

Learn How to Count Specific Words in Google Sheets with COUNTIF

Authored by
Mohammed looti

June 1, 2026

RECOMMENDED CITATION

Mohammed looti (2026). *Learn How to Count Specific Words in Google Sheets with COUNTIF*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=3676>

The ability to efficiently analyze and summarize textual data is fundamental to mastering [Google Sheets](#). When dealing with large ranges of information, users often need to determine how many cells within a specific column or row contain a particular keyword or phrase, regardless of other text present in the cell. To accurately count the occurrence of a specific word or substring within a defined range in [Google Sheets](#), we utilize the powerful **COUNTIF function** in conjunction with appropriate **wildcard characters**. This approach provides a robust and scalable method for data auditing and reporting.

The core mechanism for achieving this count is surprisingly straightforward, relying on a standard conditional counting formula. The specific structure required to count cells containing a target word--even when surrounded by other text--leverages the asterisk (*) [wildcard character](#). This character acts as a placeholder for any sequence of zero or more characters, ensuring that the criterion matches the target word wherever it appears within the cell's contents.

You can implement the following formula structure to precisely count the occurrence of a specific word within any selected range in your spreadsheet. This example demonstrates how to search for the word "Guard" within the range B2 through B15:

```
=COUNTIF(B2:B15, "*Guard*")
```

This particular formula structure instructs [COUNTIF function](#) to evaluate every cell in the designated range, **B2:B15**, and return a total count only for those cells that satisfy the defined criterion. Specifically, the use of the double asterisks (*) surrounding the search term ensures that the count includes cells where "Guard" is preceded by or followed by any text, or both. For instance, cells containing "Point Guard," "Shooting Guard," or simply "Guard" will all be included in the final tallied result.

Understanding the COUNTIF Function in Google Sheets

The [COUNTIF function](#) is one of the most frequently used statistical functions in spreadsheet software, designed to perform conditional counting. Its primary objective is to count the number of cells within a specified range that meet a single, predefined condition. The syntax is universally recognized as `COUNTIF(range, criterion)`. The **range** argument defines the collection of cells that the function will analyze, and the **criterion** argument specifies the condition that must be met for a cell to be included in the final count. Defining both of these components accurately is essential for any successful data aggregation effort.

In complex data environments, simply counting cells that equal a specific value is rarely sufficient. Often, data entries are concatenated strings, meaning they contain multiple pieces of information within a single cell. If we were to use the criterion `"Guard"` without wildcards, the function would

only count cells that contain *exactly* the text "Guard" and nothing else. This strict matching would fail to capture relevant data points like "Backup Guard" or "Veteran Guard," leading to an inaccurate and incomplete data summary. Therefore, understanding how to apply flexible matching criteria is crucial for robust data analysis in spreadsheet applications.

The criterion argument, when dealing with text, must be enclosed in double quotation marks. When utilizing the **COUNTIF function** for partial text matching, the criterion is where the **wildcard characters** come into play. The function processes this criterion as a form of [boolean logic](#), checking each cell against the pattern defined by the text and the wildcards. This powerful combination allows analysts to extract specific insights even from unstructured or semi-structured textual data, providing high utility for tasks such as categorizing items, tracking statuses, or, as demonstrated here, identifying specific roles in a list of personnel.

The Power of Wildcards: Counting Partial Matches

To effectively count words that appear as only a part of the cell content, we must leverage the concept of **wildcard characters**. In [Google Sheets](#), the primary wildcard used for variable-length text matching is the asterisk (*). This symbol represents any sequence of characters--including zero characters--of any length. By strategically placing the asterisk around our search term, we transform the criterion from a strict equality check into a flexible pattern match.

When defining the criterion as `"*Guard*"`, the first asterisk (*) tells the function to accept any text that might precede the word "Guard." The second asterisk (*) instructs the function to accept any text that might follow the word "Guard." Consequently, the function searches for the substring "Guard" anywhere within the cell's contents. This is a fundamental technique for handling descriptions, notes, or titles where the desired keyword is embedded within a longer string. Without this use of **wildcards**, the counting task would become exponentially more complex, potentially requiring nested functions or array formulas.

While the asterisk is used for matching multiple characters, it is also important to recognize the function of the question mark (?) wildcard, which matches any single character. While less common for simple substring searching, the question mark is valuable when searching for patterns of a fixed length, such as codes or abbreviations where only one character is variable (e.g., `"AB-?-10"`). However, for the purpose of counting a specific embedded word like "Guard," the use of the dual asterisk, `"*Guard*"`, remains the most efficient and standard practice, ensuring that the function captures all relevant instances regardless of the surrounding text or length of the cell entry.

Practical Example: Analyzing Sports Data

To illustrate the practical application of this counting method, consider a typical scenario involving

data analysis in a sports context. Suppose we have a detailed [dataset](#) containing information about various basketball players, including their name, team, and specific position. Our goal is to quickly determine how many players in this list are categorized as a "Guard," regardless of whether they are a Point Guard, Shooting Guard, or simply listed as a Guard role. This requires counting partial matches within the 'Position' column.

The following image represents the raw [dataset](#) we will be working with. Note that the player positions in column B are often compound terms (e.g., "Point Guard," "Forward/Center"). If we rely solely on exact matching, we would miss most of the relevant data points.

	A	B	C	D	
1	Player	Position	Points		
2	Andy	Shooting Guard	22		
3	Bob	Point Guard	34		
4	Chad	Point Guard	20		
5	Doug	Small Foward	19		
6	Eric	Power Forward	15		
7	Frank	Center	15		
8	George	Power Foward	19		
9	Henry	Point Guard	12		
10	Isaac	Shooting Guard	31		
11	John	Small Forward	38		
12	Ken	Small Forward	12		
13	Liam	Point Guard	11		
14	Mike	Center	10		
15	Nate	Center	8		
16					
17					
18					
19					

The challenge here is to isolate the role "Guard" even when it is combined with other positional descriptions. We must devise a formula that is intelligent enough to look past the ancillary text and focus only on the presence of our target keyword. This is precisely where the partial matching capability of the **COUNTIF function**, enhanced by **wildcard characters**, provides an elegant and fast solution, eliminating the need for manual inspection or complex text manipulation functions like FIND or SEARCH.

Step-by-Step Implementation of the Formula

With our objective clearly defined--to count the occurrences of the substring "Guard" in the range B2:B15--we can proceed with the step-by-step implementation of the counting formula. First, select an empty cell where you wish the result of the count to appear (for example, cell D2). Next, input the [COUNTIF function](#), specifying the range that covers all player positions, which is **B2:B15** in this example. Finally, define the criterion using the necessary **wildcards** to ensure partial matching.

The fully constructed formula, as shown below, correctly encapsulates the range and the flexible search pattern. The range **B2:B15** explicitly defines the boundaries of the data being analyzed, ensuring that no irrelevant cells are included in the calculation. The criterion `"*Guard*"` serves as the pattern matching engine, instructing [Google Sheets](#) to look for "Guard" appearing anywhere within the text string of each cell in that range.

=COUNTIF(B2:B15, "*Guard*")

Once the formula is entered, [Google Sheets](#) immediately processes the instruction, performing a sequential evaluation of each cell from B2 down to B15. For every cell where the condition is met (i.e., the cell contains the substring "Guard"), the counter increments by one. The formula is designed to handle multiple instances of the search term within the range, providing a single, consolidated result that represents the total number of cells containing the specified word. The following screenshot visually confirms the implementation of this formula within the sheet environment, demonstrating its placement and immediate calculation of the result.

E2		<i>fx</i>	=COUNTIF(B2:B15, "*Guard*")		
	A	B	C	D	E
1	Player	Position	Points		Count of Cells that contain "Guard"
2	Andy	Shooting Guard	22		6
3	Bob	Point Guard	34		
4	Chad	Point Guard	20		
5	Doug	Small Foward	19		
6	Eric	Power Forward	15		
7	Frank	Center	15		
8	George	Power Foward	19		
9	Henry	Point Guard	12		
10	Isaac	Shooting Guard	31		
11	John	Small Forward	38		
12	Ken	Small Forward	12		
13	Liam	Point Guard	11		
14	Mike	Center	10		
15	Nate	Center	8		
16					
17					
18					
19					
20					

Verifying the Results and Manual Confirmation

Upon executing the formula `=COUNTIF(B2:B15, "\*Guard\*")`, the output indicates a count of **6**. This figure represents the total number of cells within column B (rows 2 through 15) that contain the word "Guard" as part of their entry. While the automated calculation provided by the [COUNTIF function](#) is highly reliable, it is good practice, especially when learning new techniques, to confirm the result manually. This step ensures full understanding of which specific cells met the defined partial matching criterion.

The result of 6 suggests that six different players in our [dataset](#) are categorized, at least partially, as a Guard. By manually reviewing the data in column B, we can identify and highlight the specific entries that contributed to this total. This process of visual validation not only confirms the accuracy of the formula but also reinforces the conceptual understanding of how the **wildcard characters** successfully captured the partial matches.

The following visual confirmation shows the rows that were included in the count of 6. We can see entries such as "Point Guard," "Shooting Guard," and "Guard/Forward" were all successfully included because they contained the required substring "Guard." The formula's use of `\*` on both sides allowed it to ignore the preceding and succeeding text, focusing purely on the core keyword.

This manual verification step is critical for building trust in automated data analysis methods and ensuring data integrity.

	A	B	C	D	E
1	Player	Position	Points		Count of Cells that contain "Guard"
2	Andy	Shooting Guard	22		6
3	Bob	Point Guard	34		
4	Chad	Point Guard	20		
5	Doug	Small Foward	19		
6	Eric	Power Forward	15		
7	Frank	Center	15		
8	George	Power Foward	19		
9	Henry	Point Guard	12		
10	Isaac	Shooting Guard	31		
11	John	Small Forward	38		
12	Ken	Small Forward	12		
13	Liam	Point Guard	11		
14	Mike	Center	10		
15	Nate	Center	8		
16					
17					
18					
19					
20					
21					

Adapting the Formula for Other Search Terms

The beauty of using the **COUNTIF function** with **wildcards** lies in its adaptability. Once the core structure and range are established, finding the count of cells containing a different word requires only a simple modification to the criterion argument. This scalability allows analysts to reuse the same formula framework to audit an entire column for various keywords quickly.

To find the count of cells that contain a different target word, you simply replace "Guard" with the desired term, ensuring that the new term is still enclosed within the asterisk wildcards (`\*`) and the quotation marks. For example, if we now wish to count how many players have "Forward" listed in their position, we adjust the formula as follows, keeping the range **B2:B15** constant:

=COUNTIF(B2:B15, "*Forward*")

It is important to note that standard text matching in [Google Sheets](#) using [COUNTIF function](#) is generally **case-insensitive**. This means that searching for `"\*guard\*"` will yield the same result as

searching for `"*GUARD*"` or `"*Guard*"`. This inherent feature simplifies the counting process by eliminating the need for complex functions like `LOWER()` or `UPPER()` to standardize the case of the data before counting. However, if strict case-sensitive counting is ever required, advanced users would need to employ the `COUNT(FILTER(...))` structure combined with the `FIND()` function, which is inherently case-sensitive, to achieve that specific level of precision. For most general data analysis tasks, however, the simple, case-insensitive COUNTIF approach remains the most efficient methodology.

Further Resources for Data Analysis

Mastering conditional counting is a key milestone in advanced spreadsheet management. The methods discussed here using the **COUNTIF function** form the foundation for many more complex data operations. To further enhance your proficiency in data validation, aggregation, and reporting within Google Sheets, consider exploring related functions and tutorials that expand upon these foundational concepts.

The following tutorials explain how to perform other common and valuable operations in Google Sheets, building upon your understanding of conditional logic:

COUNTIFS for Multiple Criteria: Learn how to count cells based on two or more conditions simultaneously (e.g., counting players who are both a "Guard" *and* play for a specific "Team").

SUMIF and AVERAGEIF: Explore how to conditionally sum or average numerical data based on a text criterion, which is highly useful for calculating statistics tied to specific categories.

Using Regular Expressions (REGEXMATCH/REGEXEXTRACT): Understand how to implement highly complex pattern matching using regular expressions, offering finer control than standard wildcard characters.

VLOOKUP and INDEX/MATCH: Tutorials on retrieving specific values based on matching criteria, a complement to counting operations.