

Create visual calculations in Power BI Desktop

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/power-bi-visual-calculations/1-introduction>

Introduction

Completed

Visual calculations in Power BI are Data Analytics Expressions (DAX) calculations defined on a visual. Unlike measures, they aren't stored in the model, but on the visual you're working with. Visual calculations make it easier to create calculations that were previously hard to create, leading to simpler DAX, easier maintenance, and better performance.

For example, you're asked to add the following calculations to your report:

- A comparison of sales over years.
- A running sum of cost by category.
- A moving average of profit over time.

Visual calculations make these complex calculations easier by providing a user-friendly interface and real-time validation when applying the calculation on the visual.

In this module, you'll create visual calculations using templates and parameters, and be able to decide when to use visual calculations and when to use measures.

By the end, you'll be able to use visual calculations to provide insights in Power BI reports.

2. Understand visual calculations

Understand visual calculations

Completed

Visual calculations are a simple way to add calculations to your Power BI report. They're DAX calculations defined on a visual and are defined visually. Compared to measures, the DAX required is easier to write, understand, and maintain.

Visual calculations are defined on a visual, such as a table, bar chart, or a matrix. They provide easy to use functions for calculations that are commonly used in organizations.

While creating a visual calculation, you don't have to worry about the complexity of the model. You just work with what is on the visual, making it easier to understand and check your work.

A visual calculation can refer to any data in the visual including columns, measures, or other visual calculations. This ability removes the complexity of the semantic model and simplifies the process of adding calculations to your report. You can use visual calculations to complete common business calculations, such as running sums or moving averages.

Benefits of using a visual calculation

DAX in visual calculations works differently from other calculation options. The formula isn't stored in the model, but on the visual itself. This means they can only refer to what's on the visual. If you need to use something from the model, you must add it to the visual first. This approach simplifies things by avoiding the complexity of filter context and the model.

Visual calculations combine the straightforward context of calculated columns with the flexibility of on-demand calculations from measures. They work on aggregated data instead of detailed data, which often improves performance. When you can choose between a new measure or a visual calculation, the visual calculation usually performs better.

Since visual calculations are part of the visual, they can refer to the visual structure, offering more flexibility.

Note

While visual calculations are in preview, the limitations in this section are subject to change. Please see [Considerations and limitations](#) for the up-to-date information for visual calculations.

3. Create visual calculations

<https://learn.microsoft.com/en-us/training/modules/power-bi-visual-calculations/3-create-visual-calculations>

Create visual calculations

Completed

Creating a visual calculation is straightforward - select a visual and choose **New calculation**. The visual calculation window consists of three major sections, as shown from top to bottom in the following image:

- The **visual preview** which shows the visual you're working with.
- A **formula bar** where you can add visual calculations.
- The **visual matrix** which shows the data in the visual, and displays the results of visual calculations as you add them. Any styling or theming you apply to your visual isn't applied to the visual matrix.



To add a visual calculation, type the expression in the formula bar. For example, the following code calculates the profit by subtracting Total Product Cost from Sales Amount.

```
Profit = [Sales Amount] - [Total Product Cost]
```

The following image shows this Profit calculation added to a matrix visual that contains Sales Amount and Total Product Cost by Fiscal Year. Now you can see the individual fiscal years and a total for the three different values: Sales Amount, Total Product Cost, and Profit.

1 Profit = [Sales Amount]-[Total Product Cost]			
Fiscal Year	Sales Amount	Total Product Cost	Profit
FY2018	\$23,860,891.17	\$20,824,957.60	3,035,933.57
FY2019	\$34,070,108.50	\$30,362,108.34	3,708,000.16
FY2020	\$51,878,274.54	\$46,070,842.02	5,807,432.52
Total	\$109,809,274.20	\$97,257,907.95	12,551,366.25

By default, most visual calculations on a visual are evaluated row-by-row, like a calculated column. In the previous example, for each row of the visual matrix the current Sales Amount and Total Product Cost are subtracted, and the result is returned in the Profit column.

Although possible, there's no need to add an aggregation function like SUM as you would in a measure. In fact, it's better not to add such aggregates when they're not necessary, so you can more easily distinguish between measures and visual calculation expressions.

Hide fields from a visual

As you add visual calculations, they're shown in the list of fields on the visual and on the visual itself:

Y-axis

Fiscal Year

X
|
>

+Add data

X-axis

Sales Amount

👁
X
|
>

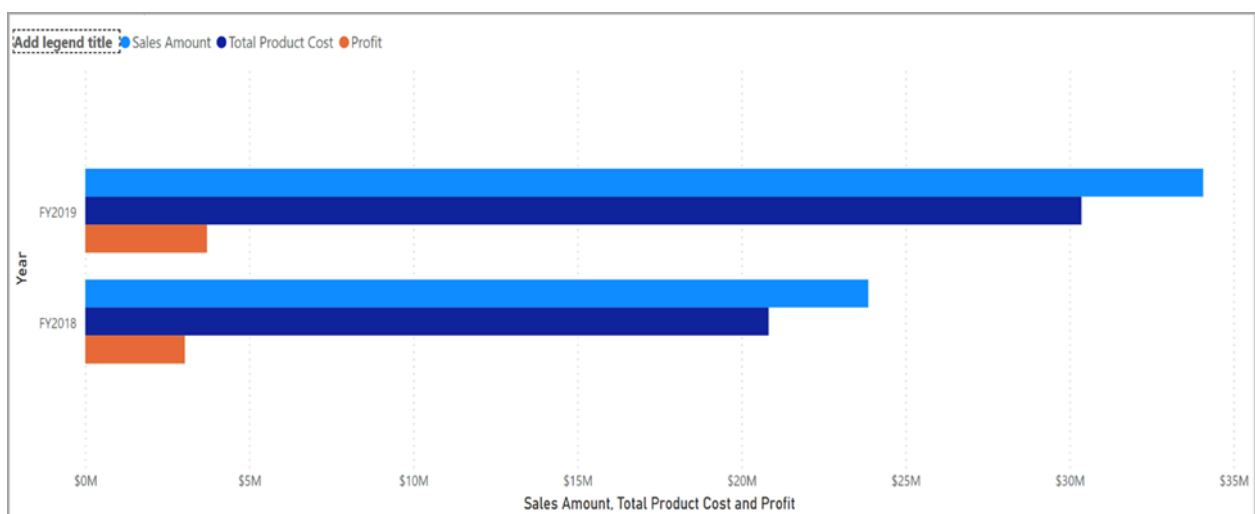
Total Product Cost

👁
X
|
>

📊 Profit

👁
X
|
>

+Add data



In visual calculations edit mode, you can hide fields from the visual just like you can hide columns and tables in the modeling view. For example, if you wanted to only show the *Profit* visual calculation, you can hide *Sales Amount* and *Total Profit* cost from view.



Hiding fields doesn't remove them from the visual or from the visual matrix, so your visual calculations can still refer to them and continue to work. A hidden field is still shown on the visual matrix but is not shown on the resulting visual. It's a recommended practice to only include hidden fields if they're necessary for your visual calculations to work.

Work with templates

Visual calculations include templates to make it easier to write common calculations. You can find templates by selecting the template button and choosing a template, such as:

- **Running sum** calculates the sum of values, adding the current value to the preceding values.
- **Moving average** calculates an average of a set of values in a given window by dividing the sum of the values by the size of the window.
- **Percent of parent** calculates the percentage of a value relative to its parent.
- **Average of children** calculates the average value of the set of child values.
- **Versus previous** compares a value to a preceding value.
- **Versus next** compares a value to a subsequent value.

Each template has a corresponding function which is added to the formula bar when you choose a template. You can also add your own expressions without relying on templates.

Available DAX functions

You can use many of the existing DAX functions in visual calculations. Since visual calculations work within the confines of the visual matrix, functions that rely on model relationships such as USERELATIONSHIP, RELATED, or RELATEDTABLE aren't available. Visual calculations also introduce a set of functions specific to visual calculations. Many of these functions are easier to use shortcuts to DAX window functions.

Note

For a full list of available functions, see the [documentation](#).

4. Use parameters in visual calculations

<https://learn.microsoft.com/en-us/training/modules/power-bi-visual-calculations/4-visual-calculation-parameters>

Use parameters in visual calculations

Completed

Visual calculations have optional parameters to help you create complex calculations with minimal code.

Use the Axis parameter

Many functions have an optional **Axis** parameter, which can only be used in visual calculations. Axis influences how the visual calculation traverses the visual matrix. The Axis parameter is set to the first axis in the visual by default. For many visuals the first axis is ROWS, which means that the visual calculation is evaluated row-by-row in the visual matrix, from top to bottom.

The following parameter values control how the data is calculated:

Value	Description	Icon
ROWS	Vertically across rows from top to bottom.	
COLUMNS	Horizontally across columns from left to right.	
ROWS COLUMNS	Vertically across rows from top to bottom, continuing column by column from left to right.	
COLUMNS ROWS	Horizontally across columns from left to right, continuing row by row from top to bottom.	

Use the Reset parameter

Many functions have an optional **Reset** parameter that is available in visual calculations only. Reset influences if and when the function resets its value to 0 or switches to a different scope while traversing the visual matrix.

The Reset parameter is set to None by default, which means the visual calculation is never restarted. The following list describes the only valid values for the Reset parameter:

- **NONE** is the default value and doesn't reset the calculation.
- **HIGHESTPARENT** resets the calculation when the value of the highest parent on the axis changes.
- **LOWESTPARENT** resets the calculations when the value of the lowest parent on the axis changes.
- **A numerical value** which tells Power BI which level in the visual's hierarchy to partition by—using positive integers to specify an absolute reset level from the top (1 = first field, 2 = second, etc.) and negative integers to specify a relative reset level above the current row—so the visual calculation knows where to restart its aggregation.

To better understand this concept, let's consider an axis that has three fields on multiple levels: Year, Quarter, and Month. As seen in the following example, **HIGHESTPARENT** is **Year** and **LOWESTPARENT** is **Quarter**, which affect how the running sum is calculated.

- `RUNNINGSUM([Sales Amount], HIGHESTPARENT)` starts from 0 **for every year**.
- `RUNNINGSUM([Sales Amount], LOWESTPARENT)` starts from 0 **for every Quarter**.

Lastly, a visual calculation that is defined as `RUNNINGSUM([Sales Amount])` doesn't reset, and continues adding the Sales Amount value for each month to the previous values without restarting.

Note

Reset expects there to be multiple levels on the axis. If there's only one level on the axis, you can use PARTITIONBY.

5. Exercise - Create visual calculations in Power BI Desktop

<https://learn.microsoft.com/en-us/training/modules/power-bi-visual-calculations/5-exercise>

Exercise - Create visual calculations in Power BI Desktop

Completed

In this exercise, you need to create visual calculations to develop a report for sales performance. You learn how to:

- Create a visual calculation
- Use the Axis parameter
- Use the Reset parameter

This lab takes approximately **30** minutes to complete.

Note

A virtual machine containing the client tools you need is provided, along with the exercise instructions. Use the "*Launch lab*" button to launch the virtual machine.

A limited number of concurrent sessions are available. If the hosted environment is unavailable, please try again later.

Alternatively, you can [open the instructions in a separate window](#).

Access your environment

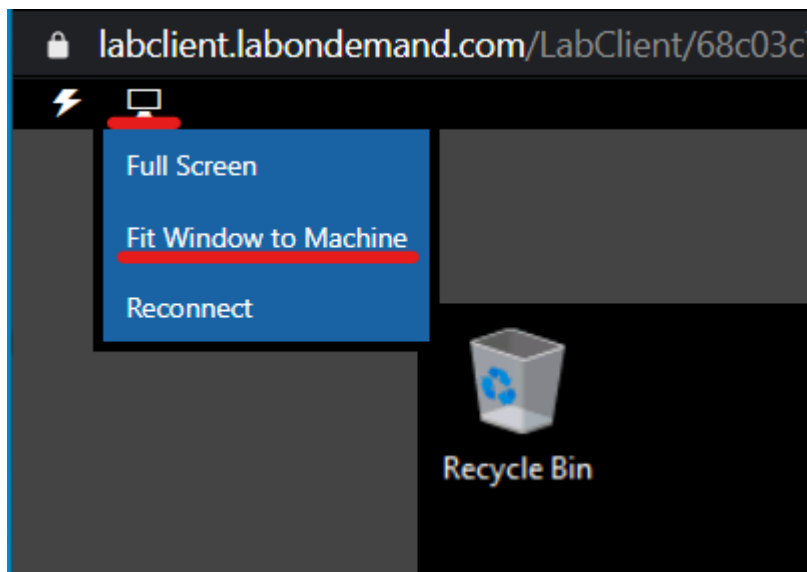
Before you start this lab (unless you are continuing from a previous lab), select **Launch lab** above.

You are automatically logged in to your lab environment as data-ai\student.

You can now begin your work on this lab.

Tip

To dock the lab environment so that it fills the window, select the PC icon at the top and then select **Fit Window to Machine**.



6. Module assessment

<https://learn.microsoft.com/en-us/training/modules/power-bi-visual-calculations/6-knowledge-check>

Module assessment

Completed

7. Summary

<https://learn.microsoft.com/en-us/training/modules/power-bi-visual-calculations/7-summary>

Summary

Completed

In this module, we explored how to use visual calculations in Power BI. Visual calculations let you add powerful calculations to your reports without needing to understand complex DAX language.

Some common problems solved with visual calculations include running sums, moving averages, and comparison calculations. Without visual calculations, users would need to rely on complex DAX functions, which can be time-consuming and difficult to manage.

Visual calculations improve report performance and flexibility, making it easier to create and manage calculations directly on the visual elements. This approach significantly reduces the time needed to add calculations to reports and enhances overall efficiency.

For a more in-depth comparison of ways of adding calculations in Power BI, see [Using calculations options in Power BI Desktop](#).