

Filtering Data Across Multiple Columns: A Google Sheets Tutorial

Authored by
Mohammed loot

November 2, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Filtering Data Across Multiple Columns: A Google Sheets Tutorial*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=8121>

Achieving true mastery of [Google Sheets](#) necessitates developing sophisticated data manipulation skills. Among the most essential techniques is the ability to apply complex and multi-faceted filtering criteria to large datasets. While standard filtering options provide quick ways to isolate data based on values in a single column, real-world data analysis often demands the simultaneous application of conditions across multiple columns. Navigating complex business requirements or large-scale research projects requires a method that transcends simple dropdown selections.

When analyzing a comprehensive [dataset](#), you frequently encounter scenarios where rows must be included or excluded based on interdependent conditions--for example, filtering transactions that occurred in a specific region OR involved a certain product type. This advanced capability, known as multi-column [filtering](#), is critical for precise reporting and targeted analysis.

This guide provides a detailed, step-by-step tutorial demonstrating exactly how to unlock the advanced potential of the "Custom formula is" feature in Google Sheets. By leveraging fundamental principles of [Boolean logic](#), specifically the powerful **OR** and **AND** functions, users can implement highly specialized filters that define criteria across any number of columns simultaneously. Understanding how to structure these custom formulas will dramatically enhance your efficiency and accuracy in data processing.

Preparing Your Dataset for Effective Multi-Column Filtering

Before diving into the technical application of custom formulas, the foundational step for any successful data analysis endeavor is ensuring that your source data is impeccably structured. The effectiveness of any filter, especially those based on advanced logical formulas, relies entirely on the consistency and organization of the underlying data. For filtering to operate flawlessly, your dataset must be arranged in clear, consistent columns, with a dedicated header row that unambiguously defines the categories within the data range.

Consistency in data entry is paramount; formulas rely on exact matches. For instance, if filtering by region, ensure that "East," "east," and "Eastern" are not used interchangeably, as this will lead to inaccurate filtered results. For the purposes of this tutorial, we will utilize a typical sales ledger dataset. This sample data includes columns defining core business metrics such as **Region** (geographical location), **Product** (item sold), and **Sales** (transaction amount).

This specific structure enables us to define specific criteria based on either the geographic location or the type of product sold, allowing us to demonstrate the power of combining conditions across separate columns. The initial dataset, which occupies columns A, B, and C, is displayed below. We will use this structure as the basis for all subsequent filtering examples, ensuring all data points are ready for logical evaluation.

	A	B	C	D	E
1	Region	Product	Revenue		
2	East	A	10		
3	East	A	6		
4	East	B	8		
5	East	C	14		
6	West	A	10		
7	West	B	19		
8	West	B	22		
9	West	C	14		
10	North	A	18		
11	North	B	8		
12	North	C	4		
13	North	C	7		
14	South	A	7		
15	South	B	11		
16	South	B	13		
17	South	C	8		
18					
19					
20					
21					

Activating the Basic Filter View

The mechanism for applying any custom filter logic in Google Sheets is inherently tied to the active filter environment. Therefore, the essential first mechanical step is to activate the standard filtering feature across your working data range. This action performs two critical functions: it transforms the header row into interactive controls, and it establishes the scope--the range of rows--over which the subsequent custom formula will be evaluated.

To initiate the filter, click on any cell that resides within your structured data range. A best practice is to select the top-left cell of the dataset, typically cell **A1**, as Google Sheets is usually adept at automatically detecting and encompassing the entire adjacent dataset. Once selected, navigate to the **Data** tab located in the main menu bar at the top of the screen. From the dropdown menu, select the option labeled **Create a filter**.

Upon successful activation, you will immediately observe small, inverted funnel icons appearing next to each column header (A1, B1, C1, etc.). The presence of these icons confirms that the filter view is active across the selected data range, establishing the necessary environment for the

"Custom formula is" condition. It is important to note that you only need to create the filter once for the entire range; subsequent formula application can be done through any of the filter icons.

The screenshot shows a Google Sheets interface with a data table. The table has columns for Region, Product, and Revenue. The 'Data' menu is open, showing options like 'Sort sheet by column A, A → Z', 'Sort sheet by column A, Z → A', 'Sort range', 'Create a filter', 'Filter views', 'Slicer', 'Data validation', 'Pivot table', 'Randomize range', 'Named ranges', 'Protected sheets and ranges', 'Cleanup suggestions', and 'Column stats'. The 'Create a filter' option is highlighted.

	A	B	
1	Region	Product	Revenue
2	East	A	
3	East	A	
4	East	B	
5	East	C	
6	West	A	
7	West	B	
8	West	B	
9	West	C	
10	North	A	
11	North	B	
12	North	C	
13	North	C	
14	South	A	
15	South	B	
16	South	B	
17	South	C	
18			

Utilizing the OR Logic for Inclusive Criteria

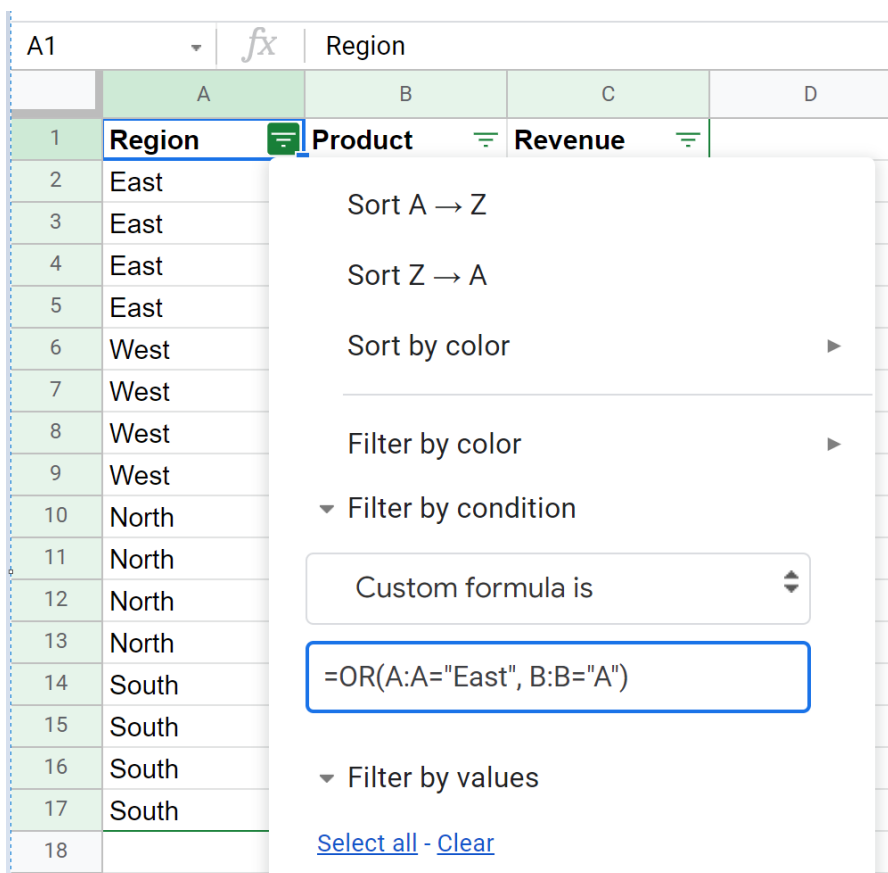
One of the most frequent requirements in business intelligence is to view data that satisfies one condition or another, providing a broad, inclusive perspective on transactions or records of interest. Consider a scenario where the marketing team needs to analyze all sales originating from the "East" geographical region *OR* any sales transaction involving "Product A," regardless of where it was sold. This kind of "either/or" scenario explicitly demands the use of the [OR function](#) within a custom filter formula.

To implement this flexible logic, begin by clicking the Filter icon associated with any column within the filtered range. While the formula logic applies across all columns, initiating the custom formula setup through the first relevant column (in this case, **Region** in column A) often provides the clearest context. In the filtering menu that appears, scroll down past the standard filtering options and click on **Filter by condition**.

From the condition dropdown menu, select the option labeled **Custom formula is**. This crucial selection bypasses the limitations of predefined conditions and opens an input box ready to accept the advanced logical expression. Enter the following formula into the designated box:

=OR(A:A="East", B:B="A")

This formula is a concise instruction to the filter mechanism: retain any row where the value in the entire range of column A (Region) is an exact match for the text string "East" OR where the value in the entire range of column B (Product) is an exact match for the text string "A." It is vital to use full column references (e.g., A:A) as the filter evaluates the condition row-by-row automatically.



	A	B	C	D
1	Region	Product	Revenue	
2	East			
3	East			
4	East			
5	East			
6	West			
7	West			
8	West			
9	West			
10	North			
11	North			
12	North			
13	North			
14	South			
15	South			
16	South			
17	South			
18				

Once you click **OK**, the dataset is dynamically filtered, providing the comprehensive view required. Notice that the resulting view is inclusive: it contains rows that satisfy only the Region condition (e.g., East/Product B), rows that satisfy only the Product condition (e.g., West/Product A), and rows that satisfy both (East/Product A). The **OR** logic is designed to be inclusive, delivering a broad sweep of all data relevant to either specified criterion.

	A	B	C	D	
1	Region ▼	Product ▾	Revenue ▾		
2	East	A	10		
3	East	A	6		
4	East	B	8		
5	East	C	14		
6	West	A	10		
10	North	A	18		
14	South	A	7		
18					
19					
20					
21					
22					
23					

Implementing the AND Logic for Strict Matching and Specificity

While the **OR** function is excellent for broad, inclusive analysis, many data requirements demand a highly strict and targeted result set. This is where the **AND** logic becomes indispensable. Unlike the flexible **OR**, a filter employing the **AND** function will only display rows where *every single specified condition* is simultaneously met. For example, if the financial controller requires a report showing only sales transactions that occurred in the "**East**" region **AND** specifically involved "**Product A**," the strict nature of the **AND** function is necessary.

To apply this stringent criterion, you must utilize the [AND function](#). Return to the filter condition box by clicking the filter icon on any column header. Select **Filter by condition**, and then **Custom formula is**. Replace the previous formula with the following refined expression:

```
=AND(A:A="East", B:B="A")
```

The structural syntax remains consistent with the OR example, but the functional behavior is fundamentally different. The use of **AND** mandates that both component condition `A:A="East"` and condition `B:B="A"` must evaluate to **TRUE** for that specific row to be retained and displayed in the filtered view. If even one of the conditions is false, the entire logical statement fails, and the row is hidden.

	A	B	C	D
1	Region	Product	Revenue	
2	East			
3	East			
4	East			
5	East			
6	West			
7	West			
8	West			
9	West			
10	North			
11	North			
12	North			
13	North			
14	South			
15	South			
16	South			
17	South			
18				
19				
20				
21				
22				

Sort A → Z

Sort Z → A

Sort by color ▶

Filter by color ▶

▼ Filter by condition

Custom formula is

`=AND(A:A="East", B:B="A")`

▼ Filter by values

[Select all](#) - [Clear](#)

Search:

✓ East

Upon executing this formula by clicking **OK**, the resulting filtered view will be significantly narrower and much more highly targeted. The result set represents a precise intersection of the two criteria. Only rows that simultaneously meet the criteria (Region is East AND Product is A) will remain visible, providing a highly specific analysis slice of your data, perfect for focused performance reviews or compliance checking.

	A	B	C	D
1	Region	Product	Revenue	
2	East	A	10	
3	East	A	6	
18				
19				
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				

Understanding the Mechanics of Custom Filter Formulas

The "Custom formula is" option is arguably the most powerful tool within the Google Sheets filtering environment because it grants exceptional, granular control over which data is displayed. It moves far beyond simple value matching and enables complex decision-making based on calculations or function outcomes. To effectively leverage this feature, it is crucial to grasp the underlying mechanism and strict requirements placed on the formula structure and output.

The core principle is simple yet rigid: the custom formula must resolve to a definitive **Boolean output** for every row in the dataset. If the formula evaluates to **TRUE** for a given row, that row is kept and displayed by the filter. Conversely, if the formula evaluates to **FALSE**, the row is hidden. This true/false requirement means that the formula must be constructed to yield a logical result, often involving comparison operators (e.g., =, >, <) or dedicated logical functions (**AND**, **OR**, **NOT**).

Furthermore, you are not limited to using just two conditions; both **AND** and **OR** can accept numerous logical arguments. For instance, you could filter for rows where the region is "East" **OR** the product is "A" **OR** the sales value in column C is greater than \$500, all within a single **OR** function. This nesting and combining ability allows for criteria complexity that is virtually limitless, serving even the most convoluted analytical needs.

Key characteristics and technical rules to remember when crafting custom filter formulas:

Absolute Boolean Output: Every formula must resolve strictly to **TRUE** or **FALSE**. Formulas

returning numerical values (unless they inherently represent a logical comparison) or text strings will generally fail to filter correctly.

Range Notation: Always use full column references (e.g., `A:A`, `C:C`) rather than specific cell ranges (e.g., `A2:A100`). When the filter applies the formula, it automatically evaluates the condition contextually for the current row it is checking, allowing the same formula to work down the entire column range seamlessly.

Flexibility and Integration: You are not restricted to just `AND` and `OR`. You can nest these functions for highly complex criteria (e.g., `=OR(AND(A:A="East", C:C>1000), B:B="B")`) or integrate other powerful functions like `NOT`, `ISBLANK`, `REGEXMATCH`, or numerical comparison operators (`>`, `<`, `>=`) for dynamic filtering based on calculations, date ranges, or text patterns across multiple columns.

Advanced Techniques and Alternative Data Management Solutions

Mastering the application of custom formulas for advanced filtering significantly elevates your capacity for sophisticated data analysis and management within [Google Sheets](#). This technique is not an isolated skill; it is often used in powerful combination with other data visualization, reporting, and cleaning tools to generate deep insights.

While the custom filter view is invaluable for dynamic, on-the-fly analysis (temporarily hiding rows you don't need), you may occasionally find yourself needing to permanently extract the filtered data into a new, separate location for archival or further processing. For this specific requirement, relying solely on the temporary filter view is inefficient. Instead, you should consider utilizing the dedicated `FILTER` function.

The standalone `FILTER` function, which is entered into a separate cell (e.g., `E1`), uses nearly identical Boolean logic syntax to the custom filter formula. However, instead of hiding rows, it returns a static array of the filtered results in a new range. For example, `=FILTER(A:C, OR(A:A="East", B:B="A"))` achieves the exact same logical result as the custom filter but outputs the resultant data block elsewhere, preserving the original dataset structure. This is essential for creating derived reports or clean subsets of data.

To continue building upon these advanced data manipulation skills, the following tutorials explain how to perform other common and complex operations in Google Sheets, which often complement the use of multi-column filtering:

Detailed methods on how to use array formulas for efficient, non-manual data processing and spillover results.

Practical techniques for setting up conditional formatting based on filter results or complex logical tests to visually highlight data patterns.

An in-depth guide to nesting logical functions (e.g., combining `AND`, `OR`, and `NOT`) for creating ultra-complex, multi-layered criteria for both filtering and calculation.