

Google Sheets: Calculate Average If Cell Contains Text

Authored by
Mohammed loot

October 28, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Google Sheets: Calculate Average If Cell Contains Text*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=4808>

When analyzing large datasets in [Google Sheets](#), users frequently encounter the need to perform calculations based on specific, conditional criteria. A powerful, yet often required, capability is calculating an [average](#) exclusively for cells whose corresponding entries contain a particular [text string](#). This technique is fundamental for dynamic data filtering, allowing analysts to extract nuanced insights from complex spreadsheets without resorting to manual sorting or complex array formulas.

Fortunately, [Google Sheets](#) provides the streamlined [AVERAGEIF](#) function, which is precisely designed to handle single criteria conditional calculations. Mastering this function, particularly when combined with text matching techniques, significantly enhances the analytical power of your spreadsheets, making data interpretation far more efficient and accurate. The key to successful partial text matching lies in understanding how to leverage special characters within your criteria.

Implementing Conditional Averaging with AVERAGEIF

To calculate the average in [Google Sheets](#) based on a partial text match, you must structure the [AVERAGEIF](#) function correctly, ensuring the inclusion of appropriate [wildcard characters](#) within the criteria argument. This specific [formula](#) structure is engineered to look for the specified text anywhere within the cells of the criteria [range](#), rather than demanding an exact match.

The standard structure for achieving this powerful conditional calculation is demonstrated below. This example calculates the average of numerical values in the average [range](#), but only if the corresponding cell in the criteria [range](#) contains the specified text string, "text."

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

In this construction, the [range](#) A2:A13 serves as the criteria range--the column containing the text we need to evaluate. The criterion, "`*text*`", determines the condition for inclusion. Finally, the [range](#) B2:B13 is the average range, containing the numerical values that will be averaged if the condition in the corresponding row of A2:A13 is met. Understanding these three components is key to utilizing [AVERAGEIF](#) effectively.

The Critical Role of Wildcard Characters

The use of [wildcard characters](#) is essential when performing partial text matching in [Google Sheets](#). The asterisk (*) is the primary wildcard used here, representing any sequence of zero or more characters. By surrounding the target text with asterisks (e.g., "`*text*`"), we instruct the [formula](#) to locate the string "text" regardless of whether it appears at the beginning, middle, or end of the cell content.

Without these [wildcards](#), the [AVERAGEIF](#) function would only average the values if the cell content

was an exact, complete match for "text" and contained absolutely nothing else. This is rarely the desired outcome when attempting to filter data based on descriptive labels or categories where the target string is embedded within a longer description. The flexibility offered by wildcards transforms conditional averaging from a restrictive process into a highly versatile analytical tool.

In addition to the asterisk (*), which handles sequences of characters, [Google Sheets](#) also supports the question mark (?) as a wildcard. The question mark matches any single character. While the asterisk is typically used for partial string containment, the question mark is useful when filtering for patterns where only one character is variable, offering another layer of precision for defining complex criteria.

Practical Example: Averaging Basketball Scores

To solidify this concept, let us walk through a practical implementation using a sample [dataset](#). Imagine a scenario where you are tracking the scores of basketball players, and you need to calculate the average points scored, but only for players belonging to specific teams identified by a partial string within their team name. This is a classic use case for conditional averaging based on text.

Consider the following [dataset](#), spanning rows 2 through 13, which lists team names (Column A) and the corresponding points scored (Column B):

	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 contains the specific substring "avs". This requires applying the [AVERAGEIF](#) function, using the full range of team names ([A2:A13](#)) as the criteria range and the range of scores ([B2:B13](#)) as the average range.

The resulting [formula](#), designed to identify and average scores associated with the partial text "avs," is entered into a target cell, such as C2:

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

Interpreting and Validating the Output

Upon execution of the [formula](#), the result returned by [Google Sheets](#) is precisely **22.75**. This figure represents the calculated conditional [average](#) of the points scored, isolating only those entries that satisfied the wildcard criterion. The following screenshot visually confirms the implementation and the immediate output generated by the spreadsheet:

	A	B	C	D	E	F
D2				=AVERAGEIF(A2:A13, "*avs*", B2:B13)		
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						

To ensure the integrity of this analytical result, it is beneficial to manually verify the calculation against the source [dataset](#). In our example, the wildcard criterion `"*avs*"` successfully matched two distinct team names within the criteria range: "Mavs" and "Cavs." The scores corresponding to these teams were 31, 23, 21, and 16.

The manual validation process confirms the accuracy: the sum of the identified scores is $31 + 23 + 21 + 16 = 91$. Since there are four relevant entries, the calculation is 91 divided by 4, which yields **22.75**. This perfect match between the manual calculation and the [AVERAGEIF](#) output confirms the precision and reliability of using conditional functions combined with [wildcard characters](#) for sophisticated data filtering.

Best Practices for Conditional Averaging

The [AVERAGEIF](#) function, particularly when utilized with partial text matching, stands as an indispensable tool for data analysts working in [Google Sheets](#). This method allows users to rapidly segment and analyze numerical data based on categorical labels, even if those labels vary slightly

or contain extra descriptive text. Adhering to specific best practices ensures that your formulas are robust and error-free.

Crucially, always confirm that your criteria [range](#) (the cells being evaluated for text) and your average [range](#) (the cells containing the numbers to be averaged) are aligned both in terms of starting row and overall size. A mismatch in these ranges is the most common cause of errors when using conditional functions. Additionally, remember that text comparisons in [Google Sheets](#) are generally case-insensitive for functions like [AVERAGEIF](#), which simplifies the criteria definition process.

For scenarios requiring more intricate filtering--where you need to average data based on two or more simultaneous conditions (e.g., average scores for teams containing "avs" AND where the score is greater than 20)--you should transition from [AVERAGEIF](#) to the more robust [AVERAGEIFS](#) function. Mastering both single-criterion and multi-criterion conditional functions will significantly elevate your capability to perform complex data analysis and create sophisticated reports directly within your spreadsheets.

Additional Resources

To further expand your proficiency in [Google Sheets](#) and tackle other common data manipulation tasks, consider exploring the following tutorials and guides:

Learn how to perform advanced filtering techniques using [QUERY](#) or [FILTER](#) functions.

Discover methods for handling errors and inconsistencies in your data using functions like [IFERROR](#).

Explore other conditional functions like [SUMIF](#) and [COUNTIF](#) for various analytical needs beyond simple averaging.