

# Learn How to Count Occurrences in Google Sheets Using UNIQUE and COUNTIF Functions

Authored by  
**Mohammed looti**

November 2, 2025

## RECOMMENDED CITATION

Mohammed looti (2025). *Learn How to Count Occurrences in Google Sheets Using UNIQUE and COUNTIF Functions*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=8576>

Accurately counting the number of times specific values appear within a dataset is a fundamental requirement of effective [data analysis](#). Whether you are auditing inventory, tracking customer preferences, or analyzing survey responses, frequency counting provides critical insights into data distribution. In [Google Sheets](#), this process is elegantly handled by combining two powerful native functions: the **=UNIQUE()** function and the **=COUNTIF()** function.

This detailed tutorial outlines the definitive, step-by-step process for generating a complete frequency table, demonstrating how these functions work in concert to deliver clear and actionable insights from any column containing repetitive entries. We will move beyond simple counting by first isolating the distinct elements and then applying conditional logic to tally their occurrences efficiently.

## The Core Methodology: Leveraging UNIQUE() and COUNTIF()

The challenge in frequency counting is twofold: first, identifying \*what\* distinct items exist in the column, and second, determining \*how many times\* each of those items appears. Relying solely on the **=COUNTIF()** function would require manually listing every possible value, a highly impractical task for large datasets. The **=UNIQUE()** function solves the first part of this problem by dynamically extracting a non-redundant list of all entries present in the specified range.

By generating this unique list, we establish the criteria necessary for the second stage. The **=COUNTIF()** function, which is designed to count cells within a range that meet a given condition, can then use each item in the unique list as its counting criterion. This pairing creates a robust and scalable method for automated frequency calculation, ensuring that no value is missed and the process remains fully dynamic if the source data is updated.

Understanding the specific roles of each function is essential for successful implementation. The [UNIQUE\(\) function](#) takes a range as input and outputs an [array](#) containing only the values that appear once in the resulting list. The [COUNTIF\(\) function](#) requires two arguments: the range to be checked (the original data column) and the criterion (the cell containing the unique value).

### Step 1: Preparing Your Dataset for Frequency Analysis

Before we implement any formulas, we must ensure our data is properly structured. For this example, we are using a column of text data--specifically, the names of professional basketball teams--which often contain repetitive entries, making them ideal candidates for frequency analysis. Data cleanliness is paramount; ensure there are no unintended trailing spaces, inconsistent capitalization, or subtle misspellings, as **Google Sheets** treats these variations as distinct unique values.

For the purposes of this demonstration, we will enter the team names into Column A. This column

represents our primary dataset that we wish to analyze. The data preparation stage is simple but critical, as the accuracy of all subsequent calculations relies entirely on the integrity of the source data.

First, let's enter the names for some professional basketball teams in column A. Note the repetitions, which necessitate the counting process:

	A	B	C	D	E
1	<b>Team Name</b>				
2	Hornets				
3	Hawks				
4	Spurs				
5	Spurs				
6	Mavericks				
7	Spurs				
8	Rockets				
9	Nets				
10	Hawks				
11	Mavericks				
12	Rockets				
13	Spurs				
14	Nets				
15	Hawks				
16	Spurs				
17					
18					
19					
20					

It is a best practice to organize your workspace by dedicating specific columns for the unique list and the final count. We will typically place the unique list in an adjacent column (e.g., Column D) and the resulting counts in the column next to it (e.g., Column E) to maintain clarity and separation from the raw input data.

## Step 2: Isolating Distinct Values Using the UNIQUE Function

Once the source data is ready, the next step is to isolate the unique values from the input column. This is achieved instantly using the **=UNIQUE()** function. This function is highly efficient because it handles the entire range in a single operation, automatically spilling the resulting list of unique items into the designated column.

To execute this, select the cell where you want the list to begin (in our example, Cell D2) and input

the function, referencing the entire data column (A2:A20, or simply A:A if your list may grow). The structure of the formula should be: `=UNIQUE(A2:A)`.

Next, let's use the [UNIQUE\(\) function](#) to create a list of every unique team name present in column A, placing the result in Column D:

	A	B	C	D	E
D2				<code>=UNIQUE(A2:A16)</code>	
1	<b>Team Name</b>			<b>Unique Teams</b>	
2	Hornets			Hornets	
3	Hawks			Hawks	
4	Spurs			Spurs	
5	Spurs			Mavericks	
6	Mavericks			Rockets	
7	Spurs			Nets	
8	Rockets				
9	Nets				
10	Hawks				
11	Mavericks				
12	Rockets				
13	Spurs				
14	Nets				
15	Hawks				
16	Spurs				
17					
18					
19					

An important characteristic of this function is its nature as an array formula. When you enter the formula in Cell D2, it automatically populates the subsequent rows (D3, D4, D5, etc.) with the unique values without requiring you to drag the formula down. The resulting range (D2:D) now serves as the dynamic criteria list for the next and final step--the frequency calculation.

### Step 3: Calculating Frequency with the COUNTIF Function

With the unique list established in Column D, we can now proceed to calculate the frequency of each item using the `=COUNTIF()` function. This function checks the original dataset (Column A) against each unique team name listed in Column D. This step requires careful consideration of absolute and relative references to ensure the formula works correctly when copied down the column.

In Cell E2, we input the formula: `=COUNTIF(A:A, D2)`. Here, `A:A` is the range containing all the basketball teams (the range to be counted), and `D2` is the first criterion (the first unique team name). Crucially, the range `A:A` must be treated as an absolute reference, which is naturally achieved by referencing the entire column (A:A) rather than a fixed range (A\$2:A\$20). The criterion cell, `D2`, must remain a relative reference so that when we copy the formula down to E3, E4, and so on, it automatically updates to check D3, D4, etc.

Next, let's use the [COUNTIF\(\) function](#) to count the number of occurrences of each unique team name, placing the results in Column E:

E2 fx =COUNTIF(\$A\$2:\$A\$16, D2)

	A	B	C	D	E
1	<b>Team Name</b>			<b>Unique Teams</b>	<b>Occurrences</b>
2	Hornets			Hornets	1
3	Hawks			Hawks	3
4	Spurs			Spurs	5
5	Spurs			Mavericks	2
6	Mavericks			Rockets	2
7	Spurs			Nets	2
8	Rockets				
9	Nets				
10	Hawks				
11	Mavericks				
12	Rockets				
13	Spurs				
14	Nets				
15	Hawks				
16	Spurs				
17					
18					
19					
20					

Note that we simply copy and pasted the formula from cell E2 to each of the remaining cells in column E that correspond to a unique value in Column D. This manual step is necessary because `=COUNTIF()` is not inherently an array formula, unlike `=UNIQUE()`. If you wished to make this process fully dynamic and avoid dragging the formula, you would need to wrap the `=COUNTIF()` structure within an `=ARRAYFORMULA()` wrapper, which is discussed in the advanced section below.

## Interpreting and Validating the Results

The resulting table, spanning Columns D and E, provides a complete and easily digestible frequency distribution of the original dataset in Column A. This output moves beyond raw data by transforming it into meaningful statistical summary. Column D lists the entire population of distinct teams, while Column E shows their corresponding frequencies.

To validate the results, you should always sum the frequency column (Column E). This sum must precisely match the total number of entries in the original data column (Column A). If the totals do not match, it suggests an error either in the range definition of the **=UNIQUE()** function or in the range definition of the **=COUNTIF()** function. This simple validation step ensures the integrity of your frequency analysis.

From the output generated, we can clearly see the distribution of team names across the dataset. This summary is instantly readable and provides insight into the most frequently occurring teams:

The team name 'Hornets' occurs **1** time in column A, indicating it is a relatively rare entry in this specific dataset.

The team name 'Hawks' occurs **3** times in column A.

The team name 'Spurs' occurs **5** times in column A, making it the most frequent entry in the sample.

The total count for all teams, when summed, should equal the total number of rows populated in Column A.

This method provides a foundational technique for frequency counting, offering unparalleled transparency and control over the counting criteria.

## Advanced Alternatives: Fully Dynamic Counting and the QUERY Function

While the combination of **=UNIQUE()** and manually dragged **=COUNTIF()** formulas is highly effective, advanced users often prefer a fully dynamic solution that requires only a single formula entry. This can be achieved by using the **=ARRAYFORMULA()** wrapper around the **=COUNTIF()** function, or by using more advanced aggregation functions like the [QUERY function](#).

One highly efficient single-cell solution involves using the **QUERY** function, which processes data using SQL-like syntax. This function can group and count data simultaneously, eliminating the need for separate **=UNIQUE()** and **=COUNTIF()** steps. The formula structure for achieving the same result would be: `=QUERY(A:A, "select A, count(A) group by A label count(A) 'Frequency' ", 1)`. This single formula automatically generates both the unique list (the grouped items) and their corresponding frequencies, making it the most compact and powerful solution for frequency distribution generation, particularly for very large spreadsheets.

Another dynamic approach involves using **ARRAYFORMULA**. Since **COUNTIF** normally only processes one cell at a time, wrapping it in **ARRAYFORMULA** allows it to process the entire list of unique criteria at once, returning multiple results without manual dragging. The structure requires the unique list to be generated first, and then the array formula applied to the count column: `=ARRAYFORMULA(IF(D2:D="", "", COUNTIF(A:A, D2:D)))`. While slightly more complex than the **QUERY** approach, this method provides flexibility if you need to perform additional calculations or manipulations on the unique list before counting.

These advanced techniques are particularly valuable when building dashboards or reports where the underlying data changes frequently, as they ensure the frequency table updates automatically without any manual intervention.

## Additional Resources

For users looking to deepen their understanding of spreadsheet functions and data manipulation in **Google Sheets**, the following resources provide comprehensive documentation and further examples regarding the application of conditional and array-based counting methods. Mastering these tools enhances your capability to perform complex [data analysis](#) tasks efficiently.