

# How to Calculate Monthly Averages in Power BI: A Step-by-Step Guide

Authored by  
**stats writer**

January 28, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Calculate Monthly Averages in Power BI: A Step-by-Step Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128221>

**Power BI** is recognized globally as a leading **data analysis** and **visualization tool**. It empowers organizations to transform raw data into rich, interactive insights, facilitating robust decision-making across all departments. Among its most powerful capabilities is the implementation of time intelligence, which allows for the straightforward tracking and comparison of metrics over specified temporal periods. Calculating key performance indicators (KPIs), such as the average value by month, is essential for identifying seasonal variations, evaluating performance consistency, and understanding underlying business drivers.

Analyzing data aggregated by month provides crucial context that daily or yearly totals often obscure. By calculating the monthly average of a specific metric--whether it is sales volume, customer service response time, or inventory turnover--users can gain deep insights into trends and patterns that influence operational efficiency and profitability. This process leverages the built-in time functions within **Power BI**, enabling analysts to structure complex calculations efficiently without needing external preprocessing steps.

The standard approach to deriving monthly averages in **Power BI** involves using **Data Analysis eXpressions (DAX)**. While simpler functions like the **AVERAGE function** can handle basic arithmetic, correctly grouping these results by month often requires creating auxiliary columns or defining advanced measures to ensure accurate aggregation across the defined date range. This tutorial will guide you through a reliable, two-step methodology to achieve accurate monthly average calculations and subsequent visualization.

## Power BI: Calculate Average By Month

### Fundamental Steps for Calculating Monthly Averages

To accurately calculate an average value, such as average sales, grouped specifically by month within your **Power BI** data model, a robust two-phase approach is necessary. This method ensures that the calculation correctly aggregates data based on the temporal context, overcoming the common pitfalls associated with automatic date hierarchies.

The calculation relies on decoupling the date field into its constituent parts--specifically the month number--before defining the aggregation logic. This enables the calculation engine to correctly group all entries falling within the same calendar month, regardless of the year, although year context is often necessary for more detailed time intelligence. This structured approach guarantees that the average is calculated correctly for all days within that defined month group.

The technical implementation involves the following sequential actions, utilizing the powerful capabilities of **DAX**:

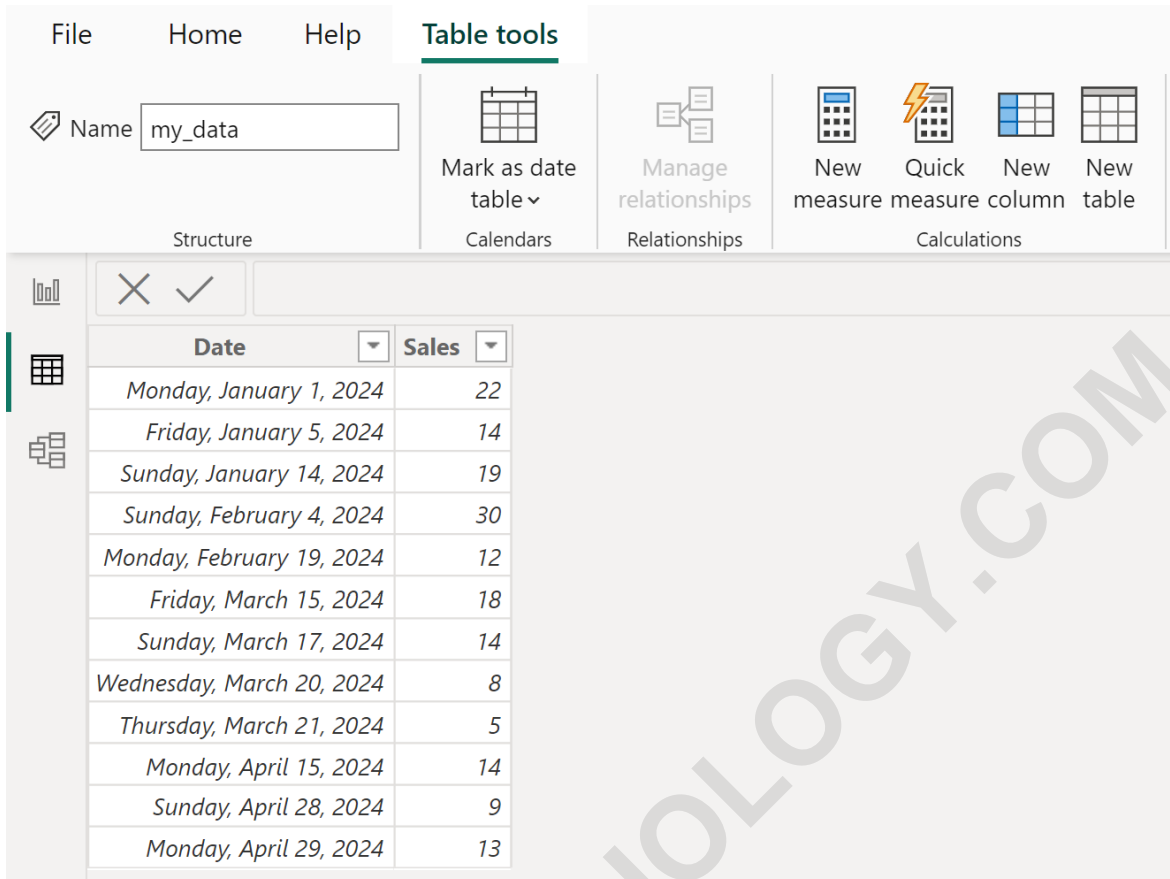
**Step 1: Data Preparation via Calculated Column.** Create a new calculated column in your table. The purpose of this column is to isolate and extract the month number from the full date column, providing a field suitable for explicit grouping.

**Step 2: Defining the Aggregation Logic.** Implement a **DAX measure**. This **measure** utilizes appropriate functions (like `DIVIDE`, `SUM`, and `COUNT`) to calculate the true average of your target metric, aggregated by the new Month column created in Step 1.

## Illustrative Example: Setting Up the Data Environment

We will now walk through a practical demonstration of this calculation using a sample dataset. Assume we have a transactional table within our **Power BI** environment, conveniently named `my_data`. This table contains various records, most importantly, the transaction `date` and the corresponding `sales` amount for that date. Our goal is to determine the average sales performance for each month present in this dataset.

The structure of our initial data table, `my_data`, is displayed below. Note that the granularity of the data is daily, but we aim to abstract this into a summarized monthly level. This initial data structure is fundamental to understanding how the subsequent **DAX** calculations will interact with the source information.



The screenshot shows the Power BI interface with the 'Table tools' ribbon selected. The ribbon is divided into four main sections: 'Structure', 'Calendars', 'Relationships', and 'Calculations'. The 'Calculations' section is currently active, displaying four options: 'New measure', 'Quick measure', 'New column', and 'New table'. Below the ribbon, a table is displayed with two columns: 'Date' and 'Sales'. The table contains 12 rows of data, including dates and corresponding sales values.

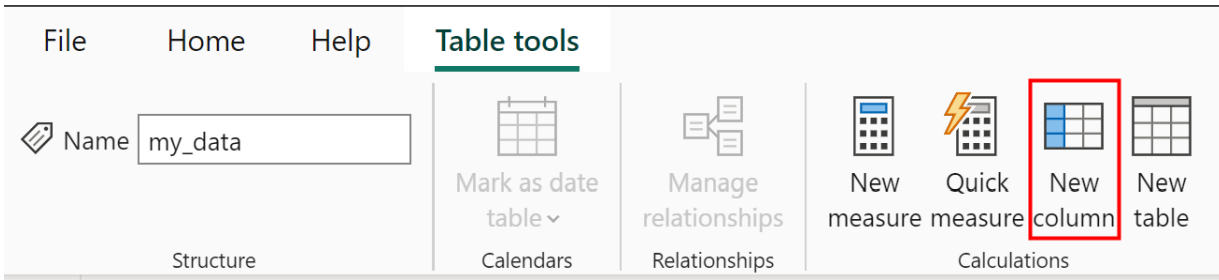
Date	Sales
Monday, January 1, 2024	22
Friday, January 5, 2024	14
Sunday, January 14, 2024	19
Sunday, February 4, 2024	30
Monday, February 19, 2024	12
Friday, March 15, 2024	18
Sunday, March 17, 2024	14
Wednesday, March 20, 2024	8
Thursday, March 21, 2024	5
Monday, April 15, 2024	14
Sunday, April 28,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

Before grouping sales data, we must ensure that we have a discrete field representing the month. While **Power BI** often automatically generates date hierarchies, creating a dedicated column for the month number streamlines the aggregation process and ensures consistency when defining the calculation **measure**. This column acts as the categorical field necessary for accurate grouping.

## Step 1: Creating the Monthly Grouping Column

The first critical step involves modifying the data model to include a column that isolates the month value numerically. This is necessary because direct aggregation functions need a simple categorical field to group by when working outside of complex time intelligence functions. We utilize the powerful **MONTH** function available in the **DAX** language for this purpose.

To execute this, navigate to the **Table tools** tab found along the top menu ribbon, typically within the Data View or Model View. From there, select the **New column** option. This action opens the formula bar, allowing you to define the calculation for the new column dynamically across the entire dataset.



Enter the following **DAX** formula into the formula bar. This formula extracts the numerical representation of the month (1 for January, 12 for December) from the `Date` column within the `my_data` table, assigning the result to the new column named `Month`:

**Month = MONTH('my\_data')**

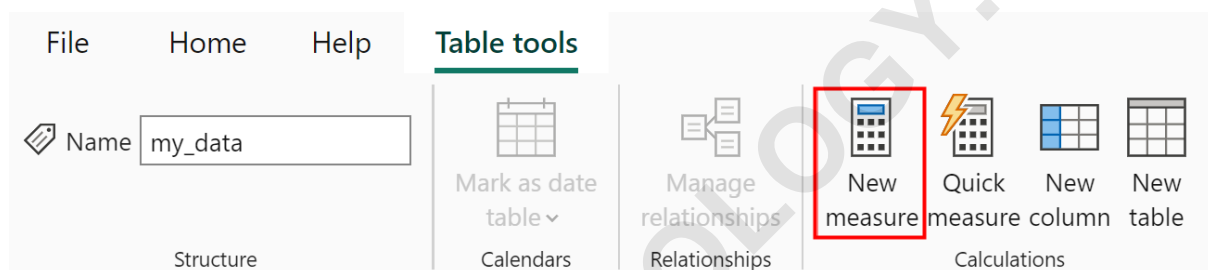
Upon execution, a new column, appropriately named `Month`, will be added to your `my_data` table. This column now holds only the month number for every corresponding entry, serving as the primary grouping mechanism for our subsequent calculations. This calculated column is crucial for ensuring accurate monthly summarization across the dataset, regardless of the year in which the transaction occurred.

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 DAX Measure

With the grouping mechanism in place (the `Month` column), the next step is to define the actual calculation--the average sales. Instead of creating another calculated column, we must create a **measure**. **Measures** are vital because they calculate results dynamically based on the current filtering context applied in the report (e.g., filtering by a specific month or year), ensuring performance efficiency.

Return to the **Table tools** tab and select the **New measure** icon. This opens the formula bar for defining the dynamic aggregation logic. The choice of a **measure** here is essential for allowing the report visuals to aggregate the sales data correctly when grouped by the newly created `Month` column.

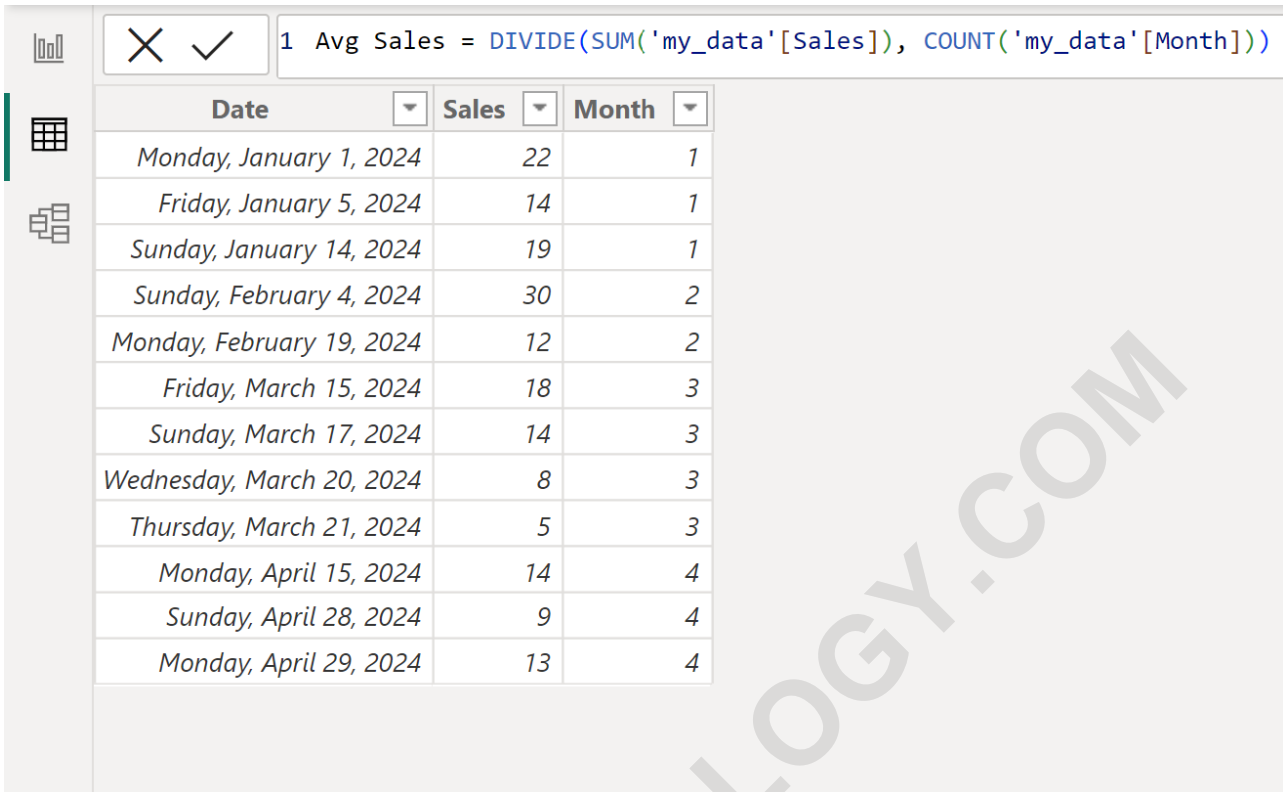


We use a combination of `SUM` and `COUNT`, wrapped within the `DIVIDE` function, to calculate the average. While the basic **AVERAGE function** exists, explicitly using `DIVIDE(SUM(), COUNT())` is often preferred in complex **DAX** scenarios as it provides better control over error handling (like division by zero) and ensures the accurate total sales are being divided by the total number of records for that specific temporal context.

Input the following **DAX** formula. This new **measure**, `Avg sales`, sums all sales values and divides them by the total count of records for the filtered month:

**Avg Sales = DIVIDE(SUM('my\_data'), COUNT('my\_data'))**

This **measure** is now ready to be deployed on your report canvas. When you pull this measure into a visual and group it by the new `Month` column, **Power BI** automatically applies the filter context for each month, calculating the summed sales and dividing it by the corresponding count of entries for that specific month, yielding the accurate average sales performance.



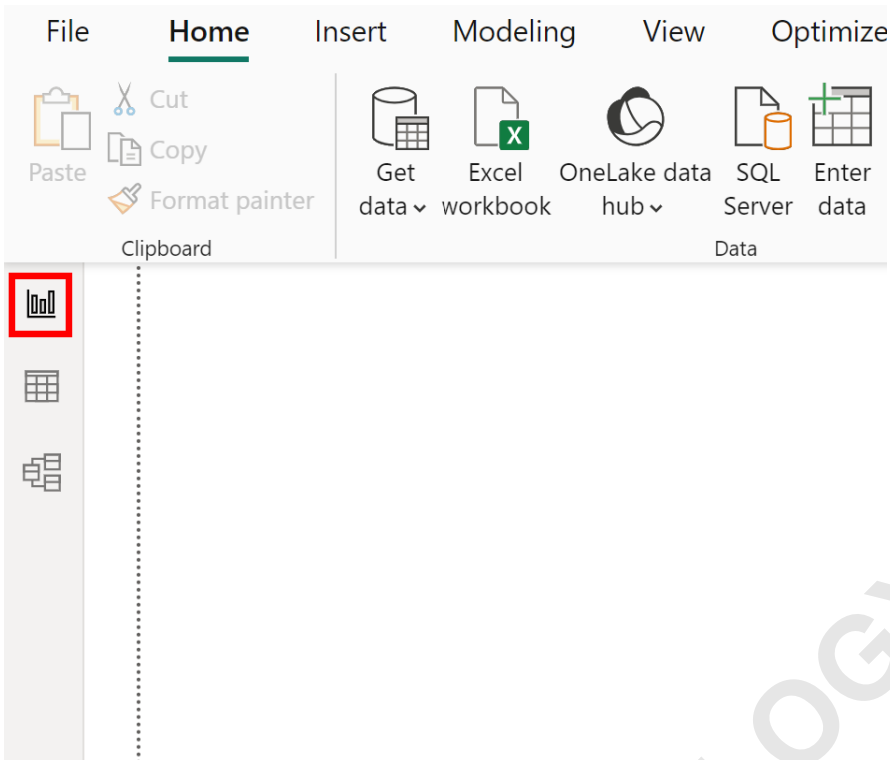
1 Avg Sales = DIVIDE(SUM('my\_data'[Sales]), COUNT('my\_data'[Month]))

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

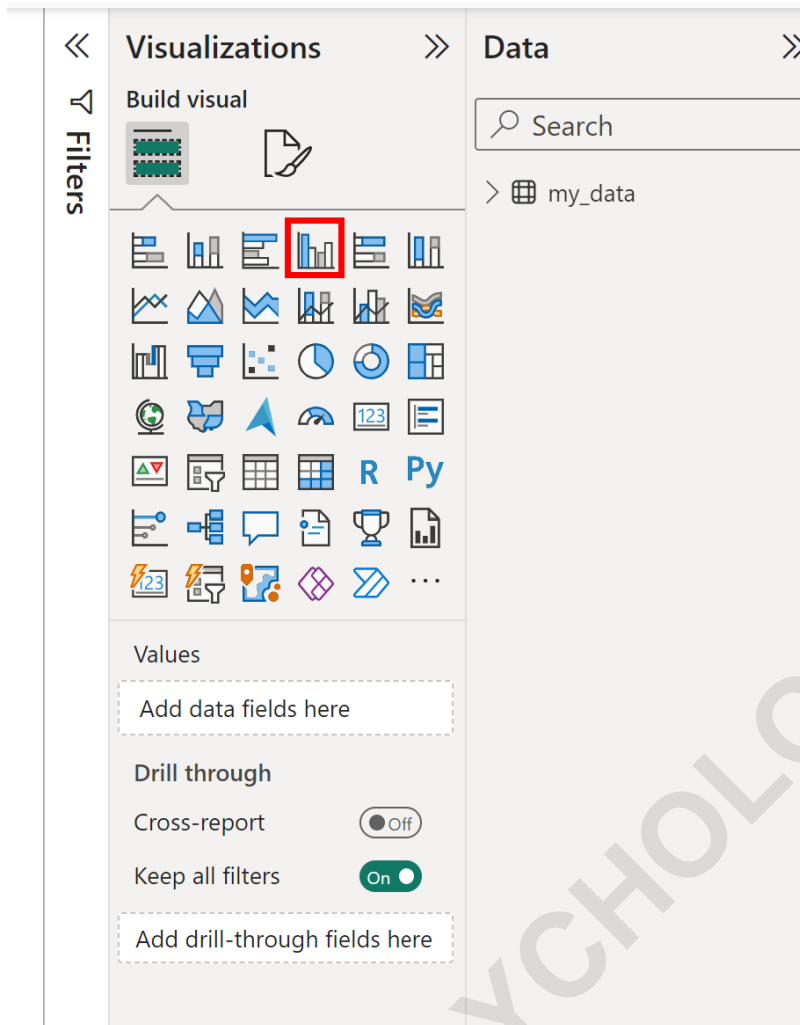
## Visualizing Monthly Performance Trends

Once the `Avg Sales` measure is successfully defined, the next logical step in the **data analysis** workflow is to visualize these monthly trends. **Power BI** excels in visualization, allowing users to transform raw metrics into insightful charts that are easy for stakeholders to understand. A common and highly effective choice for illustrating time-series data like monthly averages is the clustered column chart, which clearly displays magnitude changes between discrete periods.

To begin creating the visual, switch from the Data View or Model View to the **report view**. This is typically accessed via an icon on the left side navigation pane of the **Power BI** interface. The Report view is the dedicated workspace for designing and presenting interactive business intelligence reports.



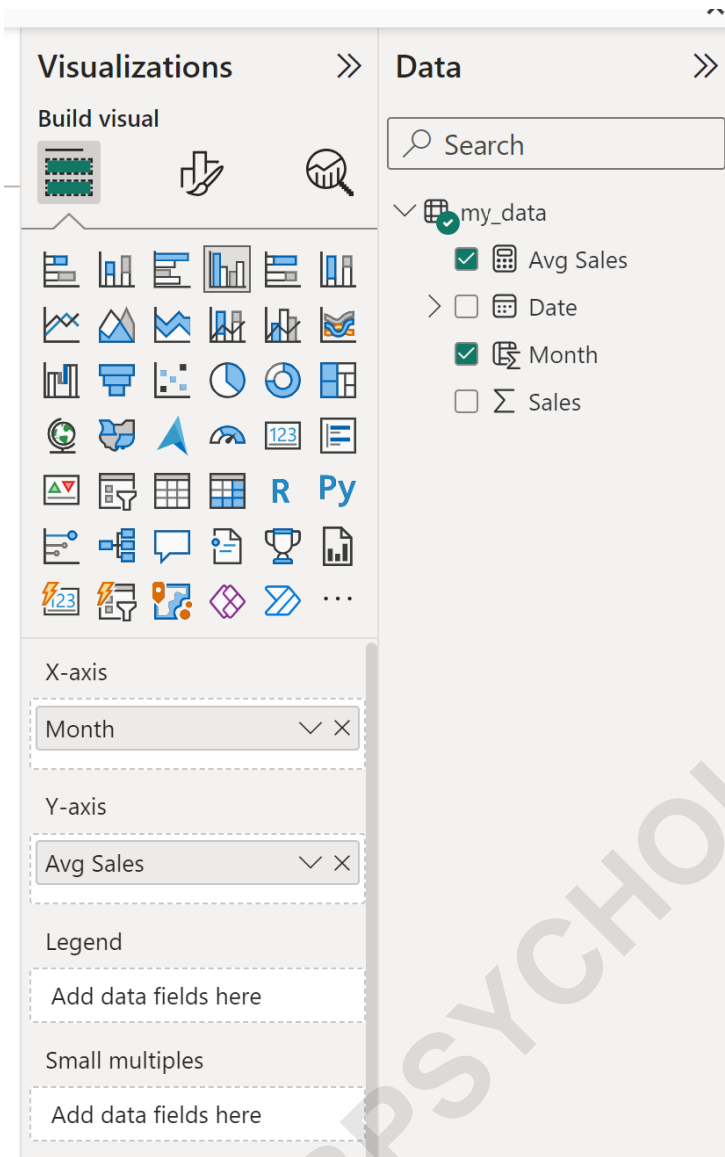
Within the Report view, access the **visualizations** pane. Select the icon representing the **clustered column chart**. This action places an empty visualization container onto your report canvas, ready for configuration. Column charts are highly effective for comparing metric performance across distinct categories, such as monthly periods, due to their ease of interpretation.



## Configuring and Interpreting the Chart Axes

The configuration of the chart involves mapping the calculated fields to the appropriate axes. For time-based analysis, the time dimension (our grouping column) should always occupy the X-axis, while the calculated metric (our measure) should define the magnitude on the Y-axis. This standard arrangement ensures logical flow and interpretability.

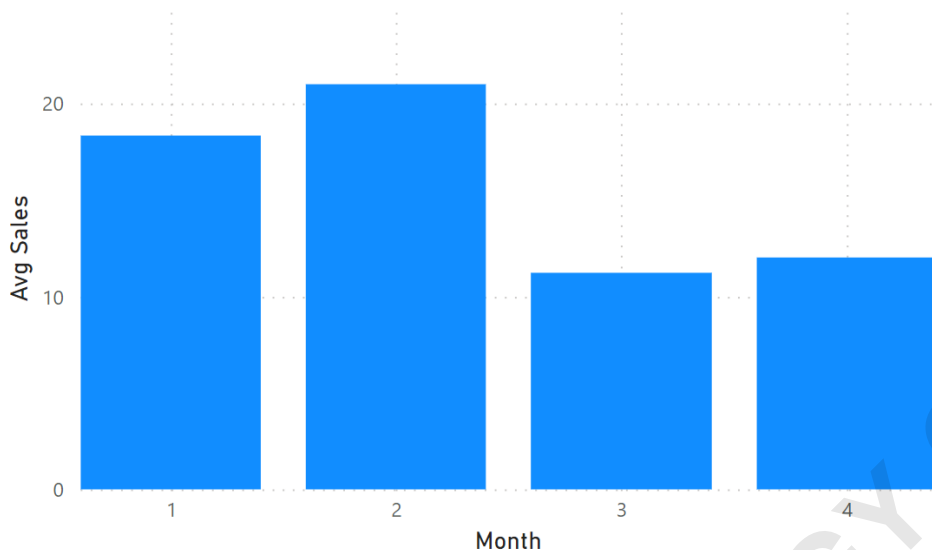
Specifically, locate the `Month` field (the calculated column from Step 1) in your Fields pane and drag it onto the **x-axis** well of the visualization. Subsequently, locate the `Avg sales` measure (the calculated measure from Step 2) and drag it onto the **y-axis** well.



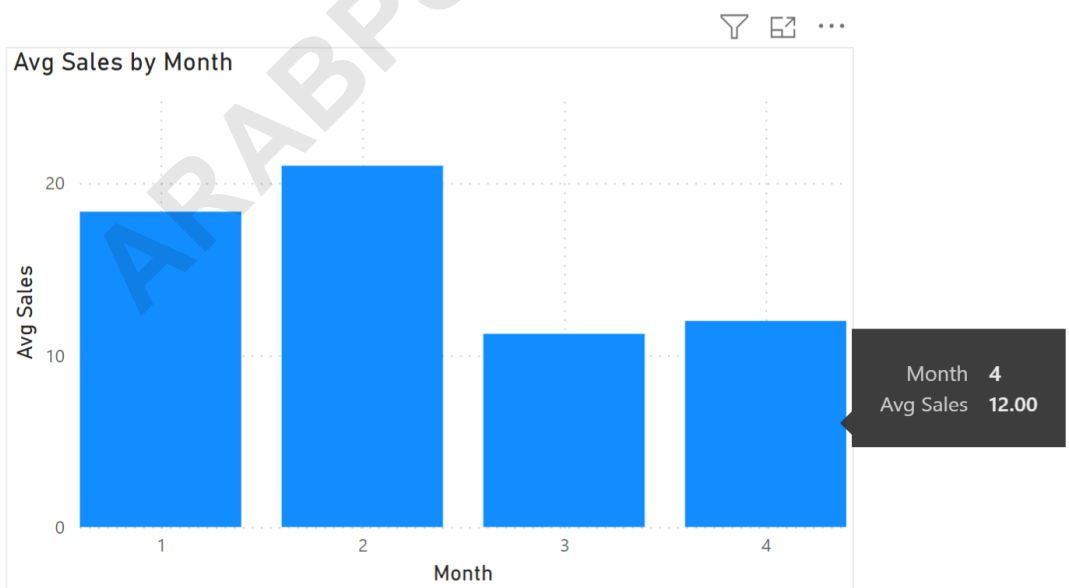
This configuration instructs **Power BI** to plot the average sales value for each unique month number present in the dataset. Note that if the dataset spans multiple years, the visual will currently aggregate all "Month 1s" (Januaries) together, providing a generalized monthly average across the entire data range. For precise year-over-year comparisons, utilizing a full Date table and appropriate time intelligence **measures** is highly recommended.

Upon successfully configuring the fields, the clustered column chart will render, visually representing the average sales performance grouped by month. The X-axis clearly delineates the month numbers, and the height of each column corresponds directly to the calculated **Avg Sales** value for that period.

Avg Sales by Month



This visual representation immediately highlights which months perform above or below the overall trend line, making pattern recognition swift and intuitive. Interactive features in **Power BI** further enhance the **data analysis** process. By hovering the mouse cursor over any individual column, you can access tooltips that display the precise average sales value for that specific month, validating the visual data with exact figures. For example, hovering over Month 4 (April) might reveal a specific average sales metric.



In this specific illustration, the tooltip confirms that for Month 4, the average sales value calculated by our **measure** is **12**. This level of detail is critical for businesses relying on precise monthly metrics to track performance targets and allocate resources effectively for the upcoming periods, enabling better strategic planning and forecasting.

## Conclusion and Next Steps in Data Analysis

Mastering the calculation of monthly averages in **Power BI** is a foundational skill for advanced **data analysis**. By following the two-step process of creating a grouping column using the **MONTH** function and defining a robust aggregation **measure** using **DIVIDE(SUM(), COUNT())**, users can generate reliable, dynamic metrics essential for business intelligence. The subsequent visualization provides immediate clarity on performance trends, transforming complex datasets into actionable insights.

This calculated monthly average serves as a strong foundation for various organizational objectives, including budgeting, sales forecasting, identifying recurring bottlenecks, and optimizing marketing campaigns based on seasonal peaks and troughs. For those interested in extending their **Power BI** capabilities, exploring other advanced **DAX** tutorials covering time intelligence functions is highly recommended, as it allows for deeper comparative analysis such as period-over-period growth.

The following tutorials explain how to perform other common tasks in Power BI: