How to Calculate Days Between Dates in Power BI

Authored by

stats writer

January 26, 2026

Welcome to this detailed guide focusing on a common, yet essential, calculation within the Power BI environment: determining the precise number of days elapsed between a historical or specific date stored in your data trends model and the current system date. This calculation is fundamentally important for various analytical tasks, including tracking project timelines, monitoring inventory aging, or assessing customer tenure. Fortunately, Power BI handles this seamlessly through its powerful formula language, DAX (Data Analysis Expressions).

To achieve this measurement of temporal distance, we utilize the robust DATEDIFF function. This function is specifically designed to calculate the difference between two date or datetime values, providing the output in specified units--in our case, days. By pairing DATEDIFF with the TODAY() function, which dynamically retrieves the current calendar date, we can construct a powerful and dynamic measure or calculated column that updates automatically whenever the Power BI report is refreshed.

Mastering this technique empowers data analysts to transform static date fields into dynamic, actionable metrics, enabling efficient tracking of time intervals and robust analysis of evolving data trends within the visualizations platform. We will now walk through the precise syntax and practical application required to implement this solution effectively.

Power BI: Calculate Days Between Date and Today

Understanding the DATEDIFF Function in DAX

The core mechanism for performing complex date calculations in Power BI is the DAX language. Among its extensive library of functions, DATEDIFF stands out as the fundamental tool for determining temporal separation. When calculating the difference between a past date and the present, we must ensure the parameters are ordered correctly and the unit of measurement is specified precisely.

The syntax for the DATEDIFF function requires three distinct arguments: the starting date, the ending date, and the interval (or unit) in which the difference should be returned. For calculating days remaining or elapsed, it is crucial that the unit is specified as **DAY**. Furthermore, the function calculates the difference by subtracting the start date from the end date. Therefore, if the end date is later than the start date, the result will be a positive number, which is generally desired when calculating elapsed time until today.

In our scenario, the start date will be the column containing the historical dates (e.g., 'my_data'), and the end date will be provided by the TODAY() function. The TODAY() function is dynamic, meaning it retrieves the date according to the system settings when the data model is refreshed or the report is loaded, ensuring that your calculation is always current relative to the moment of

viewing.

The Essential DAX Syntax for Calculating Intervals

To implement this logic, we typically create a new calculated column within the target table. Calculated columns are beneficial because they store the result for every row in the model, allowing the results to be used efficiently in slicers, filters, and row-level context calculations. The structure below represents the necessary DAX formula to derive the day count difference.

You can use the following syntax in DAX to calculate the number of days between a date and today in Power BI:

Difference = DATEDIFF('my_data', TODAY(), DAY)

This particular example creates a new column named **Difference** that contains the number of days between the date found in the **Date** column of a table named **my_data** and today's date. Let's break down the components of this powerful, concise expression to fully appreciate its functionality within the Power BI ecosystem.

Difference = : This defines the name of the new calculated column that will hold the numerical result.
DATEDIFF(...): This is the primary function responsible for the calculation.
'my_data': This serves as the **Start Date** parameter, dynamically referencing the date value for the current row within the specified table and column.
TODAY(): This serves as the **End Date** parameter. It returns the current date without the time component, ensuring a clean day count.
DAY: This specifies the required interval, instructing DATEDIFF to return the result in whole days.

Step-by-Step Example: Calculating Days Between Date and Today in Power BI

To illustrate the practical implementation, we will follow a concrete scenario based on sales data. This approach allows us to see how the theoretical DAX syntax translates into tangible analytical output within the Power BI environment. Understanding the visual steps involved in creating a calculated column is just as important as knowing the underlying formula.

Our goal is to enrich our existing data model by adding a column that quantifies how long ago each sales transaction occurred, measured in days relative to the current date. This metric is invaluable for aging analysis, understanding sales velocity, or highlighting dormant transactions. We assume a standard operational environment where the user has appropriate permissions to modify the data model structure.

The following steps detail the process, from visualizing the raw data to successfully deploying the DATEDIFF function and interpreting the final calculated values. This hands-on example reinforces the utility of dynamic date calculations for insightful data trends analysis.

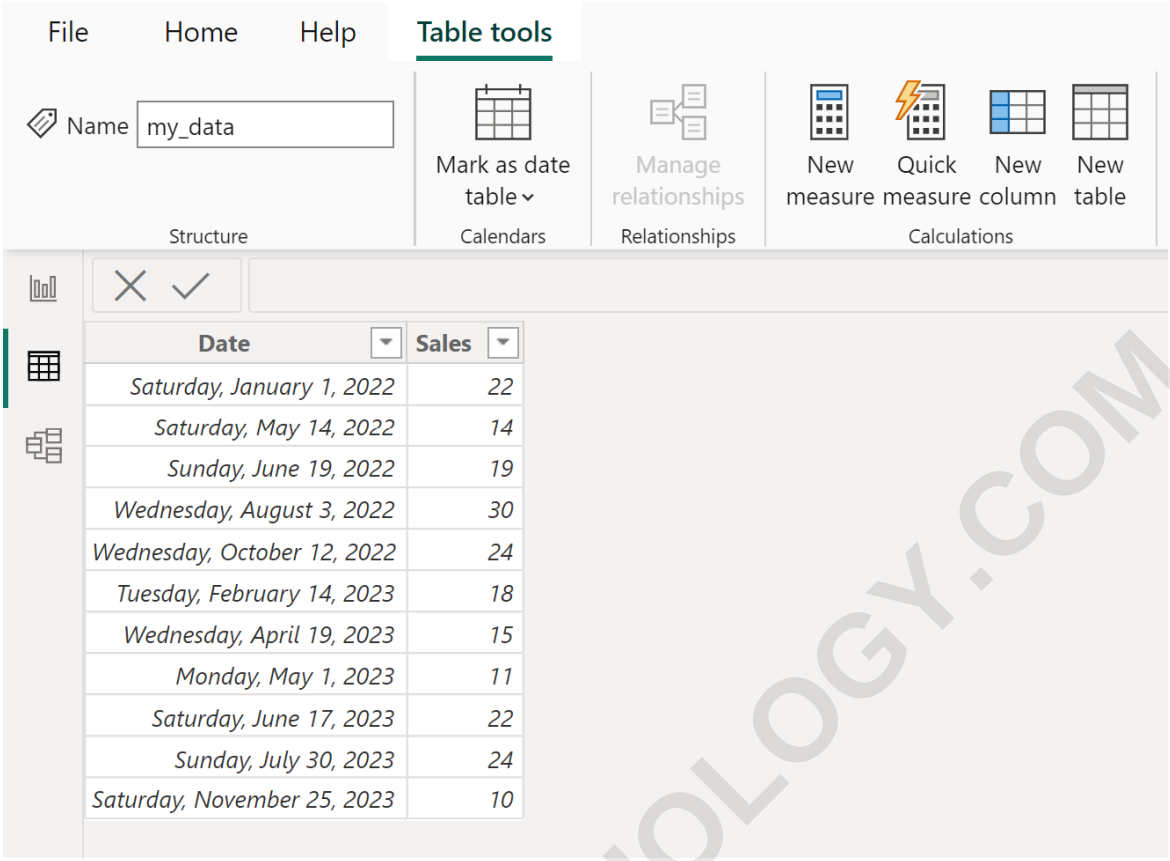
Preparing the Data Model: The `my_data` Table

Before writing any formulas, we must first examine the data structure upon which the calculation will operate. Suppose we have the following table loaded into our Power BI Desktop environment, named **my_data**. This table contains fundamental information regarding total sales recorded on various distinct dates within a hypothetical company's operations. The crucial column for our calculation is the **Date** column, which must be correctly formatted as a Date or DateTime type in the model.

The integrity of the date formatting is paramount. If the **Date** column is incorrectly categorized as text, the DATEDIFF function will fail to execute properly, as it requires valid date objects for both parameters. Ensure that you verify the data type in the Data View prior to proceeding.

For the purpose of this demonstration, we are setting a fixed reference point for "today." This article is being written on **1/8/2024**. Therefore, all calculations presented in the resulting table will be relative to this specific date. In a live Power BI report, the TODAY() function would automatically update based on the last refresh time.

Below is the representation of our initial data table:



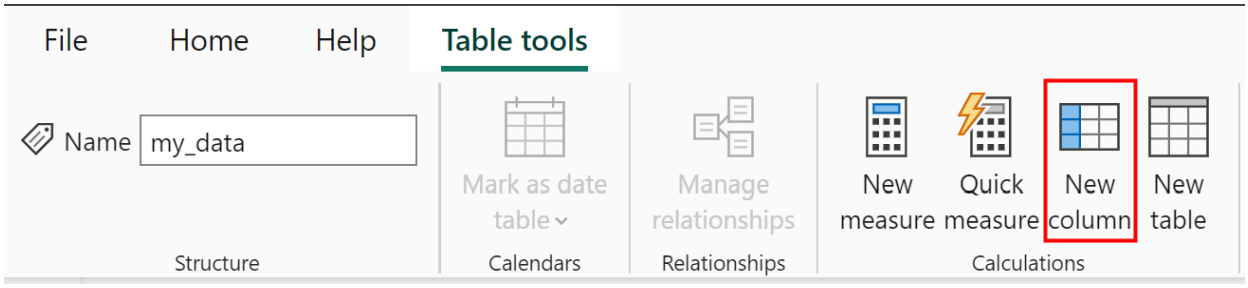
Date	Sales
Saturday, January 1, 2022	22
Saturday, May 14, 2022	14
Sunday, June 19, 2022	19
Wednesday, August 3, 2022	30
Wednesday, October 12, 2022	24
Tuesday, February 14, 2023	18
Wednesday, April 19, 2023	15
Monday, May 1, 2023	11
Saturday, June 17, 2023	22
Sunday, July 30, 2023	24
Saturday, November 25, 2023	10

Our immediate objective is to supplement this table with a new column that quantifies the span between each specific transaction date and our reference date (1/8/2024).

Implementing the New Calculated Column in Power BI

The process of adding a calculated column is straightforward within the Power BI interface. It starts in the Data or Report View, utilizing the ribbon tools designed for model management. To begin the creation of our difference column, we need to navigate to the appropriate menu option.

First, ensure you are selected on the **my_data** table, then click the **Table tools** tab located in the main ribbon interface. Within this tab, you will find the option to add new elements to your table. Locate and click the **New column** icon. This action immediately opens the formula bar, prompting you to input your DAX expression.



Once the formula bar is active, the next step is to carefully input the previously defined DAX formula. Precision in naming the column and referencing the table and date column is vital for the formula to compile correctly.

Then type the following formula into the formula bar:

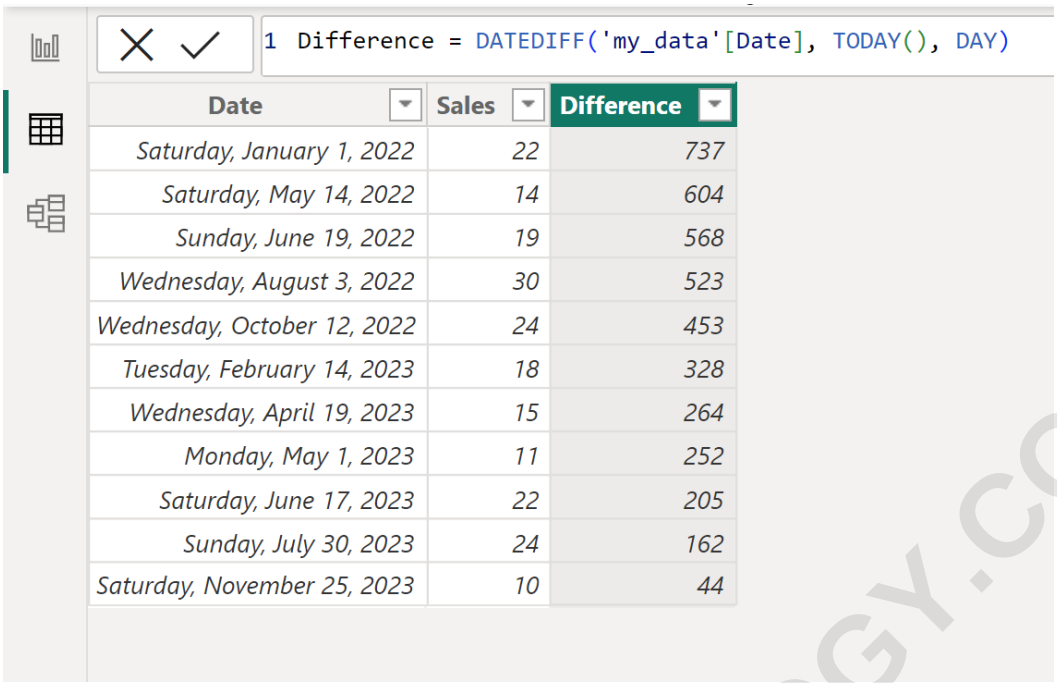
Difference = DATEDIFF('my_data', TODAY(), DAY)

Upon pressing Enter or clicking the checkmark icon in the formula bar, Power BI processes the instruction. It iterates through every row in the **my_data** table, performs the DATEDIFF calculation comparing the row's date to the current date (1/8/2024), and populates the new column named **Difference** with the result.

Interpreting the Results and Validation

The successful execution of the DAX formula yields a new column reflecting the calculated difference in days. This column provides immediate clarity regarding the age of each record. Observing the resulting table allows us to validate that the function performed its intended calculation correctly, comparing the historical dates against the fixed reference date of 1/8/2024.

This will create a new column named **Difference** that contains the number of days between the dates in the **Date** column and today's date of 1/8/2024:



Date	Sales	Difference
Saturday, January 1, 2022	22	737
Saturday, May 14, 2022	14	604
Sunday, June 19, 2022	19	568
Wednesday, August 3, 2022	30	523
Wednesday, October 12, 2022	24	453
Tuesday, February 14, 2023	18	328
Wednesday, April 19, 2023	15	264
Monday, May 1, 2023	11	252
Saturday, June 17, 2023	22	205
Sunday, July 30, 2023	24	162
Saturday, November 25, 2023	10	44

By inspecting the output, we can confirm the accuracy of the date difference computation for several key records. These results demonstrate the chronological distance from the transaction date to the reference date:

There are **737** days between the earliest recorded sale on 1/1/2022 and today's date of 1/8/2024.
There are **604** days between the transaction recorded on 5/14/2022 and today's date of 1/8/2024.
There are **568** days between the sale recorded on 6/19/2022 and today's date of 1/8/2024.

The positive numerical output confirms that the transaction dates are chronologically earlier than the reference date, as expected when calculating elapsed time. If we were to use the DATEDIFF function to calculate the time remaining until a future date, we would simply reverse the order of the parameters.

Advanced Considerations: Time Zones and Data Refresh

While the TODAY() function appears simple, its execution relies heavily on the environment in which the Power BI model is refreshed. When working in Power BI Desktop, TODAY() uses the local date of the user's machine. However, when the model is published to the Power BI Service (Cloud), the calculation is performed using the UTC standard time zone, unless gateway settings or specific refresh configurations override this default.

Analysts must be cognizant of potential time zone discrepancies, especially when dealing with transactional data that spans global regions. A slight misalignment in the date boundary (midnight UTC versus midnight local time) can occasionally lead to a one-day difference in the calculated

results for rows near the date cutoff. For calculations requiring precise time-of-day accuracy, it is advisable to use **NOW()** instead of TODAY() and adjust the interval unit (e.g., to **HOUR** or **MINUTE**) if needed, though **DAY** is sufficient for standard chronological aging analysis.

Furthermore, the calculated column only updates its values when the underlying data model is refreshed. If the report is open for several hours or days without a refresh, the "today" value remains static from the last successful refresh operation. For reports demanding real-time dynamic updates, consider using a DAX measure instead of a calculated column. Measures recalculate dynamically whenever they are placed in a visual or filter context, ensuring maximum temporal accuracy at the moment of viewing, regardless of the last data refresh time.

Conclusion and Further DAX Resources

The ability to calculate the difference between a recorded date and the current date is a cornerstone of temporal analysis in Power BI. By leveraging the straightforward structure of the DATEDIFF function coupled with TODAY(), analysts can efficiently transform static datasets into dynamic tools for tracking project completion, monitoring asset lifecycles, and deriving insightful metrics regarding performance over time.

Remember that while calculated columns provide persistence and speed, measures offer dynamic, context-aware calculation capabilities, and the choice between them depends entirely on the specific requirements of the report consumer and the frequency of required updates. Always ensure your date columns are correctly typed and your DAX syntax adheres to the necessary parameter order to guarantee accurate results.

Note: You can find the complete documentation for the **DATEDIFF** function in DAX provided by Microsoft.

Related Power BI Tutorials

To further enhance your mastery of date intelligence and data transformation within the Power BI platform, explore these related resources. The following tutorials explain how to perform other common tasks in Power BI, often involving intricate use of the DAX engine and its extensive library of time intelligence functions:

Exploring Time Intelligence Functions (e.g., TOTALYTD, SAMEPERIODLASTYEAR).
Calculating Running Totals or Cumulative Sums.
Handling Fiscal Calendars and Custom Date Tables in Power BI.