Excel for the web takes centre stage this month as Microsoft announces Office Scripts become Generally Available in the medium, together with a raft of other improvements.

As always, in our new "COVID-normal" newsletter, we have another Beat the Boredom Challenge, Charts & Dashboards, Visual Basics, Power Pivot Principles, Power Query Pointers, Power BI updates, Keyboard Shortcuts and you will also appreciate there will be no more "F" in A to Z of Excel Functions after this month – so to speak…

And we even have some news on SumProduct and yours truly…

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

Liam Bastick, Managing Director, SumProduct

## Lifetime Achievement Award

We start with some great news this month, even if it does appear to mean blowing smoke up our own proverbials, but congratulations to SumProduct's very own **Liam Bastick** who won the inaugural Lifetime Achievement Award at last week's Financial Modelling Summit 2021, held in London. If you cannot celebrate your own achievements, what can you do..?

Liam stated that he had "…been nervous about getting something called a 'Lifetime Achievement' award…", stating that he was "…scared people [would think he was] over the hill, … about to get [a] gold watch and be put out to be pasture before being shot…". Yup, sounds like Liam. On behalf of SumProduct,

he announced that to counter this, we will be introducing financial modelling for Power BI – not importing a model and presenting it, but actually building it too – very shortly. There is life in us old dogs yet.

The award is fantastic recognition for both SumProduct and Liam, who emphasised that he still had plenty to give the financial modelling community and would not be going "gently into the night" anytime soon. But he thanks the judges all the same.

You can watch the whole awards video at www.youtube.com/watch?v=48yorSjDQuM. And watch out – financial modelling in Power BI is coming *very* soon…
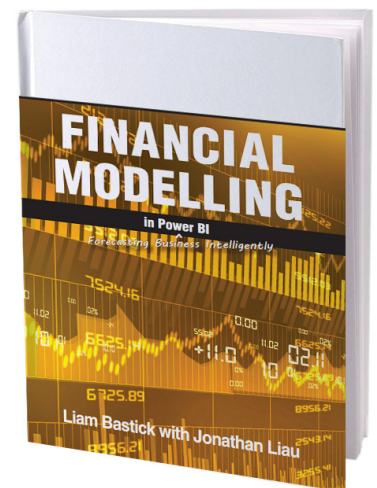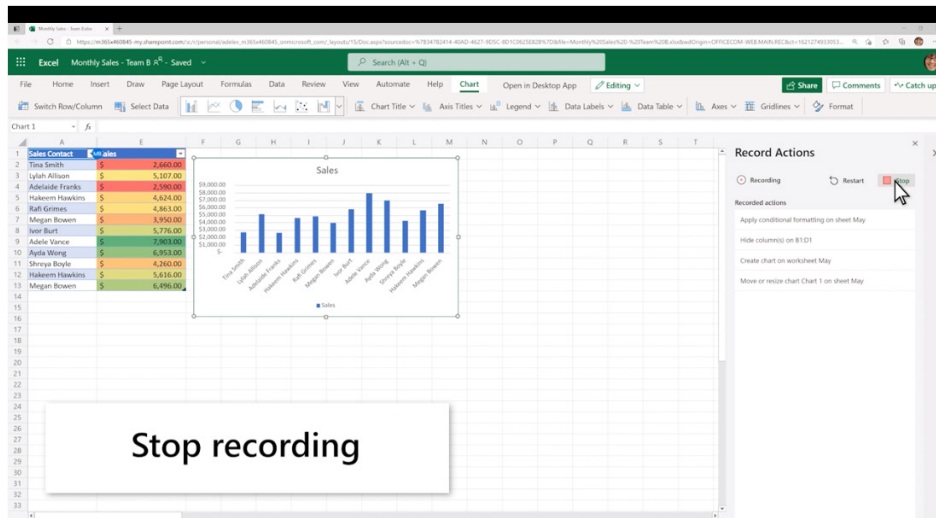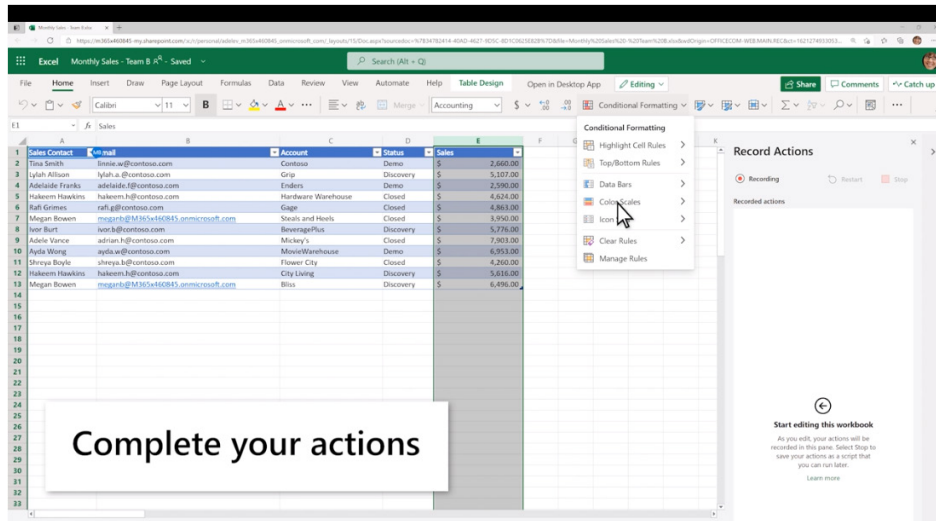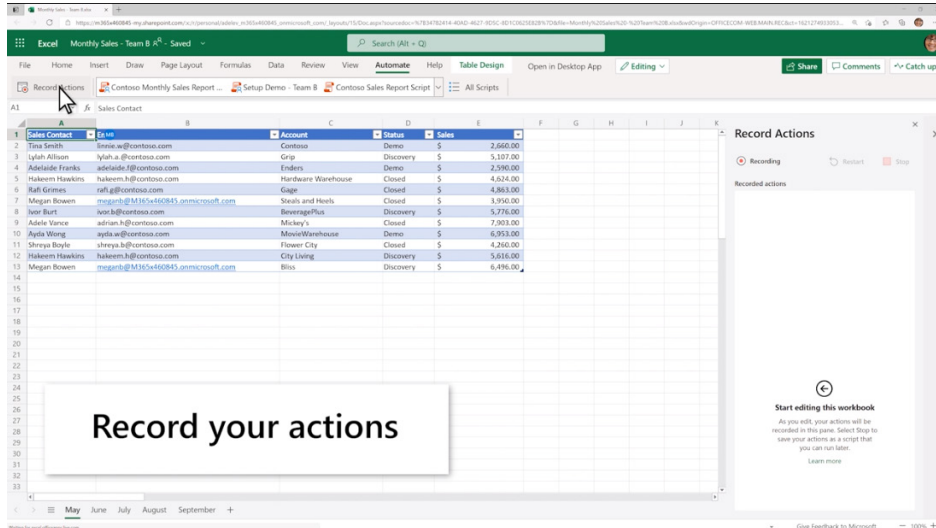
Office Scripts now Generally Available in Excel for the web

Microsoft has recently announced that Office Scripts in Excel for the web is now Generally Available for all "eligible users".

For those that may sound a little underwhelmed by this news, Office Scripts is an automation feature-set in Excel for the web that allows users with various levels of programming experience to automate their repetitive workflows. Unfortunately, I can't yet get it to work correctly for preparing this flamin' newsletter.

To get started, you should use the Action Recorder to record the actions you wish take in Excel (obviously, the Excel for the web version!). These actions are then translated into a script that you can run at any time. No programming experience is required and you can use the Code Editor afterwards to edit the resulting scripts, analagous to the actions we all take with VBA, for example. This uses a TypeScript-based editor directly within Excel for the web, and you may further implement it to edit your existing scripts or to create new ones using the Office Scripts API.

Record your actions

Complete your actions

Stop recording

Edit your script

Edit your script

Save your script

Run your script on any other workbook or worksheet

This may be used in conjunction with Power Automate to create a Flow to schedule your Office Script to run at a certain time or to trigger a script to run based upon the creation of a new file in a SharePoint site, *etc.*

Power Automate also allows you to trigger scripts based upon events from other applications and services to create cross-application workflows.
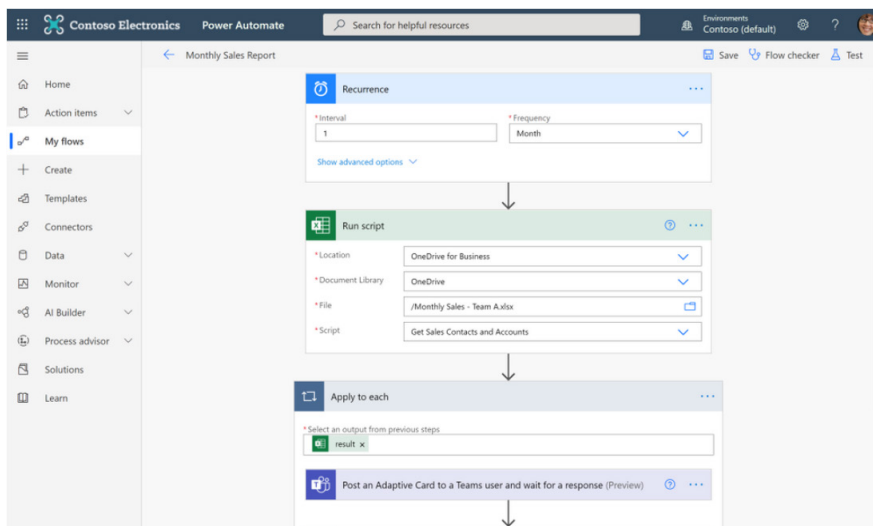
Office Scripts is currently available for all users that have a commercial or EDU license that gives access to the Microsoft 365 office desktop apps (*e.g.* Office 365 E3 and E5 licenses). If you have an eligible license, you'll find the Office Scripts feature-set in the Automate tab in the Ribbon.

Please note that if you do not have the Automate tab in Excel for the web, your administrator may have disabled the feature so you will need to go and talk to them nicely.

Other New Features in Excel for the Web

The General Availability of Office Scripts is only one of the new features and improvements to help you better analyse your data. There is more:

- Text to columns
- PivotTable improvements
- Sort improvements
- Interactive charts
- Additional keyboard shortcuts.

Let's take a look.

Text to columns

You can now split text into columns using delimiters. For example, with a few clicks, you can convert the ever-perennial "Hello World" in one cell to "Hello" and "World" across two adjacent cells. You may also select multiple delimiters including tab, semicolon, comma, space and custom [delimiter] *at the same time*. For example, "Hello, how are you?" in one cell would split into "Hello", "", "how", "are", and "you?" across five [5] cells, if you selected both comma and space as delimiters.
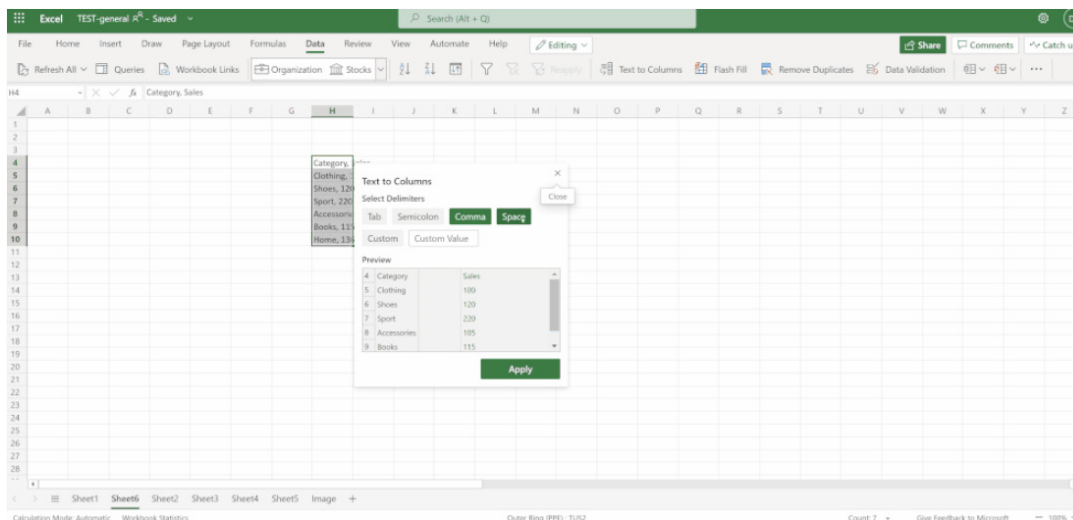
PivotTable improvements

You may analyse your data with the new and improved PivotTable experience in Excel for the web including:

- create a PivotTable quickly with Recommended PivotTables
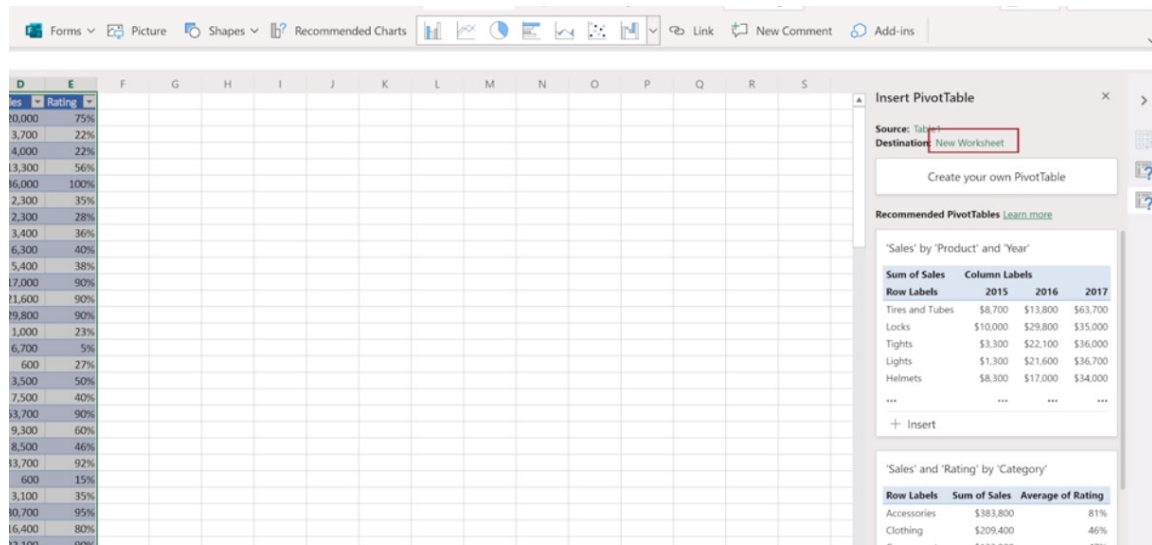
- change the settings and visualisation of your PivotTable via the new PivotTable Ribbon, including PivotTable Styles and the new 'PivotTable Settings' task pane
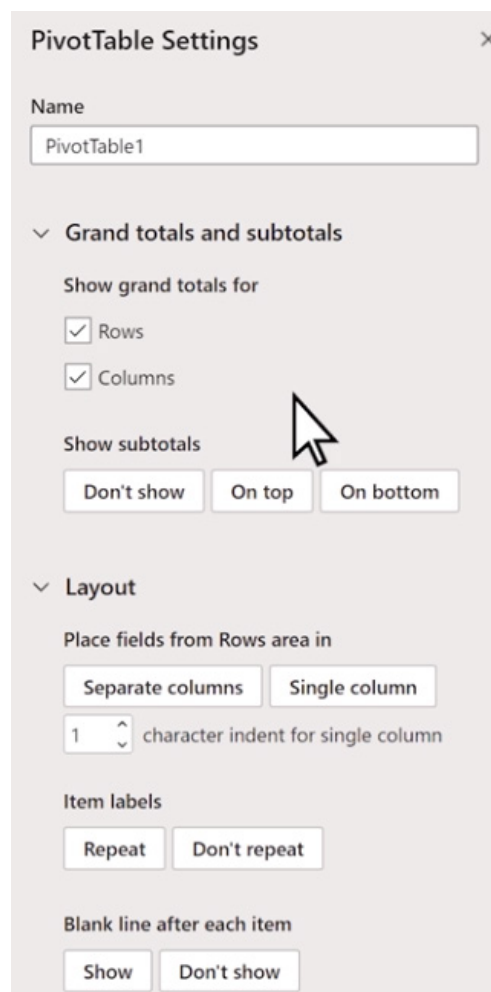
- copy, refresh and show or hide PivotTable task panes with one click, using the new PivotTable right-click menu.

Sort improvements

Sorting is fundamental in data analysis. Sorting can be more than numerical or alphabetical ordering though. In Excel for the web, you can now sort by Cell Color, Font Color, and Conditional Formatting Icon, in addition to Cell Values. Furthermore, you can sort by more than one column.
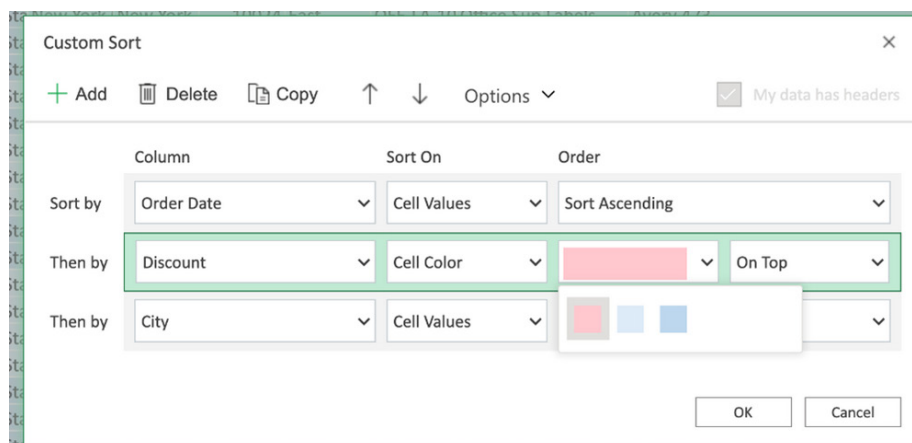
Interactive charts

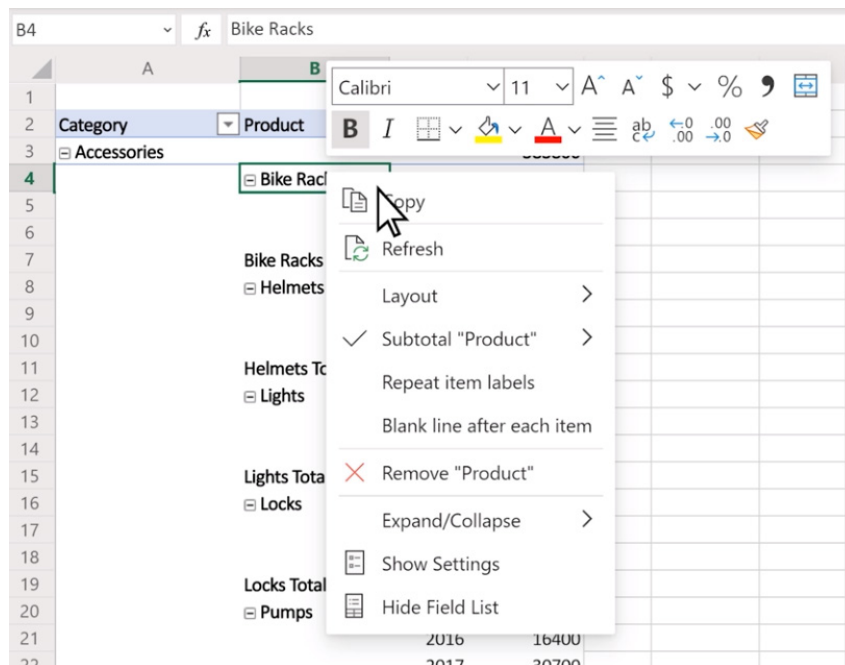You can now directly select and interact with chart elements when formatting a chart. Double-clicking a chart element launches the format task pane while single-clicking a chart element navigates you to its corresponding formatting options. In addition, non-data chart elements can now be removed from the chart using the delete / backspace keys. You will also be able to see a preview of your selection as you hover your mouse cursor over different chart elements.
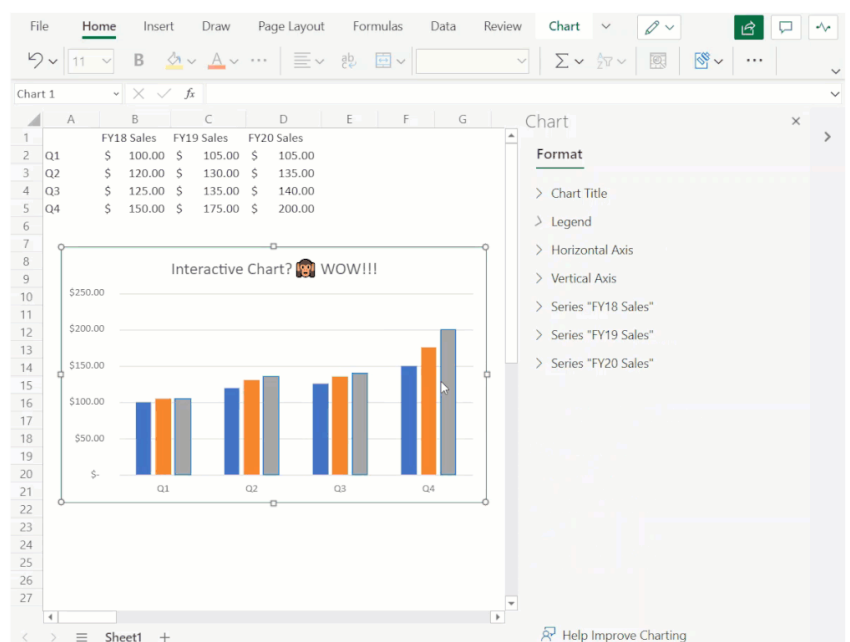
Additional keyboard shortcuts

Many users have asked for additional well-known and commonly used keyboard shortcuts in Excel for the web, so here are some of the new ones:

- **End arrows:** move to the last cell in the row or column
- **CTRL + END:** move to the last cell that contains data or formatting
- **CTRL + SHIFT + END:** extend the selection to the last used cell.

Also, coming soon are:
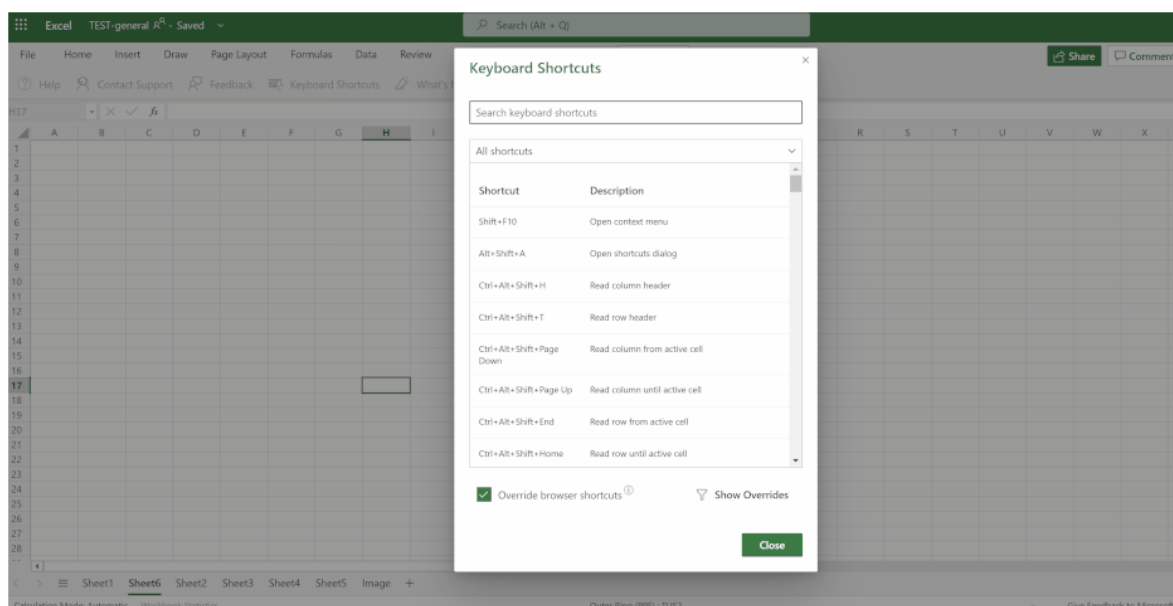
- **Page Up / Page Down:** move screen up/down
- **ALT + Page Up:** move screen left
- **ALT + Page Down:** move screen right
- **ALT + SHIFT + Page Up:** extend selection left one screen
- **ALT + SHIFT + Page Down:** extend selection right one screen.

Find the full list of keyboard shortcuts via **Help -> Keyboard Shortcuts.**

Beat the Boredom Challenge

*With many of us currently "working from home" / quarantined, there are only so Zoom / Teams calls and virtual parties you can make before you reach your (data) limit. Perhaps they should measure data allowance in blood pressure millimetres of mercury (mmHg). To try and keep our readers engaged, we will continue to reproduce some of our popular **Final Friday Fix** challenges from yesteryear in this and upcoming newsletters. One suggested solution may be found later in this newsletter. Here's this month's…*

Most models are not constructed on a daily basis. Data and calculations are consolidated on a weekly, monthly, quarterly or annual basis. Sometimes this causes problems – such as in the following example.

One such issue is when data straddles two or more consolidation periods, and this is something we have looked at before (*e.g.* Revising Forecasts). It can get worse though: what if we have two "straddles" are in the same

period, or some of the data relates to dates which are not to be included in the analysis? That's what this newsletter's challenge is all about.

Imagine you work in an education establishment, seeking to model forecast data for the calendar year 2021. There are four terms relevant to your modelling period:

Assumed Term Dates

Term	Start	End	Days in Year	Check
1	30 Nov 20	6 Mar 21	65	☑
2	27 May 21	17 Jun 21	22	☑
3	1 Sep 21	4 Oct 21	34	☑
4	17 Oct 21	7 Aug 22	76	☑
			197	☑

All we need to do is allocate the number of term days (including weekends and public holidays) to each month of 2021, *i.e.*

		Jan 21	Feb 21	Mar 21	Apr 21	May 21	Jun 21	Jul 21	Aug 21	Sep 21	Oct 21	Nov 21	Dec 21
Term Dates in Each Month													
Number of Days	197	31	28	6	-	5	17	-	-	30	19	30	31

The challenge is "simply" this: can you construct a calculation such that the correct number of days will be allocated to each month? You may assume the terms will be in chronological order, they will not overlap, there will never be more than two terms associated with any given month, and there will never be two start dates or two end dates in the same month (these assumptions are why there are checks, although the later requirements, admittedly, is not tested in our file or Monday's solution).

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

Charts and Dashboards

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

After a chart is created, if you click on a chart, Chart Tools contextual tabs will appear on the Ribbon. Here is where you may customise your chart. The tab has two sub-tabs, Design and Format. Let's break them down.
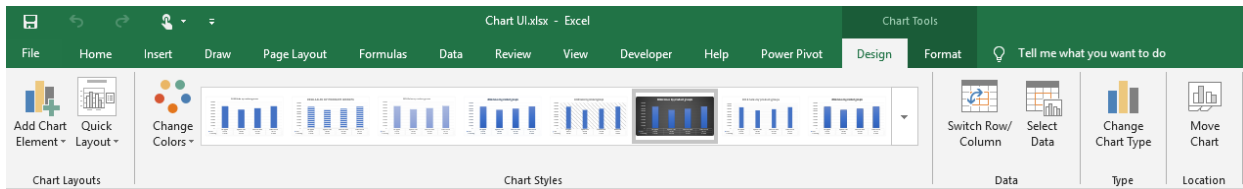
On the far-most, left-hand side is the 'Chart Layouts' category, where you may quickly fix the element and layout of the chart.
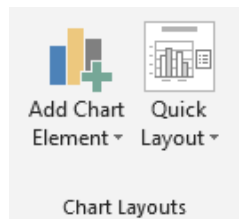
You can add a variety of element such as Axes, Axis Titles, Chart Titles, *etc.* in the 'Add Chart Element' section:
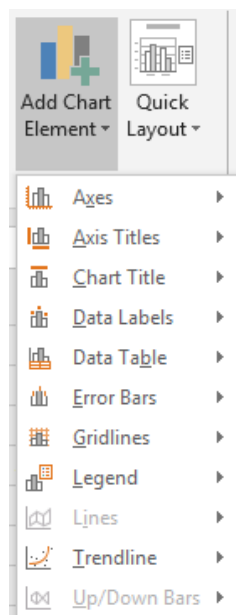
Also, you may choose how your chart and its element(s) will look like using the 'Quick Layout' option:
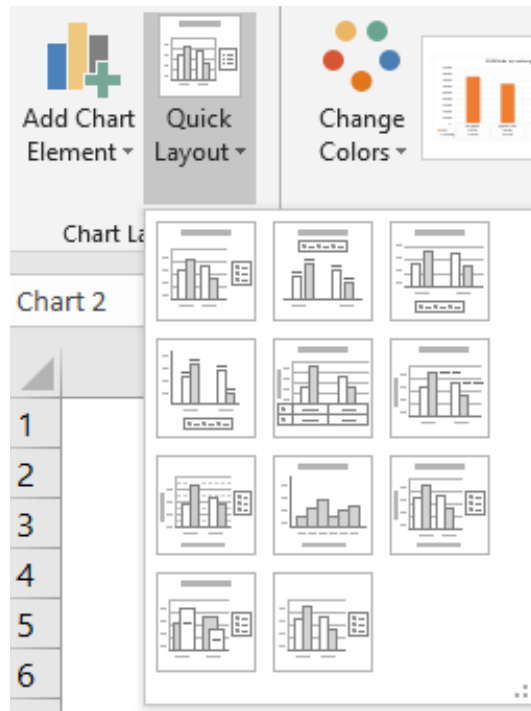
The 'Chart Styles' category is where you can change the chart color *(sic)* palette to your liking or report format requirements, <u>and also</u> there are different chart designs to choose from, rather than design your chart manually:
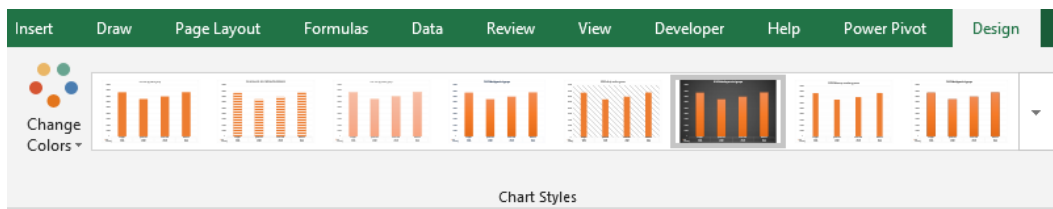
Then in the 'Data' section, you may choose to 'Switch Row/Column' in your data (just like how you do with transpose), then the chart layout will also be changed.
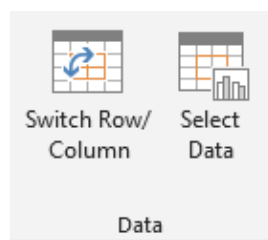
For example, note what has been changed in the chart when 'Switch Row/Column' is clicked:
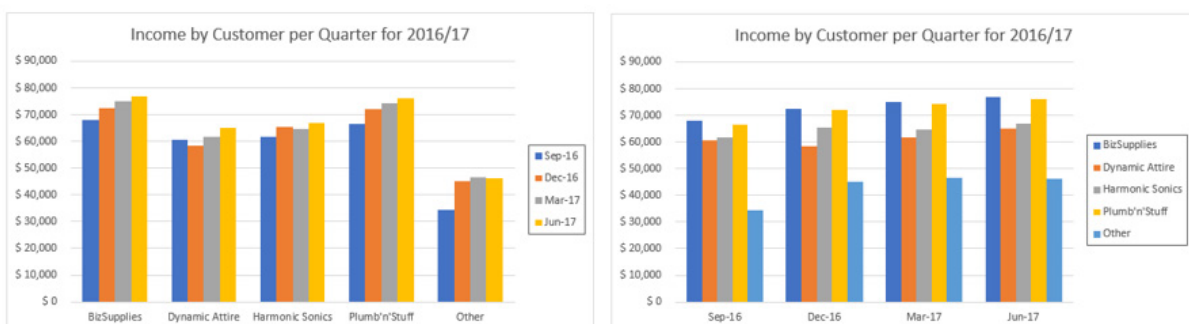
Rather than starting from scratch, choosing the entire data to insert the chart, you just need to click on 'Change Chart Type' section:
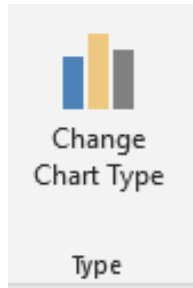
A dialog will pop out, where you can choose to change your initial chart to another chart type, or also make it a 'Combo' chart (as the last item on the left-hand list), to choose which series to be displayed as which type (Column, Line, *etc*.).
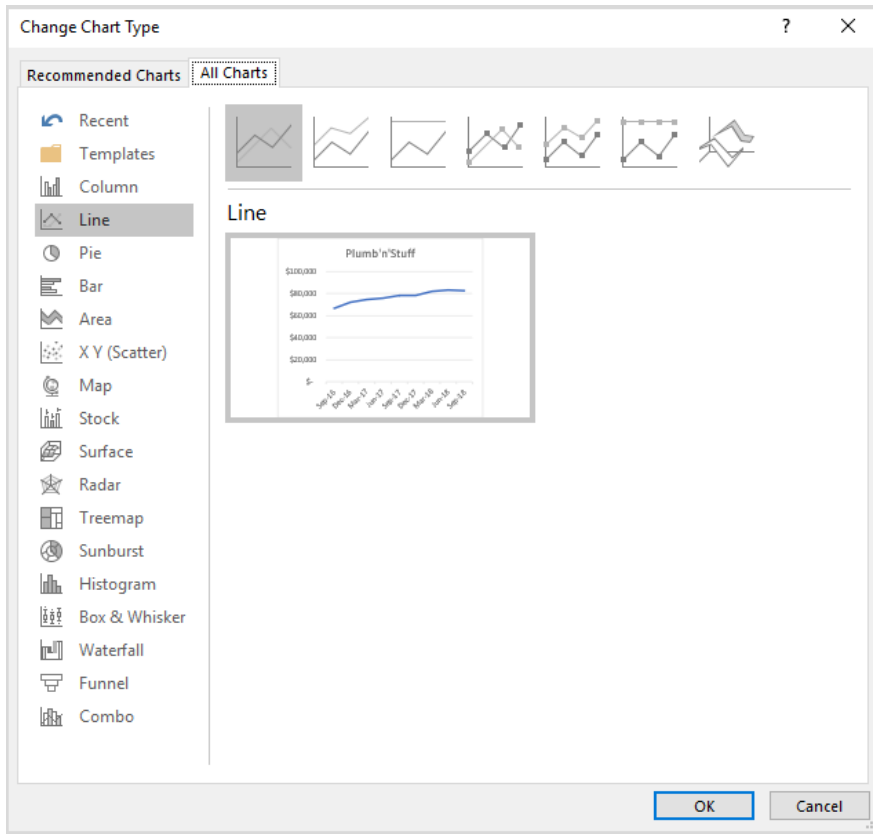
The last section on the 'Design' sub-tab is the 'Move Chart' category, where you can move the chart to another sheet or to a created dashboard:
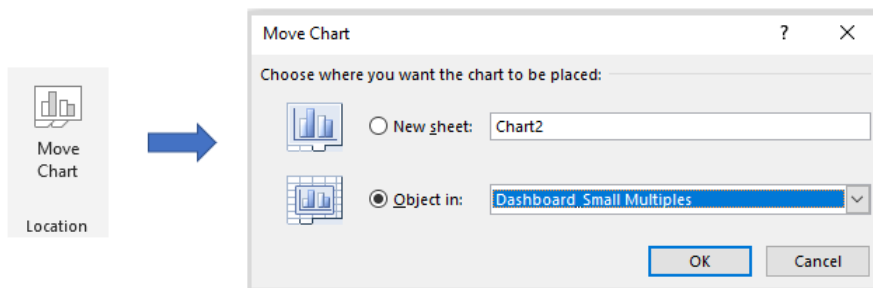
Now, let's move to the next sub-tab: Format, which helps manually design charts.
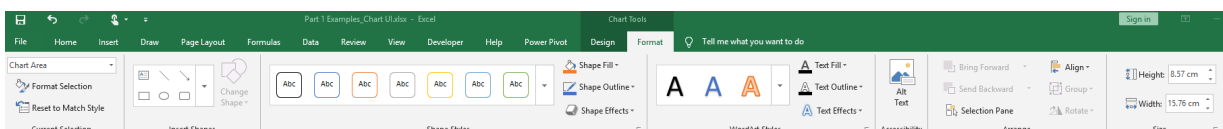
Let's break it down into smaller categories.

'Current Selection' says which part you are currently choosing to format. You may 'Reset to Match Style' which brings us back to the original one:
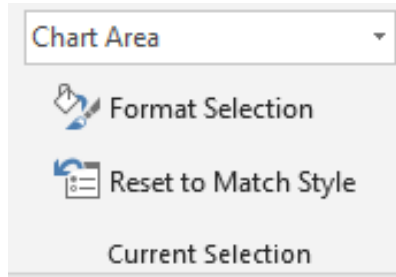
Next to it is the 'Insert Shapes', from which you may insert multiple shapes into the chart:
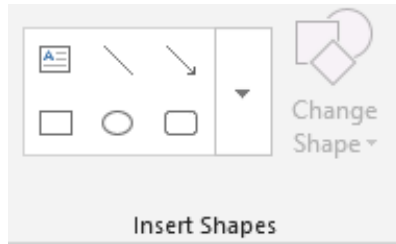
From the 'Shape Style' category, shape outlines, fills and other effect may be formatted:
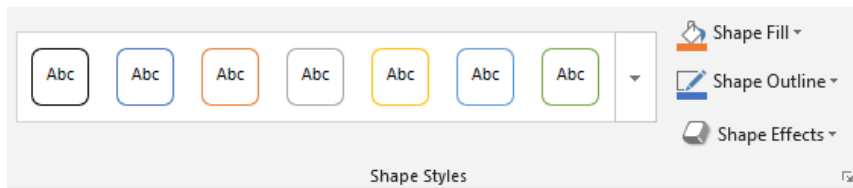
You can even add 'WordArt':
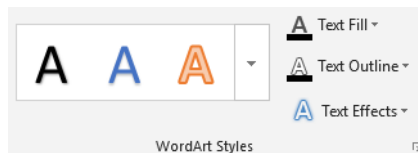
In the 'Accessibility' category, if you click 'Alt Text':
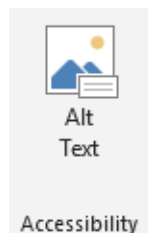
a dialog will appear like this:
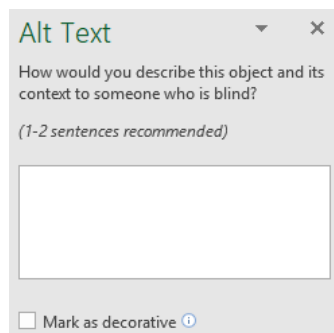
In the 'Arrange' category, the position of the chart may be adjusted:
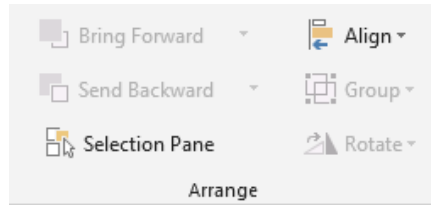
Last but not least is the 'Size' category. This is very helpful when designing a dashboard, where you may need to make charts identical in some size dimensions, by just setting the height and width to the same numbers:
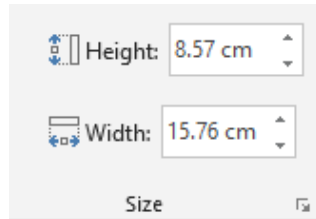
In addition to the contextual 'Chart Tools' on the Ribbon, charts may also be quickly formatted using their side-menus. If you click on a chart, you will see on the right there three option boxes:
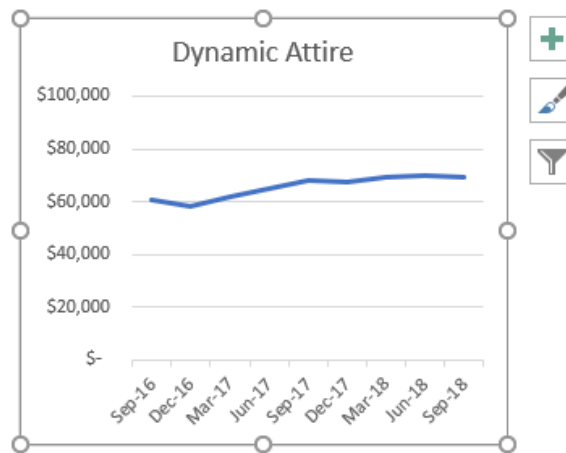
If you click the plus shaped button, there is a list of tick boxes where you can choose which chart elements *(see last month's newsletter)* should be added to your chart. You can also click on the arrows for more detailed options:
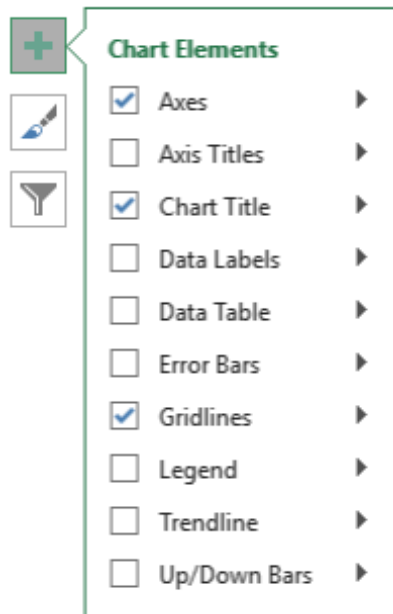
The second icon is the brush pen box, which is similar to the Design tab, where you may quickly choose your chart styles and colours:
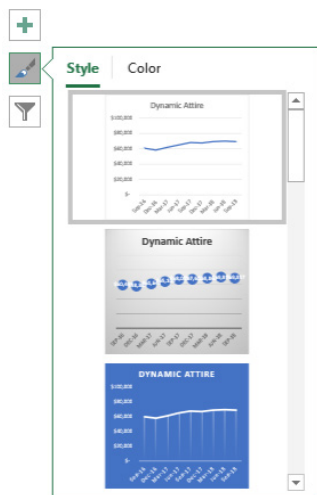
Finally, the Filter panel may be used to sort out which data you want to display on the chart:
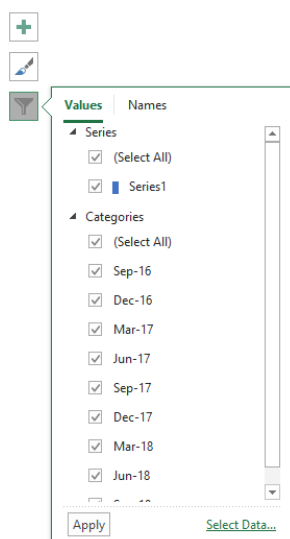
A right click on the chart will trigger a menu list, which offers most of the options from the contextual 'Chart Tools' tabs:
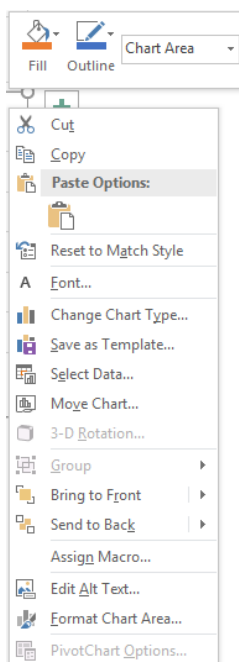
We'll continue next month…

Visual Basics

We thought we'd run an elementary series going through the rudiments of Visual Basic for Applications (VBA) as a springboard for newer users. This month, we look at declaring constants.

It is highly recommended that values that are passed into formulae, subroutines and functions are variables. However, what if the value of that variable does not change? There is a way to make it immutable by declaring it as a constant. This means that it cannot be changed during execution.

By declaring constants, they can be easy to access without hardcoding the information more than once and make code much more readable and facilitate error checking.

Constants are declared in a similar way to 'Dim' except using the keyword 'Const' – except a constant should always be assigned a value when it is declared.

```
(General)

Const GST_Rate = 10
```

If somewhere in the code, it tries to reassign the value, the VBA Compiler spits an error message before execution.

```
(General)
Option Explicit

Const GST Rate = 10

Sub TestConst()
    GST Rate   20
End Sub
```

Microsoft Visual Basic for Applications ✕

⚠ Compile error:

Assignment to constant not permitted

[OK]   [Help]
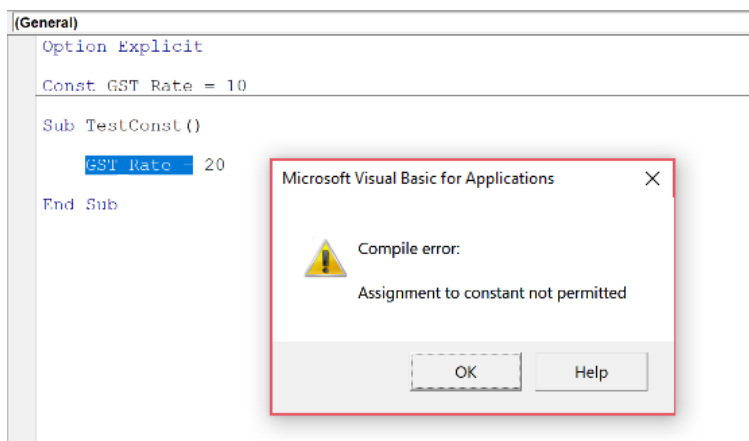
The compiler determines what the data type is from the type of expression of the constant. However, it is always good form to explicitly state a data type. This is particularly important if you plan on doing arithmetic expressions in your code that results in a Long value and having an Integer constant will cause overflow errors. Using the 'As' statement like used in 'Dim' the data type of the constant can be cast.

```
(General)
Option Explicit

Const TheAnswerToEverything As Integer = 42
```

Multiple constants can be declared on a single line as well:

```
(General)
Option Explicit

Const RankOfHumans As Long = 3, RankOfDolphins As Integer = 2, RankOfMice As String = "One"
```

Now knowing how constants are declared, like variables, the scope of the constants are also put under consideration. This is declared the same way as variables as discussed previously.

More next time.

Power Pivot Principles

We continue our series on the Excel COM add-in, Power Pivot. Having considered Calendar tables last month, this time we introduce our first time intelligence function.

DATESYTD is a time intelligence function that, like all time series functions, requires contiguous dates in order to work properly (although some are becoming "forgiving"). The **DATESYTD** function returns values in a PivotTable that contain aggregated data for the year to date. The syntax of this function is:

DATESYD(dates[, year_end_date])

It's commonly used with our old friend, the **CALCULATE** function. To demonstrate, let's create a measure using the **DATESYTD** function to show year to date sales:
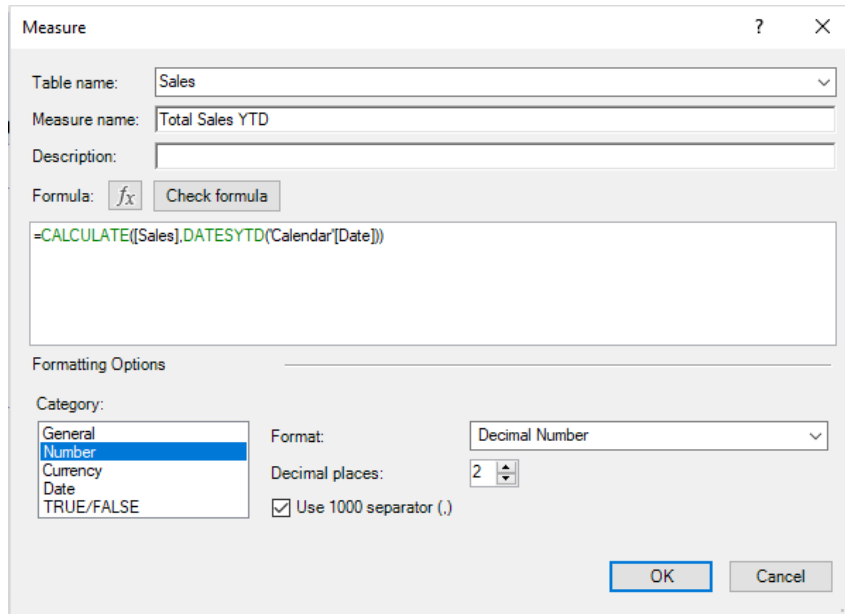
The next step is to create our PivotTable report.
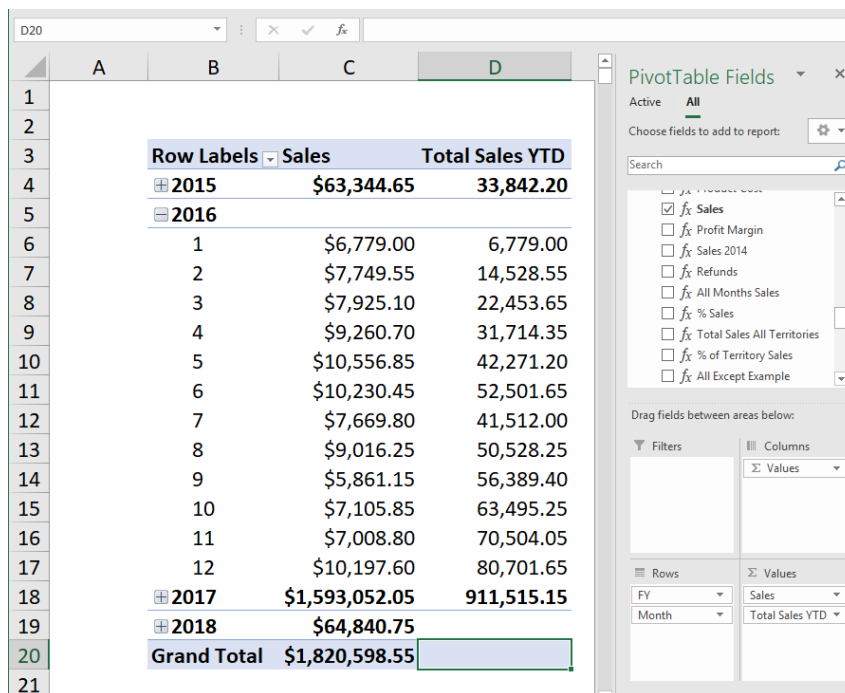
We can now see the total sales for the month and the total sales YTD (year to date) throughout the year. Notice that the year and month are both in the same column (column **B**) of the PivotTable. For many functions, this is necessary for the time series calculation to work correctly. If years were put in row 3 instead with months down column **B**, many formulae would not work as intended – hence the need for what is known as *contiguity*. But that's a tale for another day...

More *Power Pivot Principles* next month.

Power Query Pointers

Each month we'll reproduce one of our articles on Power Query (Excel 2010 and 2013) / Get & Transform (Office 365, Excel 2016 and 2019) from www.sumproduct.com/blog. If you wish to read more in the meantime, simply check out our Blog section each Wednesday. This month, we look at exporting queries to other workbooks in Power Query or Get and Transform.

Regular readers will know that this series now uses later versions of Excel, and therefore uses the term 'Get and Transform' for 'Power Query' regularly. In most cases, it's easy to follow in earlier versions of Excel which have a Power Query tab instead, but there are some improvements to 'Get and Transform' that may tempt you into an upgrade. This is one of them. There are however other ways to share queries if this particular upgrade is not available, as described later.

The functionality will only be accessible if, instead of a 'Queries' pane that appears on the right hand side of the workbook, there is a 'Queries and Connections' pane, which looks like this:
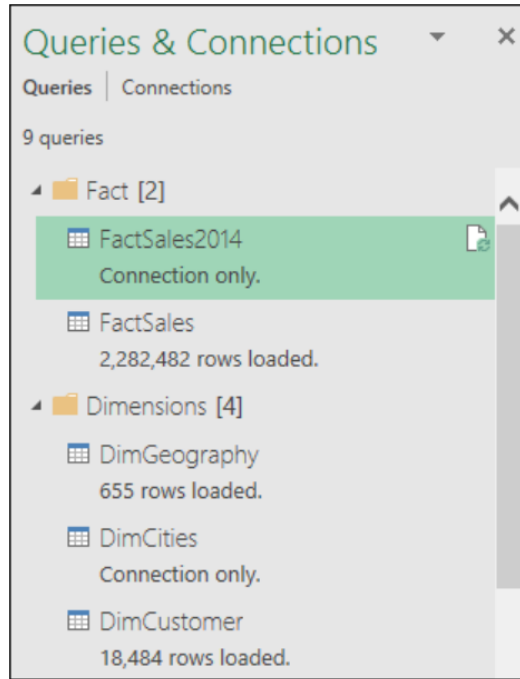
From the 'Queries and Connections' pane, you may right-click on any query and export it.
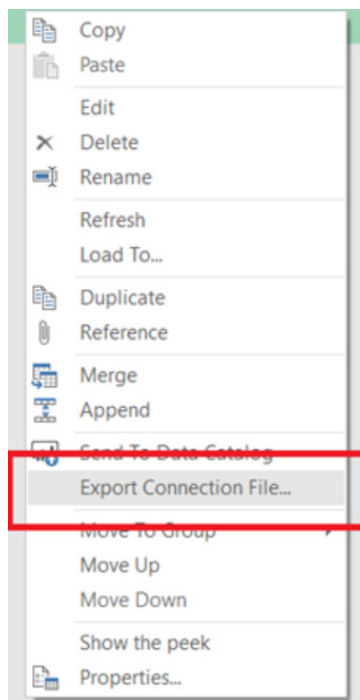
Once the query has been exported to an ODC (Office Database Connection) file, it can be sent to colleagues. If you have an ODC file on my computer, you may use it by viewing it from the 'Existing Connections' screen on the Data tab.
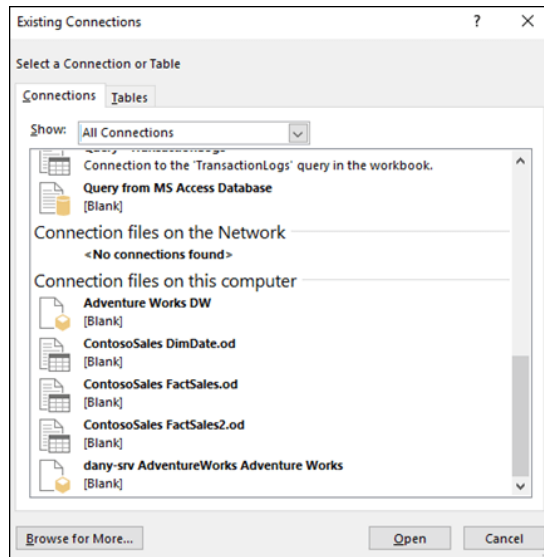
But what if you don't have the latest Office 365 update, which is where this appears to be? There are some other ways to share queries.

We have a query shown below, which will be useful in another workbook. For this example, we've chosen to use Excel 2013. First, we will copy the query and paste it into the new workbook. Note that we have selected an independent query for this example.
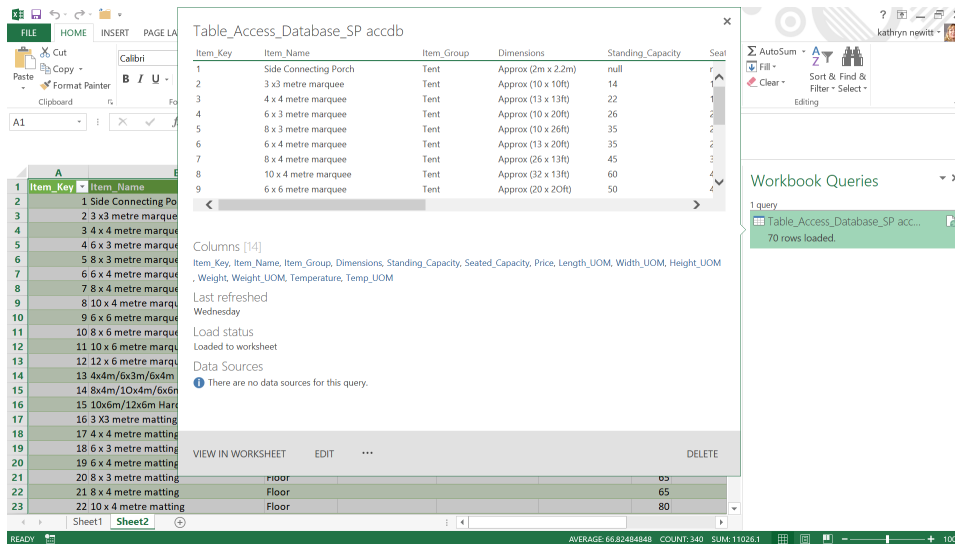
The first step is to go to the Query Editor and select the 'Advanced Editor'.
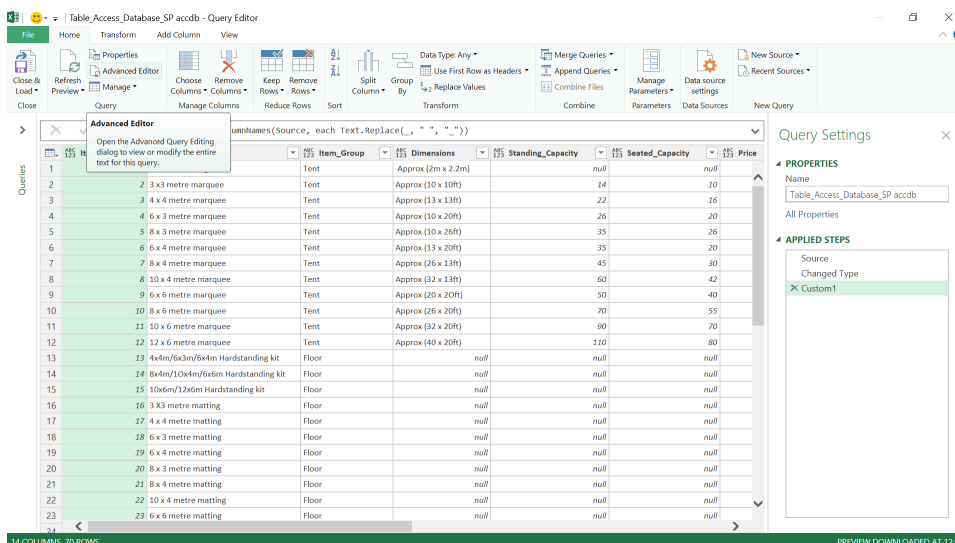
We need to copy the **M** language from the 'Advanced Editor':

In a new workbook, create a new 'Blank Query':
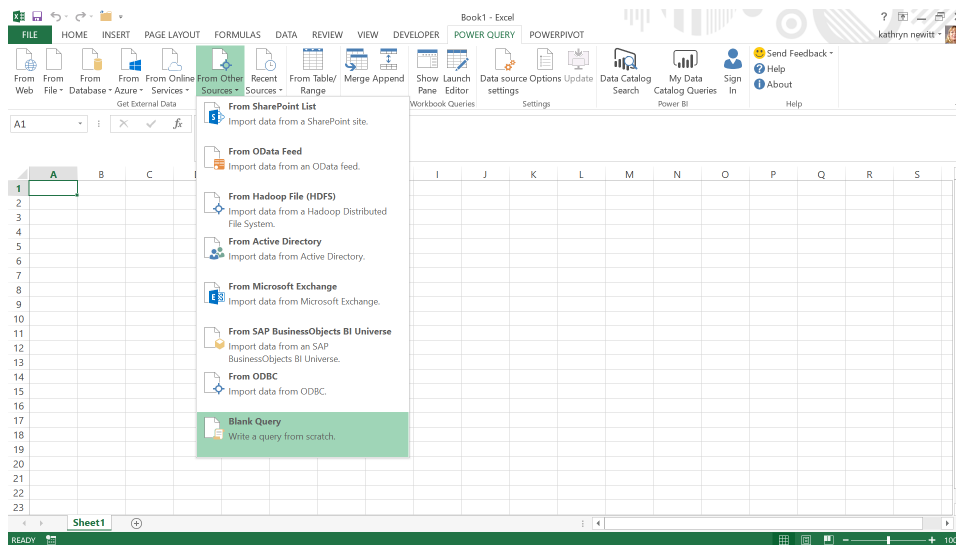
In the new 'Blank Query', access the 'Advanced Editor' and paste in the **M** code:

We can then save the query and use it in the workbook. Note that if we had chosen a query that was dependent on other queries, we would have to copy them all across, and we would choose to do this in the order in which they were created.

There is another way to share queries. If you right-click another query, you will notice you have a 'Copy' action available. Here, we are using Excel 2016 Professional Plus for this example, but there are versions of Excel 2013 with this option too.
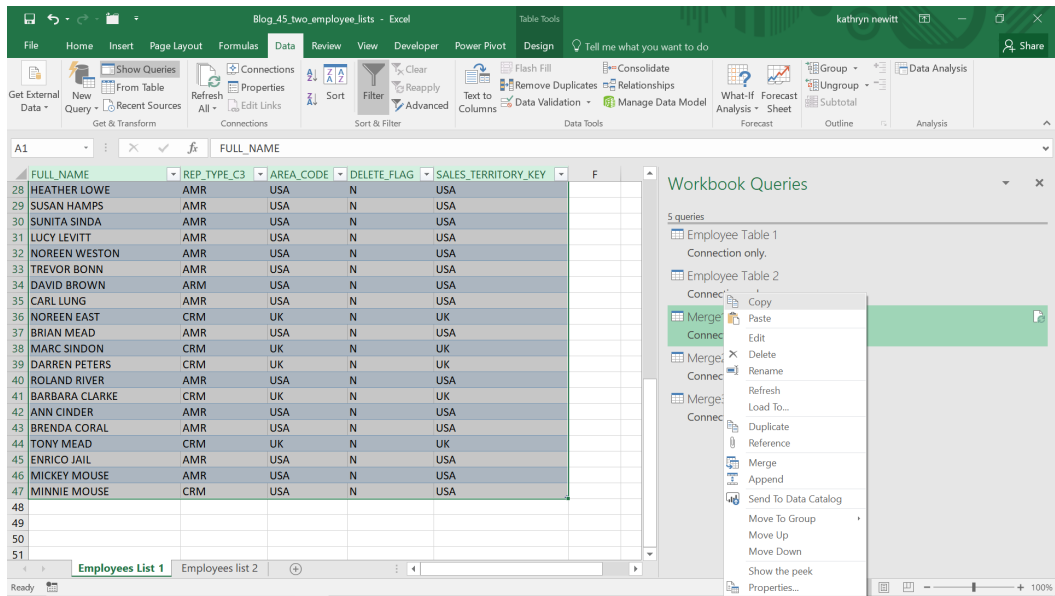
Copy the query and then paste it into the 'Query Pane' of a blank workbook.
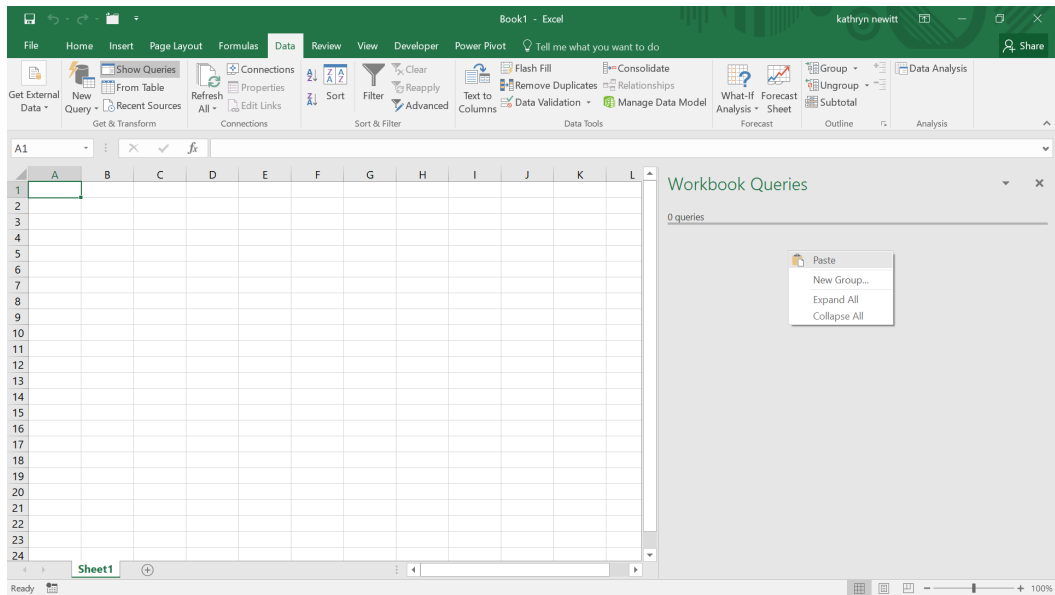
The query appears in the blank workbook.
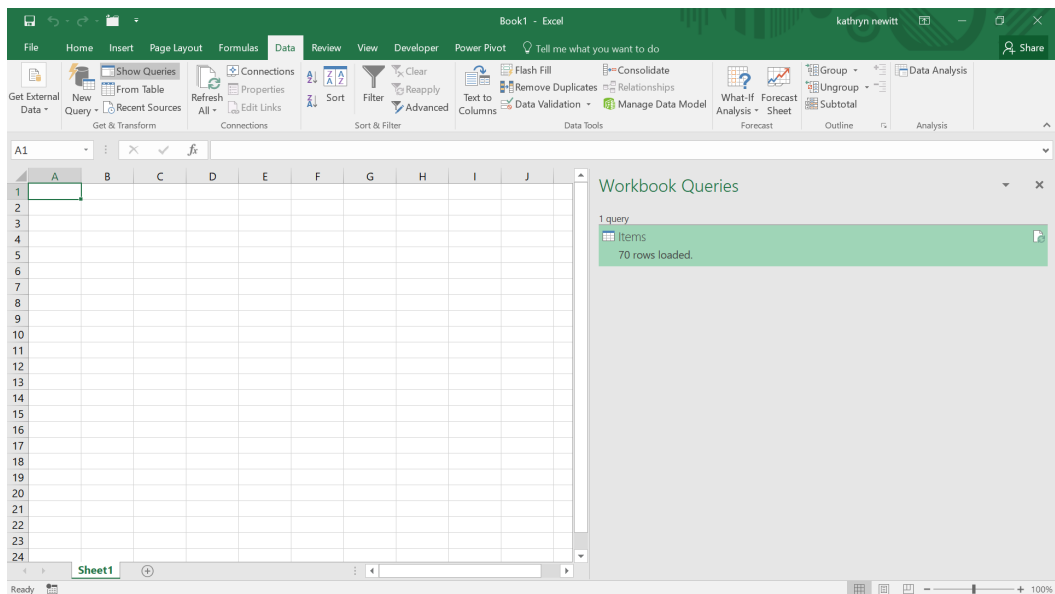
Since this method creates a query, the checks are more stringent from the start: if the query refers to a table in the original workbook, then the query will have a warning icon next to it. The copy and paste method is simpler to use as a 'template' as a basis for a new query, as this method may copy more than one query – but there is a good reason for that. Let's choose a query that is not independent:
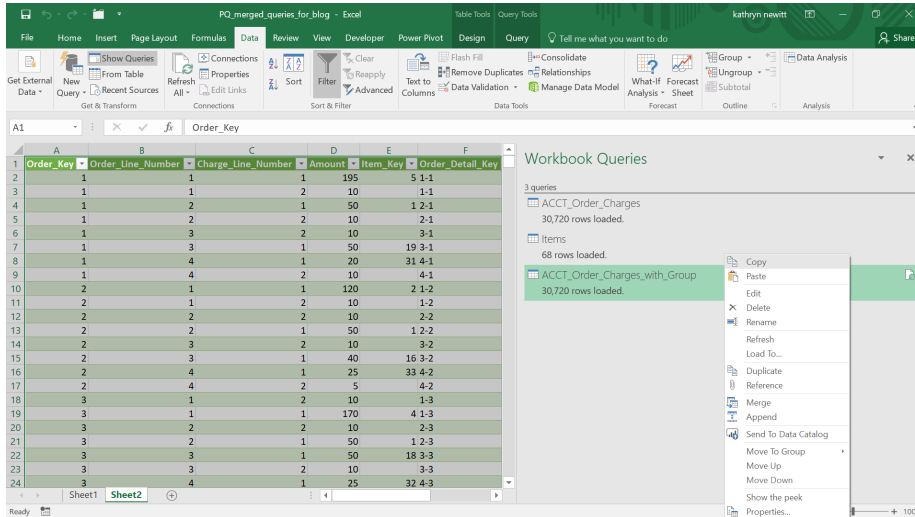
Then, paste this into a new empty workbook:
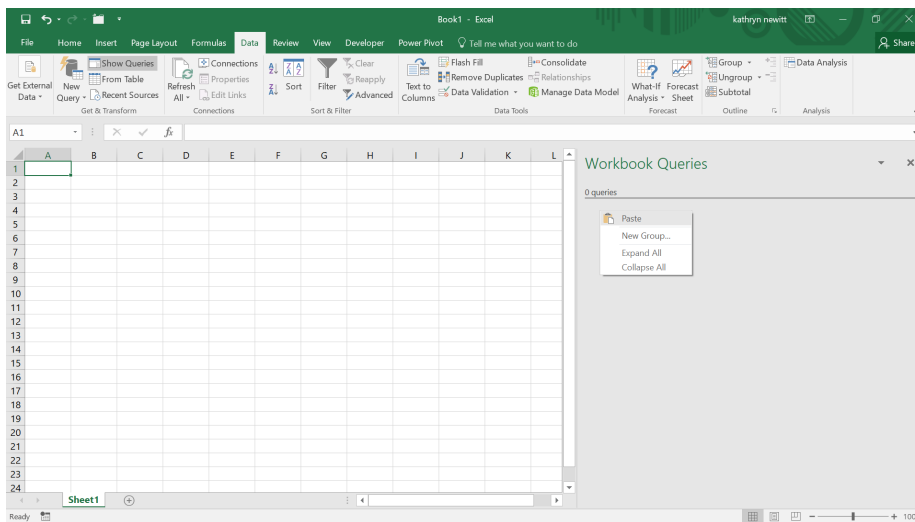
You don't just get the query you have copied:
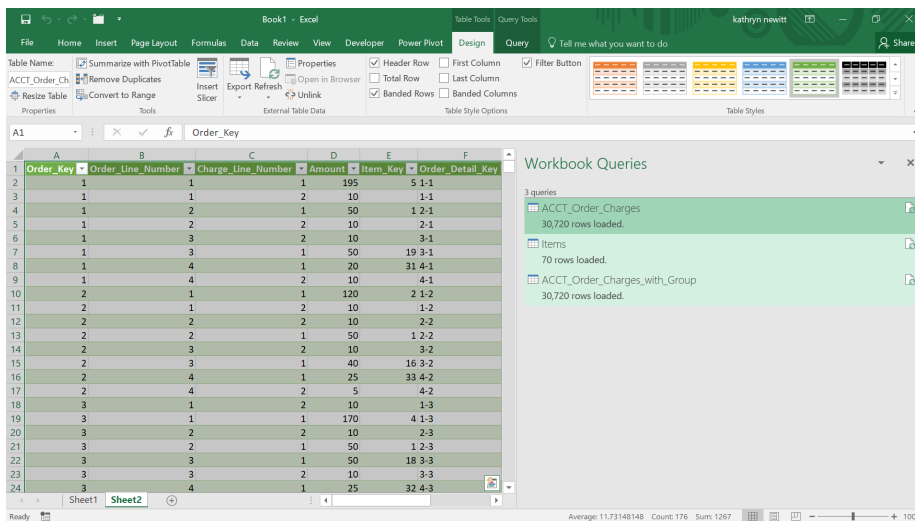
You obtain all your referenced queries too, which is much more efficient for dependent queries that are needed. Therefore, for those of us that don't have the ability to export ODC files (yet!), these can be useful methods of sharing popular queries.

More next month!

Power BI Updates

This month's updates barely made the cut-off for this month's newsletter – so these are hot off the press. They include DirectQuery support for Dataflows, which has now become Generally Available, there is a Preview of the paginated reports visual, and there are further updates for small multiples and DirectQuery for Power BI datasets, amongst others.

The full list reads as follows:

Reporting

- Paginated reports visual now in Preview
- Area chart transparency sliders
- Inner padding for continuous axes
- Small multiples: responsiveness and conditional formatting in Preview

Analytics

- Q&A improvement for inferred results

Modelling

- Format strings now persisted when using DirectQuery for Power BI datasets and Azure Analysis Services in Preview Available

Data Preparation

- DirectQuery support for Dataflows now Generally Available
- Select all operation is now supported for Dynamic M Query Parameters in Preview

Data Connectivity

- Assemble Views (new connector)
- BQE Core (new connector)
- SumTotal (new connector)
- Adobe Analytics (updated connector)
- Anaplan (updated connector)
- Azure Databricks (updated connector)
- Cognite Data Fusion (updated connector)
- Dynamics 365 Business Central (updated connector)
- FactSet Analytics (updated connector)
- Google BigQuery (updated connector)
- Starburst Enterprise (updated connector)
- Vessel Insight (updated connector)
- Workplace Analytics (updated connector)
- Snowflake (updated connector)
- Azure Consumption Insights (connector deprecated)

Service

- Datasets discoverability

- Request access to datasets
- Mandatory label policy for Microsoft Information Protection sensitivity labels in Preview
- Admin API to set and remove Microsoft Information Protection sensitivity labels
- Automate deployments with new APIs and PowerShell samples in Preview
- Manage Dataflows in deployment pipelines in Preview
- Admin APIs for deployment pipelines

Mobile

- A new look for the Power BI Windows app Preview
- Passing URL parameters to paginated reports

Visualisations

- New visuals
- Drill Down Combo Bar PRO by ZoomCharts
- Dumbbell Bar Chart by Nova Silva
- graphomate bubbles 2021.2
- Zebra BI Tables 5.0
- Zebra BI Charts 5.0

Template apps

- Template app one-click update and republish
- Salesforce Analytics for Sales Managers

Other

- Power BI Desktop Installer Changes
- Latest (June) update of Power BI Report Builder
- Report action bar now available in the Power BI SharePoint Online Webpart.

Let's go through each in turn.

Paginated reports visual now in Preview

Microsoft has announced the long-anticipated paginated report visual for Power BI reports is now available in Power BI Desktop as a public Preview. For the first time, this native Power BI visual allows you to render any paginated report you've uploaded to the Service in a Power BI report. Similar to other visuals, you may connect fields from your Power BI dataset to be used as parameter values, thus providing a fully interactive experience with these paginated reports.

Area chart transparency sliders

You can now set the transparency of the coloured areas in your area and stacked area charts. Before, this transparency was defaulted to 60%. Now, you may adjust this transparency in the 'Data colors' *(sic)* card in the Formatting pane.
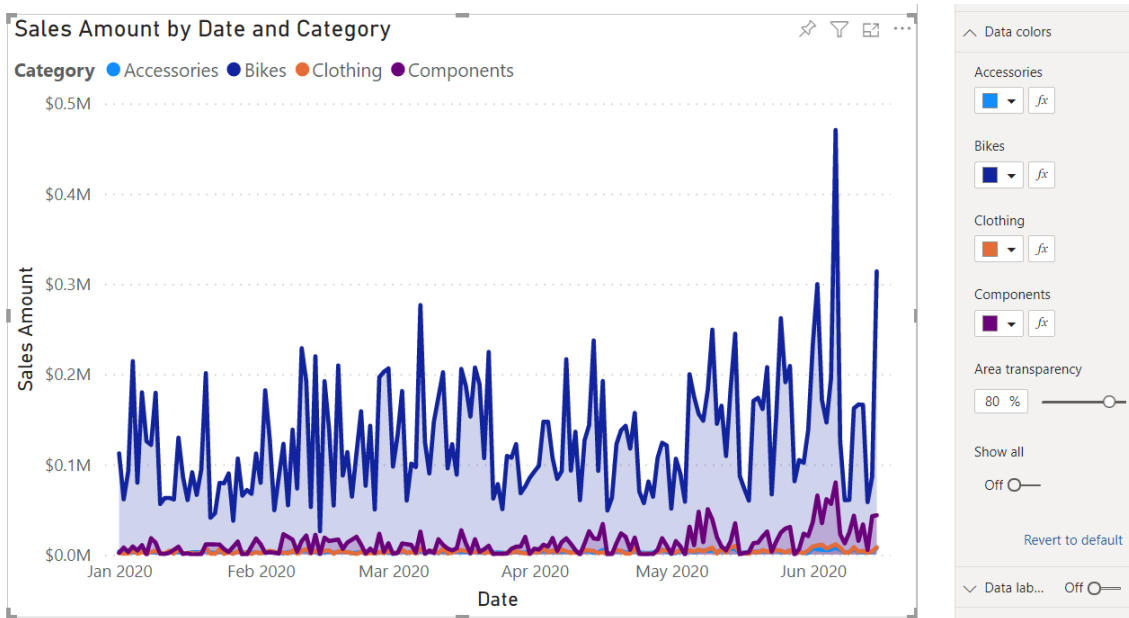
Inner padding for continuous axes

Cartesian charts with categorical axes have an 'inner padding' option in the Formatting pane, which adjusts the size of the padding between category bars, columns and clusters. This helps you determine how thick your columns and bars should be and how much whitespace should be between them. This month, this option has been added to continuous axes as well, allowing you greater control over the look and feel of your charts.
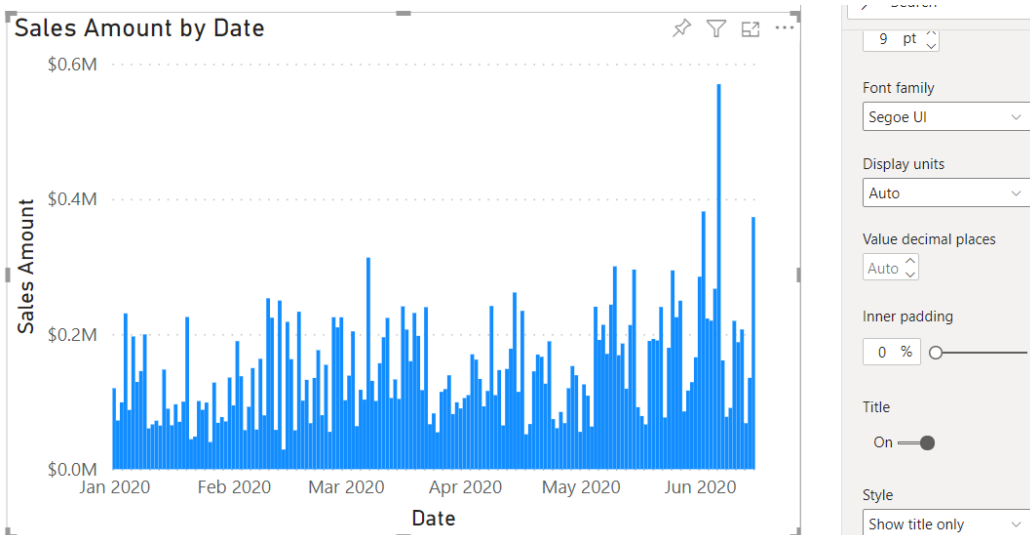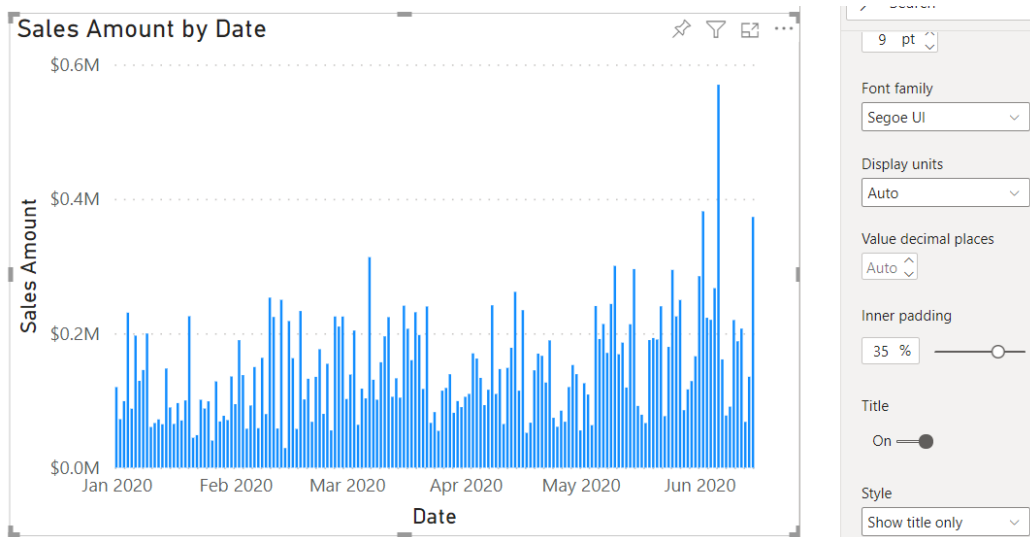
Small multiples: responsiveness and conditional formatting in Preview

There are two new updates to the small multiples Preview feature this month:

1. Support has been enabled for the responsive toggle in the General card of the Formatting pane. Responsive visuals will slowly drop chart elements like axis titles, axes and legend as their size shrinks, providing more room to the plot area. For small multiples, the responsive breakpoints have been adjusted to accommodate for the potential presence of more than one plot area in the visual. This means that small multiples visuals will generally begin shrinking padding and dropping chart elements earlier than non-small multiples visuals to make room for the multiple plot areas. Although responsiveness is on by default, responsive behaviour will generally not affect visuals large enough to provide a clear data visualisation.

2. Conditional formatting has been added to small multiple title and background colours. Click the **fx** button next to the corresponding options in the Formatting pane to launch the 'Conditional formatting' dialog, where you may set rules by which the chart elements are coloured. With this functionality, your small multiple titles and backgrounds can assist in communicating key aspects of your data.
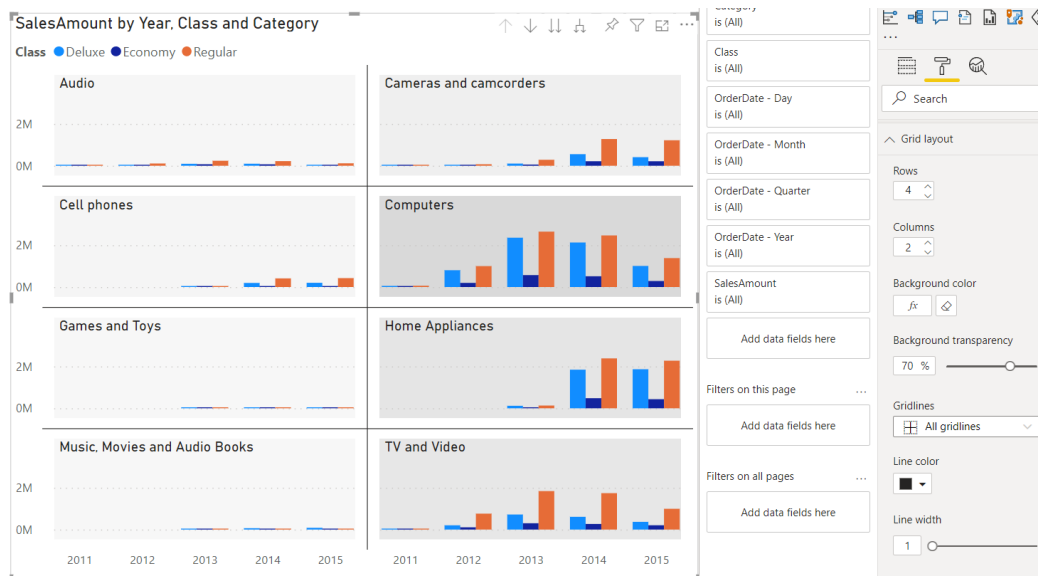
Q&A improvement for inferred results

When your end-users' questions are ambiguous or incomplete, Q&A will need to make some assumptions to produce a visual for these kinds of questions. For these scenarios, Q&A would return a visual and natural language restatement (shown under the **(i)** icon on the right), based upon Power BI's interpretation of the end-user's question. However, it was not evident to users which parts of the result were inferred, if any. In this release, this experience has been modified, by adding a 'Showing results for' restatement for these cases, with explicit emboldening for the inferences made by Q&A:
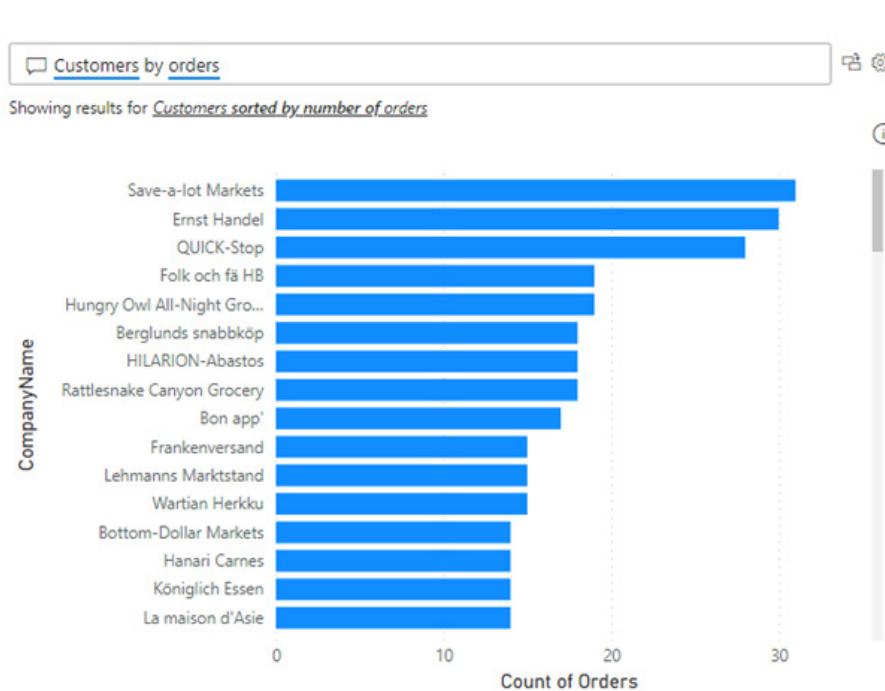
For example, for ambiguous questions, Q&A assumptions could be based upon the model or use of certain syntax, such as implicit aggregations or other derived entities. In the example question below, we see that the question "top products" is too vague, but Q&A made multiple assumptions based off the model and returned the result for *Top 10 products **by product unit price***.
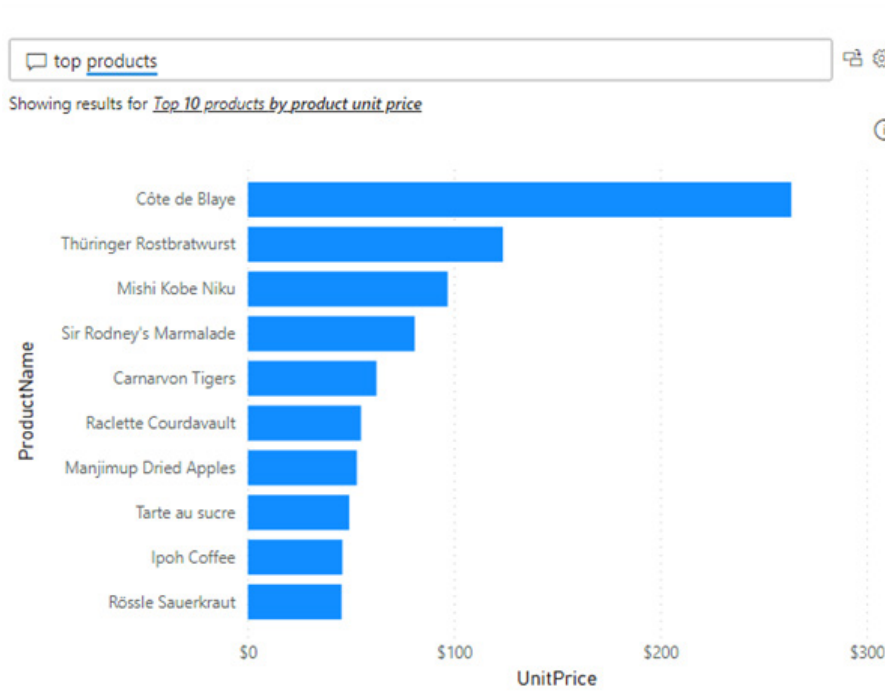
In the 'Showing results for' restatement, "10" and "by product unit price" are shown in bold because Q&A made these assumptions to return a useful result. The user can then click on the restatement for the visual to automatically update the question and tweak the question further.

Format strings now persisted when using DirectQuery for Power BI datasets and Azure Analysis Services in Preview

Microsoft has noted that it's useful to have properties set on a remote model you've connected to flow through to your local model. In this release, format strings are the latest property that will flow from your remote source to your local one, as a consequence.

DirectQuery support for Dataflows now Generally Available

DirectQuery support for Power BI dataflows is now Generally Available. You can now connect directly to a dataflow without needing to import the data into a dataset. This can be useful in a number of scenarios, including:

- working with big dataflows
- decreasing orchestration needs
- serving data to customers in a managed way
- not duplicating data between a dataflow and a dataset.

To use this capability, you will need to explicitly toggle the enhanced compute engine to 'On' in dataflow settings and then refresh the dataflow before it can be consumed in DirectQuery mode.

Microsoft has stated that they will continue to make optimisations to the underlying connector to support import scenarios as well, such as query folding support for dataset refreshes on top of dataflows. They also plan to make connecting to dataflows easier by bringing a unified Power Platform and Power BI Dataflows connector, and bring a Dataflows connector to Excel.

Select all operation is now supported for Dynamic M Query Parameters in Preview

Back in October last year, Microsoft released a Preview of Dynamic **M** Query Parameters that allowed report viewers to use filters or slicers to set the value(s) for an **M** Query Parameter, which can be especially useful for query performance optimisations. Previously, Dynamic M Query Parameters did not support the 'Select all' operation, requiring end-users to individually select each value if they wanted to see the data for all values. Now, in this release, support has finally been added for the 'Select all' operation, so that with a single-click users may select all values for the query parameter.

Confirmed Cases Analysis by Country
To analyze cases over time **Select all** countries or select a specific country to analyze. If no countries are selected, the default value is United States.

Confirmed Cases by Date

Confirmed Cases trended up, resulting in a 470.38% increase between Monday, April 6, 2020 and Friday, June 5, 2020.

Country
- Select all
- Afghanistan
- Albania
- Algeria
- American Samoa
- Andorra
- Angola
- Anguilla
- Antigua and Barbuda
- Argentina
- Armenia
- Aruba
- Australia
- Austria
- Azerbaijan
- Bahamas
- Bahrain
- Bangladesh
- Barbados
- Belarus
- Belgium
- Belize
- Benin
- Bermuda
- Bhutan
- Bolivia
- Bonaire
- Bosnia and Herzegovina
- Botswana
- Brazil
- British Virgin Islands
- Brunei

As a prerequisite, you'll need to enable and setup Dynamic **M** Query Parameters. Once you have the feature enabled and setup a Dynamic **M** Query Parameter, you may now begin the setup process to support 'Select all'.

For example, within the Model tab of Power BI Desktop, imagine you have a field called Country (list of countries) that is bound to an **M** parameter called **countryNameMParameter**:
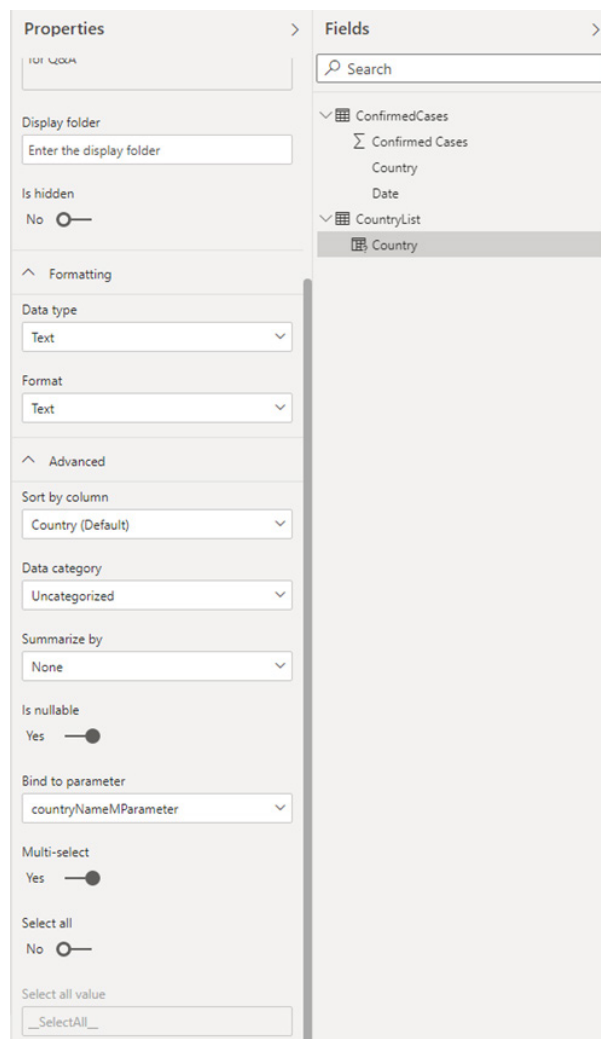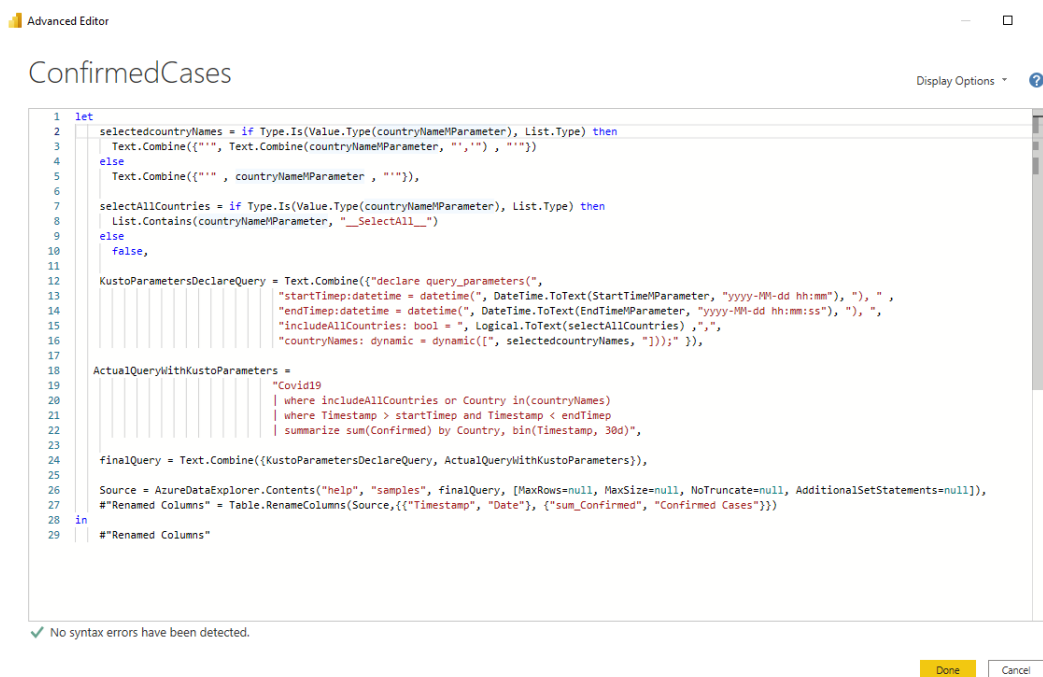
You'll also notice that this parameter is enabled for Multi-select but not enabled for 'Select all'. When you enable the 'Select all' toggle, you will then see an enabled input called 'Select all value':

The Select all value (which defaults to '__SelectAll__') is used to refer to the 'Select all' option in the **M** Query. This value is passed to parameter as a list that contains the value you defined for 'Select all'. Therefore, when you are defining this value or using the default value, you will need to make sure that this value is unique and does not exist in the field bound to the parameter.

Once you have set the value or used the default value for 'Select all', you will then need to update the **M** Query to account for this 'Select all' value.

To edit the **M** Query, you will need first launch the Power Query Editor and then select the Advanced Editor in the Query section:

In the Advanced Editor, you will need to add a Boolean expression that will evaluate to TRUE if the parameter is enabled for Multi-select and contains the 'Select all' value (else return FALSE). For our example, that would look like this:

```
selectAllCountries = if Type.Is(Value.Type(countryNameMParameter), List.Type) then
    List.Contains(countryNameMParameter, "__SelectAll__")
```

Next, you will need to incorporate the result of this 'Select all' Boolean expression into the source query. For our example, there is a Boolean query parameter in the source query called 'includeAllCountries' that is set to the result of the Boolean expression from the previous step. You may then use this parameter in a filter clause in the query, such that FALSE for the Boolean will filter to the selected country name(s) and TRUE would effectively apply no filter:

```
KustoParametersDeclareQuery = Text.Combine({"declare query_parameters(",
        "startTimep:datetime = datetime(", DateTime.ToText(StartTimeMParameter, "yyyy-MM-dd hh:mm"), "), " ,
        "endTimep:datetime = datetime(", DateTime.ToText(EndTimeMParameter, "yyyy-MM-dd hh:mm:ss"), "), ",
        "includeAllCountries: bool = ", Logical.ToText(selectAllCountries) ,",",
        "countryNames: dynamic = dynamic([", selectedcountryNames, "]));" }),

ActualQueryWithKustoParameters =
        "Covid19
        | where includeAllCountries or Country in(countryNames)
        | where Timestamp > startTimep and Timestamp < endTimep
        | summarize sum(Confirmed) by Country, bin(Timestamp, 30d)",

finalQuery = Text.Combine({KustoParametersDeclareQuery, ActualQueryWithKustoParameters}),
```

For reference, here is the full query used in this example:

let

selectedcountryNames = if Type.Is(Value.Type(countryNameMParameter), List.Type) then
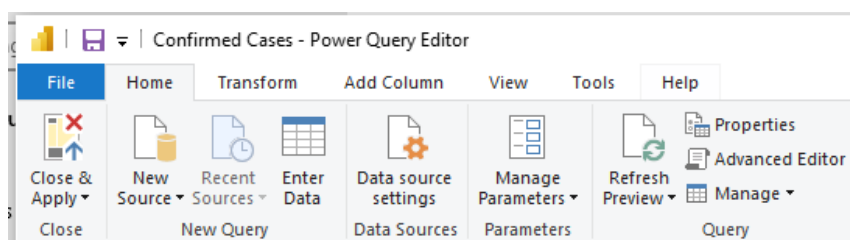
Text.Combine({"'", Text.Combine(countryNameMParameter, "','") , "'"})

else

Text.Combine({"'" , countryNameMParameter , "'"}),

selectAllCountries = if Type.Is(Value.Type(countryNameMParameter), List.Type) then

List.Contains(countryNameMParameter, "__SelectAll__")

else

false,

KustoParametersDeclareQuery = Text.Combine({"declare query_parameters(",

"startTimep:datetime = datetime(", DateTime.ToText(StartTimeMParameter, "yyyy-MM-dd hh:mm"), "), " ,

"endTimep:datetime = datetime(", DateTime.ToText(EndTimeMParameter, "yyyy-MM-dd hh:mm:ss"), "), ",

"includeAllCountries: bool = ", Logical.ToText(selectAllCountries) ,",",

"countryNames: dynamic = dynamic([", selectedcountryNames, "]));" }),

ActualQueryWithKustoParameters =

"Covid19

| where includeAllCountries or Country in(countryNames)

| where Timestamp > startTimep and Timestamp < endTimep

| summarize sum(Confirmed) by Country, bin(Timestamp, 30d)",

finalQuery = Text.Combine({KustoParametersDeclareQuery, ActualQueryWithKustoParameters}),

Source = AzureDataExplorer.Contents("help", "samples", finalQuery, [MaxRows=null, MaxSize=null, NoTruncate=null, AdditionalSetStatements=null]),

#"Renamed Columns" = Table.RenameColumns(Source,{{"Timestamp", "Date"}, {"sum_Confirmed", "Confirmed Cases"}})

in

#"Renamed Columns"

Once you have updated your **M** Query to account for this new 'Select all' value, you can now use the 'Select all' function in slicers or filters:
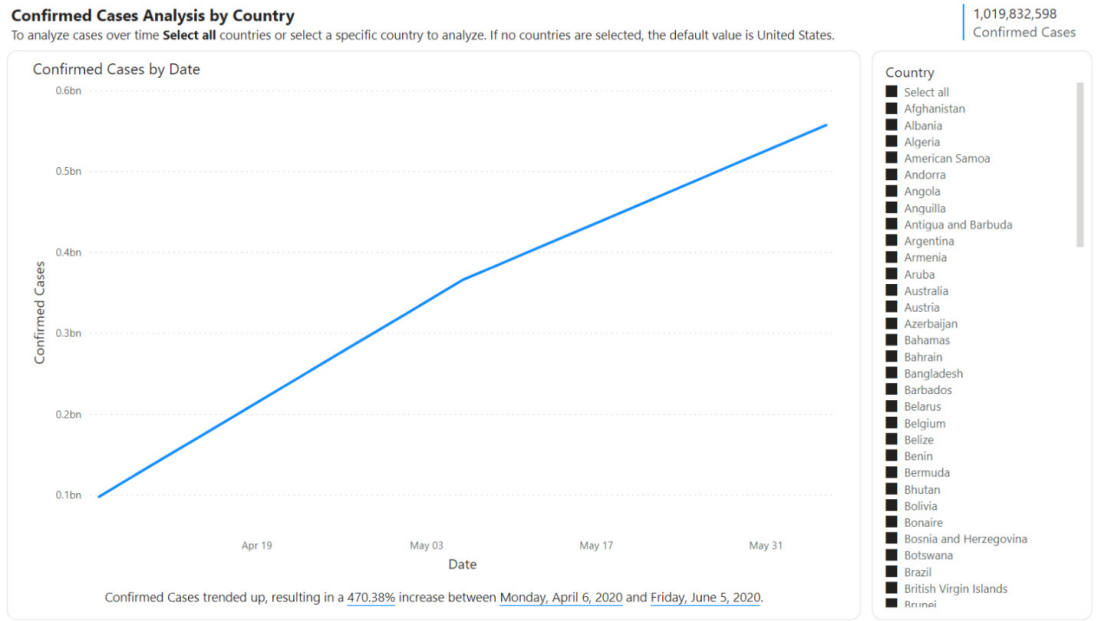
Note that Dynamic **M** Query parameters do not support 'Exclude / Not in' filters. Therefore, selecting a value after 'Select all' has been pressed will not deselect that value and produce an unsupported exclude filter. Instead, this scenario produces an Include / In filter for that selected value.

Assemble Views (new connector)

The Assemble Views connector has been released in Power BI Desktop. This is part of the Assemble Views' Autodesk Construction Solutions product family, which enables users to view and leverage building information models (BIM) in the cloud. Allowing users to condition their models using advanced filtering to aggregate, add and update fields within the model. This conditioning can then be saved as views, relevant to users with various roles on the project.

The Assemble Views Power Query connector allows users to extract this model data in pre-packaged datasets (views). Once imported into Power BI, users can employ these views using the analytics tools available to create complex workflows and reports. Data trends can be quickly identified by including a date parameter that specifies the point in time for which to load the data.

BQE Core (new connector)

The BQE Core connector is another connector now in Power BI Desktop. This connector gives users the ability to import their raw CORE data into the Power BI application, as well as transform it into centralised, visual and interactive insights for their business. While the connector

was previously offered as a manual integration through our training department, this streamlined installation from the Power BI app will help CORE customers create custom dashboards and reports.

SumTotal (new connector)

The SumTotal connector is also released in Power BI Desktop. This connector will help to launch SumTotal platform OData APIs as datasets within the Power BI Desktop application.

Adobe Analytics (updated connector)

There is a new version of the Adobe Analytics connector. The 2.0 connector will use the Adobe Analytics 2.0 API. In addition to using the dimensions and metrics of the 2.0 API, this new connector will also support paging data beyond the 50,000 row limit of the existing connector. In the near future, there will also be support for exposing Segments as Dimensions in the connector.
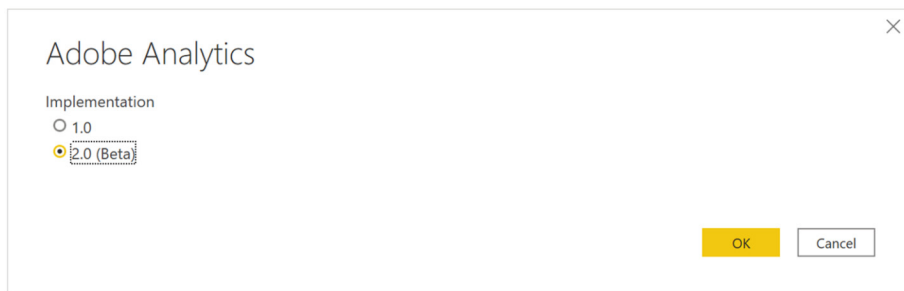
Anaplan (updated connector)

This connector is no longer considered in Beta.

Azure Databricks (updated connector)

This update improves fetch speed for some data layouts.

Cognite Data Fusion (updated connector)

This release contains a bugfix for the **TimeseriesAggregate** function. The list of tags are now escaped using `Uri.EscapeDataString`, which fixes a problem if any of the tags contained any special characters.

Dynamics 365 Business Central (updated connector)

While Business Central already offers tight integration with Power BI, it relies on web services. This new release enables APIs to be used as more performant data sources for Business Central reports hosted in Power BI.

Customers may now create reports and dashboards using modern Business Central APIs, including the built-in APIs v2.0 provided with Business Central and also any custom APIs created by partners or makers. This gives users access to better and faster data analytics in Power BI, while maintaining full backward compatibility.
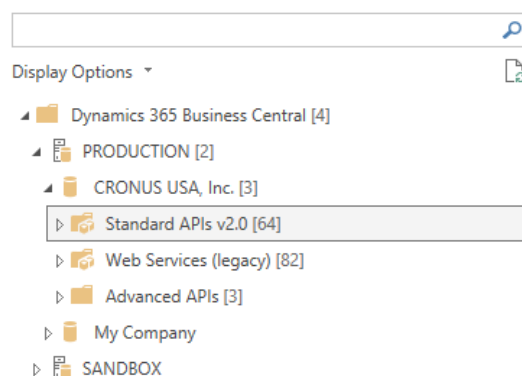
FactSet Analytics (updated connector)

The FactSet Analytics connector has been updated to support caching of calculation results. In case the user doesn't explicitly specify a 'Cached Output Age' value, calculation results are cached for 12 hours by default.

Google BigQuery (updated connector)

There is now more support for Google Cloud Service Accounts, using the Google BigQuery connector.

Starburst Enterprise (updated connector)

This update contains a renaming of the connector to Starburst Enterprise.

Vessel Insight (updated connector)

New in this version is a Voyage table that provides correlated voyage information to match the Vessel data. Voyage data includes departure and destination ports, voyage history and weather data.
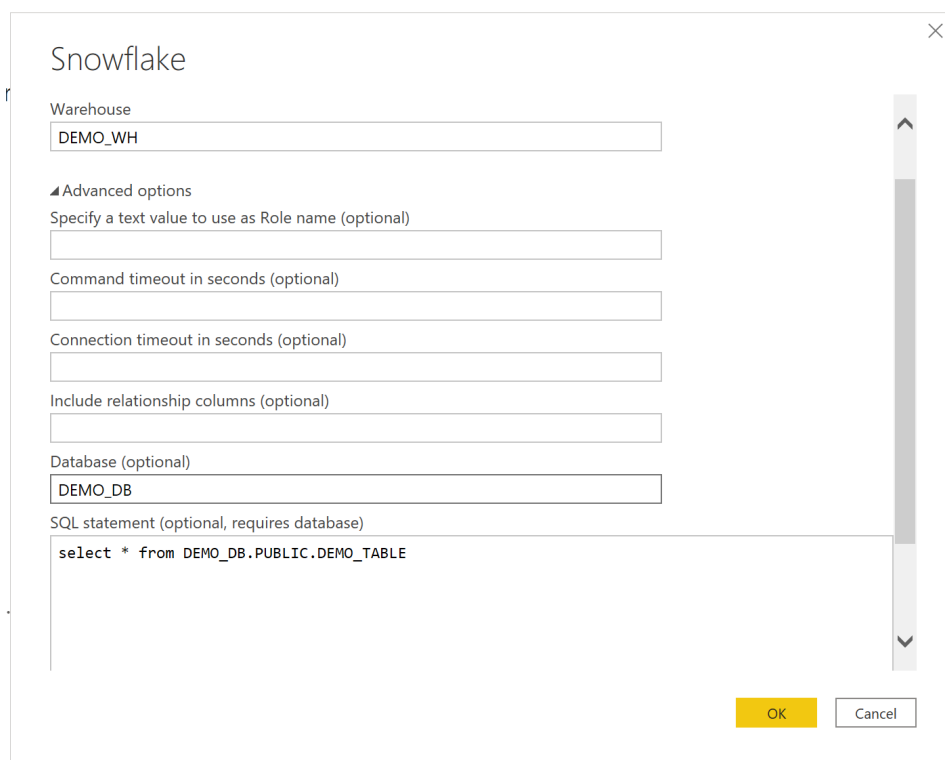
Workplace Analytics (updated connector)

This update adds support for the viva endpoint and contains many other minor bugfixes.

Snowflake (updated connector)

Microsoft has finally added the oft-requested Custom SQL support for the Snowflake connector. Like the SQL connector, this will let you input a Snowflake native query and then build reports on top of it. This will work with both Import and Direct Query mode.

In the initial version, you will need to use fully qualified table names, of the format **Database.Schema.Table.** For example, in the below demo report, there is "select * from DEMO_DB.PUBLIC.DEMO_TABLE". You will also need to fill in the 'Database' optional parameter. You can see these fields filled in below:

Azure Consumption Insights (connector deprecated)

The Microsoft Azure Consumption Insights connector is being deprecated, due to ending support from the data source service. This update will remove the connector from the 'Get Data' experience in Power BI Desktop, discouraging new connections from being made. Existing connections and reports will continue to work *for now*.

Datasets discoverability

In a culture of data reuse, analysts should be able to see what data is available in their organisation for them to use. Therefore, Power BI now allows a data owner to make their dataset discoverable even by users who don't yet have access to the data it contains.

As a data owner, when you have a dataset that is trustworthy and that you consider as having data that others can benefit from and use to create their own reports, then you most probably will want to endorse it first (either promote it or certify it).

With this release, as an administrator or a member of a workspace, when you endorse a dataset, by default you'll also be making it discoverable by other users, who can then request access to it if they don't have access already. In this way, you'll be making data accessible to users while also reducing data duplication in your organisation by making sure users can find the data they need in Power BI instead of recreating it.
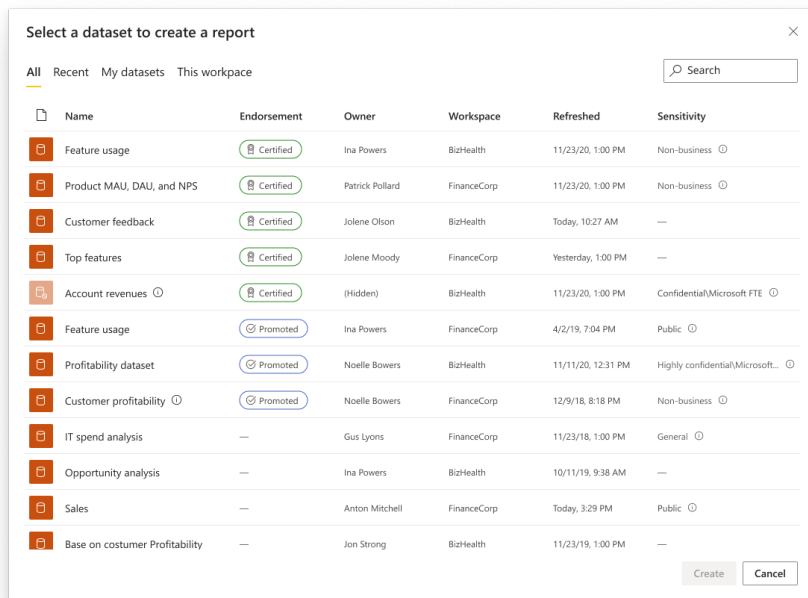
Power BI has taken care to protect this feature, using two layers:

1. Admin tenant settings control the exposure of this feature. For example, administrators may choose to allow only certified datasets to be discoverable, and to disallow promoted datasets

2. Data owners may choose to endorse their dataset, but keep it as non-discoverable by users who don't have the necessary permissions.

Note: Any dataset that was endorsed prior to this release **will not** automatically be made discoverable.

Once you have made the dataset discoverable, other users in your organisation will be able to find it and request access to it in the Datasets Hub. Apparently, in the near future, Microsoft also plans to expose these discoverable datasets in Power BI Desktop, when getting data from Power BI datasets, and in Power BI service, through the Create experience.

In the image below, Account revenues is a dataset without access that was set as discoverable by its owners. Note how it is greyed out:

Request access to datasets

Now that datasets can be made discoverable, users may find data that interests them and request access to it.

In many cases, getting access to data goes through a process that is unique to your organisation and to your dataset. Sometimes end-users are required to go through specific training to get access. Other times, end-users are directed to an organisational application where they may be added as members of an Active Directory security group. Recognising the variety of processes for requesting and granting access, Power

BI allows any dataset owner (workspace administrator / member / contributor) to customise per-dataset instructions for how to get access. When an end user requests access to the dataset, they will be shown a dialog message with details about how to get access.

By default, if no instructions were set, the access request will be sent by email to the user who created the dataset or to the one who assumed ownership.

Mandatory label policy for Microsoft Information Protection sensitivity labels in Preview

Apparently, on of the top request from Power BI users is to help ensure that new content brought into Power BI is applied with sensitivity labels. Therefore, there is now a public Preview of mandatory label policy support in Power BI. When mandatory label policy in Power BI is enabled for users, those users will be required to apply labels when they upload PBIX files to Power BI, create new content or edit content that doesn't yet have a label in Power BI.

Power BI mandatory label policy can be configured in the Microsoft 365 compliance centre, as part of a label policy published to users.

It should be noted that the Power BI mandatory label policy is separate from the mandatory label policy for Office files and emails, and can be turned on and off independently of the Office settings.

Admin API to set and remove Microsoft Information Protection sensitivity labels

To meet compliance requirements, organisations are often required to classify and label all sensitive data in Power BI. This task can be a challenge for tenants that have large volumes of data in Power BI. To make this task easier and more effective, this update introduces a new Power BI admin APIs for setting and removing sensitivity labels in Power BI. With these APIs, Power BI administrators can effectively apply sensitivity labels on large numbers of Power BI artifacts programmatically.

Automate deployments with new APIs and PowerShell samples in Preview

This update also sees the Preview release of the deployment pipelines APIs. The new set of APIs enables developers to retrieve pipelines info and trigger content deployments in a pipeline. With the PowerShell samples, it's now simpler to integrate deployment pipelines with existing DevOps tools, such as Azure Pipelines.

Manage Dataflows in deployment pipelines in Preview

Dataflows can now be managed inside deployment pipelines. These will have similar capabilities as other items managed in deployment pipelines:

- detect updates in a dataflow and highlight them using the 'compare' button

- deploy a dataflow across the different stages of the pipeline, so you can create new dataflows in the test or production stages or update existing stages

- upon deployment, datasets connected to dataflows will bind automatically to the respective dataflow in the target stage in the pipeline

- set rules for a dataflow and connect it to a specific data source or define specific parameter value in the test or production stages. Once rules are set for a data source or a parameter, they will not change upon any future updates.

For this Preview release, linked entities will not be supported yet.
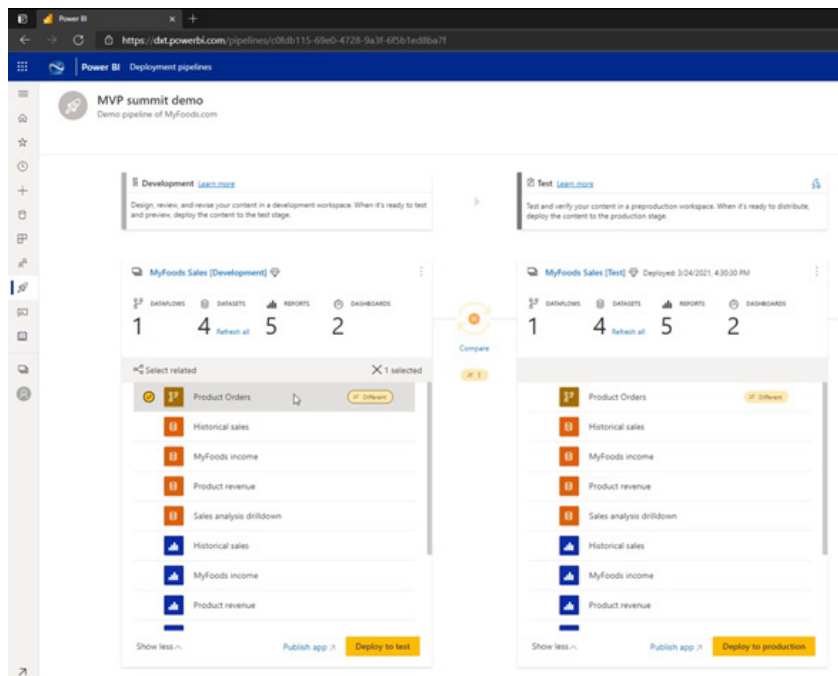
Admin APIs for deployment pipelines

A new set of Admin APIs has also been released. With these APIs, Power BI administrators may view all the existing pipelines in their tenant and add users to any pipeline.

This can help Power BI admins handle a situation of an 'orphaned pipeline', where there's no pipeline administrator available and no one can access that pipeline. Similarly, administrators may use the APIs to 'free' a workspace that is assigned to an 'orphaned pipeline'.

A new look for the Power BI Windows app Preview

In this release, there's a Preview of the Power BI Windows app's new look. The new look features a completely revamped home page that provides a centralised hub for all Power BI content, giving you quick and easy access in one place.
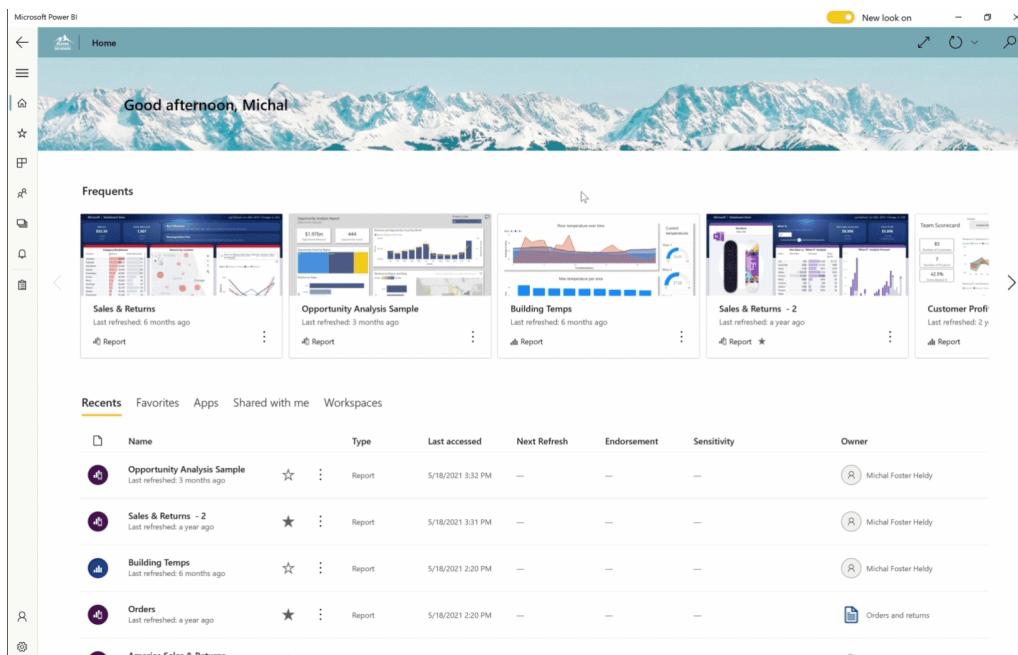
To enable the Windows app's new look, get the latest version of the Windows app from the store and open it. Once it's open, you'll see the 'New look' toggle at the top of the screen. Toggle it on, and that's pretty much it.
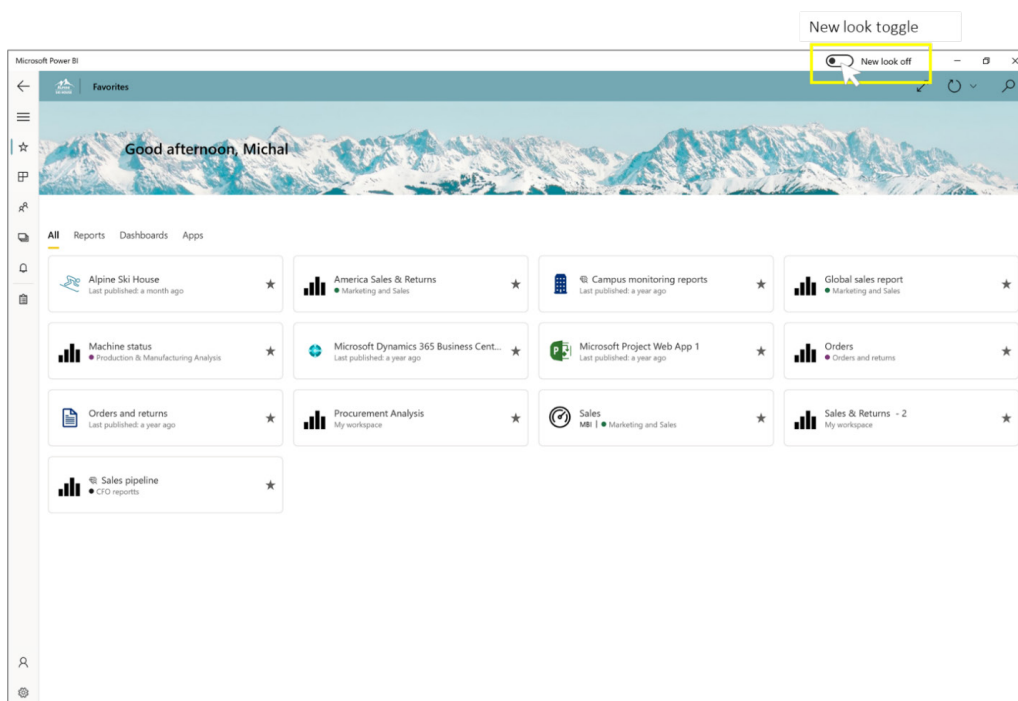
Passing URL parameters to paginated reports

The Mobile apps now support passing parameters to paginated reports via the URL. Passing parameters via the report URL automatically sets the report parameters to those values.

New visuals

New visualisations this month include:

- Power Slider by TME AG
- Growth Rate Chart
- Stratada Program Taskboard
- Charturo Interactive Line Chart
- Multiple Sparklines.

Drill Down Combo Bar PRO by ZoomCharts
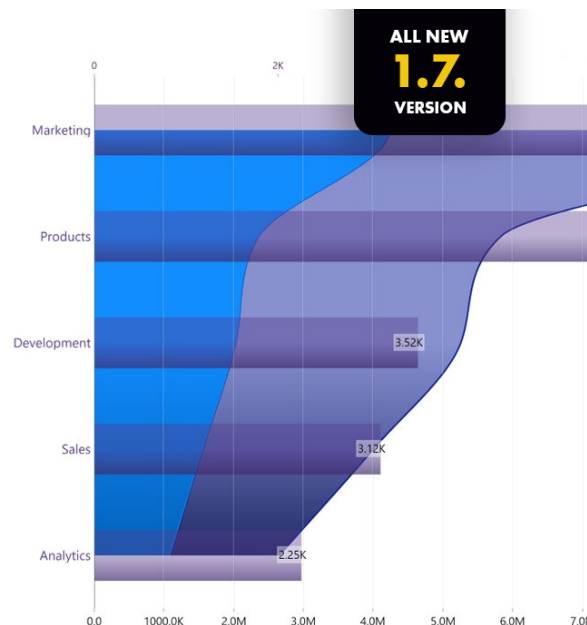
Drill Down Combo Bar PRO by ZoomCharts allows you to combine multiple chart types (line, bar, area), stack and cluster series in multiple ways and apply various customisation options for each series. You can explore multiple drill-down levels, set up to three thresholds to demonstrate targets or benchmarks, customise axes, legends, stacks, ToolTips, fill settings and outlines, and get similar experiences on any device. Among the main features of the new Drill Down Combo Bar PRO you will find:

- **1-click drill down:** click directly to the chart to explore up to nine [9] data levels
- **multiple chart types:** column, line or area
- **full customisation:** customise **X** and **Y** axes, legends, stacks, ToolTips, fill settings and outlines
- **static and dynamic thresholds:** set up to three [3] thresholds to demonstrate targets or benchmarks
- **cross-chart filtering:** instead of using slicers, select data points on multiple charts
- **multi-touch device friendly:** get similar experiences on any device.

To evaluate this Drill Down Combo Bar PRO for free, look out for the 30-day trial period in AppSource.

Dumbbell Bar Chart by Nova Silva

Data visualisations play a fundamental role in comparing results. Typical example questions include:

- how does the sales of this month compare to the sales of last month?
- what is the difference between the number of documents processed this year compared to 2020?
- how does the number of planned visitors compare to the number of unplanned visitors at our locations?

Key in answering these kinds of questions is clearly visualising the difference between the two results. The Dumbbell Bar Chart may assist with this, showing both values and the difference(s) between them.

Dumbbell Bar Chart: display the two values...
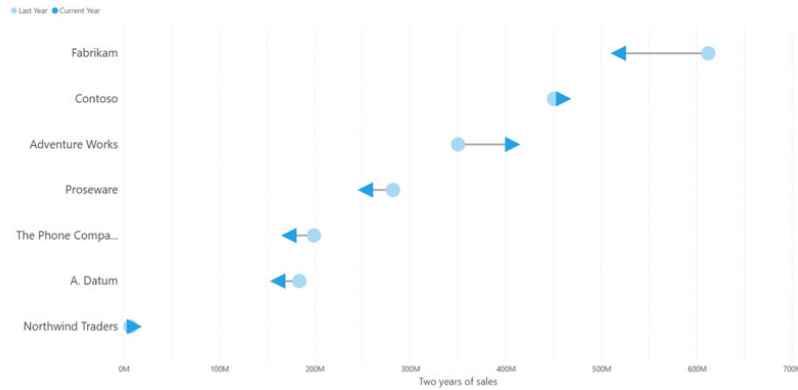
... and optionally include the variance chart
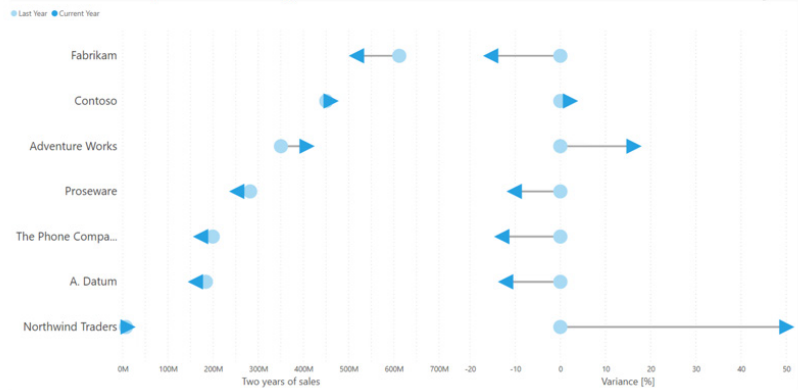
The top chart shows the two values (Last Year and Current Year). This allows the user to identify the growing (*i.e.* Adventure Works) versus the shrinking (*i.e.* Fabrikam) brands. Optionally, the user may include the variance chart (second chart) to increase the emphasis on the difference between the two values.

All features are available for free to evaluate this visual within Power BI Desktop.

graphomate bubbles 2021.2

The goal of this visual is to get a quick and easy overview of distributions and relationships among data points. Patterns and outliers can be recognised quickly for better interpretation and decision-making. The graphomate bubbles can represent portfolios. They offer numerous options for extensive analyses, with the help of circular charts.
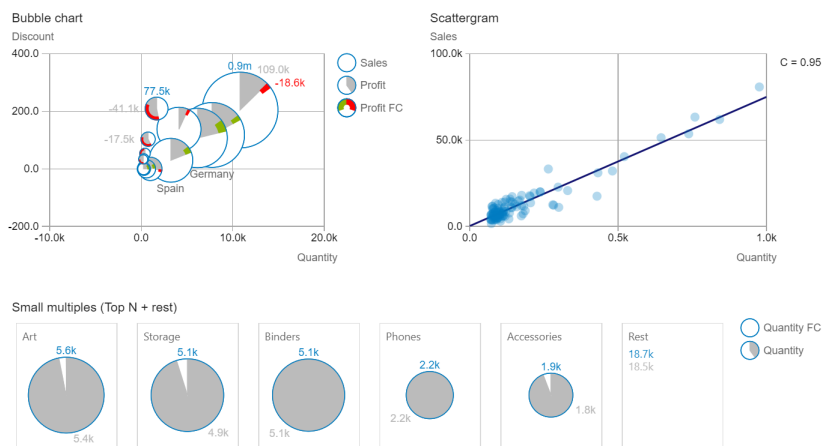
The high information density of a graphomate bubble chart results from the mapping of up to five key figures. For example, deviations from a previous year's value can be plotted on the arc.

The graphomate bubbles may be downloaded from the AppSource.

Zebra BI Tables 5.0

The latest major release of Zebra BI Tables visual 5.0 brings an comments and flexible row formatting, amongst other new features.

You can now simply add a comment data field to the visual to display your comments:

- automatically generates the title based upon the category name and calculates variance for you, if applicable
- responsive and adapts to the size of your table
- dynamic, so you can simply switch the time period to see relevant comments.
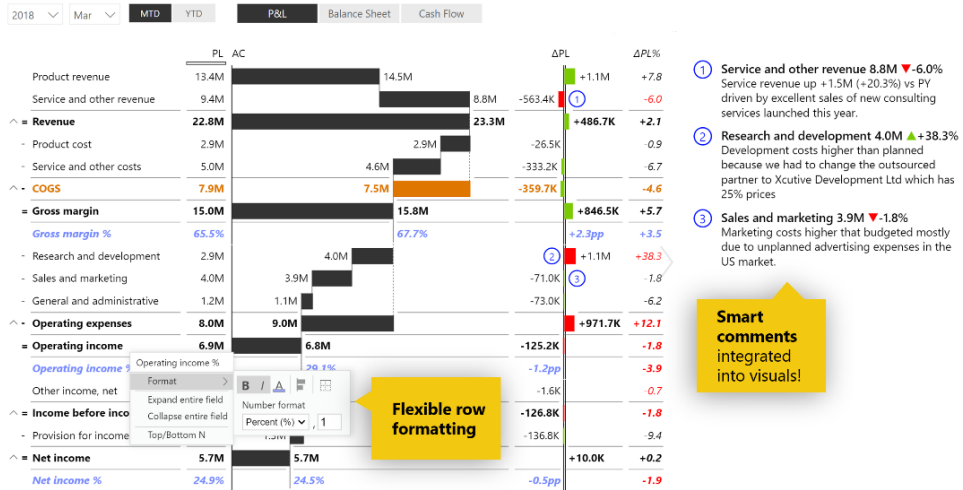
If you want to highlight important elements in your table or when displaying different KPIs in the same table, you can now completely customise the design of each row (font, colour and / or number format).

Furthermore, Zebra BI now fully supports Dark Mode. Batman will be delighted.

Zebra BI Charts 5.0

The latest release of Zebra BI Charts visual 5.0 brings advanced stacked charts, smart comments and more.

To bridge the perceived gap in data visualisation, Zebra BI launched several stacked area and column charts:

- avoid the confusion of showing too many data series: Zebra BI automatically sums up the data into Top **N** + Others for you, while allowing you to change the number of data series in one click
- the adaptive legend makes it much easier to understand the data
- simply right-click on the legend to highlight any data series.

For small multiples fans, this chart is completely integrated with Zebra BI functionalities –viewers can simply switch between stacked charts and small multiples anytime.
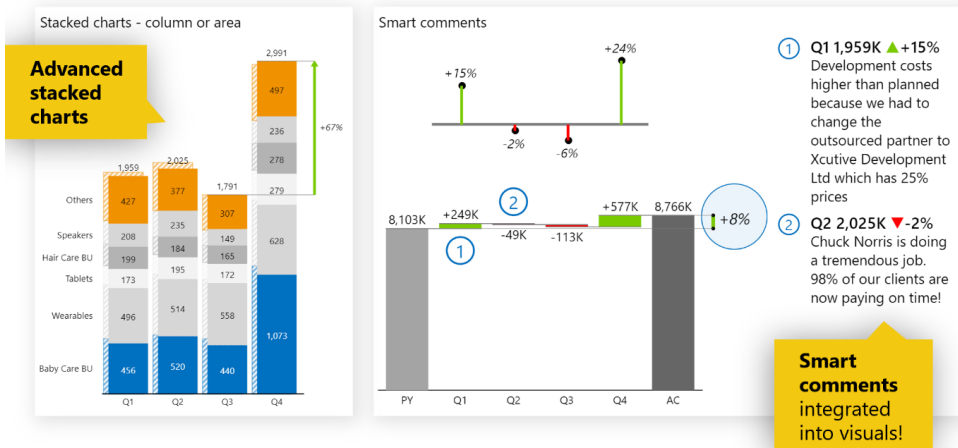
Adding comments is now simpler. Again, simply add your comment data field to the visual to display your comments. The chart will automatically generate the title based upon the category name and calculate variance for you, when applicable.

Template app one-click update and republish

Until this update, after installing a template app update, users also had to go to the template app workspace and re-publish the organisational app to implement the update. Many users found this step confusing and were unsure about what to do. The result was that the organisation app often did not get updated, and thus never included the recent changes and improvements. Now, when you install a template app update, you can choose to update the organisational app as well, enabling you to update the app, navigate to it directly, and continue your work.

The new capability works as follows. When you install an update, you'll notice a new option in the update dialog:
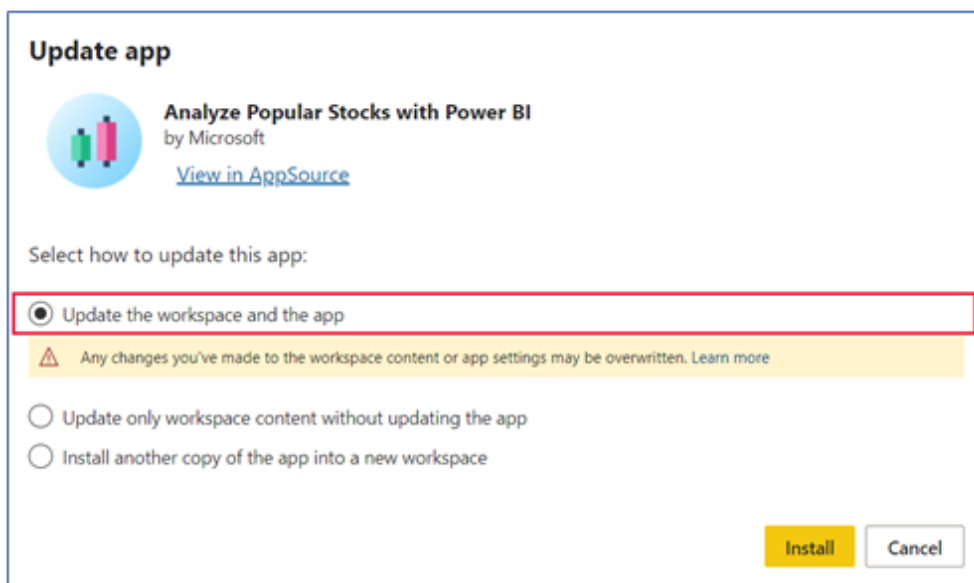
Selecting this new option will install the updates and republish the organisation app. You will be notified when update and republish finishes successfully.
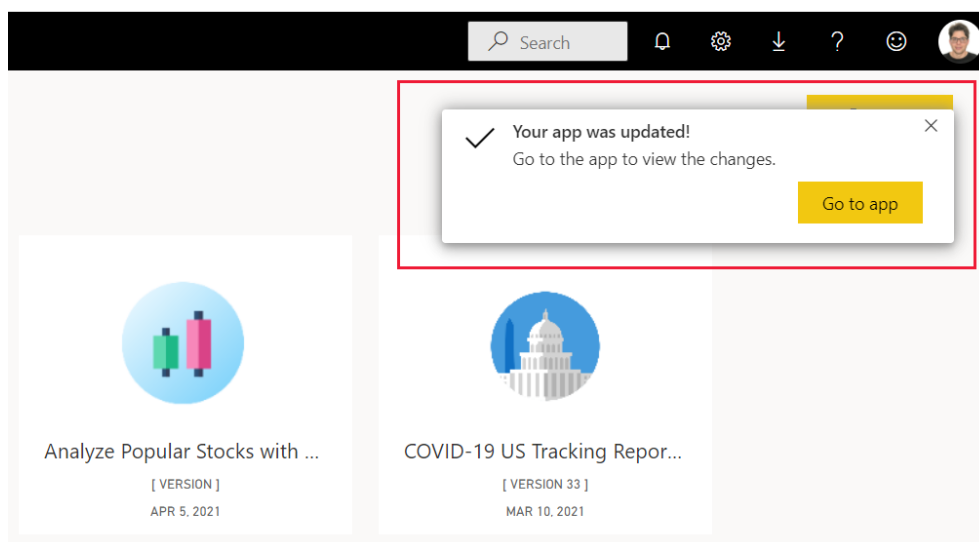
The updated app will now include any updated app branding, such as app name, logo, and navigation, as well as the latest publisher improvements to content. In fact, users who have not made changes to the app's content can still go to the application.

Salesforce Analytics for Sales Managers

Salesforce Analytics for Sales Managers includes visuals and insights for analysing your marketing performance. This dashboard provides key metrics such as your sales pipeline, best accounts and KPI's. You may drill into the report for more details on each aspect, including fully interactive visuals to help you explore your data further.
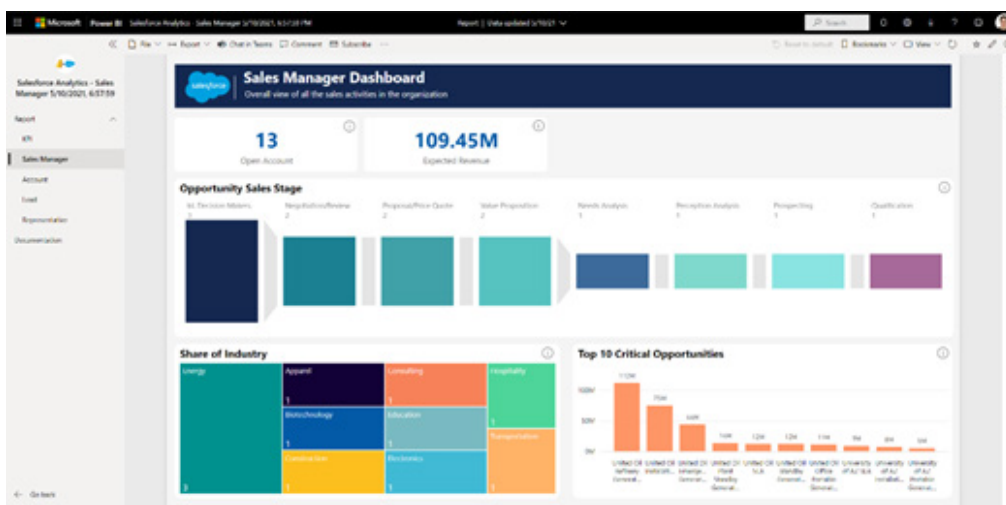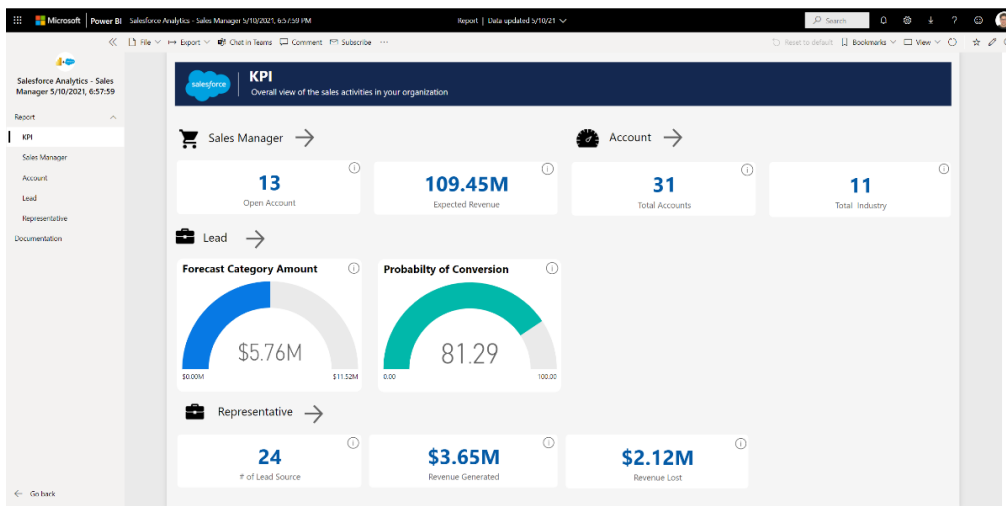
The Salesforce analytics app is brought to you by the Power BI team and is available on AppSource.

Power BI Desktop Installer Changes

Microsoft has also stated that they wanted to give everyone advanced notice to a change that will be coming to the Power BI Desktop .exe installer in July. Starting this month, if you're connected to the internet, we will attempt to download Microsoft Edge WebView2 as part of the installation process. This is currently not a requirement, so you should not notice any changes if you are installing it offline.

To give some context to this change, Microsoft is starting the process of making infrastructure changes to start using WebView2 in place of CefSharp. This switch will apparently better optimise Power BI's development and release process and means that you'll automatically get the latest security patches as the WebView2 team releases them, instead of waiting for Microsoft et al to update Power BI Desktop.

There should be more information about the transition and how it may impact the installation process down the line in next month's newsletter.

Latest (June) update of Power BI Report Builder

A new version of Power BI Report Builder is available now. This release includes a number of new features designed to improve your paginated report authoring experience, including a data tab and a streamlined DAX copy / paste experience. Furthermore, Microsoft has now added Power BI Report Builder as an app in the Microsoft Store.

Report action bar now available in the Power BI SharePoint Online Webpart

SharePoint Online is a popular and highly flexible solution for distributing Power BI reports. Embedding Power BI reports in SharePoint is a common way to engage and inform your organisation with data by providing easy access to reports right in the portals and team sites people access daily.

To provide even more options for integrating Power BI into SharePoint, Microsoft has added a new option to enable the Power BI report action bar within the Power BI SharePoint Online web part. The report action bar provides access to a variety of key features that report viewers in SharePoint can now access.

More next month, we're sure!

The A to Z of Excel Functions: FTEST

Shaded Area = alpha

$F_{critical}$
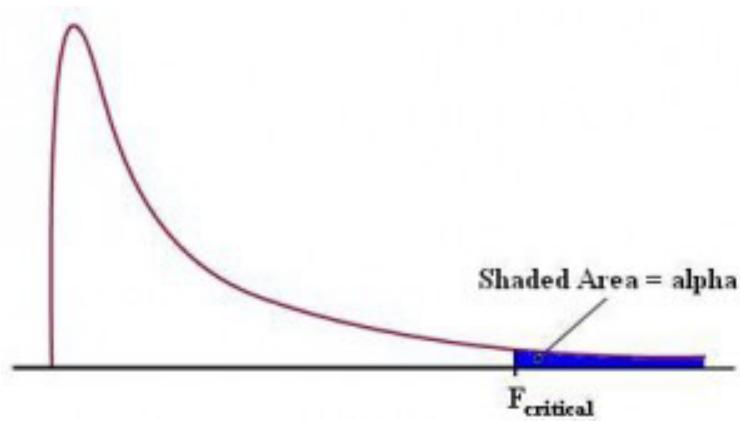
It's survival of the **FTEST** this time out…

The F-distribution is defined as a continuous distribution obtained from the ratio of two chi-square distributions and used in particular to test the equality of the variances of two normally distributed variances a continuous probability distribution, which means that it is defined for an infinite number of different values. This analysis of variance is often referred to as "ANOVA", which I still think is a cheap car.

In slightly simpler terms, the F-distribution can be used for several types of applications, including testing hypotheses about the equality of two population variances and testing the validity of what is known as a multiple regression equation, *i.e.* a formulaic attempt to model the relationship between two or more explanatory variables and a response variable by fitting a linear equation to observed data.

The F-distribution shares one important property with the Student's t-distribution: probabilities are determined by a concept known as **degrees of freedom**. Unlike the Student's t-distribution, the F-distribution is characterized by two different types of degrees of freedom: **numerator** and **denominator** degrees of freedom.

The F-distribution has two important properties:

1. It's defined only for positive values.
2. It's not symmetrical about its mean; instead, it's positively skewed.

A distribution is positively skewed if the mean is greater than the median (the mean is the average value of a distribution, and the median is the midpoint; half of the values in the distribution are below the median and half are above). This function returns the (right-tailed) F probability distribution (degree of diversity) for two datasets.

This distribution is a relatively new one. The associated F statistic is a ratio (a fraction). There are two sets of degrees of freedom; one for the numerator and one for the denominator. As mentioned above, the F distribution is derived from the Student's t-distribution. The values of the F distribution are squares of the corresponding values of the t-distribution.

To calculate the F ratio, two estimates of the variance are made:

1. **Variance between samples:** An estimate of σ^2 that is the variance of the sample means multiplied by n (when the sample sizes are the same). If the samples are different sizes, the variance between samples is weighted to account for the different sample sizes. The variance is also called variation due to treatment or explained variation

2. **Variance within samples:** An estimate of σ^2 that is the average of the sample variances (also known as a pooled variance). When the sample sizes are different, the variance within samples is weighted. The variance is also called the variation due to error or unexplained variation:

- $SS_{between}$ = the sum of squares that represents the variation among the different samples
- SS_{within} = the sum of squares that represents the variation within samples that is due to chance.

To find a "sum of squares" means to add together squared quantities that, in some cases, may be weighted. **MS** means "mean square". $MS_{between}$ is the variance between groups, and MS_{within} is the variance within groups.

To calculate the sum of squares and the mean square, let:

- **k** = the number of different groups
- n_j = the size of the jth group
- s_j = the sum of the values in the jth group
- **n** = total number of all the values combined (total sample size: **$\sum n_j$**)
- **x** = one value: $\sum x = \sum s_j$

The sum of squares of all values from every group combined: $\sum x^2$

The between group variability is therefore

$$SS_{total} = \sum x^2 - \frac{\sum x^2}{n}$$

and the total sum of squares is given by

$$\sum x^2 - \frac{\left(\sum x\right)^2}{n}$$

The explained variation is therefore the sum of squares representing variation among the different samples:

$$SS_{between} = \sum \left[\frac{(sj)^2}{n_j}\right] - \frac{\left(\sum s_j\right)^2}{n}$$

This should be compared with the unexplained variation, namely the sum of squares representing variation within samples due to chance:

$$SS_{within} = SS_{total} - SS_{between}$$

The degrees of freedom for different groups (*i.e.* the degrees of freedom for the numerator) is calculated as **df = k − 1**, whereas the equation for errors within samples (degrees of freedom for the denominator) is calculated as df_{within} = **n − k**.

Further, the mean square (variance estimate) explained by the different groups is

$$MS_{between} = \frac{SS_{between}}{df_{between}}$$

whereas the mean square (variance estimate) that is due to chance (unexplained) is

$$MS_{\text{within}} = \frac{SS_{\text{within}}}{df_{\text{within}}}$$

MS_{between} and MS_{within} can be written as follows:

$$MS_{\text{between}} = \frac{SS_{\text{between}}}{df_{\text{between}}} = \frac{SS_{\text{between}}}{k-1}$$

$$MS_{\text{within}} = \frac{SS_{\text{within}}}{df_{\text{within}}} = \frac{SS_{\text{within}}}{n-k}$$

The one-way ANOVA test depends on the fact that MS_{between} can be influenced by population differences among means of the several groups. Since MS_{within} compares values of each group to its own group mean, the fact that group means might be different does not affect MS_{within}.

The null hypothesis says that all groups are samples from populations having the same normal distribution. The alternate hypothesis says that at least two of the sample groups come from populations with different normal distributions. If the null hypothesis (H_0) is TRUE, MS_{between} and MS_{within} should both estimate the same value.

To be clear, the null hypothesis says that all the group population means are equal. The hypothesis of equal means implies that the populations have the same normal distribution, because it is assumed that the populations are normal and that they have equal variances.

So, all of this brings us to the F-ratio or F-statistic, defined as

$$F = \frac{MS_{\text{between}}}{MS_{\text{within}}}$$

If MS_{between} and MS_{within} estimate the same value (following the belief that H_0 is TRUE), then the F-ratio should be approximately equal to one (1). Mostly, just sampling errors would contribute to variations away from one. As it turns out, MS_{between} consists of the population variance plus a variance produced from the differences between the samples. MS_{within} is an estimate of the population variance. Since variances are always positive, if the null hypothesis is FALSE, MS_{between} will generally be larger than MS_{within}. Then the F-ratio will be larger than one. However, if the population effect is small, it is not unlikely that MS_{within} will be larger in a given sample.

The foregoing calculations were done with groups of different sizes. If the groups are the same size, the calculations simplify somewhat, and the F-ratio can be written as:

$$F = \frac{n \cdot s_{\bar{x}}^2}{s_{\text{pooled}}^2}$$

where

- n = the sample size
- $df_{\text{numerator}} = k - 1$
- $df_{\text{denominator}} = n - k$
- s_{pooled}^2 = the mean of the sample variances (pooled variance)
- $s_{\bar{x}}^2$ = the variance of the sample means.

Now that you have had the statistics lecture, you may use the worksheet function **FTEST** to calculate an F-ratio on the data from two samples. However, it does not return the F-ratio – no, that would be too simple. Instead, it provides the two-tailed probability of the calculated F-ratio under H_0. This means that the answer is the proportion of the area to the right of the F-ratio and to the left of the reciprocal of the F-ratio (*i.e.* 1 divided by the F-ratio).

Important: This function has been replaced by **F.TEST** so that it may provide improved accuracy and whose names better reflect their usage. Although the previous function is still available for backward compatibility, you should consider using this newer function from now on, because the older variant may not be available in future versions of Excel, according to Microsoft.

In summary, the **FTEST** function returns the result of an F-test, *i.e.* the two-tailed probability that the variances in **array1** and **array2** are not significantly different.

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

FTEST(array1,array2)

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

- **array1:** this is required and represents the first array or range of data
- **array2:** this is also required and denotes the second array or range of data.

It should be further noted that:

- the arguments must be either numbers or names, arrays or references that contain numbers
- if an array or reference argument contains text, logical values or empty cells, those values are ignored
- cells with the value zero will be included
- if the number of data points in **array1** or **array2** is less than two (2), or if the variance of **array1** or **array2** is zero, FTEST returns the *#DIV/0!* error
- the F-test value that is returned by the **LINEST** function differs from the F-test value that is returned by the **FTEST** function. **LINEST** returns the F-statistic whereas **FTEST** returns the probability.

Please see our example below:

▲	A	B	C
1	Array1	Array2	
2	6	1	
3	7	2	
4	8	4	
5	9	5	
6	10	6	
7			
8			
9	Formula	Description	Result
10	=FTEST(A2:A6,B2:B6)	F-test for the data sets in A2:A6 and B2:B6.	0.6122

The A to Z of Excel Functions: FV

If you have ever been involved calculating financials, you will appreciate interest is a fundamental aspect. Annuities often need to be calculated, *i.e.* regular, periodic payments of identical amounts earning a similar rate of return.

Perhaps the easiest way to think of it is as follows:

- Let's assume interest is set at 10% (and we will assume this is after tax)
- Something that is invested at $100 this year will increase by 10% next year, *i.e.* be valued at $110
- Something that is invested at $100 this year will increase by 10% over the next two years, *i.e.* be valued at $121
- Something that is invested at $100 this year will increase by 10% over the next three years, *i.e.* be valued at $133.10. *etc.*
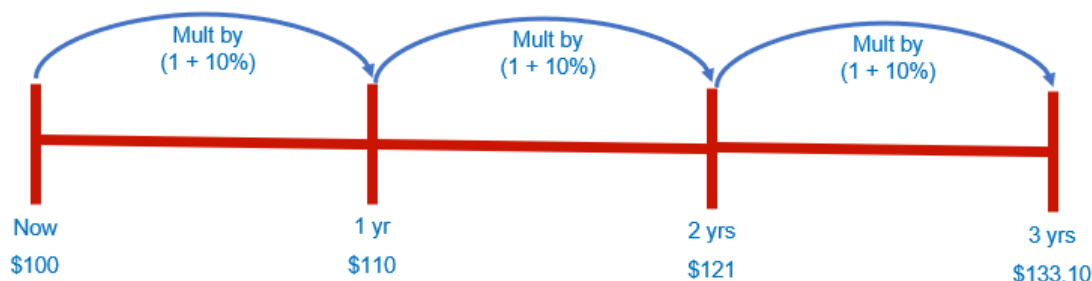
Note that all of these valuations are for a *point* of time <u>not</u> a period. This is a common mistake in modelling.

The **FV** function calculates the future value of an investment based on such a constant interest rate. You can use FV with either periodic, constant payments or a single lump sum payment.

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

FV(rate, nper, pmt, [pv], [type])

The FV function has the following arguments:

- **rate:** this is required and represents the interest rate per period
- **nper:** this is also required. This represents the total number of payment periods in an annuity
- **pmt:** again, this is required. This is the payment made each period; it cannot change over the life of the annuity. Typically, **pmt** contains principal and interest but no other fees or taxes. If **pmt** is omitted, you must include the **pv** argument
- **pv:** this argument is optional. This is the present value, or the lump-sum amount, that a series of future payments is worth right now. If **pv** is omitted, it is assumed to be zero (0), and you must include the **pmt** argument
- **type:** this is also optional. The type should either be zero (0) or one (1) and indicates when payments are due. If type is omitted, it is assumed to be zero (0).

Type	Payments due
0	At the end of the period.
1	At the beginning of the period.

Make sure that you are consistent about the units you use for specifying **rate** and **nper**. If you make monthly payments on a four-year loan at 12% annual interest, use 12%/12 for **rate** and 4*12 for **nper**. If you make annual payments on the same loan, use 12% for **rate** and 4 for **nper**.

For all the arguments, cash you pay out, such as deposits to savings, is represented by negative numbers; cash you receive, such as dividend receipts, is represented by positive numbers.

Sometimes, the numbers aren't quite what you expect for this function. That's because Microsoft uses the following relationship to be consistent across its related financial functions:

$$pv*(1+rate)^{nper} + pmt(1+rate*type)*$$
$$\left(\frac{(1+rate)^{nper}-1}{rate}\right) + fv = 0$$

Please see our example below:

	A	B	C	D	E
1	Description	Scenario 1	Scenario 2	Scenario 3	
2	Annual interest rate	6.00%	12.00%	11.00%	
3	Number of payments	10	12	35	
4	Amount of the payment	(200)	(1,000)	(2,000)	
5	Present value	(500)			
6	Payment timing	1		1	
7					
8					
9	Formula	Description		Result	
10	=FV(B2/12,B3,B4,B5,B6)	Future Value of Scenario 1		$ 2,581.40	
11	=FV(C2/12,C3,C4)	Future Value of Scenario 2		$ 12,682.50	
12	=FV(D2/12,D3,D4,,D6)	Future Value of Scenario 3		$ 82,846.25	
13					

The A to Z of Excel Functions: FVSCHEDULE

If you have ever been involved calculating financials, you will appreciate interest is a fundamental aspect. Sometimes, interest rates vary and you want to calculate what the amount, including interest, will be worth at a later point in time.

Related to the **FV** function, the **FVSCHEDULE** function calculates the future value of an investment based on variable compound interest rates and employs the following syntax to operate:

FVSCHEDULE(principal, schedule)

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

- **principal:** this is required and represents the present value
- **schedule:** this is also required. This represents an array of interest rates to apply.

The values in **schedule** can be numbers or blank cells (treated as 0%) if in cell ranges; cited explicitly, the values must be decimals (percentages and blanks will not work). Any other value produces the *#VALUE!* error value for **FVSCHEDULE**. Each period is assumed to be of equal length and the interest rate is quoted for the period in question (*e.g.* 3% per quarter if in quarters).

Please see our final example for this month below:

	A	B	C	D	E
1	Period	Rate			
2	Initial amount (principal)	100			
3	Period 1	5.00%			
4	Period 2	10.00%			
5	Period 3	20.00%			
6					
7					
8	Formula	Description		Result	
9	=FVSCHEDULE(B2,B3:B5)	Future Value given variable rates		$ 138.60	
10	=FVSCHEDULE(100,{0.05,0.1,0.2})	Alternative statement		$ 138.60	
11					

More Excel Functions next month.

Beat the Boredom Suggested Solution

In this newsletter, we considered allocating days across monthly periods, where some dates are outside the forecast period and others straddle the (monthly) reporting periods.

As a reminder from earlier, this month sees you work for an education establishment, seeking to model forecast data for the calendar year 2021. There are four terms relevant to your modelling period:

Assumed Term Dates

Term	Start	End	Days in Year	Check
1	30 Nov 20	6 Mar 21	65	☑
2	27 May 21	17 Jun 21	22	☑
3	1 Sep 21	4 Oct 21	34	☑
4	17 Oct 21	7 Aug 22	76	☑
			197	☑

All we need to do is allocate the number of term days (including weekends and public holidays) to each month of 2021, *i.e.*

|		Jan 21	Feb 21	Mar 21	Apr 21	May 21	Jun 21	Jul 21	Aug 21	Sep 21	Oct 21	Nov 21	Dec 21
Term Dates in Each Month													
Number of Days	197	31	28	6	-	5	17	-	-	30	19	30	31

The challenge was "simply" this: were you able to construct a calculation such that the correct number of days would be allocated to each month? You were advised that you may assume the terms would be in chronological order, that they would not overlap, there will never be more than two terms associated with any given month, and there will never be two start dates or two end dates in the same month.

Suggested Solution

It is possible to get into a mess with this quite quickly, because there are several scenarios to consider for any given reporting month. To get our heads around this, let's consider just one monthly period, between **x** and **y**:
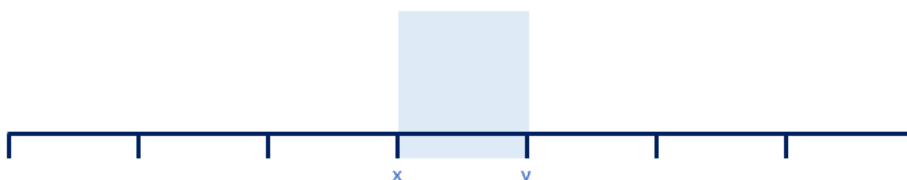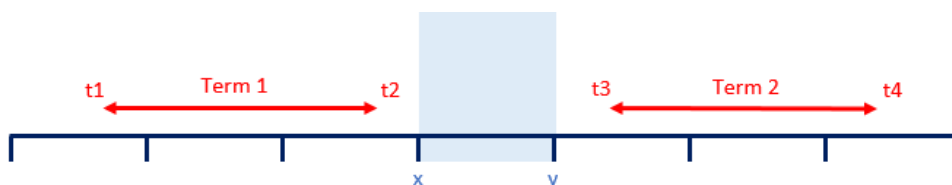

If **x** and **y** are day numbers, then the number of days in the period is
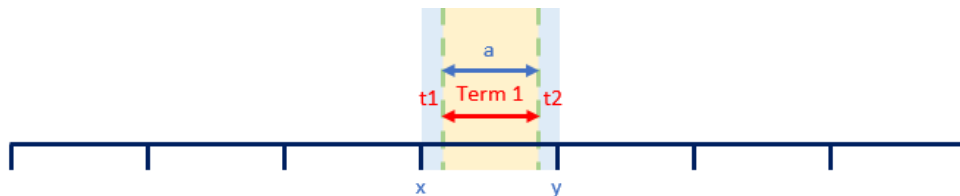
$$= y - x + 1$$

(we have to add one [1] as day **x** is actually in the period). But how many days relate to the term? Well, it depends. There are actually five scenarios to consider:

Scenario 1: No Term Days

In this instance, no term days fall in the given period, so the number of days between x and y will be zero (0).
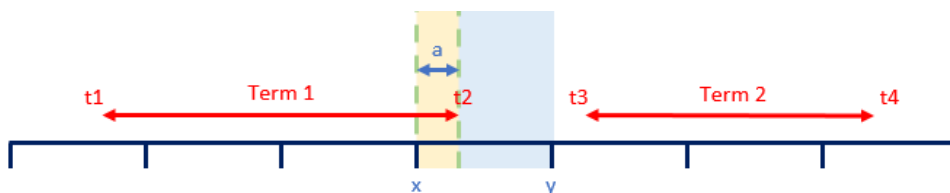
Scenario 2: Term Fully Insider Reporting Period

Here, the term starts at **t1**, which is greater than or equal to **x**, and ends at **t2**, which is less than or equal to **y**. The number of days (**a**) is therefore given by

$$= MIN(y, t2) - MAX(x, t1) + 1$$

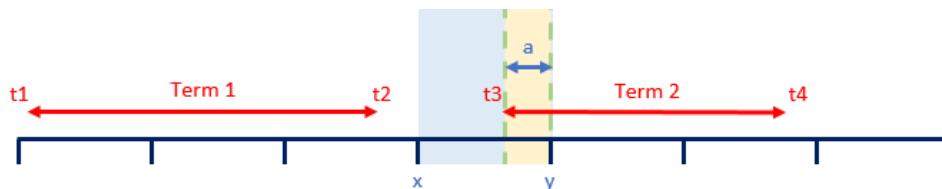## *Scenario 3: Term Ends in Reporting Period*

Here, I have Term 1 ending in the reporting period at **t2**, but Term 2 starts in a later period (*i.e.* after **y**) at time **t3**. Therefore, the number of days (**a**) is given by

$$= MIN(y, t2) - x$$

Technically, this formula is still

$$= MIN(y, t2) - MAX(x, t1) + 1$$

Scenario 4: Term Starts in Reporting Period

Here, I have Term 2 starting in the reporting period at **t3**, but Term 1 ends in an earlier period (*i.e.* before **x**) at time **t2**. Therefore, the number of days (**a**) is given by

$$= y - MAX(x, t3) + 1$$

Since there is no distinction (algebraically) between **t3** and **t1**, and similarly, between **t4** and **t2**, we may interchange these variables, so that this formula is still equivalent to

$$= MIN(y, t2) - MAX(x, t1) + 1$$

In other words, Scenarios 2, 3 and 4 may all be solved using the same formula.

Scenario 5: One Term Ends and Another Starts in the Reporting Period

This situation is distinct from the others:
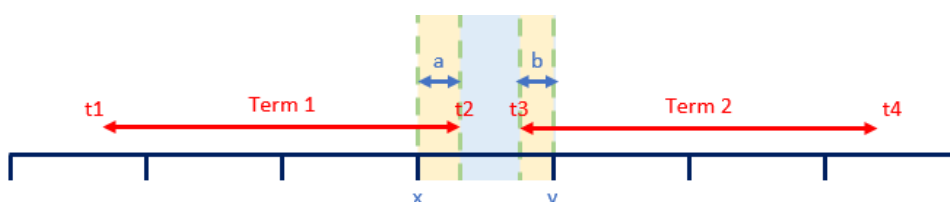
Here, I have Term 1 ending in the reporting period at **t2**, but Term 2 also starts in the same period at time t3. Therefore, the number of days (**a + b**) is given by

$$= MIN(y, t2) – MAX(x, t1) + 1 + MIN(y, t4) – MAX(x, t3) + 1$$

which may be simplified slightly to

$$= MIN(y, t2) – MAX(x, t1) + MIN(y, t4) – MAX(x, t3) + 2$$

This situation occurs when a reporting period has both a relevant start date and a relevant end date, and the relevant start date occurs *after* the relevant end date for that period.

Modelling the Solution

We are now in a position to model the number of days in a period. There are essentially three categories that the scenarios fall into:

1. No term days are in the reporting period

2. At least one of the start and / or end dates for a given term is in the reporting period (start date must be less than or equal to the end date)

3. One term ends and the next term starts within the reporting period (start date occurs after the end date for that reporting period).

Calculating the relevant end dates are not the problem. If the end date of a term falls outside the reporting period, we just use the end of the period instead (deemed end). However, start dates have two issues:

1. The start date may have occurred prior to the forecast period. If the end date is also earlier, then this constitutes no issue, but if the end date is within the period, we will have to assume the deemed start is the first day of the reporting period

2. We need to check if the start date starts before / coincides or ends after the end date for a given reporting period. This will determine the method of calculation.

With all this borne in mind, we are now in a position to model the problem. First, we need to add the assumptions into our model:

| Term	Start	End	Days in Year	Check
1	30 Nov 20	6 Mar 21	65	☑
2	27 May 21	17 Jun 21	22	☑
3	1 Sep 21	4 Oct 21	34	☑
4	17 Oct 21	7 Aug 22	76	☑
			197	☑

Assumed Term Dates

There are checks here to ensure that dates have been entered correctly a that the end date does not occur before its associated start date, *etc.*

Next, the relevant month ends associated with these dates are calculated using the EOMONTH function.

EOMONTH Reminder

As a recap, this function returns the serial number (Excel's numerical representation of a date) for the last day of the month that is the indicated number of months before or after a given **start_date**. You should use **EOMONTH** to calculate calendar dates, maturity dates or due dates that fall on the last day of the month. This is a particularly useful function for creating time series in financial models.
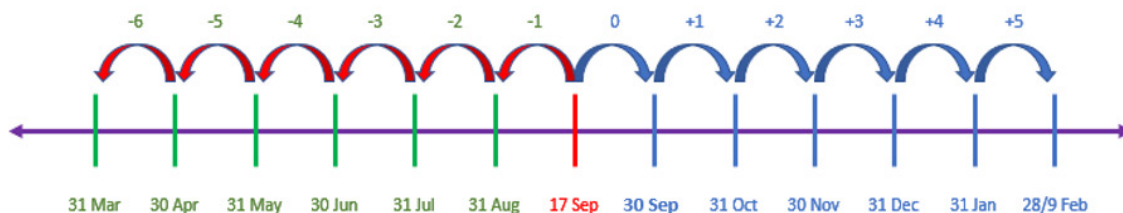
The **EOMONTH** function employs the following syntax to operate:

EOMONTH(start_date, months)

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

- **start_date:** this is required and represents the start date. Dates should be entered by using the **DATE** function, or as results of other formulas or functions. For example, consider using **DATE(**2020,11,2**)** for the second (2nd) day of November, 2020. Problems can occur if dates are entered as text

- **months:** this is also required. This represents the number of months before or after the **start_date**. A positive value for months yields a future date; a negative value yields a past date.

Returning to the Modelling

The end of months used are simple for the end dates, but the start dates formula is a little more convoluted.

The formula in cell **F32**, for example, is

$$\text{=IF(G32>=\$J\$5,EOMONTH(MAX(F18,\$J\$5),0),EOMONTH(F18,0))}$$

This checks to see if the end date occurs on or after the first day of the modelling period. If so, the end of month for the start date is assumed to be the greater of the end of month of the given start date or the end of the first reporting month, whichever is later. This is in case the start date occurred earlier than the model allows for (otherwise the number of days calculated would be incorrect).

This takes us to our final calculations:

Apologies for the still small, yet truncated, screenshot.

It is best to start with row 45, the Relevant End Date, as the concept here is simpler than its counterpart for the Relevant Start Date. The formula in cell **J45** is given by

$$\text{=IFERROR(IF(LOOKUP(J\$5,\$G\$32:\$G\$35)=J\$5,LOOKUP(J\$5,\$G\$32:\$G\$35,\$G\$18:\$G\$21),""),"")}$$

This formula uses LOOKUP.

LOOKUP Formula

The alternative form is the vector form:

$$\text{LOOKUP(lookup\_value, lookup\_vector, [result\_vector])}$$

The LOOKUP function vector form syntax has the following arguments:

- **lookup_value** is the value that **LOOKUP** searches for in the first vector
- **lookup_vector** is the range that contains only one row or one column
- result_vector is optional – if ignored, **lookup_vector** is used (which is what happens in the first part of the above formula) – this is the where the result will come from and must contain the same number of cells as the **lookup_vector**.

The **lookup_value** must be located in a range of ascending values, *i.e.* where each value is larger than or equal to the one before, which is why the terms must be in chronological order.

Returning to the Modelling (Again!)

The formula in cell **J45** is given by

$$\text{=IFERROR(IF(LOOKUP(J\$5,\$G\$32:\$G\$35)=J\$5,LOOKUP(J\$5,\$G\$32:\$G\$35,\$G\$18:\$G\$21),""),"")}$$

This checks if the end of term date occurs in that reporting period, and if so, returns that end date. If it does not, the cell is attributed the empty ("") string instead.

The formula in cell **J44** uses this result, which is why the second row has been explained first. The formula in cell **J44** is given by

$$=IFERROR(IF(LOOKUP(J\$5,\$F\$32:\$F\$35)=J\$5,LOOKUP(J\$5,\$F\$32:\$F\$35,\$F\$18:\$F\$21),$$
$$IF(I\$44>N(I\$45),I\$44,"")),"")$$

The first part of the formula is very similar to the calculation in the row below. It checks if the start of term date occurs in that reporting period, and if so, returns that start date. However, if it does not, it then checks whether there was a start and end date in the previous period. As long as the start date exists (this is why **N(I$45)** is used – so that a blank cell is treated as zero), it then checks if the start date occurs after the end date. This is to see whether we are in *Scenario 5*, as explained earlier. In this case, the start date is required.

Astute readers will note that more dates should be visible than are shown in rows 44 and 45. This is because we have hidden subsequent dates if they are not relevant to that period. Dates have been made "invisible" using conditional formatting in that instance, *viz*.
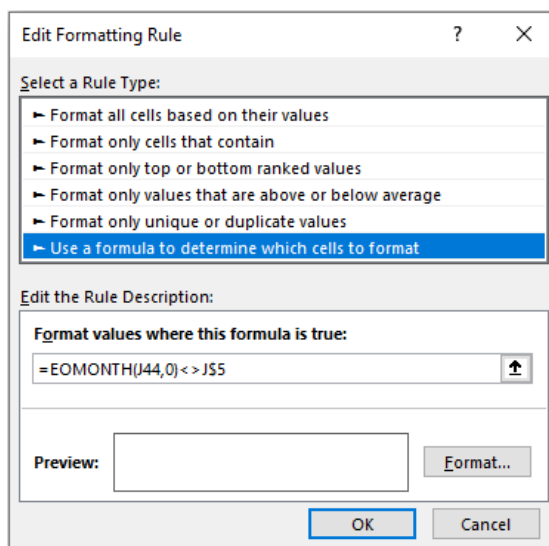
We are nearly there. The penultimate formula in cell **J46** defines the Calculation Type.

$$=(COUNT(J\$44:J\$45)<>0)*1+1+(N(J\$44)>N(J\$45))*(N(J\$45)<>0)$$

This formula generates a number between one (1) and three (3):

1. No term days are in the reporting period
2. At least one of the start and / or end dates for a given term is in the reporting period (start date must be less than or equal to the end date)
3. One term ends and the next term starts within the reporting period (start date occurs after the end date for that reporting period).

This is simply the categorisations we derived what seemed like a lifetime ago now.

This leads us elegantly to our final formula in cell **J47**, which derives the Number of Days:

$$=CHOOSE(J\$46,,MIN(J\$5,J\$45)-MAX(J\$6,J\$44)+1,J\$45+J\$5-J\$44-J\$6+2)$$

The CHOOSE function allows us to calculate the result all in one line.

CHOOSE Function

This function uses the first argument **index_number** to return a value from the list of value arguments. **CHOOSE** may be used to select one of up to 254 values based on the index number (index_number).

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

CHOOSE(index_number, value1, [value2])

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

- **index_number:** this is required and is used to specify which value argument is to be selected. The **index_number** must be a number between 1 and 254, or a formula or reference to a cell containing a number between 1 and 254.

    o   if **index_number** is 1, CHOOSE returns **value1**; if it is 2, CHOOSE returns **value2**; and so on
    o   if **index_number** is less than 1 or greater than the number of the last value in the list, **CHOOSE** returns the *#VALUE!* error value
    o   if **index_number** is a fraction, it is truncated to the lowest integer before being used.

- **value1, value2, ...: value 1** is required, but subsequent values are optional. There may be between 1 and 254 value arguments from which **CHOOSE** selects a value or an action to perform based on **index_number**. The arguments can be numbers, cell references, defined names, formulas, functions or text.

One Last Time

As stated above, the formula in cell J47 derives the Number of Days:

$$\text{=CHOOSE(J\$46,,MIN(J\$5,J\$45)-MAX(J\$6,J\$44)+1,J\$45+J\$5-J\$44-J\$6+2)}$$

Cell **J46** returns a value of 1, 2 or 3 to determine the category. If the value is:

- zero (0): the number of days returned is zero (0), as there are no term days in the reporting month
- one (1): the formula used is **MIN(J\$5,J\$45)-MAX(J\$6,J\$44)+1**, which is the Excel equivalent of the calculation derived earlier for *Scenarios 2*, *3* and *4*
- two (2): the formula used is **J\$45+J\$5-J\$44-J\$6+2**, which is the Excel equivalent of the calculation derived in *Scenario 5*.

The final check in the screenshot,

simply checks that the number of days in row 47 is equal to the number of days calculated earlier

in cell **H22**.

Word to the Wise

I am conscious there are other ways to calculate a solution to this month's challenge. Some of you may have elected to work out which term was relevant on the first day of the period, and which was relevant on the last day. That works in this situation too, as only two terms may straddle any given month (one of the assumptions stated at the beginning).

However, in more complex scenarios you may need to revert to first principles, which is why I chose to use my approach. You can see – *visualise* – my thinking. In modelling, always try to be aware of what might braeak your formula. If you become really expert at this, you might be lucky enough to spot none out of 10 such instances!

Until next time.

Location	Course	Date	Date	Duration	Duration
Online (Australia)	Power Pivot, Power Query and Power BI	19 - 21 Jul 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	26 Jul 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	27 - 28 Jul 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	23 - 25 Aug 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	30 Aug 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	31 Aug - 1 Sep 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	29 Sep - 1 Oct 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	6 Oct 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	7 - 8 Oct 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	3 - 5 Nov 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	10 Nov 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	11 - 12 Nov 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	2 Days
Online (Australia)	Power Pivot, Power Query and Power BI	8 - 10 Dec 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	3 Days
Online (Australia)	Excel Tips and Tricks	15 Dec 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	1 Day
Online (Australia)	Financial Modelling	16 - 17 Dec 2021	09:00-17:00 AEDT	(-1 day) 22:00-07:00 GMT	2 Days

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This year, we thought we'd revisit each function key in depth (there are 12 – one for each month of the year!). Given it's now July, let's look at ALL of the **F7** tips, which highlights Microsoft had probably given up at this stage!

Keystroke	What it does
F7	Check spelling
CTRL + F7	Move window

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

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

Our Services

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

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

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

Link to Others

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

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

Any Questions?

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

Training

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

Check out our more popular courses in our training brochure:

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