

How to Calculate Percent Change in Power BI: A Step-by-Step Guide

Authored by
stats writer

January 28, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Calculate Percent Change in Power BI: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128287>

Calculating the percent change is a fundamental analytical technique utilized to measure the rate of growth or decline between two distinct data points. When working within Power BI, understanding this calculation allows users to effectively track performance metrics over time, visualize emerging trends, and identify significant shifts in key performance indicators (KPIs). This guide provides a comprehensive breakdown of the mathematical principle involved and demonstrates the precise implementation using DAX, ensuring accuracy and efficiency in your data modeling efforts.

Implementing Percent Change Calculations in Power BI

The Core Mathematical Formula for Percent Change

Mathematically, calculating the percent change requires comparing a base value (Value1) to a subsequent value (Value2). The resulting difference is normalized against the base value to derive a proportion, which is then typically multiplied by 100 to express the shift as a percentage. This calculation determines the magnitude and direction (growth or decline) of the movement between the two observation periods.

The standard formula used globally for this calculation is:

$$\text{Percent change} = (\text{Value2} - \text{Value1}) / \text{Value1} * 100$$

Consider a practical scenario where a company records 85 units in sales during the initial month (Value1) and subsequently records 94 units in sales during the following month (Value2). We can apply the formula to precisely quantify the sales growth over this interval. This concrete example illustrates how the simple algebraic structure translates into meaningful business insights.

The application of the formula yields the following sequential steps:

$$\text{Percent change} = (\text{Value2} - \text{Value1}) / \text{Value1} * 100$$

$$\text{Percent change} = (94 - 85) / 85 * 100$$

$$\text{Percent change} = 10.59\%$$

The output indicates that the company experienced a percent change (growth) of **10.59%** in sales volume when moving from the first period to the second. This foundational understanding is essential before translating the logic into the Power BI environment using DAX expressions.

Structuring the Calculation using DAX

In Power BI, performing calculations that reference values from previous rows requires advanced functions, as standard measures operate on filtered contexts rather than row contexts. To achieve

a period-over-period comparison, we leverage DAX to create a calculated column that specifically identifies the 'previous' row's value relative to the current row, typically relying on a defined chronological or sequential **Index** column.

The following DAX formula is designed to calculate the percentage change between consecutive rows in a specified table (here, named **my_data**) based on the **Sales** column:

Percent Change =

VAR_max =

MAXX (FILTER ('my_data', < EARLIER ()),)

VAR_value =

MAXX (FILTER ('my_data', = _max),)

RETURN DIVIDE (- _value, _value)

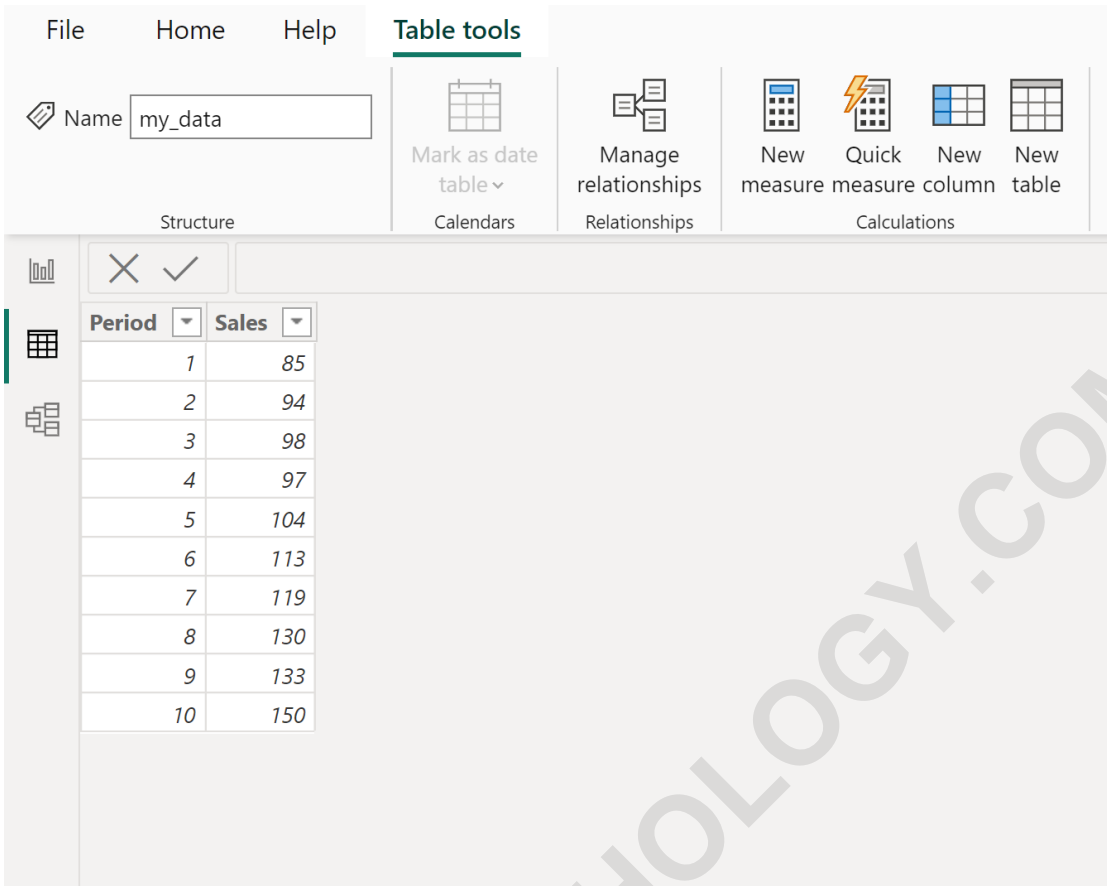
This expression utilizes two crucial variables. The first variable, _max, employs the MAXX function in conjunction with FILTER and EARLIER to find the highest index number that is still less than the current row's index, effectively identifying the immediate preceding period. The second variable, _value, retrieves the **Sales** figure corresponding to that preceding index. Finally, the RETURN statement applies the standard percent change logic using the robust DIVIDE function to handle potential division by zero errors, comparing the current sales value to the previous sales value.

It is paramount to note that this formula relies entirely on the existence of a sequential **Index** column within your data model. This index must typically range from 1 to N, providing a reliable numerical anchor for the row-context comparison required by the DAX engine.

Practical Example: Calculating Sales Performance

To demonstrate the practical application of the aforementioned DAX formula, we will work with a sample dataset loaded into Power BI Desktop. This table, conventionally named **my_data**, contains records of sales figures observed across 10 discrete and consecutive sales periods, simulating a common business scenario where time-series analysis is crucial.

The initial state of our data table is displayed below, illustrating the relationship between the period and the corresponding sales volume:



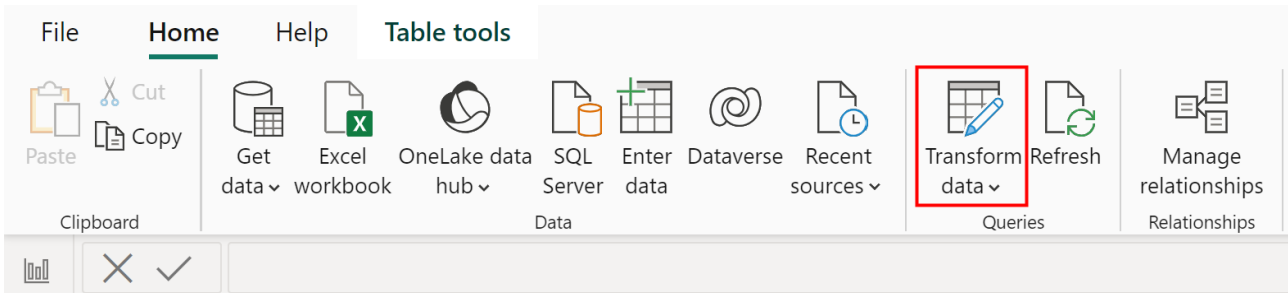
The screenshot displays the 'Table tools' ribbon in Power BI. The ribbon includes a 'Name' field containing 'my_data', a 'Structure' section with 'Mark as date table' and 'Calendars' options, a 'Relationships' section with 'Manage relationships', and a 'Calculations' section with 'New measure', 'Quick measure', 'New column', and 'New table' options. Below the ribbon, a table is visible with two columns: 'Period' and 'Sales'. The table contains 10 rows of data.

Period	Sales
1	85
2	94
3	98
4	97
5	104
6	113
7	119
8	130
9	133
10	150

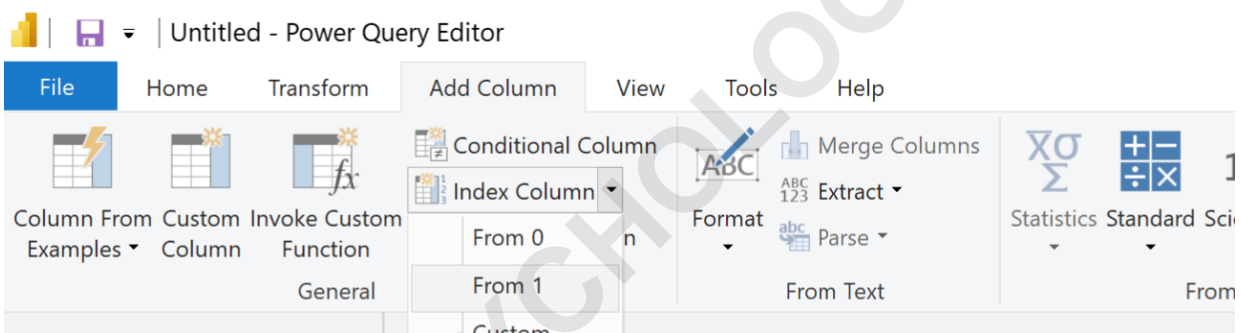
Our objective is to accurately calculate the percentage variation in sales figures from one period to the immediate next. This calculation is essential for identifying momentary dips or spikes in performance that might require further investigation or highlight successful strategies implemented during certain periods.

Preparing Data: Adding the Index Column

As previously established, the DAX formula depends on a reliable sequential identifier. Therefore, the first critical step in preparing our data is to introduce an **Index** column. This process is handled within the Power Query Editor, which is accessed by navigating to the **Home** tab in the Power BI Desktop ribbon and selecting the **Transform data** option.



Once the Power Query Editor interface is open, locate the **Add Column** tab in the top menu. Within this tab, you will find the **Index Column** option. Utilize the dropdown menu associated with this option to select the starting point for your index. While starting from 0 is permissible, choosing **From 1** often results in a more intuitive index that aligns with typical period numbering (e.g., Period 1, Period 2, etc.), which we will select for this demonstration.



Upon selection, the index column is immediately appended to the table preview within the editor, assigning a unique numerical identifier to each row:

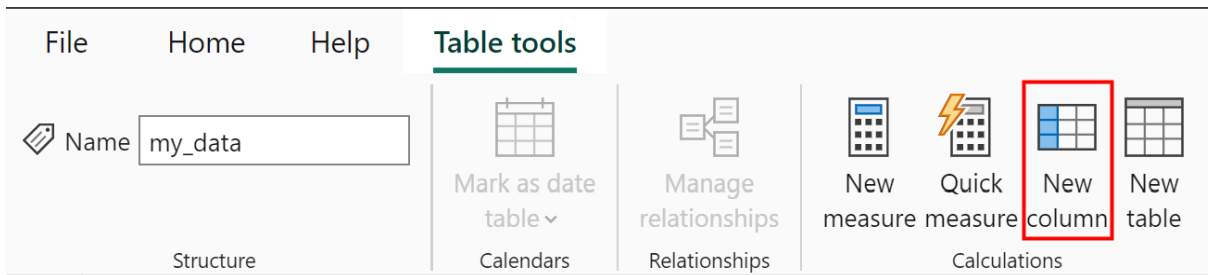
	From Text	From Number	From Date
<div style="border: 1px solid #ccc; padding: 5px;"> ✕ ✓ fx = Table.AddIndexColumn("#Changed Type", "Index", 1, 1, Int64.Type) </div>			
	1 ² 3 Period	1 ² 3 Sales	1 ² 3 Index
1	1	85	1
2	2	94	2
3	3	98	3
4	4	97	4
5	5	104	5
6	6	113	6
7	7	119	7
8	8	130	8
9	9	133	9
10	10	150	10

To finalize this change and incorporate the new **Index** column into the data model, click **Close & Apply** (or exit the editor and confirm the changes when prompted). This step ensures the necessary sequential context is available for the subsequent DAX calculation step in Power BI Desktop.

Applying the DAX Calculation

With the data preparation complete and the **Index** column successfully incorporated into the **my_data** table, we can proceed to implement the DAX logic. This calculation must be executed outside the Power Query Editor, directly within the data model view of Power BI Desktop, by adding a **New column**.

To initiate the creation of the calculated column, ensure you are in the Data view (or Modeling tab), select the table in question, and then locate the **New column** option under the appropriate tab (often **Table tools** or **Modeling**, depending on the current ribbon display):



A formula bar will appear, allowing you to define the logic for the new column. Carefully input the multi-line DAX expression detailed previously. This complex formula performs the row-context evaluation necessary to retrieve the prior period's sales figure, ensuring that the percentage change calculation is executed correctly for every row in the dataset:

Percent Change =

VAR_max =

MAXX (FILTER ('my_data', < EARLIER ()),)

VAR_value =

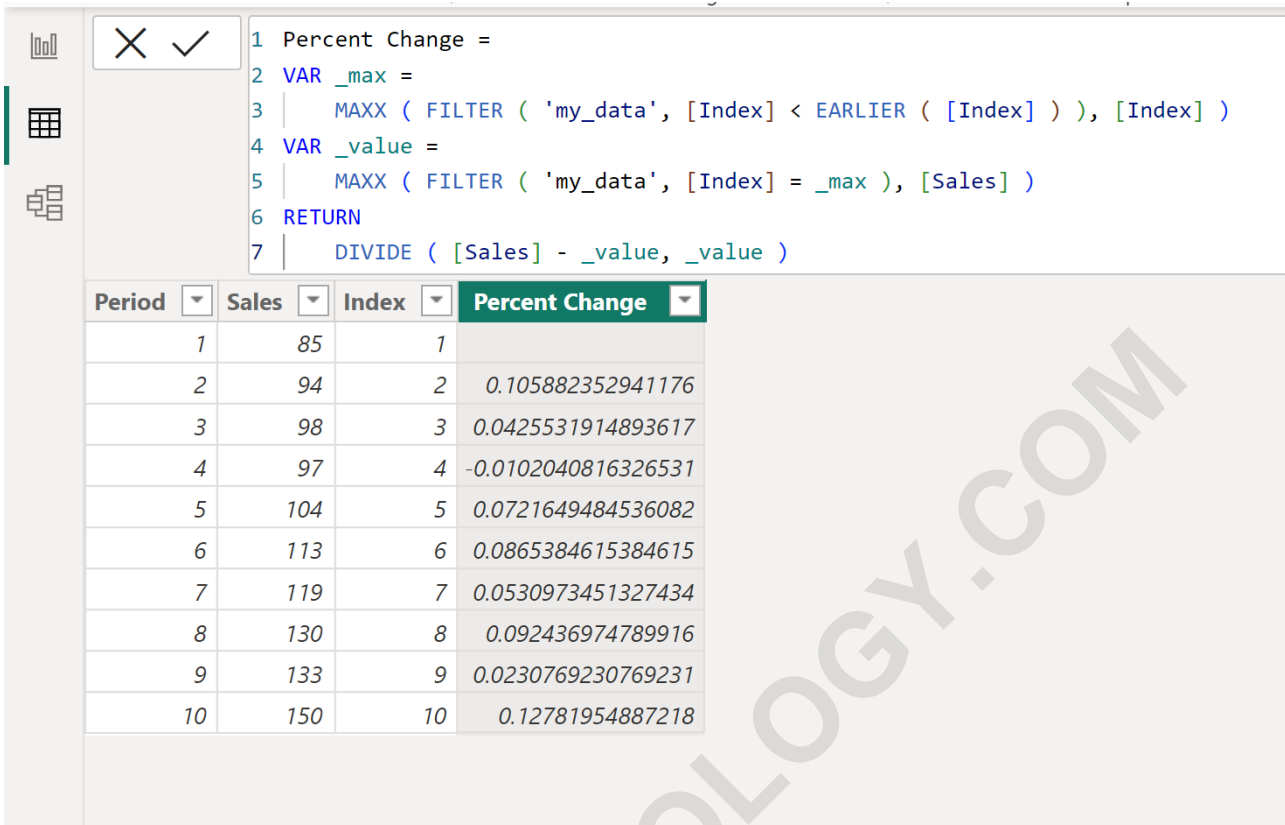
MAXX (FILTER ('my_data', = _max),)

RETURNDIVIDE (- _value, _value)

Executing this DAX code generates a new column named **Percent Change**. It is important to remember to format this new column as a percentage via the Modeling tab properties to ensure the output is displayed in the standard percentage format rather than a raw decimal value.

Analyzing and Interpreting the Results

Upon successfully implementing the DAX calculated column, the **my_data** table is updated to include the **Percent Change** column. This column clearly displays the percentage variation in sales between the current period and the period immediately preceding it. Note that the first row (Period 1) will typically show a blank or an error, as there is no preceding data point against which to calculate a change.



The screenshot shows the DAX editor interface with a measure named 'Percent Change' defined as follows:

```

1 Percent Change =
2 VAR _max =
3     MAXX ( FILTER ( 'my_data', [Index] < EARLIER ( [Index] ) ), [Index] )
4 VAR _value =
5     MAXX ( FILTER ( 'my_data', [Index] = _max ), [Sales] )
6 RETURN
7     DIVIDE ( [Sales] - _value, _value )

```

Below the DAX editor, a table displays the results of the measure. The table has four columns: Period, Sales, Index, and Percent Change. The data shows a general upward trend in sales over 10 periods, with a slight dip in period 4.

Period	Sales	Index	Percent Change
1	85	1	
2	94	2	0.105882352941176
3	98	3	0.0425531914893617
4	97	4	-0.0102040816326531
5	104	5	0.0721649484536082
6	113	6	0.0865384615384615
7	119	7	0.0530973451327434
8	130	8	0.092436974789916
9	133	9	0.0230769230769231
10	150	10	0.12781954887218

By examining the resulting calculated values, analysts can quickly identify performance trends and anomalies. A positive percentage indicates growth, while a negative percentage indicates a decline. For instance, the output provides immediate confirmation of several key performance shifts:

Between Period 1 and Period 2, sales volume increased significantly by **10.59%**.

The growth rate decelerated slightly between Period 2 and Period 3, showing an increase of **4.25%**.

A minor contraction occurred between Period 3 and Period 4, resulting in a sales decrease of **1.02%**.

This calculated column serves as a powerful foundation for visualization in Power BI reports, allowing users to plot these percentages over time using line charts or other visual elements to monitor the volatility and overall momentum of the sales performance.

Conclusion and Related Resources

Mastering the calculation of percent change in Power BI is vital for anyone engaged in serious business intelligence and data analysis. By properly setting up a sequential **Index** and utilizing carefully constructed DAX expressions involving functions like MAXX and DIVIDE, you can

transform static sales figures into dynamic, time-sensitive metrics that drive informed decision-making. These techniques are transferable and form the basis for many advanced time intelligence calculations.

For those looking to expand their knowledge of data manipulation and analysis within the platform, the following resources provide guidance on other common analytical tasks in Power BI:

ARABPSYCHOLOGY.COM