

Count Names in Google Sheets (3 Examples)

Authored by
stats writer

November 17, 2025

RECOMMENDED CITATION

stats writer (2025). *Count Names in Google Sheets (3 Examples)*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=94572>

Analyzing and managing large datasets often requires the ability to quickly tally specific entries. In Google Sheets, counting names based on precise or partial criteria is easily achieved using the powerful **COUNTIF** function. This guide provides an expert breakdown of three essential techniques for counting names, ranging from exact matches to complex multi-criteria searches.

The core concept revolves around the **COUNTIF** function, which is designed to count cells within a specified range that meet a given condition. Mastering this function, along with the strategic use of wildcards, is fundamental for efficient data analysis in any modern spreadsheet application. We will explore methods to count precise entries, locate names containing specific substrings, and even tally entries that satisfy multiple criteria simultaneously.

Below is a summary of the three fundamental formulas we will explore. These methods provide the groundwork for advanced data interrogation within your worksheets, ensuring accurate reporting and analysis:

Formula 1: Count Cells with Exact Name

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

Formula 2: Count Cells with Partial Name

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

Formula 3: Count Cells with One of Several Names

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

The following detailed examples illustrate the application of each formula using a sample employee dataset. This dataset, which we will reference throughout this tutorial, contains a column of names in the range A2:A11.

| | 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: Counting Cells with Exact Name

When your requirement is to tally only those cells that contain a precise, non-negotiable string of text, the standard application of the COUNTIF function is ideal. This method demands that the criterion matches the cell content character for character, ensuring high precision in your counts. For example, if you need to know how many times "Bob Johnson" appears in your employee roster, the exact match formula provides the definitive answer.

To implement this, you specify the range (A2:A11) and the exact text criterion ("Bob Johnson"). It is essential to remember that the text criterion must be enclosed in double quotes. Furthermore, **COUNTIF** in Google Sheets is generally case-insensitive, meaning "bob johnson" will count the same cells as "Bob Johnson." However, relying on exact matching ensures that no additional characters or spaces are inadvertently included in the count, thus validating the integrity of your data.

We use the following formula to count the number of cells in the defined range (column A) that strictly contain the exact name "Bob Johnson". The formula is concise and highly effective for singular criterion matching:

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

The resulting output confirms the count based on the data provided in the screenshot, demonstrating the formula's straightforward application in a real-world scenario. Notice that only entries where the cell content is *exactly* "Bob Johnson" are tallied; cells containing "Bob Johnson Jr." or "Bob Johnson (HR)" would be excluded.

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 illustrated in the resulting calculation, there are precisely **2** cells within the A2:A11 range that contain the exact string "Bob Johnson". This result confirms the powerful, yet simple, mechanism of **COUNTIF** when applied to precise string matching.

Deep Dive: Understanding Exact Matching Criteria

The success of the exact match method hinges on the rigorous definition of the criterion. When you input a text string into the criterion argument of the **COUNTIF** function, Google Sheets performs a comparison that requires complete identity between the criterion and the cell value. This is particularly important when dealing with datasets that might contain minor variations, such as leading or trailing spaces, which can often be invisible to the user but significant to the formula.

For instance, if cell A5 contained "Bob Johnson " (with a space at the end) and your criterion was simply "Bob Johnson," the count would not include A5. This characteristic makes exact matching a crucial tool for performing quality checks or ensuring that names adhere to strict data validation

standards. If your goal is pristine data management, utilizing functions that enforce this strict identity is non-negotiable.

While exact matching is powerful for precise analysis, it may not be suitable when dealing with inconsistent data input or needing to count occurrences of a common surname regardless of the first name. This limitation leads us to the second, more flexible method: counting partial names. Understanding when to pivot from exact matching to partial matching is key to becoming a highly proficient Google Sheets user, adapting your formulas to the specific nuances of your data.

Example 2: Counting Cells with Partial Name Using Wildcards

Often, data analysis requires counting entries where only a portion of the string is known. Perhaps you want to count all employees with the surname "Johnson," regardless of their first name. In such cases, the exact match criterion is too restrictive. This is where wildcards come into play, specifically the asterisk (*), which serves as a placeholder for any sequence of zero or more characters.

By embedding the search term within asterisks, we instruct **COUNTIF** to look for the substring anywhere within the cell. The syntax `"*Johnson"` effectively tells the function: "Count the cell if the string 'Johnson' appears, irrespective of what comes before or after it." This allows the count to include "Bob Johnson," "Mike Johnson," or even "Johnson, Arthur," providing immense flexibility in dealing with less standardized name formats.

To count the number of cells in column A that contain the substring "Johnson" anywhere within the name, we use the asterisk wildcard approach:

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

The following visualization shows the application of this formula. Note how the use of wildcards broadens the scope of the count, capturing multiple variations of the name that share the common required element. This technique is invaluable for analyses based on common attributes, such as department names, geographical locations, or family names.

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 | | | | |

After applying the formula, we observe that the count is **4**. This indicates that four unique entries contain the string "Johnson" somewhere within their name. We can confirm the accuracy of this partial count by reviewing the list of employees who meet this criteria:

Bob Johnson
 Mike Johnson
 Arthur Johnson
 Bob Johnson

The result is verified: there are indeed four employees whose names include the substring "Johnson."

Deep Dive: Leveraging Asterisks and Wildcards in Queries

The asterisk (*) is the most commonly used wildcard in Google Sheets counting functions, representing zero or more characters. However, it is worth noting the existence of the question mark (?), which represents exactly one single character. While less common for counting full names, the question mark is extremely useful when counting entries with slight, predictable variations, such as identifying both "Smith" and "Smythe."

When constructing criteria using wildcards, always ensure the entire expression is enclosed in

double quotes. If you were to reference a cell containing your wildcard criterion (e.g., cell B1 contains `*Johnson*`), you would structure the formula as `=COUNTIF(A2:A11, B1)`. However, when hardcoding the criterion directly, the quotation marks are essential for Google Sheets to interpret the asterisk as a wildcard symbol rather than literal text.

This method offers significant advantages over manual counting or filtering, especially in dynamic environments where data is frequently updated. By using a robust formula like **COUNTIF** with wildcards, you create a dynamic reporting mechanism that instantly reflects changes in the source data without requiring manual recalculation or refiltering. This efficiency is critical for reporting processes based on large or frequently changing datasets.

Example 3: Counting Cells with One of Several Names (OR Logic)

A common data analysis requirement is counting records that satisfy one of several criteria--an "OR" condition. For example, we might need to count all employees whose surname is either "Johnson" or "Smith." Since the standard COUNTIF function only evaluates a single criterion, we cannot ask it to count both conditions simultaneously within one function call.

The solution is to structure the query as a summation of individual **COUNTIF** functions. By calculating the count for the first criterion (Johnson) and adding the count for the second criterion (Smith), we achieve the desired "OR" logic. This method is effective because we are counting distinct cells; a single cell cannot contain both "Johnson" and "Smith" in the context of our name data, thus preventing double-counting.

We use the following combined formula to count the number of cells in column A that contain either "Johnson" or "Smith" somewhere in the name:

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

This formula is straightforward: it executes the partial match count for "Johnson" and independently executes the partial match count for "Smith," then sums the two results. This is the simplest and most robust way to implement multiple "OR" criteria when dealing with disjoint sets of text strings in Google Sheets.

The following screenshot demonstrates the practical result of summing these two conditions:

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 resulting calculation yields **6**. This confirms that there are six entries in the dataset that contain either the string "Johnson" or the string "Smith." This technique can be expanded to include any number of criteria, simply by appending additional **COUNTIF** functions linked by the addition operator (+).

Alternative Methods for Complex Counting

While summing multiple **COUNTIF** functions is excellent for simple "OR" logic, more complex counting scenarios involving multiple criteria (e.g., AND logic, or complex OR/AND combinations) might necessitate different, more powerful functions. For example, if you needed to count employees named "Johnson" who also work in the "Sales" department, you would use the **COUNTIFS** function (plural), which allows for multiple criteria to be evaluated simultaneously.

For scenarios requiring flexible counting based on complex patterns or filtering across multiple columns, the **QUERY** function offers unparalleled power. A **QUERY** statement can filter data using sophisticated SQL-like logic and then return the count of the resulting filtered rows. For example, a query could be written to look for names matching a regular expression pattern, offering a degree of control far beyond simple wildcards.

Furthermore, an alternative approach to counting multiple OR conditions involves using the **ARRAYFORMULA** combined with **SUM** and **REGEXMATCH**. While more advanced, this single-

cell array formula allows you to check for multiple substrings using regular expressions, often streamlining the formula presentation, especially when dealing with dozens of potential names to count.

Summary and Best Practices for Name Counting

Effective data counting in a spreadsheet environment like Google Sheets relies heavily on choosing the appropriate formula for the specific data need. We have demonstrated three core methods: the rigorous exact match, the flexible partial match using wildcards, and the combined approach for "OR" logic. Each method has its place, and understanding the context of your data--whether it requires strict adherence or flexible substring recognition--is the first step toward accurate analysis.

When applying these methods, always follow these best practices: 1) Clearly define your range before writing the formula. 2) Ensure text criteria are enclosed in double quotes. 3) Use the asterisk (*) wildcard only when partial matching is required, as unnecessary use can lead to overcounting. By integrating these structured COUNTIF strategies, you can transform raw data into meaningful metrics quickly and reliably.

These powerful counting techniques are foundational for report generation, dashboard creation, and general data management within Google Sheets. Utilizing formulas like **COUNTIF** ensures that your analyses are always dynamic and reflective of the most current data, minimizing manual intervention and maximizing efficiency.