Much of the Northern hemisphere might be on holiday in August

but we think it's **DEC** with our A to Z of Excel Functions. Certainly, it's still all systems go at SumProduct HQ – evidenced by yet another bulked-up newsletter. If this continues, we'll have to call in Weight Watchers for this monthly circular!

The Excel Summit South is now underway: by the time you read this, we'll have been and gone in Brisbane, but there's still time to catch us and our fellow MVPs in other Australian capital cities. Talking of Most Valuable Professionals (MVPs), we also have some good news for our team as both of our Excel MVPs were renewed recently – you can find out more below.

This month we also take a look at how to lay out a financial model and continue all the usual stuff with Keyboard Shortcuts, Visual Basics, Power Pivot Principles, Power Query Pointers and the usual 20 trillion updates in Power BI.

Until next month.

Liam Bastick, Managing Director, SumProduct

MVP Renewal

Liam Bastick

Tim Heng

SumProduct is pleased to announce that two of our Directors, **Liam Bastick** and **Tim Heng**, have been re-awarded Microsoft's Most Valuable Professional (MVP) award for Excel for 2019-20. This award recognises exceptional technical community leaders from around the world who voluntarily share their high quality, real world expertise with others. Microsoft MVPs are a highly select group of experts representing technology's best and brightest who share a deep commitment to community and a willingness to help others.

Worldwide, there are over 100 million participants in technical communities; of these participants, there are c.4,000 active Microsoft MVPs. In Excel, we believe there are approximately 66 (yes, we know that's a specific number!) that have received this award.

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

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

Excel Summit South: It's Underway!

We're out and about at the moment, with the Excel Summit South well and truly underway! By the time you read this, Brisbane has been and gone – hopefully, with some satisfied attendees who have learned some productivity tips for back in the workplace.

There's still time to join though – although Sydney might be a little tight! The remaining itinerary is as follows:

- **Sydney:** Thursday 1 and Friday 2 August
- **Melbourne:** Monday 5 and Tuesday 6 August
- **Perth:** Thursday 8 and Friday 9 August.

Check out the website at excelsummitsouth.com for all the details, prices and registration. Early Bird pricing is nearly over. Don't miss out – and remember, the additional 15% off using the code **NEWSLETTER15**. Hopefully, we will see you there!
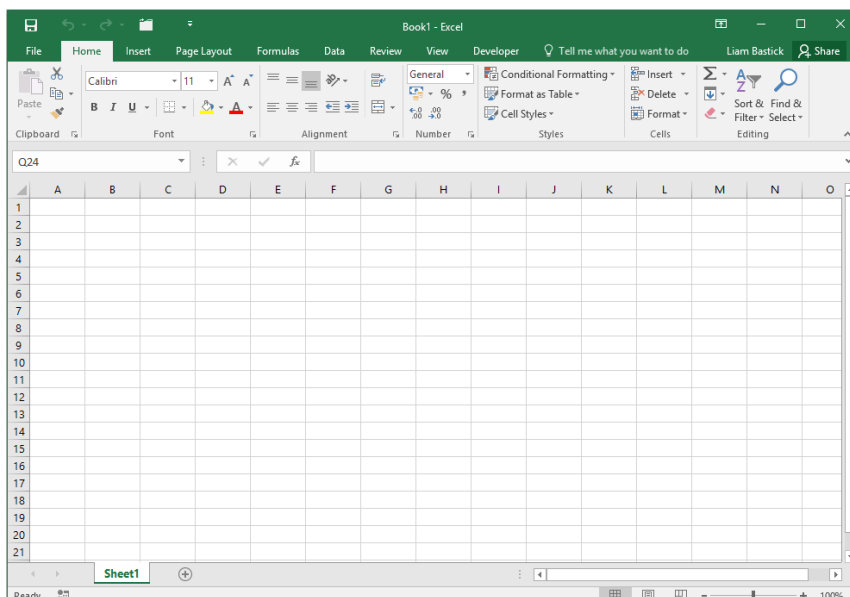
Laying Out a Financial Model

This is personal, but hopefully, you will agree with the underlying concepts!

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

Therefore, you should put some thought into designing your spreadsheets and not just the formulae. "Best Practice" should be considered a proper noun to reflect the idea that a good model has four key attributes I call **CRaFT**:

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

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

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

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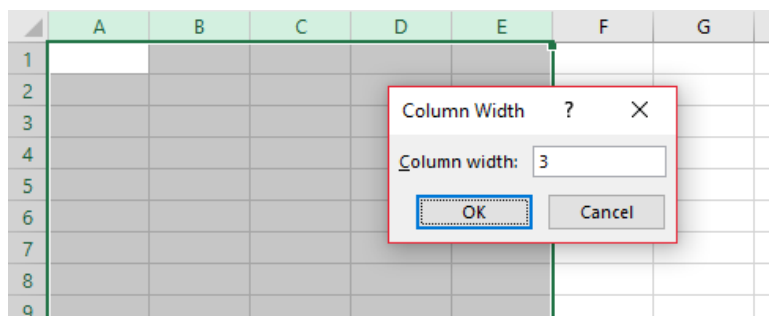

It may not be clear why I choose to do this, but read on (hey, I need to keep you on the edge of your seat!). I choose a width of 3 as this effectively makes the cells in these columns square.
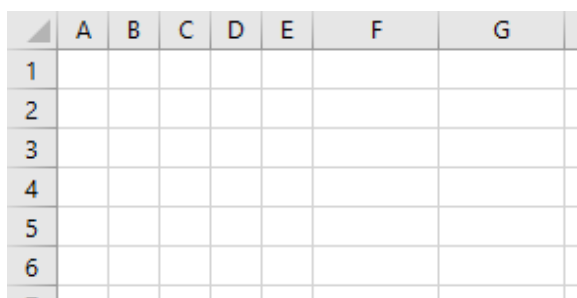
You can elect to highlight a different number of columns and you can modify the width too. There are two key points to this:

1. Keep column **A** blank other than for the sheet headings (I will explain later)

2. Be consistent, both with the widths of the columns narrowed here and with other worksheets within the same workbook (again, I will explain soon).

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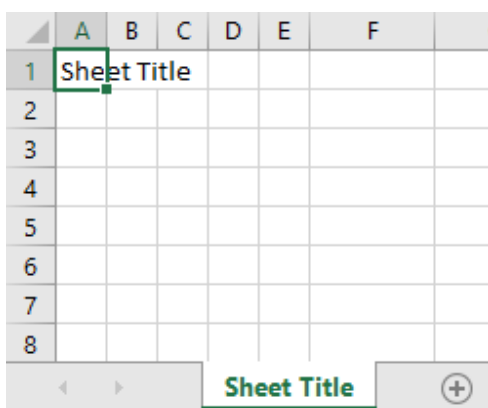
There are three reasons for this:

1. Given that sheet tab names cannot be infinitely long, sheet title is more succinct and easier for the end user to understand

2. Given that the sheet title appears on the worksheet, the name has to be written formally and cannot be an incomprehensible abbreviation, similar to many sheet tab names out there

3. This approach promotes consistency, one of the four key concepts of Best Practice modelling.

In cell **A2**, I will put the model name. This is important as often we only show extracts of a workbook in an appendix to a report or a PowerPoint presentation. Therefore, I place it here rather than in the header or footer of each worksheet instead (but you may do both):
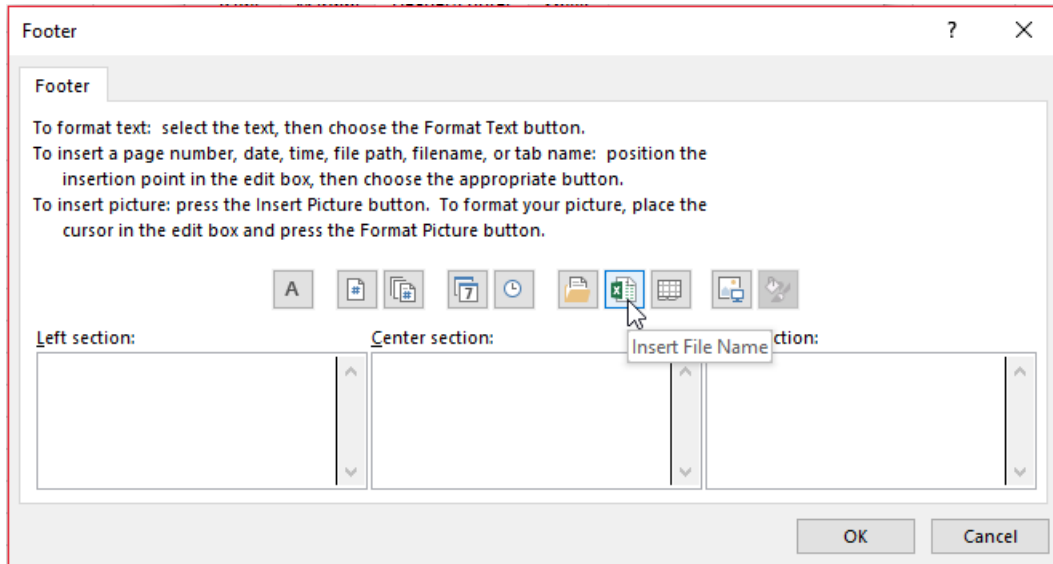
If you intend to add this to the header or footer, these attributes may be accessed by the keyboard shortcut **ALT + P + SP -> 'Header / Footer' tab -> 'Custom Footer...' button**. I acknowledge it's simpler than my alternative, but this filename will only display when the worksheet is printed. What if it is an image on a PowerPoint slide or, say, as Appendix 4 in a Word document? This is why I keep the model name front and centre on my worksheets.

There's a formula too:

=IFERROR(MID(CELL("filename",A1),FIND("[",CELL("filename",A1))+1,FIND("]",CELL("filename",A1))-FIND("[",CELL("filename",A1))-1),"")
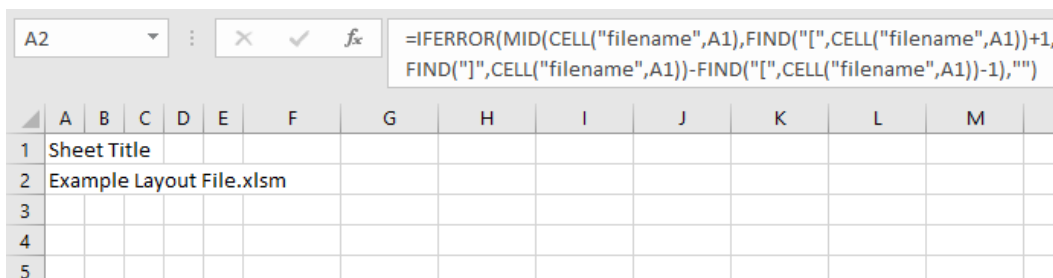
The next key item to position at the top of the sheet is a navigation aid. Today's workbooks can be quite complex with many worksheets. I recommend incorporating a central navigation page – a "Table of Contents" – which allows the end user to traverse the workbook quickly and easily.

It looks like I have added a hyperlink in cell **A3**, right? Not quite. I am a little craftier than that. Actually, I have highlighted cells **A3:F3** and then merged the cells using Excel's **Merge Across** functionality (**ALT + H + M + A**):
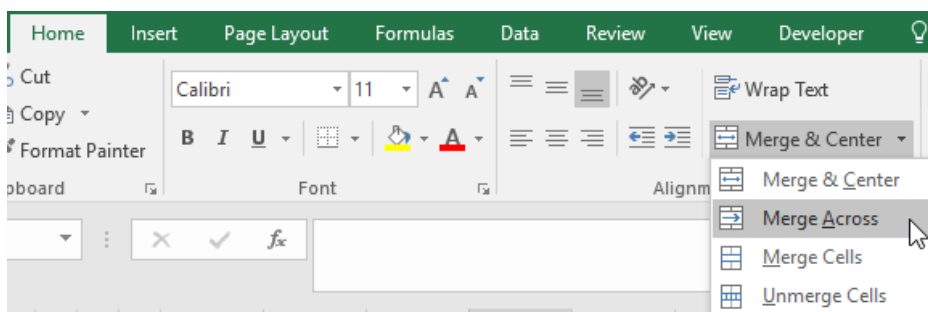
Hyperlinks may be created easily using the keyboard shortcut **CTRL + K** (or click the 'Link' button on the 'Insert' tab) to link to a place in your document. The intention is to set up a central Table of Contents worksheet where all of the hyperlinks to the other worksheets reside:

1. Table of Contents

Cover
Style Guide
Model Parameters
Timing
Error Checks
Change Log

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

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

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

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

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

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

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

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

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

This is not a drawn line. This is called a frozen pane. Frozen panes break up the worksheet in to as many four pieces and allow parts of the worksheet to remain on view ("be frozen") whilst the reader scrolls down or across the worksheet. Located in the 'Window' grouping of the 'View' tab of the Ribbon, there are three ways to create a frozen pane:
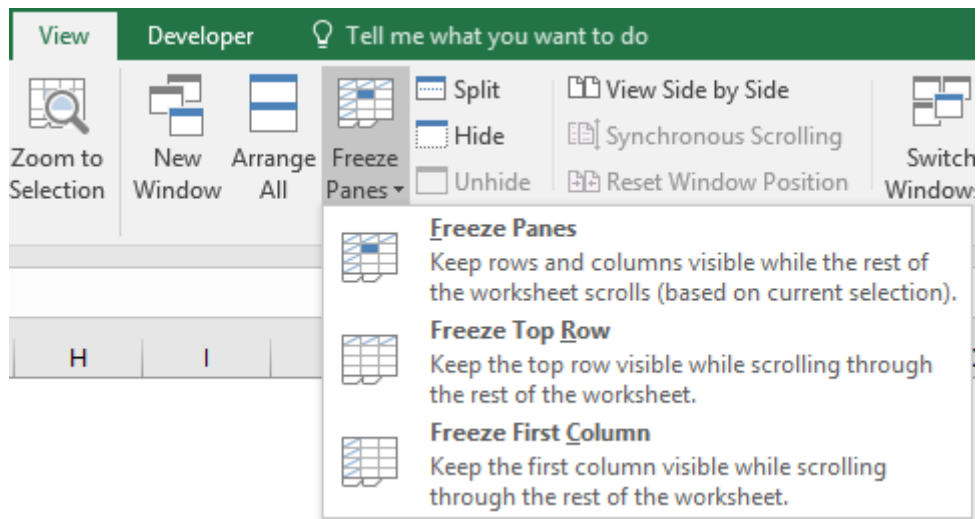
- **Freeze top row:** Keeps the top row visible no matter how far down the spreadsheet you scroll
- **Freeze first column:** Keeps the first column visible no matter how far to the right you scroll the spreadsheet
- **Custom (Freeze Panes):** Creates a frozen locus at the intersection of the top row and the first column of the cell(s) selected.

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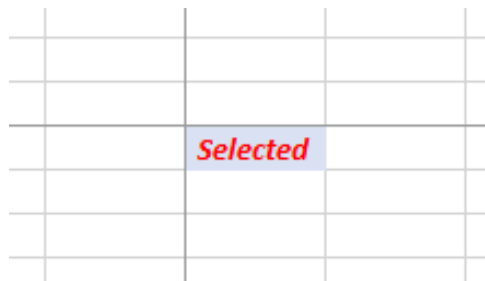
Frozen panes are created for the region the selection is in, the region directly above, the region to the immediate left and diagonally opposite the top left-hand corner of the selection. If the selection were in column **A**, there would only be two frozen panes: the rows immediately above and the remainder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

In our example, cell **A5** has been made the basis of the frozen pane, so that rows 1 to 4 will always be visible. This cell should be given a range name, *e.g.* **HL_Home**, as this is the cell hyperlinks to this sheet should link.

This cell 'resets' the sheet when a frozen pane has been added (not cell **A1**) and makes the model easier to navigate consequently. This cell can always be identified by employing the keyboard shortcut **CTRL + HOME**.

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

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

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

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

Take special note of the spacing: one blank row between headings; two lines between sections. That's my preference. You choose your own if you would prefer – just be consistent. It makes it very simple to copy sections and keep referencing if spacing is deliberate.
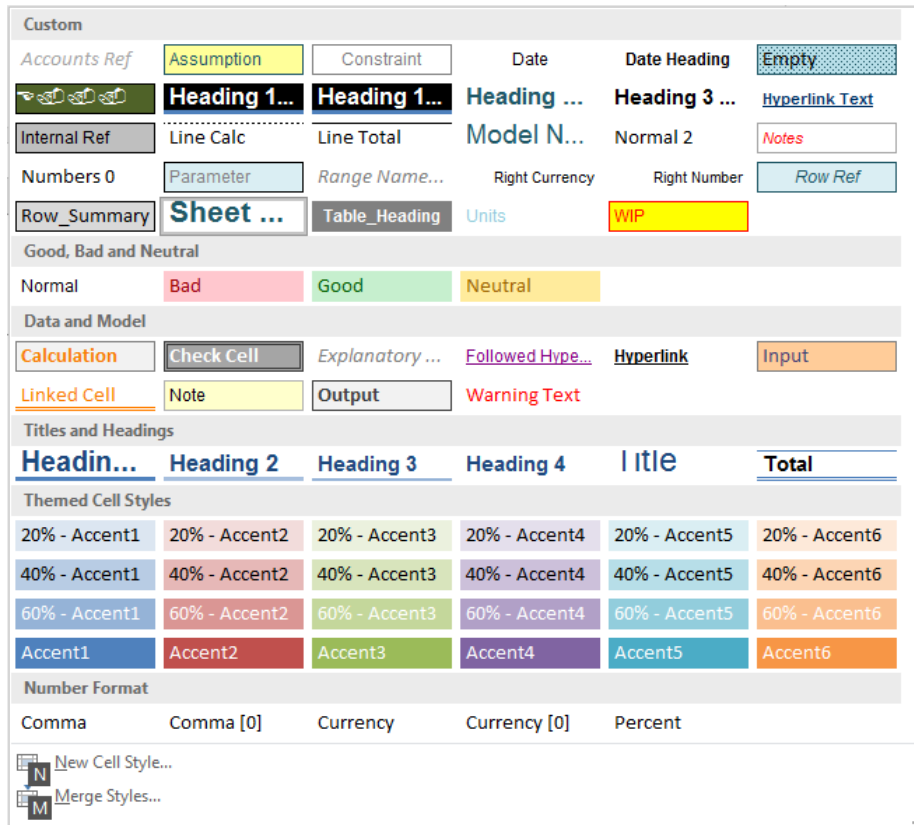
Blank columns **H** and **I** are in existence in case we have any calculations, inputs or referred values that do not refer to a particular time period. If they are not required, I tend to narrow the columns to a width of 1 (say), so that they are still there in case they are needed later.

Adding labels, data and formulae:

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

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

Very quickly, our spreadsheet is taking shape:

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

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

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

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

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

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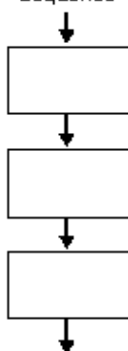
Visual Basics

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 control structures, and sequential control structures in particular.

In a programming, a control structure determines the order in which statements are executed. Control structures can be grouped into three main categories:
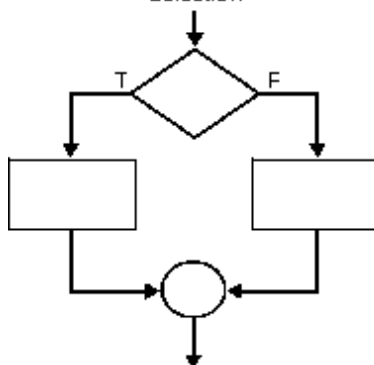
1. **Sequential:** Sequential execution is where each statement in the source code will be executed one by one in a sequential order. This is the default mode of execution

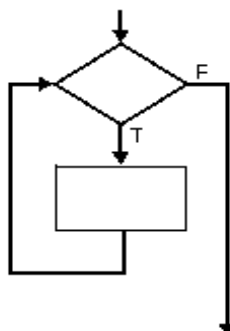2. **Selection:** The selection control structure is used for making decisions and branching statements

3. **Iteration:** The iterative control structures are used for repetitively executing a block of code multiple times

We'll consider them all, but for this month, let's take a look at the first one. **Sequential execution** is where each statement in the source code will be executed one by one in a sequential order. This is the default mode of execution.
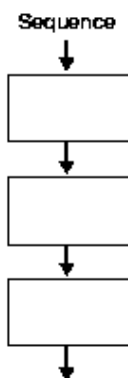
Sequence

Here are a couple of prime examples.

IF…THEN…ELSE

This is the most basic sequential control structure. Identical to the **IF** function in Excel, this calls on the most basic decision path – execute steps if a condition is met, otherwise execute other steps.

```
If expression Then
 <statements>
Else
 <statements>
End If
```

Here's an example:

```
Option Explicit
Sub IFStatement()
    Dim myNumber As Integer
    myNumber = 3
    If myNumber > 0 Then
        Debug.Print "Number is Positive!"
    Else
        Debug.Print "Number is Negative!"
    End If
End Sub
```

Note however that the **ELSE** statement is not mandatory, which is the same as the **IF** function in Excel. If no statements need to be executed should the condition prove false, then the **ELSE** section can be omitted entirely.

ELSE IF

Sometimes users are required to test for more than one condition. Quite often in Excel, one comes across multiple **IF** statements chained together in one formula.

```
IF( condition1, value_if_true1, IF( condition2, value_if_true2, value_if_false2 ))
```

Often this is referred to as "nested ts". The idea of nesting comes from embedding or "nesting" one **IF** function inside another. In VBA, this can be similarly achieved by using the **ELSEIF** statement.

The above example would be written in VBA as:

```
IF condition1 THEN
    value_if_true1
ELSEIF condition2 THEN
    value_if_true2
ELSE
    value_if_false2
END IF
```

Here's a working VBA code sample:

```vba
Option Explicit

Sub IFStatement()

    Dim myNumber As Integer

    myNumber = 3

    If myNumber < 10 Then

        Debug.Print "Number is single digit!"

    Else If myNumber < 100

        Debug.Print "Number is double digit!"

    Else

        Debug.Print "Number is very large!"

    End If

End Sub
```

More next month.

Power Pivot Principles

We continue our series on the Excel COM add-in, Power Pivot. This month, we consider some of the various menu options in the Power Pivot window: now, main course or entrée…?

From previous articles, now that we have a good understanding on what Power Pivot is and how to import data, let's run through the different menu options available in Power Pivot. Understanding the menu options will provide a basis of how to get the best out of Power Pivot.

Below is a screenshot of the Power Pivot tab option in Excel:
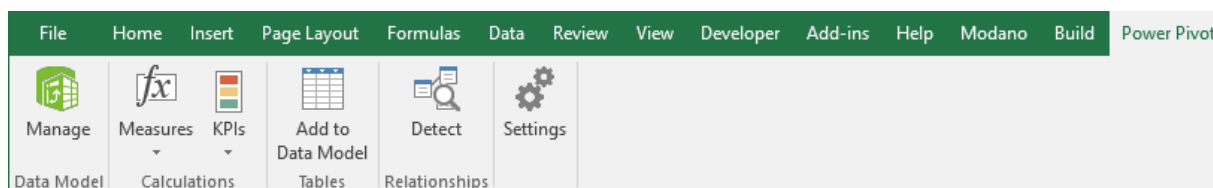
Data Model

This option allows us to access our Power Pivot data model (*i.e.* where the data is stored "behind the scenes"). We can load and prepare data or continue working on data already added to this workbook.

Calculations

Like we can do in the Power Pivot data model, the 'Measures' and the 'KPIs' options allow you to add or manage 'Measures' and 'KPIs'.

Tables

'Add to data model' will allow us to create a linked table by adding the Excel table to the data model. Linked tables are a live link between the table in Excel and the table in the data model, so updates to the table in Excel automatically update the data in the model. If this table is already in the data model then this action adds a copy to the model.

Relationships

This allows you to define settings for your Power Pivot environment and specify language options.
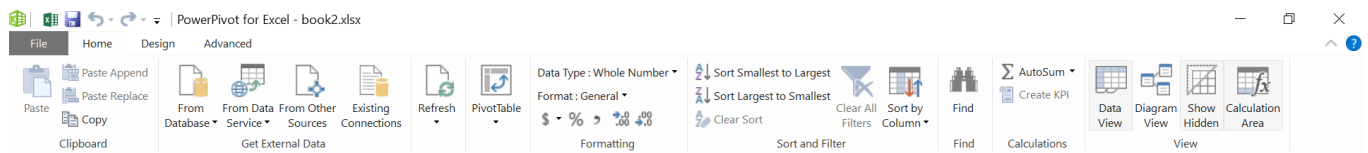
Settings

This allows you to define settings for your Power Pivot environment and specify language options.

Once you get into Power Pivot, there are three menu tabs on the Power Pivot Ribbon: 'Home', 'Design' and 'Advanced'. We'll take a look at these as well.

Let's start with the 'Home' tab:

Power Pivot Menu Options: Home

There are multiple groups that make up the 'Home' tab:

Clipboard

This section allows data to be copied from our Power Pivot data model and pasted outside of the data model. Data can also be copied from Excel and pasted as a table in the Power Pivot data model.

The 'Paste Append' option allows us to paste new data at the foot of existing data and 'Paste Replace' allows us to replace all existing data.

Please note that when using these options, the data model which we are pasting the data from needs to be in the same structure as the Power Pivot data model for it to copy over correctly.

Get External Data

This is how data is imported from external sources into Power Pivot. There are many different options to import data. Existing connections can also be accessed from this section.

Refresh

If our Power Pivot model is connected to data from another source and that source data is updated, this is where the Power Pivot data will be refreshed from. We can either refresh a specific table or the entire data set.

PivotTable

This option gives us the ability to analyse data in a PivotTable or chart. There are different options we can select – either a PivotTable or chart on their own or together or even multiple tables together.

Formatting

The formatting section allows us to change the data type: decimal number, text, date, currency *etc*. We can also decrease / increase the number of decimal places on a number. How we format the data in the Power Pivot window is how the data will be displayed in the PivotTable or chart.

Sort and Filter

The sort and filter options allow us to sort our data. Depending on what type of data format the column is showing will determine how we can sort the data.

- If the data is in date format, then the option would be to sort our data from 'oldest to newest' or 'newest to oldest'
- If the format is numeric, the sort option will show 'smallest to largest' or 'largest to smallest'
- The 'Clear All Filters' option allows us to clear any filters that are currently being used to view the data
- The 'Sort by Column' option allows us to sort a column based on another column. For example, to sort the month name column in the correct order we could sort it by the month number.

Find

The 'Find' option allows us to find metadata like fields or field names. If we need to search for specific data in our data model, we would use the filter drop-down arrows on the columns.

AutoSum

The 'AutoSum' feature allows us to create calculated fields based on the sum of the selected column and display it below the column. There's other ways to do this though – but more on that in a later article…

Create KPI

Allows us to convert calculated fields into a Key Performance Indicator (KPI).

View

The 'View' section allows us to view our data in different ways.

- The 'Data View' shows all of the columns of data
- The 'Diagram View' shows the list of our tables and shows the relationships between the tables
- The 'Show Hidden' option allows us to show objects in the model that have been hidden from the client tools
- The 'Calculation Area' is used for creating, editing and managing calculated fields and KPIs within the model.

Feel free to play around with these options in Power Pivot, to get a better feel of the software suite.

Let's now turn our attention to the 'Design' tab.
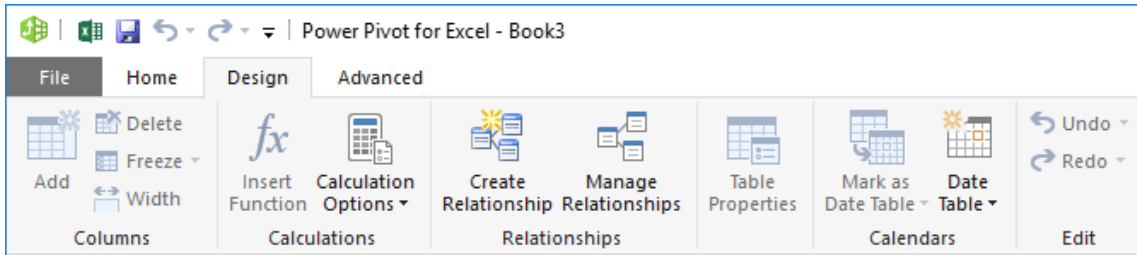
Power Pivot Menu Options: Home

There are multiple sections made up of the 'Design' tab as follows:

Columns

The 'Add' option allows you to add in a new column and then type in a formula to create a calculated column (we will discuss these formulae – known as **DAX** formulae – and calculated columns later). The 'Delete' option allows you to delete rows.

The 'Freeze' option allows us to freeze the selected columns in place, while scrolling through the table. Do note that when a column is frozen,

it will be moved to the very left of the table and will remain there even after unfrozen.

The 'Width' option allows us to adjust the width of the selected columns.

Calculations

The 'Insert Function' option allows you to insert a new calculated column with a formula. The list of functions available are sorted by category.

Calculation Options

'Auto Calculation Mode' should be set to ensure that the formulae (measures) and calculated columns are calculated automatically. If we have a large data model with many computations, we can change this section to be 'Manual Calculation Mode' so that the results will not calculate until we select 'Calculate Now'.

Relationships

'Create Relationships' allows us to create relationships between your tables of data. To create, edit or delete a relationship select 'Manage Relationships'.

Table Properties

'Table Properties' allows us to work with the data in our data model. We can change the column, row or table mappings. If we brought in columns of data in the data model that we don't need this would be one method to remove those columns.

Mark as Date Table

This option allows us to mark the selected table as a date table to enable dedicated date filtering in reports that you create in applications like Excel. This requires one of the fields to be a date field. These dates must cover all dates in the database, contain no duplicates or missing periods.

Edit

The 'Undo' and 'Redo' options allows us to undo and redo the last action performed in Power Pivot similar to the Excel functionality.

Next up is the 'Advanced' tab:
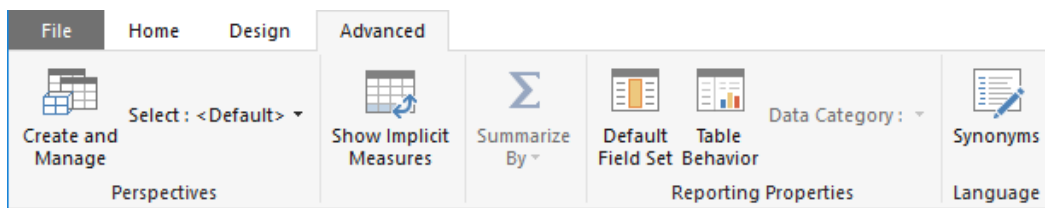
Power Pivot Menu Options: Advanced

The 'Advanced' menu was only made available in Power Pivot 2013 and later.

Perspectives

The 'Perspectives' area allows us to define views of the data. Perspectives are typically defined for a particular user group to it easier to navigate large data sets, *e.g.* we may want the finance team and the sales team to view different sections of the data set. We would use 'Perspectives' to set what each team can see (we will cover Perspectives in greater detail later).

The 'Select' option allows you to view the different perspectives that already exist.

Show Implicit Measures

The 'Show Implicit Measures' option is a toggle; switched on, it will show 'Implicit Measures'; deactivated, it will hide 'Implicit Measures'. Implicit measures are created when we drag fields into the 'Values' area in the 'Field List' in Excel (more on 'Implicit Measures' in future blogs).

Summarize By

This option allows us to change the method of calculation for a column to one of the fields below:
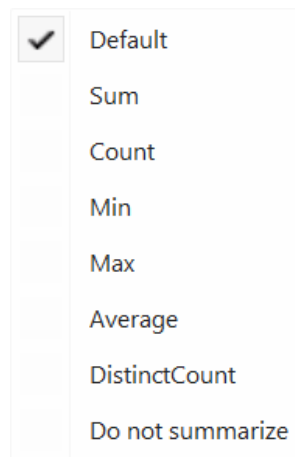
Reporting Properties

'Default Field Set' allows us to specify the column, measure and field ordering that define the default field set when visualising this table in certain client tools. 'Table Behaviour' allows us to change the default behaviour of different visualisation types and default grouping behaviour in client tools for a table.

'Data Category' allows us to categorise the fields within the selected table.

Language

'Synonyms' displays suggested alternatively names for the selected table's fields.

Stay tuned for our next post on Power Pivot. In the meantime, please remember we have training in Power Pivot which you can find out more about here. If you wish to catch up on past articles in the meantime, you can find all of our Past Power Pivot blogs here.

We'll continue next month. Stay tuned for our next post on Power Pivot. In the meantime, please remember we have training in Power Pivot which you can find out more about at www.sumproduct.com/courses/power-pivot-power-query-and-power-bi.

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 take a closer look at conditional columns.

Previously, we have used a conditional column to separate data based on the datatype. Conditional columns allow us to create some quite complex logical statements. The step created when we add a conditional column provides an opportunity to demonstrate how to tidy up the **M** code (the language that Power Query is written in) behind an individual step, in order to make it easier to read and simpler to understand.

Let's return to the data we used last month and look at some of the logic that we can choose when creating a conditional column. In the Query Editor, in the 'Add Column' tab, let's opt to add a 'Conditional Column':
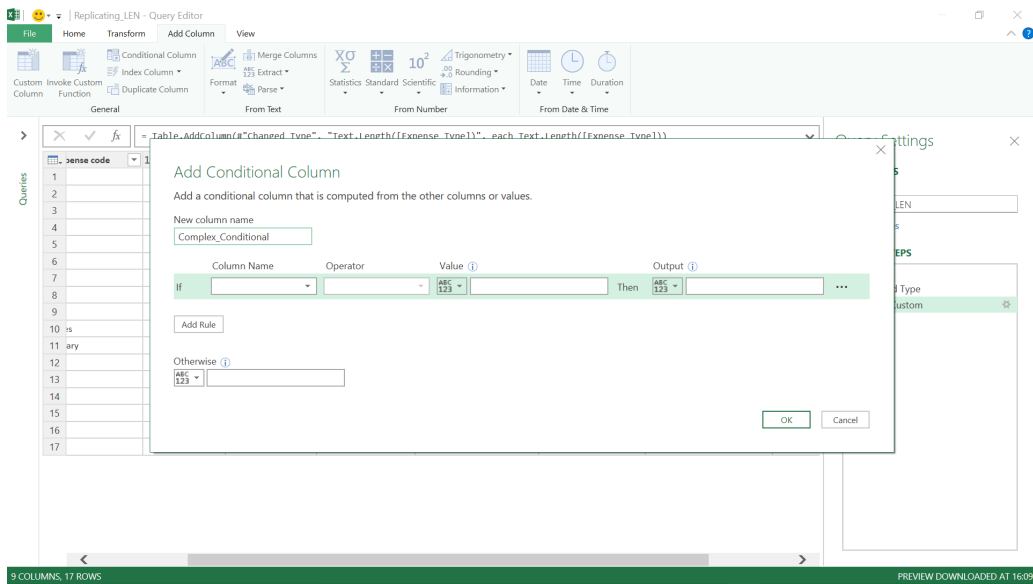
The first value we can enter is the 'New column name', which we'll decide to call **Complex_Conditional**. We're then given the opportunity to fill in the values in an 'If' statement.

We first need to specify the column name that our statement will be based on. We have a drop-down to help me so that we may pick any of the columns in the query.
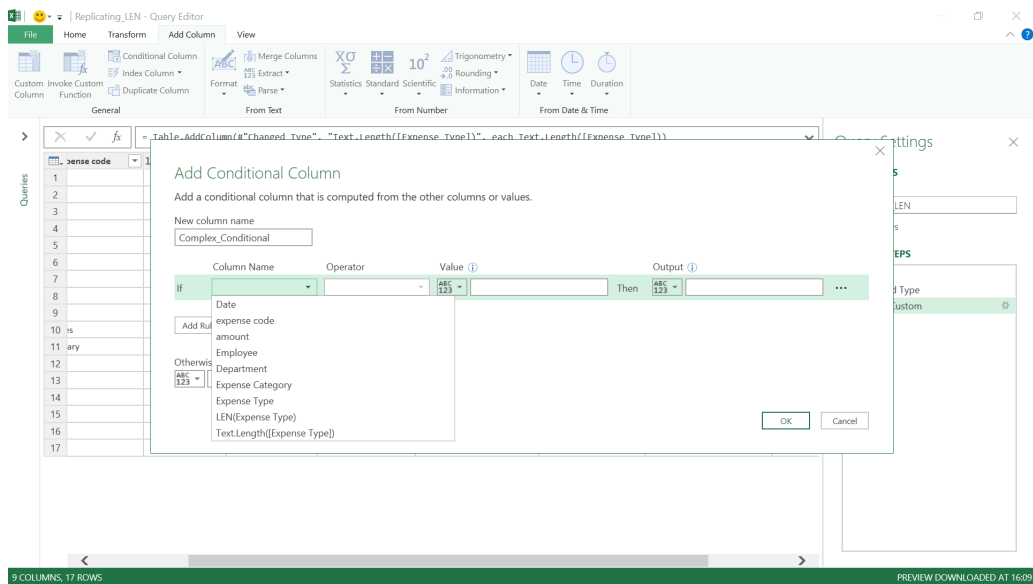

Let's begin with **Expense Category**. The next entry we're prompted for is the 'Operator' and again we have a dropdown to help us:
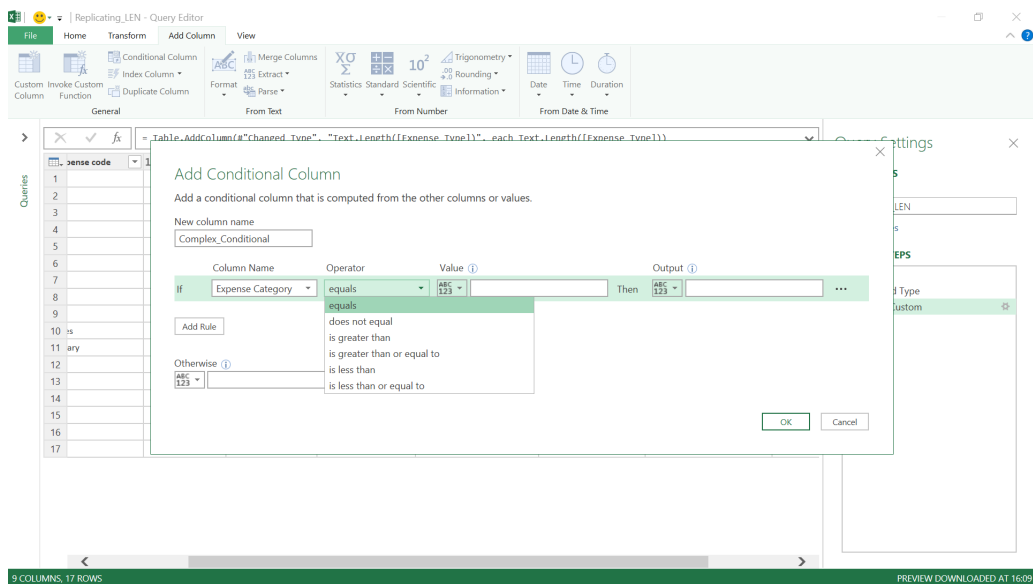
This dropdown is clever. It knows that **Expense Category** is datatype number, and so it has only given us options that apply to numbers. If we were to use **Expense Type** instead, we would be operating on a text field, so the options would be different, as shown below:
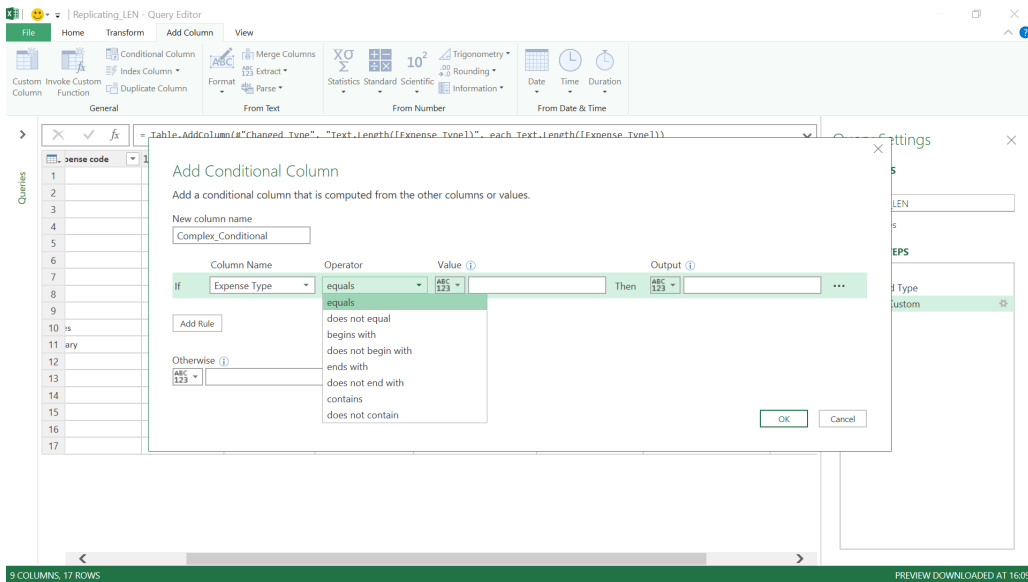
Going back to use **Expense Category**, we'll choose 'is greater than' and move to the next value which is 'Value' (the value which **Expense Category** needs to be greater than). There is a dropdown next to the data type icon:
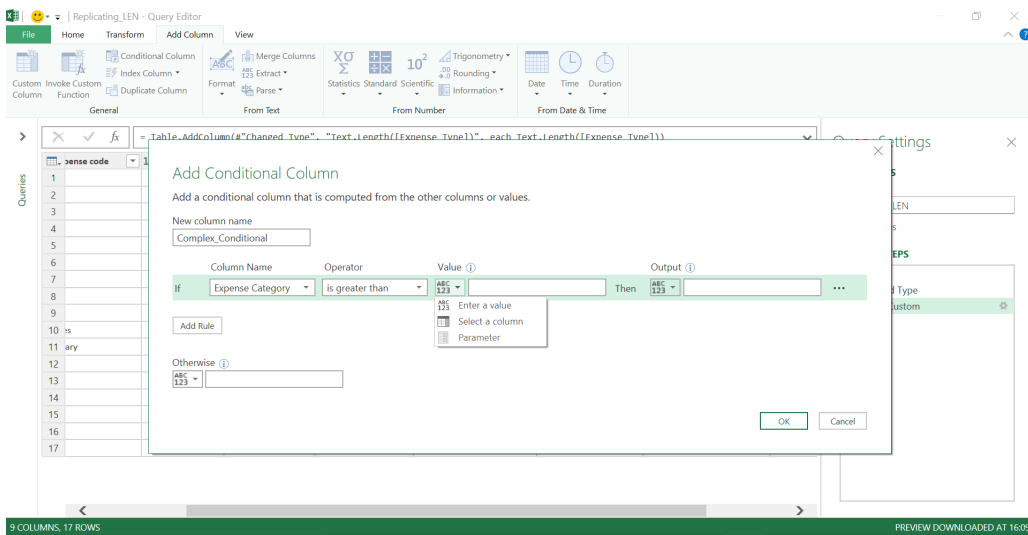
Therefore, we can either 'Enter a value', 'Select a column' or there is a third option 'Parameter'. Enter a value' is simple enough. We just pick a numerical value to compare the column with. If we choose to 'Select a column', a dropdown icon appears next to the value, but it is not quite as clever this time, as all the query columns appear, along with a little warning icon that tells us to 'Enter a number value' if we click on it. Just for fun, let's pick a column that we know is text...
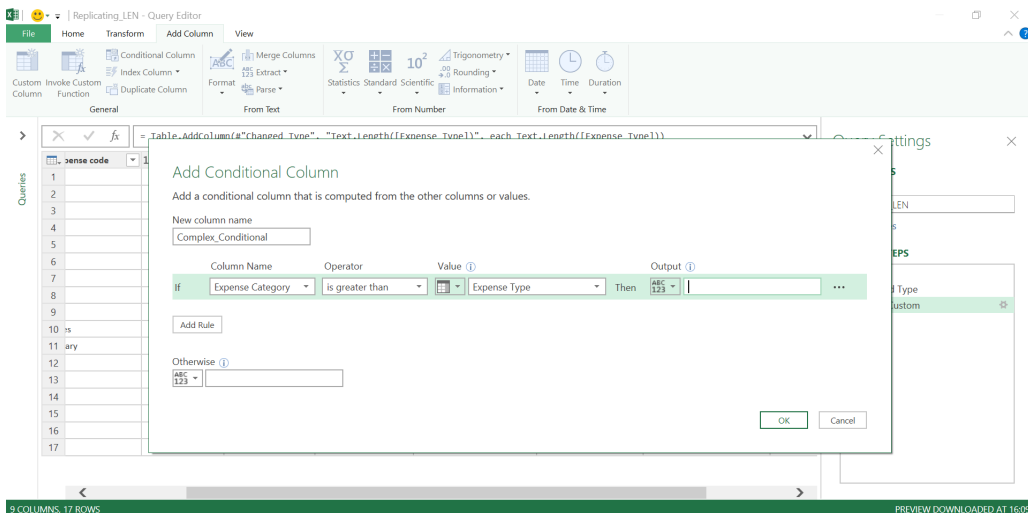
Unfortunately, Power Query will let us ignore the warning and continue, and it is not until we click 'OK' that it fails to execute the conditional statement:
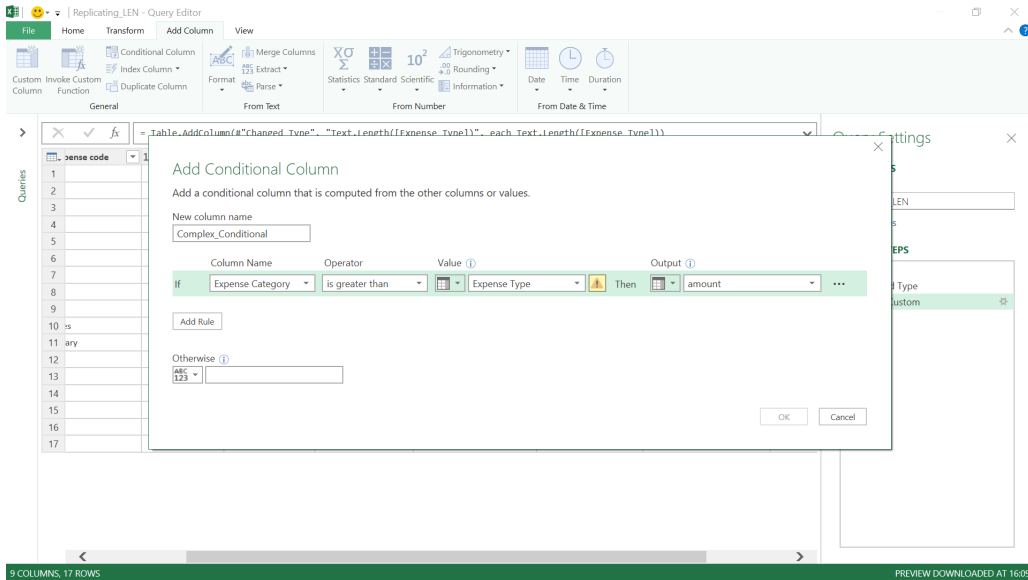
The warning icon is back and we have to go back and change the column. Our final option in the 'Value' box is to pick a 'Parameter'. However, in this case it is greyed out. This is because we have no parameters defined in our query. Parameters and their uses deserve a whole article of their own, but for now the screen below shows where parameters would be entered. On the 'Home' tab there is a section on 'Parameters' which allows us to manage, edit and add them.
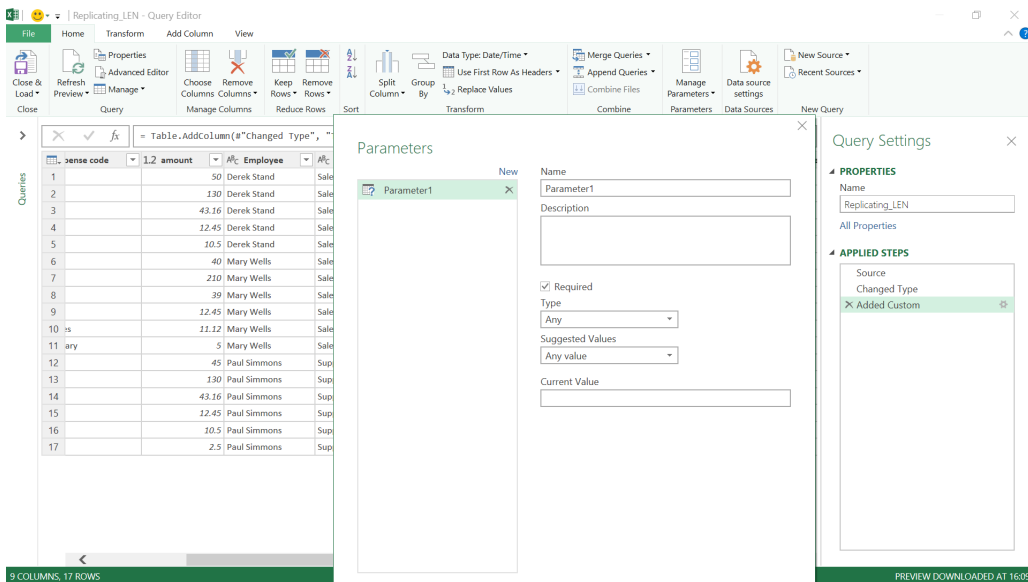
Back to the conditional column – we have picked our numeric field **Expense Category**, chosen to determine if it is greater than 200, and now we can decide what appears in our conditional column.
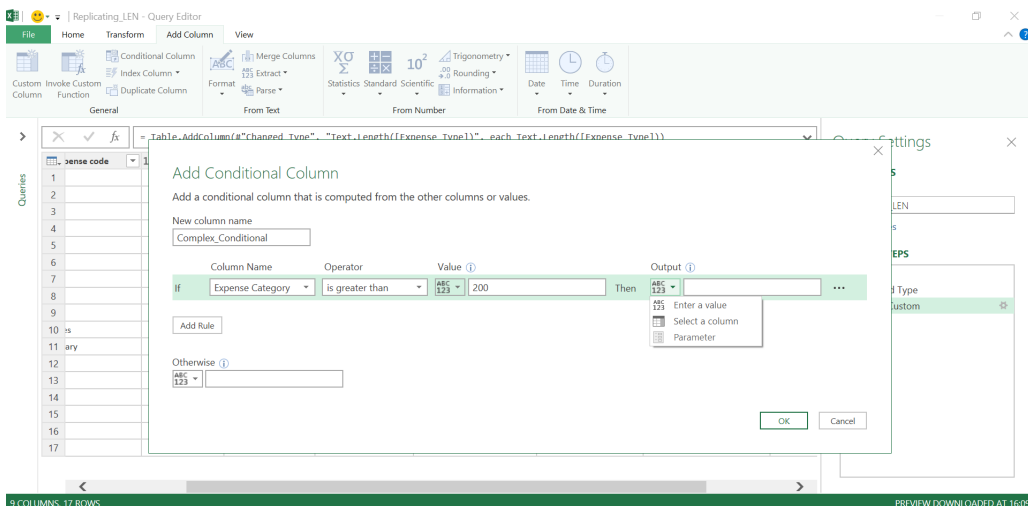
Again, we have three options: we can output a value, a column or a parameter. In this case, we can output any datatype, so there are no checks. Let's choose to show that the expense is 'Not food' and move on to decide whether to enter more rules or to just pick what happens 'Otherwise':
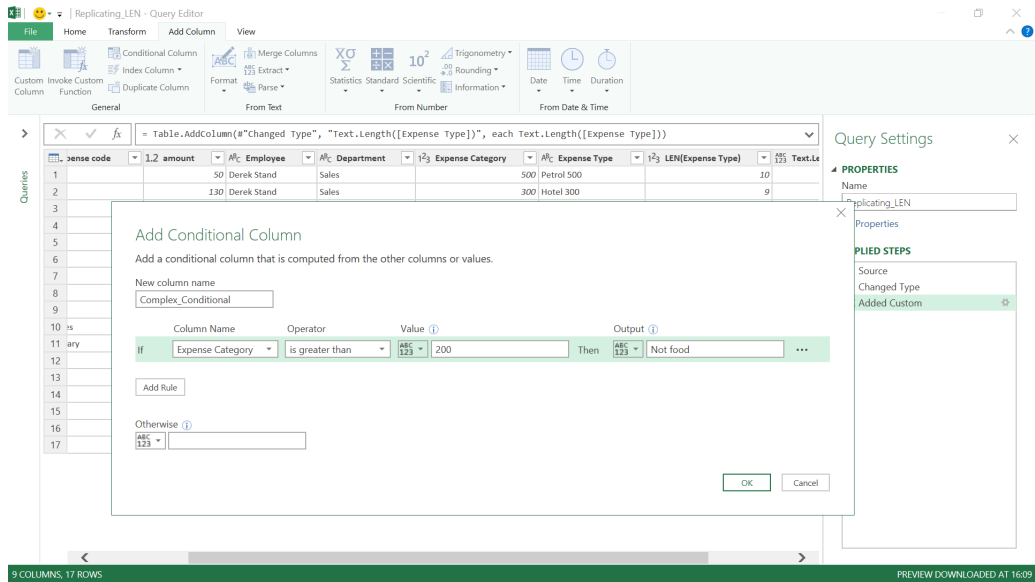
We'll choose to add a few more rules and specify a value in the 'Otherwise' section (which again can be a value, column or parameter).
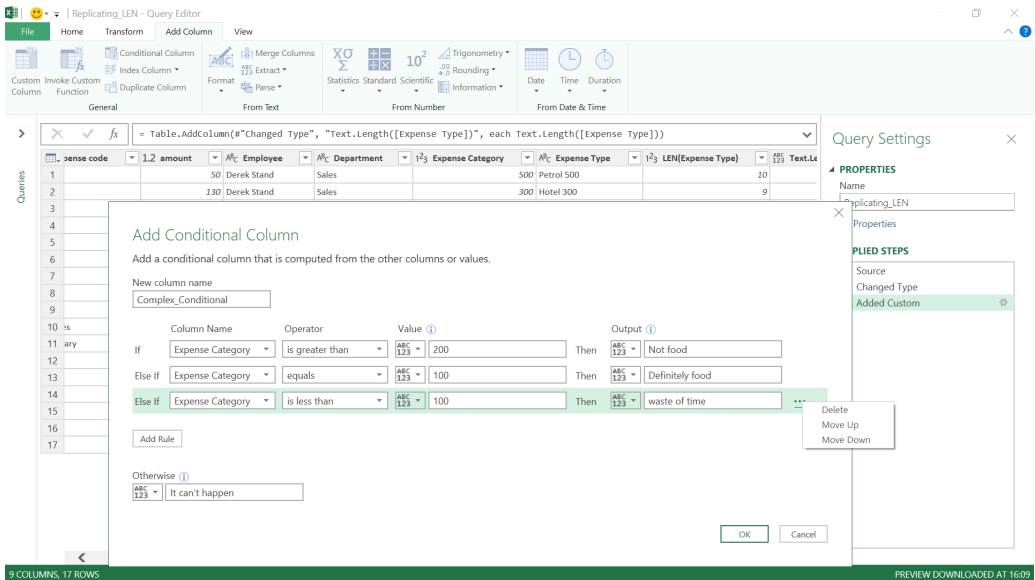

Note that, if all our data is covered by the first couple of rules, there is no point adding more, as they will never be applied. We can move rules up or down to change the priority of how they are applied else delete them altogether. Click 'OK' to see the new column and our new step.
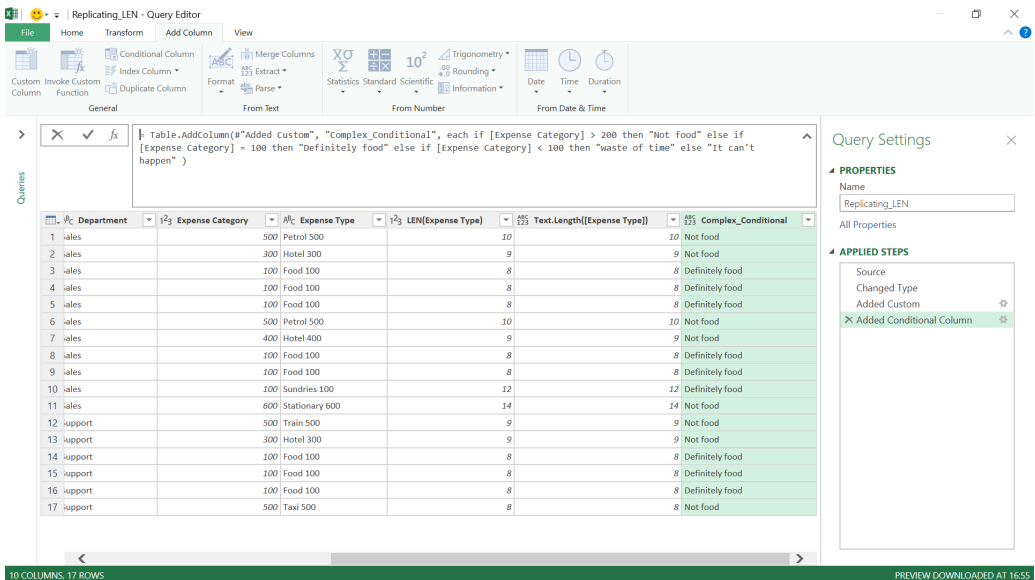
The column looks fine and in the window above the data we can see the **M** code statement:

```
= Table.AddColumn(#"Added Custom", "Complex_Conditional", each if [Expense Category] > 200 then "Not food" else if [Expense Category] = 100 then "Definitely food" else if [Expense Category] < 100 then "waste of time" else "It can't happen" )
```

It's a bit of a mouthful, so we can tidy it up and make it easier to read. We can either do this in the 'Advanced Editor' which will show us all of the steps or if we just want to change this step, we may do so by clicking in the box and lining up our data accordingly. To do this in the box, if we want to add a line break, use **ENTER** whilst holding down the **SHIFT** key.
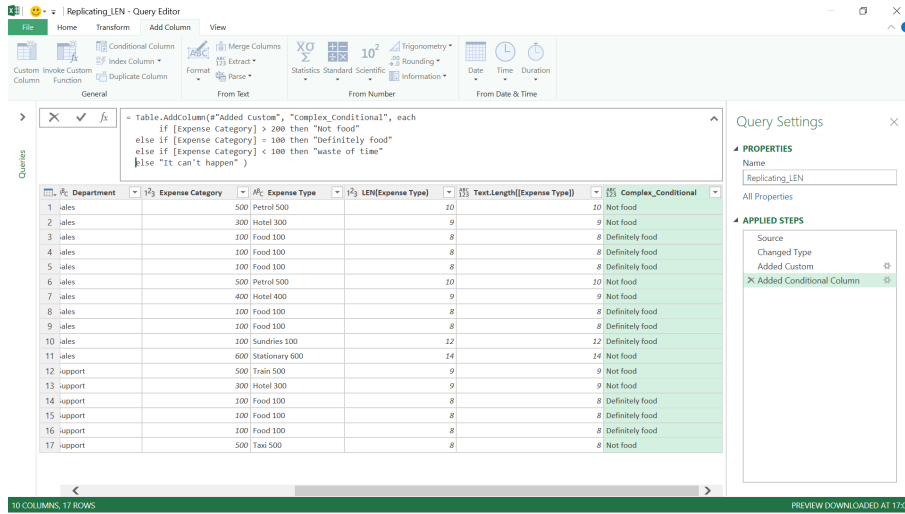
Now that is much easier to read and I can follow what the **M** code is doing.

More next month!

New AI and Enterprise features for Power BI: An Overview

In a world in which data is coming from everywhere, Power BI may assist everyone in making better decisions based on data. The growth of Power BI has been staggering: according to Microsoft, customers ingest more than 20PB of data to Power BI every month, lighting up over 30m reports and dashboards, and the Power BI service processes over 12m queries *per hour*.

To build such a data culture, everyone in an organisation needs to be empowered to work with data. At its core, BI is a collaborative process. Microsoft has considered this when improving the following capabilities:

- **New look for Power BI Desktop:** The new UI for Power BI Desktop allows you to explore data, get answers and make decisions without relying on learning specialist skills. It employs the familiar PowerPoint experience to accelerate the ability to perform data analysis regardless of role

- **New look for the Power BI service:** Report consumers, on the other hand, will have a new consumption experience including new navigation User Interface (UI) elements to match the consistent fluent design language allowing users to find and understand content more easily

- **Siri Shortcuts:** Microsoft has recently announced that the Power BI Mobile iOS app is now integrated with Siri shortcuts (albeit in Preview). Yay – now bring on Clippy (not)! Now, you'll be able to just say "Hey Siri, open my report" and you'll get the content you're looking for

- **Visual level filtering on slicers:** You may now utilise visual level filters for slicers

- **Personalized Visualization pane:** The personalized Visualization pane is available, giving report authors the ability to choose which marketplace visuals are available for them persistently in Power BI Desktop. Marketplace visuals are widely used by the Power BI community and now they are easier than ever to integrate into Power BI reports.
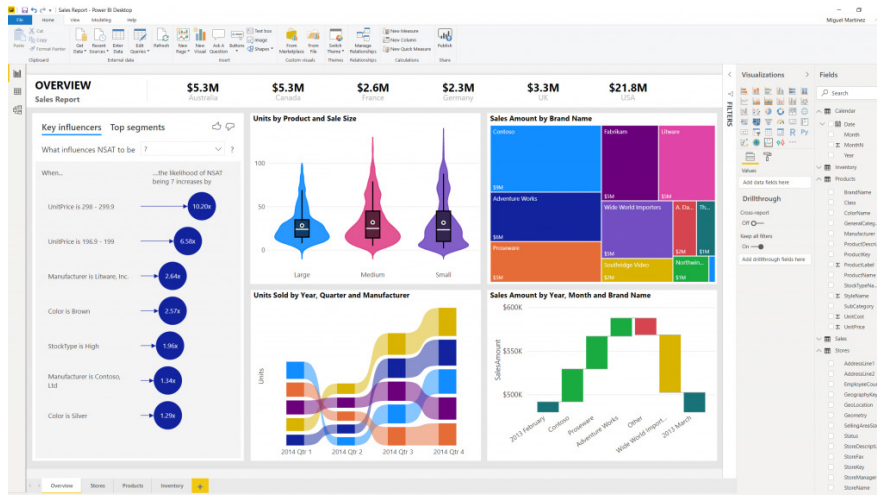
With Power BI, you can get the best of self-service analytics with centralised reporting capabilities, all atop a single enterprise-grade platform with unified governance and security. To this end, Microsoft has announced the following new and improved capabilities:

- **Paginated Reports General Availability:** Based upon the widely-used reporting technology in SQL Server Reporting Services, Paginated Reports bring new enterprise reporting capabilities to Power BI Premium. Useful for highly formatted, pixel-perfect reports with a variety of export and subscription options, Paginated Reports can be published in Power BI alongside existing interactive reports and can be built atop the same Power BI datasets
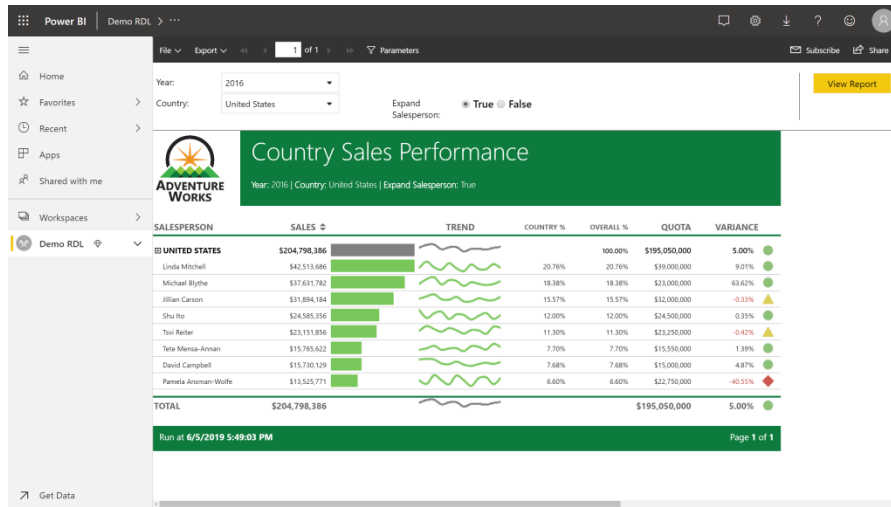
- **Shared and Certified Datasets Public Preview:** A key enabler of data culture in organisations is the pervasive availability of standard, authoritative datasets that represent a single source of truth, allowing users to make decisions on trusted data, remix to create insights, all with unified governance. Shared datasets are a set of features that enable this reuse. Microsoft has now completed the release of shared and certified datasets in the Power BI Service and have released support in the June release of Power BI Desktop:

- **Public Preview of calculation groups:** Enterprise-scale models can include hundreds of measures to facilitate diverse reporting and exploration tasks. Calculation groups simplify the presentation of similar measures to end users as well as improve management for BI professionals. Calculation groups are available now in Azure Analysis Services and will be coming soon to Power BI

- **Aggregations General Availability coming soon, including support for row-level security (RLS):** Aggregations will reach General Availability shortly (if not by the time this newsletter is published), enabling interactive analysis over petabyte-scale datasets. By GA, Microsoft means they will add support for RLS datasets. This will have huge ramifications for big datasets

- **Enhanced dataflows compute engine Public Preview:** As organisations accumulate more and more data, it becomes increasingly important to quickly perform extraction, transformation and loading (ETL) at scale over these datasets. There's now a new compute engine for dataflows to aid with this process. Now you will be able to efficiently combine and transform different data sources and manage TB's of data using dataflows. With the new engine, calculation of computed entities may be up to 20 times faster when working with large scale data volumes

- **Bring Your Own Key (BYOK) Public Preview:** With BYOK, customers can control the key used by Microsoft to encrypt data at rest in Power BI Premium. The Preview will be made soon.

Power BI has also amalgamated AI with BI, through capabilities like Quick Insights which help unearth trends in data and Q&A which enable business users to get answers by simply asking questions. Microsoft has recently announced the General Availability of Azure Cognitive Services and Azure ML dataflows integration, providing analysts with a toolkit of powerful AI functions such as:

- **Azure Cognitive Services** are sophisticated pre-trained machine learning models for intelligent applications. Analysts can use these models to extract insights from images by detecting objects. Text fields like customer feedback may be analysed for positive and negative sentiment as well as have key phrases extracted

- **Azure Machine Learning** is a powerful platform where data scientists can develop machine learning models. These models can now easily be shared and used by analysts. Power BI automatically discovers which models an analyst has permissions to and provides an intuitive point and click user interface to invoke them. Analysts can now easily collaborate with data scientists as well as visualise and use insights from the model in their reports.

There's more coming soon:

- Two new AI visuals: Distribution Changes analyse what makes a distribution look different, and the Decomposition Tree enables users to drill into any dimension to understand what is driving a key metric

- Text analytics capabilities will be improving

- Microsoft is adding entity detection and text and handwriting recognition, enabling one-click transformations for insights on unstructured data

- For enterprises that need custom lifecycle management or further tuning, models created in Power BI can be exported to Azure ML

- Power BI's natural language capabilities are to be extended. The new updates include the ability to train Q&A so it understands and adapts to company-specific language like synonyms, phrasings, or specific domains, and the ability for report authors to see every natural-language question asked so they can adjust how Power BI responds.
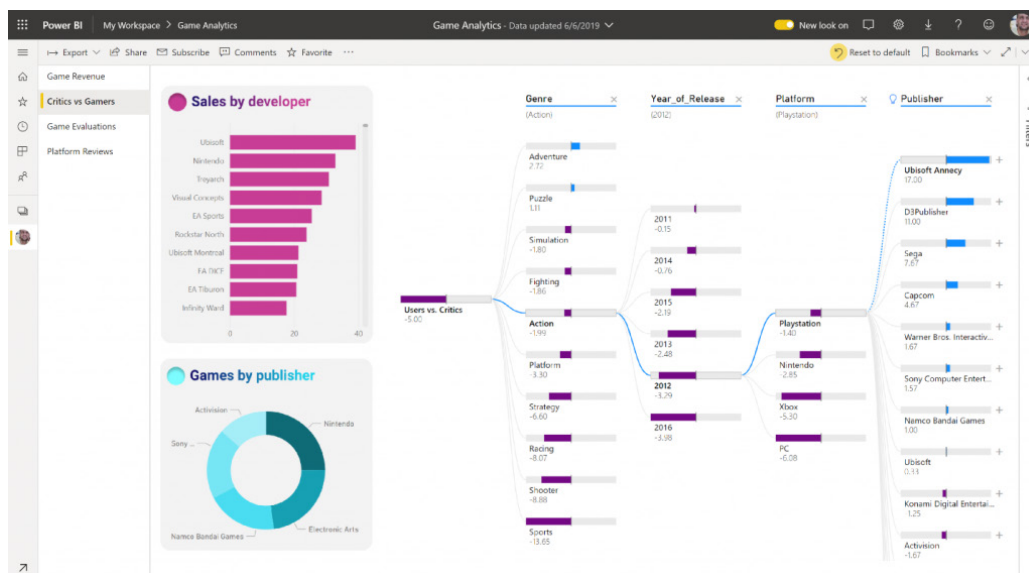
For those looking to ship Power BI Apps to customers, there's good news too.

Template apps have been developed and are now Generally Available. These are integrated packages of pre-built Power BI dashboards and reports, configured to connect to specific data sources. With them, Microsoft partners can quickly provide analytics for the apps and services they provide. Partners can also manage the Template Apps development lifecycle, from dev to marketplace to updates. With template apps, users can immediately begin exploring, learning, and acting on key data with off-the-shelf apps. Users can find and install template apps on AppSource.

Microsoft continues to roll out and try and lead the BI App marketplace. Here's a quick summary of other improvements out now in some shape or form:

- **Service principals with Power BI are now Generally Available.** With service principal, application developers can authenticate an external application to embed Power BI content or manage and automate Power BI operations with Power BI using an app-only token

- **Use the AI-based 'Key Influencers' visualisation** in your application to allow end users to see which factors affect the metric being analysed and contrast the relative importance of these factors

- **Quick Visual creator.** A new API to create visuals in an Embedded Power BI report, enabling application developers to respond to user interactions within their application by embedding an empty canvas and filling it with visuals made with code

- **Explore Power BI API with the 'Try-it' tool.** 'Try-it' enables you to explore how each REST call works, what are the required params, the structure of response or errors, and what data comes backcode

- **Performance improvements for embedding** Power BI with the new release of the client-side software development kit (SDK).

More anon.

## New Viewer Role for Power BI Workspaces

Power BI has also started to roll-out the Viewer role for the new workspace experience, useful for users who need to view content within workspaces without the ability to modify it.

The Viewer role gives a read-only experience to its users. They can view dashboards, reports or workbooks in the workspace, but can't browse the datasets or dataflows. You may use the Viewer role wherever you would previously use a classic workspace set to 'Members can only view Power BI content'.

Viewer role requires a Pro license or for the content to be in Power BI Premium. If your workspace is in Premium, users with Viewer role who don't have a Pro license can view the workspace in the workspaces list, navigate to the workspace, and view the content without getting a Pro Trial prompt.

This Viewer role has several benefits:

- It is easier to navigate in Power BI for users with read-only access

- It helps smaller teams that want stakeholders to have read-only access to all workspace content, but do not want to publish and maintain a Power BI app for this purpose

- Any row-level security (RLS) you have defined on the datasets in the Power BI workspace is enforced for users who only have the Viewer role

- You can stop using classic workspaces based on Office 365 Groups if you needed the Members to only view Power BI content option, known as 'read-only workspaces'.

For example, when you grant access in a new workspace experience workspace, you'll see a new role called Viewer in the roles option:
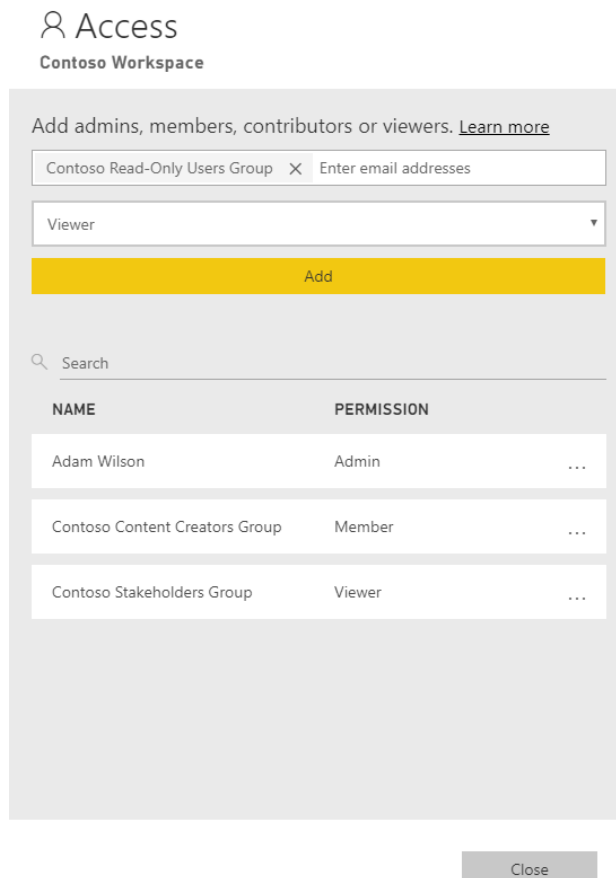
Users with the Viewer role get a read-only experience in the workspace. They can view dashboards, reports or workbooks in the workspace. They can't browse the datasets or dataflow or edit content in the workspace. This matches the experience users get in a classic workspace when they are set to 'Members can only view Power BI content'.
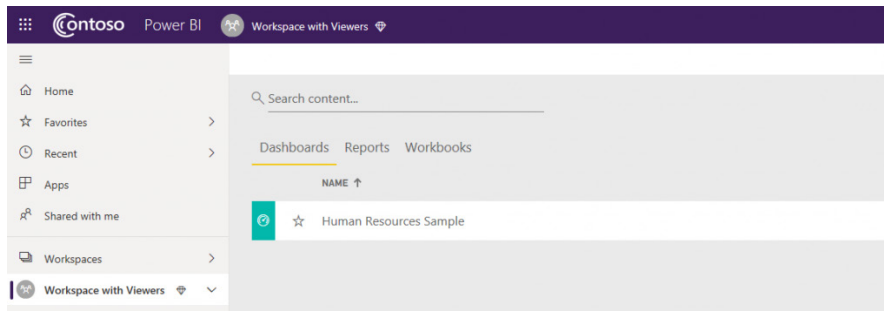
One important difference with the Viewer role is how licensing is enforced. If the workspace is in a Power BI Premium capacity, users who have the Viewer role but do not have a Pro license, can now navigate directly to the workspace and view the content. This helps smaller teams that want stakeholders to have read-only access to all workspace content, but do not want to publish and maintain a Power BI app for this purpose.

If the workspace is not in Power BI Premium, or if the user without a Pro license has a higher privilege role, like Admin, Member, or Contributor, they will continue to get the standard 'Upgrade to Power BI Pro license' dialog.
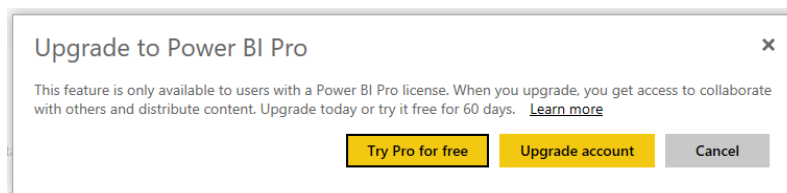
The Viewer role grants only the Read permission to items in the workspace. This means that any RLS you have set on the datasets in the workspace applies to users who only have the Viewer role. If you want these users to 'Analyze in Excel', or export data from the datasets in the workspace, also give them Build permission on the appropriate datasets.

More soon we expect!

Power BI Desktop Update for July

With this month's update comes the first feature release from Microsoft's recent announcements at the Microsoft Business Application Summit (MBAS): icon sets. There are also incremental improvements for many of Power BI's existing features, such as a counts experience for the Key Influencers visual and RLS support for aggregations. Further, Microsoft is also making several key features, such as the new filter pane and aggregations, Generally Available.

There's more going on behind the scenes too. For those not using the Microsoft Store version, the link downloads a new, more efficient, installer. Before, Microsoft had to put together 84 different .msi files for installing Power BI: 42 languages are supported and there also need to be 32-bit 64-bit variants. Now, there's just two.

Anyway, here's the complete list of July updates:

Reporting

- Icon sets for table and matrix
- Percent support for conditional formatting by rules
- New Filter pane is now Generally Available
- Data colours support when using play axis on scatter charts
- Performance improves when using relative date and dropdown slicers

Reporting

- Counts for Key Influencers visual (Preview)
- Aggregations improvements
  - o RLS support
  - o Aggregations now Generally Available

Visualisations

- PowerApps visual is now certified
- InfoCard
- Sunburst by MAQ Software
- Flying Brick visual

Data connectivity

- Azure Data Lake Storage Gen2 connector (Beta)
- Azure Cosmos DB Connector is now Generally Available
- Dynamics 365 Customer Insights connector

Data preparation

- New transform: Split column by positions.

Let's take a look at each in turn.

Icon sets for table and matrix

One of the top requests Microsoft keeps receiving is for KPI icons for reports. This update sees the release of Microsoft's "first step for this work", namely icon sets for Power BI's table and matrix visual. This feature will let you set up rules for showing different sets of icons to show in your table and matrix, very similar to the 'Icon Sets' feature in Excel"

| Country	Large	Medium	Small	Total
Australia	$8,098	$2,090	$358	$1,052
Deluxe	🟢 $8,369	🔶 $2,479	🔶 $476	$2,646
Economy		🔶 $1,640	🔶 $251	$315
Regular	🟢 $7,135	🔶 $2,004	🔶 $445	$1,098
Canada	$8,454	$2,098	$366	$1,015
France	$8,539	$2,099	$352	$1,049
Germany	$8,430	$2,045	$358	$1,084
Great Britain	$8,453	$2,124	$350	$1,075
Deluxe	🟢 $8,859	🔶 $2,505	🔶 $460	$2,724
Economy		🔶 $1,765	🔶 $237	$329
Regular	🟢 $7,012	🔶 $2,039	🔶 $443	$1,114
United States	$8,299	$2,077	$361	$1,050
Deluxe	🟢 $8,662	🔶 $2,506	🔶 $490	$2,720
Economy		🔶 $1,519	🔶 $251	$316
Regular	🟢 $6,977	🔶 $1,998	🔶 $454	$1,109
Total	$8,323	$2,085	$360	$1,049

You can turn on icons for your table or matrix just like any other conditional formatting feature, either through the contextual menu for the field you want to apply it to or through the 'Conditional Formatting' card in the Formatting pane.
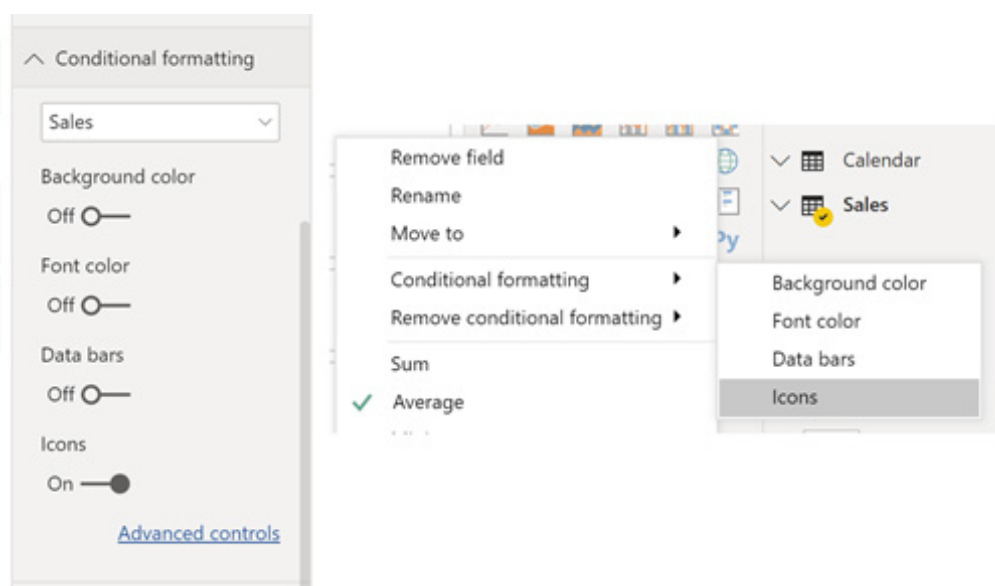
When you turn on Icons, the default rules will apply: a red diamond to the lower 33% of your data, a yellow triangle to the middle 33%, and a green circle to the upper 33%.
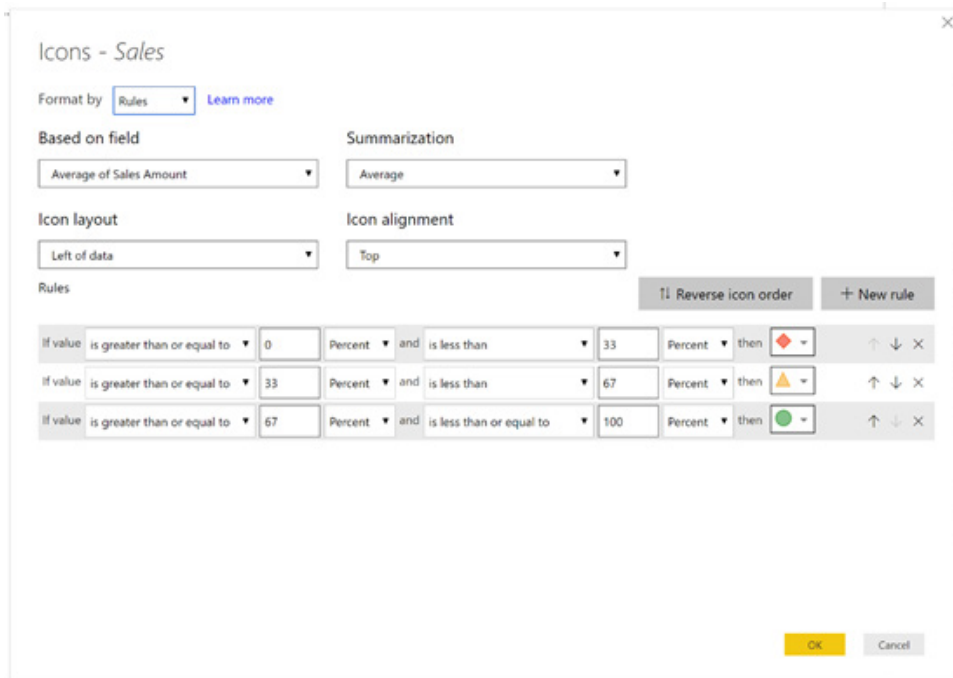
For each rule, you can set the range based on an absolute number or a percent. You can also customise the icon for each rule with any of the icons supported in the icon picker.
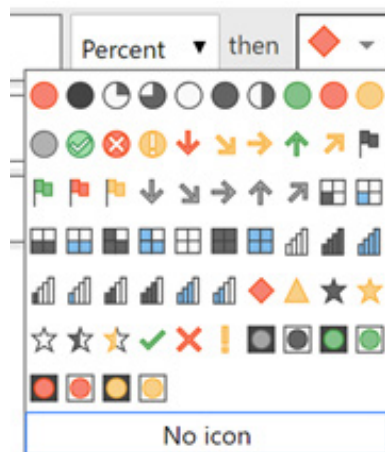
In addition to adjusting current rules, you can add new rules to the list using the 'New rule' button and swap the order of the icons in all the rules with one click using the 'Reverse icon order' button.
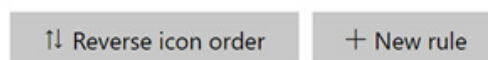
In this dialog, you can also change the field used to generate the icons, just like in other forms of conditional formatting. You can also choose, using the 'Icon layout' dropdown, whether the icon shows to the left or right of the numbers showing in the cell, or if the icons should replace the numbers all together.
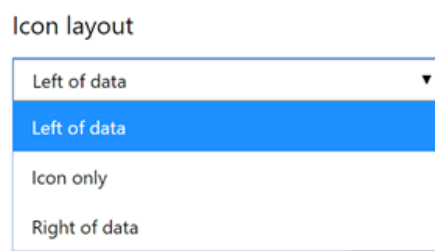
You can also control the icon alignment, switching between top, middle, and bottom alignment:
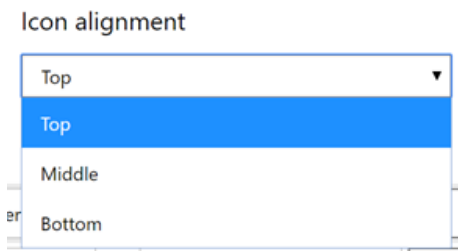
You can also format based on fields in your model, where the fields use the name of any of the existing icons, defines svgs inline, or references an image or gif online. For example, you can create a measure such as:

```
Icon Set Measure =
SWITCH(
    TRUE(),
    SUM(Sales[Units]) < 200, "data:image/svg+xml;utf8, <svg
xmlns='https://www.w3.org/2000/svg' viewBox='0 0 100 100'> <circle cx='50' cy='50' r='40'
stroke='purple' stroke-width='4' fill='purple' /> </svg>",
    SUM(Sales[Units]) < 1000, "StarMediumLight",
    "https://i.gifer.com/Omjx.gif"
)
```

and apply it to get a completely custom set of icons.
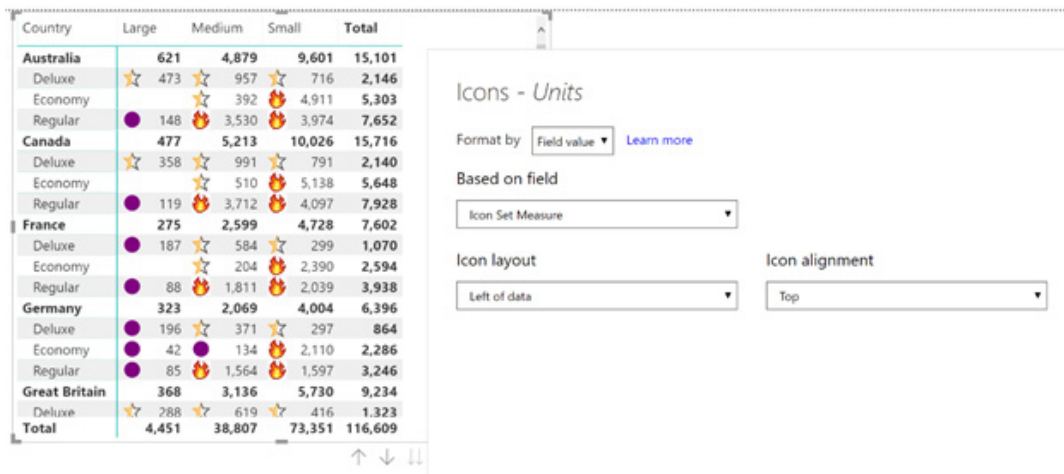
You can also change icons or add new ones through a theme file. For example, the below theme will add two new icons and replace the half-filled yellow star with a different icon:

```
{
    "name": "CustomIcons",
    "icons": {
        "fire": {
            "url": "https://i.gifer.com/Omjx.gif",
            "description": "Fire!"
        },
        "explosion": {
            "url": "https://i.gifer.com/3iCN.gif",
            "description": "Boom!"
        },
        "StarMediumLight": {
            "url": "https://www.uscsd.k12.pa.us/cms/lib/PA01000033/Centricity/Domain/1/library-icon.png",
            "description": {"resourceKey": "Icon_ColoredArrowUpRed"}
        }
    }

    }
}
```
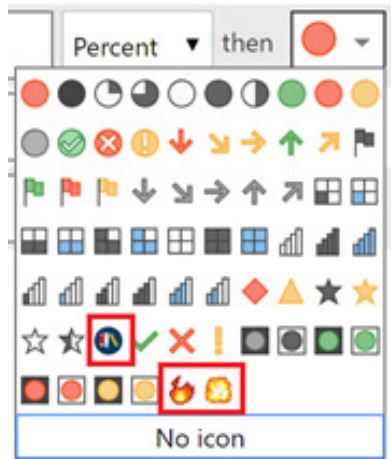
Each icon you are adding or modifying will get its own group in the icons section of the theme. The structure of each icon is:

```
"IconID": {
     "url" : " https://i.gifer.com/3iCN.gif ",
     "description": "Boom!"
}
```

The IconID section is either a unique name you are giving a custom icon or the id of any of the built-in icons you wish to modify. The URL is where you'd include either a link to web image you want to reference or embed your SVG or base64 encoded image. Lastly, the description is what will be used for the ToolTip of the icon and read out by a screen reader.

If you are using SVGs either through measures or the theme file, make sure to include the viewBox section of the code, where the viewBox is the original canvas size that was used when the SVG icon was made. This will ensure the SVG will continue to scale as the font size of your values changes.

"data:image/svg+xml;utf8, <svg xmlns='https://www.w3.org/2000/svg' viewBox='0 0 100 100'> <circle cx='50' cy='50' r='40' stroke='purple' stroke-width='4' fill='purple' /> </svg>"

In a future update, Microsoft plans to add a dropdown that allow you to pick from preset rules, similar to Excel's icon sets, to make it faster for you to create rules.

Percent support for conditional formatting by rules

In addition to adding a new type of conditional formatting, this update has extended in general the Rules form of conditional formatting to support Percent, which allows you to apply formatting on a dynamic range of data, instead of having to specify an absolute number that can become absolute as data refreshes. The 'Reverse button' has also been added from the Icon dialog for colours as well, so it's easier to quick swap the direction of your rules.
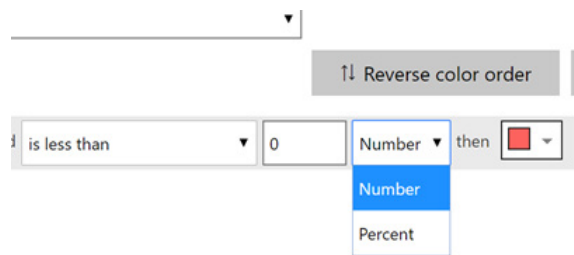

New Filter pane is now Generally Available

The new Filter pane is now Generally Available. Any new reports should automatically have the new pane on by default. For now, you'll still need to upgrade any existing report to the new pane through the Options dialog.

Data colours support when using play axis on scatter charts

You can now set the data colour for the scatter chart when using the play axis. Up until now, you could only vary / use data colours when using a scatter chart without a play axis, so this should add more flexibility when designing your reports if you're using the play axis.
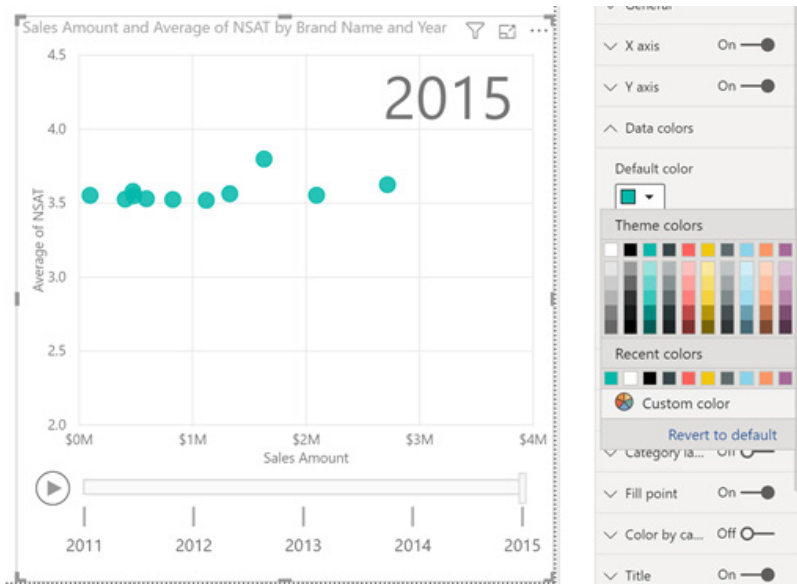
Performance improves when using relative date and dropdown slicers

If you are using the relative date slicer or the dropdown slicer, you should see some performance improvements for page loads and filtering interactions with this update. The relative date slicer will no longer run any DAX queries when loading, and the dropdown slicer will only run queries when you expand the dropdown. The more slicers you had on your page, the bigger performance improvement you should see.

Counts for Key Influencers visual (Preview)

The latest improvement for the Key Influencers visual is the addition of counts. Counts lets you get a picture of how much of your data is represented by a particular influencer, so you can focus on the influencers that impact many of your users. For example, you might have an influencer that has a very large impact, but only a handful of users are affected by it.

You can turn counts on through the Analysis card of the Formatting pane:
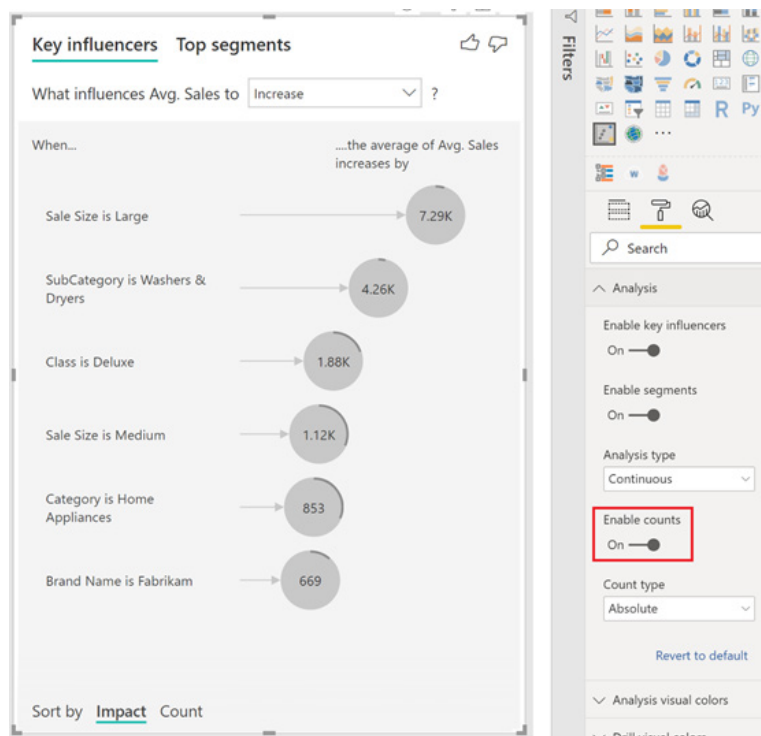
Once counts are turned on, you'll see a ring around each influencer's bubble, which represents the approximate percentage of data that influencer contains. The more of the bubble the ring circles, the more data it contains.

You can also use the 'Sort by' toggle in the bottom left of the visual to sort the bubbles by count first instead of impact.
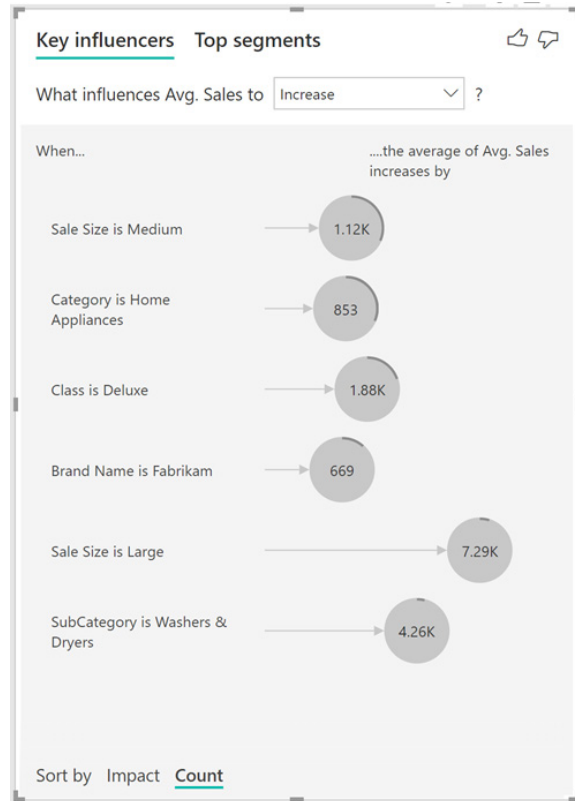
You can also change the count type to be relative to the maximum influencer using the 'Count type' dropdown in the Analysis card of the Formatting pane.
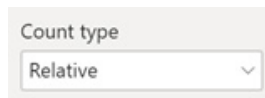
Aggregations improvements

RLS SUPPORT

RLS (row-level security) and aggregations are now supported in the same dataset. If this was preventing you from using this feature, you can now use aggregations in your reports.

AGGREGATIONS NOW GENERALLY AVAILABLE

The aggregations feature is now Generally Available and works end-to-end without needing to turn on a Preview feature switch.

PowerApps visual is now certified

The PowerApps marketplace visual is now certified. This means you can now see the PowerApps visual when exporting to PowerPoint and when you get report subscription emails.

InfoCard

The InfoCard marketplace visual is a card visual with advanced formatting options. Formatting options include:

- Large font sizes
- Custom borders with conditional formatting support
- Image support with position control
- Conditional formatting support for background, data label, category label and image
- Vertically rotate the category label.
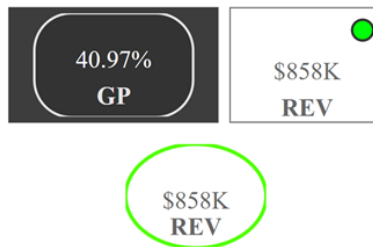
Sunburst by MAQ Software

The Sunburst marketplace visual by MAQ Software allows you to display hierarchical data as a set of nested rings. It's very similar to a donut chart visual, expect the hierarchy levels are nested within one another, allowing you to see the breakdown across the different categories more clearly.

Some key features of this visual include:

- Customisable arcs
- Customisable detail and data labels
- Zoom animations
- Support for cross-highlighting, drillthrough, report page tooltips and bookmarks.
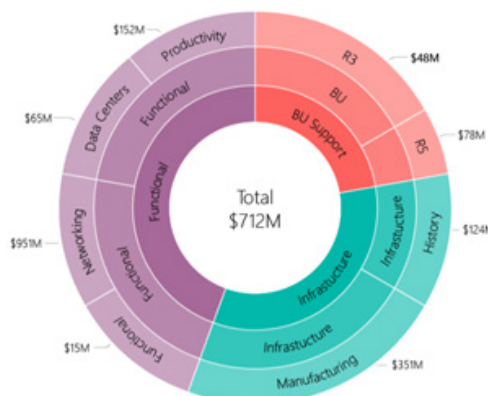
Flying Brick visual

Not the best nomenclature, we think. The flying brick marketplace visual by Balcones Enterprises is a waterfall chart visual that doesn't require a lot of work to get your data into the right format. Each category within the field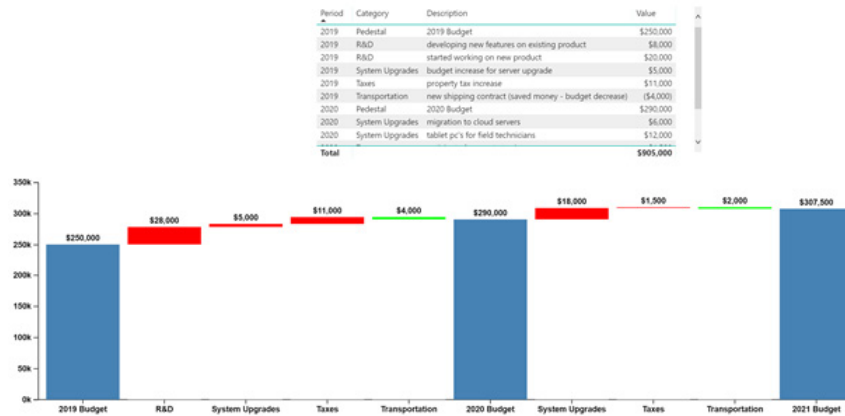 you use for the Category field well bucket will be used as a step of the waterfall chart. Any rows with 'Pedestal' as the category's text will be treated as the main columns of the waterfall chart. Below is an example of the flying brick visual next to a table showing the data its visualising.

Azure Data Lake Storage Gen2 connector (Beta)

This update sees a new connector that allows you to easily import your data from Azure Data Lake Storage Gen2. This is a highly scalable and cost-effective data lake solution for big data analytics. It combines the power of a high-performance file system with massive scale and economy to help you accelerate your time to insight. Data Lake Storage Gen2 extends Azure Blob Storage capabilities and is optimised for analytics workloads.

The new Azure Data Lake Storage Gen2 connector can be found within the 'Get Data' dialog, under the Azure category.

Azure Cosmos DB Connector is now Generally Available

The Azure Cosmos DB connector has now reached General Availability.

Azure Cosmos DB is Microsoft's globally distributed, multi-model database service. With a click of a button, Cosmos DB enables you to elastically and independently scale throughput and storage across any number of Azure regions worldwide. You can elastically scale throughput and storage, and take advantage of fast, single-digit-millisecond data access. Cosmos DB provides comprehensive service level agreements (SLAs) for throughput, latency, availability, and consistency guarantees.

You can use the connector to query and import data from Azure Cosmos DB containers into Power BI to create and publish reports. The connector now supports pushing down SQL queries to Azure Cosmos DB which allows for faster execution times.

The Azure Cosmos DB connector can be accessed from the Azure category within the Get Data dialog.

### Dynamics 365 Customer Insights connector

The Dynamics 365 Customer Insights connector becomes available in this month's release. Dynamics 365 Customer Insights helps you unify your data across sources to get a single view of customers, optimise engagement and discover insights. Customer Insights shorten time to value with a ready-to-go customer data platform (CDP) that empowers marketing, sales and service professionals to tailor digital and one-on-one interactions at scale.

The new Dynamics 365 Customer Insights connector can be found within the Get Data dialog, under the Online Services category.

New transform: Split column by positions

A new data transformation has been added in this update that allows you to split text columns at specific positions within a given text value. This new transformation can be found under the Split Column menu in the Home tab of the Power Query Editor Ribbon.
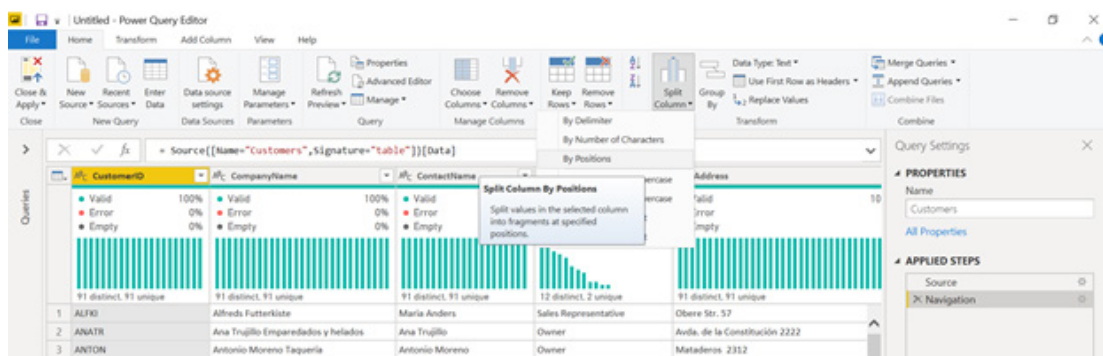
You can specify a comma-separated list of positions to split at. Power Query also tries to detect and make recommendations for those positions based on data in preview rows within the Power Query Editor. You can also customise whether to output text parts as new columns or rows.
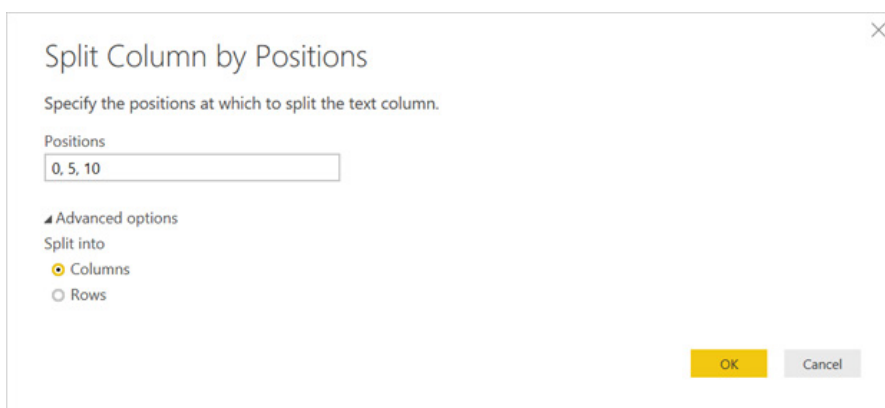
More next month we're sure!

Latest Updates for Power BI Service and Mobile

And it keeps coming! How we yearn for a two-page newsletter once more…

Here's the latest updates for Power BI Service and Mobile:

- Shared and certified datasets (Preview)
- AI metrics in the Premium Capacity Metrics app
- Update for on-premises data gateway
- General Availability of Power BI template apps
- General Availability of Paginated Reports in Power BI
- Bring Your Own Key (BYOK) for Power BI Premium (Preview)
- Viewer role for the new Power BI workspaces experience
- Updates to Power BI Mobile apps
- Roadmap updates.

Let's take a look…

Shared and certified datasets (Preview)

First up is a public Preview of shared and certified datasets. With this feature, you can now use a single dataset to build multiple reports across different workspaces. That's right: dataset owners only need to maintain a single copy of their datasets in their own separate workspace and provide permissions for report authors to use those datasets to build reports. To help users discover relevant shared datasets, Microsoft has released a new discovery experience in the Power BI Service and Desktop to make it easier for you to browse and find content. Lastly, as part of this work, new capabilities have also been added for dataset owners and tenant administrators to control the use of shared data.
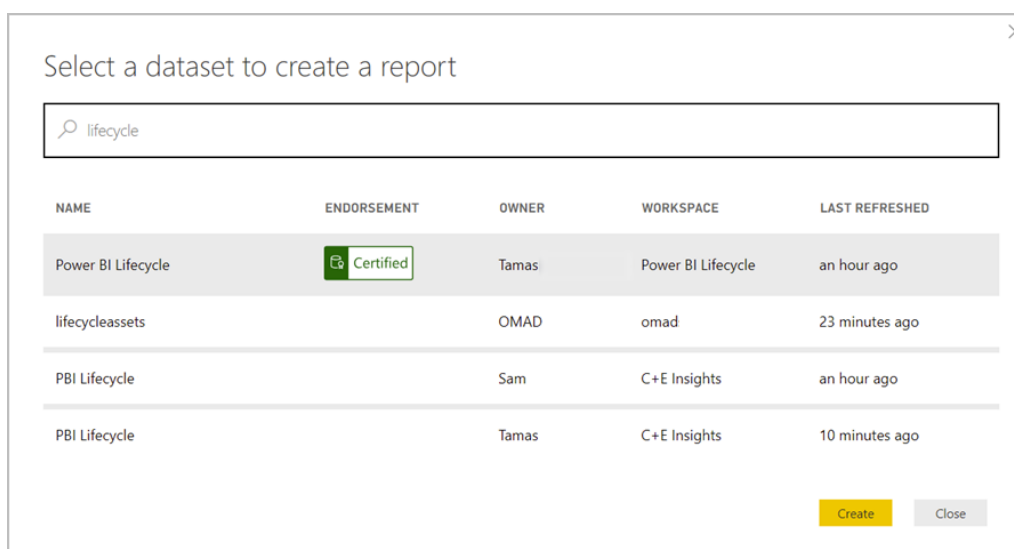
AI metrics in the Premium Capacity Metrics app

A new version of the Power BI Premium Capacity Metrics app has been added to include new AI metrics that provide details on the health of your AI workloads within your capacity. The latest version (1.10.1.7) of the app also has an AI summary dashboard to show you seven-day summaries of all the capacities that you are an administrator for.

Update for on-premises data gateway

Microsoft also updated the Power BI On-premises data gateway. Version 3000.6.204 has a new mashup engine, performance profiling based on gateway CPU, SAP Common Crypto library support for SAP BW Application Server Kerberos based SSO, and other small tweaks to keep your datasets up to date.

General Availability of Power BI template apps

In February, Microsoft announced a public Preview of template apps. These apps are integrated packages of pre-built Power BI dashboards and reports, configured to connect to specific data sources. With them, Microsoft partners can quickly access analytics for the apps and services they provide. Partners can also manage the Template Apps development lifecycle, from development to marketplace to updates. Now, these template apps have been made Generally Available.

General Availability of Paginated Reports in Power BI

Microsoft has also announced the General Availability of paginated reports. Based upon the technology used in SQL Server Reporting Services, paginated reports bring a new set of enterprise reporting capabilities to the Power BI Service. It enables developers and BI professionals to create and distribute highly formatted, pixel-perfect reports right alongside their interactive Power BI content, becoming the first cloud BI solution that combines the power of self-service BI with the needs and capabilities of traditional Enterprise BI scenarios.
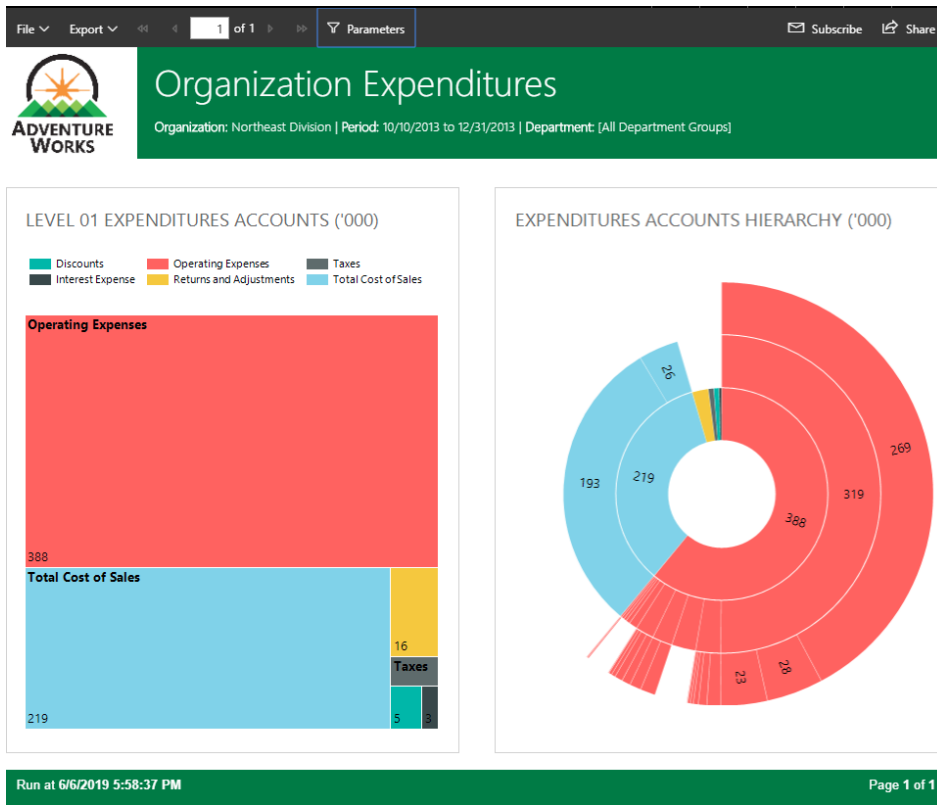
Bring Your Own Key (BYOK) for Power BI Premium (Preview)

Power BI encrypts data at rest for all customers. However, for organisations requiring additional control over how their data is encrypted, this update introduces Bring Your Own Key (BYOK). This feature allows organisations to configure Power BI to using their own keys, stored in Azure Key Vaults, for data-at-rest encryption instead of Microsoft managed keys. BYOK encryption applies to Power BI Premium capacities only, and organisations can choose to associate different capacities with different keys and / or different key vaults.

Viewer role for the new Power BI workspaces experience

Microsoft super-charged the new workspaces experience in Power BI by introducing the viewer role. Now, you can grant read-only access to users within workspaces. The viewer role requires a Pro license or for the content to be in Power BI Premium. If your workspace is in Premium, users with viewer role who don't have a Pro license can view the workspace in the workspaces list, navigate to the workspace, and view the content without getting a Pro Trial prompt.
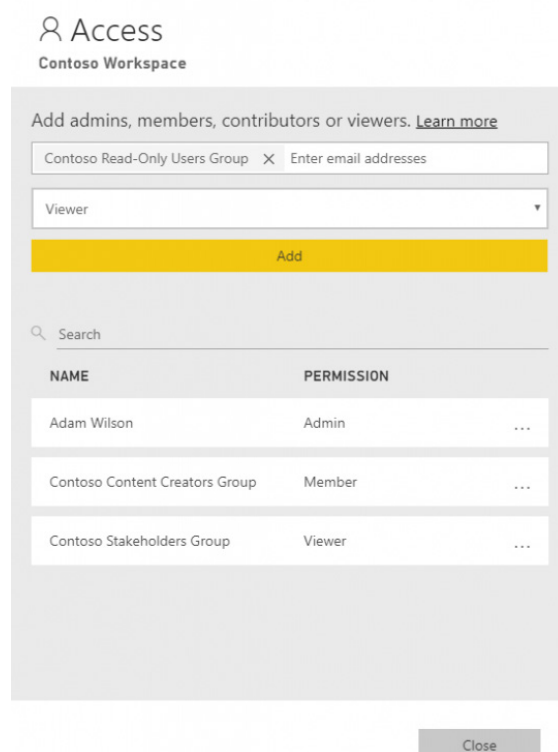
Updates to Power BI Mobile apps

There were multiple updates that went out recently for the Mobile apps in various platforms. Here's a quick summary:

- **Barcode scanning in Android:** The Power BI app on Android (phone and tablet) has been updated to scan barcodes printed on products or shelves at your store to display related Power BI reports filtered by the scanned value

- **Support for PBIX reports hosted in PBI-RS over ADFS configuration (iOS, Android):** Microsoft has added support for Power BI Reports (PBIX) hosted in PBI-RS over ADFS configuration in iOS and Android apps

- **Siri shortcuts support (iOS):** The iOS Power BI app has been updated to allow users to create Siri Shortcuts to their Power BI reports and dashboards, that can be accessed via voice later

- **Device level search (iOS):** You may now search Power BI content within iPhones and iPads that matches given search criteria

- **Protect Power BI with device Biometric ID:** Microsoft has "doubled down" on security by providing iOS app users (or admins) with the ability to configure the Power BI app to require Face ID, Touch ID, or passcode to view their data

- **Slideshow (Windows app)**: Finally, support has been added for auto-refresh for single-page reports in a slideshow. If the underlying data source of your report is updated, Power BI will pick it up and update the data on your slide automatically.

Roadmap updates

Latest disclosures (at a high level) include

- New experience for report consumption
- New Microsoft Flow actions
- Entity detection and OCR
- Data lineage capabilities.

Stay tuned for the next update.

The A to Z of Excel Functions: DEC2BIN

This function converts a decimal number (base 10) to binary (base two). It doesn't throw decimal numbers away…

The **DEC2BIN** function employs the following syntax to operate:

DEC2BIN(number, [places])

The **DEC2BIN** function has the following arguments:

- **number:** this is required and represents the decimal integer you wish to convert. If **number** is negative, valid place values are ignored and **DEC2BIN** returns a 10-character (10-bit) binary number in which the most significant bit is the sign bit. The remaining 9 bits are magnitude bits. Negative numbers are represented using two's-complement notation
- **places:** this argument is optional. This is the number of characters to use. If **places** is omitted, **DEC2BIN** uses the minimum number of characters necessary. The argument places is useful for padding the return value with leading 0s (zeros).

Two's complement is a mathematical operation on binary numbers, as well as a binary signed number representation based on this operation. The two's complement of an **N**-bit number is defined as the complement with respect to 2^N; in other words, it is the result of subtracting the number from 2^N. This is also equivalent to taking the ones' complement and then adding one, since the sum of a number and its ones' complement is all 1 bits. The two's complement of a number behaves like the negative of the original number in most arithmetic, and positive and negative numbers can coexist in a natural way.

It should be further noted that:

- if **number** < -512 or if **number** > 511, **DEC2BIN** returns the *#NUM!* error value
- if **number** is nonnumeric, **DEC2BIN** returns the *#VALUE!* error value
- if **DEC2BIN** requires more than places characters, it returns the *#NUM!* error value
- if **places** is not an integer, it is truncated
- if **places** is non-numeric, **DEC2BIN** returns the *#VALUE!* error value
- if **places** is zero or negative, **DEC2BIN** returns the *#NUM!* error value.

Here's an example:

	A	B	C
1	Formula	Description	Result
2	=DEC2BIN(9,4)	Converts decimal 9 to binary with four characters.	1001
3	=DEC2BIN(-100)	Converts decimal -100 to binary.	1110011100
4	=DEC2BIN(0)	Converts decimal 90 (zero) to binary.	0
5			

The A to Z of Excel Functions: DEC2HEX

Not to put a hex on it, but this function converts a decimal number (base 10) to hexadecimal (base 16).

The **DEC2HEX** function employs the following syntax to operate:

DEC2HEX(number, [places])

The **DEC2HEX** function has the following arguments:

- **number:** this is required and represents the decimal integer you want to convert. If **number** is negative, **places** is ignored and **DEC2HEX** returns a 10-character (40-bit) hexadecimal number in which the most significant bit is the sign bit. The remaining 39 bits are magnitude bits. Negative **numbers** are represented using two's-complement notation
- **places:** this argument is optional. This is the number of characters to use. If places is omitted, **DEC2HEX** uses the minimum number of characters necessary. The argument **places** is useful for padding the return value with leading 0s (zeros).

It should be further noted that:

- if **number** is < -549,755,813,888 or if **number** is > 549,755,813,887, **DEC2HEX** returns the *#NUM!* error value
- if **number** is non-numeric, **DEC2HEX** returns the **#VALUE!** error value
- if the result of **DEC2HEX** requires more than the number of specified **places** characters, it returns the *#NUM!* error value. For example, **DEC2HEX(**64,1**)** returns the error value because the result (40) requires two characters
- if **places** is not an integer, the value of **places** is truncated
- if **places** is non-numeric, **DEC2HEX** returns the *#VALUE!* error value
- if **places** is negative, **DEC2HEX** returns the *#NUM!* error value.

Please see an example below:

	A	B	C
1	Formula	Description	Result
2	=DEC2HEX(100,4)	Converts decimal value 100 ro hexadecimal with four characters ("padded" with two leading zeros).	0064
3	=DEC2HEX(-54)	Converts decimal value -54 to hexadecimal.	FFFFFFFFCA
4	=DEC2HEX(28)	Converts decimal value 28 to hexadecimal.	1C
5	=DEC2HEX(64,1)	Returns the *#NUM!* error because the result (4) requires two character places.	#NUM!
6			

The A to Z of Excel Functions: DEC2OCT

It may sound like 11 months of the year, however this function converts a decimal number (base 10) to octal (base 8).

The **DEC2OCT** function employs the following syntax to operate:

DEC2OCT(number, [places])

The **DEC2OCT** function has the following arguments:

- **number:** this is required and represents the decimal integer you want to convert. If **number** is negative, **places** is ignored and **DEC2OCT** returns a 10-character (30-bit) octal number in which the most significant bit is the sign bit. The remaining 29 bits are magnitude bits. Negative **numbers** are represented using two's-complement notation
- **places:** this argument is optional. This is the number of characters to use. If **places** is omitted, **DEC2OCT** uses the minimum number of characters necessary. The argument **places** is useful for padding the return value with leading 0s (zeros).

It should be further noted that:

- if **number** < -536,870,912 or if **number** > 536,870,911, **DEC2OCT** returns the *#NUM!* error value
- if **number** is nonnumeric, **DEC2OCT** returns the *#VALUE!* error value
- if **DEC2OCT** requires more than **places** characters, it returns the *#NUM!* error value
- if **places** is not an integer, it is truncated
- if **places** is nonnumeric, **DEC2OCT** returns the *#VALUE!* error value
- if **places** is negative, **DEC2OCT** returns the *#NUM!* error value.

Please see our final example for this month below:

	A	B	C
1	Formula	Description	Result
2	=DEC2OCT(58,3)	Converts decimal 58 to octal.	072
3	=DEC2OCT(-100)	Converts decimal value -100 to octal.	7777777634
4			

Upcoming SumProduct Training Courses

Location	Course	Date	Duration
Sydney	Excel Tips and Tricks	19 Aug 2019	1 Day
Melbourne	Financial Modelling	20 - 21 Aug 2019	2 Days
Sydney	Financial Modelling	20 - 21 Aug 2019	2 Days
Melbourne	Power Pivot, Power Query and Power BI	26 - 28 Aug 2019	3 Days
Sydney	Power Pivot, Power Query and Power BI	26 - 28 Aug 2019	3 Days
Melbourne	Excel Tips and Tricks	17 Sep 2019	1 Day
Sydney	Excel Tips and Tricks	23 Sep 2019	1 Day
Sydney	Financial Modelling	24 - 25 Sep 2019	2 Days
Sydney	Power Pivot, Power Query and Power BI	7 - 9 Oct 2019	3 Days
Melbourne	Financial Modelling	14 - 15 Oct 2019	2 Days
Melbourne	Power Pivot, Power Query and Power BI	22 - 24 Oct 2019	3 Days
Sydney	Excel Tips and Tricks	4 Nov 2019	1 Day
Sydney	Financial Modelling	5 - 6 Nov 2019	2 Days
Sydney	Power Pivot, Power Query and Power BI	11 - 13 Nov 2019	3 Days
Melbourne	Excel Tips and Tricks	25 Nov 2019	1 Day
Melbourne	Financial Modelling	26 - 27 Nov 2019	2 Days
Melbourne	Power Pivot, Power Query and Power BI	9 - 11 Dec 2019	3 Days
Sydney	Power Pivot, Power Query and Power BI	9 - 11 Dec 2019	3 Days
Sydney	Excel Tips and Tricks	16 Dec 2019	1 Day
Sydney	Financial Modelling	17 - 18 Dec 2019	2 Days

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This month, we thought we look at division (/) and multiplication. It's a bit short and sweet this month:

Keystroke	What it does
CTRL + Divide (/)	Select array
CTRL + Multiply (*)	Select current region

There are over 540 keyboard shortcuts in Excel. For a comprehensive list, please download our Excel file a 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 http://www.sumproduct.com/training.