

How to Filter by List of Values in Google Sheets?

Authored by
stats writer

November 22, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Filter by List of Values in Google Sheets?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=100075>

Managing and analyzing large volumes of data within Google Sheets often requires dynamic filtering capabilities. A common operational challenge arises when users need to filter a master dataset against a specific, predefined list of values, rather than just a single criterion. While manual filtering is possible, it is inefficient and prone to error when dealing with dynamic lists or frequent updates.

Fortunately, Google Sheets provides powerful function combinations to achieve this complex task programmatically. The most effective approach involves leveraging the core strength of the FILTER function in conjunction with the versatility of the COUNTIF function. This powerful synergy allows users to quickly and accurately extract relevant rows from a broad range of data based entirely on whether a key column's value appears in the specified criteria list.

Understanding this methodology is crucial for advanced data manipulation. Furthermore, once this foundation is established, these filtered results can be easily combined with other sophisticated functions like SORT, QUERY, or UNIQUE to further refine, aggregate, or visualize the resulting data, thereby maximizing the utility of your spreadsheet operations.

Understanding the Core Filtering Syntax

To successfully filter a master dataset using a separate list of values in Google Sheets, we rely on a specific syntax that integrates the array-processing nature of the FILTER function with the conditional counting ability of COUNTIF. This combination is highly efficient because COUNTIF is capable of returning an array of results when provided with a range input, which perfectly satisfies the criteria argument requirement of the FILTER function.

The standard and most efficient syntax employed for filtering a dataset based on a list of values in Google Sheets is demonstrated below:

```
=FILTER(A2:C11,COUNTIF(E2:E5,A2:A11))
```

Deconstructing the Formula Logic

To truly appreciate the power of this formula, it is essential to understand how the nested functions interact. The FILTER function requires two primary arguments: the range of data to return, and one or more criteria arrays (arrays of TRUE or FALSE values) that correspond to the rows in the range. Only rows where the criteria evaluate to TRUE are returned.

In the formula provided above, the range specified for filtering is **A2:C11**, which represents the entire block of data we wish to extract from. The crucial evaluation happens within the second argument, which is handled entirely by the COUNTIF function: `COUNTIF(E2:E5, A2:A11)`. Here,

E2:E5 is the list of permitted values (the criteria list), and **A2:A11** is the column in the primary dataset that we are checking against the criteria list.

The COUNTIF function works by iterating through every cell in the range **A2:A11** and checking how many times that cell's value appears in the criteria range **E2:E5**. If the value is found, COUNTIF returns a number greater than zero (usually 1, unless duplicates exist in the criteria list). If the value is not found, it returns 0. Since FILTER treats any non-zero number as TRUE and zero as FALSE, this results in a perfect array of Boolean criteria that dictates exactly which rows from **A2:C11** should be included in the final output.

Specifically, this particular formula filters the cells in the range **A2:C11** to only return the rows where corresponding values in the key column **A2:A11** successfully match and contain a value from the designated filtering list located in the range **E2:E5**. This method is exceptionally clean and scalable for large datasets.

The following step-by-step example demonstrates how to apply this robust syntax in a practical scenario, ensuring clarity on data structure and function placement.

Step 1: Structuring and Entering the Source Data

The first critical step in any data manipulation task is ensuring the source data is correctly structured and accurately entered into Google Sheets. For this demonstration, we will use a hypothetical dataset concerning basketball players. Proper data structure involves placing headers in the first row and maintaining consistent data types within columns, which is essential for successful filtering operations.

In this example, our source data spans columns A, B, and C, starting from row 2. The critical column we will be filtering against (the primary key column) is Column A, which contains the "Team" names. It is important to verify that the data in the key column is clean and matches the expected format of the criteria list we will define later, avoiding discrepancies such as extra spaces or inconsistent capitalization (though the FILTER function is largely case-insensitive, best practice dictates clean data).

Let's begin by accurately entering the following sample dataset into Google Sheets. This data includes information on Player Name, Team, and Points Scored:

	A	B	C	D	E
1	Team	Points	Assists		
2	Mavericks	22	4		
3	Nets	20	6		
4	Heat	26	5		
5	Kings	35	8		
6	Magic	34	8		
7	Spurs	19	9		
8	Rockets	15	12		
9	Pacers	18	5		
10	Pistons	26	6		
11	Hornets	10	11		
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					

This initial structure (A2:C11) forms the basis of our filtering operation. Every row within this range is a potential candidate for inclusion in the final filtered report, pending the evaluation performed by our criteria list.

Step 2: Defining the Filtering List of Values

After establishing the source data, the next step involves defining the specific set of values that we want to use as our filter criteria. This list should be placed in a separate, easily referenceable location within the spreadsheet. Using a designated range for the criteria list allows for dynamic updates; whenever the list changes, the filtering formula automatically reflects the new requirements without needing manual modification.

For our basketball example, let's define a list of specific team names that we wish to extract from the larger dataset. We will place this criteria list in a dedicated range, specifically cells E2 through E5:

	A	B	C	D	E
1	Team	Points	Assists		Filter for These Teams
2	Mavericks	22	4		Nets
3	Nets	20	6		Kings
4	Heat	26	5		Magic
5	Kings	35	8		Hornets
6	Magic	34	8		
7	Spurs	19	9		
8	Rockets	15	12		
9	Pacers	18	5		
10	Pistons	26	6		
11	Hornets	10	11		
12					
13					
14					
15					
16					
17					
18					
19					
20					

The values in this range (E2:E5) serve as the lookup targets for the COUNTIF function. Only rows from the main dataset whose corresponding Team value (Column A) appears in this list will be retained. It is crucial to ensure that the formatting (text/number) of the values in this criteria range matches the formatting in the column being checked (A2:A11).

By keeping the criteria list separate from the primary data, we achieve greater flexibility. This setup is particularly beneficial when managing complex projects where filtering requirements frequently change or need to be sourced from an auxiliary sheet or calculation.

Step 3: Implementing the Combined FILTER and COUNTIF Formula

With both the source data and the criteria list defined, we can now implement the final filtering formula. The result of this formula, the filtered dataset, will spill dynamically into the sheet, starting from the cell where the formula is entered. For organizational purposes, it is standard practice to place the output a few rows below the original data or on a separate tab.

In this example, we will type the complete formula into cell **A14**. This tells Google Sheets to display the filtered results starting at that position:

=FILTER(A2:C11,COUNTIF(E2:E5,A2:A11))

Upon execution, the COUNTIF array is calculated first. For instance, if the value in A2 ("Lakers") is found in E2:E5, COUNTIF returns 1 for that row. If the value in A3 ("Bulls") is not found, COUNTIF returns 0. The resulting array of 1s and 0s is passed to FILTER, which then extracts only the rows corresponding to the 1s (TRUE values).

The following screenshot visually confirms the practical implementation of this highly effective formula and the resulting output:

A14	fx	=FILTER(A2:C11,COUNTIF(E2:E5,A2:A11))			
	A	B	C	D	E
1	Team	Points	Assists		Filter for These Teams
2	Mavericks	22	4		Nets
3	Nets	20	6		Kings
4	Heat	26	5		Magic
5	Kings	35	8		Hornets
6	Magic	34	8		
7	Spurs	19	9		
8	Rockets	15	12		
9	Pacers	18	5		
10	Pistons	26	6		
11	Hornets	10	11		
12					
13					
14	Nets	20	6		
15	Kings	35	8		
16	Magic	34	8		
17	Hornets	10	11		
18					
19					
20					
21					
22					
23					

As clearly demonstrated, the resulting filtered dataset, starting in cell A14, exclusively includes rows where the Team names from the original source data (A2:C11) successfully matched one of the values listed in our criteria range (E2:E5). Rows corresponding to teams not on the criteria list, such as the "Bulls" or "Knicks," have been automatically and accurately omitted from the final

report.

Key Operational Considerations and Formula Behavior

While the combination of FILTER and COUNTIF is robust, a few operational details must be kept in mind to ensure expected results. These considerations pertain to data matching, case sensitivity, and future expansion.

One of the inherent advantages of using the FILTER function within Google Sheets is its handling of text comparisons. Unlike some database environments or other spreadsheet applications, the FILTER function is designed to be case-insensitive by default when performing comparisons. This feature significantly simplifies filtering tasks involving text data, as users do not have to worry about mismatched capitalization between the source data and the criteria list.

Note: The **FILTER** function is case-insensitive. This means if you intentionally type "nets" (lowercase) instead of "Nets" (proper case) in your criteria list, the function will still correctly identify and return the rows corresponding to the "Nets" team in the original data column. This behavior ensures that minor input errors or variations in case do not prevent successful data retrieval.

For scenarios requiring strict case sensitivity--for instance, distinguishing between product codes 'A-10' and 'a-10'--alternative functions such as FILTER(A2:C11, EXACT(A2:A11, E2:E5)) combined with array formulas might be required, although the COUNTIF approach remains the standard, user-friendly method for general list filtering.

Expanding Functionality: Handling Dynamic Ranges

For advanced spreadsheet management, relying on fixed cell ranges (like A2:C11) can be restrictive, particularly when data is constantly appended to the bottom of the source dataset. To make the filtering process fully dynamic and future-proof, it is highly recommended to use open-ended range references.

Instead of referencing **A2:C11** for the data range, professional users often employ **A2:C**, which tells Google Sheets to include all rows from A2 downwards in Columns A, B, and C. Similarly, the key column reference **A2:A11** should be converted to **A2:A**. The criteria list range **E2:E5** should also ideally be defined as **E2:E** to accommodate future additions to the filtering criteria list. This ensures that the filter formula remains accurate regardless of how much data is added to the sheet.

The revised, dynamic formula would look like this:

=FILTER(A2:C,COUNTIF(E2:E,A2:A))

Implementing this dynamic approach vastly improves the long-term maintainability and reliability of the data analysis workflow in Google Sheets.

Summary of Advantages

Utilizing the combined power of FILTER and COUNTIF for list filtering offers substantial advantages over manual filtering or simpler conditional statements. Firstly, it provides high efficiency, generating the filtered result in a single array output without requiring multiple steps or helper columns. Secondly, it offers true dynamism, automatically updating the results whenever the source data or the criteria list is modified, ensuring that the displayed report is always current.

This technique is a cornerstone of sophisticated data management in the Google Sheets environment, empowering users to move beyond basic spreadsheet operations towards complex, array-based logic for rapid data extraction and reporting.