How to Create a Monthly and Yearly Line Chart in Power BI

Authored by

stats writer

January 28, 2026

Creating effective time-series visualization is fundamental for robust data analysis. When working with large datasets in Power BI, users frequently need to aggregate and display key metrics across specific time hierarchies, such as month and year.

This detailed guide walks you through the precise steps required to construct a dynamic Line Chart in Power BI that clearly illustrates trends segmented by both month and year. This method allows for a comprehensive visual analysis of patterns and fluctuations over time, proving invaluable for business intelligence and decision-making.

We will cover the essential steps, ranging from preparing your data using a custom calculated column to configuring the visualization properties correctly. By following these instructions, you can easily create a line chart in Power BI that presents your data in a clear and organized manner by month and year, greatly aiding in the visual analysis of trends.

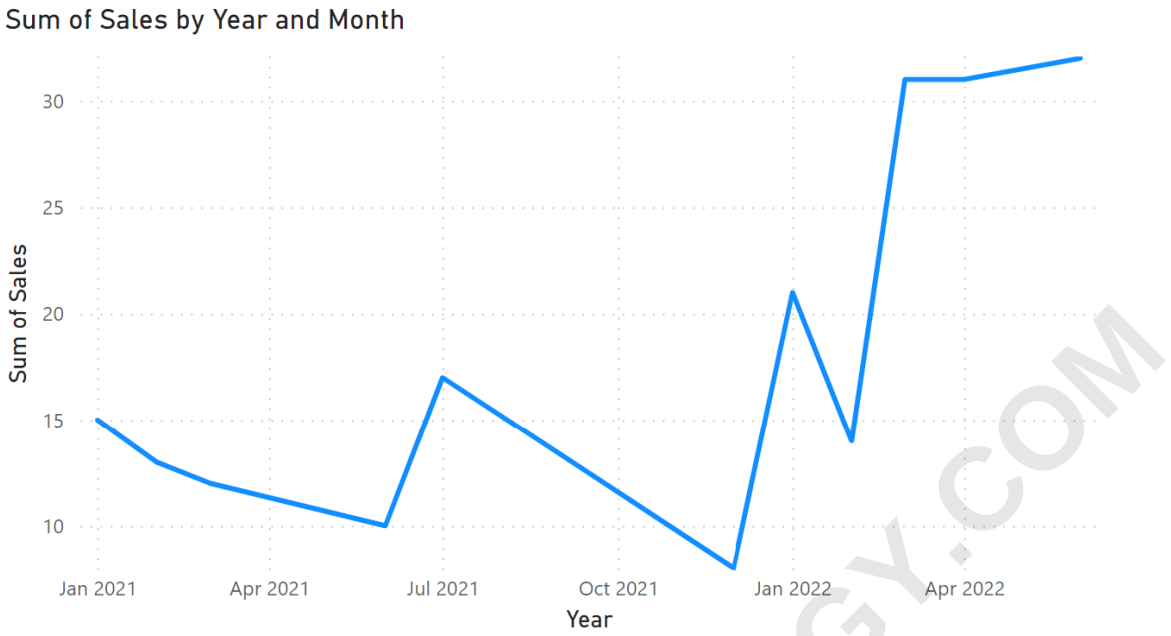
Power BI: Create a Line Chart by Month and Year

Understanding the Goal of Time-Series Visualization

Often, analyzing raw daily data proves unwieldy. To gain actionable insights, it is necessary to summarize the values of a critical variable, such as total sales, aggregated by meaningful time intervals. Displaying data summarized specifically by month and year in a line chart format provides immediate clarity on seasonality and long-term performance trends.

The method outlined below ensures that the X-axis is chronologically ordered by month, while the legend or secondary grouping handles the differentiation by year, allowing for simple year-over-year comparison within the same visual field.

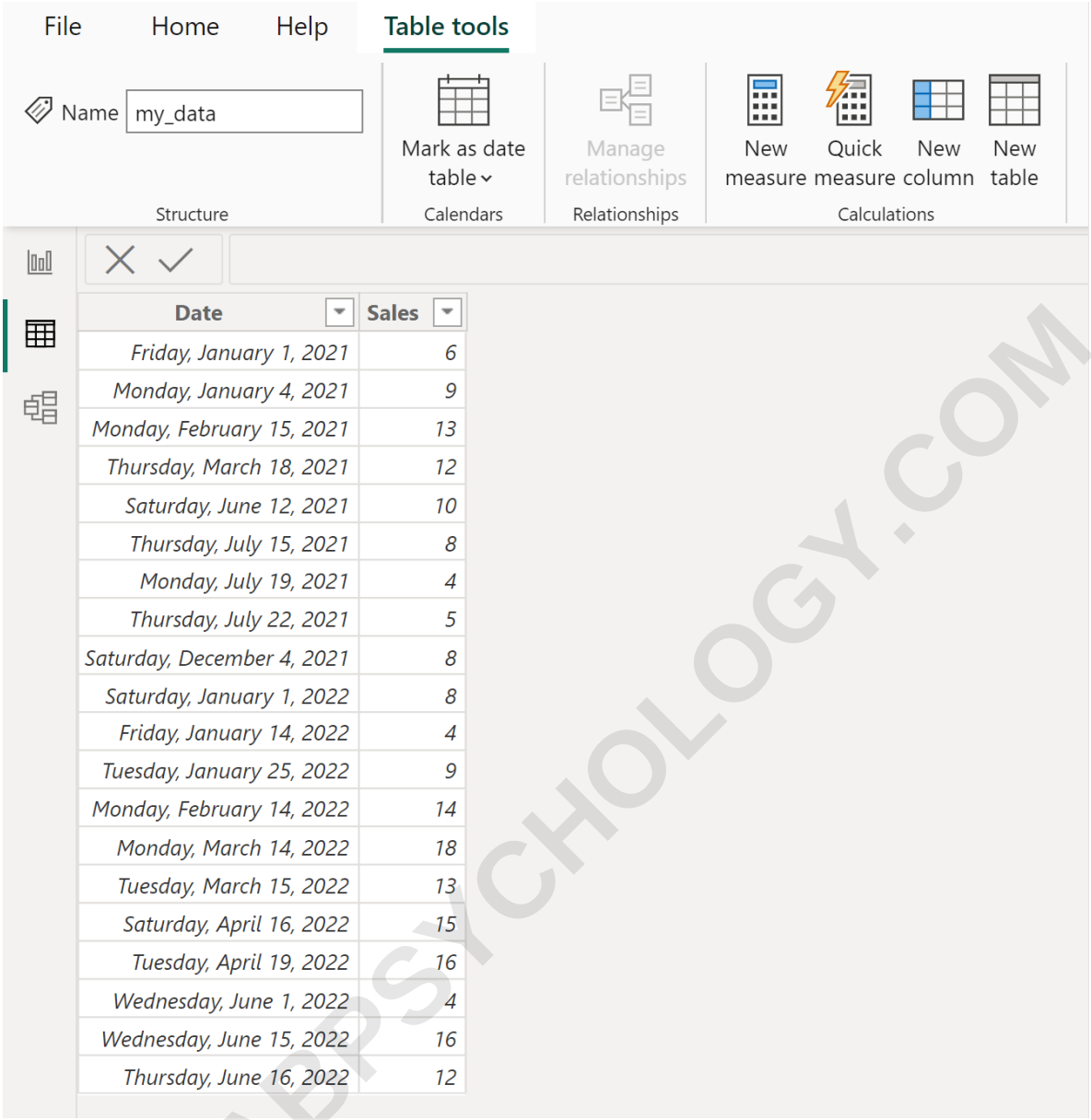
The step-by-step instructions provided below guide you in replicating the following result: a clean, effective line chart displaying aggregated sales data partitioned by monthly increments within distinct yearly timelines.



Step 1: Loading the Dataset into Power BI

The initial stage of any Power BI project requires importing and verifying the source data. We begin by loading a sample dataset into the application. This example dataset contains essential information, including records of total sales conducted by a company on various specific dates.

Ensure that your date column is correctly recognized by Power BI as a **Date/Time** data type to facilitate proper aggregation and hierarchy creation in later steps. If the column is imported as text, the necessary time intelligence functions will not operate correctly, leading to incorrect sorting on the X-axis.

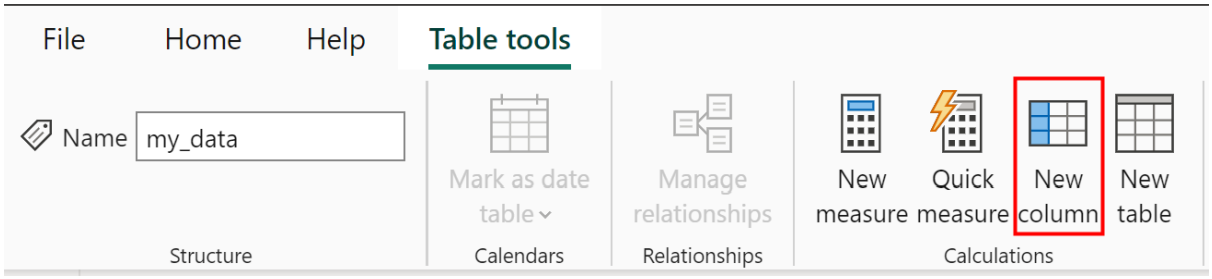


Date	Sales
Friday, January 1, 2021	6
Monday, January 4, 2021	9
Monday, February 15, 2021	13
Thursday, March 18, 2021	12
Saturday, June 12, 2021	10
Thursday, July 15, 2021	8
Monday, July 19, 2021	4
Thursday, July 22, 2021	5
Saturday, December 4, 2021	8
Saturday, January 1, 2022	8
Friday, January 14, 2022	4
Tuesday, January 25, 2022	9
Monday, February 14, 2022	14
Monday, March 14, 2022	18
Tuesday, March 15, 2022	13
Saturday, April 16, 2022	15
Tuesday, April 19, 2022	16
Wednesday, June 1, 2022	4
Wednesday, June 15, 2022	16
Thursday, June 16, 2022	12

Step 2: Data Preparation - Creating the Month and Year Column

To ensure the line chart sorts chronologically by month, irrespective of the year, and to use a clean label for the X-axis, we must create a dedicated calculated column combining the month and year attributes. This derived column will serve as the primary categorical axis for our visualization before we refine the date hierarchy.

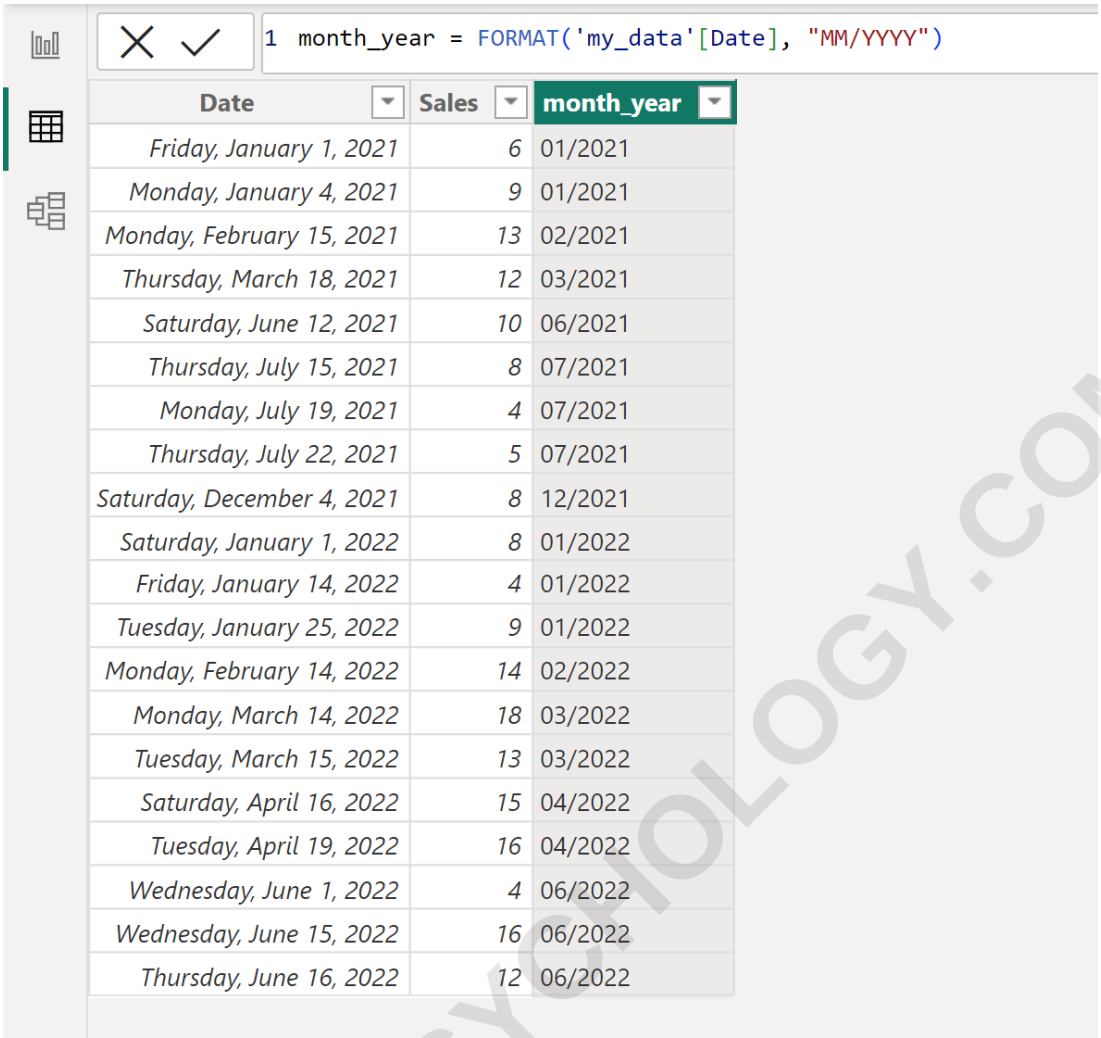
Navigate to the **Table tools** tab located in the top ribbon menu. From there, select the **New column** option. This action opens the formula bar, allowing us to input a custom calculation using DAX (Data Analysis Expressions).



Enter the following formula into the formula bar. This expression uses the `FORMAT` function to extract the month (MM) and year (YYYY) from the original field and concatenate them into a single string format (e.g., "01/2023"). This format is crucial for maintaining correct sequence on the axis.

month_year = FORMAT('my_data', "MM/YYYY")

Executing this formula generates a new column named **month_year**, which is now available in your data model. This column contains the date information structured perfectly for sequential axis plotting in the required format.

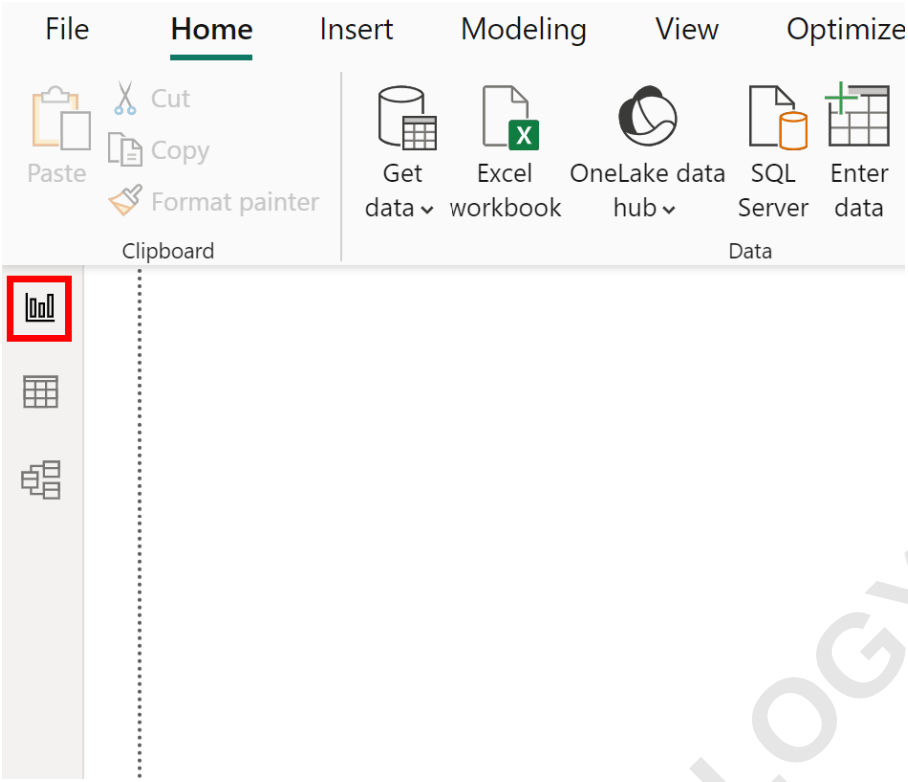


```
1 month_year = FORMAT('my_data'[Date], "MM/YYYY")
```

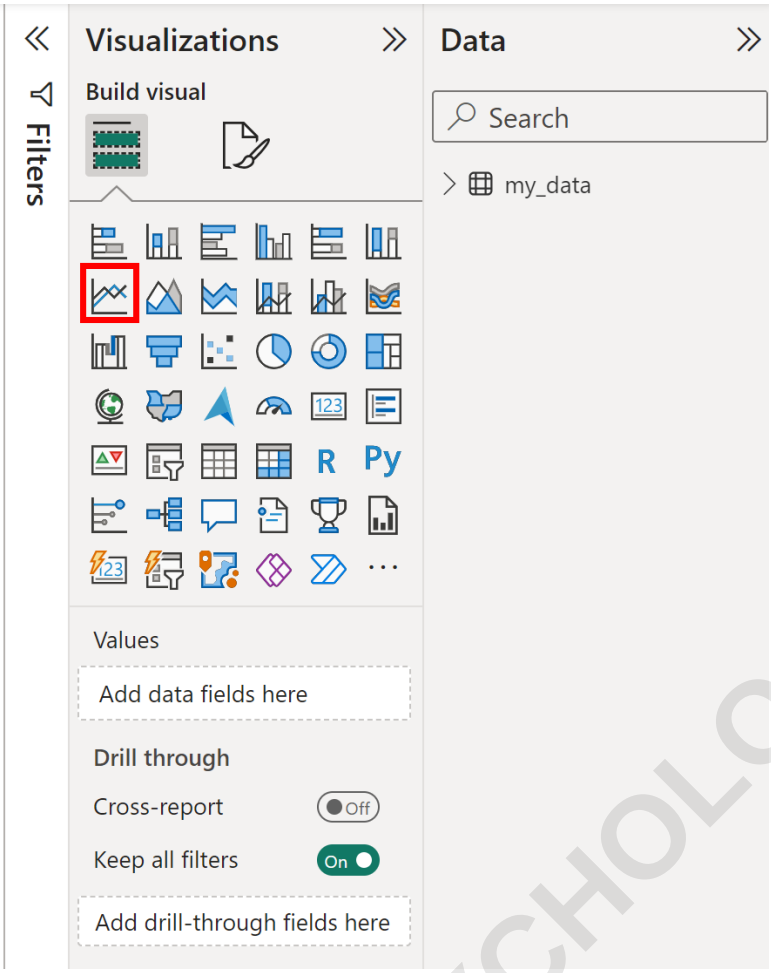
Date	Sales	month_year
Friday, January 1, 2021	6	01/2021
Monday, January 4, 2021	9	01/2021
Monday, February 15, 2021	13	02/2021
Thursday, March 18, 2021	12	03/2021
Saturday, June 12, 2021	10	06/2021
Thursday, July 15, 2021	8	07/2021
Monday, July 19, 2021	4	07/2021
Thursday, July 22, 2021	5	07/2021
Saturday, December 4, 2021	8	12/2021
Saturday, January 1, 2022	8	01/2022
Friday, January 14, 2022	4	01/2022
Tuesday, January 25, 2022	9	01/2022
Monday, February 14, 2022	14	02/2022
Monday, March 14, 2022	18	03/2022
Tuesday, March 15, 2022	13	03/2022
Saturday, April 16, 2022	15	04/2022
Tuesday, April 19, 2022	16	04/2022
Wednesday, June 1, 2022	4	06/2022
Wednesday, June 15, 2022	16	06/2022
Thursday, June 16, 2022	12	06/2022

Step 3: Inserting the Line Chart Visualization

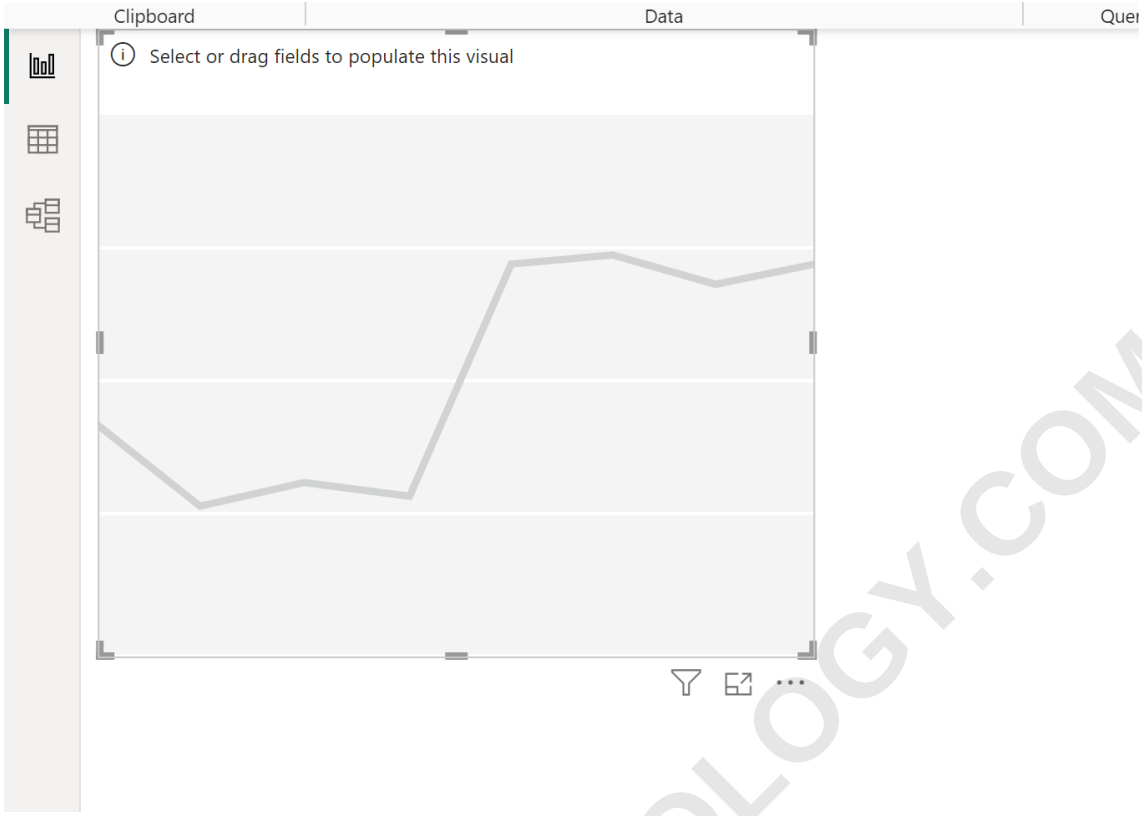
With the necessary calculated column prepared, we proceed to insert the visualization container onto the report canvas. First, click the **Report View** icon found on the left sidebar of the application interface to ensure you are in the correct workspace for building the visual report.



Next, locate the Visualizations pane, typically situated on the right side of the screen. Click on the **Line chart** icon to instantiate the empty visualization object onto your canvas. This foundational step prepares the working area for data assignment.



An empty visual placeholder will immediately appear on the report canvas, ready to be configured with your data fields. If you already have a visualization selected, ensure you click outside of it before selecting the new visual type.

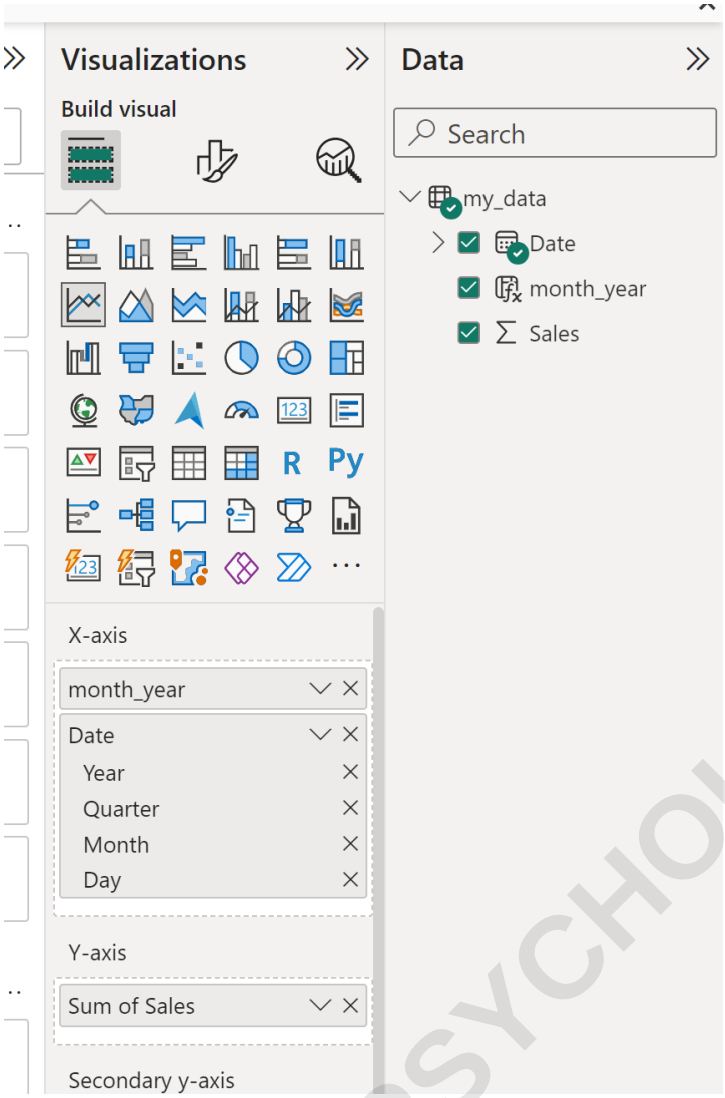


Step 4: Configuring Data Fields and Defining Axes

The critical step in creating this visualization is assigning the correct variables to the X and Y axes, and then managing the date hierarchy. We will be using both the original **Date** field and our new **month_year** field temporarily to establish the correct date hierarchy.

Follow these assignment instructions within the Visualization Fields pane:

Drag the **month_year** variable (created in Step 2) under the **X-axis** label. This initial placement is vital for sorting.
Drag the original **Date** variable under the same **X-axis** label. Power BI automatically creates a date hierarchy upon this action.
Drag the aggregated measure, **Sales**, under the **Y-axis** label. This defines the values measured over time.

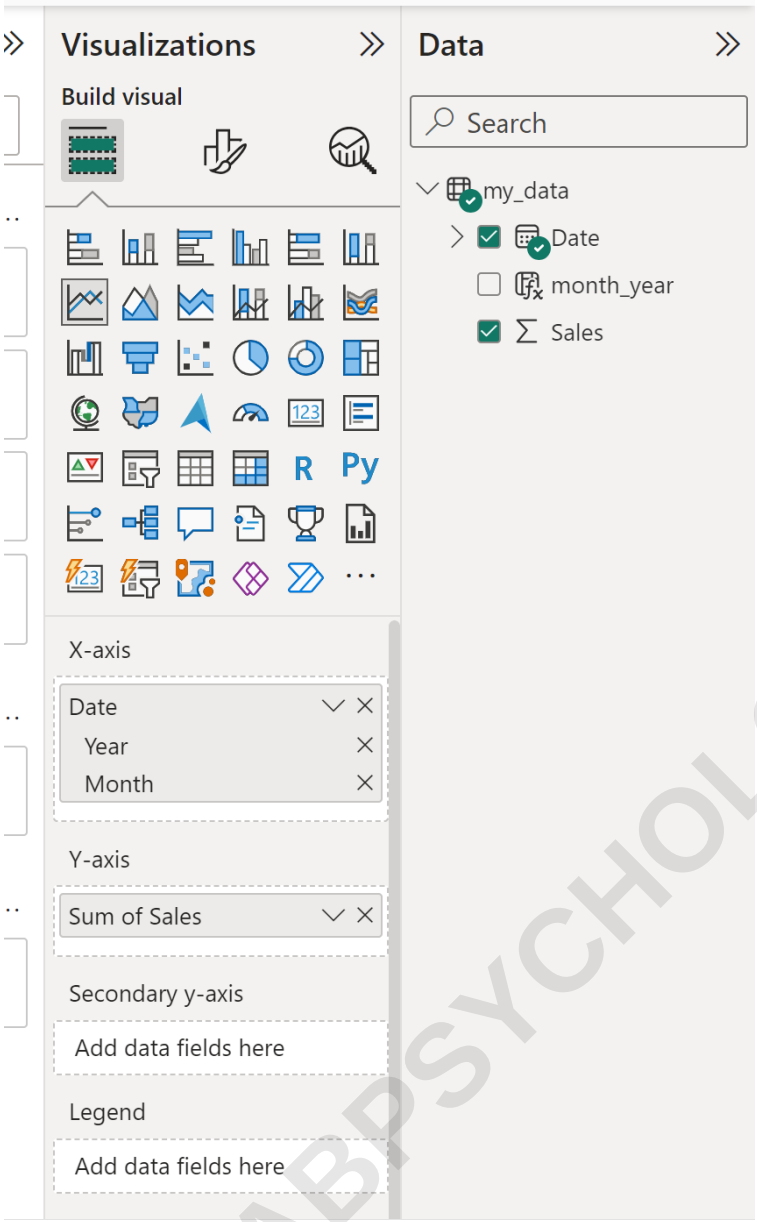


Step 5: Refining the Axis Hierarchy for Monthly Trends

After assigning the initial variables, we need to clean up the X-axis hierarchy to display only the desired month aggregation. The system automatically creates a date hierarchy when a date field is added, typically including Year, Quarter, Month, and Day. We must remove the superfluous levels generated by default.

First, locate the **X-axis** field well. Delete the **month_year** field that we previously added. This field helped establish sorting but is no longer necessary for the final visual structure.

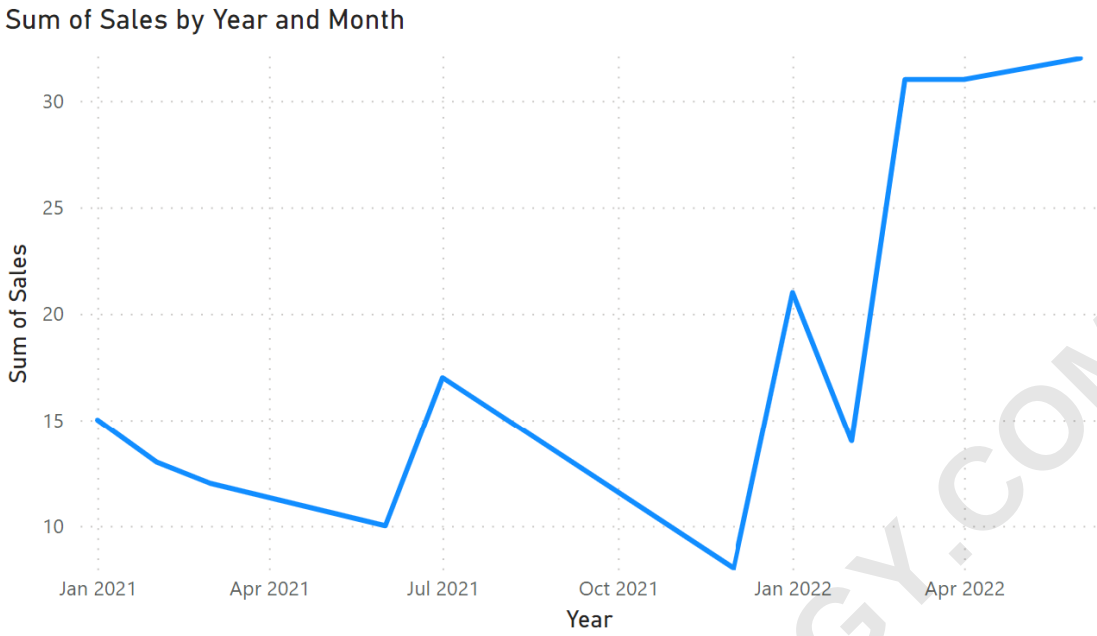
Next, click the drop-down menu associated with the remaining **Date** field under the X-axis. You will see the default hierarchical options. To maintain a clean, month-level view that is segmented by year, delete both the **Quarter** and **Day** options from this hierarchy by clicking the 'x' next to them. This action ensures that the line chart aggregates data correctly and prevents over-granularity.



Step 6: Final Review and Interpretation

Upon completing the axis refinement, the line chart will automatically update to reflect the new hierarchy. The X-axis now displays the data aggregated by month, while the chart structure incorporates the year at the highest level of the hierarchy, allowing for easy comparison of performance across different years on the same monthly timeline.

The end result is a powerful visualization displaying the sum of Sales data broken down by both month and year. This structure effectively highlights seasonal cycles and provides immediate insights into annual growth or decline, which is essential for informed data analysis and high-level reporting.



By mastering this technique of creating custom calculated columns using DAX and carefully managing the date hierarchy, you can generate sophisticated and meaningful time-series visualizations within Power BI. This foundational skill is key to transforming raw data into business intelligence.

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