

How to Count Cells Containing Specific Text in Google Sheets Using COUNTIF

Authored by
stats writer

November 22, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Count Cells Containing Specific Text in Google Sheets Using COUNTIF*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=100108>

Understanding the Power of COUNTIF Contains

The ability to analyze textual data is fundamental in modern spreadsheet management. While basic functions like the standard `COUNTIF` are excellent for counting exact matches, real-world data often requires tallying cells that only partially match a specific pattern or string. This is precisely where the "COUNTIF Contains" technique, utilizing wildcard characters, becomes an indispensable tool within Google Sheets. This approach allows users to quickly determine the total number of entries within a specified range that include a specific sequence of text, regardless of what surrounding characters might be present.

To utilize this functionality, you initiate the process by typing `=COUNTIF` into a cell. This is followed by defining the range of cells you intend to evaluate, and finally, specifying the text string (the criterion) you are searching for, appropriately wrapped in wildcards. The function executes this request and returns an integer, which represents the precise count of cells in the defined range that successfully contain the specified text string.

The core concept involves leveraging the `COUNTIF` function and wrapping the desired search string with asterisks (*). These asterisks function as placeholders for any number of characters, defining a fuzzy search pattern rather than a strict, exact-match requirement. This technique is invaluable when analyzing large datasets where text entries are descriptive or inconsistent, such as comments, addresses, or product descriptions.

The Role of Wildcard Characters in Text Matching

When working with text strings in spreadsheet environments like Google Sheets, standard comparison operators are often too rigid for real-world data analysis. To introduce the necessary flexibility for identifying partial text matches, we rely on special symbols known as wildcard characters. The primary wildcard used for this purpose in the context of `COUNTIF` and `COUNTIFS` is the asterisk (*). Understanding the placement and role of this operator is paramount for implementing containment checks successfully.

In the context of the `COUNTIF` function, the asterisk signifies any sequence of characters, of any length, including zero characters. When you structure your criterion as `"*search_term*"`, you are instructing the function to count the cell if "search_term" is present anywhere within that cell's content. For example, using `"*App*"` would count cells containing "Apple," "Application," "Reapply," or simply "App."

This powerful usage ensures that the function counts the cell regardless of where the specified substring appears--at the beginning, end, or middle of the cell content. The official documentation confirms that this * operator is the designated wildcard character for pattern matching within the

criteria arguments of aggregation functions in Google Sheets. Utilizing wildcards correctly simplifies complex data filtering and aggregation significantly.

Method 1: Counting Cells with One Specific String (COUNTIF)

When the goal is to count occurrences based solely on a single condition--such as identifying how many cells in a given column contain a specific keyword--the standard COUNTIF function is the most appropriate and efficient tool. This method is utilized for straightforward analysis requiring the inspection of one range against one partial matching pattern. We begin by reviewing the fundamental structure required for this single-criteria approach.

You can use the following formula structure in Google Sheets to count the number of cells that contain a specific string:

```
=COUNTIF(A2:A11, "*string*")
```

This powerful yet concise formula is designed to perform a dedicated count across the defined range, **A2:A11**. The crucial search criterion, `"*string*"`, dictates that a cell is tallied if the exact sequence of characters "string" appears anywhere within its textual content. This structure ensures that cells like "start string end" or "justastring" are all included in the final sum.

This specific implementation of COUNTIF counts the total number of cells in the range **A2:A11** that contain "string" anywhere in the cell. If the requirement was to find strings that only started with "string," the criterion would be `"string*"`, demonstrating the precise control afforded by the wildcard placement.

Method 2: Counting Cells with Multiple Criteria (COUNTIFS)

Data analysis often necessitates counting rows that satisfy two or more conditions concurrently, potentially across different columns. For example, counting staff members who are "Part-Time" AND work in the "Marketing" department. When multiple partial-matching criteria must be met simultaneously, the COUNTIFS function is the necessary tool. This function is specifically designed to handle multiple pairs of range and criterion arguments, linking them with an implied logical AND operation.

To implement "COUNTIFS Contains," the principle remains the same: apply the wildcard operator (*) to each individual text criterion. The structure involves sequentially listing range-criterion pairs. The function iterates through the rows, checking if the first condition is met in the first range AND if the second condition is met in the second range, and so on. Only rows that pass all defined checks are included in the final count.

This method is critical for conducting highly specific, filtered aggregations, especially where the intersection of several containing strings defines the relevant data set. Here is the generalized formula structure for handling multiple criteria containment checks:

=COUNTIFS(A2:A11, "*string1*", B2:B11, "*string2*")

This formula counts the number of rows where the cell in the range **A2:A11** contains "string1" and the corresponding cell in the range **B2:B11** contains "string2." Both conditions must be met for the row to be tallied. Failure to meet one condition results in exclusion from the count, reinforcing the logical AND nature of COUNTIFS.

It is important to reiterate that the * operator is the primary mechanism that allows this partial matching. Without it, the formula would require exact matches for "string1" and "string2," which is generally too restrictive for most containment tasks.

Practical Application: Single Criteria Example

Let us examine a concrete example using the single-criteria approach. Imagine a spreadsheet containing player data, and we are interested in counting how many rows in the **Team** column contain the partial string "Mav." This is a typical scenario where the team names might include variations like "Mavericks," "Mavs," or "Dallas Mavs."

We apply the COUNTIF function targeting the range A2:A11, where the team names are located. The formula is written as follows:

=COUNTIF(A2:A11, "*Mav*")

The execution of this formula within the sheet is visually represented below, showing the data structure and the calculated result:

	A	B	C	D	E
E1	=COUNTIF(A2:A11, "*Mav*")				
1	Team	Position	Points		5
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					
18					
19					
20					

From the output, we can clearly see that the function returns the number 5. This result indicates that 5 rows in the **Team** column contain the substring "Mav" somewhere within their cell content. This confirms that the COUNTIF function successfully utilized the wildcards to perform the required partial text match.

Practical Application: Multiple Criteria Example

For a more detailed analysis, we move to a scenario requiring two simultaneous conditions. We want to count the number of players whose **Team** name contains "Mav" AND whose **Position** contains the string "Guar" (e.g., Guard or Guarani). This necessitates using the COUNTIFS function to link criteria across two separate columns.

We must specify two distinct range-criterion pairs, each utilizing the containment logic (wildcards) for text matching:

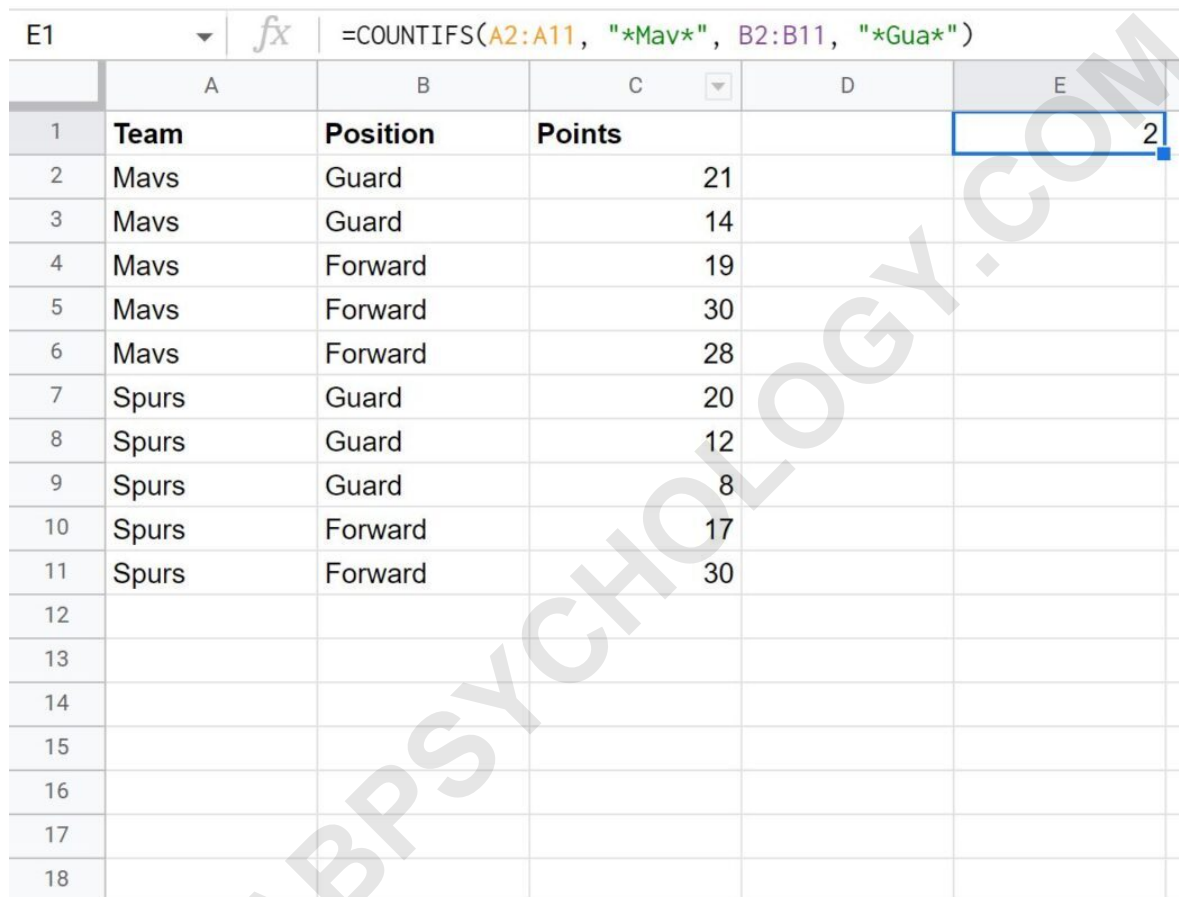
Range A2:A11 (Team column) must contain "*Mav*".

Range B2:B11 (Position column) must contain "*Guar*".

The complete formula combining these two containment checks is structured as follows:

=COUNTIFS(A2:A11, "*Mav*", B2:B11, "*Gua*")

The execution of this formula is demonstrated in the subsequent screenshot, showing the result of the intersection analysis:



The screenshot shows a Google Sheet with the following data:

	A	B	C	D	E
1	Team	Position	Points		
2	Mavs	Guard	21		2
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					
18					

The resulting output is 2. This signifies that only two rows satisfy both containment criteria simultaneously, meaning only 2 rows contain "Mav" in the **Team** column and also contain "Gua" in the **Position** column. This illustrates the effectiveness of `COUNTIFS` in performing complex, filtered counts based on partial text matches.

Best Practices for COUNTIF Contains

To ensure accuracy and efficiency when using containment logic in Google Sheets, several best practices should be observed. First, always verify that your ranges are correctly defined and uniform in length, especially when using `COUNTIFS`. Mismatched ranges will lead to calculation errors or misleading results.

Second, rigorously manage your criteria strings. The inclusion of the asterisk (*) is non-negotiable for partial matching; without it, the function reverts to exact matching. Furthermore, if you need to specifically search for an asterisk character itself (rather than using it as a wildcard), you must escape it using a tilde (~) before the asterisk (e.g., "~*") within the criterion string.

Finally, consider alternatives for extremely complex pattern matching. While the wildcard approach covers most needs, if you require advanced regular expression capabilities--such as searching for specific patterns or character sets that wildcards cannot handle--the COUNTIF and COUNTIFS functions can often be replaced or enhanced by combining functions like ARRAYFORMULA with REGEXMATCH to provide superior control over text analysis.

ARABPSYCHOLOGY.COM