Learning to Calculate Averages Between Dates in Google Sheets Using AVERAGEIFS

Authored by

Mohammed looti

February 11, 2026

Analyzing large datasets often requires the ability to calculate summaries based on very specific restrictions. One of the most common requirements in business intelligence and financial modeling is determining an average value only for data points that occurred within a defined time frame. In Google Sheets, this complex task is simplified by leveraging the robust **AVERAGEIFS function**. This comprehensive guide will walk you through the precise mechanisms of setting up conditional averages based on date boundaries, ensuring you can accurately and efficiently analyze your time-sensitive data for crucial insights.

The calculation of time-bound averages is vital across numerous fields, from tracking monthly sales quotas to monitoring project completion rates between specific milestones. Simply averaging an entire column of data can obscure meaningful trends or anomalies that are tied to particular periods. Therefore, mastering the ability to filter data using both a start date and an end date simultaneously transforms raw information into actionable business intelligence, providing a clear window into performance during critical intervals.

Understanding Conditional Averaging in Google Sheets

Data refinement is a prerequisite for meaningful aggregation. When dealing with spreadsheets containing chronological information, analysts frequently encounter the need to isolate and analyze metrics that fall within precise, user-defined timeframes. This practice is formally known as conditional averaging. By applying logical conditions to your dataset before performing the calculation, you ensure that the resulting average is highly relevant, targeted, and free from the noise of irrelevant data points outside your focus period.

While simpler functions like AVERAGEIF exist, they are limited to handling only a single logical criterion. However, calculating an average that must adhere to both a minimum date (start date) and a maximum date (end date) necessitates the use of the **AVERAGEIFS function**. This powerful formula is specifically designed to accommodate multiple criteria simultaneously, making it perfectly suited for defining complex filters like date ranges. It is the definitive tool for precision in date-based analysis.

The methodology involves defining a lower boundary (the inclusive starting date) and an upper boundary (the inclusive ending date) for your date column. By enforcing these two strict rules within a single formula, you guarantee that only the data points positioned strictly within your desired interval contribute to the final averaged result. This level of precision is indispensable for rigorous applications such as financial auditing, detailed sales performance reporting, rigorous inventory management, and academic research where the temporal dimension of data is paramount.

The AVERAGEIFS Function: Structure and Argument Breakdown

The **AVERAGEIFS function** in Google Sheets is engineered to calculate the arithmetic mean of cells based on one or more conditions being met. Understanding its underlying syntax is the first step toward effective implementation. The standard structure is highly modular and flexible, allowing for virtually unlimited criteria pairs:

```
=AVERAGEIFS(average_range, criteria_range1, criterion1, )
```

To successfully apply this structure to date-based conditional averaging, we must break down the purpose of each critical component. The first argument, `average_range`, specifies the numerical range of cells that contains the values you intend to average (e.g., sales figures or scores). The subsequent arguments work in pairs. The first pair, `criteria_range1`, refers to the column of dates that will be evaluated against the first condition, and `criterion1` is the logical test applied to that date range, typically defining the lower date boundary.

Crucially, achieving the "between two dates" requirement necessitates using the optional additional criteria pairs. Specifically, you will use a second set of `criteria_range` (which will be the same date column as the first) and `criterion` (which defines the upper date boundary). By simultaneously requiring the dates to be greater than or equal to the start date AND less than or equal to the end date, the function effectively creates a closed interval, ensuring data integrity within the desired window. Using comparison operators (like `>=` and `<=`) within quotation marks is essential for the function to properly interpret the conditions.

Consider a fundamental formula designed to calculate an average only when values correspond to dates that fall within a precise window:

=AVERAGEIFS(B2:B11, A2:A11, "<=1/15/2022", A2:A11, ">=1/5/2022")

This construction is specifically designed to calculate the average of values located in the designated numerical range **B2:B11**. This calculation is entirely conditional: it strictly mandates that the corresponding dates in the date range **A2:A11** must satisfy two distinct criteria. First, the date must be on or before **January 15, 2022** (the upper limit), AND concurrently, the date must be on or after **January 5, 2022** (the lower limit). This dual filtering ensures absolute temporal accuracy, isolating only the entries relevant to the specified period.

Step-by-Step Example: Calculating Average Sales Between Dates

To fully grasp the practical application of the **AVERAGEIFS function**, let us walk through a typical business scenario. Imagine you are tasked with analyzing the daily sales performance for a company, but management is only interested in performance during a key promotional window.

Your spreadsheet contains dates in Column A and corresponding daily sales figures in Column B. The goal is to isolate and average the sales data for this specific, critical period.

Below is a representation of the initial dataset, which spans several weeks of operation. The dates are clearly organized in the first column, while the numerical data (sales amounts) resides in the second column. This arrangement is standard for time-series data analysis and perfectly sets up the application of our conditional function.

	A	B	C	D
1	**Day**	**Sales**		
2	1/1/2022	10		
3	1/3/2022	4		
4	1/5/2022	7		
5	1/6/2022	7		
6	1/8/2022	8		
7	1/14/2022	6		
8	1/15/2022	8		
9	1/29/2022	12		
10	2/3/2022	10		
11	2/7/2022	5		
12				
13				
14				
15				
16				
17				
18				
19				
20				

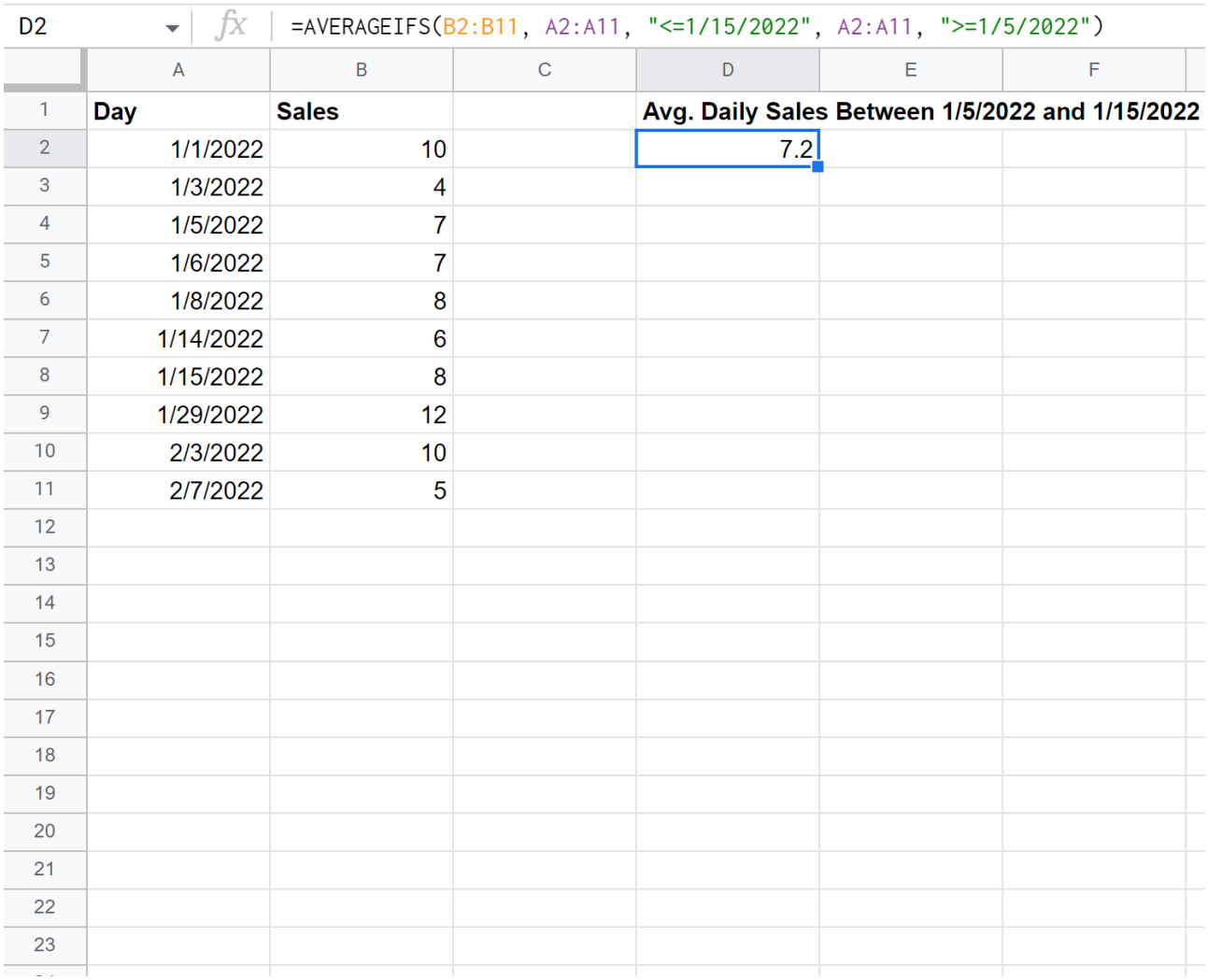
For this exercise, we define our specific analytical goal: determine the average daily sales exclusively for the period starting on **January 5, 2022**, and concluding on **January 15, 2022**. This objective requires two distinct conditions to be met simultaneously. We need the function to look at the date column (A2:A11) twice--once to check the lower bound, and once to check the upper bound--before averaging the sales column (B2:B11).

To accurately compute the average daily sales for this defined temporal window, we construct the **AVERAGEIFS function**. The formula specifies the average range (B2:B11), the first criterion

range (A2:A11) paired with the "greater than or equal to January 5" criterion, and the second criterion range (A2:A11) paired with the "less than or equal to January 15" criterion. This careful arrangement ensures that only data points that satisfy both temporal requirements contribute to the final calculation, providing the exact metric needed for the promotional window analysis.

=AVERAGEIFS(B2:B11, A2:A11, "<=1/15/2022", A2:A11, ">=1/5/2022")

The screenshot below visually demonstrates the successful implementation of this formula within the Google Sheets interface. Note how the function is entered into a target cell (G2 in this illustration), and how the various components--the average ranges and the specific criteria--are meticulously laid out. This visual confirmation is crucial for troubleshooting and validating the correct application of the function's arguments to yield the desired conditional average.



	A	B	C	D	E	F
1	**Day**	**Sales**		**Avg. Daily Sales Between 1/5/2022 and 1/15/2022**		
2	1/1/2022	10		7.2		
3	1/3/2022	4				
4	1/5/2022	7				
5	1/6/2022	7				
6	1/8/2022	8				
7	1/14/2022	6				
8	1/15/2022	8				
9	1/29/2022	12				
10	2/3/2022	10				
11	2/7/2022	5				
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						

Upon execution, Google Sheets calculates and returns a final averaged daily sales figure of **7.2** for the specified date range. This precise result offers clear, unambiguous insight into the average

performance of the sales team or product line during that particular, targeted period, differentiating it from the overall average sales figures.

Manual Verification of the Calculated Average

While the **AVERAGEIFS function** is highly reliable, performing a manual check is an excellent practice for ensuring data integrity and confirming the accuracy of complex formulas. This step serves as a validation mechanism, guaranteeing that the function correctly interpreted both the inclusion and exclusion criteria based on the dates specified (January 5, 2022, to January 15, 2022).

To verify the result, we first need to systematically identify all sales figures that fall strictly within our defined date range. Looking at the dataset, we include the following sales entries and their corresponding dates: January 5 (7), January 8 (7), January 10 (8), January 13 (6), and January 15 (8). Note that the entries for January 1 and January 20 are correctly excluded because they fall outside the specified boundaries.

The next steps involve summing these identified values and dividing the total by the count of included entries. The calculation proceeds as follows: We sum the values (7 + 7 + 8 + 6 + 8), which totals 36. Since exactly five entries met the dual date criteria, we divide the sum by 5.

Average Daily Sales = (7 + 7 + 8 + 6 + 8) / 5 = **36 / 5 = 7.2**.

This manual confirmation perfectly matches the result generated by the **AVERAGEIFS function**, proving that the formula accurately selected and averaged the data points within the specified temporal criteria. This dual validation provides confidence in the reliability of the spreadsheet output for reporting purposes.

Important Considerations and Best Practices for Date Criteria

When constructing date-based conditional averages in Google Sheets, adherence to certain best practices is necessary to avoid common errors and maximize formula efficiency. The successful execution of the **AVERAGEIFS function** heavily relies on the quality and format of your input data, as well as a precise understanding of comparison operators.

Consistent Date Formats: It is absolutely essential that all dates within your criteria range are entered and recognized consistently as date values by Google Sheets. Inconsistent date formats--such as mixing text dates with true date serial numbers--will cause the conditional function to fail or produce inaccurate results. Always verify that your date column is properly formatted and treated as numerical date values by the spreadsheet application.
Mastering Criteria Operators: The choice between inclusive (`<=`, `>=`) and exclusive (`<`, `>`)

operators dictates whether the start and end dates themselves are included in the average. Using "<=" (less than or equal to) and ">=" (greater than or equal to) is the standard method for establishing an inclusive date range where data on the boundary dates is counted. For scenarios requiring exclusion of the boundary dates, the simple less than and greater than operators must be used.

Referencing Dates from Cells for Dynamic Formulas: Hardcoding specific dates directly into the formula (e.g., "1/15/2022") makes the formula rigid and time-consuming to update. A far superior and professional practice is to reference the start and end dates from dedicated input cells (e.g., D1 for the start date and D2 for the end date). When referencing cells, the criterion must be concatenated using the ampersand (`&`) operator, yielding criteria like `"<="&D2` and `">="&D1`. This makes your reports dynamic and instantaneously adaptable to new timeframes.

Handling Data Quality Issues: Analysts must be vigilant about potential data cleanliness issues. Be mindful of cells within your criteria range that might be left empty or contain non-date text, as well as non-numeric entries in your average range. Generally, the **AVERAGEIFS function** is resilient--it ignores non-numeric data in the average range and rows where the criteria are not met. However, ensuring clean data prevents unexpected errors and maintains the integrity of your calculations.

For comprehensive details regarding all arguments, error handling, and advanced usage scenarios of the **AVERAGEIFS function**, always consult the official Google Sheets documentation.

Expanding Your Google Sheets Proficiency

The ability to harness conditional functions like **AVERAGEIFS** represents a significant professional milestone in advanced data analysis within Google Sheets. This skill allows for targeted reporting and highly accurate metric generation. To continue building upon this foundation and further enhance your overall spreadsheet proficiency, consider exploring other powerful related functions and techniques that use similar conditional logic:

SUMIFS Function: This function operates on the same multi-criteria logic as AVERAGEIFS but is designed to sum numerical values based on multiple conditions. Mastering this allows you to calculate total revenue, total costs, or total hours for a specific date range.

COUNTIFS Function: Learn how to count the number of cells or records that simultaneously meet multiple conditions. This is invaluable for quickly determining the total number of transactions or unique events that occurred within a specified date window.

QUERY Function: For highly complex data manipulation, filtering, and summarization tasks, the QUERY function provides SQL-like capabilities directly within your spreadsheet. It offers a single, flexible way to calculate conditional averages, sums, and counts simultaneously.

Conditional Formatting with Dates: Explore techniques to automatically apply visual formatting (colors, bolding) to cells based on date-related criteria. This makes tracking upcoming deadlines or

highlighting data outside a target period immediately apparent, significantly improving data visualization.

In-Depth Work with Date Formats and Functions: Deepen your knowledge of specialized date and time functions (like `DATEVALUE`, `TODAY`, and `EOMONTH`) and how to manage various date representations effectively. Consistent handling of date formats is key to reliable conditional analysis.

By integrating these advanced tools and methodologies, you can unlock the full potential of Google Sheets, transitioning from basic data entry to sophisticated, professional data analysis and reporting.