

How to Calculate Conditional Subtotals with SUBTOTAL and SUMIF in Google Sheets

Authored by
stats writer

February 1, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Calculate Conditional Subtotals with SUBTOTAL and SUMIF in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=129037>

The strategic pairing of the SUBTOTAL function with the conditional power of **SUMIF** or **SUMIFS** in Google Sheets enables a highly efficient method for conditional data aggregation. This combination is essential for filtering a dataset and then calculating sums exclusively from the rows that remain visible and meet additional criteria. This technique moves beyond standard summation, providing a dynamic and accurate way to obtain subtotal figures, which is invaluable when dealing with substantial datasets that require constant filtering and re-evaluation. Moreover, this approach guarantees that the resulting subtotal values update dynamically as the underlying data or the applied filters change, enhancing the overall utility for sophisticated data analysis and comprehensive data organization.

Google Sheets: Use SUBTOTAL with SUMIF

The Challenge of Conditional Subtotaling in Google Sheets

In many data analysis scenarios, users need to calculate summaries--such as totals or averages--only on the data that is currently visible after applying a filter. The native **SUMIF** function, while excellent for calculating sums based on criteria, unfortunately operates on the entire specified range, regardless of whether rows have been hidden by a filter. This limitation means that simply using **SUMIF** after filtering your data will still include the values from the hidden rows, rendering the result inaccurate for a visible-only subtotal.

To overcome this significant challenge, we must integrate the functionality of the SUBTOTAL function. The **SUBTOTAL** function is unique because it is designed to recognize and respect filters, calculating aggregates only on visible cells. However, **SUBTOTAL** itself does not possess the inherent capability to evaluate conditional criteria, such as "only sum if the position is 'Guard'." Therefore, a creative architectural solution involving a **helper column** is necessary to bridge the gap between these two powerful functions.

This tutorial details a robust, step-by-step methodology using a **helper column** to successfully combine the visibility awareness of SUBTOTAL with the multi-criteria capability of SUMIFS, allowing you to sum only the visible rows that meet specific conditions.

Understanding the Core Functions: SUBTOTAL and SUMIFS

Before diving into the implementation, it is essential to appreciate how the core functions involved operate. The SUBTOTAL function requires a function code (ranging from 1 to 11 for including hidden values, or 101 to 111 for ignoring hidden values) and a reference range. For our purpose--determining if a row is visible--we will use function code **103**, which corresponds to **COUNTA** (count of non-blank cells), specifically set to ignore hidden rows. By applying **SUBTOTAL(103, X)** to a single cell (X), the result will be 1 if the cell is visible and 0 if the cell is hidden by a filter.

Conversely, the **SUMIFS** function is the evolution of the simpler **SUMIF**. While **SUMIF** handles only one condition, **SUMIFS** allows you to specify multiple conditions that must all be met for a value to be included in the sum. The syntax for **SUMIFS** is generally: `=SUMIFS(sum_range, criteria_range1, criteria1, criteria_range2, criteria2, ...)`. We will use this multi-criteria capability to check two conditions simultaneously: the functional criteria (e.g., position equals 'Guard') AND the visibility criteria (the helper column equals 1).

Step 1: Establishing the Base Dataset

The first step in implementing this solution is to organize your data clearly within the spreadsheet. For this example, we utilize a typical dataset containing information about basketball players, including their name, position, conference, and points scored. A clean, structured dataset is paramount for successful filtering and aggregation operations.

We begin by entering the data shown below, ensuring the columns are labeled appropriately for clarity and ease of filtering. The dataset includes columns A through C, detailing player attributes necessary for the subsequent filtering and summation tasks.

	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					

This raw data structure provides the foundation upon which we will build the visibility tracking and conditional summation logic.

Step 2: Implementing the Visibility Check using a Helper Column

To accurately identify which rows are visible after a filter is applied, we must introduce a **helper column**. This column acts as a binary flag, marking rows with a '1' if they are visible and relevant, and a '0' if they are hidden. This flag is the crucial link between the filtering operation and the conditional summation.

In this example, we designate Column D as our helper column. In cell **D2**, enter the following specific formula:

=SUBTOTAL(103,C2)

Function code **103** instructs the SUBTOTAL function to perform a non-blank count while ignoring rows hidden by filtering. When applied to a single cell reference (C2 in this instance, though any non-blank cell in that row would work), the result is 1 if the row is visible and 0 if the row is hidden. After entering the formula into D2, drag the fill handle down to apply this formula to every remaining cell in column D. This action populates the entire helper column, establishing a real-time visibility tracker for every row in the dataset.

	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				

Step 3: Applying Filters for Data Isolation

With the helper column now tracking visibility, we can proceed to apply standard filters to the dataset. These filters define the initial group of rows that the conditional sum will analyze. This isolation step simulates a common real-world scenario where data must first be narrowed down before specific metrics are calculated.

To apply filtering, highlight the range encompassing both the data and the new helper column (cells **A1:D11**). Navigate to the **Data** tab in [Google Sheets](#) and select **Create a filter**. Once the filter is active, click the filter icon next to the **Conference** column header.

For this demonstration, we will filter the data to only display rows where the value in the **Conference** column is strictly equal to **West**. This action immediately hides all 'East' conference players, and crucially, the formulas in the helper column (Column D) immediately update, changing the values to 0 for the hidden rows and retaining 1 for the visible 'West' conference rows.

	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				

Only the visible data--the West conference players--is now primed for the final conditional aggregation step.

Step 4: Combining Visibility and Criteria using SUMIFS

The final step involves using the highly versatile **SUMIFS** function to perform the conditional subtotal calculation. This function is perfectly suited here because it allows us to impose multiple

criteria simultaneously: one criterion for the functional condition (e.g., Position = Guard) and a second criterion for visibility (Helper Column = 1).

Suppose our objective is to calculate the sum of **Points** scored, but only for players who meet two conditions: they are in the visible filtered dataset (West Conference) AND their **Position** is **Guard**. We must structure the **SUMIFS** function as follows:

Sum Range: The column containing the values we wish to sum (C2:C11, the Points column).

Criteria Range 1: The column containing the functional criterion (B2:B11, the Position column).

Criteria 1: The specific functional value required ("Guard").

Criteria Range 2: The column containing the visibility criterion (D2:D11, the Helper Column).

Criteria 2: The specific visibility value required ("1", indicating the row is visible).

The resulting formula is:

=SUMIFS(C2:C11,B2:B11,"Guard",D2:D11,"1")

Executing this formula effectively calculates the sum of points for West Conference players who are Guards, ignoring all East Conference players (because their helper column value is 0) and ignoring all non-Guard positions within the West Conference data (because they fail the primary positional criterion).

B13 | fx =SUMIFS(C2:C11, B2:B11, "Guard", D2:D11, "1")

	A	B	C	D	E
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			31		
14					
15					
16					
17					

Analyzing the Results and Verification

As illustrated in the preceding step, the formula successfully returns a sum of **31**. This result represents the total points scored only by the 'Guard' positions within the filtered 'West' conference players. This confirms that the combined logic of the SUBTOTAL (via the helper column) and SUMIFS functions has worked precisely as intended.

We can visually verify this outcome by examining the filtered data, which only displays the West Conference players. If we look specifically at the rows where the Position is 'Guard' (rows 3, 5, and 11 in the original numbering scheme), we find points values of 12, 11, and 8, respectively. The sum of these values is $12 + 11 + 8$, which equals **31**.

B2:C2 ▾ *fx* Guard

	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		31		
14				
15				
16				
17				
18				

This verification confirms the accuracy of the complex calculation, demonstrating the power of using a visibility-tracking helper column to achieve conditional subtotals in Google Sheets. This method is highly flexible and can be adapted to any combination of filters and conditional criteria required for sophisticated data analysis.