

# SumProduct

NEWSLETTER #106 - September 2021

[www.sumproduct.com](http://www.sumproduct.com) | [www.sumproduct.com/thought](http://www.sumproduct.com/thought)



## We're three-quarters of the way through the year

and we are finally in a position to make an announcement about this year's conference: Excel Virtually Global is coming back! We provide a teaser below...

We also provide details about the Navigation pane in Excel and relate the beginning of the end for Internet Explorer. But we also have the regulars: we have another Beat the Boredom Challenge, plus Charts & Dashboards, Visual Basics, Power Pivot Principles, Power Query Pointers, Power BI updates, Keyboard Shortcuts and we continue going through the "Gs" of the 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 Returns



We don't have all the details yet – but it's time to make an announcement as we're doing this mid-October!!

Microsoft's Most Valuable Professionals, or MVPs, are technology experts who passionately share their knowledge with the community. They are always on the "bleeding edge" and have an unstoppable urge to get their hands on new, exciting technologies.

How much would you be prepared to pay for the likes of Excel (and other) MVPs, Microsoft staff and other industry experts to present on a variety of Excel topics **in your house**?

Well, it's an option. For the second successive year, our usual **Excel Summit South** will not be held this year due to the COVID-19 pandemic. It will be held online – and will still be for charity.

This virtual conference, now in its sixth year, presents Excel, Data Platform and PowerPoint MVPs, together with other acknowledged experts from Microsoft and / or around the globe to present, answer questions and demonstrate the future of Excel and how it will make your life easier, personally and professionally.

Topics and presenters have not yet been agreed, but you know the drill! We'll round a few up and get them to entertain you. Dates will be in the week commencing **Monday 11 October 2021**, with precise dates and

times to be announced SOON.

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

Most sessions will be recorded so you may watch them later with downloads aplenty – there are no medals for staying up to watch the entire event live! That's just as well, as we plan to beat last year's marathon (no guarantees), which lasted a "mere" 57 hours. And it nearly killed us! 😊

From your own favourite chair, bring a laptop, an inquisitive mind and your sense of humour. Be prepared to learn heaps. And remember, it's not just value for money, we will be asking for donations to charity this time around – all profits will be donated to a global COVID-19 research charity.

More details will be available in the News section of our website soon: [www.sumproduct.com/news](http://www.sumproduct.com/news).

Hopefully, we'll see you there!

# Navigation Not Such a Pane

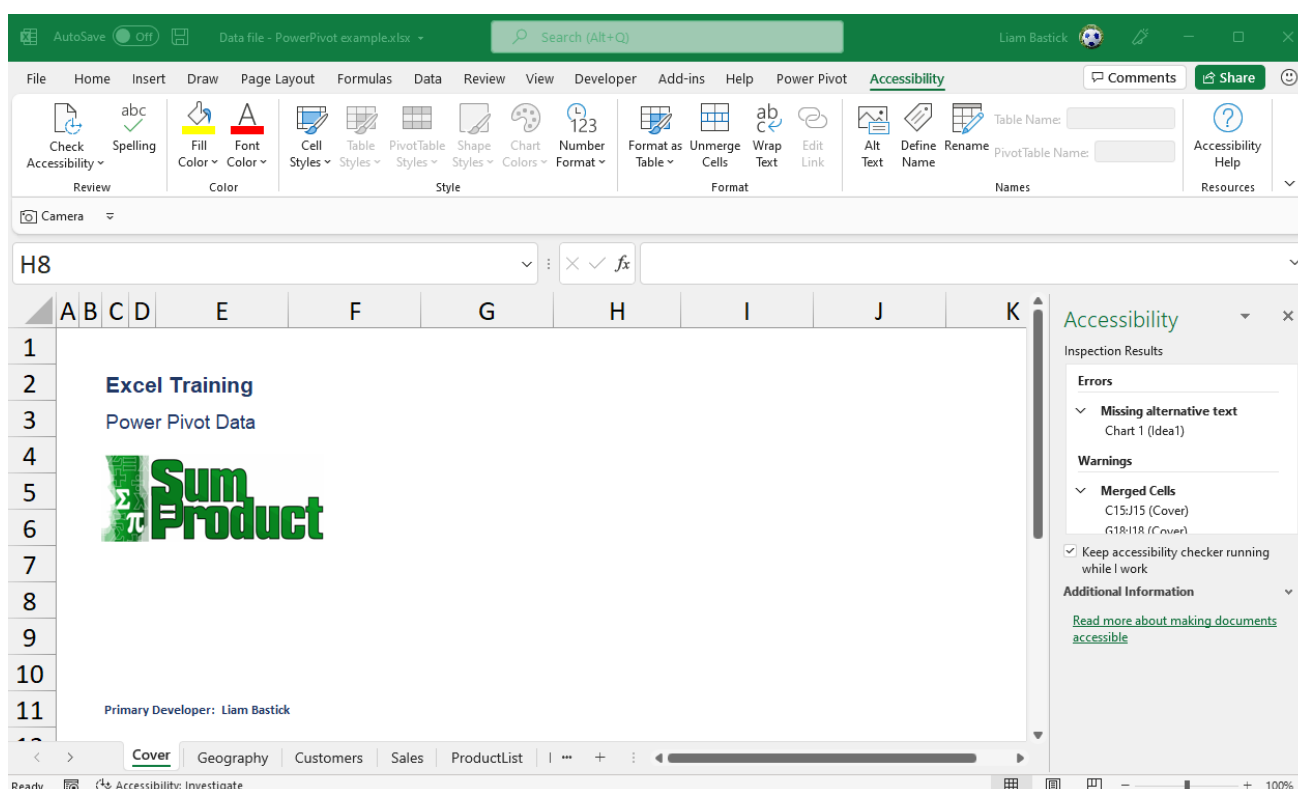
Microsoft has made a “big thing” – quite rightly – about accessibility for all of their products, to ensure they are inclusive to everyone, including more than one billion people with various disabilities around the world.

With regard to Excel in particular, Microsoft has stated that they have three overarching objectives to help development:

1. **Work seamlessly with assistive technology.** Examples of assistive technology include screen readers, dictation software, magnifiers and physical devices. Here, the intention is that everyone may use Excel in the way that works best for them, with tools that are already familiar (e.g. keyboard shortcuts, large icons)
2. **Build “inclusive and delightful experiences” inspired by people with disabilities.** The aim is that Excel should be efficient and “delightful” to use. I must remember that next time it’s 3am and I am trying to get a Balance Sheet to balance. Microsoft continue to consider opportunities to simplify workflow, summarise your content or suggest information

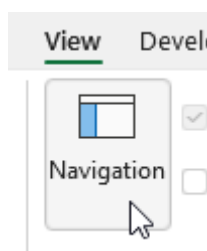
3. **Support authors to create accessible content.** Making Excel as an application accessible is only half the battle; the other half is making sure workbooks created in Excel are accessible. To support users in creating accessible content, Excel has tools and features now that let you know when something is inaccessible and help you fix accessibility issues before sharing your workbook with others.

Therefore, as a part of the goal to help you create accessible content, there is a new contextual Ribbon called ‘Accessibility’, which is rolling out in Excel. The Accessibility Ribbon will be available when you open the ‘Check Accessibility’ pane on the Review tab. On the Accessibility Ribbon you will find a collection of the most common tools you need to make your workbook accessible. The Ribbon is available presently to those in the Office Insiders program.

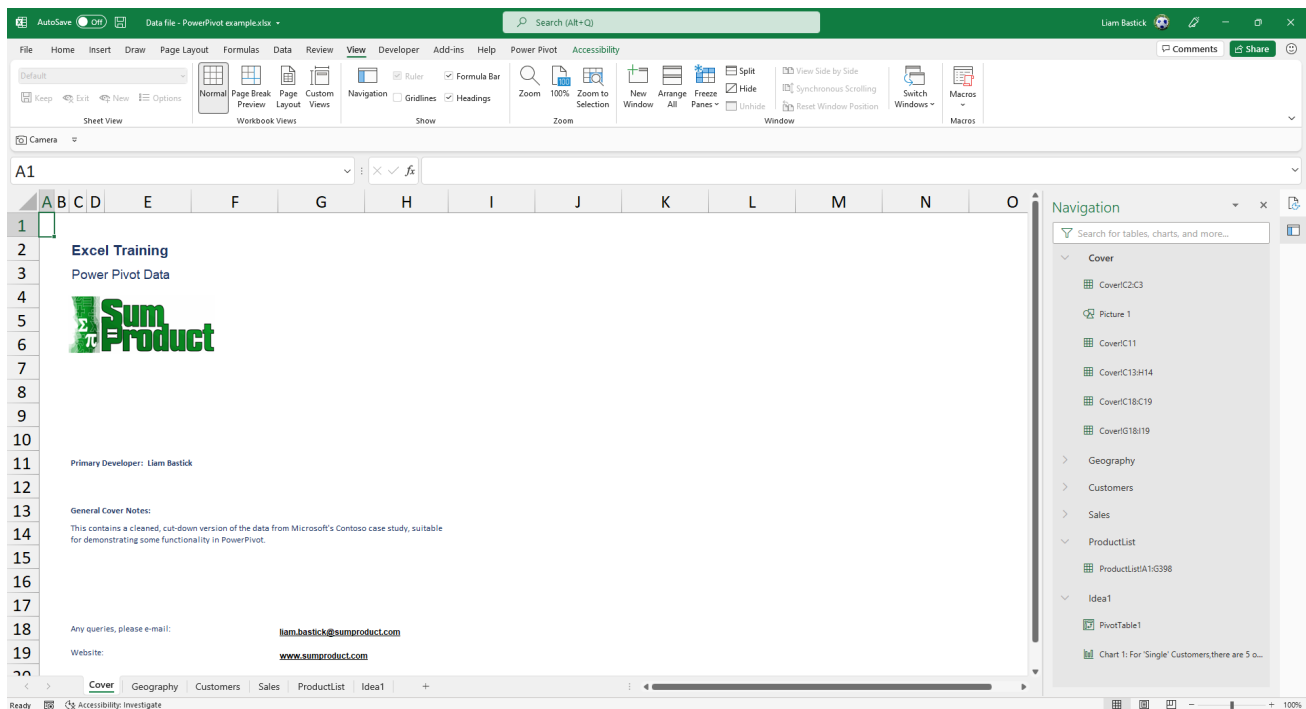


To add to this, late July saw Microsoft’s second feature announced: the introduction of the Navigation pane in Excel for Windows is now available to those in the Office Insiders program.

To access it, click the Navigation button on the View tab of the Ribbon:



This launches the Navigation pane:



The objective of the Navigation pane is to:

- orient customers quickly to the structure of an unfamiliar workbook
- help customers find content efficiently and feel the confidence that they have not missed content
- provide an easy way to navigate between different worksheets and Excel objects, and to update basic properties of those objects.

These objectives are applicable whether you are visually impaired / blind, a novice Excel user who doesn't know where to start after opening a workbook created by someone else, an Excel power user with dozens of worksheets that you need to sort through to find the data you need or mentally intoxicated (like me!). Microsoft used interviews with

people who are blind or have low vision, supplemented by feedback and requests from the larger Excel community to design Navigation pane.

Hopefully, everyone can now "see" how their file is structured!

More improvements to come shortly, we are sure.

## Internet Explorer Has Begun Its Death Throes

As previously mentioned in June's newsletter, Internet Explorer 11 (IE11) desktop application is to be retired. As of Tuesday 17 August 2021, Microsoft 365 apps and services no longer support this application. As a result, you may have a degraded experience, or be unable to connect to,

any apps and services from IE11. These apps and services will phase out over weeks and months to help ensure a smooth end of support, with each app and service phasing out on independent schedules.



Microsoft has stated that the future of Internet Explorer on Windows 10 is in Microsoft Edge, and that Internet Explorer mode (IE mode) provides legacy browser support within Microsoft Edge. This means that the IE11 desktop application will no longer be supported and afterwards will redirect to Microsoft Edge if a user tries to access it.

If your organisation has legacy sites and applications with Internet Explorer dependencies, your business will need to set up IE mode so that users can access them through Microsoft Edge.

## 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...

Consider  $n!$  where

$$n! = n(n-1)!$$

$$\text{i.e. } n! = n(n-1)(n-2) \dots 3 \times 2 \times 1.$$

For example,  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ . These numbers become astronomical (technical term) very quickly. And it gets worse.

Excel has a function, **FACT**, which calculates factorials. Except it doesn't. Consider the following summary:

Number	Factorial	Length
1	1	1
2	2	1
3	6	1
4	24	2
5	120	3
6	720	3
7	5040	4
8	40320	5
9	362880	6
10	3628800	7
11	39916800	8
12	479001600	9
13	6227020800	10
14	87178291200	11
15	1.30767E+12	13
16	2.09228E+13	14
17	3.55687E+14	15
18	6.40237E+15	16
19	1.21645E+17	18
20	2.4329E+18	19
21	5.10909E+19	20
22	1.124E+21	20
23	2.5852E+22	19

Using Excel,  $23!$  is 2.5852E+22 (Scientific notation), which when expanded is 25,852,016,738,885,000,000,000. Actually,  $23!$  is really 25,852,016,738,884,976,640,000. Microsoft Excel can only calculate to 15 significant figures and then it starts rounding. Not good.

But it's even worse than that. The **LEN** function in Excel counts how many characters there are in a text string. According to Excel,  $22!$  has 20 digits whereas  $23!$  has 19 digits. So  $23!$  is *smaller* than  $22!$  (yeah, right). What on earth..?

So here's this month's challenge: using Excel (unless you fancy a night of long multiplication), calculate  $171!$  precisely (i.e. not just to 15 significant figures).

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

## A Top Excel Blog?

We are pleased to announce that according to **Feedspot**, SumProduct has been named one of the web's Top 70 Excel blogs. That's nice to know someone is reading our rambles!

### 26. SumProduct



SumProduct is an expert in excel training. At SumProduct, we provide financial modelling, strategic data modelling, model auditing, planning & strategy and training courses. Follow to get daily excel tips and other updates. Sumproduct is expert in Excel Training Financial Modelling, Strategic Data Modelling, Model Auditing, Planning & Strategy, Training Courses, Tips & Online Knowledgebase.

**Frequency** 1 post / week

[sumproduct.com/news.html](https://sumproduct.com/news.html) [+ Follow](#)

Twitter 69

[View Latest Posts](#) · [Get Email Contact](#)

It might seem like we are one person, but we are not– and it should be noted we put up between five and 10 articles a week – so please do keep coming back!

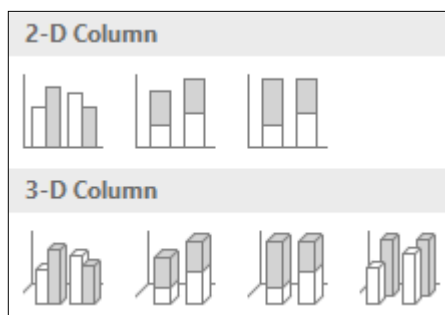


## Charts and Dashboards

*It's time to chart our progress with an introductory series into the world of creating charts and dashboards in Excel. This month, we look at Column charts.*

A Column chart shows the magnitude of values in one or more data series using (typically) rectangular bars, with the height of each corresponding to the associated value. The categories appear on the horizontal axis and the values are mapped on the vertical axis.

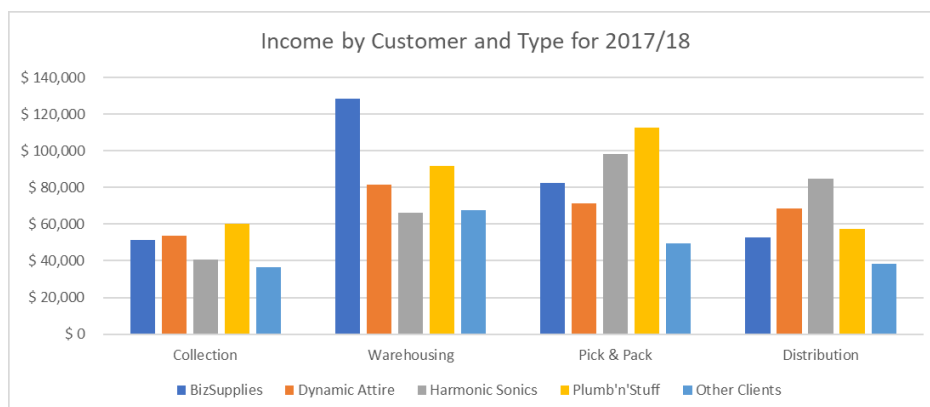
Excel provides a range of Column Chart variations, including Clustered and Stacked formats in 2-D and 3-D.



Let's have a look at the sales data by customer group:

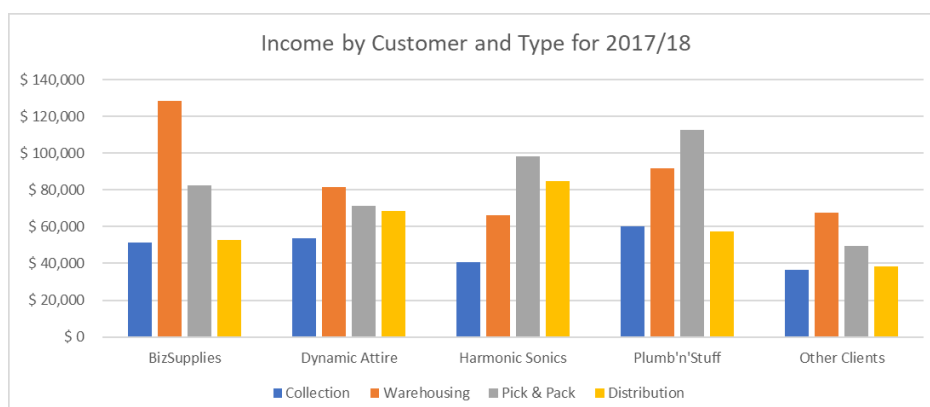
Income for 2017/18 by Type and Customer					
Income Type	BizSupplies	Dynamic Attire	Harmonic Sonics	Plumb'n' Stuff	Other Clients
Collection	\$ 51,273	\$ 53,709	\$ 40,677	\$ 60,250	\$ 36,616
Warehousing	\$ 128,194	\$ 81,318	\$ 66,315	\$ 91,765	\$ 67,401
Pick & Pack	\$ 82,399	\$ 71,216	\$ 98,094	\$ 112,581	\$ 49,320
Distribution	\$ 52,637	\$ 68,502	\$ 84,749	\$ 57,393	\$ 38,515

By selecting this data and inserting a 2-D Clustered Column chart, we may obtain an initial chart like the one below:



The current chart is great in communicating that the sales from each of the main four customer groups is more than the revenue from the Other Clients group. It also shows BizSupplies clearly provides the most Warehousing income.

However, you may be more concerned about income from each activity by customer, so you will need to flip this chart around to report columns by customer groups, not sales type. In Excel, this is easily done. We simply click on the chart and under the Design tab, and then select 'Switch Row/Column'. The chart now looks as follows:

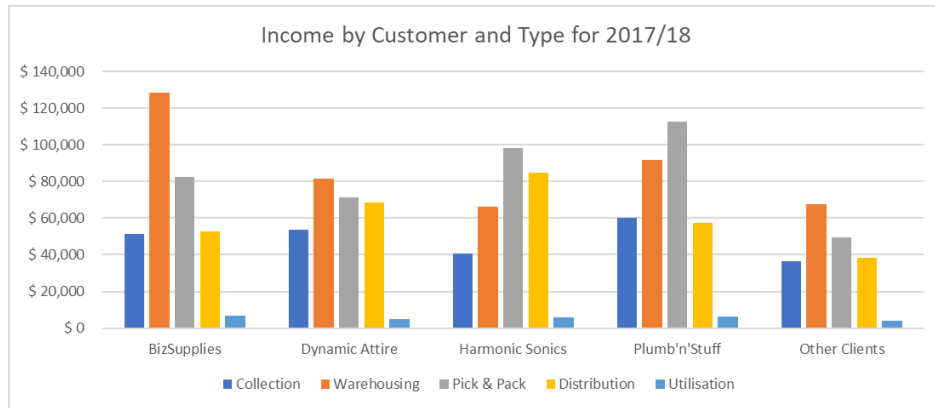


Then, add one more row of data into my previous data table, which should also be updated in the chart:

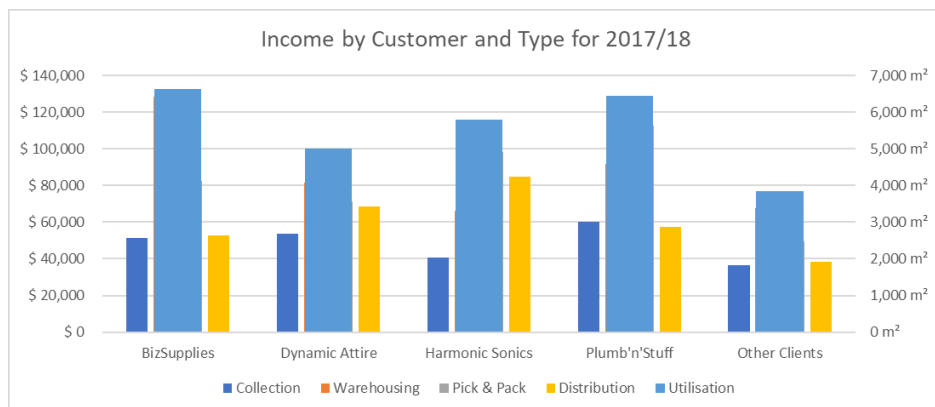
Warehouse Utilisation	BizSupplies	Dynamic Attire	Harmonic Sonics	Plumb'n'Stuff	Other Clients
	6,621 m <sup>2</sup>	4,995 m <sup>2</sup>	5,797 m <sup>2</sup>	6,440 m <sup>2</sup>	3,837 m <sup>2</sup>

Note that the newly added data is not financial data, with a different unit associated (which is m<sup>2</sup>), which means you can't just simply add this series into the column chart. Hence, we need to undertake further steps:

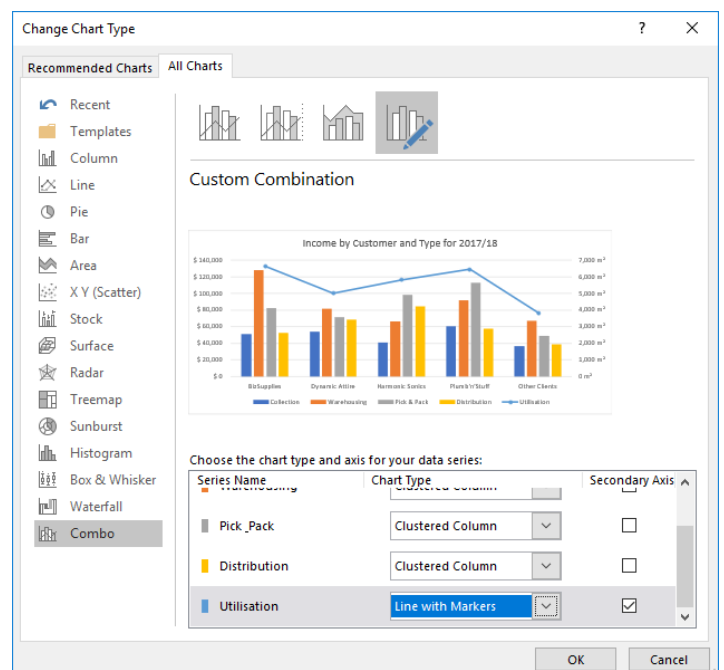
- Highlight the data and copy it
- Click on the 'Chart Area', right-click and choose Paste. The data series is added to the chart as a fifth series (the light blue Utilisation data points):



- Select this fifth data series (circular markers will appear on each corner of each column of the series), right-click and choose 'Format Data Series'
- Under 'Series Options', there is an option titled 'Plot Series On'. Select the 'Secondary Axis' and the chart will update. A vertical axis will appear on the right side of the chart and the data series will now overlay the other four series

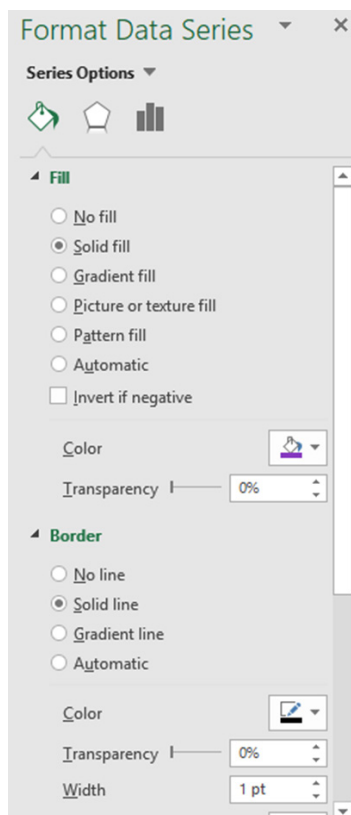


- Finally, in order to see the original sets of columns clearly once again, we need to change this new series from being columns to a line. Select this newest series, right-click, this time choose 'Change Series Chart Type' and a 'Change Chart Type' screen will appear. Scroll down in the bottom window within this screen and change the 'Chart Type' for Utilisation to 'Line with Markers'. There will be a chart within this window reflecting our requested change, then click OK to confirm the settings.

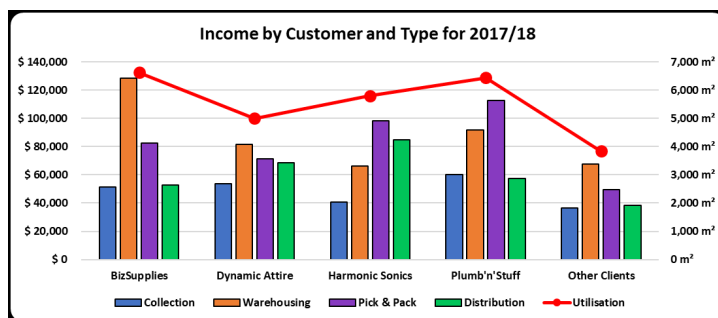


We still want to make some further modifications:

- We wish to change the colour code of each group to match my organisation colour scheme (say). Right-click on the series; in this case, we will select one of the columns that we wish to change colour, and then select 'Format Data Series'. Under the menu, click on the bucket icon, Fill. Here, we may set this to 'Solid fill' and change the colour as desired. Repeat these steps with other columns and the line indicating Utilisation
- We also want to enhance the border around each column so for all four data series relating to the columns, under Border, set it to 'Solid line' and change the colour to black, and then set the 'Width' to 1 point.



With all these modifications made, the chart is now complete:



More next month...

## Visual Basics

We thought we'd run an elementary series going through the rudiments of Visual Basic for Applications (VBA) as a springboard for newer users. This month, we look at using ListObjects to manipulate Tables within an Excel workbook in VBA.

Warning! This time out, this article was originally written whilst listening to an Adele playlist...

"Rumour Has It" that last month, how to convert a Range into a Table

object using VBA was discussed. But "Hello", what if the table already is in the workbook? How do you get access to the "One and Only"?

Let's look at the table in the workbook.

AutoSave Off File Home Insert Page Layout Formulas Data Review View Developer Cecile Table Tools Design

Table Name: Adele\_Top\_Ten

Properties: Resize Table

Tools: Summarize with PivotTable Remove Duplicates Convert to Range Insert Slicer Export Refresh

External Table Data: Properties Open in Browser Unlink ☒ Header ☒ Total Row ☒ Banded

B6 Send My Love (To Your New Lover)

	A	B	C	D	E	F
1	Rank	Song Title	Peak Position	US Singles Sold (m)	Peak Date	
2	1	Rolling in the Deep	1	8.5	21-May-2011	
3	2	Someone Like You	1	5	17-September-2011	
4	3	Hello	1	7	14-November-2015	
5	4	Set Fire to the Rain	1	4	04-February-2012	
6	5	Send My Love (To Your New Lover)	8	1	24-September-2016	
7	6	Rumour Has It	16	2	05-May-2012	
8	7	Skyfall	8	1	20-October-2012	
9	8	When We Were Young	14	1	05-March-2016	
10	9	Water Under the Bridge	26	0.5	11-February-2007	
11	10	Chasing Pavements	21	1	28-February-2009	
12	Total					

The correct way to reference a table is to use the following codes. Let's create a variable so that VBA can "Take It All" from the table to work with.

```
(General)
Option Explicit

Sub TurningTables()

Dim objListObject As ListObject

Set objListObject = ActiveSheet.ListObjects(1)

Debug.Print objListObject.Name

End Sub
```

Immediate: Adele\_Top\_Ten

This works very well if there is "One and Only" table on the worksheet and if the subroutine / function is called from the worksheet that the table is on. If there are multiple tables on the worksheet, how would one check that the right one is being referenced? One could get "Tired" trying to figure it all out.

A better way to reference it by using the Table and Worksheet Names.

```
(General)
Option Explicit

Sub TurningTables()

Dim objListObject As ListObject

Set objListObject = Worksheets("Sheet1").ListObjects("Adele_Top_Ten")

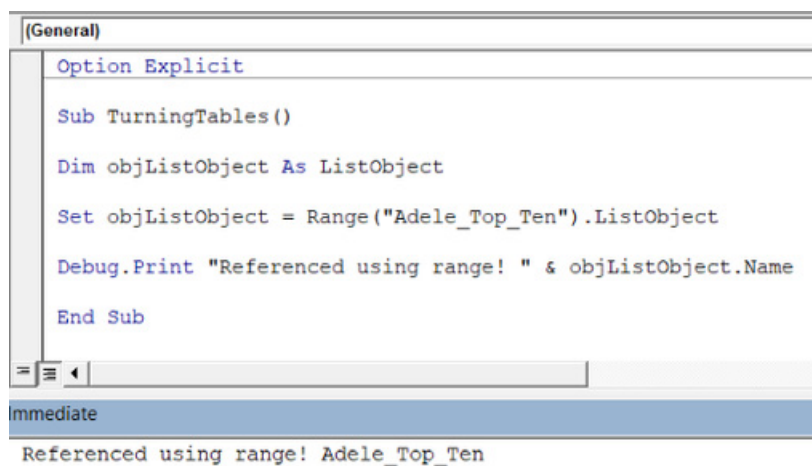
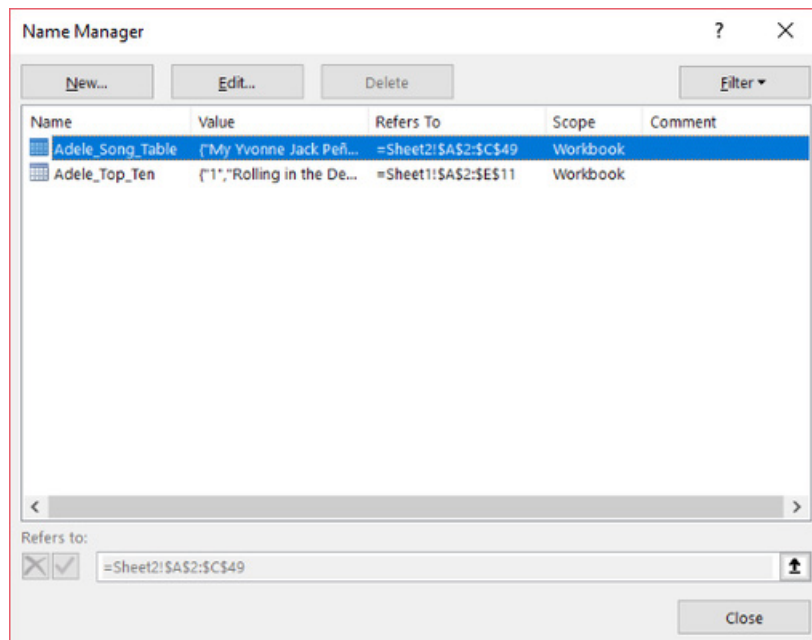
Debug.Print "Referenced by name! " & objListObject.Name

End Sub
```

Immediate: Referenced by name! Adele\_Top\_Ten

However it "Can't Let Go" using the sheet reference and the table might move around the workbook or the sheet might be renamed. "All I Ask" is to avoid using the Sheet as a reference. Saving the "Best for Last", this can be done by using Range in conjunction with Range's **ListObject** property.

Tables are also defined Named Ranges in the Excel dictionary. The Name Manager will also include all Tables that are in the Workbook so don't go "Rolling in the Deep" trying to find all the Tables!



So now having found the “Remedy”, the Table can be referenced directly from VBA. There will be problems “Now and Then” when referring to a Named Range that isn’t actually a Table, so some error checking code would be appropriate.

We’ll go through the **ListObject** properties next time, so for now “That’s It, I Quit, I’m Moving On”.

More next time.

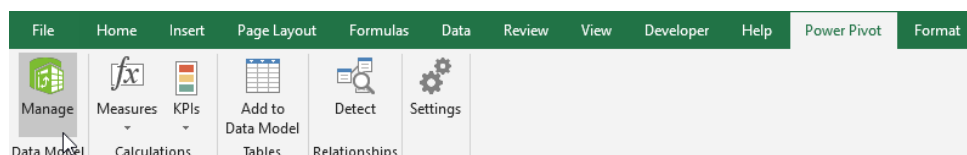
## Power Pivot Principles

*We continue our series on the Excel COM add-in, Power Pivot. This month, we provide an overview of how to refresh data.*

To refresh data in your model, navigate to the ‘Power Pivot’ tab on the Ribbon.

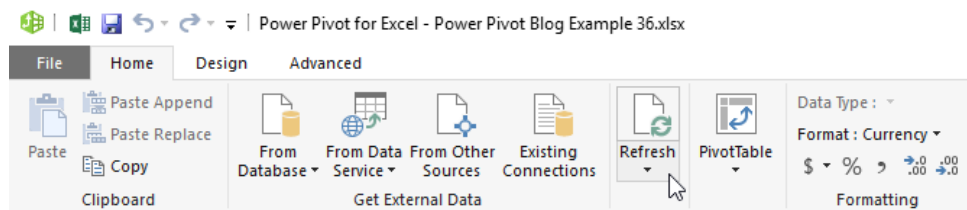


Click on the ‘Manage’ option in the ‘Data Model’ group.

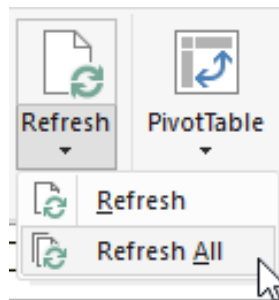


In the ‘Power Pivot for Excel’ dialog box, click on the drop-down icon on the ‘Refresh’ option on the Ribbon.

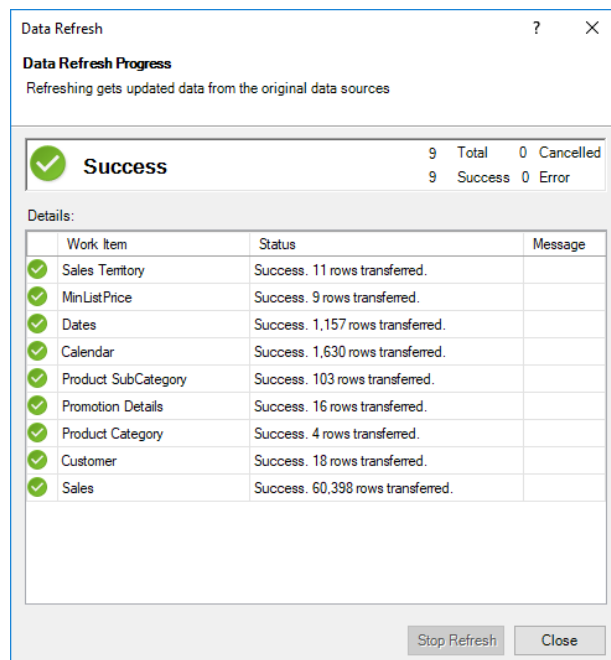




Then select the 'Refresh All' option:



The 'Data Refresh' dialog box will appear displaying the refresh stats of each connection source.



Not such a complicated article this month: was that a refreshing change?

More *Power Pivot Principles* next month.

## Power Query Pointers

Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Office 365, Excel 2016 and 2019) from [www.sumproduct.com/blog](http://www.sumproduct.com/blog). If you wish to read more in the meantime,

simply check out our Blog section each Wednesday. This month, we consider a way to store queries in a text file.

Previously ([www.sumproduct.com/blog/article/power-query-blogs/power-query-its-good-to-share-a-query](http://www.sumproduct.com/blog/article/power-query-blogs/power-query-its-good-to-share-a-query)), we have looked at a few ways to share queries between workbooks. This month, let's take a look at a slightly different approach, which relies on the following **M** function:

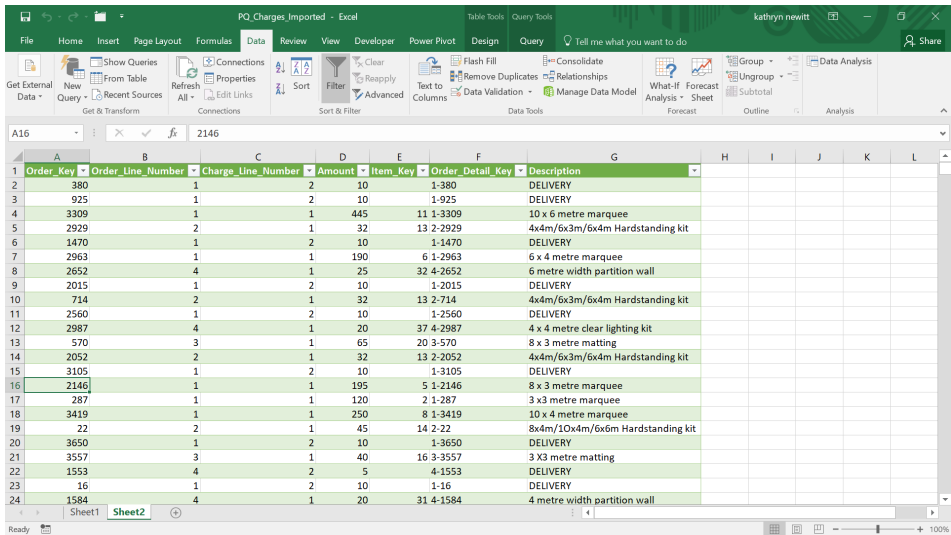
**Expression.Evaluate(expression as text, optional environment as [...])** as any

It should be noted:

- this evaluates a text expression and returns the evaluated value
- **expression** is the expression to evaluate
- **optional environment** is the **expression** environment.

That’s a very short explanation for quite a powerful function! What it does is to treat the expression as **M** language and then runs it. The best way to understand it, is to see it in action.

Consider the following. Below, we have product data. We are going to create a simple query:



Order_Key	Order_Line_Number	Charge_Line_Number	Amount	Item_Key	Order_Detail_Key	Description
380	1	2	10	1-380		DELIVERY
925	1	2	10	1-925		DELIVERY
3309	1	1	445	11 1-3309		10 x 6 metre marquee
2929	2	1	32	13 2-2929		4x4m/6x3m/6x4m Hardstanding kit
1470	1	2	10	1-1470		DELIVERY
2963	1	1	190	6 1-2963		6 x 4 metre marquee
2652	4	1	25	32 4-2652		6 metre width partition wall
2015	1	2	10	1-2015		DELIVERY
714	2	1	32	13 2-714		4x4m/6x3m/6x4m Hardstanding kit
2560	1	2	10	1-2560		DELIVERY
2987	4	1	20	37 4-2987		4 x 4 metre clear lighting kit
570	3	1	65	20 3-570		8 x 3 metre matting
2052	2	1	32	13 2-2052		4x4m/6x3m/6x4m Hardstanding kit
3105	1	2	10	1-3105		DELIVERY
2146	1	1	195	5 1-2146		8 x 3 metre marquee
287	1	1	120	2 1-287		3 x 3 metre marquee
3419	1	1	250	8 1-3419		10 x 4 metre marquee
22	2	1	45	14 2-22		8x4m/10x4m/6x6m Hardstanding kit
3650	1	2	10	1-3650		DELIVERY
3557	3	1	40	16 3-3557		3 X3 metre matting
1553	4	2	5	4-1553		DELIVERY
16	1	2	10	1-16		DELIVERY
1584	4	1	20	31 4-1584		4 metre width partition wall

In the ‘Get and Transform’ section, let’s choose to create a new query ‘From Table’. We’ll call the new query **Apply\_Tax** and we are going to add a new column to this query in the ‘Add Column’ tab:

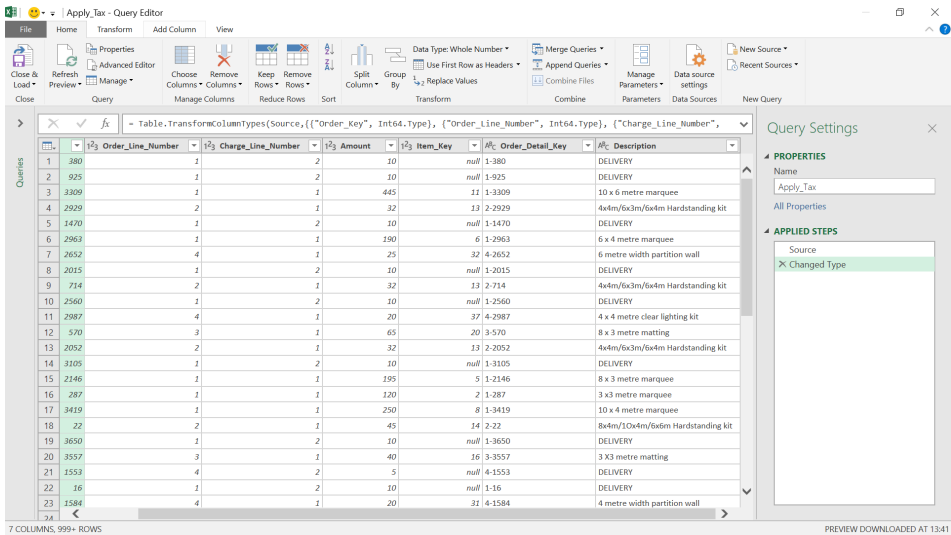


Table.TransformColumnTypes(Source,({"Order\_Key", Int64.Type), ("Order\_Line\_Number", Int64.Type), ("Charge\_Line\_Number", Int64.Type), ("Amount", Int64.Type), ("Item\_Key", Int64.Type), ("Order\_Detail\_Key", Int64.Type), ("Description", Text.Type)})

Order_Key	Order_Line_Number	Charge_Line_Number	Amount	Item_Key	Order_Detail_Key	Description
380	1	2	10	1-380		DELIVERY
925	1	2	10	1-925		DELIVERY
3309	1	1	445	11 1-3309		10 x 6 metre marquee
2929	2	1	32	13 2-2929		4x4m/6x3m/6x4m Hardstanding kit
1470	1	2	10	1-1470		DELIVERY
2963	1	1	190	6 1-2963		6 x 4 metre marquee
2652	4	1	25	32 4-2652		6 metre width partition wall
2015	1	2	10	1-2015		DELIVERY
714	2	1	32	13 2-714		4x4m/6x3m/6x4m Hardstanding kit
2560	1	2	10	1-2560		DELIVERY
2987	4	1	20	37 4-2987		4 x 4 metre clear lighting kit
570	3	1	65	20 3-570		8 x 3 metre matting
2052	2	1	32	13 2-2052		4x4m/6x3m/6x4m Hardstanding kit
3105	1	2	10	1-3105		DELIVERY
2146	1	1	195	5 1-2146		8 x 3 metre marquee
287	1	1	120	2 1-287		3 x 3 metre marquee
3419	1	1	250	8 1-3419		10 x 4 metre marquee
22	2	1	45	14 2-22		8x4m/10x4m/6x6m Hardstanding kit
3650	1	2	10	1-3650		DELIVERY
3557	3	1	40	16 3-3557		3 X3 metre matting
1553	4	2	5	4-1553		DELIVERY
16	1	2	10	1-16		DELIVERY
1584	4	1	20	31 4-1584		4 metre width partition wall

We will use the ‘Custom Column’ to create this new column.

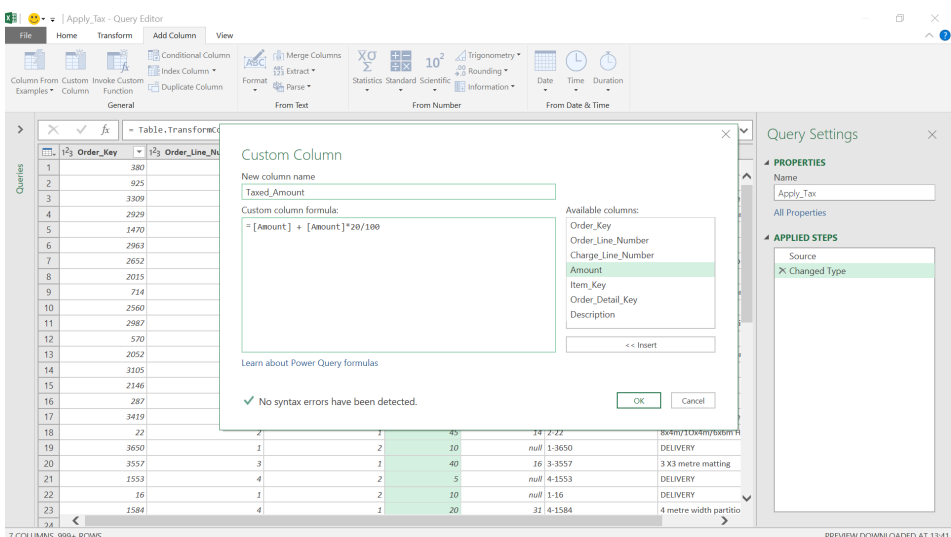


Table.TransformColumnTypes(Source,({"Order\_Key", Int64.Type), ("Order\_Line\_Number", Int64.Type), ("Charge\_Line\_Number", Int64.Type), ("Amount", Int64.Type), ("Item\_Key", Int64.Type), ("Order\_Detail\_Key", Int64.Type), ("Description", Text.Type), {"Taxed\_Amount", CustomColumn}))

Custom Column

New column name: Taxed\_Amount

Custom column formula: [Amount] \* [Amount] / 20 / 100

Available columns: Order\_Key, Order\_Line\_Number, Charge\_Line\_Number, Amount, Item\_Key, Order\_Detail\_Key, Description

Query Settings

PROPERTIES

Name: Apply\_Tax

APPLIED STEPS

Source

Changed Type

Let's choose to add 20%, and create the new column. We'll opt to change it to datatype 'Currency' for information; the formatting doesn't matter here, as we may deal with that in our Excel Worksheet later (this is not what Power Query does).

Index	Charge_Line_Number	Amount	Item_Key	Order_Detail_Key	Description	Taxed_Amount
1	1	2	10	1-380	DELIVERY	12
2	1	2	10	1-925	DELIVERY	12
3	1	1	445	11 1-3309	10 x 6 metre marquee	534
4	2	1	32	13 2-2929	4x4m/6x3m/6x4m Hardstanding kit	38.4
5	1	2	10	1-1470	DELIVERY	12
6	1	1	190	6 1-2963	6 x 4 metre marquee	228
7	4	1	25	32 4-2652	6 metre width partition wall	30
8	1	2	10	1-2015	DELIVERY	12
9	2	1	32	13 2-714	4x4m/6x3m/6x4m Hardstanding kit	38.4
10	1	2	10	1-2560	DELIVERY	12
11	4	1	20	37 4-2887	4 x 4 metre clear lighting kit	24
12	3	1	65	20 3-570	8 x 3 metre matting	78
13	2	1	32	13 2-2052	4x4m/6x3m/6x4m Hardstanding kit	38.4
14	1	2	10	1-3105	DELIVERY	12
15	1	1	195	5 1-2146	8 x 3 metre marquee	234
16	1	1	120	2 1-287	3 x3 metre marquee	144
17	1	1	250	8 1-3419	10 x 4 metre marquee	300
18	2	1	45	14 2-22	8x4m/10x4m/6x4m Hardstanding kit	54
19	1	2	10	1-3650	DELIVERY	12
20	3	1	40	16 3-3557	3 X3 metre matting	48
21	4	2	5	4-1553	DELIVERY	6
22	1	2	10	1-16	DELIVERY	12
23	4	1	20	31 4-1584	4 metre width partition wall	24

This month, we're interested in the M language generated behind the scenes. Let's access the 'Advanced Editor' from the 'Home' tab:

```

let
    Source = Excel.CurrentWorkbook(){[Name="ACCT_Order_Charges"]}[Content],
    #"Changed Type" = Table.TransformColumnTypes(Source,{{"Order_Key", Int64.Type}, {"Order_Line_Number", Int64.Type}, {"Charge_Line_Number", Int64.Type}, {"Added Custom" = Table.AddColumn(#"Changed Type", "Taxed_Amount", each [Amount] * [Amount]*20/100),
    #"Changed Type1" = Table.TransformColumnTypes(#"Added Custom",{{"Taxed_Amount", Currency.Type}})
in
    #"Changed Type1"
  
```

Copy and paste this text to a text file:

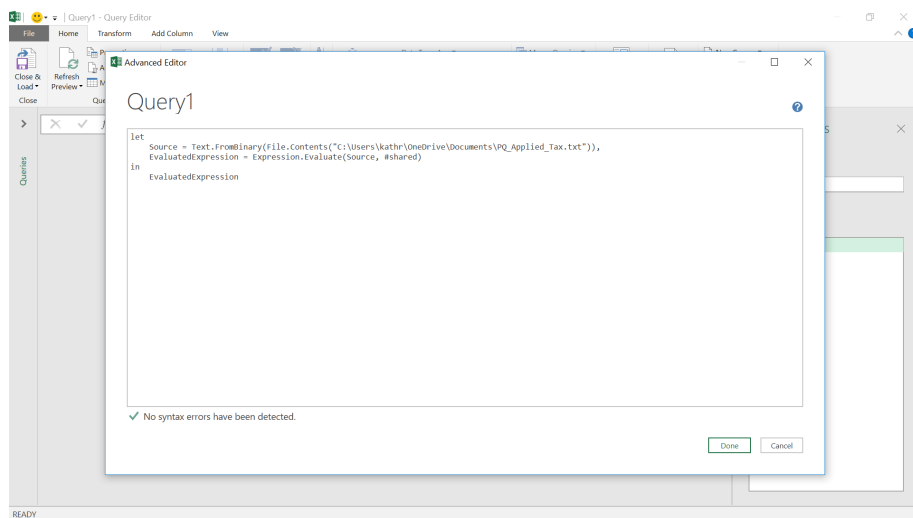
```

let
    Source = Excel.CurrentWorkbook(){[Name="ACCT_Order_Charges"]}[Content],
    #"Changed Type" = Table.TransformColumnTypes(Source,{{"Order_Key", Int64.Type}, {"Order_Line_Number", Int64.Type}, {"Charge_Line_Number", Int64.Type}, {"
    #"Added Custom" = Table.AddColumn(#"Changed Type", "Taxed_Amount", each [Amount] * [Amount]*20/100),
    #"Changed Type1" = Table.TransformColumnTypes(#"Added Custom",{{"Taxed_Amount", Currency.Type}})
in
    #"Changed Type1"
  
```

If you return to the original data, you may now use the M language stored in this text file. We'll discard the **Apply\_Tax** query and create a new Blank Query, by choosing 'From Other Sources' and then 'Blank Query' from the 'New Query' option in the 'Get and Transform' section.

Order	Charge_Line_Number	Amount	Item_Key	Order_Detail_Key	Description
1	2	10	1-380		DELIVERY
1	2	10	1-925		DELIVERY
3	1	445	11 1-3309		10 x 6 metre marquee
3	2	32	13 2-2929		4x4m/6x3m/6x4m Hardstanding kit
5	1	10	1-1470		DELIVERY
6	1	190	6 1-2963		6 x 4 metre marquee
7	4	25	32 4-2652		6 metre width partition wall
8	2	10	1-2015		DELIVERY
9	1	32	13 2-714		4x4m/6x3m/6x4m Hardstanding kit
10	2	10	1-2560		DELIVERY
11	4	20	37 4-2987		4 x 4 metre clear lighting kit
12	1	65	20 3-570		8 x 3 metre matting
13	2	32	13 2-2052		4x4m/6x3m/6x4m Hardstanding kit
14	1	10	1-3105		DELIVERY
15	1	195	5 1-2146		8 x 3 metre marquee
16	1	120	2 1-287		3 x 3 metre marquee
17	1	250	8 1-3419		10 x 4 metre marquee
18	2	45	14 2-22		8x4m/10x4m/6x6m Hardstanding kit
19	1	10	1-3650		DELIVERY
20	3	40	16 3-3557		3 X3 metre matting
21	5	5	4-1553		DELIVERY
22	1	10	1-16		DELIVERY
23	4	20	31 4-1584		4 metre width partition wall

We need to enter the **M** language below in order to run the code in this text file:



**let**  
**Source=Text.FromBinary(File.Contents("C:\Users\kathr\OneDrive\Documents\PQ\_Applied\_Tax.txt")),**  
**EvaluatedExpression = Expression.Evaluate(Source, #shared)**  
**in**  
**EvaluatedExpression**  
 We'll create this new query and check the results.

Index	Charge_Line_Number	Amount	Item_Key	Order_Detail_Key	Description	Taxed_Amount
1	2	10	1-380		DELIVERY	12
2	2	10	1-925		DELIVERY	12
3	1	445	11 1-3309		10 x 6 metre marquee	534
4	2	32	13 2-2929		4x4m/6x3m/6x4m Hardstanding kit	38.4
5	1	10	1-1470		DELIVERY	12
6	1	190	6 1-2963		6 x 4 metre marquee	228
7	4	25	32 4-2652		6 metre width partition wall	30
8	2	10	1-2015		DELIVERY	12
9	1	32	13 2-714		4x4m/6x3m/6x4m Hardstanding kit	38.4
10	2	10	1-2560		DELIVERY	12
11	4	20	37 4-2987		4 x 4 metre clear lighting kit	24
12	1	65	20 3-570		8 x 3 metre matting	78
13	2	32	13 2-2052		4x4m/6x3m/6x4m Hardstanding kit	38.4
14	1	10	1-3105		DELIVERY	12
15	1	195	5 1-2146		8 x 3 metre marquee	234
16	1	120	2 1-287		3 x 3 metre marquee	144
17	1	250	8 1-3419		10 x 4 metre marquee	300
18	2	45	14 2-22		8x4m/10x4m/6x6m Hardstanding kit	54
19	1	10	1-3650		DELIVERY	12
20	3	40	16 3-3557		3 X3 metre matting	48
21	5	5	4-1553		DELIVERY	6
22	1	10	1-16		DELIVERY	12
23	4	20	31 4-1584		4 metre width partition wall	24

The query has been executed correctly and we may easily share the text on an accessible drive for other users.  
 More next month!

# Power BI Updates

This month sees updates for Q&A, shapes and automatic aggregations for DirectQuery, amongst other inclusions. The full list reads as follows:

## Reporting

- Customise shape formatting
- x-axis constant line improvements
- Default sensitivity label policy in Power BI Desktop (Preview)

## Analytics

- Sharing Q&A synonyms

## Modelling

- New way of expressing Date and DateTime values
- Evaluation configuration settings

## Data Connectivity and Preparation

- One Dataflow connector (connector update)
- Automy Data Analytics (new connector)
- Azure Databricks (connector update)
- Dremio (connector update)
- Google BigQuery (connector update)
- Quickbase (connector update)

## Service

- Sensitivity labels for paginated reports (Preview)
- Leverage automatic aggregations to enhance performance over DirectQuery datasets
- Power BI REST API support for DAX Queries (Preview)
- Detailed historical activity information for Premium datasets (Preview)

## Mobile

- New mobile report footer

## Visualisations

- New Power BI visuals store experience
- New visuals
- Timeline by Queryon
- Drill Down Graph PRO by ZoomCharts
- Blpm custom visual by ProcessM.

Let's go through each in turn.

## Customise shape formatting

You can now customise the formatting of the shapes added to your Power BI reports. This update sees a number of controls inserted into the Shape card of the Formatting pane for a variety of different Shape

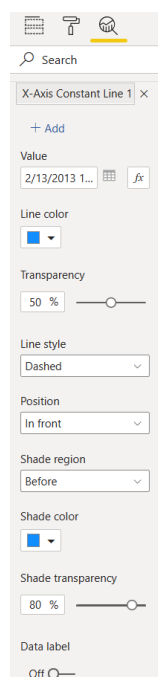
options, including "rounded" rectangles, chevrons, arrows, and more. You may now customise the look of each of the shapes you create, allowing more flexible designs and structures.



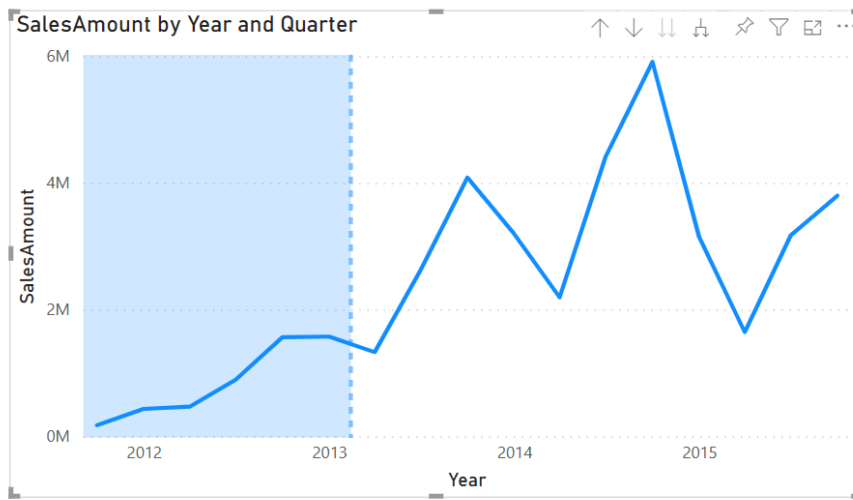
## x-axis constant line improvements

Since the x-axis constant line was introduced, Microsoft has advised they have received much constructive feedback. As a result, conditional formatting for the x-axis constant line value and shading for the regions

before or after the constant line have been added. You can find these new options in the x-axis constant line card in the Analytics pane.



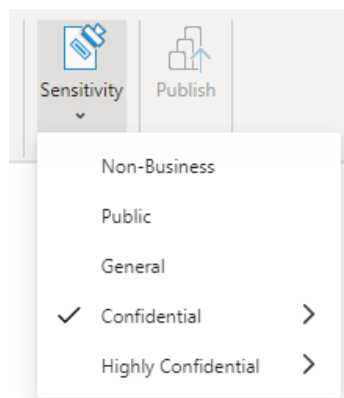




### Default sensitivity label policy in Power BI Desktop (Preview)

Security administrators may define a default label to be set on a PBIX file when a user creates a new file or edits an existing file that has no label. Users may also change the default label if they see fit. Moving forward,

default label policies will be applicable to new content in the Power BI Service as well.



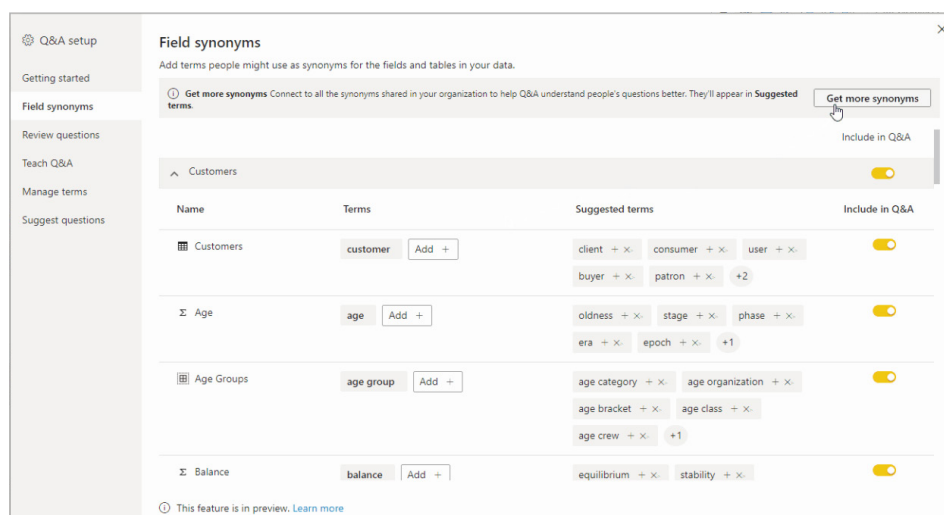
### Sharing Q&A synonyms

Field synonyms are an important part of Q&A because users often have a variety of terms that they use to refer to the same field, such as total sales, net sales or total net sales. This update now supports sharing

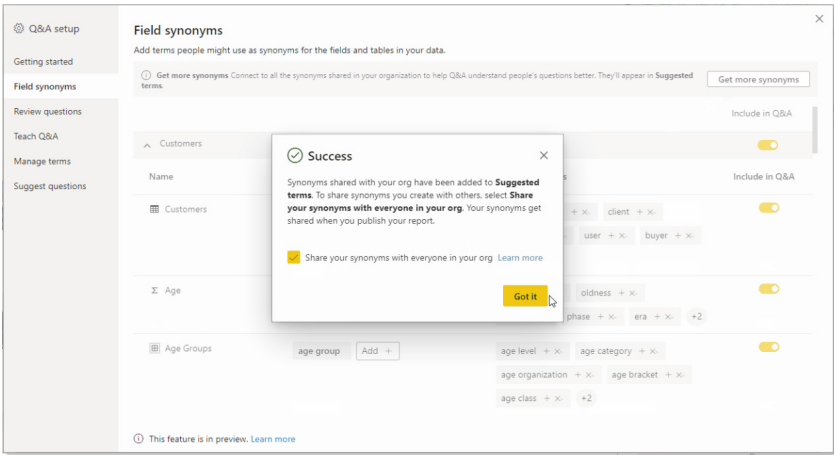
synonyms between everyone in your organisation. This means that you can now leverage your co-workers' synonyms and minimise duplicate effort when adding synonyms to your models.

### HOW TO GET SHARED SYNONYMS

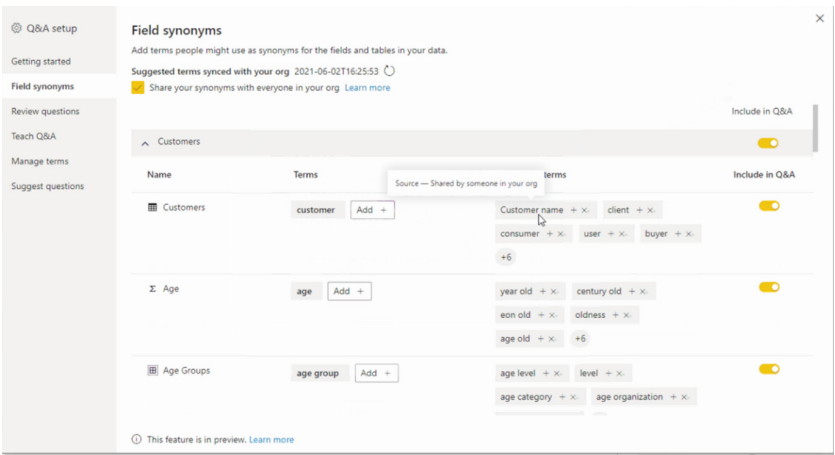
To connect to shared synonyms, navigate to **Q&A setup -> Field synonyms** and click on 'Get more synonyms'.



This process may take a few seconds, but once it's completed, you also have the option to share your synonyms with everyone else in your workplace too. It should be noted that the synonyms are shared when you publish the report.




Any shared synonyms that match with fields in your model will show up in the 'Suggested terms' section. Then, you may click the plus icon (+) on the suggested term to add it as a field synonym.




As new synonyms get shared within your organisation, you can click the Refresh icon to ensure that your 'Suggested terms' are in-sync with your organisation, viz.

### Field synonyms

Add terms people might use as synonyms for the fields and tables in your data.

Suggested terms synced with your org 2021-06-02T16:25:53 

 Share your synonyms with everyone in your org [Learn more](#)

## HOW TO CHANGE YOUR SHARING SYNONYMS SETTINGS

In Power BI Desktop, you may update your sharing synonyms settings in the Q&A setup itself or within the Options dialog, under **Data Load -> Q&A -> Share your synonyms with everyone in your org**:

### Options

GLOBAL

Data Load

Power Query Editor

DirectQuery

R scripting

Python scripting

Security

Privacy

Regional Settings

Updates

Usage Data

Diagnostics

Preview features

Auto recovery

Report settings

CURRENT FILE

Data Load

Regional Settings


Privacy


Auto recovery


Type Detection

☒ Detect column types and headers for unstructured sources

Relationships


☒ Import relationships from data sources on first load 

☐ Update or delete relationships when refreshing data 

☒ Autodetect new relationships after data is loaded 

[Learn more](#)


Time intelligence

☒ Auto date/time  [Learn more](#)


Background Data

☒ Allow data previews to download in the background

Parallel loading of tables

☒ Enable parallel loading of tables 

Q&A

☒ Turn on Q&A to ask natural language questions about your data  [Learn more](#)

☒ Share your synonyms with everyone in your org

OK Cancel

You can also update this setting in the 'Dataset settings' within Power BI Service:

**Datasets** Workbooks

### Settings for Hotel Reviews + Drivers updated

[Refresh history](#)

- ▶ Gateway connection
- ▶ Data source credentials
- ▶ Sensitivity label
- ▶ Parameters
- ▶ Scheduled refresh
- ◀ Q&A

① If Q&A is turned on, we'll create an index of your data in the service. [Learn more](#)

☒ Turn on Q&A to ask natural language questions about your data

☒ Share your synonyms with everyone in your org

Apply Discard

Additionally, administrators may choose to disable Synonym sharing within the Admin Portal by navigating to **Tenant settings -> Q&A settings**.

### ***New way of expressing Date and DateTime values***

Microsoft is introducing a new way to express Date and DateTime-typed values as what is known as a *DAX literal*.

This will allow you to directly specify dates and times (up to the second) in your DAX queries, without having to use other functions. Starting with this latest release, Power BI now supports either a complete date value or a complete date and time value. The syntax is as follows:

Date format: **dt"YYYY-MM-DD"**

For example, **dt"1999-12-31"** would represent 31 December 1999.

Furthermore, the DateTime format is as follows:

DateTime format: **dt"YYYY-MM-DDThh:mm:ss"** or

DateTime format: **dt"YYYY-MM-DD hh:mm:ss"**

For examples, **dt"2021-05-24T12:00:00"** and **dt"2021-05-24 12:00:00"** would both represent noon on 24 May 2021.

It should be noted that in these instances, DAX supports valid date / time components with fewer digits than indicated for ease of use, e.g. January may be specified as either "1" or "01".

To show how this makes things simpler, consider the following example, where in the past you may have filtered sales to within a specific order date range by using the following DAX code:

**EVALUATE**

**FILTER (Sales, [OrderDate] > (DATE(2015,1,9) + TIME(2,30,0)) && [OrderDate] < (DATE(2015,12,31) + TIME(11,59,59)))**

This now simplifies to

**EVALUATE**

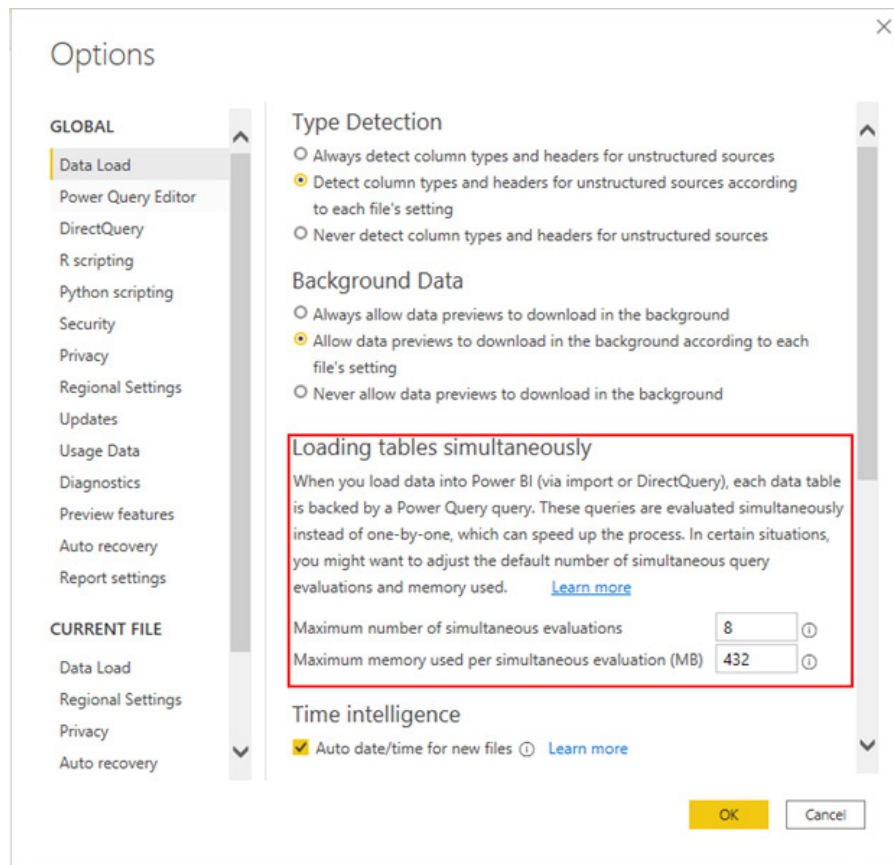
**FILTER (Sales, [OrderDate] > dt"2015-1-9T02:30:00" && [OrderDate] < dt"2015-12-31T11:59:59").**

### ***Evaluation configuration settings***

Power BI Desktop optimises query performance when importing data or when using DirectQuery by evaluating tables simultaneously. However, in specific situations, you might want to influence the behaviour and change the defaults. For example, you might wish to modify when the

data import is taking too long or Power BI Desktop is taking too much resource on the machine. Until now, you could only influence these settings by making changes to the registry. Starting with this update, there are now two configuration options to Power BI Desktop:

1. **Maximum number of simultaneous evaluations:** this configures the level of parallelisation for query execution in Power Query
2. **Maximum memory used per simultaneous evaluation:** this configures the available memory per evaluation.



Using these settings, you can make sure the loading of data is optimised for your machine, so you get the most pragmatic experience.

### ***One Dataflow connector (connector update)***

There is an update to Dataflow Connectors in Power BI, together with the consolidation of the two existing Dataflow connectors into a single Dataflows connector. Dataflow users will now be able to leverage a single connector to discover, navigate and get data from dataflows created in either Power BI workspaces or Power Apps Environments. In addition, the new connector will provide Power BI Premium dataflow

customers with improved performance when authoring dataflows or datasets by leveraging Power BI's enhanced compute engine. The Dataflows connector may significantly reduce evaluation time for steps performed over computed entities such as performing joins, distinct, filters and group by operations. There are two specific benefits:

1. Downstream users connecting to this new Dataflows connector in Power BI Desktop may take advantage of better performance in authoring scenarios because the new connector supports query folding
2. Dataset refresh operations can also fold to the enhanced compute engine, meaning even incremental refresh from a dataset can fold to a dataflow, improving refresh performance and potentially decreasing latency between refresh cycles. All you have to do is ensure the compute engine is explicitly set to 'On'.

As part of the consolidation into one connector, the Previously named 'Power Platform Dataflows' connector will be renamed to 'Dataflows' connector, and the 'Power BI Dataflows' connector will be renamed to 'Power BI Dataflows (Legacy)' connector, which will eventually be removed from the 'Get Data' menu.

All existing Dataflows and Datasets created using either connector will continue to function without interruption, and will not be impacted by

the name changes. No action is required to maintain their long-term function ability. When creating new Datasets from Dataflows, Microsoft recommends you select the 'Dataflows' connector in 'Get Data' in Power BI Desktop under the 'Power Platform' category. If you have existing datasets connected to Dataflows, you might consider trying out the new connector to take advantage of the improvements in this initial release.

## Automy Data Analytics (new connector)



The Automy Get Data connector has been released, which extends the reporting capabilities of Automy and allows you to capture it within your Power BI architecture for sharing with colleagues. Automy provides integrated software and data solutions for process automation. It enables

you to combine multiple data from workflows and processes into a single virtual data layer on-demand without copying or moving data.

This connector is located in the 'Online Services' category of the 'Get Data' dialog.

## Azure Databricks (connector update)

The optional flag 'Fast Evaluation' improves refresh performance of DirectQuery reports and imports. This is available for Databricks SQL

and for Databricks Runtime versions, starting at 8.3 and later.

## Dremio (connector update)

The connector has been updated to Version 1.0.7. This connector has now added support for connecting to Dremio Cloud. Furthermore, it shows up as a separate connector along with the previous connector for software now renamed as Dremio Software. The methods of

authentication available for Dremio Cloud are Azure Active Directory and Personal Access Token, whilst the method of authentication for Dremio Software remains unchanged.

## Google BigQuery (connector update)

This update sees support for native database queries for the Google BigQuery connector. Similar to SQL and Snowflake connectors, this will allow you to input a Google BigQuery native query and build reports on

top of it. In this initial version, you will need to specify a Project ID and use fully qualified table names of the format **ProjectID.Schema.Table**, e.g. **SELECT \* FROM ProjectID.Schema.Table**.

Google BigQuery

Advanced options

Billing Project ID (optional)  
Example: MyBillingProject

Use Storage Api (optional)  
Example: true

Connection timeout duration  
Example: 0.00:01:02

Command timeout duration  
Example: 0.00:01:02

Project ID (optional)  
Example: project-id

SQL statement (optional, requires Project ID)  
Example: select \* from project-id.demo.table

OK Cancel

## Quickbase (connector update)

This revision sees a new version of the Quickbase connector introduced.

Firstly, the logo and company name have been updated to a new brand identity.

Secondly, the new connector will use the more modern RESTful API. This API has been designed from the ground up to ensure a consistent experience. As a result of this upgrade, several backend performance improvements and bug fixes have been developed. Functionally, support

for Custom Fields that are defined in reports has also been added. As new capabilities are added to the RESTful API, they will continue to be passed through the updated connector.



### Sensitivity labels for paginated reports (Preview)

Now you can set MIP sensitivity labels on paginated reports in the Power BI Service, just like you can in Power BI Desktop reports. When data from

a paginated report is exported to a Word, Excel, PowerPoint or a PDF file, the report's label and protection settings be applied to the exported file.

### Leverage automatic aggregations to enhance performance over DirectQuery datasets

Automatic aggregations enhance performance over DirectQuery datasets by automatically creating aggregations, based upon your historic query patterns. Using machine learning, automatic aggregations determine, create and maintain the most optimal set of aggregations for

your workload. You no longer need to have extensive data-modelling or advanced query optimisation experience in order to leverage the power of this feature with just a few clicks.

### Power BI REST API support for DAX Queries (Preview)

There is a Preview of the new REST API to query datasets in Power BI by using Data Analysis Expressions (DAX). Amongst other things, this new DAX REST API helps to address feedback concerning programmatic access to the data in a dataset (e.g. the idea REST API should have access to read datasets). The new DAX REST API avoids dependencies on Analysis Services client libraries, requires no connection to XMLA endpoints and works in Power BI Premium, as well as in Power BI shared capacity.

One of the advantages of a REST API to query datasets is that you may use this REST API in almost any modern development environment on any platform, including low-code no-code Power Apps, Power Automate,

Logic Apps, JavaScript-based languages, PowerShell, Java, PHP, Ruby, Python or any other technology that can authenticate against Azure Active Directory (AAD) and construct a Web request. You may also use .NET to call this REST API. As with other Power BI REST APIs, the DAX REST API supports user accounts, service principals and works in B2B scenarios. Since the caller is fully identifiable, row-level security and other controls at the dataset level are applied as expected too.

You should note that the tenant-level setting Allow XMLA endpoints and 'Analyze in Excel' with on-premises datasets must be enabled in the Power BI Admin Portal.

#### Integration settings

Allow XMLA endpoints and Analyze in Excel with on-premises datasets

Enabled for the entire organization

Users in the organization can use Excel to view and interact with on-premises Power BI datasets. This also allows connections to XMLA endpoints.

Enabled

Apply to:

☒ The entire organization

☐ Specific security groups

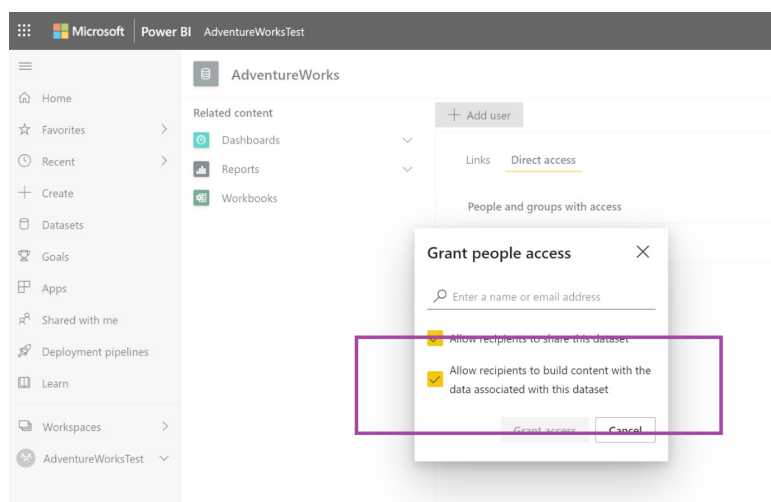
☐ Except specific security groups

Apply

Cancel

This setting is enabled by default, but you should double-check with your Power BI administrator. At least for the duration of the Preview period, Microsoft believed it seemed appropriate to govern this new DAX REST API with the existing XMLA endpoints tenant setting: the DAX query functionality is comparable to XMLA read access.

As well as XMLA read access, you will also require the Build permission for the dataset you want to query by using this new REST API. Contributors and workspace administrators have this permission, but you can also grant it directly.



You should also note that the workspace must be a modern (v2) workspace. Classic (legacy) workspaces are not supported. Therefore, if your datasets still reside in a classic workspace, make sure you upgrade the workspace to the new (v2) workspace experience first.

One way to test the DAX REST API is to query a dataset by using the Power BI Management cmdlets. As an example, consider Microsoft's much used *AdventureWorks* dataset:

```

1 $requestUrl = "datasets/88f9df14-be84-4473-a3d1-55ac7c8d1ff4/executeQueries"
2
3 $requestBody = @"
4 {
5   "queries":
6   [
7     {
8       "query": "EVALUATE SUMMARIZECOLUMNS(
9         ROLLUPADDISSUBTOTAL(
10          ROLLUPGROUP('DimCustomer'[LastName], 'DimProduct'[ModelName]), \"IsGrandTotalRowTotal\"
11        ),
12        TREATAS ( {\"Adams\"}, DimCustomer[LastName]),
13        \"SumSalesAmount\", CALCULATE(SUM('FactInternetSales'[SalesAmount]))
14      )\"
15    }
16  ]
17  \"includeNulls\": false
18 }
19 @"
20 Login-PowerBI
21 $result = Invoke-PowerBIRestMethod -Method Post -Url $requestUrl -Body $requestBody
22
23 $parsed = $result | ConvertFrom-Json
24 $parsed.results[0].tables[0].rows | Format-List

```

```

DimCustomer[LastName] : Adams
DimProduct[ModelName] : Touring Tire
[IsGrandTotalRowTotal] : False
[SumSalesAmount] : 86.97

DimCustomer[LastName] : Adams
DimProduct[ModelName] : Touring-1000
[IsGrandTotalRowTotal] : False
[SumSalesAmount] : 9536.28

DimCustomer[LastName] : Adams
DimProduct[ModelName] : Touring-3000
[IsGrandTotalRowTotal] : False
[SumSalesAmount] : 1484.7

DimCustomer[LastName] : Adams
DimProduct[ModelName] : Road-350-W
[IsGrandTotalRowTotal] : False
[SumSalesAmount] : 10205.94

DimCustomer[LastName] : Adams
DimProduct[ModelName] : Road-750
[IsGrandTotalRowTotal] : False
[SumSalesAmount] : 2159.96

DimCustomer[LastName] : Adams
DimProduct[ModelName] : Mountain-400-W

```

In summary, the above script performs the following main steps:

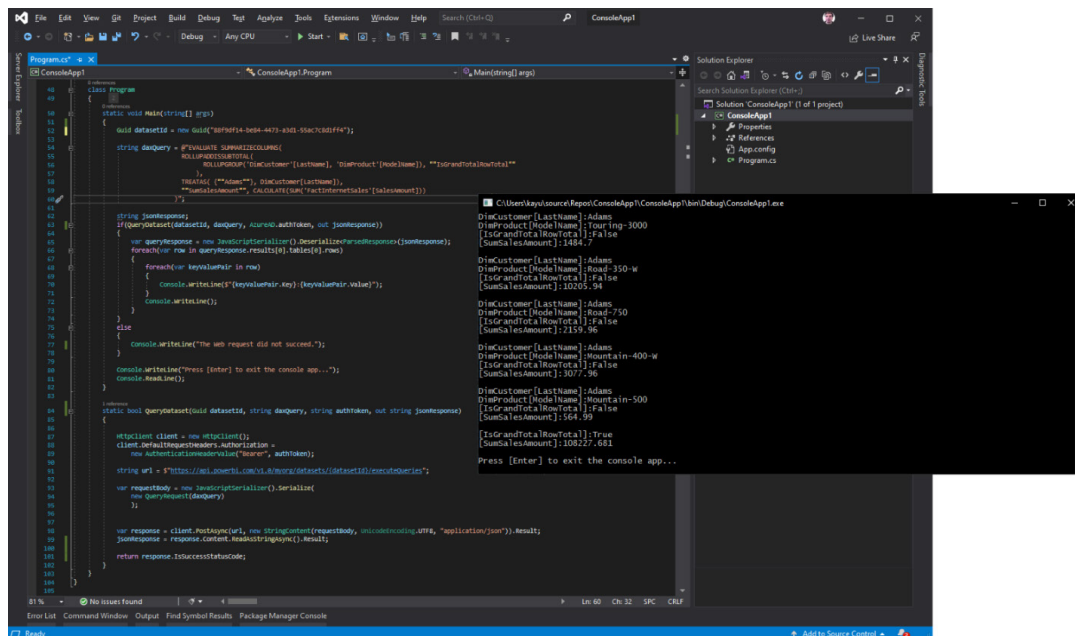
- construct the request parameters; this includes the request URL and the request body
- the request URL references **executeQueries** endpoint of the dataset you want to query. The dataset ID (**id**) identifies the dataset. There are many ways to determine the dataset ID. Perhaps the easiest is to display the dataset settings in the browser and then copy the dataset Id from the URL in the address bar
- the URL has the format: **https://api.powerbi.com/v1.0/myorg/datasets/{datasetId}/executeQueries**. If you use the Power BI Management cmdlets, you also can work with relative URLs, as in the script above
- the request body is a JSON document containing the DAX query you want to run and an optional parameter to specify if blanks should be returned as *null* values or not. The default omits blanks, which helps to reduce the data volume. More importantly, you should note that the request body consists of a queries array, but currently you can only specify a single query. The queries array provides flexibility for future improvements, although presently it can only contain a single query item. As a template for the request body, replace '**<Your DAX Query>**' with your actual DAX expression and don't forget to escape quotation marks with a backslash ("**\**")

**{ "queries": [{"query": "<Your DAX Query>"}, {"serializerSettings": {"includeNulls": true}}]**

- send the request to Power BI. The DAX REST API expects a POST request, which you can construct in PowerShell by using the **Invoke-PowerBIRestMethod** cmdlet. Don't forget to login to Power BI and then specify Post as the Method and pass the request URL and request body as demonstrated above. The Power BI Management cmdlets have the required app permissions to use the DAX REST API
- if you want to use your own app registration, make sure your app has the **Dataset.Read.All** or **Dataset.ReadWrite.All** permission
- parse the JSON response and process the results. The DAX REST API returns a JSON document with a results array containing one result per input query. Because the API currently only supports a single DAX query, the results array will also only include a single item. This item might include one or multiple tables, depending upon what the DAX query returns, which in turn contain the rows with the actual results as key / value pairs with the keys referring to the column names.

In the script above, the line **\$parsed.results[0].tables[0].rows | Format-List** outputs the rows of the first (and only) table of the first (and only) result item returned for the (one and only) DAX query.

A .NET implementation that produces the same results as the PowerShell script above is equally uncomplicated (*see below*).



Do remember that the request body currently can only contain one DAX query and accordingly there is only one result set. Future versions of the API might support more. Also, you should note that the result set is currently capped at 100,000 rows. If you need to retrieve more rows, you must construct and submit multiple DAX queries that each only retrieves an appropriate portion of the results. Moreover, the result set does not support binary data. Make sure your values are of the type string, numeric, Boolean, blank, Date Time or variant. Furthermore, this is a Power BI API and therefore not available in Azure Analysis Services or SQL Server Analysis Services. Also, because this is a DAX REST API, you cannot submit MDX queries.

While it is certainly not too difficult to construct a web request manually, Microsoft is also aiming at adding the DAX REST API to the Power BI .NET SDK and the Power BI Connector for Power Automate, Power Apps and Logic Apps so that it is even easier to leverage the data from your Power BI datasets in your business solutions. For example, you could then query certain key performance indicators in Power Automate and trigger appropriate subprocesses based on the results.

### Detailed historical activity information for Premium datasets (Preview)

Workspace administrators can now enable collection of Power BI user activities and metrics for any Premium workspaces they own. This Preview capability provides near real-time information reported by the Analysis Services engine about dataset activity. This allows you to monitor load, observe usage trends, spot periods of unusual activity, and identify performance degradation with queries or dataset refresh operations.

Microsoft has partnered with Azure Log Analytics to allow you to send your Power BI activity data to your own Azure environment. This gives you up to two years of data retention with ad-hoc query and alerting.

The Power BI Service administrator can enable this capability for Workspaces in Tenant Settings:

### Admin portal

- Usage metrics
- Users
- Premium Per User
- Audit logs
- Tenant settings
- Capacity settings
- Refresh summary
- Embed Codes
- Organizational visuals
- Azure connections
- Workspaces
- Custom branding
- Protection metrics
- Featured content

### Connect to Azure resources (preview)

- ▶ Tenant-level storage (preview)
- ▶ Workspace-level storage permissions (preview)

#### ◀ Workspace-level Log Analytics permissions (preview)




Allow workspace admins to connect their own Log Analytics workspace

Save

Cancel

A workspace administrator can then set up logging within 'Workspace Settings'. The example (below) shows a completed configuration:

 Settings


v2 MSIT Testing

AboutPremiumAzure connections

Storage


Log Analytics

Connect an Azure Log Analytics workspace to collect usage and performance logs for this workspace. [Learn more](#)

**Subscription** 

**Resource group** rg-byola

**Log Analytics workspace** bhmerc-msit-ws-0407

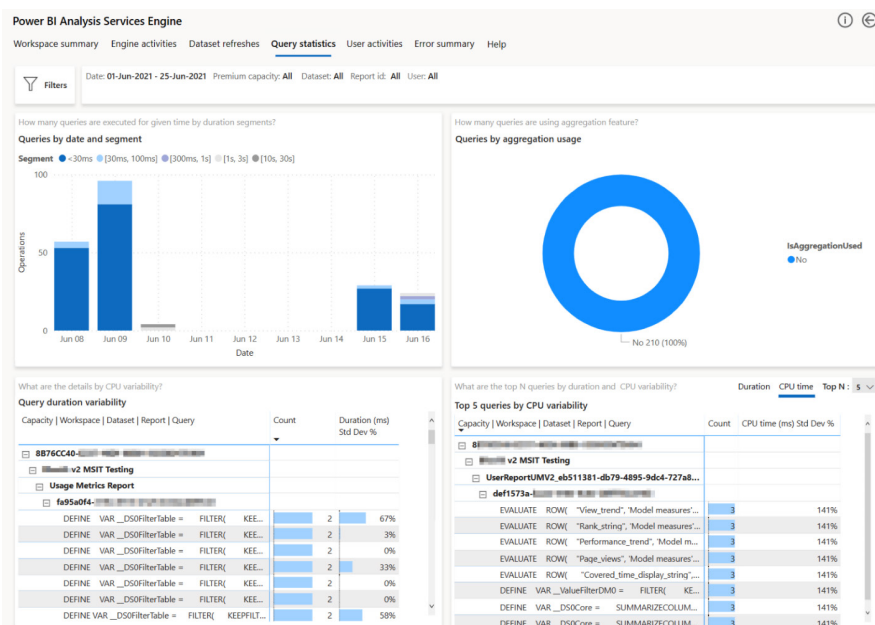
Configured by  on 2021-04-07T12:00:00Z

**Disconnect from Azure**

Disconnecting will stop usage and performance data from flowing into your Azure Log Analytics workspace.

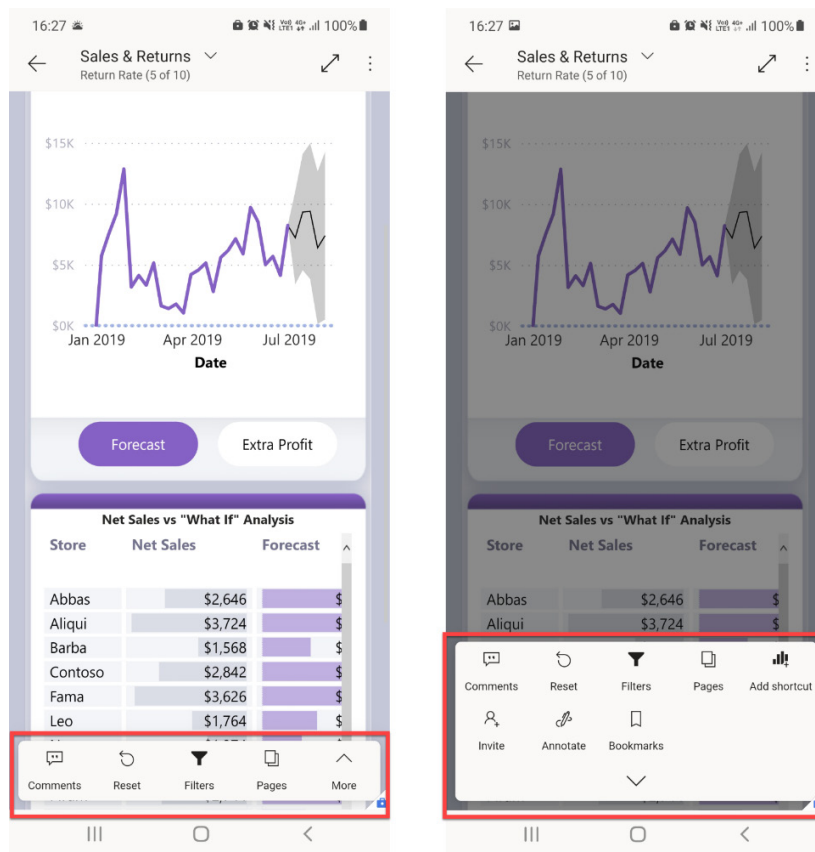
Disconnect from Azure

In the supporting documentation for this month's update, Microsoft has published a template report to allow you to get insights on your activity data right away, without having to understand how to parse and transform the log data. The example below shows the Query Statistics page which can be used to identify queries with inconsistent performance:



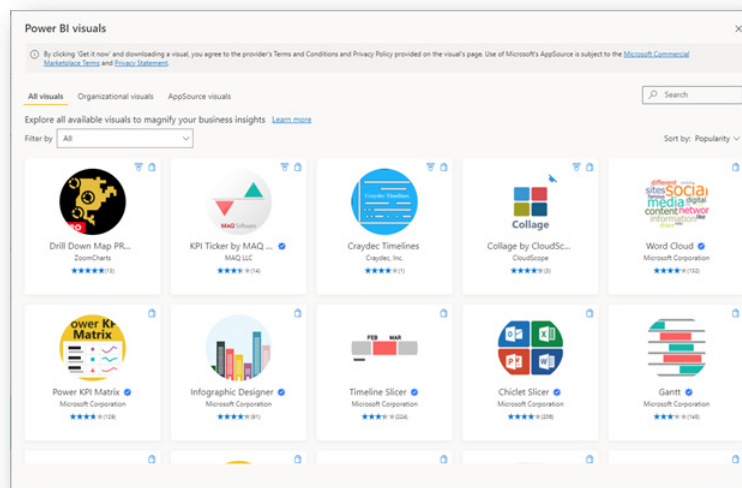
## New mobile report footer

This edition has updated the report footer to make it easier to find and understand all the actions you can perform with your reports.



### New Power BI visuals store experience

Power BI visuals' embedded store has been given a new look, where you can browse all visuals allowed for your organisation.



### New visuals

New visuals this month include:

- Floor Plan Visual by Simpson Associates
- Timeline Box by TheraTraq
- Timeline Basic by TheraTraq
- Phrazor by vPhrase Analytics Solutions Pvt. Ltd
- Balance Sheet Visual by K Team Solutions GmbH
- Drill Down Map PRO by ZoomCharts
- Craydec Timelines
- KPI Ticker by MAQ Software
- Collage by CloudScope.



## Timeline by Queryon

Timeline by Queryon makes the single-axis timeline flexible and provides you with useful visualisations. You may use Timeline to visualise project milestones, key events and project finish times.

## Drill Down Graph PRO by ZoomCharts

This interactive graphic allows you to control column sequences, add subtotals and drill down into each category with just a click. It supports visualising totals, subtotals and column sequences.

Features include:

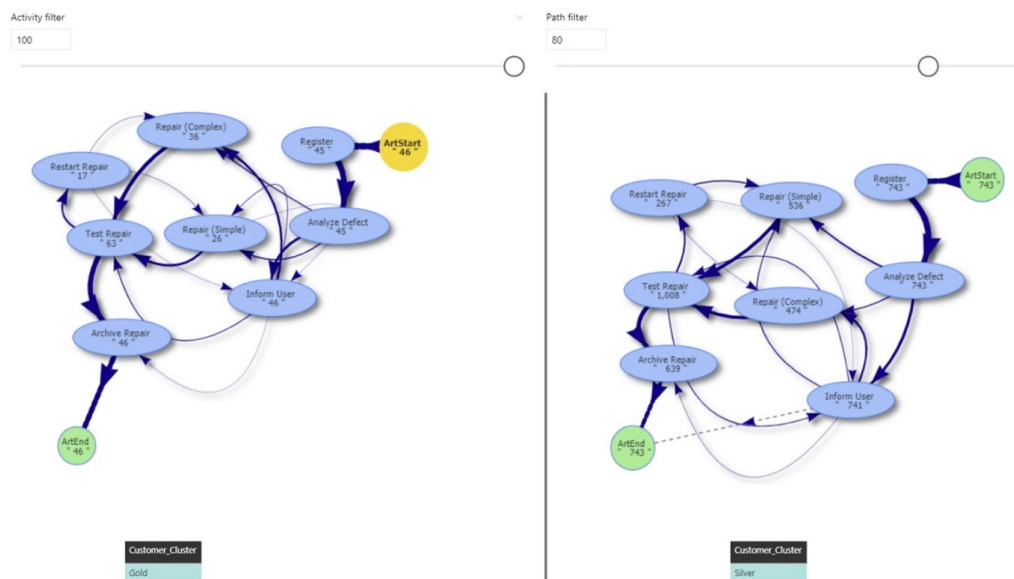
- **touch-driven slicer:** you may filter the report page by using the visual itself (i.e. there is no need for external slicers)
- **on-chart interactions:** zoom, click and drag or drill down to explore and filter data
- **display of total:** toggle on / off Total column
- **sub-totals:** set display values in the dataset or let the visual calculate them automatically
- **various customisation options:** customise increasing, decreasing and totals series separately (i.e. colours, outlines, column widths, connectors, value labels and more)
- **static and dynamic thresholds:** set up to four thresholds to demonstrate targets or benchmarks
- **mobile friendly:** use on touch and multi-touch devices.

To evaluate Drill Down Graph PRO for free, there is a 30-day trial period available.

## Blpm custom visual by ProcessM

For this update, ProcessM has introduced the Blpm custom visual. It's an independent visual undertaking process mining to investigate real behaviours over your process data. This custom visual is intended as a simple, interactive and user-friendly solution to study process models over multidimensional event logs in Power BI, thereby having a reliable platform for comprehensive enterprise mining.

As an example of many practical analytical tasks by Blpm, you can compare the various process models of the input data's multi aspects by setting the different filters. For instance, the following figure shows the processes for gold and silver customers separately:

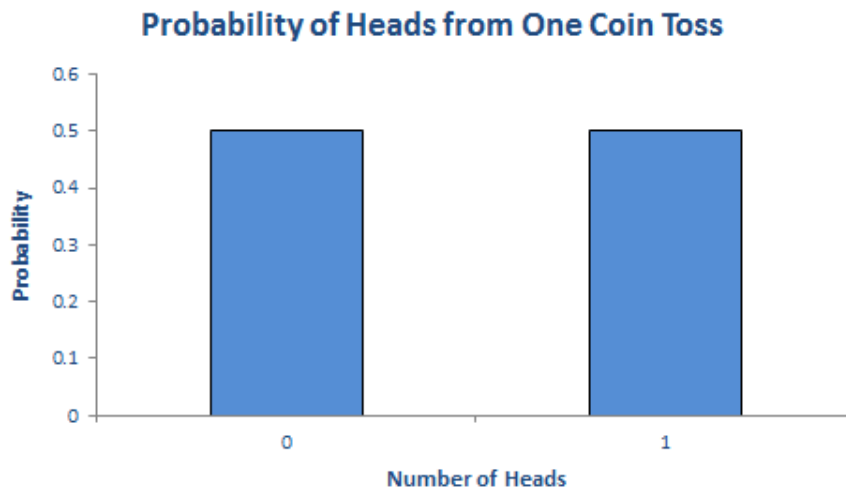


You may evaluate the process mining now on your process data (event log) by visiting AppSource and downloading Blpm.

More next month, we're sure!

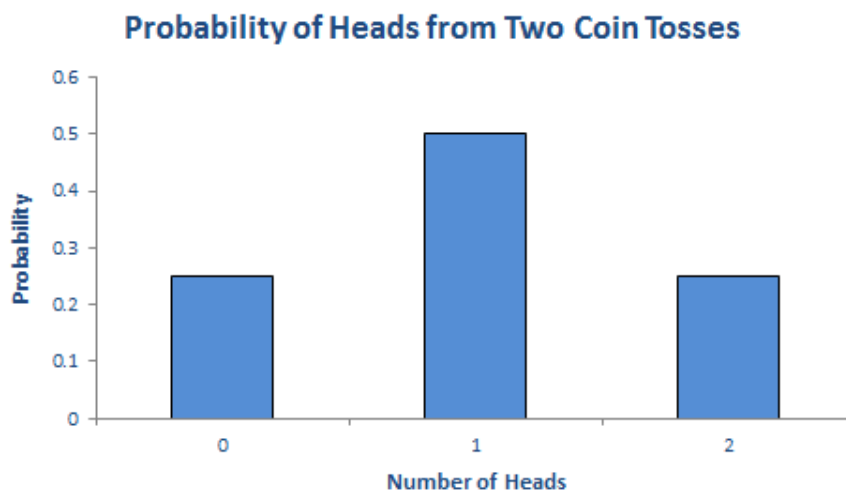
# The A to Z of Excel Functions: GAUSS

Let's start simply. Imagine you toss an unbiased coin: half of the time it will come down heads, half tails:



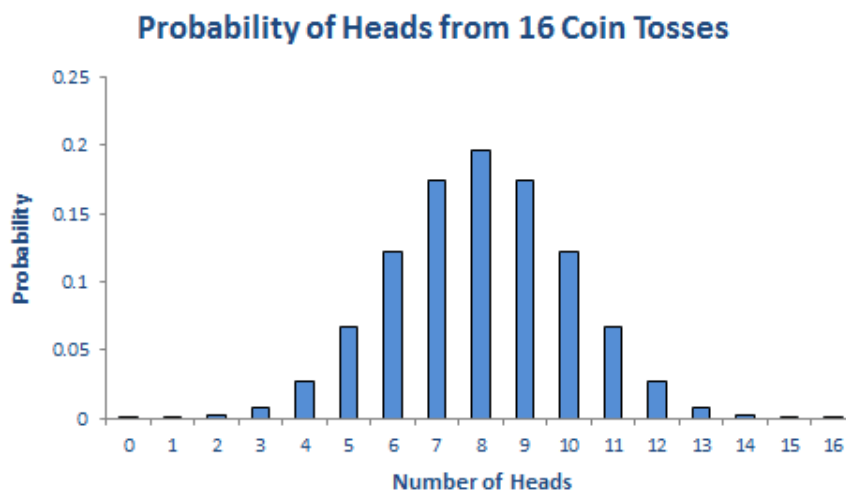
It is not the most exciting chart ever constructed, but it's a start.

If you toss two coins, you get four possibilities: two heads, a head and a tail, a tail and a head, and two tails.



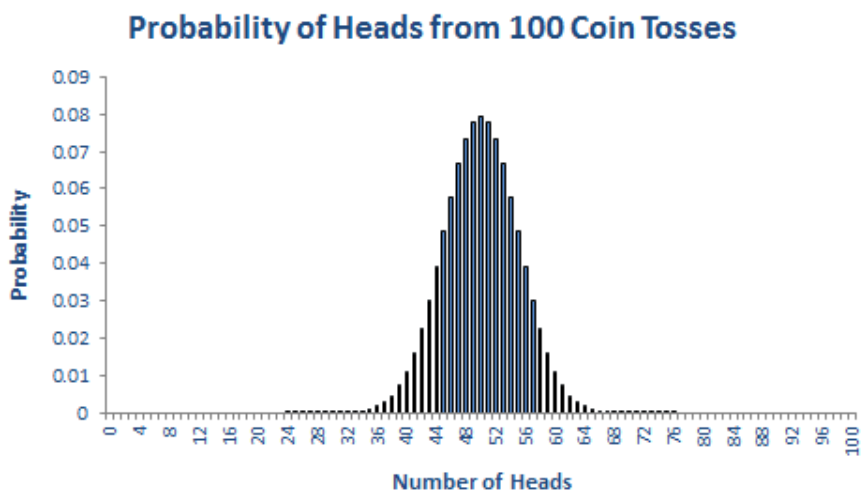
In summary, we should get two heads a quarter of the time, one head half of the time and no heads a quarter of the time. Note that  $(1/4) + (1/2) + (1/4) = 1$ . These fractions are the probabilities of the events occurring and the sum of all possible outcomes must always add up to one [1].

The story is similar if we consider 16 coin tosses say:



Again, if you were to add up all of the individual probabilities, they would total to one [1]. Notice that in symmetrical distributions (such as this one) it is common for the most likely event (here, eight heads) to be the event at the midpoint.

Of course, why should we stop at 16 coin tosses?



All of these charts represent **probability distributions**, *i.e.* it displays how the probabilities of certain events occurring are distributed. If we can formulate a probability distribution, we can estimate the likelihood of a particular event occurring (*e.g.* probability of precisely 47 heads from 100 coin tosses is 0.0666, probability of less than or equal to 25 heads occurring in 100 coin tosses is  $2.82 \times 10^{-7}$ , *etc.*).

Now, try to verify this last chart. Assuming you can toss 100 coins, count the number of heads and record the outcome at one coin toss per second, it shouldn't take you more than  $4.0 \times 10^{22}$  **centuries** to generate every permutation. Even if we were to simulate this experiment using a computer programme capable of generating many calculations a second

it would not be possible. For example, in February 2012, the *Japan Times* announced a new computer that could compute 10,000,000,000,000,000 calculations per second. If we could use this computer, it would only take us a mere 401,969 years to perform this computation. Sorry, but we can't afford the electricity bill.

Let's put this all into perspective. All we are talking about here is considering 100 coin tosses. If only business were that *simple*. Potential outcomes for a business would be much more complex. Clearly, if we want to consider all possible outcomes, we can only do this using some sampling technique based on understanding the underlying probability distributions.

### Probability Distributions

If you plotted charts for 1,000 or 10,000 coin tosses similar to the above, you would generate similarly shaped distributions. This classic distribution which only allows for two outcomes is known as the **Binomial distribution** and is regularly used in probabilistic analysis.

The 100 coin toss chart shows that the average (or '**expected**' or '**mean**') number of heads here is 50. This can be calculated using a weighted

average in the usual way. The 'spread' of heads is clearly quite narrow (tapering off very sharply at less than 40 heads or greater than 60). This spread is measured by statisticians using a measure called **standard deviation** which is defined as the square root of the average value of the square of the difference between each possible outcome and the mean, *i.e.*

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

where:  $\sigma$  = standard deviation

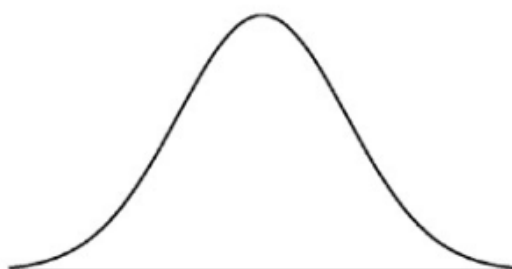
$N$  = total number of possible outcomes

$\Sigma$  = summation

$x_i$  = each outcome event (from first  $x_1$  to last  $x_n$ )

$\mu$  = mean or average.

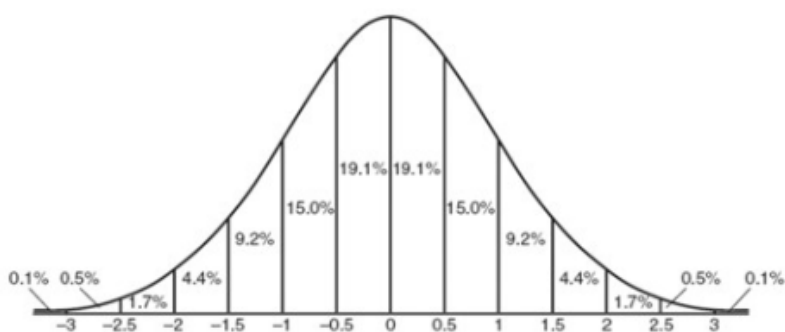
The Binomial distribution is not the most common distribution used in probability analysis: that honour belongs to the **Gaussian** or **Normal distribution**:



Generated by a complex mathematical formula, this distribution is defined by specifying the **mean** and **standard deviation** (see above). The reason it is so common is that in probability theory, the Central Limit Theorem (CLT) states that, given certain conditions, the mean of a sufficiently large number of independent random variables, each

with finite mean and standard deviation, will approximate to a Normal (Gaussian) distribution.

The Normal distribution's population is spread as follows:



*i.e.* 68% of the population is within one standard deviation of the mean, 95% within two standard deviations and 99.7% within three standard deviations.

The formula for the Normal distribution is given by

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

The **GAUSS** function calculates the probability that a member of a standard normal population will fall between the mean and z standard deviations from the mean. It has the following syntax:

**GAUSS(z)**

The **GAUSS** function has the following argument:

- **z**: this is required and this represents a number.

It should be noted that:

- if **z** is not a valid number, **GAUSS** returns the **#NUM!** error value
- if **z** is not a valid data type (*i.e.* it is not a number), **GAUSS** returns the **#VALUE!** error value
- because **NORM.S.DIST(0,TRUE)** always returns 0.5, **GAUSS(z)** will always be 0.5 less than **NORM.S.DIST(z,TRUE)**.

Please see our example below:

	A	B	C
1	Formula	Description	Result
2	<b>=GAUSS(1)</b>	Probability that a member of a standard normal population will fall between the mean and 1 standard deviation from the mean	<b>0.341</b>
3	<b>=GAUSS(2)</b>	Probability that a member of a standard normal population will fall between the mean and 2 standard deviations from the mean	<b>0.477</b>
4	<b>=GAUSS(3)</b>	Probability that a member of a standard normal population will fall between the mean and 3 standard deviations from the mean	<b>0.499</b>
5			

## The A to Z of Excel Functions: GCD

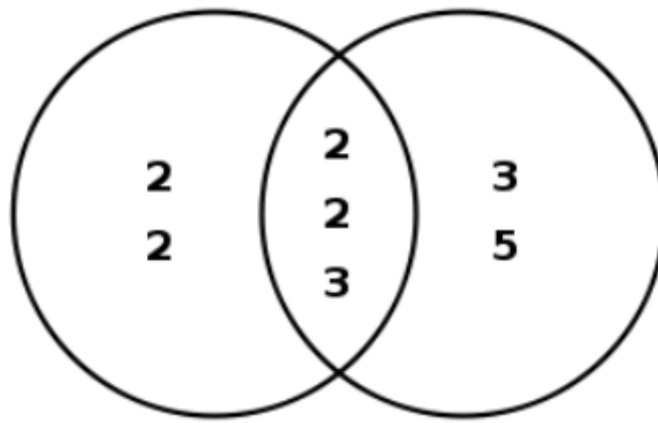
In mathematics, the greatest common divisor (GCD), also known as the greatest common denominator or the highest common factor, of two or more non-zero integers, is the largest positive integer that divides the numbers without a remainder. For example, the GCD of 180 and 48 is 12.

To show this, if we factorise the numbers down to primes:

$$180 = 2 \times 2 \times 3 \times 3 \times 5$$

$$48 = 2 \times 2 \times 2 \times 2 \times 3$$

Therefore, the prime numbers shared are:



$2 \times 2 \times 3 = 12$ .

A much more efficient method is the Euclidean algorithm, which uses the division algorithm in combination with the observation that the GCD of two numbers also divides their difference:

- Divide 180 by 48 is 3 remainder 36, so
- Divide 48 by 36 is 1 remainder 12, so
- Divide 36 by 12 is 3 with no remainder.

Therefore, 12 is the GCD of 180 and 48.

The Excel function **GCD** returns the greatest common divisor of two or more integers. The greatest common divisor is the largest integer that divides both **number1** and **number2** without a remainder. It has the following syntax:

**GCD(number1, [number2], ...)**

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

- **number1, number2, ...** where **number1** is required, and subsequent numbers are optional. There can be between one [1] and 255 numbers. If any value is not an integer, it is truncated.

It should be further noted that:

- if any argument is nonnumeric, **GCD** returns the **#VALUE!** error value
- if any argument is less than zero, **GCD** returns the **#NUM!** error value
- one divides any value evenly
- a prime number has only itself and one as divisors
- if a parameter to **GCD** is  $\geq 2^{53}$ , **GCD** returns the **#NUM!** error value.

Please see our example below:

	A	B	C
1	Formula	Description	Result
2	<b>=GCD(180,48)</b>	Greatest common divisor of 180 and 48	<b>12</b>
3	<b>=GCD(343,217,49)</b>	Greatest common divisor of 343, 217 and 49	<b>7</b>
4	<b>=GCD(7,1)</b>	Greatest common divisor of 7 and 1	<b>1</b>
5	<b>=GCD(7,0)</b>	Greatest common divisor of 7 and 0	<b>7</b>
6	<b>=GCD(7)</b>	Greatest common divisor of 7	<b>7</b>
7			

## The A to Z of Excel Functions: GEOMEAN

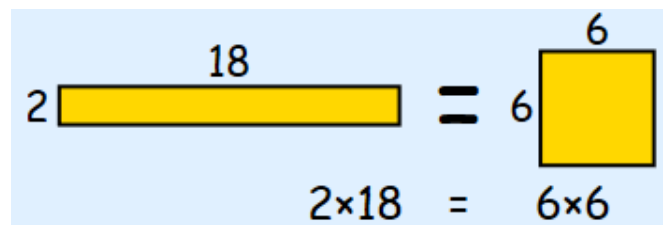
In mathematics, the geometric mean is a mean or average, which indicates the central tendency or typical value of a set of numbers by using the product of their values (as opposed to the arithmetic mean

which uses their sum). The geometric mean is defined as the  $n$ th root of the product of  $n$  numbers, *i.e.* for a set of numbers  $x_1, x_2, \dots, x_n$ , the geometric mean is defined as

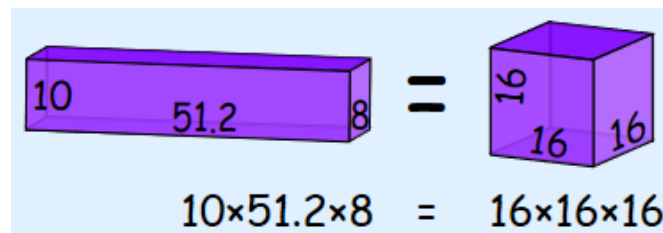
$$\left( \prod_{i=1}^n x_i \right)^{\frac{1}{n}} = \sqrt[n]{x_1 x_2 \cdots x_n}$$



In two dimensions, it is the equivalent of finding the equivalent square with the same area as the rectangle given by the two dimensions cited:



In three dimensions, it is the equivalent of finding the equivalent cube with the same volume as the given hexahedron with the three dimensions cited:



The idea continues in  $n$  dimensions.

The Excel function **GEOMEAN** returns the geometric mean of an array or range of positive data. For example, you can use **GEOMEAN** to calculate average growth rate given compound interest with variable rates. It has the following syntax:

**GEOMEAN(number1, [number2], ...)**

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

- number1, number2, ...** where **number1** is required, and subsequent numbers are optional. There can be between one (1) and 255 numbers. You can also use a single array or a reference to an array instead of arguments separated by commas.

It should be further noted that:

- arguments can either 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 array or reference argument contains text, logical values or empty cells, those values are ignored; however, cells with the value zero are included
- arguments that are error values or text that cannot be translated into numbers cause errors
- if any data point  $\leq 0$ , **GEOMEAN** returns the #NUM! error value
- the equation for the geometric mean is:

$$GM_{\bar{y}} = \sqrt[n]{y_1 y_2 y_3 \cdots y_n}$$

Please see our final example for this month below:

	A	B	C
1	Data		
2	6		
3	7		
4	9		
5	15		
6	21		
7			
8			
9	Formula	Description	Result
10	=GEOMEAN(A2:A6)	Geometric mean of the data set contained in A2:A6	10.3552

More Excel Functions next month.

## Beat the Boredom Suggested Solution

Earlier in this newsletter I discussed factorials denoted by  $n!$  where

$$n! = n(n-1)!$$

$$\text{i.e. } n! = n(n-1)(n-2) \dots 3 \times 2 \times 1.$$

For example,  $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$ . I highlighted the Excel function, **FACT**, which calculates factorials:

Number	Factorial	Length
1	1	1
2	2	1
3	6	1
4	24	2
5	120	3
6	720	3
7	5040	4
8	40320	5
9	362880	6
10	3628800	7
161	7.5907E+286	20
162	1.2297E+289	20
163	2.0044E+291	20
164	3.2872E+293	20
165	5.4239E+295	20
166	9.0037E+297	20
167	1.5036E+300	19
168	2.5261E+302	20
169	4.2691E+304	20
170	7.2574E+306	19

Do you see what happens from **FACT(15)** onwards? These numbers are expressed – and more importantly, stored – in Scientific (Exponential) notation.  $15!$  is actually 1,307,674,368,000, although it is displayed as 1.30767E+12 and is stored as 1307674368000 (deliberately no commas), which is why its length is 13 (there are 13 digits – more precisely, characters – in the text string).

Similarly:

- $22!$  is actually 1,124,000,727,777,607,680,000, which is displayed by default as 1.124E+21 and is stored as 1.12400072777761E+21. This stored string has 20 characters: 15 digits (the maximum significant figures Excel can display), a decimal point and four characters for “E+21”. This means that the maximum length for a number smaller than  $10^{100}$  is 20 characters

- $23!$  is actually 25,852,016,738,884,976,640,000, which is displayed by default as 2.5852E+22 and is stored as 2.5852016738885E+22. This stored string has 19 characters. This time only 14 digits are included: this is because the 15<sup>th</sup> number would be “0” and hence is suppressed. This is why  $23!$  has a length of 19 rather than  $22!$  which has a length of 20.  $23!$  is definitely larger than  $22!$

Therefore, precision is of concern with larger numbers in Excel as they will only display to a maximum of 15 significant figures.

There are two more issues though.

Firstly, there’s a reason the challenge was to calculate  $171!$

Number	Factorial	Length
1	1	1
2	2	1
3	6	1
4	24	2
5	120	3
6	720	3
7	5040	4
8	40320	5
9	362880	6
10	3628800	7
161	7.5907E+286	20
162	1.2297E+289	20
163	2.0044E+291	20
164	3.2872E+293	20
165	5.4239E+295	20
166	9.0037E+297	20
167	1.5036E+300	19
168	2.5261E+302	20
169	4.2691E+304	20
170	7.2574E+306	19
171	#NUM!	#NUM!

Secondly, I note using numbers only will not retain precision. I need

I need to get creative.

171! is (big drum roll please)...

1241018070217667823424840524103103992616605577501693  
1853889518036119960752216917529927519781204875855764  
6495950167038705280988985869071076733124203221848436  
4310473577889968548278290754541561964852153468318044  
2932395981736968996572359039476161522785581800611763  
65108428800

How was this achieved? Without any VBA, I created a table in Excel to calculate the following:

1. I need to identify the largest factorial in my table (171) and multiply this by 9.999999999999999... (rounded up to 10) to work out the number of characters I need to keep in reserve, which is  $171 \times 10$  which is 1710 (without commas). This is four characters. Given one character is already used, I need to keep three characters in reserve (this is used for the next step)
2. Since the maximum number of significant figures is 15, I need to separate my number into blocks (what I called “batches” in the model) into text strings of no more than  $15 - 3 = 12$  characters. This was defined as the **Batch\_Length**
3. Define 1! as 1
4. Convert 1! (1) to a text string (“1”)
5. Now we need to calculate the next factorial
6. Calculate the character length of this string
7. Divide this length by **Batch\_Length** (i.e. 12) rounded up to the nearest whole number. This is how many batches are needed
8. Split the text factorial into batches: Batch 1 is the final 12 characters, Batch 2 is the 24<sup>th</sup> from last to 13<sup>th</sup> from last digit, Batch 3 is the 36<sup>th</sup> from last to the 25<sup>th</sup> from last digit, and so on
9. For each batch, convert the text string back to a number simply by multiplying by the factorial you are trying to calculate (e.g. if you have just calculated 1! then multiply by two, if it is 17! then multiply by 18)
10. If the resulting number contains more than 12 characters, split the number in two: the “edited” number is the last 12 characters and the “residual” is any digits at the beginning not included
11. Add the residual from Batch **n-1** to the Edited number in Batch **n**
12. Convert all of the numbers back to text strings
13. Concatenate the batches using the ampersand (&) operator (note this cannot be done with a text function because only 255 characters are allowed and only 15 significant figures would be displayed)
14. This is the next factorial in the series
15. To calculate the next factorial in the sequence, return to Step 5 (above).

### ***Why is this Important?***

It's not that we are trying to create hideous challenges. It's just that this example – and its solution highlight some key points for Excel users:

- Until next time.

## Upcoming SumProduct Training Courses - COVID-19 update

Due to the COVID-19 pandemic that is currently spreading around the globe, we are suspending our in-person courses until further notice. However, to accommodate the new working-from-home dynamic, we are switching our public and in-house courses to an online delivery stream, presented via Microsoft Teams, with a live presenter running through the same course material, downloadable workbooks to complete the hands-on exercises during the training session, and a recording of the sessions for

your use within 1 month for you to refer back to in the event of technical difficulties. To assist with the pacing and flow of the course, we will also have a moderator who will help answer questions during the course.

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

Location	Course	Date	Date	Duration	Duration
Online (Australia)	Power Pivot, Power Query and Power BI	29 Sep - 1 Oct 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	6 Oct 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	7 - 8 Oct 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	3 - 5 Nov 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	10 Nov 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	11 - 12 Nov 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	8 - 10 Dec 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	15 Dec 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	16 - 17 Dec 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	2 Days

## Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This year, we thought we'd revisit each function key in depth (there are 12 – one for each month of the year!). Given it's now September, let's look at the **F9** tips.

Keystroke	What it does
F9	Calculate now
CTRL + F9	Minimise window
CTRL + ALT + F9	Full recalculation

Keystroke	What it does
CTRL + ALT + SHIFT + F9	Forced full recalculation (rebuild)
SHIFT + F9	Calculate active sheet(s)

Hey, it's not our fault so many do the same thing this month!

There are c.550 keyboard shortcuts in Excel. For a comprehensive list, please download our Excel file at [www.sumproduct.com/thought/keyboard-shortcuts](http://www.sumproduct.com/thought/keyboard-shortcuts). Also, check out our new daily **Excel Tip of the Day** feature on the [www.sumproduct.com](http://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](mailto: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](http://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](mailto: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](mailto:training@sumproduct.com) for a copy of the brochure or download it directly from [www.sumproduct.com/training](http://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**