

# How do I Count Filtered Cells with Text in Excel?

Authored by  
**stats writer**

November 18, 2025

## RECOMMENDED CITATION

stats writer (2025). *How do I Count Filtered Cells with Text in Excel?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=95939>

## The Challenge of Counting Filtered Data in Excel

When working with large datasets in Excel, applying filters is a standard procedure used to isolate and analyze specific subsets of information. However, traditional counting functions often fail to accurately tally only the visible cells after a filter has been applied. This limitation becomes particularly problematic when you need to count how many of those visible cells contain specific data types, such as text.

Fortunately, there is a powerful and reliable formula designed to overcome this challenge. This method combines several advanced Excel functions, including SUBTOTAL, SUMPRODUCT, and ISTEXT, allowing for precise counting of visible entries that match specific criteria. Understanding this complex function is key to mastering data analysis in filtered environments.

This sophisticated technique ensures that your analysis reflects only the data currently displayed on the worksheet, providing accurate summary statistics for your filtered view. This is essential for reporting and decision-making where hidden data should not influence the results. We will explore the components of this formula in detail and demonstrate its application using a practical example involving employee sales data.

## The Essential Formula for Visible Text Counts

To accurately count the number of filtered cells in a specified range that contain text values, you must employ the following robust formula structure. This structure is specifically engineered to interact correctly with Excel's filtering mechanism, bypassing the limitations of simple counting functions.

**=SUMPRODUCT(SUBTOTAL(103, INDIRECT("A"&ROW(A2:A13))), --(ISTEXT(A2:A13)))**

In this specific instance, the formula is configured to count the visible cells containing text within the range **A2:A13**. This formula operates by creating two separate arrays: one array that identifies which rows are visible, and another array that identifies which cells contain text. The SUMPRODUCT function then multiplies these two arrays together, summing up only those entries where both conditions (visible AND contains text) are true.

While this formula may appear daunting at first glance, its efficiency and accuracy are unmatched when dealing with dynamic, filtered datasets. The power lies in the combination of the specific function arguments, particularly the use of SUBTOTAL with the function number 103, which is crucial for identifying visibility. We will break down each function used in the array calculation to fully understand its role.

## Deconstructing the Core Components: SUBTOTAL(103)

The success of this counting method hinges on the intelligent use of the SUBTOTAL function. Unlike standard summary functions (like SUM or COUNT), SUBTOTAL is designed to ignore values in rows that have been hidden by a filter. The key to counting text specifically, rather than numerical data, lies in the function number argument.

The SUBTOTAL function requires two main arguments: the function code and the range. Function codes 1 through 11 include hidden values, whereas codes 101 through 111 ignore values in hidden rows. We use the code 103, which corresponds to the COUNTA function (counting non-blank cells), but critically, it only counts those cells that are visible.

The structure of the visibility check is complex: SUBTOTAL(103, INDIRECT("A"&ROW(A2:A13))). By combining `ROW()` with `INDIRECT()`, we force SUBTOTAL to calculate a count for \*each individual cell\* in the range A2:A13, rather than the entire range at once. If a cell is visible, the calculation returns 1; if it is hidden by a filter, it returns 0 (or an empty array element depending on the Excel version's handling of array processing). This sequence effectively creates a boolean array representing the visibility status of every row in the target range.

## Leveraging SUMPRODUCT and Array Operations

The SUMPRODUCT function is the engine that drives this formula. Its primary role is to multiply corresponding elements in two or more arrays and then return the sum of those products. In this specific scenario, we are multiplying the Visibility Array (created by the SUBTOTAL/INDIRECT mechanism) by the Text Identification Array (created by the ISTEXT function).

When the two arrays are multiplied element by element, the resulting array contains a 1 only when a cell meets \*both\* criteria: it must be visible (1 from the SUBTOTAL array) AND it must contain text (1 from the ISTEXT array). If a cell is visible but contains a number, the result is  $1 * 0 = 0$ . If a cell is hidden but contains text, the result is  $0 * 1 = 0$ . Only visible text cells yield  $1 * 1 = 1$ .

The final step performed by SUMPRODUCT is summing all the resulting 1s, which gives us the precise count of visible, text-containing cells. This ability to handle complex conditional logic across large arrays makes SUMPRODUCT indispensable for advanced Excel array formulas, eliminating the need for the traditional Ctrl+Shift+Enter entry method often required for older array calculations.

## Identifying Text Values with ISTEXT

The second essential component of the counting formula is the expression `--(ISTEXT(A2:A13))`. This section is responsible for evaluating every cell in the specified range and determining whether

its content is classified as text.

The ISTEXT function returns a boolean value: TRUE if the cell contains text, and FALSE otherwise. However, SUMPRODUCT requires numerical input (0s and 1s) for its multiplication operation.

This is where the double negative (``) comes into play. Applying the double unary operator forces Excel to convert the boolean results (TRUE/FALSE) into their numerical equivalents (1/0). Specifically, TRUE becomes 1, and FALSE becomes 0. This conversion process is vital, as it standardizes the output into an array of binary values that the SUMPRODUCT function can then successfully multiply against the visibility array.

Understanding the role of the double negative is fundamental to writing effective conditional array formulas in Excel. It is an efficient way to coerce logical values into numerical values, ensuring that the multiplication yields accurate results only for those records that satisfy the text criterion.

### **Practical Application: Setting up the Dataset Example**

To illustrate the power of this formula, let us consider a common business scenario involving sales tracking. Suppose we are analyzing a dataset that contains information regarding total sales generated by various employees within an organization. Our goal is to count the employee names (which are text values) that correspond to specific sales thresholds after applying a filter.

The initial dataset structure is displayed below. Column A lists the employee names, which are the text values we intend to count, and Column B contains the numerical sales figures. The entire range of data spans from A1 to B13.

	A	B	C	D	E
1	<b>Employee</b>	<b>Sales</b>			
2	50094	22			
3	50086	14			
4	Michael	15			
5	Tony	13			
6	51009	24			
7	Andy	31			
8	49006	15			
9	51002	19			
10	52007	18			
11	45966	11			
12	Jim	18			
13	Craig	24			
14					
15					
16					
17					
18					

Now, we proceed to filter this data. For our analysis, let us assume we are only interested in employees whose sales performance exceeded a value of 15. We apply a filter to the **Sales** column to show only rows where the value is greater than 15. This action immediately hides several rows that do not meet the criteria, leaving only the high-performing employees visible.

The visual result of applying the filter is shown in the subsequent image. Notice that while the original dataset had 12 rows of data (A2:A13), the filtered view now displays only a subset of these rows.

	A	B	C	D	E
1	<b>Employ</b>	<b>Sales</b>			
2	50094	22			
6	51009	24			
7	Andy	31			
9	51002	19			
10	52007	18			
12	Jim	18			
13	Craig	24			
14					
15					
16					
17					
18					
19					
20					
21					

## The Failure of Standard Counting Functions

Before implementing the advanced solution, it is important to understand why standard Excel functions cannot correctly solve this problem. If we wanted to count the total number of cells containing text in the **Employee** column (A2:A13) without any filtering, we would typically use a straightforward function like COUNTIF.

A standard formula to count text in the range A2:A13 is:

**=COUNTIF(A2:A13, "\*")**

The asterisk (\*) acts as a wildcard character, instructing Excel to count any cell that contains text. When applied to the original, unfiltered dataset, this formula would correctly return 12, as all 12 cells contain employee names (text).

However, applying this exact formula to our filtered data produces an incorrect result relative to the visible data. Standard counting functions like COUNTIF or COUNTA operate on the underlying, entire range, regardless of whether rows have been hidden by a filter. They do not possess the inherent intelligence to respect the filter criteria.

As shown in the image below, if we attempt to use the simple COUNTIF formula on the filtered dataset, the formula returns 12, which is the total number of employee names in the original

dataset. This result clearly misrepresents the count of visible text entries, which is significantly lower. This discrepancy necessitates the use of the complex array formula featuring SUBTOTAL(103).

	A	B	C	D	E	F
1	Employee	Sales		Cells with Text		
2	50094	22		5		
6	51009	24				
7	Andy	31				
9	51002	19				
10	52007	18				
12	Jim	18				
13	Craig	24				
14						
15						
16						
17						
18						
19						

## Implementing the Advanced Solution

To obtain the accurate count of only the visible text cells in the **Employee** column after filtering for sales greater than 15, we must utilize the advanced array mechanism detailed previously. The formula ensures that every row is checked both for visibility and for content type.

The required formula to count the number of filtered cells with text in the range A2:A13 is:

**=SUMPRODUCT(SUBTOTAL(103, INDIRECT("A"&ROW(A2:A13))), --(ISTEXT(A2:A13)))**

When entered into a cell--for instance, cell D1--this formula immediately calculates the correct count based on the current filtering state of the dataset. Since SUMPRODUCT inherently handles array processing, no special entry method is needed.

The following screenshot demonstrates the application of this formula within the filtered workbook, showing the correct calculated output:

	A	B	C	D	E	F	G
1	Employee	Sales		Cells with Text			
2	50094	22		3			
6	51009	24					
7	Andy	31					
9	51002	19					
10	52007	18					
12	Jim	18					
13	Craig	24					
14							
15							
16							
17							
18							
19							
20							
21							
22							

## Verifying the Results and Conclusion

As demonstrated in the final implementation, the formula correctly returns a value of **3**. This outcome accurately reflects the number of visible rows in the dataset that contain text values in the specified column (A2:A13).

We can easily verify this result by manually reviewing the visible rows in the Employee column after the filter for Sales > 15 was applied. The visible employee names are: **Andy** (Row 3), **Jim** (Row 6), and **Craig** (Row 12). Since these are the only three rows meeting both the visibility and the text content criteria, the formula's calculation is validated.

In conclusion, while Excel provides numerous built-in functions for counting data, specific analytical requirements, such as counting based on visibility and data type simultaneously, often demand a more sophisticated approach. The combination of SUMPRODUCT, SUBTOTAL(103), and ISTEXT provides the robust solution necessary for accurate text counting in dynamically filtered spreadsheets, empowering users to perform reliable data segmentation and reporting.