

Sum Product

NEWSLETTER #87 - February 2020

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



It's Good News Month!

We are only a month in and there is plenty of good stuff to announce: dynamic arrays, **XLOOKUP** and **XMATCH** are noticeably propagating through Excel Office 365 now – but they have brought friends with them. You can also inspect the new features Workbook Statistics and Office Scripts (not to be confused with the similarly named Script Lab). Find out more in this newsletter.

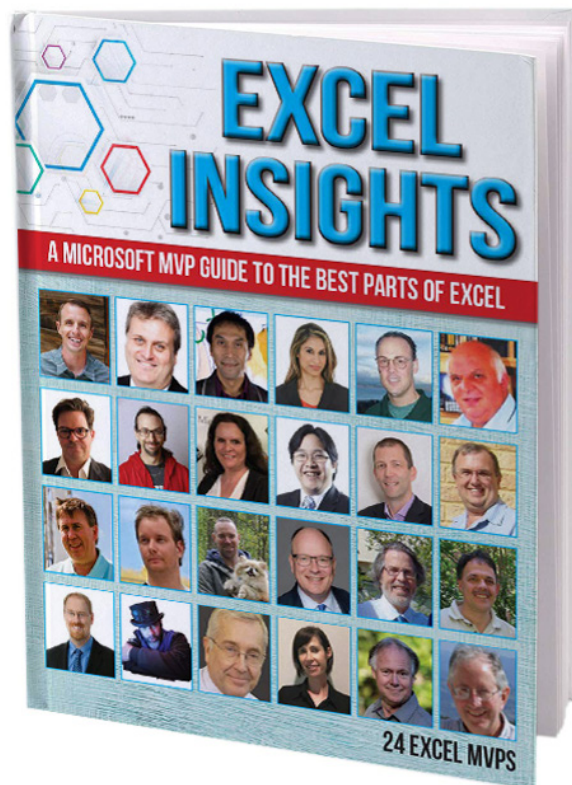
Aside from another book announcement, a tip on comparing lists and a brief briefing regarding me presenting in Taipei, we have all of the regulars too: VBA Basics, Power Pivot Principles, Power Query Pointers, the Power BI updates, and of course the latest instalment of the *A-Z of Excel Functions*.

Until next month.

Liam Bastick, Managing Director, SumProduct



Another Book!



Some more exciting news for the New Year! There's another book coming out. Written by 24 MVPs (there's only 66 Excel MVPs), including SumProduct's very own **Tim Heng** and **Liam Bastick**, this book covers a myriad of the MVPs' favourite topics, including dynamic arrays, **XLOOKUP**, Power Query, charts, dashboards, VBA, Uncle Tom Cobley, *et al*.

Microsoft MVPs are technology experts who have very deep knowledge of Microsoft products. Each year, Microsoft recognises experts who have shown a passion for sharing that knowledge with the community. This book is written by 24 such MVPs (one actually is a PowerPoint MVP, but we won't make note of the error on the cover!): Jon Acampora, Liam Bastick, Leila Gharani, Mike Girvin, a rather orange Roger Govier,

Frederic le Guen, Mathieu Guindon, Ingeborg Hawighorst, Tim Heng, Wyn Hopkins, Ian Huitson, Bill Jelen, Tony de Jonker, Gasper Kamensek, John MacDougall, Dave Paradi, Jon Peltier, Jan Karel Pieterse, Ken Puls, Oz du Soleil, Herve Thiriez, Mynda Treacy, Henk Vlootman and Charles Williams. It's fun seeing if you can spot which photos on the cover were taken within the past 10 years...

Published by *Holy Macro!* Books on April Fool's Day [April 1] (so appropriate, lol), you can pre-order now on Amazon.



Workbook Statistics

Happy New Year – and happy new Excel! To celebrate the start of the third decade, Microsoft has gone “in with the new” in Excel – well the Office 365 Insider Fast variant anyway.

Check out the new **Workbook Statistics** feature (located on the Insert tab in the Proofing group, **ALT + R + B**), which provides information for both the selected worksheet (if you have more than one selected, it’s the one you can see) and the associated workbook. It’s like a mini *Spreadsheet Inquire*.

Current data includes:

- where the end of the sheet is located
- how many cells contain data
- number of Excel Tables (**CTRL + T**)
- number of PivotTables
- number of formulae
- number of charts
- number of objects
- how many worksheets there are
- number of macros.

Yes, there are some things we’d like to see (number of unique formulae, number of hidden sheets, distinction between charts and PivotCharts) and some we’d rather not (apparently there is a macro in a brand new .xlsx workbook?), but that’s what Insider Fast is all about. Suck it and see: if it sucks, you’ll see, and they’ll fix it! This is all part of the great new dawn for Office. Take a look for yourself.

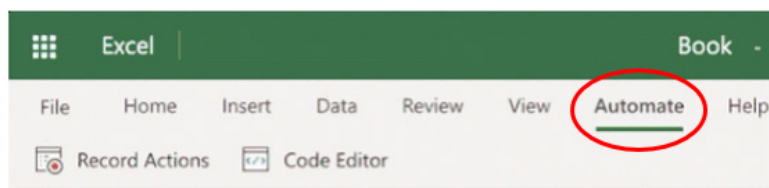
Office Scripts Preview Rolling Out Now

A public Preview of Office Scripts in Excel on the web has recently been announced by Microsoft. Office Scripts is a feature that enables you to record scripts and automate tasks. It is rolling out gradually, but it's well worth a look as it will allow you to create new automations using the script recorder and editor functionality on the web.

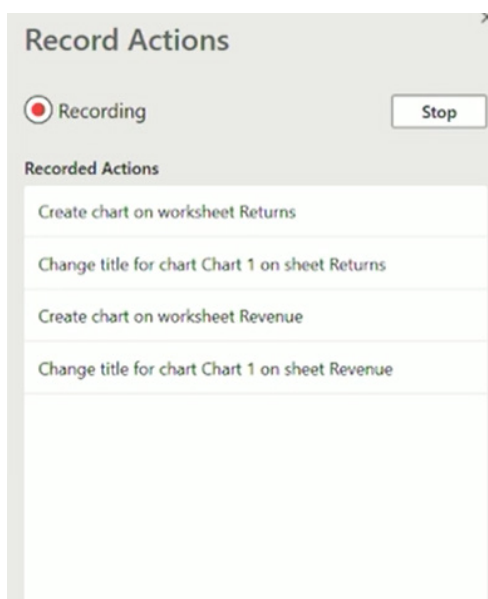
The problem with VBA is that the internet and VBA do not play nicely together. This is where Office Scripts comes in: with this new feature, end-users may record all actions performed such as renaming the worksheet,

inputting the data, and changing the format etc. inside an online Excel workbook, and save the steps as a script. The script generated may then be used with Power Automate (formerly known as Microsoft Flow) or be integrated with another workflow. Furthermore, once the script is created, users may share it with others and a central management system, which can facilitate the sharing of all created scripts.

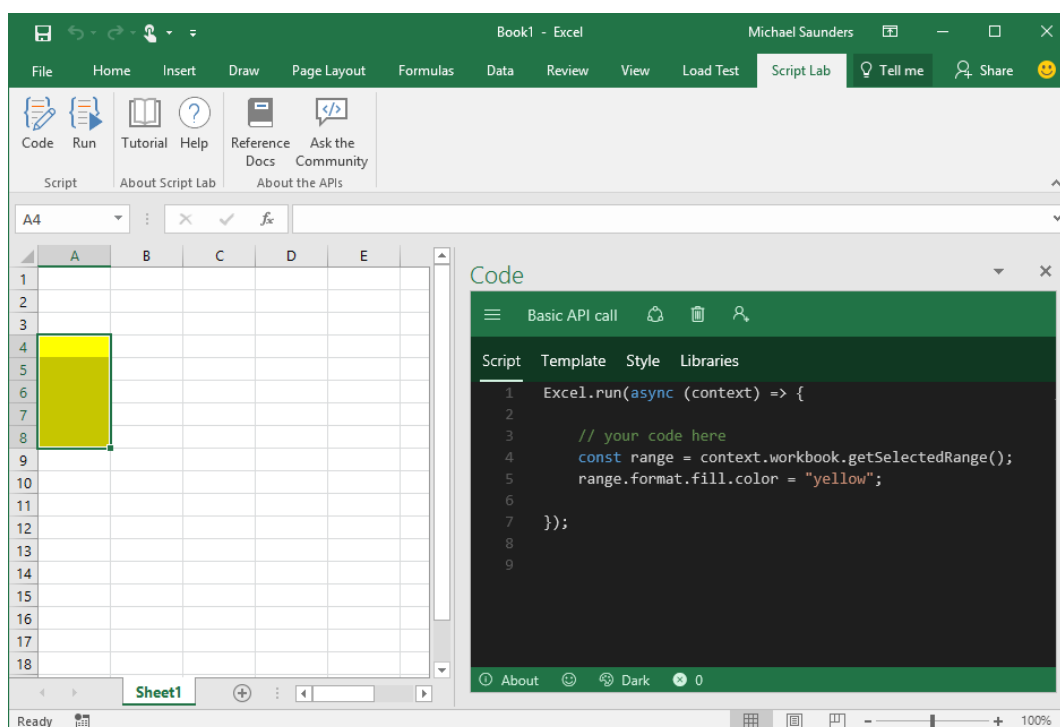
The new feature is incorporated in the Automate tab on the Ribbon for Excel on the web:



From here, it is simple to Record and Edit:



It's easy to confuse Office Scripts with the similarly-named Script Lab, especially as the latter is now available for Excel on the web. Two and a half years ago, Microsoft released Script Lab, a little-known add-in for Excel (and other applications).



Script Lab was created as a tool for anyone who wanted to learn about writing add-ins for Excel, Word or PowerPoint. Microsoft were – and still are – keen to push the JavaScript application program interface (API), as they see it as the technology needed for building Office add-ins that run across platforms.

Script Lab has three main features:

1. Code in a pane beside your spreadsheet
2. Run the code in another pane beside the editor
3. Share your snippets elsewhere.

The problem was, it was really aimed at expert users and, quite frankly, didn't really take off. Office Scripts, on the other hand, is aimed at the "point and click brigade", of which this author is a fully signed up member!

Office Scripts will be rolling out – gradually, alas – over the next few weeks to Excel users who have Office 365 E3 and E5 licenses. This allows Microsoft to gather feedback and react as necessary. During this initial Preview, Office Scripts will only be available in Excel on the web, although it will extend to other Excel media in due course.

In order to obtain access, Office Scripts will need to be enabled through the admin center (sic) and will become visible in tenants as the feature rolls out. The feature may be enabled by following the steps below:

- Go to [admin center](#)
- Select Settings -> Settings (or Settings -> Services & add-ins in the new admin center, depending upon your interface) and then choose Office Scripts from the list.

Check the 'Let users automate their tasks in Office on the web' box in the Office Scripts panel and save changes.

Office Scripts

Office Scripts platform allows users to automate their tasks. This feature is available in preview for Excel on the web. By enabling this feature, users can try Office Scripts and provide feedback while it's in preview.

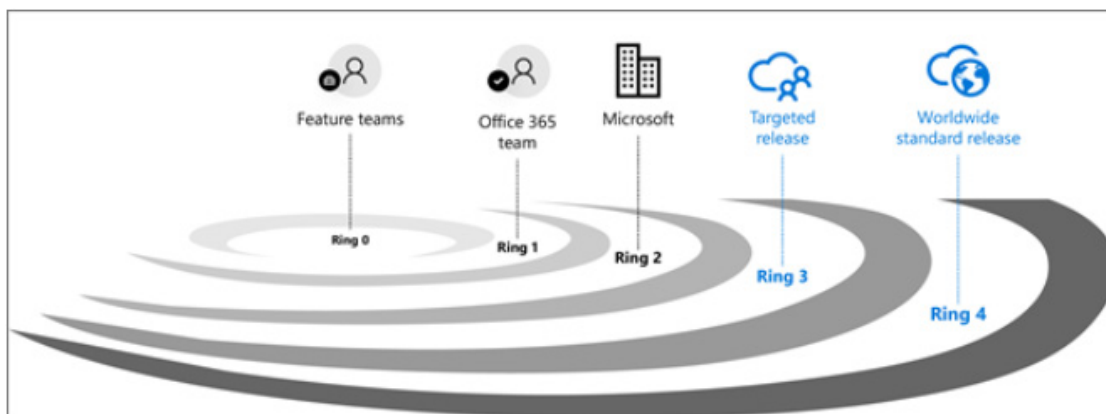
[Learn more about Office Scripts](#)

Let users automate their tasks in Office on the web

Note: Office Scripts is not currently available in Internet Explorer (ironically).

Office Scripts is currently in what is known as the **Targeted** release (previously known as **First Release**). If you have enabled Office Scripts, but do not yet see the Automate tab, you may need to wait until the feature is released to your organisation.

Any new release is first tested and validated by the feature team, then by the entire Office 365 feature team, followed by all of Microsoft. After internal testing and validation, the next step is a Targeted release to customers who opt in. At each release ring, Microsoft collects feedback and further validates quality by monitoring key usage metrics. This series of progressive validation is in place to make sure the worldwide-release is as robust as possible. The releases may be visualised as follows:



If you are an administrator and wish to swap to Targeted release:

1. In the [admin center](#), go to the **Settings -> Organization profile page**
2. Next to **Release preferences**, select **Edit**
3. To disable targeted release, select **Standard release**, then select **Next**, and say **Yes** to the confirmation. Skip to the step 7
4. To enable targeted release for all users in your organization, select **Targeted release for everyone**, then select **Next**, and say **Yes** to the confirmation. Skip to step 7
5. To enable targeted release for some people in your organization, select **Targeted release for selected users**, then select **Next**, and say **Yes** to the confirmation
6. Select **Add people** to add users individually. Search for their names and select **+** to **Add**.
7. When you're done adding users, select **Save** and then **Close**.

Dynamic Arrays, XLOOKUP and XMATCH

F73 =FILTER(F48:I63,(G48:G63=G69)*(H48:H63=G70),{"", "None", "N/A", "N/A"})

Original Data

Item	Shape	Colour	Sides
1	Triangle	Red	3
2	Rectangle	Amber	4
3	Circle	Green	1
4	Triangle	Red	3
5	Square	Blue	4
6	Rectangle	Blue	4
7	Rectangle	Amber	4
8	Circle	Amber	1
9	Triangle	Red	3
10	Square	Green	4
11	Circle	Blue	1
12	Square	Amber	4
13	Triangle	Blue	3
14	Circle	Green	1
15	Rectangle	Blue	4

Filter Results

Shape
 Colour

Item	Shape	Colour	Sides
1	Triangle	Red	3
4	Triangle	Red	3
9	Triangle	Red	3

These long-awaited Office 365 features continue to roll out. It is now believed anyone with Office 365 Monthly Channel should be seeing dynamic arrays, and members of our team have reported seeing **XLOOKUP** and **XMATCH** pop up in Excel too.

Example

Data

Student	ID
Graham	1333
Alice	1331
Mitch	1392
Wendy	1867
Xiu	3589
Brian	2364
Toby	3717

Solution

Student

XLOOKUP =XLOOKUP(H52,F41:F47,G41:G47)

VLOOKUP (3 args) =VLOOKUP(H52,F41:G47,COLUMNS(F39:G39))

VLOOKUP (4 args) =VLOOKUP(H52,F41:G47,COLUMNS(F39:G39),FALSE)

If you have Office 365 and you haven't got them yet, be patient; they are definitely rolling out.

A Useful Tip

I am not a great fan of PivotTables, truth be known. Their results are sometimes a little too "black box" for my liking and many users forget to refresh them when data has been updated. Power Pivot supersedes much of their functionality, but they are useful upon occasion.

And this is one such occasion.

Let's imagine you have a workbook with multiple lists contained in different sections or even worksheets. For my example, I am going to assume we are comparing lists of colours, but these data points could just as simply be account codes, business units, employee names, etc.

Colour
Red
Orange
Indigo
Green
Pink
White
Black
Orange
Purple
Turquoise

Colour
Yellow
Indigo
Black
Pink
Cyan
White
Blue
Brown
Red
Purple
Mauve
Magenta

Colour
Red
Orange
Black
Yellow
Indigo
Blue
Green

Colour
Violet
White
Pink
Magenta
Red
Green
Orange
Blue
Turquoise
Black

Colour
Red
Orange
Yellow
Green
Blue
Indigo
Violet
Black
White
Pink
Magenta
Cyan
Turquoise
Mauve

Which colours are in all five lists? Are there any duplicates within a list? Do some colour descriptions have additional blanks after the end of the visible characters which may cause problems for **LOOKUP** or **SUMIF** operations?

How would you go about analysing these lists? Many analysts would use conditional formatting to highlight duplicates or use functions such as **VLOOKUP**, **COUNTIF** and **SUMIF** to identify and reconcile list elements. Others would resort to VBA and create a report that lists the number of occurrences of each item in list and in combination.

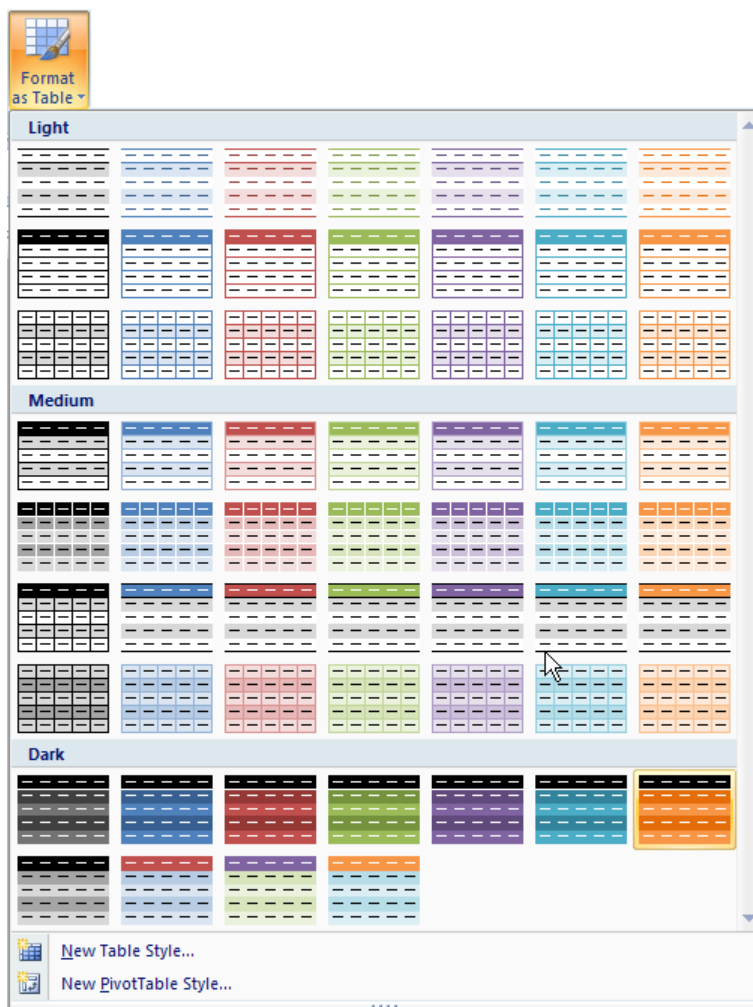
There's a simpler way: PivotTables.

However, before going into full solution mode, I am going to make these lists more versatile by converting them into Excel Tables.

With the release of Excel 2007, the Table functionality was introduced, superseding Excel 2003's 'List' feature, in effect adding functionality. This has nothing at all to do with Data Tables.

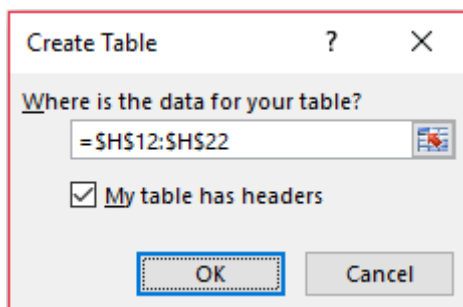
Creating a Table in Excel 2007 and later versions is straightforward, and simply requires the user to select the data to be used in the Table. You do not even need to select the whole range: Excel will prompt for the whole Table even if only one cell is selected (PivotTable creation works similarly).

Next, from the Ribbon, in the 'Styles' group of the 'Home' tab, click on the button that says 'Format as Table':



After clicking this button, Excel shows a new user interface element called a gallery, with a number of formatting choices for your Table, as displayed above. New styles may be created by selecting 'New Table Style...' if required.

After one of the formats has been chosen, Excel will prompt the user regarding which cells are to be converted to a Table:

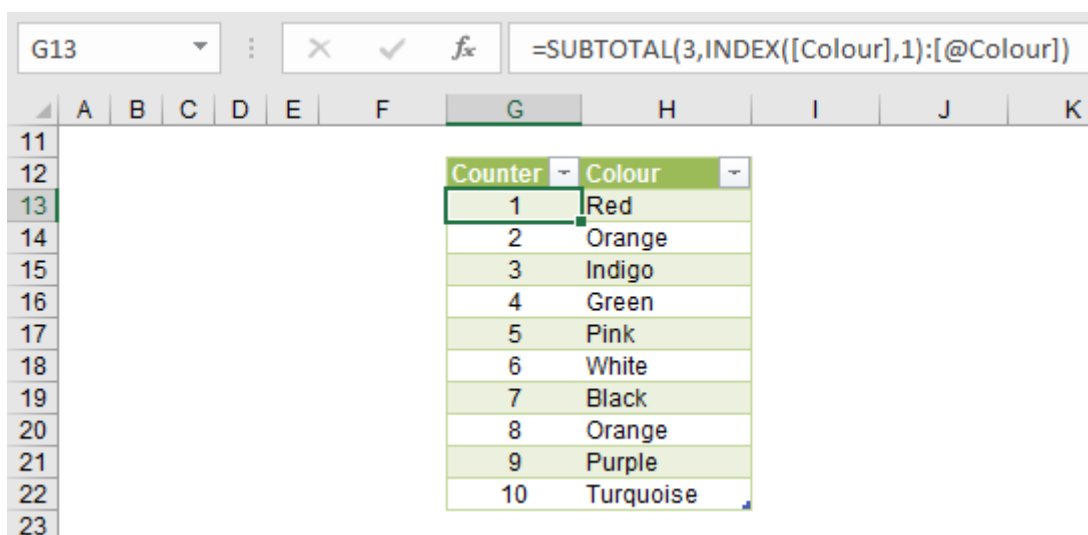


If the Table contains a heading row, ensure that the 'My table has headers' checkbox is checked. Click 'OK' to convert the range to a Table. The keyboard shortcut **CTRL + T** will have the same effect without allowing you to choose the formatting.

Tables have various useful functionalities, one such being the filtering which was done in lists in Excel 2003 and earlier versions. For example, provided the Table has a header row, it will always have in-built filter and sorting, which can be readily accessed from this top row.

A Table will automatically resize to accommodate additional rows and / or columns, provided that data is entered in a cell immediately after the last column or row. It is this feature that is particularly useful when dealing with our lists problem.

Returning to my example, I have turned my lists into Tables. For example, the first list,



This Table has been given the unimaginative name **First_Table** (it's always a good idea to name your Tables) and had a counter added for reasons that will become apparent shortly.

As an aside, for those that don't work with Tables often, or do but encounter the following issue, do note how I have created the counter here. I have used the formula

=SUBTOTAL(3,INDEX([Colour],1):[@Colour])

rather than **=SUM(\$G\$13:\$G13)** (say). The **SUBTOTAL(3,...)** function works like **COUNTA**, i.e. it counts non-blank cells within the range – but with the added benefit of counting only the cells visible after filtering. The second argument requires a little more explaining though.

The problem with **=SUM(\$G\$13:\$G13)** is that if the table is expanded with rows added at the bottom of the Table, the formula does not update correctly in the bottom rows. Whilst it works correctly in the final row, rows directly above it have the same formula which leads to incorrect running totals, counters, etc.

The reference **INDEX([Colour],1):[@Colour]** circumvents this issue. **[Colour]** is the entire field 'Colour' and **INDEX([Colour],1)** is therefore the first item in the field which is held constant (similar to **\$G\$13**). **[@Colour]** gives the contents of the field 'Colour' on the row that the formula is in, so essentially **INDEX([Colour],1):[@Colour]** is **\$G\$13:\$G13** – it's just that it works for some reason!

The second list is also turned into a Table with a counter:

G13 =MAX(First_Table[Counter])+SUBTOTAL(3,INDEX([Colour],1):[@Colour])

Counter	Colour
11	Yellow
12	Indigo
13	Black
14	Pink
15	Cyan
16	White
17	Blue
18	Brown
19	Red
20	Purple
21	Mauve
22	Magenta

Here, note that the formula is slightly different:

=MAX(First_Table[Counter])+SUBTOTAL(3,INDEX([Colour],1):[@Colour])

This formula adds **MAX(First_Table[Counter])** to the counter so that the counter continues from where the **First_Table** counter ends. This logic is then applied to similar formulae in the Tables created for the next three lists too.

Next, a **Table_Summary** is created. This summarises each Table's name and the first and last counter in each using **MIN** and **MAX** functions respectively:

G13 =MIN(First_Table[Counter])

Table	Min	Max	Table Name
1	1	10	Table 1
2	11	22	Table 2
3	23	29	Table 3
4	30	39	Table 4
5	40	53	Table 5

A dynamic, combined list can now be constructed using **LOOKUP** formulae. This may seem convoluted (for those who know and have access to Power Query / Get & Transform [Excel 2016] with Power Pivot,

a similar result can be achieved simply with appending Tables), but it will make your life simpler:

=IF([@Counter]="", "",
 LOOKUP([@Counter],CHOOSE([@[Table Number]],First_Table[Counter],Second_Table[Counter],Third_Table[Counter],Fourth_Table[Counter],Fifth_Table[Counter]),
 CHOOSE([@[Table Number]],First_Table[Colour],Second_Table[Colour],Third_Table[Colour],Fourth_Table[Colour],Fifth_Table[Colour]))

Counter	Table Number	Colour	Table Name
1	1	Red	Table 1
2	1	Orange	Table 1
3	1	Indigo	Table 1
4	1	Green	Table 1
5	1	Pink	Table 1
6	1	White	Table 1
7	1	Black	Table 1
8	1	Orange	Table 1
9	1	Purple	Table 1
10	1	Turquoise	Table 1
11	2	Yellow	Table 2
12	2	Indigo	Table 2
13	2	Black	Table 2
14	2	Pink	Table 2
15	2	Cyan	Table 2
16	2	White	Table 2
17	2	Blue	Table 2
18	2	Brown	Table 2
19	2	Red	Table 2
20	2	Purple	Table 2
21	2	Mauve	Table 2
22	2	Magenta	Table 2
23	3	Red	Table 3
24	3	Orange	Table 3
25	3	Black	Table 3
26	3	Yellow	Table 3
27	3	Indigo	Table 3
28	3	Blue	Table 3
29	3	Green	Table 3
30	4	Violet	Table 4
31	4	White	Table 4

Upon first glance, the formula in the 'Colour' column looks horrible:

```
=IF([@Counter]="", "",
LOOKUP([@Counter], CHOOSE([@Table Number], First_Table[Counter], Second_Table[Counter], Third_Table[Counter],
Fourth_Table[Counter], Fifth_Table[Counter]),
CHOOSE([@Table Number], First_Table[Colour], Second_Table[Colour], Third_Table[Colour],
Fourth_Table[Colour], Fifth_Table[Colour])))
```

Once you get your head around it, it's not *quite* so bad. The **LOOKUP** seeks out the counter in the correct Table's 'Counter' column and returns the corresponding colour from that Table. The **IF** statement is applied simply to return a blank if the consolidated Table has more rows than there are in total in the individual Tables.

Once this has been generated, this can be used as the source for a standard Excel PivotTable (**ALT + N + V**, **ALT + D + P** or else highlight the entire Table and click on the Quick Analysis button in the bottom right-hand corner of the selection in Excel 2013 onwards), viz.

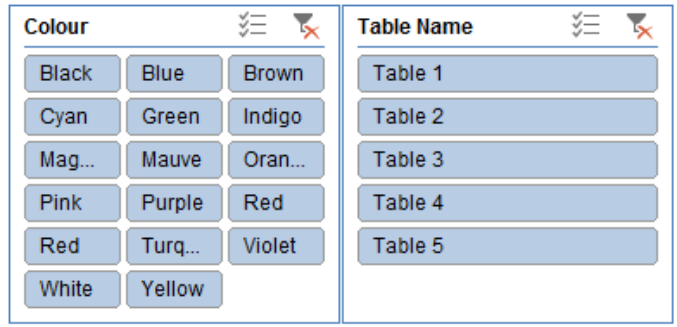


	Table 1	Table 2	Table 3	Table 4	Table 5
Black	1	1	1	1	1
Blue		1	1	1	1
Brown		1			
Cyan		1			1
Green	1		1	1	1
Indigo	1	1	1		1
Magenta		1		1	1
Mauve		1			1
Orange	2		1	1	1
Pink	1	1		1	1
Purple	1	1			
Red	1		1	1	1
Red		1			
Turquoise	1			1	1
Violet				1	1
White	1	1		1	1
Yellow		1	1		1

Now look how user-friendly this is! It has slicers added to drill down on comparisons between the lists or even for each colour – and you can add items to all five lists or even add more lists (as long as you remember to refresh the PivotTable first). Did you catch one 'Red' has an extra space or that there were two occurrences of 'Orange' in the first list? Me neither.

I have taken the liberty of adding slicers to make the experience more user-friendly. The numbers in the PivotTable could be formatted differently (e.g. conditional formatting) to make the summary even easier to follow. How hard would this have been to do with formulae..?

No macros, no complex output formulae. Simple to use. Result.

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 consider using message dialog boxes in Excel.

It's important to ensure that you can give feedback to users when something wrong occurs. Often, it's a nice touch to give feedback when everything works perfectly as well, if only just to let them know that the macro has finished running, and that they can start working in Excel again. To do this, we're going to move outside the realm of recorded

macros, and start writing some simple VBA code from scratch.

The **MsgBox** function in Excel gives us one simple way that we can inform users about what we have done with our macros. The syntax for this is pretty easy:

```
MsgBox <Prompt>, [Buttons], [Title], [HelpFile], [Context]
```

What do these all mean?

- **Prompt:** This is the only mandatory part of the message box. Whatever you type in here - either by putting it in speech marks or by providing a value to output (e.g. Range("A1").Value) – will appear in the body of the message that pops up. So, our first example might be:



- **Buttons:** Have you ever wondered how applications determine what buttons to show you ("Ok", "Cancel", "Export to Excel")? The Buttons parameter gives you the flexibility to decide what buttons to display. You can choose from a range of options including:
 - o OK only
 - o OK and Cancel
 - o Abort, Retry and Ignore
 - o Yes, No and Cancel
 - o Yes and No

You can see other options on Microsoft's help page [here](#). By default, OK will be the only button that shows. These buttons are useful to help capture in VBA whether a user wants to continue or cancel a macro

- **Title:** Relatively self-explanatory, the Title parameter allows you to change the heading of the message box to be something relevant. If you leave it blank, it will display the application name (in our case, Microsoft Excel)
- **HelpFile and Context:** These allow you to reference a custom help file that you can specify, along with what message you would like displayed if someone clicks on the Help button that will appear. To be completely honest, this is uncommon for most purposes, and we're not going to be looking at specific examples of how you can make this work here (does anyone still use help files these days, rather than Google..?).

Therefore, putting all these things together, we could create a message box that checks whether a user definitely wants to continue to run the macro. This might look something along the lines of:

```
Sub Continue()
    MsgBox "Warning - any changes that may occur cannot be undone.", vbOKCancel, "Warning"
End Sub
```

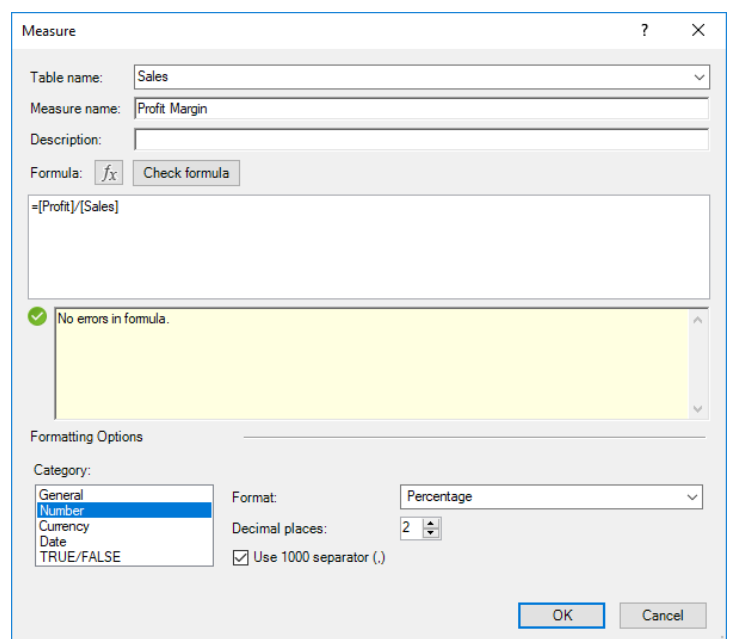
This is a simple way of providing a message to users and getting their feedback.

Next month, we will cover how to use the Status Bar to provide updates while the macro is running, without needing to wait for user input to keep things running behind the scenes. Stay tuned!

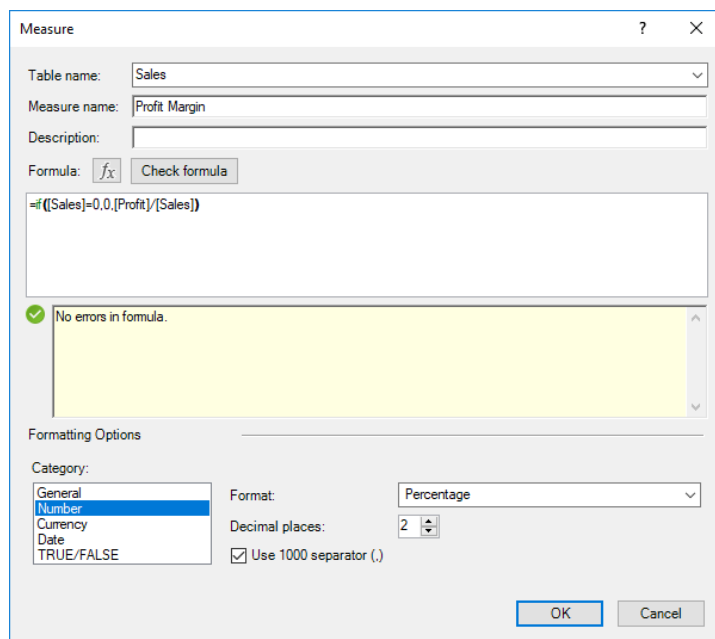
Power Pivot Principles

We continue our series on the Excel COM add-in, Power Pivot. This month, we look at how to include error traps in our measure formulae.

Here, let's continue from last month's article. This time, another common measure that we might want to create is a 'Profit Margin' measure:



Nothing seems to be wrong with this formula, even Power Pivot thinks so. But what if there were no sales in one period or data were missing? We would get a division by zero error. To avoid the potential error, we can use the **IF** function to provide us with an error trap:



Measure

Table name: Sales

Measure name: Profit Margin

Description:

Formula:

✓ No errors in formula.

Formatting Options

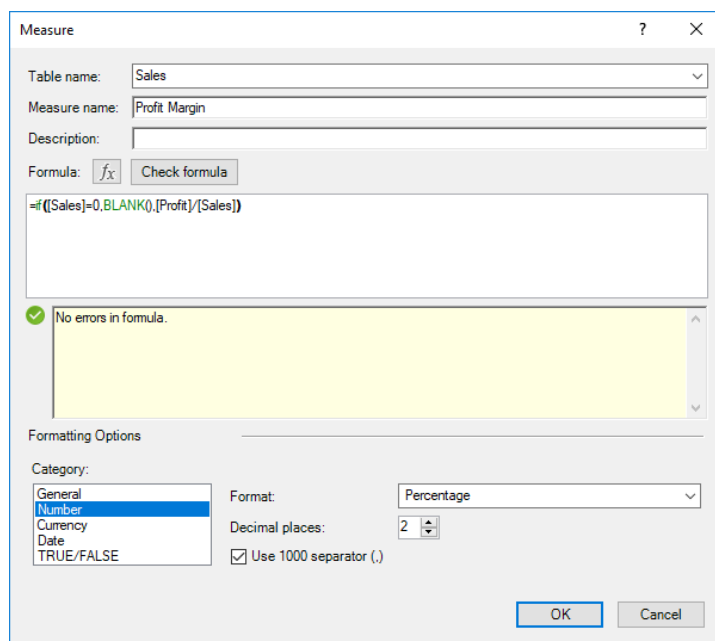
Category: Number

Format: Percentage

Decimal places: 2

Use 1000 separator (.)

This will return a zero value when the sales are nil. But why report it anyway? We can use the **BLANK** function to further streamline this measure viz.



Measure

Table name: Sales

Measure name: Profit Margin

Description:

Formula:

✓ No errors in formula.

Formatting Options

Category: Number

Format: Percentage

Decimal places: 2

Use 1000 separator (.)

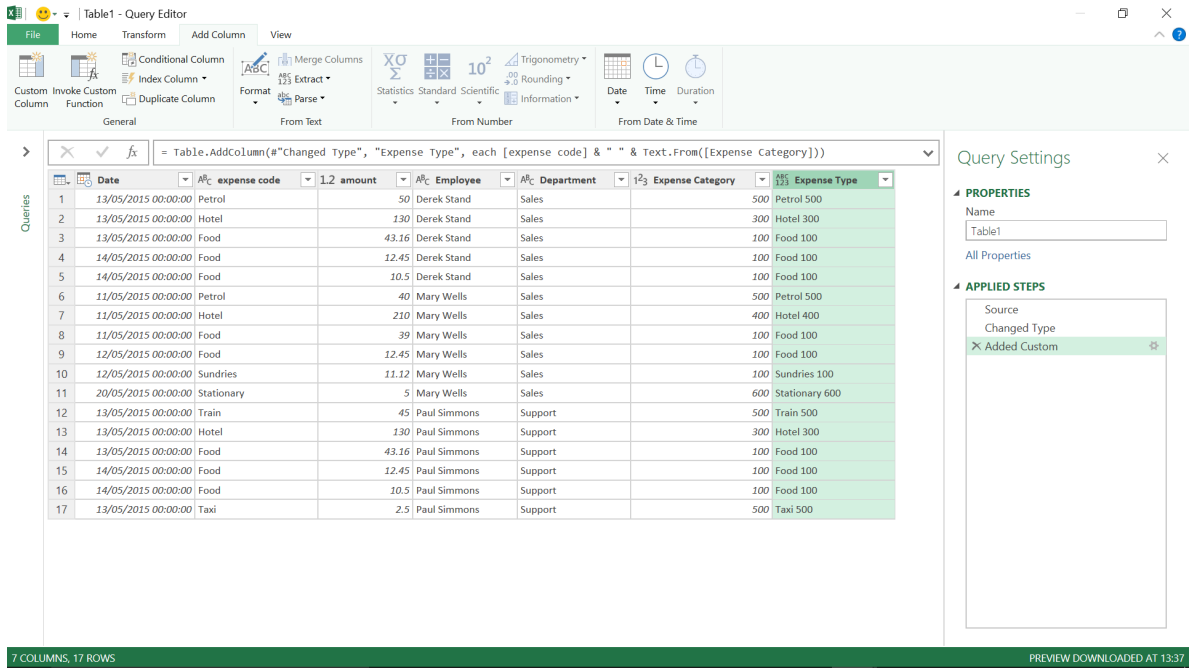
Now the value will be suppressed in the Pivot Table if sales are blank, leading to clearer, more concise reports.

More *Power Pivot Principles* next month.

Power Pivot Principles

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 create a simple function and show how to then access it when building a query.

Previously, we used **M** code to combine two columns, as in the screenshot below:

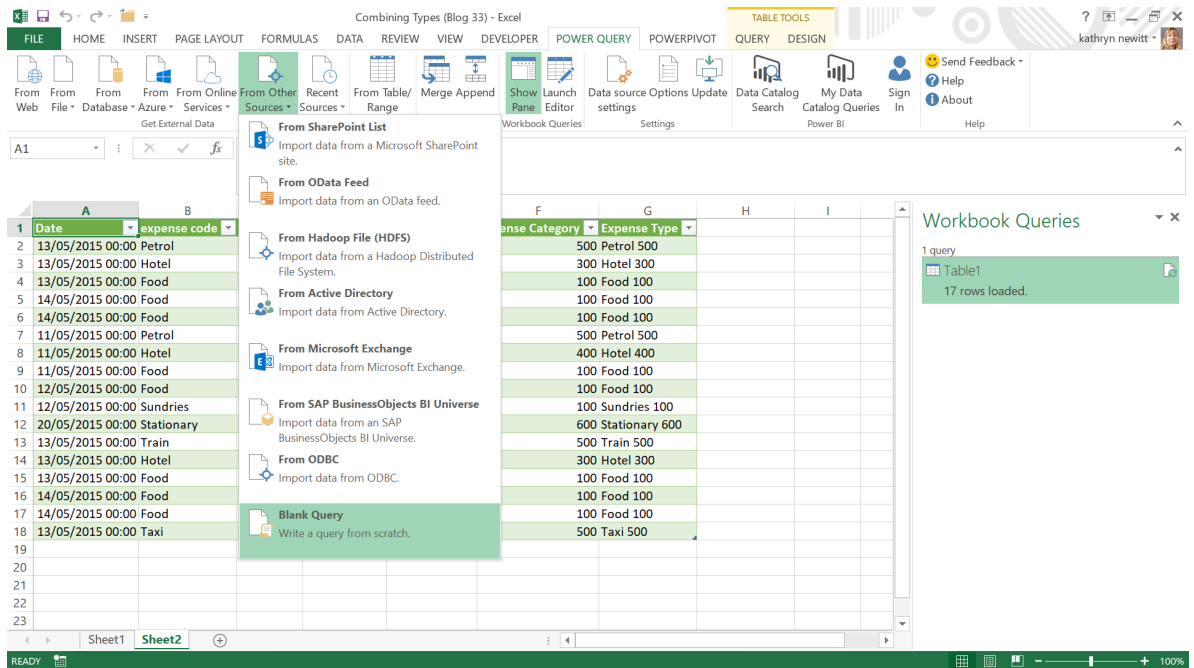


If we often need to combine columns in this way, we can take the M code

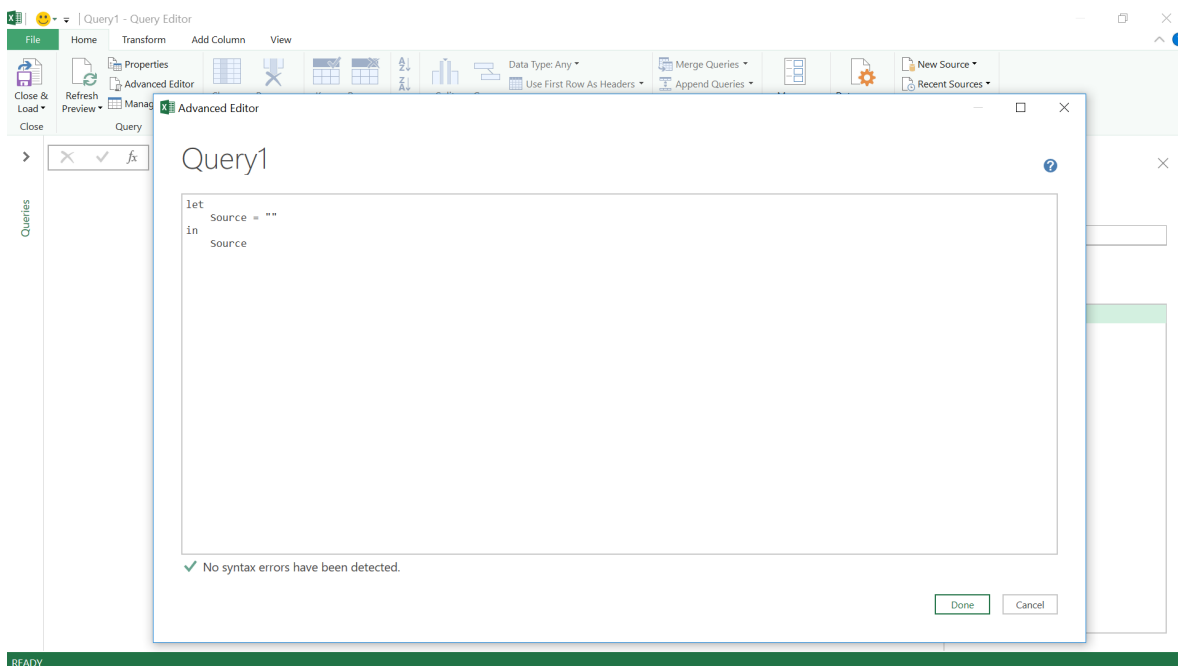
= [expense code] & " " & Text.From([Expense Category])

and put it into a function.

Let's begin by creating a blank query from the Excel worksheet:



Then, we need to access the Advanced Editor:



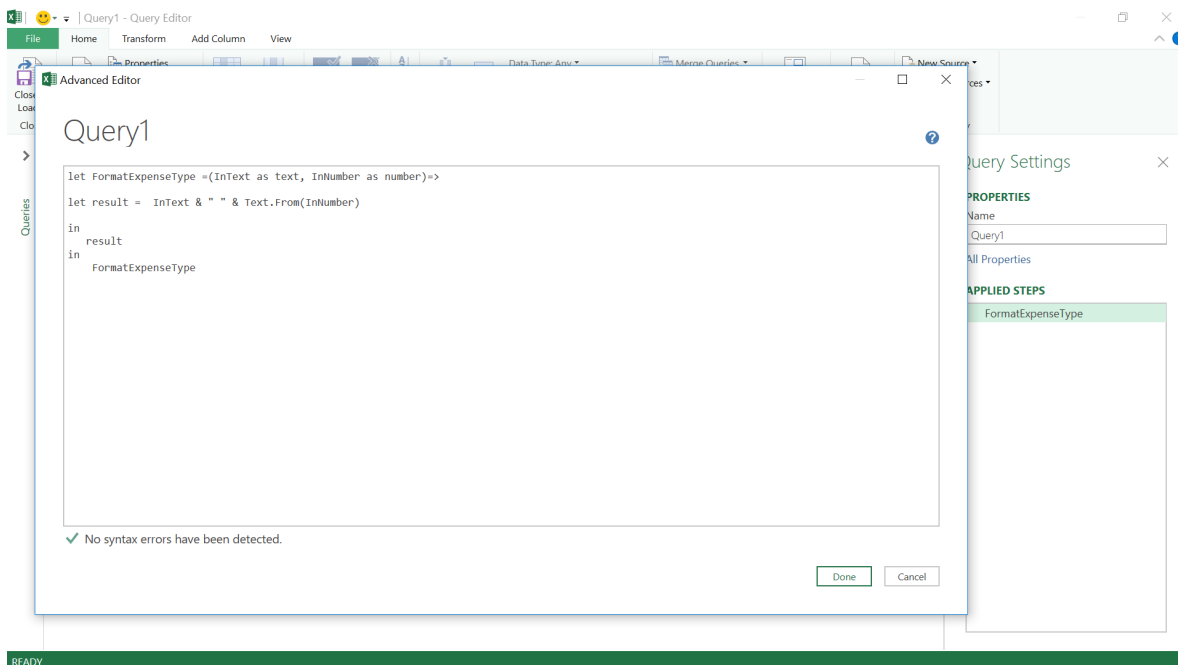
At the beginning of our function code, we need to give it a name – I'll choose '**FormatExpenseType**'. We need to define what parameters our function accepts (I've called them '**InText**' and '**InNumber**'). Next, we'll apply this logic to combine the parameters and finish our function syntax by defining the output ('**result**'). We may specify a value for my function should we wish (I have chosen not to, but I still need the final line):

let FormatExpenseType =(InText as text, InNumber as number)=>

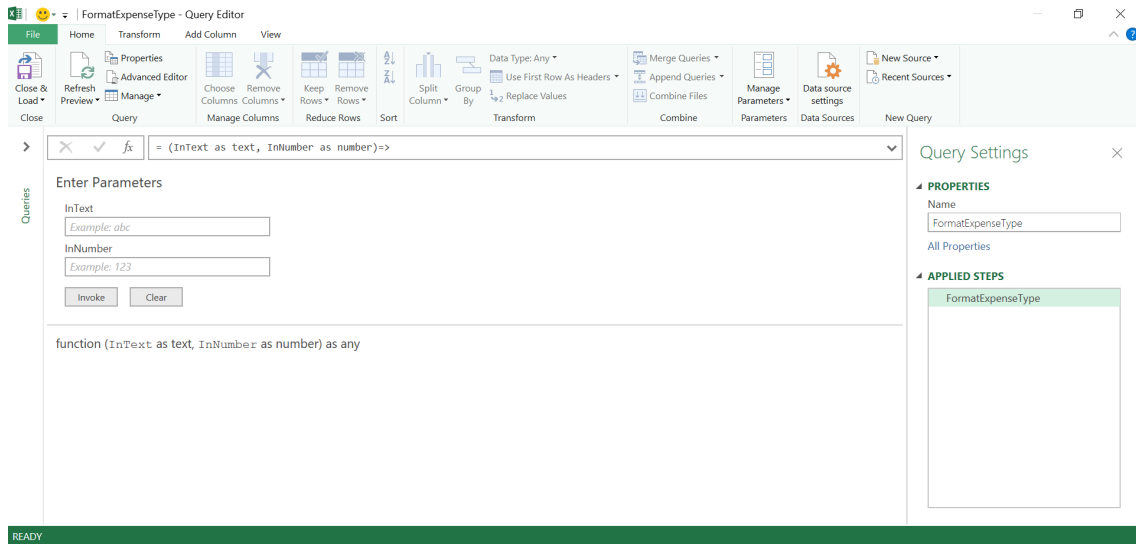
let result = InText & " " & Text.From(InNumber)

in result

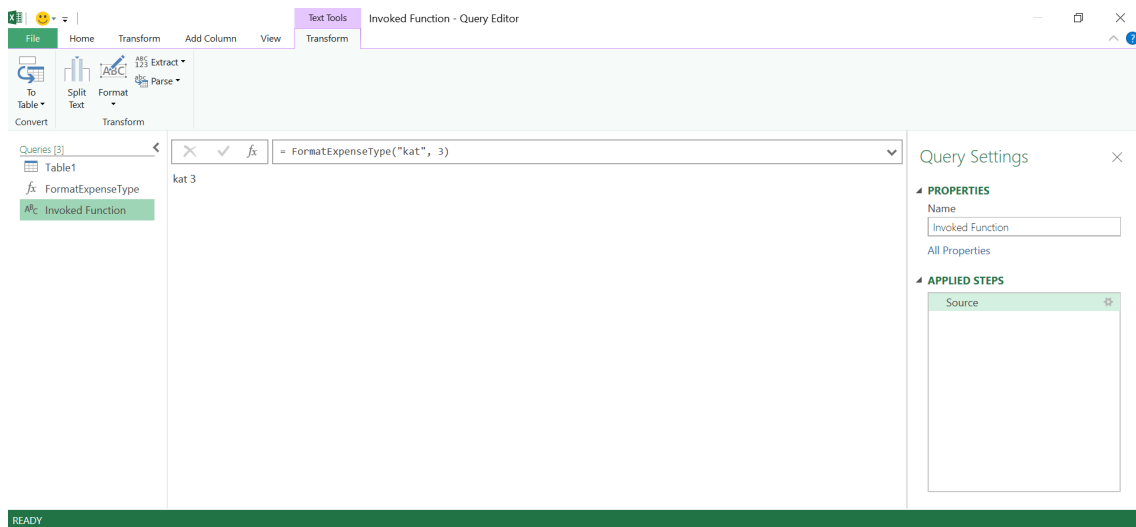
in FormatExpenseType



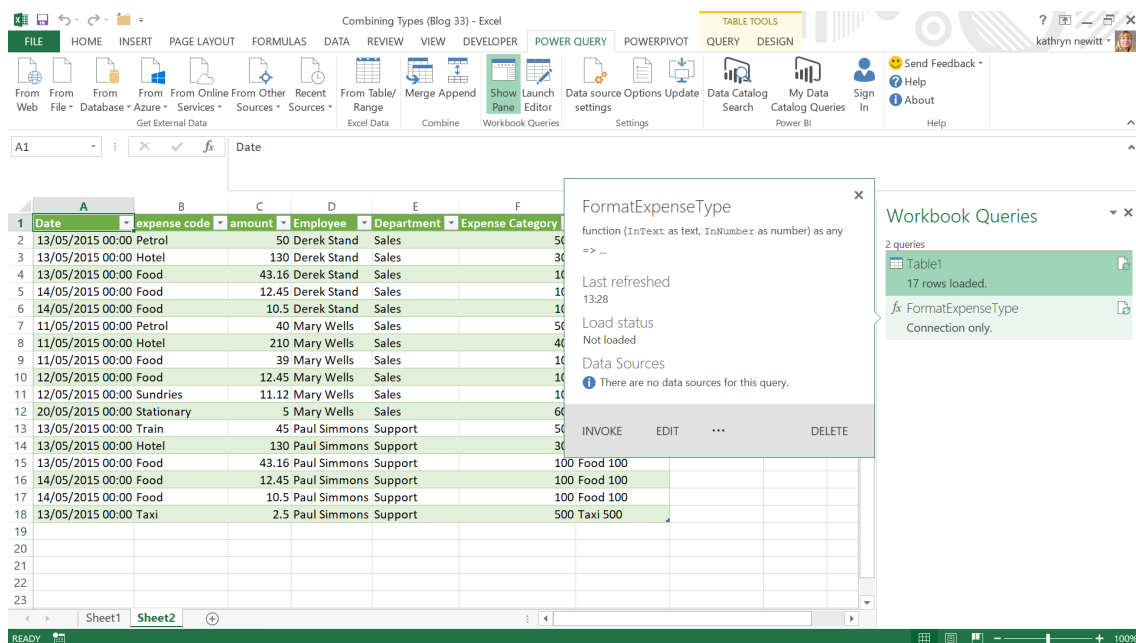
Once we have created our query, we can name it '**FormatExpenseType**', ready to use in other queries. You can test it first by clicking 'Invoke', or choose to 'Close and Load' from the 'Home' or 'File' tabs to store the function in the workbook.



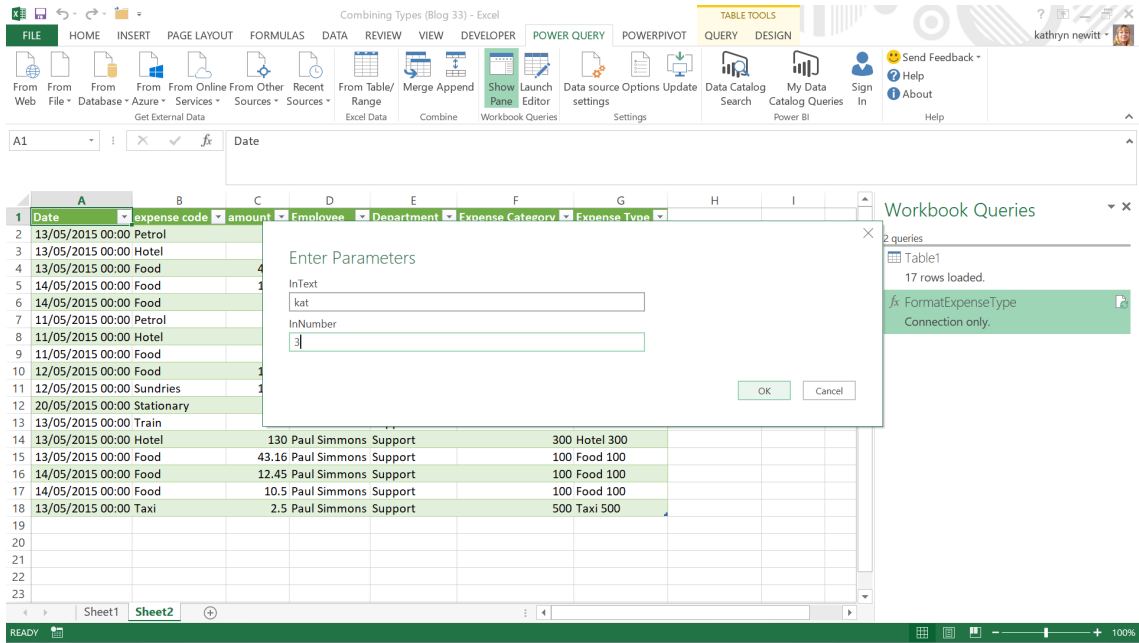
Choosing to invoke creates a new query rather than adding an unnecessary step to our function:



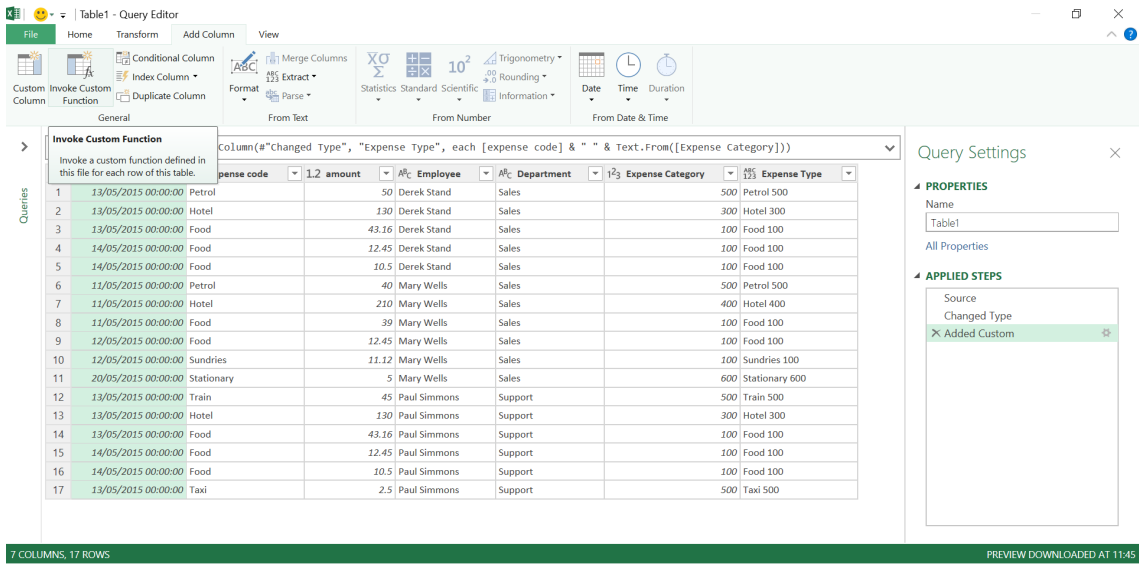
The query is now shown as a function in the workbook, and hovering over it shows us the details behind it.



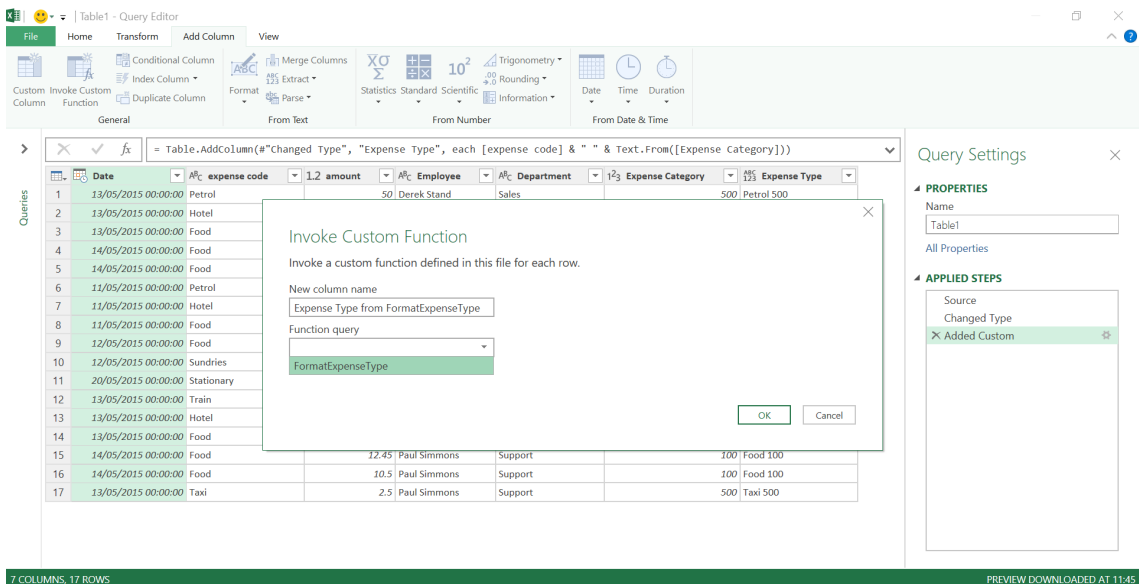
We can also choose to invoke from here, which would create a new query in exactly the same way as invoking from the Query Editor.



We are now ready to call this function from my 'Table1' query. First, double click on 'Table1'. In the 'Add Column' tab in the Query Editor, we may 'Invoke Custom Function':



We are prompted to enter a new column name and select a function; in this case, we can only see the one we have created.



This time, instead of values, let's choose columns as our parameters:

Invoke Custom Function

Invoke a custom function defined in this file for each row.

New column name
Expense Type from FormatExpenseType

Function query
FormatExpenseType

InText
Example abc

Column Name

OK Cancel

7 COLUMNS, 17 ROWS

PREVIEW DOWNLOADED AT 11:45

Select the **expense code** and **Expense Category** columns.

Invoke Custom Function

Invoke a custom function defined in this file for each row.

New column name
Expense Type from FormatExpenseType

Function query
FormatExpenseType

InText
expense code

InNumber
Expense Category

OK Cancel

7 COLUMNS, 17 ROWS

PREVIEW DOWNLOADED AT 11:45

We have deliberately used a query where we already have the original **Expense Type** column for comparison, so now you may test this function:

Invoke Custom Function

Invoke a custom function defined in this file for each row.

New column name
Expense Type from FormatExpenseType

Function query
FormatExpenseType

InText
expense code

InNumber
Expense Category

OK Cancel

8 COLUMNS, 17 ROWS

PREVIEW DOWNLOADED AT 13:45

The function works perfectly – and because it will work on any text and number column combination you could rename it to be more generic (e.g. **FormatTextPlusNumber**) and use it for other columns.

More next month!

Power BI Desktop Update for December

With the newsletter printing deadline always so early in December, it's always difficult to detail the Power BI Desktop updates for that month, as Microsoft typically releases them nearer the Christmas break. 2019 was no exception, so we have the usual February landslide below!

In Preview, there is the new 'Customize theme' dialog that allows you

But that's not all. Here's the complete list of December updates:

Reporting

- Customize current theme (Preview)
- Export current theme
- Setting table column or matrix value as a custom URL
- KPI visual formatting settings
- New decomposition tree formatting
- Filter pane toggle button in the new Ribbon
- Automatic Page Refresh query details (Preview)

Analytics

- Load more for Analyze Insights
- New DAX function: **QUARTER**

Visualizations

- Personalisation of the Visualization pane is now Generally Available
- New xViz visuals:
 - o Hierarchy tree
 - o Bullet chart
 - o Parallel Coordinates
 - o Gantt chart
- New Zebra BI visuals:
 - o Zebra BI charts
 - o Zebra BI tables
- Drill Down Combo Bar PRO by ZoomCharts
- Annotated Bar by Queryon
- Tachometer 5 Ranges
- ValQ licensing updates

Customize current theme (Preview)

This update sees two significant updates for theming. The first is that you may now customise the most common options of your current theme through a dialog in Power BI Desktop instead of needing to write or modify a JSON file (which, of course, we know you all did!).

to tweak common theme settings without having to write or modify a JSON file. Additionally, you may now export the current theme as a JSON so you can easily share, reuse, or make modifications to the JSON in an editor. The recently added decomposition tree receives several new formatting options with this update too.

Data Connectivity

- Azure Data Lake Storage Gen2 connector is now Generally Available
- Power Platform Dataflows connector
- PostgreSQL connector now includes the Npgsql provider
- AtScale connector is now Generally Available
- Azure Time Series Insights connector
- Data Virtuality connector
- Zucchetti HR Infinity connector

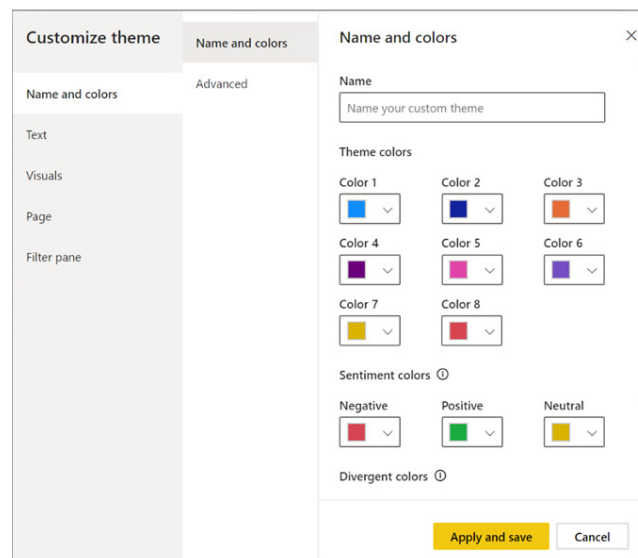
Data Preparation

- Updates to AI Insights functions (Preview)

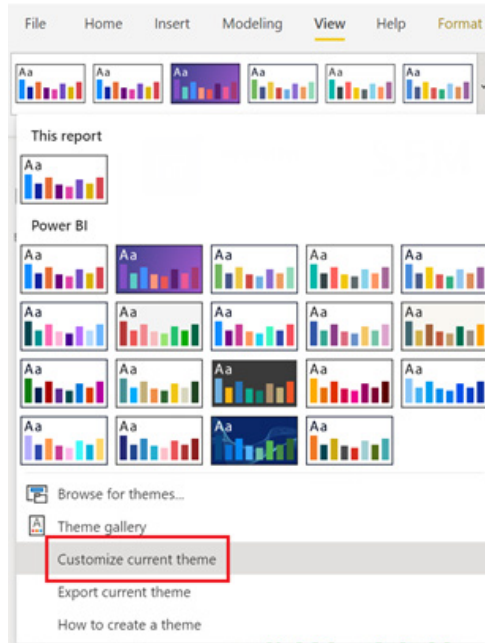
Template Apps

- Omnichannel insights for Dynamics 365
- Customer service analytics for Dynamics 365
- Microsoft Forms Pro for Customer Service.

Let's take a look at each in turn.



After you've enabled the Preview feature in the Options dialog, you can launch this new theming dialog by selecting the 'Customize current theme' option in the 'Theme gallery' dropdown:

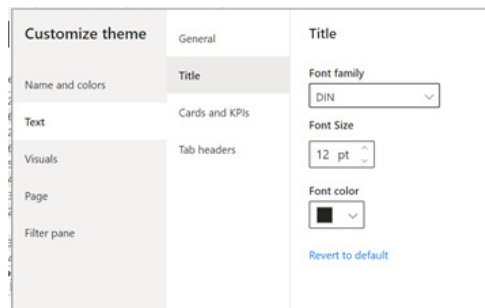


Whether you are using one of the built-in themes or a custom one, this new theme dialog is pre-populated with the settings from your current theme. From here, you may adjust any or all the settings you want and save it back to your report.

The settings that may be customised are divided into different categories, shown in the left tab navigation of the dialog. Under 'Name and colors', you'll find:

- Theme colors
- Sentiment colors, used by the waterfall and KPI charts
- Divergent colors, used by the color scales version of conditional formatting
- Advanced color classes, introduced to the theme JSON structure back in September of last year.

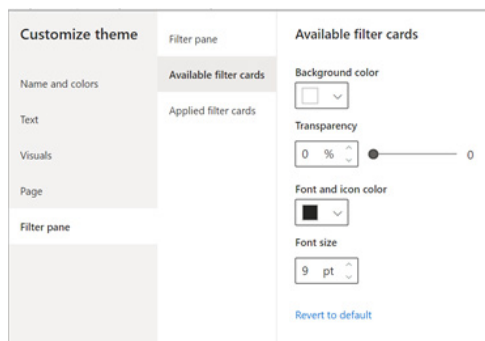
The Text tab lets you customise all the major text classes, which were also introduced back in September.



The Visuals section lets you modify:

- Backgrounds
- Borders
- The visual header formatting (although it does not include on / off toggles for each icon)
- ToolTip formatting.

The Page settings include wallpaper and page background colour and transparency, and lastly, the Filter pane section include all the formatting options available to the Filter pane itself and the applied and available Filter cards.



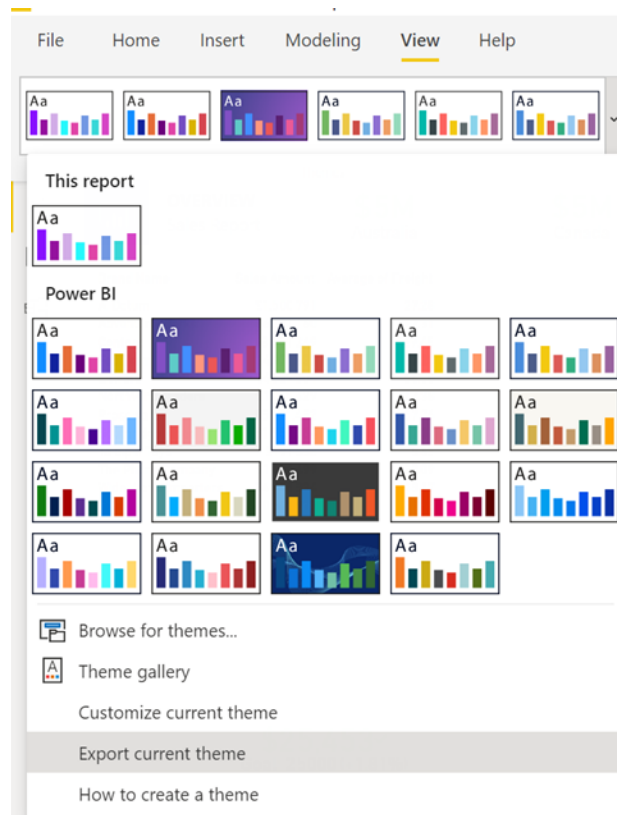
The color picker for this dialog is also new and improved. In the new color picker, you have a slider to change the colour you want to use and then a large section you can drag within to pick the right hue of that colour. You can also set the colour using hexcode or RGB value.



Export current theme

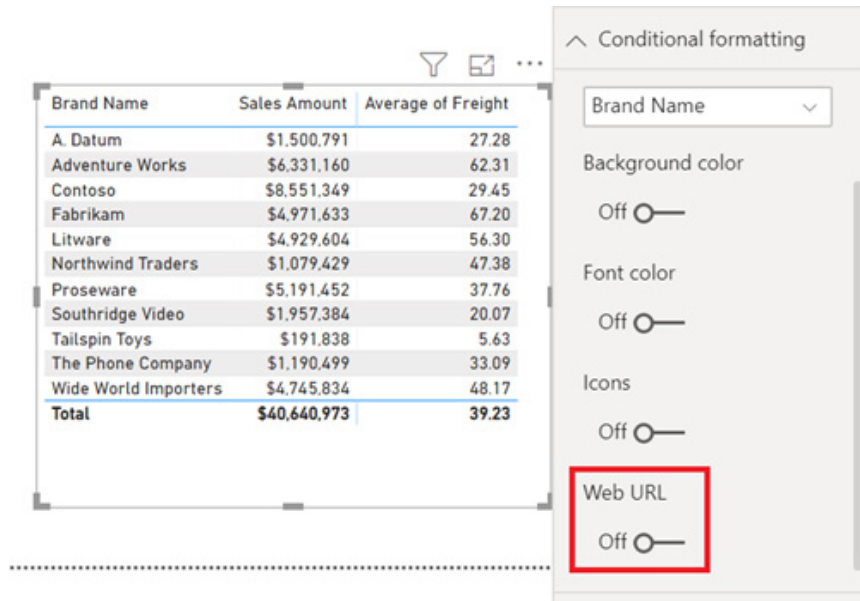
The second theme related update is that you can now download the currently applied theme. If you have a downloaded or shared report with a theme that you really like, this is a great way to extract that theme file to reuse in your own reports. This option will also allow you

to export the JSON for most of our built-in themes. The only exceptions to this are Microsoft's "base" themes, Classic and Default, which other themes build upon when imported. You can find this option under the theming dropdown.

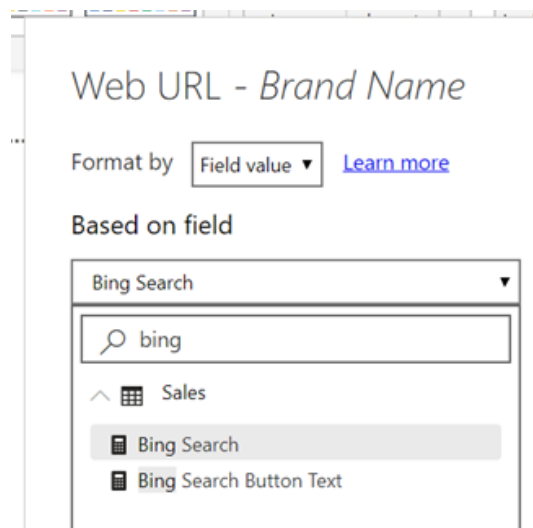


Setting table column or matrix value as a custom URL

Tables and matrices get an update too. Microsoft has introduced a new conditional formatting option that allows you set a table's column or matrix's value as a custom URL. You'll find this new option under the 'Conditional formatting' card in the Formatting pane.



Once you turn it on, you'll be able to pick a field to use as the URL for the column.



Once you've picked the field and confirmed the change, you'll see the field values formatted as links in the table or matrix. If you hover over the link, you'll see a ToolTip showing the URL; clicking on it will open up the URL in a new tab.

Brand Name	Sales Amount	Average of Freight
A. Datum	\$1,500,791	27.28
Adventure Works	\$6,331,160	62.31
Contoso	\$8,551,349	29.45
Fabrikam	\$4,971,633	67.20
Litware	\$4,929,604	56.30
Northwind Traders	\$1,079,429	47.38
https://www.bing.com/search?q=Litware, Inc.	\$191,838	5.63
The Phone Company	\$1,190,499	33.09
Wide World Importers	\$4,745,834	48.17
Total	\$40,640,973	39.23

KPI visual formatting settings

The KPI visual itself has been around for a while, but when compared to other Power BI visuals, the formatting options were a bit limited previously. However, with this update, the new KPI formatting options have been upgraded / improved:

- Indicator text formatting (font family, colour and alignment)
- Trend axis transparency
- Goal and distance text formatting (label text, font family, colour and size)
- Distance text formatting (label text, positive direction, font family, colour and size)
- Adding a date label with formatting (font family, colour and size).



\$25,453 ✓
Goal: 25000 (+453 +2%)
12/31/2015

Some of these new formatting options may also be conditionally formatted:

- Indicator font color
- Goal font color and Goal Distance font color
- The good / bad / neutral status colours
- Date font color.

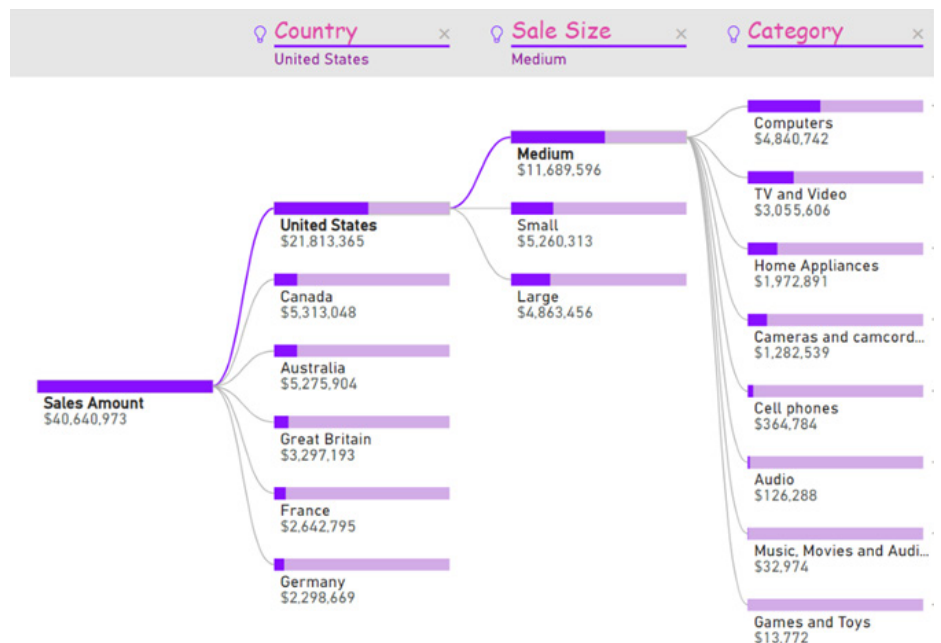
(The spellings of “colour” and “color” are deliberate!! Don’t ask us how many committee meetings we held to make the decisions we did – Ed.)

New decomposition tree formatting

The decomposition tree visual has been updated with many more formatting options, including:

- Category labels font family, size and color
- Data labels font family, size, color, display units and decimal places precision
- Level header title font family, size and color
- Show subtitles toggle
- Subtitles font family
- Level header background color
- Data bars background color and width
- Tree connector color.

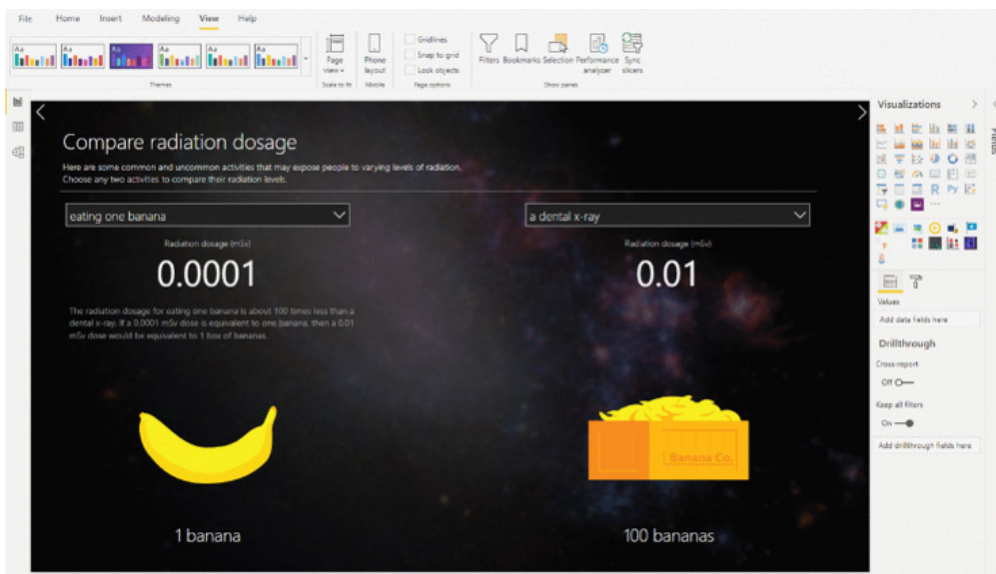
With all these additional formatting options, you can completely customise the tree to look the way you want and make it feel cohesive with your report.



Filter pane toggle button in the new Ribbon

There has been a small improvement to the new Ribbon in Preview. There is now a Filters toggle button in the View tab that allows you to toggle the Filter pane on and off in Power BI Desktop. This feature is useful when you aren't using the Filter pane and need extra real estate. This addition aligns the Filter pane with the on / off experience available

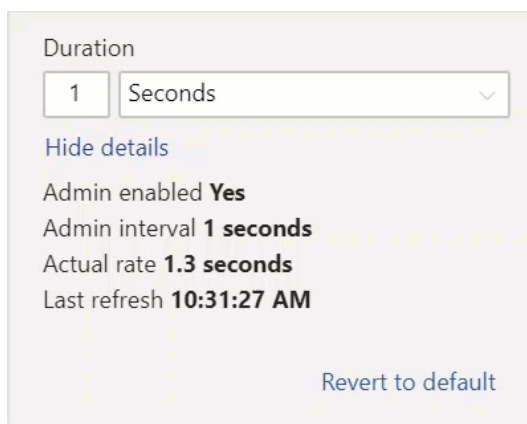
for other panes that can be opened and closed, such as the Bookmarks pane. However, do keep in mind that this setting only impacts the viewing of the Filter pane in Power BI Desktop. If you want to hide the Filter pane for your end-users, you will need to select the eye icon next to Filters.



Automatic Page Refresh query details (Preview)

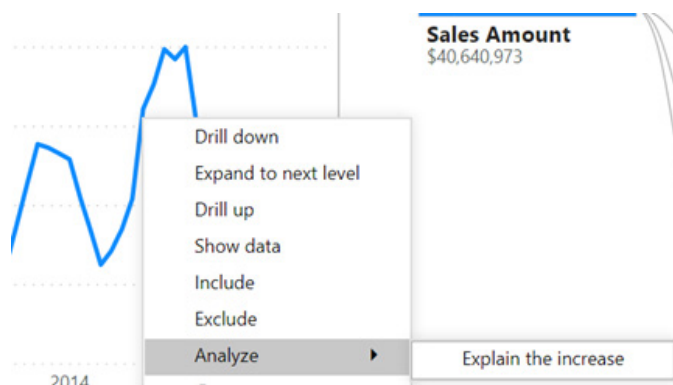
Back in the October update, support was announced for automatic page refresh (APR for short) in Preview. With this release, Microsoft is adding a details section on each page to help you define the best refresh interval and monitor performance. To access this details menu, go to

the Page refresh tab in the Formatting pane and click on 'Show details'. In the Service, it will show you if the feature is enabled in your premium capacity by the administrator, the defined minimum interval, the actual refresh rate and the last refresh.

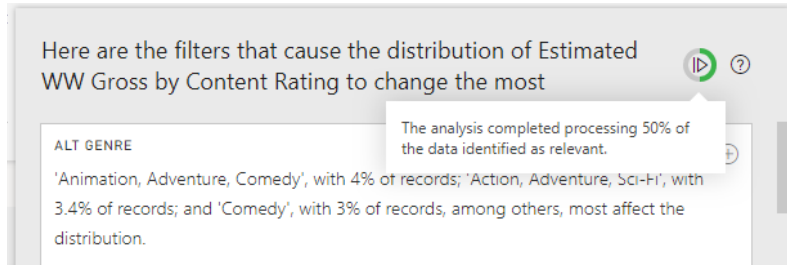


Load more for Analyze Insights

When running the Analyze feature to find insights in your data, such as 'Explain the increase', Power BI Desktop only runs the Machine Learning models for a period of time to show you insights in a timely manner. However, if there's significant data to analyse, you can now choose to continue to run the analysis after the initial timeout.



You'll be able to use the play button in the top right of the flyout to continue running the analysis. After each continuation, the insights shown will update to include the top 10 most relevant insights.



New DAX function: QUARTER

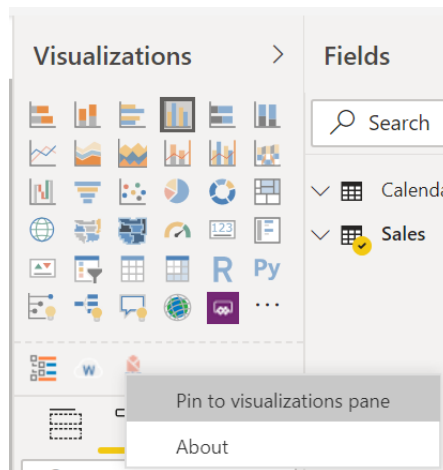
There is a new DAX function: **QUARTER**, which we will show no quarter to. The **QUARTER** function returns the quarter corresponding to a specified date.

Personalisation of the Visualization pane is now Generally Available

The ability to personalise the Visualization pane is now Generally Available. Some highlights of this feature include:

- You can pin or unpin all types of visuals and personalise the Visualization pane
- You can see the personalisation in the Power BI Service as well
- You can unpin default visuals and restore them using the 'Restore default visuals' option.

As a reminder, unpinned visuals in your report will show up below a dotted line and the pinned visuals above it. You can pin or unpin visuals just by opening the context menu and selecting the first option.

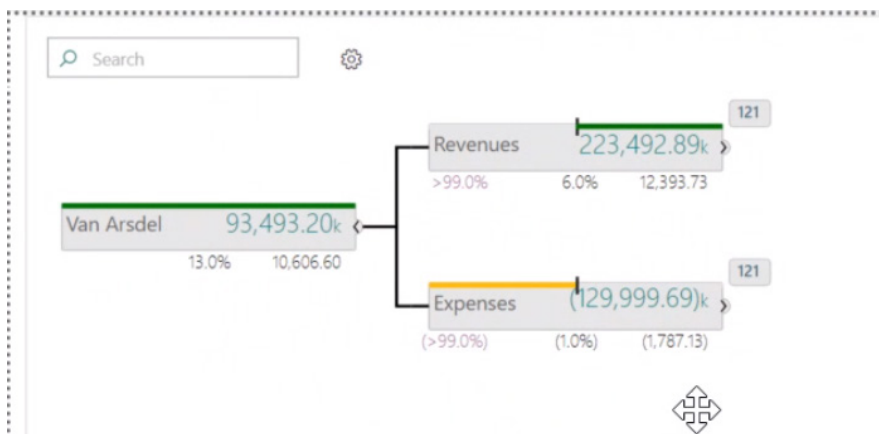


New xViz visuals

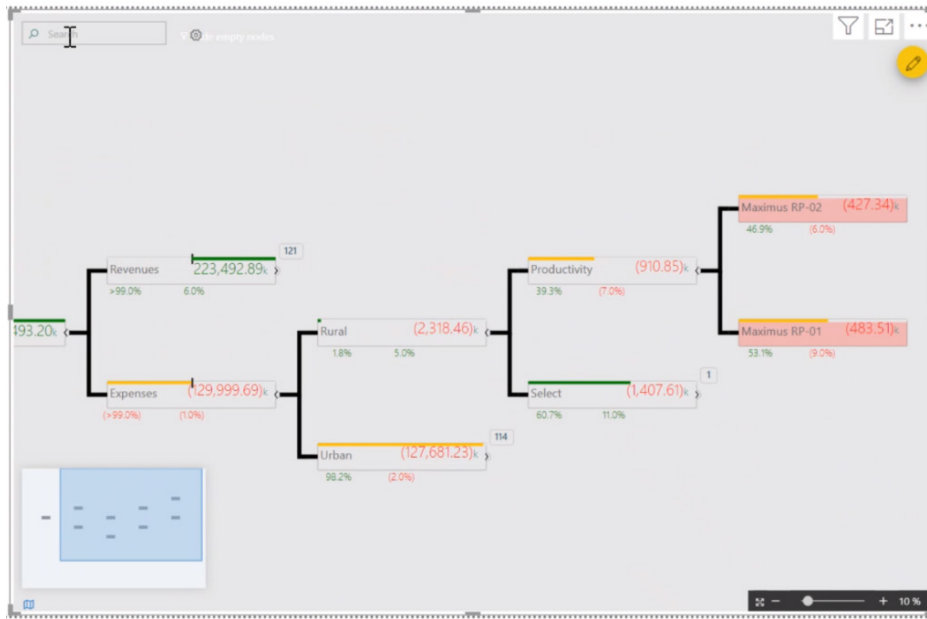
xViz has added four new visuals to their suite. As a reminder, xViz is a package of visuals with advanced formatting and configuration capabilities. You can use the basic capabilities for free or purchase a license that will give you access to all the visuals for one price.

HIERARCHY TREE

The hierarchy tree visual lets you display hierarchical data in a tree view. This visual is similar in layout to Microsoft's decomposition tree, but it allows you to compare two measures, such as budget and actual, and see the variance between them, viz.

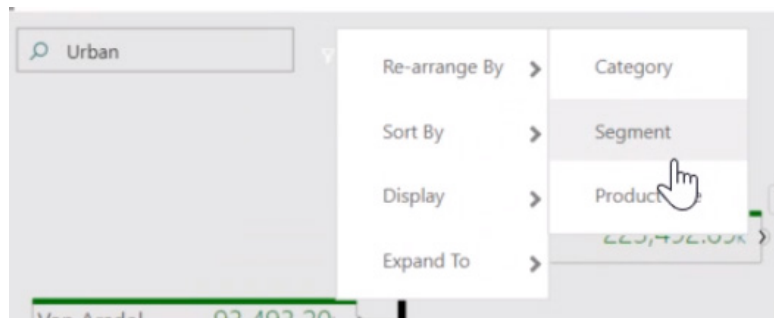


When your tree extends beyond the viewport, you have the ability to continuously pan with zoom control and there is even a mini-map to help you navigate. You can also search for nodes with the search box in the top right, which helps when the tree is quite large.



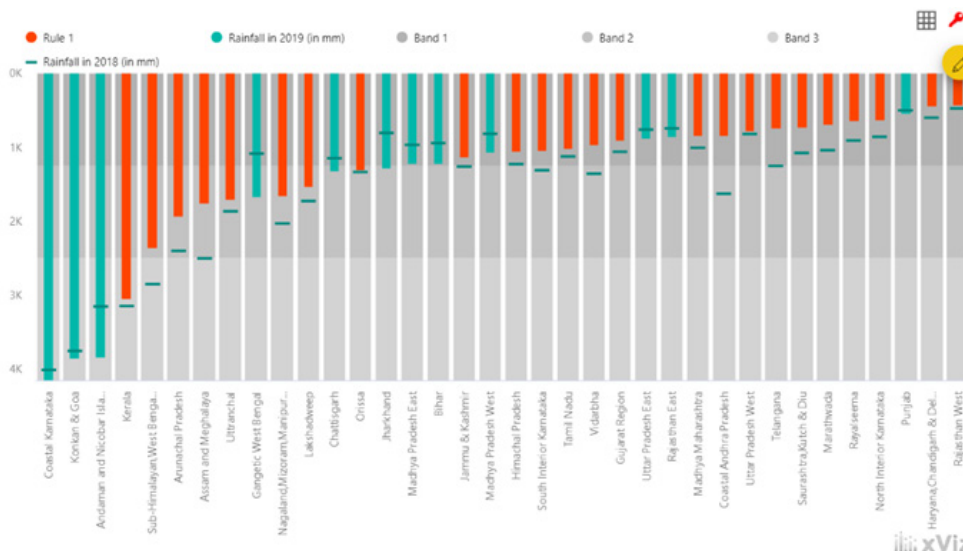
Optionally, for each child you can also see the number of nodes that haven't been expanded out. You can conditionally format the nodes based on the values displayed in them, and you can even control the look and precision of the values with advanced number and semantic formatting. The connector lines, the font formatting, and background

and node colours may also be controlled through the Formatting pane. Lastly, end-users may also change and rearrange the tree by using the context menu:



BULLET CHART

The bullet chart visual is a variation of a bar chart that is similar to a gauge, by allowing you to analyse actual values against a series of qualitative bands, and a comparison or goal value.

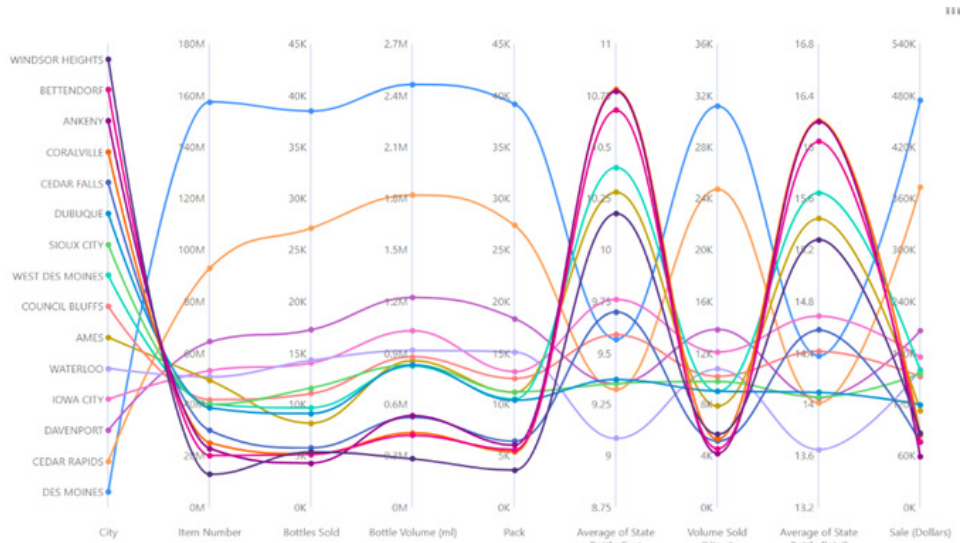


The three comparison bands can be configured to specific percentages or controlled through a field in your model. You can also give each band a name and colour. Additionally, you can control the marker type and size.

There are also advanced formatting capabilities, such as number and semantic formatting, chart orientation and axis orientation colour and text formatting.

PARALLEL COORDINATES

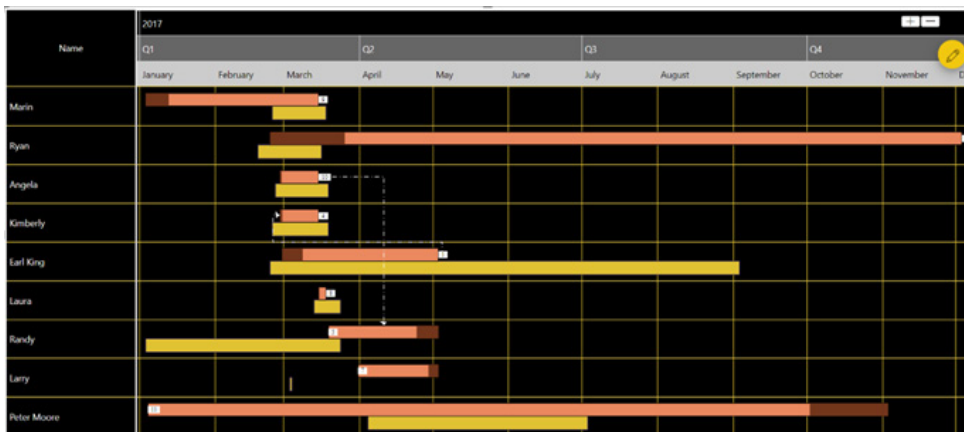
The parallel coordinates visual allows you to plot multivariate numerical data, enabling you to showcase multi-dimensional data and their relationships. For example, you may wish to compare different products in different stores based on multiple measures such as revenue or units sold.



This visual allows you to extensively customise the x-axis and y-axis. You can format the orientation, colour, font size and family, and axis colours. It also has the same advanced number formatting and utility menu capabilities that all xViz visuals have

GANTT CHART

The Gantt chart visual is a useful tool for project managers that allows you to track deliverables, understand dependencies between tasks, spot missed deadlines and see milestones.



For each task you can set:

- Task Name
- Actual start and end dates
- Planned start and end dates
- Progress
- Connectors to show related tasks.

This visual is also highly customisable. You can set the background and font colour for the grid and headers. For the timeline, you can control each level separately, setting the fill and font colours and font size. For milestones, you can set the fill and border colour as well as the shape

of the milestone. You can use number and semantic formatting for both the progress and ranking numbers and can even apply conditional formatting to the progress. You can turn on or off the zoom buttons, which allow you to more easily navigate long or complex projects.



You can also switch the Gantt chart to a resource chart if you just want to understand the amount of work assigned to each of your resources.



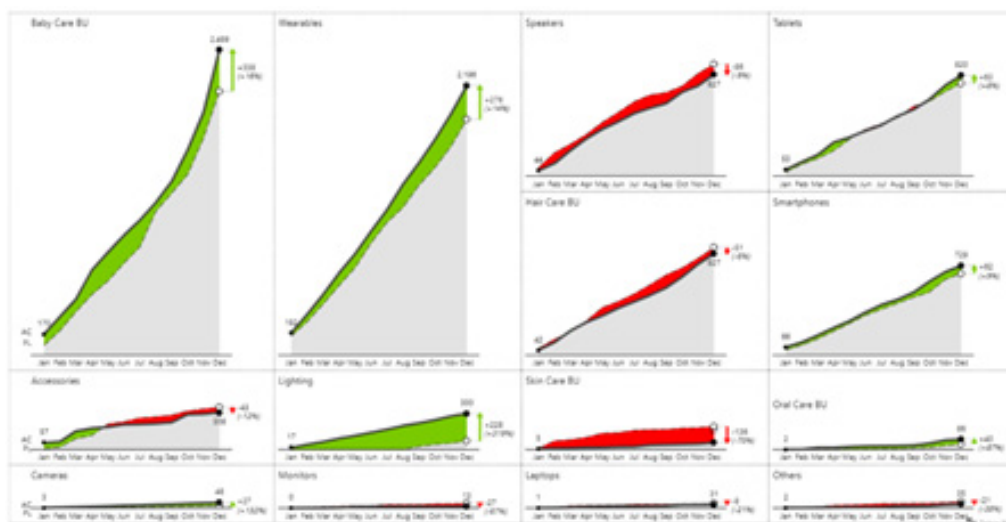
New Zebra BI visuals

ZEBRA BI CHARTS

The Zebra BI charts visual is one graphic that can be many different charts. Users are able to pick between 12 different charts, including:

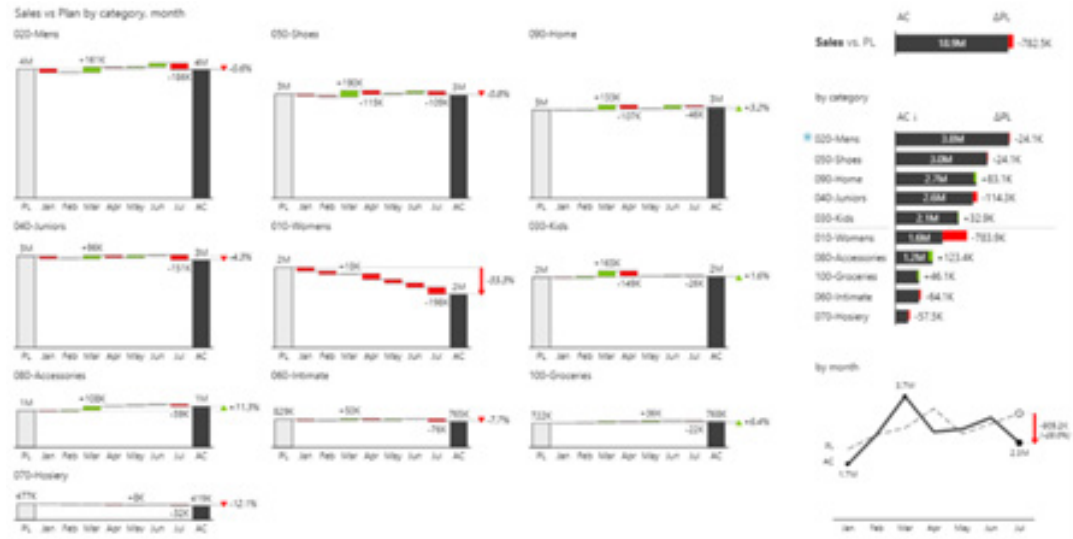
- Waterfall
- Variance
- Column
- Area
- Line
- Dot
- Combo.

Additionally, all of these can be visualised as small multiples:



This visual is both customisable and feature rich. Some of the highlights of the visual include:

- Calculates absolute and relative variances
- Calculates YoY growth rates
- Advanced layouts for small multiples
- Advanced filters such as Top / Bottom N
- Difference highlights
- Drill-down support
- Axis breaks
- Custom number formats
- Works in Report Server and mobile applications
- Fully responsive visual.



ZEBRA BI TABLES

The Zebra BI Tables visual is a flexible table and matrix visual that supports embedding visuals with the rows. You're able to show hierarchies with support for expanding and collapsing rows, reordering columns and more.



This visual is also highly customisable and includes the following features:

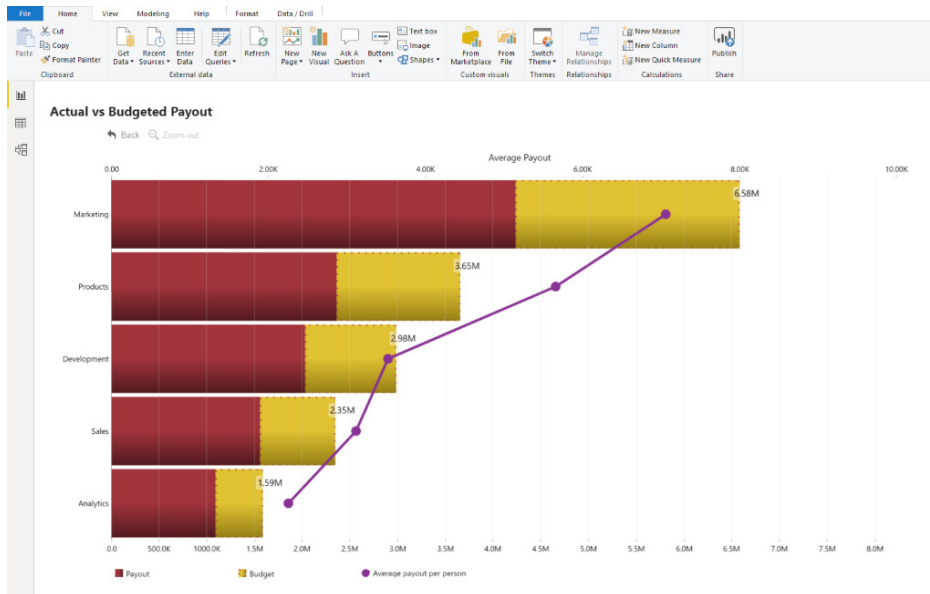
- Automatic variance charts
- Variance charts, waterfall charts, bar charts, lollipop charts within the rows
- P&L calculations with subtotals
- Flexible table or matrix layouts
- 1-click sort by any column
- Expand or collapse rows – unlimited levels
- Reorder columns by dragging
- Top N + others analysis
- Hide or unhide columns
- Column hierarchies
- Custom number formats
- Works in Report Server and mobile applications.



Drill Down Combo Bar PRO by ZoomCharts

Drill Down Combo Bar PRO is another addition to the ZoomCharts custom visuals suite and has all ZoomCharts signature features, such as interactive drill down, smooth animations, rich customisation options and full measure support. The Drill Down Combo Bar PRO puts a fresh

spin on the traditional bar chart by enabling you to combine bars, lines, and areas. The visual offers multiple stacking and clustering options. Each series may be customised individually, and it is possible to add thresholds to visualise targets on top of actual data.



The key features are:

- Multiple chart types: combine bars with lines and areas
- Many stacking and clustering configuration options
- Multiple x-axes: separate values with scale differences
- Rich customisation options: customise each series separately (colours, outlines, column widths, fonts, value labels and more)
- Static and dynamic thresholds: set up to four (4) thresholds to demonstrate targets or benchmarks
- Touch-driven slicer: filter the report page by using the visual itself (no need for external slicers)
- On-chart interactions: zoom, click and drag or drill down to explore and filter data
- Mobile friendly: use on touch and multi-touch devices.

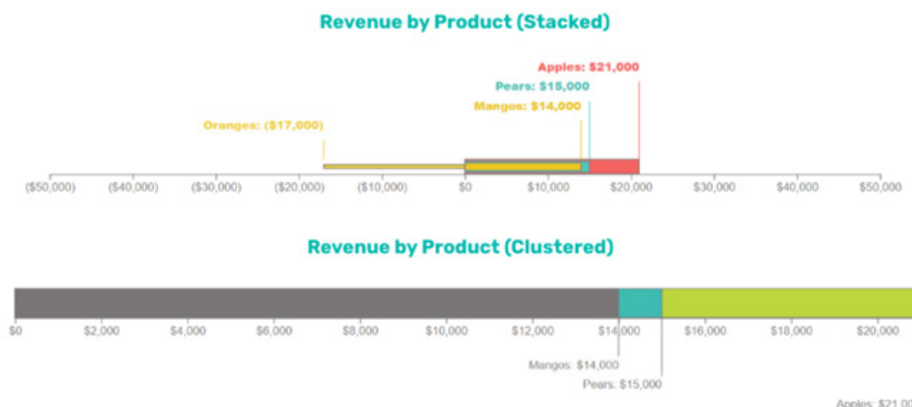
Some common use cases for the visual include:

- Sales & Marketing: monitoring sales results and campaign-by-campaign marketing metrics
- Human Resource: managing hiring, sick days, overtimes and efficiency ratios by department
- Accounting and Finance: compare financial performance by region, office or business line
- Manufacturing: view production efficiencies and quality metrics by product line.

Annotated Bar by Queryon

The Annotated Bar by Queryon visual lets you create various forms of bar charts, such as a stacked, clustered or overlapping bar charts, and combine that with customisable labels to help your users notice to key

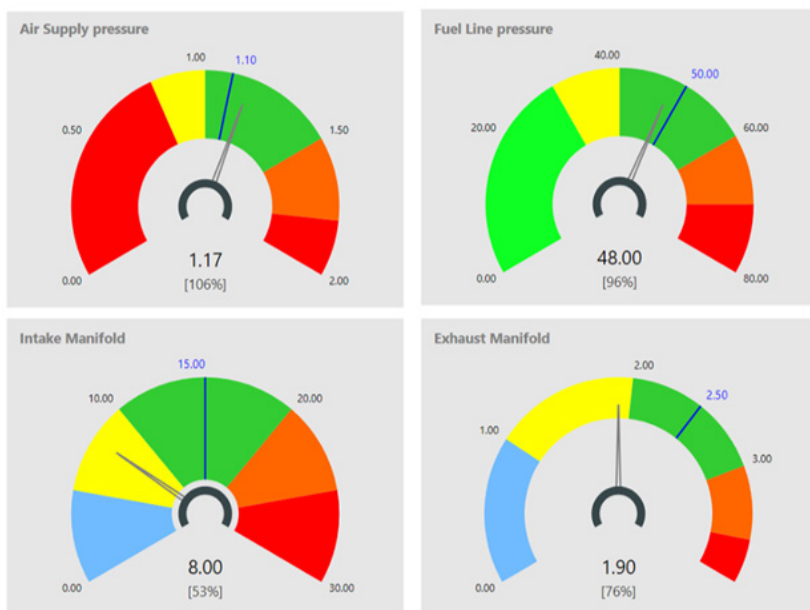
data points. The labels can be formatted with control over the colour, font family, size and position. The visual also supports drilldown, which can be performed by clicking on the bar or the label.



Tachometer 5 Ranges

The Tachometer 5 Ranges visual by Dimensional Strategies Inc. is a gauge visual that allows you to visualise your metric using five (5) different ranges. Some of the customisation options include changing

the colours and labels, setting data driven or static range limits, and setting start and end values and angles.



ValQ licensing updates

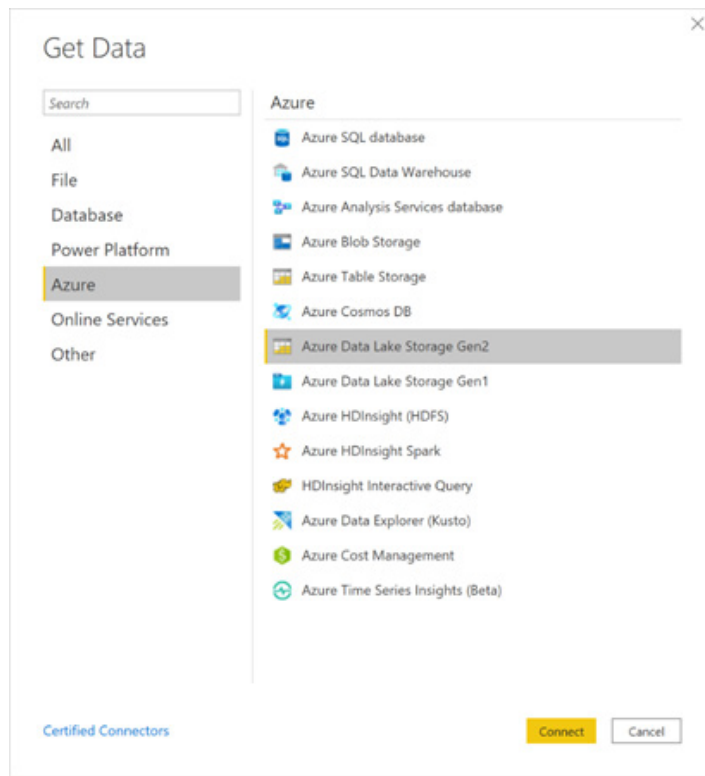
We featured ValQ for Modern Digital Planning back in June 2019's newsletter. Since then, ValQ has had several major enhancements bringing several new use cases which are more than just visualisations

and what-if analysis. Recently, they have changed their pricing to introduce per user pricing model and a freemium offering with reasonable data / node limits.

Azure Data Lake Storage Gen2 connector is now Generally Available

For the past few months, the Azure Data Lake Storage Gen2 connector was available as a Beta connector in Power BI Desktop. From late December, Microsoft has announced that this connector is now

supported for the Power BI Service refresh and has also reached General Availability. This connector can be found in the Azure category of the Get Data dialog.

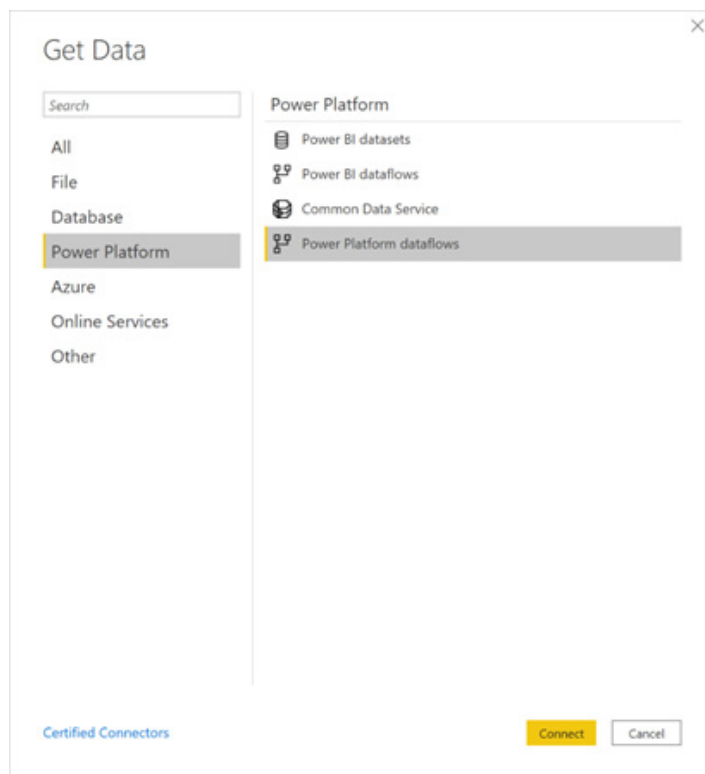


Power Platform Dataflows connector

One of the greatest additions to the Power Platform over the last year has been Dataflows. Dataflows enable self-service data preparation and reusability of data for users across Power BI, Power Apps and the rest of the Power Platform.

With this release, Microsoft has issued a new Power Platform Dataflows connector that allows you to create reports in Power BI on top of

Dataflows created from Power Apps environments. This new ability is designed for users who mainly work and develop inside Power Apps and have existing dataflows they would like to leverage inside Power BI. This connector can be found in the Power Platform category of the Get Data dialog.



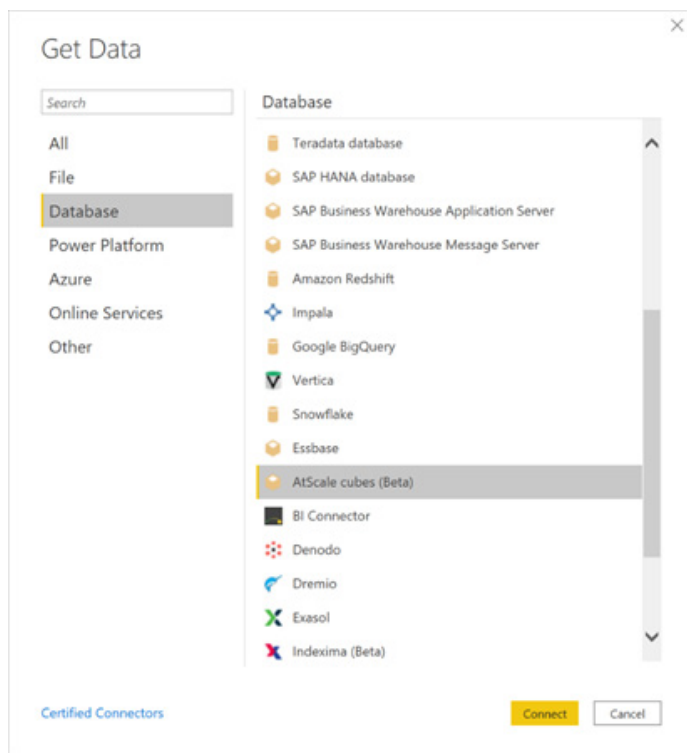
PostgreSQL connector now includes the Npgsql provider

An update to the PostgreSQL connector is that it now includes the Npgsql provider (that's a lot of points in Scrabble). This means that with the out-of-box experience, you will no longer need to install Npgsql

yourself. In addition, the Power BI Service now supports cloud-to-cloud refresh of PostgreSQL. If you want to continue using your Gateway (for firewall related reasons) you will still be able to do so.

AtScale connector is now Generally Available

The AtScale connector is now Generally Available. This connector can be found in the Database category of the Get Data dialog.



Azure Time Series Insights connector

Azure Time Series Insights is an end-to-end Platform-as-a-Service (PaaS) offering used to collect, process, store, analyse and query data (at an Internet of Things scale) that is highly contextualised and optimised for time series. Time Series Insights now seamlessly integrates with Power

BI, providing users with more powerful visualisation and dashboarding capabilities over streaming data, and allowing users to share insights and results across their organisation.

Data Virtuality connector

The Data Virtuality Logical Data Warehouse, a high-performance data virtualisation solution, allows you to leverage your existing data environment through instant data access, data centralisation, automation and data governance. This solution marries two distinct technologies to create an entirely new way to integrate data. The combination of data virtualisation and next generation Extract

Transform and Load (ETL) enables an agile data infrastructure with high performance. With the Data Virtuality Logical Data Warehouse, you can connect to multiple data sources and query data by using SQL, regardless of the format of the source file. It enables you to integrate your data and create a central data logic that covers the business logic and the logical connections between the different systems.

Zucchetti HR Infinity connector

Zucchetti HR software supports the HR department in valuing talents, in improving the corporate environment and in reaching strategic goals: this database and its native integration seeks the optimisation of all administrative, management and HR analysis processes.

Updates to AI Insights functions (Preview)

November saw the release of Azure ML, text analytics and image tagging in Preview. There are two updates related to these features in this release.

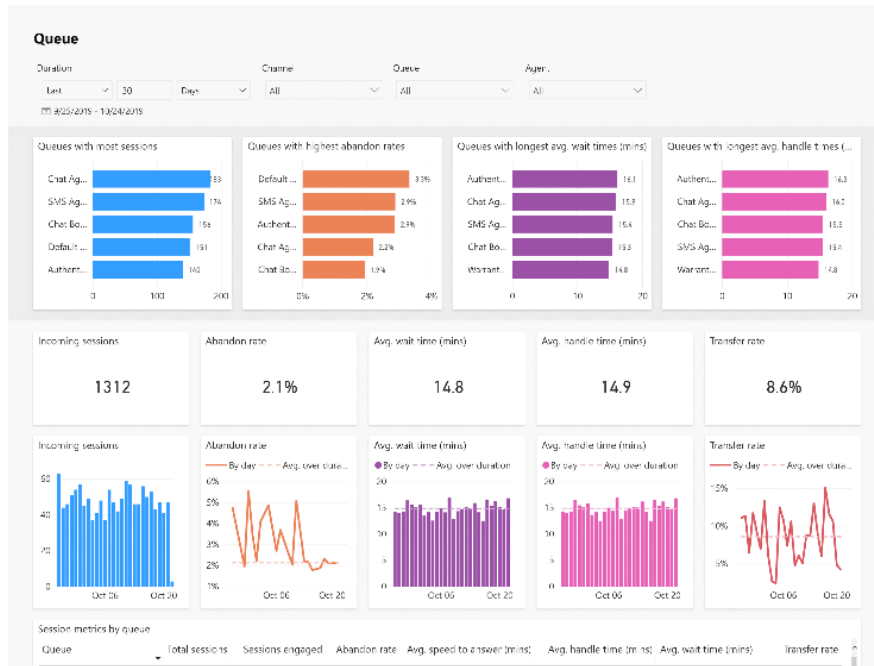
Secondly, some users might have experienced an issue with retrieving the text analytics and image tagging functions, even though you may have a Premium capacity. This happened only for capacities where the administrator had not made changes to the AI workload settings. This issue is now fixed.

First, when you connect to an Azure ML model, the columns in your table are automatically mapped to similarly named parameters in the Azure ML model. For example, if the model is expecting 'Amount', 'Product type', and 'Tenure', the columns in your table with these named are automatically selected from the dropdown lists.

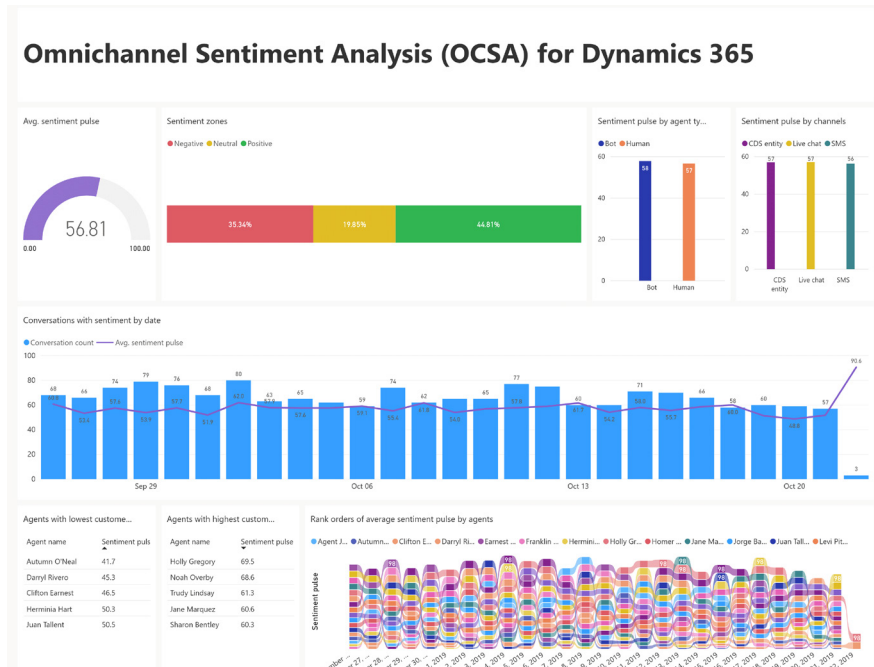
Omnichannel insights for Dynamics 365

Microsoft has also announced Omnichannel Insights for Dynamics 365 supporting Chat, Bot and SMS channels, all now Generally Available. Omnichannel dashboard helps supervisors with insights on operational metrics across different channels, queues, bots and agents. It also uses AI to provide in-depth sentiment analysis to help supervisors take actions to improve customer satisfaction.

This template app has two dashboards and six visually rich report pages with filters for supervisors to get a summary on how their organisation is engaging and connecting to their customers through Omnichannel for customer service.



The Omnichannel Sentiment Analysis dashboard, shown below, provides an overview of important KPIs and trends relative to the sentiment analysis of conversation offered.

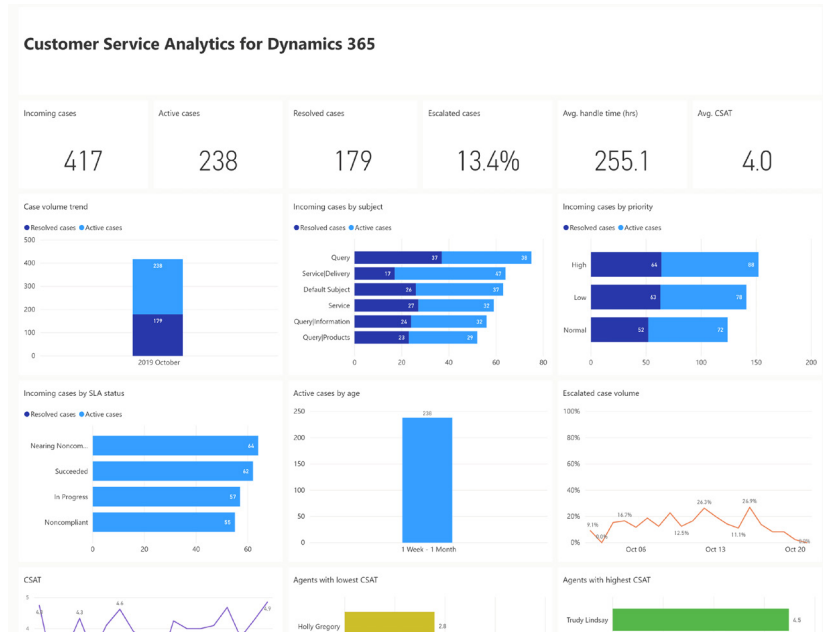


With this release, you can download the Power BI reports and customise them to meet your organisational needs.

It should be noted that Omnichannel for Customer Service is a paid add-on to Dynamics 365 Customer Service apps that use the Unified Interface. It is available only when you purchase a subscription to Chat for Dynamics 365 Customer Service or Dynamics 365 for Digital Messaging.

Customer service analytics for Dynamics 365

The Customer Service Analytics for Dynamics 365 is also now available. The Customer Service Analytics dashboard provides information about the historical operational metrics and KPIs to effectively manage contact centres.



With this release, you can also download the Power BI reports and customise them to your organisational needs.

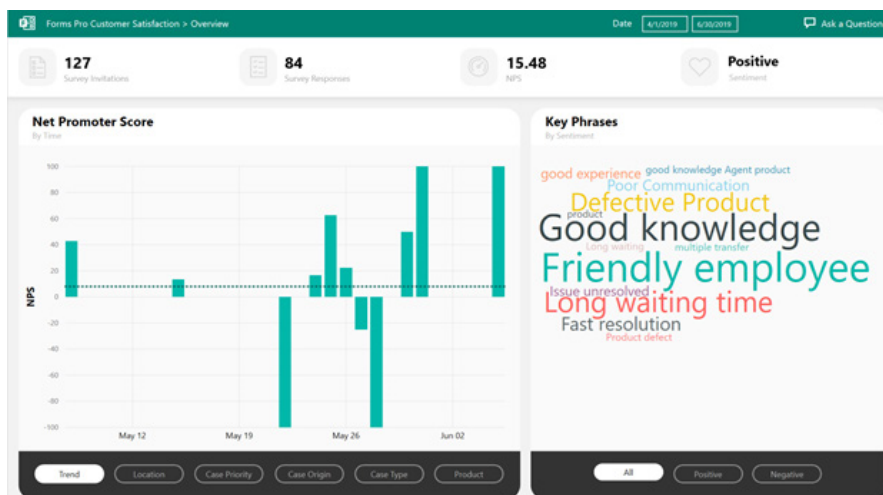
Microsoft Forms Pro for Customer Service

Microsoft Forms Pro is an enterprise survey capability that helps businesses obtain the feedback they need to make smarter decisions. It enables you to collect feedback using surveys and connect the survey response data with the business data to obtain meaningful insights.

If you are using Forms Pro to collect customer feedback after a case is resolved in Dynamics 365 Customer Service, you can automate the feedback collection and association of survey response data and case data using Power Automate. The Forms Pro survey response data will be stored in same Dynamics 365 organisation along with your business data. The Forms Pro Customer Satisfaction app will enable you to get analytics combining survey data and case data. To connect to the

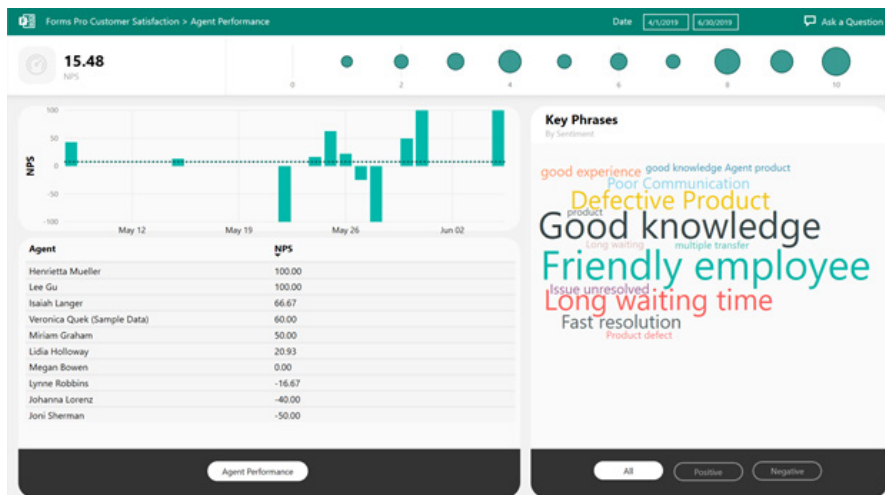
Dynamics 365 Customer Service instance, you will need to provide the URL for your organisation and the credentials for authentication.

This app template brings together the survey feedback and Dynamics 365 Customer Service data into a three-page report to provide a useful out-of-box experience for customer service managers. The first page contains overview of the survey response data and provides metrics like NPS, NPS trend, word cloud of the customer verbatim phrases, and a number of survey invitations and responses. It also provides NPS distribution with respect to the different case parameters like location, case priority, case origin, case type and product.

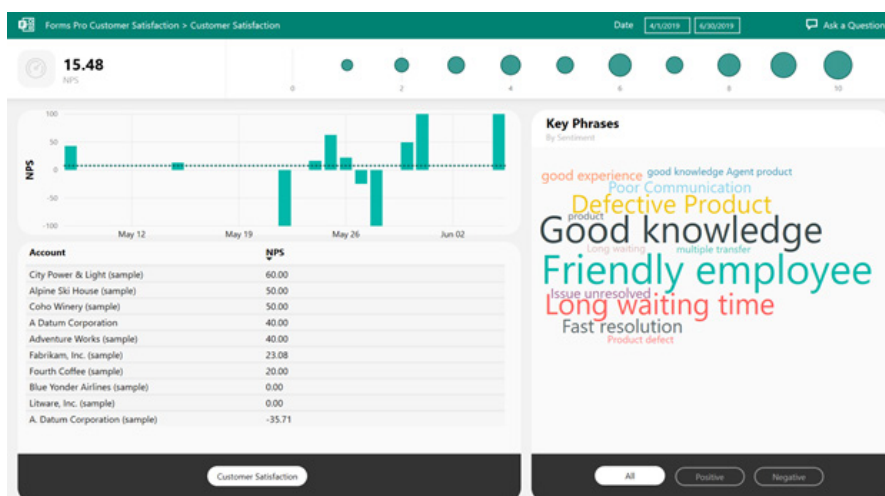


The second page provides an agent-centric view of the satisfaction data. This page will help customer service managers to analyse an agent's performance based on the survey responses. They can view

how the customer service agents are performing and what feedback the customers are providing about them.



The third page provides customer-centric view of the satisfaction data, which shows the satisfaction rating trend for the customers and their feedback.



More soon, we're sure!

Microsoft Ignite the Tour: Taipei



SumProduct is proud to announce that our very own Liam Bastick will be presenting in Taipei (Taiwan) in February. The event is held on Monday 17 and Tuesday 18 February, and more details may be found at www.microsoft.com/zh-tw/ignite-the-tour/taipei.

Liam's topic will be *Financial Modelling in Power BI*, where he will show you how to build three-way integrated financial statements into the business intelligence suite.

The A to Z of Excel Functions: DPRODUCT

This function multiplies the values in a **field** (column) of records in a list or **database** that match conditions that you specify.



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

DPRODUCT(database, field, criteria)

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

- **database**: this is required and represents the range of cells that makes up the list or database. A **database** is a list of related data in which rows of related information are records and columns of data are fields. The first row of the list contains labels for each column
- **field**: indicates which column is used in the function. Make sure you enter the column label enclosed between inverted commas (double quotation marks), e.g. "Age" or "Yield", or a number (without quotation marks) that represents the position of the column within the list, that is, 1 for the first column, 2 for the second column, and so on. Microsoft's documentation states that this argument is required. We're not convinced. If **field** is omitted, **DPRODUCT** multiplies all of the records in the table that match the criteria
- **criteria**: is the range of cells that contains the conditions you specify. You can use any range for the criteria argument, as long as it includes at least one column label and at least one cell below the column label in which you specify a condition for the column.

It should be further noted that:

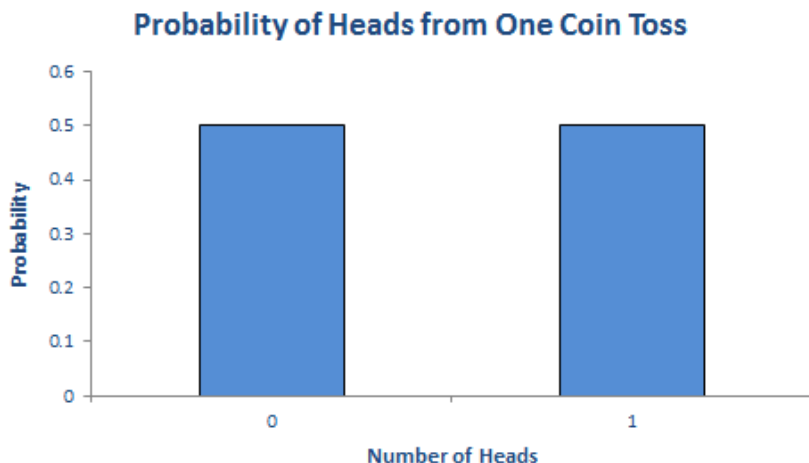
- you can use any range for the **criteria** argument, as long as it includes at least one column label and at least one cell below the column label for specifying the condition, e.g. if the range **G1:G2** contains the column label Income in **G1** and the amount 10,000 in **G2**, you could define the range as **MatchIncome** and use that name as the **criteria** argument in the database functions
- although the **criteria** range can be located anywhere on the worksheet, do not place the criteria range below the list. If you add more information to the list, the new information is added to the first row below the list. If the row below the list is not blank, Excel cannot add the new information
- make sure that the **criteria** range does not overlap the list
- to perform an operation on an entire column in a **database**, enter a blank line below the column labels in the **criteria** range.

Please see the example below:

	A	B	C	D	E
1	Tree	Height	Height		
2	Apple	>3	<=4		
3	Pear				
4					
5	Tree	Height	Age	Choices	Profit
6	Apple	6	20	2	105
7	Apple	4	12	2	96
8	Cherry	4.2	14	3	105
9	Apple	4.7	15	2	75
10	Pear	3	8	4	16.8
11	Apple	2.7	9	2	45
12					
13	Formula	Description	Result		
14	=DPRODUCT(A5:E11,"Choices",A1:C3)	The product of the number of choices for trees which are either apple trees with a height greater than three metres and less than or equal to four metres, or pear trees.	8		
15					

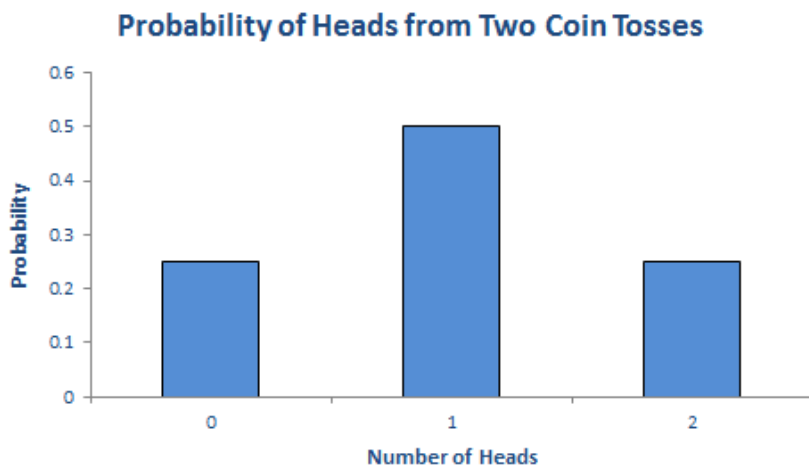
The A to Z of Excel Functions: DSTDEV

Imagine you toss an unbiased coin: half of the time it will come down heads, half tails:



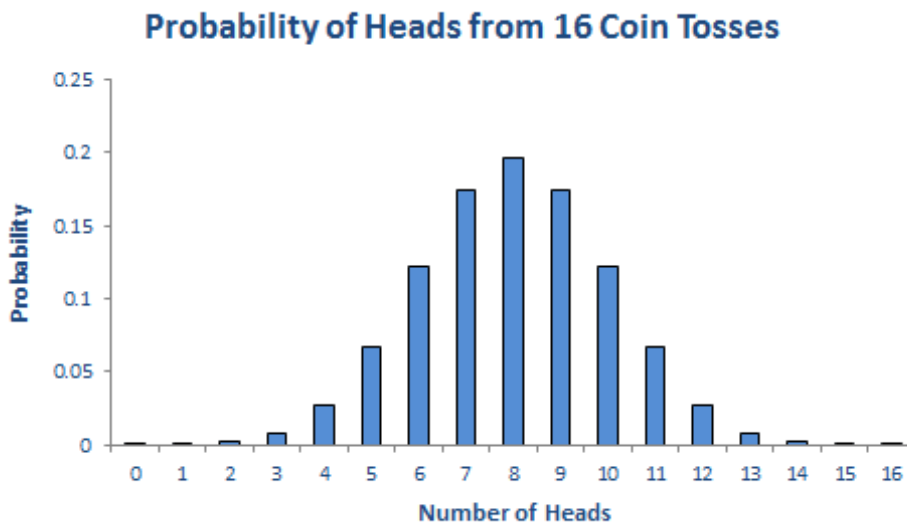
It is not the most exciting chart ever constructed, but it's a start.

If you toss two coins, there's four possibilities: two Heads, a Head and a Tail, a Tail and a Head, and two Tails.



You 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 1.

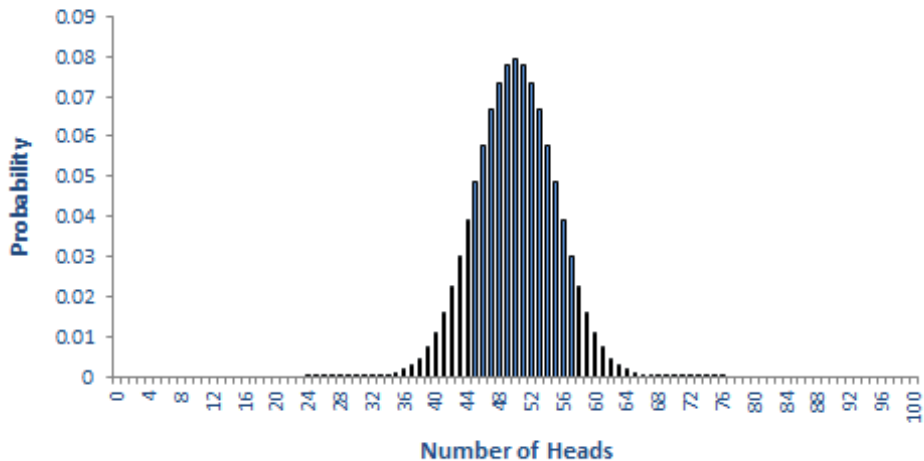
The story is similar if we consider 16 coin tosses say:



Again, if you were to add up all of the individual probabilities, they would total to 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?

Probability of Heads from 100 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 2.82×10^{-7} , *etc.*).

Now, I would like to ask the reader 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 4.0×10^2 **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, the *Japan Times* recently announced a new computer that could compute 10,000,000,000,000 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 I can't afford the electricity bill.

Let's put this all into perspective. All I am 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 I plotted charts for 1,000 or 10,000 coin tosses similar to the above, I 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: σ = standard deviation

N = total number of possible outcomes (for population, or is replaced by $n-1$, where n is the number of observations in a sample)

Σ = summation

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

μ = mean or average

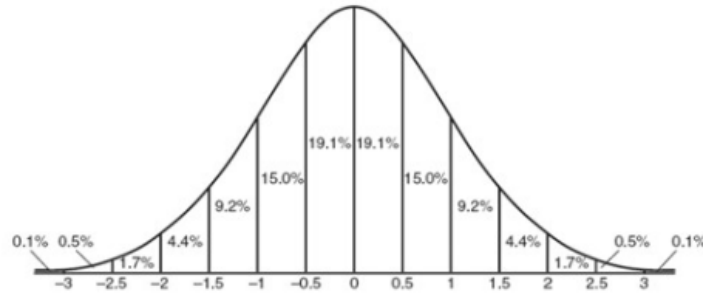
The Binomial distribution is not the most common distribution used in probability analysis: that honour belongs to the **Gaussian** or **Normal distribution**:



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 distribution.

The Normal distribution's population is spread as follow:



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.

– and here I will focus on the Normal distribution – it is possible to predict the mean and range of outcomes using a sampling method.

Therefore, if we know the formula to generate the probability distribution

For the record, the formula for the Normal distribution is given by

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

However, here we are concerned about the standard deviation.

The **DSTDEV** function estimates the standard deviation of a sample by using the numbers in a field (column) of records in a list or database that match conditions that you specify.

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

DSTDEV(database, field, criteria)

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

- **database:** this is required and represents the range of cells that makes up the list or **database**. A database is a list of related data in which rows of related information are records and columns of data are fields. The first row of the list contains labels for each column
- **field:** indicates which column is used in the function. Make sure you enter the column label enclosed between inverted commas (double quotation marks), *e.g.* "Age" or "Yield", or a number (without quotation marks) that represents the position of the column within the list, that is, 1 for the first column, 2 for the second column, and so on. Microsoft's documentation states that this argument is required. We're not convinced. If **field** is omitted, **DSTDEV** calculates the standard deviation of the sample of the records in the table that match the criteria
- **criteria:** is the range of cells that contains the conditions you specify. You can use any range for the **criteria** argument, as long as it includes at least one column label and at least one cell below the column label in which you specify a condition for the column.

It should be further noted that:

- you can use any range for the **criteria** argument, as long as it includes at least one column label and at least one cell below the column label for specifying the condition, *e.g.* if the range **G1:G2** contains the column label Income in **G1** and the amount 10,000 in **G2**, you could define the range as MatchIncome and use that name as the **criteria** argument in the database functions
- although the **criteria** range can be located anywhere on the worksheet, do not place the **criteria** range below the list. If you add more information to the list, the new information is added to the first row below the list. If the row below the list is not blank, Excel cannot add the new information
- make sure that the **criteria** range does not overlap the list
- to perform an operation on an entire column in a **database**, enter a blank line below the column labels in the **criteria** range.

Please see the example to the right:

	A	B	C	D	E
1	Tree	Height	Height		
2	Apple	>3	<=4		
3	Pear				
4					
5	Tree	Height	Age	Yield	Profit
6	Apple	6	20	14	105
7	Apple	4	12	10	96
8	Cherry	4.2	14	9	105
9	Apple	4.7	15	10	75
10	Pear	3	8	8	16.8
11	Apple	2.7	9	6	45
12					
13	Formula	Description	Result		
14	=DSTDEV(A5:E11,"Yield",A1:C3)	The estimated standard deviation in the yield of apple and pear trees in the second table based upon the criteria cited in the first table.	1.41421		

The A to Z of Excel Functions: DSTDEVP

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

DSTDEVP(database, field, criteria)

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

- **database:** this is required and represents the range of cells that makes up the list or database. A **database** is a list of related data in which rows of related information are records and columns of data are fields. The first row of the list contains labels for each column
- **field:** indicates which column is used in the function. Make sure you enter the column label enclosed between inverted commas (double quotation marks), e.g. "Age" or "Yield", or a number (without quotation marks) that represents the position of the column within the list, that is, 1 for the first column, 2 for the second column, and so on. Microsoft's documentation states that this argument is required. We're not convinced. If **field** is omitted, **DSTDEVP** calculates the standard deviation of the total population of records in the table that match the criteria
- **criteria:** is the range of cells that contains the conditions you specify. You can use any range for the **criteria** argument, as long as it includes at least one column label and at least one cell below the column label in which you specify a condition for the column.

It should be further noted that:

- you can use any range for the **criteria** argument, as long as it includes at least one column label and at least one cell below the column label for specifying the condition, e.g. if the range **G1:G2** contains the column label Income in **G1** and the amount 10,000 in **G2**, you could define the range as **MatchIncome** and use that name as the **criteria** argument in the database functions
- although the **criteria** range can be located anywhere on the worksheet, do not place the **criteria** range below the list. If you add more information to the list, the new information is added to the first row below the list. If the row below the list is not blank, Excel cannot add the new information
- make sure that the **criteria** range does not overlap the list
- to perform an operation on an entire column in a **database**, enter a blank line below the column labels in the **criteria** range.

Please see the example below:

	A	B	C	D	E
1	Tree	Height	Height		
2	Apple	>3	<=4		
3	Pear				
4					
5	Tree	Height	Age	Yield	Profit
6	Apple	6	20	14	105
7	Apple	4	12	10	96
8	Cherry	4.2	14	9	105
9	Apple	4.7	15	10	75
10	Pear	3	8	8	16.8
11	Apple	2.7	9	6	45
12					
13	Formula	Description	Result		
		The estimated standard deviation in the yield of apple and pear trees in the second table based upon the criteria cited in the first table, assuming the second table is the entire population.			
14	=DSTDEVP(A5:E11,"Yield",A1:C3)		1.00000		
15					

The A to Z of Excel Functions: DSUM

This function adds the numbers in a **field** (column) of records in a list or **database** that match conditions that you specify.



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

DSUM(database, field, criteria)

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

- **database:** this is required and represents the range of cells that makes up the list or database. A **database** is a list of related data in which rows of related information are records and columns of data are fields. The first row of the list contains labels for each column
- **field:** indicates which column is used in the function. Make sure you enter the column label enclosed between inverted commas (double quotation marks), e.g. "Age" or "Yield", or a number (without quotation marks) that represents the position of the column within the list, that is, 1 for the first column, 2 for the second column, and so on. Microsoft's documentation states that this argument is required. We're not convinced. If **field** is omitted, **DSUM** identifies the sum of all of the records in the table that match the criteria
- **criteria:** is the range of cells that contains the conditions you specify. You can use any range for the **criteria** argument, as long as it includes at least one column label and at least one cell below the column label in which you specify a condition for the column.

It should be further noted that:

- you can use any range for the **criteria** argument, as long as it includes at least one column label and at least one cell below the column label for specifying the condition, e.g. if the range **G1:G2** contains the column label Income in **G1** and the amount 10,000 in **G2**, you could define the range as **MatchIncome** and use that name as the criteria argument in the database functions
- although the **criteria** range can be located anywhere on the worksheet, do not place the **criteria** range below the list. If you add more information to the list, the new information is added to the first row below the list. If the row below the list is not blank, Excel cannot add the new information
- make sure that the **criteria** range does not overlap the list
- to perform an operation on an entire column in a **database**, enter a blank line below the column labels in the **criteria** range.

Please see our final example below:

	A	B	C	D	E
1	Tree	Height	Height		
2	Apple	>3	<=4		
3	Pear				
4					
5	Tree	Height	Age	Yield	Profit
6	Apple	6	20	14	105
7	Apple	4	12	10	96
8	Cherry	4.2	14	9	105
9	Apple	4.7	15	10	75
10	Pear	3	8	8	16.8
11	Apple	2.7	9	6	45
12					
13	Formula	Description	Result		
14	=DSUM(A5:E11,"Profit",A1:A2)	The total profit from all of the apple trees	321		
15	=DSUM(A5:E11,4,A1:C3)	The total yield (column 4) from apple trees which are greater than three metres, but are less than or equal to four metres, or pear trees.	18		
16					

Criteria Examples

Typing an equal sign in a cell indicates you want to enter a formula. To display text that includes an equal sign, surround the text and the equal sign with double quotes, like so:

"=Liam"

You also do that if you're entering an expression (a combination of formulas, operators, and text) and you want to display the equal sign instead of have Excel use it in a calculation. For example:

">= entry"

Where *entry* is the text or value you want to find. For example:

What you type in the cell	What Excel evaluates and displays
"=Liam"	=Liam
">=1"	>=1

- When filtering text data, Excel does not distinguish between uppercase and lowercase characters. However, you can use a formula to perform a case-sensitive search (*see below*).

The following sections provide examples of complex criteria.

Multiple criteria in one column

Boolean logic: (Salesperson = "Tim" **OR** Salesperson = "Kathryn")

To find rows that meet multiple criteria for one column, type the criteria directly below each other in separate rows of the criteria range.

e.g. In the following data range (A5:C9), the criteria range (B1:B3) displays the rows that contain either "Tim" or "Kathryn" in the **Salesperson** column (B5:B9).

	A	B	C
1	Service	Salesperson	Sales
2		=Tim	
3		Kathryn	
4			
5	Service	Salesperson	Sales
6	Consulting	Kathryn	\$ 1,744
7	Training	Tim	\$ 9,183
8	Auditing	Kathryn	\$ 2,599
9	auditing	Cecile	\$ 1,117
10			

Multiple criteria in multiple columns where all criteria must be true

Boolean logic: (Service = "Auditing" AND Sales > 1500)

To find rows that meet multiple criteria in multiple columns, type all of the criteria in the same row of the criteria range.

In the following data range (A5:C9), the criteria range (A1:C2) displays all rows that contain "Auditing" in the **Service** column and a value greater than \$1,500 in the **Sales** column (C5:C9).

	A	B	C
1	Service	Salesperson	Sales
2	=Auditing		>1500
3			
4			
5	Service	Salesperson	Sales
6	Consulting	Kathryn	\$ 1,744
7	Training	Tim	\$ 9,183
8	Auditing	Kathryn	\$ 2,599
9	auditing	Cecile	\$ 1,117
10			

Multiple criteria in multiple columns where any criteria can be true

Boolean logic: (Service = "Auditing" OR Salesperson = "Kathryn")

To find rows that meet multiple criteria in multiple columns, where any criteria can be true, type the criteria in different rows of the criteria range.

In the following data range (A5:C9), the criteria range (A1:B3) displays all rows that contain "Auditing" in the **Service** column (C5:C9) or "Kathryn" in the **Salesperson** column (B5:B9).

	A	B	C
1	Service	Salesperson	Sales
2	=Auditing		
3		Kathryn	
4			
5	Service	Salesperson	Sales
6	Consulting	Kathryn	\$ 1,744
7	Training	Tim	\$ 9,183
8	Auditing	Kathryn	\$ 2,599
9	auditing	Cecile	\$ 1,117
10			

Multiple sets of criteria where each set includes criteria for multiple columns

Boolean logic: ((Salesperson = "Kathryn" AND Sales > 2000) OR (Salesperson = "Tim" AND Sales > 1500))

To find rows that meet multiple sets of criteria, where each set includes criteria for multiple columns, type each set of criteria in separate rows.

In the following data range (A5:C9), the criteria range (B1:C3) displays the rows that contain both "Kathryn" in the **Salesperson** column and a value greater than \$2,000 in the **Sales** column, or displays the rows that contain "Tim" in the **Salesperson** column (B5:B9) and a value greater than \$1,500 in the **Sales** column (C5:C9).

	A	B	C
1	Service	Salesperson	Sales
2		=Kathryn	>2000
3		Tim	>1500
4			
5	Service	Salesperson	Sales
6	Consulting	Kathryn	\$ 1,744
7	Training	Tim	\$ 9,183
8	Auditing	Kathryn	\$ 2,599
9	auditing	Cecile	\$ 1,117
10			

Multiple sets of criteria where each set includes criteria for one column

Boolean logic: ((Sales > 2000 AND Sales <= 3000) OR (Sales < 1500))

To find rows that meet multiple sets of criteria, where each set includes criteria for one column, include multiple columns with the same column heading.

In the following data range (A5:C9), the criteria range (C1:D3) displays rows that contain values between 2,000 and 3,000 and values less than 1,500 in the **Sales** column (C5:C9).

	A	B	C	D
1	Service	Salesperson	Sales	Sales
2			>2000	<=3000
3			<1500	
4				
5	Service	Salesperson	Sales	
6	Consulting	Kathryn	\$ 1,744	
7	Training	Tim	\$ 9,183	
8	Auditing	Kathryn	\$ 2,599	
9	auditing	Cecile	\$ 1,117	
10				

Criteria to find text values that share some characters but not others

To find text values that share some characters but not others, do one or more of the following:

- type one or more characters without an equal sign (=) to find rows with a text value in a column that begin with those characters. For example, if you type the text **Lia** as a criterion, Excel finds "Liam", "Liar" and "Lianne"
- use a wildcard character.

The following wildcard characters can be used as comparison criteria:

Use	To Find
? (question mark)	Any single character For example, sm?th finds "smith" and "smyth"
* (asterisk)	Any number of characters For example, *east finds "Northeast" and "Southeast"
~ (tilde) followed by ?, * or ~	A question mark, asterisk, or tilde For example, fy91~? finds "fy91?"

In the following data range (A5:C9), the criteria range (A1:B3) displays rows with "Co" as the first characters in the **Service** column or rows with the second character equal to "i" in the **Salesperson** column (B5:B9).

	A	B	C
1	Service	Salesperson	Sales
2	Co		
3		=?i*	
4			
5	Service	Salesperson	Sales
6	Consulting	Kathryn	\$ 1,744
7	Training	Tim	\$ 9,183
8	Auditing	Kathryn	\$ 2,599
9	auditing	Cecile	\$ 1,117
10			

Criteria created as the result of a formula

You can use a calculated value that is the result of a formula as your criterion. Remember the following important points:

- the formula must evaluate to TRUE or FALSE
- because you are using a formula, enter the formula as you normally would, and do not type the expression in the following way:
="= entry "
- do not use a column label for criteria labels; either keep the criteria labels blank or use a label that is not a column label in the range (in the examples below, **Calculated Average** and **Exact Match**)
- if you use a column label in the formula instead of a relative cell reference or a range name, Excel displays an error value such as #NAME? or #VALUE! in the cell that contains the criterion. You can ignore this error because it does not affect how the range is filtered
- the formula that you use for criteria must use a relative reference to refer to the corresponding cell in the first row (in the examples below, **C6** and **A6**)
- all other references in the formula must be absolute references.

The following subsections provide specific examples of criteria created as the result of a formula.

Filtering for values greater than the average of all values in the data range

In the following data range (**A5:C9**), the criteria range (**D1:D2**) displays rows that have a value in the Sales column greater than the average of all the values (**C6:C9**). In the formula, "**C6**" refers to the filtered column (**C**) of the first row of the data range (**6**).

	A	B	C	D
1	Service	Salesperson	Sales	Calculated Average
2				=C6>AVERAGE(\$C\$6:\$C\$9)
3				
4				
5	Service	Salesperson	Sales	
6	Consulting	Kathryn	\$ 1,744	
7	Training	Tim	\$ 9,183	
8	Auditing	Kathryn	\$ 2,599	
9	auditing	Cecile	\$ 1,117	
10				

Filtering for text by using a case-sensitive search

In the data range (**A5:C9**), the criteria range (**D1:D2**) displays rows that contain "Auditing" in the Service column by using the **EXACT** function to perform a case-sensitive search (**A5:A9**). In the formula, "**A6**" refers to the filtered column (**A**) of the first row of the data range (**6**).

	A	B	C	D
1	Service	Salesperson	Sales	Exact Match
2				=EXACT(A6,"Auditing")
3				
4				
5	Service	Salesperson	Sales	
6	Consulting	Kathryn	\$ 1,744	
7	Training	Tim	\$ 9,183	
8	Auditing	Kathryn	\$ 2,599	
9	auditing	Cecile	\$ 1,117	
10				

Upcoming SumProduct Training Courses

Location	Course	Date	Duration
Sydney	Power Pivot, Power Query and Power BI	17 - 19 Feb 2020	3 Days
Sydney	Excel Tips and Tricks	2 Mar 2020	1 Day
Sydney	Financial Modelling	3 - 4 Mar 2020	2 Days
Melbourne	Power Pivot, Power Query and Power BI	3 - 5 Mar 2020	3 Days
Sydney	Excel Tips and Tricks	6 Apr 2020	1 Day
Sydney	Financial Modelling	7 - 8 Apr 2020	2 Days
Melbourne	Excel Tips and Tricks	7 Apr 2020	1 Day
Melbourne	Financial Modelling	8 - 9 Apr 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	11 - 13 May 2020	3 Days
Melbourne	Power Pivot, Power Query and Power BI	18 - 20 May 2020	3 Days
Melbourne	Excel Tips and Tricks	25 May 2020	1 Day
Melbourne	Financial Modelling	26 - 27 May 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	16 - 18 Jun 2020	3 Days
Melbourne	Power Pivot, Power Query and Power BI	22 - 24 Jun 2020	3 Days
Melbourne	Excel Tips and Tricks	6 Jul 2020	1 Day
Melbourne	Financial Modelling	8 - 9 Jul 2020	2 Days
Sydney	Excel Tips and Tricks	14 Jul 2020	1 Day
Sydney	Financial Modelling	15 - 16 Jul 2020	2 Days
Melbourne	Power Pivot, Power Query and Power BI	20 - 22 Jul 2020	3 Days
Sydney	Power Pivot, Power Query and Power BI	27 - 29 Jul 2020	3 Days
Sydney	Excel Tips and Tricks	17 Aug 2020	1 Day
Sydney	Financial Modelling	18 - 19 Aug 2020	2 Days
Melbourne	Excel Tips and Tricks	18 Aug 2020	1 Day
Melbourne	Financial Modelling	19 - 20 Aug 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	24 - 26 Aug 2020	3 Days
Melbourne	Power Pivot, Power Query and Power BI	7 - 9 Sep 2020	3 Days
Sydney	Excel Tips and Tricks	21 Sep 2020	1 Day

Location	Course	Date	Duration
Sydney	Financial Modelling	22 - 23 Sep 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	6 - 8 Oct 2020	3 Days
Melbourne	Power Pivot, Power Query and Power BI	9 - 11 Nov 2020	3 Days
Sydney	Excel Tips and Tricks	9 Nov 2020	1 Day
Sydney	Financial Modelling	10 - 11 Nov 2020	2 Days
Melbourne	Excel Tips and Tricks	16 Nov 2020	1 Day
Melbourne	Financial Modelling	17 - 18 Nov 2020	2 Days
Sydney	Power Pivot, Power Query and Power BI	8 - 10 Dec 2020	3 Days
Melbourne	Power Pivot, Power Query and Power BI	14 - 16 Dec 2020	3 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 combine, **CTRL**, **ALT** and **SHIFT** with the function keys:

Keystroke	What it does
CTRL + ALT + SHIFT + F2	Print
CTRL + ALT + SHIFT + F4	Close application
CTRL + ALT + SHIFT + F9	Recalculate, after full re-build

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>.

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, Level 9, 440 Collins Street, Melbourne, VIC 3000
Registered Address: SumProduct Pty Ltd, Level 6, 468 St Kilda Road, Melbourne, VIC 3004

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