

Counting Names in Google Sheets: A Tutorial with Examples

Authored by
Mohammed loot

November 10, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Counting Names in Google Sheets: A Tutorial with Examples*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=16056>

Mastering Conditional Counting in Google Sheets

Effectively analyzing large datasets often hinges on the ability to summarize information based on specific, predefined criteria. Whether you are managing complex inventory records, tracking project assignments, or simply processing a long list of employee names, determining the frequency of a particular textual entry is a fundamental requirement of [data analysis](#). Fortunately, [Google Sheets](#) provides an array of powerful conditional functions designed to streamline this process. The most critical tool in our arsenal for counting names is the highly versatile **COUNTIF** function, which allows us to evaluate a designated range of cells against a single, specific condition.

This guide details the three essential techniques required for accurate conditional counting within spreadsheets. We will explore methods for achieving an **exact match**, which is vital for standardized data; conducting **partial matches** using flexible [wildcard characters](#) for broader searches; and finally, aggregating counts when cells must match **one of several names**. Mastering these applications of the [COUNTIF](#) function ensures you can perform flexible and powerful data aggregation, regardless of the size or complexity of your dataset.

The following sections will walk through each technique step-by-step, providing clear examples and detailed formula breakdowns that transition directly from theoretical knowledge to practical application. These methods form the foundation of efficient data reporting and querying in [Google Sheets](#), enabling you to quickly extract meaningful insights from raw data.

Quick Reference: Essential Formulas for Name Counting

To facilitate quick reference, here are the three core formulas used to count names, categorized by the string matching requirement. These formulas utilize the [COUNTIF](#) function, which is the standard method for conditional counting in spreadsheets.

Formula 1: Count Cells with Exact Name Match

=COUNTIF(A2:A11, "Bob Johnson")

This foundational structure counts entries only when the cell content is **absolutely identical** to the specified criterion, "Bob Johnson." The function analyzes the defined range (A2:A11) and only includes cells that are an exact textual match, ensuring the highest level of precision.

Formula 2: Count Cells Using Partial Name Match (Wildcards)

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

When a precise match is too restrictive, **partial matching** offers flexibility. By enclosing the search string ("Johnson") in [wildcard characters](#) (asterisks, `*`), we instruct [COUNTIF](#) to look for that string anywhere within the cell. This is ideal for counting last names regardless of first names or titles.

Formula 3: Count Cells Matching One of Several Names (OR Condition)

=COUNTIF(A2:A11, "*Johnson*") + COUNTIF(A2:A11, "*Smith*")

To tally entries that satisfy **multiple distinct criteria**, we must combine separate [COUNTIF](#) functions using the addition operator (`+`). This method simulates an OR condition: counting all cells that match the first name OR the second name. This approach is highly effective for gathering totals across different categories.

The following dataset, representing a list of employees, will be used as the reference data for all subsequent demonstrations in [Google Sheets](#):

	A	B	C	D
1	Employee	Sales		
2	Bob Johnson	22		
3	Doug Arnold	25		
4	Doug Mandlen	14		
5	Andy Smith	9		
6	Mike Johnson	8		
7	Dave Stenlen	12		
8	Craig Reynolds	12		
9	Arthur Johnson	10		
10	Ron Smith	20		
11	Bob Johnson	31		
12				
13				
14				
15				
16				

Example 1: Precision Counting with Exact Matches

The first and most stringent requirement in name counting is determining the exact frequency of a specific, complete name within a dataset. This method is indispensable when verifying standardized user inputs, reconciling unique IDs, or ensuring absolute consistency in data entry, where even minor variations like extra spaces or appended suffixes must be ignored. When the [COUNTIF](#) function is used without any [wildcard characters](#), it defaults to a direct, character-for-character comparison against the specified text string.

To illustrate, suppose we need to count the precise number of times the name "Bob Johnson" appears in column A. The formula is straightforward, demanding an absolute match against the exact text within the quotation marks:

=COUNTIF(A2:A11, "Bob Johnson")

In this scenario, [COUNTIF](#) meticulously examines the range A2:A11. Any entries that deviate, such as "Bob Johnson, Jr." or "Bob Johnson (Consultant)," are strictly excluded because they do not satisfy the strict exact match criterion. This functionality is crucial for maintaining data integrity. The

following visual confirms the outcome of applying this precise counting formula in a live sheet environment:

D2 fx =COUNTIF(A2:A11, "Bob Johnson")

	A	B	C	D
1	Employee	Sales		Count of Bob Johnson
2	Bob Johnson	22		2
3	Doug Arnold	25		
4	Doug Mandlen	14		
5	Andy Smith	9		
6	Mike Johnson	8		
7	Dave Stenlen	12		
8	Craig Reynolds	12		
9	Arthur Johnson	10		
10	Ron Smith	20		
11	Bob Johnson	31		
12				
13				
14				
15				
16				
17				

As demonstrated by the output, there are exactly **2** cells that contain the precise text "Bob Johnson." This result underscores the necessity of precise criteria when variations in data must be intentionally filtered out.

Example 2: Leveraging Wildcards for Partial Name Counting


Often, the analytical goal is to count entries based on a component of the name, such as only a last name or a specific keyword that might be embedded within a longer string. In these cases, the exact match approach fails, and **partial matching** becomes essential. [Google Sheets](#) supports the use of the [wildcard character](#), the asterisk (*), which serves as a powerful placeholder for any sequence of zero or more characters. This flexibility allows us to search for names irrespective of titles, prefixes, or suffixes.

If our objective is to count all cells in column A that contain the last name "Johnson" anywhere

within the text, we must strategically wrap the criterion in asterisks. This tells the [COUNTIF](#) function to search for the literal string "Johnson," while disregarding any content that appears either before or after it:

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

It is crucial to understand the placement of the [wildcard](#). If the formula were written as `"Johnson"` (without the leading asterisk), it would only capture names that **begin** with "Johnson" (e.g., "Johnson, M."). By including the leading asterisk, we ensure that entries like "Mike Johnson" and "Arthur Johnson" are correctly included in the count, as the asterisk accounts for their first names. The result of this partial matching criteria is shown below:

D2  =COUNTIF(A2:A11, "*Johnson*")

	A	B	C	D
1	Employee	Sales		Count of Johnson
2	Bob Johnson	22		4
3	Doug Arnold	25		
4	Doug Mandlen	14		
5	Andy Smith	9		
6	Mike Johnson	8		
7	Dave Stenlen	12		
8	Craig Reynolds	12		
9	Arthur Johnson	10		
10	Ron Smith	20		
11	Bob Johnson	31		
12				
13				
14				
15				

The resulting output confirms that there are **4** cells containing "Johnson" somewhere in the name. We can manually verify this by checking the employee list:

Bob Johnson
 Mike Johnson
 Arthur Johnson
 Bob Johnson

This validation confirms the accuracy and utility of the double [wildcard](#) approach for flexible text searching.

Example 3: Combining Criteria for Counting Multiple Names

A frequent requirement in [data analysis](#) is counting entries that satisfy **one of several possible criteria**--a logical OR operation. For instance, a human resources analyst might need to count all employees whose last name is either "Johnson" or "Smith." A key limitation of the single [COUNTIF](#) function is that it does not natively process multiple criteria as an OR condition within one function call.

The effective solution to this constraint is executing a separate [COUNTIF](#) calculation for each criterion, and then simply summing the results together using the addition operator (+). This technique correctly tallies the count of "Johnson" entries and adds it to the count of "Smith" entries, yielding the total number of cells that match at least one of the desired names. We use the following formula to count the number of cells in column A that contain either "Johnson" or "Smith":

=COUNTIF(A2:A11, "*Johnson*") + COUNTIF(A2:A11, "*Smith*")

This additive structure ensures accurate aggregation, provided the two criteria are mutually exclusive (i.e., one cell cannot simultaneously contain both "Johnson" and "Smith" in a way that satisfies both partial matches). If the criteria were not mutually exclusive (e.g., searching for text that includes both "Manager" and "New York"), this simple SUM method would lead to overcounting, necessitating more advanced functions like `COUNTIFS` or the powerful `QUERY` function. For typical name counting, however, this approach is straightforward and highly reliable.

The following screenshot illustrates the practical execution of this formula, showcasing the combined count of both target names within the specified range:

D2 ∇ | fx =COUNTIF(A2:A11, "*Johnson*") + COUNTIF(A2:A11, "*Smith*")

	A	B	C	D	E
1	Employee	Sales		Count of Johnson or Smith	
2	Bob Johnson	22		6	
3	Doug Arnold	25			
4	Doug Mandlen	14			
5	Andy Smith	9			
6	Mike Johnson	8			
7	Dave Stenlen	12			
8	Craig Reynolds	12			
9	Arthur Johnson	10			
10	Ron Smith	20			
11	Bob Johnson	31			
12					
13					
14					
15					

The final count is **6**, representing the total number of cells that contain either "Johnson" or "Smith." This result is derived from the sum of 4 "Johnson" entries (validated in Example 2) and 2 "Smith" entries present in the dataset.

Deep Dive: Understanding Case Sensitivity and Data Integrity

The efficacy of the [COUNTIF](#) function in [Google Sheets](#) is fundamentally governed by a clear understanding of its syntax: `COUNTIF(range, criterion)`. The **range** defines the specific set of cells to be analyzed (e.g., A2:A11), and the **criterion** sets the condition that must be met for a cell to be counted. When working with text, the criterion must always be correctly enclosed in double quotation marks.

A critical point of awareness when counting names is how Sheets handles **case sensitivity**. In virtually all standard text comparisons within [Google Sheets](#), including `COUNTIF`, the function is inherently **case-insensitive**. This means that a search criterion of "bob johnson" will yield the exact same result as searching for "Bob Johnson" or "BOB JOHNSON." While this is often convenient for general name counting, if your application requires strict case matching (e.g., distinguishing specific product codes like 'aBC' from 'abc'), you must employ more complex, case-sensitive functions. These alternatives typically involve combining array formulas like

`SUMPRODUCT` with the `EXACT` function, or utilizing specialized functions within the `QUERY` framework.

Furthermore, when employing partial matching using the [wildcard character](#) (`*`), remember that the asterisk matches any sequence of characters, including problematic extra spaces. If your source data contains inconsistent formatting, such as accidental leading or trailing spaces, the wildcard method can often be more forgiving than the exact match. However, for maintaining robust data quality standards, it is always best practice to clean data beforehand using functions like `TRIM` to remove extraneous whitespace, thereby ensuring consistent and predictable results regardless of the counting method used.

Advanced Considerations and Alternative Resources

While [COUNTIF](#) effectively handles basic exact and partial name counting, complex analytical scenarios often exceed its capabilities, requiring more specialized functions. For instance, if you need to count names based on **multiple criteria that must all be true simultaneously** (an AND logical operation), such as counting employees named "Johnson" who *also* work in the "Sales" department, you must transition to the `COUNTIFS` function. This function permits the evaluation of multiple range/criterion pairs concurrently.

For the most dynamic filtering, aggregation, and conditional counting--especially when criteria are externalized in separate cells or require highly complex logic--the **QUERY function** provides unparalleled power. The QUERY function uses a simplified form of [SQL](#) (Structured Query Language) that is integrated directly into [Google Sheets](#). This enables users to write sophisticated, database-like statements for filtering, sorting, and aggregating data, far surpassing the limitations of basic conditional counting formulas.

Additional Resources

The following tutorials explain how to perform other common tasks in Google Sheets: