Learning Conditional Counting: Using COUNTIF with OR Logic in Google Sheets

Authored by

Mohammed looti

November 1, 2025

Introduction to Advanced Conditional Counting

In the realm of data analysis, particularly within a powerful tool like Google Sheets, a frequent requirement is determining the total number of cells within a specified range that satisfy one of many conditions. This process is formally known as conditional counting using OR logic. Unlike simple counting, this method demands that we aggregate results where a cell meets criterion A **or** criterion B **or** criterion C. Standard spreadsheet functions often struggle with this complexity.

The native COUNTIF function, while excellent for single conditions, is inherently limited; it is designed to evaluate only one criterion at a time against a given range. Therefore, achieving robust OR functionality requires a sophisticated combination of functions working in concert. This article will detail the definitive, scalable, and most efficient methodology for implementing multi-criteria counting in Google Sheets.

To successfully execute true OR conditional counting, we must employ a trio of essential functions: the powerful ArrayFormula, the aggregation tool SUM function, and the core counting mechanism, COUNTIF. This composite technique compels Google Sheets to process multiple criteria simultaneously, generate intermediate results, and ultimately sum them up to produce the accurate final count.

Mastering the Core Array Formula Structure

When we need to count cells based on multiple, non-mutually exclusive conditions (i.e., criteria linked by OR logic), the solution lies in utilizing an array constant directly within the COUNTIF function. This structure forces COUNTIF to perform its calculation repeatedly--once for each item in the array--generating an internal array of counts. This temporary array must then be processed further to deliver the final, aggregated total.

The following syntax represents the standard and most reliable foundation for counting cells that satisfy any of the specified criteria defined within the curly braces `{}`. Understanding and correctly applying this precise structure is paramount for advanced conditional reporting:

=ArrayFormula(SUM(COUNTIF(A:A,{"Value1", "Value2", "Value3"})))

In this powerful formula, the target range (specified here as A:A) is systematically evaluated against every criterion listed in the array: "Value1," "Value2," and "Value3." The aggregated output represents the total count of cells in column A that match any of these conditions. This sophisticated methodology successfully replicates the necessary OR functionality that the single COUNTIF function lacks.

Deep Dive into Array-Based OR Logic Execution

To fully grasp the mechanism behind this composite formula, it is crucial to understand the distinct role played by each functional wrapper. By default, the standard COUNTIF function is designed to operate in a scalar context, meaning it accepts one condition and returns one single numeric result. However, introducing an array of criteria, such as `{"Value1", "Value2"}`, dramatically changes its behavior.

The essential ingredient enabling this change is the ArrayFormula wrapper. This wrapper forces the inner COUNTIF operation to execute in an array processing mode. Instead of halting after the first calculation, the COUNTIF performs separate counts for every element within the criteria array. For instance, if "Value1" appears 4 times and "Value2" appears 7 times, the COUNTIF function, driven by ArrayFormula, returns the intermediate array `{4, 7}`.

The final step relies on the outer SUM function. It receives this resulting array--in our example, `{4, 7}`--and aggregates all the elements (4 + 7 = 11). This total represents the final, accurate, aggregated count for all conditions met across the range. This elegant, three-part combination is recognized as the definitive method for implementing complex OR counting logic in Google Sheets.

Practical Application 1: Counting Text Strings

Let us translate this powerful theoretical technique into a concrete, practical scenario. Suppose we are managing a dataset of regional sales and need to quickly ascertain the total number of transactions originating from either the "East" region **or** the "South" region. Our data is structured in a Google Sheet, with the region identifier located in Column A:

	A	B	C	D	E
1	**Division**	**Team**	**Points**		
2	West	Mavericks	99		
3	South	Hawks	95		
4	West	Suns	98		
5	West	Spurs	105		
6	West	Rockets	103		
7	West	Lakers	114		
8	East	Celtics	109		
9	East	Hornets	85		
10	West	Blazers	91		
11	North	Knicks	95		
12	North	Bulls	99		
13	South	Magic	104		
14	South	Heat	103		
15					
16					
17					
18					
19					

To calculate the total number of cells in column A that match either "East" or "South," we must construct our criteria array carefully. Since we are counting text strings, the values must be enclosed in double quotation marks within the curly braces `{}`:

=ArrayFormula(SUM(COUNTIF(A:A,{"East", "South"})))

Upon execution, the formula first generates a count for "East" (3 instances) and a count for "South" (2 instances), resulting in the intermediate array `{3, 2}`. The SUM function then totals this array, yielding 5. The resulting calculation is clearly demonstrated in the screenshot below, confirming the robust nature of array-based OR counting for textual data.

E2 =ArrayFormula(SUM(COUNTIF(A:A,{"East", "South"})))

	A	B	C	D	E
1	**Division**	**Team**	**Points**		
2	West	Mavericks	99		5
3	South	Hawks	95		
4	West	Suns	98		
5	West	Spurs	105		
6	West	Rockets	103		
7	West	Lakers	114		
8	East	Celtics	109		
9	East	Hornets	85		
10	West	Blazers	91		
11	North	Knicks	95		
12	North	Bulls	99		
13	South	Magic	104		
14	South	Heat	103		
15					
16					
17					
18					
19					

As confirmed by the output, a total of **5** cells in column A contain the value of "East" or "South." This example successfully illustrates the efficiency and accuracy of the array-based OR methodology for multiple text criteria.

Practical Application 2: Counting Numeric Values

The same fundamental array technique is equally applicable and effective when counting specific numeric values, though a small but crucial syntax difference exists in the criteria array definition. When specifying numerical conditions, we must omit the double quotation marks that are required for text strings to ensure Google Sheets treats them as numeric constants.

Consider a scenario where we are analyzing test scores and need to determine the frequency of three distinct, specific outcomes: scores of 95, 99, or 103. This is essential for identifying how often these precise numerical results occur within our dataset, located in Column C.

The specific formula designed to count these precise numeric criteria in Column C is constructed as follows:

=ArrayFormula(SUM(COUNTIF(C:C,{95, 99, 103})))

Crucially, observe that the numeric values `95, 99, 103` are placed directly inside the curly braces `{}` without any surrounding quotes. This definition ensures they are recognized as numerical constants by the array processor. The resulting calculation returns the total number of entries in Column C that match any of the three specified scores.

E2 =ArrayFormula(SUM(COUNTIF(C:C,{95, 99, 103})))

	A	B	C	D	E
1	**Division**	**Team**	**Points**		
2	West	Mavericks	99		6
3	South	Hawks	95		
4	West	Suns	98		
5	West	Spurs	105		
6	West	Rockets	103		
7	West	Lakers	114		
8	East	Celtics	109		
9	East	Hornets	85		
10	West	Blazers	91		
11	North	Knicks	95		
12	North	Bulls	99		
13	South	Magic	104		
14	South	Heat	103		
15					
16					
17					
18					
19					

Why the Array Formula Outperforms Nested COUNTIFs

A frequent error among intermediate spreadsheet users is attempting to implement OR logic by simply adding multiple standard COUNTIF function calls together. While mathematically sound--since addition effectively serves as OR logic when counting non-overlapping sets--this method quickly becomes impractical, inefficient, and difficult to manage as the number of criteria grows.

For instance, while the formula `=COUNTIF(A:A, "East") + COUNTIF(A:A, "South")` achieves the correct result for two criteria, scaling this to handle ten or twenty different criteria would require manually typing and maintaining a cumbersome formula involving numerous addition operators. This lack of scalability makes the nested approach non-viable for robust data analysis.

In contrast, the array-based solution utilizing ArrayFormula offers superior elegance and

dynamism. By centralizing all conditions within a single, concise array constant `{...}`, the entire set of criteria is managed in one location. This centralization dramatically simplifies auditing, maintenance, and future updates, solidifying the ArrayFormula approach as the professional standard for multi-criteria conditional counting.

Resources for Advanced Conditional Counting

Mastering the powerful combination of ArrayFormula, SUM, and COUNTIF is a foundational technique that unlocks advanced data manipulation capabilities within Google Sheets. This methodology allows analysts to construct complex reporting structures that would be impossible to achieve using basic, single-condition functions alone.

Furthermore, the array-based technique is exceptionally flexible. Instead of hardcoding values directly into the criteria array, you can use cell references (e.g., `{B1, B2, B3}`), which further increases the dynamic capabilities of the formula, allowing for the construction of highly interactive dashboards and reports where criteria can be changed instantly by updating a single cell.

For those looking to expand their knowledge beyond OR logic, the following resources cover other common conditional counting operations in Google Sheets, including those requiring AND logic or comparisons using numerical operators:

Tutorial: Using COUNTIF with Wildcards for Partial Matches

Tutorial: Implementing COUNTIF with AND Logic using COUNTIFS

Tutorial: Applying COUNTIF with Comparison Operators (>, <, <=)