

Get First Day of Month in Google Sheets

Authored by
Mohammed looti

November 15, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Get First Day of Month in Google Sheets*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=1875>

For analysts and data professionals working with time-series data, standardizing dates to the beginning of their respective months is a frequent requirement. Whether you are generating monthly reports or grouping data aggregates, knowing how to quickly determine the first day of the month is essential in [Google Sheets](#). Fortunately, there are two highly effective formula-based approaches to achieve this conversion seamlessly.

The following two formulas represent the quickest and most reliable ways to calculate the first day of the month based on a given date input:

Formula 1: Direct Calculation using the [DAY\(\) Function](#)

=A2-DAY(A2)+1

Formula 2: Date Offset utilizing the [EOMONTH\(\) Function](#)

=EOMONTH(A2,-1)+1

Both of these powerful formulas will yield the correct first day of the current month for the date provided in cell **A2**. The choice between them often depends on personal preference or the need for compatibility with other spreadsheet systems.

Understanding Date Systems: The Basis for Calculation

Before diving into the mechanics of the formulas, it is crucial to understand how spreadsheet applications like [Google Sheets](#) handle dates internally. Unlike simple text strings, dates are stored as numerical values known as a [serial date system](#). This system assigns an integer to every day, typically starting from December 30, 1899. Therefore, when we perform calculations like subtraction or addition on a date, we are manipulating these underlying integers, which represent the number of days elapsed since the starting date. This fundamental concept is essential for successfully manipulating date data types in any spreadsheet environment.

For instance, if a cell contains the date 2023-10-15, [Google Sheets](#) recognizes this as a large integer (e.g., 45214). Subtracting 1 from this date effectively means subtracting one day from its serial value, resulting in 2023-10-14 (45213). This reliance on the [serial date system](#) is what allows the formulas below to work reliably across different months, ensuring accuracy regardless of whether the month has 28, 29, 30, or 31 days. Understanding this numerical basis demystifies date calculations and aids in troubleshooting common formatting issues.

Preparation and Dataset Overview

To demonstrate the practical application of both date calculation methods, we will utilize a cohesive

sample dataset. This dataset, clearly displayed in the visual below, contains various chronological dates located in column A. Our objective is to calculate the corresponding first day of the month for each entry and place the resulting standardized dates into column C.

Carefully reviewing this setup ensures you understand the context for the cell references used throughout the examples (specifically starting with **A2**, which holds our first date input). We will perform the initial calculation in **C2** and then efficiently drag the formula down to process the remaining rows in the dataset.

	A	B	C	D
1	Date	Sales		
2	1/4/2022	22		
3	2/15/2022	31		
4	3/23/2022	14		
5	4/15/2022	15		
6	4/18/2022	15		
7	5/1/2022	17		
8	8/7/2022	22		
9	10/31/2022	25		
10	11/4/2022	24		
11	12/19/2022	40		
12				
13				
14				
15				
16				
17				
18				

Method 1: Leveraging Subtraction with the DAY() Function

The first approach is perhaps the most fundamental and relies on standard arithmetic operations combined with the [DAY\(\) function](#). This method works by neutralizing the current day component of the date and then advancing the date by one day to reach the start of the required month. It is highly intuitive for users who are comfortable with how dates are represented numerically in spreadsheets.

To implement this technique, navigate to the first output cell, **C2**, and accurately input the following

formula, ensuring it references the initial date in cell **A2**. This specific construction resets the date by subtracting the current day number and then adding one day back:

=A2-DAY(A2)+1

Once the formula is correctly entered in **C2**, you can efficiently apply it to the remainder of the dataset by utilizing the fill handle. Dragging this handle down column C ensures that all relative references (such as **A2** changing to **A3**, **A4**, and so on) are correctly updated for every subsequent row, automating the standardization process for the entire column A.

C2 ▾ fx =A2-DAY(A2)+1				
	A	B	C	D
1	Date	Sales	First Day of Month	
2	1/4/2022	22	1/1/2022	
3	2/15/2022	31	2/1/2022	
4	3/23/2022	14	3/1/2022	
5	4/15/2022	15	4/1/2022	
6	4/18/2022	15	4/1/2022	
7	5/1/2022	17	5/1/2022	
8	8/7/2022	22	8/1/2022	
9	10/31/2022	25	10/1/2022	
10	11/4/2022	24	11/1/2022	
11	12/19/2022	40	12/1/2022	
12				
13				
14				
15				
16				
17				

The resulting Column C clearly shows the accurate first day of the month corresponding to each original date in Column A. This method's success hinges on a solid grasp of its three operational components:

The **DAY() function** extracts the specific day of the month (a numerical value between 1 and 31) from the date provided in **A2**. For example, if **A2** holds October 15, 2023, **DAY(A2)** returns 15.

The crucial subtraction **A2 - DAY(A2)** subtracts the current day number from the full date's [serial date system](#) value. This calculation rolls the date back precisely to the last day of the preceding

month, regardless of the month's length.

Finally, adding **+1** increments the date value by exactly one day, moving the result forward from the last day of the previous month to land precisely on the **first day of the current month**, thereby achieving the intended date standardization.

Method 2: Utilizing the EOMONTH() Function for Efficiency

The second highly reliable method utilizes the specialized, built-in **EOMONTH() function** (End of Month). This function is specifically designed to handle date offsets and month-end calculations, providing a structure that is often favored by advanced users for its clarity and intended purpose. While the **DAY()** method uses generic arithmetic, **EOMONTH()** is a dedicated financial and date function.

The syntax of the **EOMONTH()** function requires two parameters: the starting date and an integer representing the number of months to offset. To find the first day of the **current** month, we must first determine the last day of the **previous** month. We achieve this by setting the offset argument to **-1**. We then add 1 to the result to move forward to the beginning of the desired month.

Enter the following formula into cell **C2**. This calculation is notably concise and effective, using the date offset mechanism:

=EOMONTH(A2,-1)+1

Once the formula is established in **C2**, proceed to drag and fill it down the entirety of column C. This quick step replicates the logic across all dates in column A, confirming that this specialized function achieves the identical, accurate results as the **DAY()** subtraction method.

C2 fx =EOMONTH(A2, -1)+1

	A	B	C	D
1	Date	Sales	First Day of Month	
2	1/4/2022	22	1/1/2022	
3	2/15/2022	31	2/1/2022	
4	3/23/2022	14	3/1/2022	
5	4/15/2022	15	4/1/2022	
6	4/18/2022	15	4/1/2022	
7	5/1/2022	17	5/1/2022	
8	8/7/2022	22	8/1/2022	
9	10/31/2022	25	10/1/2022	
10	11/4/2022	24	11/1/2022	
11	12/19/2022	40	12/1/2022	
12				
13				
14				
15				
16				
17				
18				

The output in Column C confirms the successful application, showing the standardized start date for each entry. The operational logic of the **EOMONTH()** method is structured as follows:

The core **EOMONTH() function** executes two tasks: navigating to the specified offset month and returning the last day of that month.

The expression **EOMONTH(A2, -1)** instructs [Google Sheets](#) to move backward one month (due to the -1 offset) relative to the start date in **A2**, and return the numerical date value corresponding to the last day of that preceding month.

The final component, adding **+1**, ensures the date is incremented by a single day, successfully transitioning the calculated date from the last day of the previous month to the precise **first day of the current month**.

Choosing the Best Formula and Troubleshooting

When deciding between the **DAY()** and **EOMONTH()** methods, practitioners often weigh computational overhead against semantic clarity. The **DAY()** method is mathematically simpler and relies purely on fundamental arithmetic, which might offer marginal speed benefits in extremely massive datasets. However, the **EOMONTH()** method is generally preferred for its clear intent,

making the spreadsheet easier to audit and maintain, especially when dealing with complex time-based calculations involving financial reporting periods.

A frequent troubleshooting step is addressing output formatting. After applying either formula, the result might appear as a large number (e.g., 45199) rather than a recognizable date (e.g., 10/1/2023). This happens because the calculation operates on the underlying [serial date system](#) value. To correct this, simply select the output cells (Column C), navigate to the Format menu, select "Number," and choose the desired date format (e.g., Date or a Custom date and time format). This step ensures the numerical result is interpreted and displayed correctly as a date.

Mastering these date manipulation techniques is critical for accurate time-series analysis and reporting in [Google Sheets](#). Furthermore, the EOMONTH function's flexibility extends far beyond finding the current month's start; it allows for easily calculating dates several months in the future or past by simply adjusting the offset integer, proving its value as a comprehensive temporal tool.

Additional Resources for Google Sheets Mastery

For those seeking to deepen their knowledge of date and time manipulation in spreadsheets, continuous reference to official documentation is advised. The concepts explored here build the foundation for complex temporal analysis.

The following tutorials explain how to perform other common and essential operations within the Google Sheets environment, helping you transition from basic calculations to advanced data handling: