

What is the process for obtaining the first day of the week in Power BI and can you provide some examples?

Authored by
stats writer

January 27, 2026

RECOMMENDED CITATION

stats writer (2026). *What is the process for obtaining the first day of the week in Power BI and can you provide some examples?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=127977>

Analyzing time series data often requires grouping observations by consistent intervals, and one of the most fundamental requirements in business intelligence is accurately identifying the start date of any given week. In Power BI, the process of accurately determining the first day of the week involves leveraging the powerful capabilities of the DAX (Data Analysis Expressions) language. This is crucial for creating robust time intelligence calculations, generating aggregated weekly reports, and ensuring consistency across different visualizations.

To achieve this calculation, developers typically implement a new custom column within the data model. The core mechanism relies on subtracting the calculated offset of the current day within the week from the original date. This offset calculation is handled efficiently using the native WEEKDAY function, which is highly customizable to adhere to different regional standards (e.g., setting the start of the week to Sunday, Monday, or another day).

Understanding the proper syntax and parameters of the WEEKDAY function is paramount. While the exact implementation details might shift slightly based on whether you are defining a calculated column or a measure, the underlying mathematical logic remains consistent. We must define the return type argument carefully to ensure the result aligns with the organizational definition of the start of the week--whether that is Sunday (often designated as return type 1) or Monday (often designated as return type 2 or 3).

For instance, if the organizational standard requires the week to begin on **Sunday**, the DAX formula must utilize a return type that designates Sunday as the first day (Day 1). A common approach to achieve this offset looks like the following structure, although alternative methods sometimes prove more efficient depending on the specific return type used:

= - WEEKDAY(, 1) + 1

This expression calculates the number of days the current date is past the required Sunday start and subtracts that duration, effectively returning the date corresponding to the Sunday of that week. Conversely, if the requirement is for the week to begin on **Monday**--a standard common in ISO 8601 contexts--a slightly different adjustment is necessary, often relying on the default return type or a specific modifier:

= - WEEKDAY() + 1

This latter formula demonstrates how flexible DAX calculations can be when aligning calendar requirements. In essence, mastering the calculation of the first day of the week in Power BI is fundamental for any serious data analyst working with time-based datasets.

Calculating the First Day of the Week in Power BI Using DAX

Core DAX Formulas for Week Alignment

The goal of week alignment is to map every date within a week to a single, consistent reference date--the starting date of that week. This process simplifies grouping and aggregation tasks within the Power BI environment. While there are several methods to achieve this, the most common and robust approach involves utilizing the WEEKDAY function within a calculated column, as it directly manipulates the underlying date values based on defined calendar standards.

The following two formulas represent the fundamental approaches for defining the week start, catering to the two most common regional standards: beginning the week on Sunday or beginning the week on Monday. It is crucial to select the formula that accurately reflects the business rules governing your data analysis, as inconsistencies here can lead to significant reporting errors when combining data across different sources or systems.

The structure of these formulas relies on the DAX date functionality, where a date is treated internally as a decimal number. Subtracting an integer value (the calculated day offset) from the date column effectively moves the date backward in time to the required starting point of the week.

Formula 1: Determining Week Start (Assuming Sunday is the First Day)

Week Start = 'my_data' - WEEKDAY('my_data', 2)

Formula 2: Determining Week Start (Assuming Monday is the First Day)

Week Start = 'my_data' - WEEKDAY('my_data', 2) + 1

Both of these calculated expressions create a new custom column, typically named **Week Start** or similar descriptive title, which populates the corresponding date that marks the beginning of the week for every record in the original **Date** column. The key difference lies in the final adjustment (+1) and the use of the return type parameter within the WEEKDAY function, which we will explore in detail below.

Deep Dive into the WEEKDAY Function Parameters

The effectiveness of these formulas hinges entirely on the correct application of the WEEKDAY function. This function returns an integer ranging from 1 to 7, representing the day of the week for

a given date. Its second argument, the **return_type** parameter, dictates which day corresponds to the number 1 (the first day of the week) and subsequently how the remaining days are numbered.

In the examples provided in the introductory section, the return type was implicitly defined, but in the optimized formulas shown above, the return type **2** is used. Return type 2 specifies that **Monday is Day 1** and the numbering continues sequentially until Sunday (Day 7). Understanding this parameter is vital for adjusting the final formula correctly, particularly when aiming for a Sunday start.

Consider the structure of the formula: `Date - WEEKDAY(Date, return_type) + adjustment`. If we use return type 2 (Monday=1, Sunday=7), and we want Sunday to be the start date, we must subtract the correct number of days to land on Sunday. Since Monday is 1, and Sunday is 7, the simple subtraction of `WEEKDAY(Date, 2)` will always land on the previous day relative to the return type's start. This requires a slight adjustment depending on whether we target Monday or Sunday as the final output.

Scenario 1: Calculating Week Start Based on Sunday

When the desired outcome is for the calculated **Week Start** column to display the Sunday preceding or coinciding with the date, we must construct the DAX expression such that the subtraction logic accurately offsets the date to Sunday. Let's analyze the formula presented:

Week Start = 'my_data' - WEEKDAY('my_data', 2)

In this particular formula, we are leveraging return type 2, where Monday is counted as 1 and Sunday as 7. If today is Wednesday (which returns 3 via WEEKDAY), subtracting 3 days from Wednesday lands us on Sunday. If today is Sunday (which returns 7), subtracting 7 days lands us on the previous Sunday. Therefore, when using return type 2, the raw subtraction results in a Sunday start, as long as the date column 'my_data' refers to a valid date column within your data model.

This is often the most concise way to define a Sunday start when utilizing return type 2. It requires no additional '+1' adjustment because the numbering convention (Monday=1 through Sunday=7) conveniently provides the exact offset needed to land on the Sunday date when subtracted directly.

Scenario 2: Calculating Week Start Based on Monday (ISO Standard)

For organizations adhering to the ISO 8601 standard, the week must always begin on Monday. This requires a minor modification to the previous formula to ensure that Monday is correctly identified as Day 1. We continue to use return type 2 for consistency, but we add an increment to

account for the resulting Sunday alignment:

Week Start = 'my_data' - WEEKDAY('my_data', 2) + 1

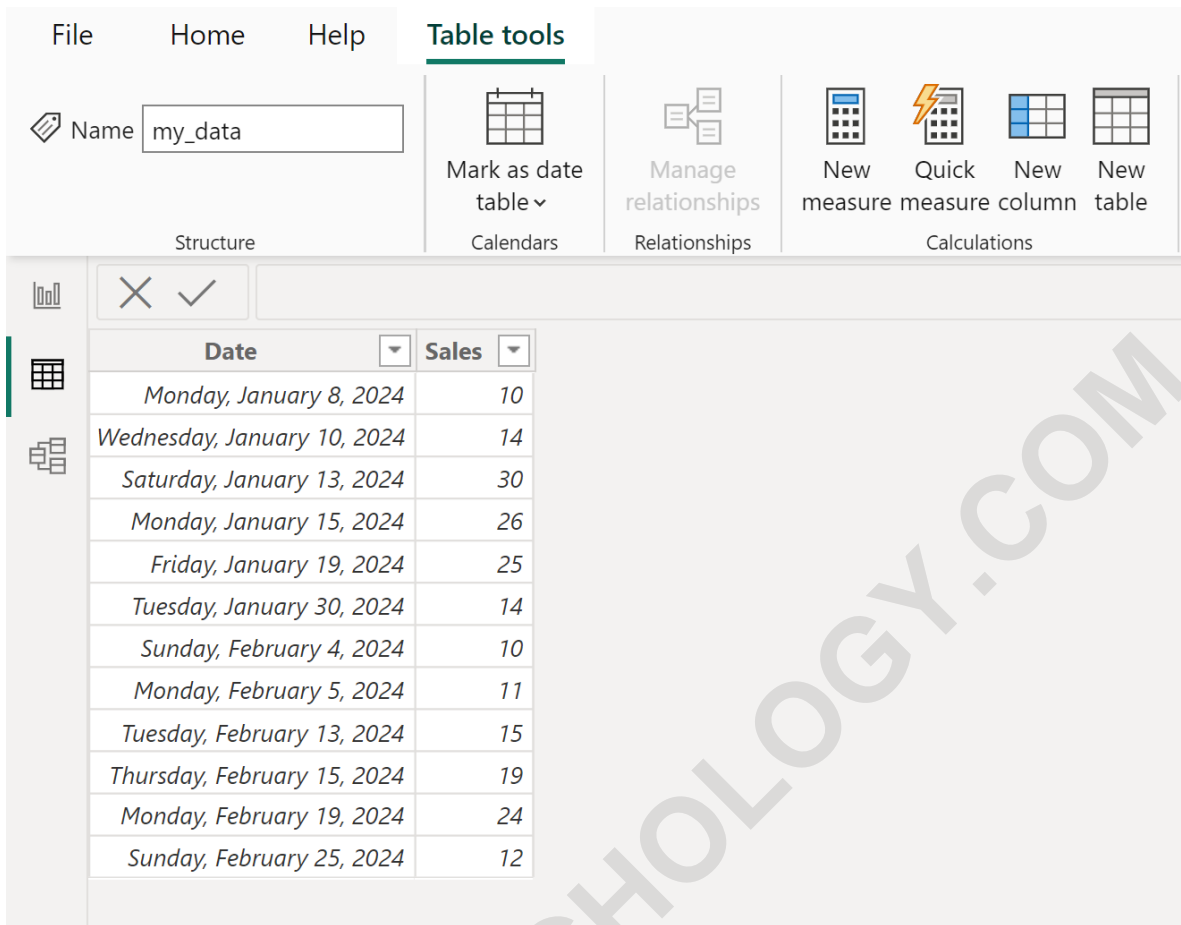
If we examine the mechanics of this formula, we first calculate the Sunday alignment by executing `'my_data' - WEEKDAY('my_data', 2)`. As established previously, this interim result yields the Sunday of that week. By adding `+ 1` to this Sunday date, the calculation is immediately shifted forward by one day, precisely landing on the following Monday, which is the required start of the week.

This method ensures accurate weekly aggregation for Monday-start weeks, a requirement frequently encountered in European and international reporting contexts. This flexibility within DAX allows analysts to meet diverse geographic and industrial reporting standards without complex conditional logic.

Practical Implementation in Power BI: A Step-by-Step Example

To illustrate the application of these formulas, we will walk through the process of creating a new calculated column in Power BI Desktop. Assume we have a dataset loaded into the canvas that details daily sales figures, and we need to group these figures by their corresponding week start date.

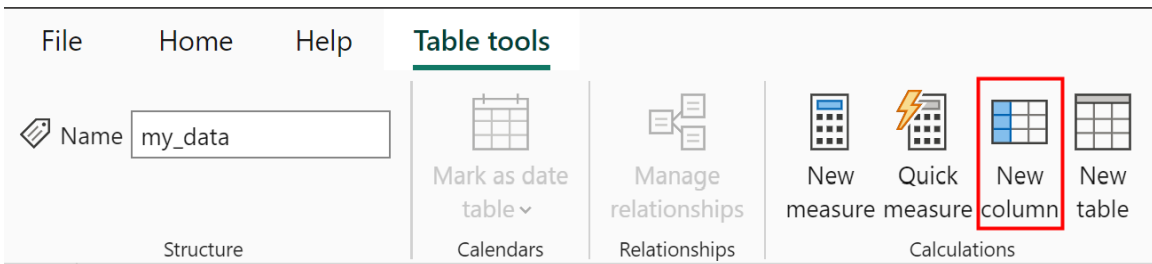
Suppose our initial dataset in Power BI contains transactional data, including a column named **Date**:



Our objective is to generate a new custom column that contains the reference date for the beginning of the week for every row in this sales table. This column will serve as the primary key for any subsequent weekly analysis or aggregation. The procedure begins by navigating to the modeling tools within the Power BI interface.

Creating a New Column for Weekly Alignment

The first step involves activating the column creation feature. This is typically done within the Data View or Report View by selecting the relevant table and navigating to the toolbar. Specifically, you must click the **Table tools** tab at the top of the application window and then select the **New column** icon:



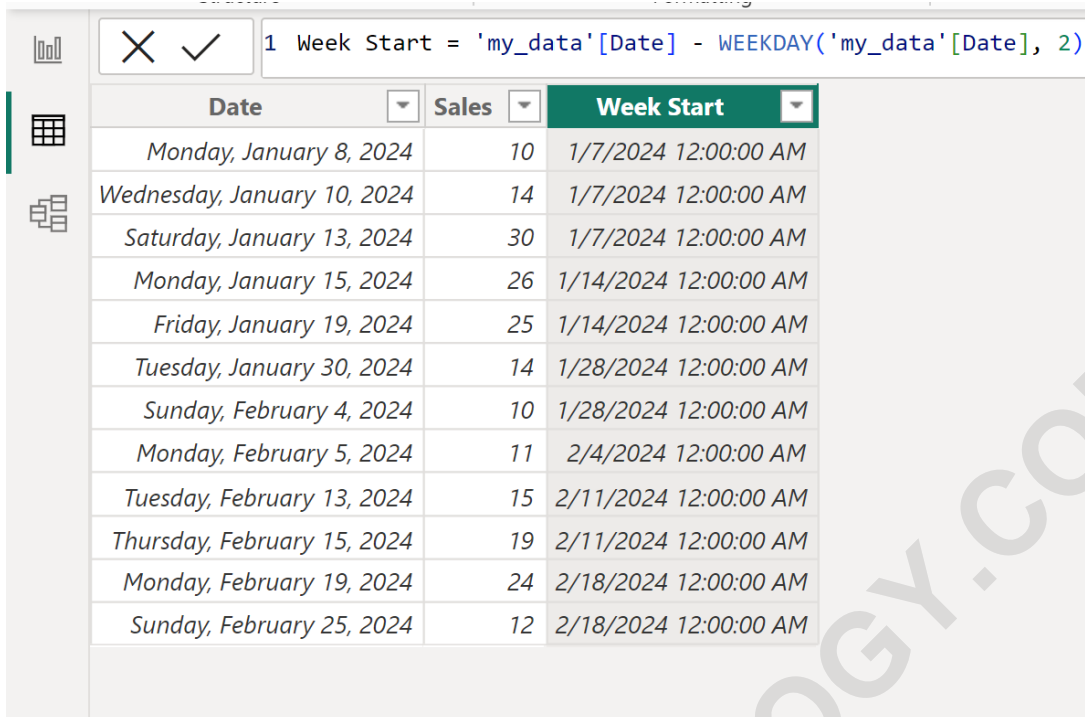
Upon clicking **New column**, the DAX formula bar will appear, prompting the user to define the calculation logic for the new field. This is where we input the chosen formula based on whether we require a Sunday or Monday start date.

Implementing the Sunday Start Formula

If the requirement dictates that the week begins on **Sunday**, the following DAX formula is entered into the formula bar:

Week Start = 'my_data' - WEEKDAY('my_data', 2)

Executing this formula instantaneously creates the new column named **Week Start**. This column now holds the Sunday date corresponding to the week of the original transaction date. For any date that falls between Monday and Sunday, the calculated column will uniformly display the preceding Sunday's date, establishing a consistent weekly grouping key.



Date	Sales	Week Start
Monday, January 8, 2024	10	1/7/2024 12:00:00 AM
Wednesday, January 10, 2024	14	1/7/2024 12:00:00 AM
Saturday, January 13, 2024	30	1/7/2024 12:00:00 AM
Monday, January 15, 2024	26	1/14/2024 12:00:00 AM
Friday, January 19, 2024	25	1/14/2024 12:00:00 AM
Tuesday, January 30, 2024	14	1/28/2024 12:00:00 AM
Sunday, February 4, 2024	10	1/28/2024 12:00:00 AM
Monday, February 5, 2024	11	2/4/2024 12:00:00 AM
Tuesday, February 13, 2024	15	2/11/2024 12:00:00 AM
Thursday, February 15, 2024	19	2/11/2024 12:00:00 AM
Monday, February 19, 2024	24	2/18/2024 12:00:00 AM
Sunday, February 25, 2024	12	2/18/2024 12:00:00 AM

Examining the results confirms the alignment:

For the date 1/8/2024 (a Monday), the calculated first day of the week is correctly identified as **1/7/2024** (Sunday).

For 1/10/2024 (a Wednesday), the week start remains **1/7/2024**.

For 1/13/2024 (a Saturday), the week start is still **1/7/2024**.

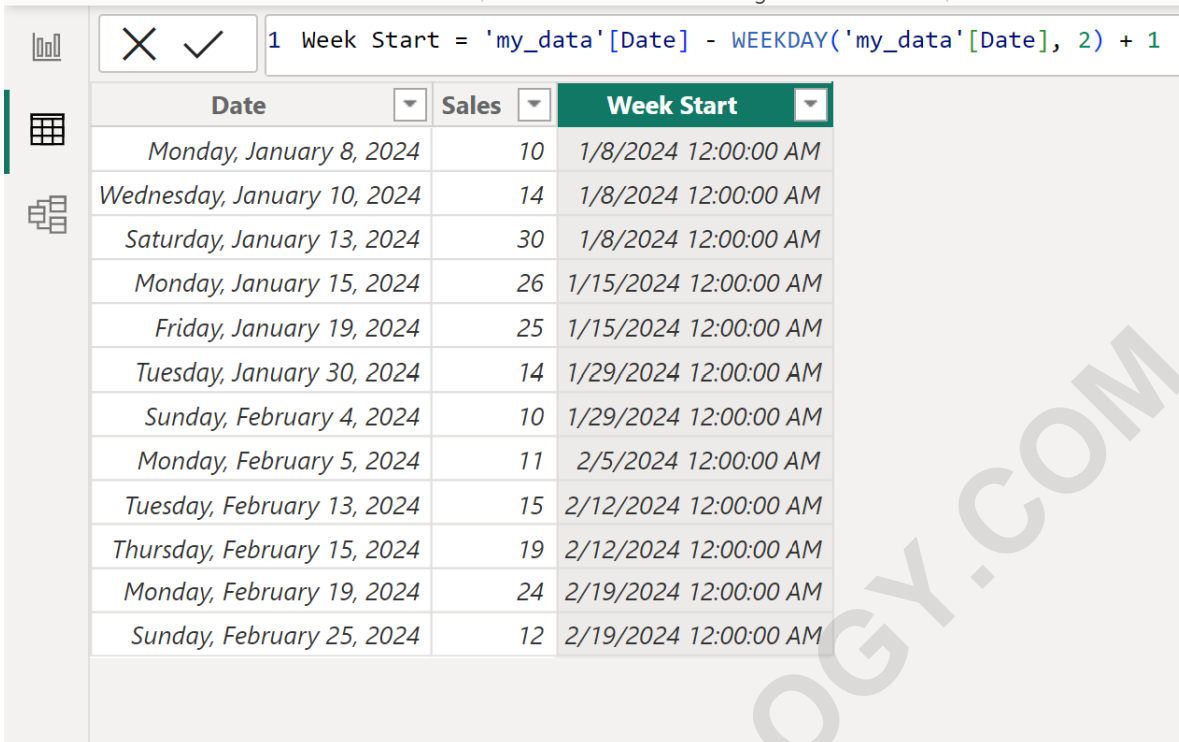
The date 1/15/2024 (the subsequent Monday) correctly aligns with the next Sunday start, which is **1/14/2024**.

Implementing the Monday Start Formula

If, conversely, the business requirement is to adhere to a **Monday** start date, the formula requires the previously discussed +1 adjustment. The analyst would input the following expression into the formula bar:

Week Start = 'my_data' - WEEKDAY('my_data', 2) + 1

This calculation ensures that the resulting **Week Start** column is shifted one day forward from the Sunday baseline, thereby designating Monday as the start of the week for all records.



The screenshot shows a Power BI interface with a calculated column formula and a data table. The formula is: `1 Week Start = 'my_data'[Date] - WEEKDAY('my_data'[Date], 2) + 1`. The table below shows the results of this formula applied to a dataset.

Date	Sales	Week Start
Monday, January 8, 2024	10	1/8/2024 12:00:00 AM
Wednesday, January 10, 2024	14	1/8/2024 12:00:00 AM
Saturday, January 13, 2024	30	1/8/2024 12:00:00 AM
Monday, January 15, 2024	26	1/15/2024 12:00:00 AM
Friday, January 19, 2024	25	1/15/2024 12:00:00 AM
Tuesday, January 30, 2024	14	1/29/2024 12:00:00 AM
Sunday, February 4, 2024	10	1/29/2024 12:00:00 AM
Monday, February 5, 2024	11	2/5/2024 12:00:00 AM
Tuesday, February 13, 2024	15	2/12/2024 12:00:00 AM
Thursday, February 15, 2024	19	2/12/2024 12:00:00 AM
Monday, February 19, 2024	24	2/19/2024 12:00:00 AM
Sunday, February 25, 2024	12	2/19/2024 12:00:00 AM

Conclusion and Advanced Considerations in Time Intelligence

The ability to accurately and dynamically determine the first day of the week is a fundamental building block in advanced time intelligence within Power BI. By leveraging the flexibility of the WEEKDAY function and strategic arithmetic adjustments, analysts can create reliable date dimensions necessary for complex reporting, such as calculating week-over-week growth or performing rolling weekly averages.

While the demonstrated method using calculated columns is highly effective for basic data models, analysts should note that for extremely large datasets or complex, multiple-calendar scenarios, alternative approaches might be considered. These include utilizing robust date dimension tables (which typically pre-calculate all necessary date attributes like Week Start Date) or employing other specialized DAX functions like `CALENDARAUTO` or `M/Query` transformations for initial data preparation. However, for quick and specific alignment needs within an existing transactional table, the calculated column approach remains the most straightforward and powerful tool.

It is highly recommended that users consult the official documentation for the **WEEKDAY** function in DAX to explore all available return types (1, 2, 3, etc.), as these options allow for customization based on region-specific or specialized calendar requirements that deviate from the standard Sunday or Monday starts. Choosing the correct return type minimizes the complexity of the adjustment logic needed in the final formula.

In addition to these core calculations, the analyst should also explore related tutorials detailing other common time intelligence tasks in Power BI to build a comprehensive analytical toolkit. Mastering these foundational techniques ensures the integrity and reliability of all temporal analyses conducted within the business intelligence platform.

ARABPSYCHOLOGY.COM