

How to Count with Criteria in Google Sheets Using SUBTOTAL and COUNTIF

Authored by
stats writer

January 17, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Count with Criteria in Google Sheets Using SUBTOTAL and COUNTIF*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=126480>

1. Introduction: Mastering Conditional Counting in Google Sheets

The ability to perform sophisticated data analysis is paramount in modern spreadsheet software. While functions like `COUNTIF` or its plural counterpart, **COUNTIFS**, are excellent for tallying cells based on specific criteria, they often fall short when dealing with filtered data. When you filter a dataset in Google Sheets, standard counting functions continue to evaluate all rows, including those hidden by the filter. This fundamental limitation necessitates the integration of the powerful **SUBTOTAL** function.

The **SUBTOTAL** function is uniquely designed to operate exclusively on visible cells within a range. By combining **SUBTOTAL** with conditional counting logic, analysts gain precise control over their metrics, ensuring that counts reflect only the currently displayed subset of data. This technique is particularly valuable when managing large, dynamic datasets where filtering and slicing the information are regular requirements. Understanding how to link these two functions--a conditional counter and a visibility assessor--is key to accurate spreadsheet reporting.

This detailed guide will walk through the exact steps required to leverage the **SUBTOTAL** function alongside **COUNTIFS** (which we use because it supports multiple criteria, essential for combining a data criterion with the visibility criterion). We will use a practical example involving basketball player statistics to illustrate how a strategically placed helper column enables this synergy, allowing you to count only the visible rows that satisfy predetermined conditions. This methodology transforms basic counting into a dynamic, filter-aware operation, providing highly relevant insights into your segmented data.

2. Understanding the SUBTOTAL Function in Context

The **SUBTOTAL** function is highly versatile, capable of performing various calculations--including sums, averages, maximums, and counts--but its distinctive feature is its ability to recognize and ignore rows that have been hidden manually or, more importantly, hidden by a filter. This behavior is controlled by the function's first argument, known as the `function_code`. There are two primary groups of codes: codes 1 through 11 include manually hidden rows in the calculation, while codes 101 through 111 exclude both filtered and manually hidden rows.

For our specific goal of counting only visible rows in a filtered list, we must utilize a code from the 100-series. The most relevant codes for counting are 102 (for **COUNT**, counting numbers) and 103 (for **COUNTA**, counting non-blank values). Since our helper column needs to assign a visible/invisible status regardless of the data type in the corresponding row, using **COUNTA** (code **103**) is the standard and most reliable approach. This code tells Google Sheets, "Count all non-empty, visible cells in this range."

When **SUBTOTAL(103, C2)** is applied to a single cell (C2 in our example), the result is either 1 (if

the row is visible) or 0 (if the row is hidden by a filter). This output is critical because it generates a binary marker in the helper column. This binary marker--1 for visible, 0 for invisible--serves as the necessary visibility criteria that we will feed into the **COUNTIFS** function, effectively bridging the gap between visibility assessment and conditional counting. This preparation is the cornerstone of the entire process.

3. Step 1: Initial Data Entry and Preparation

The first essential step involves setting up the data structure. For this demonstration, we will use a small dataset representing basketball player statistics. This dataset should include various categorical fields that can be used for filtering and conditional counting, such as Player Name, Position, and Conference. A clean, correctly formatted dataset is the foundation for accurate data analysis in any spreadsheet environment.

We begin by entering the raw data into our Google Sheets worksheet. Ensure that the first row is used for informative headers (e.g., Player, Position, Conference) as these headers are crucial for applying filters correctly in subsequent steps. The following image represents the initial dataset, which includes players spanning different positions and conferences (East and West).

It is important to recognize that at this stage, any standard counting function like **COUNTIF** would count all eight players, regardless of how we might intend to filter them later. The goal is to set up a system where, if we filter for only the 'West' Conference, a conditional count for 'Guard' will only return the count from the visible, 'West' players.

First, let's enter the following dataset that contains information about various basketball players:

	A	B	C	D	
1	Conference	Position	Points		
2	West	Guard	12		
3	West	Forward	22		
4	East	Guard	24		
5	West	Center	30		
6	East	Center	25		
7	East	Guard	24		
8	West	Guard	19		
9	West	Forward	16		
10	East	Forward	22		
11	East	Guard	28		
12					
13					
14					
15					

4. Step 2: Implementing the Helper Column for Visibility Tracking

The key innovation in this method is the introduction of a helper column, which acts as a flag for visible rows. This column will hold the results of the **SUBTOTAL** function, specifically utilizing the function_code **103**, which corresponds to **COUNTA** for visible cells. We will label this column clearly, perhaps as "Visible Flag" or simply "Helper," to maintain clarity in the spreadsheet structure and facilitate easy reference in the final counting formula.

We must enter the specific SUBTOTAL formula into the second row of this new helper column. Assuming our data starts in row 2, and we place the helper column in column D, the formula will be placed in cell **D2**. The reference within the formula should point to a non-empty cell in the corresponding row (e.g., cell C2, which contains the conference data). Since we are using **103 (COUNTA)**, any non-empty cell in that row will suffice for checking visibility, as the function only cares if the row itself is displayed.

Next, type the following formula into cell **D2**:

=SUBTOTAL(103, C2)

Once the formula is entered into **D2**, it should be copied down to the remaining cells corresponding to the data rows (D3, D4, D5, etc.). Initially, before any filtering is applied, every cell in the helper column will display the value **1**, indicating that every row is currently visible. This value will

dynamically change to **0** only when a filter is activated and the specific row is hidden from view. This dynamic flag is the critical component that integrates visibility awareness into the counting process.

Then click and drag this formula down to each remaining cell in column D:

D2 =SUBTOTAL(103, C2)

	A	B	C	D
1	Conference	Position	Points	Helper
2	West	Guard	12	1
3	West	Forward	22	1
4	East	Guard	24	1
5	West	Center	30	1
6	East	Center	25	1
7	East	Guard	24	1
8	West	Guard	19	1
9	West	Forward	16	1
10	East	Forward	22	1
11	East	Guard	28	1
12				
13				
14				
15				

5. Step 3: Applying Filters to Isolate Data Segments

With the helper column in place, we are now ready to activate the filtering capabilities within Google Sheets. Filtering is the mechanism that triggers the SUBTOTAL function to output zeros for hidden rows, thus demonstrating its utility for targeted data analysis. The process for creating a filter is straightforward and standardized across most spreadsheet platforms.

To add a filter to this data, we must first highlight the entire range, including the headers and the newly created helper column. In our example, this range is typically **A1:D11**. After selecting the range, navigate to the **Data** tab in the Google Sheets menu and select the **Create a filter** option. This action places a small filter icon next to each column header, enabling interactive data segmentation.

For our scenario, let's establish a specific filter criteria. We want to analyze only players belonging to the 'West' Conference. Click the filter icon next to the **Conference** column header. In the filtering dialogue box, deselect all options except for **West**. Applying this filter hides all rows where the

Conference value is 'East'. Crucially, observe the helper column (Column D): the values in D turn to **0** for the hidden rows, while the visible 'West' rows retain the value **1**.

To add a filter to this data, we can highlight cells **A1:D11**, then click the **Data** tab, then click **Create a filter**.

Then click the filter icon next to the **Conference** column and filter the data to only show rows where the value in the **Conference** column is equal to **West**:

	A	B	C	D
1	Conference ▼	Position ▼	Points ▼	Helper ▼
2	West	Guard	12	1
3	West	Forward	22	1
5	West	Center	30	1
8	West	Guard	19	1
9	West	Forward	16	1
12				
13				
14				
15				
16				
17				
18				
19				

6. Step 4: Using COUNTIFS to Count Visible Rows Conditionally

The final and most critical step involves using the **COUNTIFS** function to perform the conditional count, utilizing the binary visibility flag we generated in the helper column. We are no longer relying on the simpler COUNTIF, but rather its multi-criteria sibling, COUNTIFS, because we now have two conditions that must be met simultaneously: the data criterion (e.g., Position = Guard) AND the visibility criterion (Helper Column = 1).

Suppose our objective is to determine how many players in the currently filtered data subset (the 'West' Conference) play the **Guard** position. The **COUNTIFS** function structure requires pairs of ranges and criteria. Our formula will incorporate two such pairs: first, checking the Position column for "Guard," and second, checking the Helper column for the value 1.

The resulting formula effectively isolates the desired count. The function checks every row in the

defined range. Only if a row contains "Guard" in the Position column (B2:B9) **AND** the value 1 in the Helper column (D2:D9) will that row be included in the final tally. This ensures that only players who are both Guards **and** visible (i.e., belonging to the West Conference) are counted, providing a precise and context-aware metric.

Now suppose that we would like to count the number of values in the **Position** column that are equal to **Guard**.

=COUNTIFS(B2:B9,"Guard",D2:D9,"1")

The following screenshot shows how to use this formula in practice:

B13 **fx** =COUNTIFS(B2:B9,"Guard",D2:D9,"1")

	A	B	C	D
1	Conference ▼	Position ▼	Points ▼	Helper ▼
2	West	Guard	12	1
3	West	Forward	22	1
5	West	Center	30	1
8	West	Guard	19	1
9	West	Forward	16	1
12				
13		2		
14				
15				
16				
17				
18				

7. Verifying the Results and Practical Applications

Upon executing the combined COUNTIFS and SUBTOTAL formula, the result returned is **2**. This output must be rigorously verified against the filtered data currently visible on the screen. By manually inspecting the rows visible after applying the 'West' Conference filter, we can confirm that only two rows correspond to players holding the 'Guard' position. This verification confirms the accuracy and effectiveness of the helper column methodology.

The power of this technique lies in its adaptability. If we were to change the filter--for instance, filtering the data to show only the 'East' Conference--the helper column values for the newly visible

'East' rows would instantly revert to 1, and the formula would automatically recalculate the number of Guards in the East, without any modification to the core **COUNTIFS** formula itself. This dynamic functionality is a significant benefit for users engaged in continuous data exploration and reporting.

This approach is highly valuable across numerous professional disciplines. Whether performing inventory management, tracking customer service metrics, or conducting financial statement analysis, the ability to generate accurate conditional counts on segmented data ensures reliable reporting. By mastering the synergy between **SUBTOTAL(103)** and **COUNTIFS**, users can elevate their spreadsheet skills, moving beyond static calculations to achieve true, filter-aware data analysis within Google Sheets. The formula correctly returns a count of **2**, demonstrating the method's precision, which we verified by looking at the filtered data, showing there are indeed 2 rows in the **Position** column with a value of **Guard**.