How to Calculate Averages with Multiple Criteria in Google Sheets: A Simple AVERAGEIFS Guide

Authored by

stats writer

November 30, 2025

The ability to calculate conditional statistics is fundamental to effective data analysis in spreadsheets. While the basic Google Sheets AVERAGE function calculates a simple mean across a range, analysts often need to restrict this calculation based on specific conditions. This is where the powerful function, AVERAGEIFS, becomes indispensable. It allows users to compute the average of a range of cells by applying constraints defined by one or more sets of criteria simultaneously. For instance, you might need to find the average sales figure only for products sold in a specific region and exceeding a certain price threshold. AVERAGEIFS handles these complex, multi-conditional requirements with ease and precision, significantly streamlining data analysis workflows. It is capable of managing up to 127 criteria ranges and their corresponding criteria, offering immense analytical depth.

Understanding the Structure of AVERAGEIFS

The **AVERAGEIFS** function is a key component of Google Sheets' conditional calculation suite, offering far greater flexibility than its single-criterion counterpart, AVERAGEIF. It is designed specifically to handle scenarios where multiple conditions must be met simultaneously for a data point to be included in the final calculation. The core purpose of the function is to determine the arithmetic mean of values within a specified range only when corresponding values in other ranges satisfy predefined conditions, employing an inherent AND logic.

The **AVERAGEIFS** function in Google Sheets can be used to find the average value in a range if the corresponding values in another range meet certain criteria.

This function uses the following critical syntax:

AVERAGEIFS(average_range, criteria_range1, criterion1,)

where the arguments are defined as follows:

average_range: The numerical range containing the values from which the average will be calculated. This must always be the first argument.
criteria_range1: The first range of cells that will be evaluated against a specific criterion. It must be dimensionally aligned with the **average_range**.
criterion1: The condition or pattern (which can be a number, text, or logical expression) applied to the first criteria range.

It is important to remember that all criteria ranges must be aligned dimensionally with the **average_range**. If the ranges are mismatched, the function will return an error. When using text or logical operators (like greater than or less than), the criterion must be enclosed in quotation marks.

Preparing the Dataset for Conditional Analysis

To effectively demonstrate the practical application of **AVERAGEIFS**, we utilize a sample dataset commonly found in statistical reporting, focused here on sports performance metrics. This dataset provides a clear structure featuring Team affiliation, Player Position, and Points scored, allowing us to build conditions based on both text and numerical data types. Understanding this raw data layout is the precursor to crafting accurate conditional formulas.

The dataset used throughout the following examples is presented below:

	A	B	C	D
1	**Team**	**Position**	**Points**	
2	Mavs	Guard	22	
3	Mavs	Guard	28	
4	Mavs	Forward	25	
5	Mavs	Forward	30	
6	Mavs	Center	18	
7	Spurs	Guard	13	
8	Spurs	Guard	19	
9	Spurs	Forward	22	
10	Spurs	Forward	30	
11	Spurs	Center	11	
12				
13				
14				
15				
16				

Columns A, B, and C define the key fields: Team, Position, and Points, respectively. Our conditional averaging will consistently target the numerical values in the Points column (C2:C11), filtering this range based on conditions set in columns A and B.

Example 1: Calculating Averages with Single Text Criteria

Our initial illustration focuses on the simplest application of AVERAGEIFS: filtering based on a single text criterion. We aim to isolate all players belonging to the "Mavs" team and calculate their average points scored. This exercise confirms that AVERAGEIFS can seamlessly replace AVERAGEIF while maintaining the capacity for expansion to multiple criteria.

The goal is to calculate the average score from the **Points** column (C2:C11) only for rows where

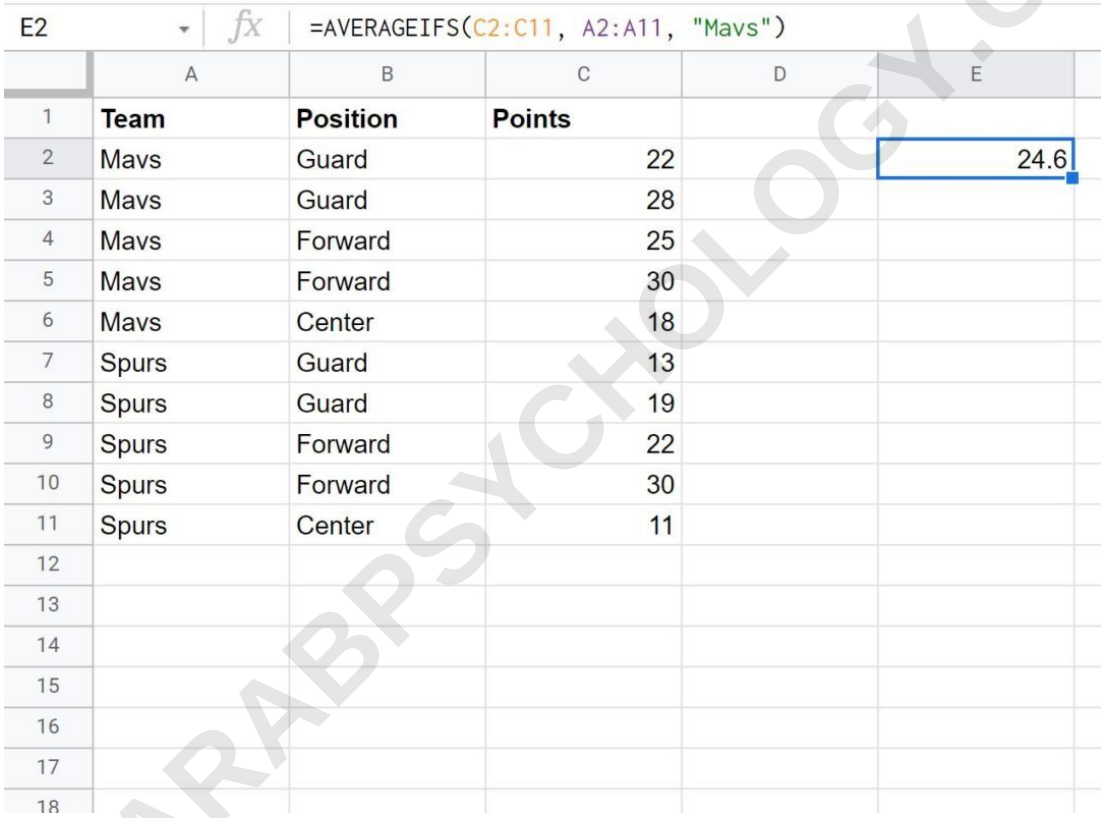
the corresponding **Team** column (A2:A11) contains the exact text match "Mavs".

We use the following formula to execute this single-criterion calculation:

=AVERAGEIFS(C2:C11, A2:A11, "Mavs")

In this formula, the criteria_range1 (A2:A11) is checked for criterion1 ("Mavs"). Only values from the average_range (C2:C11) that correspond to these successful matches are aggregated and averaged.

The following screenshot demonstrates the practical result of implementing this formula:



	A	B	C	D	E
1	**Team**	**Position**	**Points**		
2	Mavs	Guard	22		24.6
3	Mavs	Guard	28		
4	Mavs	Forward	25		
5	Mavs	Forward	30		
6	Mavs	Center	18		
7	Spurs	Guard	13		
8	Spurs	Guard	19		
9	Spurs	Forward	22		
10	Spurs	Forward	30		
11	Spurs	Center	11		
12					
13					
14					
15					
16					
17					
18					

The resulting average value in the **Points** column for players on the "Mavs" team is calculated as **24.6**. We confirm this result by manually summing the Mavs scores (22 + 28 + 25 + 30 + 18) and dividing by the count (5), yielding **24.6**.

Example 2: Applying Multiple Text Criteria for Precision

The essential utility of AVERAGEIFS becomes evident when multiple conditions must be satisfied. Here, we aim for a more granular analysis: calculating the average points only for players who are on the "Mavs" team AND who play the "Guard" position. This requires linking two separate criteria

ranges, each with its own criterion, using the built-in AND logic of the function.

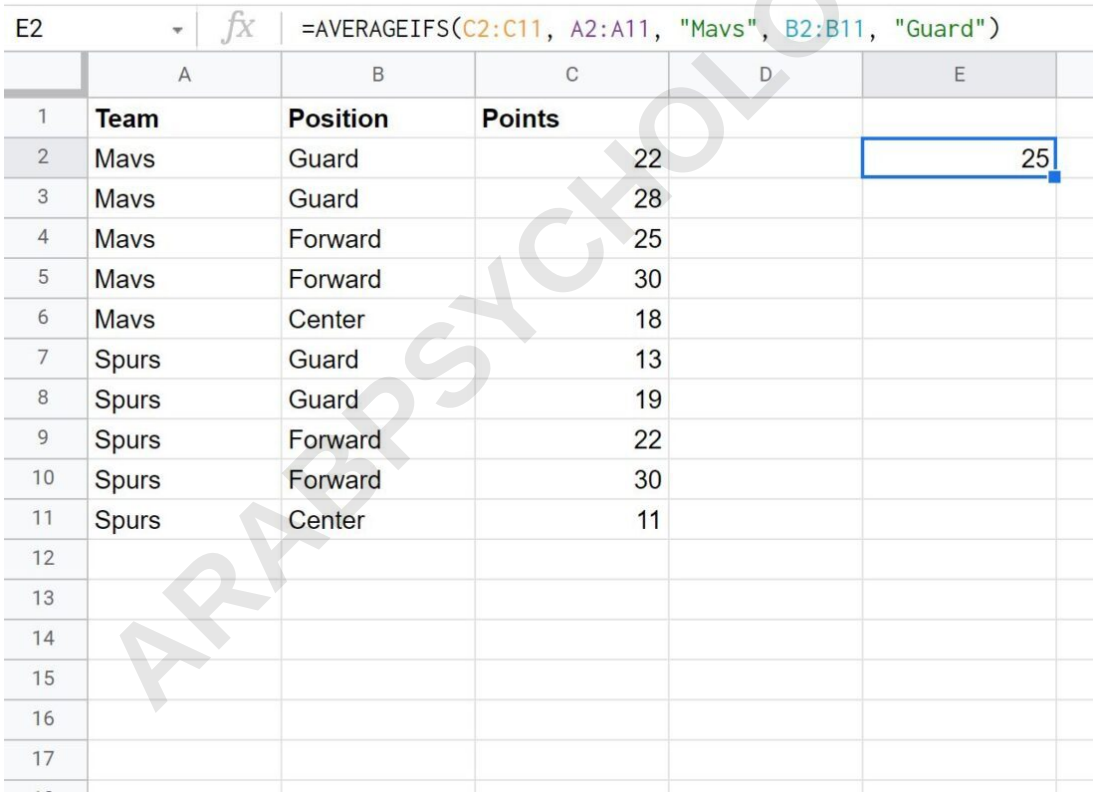
We need to specify the points range (C2:C11), the team range (A2:A11) with the "Mavs" criterion, and the position range (B2:B11) with the "Guard" criterion.

We can use the following formula to calculate the average value in the **Points** column where the **Team** column is equal to "Mavs" and the **Position** column is equal to "Guard":

=AVERAGEIFS(C2:C11, A2:A11, "Mavs", B2:B11, "Guard")

This formula ensures that only rows where both conditions (Team = Mavs AND Position = Guard) are simultaneously true contribute to the average calculation, resulting in a highly focused metric.

The following screenshot shows how to use this formula in practice, yielding a more selective result:



	A	B	C	D	E
1	**Team**	**Position**	**Points**		
2	Mavs	Guard	22		25
3	Mavs	Guard	28		
4	Mavs	Forward	25		
5	Mavs	Forward	30		
6	Mavs	Center	18		
7	Spurs	Guard	13		
8	Spurs	Guard	19		
9	Spurs	Forward	22		
10	Spurs	Forward	30		
11	Spurs	Center	11		
12					
13					
14					
15					
16					
17					

The calculated average value in the **Points** column for this highly refined subset of players is **25**. This is verified by identifying the two Mavs players who are Guards (22 and 28 points). Their average is (22 + 28) / 2 = **25**. This clearly demonstrates the restrictive power of utilizing multiple text-based criteria.

Example 3: Combining Text and Numeric Criteria

In many analytical tasks, conditions must combine textual categories with logical numeric constraints. Our third example focuses on calculating the average points for players on the "Spurs" team, but only including those who scored greater than 15 points. This involves using the Points column both as the average range and as a criteria range with a logical operator.

It is crucial to correctly format the numeric criterion using a logical operator. The entire expression (e.g., ">15") must be treated as a text string by enclosing it in quotation marks so that Google Sheets interprets it as a condition rather than a simple numeric value.

We can use the following formula to calculate the average value in the **Points** column where the **Team** column is equal to "Spurs" and the **Points** column is greater than 15:

=AVERAGEIFS(C2:C11, A2:A11, "Spurs", C2:C11, ">15")

The final criterion pair (C2:C11, ">15") ensures that only points greater than 15 are evaluated for inclusion, even if the team condition is met.

The following screenshot confirms the result of this combined logic:

E2 fx =AVERAGEIFS(C2:C11, A2:A11, "Spurs", C2:C11, ">15")

	A	B	C	D	E
1	**Team**	**Position**	**Points**		
2	Mavs	Guard	22		23.66666667
3	Mavs	Guard	28		
4	Mavs	Forward	25		
5	Mavs	Forward	30		
6	Mavs	Center	18		
7	Spurs	Guard	13		
8	Spurs	Guard	19		
9	Spurs	Forward	22		
10	Spurs	Forward	30		
11	Spurs	Center	11		
12					
13					
14					
15					
16					
17					

The average value in the **Points** column where the **Team** column is equal to "Spurs" and the **Points** column is greater than 15 is approximately **23.67**. This result is verified by identifying the Spurs players who meet the score threshold (19, 22, and 30 points). The average is (19 + 22 + 30) / 3 = **23.67**, demonstrating accurate conditional filtering based on mixed data types.

Using Wildcard Characters in Criteria

For scenarios requiring pattern matching instead of exact text matches, **AVERAGEIFS** supports the use of wildcard characters within the criteria arguments. The asterisk (*****) represents any sequence of zero or more characters, and the question mark (**?**) represents any single character. This capability is invaluable for working with inconsistent or partial data entries.

For example, if you wanted to average the points for all teams whose name starts with 'M', you would use the criterion **"M*"**. If you were searching for a position that had exactly five letters, you might use **"?????"**. When using wildcards, remember that the criteria must still be enclosed in quotation marks, treating the entire pattern as a text string for evaluation. This flexibility prevents the need for complex formulas involving search functions to achieve partial matches.

Handling Errors and Zero Results

While **AVERAGEIFS** is highly reliable, specific conditions can lead to errors that analysts must anticipate. The most common error is the **#DIV/0!** error. This error occurs in two primary situations: first, if the **average_range** contains no numeric values (only text or empty cells); and second, if no rows in the entire dataset satisfy all of the applied criteria, resulting in division by a count of zero.

To prevent abrupt errors in dynamic dashboards or complex reports, it is standard practice in Google Sheets to wrap the **AVERAGEIFS** function within the IFERROR function. This wrapper allows the user to specify a custom output, such as 0 or a user-friendly message, in place of the error message when no data meets the specified conditions. For example: `=IFERROR(AVERAGEIFS(...), 0)`. This practice enhances the robustness and readability of the spreadsheet results.

Conclusion: Mastering Advanced Conditional Calculations

The AVERAGEIFS function represents a fundamental building block for advanced analysis in Google Sheets. By allowing precise control over which data points are included in an average calculation based on multiple conditions, it moves analysis beyond simple summation to targeted, insightful reporting. Whether filtering based on categories, numerical limits, or mixed criteria, maintaining a strict understanding of the average_range position and the correct formatting of criteria ensures accurate and repeatable results. Mastery of this function is essential for creating robust, dynamic, and actionable spreadsheet models.

Note: You can find the complete online documentation for the **AVERAGEIFS** function here: Google Sheets Help Documentation.