

How to Find the First Sunday of Every Month in Excel

Authored by
stats writer

January 21, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Find the First Sunday of Every Month in Excel*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=126757>

Microsoft Excel provides powerful tools for manipulating time and date data, which is essential for reporting, scheduling, and analysis across various fields. One common requirement in business intelligence and personal finance is locating specific days, such as the first Sunday, within any given month. While this might seem complex due to the varying number of days in a month and how weeks align, **Excel** offers specialized functions to solve this efficiently and mathematically. The core of this solution lies in combining several fundamental date functions to construct a robust and reusable calculation.

The objective is to create a dynamic **formula** that automatically adjusts, regardless of the year or month provided, to pinpoint the first occurrence of Sunday. By leveraging functions that identify the day number of the week and those that manipulate the date itself, we can bypass tedious manual calendar lookups. This article serves as an expert guide, detailing the required functions, explaining the logic behind the complex calculation, and providing a step-by-step example for finding the first Sunday of every month in your dataset. Understanding this methodology unlocks significant efficiency in handling chronological data within the Excel environment.

Mastering Date Calculations in Excel

Effective date manipulation in **Excel** relies on understanding how the system stores and interprets dates. Excel treats dates as serial numbers, where January 1, 1900, is serial number 1, and subsequent dates are incremented accordingly. This numerical representation allows standard mathematical operations (addition, subtraction) to be performed on dates, making complex scheduling and chronological calculations possible. When aiming to locate a specific weekday, such as the first Sunday, we must convert the date serial number back into recognizable weekday numbers.

The challenge of finding the first Sunday is twofold: first, identifying the starting day of the month, and second, calculating the offset needed to reach the next Sunday. Since months begin on different days of the week, the offset required to hit the first Sunday can range from zero (if the month starts on a Sunday) up to six days. Using a combination of functions allows us to standardize this calculation into a single, elegant **formula** that applies universally across all months and years. This avoids the necessity of creating multiple conditional statements (IF functions) which would complicate the spreadsheet design unnecessarily.

Achieving this level of precision requires harnessing functions that are specifically designed for date component extraction and manipulation. We must be able to isolate the day of the month, construct a date based on specified components, and determine the numerical day of the week for any given date. These core tools--the WEEKDAY function, the DATE function, and the DAY function--form the foundation of our solution, providing the necessary mathematical framework for complex **date calculations**.

Understanding the Core Excel Functions

To fully appreciate the resulting formula, it is essential to understand the roles of the primary functions utilized. The solution relies heavily on how Excel assigns numerical values to days of the week and how we can use these numbers to calculate required date offsets.

The WEEKDAY function is paramount. It takes a date as input and returns a number from 1 to 7 corresponding to the day of the week. By default (or when using type 1), Sunday is represented by 1, Monday by 2, and so on, up to Saturday being 7. This numerical output is critical because it tells us exactly how many days we have passed since Sunday, allowing us to calculate the required backward adjustment to align any date precisely to the preceding or succeeding Sunday.

The DAY function is used for extracting information. It takes a date and returns only the day of the month as a number (1 through 31). In our context, we use the **DAY function** to determine how many days to subtract from a date to effectively "roll back" to the last day of the previous month. For instance, if a date is January 15th, subtracting 15 days takes us back to December 31st of the previous year. This is a crucial step in isolating the beginning of the desired month.

While not explicitly visible in the final condensed formula presented below, the logic often relies on the functionality of the DATE function when constructing dates from scratch. For this specific calculation, we utilize the arithmetic properties of **Excel's** serial numbering system to manipulate the date directly, achieving the same result as complex nested date constructions, but in a far more compact and efficient manner.

Deconstructing the First Sunday Formula Logic

To accurately locate the first Sunday, we must manipulate the date provided in a reference cell (e.g., **A2**) to find the start of its corresponding month and then apply a standard offset to land on the first Sunday. The complete formula used is:

You can use the following highly efficient **formula** in Excel to find the first Sunday of a given month, based on any date within that month stored in the reference cell:

=A2-DAY(A2)+8-WEEKDAY(A2-DAY(A2))

This particular example returns the first Sunday of the month for the date contained in cell **A2**.

Let us break down this sophisticated calculation into its three core components to understand how it functions:

Finding the Last Day of the Previous Month: `A2 - DAY(A2)`: This segment establishes the

baseline. If **A2** contains a date (e.g., 1/15/2023), the DAY function returns 15. Subtracting 15 from the date 1/15/2023 results in 12/31/2022. This date is mathematically equivalent to the 0th day of the current month cycle. Establishing this zero point is necessary because all subsequent calculations must start from the month boundary, not an arbitrary mid-month date.

Establishing the Target Week Range: `... + 8`: By adding 8 days to the last day of the previous month (the result of step 1), we land on a date between the 1st and the 8th of the current month. For example, if the previous step resulted in 12/31/2022, adding 8 yields 1/8/2023. This ensures that the calculated date falls within the range where the first Sunday must exist, while also providing enough mathematical cushion to adjust backward to Sunday regardless of the starting day of the month.

Calculating the Backward Offset: `-WEEKDAY(...)`: The final part of the WEEKDAY function subtracts the day number of the week from the date established in step 2. Crucially, the argument within the **WEEKDAY** function is `(A2-DAY(A2))`, which calculates the day of the week for the last day of the *previous* month. By using this logic and subtracting the result, we align the date precisely to the preceding Sunday relative to that boundary point. This complex adjustment ensures that the calculation lands exactly on the first Sunday of the current month. If the result of the WEEKDAY calculation is 1 (Sunday), we subtract 1, and the overall calculation is balanced to land correctly. If the result is 7 (Saturday), subtracting 7 adjusts the date backward seven days from the 8th of the month, placing it correctly on the 1st (if the 1st is Sunday) or 7 days back from the 8th.

Applying the Universal Formula

The true power of this formula lies in its applicability. Because it uses the date serial numbers and the standardized WEEKDAY function output, this single string of code can be copied and applied across hundreds or thousands of rows in your **Excel** spreadsheet, instantly calculating the required date for every entry. This capability is vital for scheduling tasks that recur weekly or monthly, such as calculating payroll deadlines, scheduling maintenance routines, or determining reporting cutoff dates.

To implement this in practice, you only need a column containing any valid date within the target month. Even if your dates are inconsistent (e.g., one row has 1/5/2023 and the next 1/28/2023), the formula will correctly identify the first Sunday of January 2023 for both entries because the `A2-DAY(A2)` portion always normalizes the starting point to the end of the previous month. This robustness minimizes data cleaning requirements before calculation.

Remember that when you enter the formula into the target cell, the result will initially be displayed as an **Excel** serial number. You must ensure that the cell formatting is set to a proper Date format (e.g., Short Date or Long Date) to view the result clearly, such as "1/1/2023" instead of "44927."

This conversion step is standard practice in all spreadsheet **date calculations**.

Step-by-Step Practical Example

To illustrate the application of this formula, we will work through a practical scenario where we have a list of dates and need to extract the first Sunday corresponding to each month represented in that list.

Suppose we have the following list of reference dates stored in Column A of our **Excel** worksheet. These dates cover several different months throughout the year:

	A	B	C	D	E	F
1	Date					
2	1/1/2023					
3	2/1/2023					
4	3/1/2023					
5	4/1/2023					
6	5/1/2023					
7	6/1/2023					
8	7/1/2023					
9	8/1/2023					
10	9/1/2023					
11	10/1/2023					
12	11/1/2023					
13	12/1/2023					
14						
15						
16						
17						
18						

Our goal is to populate Column B with the date of the first Sunday for the month corresponding to the date in the adjacent cell in Column A. This allows for quick reference and further calculations based on these significant monthly milestones.

Inserting the Formula

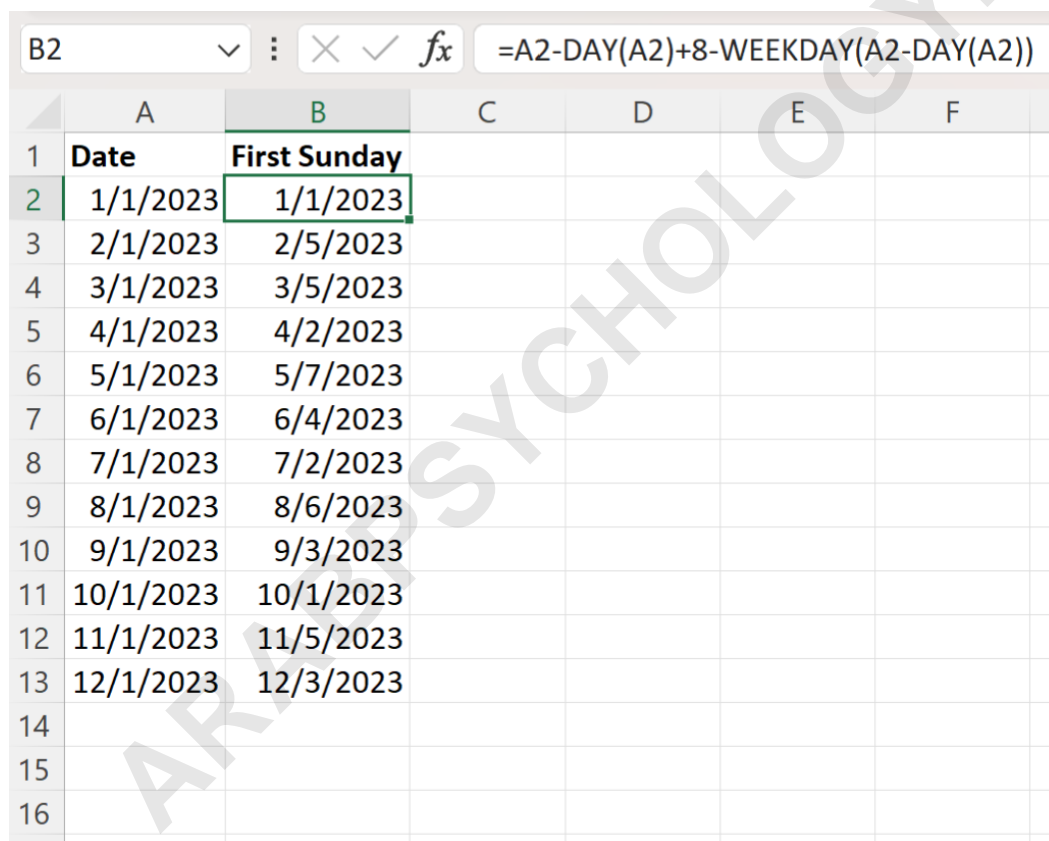
We begin by selecting cell **B2**, the location where we want the result for the first date (1/1/2023). We then type or paste the complete formula, referencing the date in **A2**:

=A2-DAY(A2)+8-WEEKDAY(A2-DAY(A2))

Upon pressing Enter, **Excel** performs the complex calculation, determining the first Sunday of January 2023 based on the entry in **A2**.

Applying the Formula to the Dataset

Once the formula is correctly entered in **B2**, we can efficiently apply it to the rest of the dataset. This is done by utilizing **Excel's** "fill handle"--the small square located at the bottom right corner of the selected cell. We click and drag this formula down to each remaining cell in Column B. Since we used relative referencing (A2), the formula automatically adjusts to A3, A4, and so on, for subsequent rows:



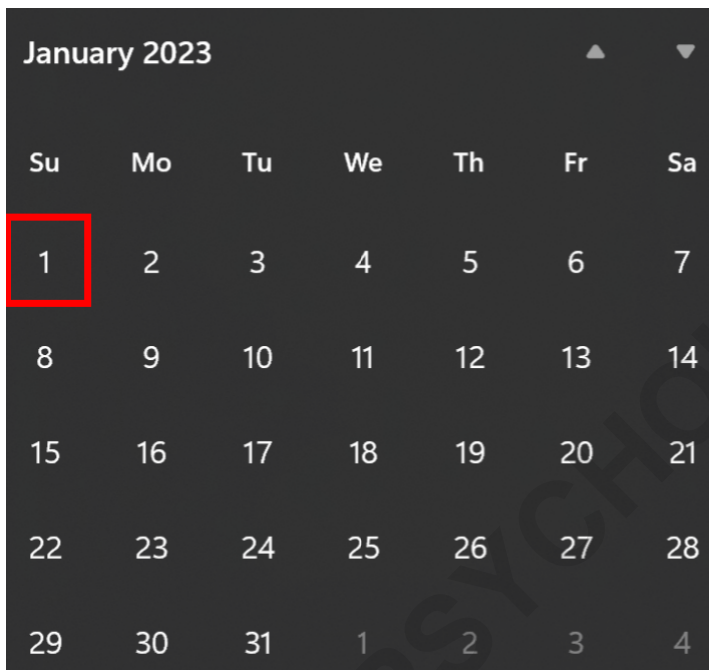
	A	B	C	D	E	F
1	Date	First Sunday				
2	1/1/2023	1/1/2023				
3	2/1/2023	2/5/2023				
4	3/1/2023	3/5/2023				
5	4/1/2023	4/2/2023				
6	5/1/2023	5/7/2023				
7	6/1/2023	6/4/2023				
8	7/1/2023	7/2/2023				
9	8/1/2023	8/6/2023				
10	9/1/2023	9/3/2023				
11	10/1/2023	10/1/2023				
12	11/1/2023	11/5/2023				
13	12/1/2023	12/3/2023				
14						
15						
16						

As depicted in the resulting table, Column B now consistently contains the date of the first Sunday of each month corresponding to the input dates found in Column A. This demonstrates the efficiency and scalability of using a well-constructed mathematical **formula** rather than relying on manual lookups or intricate conditional formatting.

Verifying the Results with a Calendar

While the calculation provides the mathematically correct answer, it is always prudent to perform external verification, especially when dealing with complex date logic. We can confirm that each date returned in column B indeed represents the first Sunday of the month referenced in column A.

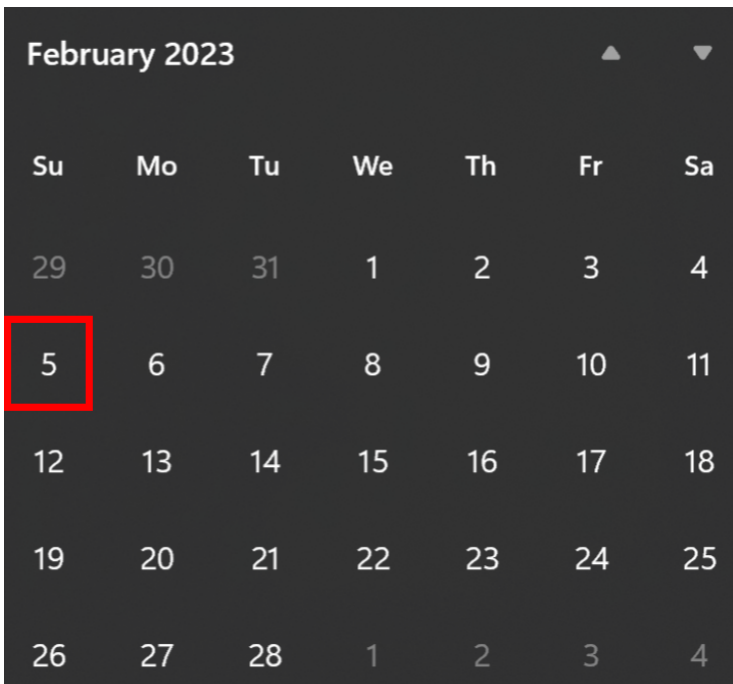
For the first entry, the source date is **1/1/2023**, which corresponds to the month of January 2023. Our formula returns the date 1/1/2023 itself. This is confirmed by referring to a January 2023 calendar, which clearly shows that January 1st fell on a Sunday:



A calendar for January 2023. The days of the week are listed at the top: Su, Mo, Tu, We, Th, Fr, Sa. The dates are arranged in a grid. The date 1 is highlighted with a red box, indicating it is a Sunday.

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	1	2	3	4

Moving to the next entry, the source date is **2/1/2023**, which corresponds to the month of February 2023. The calculated result in Column B is **2/5/2023**. Since February 1st was a Wednesday, we expect the first Sunday to fall several days later. Consulting a calendar for February 2023 confirms that the 5th of the month was indeed the first Sunday:



The image shows a calendar for February 2023. The days of the week are listed at the top: Su, Mo, Tu, We, Th, Fr, Sa. The dates are arranged in a grid. The date 5 is highlighted with a red box, indicating it is the first Sunday of the month.

Su	Mo	Tu	We	Th	Fr	Sa
29	30	31	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	1	2	3	4

This process of verification confirms the absolute accuracy of the derived date. We can be confident that every date populated in column B correctly represents the first Sunday of the month for each corresponding date in column A, thus completing the required **date calculation** task effectively within the **Excel** environment.

Expanding Your Excel Date Capabilities

Mastering the technique for finding the first Sunday is a gateway to performing far more intricate calendar calculations in **Excel**. The underlying principles--using arithmetic to manipulate date serial numbers and leveraging the WEEKDAY function for offset adjustments--can be adapted to solve a wide variety of date problems.

For instance, if you wished to find the third Tuesday of the month, you would modify the numerical offset used in the calculation. If Sunday has a WEEKDAY value of 1, and Tuesday has a value of 3, you would adjust the formula to target that specific day number and ensure the offset lands in the correct week (i.e., the third week instead of the first). Such adaptations make **Excel** an incredibly flexible tool for complex scheduling tasks where fixed calendar rules must be dynamically applied to large datasets.

By understanding how to combine ``DATE``, ``DAY``, and ``WEEKDAY``, you gain the ability to tackle challenges like calculating workdays, determining quarter start and end dates, or implementing custom fiscal calendars. We encourage users to explore related tutorials to further enhance their proficiency in advanced date and time management within the **Excel** application.

The following tutorials explain how to perform other common operations in Excel, building upon the foundational knowledge of date arithmetic established here:

ARABPSYCHOLOGY.COM