

# How to Calculate Averages When Cells Contain Text: A Simple Guide

Authored by  
**stats writer**

December 1, 2025

## RECOMMENDED CITATION

stats writer (2025). *How to Calculate Averages When Cells Contain Text: A Simple Guide*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=102894>

When working with large data sets, analysts frequently encounter situations where cells intended for calculation contain descriptive text rather than pure numerical values. A fundamental rule in spreadsheet software, particularly in Excel, is that direct arithmetic functions like the AVERAGE function require exclusively numerical inputs. If a cell contains text--even if that text represents a quantitative concept (like "low," "medium," or "high")--it is automatically excluded or causes an error in standard average calculations, leading to inaccurate results or calculation failures.

To accurately calculate the average within a range that includes descriptive text, two primary strategies exist. The first involves complex data preprocessing, where you replace the text with an appropriate numerical proxy. For instance, if the cell contains the qualitative descriptor "low", you might assign it the numerical equivalent of 1, and if it contains "high" you might assign the value 5. This process requires creating a mapping scheme and using functions like VLOOKUP or nested IF statements to systematically convert the text entries into their numerical counterparts across the entire data range.

However, the most efficient and robust method, particularly when you only need to calculate the average of numerical values contingent upon specific text criteria being met in an accompanying column, is utilizing the powerful AVERAGEIF function. This function allows for conditional averaging, enabling the user to specify inclusion criteria based on the content of another cell. This capability eliminates the need for manual data transformation in many common analytical scenarios, saving considerable time and minimizing the risk of data entry errors associated with large-scale conversion tasks. It is essential to understand the distinction: while raw text cannot be averaged, it can be used as a criterion to filter which numerical values **are** included in the average calculation.

## Understanding Conditional Averaging with AVERAGEIF

The AVERAGEIF function is specifically designed to calculate the arithmetic mean of all cells in a range that meet a given criterion. Crucially, this criterion can be based on text located in a separate (or the same) conditional range. This is the cornerstone of solving the problem when you need to average a set of values, but only those values associated with a specific textual identifier, such as a category name, status, or group label. The formula allows for sophisticated filtering without altering the underlying data structure.

To calculate the average in Excel only for the cells that contain a specific text string, you must define three distinct arguments: the range containing the criteria (the text), the criterion itself (the specific text you are searching for), and the range containing the values to be averaged. This method ensures that only the relevant numerical data points are considered in the final computation, maintaining data integrity while providing the necessary analytical insight. When constructing the criterion, using quotation marks is mandatory, and incorporating special symbols

like asterisks is often necessary for flexible matching.

Consider the standard structure required for this operation. The flexibility of AVERAGEIF shines when dealing with partial matches, allowing you to search for a substring within a larger cell entry. This capability distinguishes it from simple comparisons that require exact matches. The following formula illustrates the foundational implementation structure, where the ranges and criteria are clearly delineated to isolate the calculation based on the specified text:

**=AVERAGEIF(A1:A13,"\*text\*",B1:B13)**

This particular formula is executed by first scanning the criteria range, defined here as **A1:A13**. It then identifies every cell within this range that contains the substring "text", regardless of any other characters present in that cell. For every cell in **A1:A13** that meets this text-based criterion, the corresponding numerical value in the average range, **B1:B13**, is selected. Finally, the Excel function calculates the average of these selected numerical values, excluding any numerical entries that did not have a matching criterion in the adjacent column.

**Note:** The asterisks (\*) used within the criteria argument are essential wildcard characters that instruct Excel to look for the specified string ("text" in this example) anywhere within the cell content--whether it appears before, after, or is surrounded by other characters. Without these wildcards, the function would only return a result if the cell in the criteria range contained **only** the word "text" and nothing else.

## Syntax Deep Dive: Deconstructing the AVERAGEIF Arguments

To master conditional averaging, a thorough understanding of the three required arguments within the AVERAGEIF function is mandatory. These arguments are structured as follows: AVERAGEIF(range, criteria, ). The first argument, range, specifies the cell range where the condition (the text string) must be checked. This is the search area. The second argument, criteria, defines the condition itself; this must be enclosed in quotation marks and often incorporates wildcard characters for flexible text matching, enabling the function to identify partial text strings or specific prefixes/suffixes.

The third argument, average\_range, is optional but is almost always required in scenarios where the text criteria and the numerical values reside in different columns, which is typically the case when calculating averages based on categorical labels. This argument specifies the actual range containing the numerical values that will be averaged. It is paramount that the range and the average\_range have the exact same dimensions and orientation (e.g., both must be 13 rows high) to ensure that Excel correctly pairs the text criterion with its corresponding numerical entry. If the average\_range argument is omitted, AVERAGEIF will use the range argument as the range to

average, which is only appropriate if the criteria range itself contains numerical values that need conditional averaging.

Furthermore, when constructing complex criteria involving comparisons (e.g., greater than 50), the operator must also be enclosed within the quotation marks, such as ">50". When linking a comparison operator to a cell reference or the result of another function, concatenation using the ampersand symbol (&) is necessary. For instance, to average values greater than the value in cell C1, the criterion would be written as ">"&C1. While our focus here is on text criteria, understanding the full range of criteria expression enhances the overall utility of this powerful conditional function.

## The Role of Wildcard Characters in Text Criteria

The effective use of wildcard characters--the asterisk (\*) and the question mark (?)--is fundamental to achieving flexibility when searching for text criteria within spreadsheet applications. The asterisk acts as a stand-in for any sequence of characters of any length, including zero characters. When placed around the search term, as in "\*mavs\*", it tells Excel to find cells that contain "mavs" somewhere within their content. Conversely, using "mavs\*" would find cells that begin with "mavs" (e.g., "mavs team", "mavs 2023"), while "\*mavs" would find cells that end with "mavs".

The question mark (?) serves a different purpose; it represents any single character. This is particularly useful when dealing with patterns where you know the exact length but have one or two variable characters. For example, the criterion "J?n" would match "Jan", "Jen", or "Jun". Combining these wildcards allows for highly precise yet flexible conditional matching, which is crucial when dealing with inconsistent data entry or complex categorical labels.

It is important to note the exception: if you specifically need to search for an actual asterisk or question mark character within the cell, you must escape the wildcard by preceding it with a tilde (~). For instance, the criterion "~\*" would search for the literal asterisk symbol, rather than treating it as a wildcard. Mastering these nuances of text manipulation ensures that your conditional average calculations are both accurate and exhaustive, capturing all necessary data points based on the specified textual criteria.

## Practical Implementation: Calculating Team Averages

We will now proceed with a practical demonstration to illustrate how the AVERAGEIF function handles real-world data analysis. Suppose we are analyzing a hypothetical dataset that shows the individual points scored by 12 different basketball players, alongside the team they play for. Our objective is to calculate the average points scored exclusively by the players associated with one specific team, which is identified by its text label in the adjacent column.

The following visual representation shows the structure of our dataset, where Column A contains

the team names (the criteria range) and Column B contains the points scored (the average range). This layout is typical for analytical tasks where numerical performance metrics must be segregated and analyzed based on non-numerical categorical identifiers.

	A	B	C	D	E
1	<b>Team</b>	<b>Points</b>			
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					

To calculate the average points scored solely by players on the "Mavs" team, we must establish the boundaries of our ranges and define the criterion. The criterion range will be the team column (A2:A13), and the average range will be the points column (B2:B13). The criterion itself must be formulated to match the text "mavs" regardless of any surrounding characters, ensuring flexibility if the cell contains "The Mavs" or "Mavs (Home)". We will use the following precisely constructed formula:

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

Upon entering this formula into a designated results cell, Excel processes the request by filtering the B2:B13 range against the A2:A13 range based on the textual match. The result provides an immediate analytical summary, isolating the performance metric specifically for the targeted subset of the data. The following screenshot visually confirms the implementation and the resulting calculated average:

	A	B	C	D	E	F
1	<b>Team</b>	<b>Points</b>		<b>Mavs Average Points</b>		
2	Mavs	31		25		
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						

## Verification and Case Sensitivity Considerations

As demonstrated in the final calculation, the conditional average points scored by players on the "Mavs" team is precisely **25**. It is always good practice in complex spreadsheet tasks to manually verify the result derived from the formula, particularly when dealing with conditional logic, to ensure that the ranges were correctly specified and the criterion was accurately matched. In our dataset, the Mavs players scored 31, 23, and 21 points, respectively.

We can manually verify the calculation process by summing the points scored by the Mavs players and dividing by the count of those players: Average Points Scored =  $(31 + 23 + 21) / 3 = 75 / 3 = \mathbf{25}$ . This manual verification confirms the accuracy and effectiveness of the AVERAGEIF function in filtering numerical values based on a textual condition in an accompanying column.

A crucial characteristic of the AVERAGEIF function, shared by most standard Excel conditional functions (like COUNTIF and SUMIF), is that it is inherently case-insensitive. This means that the function treats uppercase and lowercase letters as identical when evaluating the criteria. For example, if we had used `"*MAVS*"` or `"*mavs*"` or `"*Mavs*"` or even mixed capitalization like `"*mAvS*"` in the criteria argument, Excel would have returned the exact same result (25). This feature generally simplifies data analysis, as it removes the need for preparatory text transformation to ensure case uniformity before applying the criteria.

## Advanced Conditional Averaging: Introducing AVERAGEIFS

While AVERAGEIF is ideal for a single criterion, real-world data analysis often requires filtering based on multiple, simultaneous conditions. For scenarios demanding two or more criteria--for example, calculating the average score for the "Mavs" team **only** for games played "Home"--you must utilize the more powerful **AVERAGEIFS** function. The syntax for AVERAGEIFS is distinct, placing the average\_range (the range of numerical values) as the first argument, followed by pairs of criteria range and criteria.

The structure for this multi-conditional function is: `AVERAGEIFS(average_range, criteria_range1, criteria1, , ...)`. This structure allows for virtually unlimited criteria pairs, providing immense flexibility for complex filtering tasks. For instance, to calculate the average points for Mavs players in games where the score was also greater than 20, the formula would look like `=AVERAGEIFS(B2:B13, A2:A13, "*mavs*", B2:B13, ">20")`. Note that the conditional range B2:B13 is used both as the average range (first argument) and as a criteria range (third criteria range argument).

Understanding when to transition from AVERAGEIF to AVERAGEIFS is key to efficient spreadsheet modeling. If you are ever required to use an 'AND' logic--where multiple conditions must simultaneously be met for a numerical value to be included in the average--AVERAGEIFS is the mandatory choice. It handles text-based criteria, numerical comparisons, and date-based criteria with equal efficacy, all while maintaining the strict adherence to the defined ranges and logical conditions.

## Alternative Data Transformation Techniques

While conditional functions like AVERAGEIF are perfect for filtering, there are scenarios where the text itself needs to be quantified **before** averaging. This is common when qualitative assessments (e.g., "Good," "Fair," "Poor") need to be converted into a measurable scale (e.g., 3, 2, 1). In such cases, one must use data transformation functions to create a new helper column containing the numerical values.

The most common tools for this conversion are nested **IF** functions or the **VLOOKUP** function paired with a lookup table. Using VLOOKUP is generally cleaner and more scalable: you create a separate table mapping "Good" to 3, "Fair" to 2, and so on. Then, in the helper column, you use VLOOKUP to pull the corresponding numerical score for every text entry in the original column. Once the helper column is populated entirely with numerical data, you can use the standard AVERAGE function directly on this new column.

This preprocessing step is required primarily when the text **is** the variable being averaged (after conversion) rather than merely a category used for filtering. For instance, if you want the average

quality rating based on a 1-5 scale derived from text descriptions, transformation is necessary. If you only want the average score of items **labeled** "Good," then AVERAGEIF remains the superior and more direct approach, as it avoids generating redundant data columns.

ARABPSYCHOLOGY.COM