Learning to Sum Values Based on Partial Text Matches in Google Sheets Using SUMIF and SUMIFS

Authored by

Mohammed looti

October 31, 2025

In the modern landscape of data analysis and management, the ability to accurately and flexibly summarize information is essential. Google Sheets provides a robust suite of functions designed to tackle complex summation challenges, particularly those involving textual criteria. This comprehensive guide details the effective implementation of the SUMIF and SUMIFS functions to calculate sums based on cells that **contain** specific text fragments, rather than relying solely on exact matches. This powerful capability is critical for cleaning, filtering, and aggregating data within large or dynamic datasets.

Real-world data often features inconsistent entries, or you may simply need to aggregate values based on partial pattern recognition. For example, a business analyst might need to sum total revenue for all product codes that include "XYZ" in their nomenclature, or a coach might want to total points for athletes whose position name contains "Wing." Mastering the "contains" logic using `SUMIF` and `SUMIFS` is a significant step toward advanced spreadsheet proficiency. We will systematically explore the two primary methods: applying partial matching for a single condition and extending this logic to handle multiple concurrent conditions, providing precise explanations and practical, verifiable examples for both scenarios.

Leveraging Wildcard Characters for Partial Matching

The foundation for performing "contains" operations within Google Sheets using functions like `SUMIF` and `SUMIFS` rests on the strategic application of wildcard characters. These unique characters serve as placeholders, allowing you to define patterns within your search criteria, thereby enabling partial text matches. The most vital wildcard for implementing "contains" logic is the asterisk (*****).

The asterisk (*****) is designated to represent any sequence of characters, encompassing zero or more characters. When you construct a string by placing an asterisk both before and after your search term--such as **"*text*"**--you are instructing Google Sheets to identify any cell that includes "text" at any position within its content. To illustrate, the criterion **"*apple*"** will successfully match entries like "Red apple pie", "Granny Smith applesauce", or simply "apple". This inherent flexibility is the key mechanism that permits `SUMIF` and `SUMIFS` to execute effective "contains" logic.

While the asterisk is indispensable for finding a string anywhere within a cell, it is worth noting the existence of the question mark (**?**) wildcard, which acts as a placeholder for any single character. For instance, the pattern **"te?t"** would match "test" or "text" but not "teeeest". However, for the general task of checking if a cell includes a specific substring, the asterisk remains the essential and primary tool. Understanding these wildcard characters is paramount for constructing dynamic and robust formulas capable of adapting to diverse and imperfect data inputs.

Method 1: Using SUMIF for a Single "Contains" Condition

The SUMIF function is specifically engineered to sum numeric values within a range that satisfy one single, specified criterion. When the requirement is to sum values corresponding to cells that **contain** a particular string, `SUMIF` offers the most direct and accessible solution. Its simple structure makes it a fundamental tool for nearly all Google Sheets users.

The standard syntax for the `SUMIF` function is defined as: `SUMIF(range, criterion, )`.

`range`: This specifies the range of cells that the function will evaluate against the specified criterion.

`criterion`: This is the condition or pattern that must be met by the cells within the `range`. For achieving a "contains" match, you must encapsulate the desired string using asterisks (e.g., **"*substring*"**).

: (Optional) This defines the actual range of cells whose values will be totaled. If this argument is omitted, the function defaults to summing the cells in the initial `range`.

To calculate the total sum of cells based on a single "contains" criterion, you should implement the following formula structure, ensuring that the wildcards are correctly positioned:

=SUMIF(A2:A11, "*string*", C2:C11)

This powerful yet simple formula instructs the spreadsheet to sum all values located in the range **C2:C11**. The crucial condition for inclusion is that the corresponding cell within the evaluation range **A2:A11** must include the specified partial string--represented by "string" in the example. The surrounding asterisks guarantee that "string" can be found anywhere within the cell's textual content, making the match flexible.

Method 2: Using SUMIFS for Multiple "Contains" Conditions

While SUMIF handles single conditions effectively, many analytical scenarios demand the summation of values based on two or more simultaneous criteria. For these complex requirements, the SUMIFS function is the indispensable tool in Google Sheets. `SUMIFS` allows you to first define the primary summation range and then apply multiple pairs of `criteria_range` and `criterion` that must all be satisfied for a value to be included.

The standardized syntax for `SUMIFS` is structured as: `SUMIFS(sum_range, criteria_range1, criterion1, )`.

`sum_range`: This is the definitive range of numeric values that will be aggregated. It is important to

remember that this is the first argument, which differs from the argument order in `SUMIF`.

`criteria_range1`: This specifies the first range of cells that the function will evaluate.

`criterion1`: This is the specific criterion applied to the `criteria_range1`. For any "contains" match, you must utilize wildcard characters (e.g., **"*string1*"**).

: (Optional) This section allows for the inclusion of additional ranges and their corresponding criteria, all of which must be concurrently true.

To execute a summation based on two or more simultaneous "contains" conditions, you must structure your formula to link each criteria range with its corresponding wildcard criterion, as shown below:

=SUMIFS(C2:C11, A2:A11, "*string1*", B2:B11, "*string2*")

This highly effective formula aggregates numerical values from the range **C2:C11** only if two strict conditions are met: the corresponding cell in range **A2:A11** must contain "string1," AND the corresponding cell in range **B2:B11** must contain "string2." The core principle of `SUMIFS` is that every specified criterion must evaluate to true for a row's value to be successfully included in the final aggregated sum.

Example 1: Practical Application of SUMIF Contains

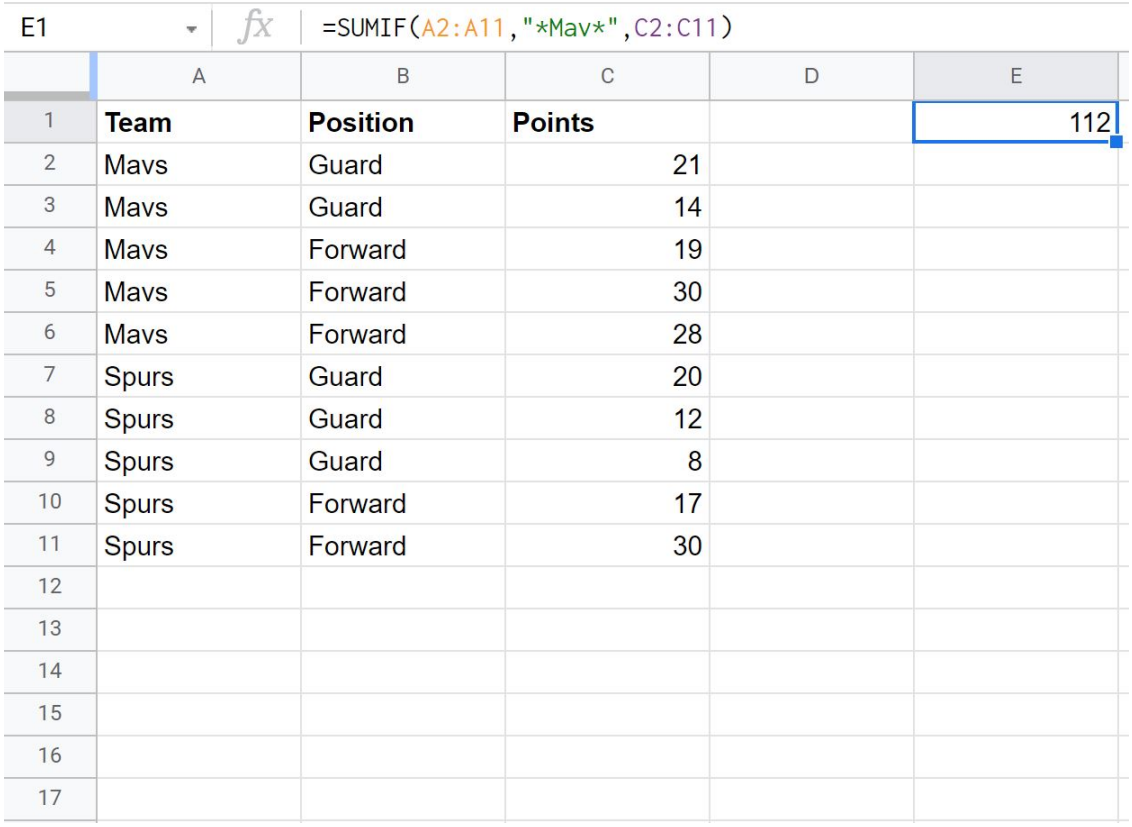
To demonstrate the utility of `SUMIF`, let us consider a typical data scenario. Suppose you are analyzing a dataset listing basketball teams and their corresponding points. Your specific goal is to calculate the total points accumulated for all rows where the team name includes a partial text match--in this instance, we aim to sum the points for any team name that contains the string "Mav".

Assuming the team names are situated in column A (the evaluation range) and the points are in column C (the summation range). To achieve this targeted summation, we must construct the following formula using the asterisk wildcard:

=SUMIF(A2:A11,"*Mav*",C2:C11)

Within this formula, **A2:A11** serves as the `range` where the partial match criterion is checked. The string **"*Mav*"** functions as the `criterion`, compelling Google Sheets to search for any cell in **A2:A11** that contains the substring "Mav" (e.g., "Mavericks" or "Mavs"). Lastly, **C2:C11** is designated as the `sum_range`; if a cell in **A2:A11** meets the specified `criterion`, its corresponding point value in **C2:C11** is added to the running total.

The following visual representation captures the implementation of this formula and its output within a Google Sheets environment:



	A	B	C	D	E
1	**Team**	**Position**	**Points**		112
2	Mavs	Guard	21		
3	Mavs	Guard	14		
4	Mavs	Forward	19		
5	Mavs	Forward	30		
6	Mavs	Forward	28		
7	Spurs	Guard	20		
8	Spurs	Guard	12		
9	Spurs	Guard	8		
10	Spurs	Forward	17		
11	Spurs	Forward	30		
12					
13					
14					
15					
16					
17					

As clearly illustrated by the resulting output, the calculation yields a total sum of **112** points for all rows where the team name contains "Mav". This result is accurately derived by isolating all matching team names and aggregating their respective point totals. We can confirm this by manually verifying the involved rows: 21 + 14 + 19 + 30 + 28 = **112**. This manual confirmation validates the correct and precise application of the `SUMIF` function using a "contains" criterion.

Example 2: Practical Application of SUMIFS Contains

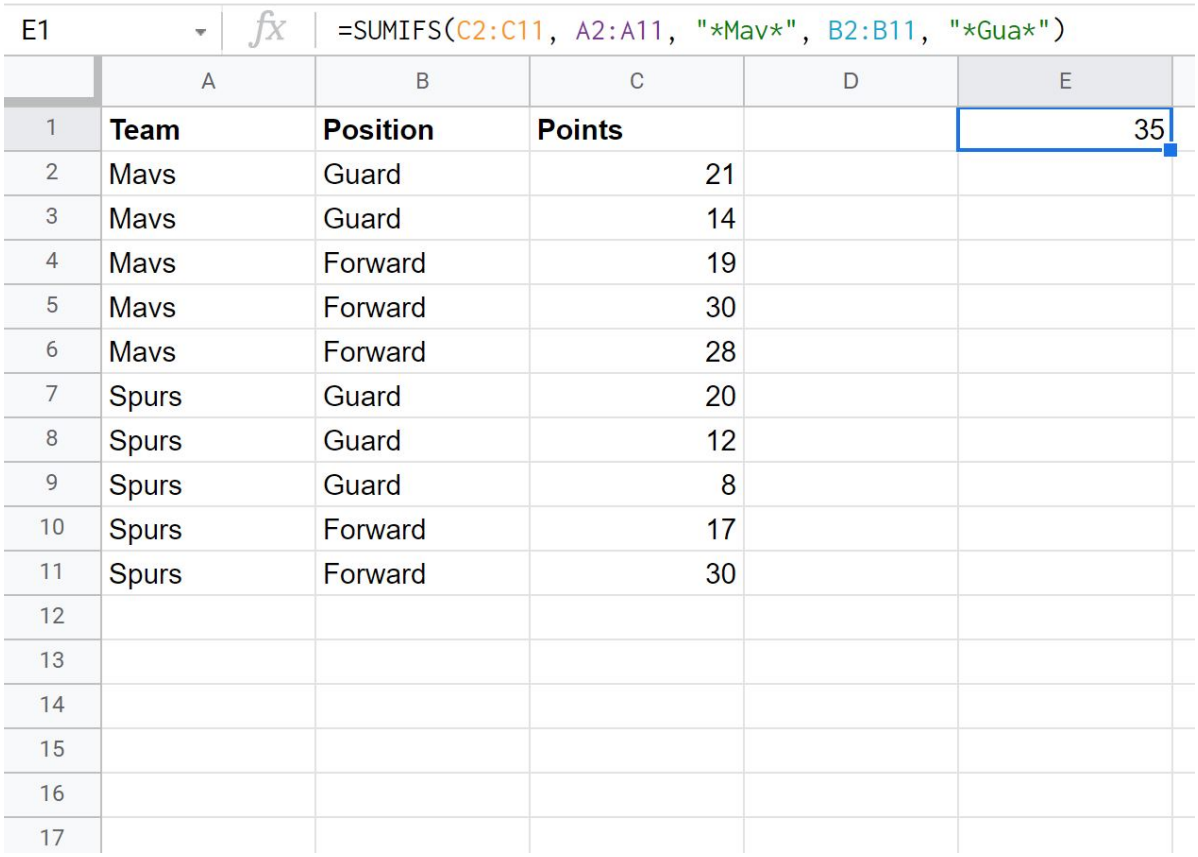
Next, we tackle a more sophisticated analytical requirement involving simultaneous conditions. Imagine your dataset is expanded to include not only team names and points but also player positions. The objective is now to sum points only for players who satisfy a specific combination of criteria: their team name must contain "Mav" AND their position must contain "Gua". This level of precision is achieved through the use of the SUMIFS function.

Assuming team names are in column A, positions are in column B, and points are in column C, the specific formula designed for this multi-conditional summation is:

=SUMIFS(C2:C11, A2:A11, "*Mav*", B2:B11, "*Gua*")

In this advanced arrangement, **C2:C11** is established as the `sum_range`. The first criterion dictates that cells in **A2:A11** (Team column) must contain the string "Mav". Concurrently, the second criterion strictly requires that cells in **B2:B11** (Position column) must contain the substring "Gua" (matching entries like "Guard" or "Guardsman"). Only those rows that satisfy **both** mandatory conditions will contribute their corresponding point values to the final sum.

The image below provides a visual confirmation of this formula applied within Google Sheets, showcasing the precise filtering process:



	A	B	C	D	E
1	**Team**	**Position**	**Points**		35
2	Mavs	Guard	21		
3	Mavs	Guard	14		
4	Mavs	Forward	19		
5	Mavs	Forward	30		
6	Mavs	Forward	28		
7	Spurs	Guard	20		
8	Spurs	Guard	12		
9	Spurs	Guard	8		
10	Spurs	Forward	17		
11	Spurs	Forward	30		
12					
13					
14					
15					
16					
17					

The resulting output reveals that the total points for players whose team name contains "Mav" and whose position contains "Gua" is **35**. This precise aggregation capability demonstrates the immense filtering power of `SUMIFS` when managing complex, multi-conditional data requirements. To verify this result manually, we look for rows meeting both conditions and sum their points: 21 + 14 = **35**. This confirms the correct execution of the SUMIFS function using multiple "contains" criteria.

Best Practices and Advanced "Contains" Techniques

When utilizing SUMIF and SUMIFS with "contains" criteria in Google Sheets, adhering to certain best practices is crucial for ensuring accuracy and optimizing performance. Understanding the nuances of how these functions interpret text will help you construct more dependable and efficient formulas.

A critical characteristic to note is the issue of **case sensitivity**. By default, `SUMIF` and `SUMIFS` in Google Sheets operate in a case-insensitive manner when evaluating text criteria. Consequently, searching for the pattern `"*mav*"` will match "Mav", "mav", "MAV", or "Maverick" without requiring auxiliary functions like `LOWER()` or `UPPER()`. This default behavior greatly simplifies the majority of common text-matching tasks, saving time and complexity in formula construction.

For enhanced flexibility, it is highly recommended to use a cell reference instead of hardcoding your criterion directly into the formula. Rather than inputting **"*Mav*"**, you can store the base search string (e.g., "Mav") in a designated cell (such as **B1**) and concatenate the wildcards around that reference. The resulting formula would look like: `=SUMIF(A2:A11, "*"&B1&"*", C2:C11)`. This method significantly increases the adaptability of your spreadsheets, allowing users to instantly modify the search term without having to edit the underlying function.

Finally, precision in wildcard characters usage dictates the outcome. It is vital to distinguish between the four main types of text pattern matches:

"*string*": Identifies cells that **contain** "string" anywhere within the text.

"string": Requires an **exact match** to "string", with no other characters allowed.

"string*": Matches cells that **start with** "string".

"*string": Matches cells that **end with** "string".

Understanding these subtle yet crucial variations is essential for accurate data manipulation. If, after evaluation, no cells satisfy the criteria you have specified, both `SUMIF` and `SUMIFS` will reliably return the value **0**, which is a predictable and manageable outcome. While these functions are highly efficient for most standard datasets, analysts dealing with exceptionally large data volumes might consider the `QUERY` function for potential performance benefits, although `SUMIF` and `SUMIFS` remain the standard for daily summation tasks.

Conclusion: Mastering Conditional Summation

Developing proficiency in using SUMIF and SUMIFS functions alongside "contains" criteria is an

invaluable skill for any professional working extensively with Google Sheets. These functions grant you the necessary flexibility to aggregate data based on partial text matches, effectively handling the complexities inherent in real-world datasets where exact matches are often impractical or undesirable. By strategically deploying wildcard characters--most importantly, the asterisk (**)--you can construct sophisticated formulas that dynamically filter and precisely sum your information.

Regardless of whether your task requires summing values based on a single condition using `SUMIF` or multiple conditions using `SUMIFS`, the underlying principles of wildcard characters application remain fundamentally consistent. We have provided detailed demonstrations on how to implement these functions, ranging from simple aggregation based on a team name substring to complex filtering that combines criteria for both team and player position. Integrating these powerful techniques into your workflow will significantly enhance your data analysis capabilities, enabling you to extract profound and meaningful insights with increased efficiency and statistical rigor.

Related: How to Use COUNTIF Contains in Google Sheets

Additional Resources for Google Sheets Proficiency

The following tutorials offer comprehensive explanations on executing other commonly required data operations in Google Sheets: