Learning to Calculate Monthly Averages in Power BI with DAX

Authored by

Mohammed looti

November 12, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Learning to Calculate Monthly Averages in Power BI with DAX*. PSYCHOLOGICAL STATISTICS. Retrieved from https://statistics.arabpsychology.com/?p=17501

Introduction to Monthly Aggregation in Power BI

The ability to calculate accurate and meaningful averages grouped by specific time periods is fundamental to effective business intelligence and data analysis. In Power BI, analyzing sales or performance metrics monthly provides crucial insights into seasonal trends, long-term growth, and performance volatility that daily data might obscure. Calculating such time-based aggregations requires careful utilization of DAX (Data Analysis Expressions) and proper data modeling techniques.

This detailed guide outlines the robust, two-step methodology required to calculate the average value of a metric (such as sales) aggregated monthly within the Power BI environment. This process ensures that the resulting calculation is dynamic, responding correctly to filters and slicers applied in the final report view, which is essential for interactive dashboards.

By separating the date context (creating a grouping column) from the calculation logic (creating a measure), we establish a flexible and efficient model. This separation is a best practice in Power BI, allowing the calculation engine to perform sophisticated mathematical operations while respecting the visual context applied by the user.

The Foundational Steps for Time-Based Averages

To successfully compute a monthly average, we cannot simply rely on basic aggregation functions applied directly to the original date column. The Power BI engine needs a discrete numerical or textual identifier to group the data accurately. Therefore, we must undertake preparatory data manipulation before defining the final calculation.

The methodology involves creating a grouping dimension and then using that dimension to scope a dynamic calculation. If the goal were merely a static, row-by-row calculation, a simple calculated column might suffice, but for aggregated values that respond to visualization filtering, a powerful Measure is necessary.

The two critical steps involved in calculating an average value by month in Power BI are detailed below. Following this structured approach ensures data integrity and model efficiency:

Data Preparation: Create a new calculated column that extracts and stores the month number or name from the original date field. This column serves as the grouping context.

Calculation Definition: Use DAX to create a new Measure that calculates the average value, utilizing the newly created month column for context within the visual.

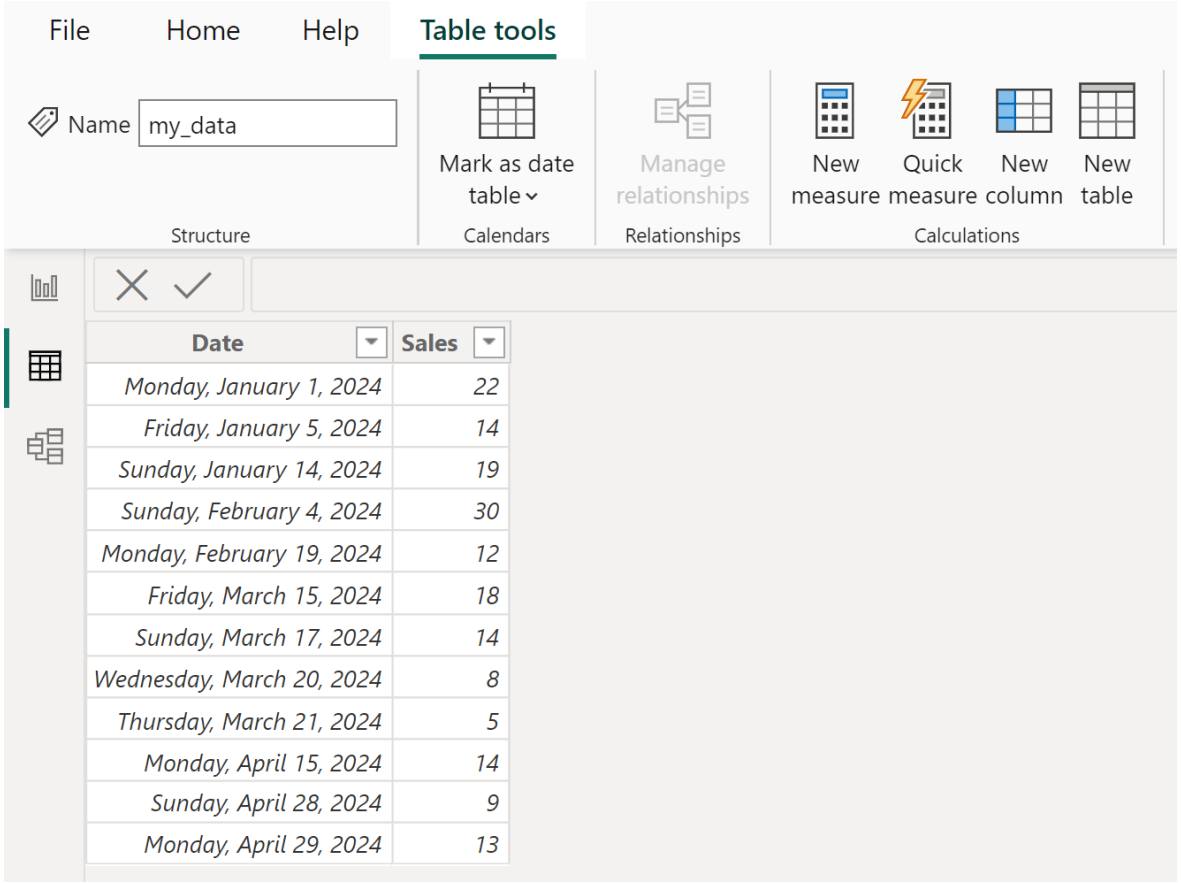
The following practical example utilizes a sample dataset to illustrate these steps clearly, moving from raw data to a finished, insightful visualization.

Example Scenario: Calculating Average Sales by Month

For demonstration purposes, assume we are working with a data table in Power BI named **my_data**. This table contains standard transactional business data, specifically detailing the total sales generated on various dates throughout the year. Our objective is to determine the average sales performance for each calendar month represented in the dataset.

This initial table, named **my_data**, contains at least two key fields: a **Date** column (containing the transaction date) and a **Sales** column (containing the monetary amount of the sale). We aim to calculate the average of the **Sales** column, grouped distinctly by the month of the transaction date.

The structure of the raw data, before any transformation, appears as follows:



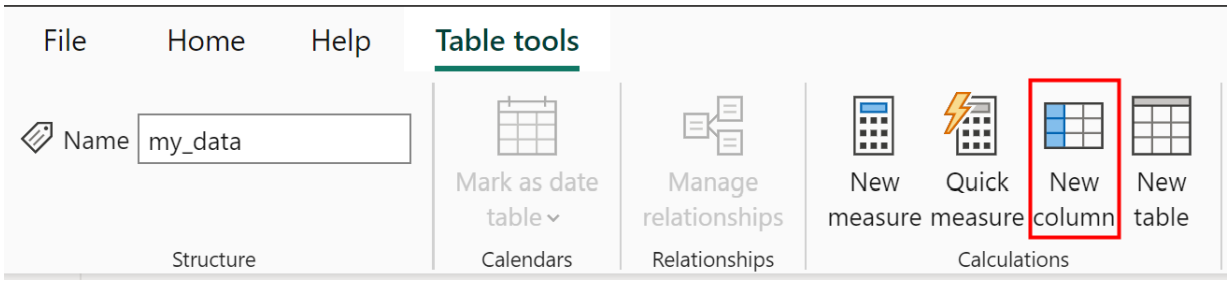
Date	Sales
Monday, January 1, 2024	22
Friday, January 5, 2024	14
Sunday, January 14, 2024	19
Sunday, February 4, 2024	30
Monday, February 19, 2024	12
Friday, March 15, 2024	18
Sunday, March 17, 2024	14
Wednesday, March 20, 2024	8
Thursday, March 21, 2024	5
Monday, April 15, 2024	14
Sunday, April 28, 2024	9
Monday, April 29, 2024	13

Our primary requirement is to calculate the average value residing in the **Sales** column, ensuring this aggregation is performed independently for each month present in the **Date** column. This process starts with data augmentation--the creation of a new grouping column.

Step 1: Extracting the Month Identifier Using a Calculated Column

The first essential step is to generate a numerical representation of the month from the existing **Date** column. While Power BI can handle implicit date hierarchies, creating an explicit month number column provides greater control and simplifies the subsequent measure calculation, especially when dealing with data spanning multiple years (though for simplicity, this example assumes single-year data).

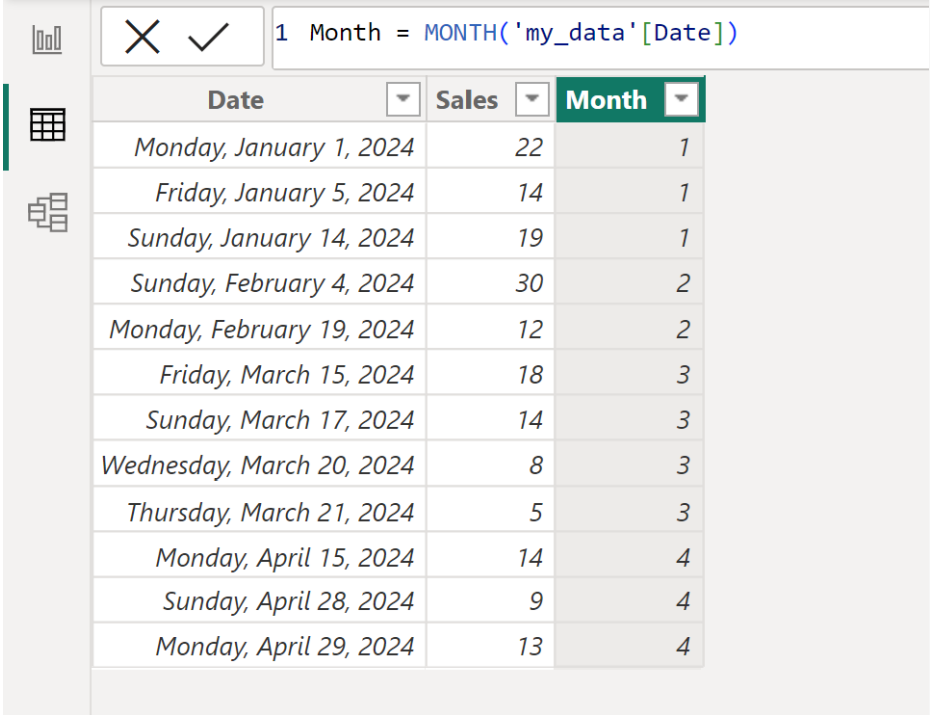
To initiate the column creation process, navigate to the data view within the modeling ribbon. Click the **Table tools** tab located along the top ribbon interface. Within this tab, select the **New column** icon. This action opens the DAX formula bar, prompting you to define the calculation for the new field.



In the DAX formula bar, input the following expression. This formula utilizes the built-in DAX function MONTH, which extracts the month number (1 through 12) from the specified date column:

Month = MONTH('my_data')

Executing this formula successfully adds a new column named **Month** to the **my_data** table. This column assigns the corresponding month number to every transaction row, effectively providing the necessary dimension for grouping our average sales calculation later in the process.

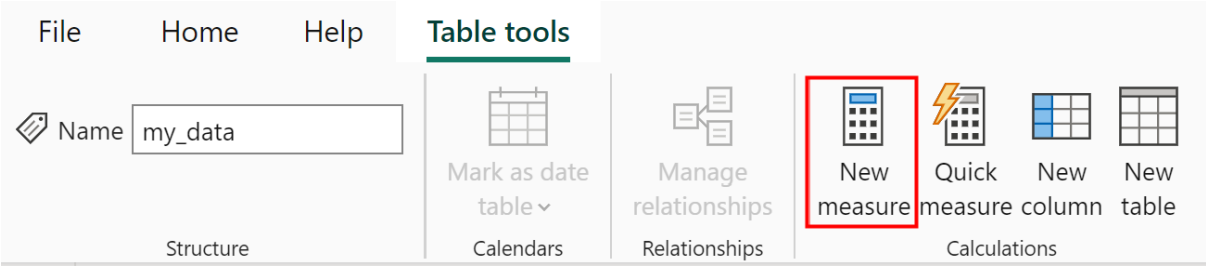


Date	Sales	Month
Monday, January 1, 2024	22	1
Friday, January 5, 2024	14	1
Sunday, January 14, 2024	19	1
Sunday, February 4, 2024	30	2
Monday, February 19, 2024	12	2
Friday, March 15, 2024	18	3
Sunday, March 17, 2024	14	3
Wednesday, March 20, 2024	8	3
Thursday, March 21, 2024	5	3
Monday, April 15, 2024	14	4
Sunday, April 28, 2024	9	4
Monday, April 29, 2024	13	4

Step 2: Defining the Average Sales Measure with DAX

With the necessary grouping column established, the next stage involves creating a dynamic calculation--a Measure. A Measure is preferred over a calculated column for aggregations because it is evaluated at query time, dynamically adjusting its result based on the filtering context (e.g., the specific month selected in a visualization).

To create the Measure, return to the **Table tools** tab and click the **New measure** icon. This action again opens the formula bar, where we will define the average sales logic.



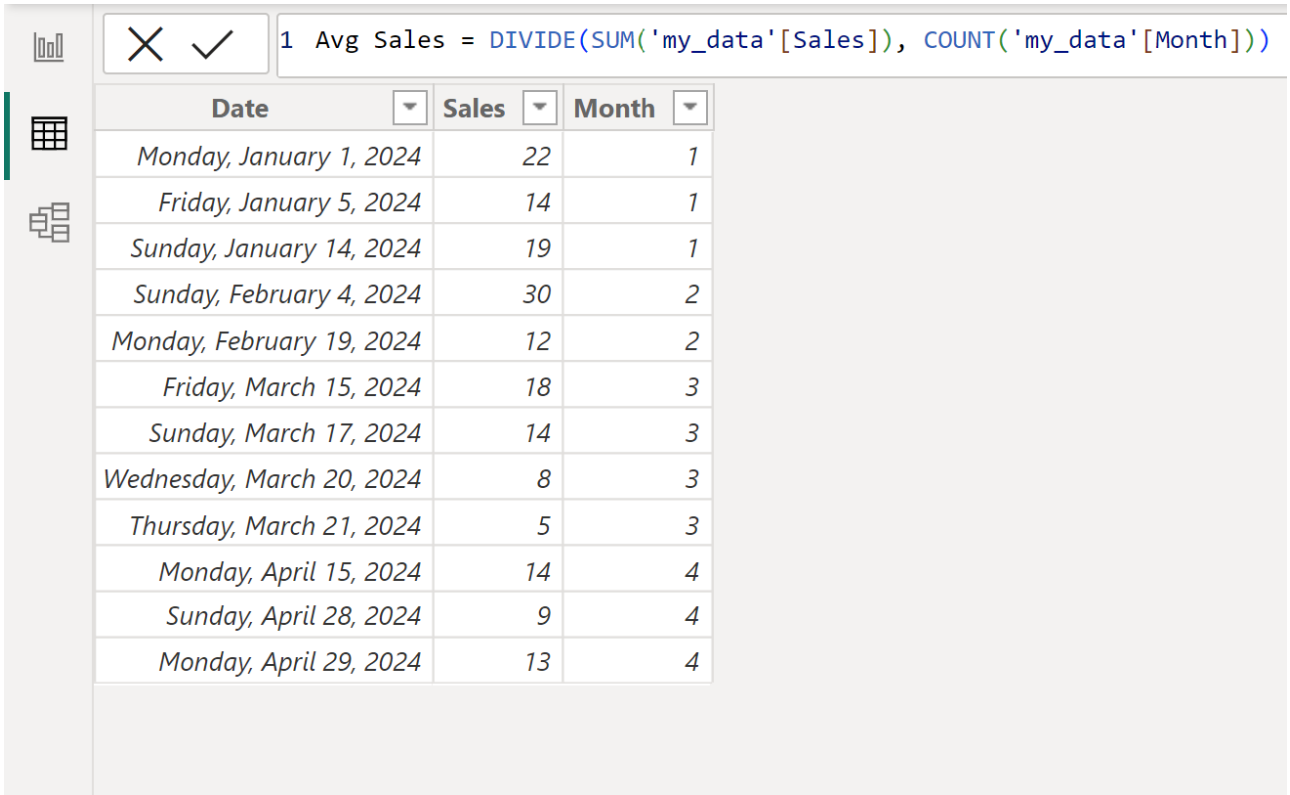
The calculation for the average must be explicitly written as the total sum divided by the count of observations. While DAX has an `AVERAGE` function, explicitly defining the average using the DIVIDE function, coupled with `SUM` and `COUNT`, provides greater control and robustness, particularly preventing errors if the denominator is zero. The `COUNT` function applied to the

Month column effectively counts the number of rows (transactions) associated with that month.

Enter the following formula into the formula bar to define the measure named **Avg Sales**:

Avg Sales = DIVIDE(SUM('my_data'), COUNT('my_data'))

This operation successfully creates the **Avg Sales** Measure. When placed into a visualization, this measure will calculate the sum of sales for the current context (e.g., Month 1) and divide it by the total count of transactions in that same context, yielding the accurate average sales value per month.

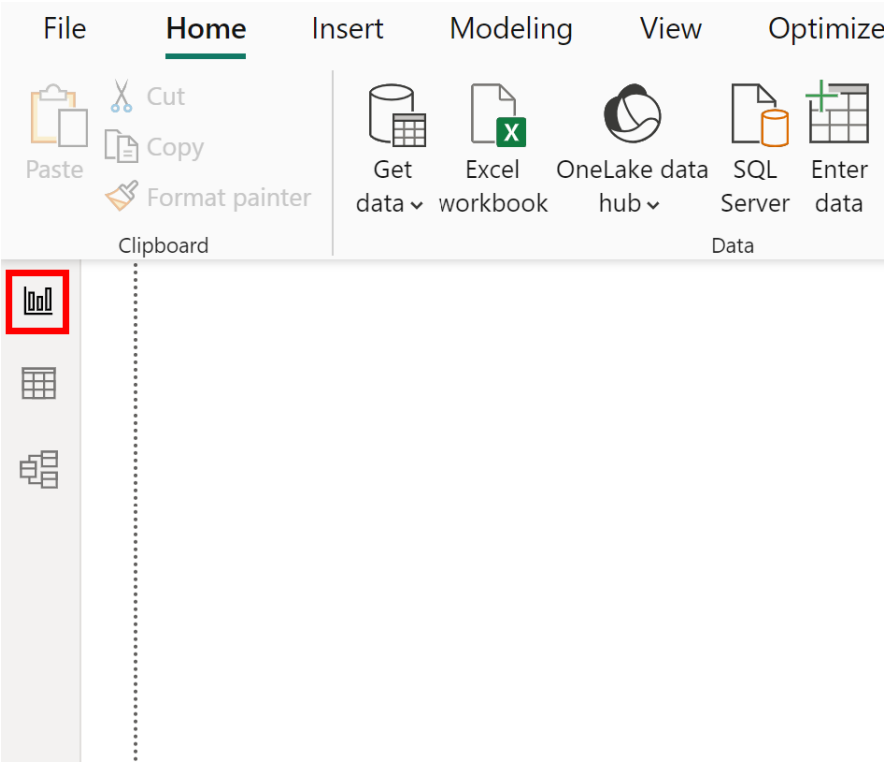


Date	Sales	Month
Monday, January 1, 2024	22	1
Friday, January 5, 2024	14	1
Sunday, January 14, 2024	19	1
Sunday, February 4, 2024	30	2
Monday, February 19, 2024	12	2
Friday, March 15, 2024	18	3
Sunday, March 17, 2024	14	3
Wednesday, March 20, 2024	8	3
Thursday, March 21, 2024	5	3
Monday, April 15, 2024	14	4
Sunday, April 28, 2024	9	4
Monday, April 29, 2024	13	4

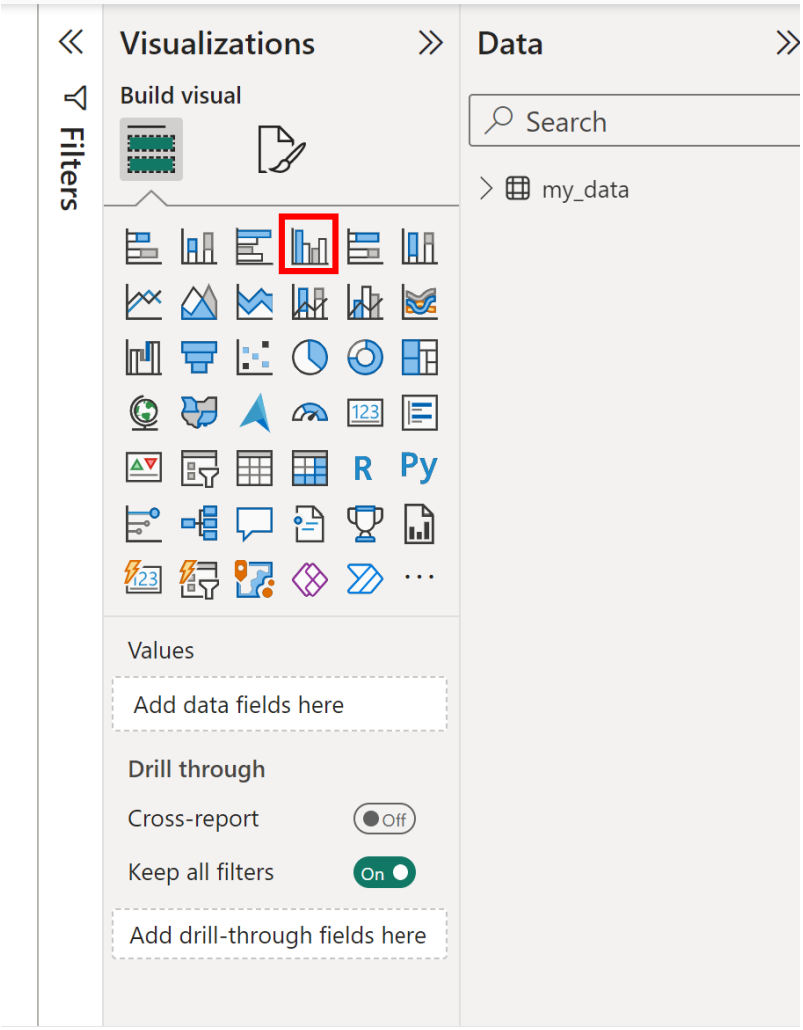
Visualizing the Results in the Report View

Once the data model is prepared with the necessary calculated column and the dynamic measure, the final step is to transition to the reporting phase and visualize the results. Visualization transforms the raw calculated numbers into actionable business insights, making trends and outliers immediately apparent.

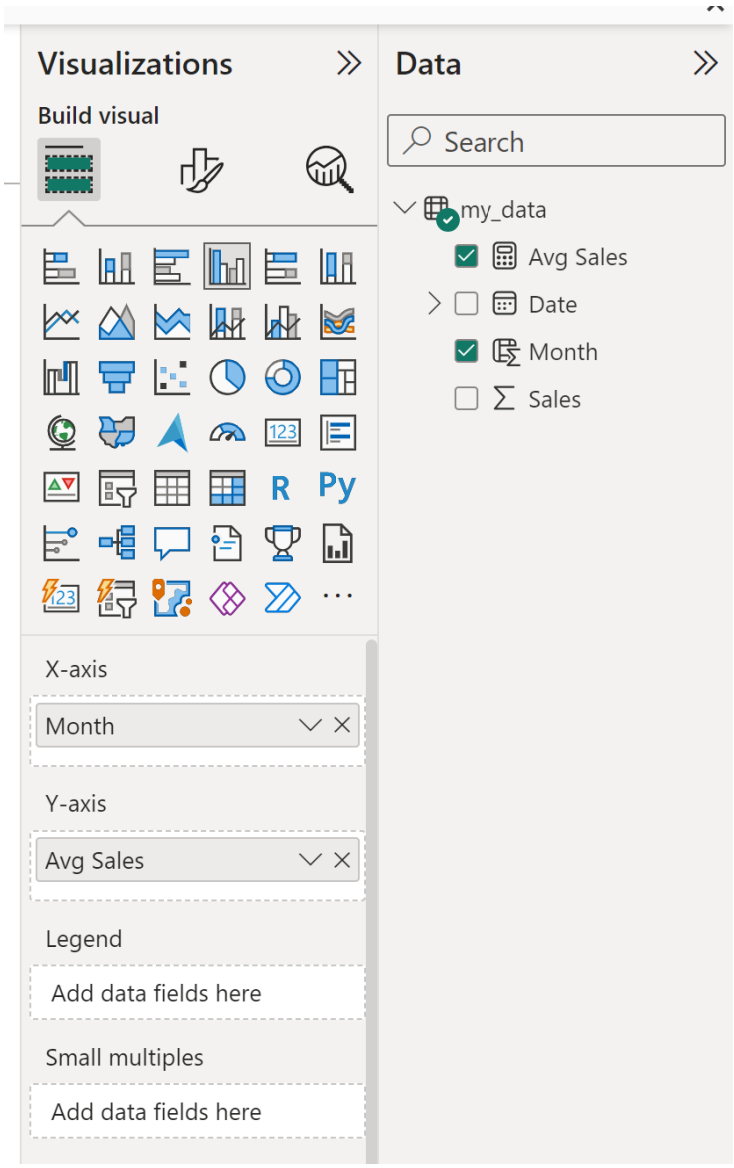
To begin visualization, click the **Report view** icon, typically located on the left side of the Power BI Desktop interface. This shifts the workspace from the data model editor to the interactive report canvas.



For time-series or categorical comparisons like monthly averages, a column chart is highly effective. Navigate to the **Visualizations** pane and click the icon corresponding to the **Clustered column chart**. This action places an empty visual placeholder onto the report canvas.

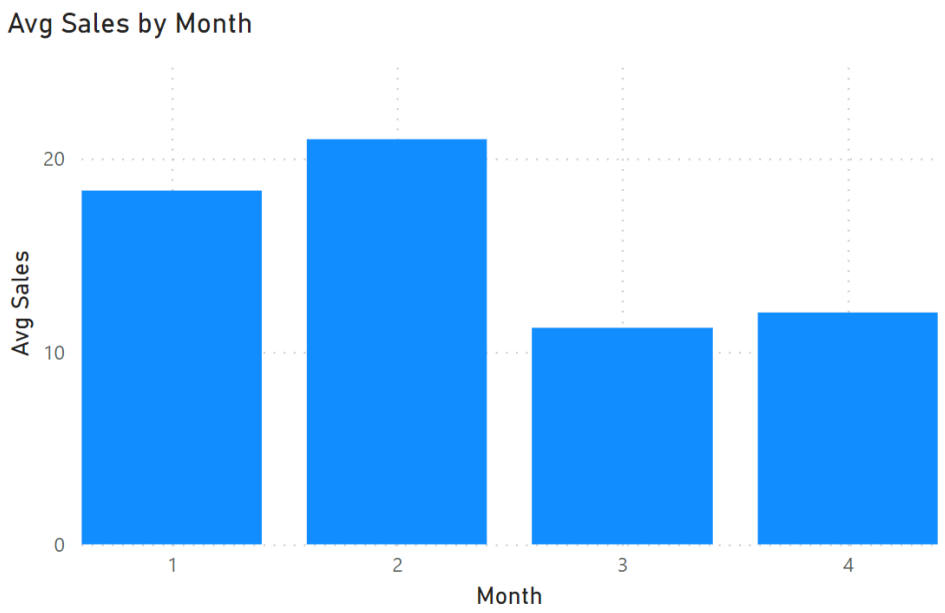


The next critical step is mapping the calculated fields to the correct visual axes. Drag the **Month** column (the grouping dimension created in Step 1) and place it under the **X-axis** label. Subsequently, drag the **Avg Sales** measure (the dynamic calculation created in Step 2) and place it under the **Y-axis** label. This tells Power BI to plot the monthly average against the month number, generating the final report visual.



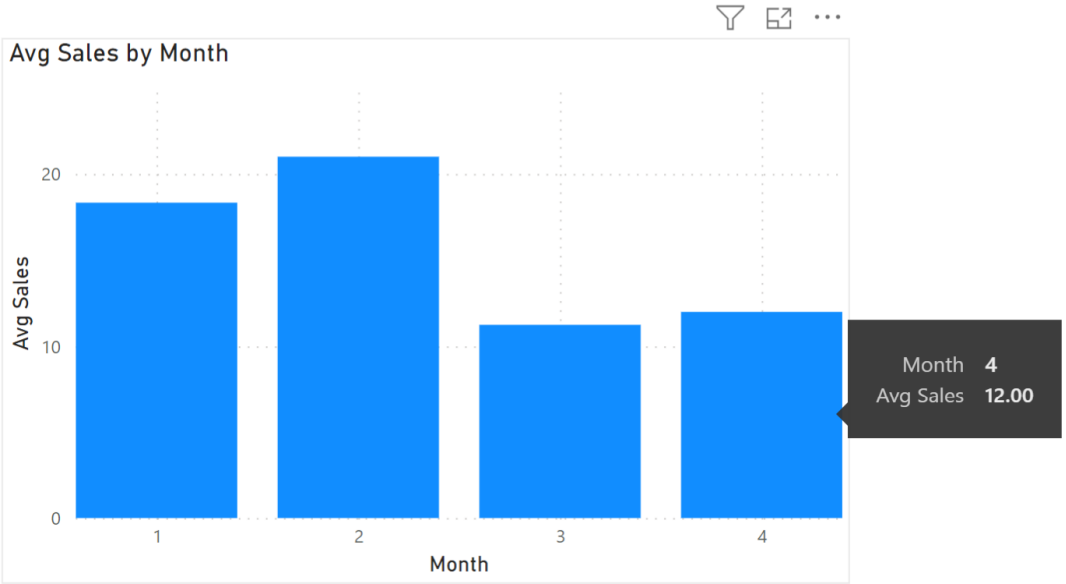
Interpreting the Visualization and Contextualizing the Data

The visualization produced clearly demonstrates the average sales performance segmented by month. The horizontal (X) axis represents the unique month numbers (1 through 12), corresponding to the column we created, while the vertical (Y) axis quantifies the calculated average sales value for that specific month. This visual representation is the end product of our successful data modeling and DAX efforts.



The primary benefit of this dynamic visualization is its interactivity. By hovering the cursor over any individual column in the chart, the tooltip feature provides precise, detailed information regarding that specific data point. This confirms the accuracy of the underlying calculation for each monthly segment.

For instance, if we hover over the bar corresponding to month 4 (April), the tooltip confirms that the calculated average sales value for all transactions recorded in that month is exactly **12**. This high level of detail is crucial for precise financial and operational reporting. This successful output confirms that our two-step process--creating a static month column for grouping and a dynamic Measure for calculation--has yielded the desired monthly data analysis result.



Summary and Further Resources

Mastering time intelligence in Power BI, especially calculating averages over periods like months, is a cornerstone of advanced report creation. By following the best practice of separating the data context (the calculated column) from the aggregation logic (the Measure), users ensure their reports are scalable, efficient, and responsive to complex filtering. This technique ensures that your data analysis is always accurate, regardless of how the end-user interacts with the visualizations.

The following tutorials explain how to perform other common tasks in Power BI, building upon the foundational knowledge of DAX and data modeling presented here:

Further Tutorials Link 1
Further Tutorials Link 2
Further Tutorials Link 3