it's 3×10^{200} squared) but Excel cannot evaluate it because it's too large. Meh, you can't have everything!

Keep reading these articles, but stay cynical. I am as prone to mistakes as the next persona (that's a joke!). By all means research, but convince yourself that it's right. Take nothing at face value.

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)	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 saying **HI** last month, we've asked Freddie for his suggestions (you know him – he lives on **LM** street...).

Keystroke	What it does
<u>L</u> eft Arrow	Move left one cell
ALT + <u>L</u> eft Arrow	Back (hyperlink navigation)
CTRL + <u>L</u> eft Arrow	Select the last cell in the area left
SHIFT + <u>L</u> eft Arrow	Extend selection left one cell
ALT + SHIFT + <u>L</u> eft Arrow	Ungroup
CTRL + ALT + <u>L</u> eft Arrow	Intel Chipset: Turn screen +90 degrees; else: move active cell to previous non adjacent area within selection
CTRL + SHIFT + <u>L</u> eft Arrow	Extend selection down to last cell in area left
CTRL + <u>M</u> ultiply (*)	Select current region

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

Our Services

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

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

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

Link to Others

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

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

Any Questions?

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

Training

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

Check out our more popular courses in our training brochure:



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

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

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