

How to Calculate Averages in Google Sheets Even When Cells Contain Text

Authored by
stats writer

November 28, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Calculate Averages in Google Sheets Even When Cells Contain Text*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=100778>

The Power of Conditional Averaging in Spreadsheets

Modern spreadsheet software, such as Google Sheets, offers powerful tools for sophisticated data analysis that goes far beyond simple arithmetic. One common requirement in data handling is the ability to calculate statistical metrics, like the average, based on specific conditions met in other columns. This capability is essential when raw data streams contain mixed information--for instance, numeric values coupled with descriptive text categories. While standard averaging functions typically fail or return errors when encountering text strings, conditional functions provide a robust solution, allowing analysts to accurately filter and summarize specific subsets of information.

The core challenge addressed by this method involves isolating numerical entries that correspond precisely to a specified textual label or pattern. Imagine a scenario where you have sales figures alongside product descriptions; calculating the average sales only for products containing the word "Premium" requires selective processing. Attempting to manually sort or filter large datasets is tedious and error-prone. Fortunately, Google Sheets provides the dedicated AVERAGEIF function, designed specifically to handle these conditional calculations efficiently and dynamically.

Mastering conditional averaging is a foundational skill for anyone working extensively with tabular data. It ensures that analyses are both precise and repeatable, minimizing the risk associated with human error during manual data manipulation. This guide will delve into the mechanisms of using the AVERAGEIF function to calculate the average of numerical cells only when an associated cell contains specific text, providing clarity on syntax, practical examples, and important technical considerations, such as the use of wildcard characters for pattern matching.

Understanding the AVERAGEIF Function in Google Sheets

The core tool for this conditional calculation is the AVERAGEIF function. Unlike the standard AVERAGE function, which calculates the mean of an entire numerical range, AVERAGEIF introduces a filtering mechanism. It first evaluates a designated range against a specified criterion and subsequently calculates the average of corresponding values in a separate, parallel range. This functionality is crucial for achieving targeted summaries within complex datasets, making it a cornerstone of efficient data analysis in environments like Google Sheets.

To calculate the average of numerical values contingent upon a cell containing specific text, the following structure is utilized. This example demonstrates a generalized format before diving into a detailed application:

```
=AVERAGEIF(A2:A13,"*text*",B2:B13)
```

In this structure, the formula performs a very specific task: it instructs the spreadsheet software to

look through the cells from **A2** through **A13**. For every cell in that range that satisfies the condition--in this case, containing the string "text"--the function retrieves the corresponding numerical value from the parallel range, **B2:B13**. Finally, it computes the arithmetic mean only of those retrieved numerical values. This powerful conditional logic eliminates the need for intermediate filtering steps, streamlining the analytic process considerably.

Syntax Breakdown: Mastering the Three Arguments

The AVERAGEIF function requires three distinct arguments to operate correctly. Understanding the precise role of each argument is essential for constructing flawless formulas tailored to complex data needs. The structure is universally defined as `AVERAGEIF(range, criterion, average_range)`.

The first argument, `range` (e.g., **A2:A13**), specifies the range of cells where the condition will be checked. This is typically the column containing the descriptive or categorical data--the text strings we are evaluating. This range must be the same size as the final averaging range, although it often contains non-numerical data.

The second argument, `criterion` (e.g., `"*text*"`), defines the condition that must be met in the corresponding cell within the `range`. When dealing with text strings, this criterion must be enclosed in quotation marks. For partial text matching, which is often required when searching for a substring within a larger cell entry (like finding "avs" within "Mavs"), we must employ specific tools known as wildcard characters.

The third argument, `average_range` (e.g., **B2:B13**), identifies the actual numerical values that will be averaged. This range should parallel the `range` argument in size and structure, ensuring that the numerical data aligns correctly with the criteria checked in the first range. The function only considers values from this range where the condition in the first range was met.

Proper sequencing and definition of these arguments ensure that the calculation remains accurate and that the filtered result truly represents the average of the desired subset of data points. Any misalignment or error in defining these three components will lead to incorrect calculations or formula errors, highlighting the importance of meticulous syntax review.

The Role of Wildcard Characters for Text Matching

When specifying the criterion for text containing a specific substring, standard exact match criteria are often insufficient. For instance, if you searched for exactly `"text"`, the formula would ignore cells containing `"Prefix text"` or `"text Suffix"`. This is where wildcard characters become indispensable within Google Sheets formulas. Wildcards allow us to define flexible search patterns rather than fixed strings.

The most commonly used wildcard for partial text matching is the asterisk (*). The asterisk acts as a placeholder for zero or more characters. When placed before and after a string, as seen in `"*text*"`, it tells Google Sheets to look for any cell that contains "text" anywhere within its content, regardless of what precedes or follows it. For example, `"*avs*"` successfully matches "Mavs," "Cavs," and even "Denver Avalanche."

There is also the question mark (?) wildcard, which acts as a placeholder for exactly one character. While less common for simple substring searches, it is useful for highly specific pattern matching, such as finding all four-letter words starting with 'B', where the criterion might be `"B???"`. Understanding both types of wildcards vastly enhances the flexibility and power of conditional functions like AVERAGEIF and COUNTIF when performing advanced data analysis involving textual criteria across large spreadsheet structures.

Note: The asterisks (*) are wildcard characters that instruct Google Sheets to ignore any preceding or succeeding text relative to the specified string. This ensures that the function correctly identifies matches even when the target text is embedded within a longer cell entry.

Practical Application: A Step-by-Step Example

To solidify the understanding of conditional averaging, let us apply the AVERAGEIF function to a realistic dataset. We will use a hypothetical roster detailing the performance of several basketball players, where the goal is to determine the average points scored specifically by players belonging to teams whose names contain a specific identifying substring. This process demonstrates how text criteria drive numerical aggregation.

Consider the following dataset, structured in two columns: Column A contains the team names, and Column B contains the corresponding points scored by individual players. This mixed data format is typical in real-world scenarios where descriptive categories must be linked to quantitative results.

	A	B	C	D	E
1	Team	Points			
2	Mavs	31			
3	Spurs	22			
4	Rockets	24			
5	Mavs	23			
6	Nets	28			
7	Mavs	21			
8	Warriors	19			
9	Rockets	23			
10	Magic	25			
11	Heat	16			
12	Heat	22			
13	Wizards	30			
14					
15					
16					
17					
18					

Our objective is to calculate the average points scored by players on any team whose name includes the letters "avs." We are not looking for an exact match to "avs," but rather for teams like "Mavs" or "Cavs." To achieve this partial match, we must employ the asterisk wildcard in our criterion. We specify the range of teams (A2:A13) as the condition range and the range of points (B2:B13) as the averaging range.

The complete formula entered into an empty cell (e.g., C2) is structured as follows, meticulously defining the criteria range, the conditional search string, and the final data range:

=AVERAGEIF(A2:A13,"*avs*",B2:B13)

Upon executing this formula in [Google Sheets](#), the software systematically evaluates each cell in A2:A13. If a cell contains "avs," the corresponding numerical value from B2:B13 is included in the averaging pool. This immediate calculation provides the desired summary statistic without requiring any manual data rearrangement.

Visualizing the Formula Execution

The resulting output confirms the efficacy of the conditional formula. The [spreadsheet](#) automatically identifies the qualifying entries (Mavs and Cavs) and extracts their associated scores

(31, 23, 21, and 16). The final numerical output represents the mean of these four scores.

D2		=AVERAGEIF(A2:A13, "*avs*", B2:B13)				
	A	B	C	D	E	F
1	Team	Points		Average points of teams that contain "avs"		
2	Mavs	31		22.75		
3	Spurs	22				
4	Rockets	24				
5	Mavs	23				
6	Nets	28				
7	Mavs	21				
8	Warriors	19				
9	Rockets	23				
10	Magic	25				
11	Cavs	16				
12	Heat	22				
13	Wizards	30				
14						
15						
16						
17						
18						
19						
20						
21						
22						

As depicted in the screenshot, the result of the calculation is **22.75**. This figure is instantly generated, demonstrating the speed and precision of using AVERAGEIF for complex filtering tasks. This efficiency is particularly valuable when working with datasets containing hundreds or thousands of rows, where manual filtering would be impractical and extremely time-consuming for repetitive data analysis tasks.

Manual Verification and Results Interpretation

To ensure full confidence in the formula's output, it is beneficial to manually verify the calculation using the identified data points. This step confirms that the conditional logic, driven by the wildcard characters, successfully isolated the correct subset of scores. In our dataset, the team names containing "avs" are "Mavs" and "Cavs."

By isolating the scores associated with these two teams, we find the following four data points in

column B:

Mavs Player 1: 31 Points

Mavs Player 2: 23 Points

Cavs Player 1: 21 Points

Cavs Player 2: 16 Points

The verification involves summing these four scores and dividing by the count (N=4). The manual calculation confirms the result derived by the conditional formula:

Average Points Scored = $(31 + 23 + 21 + 16) / 4 = 91 / 4 = 22.75$.

This exact match between the manual calculation and the formula output (**22.75**) validates the effective operation of the function using partial text criteria. This process confirms that the conditional structure is robust and correctly executed the requested filtering operation within the spreadsheet environment.

Advanced Considerations and Limitations of AVERAGEIF

While AVERAGEIF is powerful for single-condition averaging, users should be aware of its limitations, especially when dealing with increasingly complex data analysis requirements. The primary limitation is that AVERAGEIF can only evaluate one single criterion. If, for instance, you needed to calculate the average points for players on the "Mavs" AND who scored more than 25 points, AVERAGEIF alone would be insufficient.

For scenarios involving multiple criteria (e.g., averaging based on text in one column AND a numerical threshold in another column), users must transition to the more advanced function, **AVERAGEIFS**. This pluralized function allows for the inclusion of multiple range/criterion pairs, providing significantly greater flexibility in complex conditional summaries. AVERAGEIFS requires the average range to be listed first, differing slightly in syntax from AVERAGEIF, a key point to remember during implementation.

Furthermore, users must ensure data consistency. Text matching is case-insensitive in Google Sheets by default when using conditional functions, meaning "avs" matches "AVS" or "Avs." However, leading or trailing spaces in the data cells will prevent a match unless the criteria explicitly account for them. Maintaining clean source data is paramount to reliable results, regardless of the complexity of the conditional function employed in the spreadsheet.