

Sum Product

NEWSLETTER #151 - June 2025

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



We say goodbye to an old friend this month Skype goes dark as it gets subsumed by Teams. Perhaps it missed its calling – get it...? [Groan – Ed.]

In another packed newsletter, we go back to our roots and consider how to balance that Balance Sheet in your financial model. It's been a while, but we decided it was time to discuss modelling with another meaty article.

But then there are all the usual features too. There is another Beat the Boredom Challenge, Charts & Dashboards Tips, Over to AI, Excel for Mac, Visual Basics, Power Pivot Principles, Power Query Pointers, Power BI Updates, Excel Updates, plus we see the A to Z of Excel functions consider **PIVOTBY** and we conclude by putting a **SHIFT** in on our Keyboard Shortcuts.

Happy reading and remember: stay safe, stay happy, stay healthy.

Liam Bastick, Managing Director, SumProduct



Skype Moves to the Dark Side



If May 4 was Star Wars day ("May the Fourth be with you"), then May 5 will go down as the day Skype was retired with existing users needing to export their data or else migrate to (a free version of) Teams.

Existing Skype users will be able to log in to the Microsoft Teams application and have their message history, group chats and contacts all automatically available without having to create another account, if they don't wish to export their data elsewhere. However, it should be noted that Microsoft will be phasing out support for calling domestic or international numbers.

Any contacts and chats from Skype will automatically appear when users

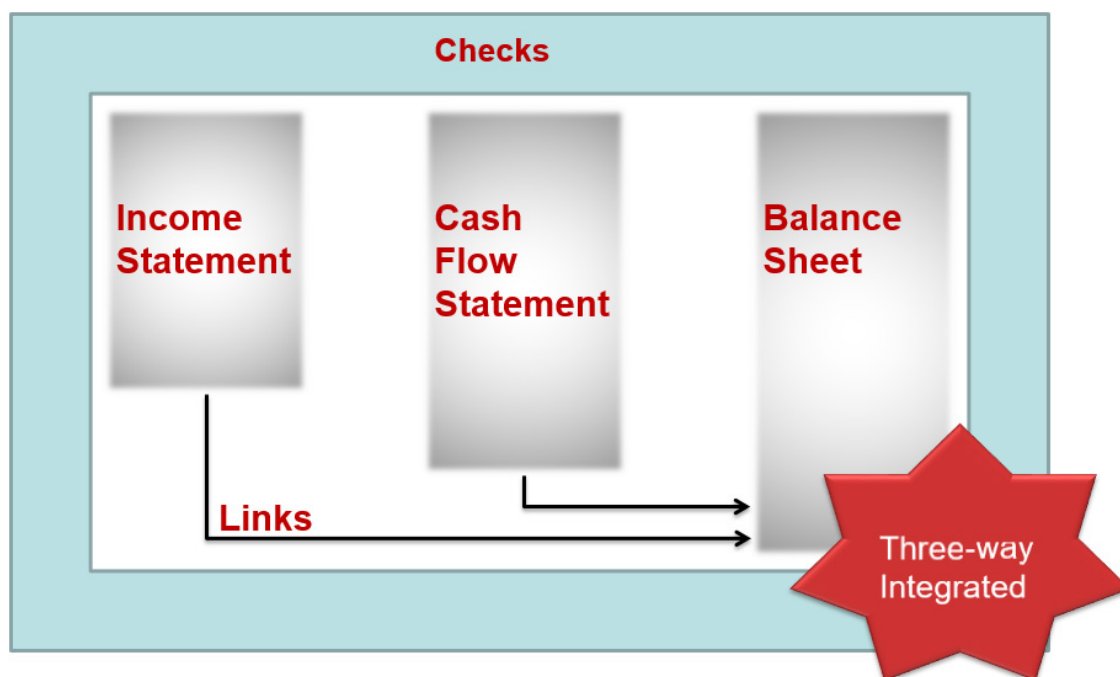
log into the free version of Teams with their Skype credentials. Skype accounts will continue to exist but will be subsumed into Teams.

Contrary to some early reports, Microsoft will honour existing Skype credits. Existing Skype subscription users will be able to use their Skype credits and subscriptions inside Microsoft Teams until the end of their next renewal period. Existing Skype Number users will also need to port their number over to another provider, as Microsoft is no longer supporting this, either. The Skype Dial Pad will be part of Teams temporarily for existing credits and subscriptions, but Microsoft isn't going to offer calling plans to Teams consumers unlike the service for businesses.



Balancing the Balance Sheet

We have been building financial models for many years and in that time, we've realised our modelling has become simpler and simpler. We talk of creating "three-way integrated" financial statements, but what exactly does that mean? Consider the following:



The three financial statements, the Income Statement, Cash Flow Statement and Balance Sheet are linked together with the appropriate checks incorporated to ensure dataflows are consistent, there are no errors, values flow through the model and the Balance Sheet balances.

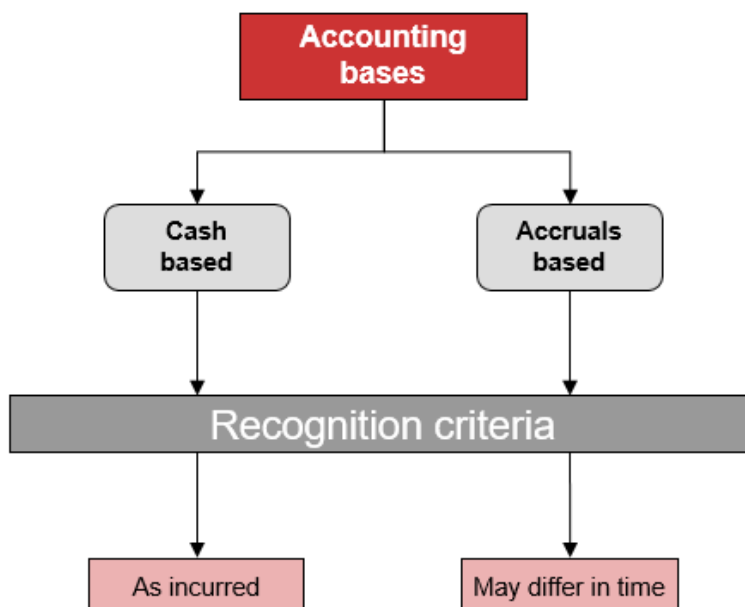
Therefore, as we build calculations, we need to ensure all entries to the

financial statements are included, linked accordingly and checked for balancing issues. It really is as simple as that.

Hence, we wish to propose a method of building up a financial model. To that end, let's consider the three [3] key outputs.

1. Income Statement

This is the financial statement that shows the net operating profit of an entity for a given period of time. It works on an accruals basis, which is pretty much how *everybody* thinks, not just accountants.



Income is recognised when products are delivered or services are provided, not when payment is received. Similarly, costs are attributed to the period they are incurred, not when they are necessarily paid. If our company sells one million widgets within the financial year at \$1 each and incurs direct costs of 75c per widget, we would expect a gross profit of \$250,000, viz.

Number of Widgets:	1,000,000
Unit Price:	\$ 1.00
Unit Cost:	\$ 0.75

	\$
Sales	1,000,000
Cost of Goods Sold	(750,000)
Gross Profit	250,000

The cash position could be radically different. We may have had to pay all of the costs and not yet received any monies. However, this is not how we think. We all attribute on an accruals basis.

And there’s more. If you were asked, what would you model first, then second, then third, who here was thinking, revenue, costs of goods sold and then operating expenditure? Like it or not, we are walking talking income statements:

Revenue	X
COGS	(X)
Gross Profit	X
Operating Expenditure	(X)
EBITDA	X
Depreciation	(X)
EBIT	X
Interest Expense	(X)
Net Profit Before Tax (NPBT)	X
Tax Expense	(X)
Net Profit After Tax (NPAT)	XX

Why have we made such a big deal of the order? Go and ask a non-accountant the order they would build a model in: the vast majority would build the income calculations first, the direct costs second and so

on. The reason the Income Statement is always so popular is this is how people think. We are great believers in “if it ain’t broke, don’t fix it” and this ideology very much applies here.

We should consider building a model in the P&L order as this is most intuitive to modellers and end users alike.

2. Balance Sheet

How would you explain the Balance Sheet to a non-accountant? For those going, "it's the comparison for a moment in time of an entity's assets, liabilities and equity", please explain "assets", "liabilities" and "equity". The Balance Sheet used to be known as the Net Worth statement, summarising the financial position. For a particular moment in time (the date stated) it displayed what a business was worth.

Modellers tend to create Balance Sheets as a bit of an afterthought, but they are more important than that. In fact, modellers *hate* Balance Sheets. They never seem to balance, reconcile or be understood. However, this may all be circumvented with **control accounts** which

will be discussed below. Balance Sheets are essential for three-way integrated financial statement modelling.

There is one other issue. Balance Sheets, by their very nature, are cumulative. They are stated at a point in time. They have to balance. So what if the Balance Sheet did not balance at the model start date? As a modeller, there is nothing you can do about this: this is an opening assumption (the Opening Balance Sheet). If this were to happen to you, reject the Opening Balance Sheet and wait until someone who knows what they are doing gives you a proper one.

All a modeller may ever be held accountable for is that the change in Net Assets equals the change in Total Equity.

3. Cash Flow Statement

Let's consider a Cash Flow Statement:

Operating Cash Flows	
Cash Receipts	X
Cash Payments	X
Interest Paid	X
Tax Paid	X
Net Operating Cash Flows	X
Investing Cash Flows	
Interest Received	X
Dividends Received	X
Purchase of Non-Current Assets	X
Net Investing Cash Flows	X
Financing Cash Flows	
Debt Drawdowns	X
Debt Repayments	X
Ordinary Equity Issuance	X
Ordinary Equity Buybacks	X
Dividends Paid	X
Net Financing Cash Flows	X
Net Inc / (Dec) in Cash Held	X

The above is an example of what is known as a **direct** Cash Flow Statement. There are two forms to the Cash Flow Statements: direct and indirect. In many accounting jurisdictions it is stipulated that one variant must be displayed in the financial statements and the other should be the reconciliatory note to said accounts. It usually does not matter which way round this is done, as long as it is consistent from one period to the next.

Both variants affect the Net Operating Cash Flow section only of the Cash Flow Statement. As you are probably aware, they are defined as follows:

- **direct:** This can reconcile Operating Cash Flows back to a large proportion of the bank statements. It is a summary of Cash Receipts, Cash Paid, (sometimes) Interest Paid and Tax Paid.
- **indirect:** This starts with an element of the Income Statement and adds back non-cash items (deducting their cash equivalents) and adjusts for working capital movements.

A typical indirect Cash Flow Statement may compare to the direct version as follows:

Cash Flow Statement

Example

Go to Navigator

Error Checks:

Period start

Period end

Counter

01-Jan-20 01-Jan-21 01-Jan-22 01-Jan-23 01-Jan-24

31-Dec-20 31-Dec-21 31-Dec-22 31-Dec-23 31-Dec-24

1 2 3 4 5

Operating Cash Flow (Direct)

Cash Receipts	US\$'000	384	440	485	526	555
Cash Payments	US\$'000	(120)	(131)	(145)	(157)	(166)
Interest Paid	US\$'000	(20)	(9)	(11)	(13)	(13)
Tax Paid	US\$'000	(40)	-	-	-	(63)
Net Operating Cash Flow (Direct)	US\$'000	204	300	330	356	313

Operating Cash Flow (Indirect)

NPAT	US\$'000	161	159	157	160	115
Add back:						
Depreciation Expense	US\$'000	38	83	113	135	213
Interest Expense	US\$'000	6	7	8	6	4
Tax Expense	US\$'000	75	64	67	71	60
Movements in working capital:						
(Inc) / Dec in Current Assets	US\$'000	(16)	(8)	(7)	(6)	(4)
Inc / (Dec) in Current Liabilities	US\$'000	(0)	4	3	3	2
Deduct:						
Interest Paid	US\$'000	(20)	(9)	(11)	(13)	(13)
Tax Paid	US\$'000	(40)	-	-	-	(63)
Net Operating Cash Flow (Indirect)	US\$'000	204	300	330	356	313

As explained above, the indirect version is calculated as follows:

- Start with a line item from the Income Statement (here, Net Profit After Tax)
- Add back non-cash items (Depreciation Expense, Interest Expense and Tax Expense)
- Adjust for working capital movements (increases and decreases in Current Assets and Current Liabilities)
- Deduct the cash equivalents of the non-cash items added back:
 - Instead of Interest **Expense** deduct Interest **Paid**
 - Instead of Tax **Expense** deduct Tax **Paid**
 - Instead of Depreciation Expense, exclude as the cash effect is considered in Cashflows from Investing Activities.

So which one should you model? Most modellers will create an Income Statement and (at least extracts from) the Balance Sheet, so the indirect approach represents "simplest bestest". The issue is items double counted or omitted in the Income Statement may be missed using this approach. This is often what causes problems in Balance Sheet reconciliations.

My approach facilitates the incorporation of **control accounts** and control accounts are a financial modeller's best friends. These require you to use the direct method, but you may always calculate the indirect variant later, if required.

Model Cash Flow Statements on a direct basis.

Linking Financial Statements

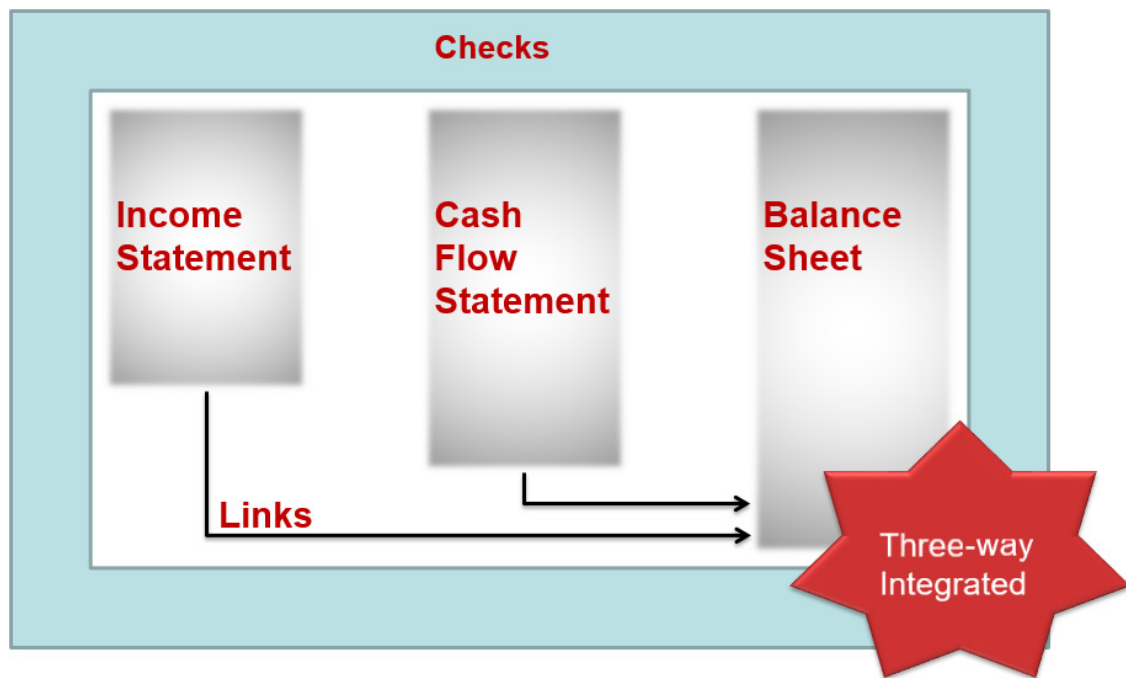
This requires linking up the financial statements. So how many do you need? One [1]? Eight? [8]? 50? The correct answer, believe it or not, is **two** [2]:

Income Statement	Balance Sheet	Cash Flow Statement
Sales Cost of Goods Sold Gross Profit Operating Expense EBITDA Depreciation EBIT Interest Expense Net Profit Before Tax (NPBT) Tax Expense Net Profit After Tax (NPAT)	Current Assets Opening Cash Net Increase/Decrease in Cash Held Cash Accounts Receivable Other Current Assets Total Current Assets Non-Current Assets PP&E Deferred Tax Assets Total Non-Current Assets Total Assets Current Liabilities Accounts Payable Interest Payable Dividends Payable Income Tax Payable Other Current Liabilities Total Current Liabilities Non-Current Liabilities Debt Deferred Tax Liabilities Total Non-Current Liabilities Total Liabilities Net Assets Equity Ordinary Equity Opening Retained Profits Net Profit After Tax (NPAT) Dividends Declared Retained Profits Total Equity	Operating Cash Flow Cash Receipts Direct Cash Payments Indirect Cash Payments Cash Payments Interest Paid Tax Paid Net Operating Cash Flows Investing Cash Flows Purchases of Capital Net Investing Cash Flows Financing Cash Flows Debt Drawdowns Debt Repayments Ordinary Equity Issuances Ordinary Equity Buybacks Dividends Paid Net Financing Cash Flows Net Increase/Decrease in Cash Held

As long as the Net Profit After Tax links into the Retained Earnings section of the Balance Sheet and the Net Increase / Decrease in Cash Held from the Cash Flow Statement links into the Current Assets section of the Balance Sheet, you have all of the links you require to put a financial model together.

Appropriate Order of the Financial Statements

Let's revisit the first image of this section:



Do you see the three statements have not been displayed as of equal size? There is a reason for this:

- **Income Statement:** this is arguably the “smallest” financial statement, albeit conceptually. We’re neither talking about the magnitude of the numbers nor the number of line items within the financial statements. We are considering how small or large the Income Statement is conceptually compared with the other statements.

The Income Statement considers the Net Operating Profit after Taxation (NPAT). The Cash Flow Statement considers Operating Cash Flows, but it also considers Investing and Financial ones too. The Balance Sheet incorporates the summary NPAT of the Income Statement, so it must also be at least as large.

- **Cash Flow Statement:** as discussed above, it considers more factors than the Income Statement (albeit from a different perspective), but since it is also summarised in the Balance Sheet (Cash), it is the ‘middle’ statement.
- **Balance Sheet:** not only does it summarise the other two financial statements, but it also details financials not captured elsewhere, *e.g.* movements between Non-Current and Current, and transfers in Reserves.

So why is this important? Earlier, we stated that when we start to build a model, in general we start to work our way down the Income Statement. That made sense and is commensurate with the magnitude of the concept of the financial statement. It also suggests that the Cash Flow Statement should be built second, which again makes sense, given the Balance Sheet includes a summary of the other two statements.

Therefore, to summarise, we should:

- develop the three financial statements, building up by line item and total
- link the Income Statement and Cash Flow Statement into the Balance Sheet
- add error checks to ensure no errors, that the Balance Sheet balances and is solvent (for example)
- model our way down the smallest financial statement (Income Statement), then what remains in the Cash Flow Statement and finally, whatever else is left in the Balance Sheet.

We have a plan!

Control Accounts

Now it's time to let you in to one of the finance world's best kept secrets. They are easy to construct and even easier to understand. Let's consider the reconciliation of the line item Accounts Receivable (or Debtors):

	\$
Accounts Receivable b/f	120,000
Sales	64,700
Cash Receipts	<u>(82,750)</u>
Accounts Receivable c/f	101,950

This reconciliation is known as a **control account**: it is a reconciliation of a Balance Sheet item from one period to the next.

Typically (although not always), the line items between the opening and closing balances come from the Income Statement and Cash Flow Statement. This is consistent with the idea that the Balance Sheet is stated at a point in time whereas the other two statements are for periods of time.

- Sales of \$64,700 in the Income Statement;
- Cash Receipts of \$82,750 (as a positive number) in the Cash Flow Statement; *and*
- Closing Accounts Receivable of \$101,950 in the Balance Sheet

In the example above, if the opening balance of Accounts Receivable is \$120,000 and we make further sales in the period of \$64,700, assuming there are no bad debts (more on that later) and the cash received is \$82,750, then the closing balance for Accounts Receivable has to be \$101,950. In other words, assuming the opening balance was \$120,000, entering:

means that the three-way integrated financial statements must balance. The end. Modelling financial statements really is that simple.

Control accounts tell you three [3] key things:

1. **Number of calculations that need to be entered into the financial statement so that they balance:** This is always one less than the number of rows in the control account. The reason it is one less is because the opening balance is simply the closing balance calculated from the period before.
2. **The order to build the calculations into the financial statements:** This is always row 2 first, then row 3, then row 4 and so on. Think of it this way: assuming no opening balance (which there would not be in the beginning), if there were no sales, there could be no payments received. If there are no sales and no receipts, the difference between them (the amount owed, the Accounts Receivable) would also be zero. It is a logical order.
3. **It identifies the key driver:** Often you want to undertake sensitivity and scenario analysis in your models, but sometimes you may be unsure which variables should be included in the analysis. Line 2 of the control account is always the key driver. As above, if there were no sales, there could be no payments received. If there are no sales and no receipts, the difference between them (the amount owed, the Accounts Receivable) would also be zero. To make a point, I have repeated myself deliberately. To make a point, I have repeated myself deliberately. To make a point, I have repeated myself deliberately...

Therefore, in our example, we can conclude:

- In order to make the Balance Sheet balance we need to construct three calculations which need to be incorporated into the financial statements: Sales, Cash Receipts and closing Accounts Receivable.
- The order to calculate them should be Sales, Cash Receipts and finally, closing Accounts Receivable.
- The key driver of Accounts Receivable is Sales.

Let's consider the following modelling example (at last – something to do with Excel):

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Calculations															
2	SP Balancing the Balance Sheet Example.xlsm															
3	Navigator															
4	Error Checks: <input checked="" type="checkbox"/>															
5										Jun 25	Jun 26	Jun 27	Jun 28	Jun 29		
6		Start Date								1 Jul 24	1 Jul 25	1 Jul 26	1 Jul 27	1 Jul 28		
7		End Date								30 Jun 25	30 Jun 26	30 Jun 27	30 Jun 28	30 Jun 29		
8		Number of Days								365	365	365	366	365		
9		Counter								1	2	3	4	5		
10																
11	1. Calculations															
12																
13	Revenue and related															
14																
15	Sales															
16																
17		Projected sales	#							100	200	300	400	500		
18		Unit price	US\$'000							15						
19		Inflation	%								3%	4%	5%	6%		
20																
21		Price per unit	US\$'000							15	15	16	17	18		=IF(J\$9=1,\$J\$18,I21*(1+J19))
22																
23		Revenue	US\$'000							1,500	3,090	4,820	6,749	8,942		=J17*J21
24																
25	Working capital															
26																
27		Days receivable	# Days							60	60	60	60	60		=Assumptions/J23
28																
29		Days in period	# Days							365	365	365	366	365		=J\$8
30																
31		Closing receivables	US\$'000							247	508	792	1,106	1,470		=J23*J27/J29
32																
33	Control account															
34																
35		Opening receivables	US\$'000							-	247	508	792	1,106	BS	=I38
36		Revenue	US\$'000							1,500	3,090	4,820	6,749	8,942	IS	=J23
37		Cash receipts	US\$'000							(1,253)	(2,829)	(4,536)	(6,435)	(8,578)	CFS	=J38-SUM(J35:J36)
38		Closing receivables	US\$'000							247	508	792	1,106	1,470	BS	=J31

There are four [4] line items in our control account (rows 35 to 38), so there are three [3] calculations included in the financial statements. Revenue (line 36) has been linked to the Income Statement:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Income Statement															
2	SP Balancing the Balance Sheet Example.xlsm															
3	Navigator															
4	Error Checks:						<input checked="" type="checkbox"/>									
5																
6		Start Date								Jun 25	Jun 26	Jun 27	Jun 28	Jun 29		
7		End Date								1 Jul 24	1 Jul 25	1 Jul 26	1 Jul 27	1 Jul 28		
8		Number of Days								30 Jun 25	30 Jun 26	30 Jun 27	30 Jun 28	30 Jun 29		
9		Counter								365	365	365	366	365		
10										1	2	3	4	5		
11	1. Income Statement															
12																
13		Revenue				US\$'000				1,500	3,090	4,820	6,749	8,942	=Calculations!J36	
14		COGS				US\$'000										
15		Gross Profit				US\$'000				1,500	3,090	4,820	6,749	8,942		
16																
17		Operating expenditure				US\$'000										
18		EBITDA				US\$'000				1,500	3,090	4,820	6,749	8,942		
19																
20		Depreciation				US\$'000										
21		EBIT				US\$'000				1,500	3,090	4,820	6,749	8,942		
22																
23		Interest Expense				US\$'000										
24		NPBT				US\$'000				1,500	3,090	4,820	6,749	8,942		
25																
26		Tax Expense				US\$'000										
27		NPAT				US\$'000				1,500	3,090	4,820	6,749	8,942		
28																

Cash receipts (row 37) has been linked (as a positive number) to the **direct** Cash Flow Statement:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Cash Flow Statement															
2	SP Balancing the Balance Sheet Example.xlsm															
3	Navigator															
4	Error Checks:						<input checked="" type="checkbox"/>									
5																
6		Start Date								Jun 25	Jun 26	Jun 27	Jun 28	Jun 29		
7		End Date								1 Jul 24	1 Jul 25	1 Jul 26	1 Jul 27	1 Jul 28		
8		Number of Days								30 Jun 25	30 Jun 26	30 Jun 27	30 Jun 28	30 Jun 29		
9		Counter								365	365	365	366	365		
10										1	2	3	4	5		
11	1. Cash Flow Statement															
12																
13		Direct Cash Flow Statement														
14																
15		Operating cash flow														
16		Cash receipts				US\$'000				1,253	2,829	4,536	6,435	8,578	=Calculations!J37	
17		Direct cash payments				US\$'000										
18		Indirect cash payments				US\$'000										
19		Cash payments				US\$'000				-	-	-	-	-		
20		Interest paid				US\$'000										
21		Tax paid				US\$'000										
22		Net Operating cash flow				US\$'000				1,253	2,829	4,536	6,435	8,578		
23																
24		Investing cash flows														
25		Interest received				US\$'000										
26		Purchases of Non-Current Assets				US\$'000										
27		Net Investing cash flows				US\$'000				-	-	-	-	-		
28																
29		Financing cash flows														
30		Debt drawdowns				US\$'000										
31		Debt repayments				US\$'000										
32		Ordinary equity issuances				US\$'000										
33		Ordinary equity buybacks				US\$'000										
34		Dividends paid				US\$'000										
35		Net Financing cash flows				US\$'000				-	-	-	-	-		
36																
37		Net increase / (decrease) in cash held				US\$'000				1,253	2,829	4,536	6,435	8,578		
38																

Finally, Closing receivables have been linked to the Balance Sheet. It all balances:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Balance Sheet															
2	SP Balancing the Balance Sheet Example.xlsm															
3	Navigator															
4	Error Checks:															
5																
6																
7																
8																
9																
10																
11	1. Balance Sheet															
12																
13	Current assets															
14	Cash															
15	Accounts receivable															
16	Other current assets															
17	Total current assets															
18																
19	Non-current assets															
22	Total non-current assets															
23																
24	Total assets															
25																
26	Current liabilities															
32	Total current liabilities															
33																
34	Non-current liabilities															
37	Total non-current liabilities															
38																
39	Total liabilities															
40																
41	Net assets															
42																
43																
44	Equity															
45	Ordinary equity															
46	Opening retained profits															
47	NPAT															
48	Dividends declared															
49	Retained profits															
50	Total equity															

That's it! Use control accounts and follow the order of calculations, working your way down the financial statements ensuring it is correctly referenced at all junctures – and you will *never* have to worry about the Balance Sheet balancing again.

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...

Given an array of numeric values, how do you produce an array of row running totals? You might be familiar with using a **SUM** function and anchoring the beginning of ranges. This month, we challenge you to do that for a whole array in one [1] formula!

The outputs should look similar to the following upon completion:

Data	4	12	(9)	(5)	(4)	9	1	(12)	(7)	-
	-	10	(11)	(12)	7	(2)	-	9	(10)	(4)
	(5)	(10)	11	(11)	4	2	(6)	11	(1)	-
	(12)	(7)	4	(8)	6	(1)	(12)	8	7	1
	(5)	(3)	(5)	7	6	8	4	(12)	1	(3)
	10	(9)	(10)	6	(1)	8	-	-	(6)	(8)
	10	4	(2)	8	9	2	4	(7)	(2)	(9)
	(8)	(9)	4	(7)	6	(6)	(6)	6	(2)	(11)
	-	4	(5)	(8)	(5)	(1)	(3)	-	(3)	9
Cumulative Sum	4	16	7	2	(2)	7	8	(4)	(11)	(11)
	-	10	(1)	(13)	(6)	(8)	(8)	1	(9)	(13)
	(5)	(15)	(4)	(15)	(11)	(9)	(15)	(4)	(5)	(5)
	(12)	(19)	(15)	(23)	(17)	(18)	(30)	(22)	(15)	(14)
	(5)	(6)	(13)	(6)	-	8	12	-	1	(2)
	10	1	(9)	(3)	(4)	4	4	4	(2)	(10)
	10	14	12	20	29	31	35	28	26	17
	(8)	(17)	(13)	(20)	(14)	(20)	(26)	(20)	(22)	(33)
	-	4	(1)	(9)	(14)	(15)	(18)	(18)	(21)	(12)
Cumulative Product	4	48	(432)	2,160	(8,640)	(77,760)	(77,760)	933,120	(6,531,840)	-
	-	-	-	-	-	-	-	-	-	-
	(5)	50	550	(6,050)	(24,200)	(48,400)	290,400	3,194,400	(3,194,400)	-
	(12)	84	336	(2,688)	(16,128)	16,128	(193,536)	(1,548,288)	(10,838,016)	(10,838,016)
	(5)	15	(75)	(525)	(3,150)	(25,200)	(100,800)	1,209,600	1,209,600	(3,628,800)
	10	(90)	900	5,400	(5,400)	(43,200)	-	-	-	-
	10	40	(80)	(640)	(5,760)	(11,520)	(46,080)	322,560	(645,120)	5,806,080
	(8)	72	288	(2,016)	(12,096)	72,576	(435,456)	(2,612,736)	5,225,472	(57,480,192)
	-	-	-	-	-	-	-	-	-	-

As always, there are some requirements:

- each formula needs to be within one cell
- the function **LAMBDA** and any of its helper functions (e.g. **LET**, **BYROW** or **MAP**) are not allowed
- this is a formula challenge; no Power Query / Get & Transform or VBA.

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

Over to AI



Here at SumProduct we write many articles for many magazines and our authoring contracts have been revised recently to ensure Artificial Intelligence hasn't taken over our duties. We can assure everyone they have not.

But it did get us thinking: *what if we did let AI loose and write a regular article for the newsletter?*

This month, we decided to consider financial modelling *per se*.

This Month's Prompt:

Some argue that AI will never be able to build a financial model, as such spreadsheets are inherently too complex and Large Language Models are not built to recognise non-sequential patterns.

Others argue that AI will learn patterns in financial models and will eventually understand how to build three-way integrated financial models.

Who is right? Why?

Introduction

The debate around whether artificial intelligence (AI), particularly Large Language Models (LLMs), can build complex financial models, such as three-way integrated financial models, has sparked interest among finance professionals and technologists alike. On one side, sceptics argue that financial models are inherently too complex for AI due to their

reliance upon non-sequential patterns and nuanced domain expertise. On the other side, proponents believe that AI's ability to learn patterns and adapt to increasingly sophisticated tasks means it will eventually master the art of building integrated financial models. This article delves into both perspectives and examines the validity of these claims.

The Sceptical Perspective

Sceptics argue that financial modelling is an intricate art and science that demands an understanding of interdependencies across income statements, balance sheets and cash flow statements. These models

often incorporate non-linear relationships, requiring subjective judgment, domain expertise and forward-thinking assumptions that LLMs may struggle to grasp.

Key arguments include:

- **Complexity of financial models:** financial models require handling non-sequential patterns, which differ starkly from the textual data that LLMs are traditionally trained on. For example, the interdependency between the financing activities on a cash flow statement and the equity section of a balance sheet is not sequentially presented but rather closely intertwined within the model

- **Subjectivity and context:** human judgment plays a critical role in financial modelling, such as choosing relevant assumptions (e.g. revenue growth rates or cost structures) and interpreting market dynamics. AI might lack the ability to replicate these subjective elements
- **Dynamic updates:** financial models are living documents, continuously refined based upon new data, regulatory changes and market conditions. AI models might struggle to adapt to such dynamic environments without explicit programming and real-time contextual understanding.

The Optimistic Perspective

Proponents of AI argue that the technology's capabilities are evolving rapidly and its ability to learn patterns and integrate complex systems makes it well-suited for mastering financial modelling over time. AI has

already demonstrated remarkable success in domains involving pattern recognition, automation and predictive analysis, which are foundational to financial modelling.

Key arguments include:

- **Pattern recognition and data integration:** AI excels at finding patterns in large datasets, which is a core activity in financial modelling. For instance, given a dataset containing historical financial statements, AI can identify correlations and trends, which could aid in automating the creation of three-way models
- **Training on domain-specific data:** by training LLMs with large datasets of existing financial models, combined with additional domain knowledge, AI could learn the interdependencies between financial statements and understand how to build integrated models
- **Improvement through reinforcement learning:** AI can improve iteratively through reinforcement learning, refining its approach to financial modelling with feedback and corrections from finance professionals
- **Success in adjacent domains:** AI has already proven capable of creating financial dashboards, automating bookkeeping tasks and generating financial projections based upon historical data. These successes suggest the possibility of mastering the complexities of financial modelling in the future.

Who Is Right?

Both perspectives bring valid points to the table. The sceptics are correct in highlighting that financial modelling involves subjective judgment and complexity that extend beyond pattern recognition. However, the optimistic perspective underscores AI's adaptability and learning capabilities, which are advancing at an unprecedented pace.

Current limitations:

- As of now, AI faces significant challenges in building sophisticated models autonomously:
 - Lack of domain-specific intuition and subjective decision-making
 - Difficulties in understanding non-sequential relationships inherent in financial data
 - Challenges in incorporating contextual and real-time updates into models.

Future potential:

- Despite these limitations, the trajectory of AI development suggests that it may eventually overcome these obstacles:
 - Specialised training datasets could help AI understand financial nuances
 - Collaborative tools that pair AI with human oversight could leverage the strengths of both
 - Advances in AI's ability to handle symbolic reasoning and causal inference could improve its understanding of non-sequential patterns.

Conclusion

The question of whether AI will ever build complex financial models is not a matter of "if" but "when". While AI is not yet equipped to independently create models with the intricacy and judgment of a seasoned finance professional, its capabilities are expanding. As AI evolves, the integration of technology into financial modelling may shift from being a tool for

automation to a partner in strategy and decision-making. For now, the balance lies in harnessing AI's strengths while recognising its limitations, allowing human expertise to guide its development toward mastering financial modelling.

Do you agree that AI will build financial models? We asked AI to quite to build a financial model with only 20 unique formulae in it – and it suggested it would take four to six weeks to build!! Send us your comments and ideas for questions to contact@sumproduct.com.

Excel for Mac

Each month, we'll cover a different topic to help you understand how Excel for Mac is different than Excel for Windows. This month, in our series about Microsoft Excel for Mac, we continue on our topic about the

Filter dialog, discussing a missing option and the different way to apply text filters. We also have a tip about wildcards.

If you use Excel for Mac, you're undoubtedly going to filter your data at some point and it's important to know how the Filter dialog works, including what limitations you have to deal with.

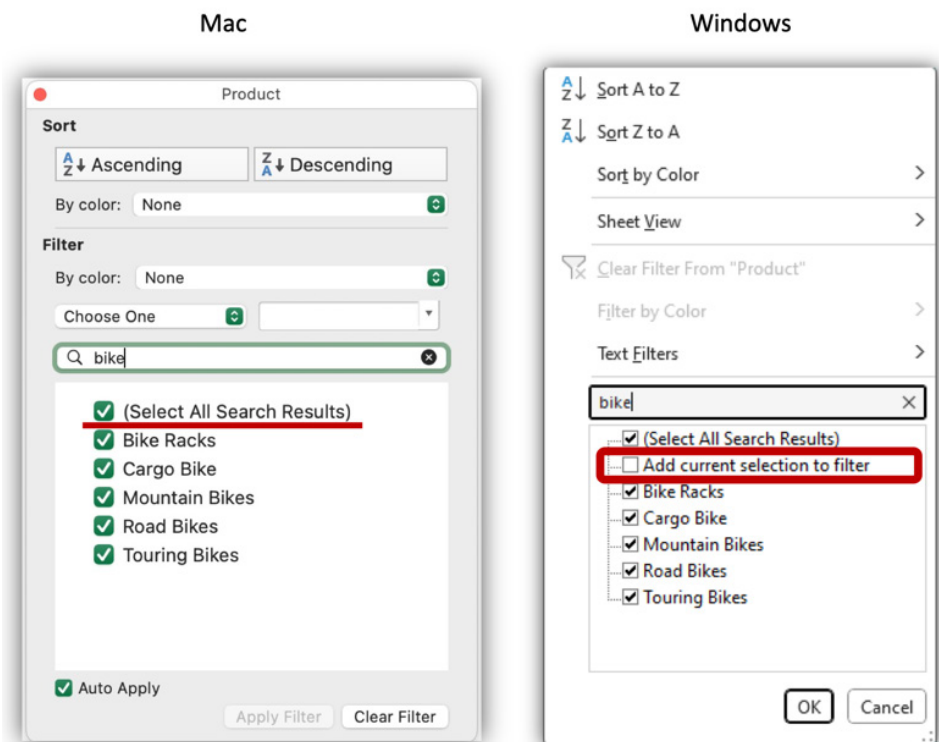
Missing from Mac: Add Current Selection to Filter

If you have a long list of values in your Filter dialog, you can use the search field rather than scrolling through the entire list. This makes it easy to find and select what you want. Unfortunately, there's one handy capability that's missing on Mac: the ability to add current selections to the filter. It's found on Windows, as in the screenshots below. On Mac,

you're limited to filtering with selections that are shown in the dialog. If you use the search feature, then you can only filter for items that match your search. It's more flexible on Windows, because you can use the search filter, select some items, and choose 'add current selection to filter', then do a different search, and add more items to the filter.

For example, if you had a list of products and you searched for “bike”, you could add any items that contain the word “bike” to the filter. Then you could search for “chain” and add it to the filter. On Mac, if you want to select both “bike” and “chain” at the same time, the best you could do is to search for “i”, which is the only common letter.

In the screen shots below, we show the Filter dialog from Mac and Windows, indicating that ‘Add current selection to filter’ is only available on Windows.

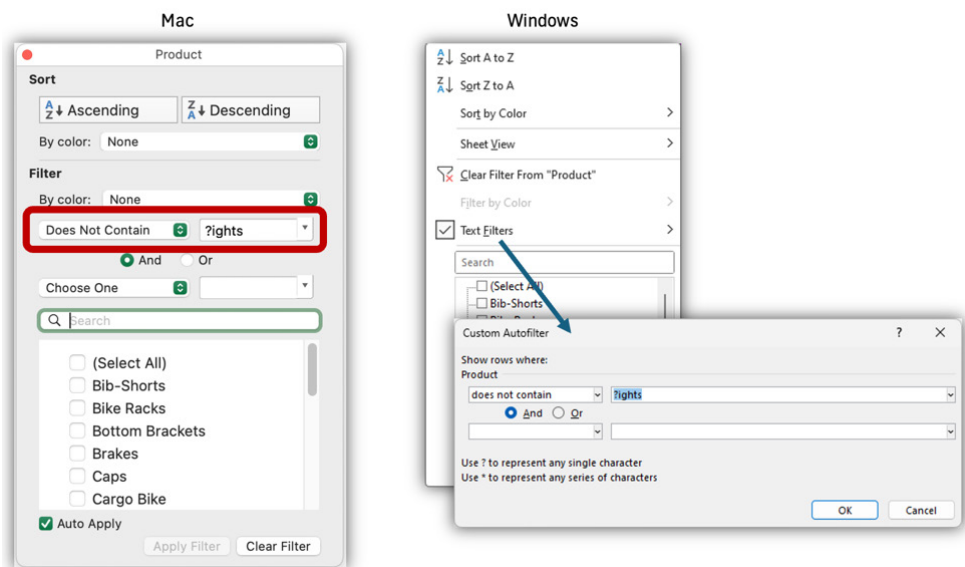


Text filtering

You can apply text filters on both Mac and Windows, but it’s slightly different. As shown in the screenshots below, the text filtering options are found in the Filter dialog on Mac, but on Windows, they’re found in the ‘Custom Autofilter’ dialog, which you can open from the Text Filters menu in the main dialog. Filtering works the same once you’ve set up your criteria.

One important thing to note is that there are some instructions about how to use wildcards, but the instructions aren’t shown on Mac. You just need to know that you can use wildcards, as below:

- use ? to represent any single character. For example, if you want to remove “lights” and “tights” from your list, you could use the criteria “Does not contain: ?ights”
- use * to represent any series of characters.



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 continue our short series on some of the common traps we sometimes fall for.

This month, we're going to highlight a common issue that follows on from last newsletter's macro. In last month's article, we copied and pasted a cell value into a new location:

	A	B	C	D	E
1	Amy	80000		Total Salary	180000
2	Billy	60000			
3	Charlie	40000			
4	Total	180000			

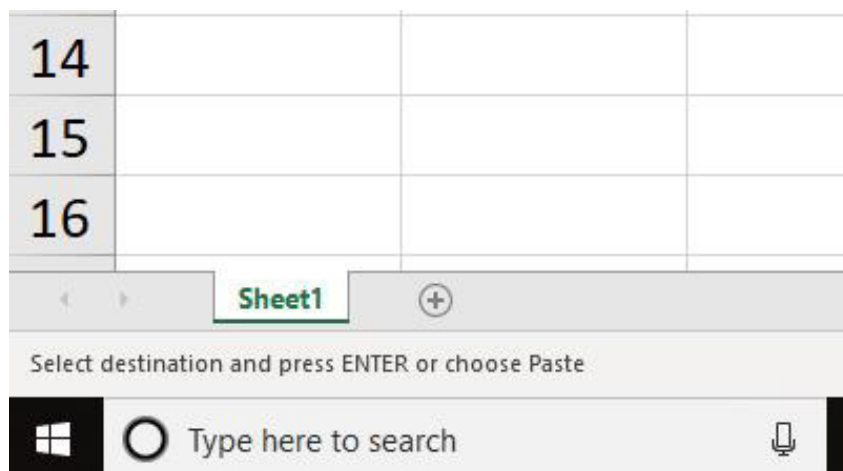
Of course, once the macro finishes running, we might click somewhere on the sheet and hit **ENTER** to move down a row:

	A	B	C	D	E
1	Amy	80000		Total Salary	180000
2	Billy	60000		#REF!	
3	Charlie	40000			
4	Total	180000			

Goodness, what's happened here? If we go back to the moment just before we hit Enter, you might notice something about the image:

	A	B	C	D	E
1	Amy	80000		Total Salary	180000
2	Billy	60000			
3	Charlie	40000			
4	Total	180000			

See the dotted green border around the cell **B4**? Often referred to as the "dancing ants" colloquially, this indicates that a cell has been cut or copied or is otherwise on the clipboard for use. Zooming out further, we might have seen the note in the bottom left corner of the screen:



This is a common problem when macros are written, that clean-up isn't properly done once the macro finishes, so people may be caught unaware by what is left over in the clipboard. How do we avoid this? After pasting our result, we simply need to use the line:

```
Application.CutCopyMode = False
```

Thus, we end up with the code:

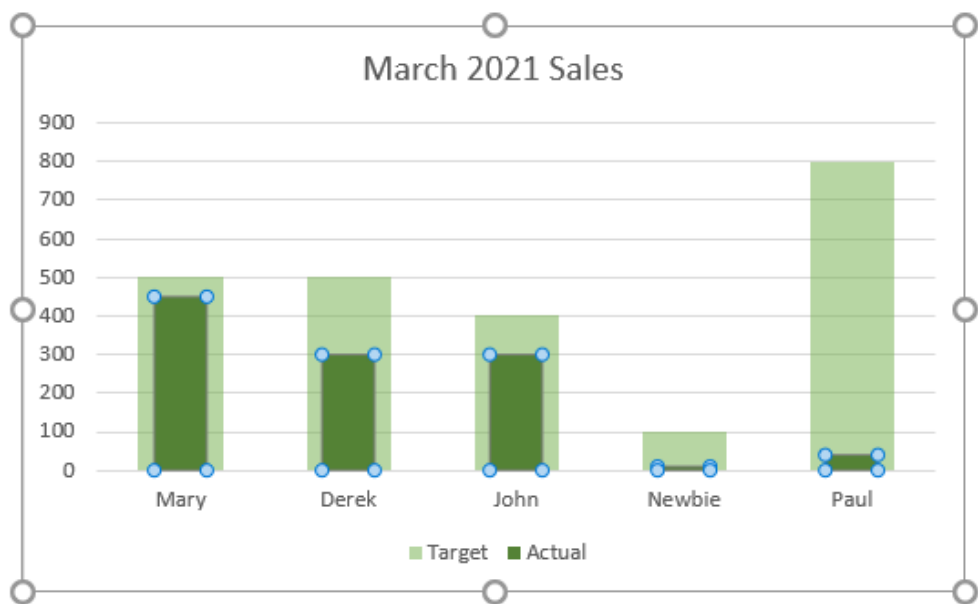
```
Sub Macro1()
'
' Macro1 Macro
'
Range("Salary_Copy").Select
Selection.Copy
Range("Salary_Paste").Select
Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False
Application.CutCopyMode = False
End Sub
```

More next month...

Charts and Dashboards

It's time to chart our progress with an introductory series into the world of creating charts and dashboards in Excel. This month, we look at another way to create a presentation of actual sales vs. target sales.

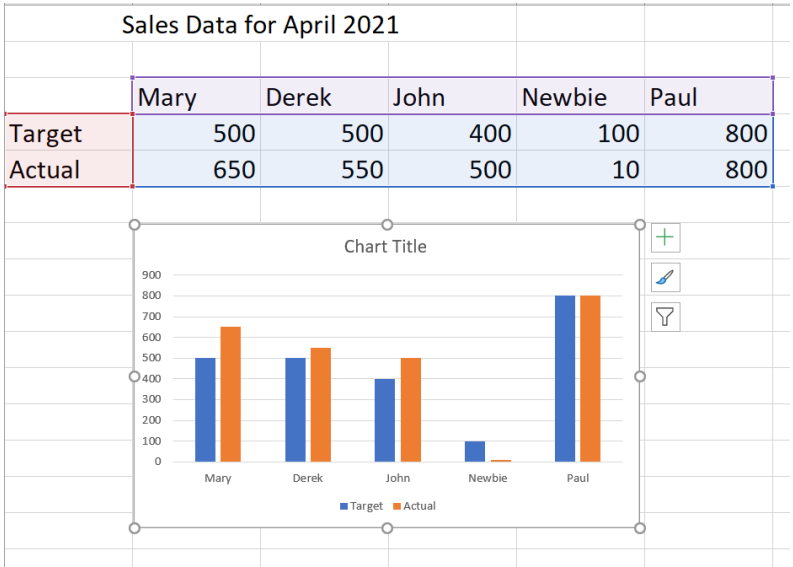
Last time, we looked at a way to represent the actual and target sales on a chart, viz.



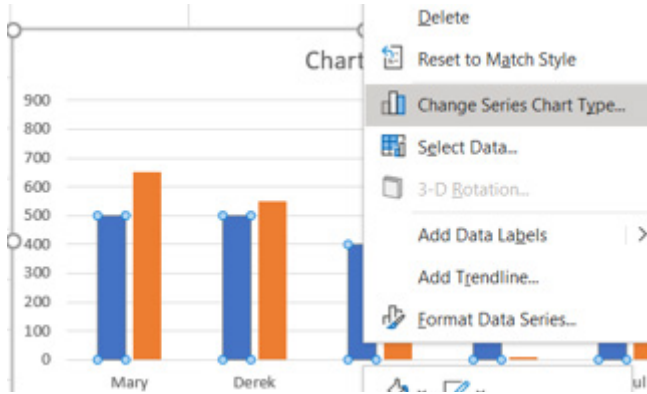
In April, our imaginary salespeople did much better (on the whole):

	A	B	C	D	E	F
1	Sales Data for April 2021					
2						
3		Mary	Derek	John	Newbie	Paul
4	Target	500	500	400	100	800
5	Actual	650	550	500	10	800
6						

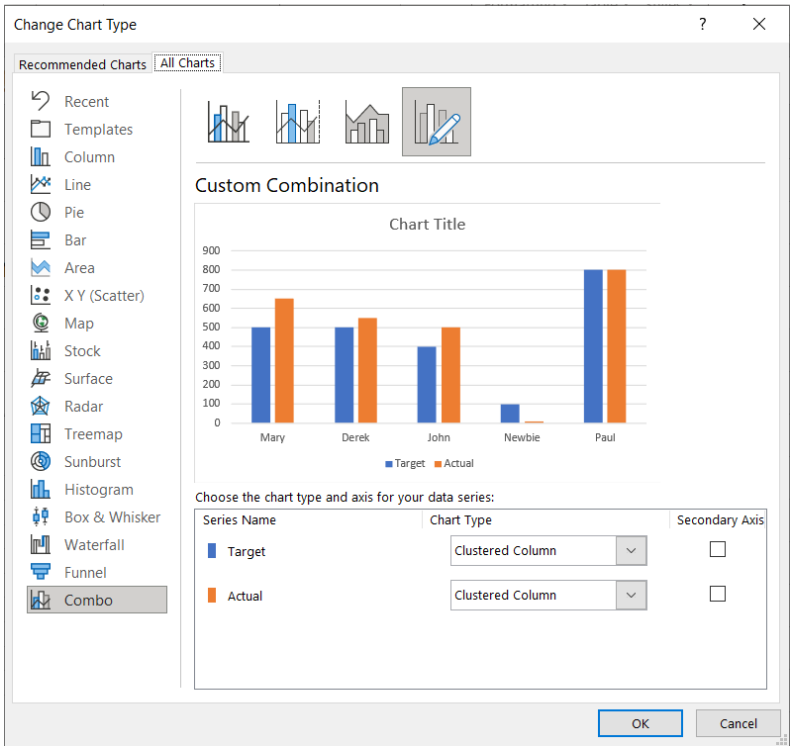
This time, we are going to present the data in a slightly different way. We will start the same way as last time, by selecting my data and pressing **ALT + F1**.



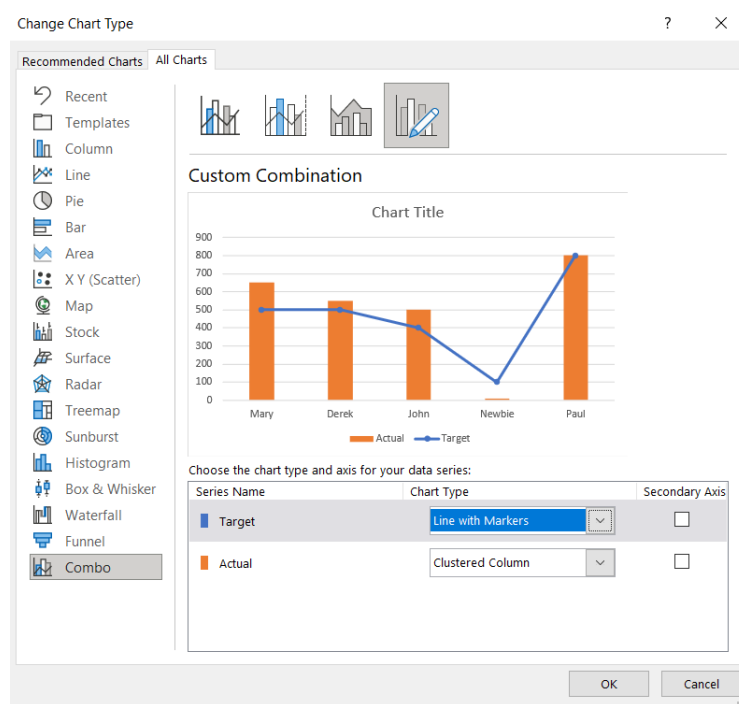
A Column chart is inserted in the sheet, and we may click on a bar from either of the data series and right click to access the menu. This time, let's select 'Change Series Chart Type'.



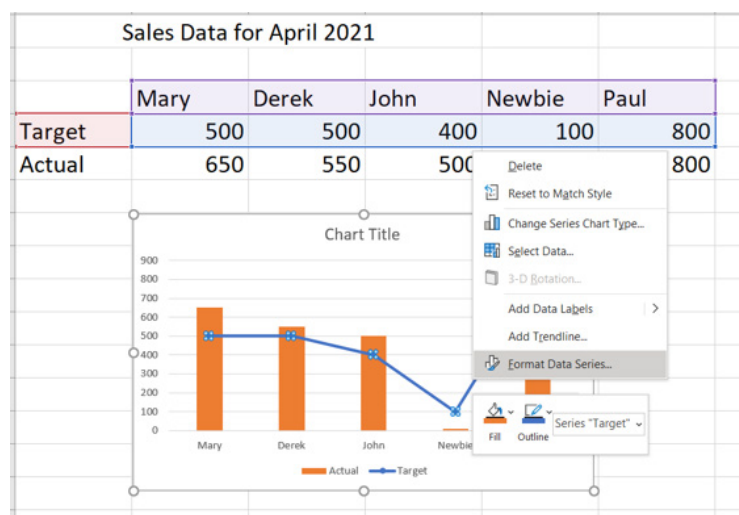
A dialog box appears:



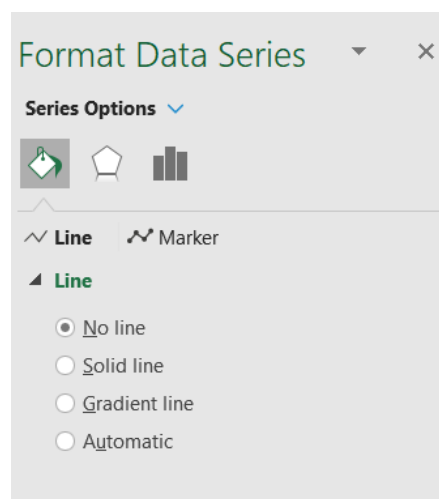
We will change the **Target** data series to 'Line Chart with Markers'.



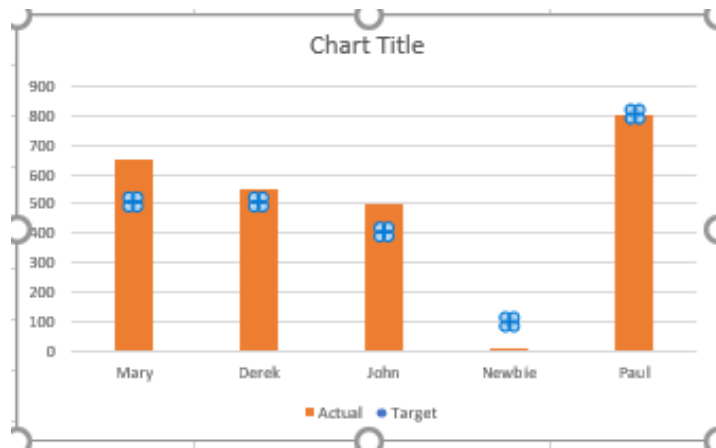
So far, this doesn't appear to make anything clearer, but we have more changes to make. Let's click 'OK' and select the Target line and right-click to access the 'Format Data Series' pane.



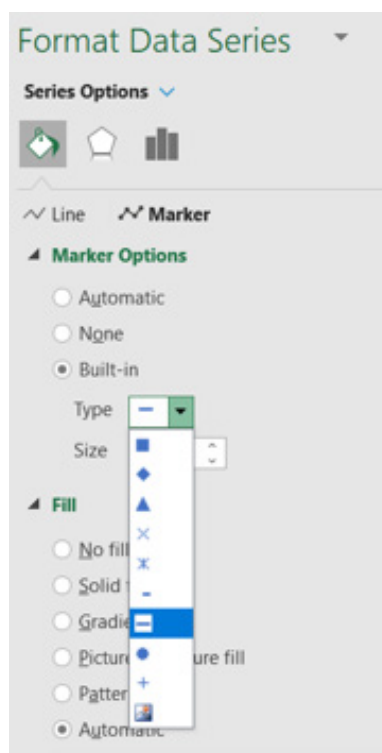
In the 'Fill and Line' tab, I select 'No Line'.



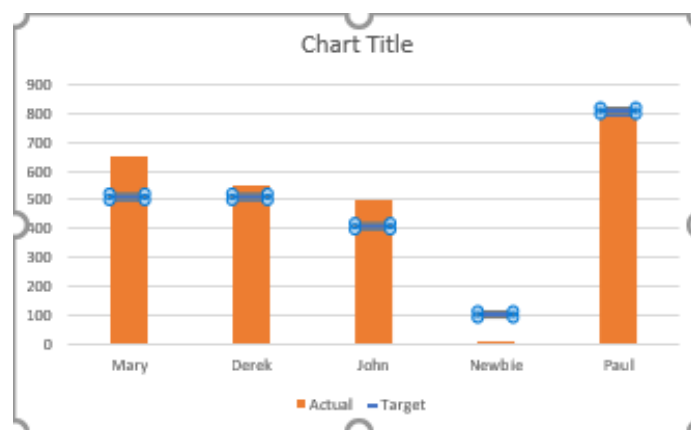
This replaces the line with markers.



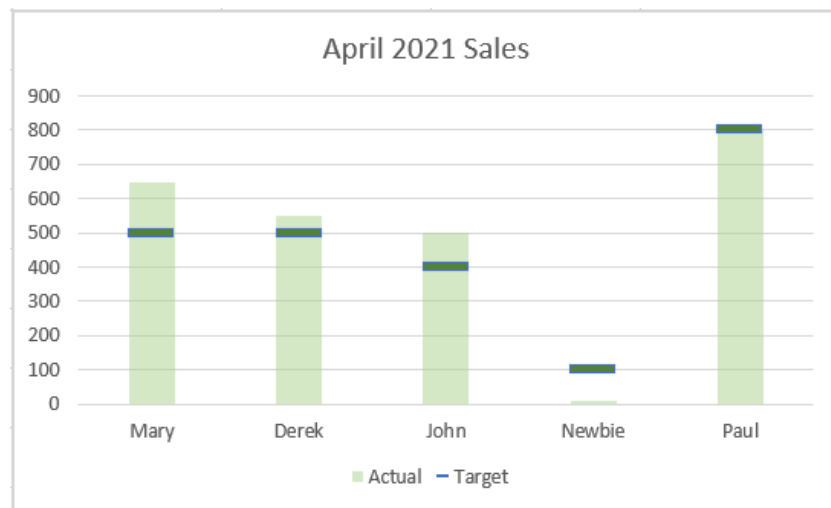
Next, we will need to format the Markers on the 'Format Data Series' pane. Let's select 'Built-in' and pick the long horizontal line. We will set the size of the line to 20.



This changes the look of our chart.



We may now change the colours in the 'Format Data Series' pane for each series and edit the chart title.



Now, we can quickly see who is exceeding their targets. Time for a chat with Newbie!

More next month.

Power Pivot Principles

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

The **ISFILTERED** function uses the following syntax to operate:

ISFILTERED(ColumnName)

The **ColumnName** parameter is a column that currently exists in a table that is currently loaded in the Data Model (*i.e.* it is a pre-existing field).

The **ISFILTERED** function returns with a TRUE value when the **ColumnName** parameter is filtered in the PivotTable (by definition, all simple, non-total rows and columns in a PivotTable are filtered by a given context). If there is no filter related to the **ColumnName**, the function will return with FALSE.

Let's consider an example:

Product	Sales	Weight
Coffee	5	0.1
Rabbit Skin	10	0.1
Big Game Meat	30	0.5
Small Game Mea	25	0.3

After loading the Table above into our data model, we create the following PivotTable:

	A	B	C	D	E	F
1						
2						
3		Row Labels	Sum of Sales	Sum of Weight	Weighted Sales	
4		Big Game Meat	30	0.5	15.00	
5		Coffee	5	0.1	0.50	
6		Rabbit Skin	10	0.1	1.00	
7		Small Game Mea	25	0.3	7.50	
8		Grand Total	70	1	70.00	
9						
10						
11						
12						
13						
14						

PivotTable Fields

Active All

Choose fields to add to report:

Search

ProductSales

☒ Product

☒ Sales

☒ Weight

☒ Weighted Sales

Drag fields between areas below:

Filters

Columns

Σ Values

Rows

Product

Σ Values

Sum of Sales

Sum of Weight

We simply want to create a 'Weighted Sales' column that will multiply the sale value with the corresponding weights. However, upon close inspection the 'Grand Total' is returning with the value of '70'. This is incorrect: it should rightfully evaluate to 24 (15 + 0.5 + 1 + 7.5 = 24).

The solution is to use a combination of the **ISFILTERED**, **IF** and **SUMX** functions:

Measure

Table name: ProductSales

Measure name: ISFILTERED Total

Description:

Formula: `=IF(ISFILTERED(ProductSales[Product]), SUM(ProductSales[Sales])*SUM(ProductSales[Weight]), SUMX(ProductSales, ProductSales[Sales] * ProductSales[Weight]))`

Formatting Options

Category: Number

Format: Decimal Number

Decimal places: 2

☒ Use 1000 separator (.)

OK Cancel

The **IF** function will calculate the weighted totals of our sales when the **ISFILTERED** function returns with TRUE for each row that contains a Product. In this PivotTable, **ISFILTERED** will return with TRUE for each of the product rows and will return with FALSE on the 'Grand Total' row.

The **IF** function will then evaluate the **SUMX** portion of the measure instead. This will properly calculate the total weighted sales in our PivotTable.

	A	B	C	D	E	F	G
1							
2							
3		Row Labels	Sum of Sales	Sum of Weight	Weighted Sales	ISFILTERED Total	
4		Big Game Meat	30	0.5	15.00	15	
5		Coffee	5	0.1	0.50	0.5	
6		Rabbit Skin	10	0.1	1.00	1	
7		Small Game Mea	25	0.3	7.50	7.5	
8		Grand Total	70	1	70.00	24	
9							
10							
11							
12							
13							

PivotTable Fields

Active All

Choose fields to add to report:

Search

☒ Product

☒ Sales

☒ Weight

☒ f_x Weighted Sales

☒ f_x ISFILTERED Total

Drag fields between areas below:

Filters

Columns

Values

Rows

Product

Sum of Sales

Sum of Weight

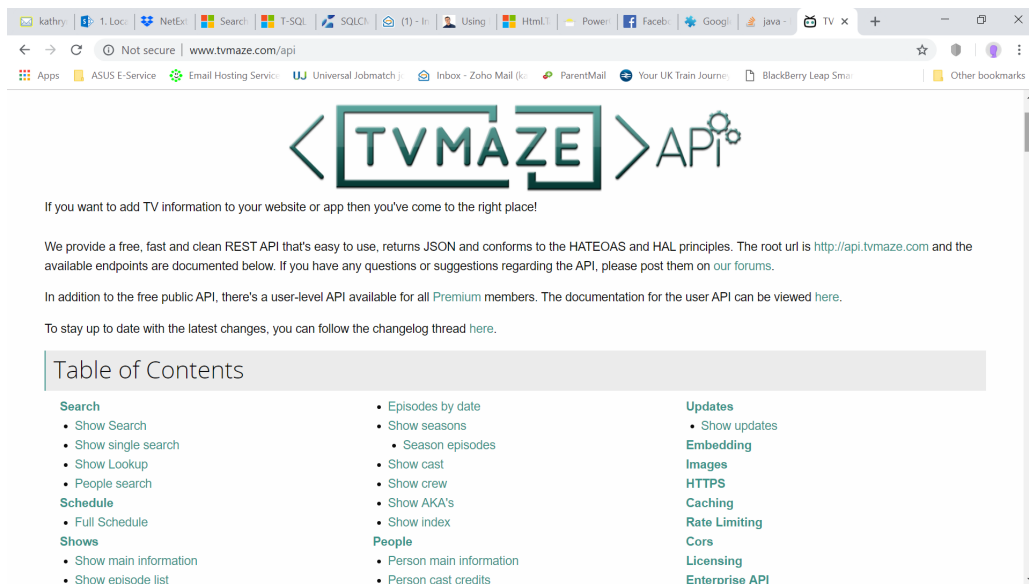
There you have it – how to properly calculate weighted totals in a PivotTable.

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

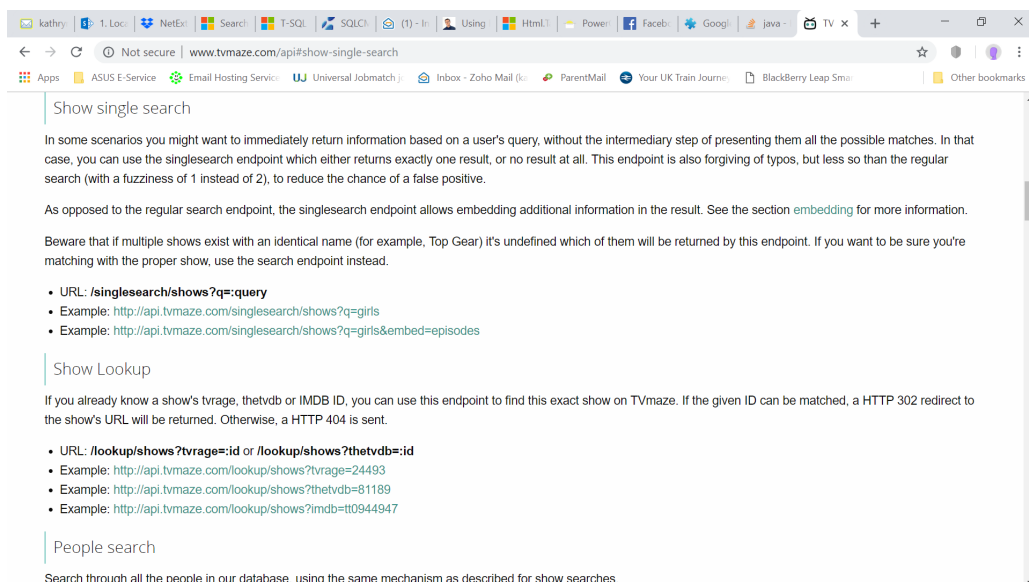
Power Query Pointers

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

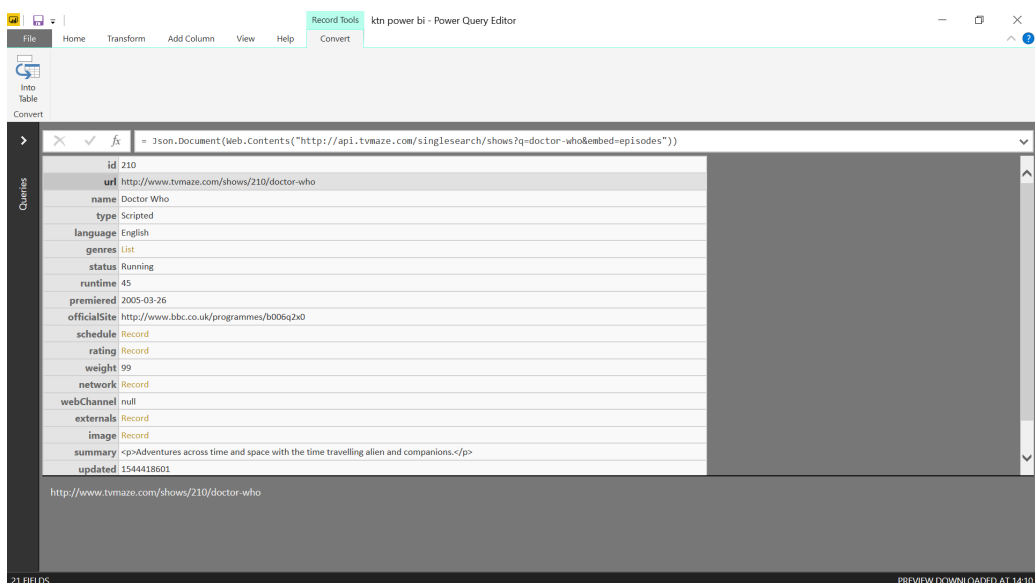
Last newsletter, we looked at viewing links in Power BI Power Query. This month, we will stay in Power BI, as it enables to create a complex **JSON** record from a webpage. The site we are going to use has a page www.tvmaze.com/api, which is designed to output **JSON** data for use in applications.



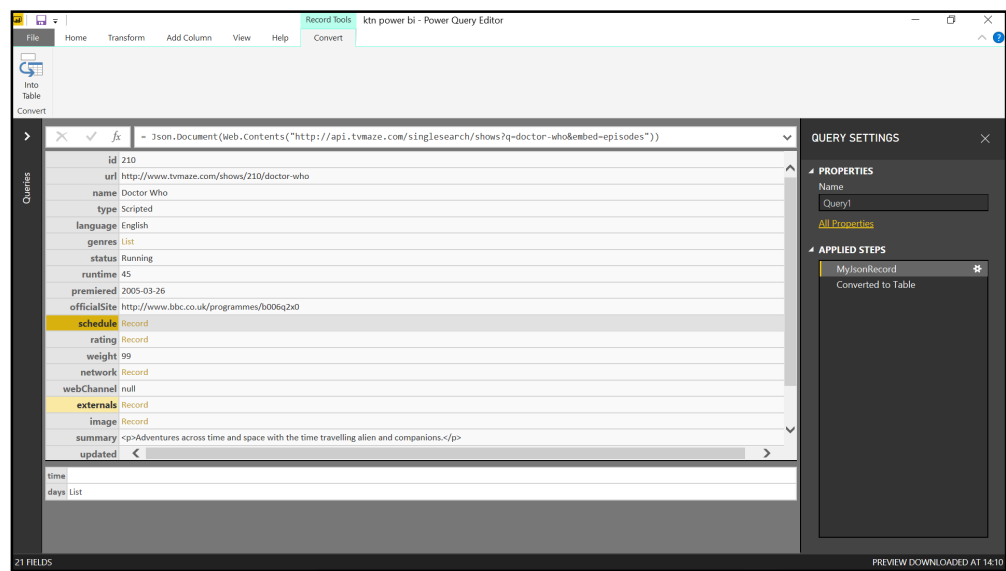
We will use the ‘Show single search’ option.



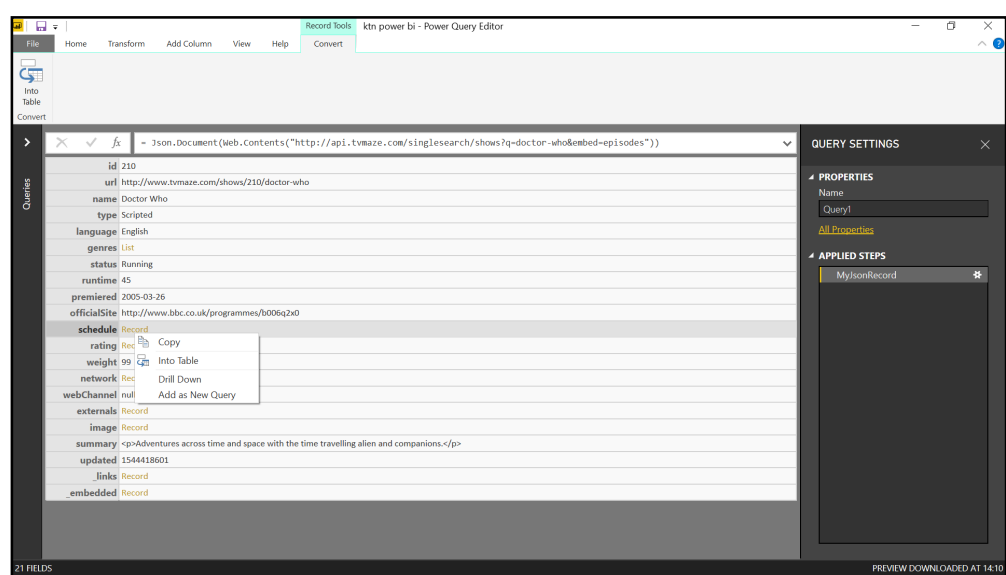
In particular, we are going to search for ‘Doctor Who’ (yup, we’re all nerds here):



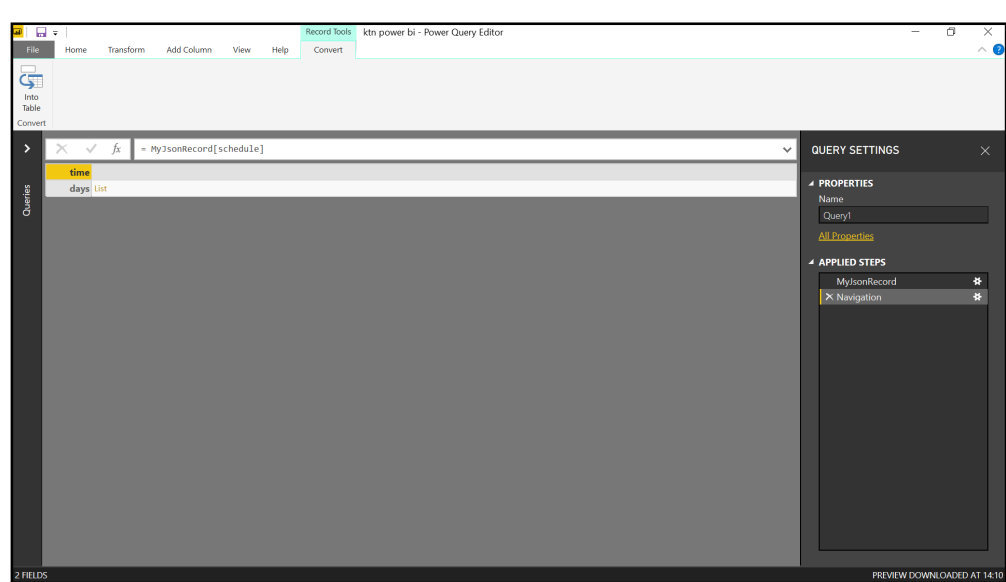
This record is quite detailed, so we can try expanding the information. It's easy enough to view it:



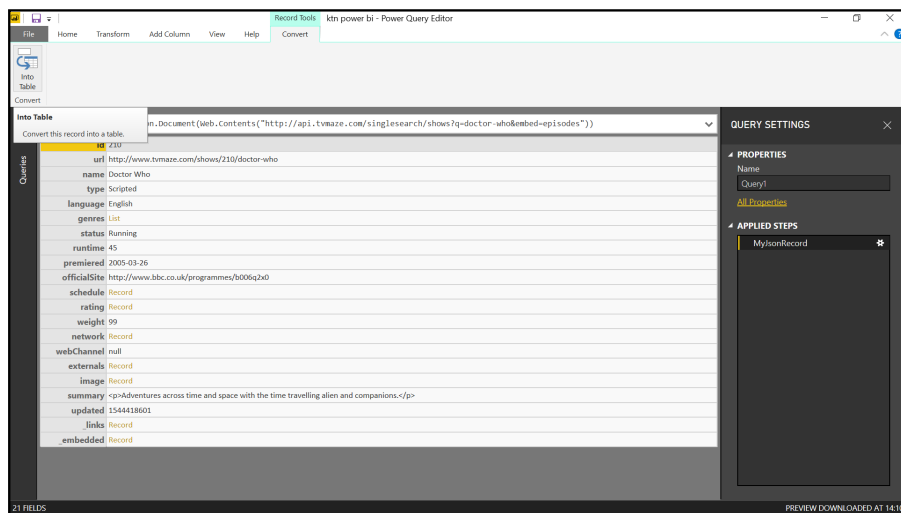
However, if we wish to expand the data so we may use it, we encounter problems:



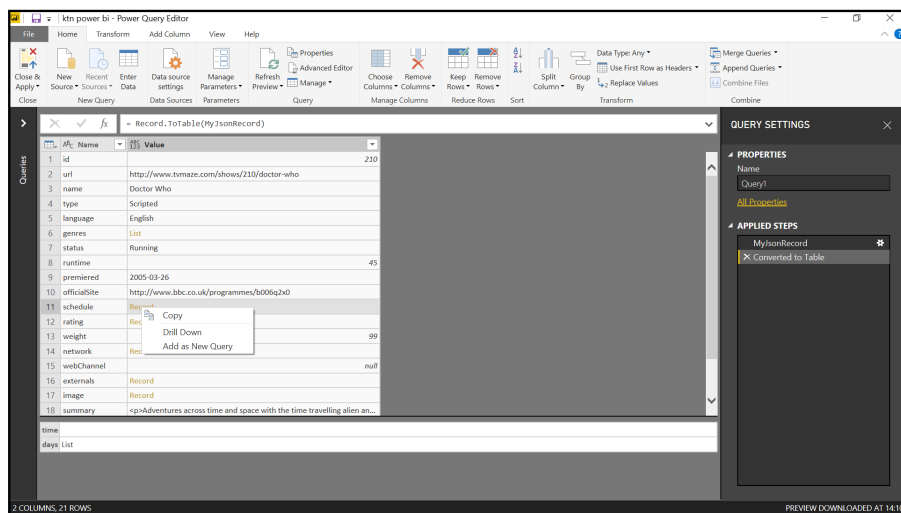
If we 'Drill Down' or double click on 'Record' then we are taken out of our main data:



We could try the option available in the 'Record Tools' tab, to convert to a record.



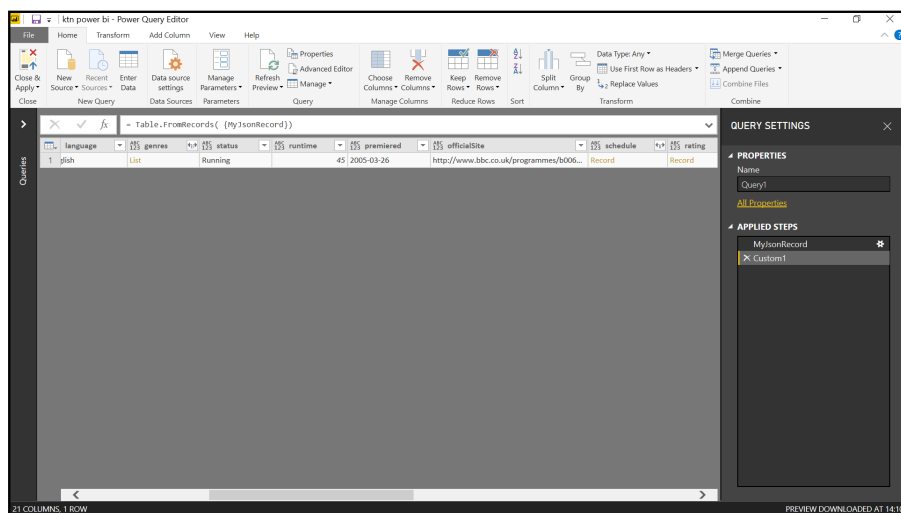
However, this doesn't help much:



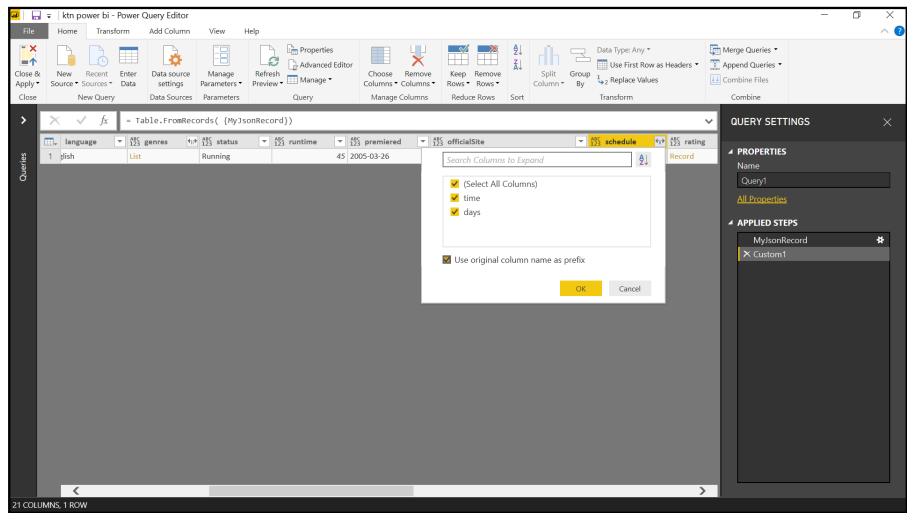
We are still taken away from the rest of our data in order to expand the record, and we'd need to do more work to keep the data in one table. There is another approach, using the **M** code

Table.FromRecords({ MyJsonRecord })

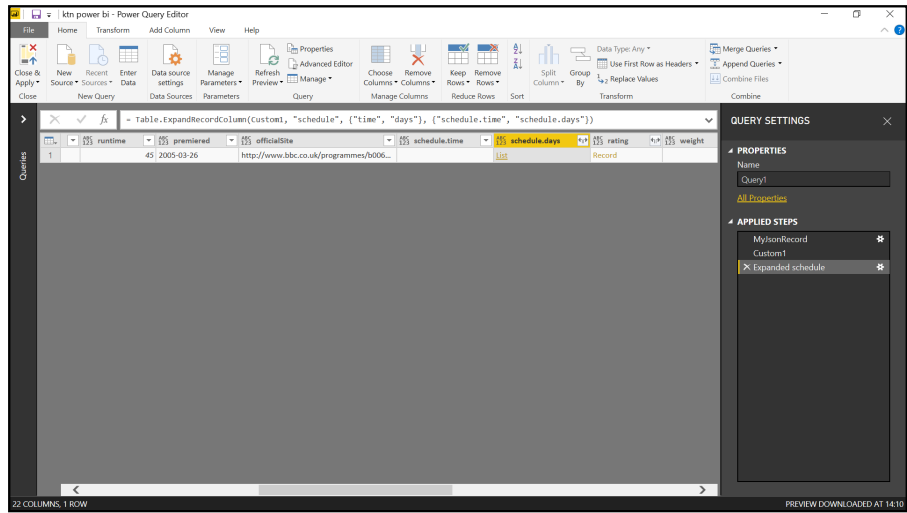
This creates a table from the list of **JSON** data:



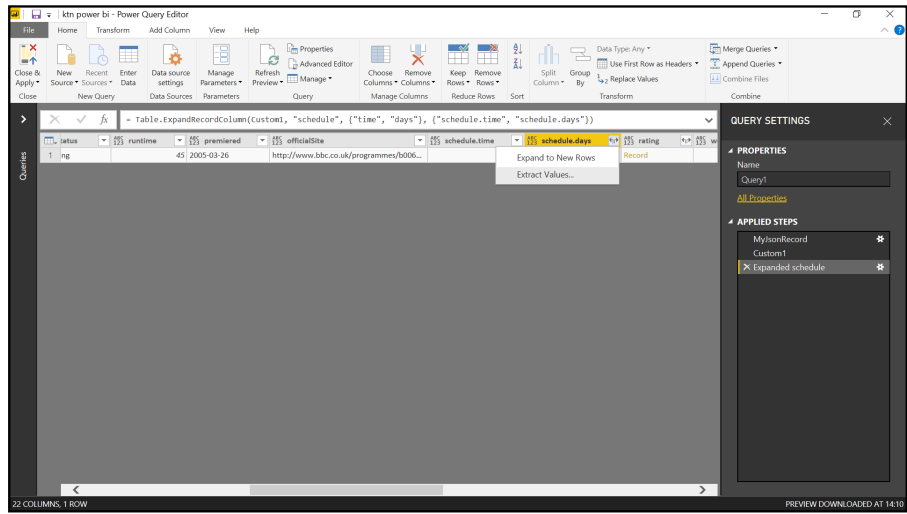
This is much more promising, as we may expand the **schedule** column and keep the rest of the data.



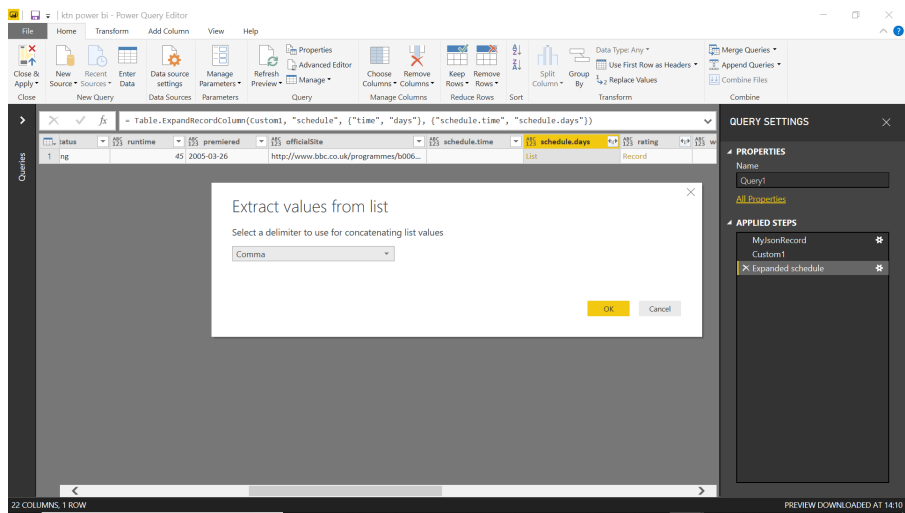
This is one time when ‘Use original column name as prefix’ is useful!



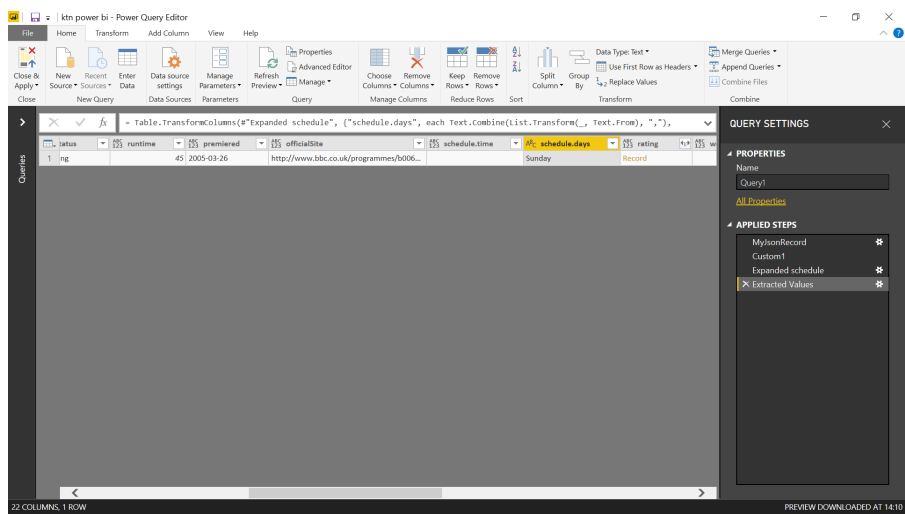
Now we may extract the list of **schedule.days**.



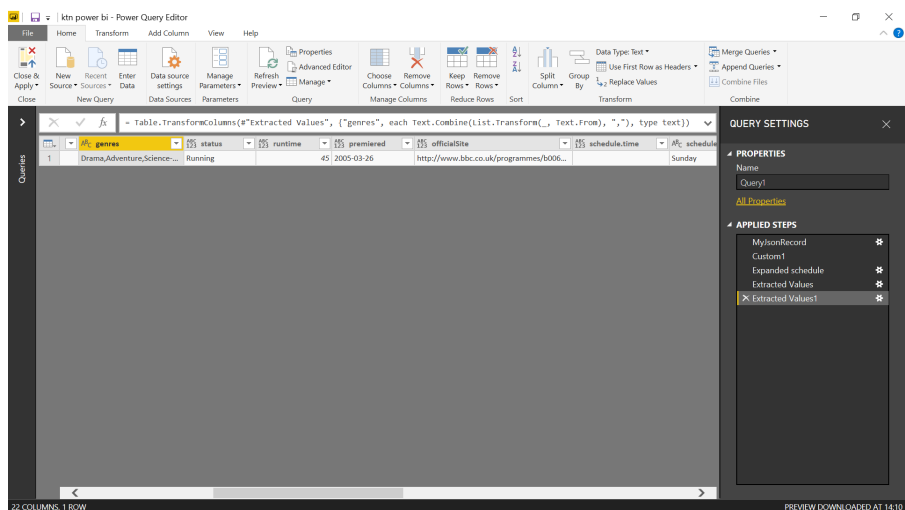
Let’s choose to extract the values.



We will choose to separate the **schedule.days** with a comma.



We can view it on Sundays. We may expand all the data held in records and lists so that we have all the information in one table.



We now have all the information we need in one place.

Until next month.

Power BI Updates

The latest update introduces a range of exciting advancements to Power BI, including a standalone Copilot feature allowing users to “Ask Anything!” in Preview. Something called “translytical task flows” are introduced to assist with automation in Power BI too. The Azure Maps visual now has enhanced base map customisation. Yup, it’s all mapped out this month...

The full list of updates is as follows:

General

- Power BI Desktop 32-bit build deprecation update

Copilot and AI

- Standalone Copilot in Power BI: Ask Anything! (Preview)
- Quality improvements to language understanding for data questions
- Get your data AI ready (Preview)
- AI Data Schema
- Verified Answers
- AI instructions
- Testing with the skill picker
- Preparing a semantic model for AI
- Mark a semantic model as prepared for AI

Reporting

- Automate report actions and data writeback with Translytical task flows (Preview)
- Persisted sorting for field parameters (Preview)
- Easily refer to values in your visual with visual calculations (Preview)
- Updates to the new List slicer (Preview)
- Paste into List and Button slicers (Preview)
- Customise line segment colours in Line charts
- Additional Base Map customisation options for Azure Maps visual
- Upcoming Azure Maps tenant settings and layer ordering updates
- Upcoming changes to the Power BI application in Teams Navigation menu

- Visual updates for semantic models with discourage implicit measures enabled

Modelling

- Selection Expressions for calculation groups
- TMDL view enhancements
- Direct Lake relationship improvements
- Direct Lake and Import tables in the same semantic model

Data Connectivity

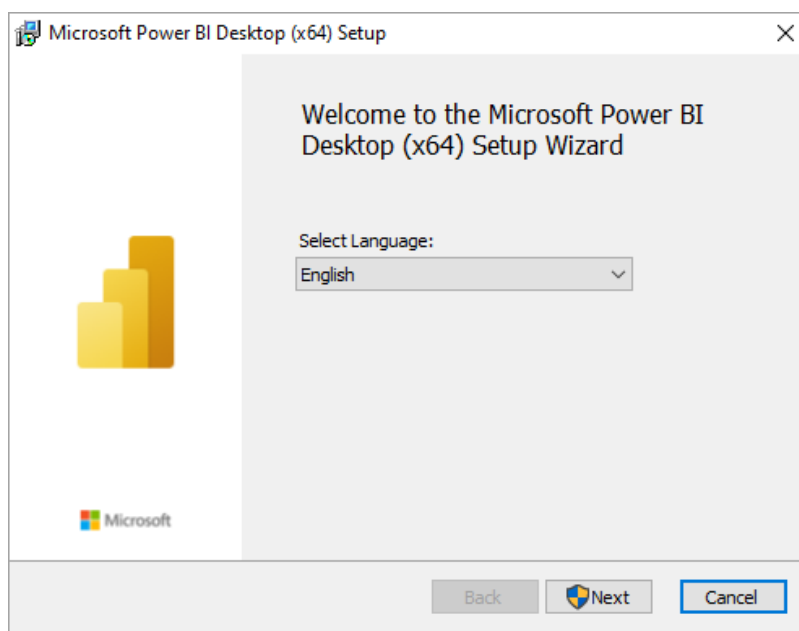
- New Google BigQuery connector implementation (Preview)
- Connect to the Vertica database with the user-installed ODBC driver – updates (Preview)
- Connect to the Oracle database with the built-in driver – updates (Preview)
- Improvements in the Snowflake connector (Preview)

Visualisations

- Dandelion chart
- Dynamic Radial Bar chart by JTA
- Action dots: an innovative way to visualise variances in Zebra BI Tables
- KPI by Powerviz
- Drill Down Timeline PRO by ZoomCharts
- PowerGantt chart by Nova Silva
- Profitbase Gantt revamped.

Let’s look at each in turn.

Power BI Desktop 32-bit build deprecation update



As part of Microsoft’s ongoing efforts to enhance the performance and security of Power BI, the company has announced the deprecation of the 32-bit build of Power BI Desktop. Starting in August 2025, there will no longer be any 32-bit builds for Power BI Desktop supported.

The main headlines are as follows:

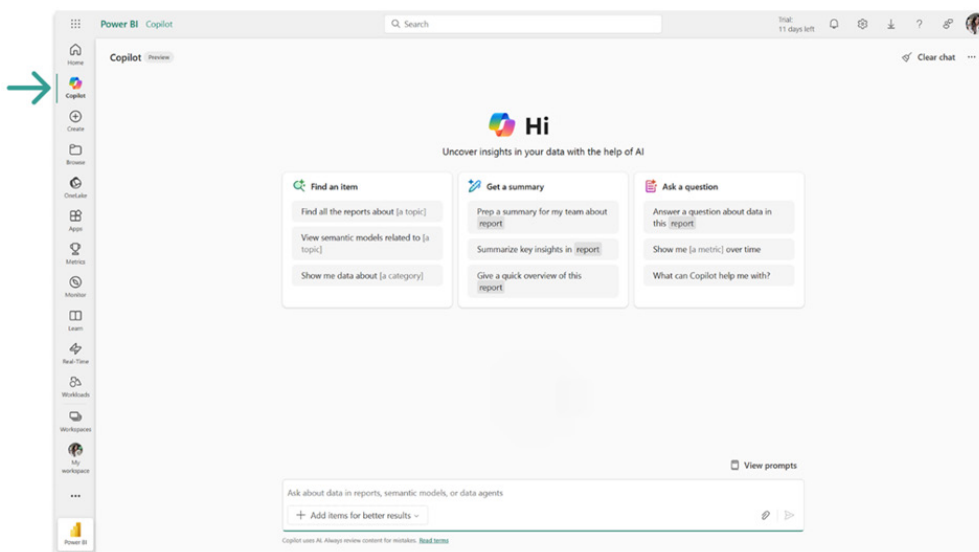
- **Deprecation timeline:** the 32-bit build will be deprecated in the August 2025 release. Customers using the 32-bit build are encouraged to transition to the 64-bit build to continue receiving updates and support
- **Benefits of transitioning to 64-bit:** the 64-bit build provides enhanced performance, better security and support for the latest features and improvements in Power BI Desktop.

It is recommended all users to make the switch to the 64-bit build to take full advantage of these benefits. It can still work in an otherwise 32-bit environment.

Standalone Copilot in Power BI: Ask Anything! (Preview)

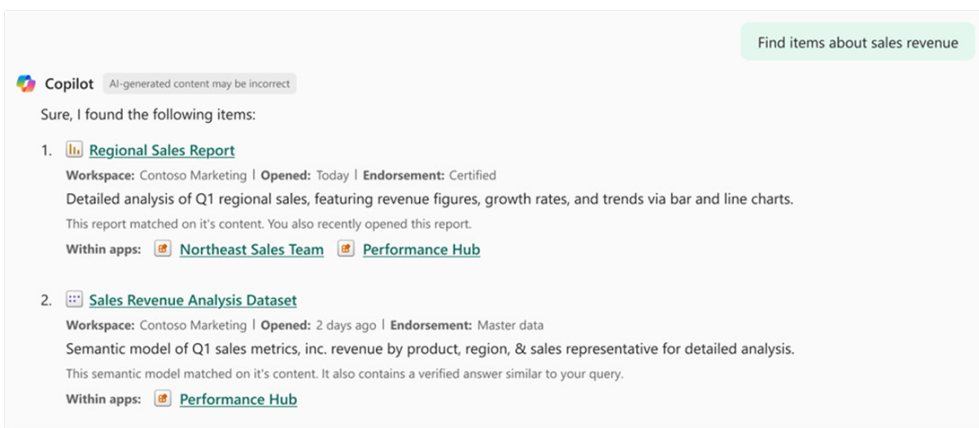
There is now a new standalone Copilot in Power BI. The intrusive AI system clearly knows no bounds (I am especially enjoying the Notepad version). This full-page Copilot experience allows users to find and analyse reports, semantic models, applications and data agents they have access to. Unlike the Copilot pane, which only lets you ask

questions about the report you currently have opened, the standalone Copilot Pane lets you ask questions about any data you have access to and attempt to provide an answer. It is noted that it will respect all existing permissions, including Row Level Security (RLS). This experience will be available in Power BI Service in the upcoming weeks.

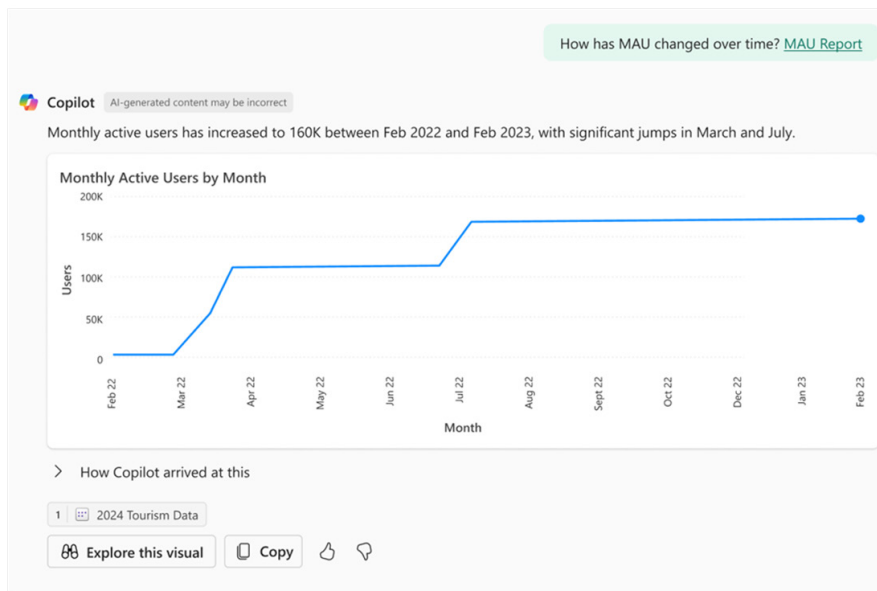


Capabilities include:

- **Search:** ask Copilot to find reports, semantic models, applications and data agents that you have access to. Power BI will return a hyperlinked list of relevant items



- **Ask questions about your data:** Copilot can answer questions and create visuals using the measures and other data fields in your model and it can also create new **DAX** calculations. You can link a report or model to your question, with application functionality coming soon



Alternatively, simply type your question and Copilot will locate the relevant data for you. Simply choose one of the suggested report buttons at the bottom of the screen, and it will generate the answer based upon your selection

Show visits over time in Hawaii

Copilot AI-generated content may be incorrect

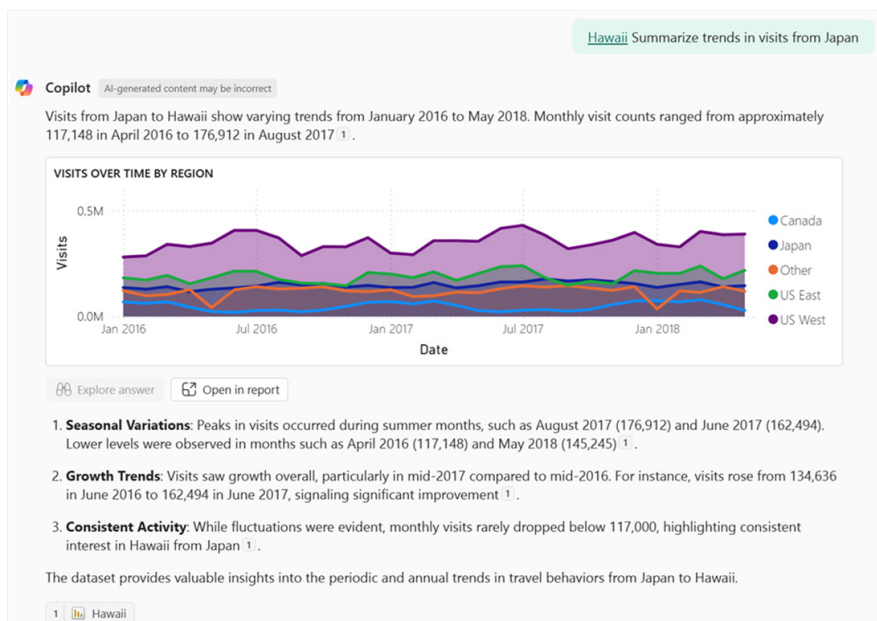
I found several items that could help answer your question.

- [Hawaii Tourism Verified Answers](#)
Workspace: PBIFabricDoNotDelete
Report containing verified answers prompts to help answer questions about hawaii tourism
This was matched on the title, the description and its contents.
- [hawaii tourism calendar table](#)
Workspace: PBIFabricDoNotDelete
This was matched on the title and its contents.
- [Hawaii Tourism Mike](#)
Workspace: PBIFabricDoNotDelete
This was matched on the title and its contents.

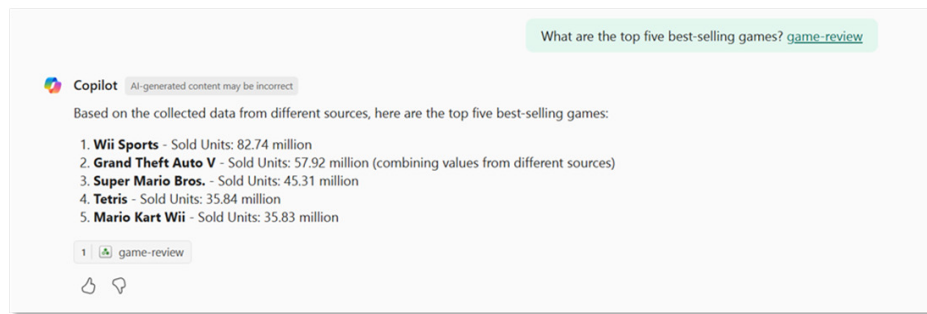
Which one would you like to use?

[Use 1. Hawaii_Tourism_Verified_Answers](#) [Use 2. hawaii_tourism_calendar_table](#) [Use 3. Hawaii Tourism Mike](#)

- **Get summaries** about an entire report or a specific topic within that report. Ask things like, “Summarise the relationship between weather and visits to Hawaii” or “Summarise trends in sales last year”. Do note though, at this time, you can only get summaries about reports, not models



- **Attach Fabric data agents** and get answers without leaving Power BI Copilot (think of these as experts in a specific topic, customised and trained by authors).



Things you should consider:

- **Turn on the tenant setting:** to try it out, administrators need to enable Copilot in Power BI, additionally enable the following new tenant setting. Users can access a standalone, cross-item Power BI Copilot experience
- **Prepare your data for good results:** you must prepare your data for it to work well with Copilot
- **Mark your data as prepared (prepped):** semantic model authors will have the ability to mark a semantic model as prepped in model settings. This impacts the appearance of 'Needs AI prep' warnings
- **Warnings on unprepped data:** models that aren't marked as prepped will be accompanied by warnings that the answer quality could be low. At the beginning of the Preview, you'll only see a warning experience on answers. However, in the near future, warning labels will start showing up on items in search responses
- **Administrators can hide unprepped data (coming soon):** capacity administrators will have the ability to hide data that hasn't been marked as prepped by authors. This administrative setting will be available shortly.

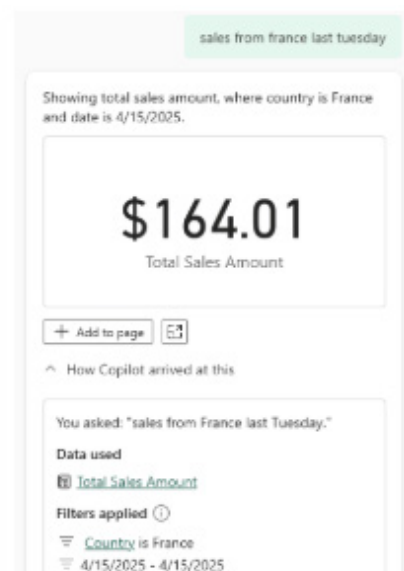
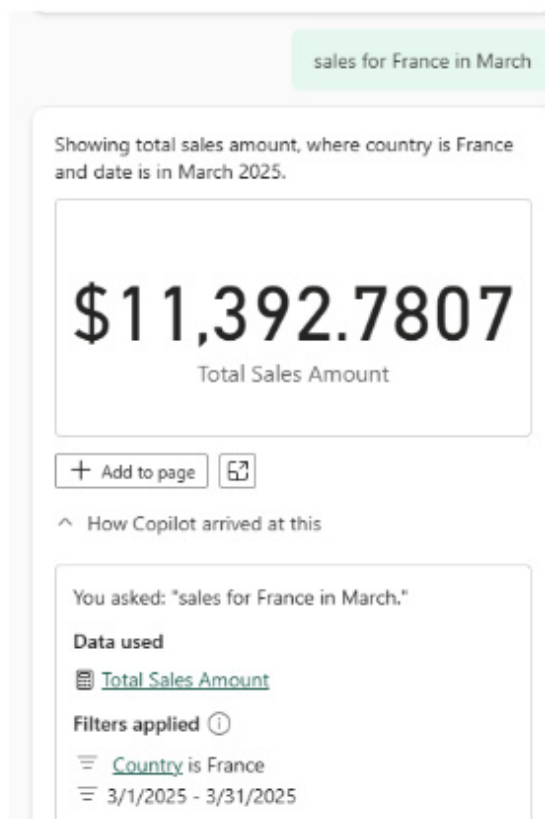
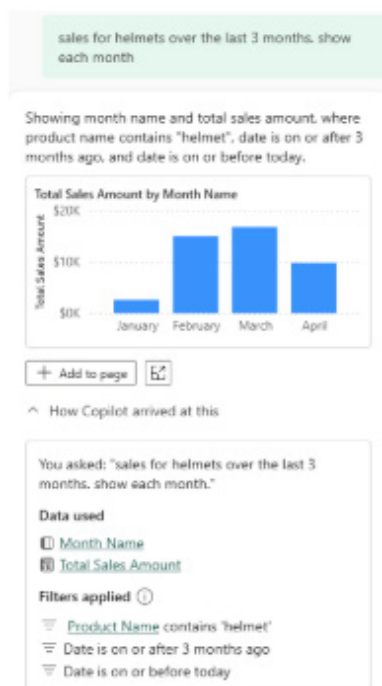
Quality improvements to language understanding for data questions

This month sees Power BI make improvements to its enhanced language understanding layer that was shipped back in March. With this revision, you should notice better handling of relative dates and generation of proper filters to provide more reliable visual output.

Relative dates by month

Partial dates like "March" use the current year even if not specified, e.g. "March 2025"

Relative dates by day of the week



These new Copilot tooling features have been incorporated to ensure the work authors have done by adding AI instructions, reducing schema and curating verified answers work in tandem with the data question capability in Copilot.

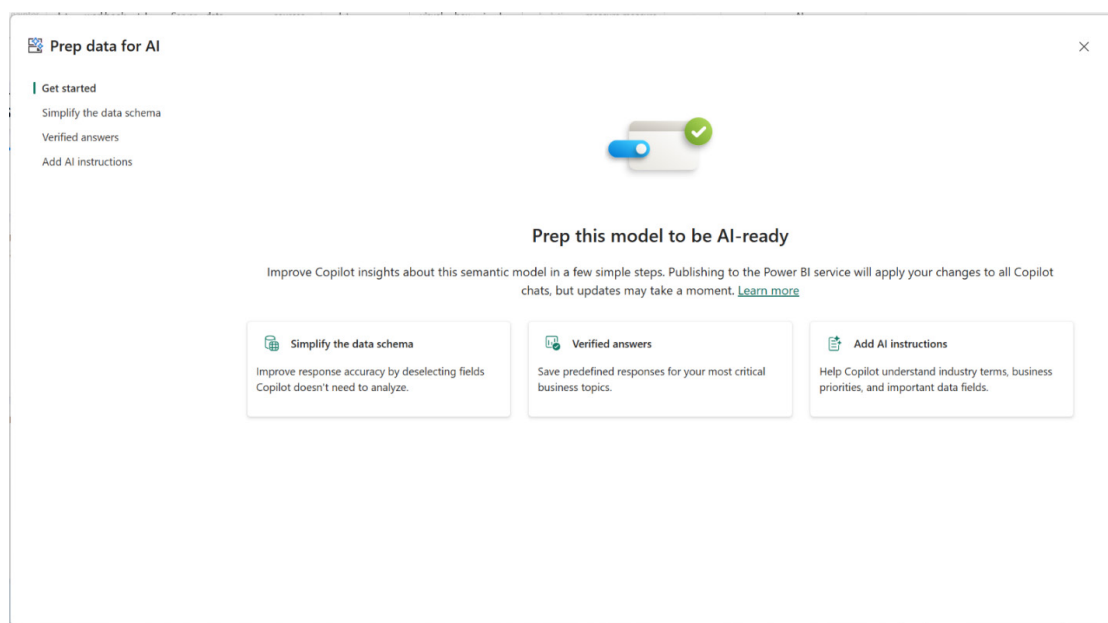
Get your data AI ready (Preview)

Preparing data, ensuring it's clean, structured and fit for purpose, is the foundation for building reliable, scalable AI solutions. To help you unlock the full potential of AI, this update sees the Preview of new capabilities designed to help you prepare your data to work better with Copilot.

Three [3] essential capabilities to assist in preparing your semantic models for AI are now available in Preview:

1. **AI Data Schema:** select a curated subset of your model's schema for Copilot to use in its responses. By narrowing the scope to only the most relevant fields, you help Copilot reason more effectively and deliver more accurate results
2. **Verified answers:** curate verified, visual answers directly from your report to guide Copilot to respond with confidence to the most frequently asked questions. This helps ensure consistency and trust in how Copilot answers your users' questions
3. **AI instructions:** add contextual details, business logic and interpretation guidelines directly into Copilot prompts. This helps tailor Copilot's responses to align with your organisation's specific needs.

Now, you'll see a new 'Prep data for AI' button in the Ribbon, right next to the familiar Copilot button. Clicking opens a new dialog with these three [3] tools to help you get started:



It should be noted:

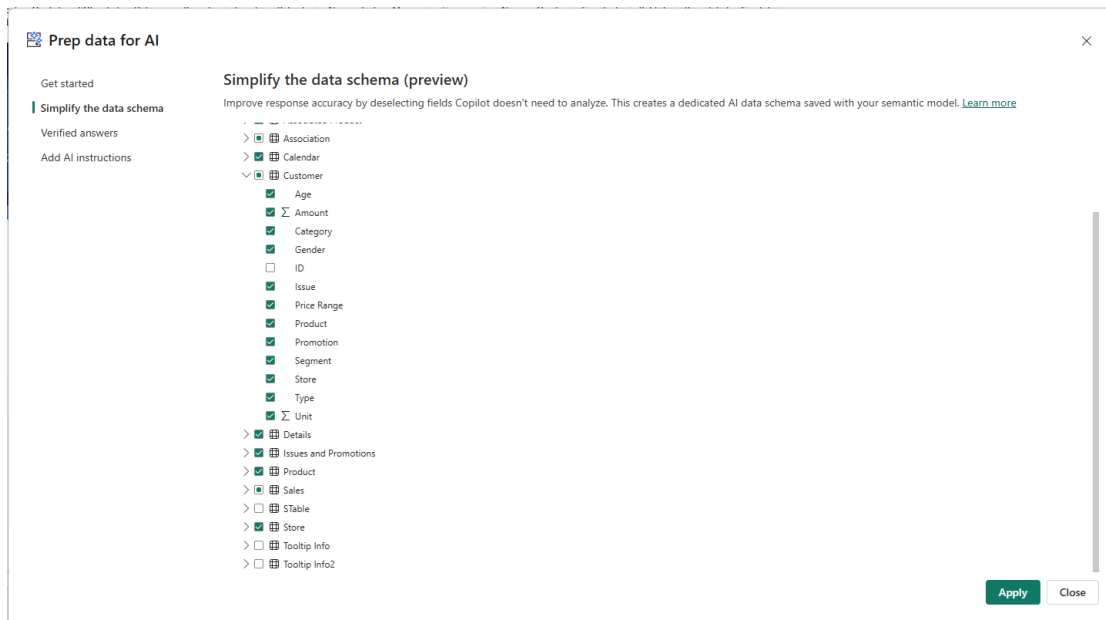
- all prep data for AI features are saved at the semantic model level
- 'Prep data for AI' features work best with models in import mode
- you must have written permission on the model to use these capabilities
- it is recommended that you use import mode. Composite and DirectQuery have limitations. DirectLake is not yet supported
- in Power BI Desktop, after preparing your data, test the impact directly in the Copilot pane (you may need to refresh or reopen the pane to see updates)
- in Power BI Service (including Standalone Copilot): once your report is published, users will benefit from your data prep efforts immediately in their Copilot experiences. Be sure to mark your semantic model as prepped for AI.

AI Data Schema

AI Data Schema is available as part of the efforts to assist in preparing your data for AI applications. The semantic models are optimised for Power BI Reporting, which means they often contain fields that don't necessarily need to be considered for Copilot responses. AI Data Schema allows semantic model authors to select a subset of the semantic model's schema as the AI Data Schema for Copilot to reason over in its responses. By selecting only the most important fields, you help Copilot focus on what matters, improving the relevance, clarity and accuracy of

its responses. It's a simple way to guide AI behaviour without having to restructure your entire model.

To get started, simplify your data schema for Copilot through the 'Prep data for AI' button in the Desktop Ribbon. To configure your AI data schema, choose the tables, columns, measures and hierarchies for Copilot's use. Confirm your selections and close the dialog.



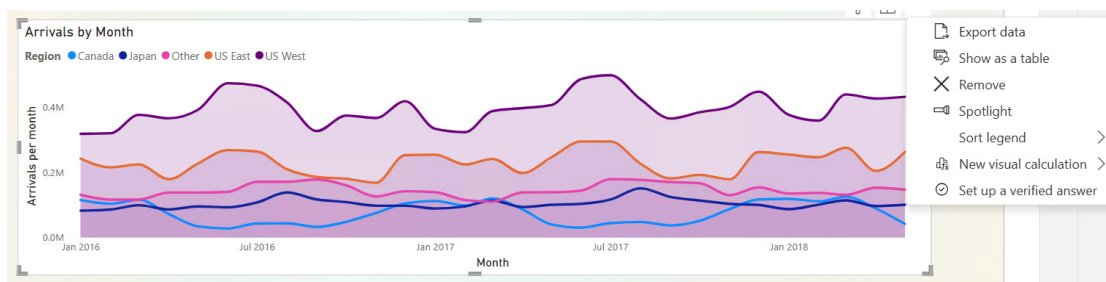
Verified Answers

Verified Answers are curated responses that are automatically triggered when users input predefined phrases in Copilot chat experiences. Each Verified Answer pairs a predefined trigger phrase with a specific visual from your report, both stored directly in the semantic model.

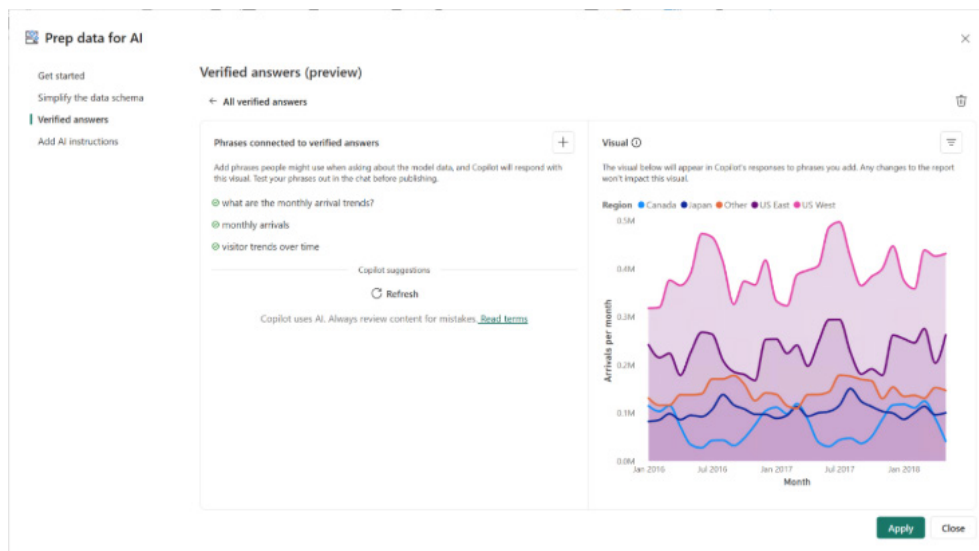
By guiding Copilot to respond with trusted, pre-defined answers, you help ensure users get consistent, accurate information, especially for high-impact or frequently asked questions. Indeed, there are two [2] primary goals of Verified Answers:

- **Improve response accuracy:** surface curated visuals and content to deliver more precise, context-aware answers, especially for nuanced or frequently asked questions
- **Enhance Copilot intelligence:** teach Copilot what a 'good' answer looks like by linking specific questions or keywords to trusted responses. Over time, this helps Copilot respond more intelligently to similar prompts.

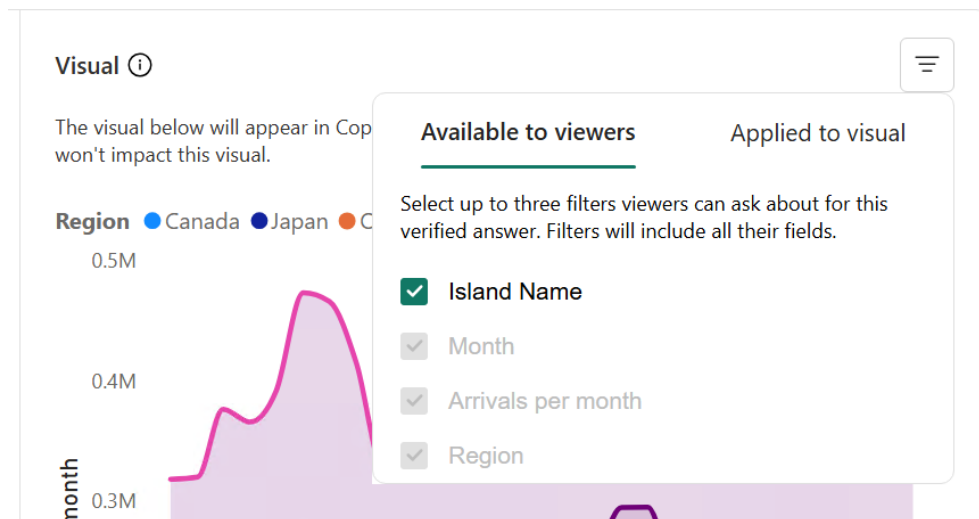
To get started with Verified Answers, right-click the report visual you want to use, open the three-dot menu, and select 'Set up a verified answer'.



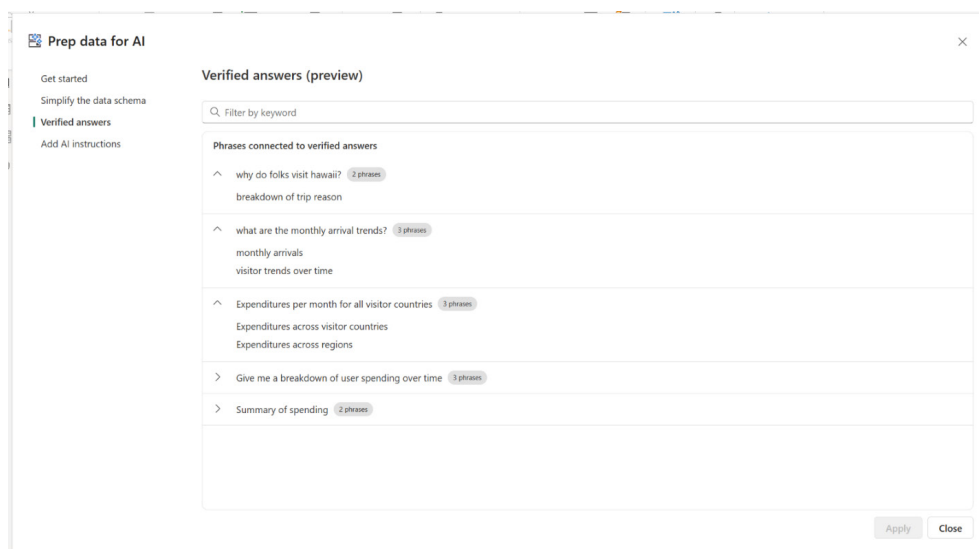
In the setup dialog, you can define trigger phrases, keywords or full questions that users are likely to ask. Once configured, the associated visual will be returned when a user enters a matching or similar phrase into a Copilot chat.

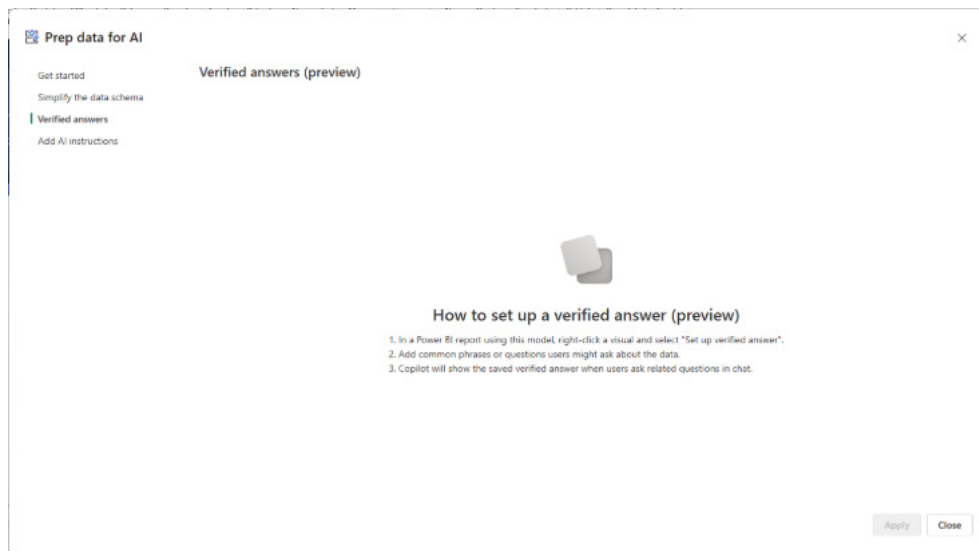


You can also include filters to ensure that filtered data is reflected in the Verified Answer. This allows users to receive accurate, filtered responses without requiring authors to manually define variations of trigger phrases for each filter state.



To manage your verified answers, access the dialog via 'Prep data for AI'. From here, you can see your Verified Answers, add additional trigger phrases, and delete any Verified Answers that no longer apply.





AI instructions

With AI Instructions you can add key business context, guidance and domain-specific logic directly into the prompts Copilot sees. This helps it generate more relevant, accurate responses. This capability empowers you to tailor Copilot's responses to align with their specific business needs while providing deeper insights into the nuances and context of your operations that Copilot might not otherwise understand.

Whether it's highlighting busy seasons, defining how metrics should be interpreted or excluding irrelevant data, AI Instructions allow you to shape how Copilot understands and interacts with your data, resulting in more tailored insights and greater confidence across Copilot outputs.

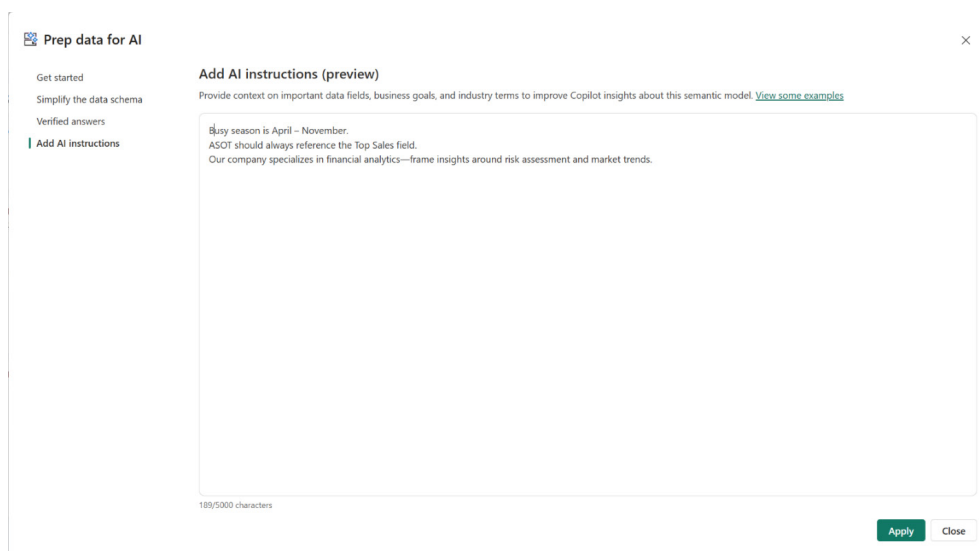
AI Instructions can be used in various ways:

- Business context and interpretation:
 - The busy season is April – November
 - "ASOT" should always reference the "Top Sales" field
 - The company specialises in financial analytics: frame insights around risk assessment and market trends
- Specific guidelines or ways to analyse the data in the model:
 - When asked about sales, break results down by location and salesperson
 - Exclude data about part-time students
 - Only show information about Golden Retrievers when asked about dogs.

These instructions help ensure Copilot aligns with your unique goals and organisational logic; no extra manual intervention is required.

To get started with AI Instructions:

- you can find AI instructions in the 'Prep data for AI' dialog box
- to add AI Instructions, navigate to the 'Add AI Instructions' tab and add in your instructions, hit Apply and close the dialog.



Testing with the skill picker

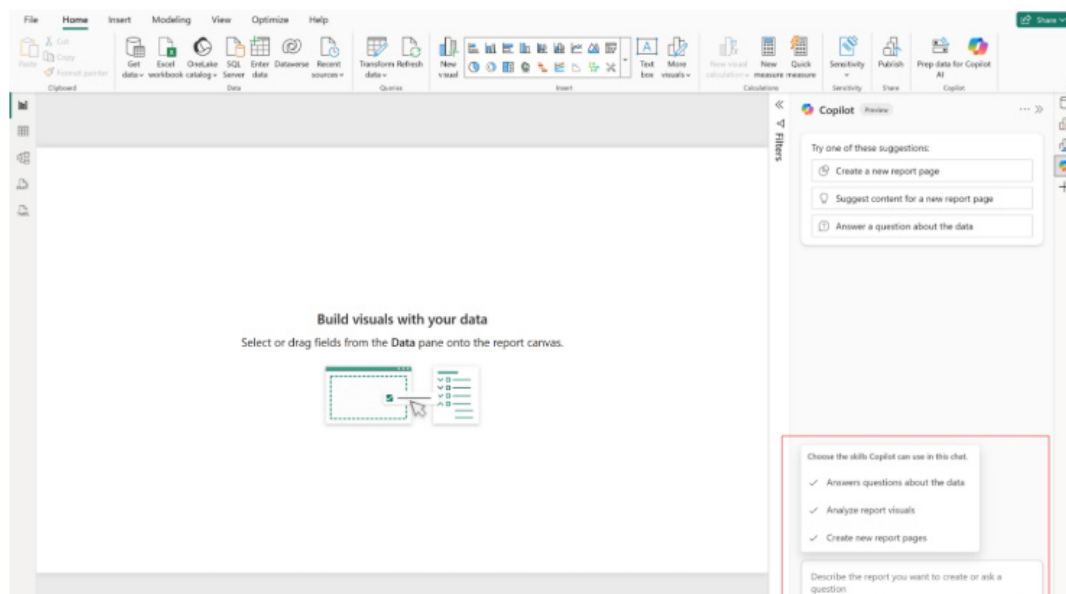
After preparing your data for AI, you can test what your end-users will see through the Desktop report pane. Power BI has added a new capability, the skill picker, to assist. The skill picker is a tool that gives you greater control over how Copilot responds to you by allowing you to select specific Copilot capabilities to enable.

Currently, the skill picker includes three [3] capabilities:

1. **Answer questions about the data:** leverages Copilot to respond to questions based upon a given semantic model
2. **Analyze (sic) report visuals:** enables Copilot to interpret and answer questions about the visuals within a report
3. **Create new report pages:** allows Copilot to generate new report pages based upon your prompts.

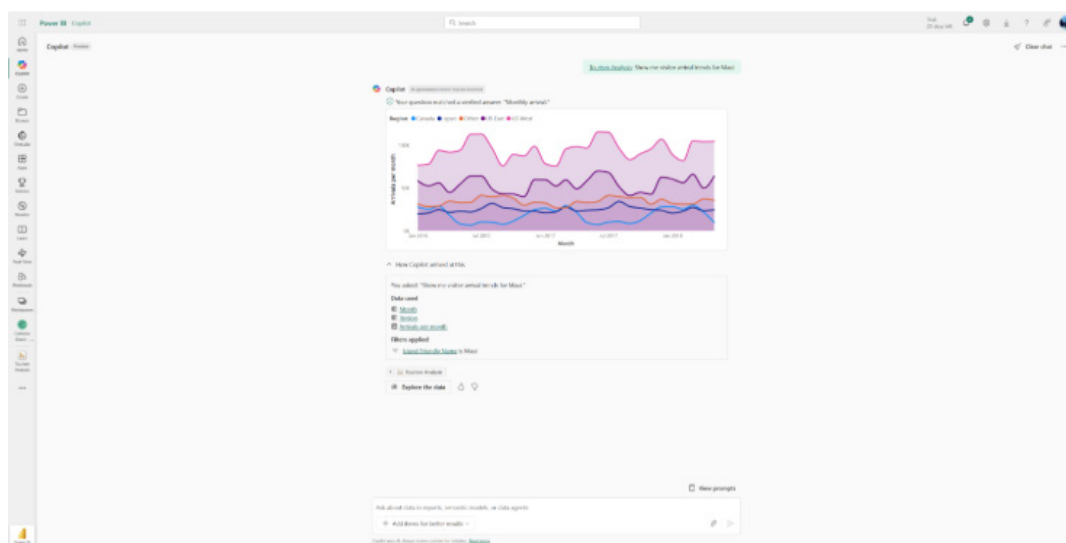
By selecting which capabilities to enable, authors can simulate different environments and test how their changes impact the Copilot experience.

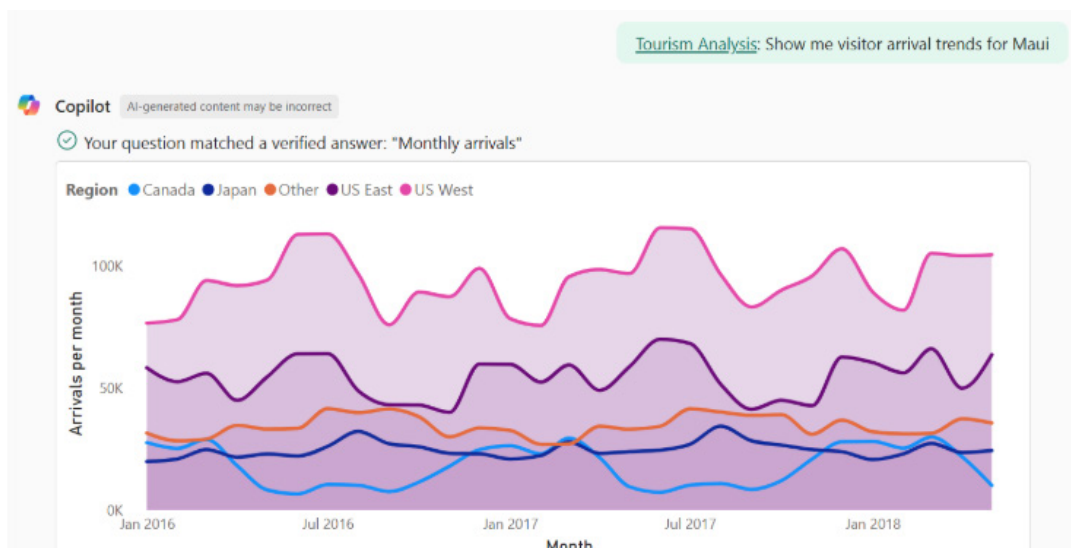
To get started with the skill picker, by default, all three [3] skills are enabled in Power BI Desktop, giving you immediate access to the full range of Copilot functionality. To customise the skills for your session, simply click the 'Select Skills' dropdown in the Copilot chat box and choose the combination that fits your needs.



Preparing a semantic model for AI

After setting up and testing your data with the available tools, you can publish your Power BI report online. This allows users to interact with Copilot, either within the report pane or through the standalone Copilot experience. The responses they see will be based on verified data, an optimised AI schema and clear AI instructions.





Mark a semantic model as prepared for AI

Once your semantic model is sufficiently configured, you can mark it as 'Prepped for AI'. To do this:

- in Power BI Service, navigate to your semantic model's settings
- open the 'AI Preparation' section
- check the box to mark the model as prepped and select Apply.

▸ Featured Q&A questions

▸ AI preparation

Mark this model as prepped for AI when it's ready for use with Copilot. This action clears any warnings about incomplete data prep for the model and its connected reports. [Learn more](#)

☒ Prepped for AI

Apply

Discard

▸ Endorsement and discovery

Marking your model as prepped for AI will optimise this data for Copilot experiences, especially in the Standalone Copilot experience. This action will remove any friction treatment or warning within the standalone

Copilot experience. It will also help optimise finding models that are ready for AI consumption to help users get the answers they need.

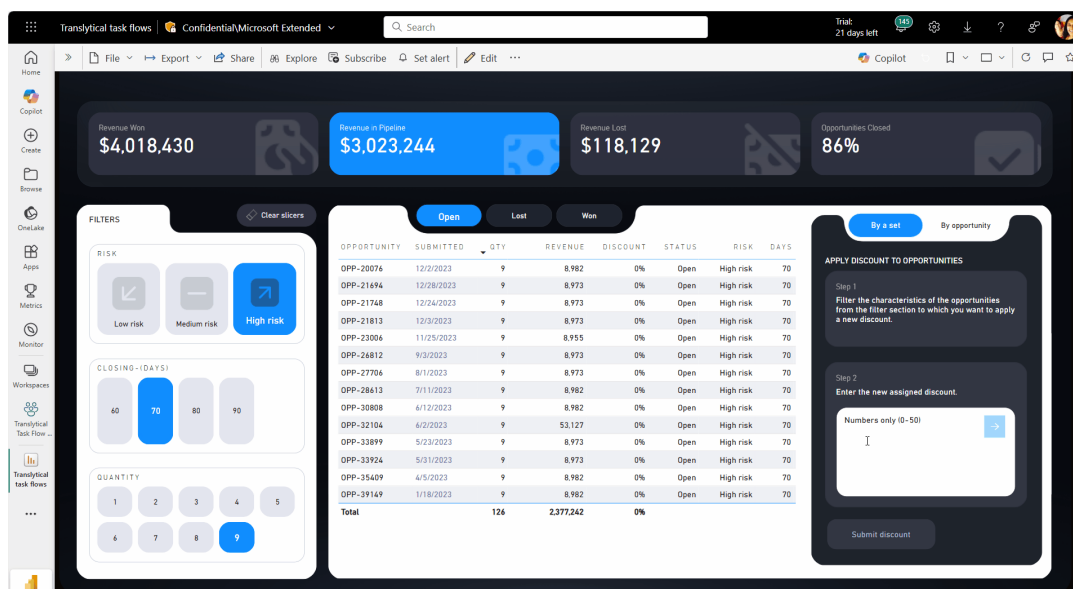
Automate report actions and data writeback with Translytical task flows (Preview)

Power BI report buttons may now run Fabric User data functions for custom, automated action, including data write-back. This marks a major evolution in Power BI reports to support all kinds of "translytical" task flows, such as updating records, dynamic notifications, adding annotations or even creating powerful workflows that trigger actions in other systems.

Translytical task flows can enable the highly requested write-back capability natively in Fabric. Leveraging the Fabric User data function,

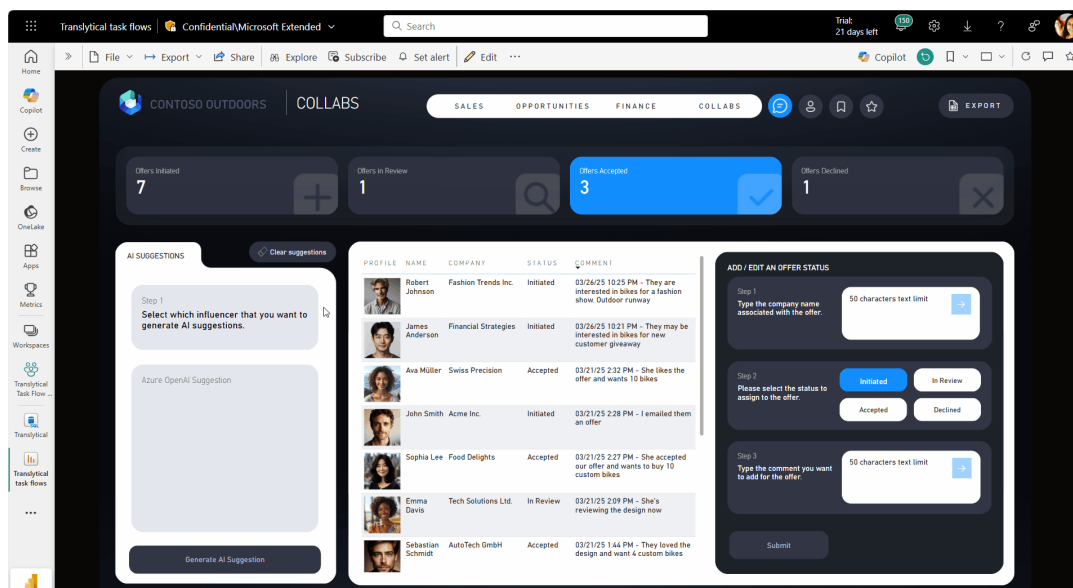
you can programmatically update, add or delete records of data based upon the filter context passed from the report.

For example, in this Power BI report, you can modify the discount value seen in the table without ever leaving the report. You simply enter in the new value in the text slicer and click the 'Submit discount' button, which runs a Fabric User data function that instantly updates the data source records that match the applied filters.



Translytical task flows can enable more than just data write-back, but they can also automate a wide variety of tasks and even take actions in other systems via external APIs. You can submit approval workflows, trigger dynamic notifications, augment data on the fly and more.

For example, in this Power BI report, you can generate tailored AI suggestions for the influencers seen in the table using Azure OpenAI Responses API. You simply select the influencer and click the ‘Generate AI Suggestion’ button, which runs a Fabric User data function that instantly provides an Azure OpenAI response based upon a fully customisable prompt.

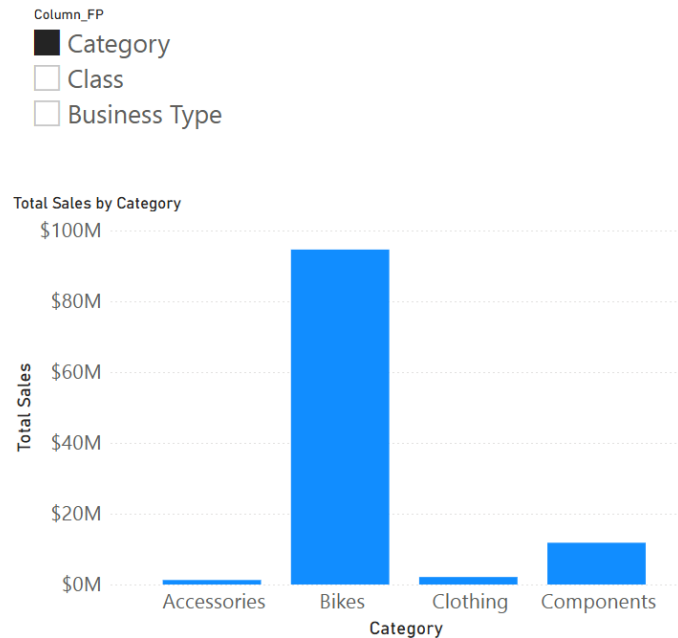


To build your own translytical task flow, you'll first need to turn on the 'Translytical task flows' preview feature in Power BI Desktop by navigating to **File -> Options and settings -> Options -> Preview features**.

Persisted sorting for field parameters (Preview)

Field parameters are a great way to provide flexibility to your report users. However, there have been issues. One of the problems has to do with how changes in the field selected by the field parameter resets the sorting of a visual. This month resolves this by persisting in the sorting even if you change the field selected by the field parameter.

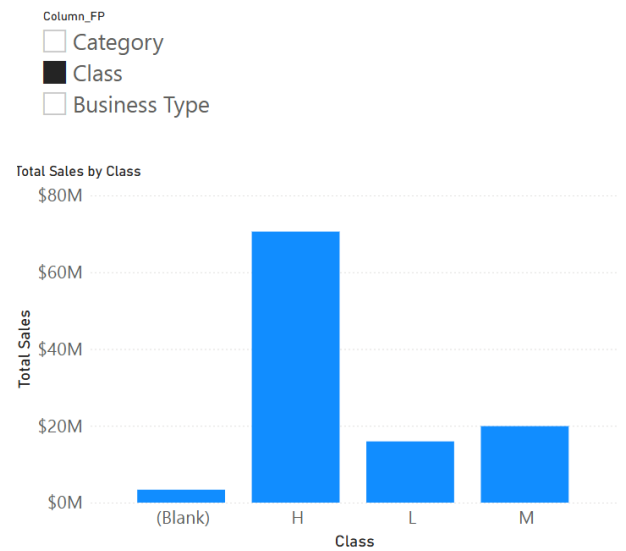
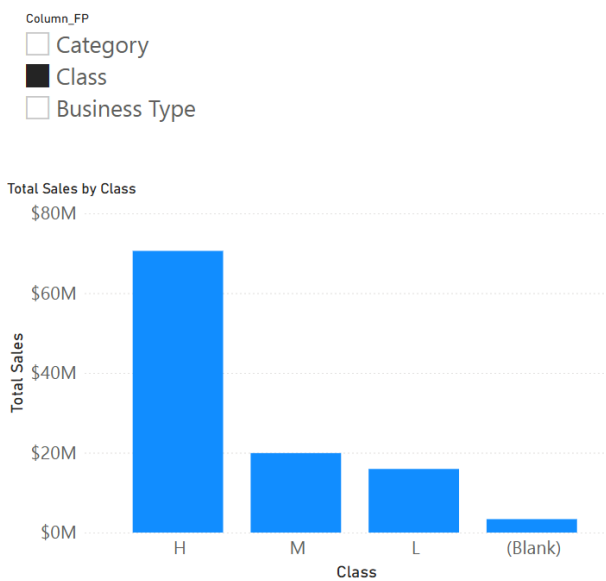
For instance, a visual can use a field parameter to display 'Total Sales' based upon the selected field. In this example, the selected field is Category. Notice that we have sorted the visual by Category ascending:



Now let's switch the field parameter to Class instead of Category. You can compare and contrast the old and new behaviour as follows:

Old (Before May 2025)

New (From May 2025)



As soon as you switch the field selected by the field parameter to something else (in our example Class), the sort order of the visual would change and your visual would be sorted by 'Total Sales' descending (which is the default sort order for this type of visual). Starting with this update, that is no longer the case, and the sort order will persist and the

visual is still sorted by the field selected by the field parameter even after it is switched.

This is one of the key improvements to field parameters. More updates are to follow, apparently. We look forward to them with interest.

Easily refer to values in your visual with visual calculations (Preview)

This month, two new functions are joining the visual calculations family: **LOOKUP** and **LOOKUPWITHTOTALS**. Visual calculations simplify DAX by focusing on one visual at a time, allowing for easy creation of powerful calculations and verification of your work.

The new **LOOKUP** and **LOOKUPWITHTOTALS** functions enable retrieval of values from the visual matrix through absolute navigation with one or more filters applied. Although both functions serve similar purposes, they differ in handling omitted filters.

Consider a visual that shows the Sales per product Category and Class. In this visual, let's add the following visual calculations:

LOOKUPExample = LOOKUP(AVERAGE([Sales Amount]), [Class], "H")

LOOKUPWITHTOTALSExample = LOOKUPWITHTOTALS(AVERAGE([Sales Amount]), [Class], "H")

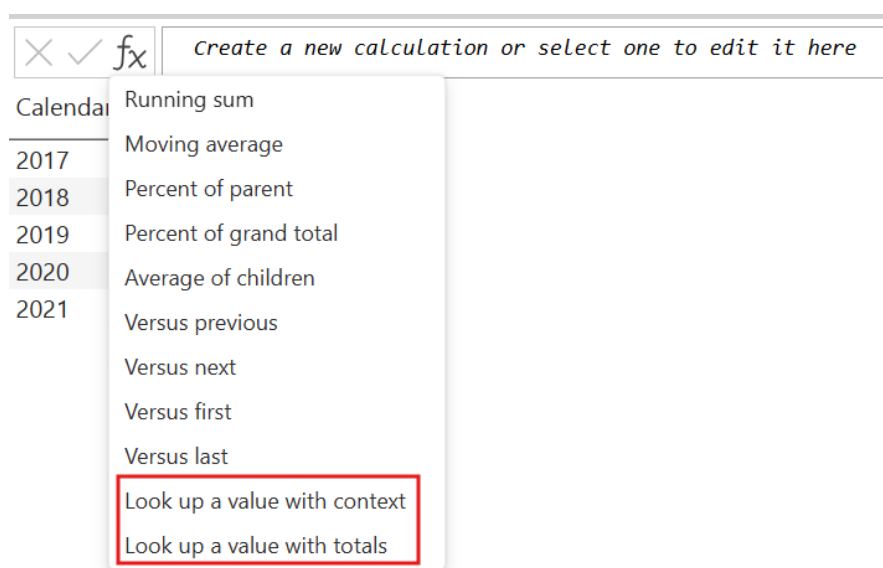
Category	Sales	LOOKUPExample	LOOKUPWITHTOTALSExample
⊖ Accessories	1,272	77	23,530
	1,093	77	23,530
H	77	77	23,530
L	44	77	23,530
M	58	77	23,530
⊖ Bikes	94,621	63,654	23,530
H	63,654	63,654	23,530
L	13,612	63,654	23,530
M	17,354	63,654	23,530
⊖ Clothing	2,118		23,530
	2,118		23,530
⊖ Components	11,799	6,858	23,530
	153	6,858	23,530
H	6,858	6,858	23,530
L	2,286	6,858	23,530
M	2,503	6,858	23,530
Total	109,809	23,530	23,530

As demonstrated by the background colours in the image above, the **LOOKUPExample** visual calculation returns the average Sales for the products in the “H” class within each Category, as **LOOKUP** infers any omitted filters from the context.

In contrast, **LOOKUPWITHTOTALS** treats any omitted filters as referring to the total. Since we didn’t specify a filter for Category, the

LOOKUPWITHTOTALSExample visual calculation returns average Sales for the products in the “H” class across all categories. This is the same value returned by the **LOOKUPExample** on the Total row.

To make things even easier, Power BI has also added templates for both **LOOKUP** and **LOOKUPWITHTOTALS**:

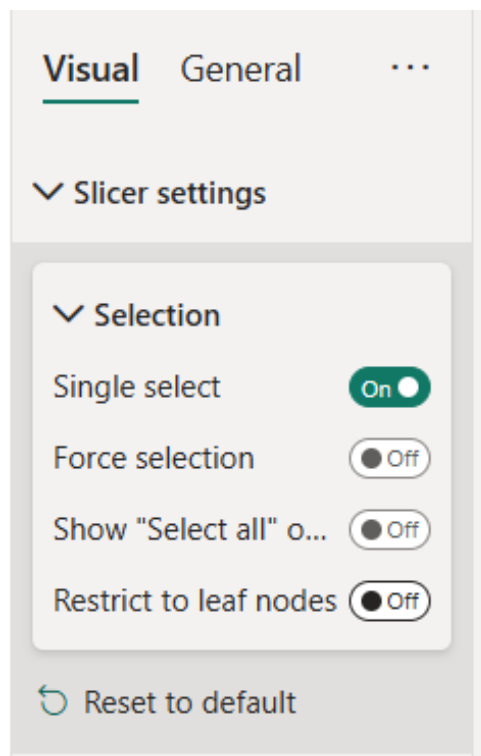


Updates to the new List slicer (Preview)

A significant update has been made to the new List slicer, which was released in Preview in October last year. This update allows for even more control over how the slicer both looks and functions, allowing you to create almost any list slicer design you can think of.

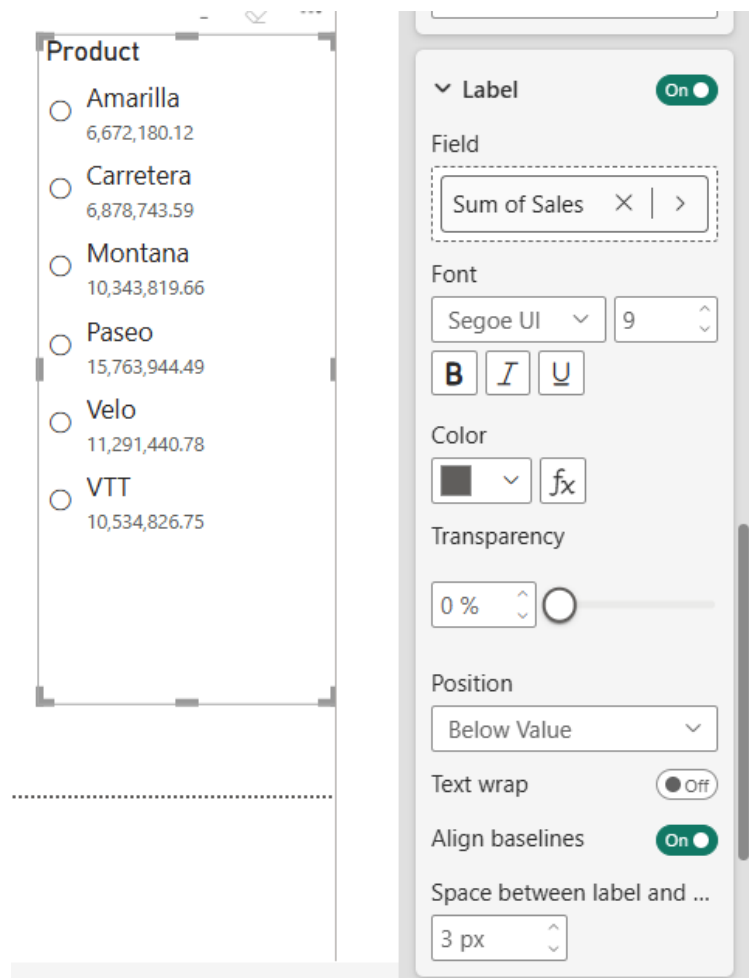
For a start, you may now restrict selection in the slicer to only the last level (leaf nodes) when using a hierarchy with the slicer. This means that

users can only select the most specific items within the list. To use this option, just select the ‘Restrict to leaf nodes’ option with the Selection card of the Formatting pane. Once selected, selecting any of the higher nodes of the slicer will always expand / collapse the level instead of selecting it.

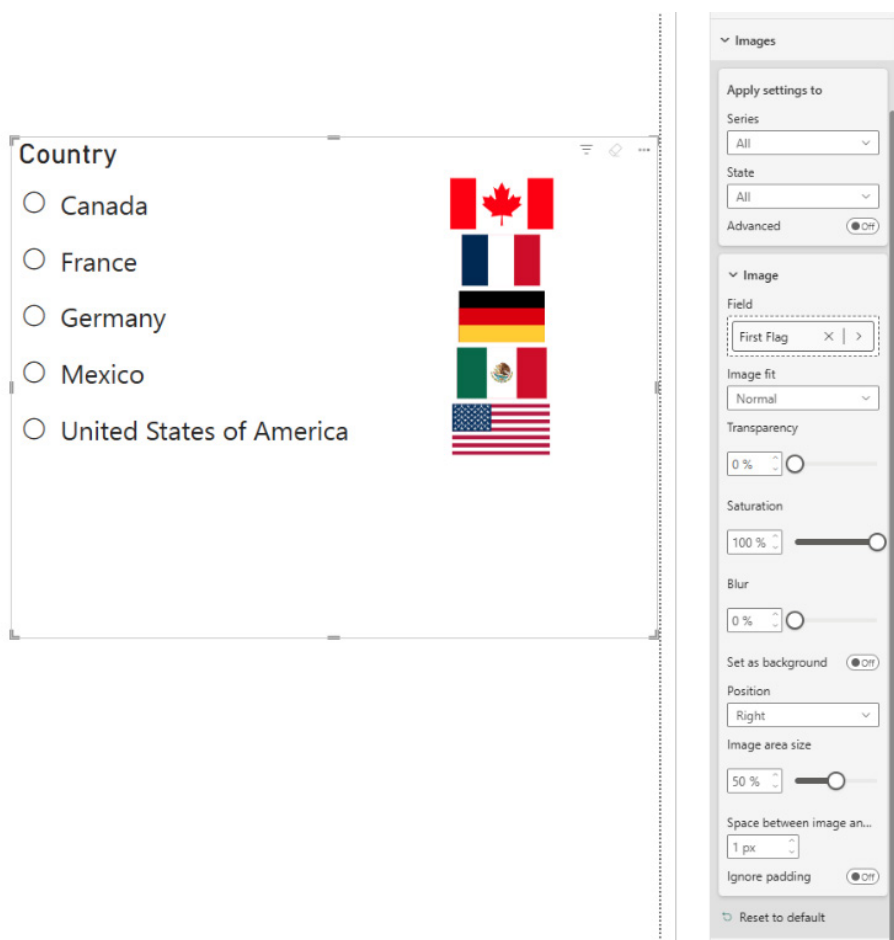


Next are two [2] new slicer formatting options: Label and Images.

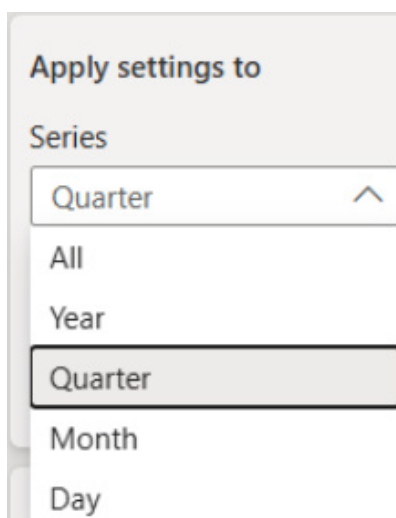
1. **Label** allows you to add a secondary row to each value in the slicer to provide additional information, whether that be a subtitle or an additional metric. You can find Label as a section of the Callout values card. You have similar formatting options to the values themselves



2. The new **Images** card lets add, er, images to your slicer. You can add a unique image from your model to go along with each value of your slicer and control some basic formatting for that image including (but not limited to) image fit, transparency, saturation and position.

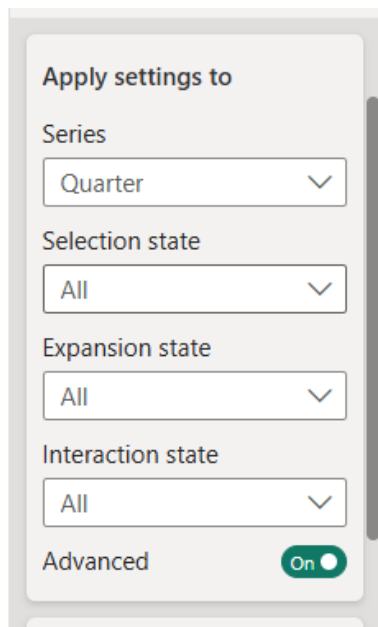


The method for controlling the formatting of the various states of the slicer has been thoroughly revised. Previously, you were limited to Default, Hover, Press, Selected and Mixed as your states, and the formatting was always shared across all fields within the hierarchy. Now, you may format each field of your hierarchy separately using the Series dropdown of the Apply settings to card.



For each of those, you'll have control over a subset of the following states: All, Rest, Hover, Press, Selected and Mixed. Power BI has been updated: the dropdown now conditionally shows only the states that are most commonly applicable for that card's component, so for example, you won't see Mixed as an Option for the expand / collapse icon.

Additionally, Power BI has an Advanced toggle that breaks down the states you can configure into Selection, Expansion and Interaction States.



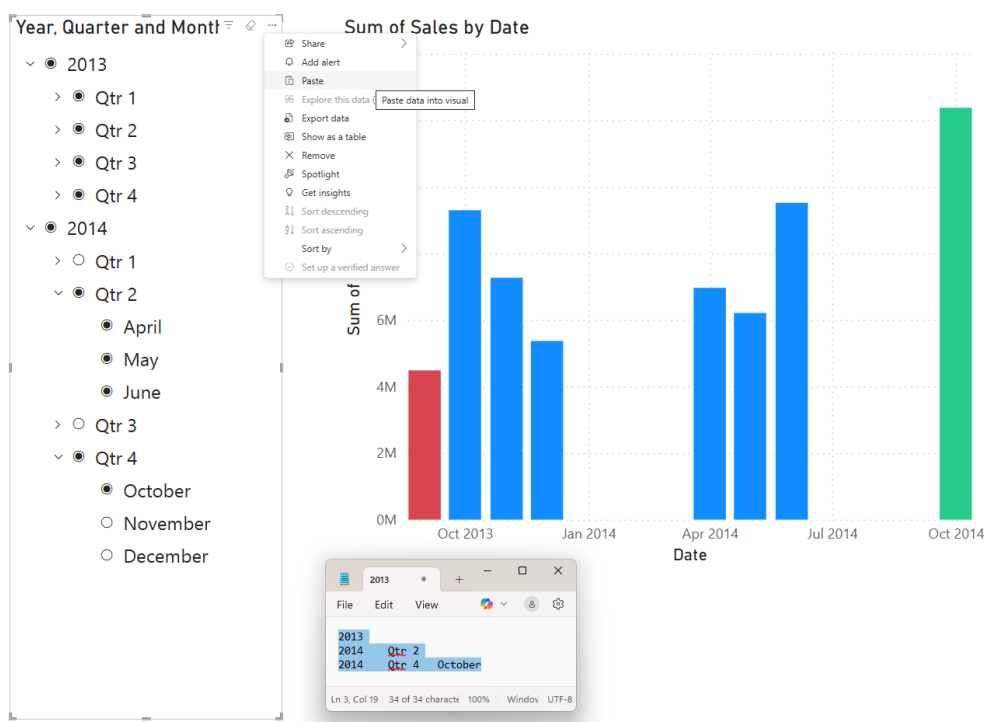
The breakdown of what's formattable in each of these is as follows:

- **Selection state:** All, Selected, Unselected, Mixed
- **Expansion state:** All, Expanded, Collapsed
- **Interaction state:** All, Hover, Press, Rest, Disabled.

This allows you to get incredibly precise about the formatting for every possible state of individual values of the slicer. It should be remembered that you may do this separately for each level of the hierarchy if you want to. You have these new state controls for the Callout values, Images, Selection icon, expand / collapse icon and Buttons cards.

Paste into List and Button slicers (Preview)

Starting with this update, you can now paste a series of values into the new List and Button slicers that are in Preview. Simply copy the values you want to filter by to your clipboard from tools just as Excel or Notepad, and either select the Paste option from the '...' button in the visual header or select the slicer and use **CTRL + V**.



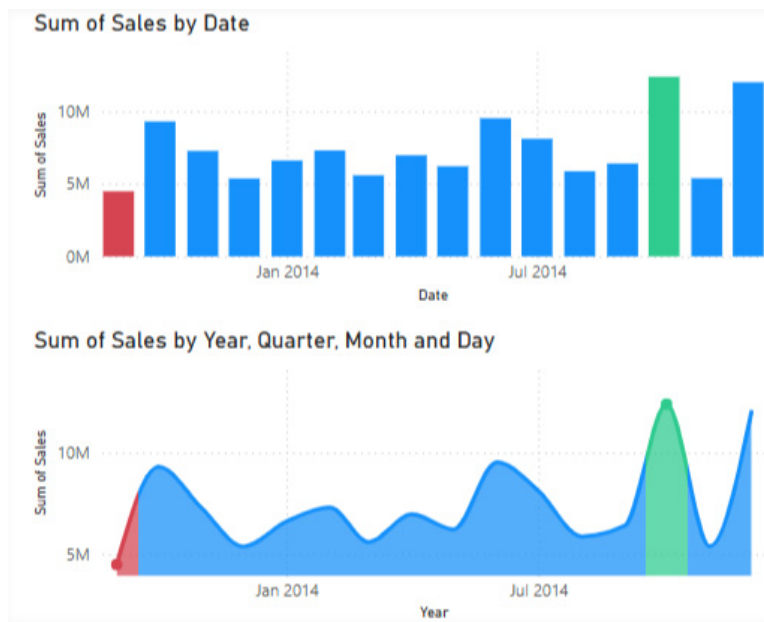
Each individual value should be on its own line, and levels of the hierarchy should be separated by tab. As an example, if you wanted to filter by the second quarter of a year, your row would be '2025 Quarter 2', with a tab in between. Also, the order of the values should be in the same order as the columns in the field well. Your data should also not include any headers.

It should be noted that:

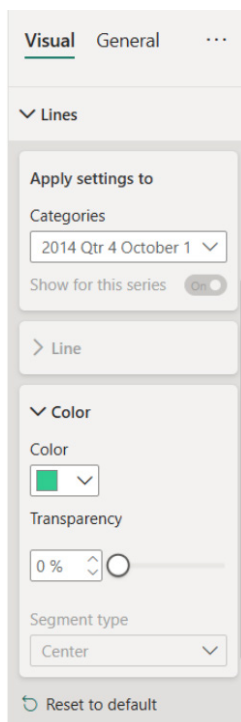
- columns that use group on keys are not supported
- when reading numbers and dates, Microsoft supports both the short format for the current culture and the current column format in Power BI
- Empty values within a row are treated as blanks.

Customise line segment colours in Line charts

This month introduces the ability to format individual line segments within Line charts. Just as you've been able to assign a specific colour to an individual bar of a Column or Bar chart, you can now colour the segments of your Line chart, allowing you to easily highlight interesting data points. You'll be to individually format both the colour and transparency of the line and shade area, in addition to the already supported markers.



To apply formatting to a line segment, just select the data point the formatting should apply to within the Apply settings to drop down within the Lines and Shade area cards of the Formatting pane.



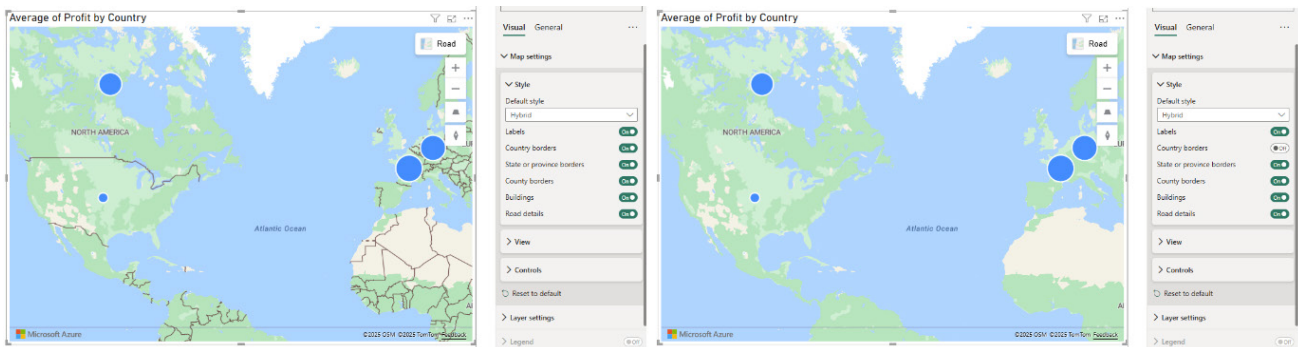
This feature not only closes a gap between Column, Bar and Line carts, but it also sets the stage for Power BI to support more advanced segment formatting like conditional formatting in the future. Cool.

Additional Base Map customisation options for Azure Maps visual

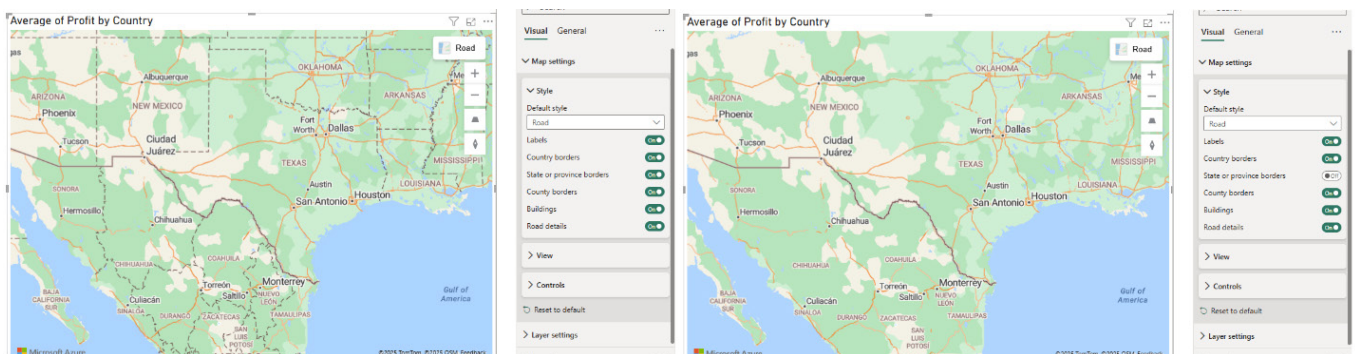
This month also sees the introduction of the enhanced base map customisation features for the Azure Maps visual in Power BI. These updates give you greater control over the appearance of the base map, helping tailor visuals to better support specific analytic goals.

You can now independently toggle the visibility of key map features to either highlight important boundaries or simplify the visual display.

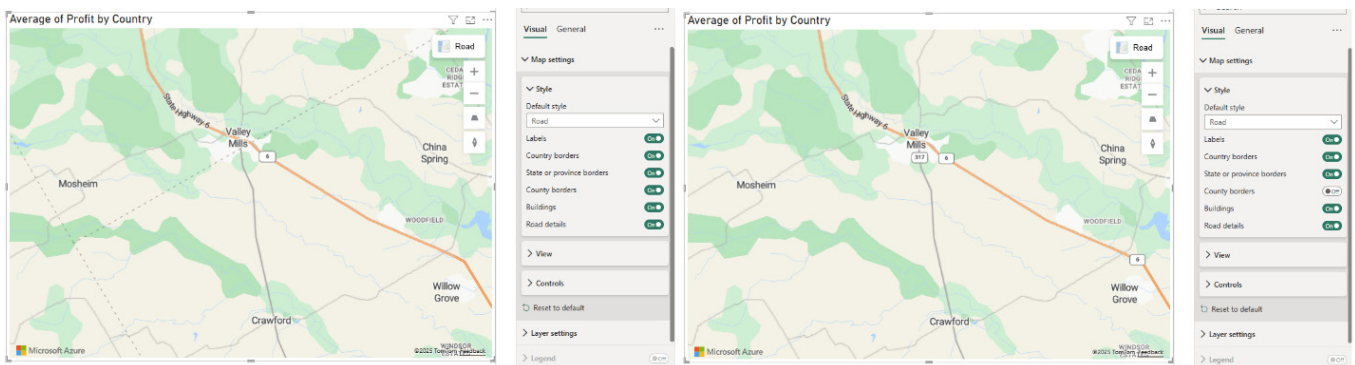
Country borders



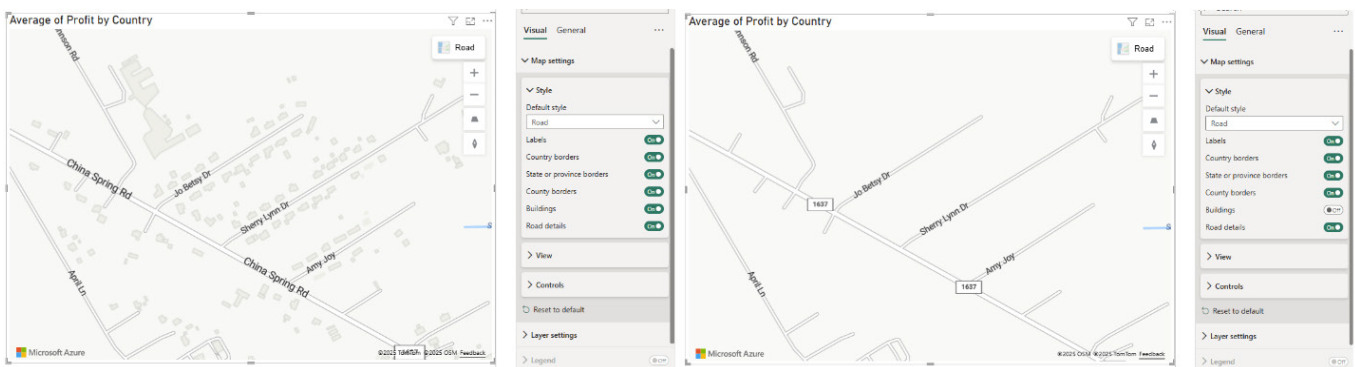
State or province borders



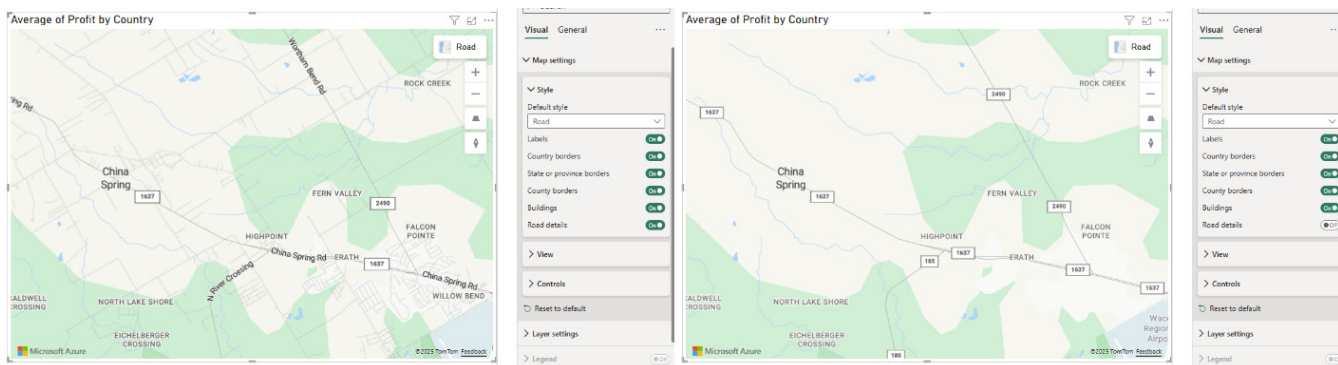
County borders



Buildings



Road details



It should be noted that this does not remove all roads but instead removes the more detailed / minor roads within a given viewport.

This added flexibility makes it easier to create cleaner, more focused geospatial reports, whether you're analysing trends across administrative regions or decluttering the map for a sharper presentation.

Upcoming Azure Maps tenant settings and layer ordering updates

As a reminder, beginning this month (June), Power BI will introduce new tenant settings for the Azure Maps visual, giving you more control over data residency and compliance. These more granular tenant settings will let you independently control:

- whether data may be processed outside your tenant's geographic region
- the use of certified Microsoft Online Services sub processors (used by the selection tool).

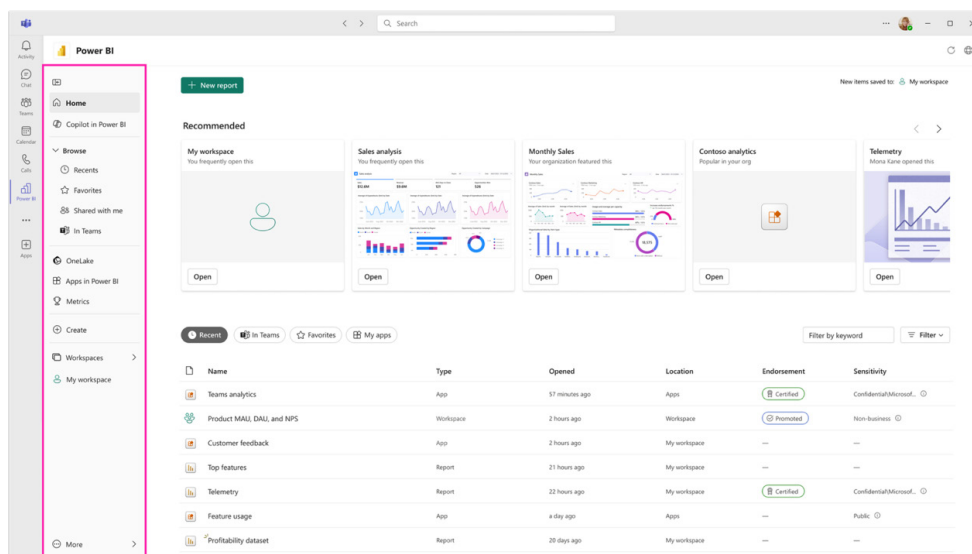
However, to take advantage of these changes, there is an important step to take: your organisation must be on the April version of Power BI Desktop (last month) or later. If you're still using an older version of Power BI Desktop in June, Azure Maps visuals will no longer work in your reports when viewing them in Desktop.

Also starting with this latest release, you may notice changes in the ordering of map layers in the Azure Maps visual. This update resolves

several known issues that previously affected the correct rendering sequence of layers, which results in layers and their associated labels occasionally overlapping in unexpected ways. The updated behaviour may impact the appearance of your reports, specifically when you have multiple layers visible at the same time. You can customise the layer order using the 'Layer position' setting available for each layer.

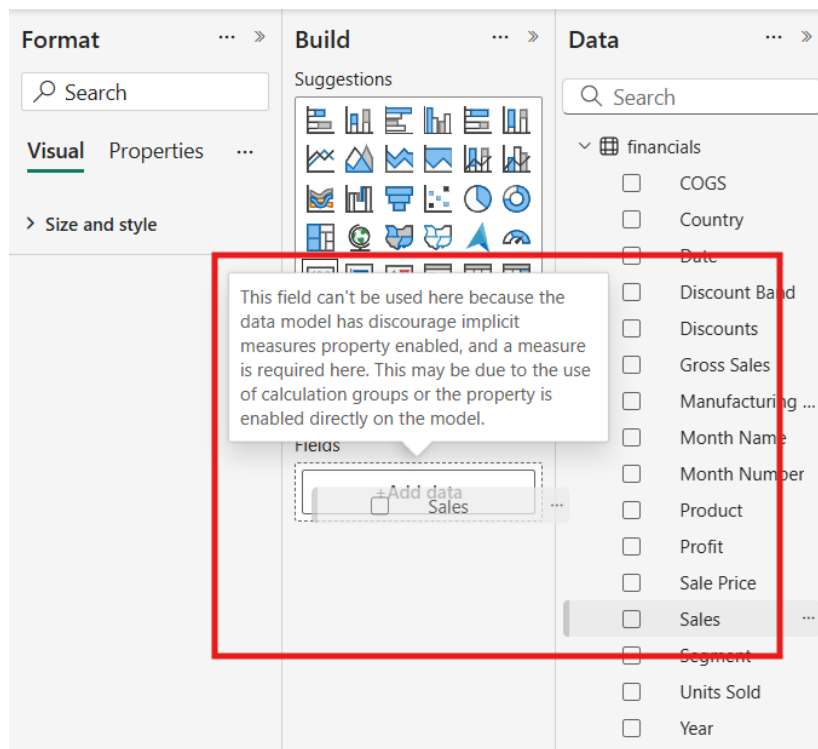
Upcoming changes to the Power BI application in Teams Navigation menu

There is a new left navigation menu that will be coming to the Power BI application for Teams in the upcoming weeks. This update will aim to simplify navigation within the app and make it easier to access Power BI items in Teams. According to Microsoft, the new navigation menu is designed to be more intuitive, helping users locate the tools and features they need faster and more efficiently.



Visual updates for semantic models with discourage implicit measures enabled

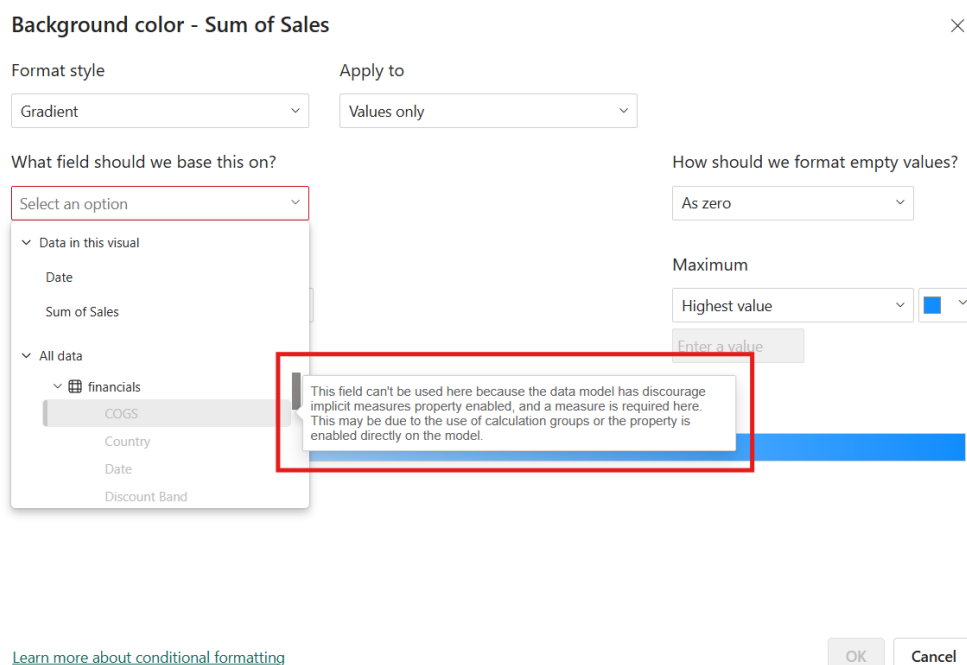
Visuals in Power BI reports now show helpful information when attempting to create or edit implicit measures when discourage implicit measures property is enabled on the underlying Power BI semantic model.



Semantic models can have the property discourage implicit measures enabled (disabled by default) to encourage the use of explicit measures and visual calculations in the model to aggregate data. Simply adding a data column and summing it in a visual is called creating an implicit measure and when this property is enabled, adding implicit measures does not work. This includes in the values and ToolTip sections, as well as in conditional formatting sections on visuals. For example, actions such as adding a data column to a card visual to show the first, last or sum of it do not work with this property enabled. Report authors can

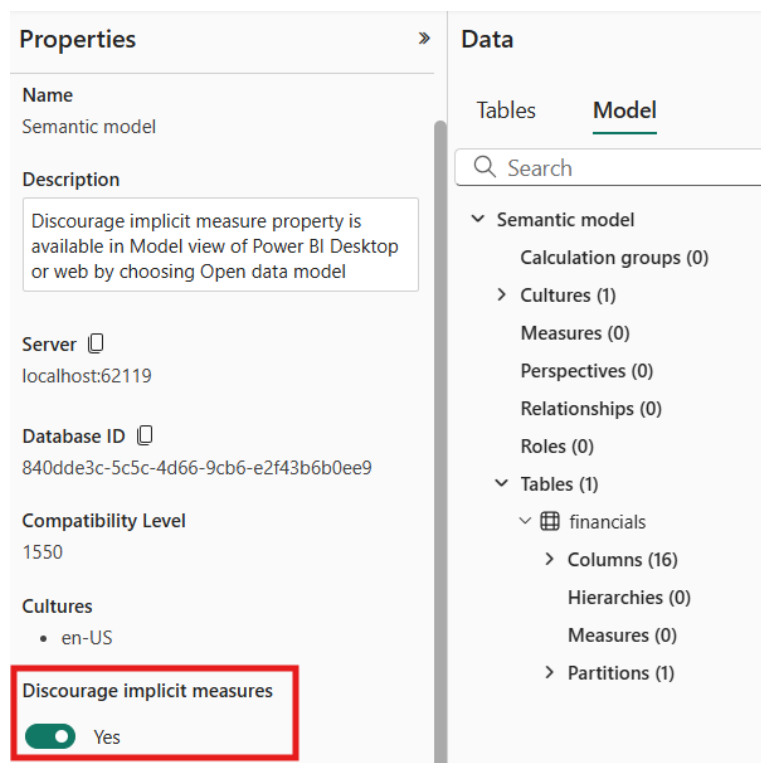
create a measure or use a visual calculation on the visual to visualise this aggregated data instead.

If an implicit measure were already created in a visual, on the visual itself or in conditional formatting, they will still work with discourage implicit measure now enabled. Editing or changing the implicit measure does not work, and disabled options show a helpful message to explain what is happening to the report author.



Typically, this property is enabled when using calculation groups in the semantic model, as calculation groups only apply to explicit measures, not implicit measures. Even when a calculation group is removed, the

property stays enabled until the model author disables it. The property can also be enabled without a calculation group present.



To enable or disable the discourage implicit measure property, go to Model view in Power BI Desktop or web, and in the Data pane pick Model, then select the 'Semantic model node'. The property is listed with a toggle in the Properties pane.

Selection Expressions for calculation groups

If you've worked with calculation groups, you have probably realised how powerful they are for simplifying measure logic and enhancing report functionality. There is more to come though with the Preview of selection expressions.

Selection expressions allow fine-tuned control over how calculations behave when certain conditions are met. They introduce additional

logic for handling cases where multiple calculation items are selected or when no specific selection is made on a calculation group. This provides a way to do better error handling but also opens interesting scenarios that provide some good default behaviour, *e.g.* automatic currency conversion. Selection expressions are optionally defined on a calculation group and consist of an expression as well as an optional dynamic format expression.

There are two [2] types of selection expressions:

1. **multipleOrEmptySelectionExpression:** this selection expression is applied when:
 - multiple calculation items have been selected
 - a non-existing calculation item has been selected *or*
 - a conflicting selection has been made
2. **noSelectionExpression:** this selection expression is applied when the calculation group is not filtered.

Additionally, using the new **selectionExpressionBehavior** setting on the semantic model, you can decide how calculations are evaluated for (sub)totals and what they return if no selection expressions are defined yet the user makes selections on the calculation group that would have triggered the evaluation of the selection expressions.

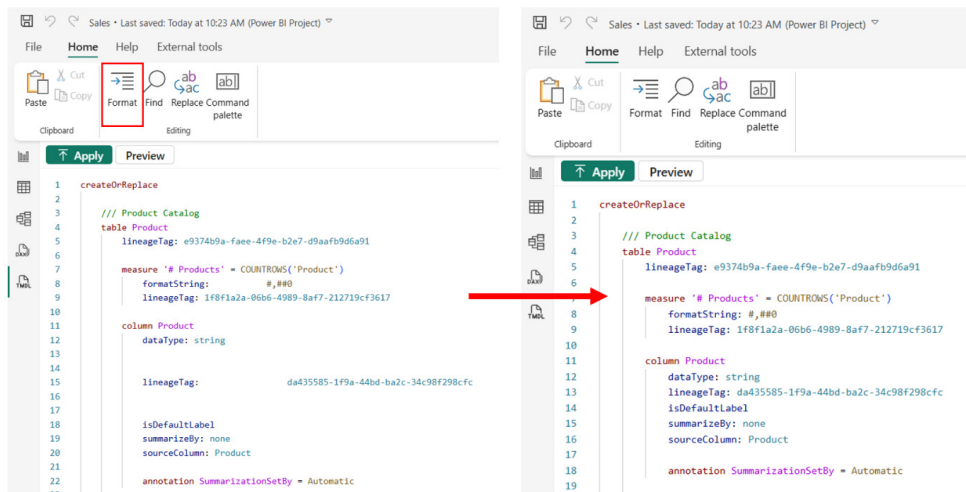
TMDL view enhancements

The Tabular Model Definition Language (TMDL) view provides an alternative to semantic modelling by using code rather than a graphical interface, offering robust editor features. This month, the TMDL view code editor has added several new functionalities detailed below.

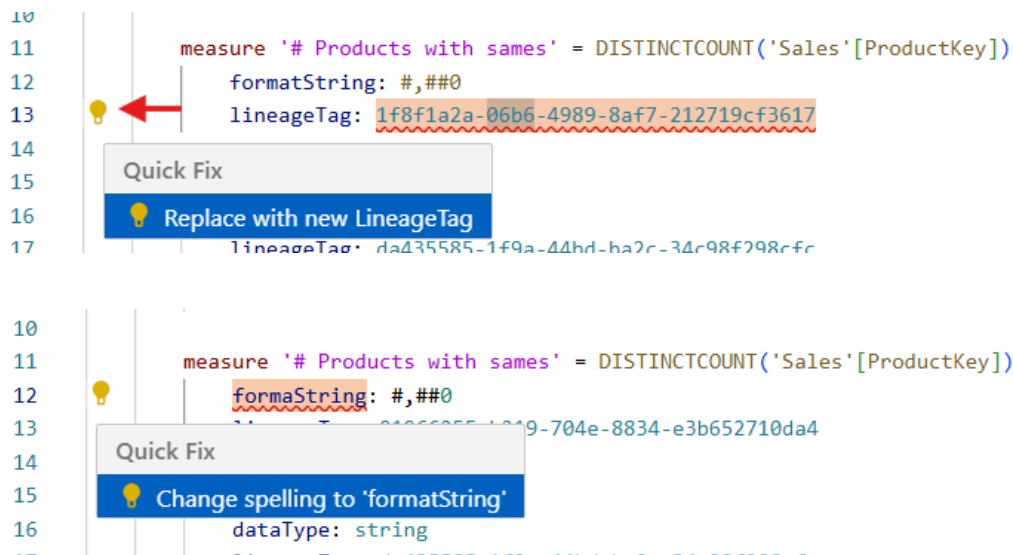
To start, context ToolTips on mouse hover provide information about each TMDL object or property.



You may also format your TMDL code by pressing **SHIFT + ALT + F** or clicking the Format option in the Ribbon.

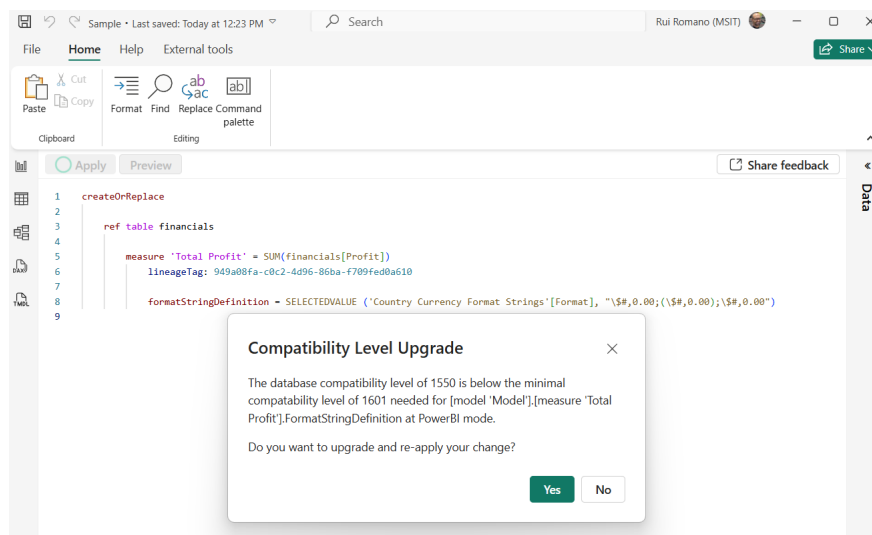


Automatic code actions offering useful quick fixes to your TMDL code, such as generating a new lineage tag or correcting misspelled property names.



The semantic model compatibility level defines the features accessible within a Power BI semantic model, which Power BI Desktop updates automatically. The TMDL view now also upgrades the compatibility level

automatically when utilising a feature that necessitates this upgrade. This functionality is particularly beneficial when copying scripts from other semantic models that may operate at a higher compatibility level.



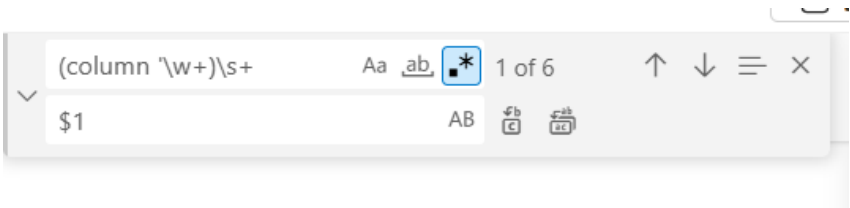
Renaming columns in TMDL view was previously possible, but the old names would return after a Refresh. After renaming the TMDL view, a Power Query rename step had to be added:

```

160 | partition financials = m
161 | mode: import
162 | source =
163 | let
164 |     Source = Excel.Workbook(File.Contents("C:\Program Files\WindowsApps\Microsoft.MicrosoftPowerBI
165 |     financials_Table = Source[[Item="financials",Kind="Table"]][Data],
166 |     #"Changed Type" = Table.TransformColumnTypes(financials_Table,{{"Segment", type text}, {"Cc
167 |     Int64.Type}, {"Gross Sales", type number}, {"Discounts", type number}, {"Sales", type numbe
168 |     #"Renamed Columns" = Table.RenameColumns(#"Changed Type",{{"Month Name", "Month"}})
169 | in
170 |     #"Renamed Columns"

```

This requirement has been removed and it is now possible to rename your columns using TMDL view or external tools without updating the Power Query expression. This can be very useful for scenarios involving bulk renaming of columns. For instance, if there is a need to remove spaces from all columns, the following RegEx find and replace can be utilised:



After performing a find and replace and applying the changes, the column name will differ from the sourceColumn. In this state, Power BI Desktop will maintain this change and stop synchronising the column names of your semantic model with those in Power Query. The connection between the semantic model columns and the query columns is determined solely by the sourceColumn property.

```

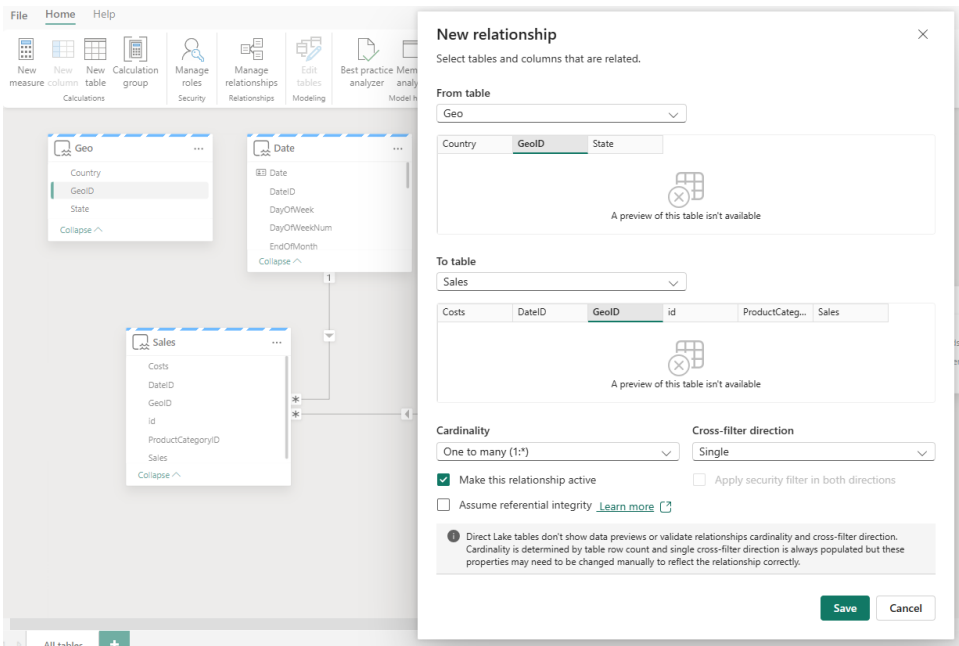
40 | column 'UnitsSold'
41 |     dataType: double
42 |     lineageTag: 34d60fd1-0c2a-46eb-bea8-3a8bdbdcf9fa
43 |     summarizeBy: sum
44 |     sourceColumn: Units Sold
45 |

```

Columns with different names from sourceColumn will no longer have automatic Power Query steps added.

Direct Lake relationship improvements

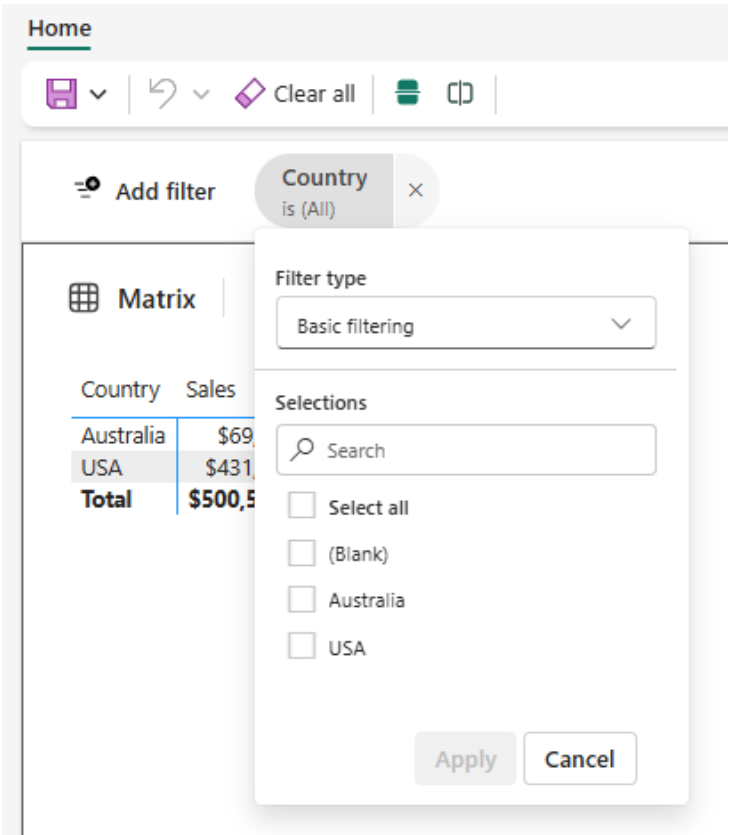
Creating relationships with Direct Lake tables has now become easier. When dragging and dropping columns to create a relationship, the cardinality is now populated based upon the number of rows in each table. The table with a higher number of rows is chosen to be the many-side and the other table is chosen to be on the one-side of the relationship.



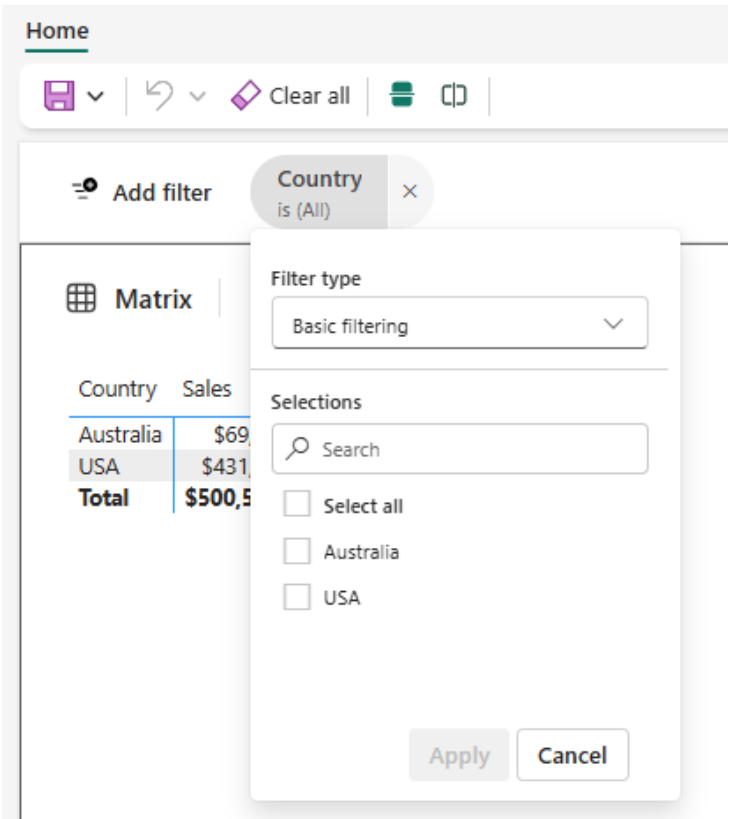
As you are data modelling, to be transparent on the limitations of relationships editor with Direct Lake tables, additional information is shown: "Direct Lake tables don't show data previews or validate relationships cardinality and cross-filter direction. Cardinality is determined by table row count and single cross-filter direction is always populated but these properties may need to be changed manually to reflect the relationship correctly".

'Assume referential integrity' is also now available for relationships between Direct Lake tables. When this is checked, slicers will no longer show a blank option and assume every value has a matching value in both tables.

Without assume referential integrity enabled on the relationship, a blank option is showing:



After assuming referential integrity enabled, no blank option is shown:



Direct Lake and Import tables in the same semantic model

Direct Lake on OneLake semantic models were announced recently in Power BI Desktop, simplifying semantic model and report creation with OneLake data. Now, you can also include import tables from any data source alongside Direct Lake tables in the same model.

Import tables can come from any supported data source and relationships between Direct Lake on OneLake and Import tables are regular relationships. Small dimension or lookup tables already in Direct Lake storage mode can instead use import storage mode, giving you the option to extend the table with calculated columns and structuring the table with hierarchies for use in Power BI reports and Excel PivotTables.

Microsoft is still working on adding composite semantic model authoring to the web and live editing in Power BI Desktop. Meanwhile, you can use XMLA and community-based tools like Tabular Editor, Fabric Studio and Semantic Link Labs to author them now. Once created, in addition to continuing to use those tools, you can open these models in the web and make semantic models changes, such as adding measures, relationships, columns, hierarchies and other semantic model objects, but you can't perform a schema refresh of the tables yet.

New Google BigQuery connector implementation (Preview)

To enhance the integration with Google BigQuery, this update brings a new implementation for Google BigQuery connector, currently available in Preview. It uses Arrow Database Connectivity (ADBC) instead of ODBC to connect to and retrieve data from Google BigQuery which improves performance especially for large result sets.

To access this feature, in Power BI Desktop, navigate to **Options and settings** (under the File Menu) -> **Options** -> **Preview features**, click

the checkbox to enable the 'Use new Google BigQuery connector implementation' option. Once the option is on, all the newly created connections will automatically use the new connector implementation.

Your existing connections will remain unchanged. You may also test the new feature by editing the queries. When you use the On-premises Data Gateway, make sure you have the latest version to use this feature.

Connect to the Vertica database with the user-installed ODBC driver – updates (Preview)

Back in April, Microsoft announced the option of using the user-installed ODBC driver to connect to Vertica database is enabled by default in Power BI Desktop. With this release, this option will be enabled by default in On-premises Data Gateway. Additionally, you will need to install the ODBC

driver on your gateway machine to use the Vertica database connector.

If you would like to revert to the old built-in driver experience, you need to update the gateway configuration setting.

Connect to the Oracle database with the built-in driver – updates (Preview)

Also in April, a new option was provided to use the built-in driver to connect to Oracle database in Power BI Desktop. With this release, you can now use this Oracle managed ODP.NET driver in On-premises Data Gateway as well. This feature removes the necessity for users to install and manage the driver.

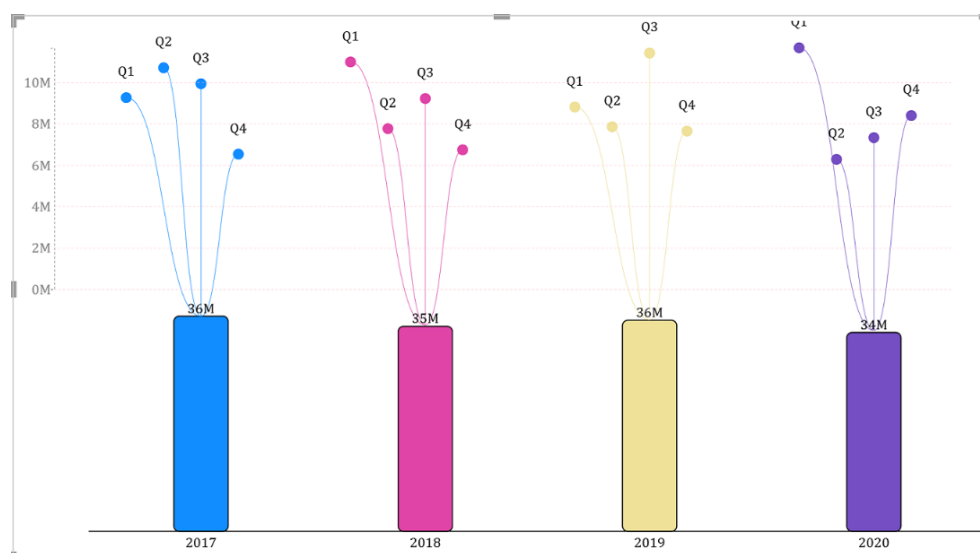
Improvements in the Snowflake connector (Preview)

This month sees the query tag support added to enable you to easily identify the queries issued from Power Query in your Snowflake query history.

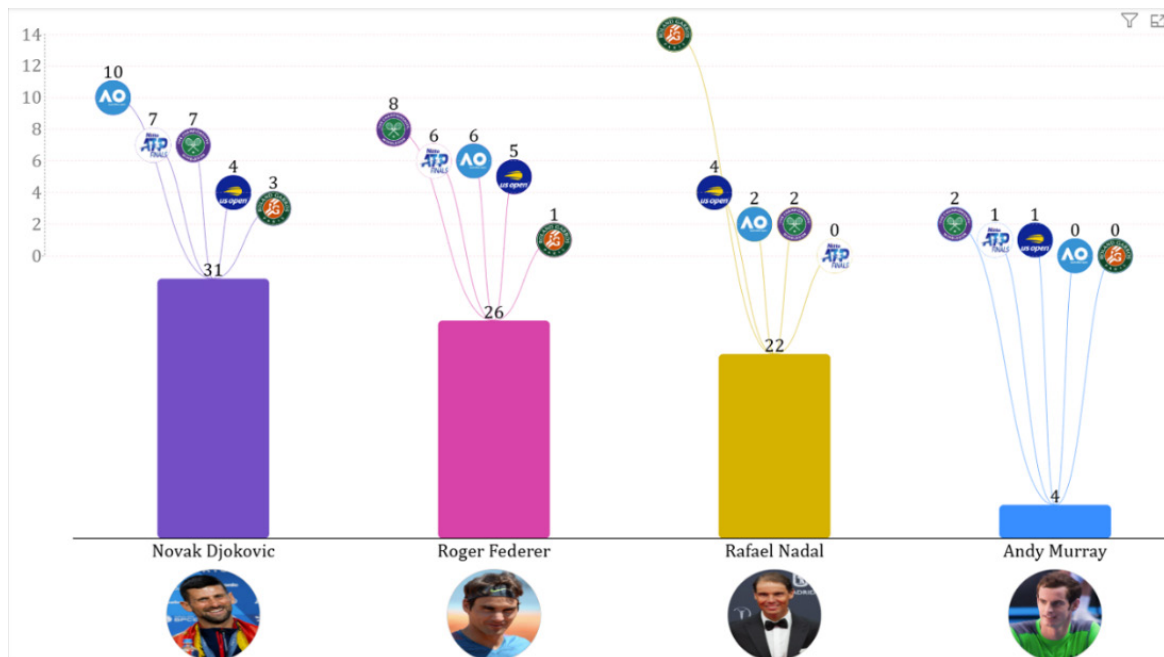
Dandelion chart

The Dandelion chart visualises your data in the form of a Column chart plus the breakdown of your data as Dandelion flowers above the columns. This visual offers a unique and holistic way to depict the parent / child relationship between two [2] fields in the 'Category' section of the visual.

For example, if you have yearly and monthly sales data, you can drag and drop 'Years' and 'Months' fields in the Category section and your Sales value field in the Measure section of this visual. Your annual data will be depicted as Column chart and its breakdown by months will be depicted as flowers above each column / bar.



The visual allows you to add images for each of the nodes as well.



You may download this visual from AppSource.

Dynamic Radial Bar chart by JTA

This is a Power BI custom visual that combines two [2] of the most effective and common charts in the data visualisation field, the Bar chart and the Radial chart.

With this visual, you can navigate and explore your data with multiple configuration options and drill down into your data. You may set global or category-specific targets, compare results and choose from multiple fields when drilling down all with full customisation.

Features of Dynamic Radial Bar Chart by JTA include:

- drill down into categories and sub-groups with a single click
- define a target value and respective reference line
- if you wish to show additional values in the ToolTip, just drag them to the Fields pane and define any explanation / description or hide unnecessary values
- adjust or disable animations to match your reporting style
- change the scale
- customise all the labels text in the visual
- change the formatting of all the values
- change the background
- customise the reference lines.

You may download the Dynamic Radial Bar Chart by JTA for free from AppSource.



DYNAMIC RADIAL BAR CHART BY JTA

PBI Certified

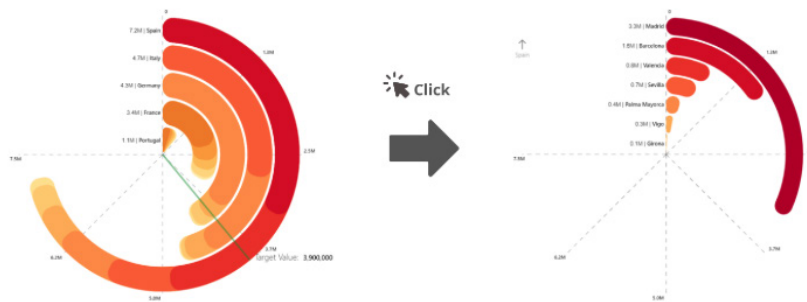


Interactive drilldown radial bar chart with multiple configurations

A PBI custom visual that combines two of the most effective and common charts in the data visualization field, the bar chart and the radial chart.

Easily navigate and explore your data with multiple configuration options.

Drill down into your data with ease. Set global or category-specific targets, compare results, and choose from multiple fields when drilling down—all with full customization.



- Seamlessly **drill down** into categories and sub-groups with a single click.
- Define a **target** value and respective **reference line**
- Want to show **additional** values in the **tooltip**? Just drag them to the Fields pane, and define any explanation/description or hide unnecessary values.
- Adjust or disable **animations** to match your reporting style.
- Change the **scale**
- Customize all the labels **text** in the visual
- Change the **formatting** of all the values
- Change the **background**
- Customize the **reference lines**

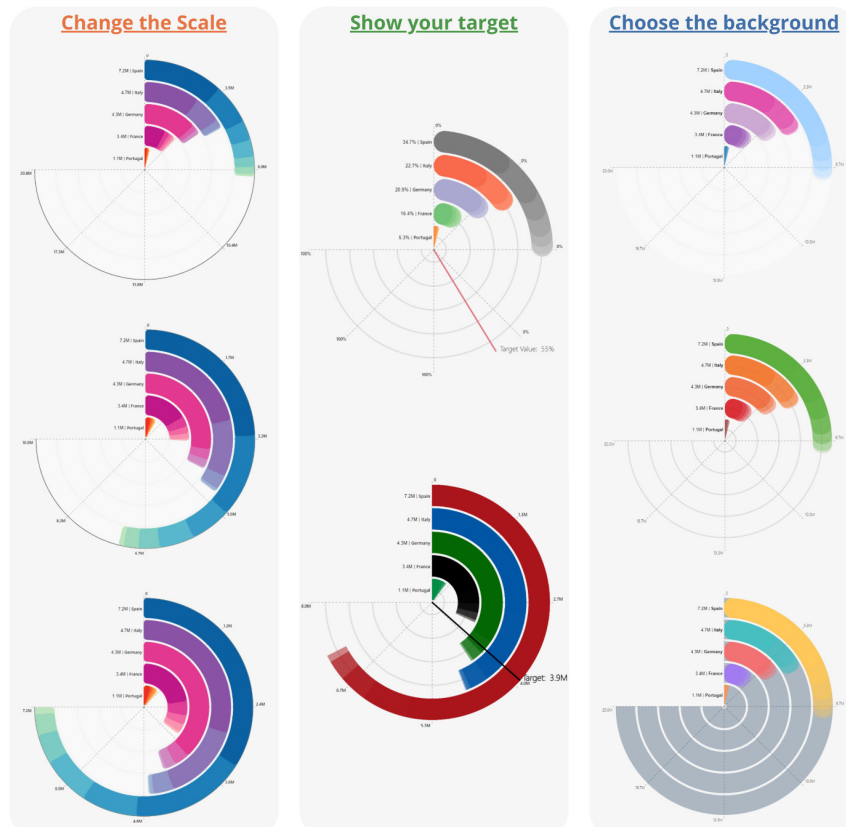


DYNAMIC RADIAL BAR CHART BY JTA

PBI Certified



Interactive drilldown radial bar chart with multiple configurations



Action dots: an innovative way to visualise variances in Zebra BI Tables

Action dots transform data into clearer, visual insights, making variance tracking simpler, especially for dashboards where space may be tight. By combining smart sizing and colour, the size of each dot directly represents the magnitude of the variance, making it easy to identify areas of over- or underperformance. Action dots are a compact addition to data visualisation, offering another way to represent relative variances.

By saving space and simplifying comparisons, this visual type may assist business insights.

Zebra BI has also brought back “ragged hierarchies”, a feature originally available in Zebra BI Tables but temporarily unavailable after the 6.0 release. You may now manage multi-level reports, eliminating duplication issues and reducing load times.

Action dots - New visualization option for relative variances

zebra bi



KPI by Powerviz

KPI by Powerviz is another custom visual for Power BI that allows users to visualise Key Performance Indicators (KPIs). Key features include 100+prebuilt KPI templates within the visual and the option to create your own templates too.

The design includes:

- 16 layers and 40+ chart variations to create infographic designs
- Various customisation, formatting options and colour styles
- You can create KPI objects in layers, combining charts, metrics and icons.

Analytical considerations include:

- **Data visualisation types:**
 - **Categorical:** compare values across categories
 - **Comparison:** analyse differences between values
 - **Composition:** show parts of a whole
 - **Progression:** display trends over time
 - **Actual vs Target:** compare actual against targets
- **Formatting features:** configure the Ranking, Sorting, Axis, Number-Formatting, ToolTip, Gridlines, Data Labels and Series Labels for visuals
- **IBCS theme support:** includes deviation bars, series labels and consistent colour schemes
- **Small multiples:** support for all chart types, including Fixed / Fluid with a change chart feature.

Other features include multi-categories comparison, highlight values, layer flexibility, and more.

You may try this KPI Visual for free from AppSource.

INTRODUCING KPI BY POWERVIZ

Most advanced KPI Visual with 16 Chart Layers, IBCS Theme Support, Conditional Formatting & On-object Interaction.

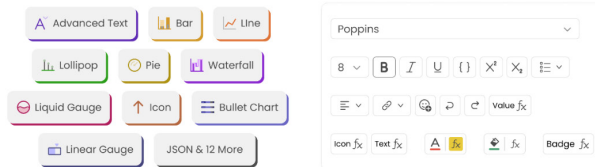
Don't settle for normal looking KPI Cards. Make them look stunning.



Unlock A World With Endless Possibilities With The KPI By Powerviz

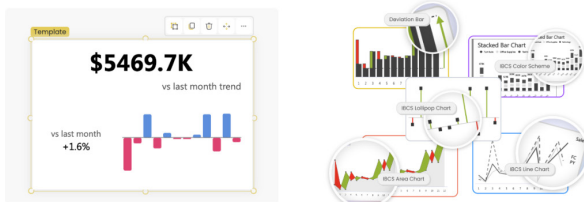
INTRODUCING KPI BY POWERVIZ

Advanced Text Editor, Highlighting & Styling Options, User Friendly UI, Multi Categories/Measures & More.



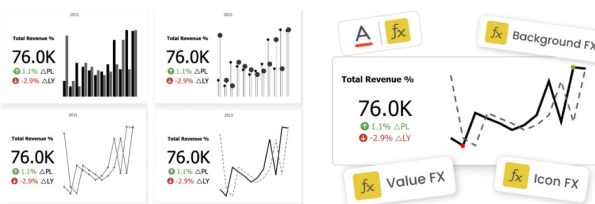
Upto 16 Layers And 40+ Chart Variations

Advanced Text Editor



On-Object Interaction

IBCS Support



Small Multiples

Conditional Formatting

Unlock A World With Endless Possibilities With The KPI By Powerviz

Drill Down Timeline PRO by ZoomCharts

Drill Down Timeline PRO is a column, line and area chart visualisation that is purpose-built for time series data. Users may easily drill down by simply clicking on data. For example, if the initial view shows aggregated monthly totals, clicking on a specific month will instantly reveal daily values. Creators can choose which time units are available to the user and

even create their own time unit hierarchy, from decades to milliseconds.

You may visualise up to 25 series on the same chart and combine different chart types. Each series can be fully customised so you can build the perfect visual for your use case, including Stacked and Clustered Columns, Candlestick charts, Streamgraphs, Histograms, Line charts, *etc.*

Features include:

- on-chart drill down
- up to 25 series (columns, lines and areas)
- 'change' indicators for comparison between two [2] values
- split series values by 'Legend' field
- custom date / time hierarchy (from milliseconds to decades)
- **DAX** measure support
- Customisation: defaults or individual series settings
- up to four [4] threshold lines and areas
- conditional formatting: series colours and value labels
- cross-chart filtering and touch support.

Again, this is available from AppSource.

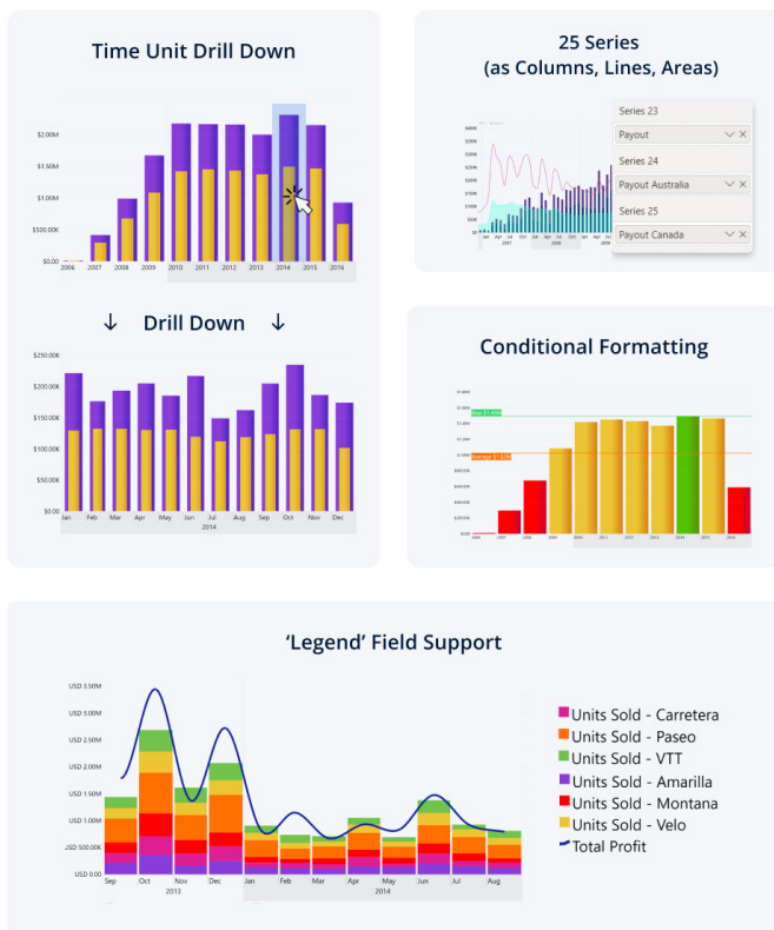


Drill Down Timeline PRO



SIMPLE AND POWERFUL

Create beautiful and interactive charts for your date/time data that reveal valuable insights with just a click.





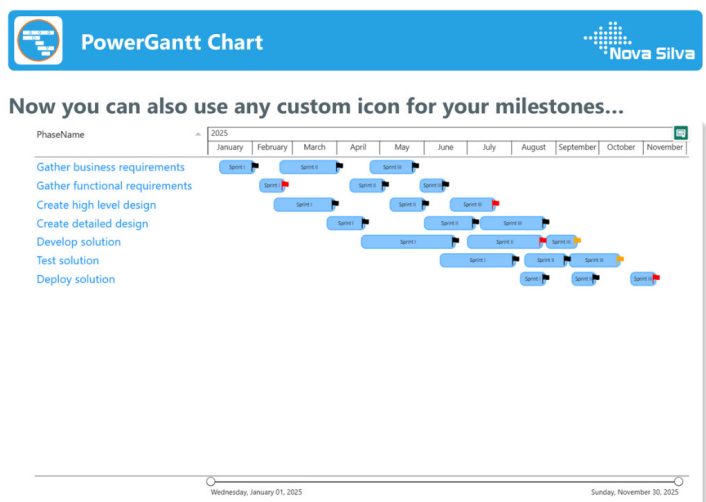
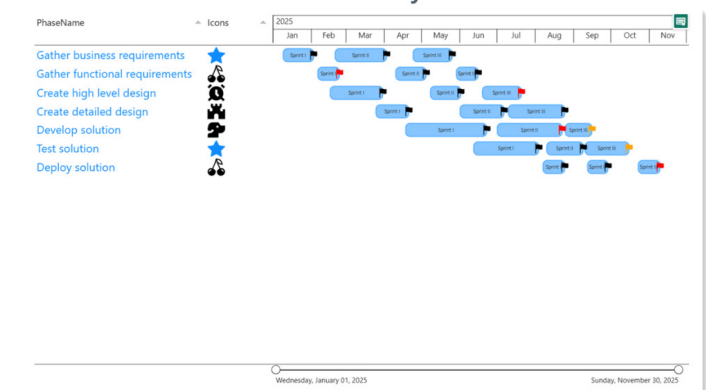
PEERLESS CUSTOMIZATION

Adapt this visual for any use case with over 300 configuration options.
Configure each series separately or apply the same settings to all series at once.

**PowerGantt chart by Nova Silva**

There is a new feature in the latest PowerGantt release: custom milestone markers. With this update, you can now add any icon to your PowerGantt visual to represent milestones.

These icons support conditional formatting, enabling dynamic styling based on status, category, or any field in your data. For example, you can automatically display a red warning icon for delayed tasks or a green checkmark for completed milestones.

**or use them as an additional column in your PowerGantt**

Custom icons give you the flexibility to match your visual to your organisation’s language and branding, making project tracking more intuitive and visually engaging.

You may try the PowerGantt Chart for free now on your own project data by downloading it from the AppSource.

Profitbase Gantt revamped

With the release of Profitbase Gantt v.3, this update offers an improved experience, having added user requested features, in addition to offering more flexibility in both data type and formatting options.

This visual now allows for the use of measures in more field buckets than in previous versions. In addition, conditional formatting has seen a significant overhaul, with added options for where to apply formatting.

Highlighted new features and improvements include:

- group column headers in the data grid
- zoom to fit button
- formatting options for milestones
- set focus period
- set default expansion level
- create your own timeline filters (e.g. for the next three weeks)
- copy data from the data grid
- layer placement of event ranges and today lines
- days off settings, which highlight weekends, custom dates and you may also choose to exclude from duration calculations.

You may create your own predefined period filters and choose which should be available in the filter list. In this case, selecting the ‘Weapons test phase’ from the custom filter dropdown, changes the timeline to show the dates specified for that custom filter:



‘Zoom to fit’ will adjust the timeline zoom level so the start of the first task and the end of the last task will fit within the timeline, without having to scroll horizontally.



The new features will supplement the pre-existing key features, which include:

- unlimited hierarchy
- format hierarchy levels individually
- supports additional columns
- supports multiple milestones per task apply expansion styles
- zoom in / out using zoom buttons
- apply highlighting curtains to important periods
- sort, filter and search for any information across all columns
- use theming templates to set formatting
- ragged hierarchy
- dynamic event markers from data.

And yes, you can get the visual from AppSource!

More next month.

New Features for Excel

This month introduces several new features across platforms, including value tokens, that infernal Dark Mode and the ability to create Tables with Copilot in Excel (yes, **CTRL + T** was too difficult).

Here is this month's list:

Excel for Windows

- Introducing value tokens in Excel (Insiders)
- Dark Mode in Excel
- ActiveX disabled by default (Insiders)

Excel for Windows and web

- Table creation with Copilot in Excel
- Upload phone images with Copilot

Excel for Windows and Mac

- View multiple worksheets side by side (Insiders).

Let's get started.

Introducing value tokens in Excel (Insiders)

Microsoft Excel copes with more than just plain text, numbers and errors. For example, over the years Microsoft has introduced the ability to put stocks, geography, currencies and formatted number values, amongst other data types, into cells, allowing them to hold what is known as "richer" data.

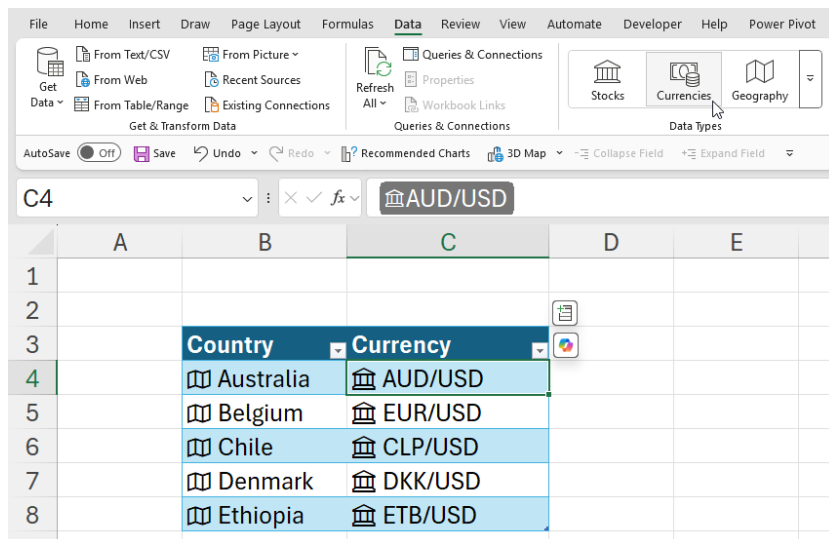
Due to their history and origination profiles, these data types were never able to be shown in the Formula bar in a coherent, cohesive and consistent way – until now. This enhancement in Excel for Windows provides a clear visual indication that a cell contains a data type or other richer formats. The addition of an icon next to the value further helps differentiate between data types in a spreadsheet.

This new feature works as follows:

- Open a new or existing Excel spreadsheet in Excel for Windows
- Select a cell and enter a geography, currency or another data type
- Select **Data -> Data Types**, select the data type you want (e.g. Currencies, Geography) and notice that a value token appears in the cell.

In the future, Microsoft plans to integrate value tokens into Excel's formula editing experience more deeply. Can't wait.

Value tokens are available to Excel for Windows users running Version 2502 (Build 18623.20020) or later. They will be available in Excel for Mac, Excel for Android and iOS, and Excel for the web soon.

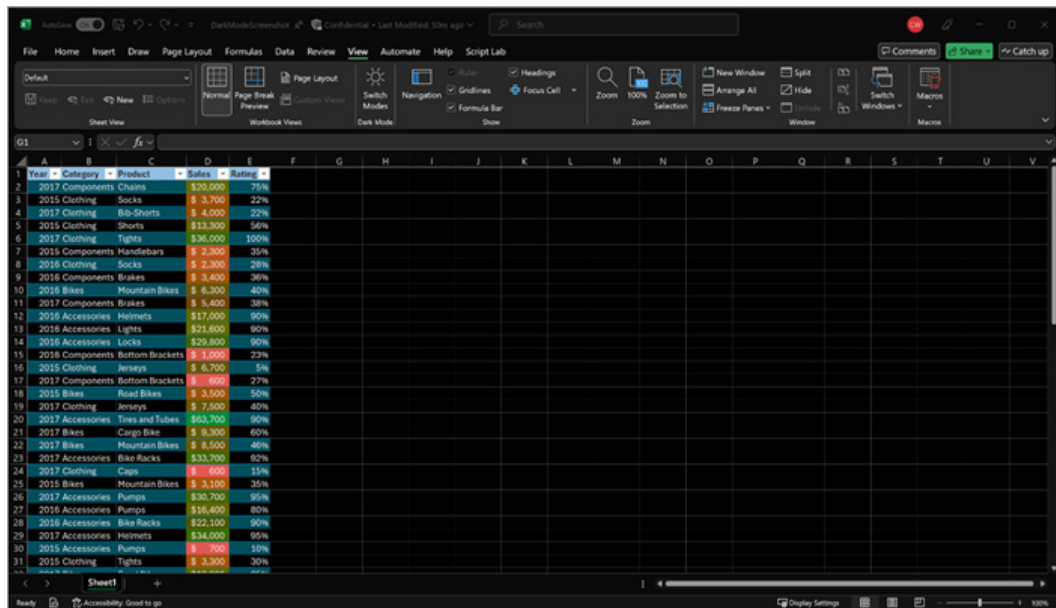


Dark Mode in Excel

Excel for Windows now supports Dark Mode, which offers a dark colour scheme for both the menu controls and the sheet background. Dark Mode can help reduce eye strain and improve accessibility for individuals with light sensitivity, while providing a modern look and feel. It is also becoming the daily default on some PCs which is particularly annoying

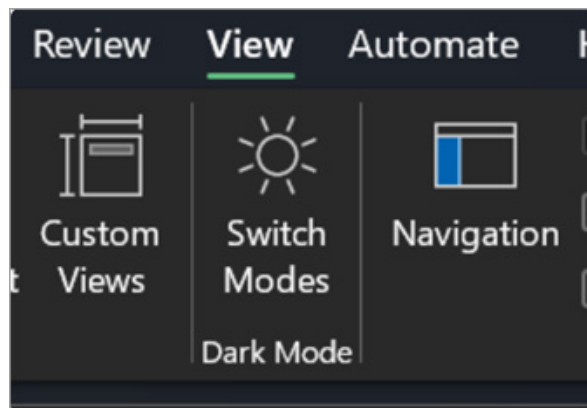
for those of us that don't want Dark Model.

The dark sheet background does not convey how your worksheet information will print, or the default view your collaborators will see when they open it.



To use Dark Mode in Excel, you must first set your Office theme or Windows theme to Black:

1. Open an existing worksheet or create one
2. To make sure you are using a supported Office theme, go to **File -> Account -> Office Theme** and confirm that either **Black** or **Use system setting** is selected
3. To turn on Dark Mode, select the **View** tab, then **Switch Modes** button



4. To turn Dark Mode off, select **View -> Switch Modes** again.

Regardless of your Dark Mode settings, your information will print with the light mode page colour. Also, your Dark Mode settings do not impact your colleagues and Excel will respect individual view preferences. To preview your document for printing and sharing, use the **Switch Modes** button to change the page background to light.

ActiveX disabled by default (Insiders)

Now this is a little more controversial.

ActiveX is a powerful technology that enables interactions within Microsoft 365 applications, but it is recognised that its deep access to system resources may also increase security risks. With this update, the Windows versions of Microsoft Word, Microsoft Excel, Microsoft PowerPoint and Microsoft Visio will have a new default configuration for ActiveX controls: **Disable all controls without notification**.

The previous default setting, **Prompt me before enabling all controls with minimal restrictions**, allowed you to enable potentially dangerous ActiveX controls, which could be exploited by attackers through social engineering or malicious files. The new default setting is more secure because it blocks these controls entirely, reducing the risk of malware or unauthorised code execution.

To see how it works:

1. Open a file that contains ActiveX controls
2. Notice a business bar appears at the top that reads **BLOCKED CONTENT: The ActiveX content in this file is blocked**, with an option to [learn more](#)



NOTE: This notification will only appear if you have not manually set the ActiveX settings in the Trust Center prior to this default change. When ActiveX is disabled, you will no longer be able to create or interact with ActiveX objects in Microsoft 365 files. Some existing ActiveX objects will still be visible as a static image, but it will not be possible to interact with them

3. To re-enable ActiveX in a file, select **File -> Options -> Trust Center**, then select the **Trust Center Settings** button
4. In the **Trust Center** dialog box, select **ActiveX Settings -> Prompt me before enabling all controls with minimal restrictions**, and then select the **OK** button.

It should be noted that if the ActiveX settings page is greyed out, it means your administrator has configured this policy centrally. Please contact your administrator for support. Administrators can enable ActiveX for their tenant in Group Policy Editor by selecting **Group Policy Path -> User configuration -> Administrative templates -> Microsoft Office 2016 -> Security Settings -> Disable All ActiveX -> Disabled**.

Alternatively, Cloud policies may be deployed instead.

This feature is available to all Beta Channel users and is rolling out to Current Channel (Preview) users running Version 2504 (Build 18730.20030) or later.

Table creation with Copilot in Excel

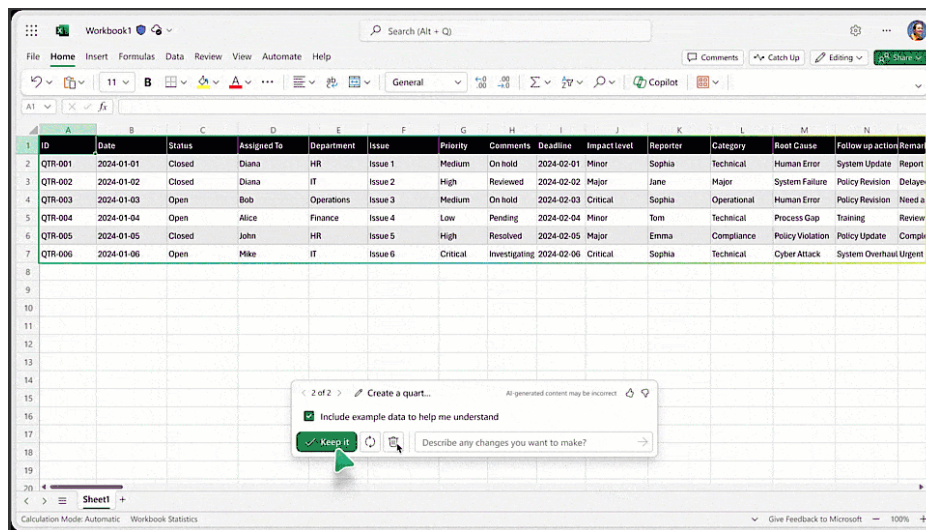
For users who don't have time to analyse text data (!), Copilot can now quickly identify sentiments in text and summarise themes, providing sources for efficient validation. Copilot can then create charts or PivotTables to help users uncover deeper insights into key issues or trends

from their text analysis. Additionally, Copilot now supports analysing tens of thousands of rows, so users can save time and uncover deeper insights across larger text datasets. This feature has just rolled out.



As a reminder, for users who are not sure where to begin in Excel, now they can just tell Copilot what they want to create and Copilot will suggest and refine a template with headers, formulae and visuals to get them going. Users may either choose from suggested prompts or simply

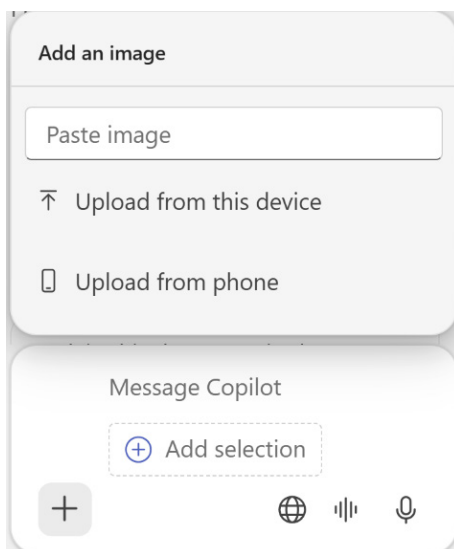
describe what they want and Copilot will create a preview Table. Then users can keep, regenerate, refine or discard it with a click. This feature rolled out on the web back in March.



Upload phone images with Copilot

Microsoft 365 Copilot now enables you to import your phone's photos directly into your Excel, PowerPoint or Word workspace in Excel for Windows and the web, so you may transform visual data into actionable insights without the hassle of emailing or downloading files. To do this, simply open a document on desktop or web, click Copilot from the Home tab, tap the + button in the Copilot prompt box, select 'Add an image',

and select 'Upload from phone'. A QR code appears for the user to scan with their mobile device, enabling you to quickly snap a new picture or choose an existing image from their gallery. Once uploaded, the image (like photos of handwritten notes, whiteboard sketches, or receipts) appears in the Copilot chat where you may ask it to extract and process text, summarise information or answer any questions you have.



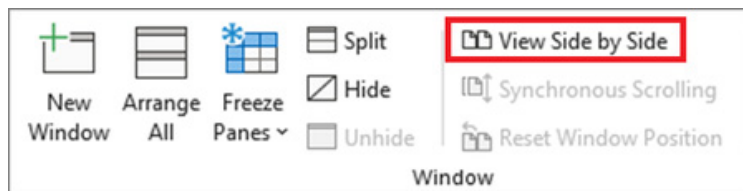
View multiple worksheets side by side (Insiders)

Some users collaborate on several Excel worksheets at a time or reference one worksheet while working on another. Now, Excel for Mac supports the ability to view multiple worksheets from the same or different workbooks side by side, providing benefits such as:

- **Enhanced productivity:** viewing worksheets simultaneously allows for comparison and data analysis without the need to constantly switch tabs
- **Efficient workflow:** this feature can speed up and streamline tasks like tracking changes, verifying data across different sheets and consolidating information
- **Improved accuracy:** by having multiple sheets open side by side, you can ensure consistency and minimise errors when transferring or referencing data.

It works as follows:

1. Open a new or existing worksheet in Excel on your Mac device
2. Select which sheet you would like to pop out in a new window
3. Select the View tab, then 'New Window'. This will open the selected sheet in a second window. To view worksheets from different workbooks, select 'New Window' in one of the workbooks, then **File -> Open** to find a second worksheet from another workbook
4. Select the 'View Side by Side' icon (two pieces of paper next to each other) to view the worksheets on top of one another.



Tips and tricks:

- If you rearrange the placement of the windows while working, you can select 'Reset Window Position' to reset them back to viewing side by side
- To scroll both worksheets at the same time, select 'Synchronous Scrolling' (this option is available only when 'View Side by Side' is turned on).

It's not perfect yet. One known issue is that the View Side by Side feature is not currently findable via the Search feature.

If you're interested in this feature for your Mac, this update is available to Excel for Mac users running Version 16.97 (Build 25041535) or later.

Until next month.

The A to Z of Excel Functions: PIVOTBY



The **PIVOTBY** function allows you to create a summary of your data via a formula, akin to a formulaic PivotTable. It supports grouping along two axes and aggregating the associated values. For instance, if you had a table of sales data, you might generate a summary of sales by state and year.

It should be noted that **PIVOTBY** is a function that returns an array of values that can spill to the grid. Furthermore, at this stage, not all features of a PivotTable appear to be replicable by this function.

The syntax of the **PIVOTBY** function is:

PIVOTBY(row_fields, col_fields, values, function, [field_headers], [row_total_depth], [row_sort_order], [col_total_depth], [col_sort_order], [filter_array], [relative_to])

It has the following arguments:

- **row_fields**: this is required and represents a column-oriented array or range that contains the values which are used to group rows and generate row headers. The array or range may contain multiple columns. If so, the output will have multiple row group levels
- **col_fields**: also required and represents a column-oriented array or range that contains the values which are used to group columns and generate column headers. The array or range may contain multiple columns. If so, the output will have multiple column group levels

- **values:** this is also required and denotes a column-oriented array or range of the data to aggregate. The array or range may contain multiple columns. If so, the output will have multiple aggregations
- **function:** also required, this is an explicit or eta reduced lambda (e.g. **SUM**, **PERCENTOF**, **AVERAGE**, **COUNT**) that is used to aggregate **values**. A vector of lambdas may be provided. If so, the output will have multiple aggregations. The orientation of the vector will determine whether they are laid out row- or column-wise
- **field_headers:** this and the remaining arguments are all optional. This represents a number that specifies whether the **row_fields**, **col_fields** and **values** have headers and whether field headers should be returned in the results. The possible values are:
 - **Missing:** Automatic
 - **0:** No
 - **1:** Yes and don't show
 - **2:** No but generate
 - **3:** Yes and show

It should be noted that "Automatic" assumes the data contains headers based upon the **values** argument. If the first value is text and the second value is a number, then the data is assumed to have headers. Fields headers are shown if there are multiple row or column group levels

- **row_total_depth:** this optional argument determines whether the row headers should contain totals. The possible values are:
 - **Missing:** Automatic, with grand totals and, where possible, subtotals
 - **0:** No Totals
 - **1:** Grand Totals
 - **2:** Grand and Subtotals
 - **-1:** Grand Totals at Top
 - **-2:** Grand and Subtotals at Top

It should be noted that for subtotals, **row_fields** must have at least two [2] columns. Numbers greater than two [2] are supported provided **row_field** has sufficient columns

- **row_sort_order:** again optional, this argument denotes a number indicating how rows should be sorted. Numbers correspond with the columns in **row_fields** followed by the columns in **values**. If the number is negative, the rows are sorted in descending / reverse order. A vector of numbers may be provided when sorting based upon only **row_fields**
- **col_total_depth:** this optional argument determines whether the column headers should contain totals. The possible values are:
 - **Missing:** Automatic, with grand totals and, where possible, subtotals
 - **0:** No Totals
 - **1:** Grand Totals
 - **2:** Grand and Subtotals
 - **-1:** Grand Totals at Top
 - **-2:** Grand and Subtotals at Top

It should be noted that for subtotals, **col_fields** must have at least two [2] columns. Numbers greater than two [2] are supported provided **col_field** has sufficient columns

- **col_sort_order:** again optional, this argument denotes a number indicating how they should be sorted. Numbers correspond with the columns in **col_fields** followed by the columns in **values**. If the number is negative, these are sorted in descending / reverse order. A vector of numbers may be provided when sorting based upon only **col_fields**
- **filter_array:** this represents a column-oriented one-dimensional array of Boolean values [1, 0] that indicate whether the corresponding row of data should be considered. It should be noted that the length of the array must match the length of **row_fields** and **col_fields**
- **relative_to:** this new, final argument allows you to summarise functions relative to row and column totals or the grand total. Five alternatives are possible:
 - **0:** Column Totals (default) (*Screentip: Calculation performed relative to all values in column*)
 - **1:** Row Totals (*Calculation performed relative to all values in row*)
 - **2:** Grand Total (*Calculation performed relative to all values*)
 - **3:** Parent Column Total (*Calculation performed relative to all values in column parent*)
 - **4:** Parent Row Total (*Calculation performed relative to all values in row parent*).

Similar in many ways to **GROUPBY**, **PIVOTBY** is fairly straightforward to use. Consider the following Excel Table called **tbl**:

Example Data

Table Used for Formulae

Year	Category	Item	Sales	Rating
2020	Components	Wheels	4,000	10%
2022	Components	Pedals	3,200	50%
2020	Components	Brakes	3,300	45%
2020	Clothing	Jerseys	1,100	10%
2020	Components	Saddles	500	85%
2020	Clothing	Jerseys	1,500	30%
2021	Accessories	Bike Racks	2,600	85%
2020	Bikes	Touring Bikes	1,100	30%
2022	Clothing	Tights	800	65%
2021	Clothing	Bib-Shorts	1,000	45%
2021	Accessories	Helmets	2,700	45%
2020	Clothing	Gloves	800	20%
2022	Clothing	Vests	1,100	30%
2021	Components	Brakes	1,100	100%
2022	Components	Handlebars	3,200	25%
2022	Accessories	Locks	400	55%
2021	Accessories	Tyres and Tubes	500	70%
2020	Components	Pedals	1,000	45%
2021	Accessories	Helmets	3,600	60%
2020	Bikes	Touring Bikes	200	55%
2021	Clothing	Gloves	4,000	100%
2020	Accessories	Locks	1,500	75%
2022	Bikes	Road Bikes	600	75%
2022	Clothing	Gloves	900	65%
2022	Components	Chains	100	10%
2022	Components	Chains	1,600	45%
2021	Bikes	Touring Bikes	2,400	70%

Consider the following formula:

=PIVOTBY(tbl[Category],tbl[Year],tbl[Sales],AVERAGE)

	2020	2021	2022	Total
Accessories	2,059	1,795	2,065	1,974
Bikes	1,659	1,926	2,062	1,885
Clothing	2,204	2,019	1,990	2,072
Components	2,116	2,064	1,947	2,038
Total	2,007	1,948	2,014	1,990

=PIVOTBY(tbl[Category],tbl[Year],tbl[Sales],AVERAGE)

You can get more imaginative and sort in descending order by the **AVERAGE** of **Rating**, viz.

=PIVOTBY(tbl[Item],tbl[Year],tbl[Rating],AVERAGE,,-2)

	2020	2021	2022	Total
Brakes	46%	61%	65%	59%
Locks	50%	58%	71%	56%
Pumps	70%	41%	56%	56%
Shorts	52%	52%	64%	55%
Vests	54%	51%	60%	55%
Touring Bikes	54%	51%	59%	55%
Pedals	58%	57%	50%	54%
Bib-Shorts	43%	48%	69%	54%
Gloves	49%	64%	52%	53%
Road Bikes	46%	54%	58%	53%
Cargo Bikes	56%	51%	51%	53%
Saddles	63%	43%	47%	52%
Tights	56%	53%	46%	51%
Handlebars	53%	52%	47%	51%
Caps	60%	39%	51%	51%
Chains	52%	49%	51%	51%
Socks	50%	45%	56%	50%
Wheels	41%	67%	45%	50%
Tyres and Tubes	46%	54%	46%	50%
Lights	47%	46%	49%	47%
Mountain Bikes	48%	42%	48%	46%
Bottom Brackets	48%	42%	45%	46%
Helmets	47%	43%	47%	45%
Bike Racks	31%	51%	54%	44%
Jerseys	27%	53%	45%	44%
Total	50%	50%	53%	51%

=PIVOTBY(tbl[Item],tbl[Year],tbl[Rating],AVERAGE,,-2)

You can get more sophisticated. Let’s create another Table, **Data**, slightly different to the original source data:

Source Table

Year	Quarter	Category	Item	Sales	Rating
2022	Q1	Components	Wheels	\$ 4,000	10%
2024	Q1	Components	Pedals	\$ 3,200	50%
2022	Q4	Components	Brakes	\$ 3,300	45%
2022	Q4	Clothing	Jerseys	\$ 1,100	10%
2022	Q2	Components	Saddles	\$ 500	85%
2022	Q1	Clothing	Jerseys	\$ 1,500	30%
2023	Q2	Accessories	Bike Racks	\$ 2,600	85%
2022	Q1	Bikes	Touring Bikes	\$ 1,100	30%
2024	Q4	Clothing	Tights	\$ 800	65%
2023	Q1	Clothing	Bib-Shorts	\$ 1,000	45%
2023	Q1	Accessories	Helmets	\$ 2,700	45%
2022	Q4	Clothing	Gloves	\$ 800	20%
2024	Q3	Clothing	Vests	\$ 1,100	30%
2023	Q4	Components	Brakes	\$ 1,100	100%
2024	Q2	Components	Handlebars	\$ 3,200	25%
2024	Q3	Accessories	Locks	\$ 400	55%
2023	Q3	Accessories	Tyres and Tubes	\$ 500	70%
2022	Q4	Components	Pedals	\$ 1,000	45%
2023	Q4	Accessories	Helmets	\$ 3,600	60%
2022	Q4	Bikes	Touring Bikes	\$ 200	55%
2023	Q4	Clothing	Gloves	\$ 4,000	100%
2022	Q4	Accessories	Locks	\$ 1,500	75%
2024	Q1	Bikes	Road Bikes	\$ 600	75%
2024	Q3	Clothing	Gloves	\$ 900	65%
2024	Q1	Components	Chains	\$ 100	10%
2024	Q1	Components	Chains	\$ 1,600	45%
2023	Q4	Bikes	Touring Bikes	\$ 2,400	70%

Here, we have two [2] parent / child relationships:

- Year and Quarter
- Category and Item.

We may create another formulaic alternative to a PivotTable (with crafty formatting) using the following formula:

=PIVOTBY(Data[[Category]:[Item]],Data[[Year]:[Quarter]],Data[Sales],PERCENTOF)

1. Default Presentation

Sales as a Percentage, Displayed by Category and Item vs. Year and Quarter

Default

		2022	2022	2022	2022	2023	2023	2023	2023	2024	2024	2024	2024	Total
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	
Accessories	Bike Racks	10.7%	4.8%	2.8%	0.3%	1.6%	3.1%		5.2%	1.2%	5.0%	2.0%	11.2%	4.2%
Accessories	Helmets	4.4%	2.0%	4.4%	11.1%	2.8%	10.3%	10.9%	4.0%	1.8%	4.5%	4.7%	10.2%	5.8%
Accessories	Lights	6.4%	2.7%	6.4%	3.3%	5.1%	1.1%	4.2%	1.9%	2.1%	3.9%	1.4%	0.7%	3.3%
Accessories	Locks	5.1%	4.0%	2.9%	12.3%	3.1%	3.9%	1.0%	0.8%	3.2%	3.6%	1.5%	1.1%	3.6%
Accessories	Pumps	1.7%	3.7%	9.3%	0.8%	11.8%	0.1%		0.6%	7.0%	3.6%	2.2%	3.9%	3.7%
Accessories	Tyres and Tubes	5.4%	1.8%	3.9%	1.2%	5.0%	2.7%	7.5%	7.1%	3.3%	3.2%	5.9%	0.4%	3.9%
Bikes	Cargo Bikes	5.6%	2.5%	8.7%	6.4%	9.2%	6.3%	5.8%	10.4%	4.6%	8.7%	14.2%	9.0%	7.5%
Bikes	Mountain Bikes	3.4%	6.5%	3.7%	7.2%	5.6%	6.6%	4.6%	9.1%	7.3%	5.3%	8.7%	5.8%	6.2%
Bikes	Road Bikes	3.3%	2.9%	3.7%	10.7%	0.7%	8.7%	6.9%	3.7%	8.0%	4.3%	5.6%	7.1%	5.4%
Bikes	Touring Bikes	4.5%	3.7%	5.4%	5.3%	5.6%	5.4%	6.9%	7.5%	6.9%	6.2%	8.3%	4.9%	5.8%
Clothing	Bib-Shorts	2.0%	2.0%	4.6%	5.2%	2.2%	1.6%	0.4%	1.9%	1.8%	2.7%	4.9%	2.2%	2.6%
Clothing	Caps	2.1%	5.2%	3.1%	1.1%	4.4%	4.9%		0.7%	1.3%	6.3%	2.5%	0.5%	2.7%
Clothing	Gloves	4.3%	10.7%	3.9%	2.6%		2.6%	1.4%	6.0%	2.9%		2.9%	4.8%	3.6%
Clothing	Jerseys	0.8%	2.6%		7.5%	6.3%	6.8%	6.4%	4.1%	5.9%	0.1%	3.5%	4.6%	4.0%
Clothing	Shorts	3.1%	6.2%	0.7%	0.7%	2.0%	4.7%	3.7%	8.7%	2.3%	3.7%	1.2%	2.6%	3.4%
Clothing	Socks	4.6%	2.2%	6.7%	3.1%	1.7%	1.6%	1.8%	2.8%	7.0%	2.6%	0.3%	0.8%	3.0%
Clothing	Tights	1.1%		7.8%	2.6%	3.8%	4.5%	1.9%	6.2%	5.9%	1.8%	2.3%	2.2%	3.3%
Clothing	Vests	3.3%	3.0%	0.3%	2.1%	7.2%	2.8%	4.8%	2.2%	0.5%	2.7%	3.9%	5.4%	3.1%
Components	Bottom Brackets	6.1%	2.9%	3.0%	3.0%	2.5%		3.9%	4.0%		3.6%	2.5%	4.2%	3.0%
Components	Brakes	2.6%	3.3%		1.9%	7.5%		7.0%	0.6%	2.7%	0.8%	3.5%	2.9%	2.7%
Components	Chains	1.8%	3.8%	0.4%	3.1%	1.3%	2.0%	5.0%	3.5%	5.6%	3.3%	5.2%	4.1%	3.3%
Components	Handlebars	6.8%	8.5%	5.2%	2.5%	2.8%	7.8%	5.8%	1.8%	1.4%	2.7%	4.2%	2.4%	4.3%
Components	Pedals	0.1%	5.5%	2.2%	4.8%	4.7%	3.9%	1.8%	2.1%	7.8%	11.1%	2.9%	5.8%	4.4%
Components	Saddles	2.7%	5.8%	2.9%	0.9%	2.8%	3.0%	2.7%	3.2%	5.0%	6.5%		2.9%	3.3%
Components	Wheels	8.1%	3.7%	8.1%		0.1%	5.4%	5.6%	1.9%	4.5%	3.8%	5.6%	0.3%	3.9%
Total		100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

Note that each column of sales is represented as a percentage of that column (including the Total column). Whilst it was a great start, Microsoft received feedback that end users wanted to see percentages summarised in alternative ways – and that is what we will now address.

This recently introduced final argument, **relative_to**, behaves the same in scenario 0: Column Totals. This is the default view:

=PIVOTBY(Data[[Category]:[Item]],Data[[Year]:[Quarter]],Data[Sales],PERCENTOF,,,,,0)

2. Column Totals (Relative Value 0)

Sales as a Percentage, Displayed by Category and Item vs. Year and Quarter

Column Totals (Relative Value 0)

		2022	2022	2022	2022	2023	2023	2023	2023	2024	2024	2024	2024	Total
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	
Accessories	Bike Racks	10.7%	4.8%	2.8%	0.3%	1.6%	3.1%		5.2%	1.2%	5.0%	2.0%	11.2%	4.2%
Accessories	Helmets	4.4%	2.0%	4.4%	11.1%	2.8%	10.3%	10.9%	4.0%	1.8%	4.5%	4.7%	10.2%	5.8%
Accessories	Lights	6.4%	2.7%	6.4%	3.3%	5.1%	1.1%	4.2%	1.9%	2.1%	3.9%	1.4%	0.7%	3.3%
Accessories	Locks	5.1%	4.0%	2.9%	12.3%	3.1%	3.9%	1.0%	0.8%	3.2%	3.6%	1.5%	1.1%	3.6%
Accessories	Pumps	1.7%	3.7%	9.3%	0.8%	11.8%	0.1%		0.6%	7.0%	3.6%	2.2%	3.9%	3.7%
Accessories	Tyres and Tubes	5.4%	1.8%	3.9%	1.2%	5.0%	2.7%	7.5%	7.1%	3.3%	3.2%	5.9%	0.4%	3.9%
Bikes	Cargo Bikes	5.6%	2.5%	8.7%	6.4%	9.2%	6.3%	5.8%	10.4%	4.6%	8.7%	14.2%	9.0%	7.5%
Bikes	Mountain Bikes	3.4%	6.5%	3.7%	7.2%	5.6%	6.6%	4.6%	9.1%	7.3%	5.3%	8.7%	5.8%	6.2%
Bikes	Road Bikes	3.3%	2.9%	3.7%	10.7%	0.7%	8.7%	6.9%	3.7%	8.0%	4.3%	5.6%	7.1%	5.4%
Bikes	Touring Bikes	4.5%	3.7%	5.4%	5.3%	5.6%	5.4%	6.9%	7.5%	6.9%	6.2%	8.3%	4.9%	5.8%
Clothing	Bib-Shorts	2.0%	2.0%	4.6%	5.2%	2.2%	1.6%	0.4%	1.9%	1.8%	2.7%	4.9%	2.2%	2.6%
Clothing	Caps	2.1%	5.2%	3.1%	1.1%	4.4%	4.9%		0.7%	1.3%	6.3%	2.5%	0.5%	2.7%
Clothing	Gloves	4.3%	10.7%	3.9%	2.6%		2.6%	1.4%	6.0%	2.9%		2.9%	4.8%	3.6%
Clothing	Jerseys	0.8%	2.6%		7.5%	6.3%	6.8%	6.4%	4.1%	5.9%	0.1%	3.5%	4.6%	4.0%
Clothing	Shorts	3.1%	6.2%	0.7%	0.7%	2.0%	4.7%	3.7%	8.7%	2.3%	3.7%	1.2%	2.6%	3.4%
Clothing	Socks	4.6%	2.2%		3.1%	1.7%	1.6%	1.8%	2.8%	7.0%	2.6%	0.3%	0.8%	3.0%
Clothing	Tights	1.1%		7.8%	2.6%	3.8%	4.5%	1.9%	6.2%	5.9%	1.8%	2.3%	2.2%	3.3%
Clothing	Vests	3.3%	3.0%		2.1%	7.2%	2.8%	4.8%	2.2%	0.5%	2.7%	3.9%	5.4%	3.1%
Components	Bottom Brackets	6.1%	2.9%	3.0%	3.0%	2.5%		3.9%	4.0%		3.6%	2.5%	4.2%	3.0%
Components	Brakes	2.6%	3.3%		1.9%	7.5%		7.0%	0.6%	2.7%	0.8%	3.5%	2.9%	2.7%
Components	Chains	1.8%	3.8%	0.4%	3.1%	1.3%	2.0%	5.0%	3.5%	5.6%	3.3%	5.2%	4.1%	3.3%
Components	Handlebars	6.8%	8.5%	5.2%	2.5%	2.8%	7.8%	5.8%	1.8%	1.4%	2.7%	4.2%	2.4%	4.3%
Components	Pedals	0.1%	5.5%	2.2%	4.8%	4.7%	3.9%	1.8%	2.1%	7.8%	11.1%	2.9%	5.8%	4.4%
Components	Saddles	2.7%	5.8%	2.9%	0.9%	2.8%	3.0%	2.7%	3.2%	5.0%	6.5%		2.9%	3.3%
Components	Wheels	8.1%	3.7%	8.1%		0.1%	5.4%	5.6%	1.9%	4.5%	3.8%	5.6%	0.3%	3.9%
Total		100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

It is clear to see this is identical to the first output. But let's see what happens when we start playing with the final argument. Let's change this value to **1**: Row Totals.

=PIVOTBY(Data[[Category]:[Item]],Data[[Year]:[Quarter]],Data[Sales],PERCENTOF,,,,,1)

Now, each row of sales is represented as a percentage of that row (including the Total row), viz.

3. Row Totals (Relative Value 1)

Sales as a Percentage, Displayed by Category and Item vs. Year and Quarter

Row Totals (Relative Value 1)

		2022	2022	2022	2022	2023	2023	2023	2023	2024	2024	2024	2024	Total
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	
Accessories	Bike Racks	24.1%	10.9%	5.1%	0.7%	2.9%	5.6%		11.2%	2.5%	10.3%	3.7%	23.0%	100.0%
Accessories	Helmets	7.1%	3.3%	5.7%	16.9%	3.7%	13.1%	13.6%	6.1%	2.6%	6.7%	6.2%	15.0%	100.0%
Accessories	Lights	18.5%	7.9%	14.6%	8.8%	12.0%	2.5%	9.4%	5.4%	10.3%	3.4%	1.8%		100.0%
Accessories	Locks	13.2%	10.3%	6.0%	29.9%	6.5%	8.1%	2.1%	2.1%	7.6%	8.6%	3.2%	2.5%	100.0%
Accessories	Pumps	4.4%	9.5%	19.1%	1.9%	24.6%	0.3%		1.5%	16.5%	8.5%	4.7%	9.2%	100.0%
Accessories	Tyres and Tubes	12.9%	4.3%	7.6%	2.7%	9.8%	5.2%	14.0%	16.4%	7.4%	7.1%	11.6%	0.9%	100.0%
Bikes	Cargo Bikes	7.0%	3.1%	8.7%	7.5%	9.3%	6.2%	5.6%	12.5%	5.2%	9.9%	14.6%	10.2%	100.0%
Bikes	Mountain Bikes	5.2%	9.8%	4.5%	10.3%	6.9%	7.9%	5.4%	13.4%	10.2%	7.4%	10.9%	8.0%	100.0%
Bikes	Road Bikes	5.6%	5.0%	5.2%	17.4%	1.0%	11.9%	9.3%	6.1%	12.7%	6.8%	8.0%	11.1%	100.0%
Bikes	Touring Bikes	7.2%	5.9%	6.9%	8.0%	7.3%	6.8%	8.6%	11.6%	10.2%	9.1%	11.0%	7.2%	100.0%
Clothing	Bib-Shorts	7.1%	7.3%	13.0%	17.4%	6.5%	4.6%	1.1%	6.7%	5.9%	8.8%	14.5%	7.1%	100.0%
Clothing	Caps	7.5%	18.2%	8.6%	3.6%	12.6%	13.7%		2.4%	4.3%	20.1%	7.3%	1.7%	100.0%
Clothing	Gloves	11.0%	27.3%	8.0%	6.4%		5.4%	2.8%	14.8%	6.8%		6.2%	11.3%	100.0%
Clothing	Jerseys	1.9%	6.2%		16.6%	12.0%	12.8%	11.6%	9.4%	12.7%	0.1%	6.8%	9.9%	100.0%
Clothing	Shorts	8.7%	17.2%	1.6%	1.8%	4.5%	10.3%	8.1%	23.3%	5.8%	9.4%	2.7%	6.7%	100.0%
Clothing	Socks	14.5%	6.8%		17.0%	9.4%	4.4%	3.9%	4.4%	8.7%	20.3%	7.7%	0.7%	100.0%
Clothing	Tights	3.1%		17.9%	7.1%	9.0%	10.2%	4.2%	17.2%	15.5%	4.6%	5.4%	5.9%	100.0%
Clothing	Vests	9.8%	9.0%	0.6%	5.9%	17.5%	6.6%	11.1%	6.4%	1.4%	7.4%	9.6%	14.6%	100.0%
Components	Bottom Brackets	18.8%	8.8%	7.5%	8.8%	6.3%		9.5%	11.8%		10.2%	6.3%	11.8%	100.0%
Components	Brakes	9.2%	11.5%		6.2%	21.3%		19.0%	2.1%		2.4%	10.2%	9.4%	100.0%
Components	Chains	5.2%	10.8%	0.9%	8.5%	2.9%	4.6%	11.1%	9.7%	14.8%	8.6%	12.3%	10.6%	100.0%
Components	Handlebars	14.7%	18.5%	9.1%	5.1%	5.0%	13.4%	9.8%	3.7%	2.8%	5.4%	7.6%	4.8%	100.0%
Components	Pedals	0.2%	11.6%	3.8%	9.6%	8.1%	6.6%	3.0%	4.3%	15.1%	21.5%	5.1%	11.1%	100.0%
Components	Saddles	7.9%	16.7%	6.6%	2.5%	6.6%	7.0%	6.0%	8.8%	13.1%	17.2%		7.6%	100.0%
Components	Wheels	19.5%	8.8%	15.7%		0.3%	10.4%	10.5%	4.6%	10.0%	8.3%	11.2%	0.7%	100.0%
Total		9.4%	9.4%	7.5%	8.8%	7.6%	7.4%	7.3%	9.1%	8.6%	8.6%	7.7%	8.6%	100.0%

If you wish, you can show the sales as a percentage of the Grand Total, using **2**: Grand Total:

=PIVOTBY(Data[[Category]:[Item]],Data[[Year]:[Quarter]],Data[Sales],PERCENTOF,,,,,2)

4. Grand Total (Relative Value 2)

Sales as a Percentage, Displayed by Category and Item vs. Year and Quarter

Grand Total (Relative Value 2)

		2022	2022	2022	2022	2023	2023	2023	2023	2024	2024	2024	2024	Total
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	
Accessories	Bike Racks	1.0%	0.5%	0.2%	0.0%	0.1%	0.2%	0.0%	0.5%	0.1%	0.4%	0.2%	1.0%	4.2%
Accessories	Helmets	0.4%	0.2%	0.3%	1.0%	0.2%	0.8%	0.8%	0.4%	0.2%	0.4%	0.4%	0.9%	5.8%
Accessories	Lights	0.6%	0.3%	0.5%	0.3%	0.4%	0.1%	0.3%	0.2%	0.2%	0.3%	0.1%	0.1%	3.3%
Accessories	Locks	0.5%	0.4%	0.2%	1.1%	0.2%	0.3%	0.1%	0.1%	0.3%	0.3%	0.1%	0.1%	3.6%
Accessories	Pumps	0.2%	0.3%	0.7%	0.1%	0.9%	0.0%	0.0%	0.1%	0.6%	0.3%	0.2%	0.3%	3.7%
Accessories	Tires and Tubes	0.5%	0.2%	0.3%	0.1%	0.4%	0.2%	0.5%	0.5%	0.3%	0.3%	0.5%	0.0%	3.9%
Bikes	Cargo Bikes	0.5%	0.2%	0.7%	0.5%	0.7%	0.5%	0.4%	0.9%	0.4%	0.7%	1.1%	0.8%	7.5%
Bikes	Mountain Bikes	0.3%	0.6%	0.3%	0.6%	0.4%	0.5%	0.3%	0.8%	0.6%	0.5%	0.7%	0.5%	6.2%
Bikes	Road Bikes	0.3%	0.3%	0.3%	0.9%	0.1%	0.7%	0.5%	0.3%	0.7%	0.4%	0.4%	0.6%	5.4%
Bikes	Touring Bikes	0.4%	0.3%	0.4%	0.5%	0.4%	0.4%	0.5%	0.7%	0.6%	0.5%	0.6%	0.4%	5.8%
Clothing	Bib-Shorts	0.2%	0.2%	0.3%	0.5%	0.2%	0.1%	0.0%	0.2%	0.2%	0.2%	0.4%	0.2%	2.6%
Clothing	Caps	0.2%	0.5%	0.2%	0.1%	0.3%	0.4%	0.1%	0.1%	0.5%	0.2%	0.0%	0.0%	2.7%
Clothing	Gloves	0.4%	1.0%	0.3%	0.2%	0.2%	0.2%	0.1%	0.5%	0.2%	0.2%	0.4%	0.4%	3.6%
Clothing	Jerseys	0.1%	0.2%	0.7%	0.7%	0.5%	0.5%	0.5%	0.4%	0.5%	0.0%	0.3%	0.4%	4.0%
Clothing	Shorts	0.3%	0.5%	0.1%	0.1%	0.2%	0.3%	0.3%	0.8%	0.2%	0.3%	0.1%	0.2%	3.4%
Clothing	Socks	0.4%	0.2%	0.5%	0.3%	0.1%	0.1%	0.1%	0.3%	0.6%	0.2%	0.0%	0.1%	3.0%
Clothing	Tights	0.1%	0.6%	0.2%	0.3%	0.3%	0.3%	0.1%	0.6%	0.5%	0.2%	0.2%	0.2%	3.3%
Clothing	Vests	0.3%	0.3%	0.0%	0.2%	0.5%	0.2%	0.3%	0.2%	0.0%	0.2%	0.3%	0.5%	3.1%
Components	Bottom Brackets	0.6%	0.3%	0.2%	0.3%	0.2%	0.3%	0.4%	0.3%	0.4%	0.3%	0.2%	0.4%	3.0%
Components	Brakes	0.2%	0.3%	0.2%	0.2%	0.6%	0.5%	0.1%	0.2%	0.1%	0.3%	0.3%	0.3%	2.7%
Components	Chains	0.2%	0.4%	0.0%	0.3%	0.1%	0.2%	0.4%	0.3%	0.5%	0.3%	0.4%	0.3%	3.3%
Components	Handlebars	0.6%	0.8%	0.4%	0.2%	0.2%	0.6%	0.4%	0.2%	0.3%	0.2%	0.3%	0.2%	4.3%
Components	Pedals	0.0%	0.5%	0.2%	0.4%	0.4%	0.3%	0.1%	0.2%	0.7%	1.0%	0.2%	0.5%	4.4%
Components	Saddles	0.3%	0.5%	0.2%	0.1%	0.2%	0.2%	0.2%	0.3%	0.4%	0.6%	0.2%	0.3%	3.3%
Components	Wheels	0.8%	0.3%	0.6%	0.0%	0.0%	0.4%	0.4%	0.2%	0.4%	0.3%	0.4%	0.0%	3.9%
Total		9.4%	9.4%	7.5%	8.8%	7.6%	7.4%	7.3%	9.1%	8.6%	8.6%	7.7%	8.6%	100.0%

There are still two further scenarios – and this is why our example contained two parent / child relationships. The first is **3: Parent Column Total**:

=PIVOTBY(Data[[Category]:[Item]],Data[[Year]:[Quarter]],Data[Sales],PERCENTOF,,,,,,3)

5. Parent Column Totals (Relative Value 3)

Sales as a Percentage, Displayed by Category and Item vs. Year and Quarter

Parent Column Totals (Relative Value 3)

		2022	2022	2022	2022	2023	2023	2023	2023	2024	2024	2024	2024	Total
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	
Accessories	Bike Racks	59.1%	26.7%	12.5%	1.8%	14.7%	28.2%	37.1%	57.1%	6.4%	26.0%	9.5%	58.1%	100.0%
Accessories	Helmets	21.5%	10.0%	17.3%	51.2%	10.2%	35.9%	37.1%	16.8%	8.5%	21.8%	20.4%	49.3%	100.0%
Accessories	Lights	37.2%	15.8%	29.4%	17.6%	41.1%	8.4%	32.1%	18.4%	25.7%	49.3%	16.2%	8.8%	100.0%
Accessories	Locks	22.2%	17.3%	10.1%	50.4%	34.8%	43.0%	11.1%	11.1%	34.8%	39.2%	14.6%	11.4%	100.0%
Accessories	Pumps	12.6%	27.2%	54.7%	5.5%	93.2%	1.0%	5.7%	42.4%	21.9%	12.0%	23.7%	100.0%	100.0%
Accessories	Tires and Tubes	46.9%	15.5%	27.7%	9.9%	21.7%	11.4%	30.8%	36.2%	27.3%	26.3%	43.1%	3.3%	100.0%
Bikes	Cargo Bikes	26.8%	11.9%	33.0%	28.4%	27.7%	18.5%	16.7%	37.1%	13.1%	24.8%	36.5%	25.6%	100.0%
Bikes	Mountain Bikes	17.5%	32.9%	15.1%	34.5%	20.6%	23.5%	16.0%	38.8%	27.9%	20.3%	29.9%	21.9%	100.0%
Bikes	Road Bikes	17.0%	15.0%	15.6%	52.4%	3.6%	42.2%	32.7%	21.6%	32.9%	17.5%	20.7%	28.8%	100.0%
Bikes	Touring Bikes	25.8%	20.9%	24.6%	28.6%	21.4%	19.8%	25.1%	33.7%	27.1%	24.3%	29.4%	19.3%	100.0%
Clothing	Bib-Shorts	15.8%	16.2%	29.1%	38.9%	34.3%	24.2%	6.1%	35.4%	16.3%	24.2%	40.0%	19.5%	100.0%
Clothing	Caps	19.8%	48.0%	22.8%	9.4%	43.8%	47.7%	8.5%	12.9%	60.1%	21.9%	5.1%	100.0%	100.0%
Clothing	Gloves	20.9%	51.8%	15.2%	12.0%	23.5%	12.0%	64.5%	27.8%	25.6%	46.6%	100.0%	100.0%	100.0%
Clothing	Jerseys	7.7%	25.1%	67.2%	26.2%	27.9%	25.4%	20.4%	42.9%	0.4%	23.2%	33.5%	100.0%	100.0%
Clothing	Shorts	29.6%	58.7%	5.6%	6.1%	9.7%	22.3%	17.5%	50.5%	23.6%	38.2%	10.9%	27.3%	100.0%
Clothing	Socks	30.4%	14.3%	35.7%	19.6%	20.6%	18.3%	20.6%	40.5%	65.7%	24.9%	2.2%	7.2%	100.0%
Clothing	Tights	11.0%	63.7%	25.3%	22.1%	25.2%	10.3%	42.4%	49.3%	14.8%	17.2%	18.7%	100.0%	100.0%
Clothing	Vests	38.6%	35.4%	2.5%	23.4%	42.1%	15.8%	26.6%	15.4%	4.4%	22.3%	29.1%	44.2%	100.0%
Components	Bottom Brackets	42.8%	20.1%	17.0%	20.1%	22.9%	34.3%	42.8%	4.9%	35.9%	22.4%	41.8%	100.0%	100.0%
Components	Brakes	34.3%	42.7%	23.1%	50.2%	16.3%	44.9%	4.9%	28.2%	8.0%	33.1%	30.7%	100.0%	100.0%
Components	Chains	20.8%	12.4%	3.6%	33.3%	30.3%	31.8%	34.6%	18.6%	25.6%	22.6%	100.0%	100.0%	100.0%
Components	Handlebars	31.0%	38.9%	19.2%	10.8%	15.7%	42.0%	30.7%	11.7%	13.6%	26.1%	36.9%	100.0%	100.0%
Components	Pedals	0.9%	46.2%	14.9%	38.0%	36.8%	30.1%	13.5%	19.7%	28.6%	40.6%	9.7%	21.1%	100.0%
Components	Saddles	23.4%	49.5%	19.7%	7.3%	23.4%	24.5%	21.2%	31.0%	34.7%	45.3%	20.0%	100.0%	100.0%
Components	Wheels	44.2%	20.1%	35.7%	1.0%	40.4%	40.9%	17.7%	33.2%	27.6%	37.1%	2.2%	100.0%	100.0%
Total		26.8%	26.7%	21.5%	25.1%	24.3%	23.7%	23.2%	28.8%	25.7%	25.6%	23.1%	25.6%	100.0%

Here, the Total column is 100% throughout. It is a little confusing as, if anything, it looks a little like Scenario 1: Row Totals. This is because the column here refers to the headings in each column, *i.e.* **Year** and **Quarter**. You can see that for any row the sum of the four quarters for any given year totals 100% (including the Total row).

Finally, Scenario 4: Parent Row Total considers the other parent / child relationship:

=PIVOTBY(Data[[Category]:[Item]],Data[[Year]:[Quarter]],Data[Sales],PERCENTOF,,,,,,4)

6. Parent Row Totals (Relative Value 4)

Sales as a Percentage, Displayed by Category and Item vs. Year and Quarter

Parent Row Totals (Relative Value 4)

		2022	2022	2022	2022	2023	2023	2023	2023	2024	2024	2024	2024	Total
		Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	Q1	Q2	Q3	Q4	
Accessories	Bike Racks	31.7%	25.4%	9.5%	1.2%	5.4%	14.6%	44.9%	26.4%	6.6%	20.8%	11.4%	40.6%	17.0%
Accessories	Helmets	13.1%	10.7%	14.9%	38.4%	9.6%	48.4%	46.0%	20.2%	9.4%	18.9%	26.5%	37.2%	23.8%
Accessories	Lights	19.1%	14.4%	21.4%	11.2%	17.4%	5.1%	17.9%	9.9%	11.0%	16.4%	8.1%	2.6%	13.4%
Accessories	Locks	15.1%	20.8%	9.7%	42.3%	10.5%	18.5%	4.4%	4.3%	17.3%	15.2%	8.5%	3.8%	14.8%
Accessories	Pumps	5.1%	19.4%	31.3%	2.8%	40.0%	0.5%	3.1%	37.7%	15.2%	12.5%	14.3%	15.0%	15.0%
Accessories	Tires and Tubes	15.9%	9.3%	13.3%	4.1%	17.0%	12.7%	31.7%	36.1%	17.9%	13.5%	33.1%	1.5%	15.9%
Bikes	Cargo Bikes	33.4%	16.3%	40.5%	21.6%	43.4%	23.4%	24.0%	33.8%	17.0%	35.4%	38.5%	33.6%	30.1%
Bikes	Mountain Bikes	20.4%	41.5%	17.1%	24.3%	26.6%	24.4%	18.9%	29.8%	27.3%	21.8%	23.7%	21.5%	24.7%
Bikes	Road Bikes	19.4%	18.7%	17.4%	36.2%	3.4%	32.4%	28.6%	12.0%	29.9%	17.5%	15.2%	26.4%	21.8%
Bikes	Touring Bikes	26.8%	23.5%	24.9%	17.9%	26.6%	19.8%	28.6%	24.4%	25.8%	25.4%	22.6%	16.5%	23.4%
Clothing	Bib-Shorts	9.3%	6.4%	16.9%	20.8%	8.1%	5.5%	2.0%	6.0%	6.6%	13.6%	23.0%	9.4%	10.3%
Clothing	Caps	10.1%	16.4%	11.4%	4.3%	16.0%	16.7%	2.2%	4.9%	31.7%	11.8%	2.3%	10.5%	10.5%
Clothing	Gloves	20.2%	33.4%	14.4%	10.5%	10.5%	8.9%	6.8%	18.2%	10.4%	13.6%	20.9%	14.2%	14.2%
Clothing	Jerseys	3.8%	8.3%	30.0%	22.7%	23.2%	31.3%	12.6%	21.3%	0.3%	16.3%	19.8%	15.5%	15.5%
Clothing	Shorts	14.6%	19.4%	2.7%	2.7%	7.2%	15.8%	18.4%	26.6%	8.3%	18.6%	5.4%	11.5%	13.1%
Clothing	Socks	21.5%	6.7%	24.8%	12.6%	6.2%	5.3%	8.8%	8.7%	25.3%	13.3%	1.2%	3.3%	11.5%
Clothing	Tights	5.1%	28.8%	10.5%	13.8%	13.8%	15.1%	9.2%	18.9%	21.3%	8.9%	10.6%	9.7%	12.7%
Clothing	Vests	15.4%	9.4%	1.0%	8.5%	26.0%	9.4%	1.9%	13.6%	18.1%	23.2%	18.1%	12.2%	12.2%
Components	Bottom Brackets	21.5%	8.5%	13.8%	18.6%	11.6%	12.4%	23.1%	10.3%	10.3%	18.5%	12.2%	100.0%	100.0%
Components	Brakes	9.3%	9.8%	11.5%	34.3%	22.0%	3.5%	10.0%	2.4%	14.7%	13.1%	10.8%	10.8%	10.8%
Components	Chains	6.5%	11.3%	1.8%	19.3%	5.8%	9.1%	15.7%	20.5%	20.8%	10.4%	21.7%	18.0%	13.2%
Components	Handlebars	24.0%	25.5%	23.9%	15.4%	13.1%	35.1%	18.3%	10.4%	5.2%	8.5%	17.7%	10.7%	17.4%
Components	Pedals	0.4%	16.5%	10.1%	29.5%	21.6%	17.7%	5.7%	12.4%	28.9%	35.0%	12.2%	25.6%	17.8%
Components	Saddles	9.7%	17.4%	13.2%	5.6%	13.1%	13.7%	8.5%	18.6%	18.4%	20.6%	12.8%	13.1%	13.1%
Components	Wheels	28.6%	11.0%	37.1%	0.6%	40.4%	24.5%	17.6%	11.4%	16.7%	11.9%	23.4%	1.3%	15.6%
Total		100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

In this final illustration, the Total row is 100% throughout. This looks similar to the default Scenario 0: Column Totals. This is because the row here refers to the headings in each row, *i.e.* **Category** and **Item**. You can see that for any row the sum of any category for any given **Quarter** and **Year** totals 100% (including the Total column).

The A to Z of Excel Functions: PMT



This function is often referred to as the *mortgage calculator*. Here, the aim is to calculate what the regular repayment is per period to service and pay off a debt over a given amount of time – just like a common mortgage, but with a fixed (rather than a variable) interest rate.

For example, if we borrow \$300,000 over 25 years at an interest rate of 6% p.a. what will our regular monthly payments be (assuming no change of rate)?

The answer to this question is given by the formula:

$$P = Ai / (1 - (1 + i)^{-N}) \text{ where:}$$

- **P** = regular periodic payment
- **A** = amount borrowed
- **i** = periodic interest rate
- **N** = total number of repayment periods

(Interesting that the acronym for remembering the mortgage variables is **PAiN!**)

In our example, crunching the numbers (using a periodic interest rate of 0.50%, being 6% / 12 (since the interest does not compound as it is paid monthly) and total number of periods being 25 x 12 = 300) gives a monthly repayment of \$1,932.90, viz.

1. Minimum Payment PMT

Payments in Arrears

Scenario

Imagine you borrow an amount **A** to be repaid over **N** periods with a periodic interest rate of **i**. Assuming payments are made at the end of each period, the amount to be paid each period will be **P**:

$$P = Ai / (1 - (1 + i)^{-N})$$

Illustration

This formula shows how it works:

Amount (A)	\$ 300,000
Repayment Period (Years)	25
Repayment Period (Months) (N)	300
Annual Interest Rate (%)	6.00%
Monthly Interest Rate (%) (i)	0.50%
Monthly Payment	\$ 1,932.90

It should be noted that using **PMT** will give the same solution but be negative instead. This is because Excel's financial functions distinguish between cash inflows (positive) and outflows (negative).

+

PMT(rate, nper, pv, [fv], [type])

- **rate:** this is required and represents the constant interest rate for the loan
- **nper:** this is also required and denotes the total number of payments for the loan
- **pv:** also necessary, this is the present value, or the total amount that a series of future payments is worth now, also known as the principal (*i.e.* what you are borrowing)
- **fv:** this is the first of two optional arguments. This is the future value, or a cash balance you want to attain, after the last payment is made. If **fv** is omitted, it is assumed to be zero (0), *i.e.* the future value of a loan is nil
- **type:** this final argument is also optional. This the number zero (0) or one (1) and indicates when payments are due:

type	Payments due
0 or omitted	At the end of the period
1	At the beginning of the period

- the payment returned by **PMT** includes principal and interest but considers no taxes, reserve payments or other fees sometimes associated with loans
- 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 an annual interest rate of 12%, use 12%/12 for **rate** and 4*12 for **nper**. If you make annual payments on the same loan, use 12% for the **rate** and 4 for **nper**
- to find the total amount paid over the duration of the loan, multiply the returned **PMT** value by **nper**.

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

Beat the Boredom Suggested Solution

Earlier in this newsletter, we challenged you to produce row cumulative sums for a numeric array using only one [1] Excel formula, and to produce row cumulative products for the same array with, again, “just” one [1] formula.

The outputs were to look like the following upon completion:

Data	4	12	(9)	(5)	(4)	9	1	(12)	(7)	-
	-	10	(11)	(12)	7	(2)	-	9	(10)	(4)
	(5)	(10)	11	(11)	4	2	(6)	11	(1)	-
	(12)	(7)	4	(8)	6	(1)	(12)	8	7	1
	(5)	(3)	(5)	7	6	8	4	(12)	1	(3)
	10	(9)	(10)	6	(1)	8	-	-	(6)	(8)
	10	4	(2)	8	9	2	4	(7)	(2)	(9)
	(8)	(9)	4	(7)	6	(6)	(6)	6	(2)	(11)
	-	4	(5)	(8)	(5)	(1)	(3)	-	(3)	9

Cumulative Sum	4	16	7	2	(2)	7	8	(4)	(11)	(11)
	-	10	(1)	(13)	(6)	(8)	(8)	1	(9)	(13)
	(5)	(15)	(4)	(15)	(11)	(9)	(15)	(4)	(5)	(5)
	(12)	(19)	(15)	(23)	(17)	(18)	(30)	(22)	(15)	(14)
	(5)	(8)	(13)	(6)	-	8	12	-	1	(2)
	10	1	(9)	(3)	(4)	4	4	(2)	(10)	
	10	14	12	20	29	31	35	28	26	17
	(8)	(17)	(13)	(20)	(14)	(20)	(26)	(20)	(22)	(33)
	-	4	(1)	(9)	(14)	(15)	(18)	(18)	(21)	(12)

Cumulative Product	4	48	(432)	2,160	(8,640)	(77,760)	(77,760)	933,120	(6,531,840)	-
	-	-	-	-	-	-	-	-	-	-
	(5)	50	550	(6,050)	(24,200)	(48,400)	290,400	3,194,400	(3,194,400)	-
	(12)	84	336	(2,688)	(16,128)	16,128	(193,536)	(1,548,288)	(10,838,016)	(10,838,016)
	(5)	15	(75)	(525)	(3,150)	(25,200)	(100,800)	1,209,600	1,209,600	(3,628,800)
	10	(90)	900	5,400	(5,400)	(43,200)	-	-	-	-
	10	40	(80)	(640)	(5,760)	(11,520)	(46,080)	322,560	(645,120)	5,806,080
	(8)	72	288	(2,016)	(12,096)	72,576	(435,456)	(2,612,736)	5,225,472	(57,480,192)
	-	-	-	-	-	-	-	-	-	-

There were some requirements:

- each formula needed to be within one cell
- the function **LAMBDA** and any of its helper functions (e.g. **LET**, **BYROW** or **MAP**) were not allowed
- this was a formula challenge; no Power Query / Get & Transform or VBA.

Suggested Solution

Our suggested solution contains several stages. The steps are detailed below.

The OFFSET Approach

The **OFFSET** function has four [4] arguments:

=OFFSET(reference, rows, columns, [height], [width])

The argument **reference** accepts a cell or range of adjacent cells, and it is the base of the **OFFSET**. The arguments **rows** and **columns** are displacements from **reference**, i.e. the number of rows downwards and the number of columns to the right. They accept both positive and negative values, and negative values simply represent movements in the opposite direction. For example,

=OFFSET(B1, 2, -1)

would take us two [2] rows down and one [1] column to the left, ending up at cell **A3**.

	A	B	C	D	E	F
1	1	2	3	4	5	6
2	7	8	9	10	11	12
3	13	14	15	16	17	18
4	19	20	21	22	23	24
5	25	26	27	28	29	30
6	31	32	33	34	35	36

The optional arguments **height** and **width** are dimensions of the output array from **reference** after displacement, and they only accept positive values. Again, **height** and **width** are measuring downwards and to the right, and they both have a default value one [1].

Increasing the **width** argument of an **OFFSET** formula applied on the input array whilst keeping **height** as one [1] will produce longer and longer one-row segments from the input. This way, we can extract “running segments” from the array.

</

For example, from the top-left corner (**F13**) of our input array, if we specify a **width** of one [1]

=OFFSET(F13, 0, 0, 1, 1)

then we are outputting the cell **F13** itself. If we then specify a **width** of two [2]

=OFFSET(F13, 0, 0, 1, 2)

then we are outputting a spilled range **F13:G13**, the first two [2] cells in the first row of our input array. Following this idea, we can increase the **width** argument in the **OFFSET** to sequentially spill “running segments” of a row. For example, a **width** of nine [9] will give us the first nine cells in the first row (**F13:N13**):

F23

</

Hopefully, it is intuitive at this point that, if we can find a way to output all “running segments” with one [1] formula and output the totals of these spilled ranges instead of spilling them, we will obtain an output row of running totals.

The Rows Argument

Before implementing the above idea, let’s look at the **rows** argument in the **OFFSET** function and how we may cover all rows of the input array. The **rows** argument is the vertical displacement from **reference**, *i.e.* the number of rows downwards. Using the top-left corner **F13** as

the **reference** and increasing **rows** from zero [0] to eight [8], we can output from the first row to the ninth, or the last row of the array. For example, with a **rows** value of seven [7] and a **width** value of nine [9], we can output the first nine [9] cells of the second last row (**F20:N20**):

F23	=OFFSET(\$F\$13, 7, 0, 1, 9)														
1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
2	SP Cumulative Sum & Product.xlsm														
3	Navigator														
4	Error Checks:														
5															
6	1. OFFSET														
7															
8	Inputs														
9															
10	Number of rows														
11	Number of columns														
12															
13	Data														
14		4	12	(9)	(5)	(4)	9	1	(12)	(7)	-				
15		-	10	(11)	(12)	7	(2)	-	9	(10)	(4)				
16		(5)	(10)	11	(11)	4	2	(6)	11	(1)	-				
17		(12)	(7)	4	(8)	6	(1)	(12)	8	7	1				
18		(5)	(3)	(5)	7	6	8	4	(12)	1	(3)				
19		10	(9)	(10)	6	(1)	8	-	-	(6)	(8)				
20		10	4	(2)	8	9	2	4	(7)	(2)	(9)				
21		(8)	(9)	4	(7)	6	(6)	(6)	6	(2)	(11)				
22		-	4	(5)	(8)	(5)	(1)	(3)	-	(3)	9				
23	OFFSET output														
24		(8)	(9)	4	(7)	6	(6)	(6)	6	(2)					

What we need to do next is to figure out how to output all “running segments” of each row and all rows with only one [1] **OFFSET** formula. We also need to find a way to sum for each of the output spilled ranges.

However, **OFFSET** doesn’t work well with array inputs if we try to be dynamic with either **width** or **rows**. For example:

F23	=OFFSET(\$F\$13, SEQUENCE(9,0), 0, 1, 10)														
1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
2	SP Cumulative Sum & Product.xlsm														
3	Navigator														
4	Error Checks:														
5															
6	1. OFFSET														
7															
8	Inputs														
9															
10	Number of rows														
11	Number of columns														
12															
13	Data														
14		4	12	(9)	(5)	(4)	9	1	(12)	(7)	-				
15		-	10	(11)	(12)	7	(2)	-	9	(10)	(4)				
16		(5)	(10)	11	(11)	4	2	(6)	11	(1)	-				
17		(12)	(7)	4	(8)	6	(1)	(12)	8	7	1				
18		(5)	(3)	(5)	7	6	8	4	(12)	1	(3)				
19		10	(9)	(10)	6	(1)	8	-	-	(6)	(8)				
20		10	4	(2)	8	9	2	4	(7)	(2)	(9)				
21		(8)	(9)	4	(7)	6	(6)	(6)	6	(2)	(11)				
22		-	4	(5)	(8)	(5)	(1)	(3)	-	(3)	9				
23	OFFSET output														
24		#VALUE!													
25		#VALUE!													
26		#VALUE!													
27		#VALUE!													
28		#VALUE!													
29		#VALUE!													
30		#VALUE!													
31		#VALUE!													

The Excel engine allows some functions to spill, but not others due to limitations with the prohibition of creating arrays of arrays.

Even if we obtained an array of spilled “running segments” as planned, we couldn’t simply use **SUM** or **PRODUCT** to produce running totals or running products from those “running segments”. The reason is that **SUM** and **PRODUCT** are aggregate functions, and they will produce one

[1] output aggregating (adding / multiplying) everything in one [1] cell, instead of preserving the array structure.

So far, the intermediate output before addition or multiplication we can visualise to have the structure of an array with spilled rows or spilled “running segments”. In the **OFFSET** we will be using numbers one [1] to 10 for **width**, and zero [0] to eight [8] for **rows**.

J27	=OFFSET(\$G\$13, \$F27, 0, 1, J\$23)														
1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
2	SP Cumulative Sum & Product.xlsm														
3	Navigator														
4	Error Checks:														
5															
6	1. Spilling Array														
7															
8	Inputs														
9															
10	Number of rows														
11	Number of columns														
12															
13	Data														
14		4	12	(9)	(5)	(4)	9	1	(12)	(7)	-				
15		-	10	(11)	(12)	7	(2)	-	9	(10)	(4)				
16		(5)	(10)	11	(11)	4	2	(6)	11	(1)	-				
17		(12)	(7)	4	(8)	6	(1)	(12)	8	7	1				
18		(5)	(3)	(5)	7	6	8	4	(12)	1	(3)				
19		10	(9)	(10)	6	(1)	8	-	-	(6)	(8)				
20		10	4	(2)	8	9	2	4	(7)	(2)	(9)				
21		(8)	(9)	4	(7)	6	(6)	(6)	6	(2)	(11)				
22		-	4	(5)	(8)	(5)	(1)	(3)	-	(3)	9				
23	OFFSET output														
24		1	2	3	4	5	6	7	8	9	10				
25	0	4													
26	1		-	10											
27	2			(5)	(10)	11									
28	3				(12)	(7)	4	(8)							
29	4					(5)	(3)	(5)	7	6					
30	5														
31	6														
32	7														
33	8														

In the above illustration, we haven’t filled up the array but only showed a few spilled segments, so we don’t bother you with a full screen of **#SPILL!** errors.

A SUBTOTAL Structure

Now here is where the function **SUBTOTAL** comes into play. It has the following syntax:

=SUBTOTAL(function_number, reference1, [reference2], ...)

The first argument **function_number** specifies the function to perform. For example, one [1] is **AVERAGE** and nine [9] is **SUM**.

If we use outputs from **OFFSET** as the **reference** inside a **SUBTOTAL**, we are allowed to use arrays as **width** and **rows** for the **OFFSET**. Moreover, **SUBTOTAL** performs quite similarly to **BYROW** or **MAP**, that it executes the specified function on each **reference**, and preserves the array structure instead of aggregating everything like **SUM** or **PRODUCT**.

We can design a two-dimensional structure using **SEQUENCE** to cover the whole array. We first use a horizontal sequence of length ten [10] in **width** to produce "running segments" of a row:

=SEQUENCE(, 10)

1	2	3	4	5	6	7	8	9	10
---	---	---	---	---	---	---	---	---	----

Then we use a vertical sequence of length nine [9] but starting from zero [0] in **rows**, to cover all nine [9] rows of the array:

=SEQUENCE(9, , 0)

0
1
2
3
4
5
6
7
8

For running totals, we can combine the above inside **OFFSET** and **SUBTOTAL**, and specify a **function_number** nine [9] for **SUBTOTAL**:

=SUBTOTAL(9, OFFSET(\$F\$13, SEQUENCE(\$F\$10,, 0), 0, 1, SEQUENCE(, \$F\$11)))

and the output will be a dynamic array of row running totals for the input array:

F28 =SUBTOTAL(9, OFFSET(\$F\$13, SEQUENCE(\$F\$10,, 0), 0, 1, SEQUENCE(, \$F\$11)))

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Question Data														
2	SP Cumulative Sum & Product.xlsm														
3	Navigator														
4	Error Checks:														
5															
6	1. Question Data														
7															
8	Inputs														
9															
10	Number of rows 9														
11	Number of columns 10														
12															
13	Data														
14	4	12	(9)	(5)	(4)	9	1	(12)	(7)	-					
15	-	10	(11)	(12)	7	(2)	-	9	(10)	(4)					
16	(5)	(10)	11	(11)	4	2	(6)	11	(1)	-					
17	(12)	(7)	4	(8)	6	(1)	(12)	8	7	1					
18	(5)	(3)	(5)	7	6	8	-	4	(12)	1					
19	10	(9)	(10)	6	(1)	8	-	-	(6)	(8)					
20	10	4	(2)	8	9	2	4	(7)	(2)	(9)					
21	(8)	(9)	4	(7)	6	(6)	(6)	6	(2)	(11)					
22	-	4	(5)	(8)	(5)	(1)	(3)	-	(3)	9					
23															
24	2. Desired Output														
25															
26	Answer														
27															
28	Cumulative Sum														
29	4	16	7	2	(2)	7	8	(4)	(11)	(11)					
30	-	10	(1)	(13)	(6)	(6)	(8)	1	(9)	(13)					
31	(5)	(15)	(4)	(15)	(11)	(9)	(15)	(4)	(5)	(5)					
32	(12)	(19)	(15)	(23)	(17)	(18)	(30)	(22)	(15)	(14)					
33	(5)	(8)	(13)	(6)	-	8	12	-	1	(2)					
34	10	1	(9)	(3)	(4)	4	4	(2)	(10)						
35	10	14	12	20	29	31	35	28	26	17					
36	(8)	(17)	(13)	(20)	(14)	(20)	(26)	(22)	(23)						
37	-	4	(1)	(9)	(14)	(15)	(18)	(18)	(21)	(12)					

Similarly, specifying a **function_number** of six [6] for **SUBTOTAL** produces a dynamic array of row running products for the input array:

=SUBTOTAL(6, OFFSET(\$F\$13, SEQUENCE(\$F\$10,,0), 0, 1, SEQUENCE(, \$F\$11)))

F38

✓

<

Thus, we achieve to produce arrays of row running totals and row running products with only one [1] Excel formula for each. We also encourage you to play around with other functions in **SUBTOTAL**. For example, running maximum and running minimum will be both interesting and useful.

Did you have a better solution? Let us know, we'd be glad to hear if there are better ways to work around this problem!

Upcoming SumProduct Training Courses

Location	Course	Course Date	Local Time	UTC	Duration
Melbourne Australia	Power Pivot, Power Query and Power BI	16 June 2025 - 17 June 2025	09:00 - 17:00 AEDT	15 June 2025 23:00 UTC - 17 June 2025 07:00 UTC	2 Days
Melbourne Australia	Excel Tips and Tricks	18 June 2025	09:00 - 17:00 AEDT	17 June 2025 23:00 UTC - 18 June 2025 07:00 UTC	1 Day
Melbourne Australia	Financial Modelling	19 June 2025 - 20 June 2025	09:00 - 17:00 AEDT	18 June 2025 23:00 UTC - 20 June 2025 07:00 UTC	2 Days
Virtual (Australia)	ChatGPT Part 1	23 June 2025	13:30 - 17:00 AEDT	23 June 2025 03:30 UTC - 23 June 2025 07:00 UTC	1 Day
Virtual (Australia)	ChatGPT Part 2	30 June 2025	13:30 - 17:00 AEDT	30 June 2025 03:30 UTC - 30 June 2025 07:00 UTC	1 Day
Sydney Australia	Power Pivot, Power Query and Power BI	14 July 2025 - 15 July 2025	09:00 - 17:00 AEDT	13 July 2025 23:00 UTC - 15 July 2025 07:00 UTC	2 Days
London UK	Financial Modelling	14 July 2025 - 15 July 2025	09:00 - 17:00 BST	13 July 2025 23:00 UTC - 15 July 2025 07:00 UTC	2 Days
Sydney Australia	Financial Modelling	17 July 2025 - 18 July 2025	09:00 - 17:00 AEDT	16 July 2025 23:00 UTC - 18 July 2025 07:00 UTC	2 Days
London UK	Excel Tips and Tricks	28 July 2025	09:00 - 17:00 BST	27 July 2025 23:00 UTC - 28 July 2025 07:00 UTC	1 Day

Key Strokes

Each newsletter, we'd like to introduce you to useful keystrokes you may or may not be aware of. This time, we thought we would put a **SHIFT** in with some of our keyboard commands:

Keystroke	What it does
SHIFT + Backspace	Collapse selection to the active cell
SHIFT + Delete	Cut
SHIFT + Down	Extend selection down one cell
SHIFT + End	Toggle End Mode (and extend when you press SHIFT + Arrow key)
SHIFT + ENTER	Enter value and move up
SHIFT + ESC(APE)	Cancel (edit, copy, cut, dialog, etc.)
SHIFT + Execute	Enter value and move up
SHIFT + Help	What is..?
SHIFT + Home	Extend selection to the first cell in the row
SHIFT + Insert	Paste
SHIFT + Left	Extend selection left one cell
SHIFT + Page Down	Extend selection page down
SHIFT + Page Up	Extend selection page up
SHIFT + RETURN	Enter value and move up
SHIFT + Right	Extend selection right one cell
SHIFT + Spacebar	Select row
SHIFT + Subtract (-)	Add to Selection mode
SHIFT + TAB	Tab backwards (previous cell to edit)
SHIFT + Up	Extend selection up one cell

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

Our Services

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

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

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

Link to Others

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

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

Any Questions?

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

Training

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

Check out our more popular courses in our training brochure:



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

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

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