

Learn How to Use “Does Not Contain” with Excel Advanced Filter

Authored by
Mohammed loot

October 31, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Learn How to Use “Does Not Contain” with Excel Advanced Filter*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=7032>

The [Excel Advanced Filter](#) stands as an essential utility for power users, offering sophisticated data manipulation capabilities far exceeding the basic functionality of standard autofilters. This feature allows users to apply highly complex, multi-layered criteria to large [datasets](#), streamlining analysis and reporting tasks. Among its most powerful, yet frequently overlooked, functionalities is the ability to specifically exclude rows that contain a particular text string--the "does not contain" operation. Mastering this technique is crucial for precise data cleansing, ensuring that irrelevant or noisy entries are systematically removed, allowing analysts to focus strictly on the necessary information.

Implementing a "does not contain" filter requires adherence to a specific and precise syntax within the [Advanced Filter](#) mechanism. Unlike simple comparisons, exclusion filtering relies on a combination of the standard "not equal to" comparison operator and strategic use of [wildcard characters](#). This combination creates a pattern definition that clearly identifies and rejects any cell content matching the specified exclusion criteria. Properly constructing the [criteria range](#) is the foundational step to unlocking this powerful filtering capability, ensuring accurate results even when dealing with variable text formats.

The fundamental syntax required by the [Advanced Filter](#) to exclude cells that include a specific text fragment is always structured as a formula entry within the [criteria range](#), ensuring Excel interprets the command correctly.

`<>*sometext*`

Within this crucial formula, the `<>` operator performs the function of "not equal to." This instructs Excel to look for anything that is not an exact match to the subsequent pattern. The surrounding `*`` [wildcard characters](#) are placeholders, representing zero or any number of characters. By enveloping the target string, `sometext``, with asterisks, we ensure that the filter will exclude any cell where `sometext`` appears anywhere--at the beginning, end, or middle--within the cell's total content. This precise wildcard deployment is what differentiates a simple "not equal to" filter from a comprehensive "does not contain" filter.

Understanding the "Does Not Contain" Logic

The underlying logical framework of the "does not contain" function within [Excel's Advanced Filter](#) hinges entirely on the synergistic combination of the inequality operator (`<>`) and the ubiquitous asterisk [wildcard](#). The inequality operator establishes the rejection criterion, stating that the resulting data must be different from the specified pattern. When this operator is coupled with wildcards, the filtering power shifts from seeking an exact match to evaluating content inclusion.

The asterisk (`*`) is defined as representing an arbitrary sequence of characters of any length,

including an empty string. Consequently, the pattern `*sometext*` is interpreted by Excel as "any text string that includes 'sometext' somewhere within its boundaries." For illustrative purposes, if the criteria were `*North*`, it would successfully match "NorthAmerica," "Northern Region," "The North," or simply "North." The strategic move is to precede this inclusive pattern with the inequality operator, resulting in `<>*sometext*`. This instruction compels Excel to select and display all data entries that do **not** conform to the specified pattern. In essence, the command translates precisely to: "Retrieve all records where the specified field does not hold the text 'sometext' in any position."

This advanced filtering methodology proves exceptionally valuable when confronting real-world data issues, such as cleaning up inconsistent data entries, or when an analyst needs to systematically exclude specific categories or descriptive tags that may be embedded within larger text fields. The subsequent sections will provide detailed, practical demonstrations, walking through the application of this powerful exclusion method in scenarios involving both single and multiple exclusion criteria, particularly demonstrating how to enforce strict exclusion through [AND logic](#).

Example 1: Filtering for Rows that Do Not Contain One Specific Text

We begin with a fundamental and common data analysis requirement: filtering a comprehensive [dataset](#) to ensure that rows containing a specific text string in a target column are entirely excluded from the view. Consider a scenario where an analyst is reviewing a large sales ledger and needs to isolate the sales performance results for all regions, explicitly excluding all data associated with any region that includes the word "East."

The following sample [dataset](#) serves as the basis for this example, detailing product sales totals across various operational regions for a hypothetical company. Note the varying text entries in the "Region" column:

	A	B	C	D	E	F
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						

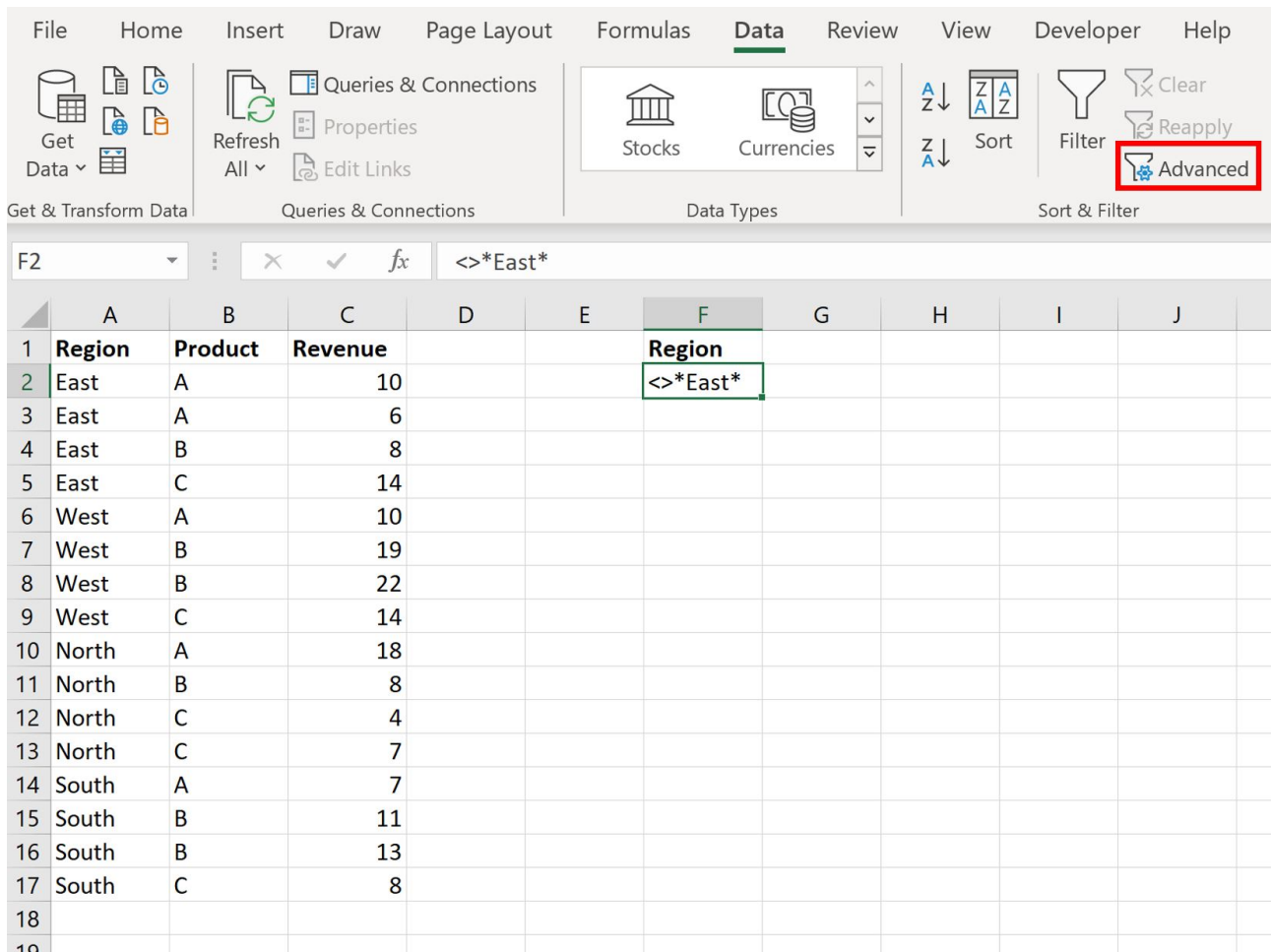
Our immediate objective is to refine this [dataset](#), ensuring that the resulting filtered view only displays rows where the "Region" column emphatically does **not** contain the target word "East." This rigorous exclusion means that any entry such as "East," "Eastern," "NorthEast," or "Southeast" must be hidden from the final analysis, isolating the remaining regional data points.

Step-by-Step Guide for Example 1

Define the Criteria Range: The foundational step for utilizing the [Advanced Filter](#) involves meticulously setting up the [criteria range](#). This range must consist of a minimum of two rows: the first row must contain the exact column header from your primary [dataset](#) (here, "Region"), and the second row must house the exclusion criterion itself. For our goal, in the cell immediately below the "Region" header in your criteria area, enter the formula `=<>*East*`. It is paramount that this criteria range is positioned in a vacant area of your worksheet, distinctly separate from your main list range, to avoid confusion or data corruption.

	A	B	C	D	E	F	G
1	Region	Product	Revenue			Region	
2	East	A	10			<>*East*	
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							

Access the Advanced Filter Utility: Once the [criteria range](#) is established, navigate to the **Data** tab located on the [Excel](#) ribbon. Within the dedicated "Sort & Filter" command group, locate and click the **Advanced** button. This action will initiate the launch of the [Advanced Filter](#) dialog box, which manages the application process.



The screenshot shows the Microsoft Excel interface with the **Data** ribbon selected. The **Advanced** filter option is highlighted with a red box. Below the ribbon, the spreadsheet is visible with the following data:

	A	B	C	D	E	F	G	H	I	J
1	Region	Product	Revenue			Region				
2	East	A	10			<*>East*				
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										

Configure the Filter Settings: Inside the [Advanced Filter](#) dialog box, select the desired action. Typically, "Filter the list, in-place" is chosen to modify the current view of your data. Next, meticulously define the ranges: for the **List range**, select the entire dataset, including all headers (e.g., `A1:C17`). For the **Criteria range**, select the two cells defining your criterion (e.g., `F1:F2`), ensuring both the header and the formula are included in the selection.

	A	B	C	D	E	F	G	H
1	Region	Product	Revenue			Region		
2	East	A	10			<>*East*		
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								
22								
23								

Advanced Filter ? X

Action

Filter the list, in-place
 Copy to another location

List range: \$A\$1:\$C\$17 ↑

Criteria range: \$F\$1:\$F\$2 ↑

Copy to: ↑

Unique records only

OK Cancel

Apply the Filter and Review Results: After confirming that all range selections are accurate, click **OK**. [Excel](#) will immediately execute the Advanced Filter command. Your dataset will now be dynamically filtered to display only those rows where the "Region" column does **not** contain the text "**East**," effectively concealing all data points associated with East, Eastern, or NorthEast regions.

	A	B	C	D	E	F	G
1	Region	Product	Revenue			Region	
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							
22							
23							
24							

Example 2: Filtering for Rows that Do Not Contain One of Multiple Text Strings (AND Logic)

Expanding upon the capabilities demonstrated in the initial example, advanced data analysis often necessitates the exclusion of rows based on the presence of any one of several predefined text strings. Imagine a scenario where the sales analysis must specifically exclude regions containing "East" as well as regions containing "West." This requirement demands the simultaneous application of multiple "does not contain" conditions, which, within the structure of the [Advanced Filter](#), is achieved using a horizontal [AND logic](#) setup.

We will continue to use the same sales [dataset](#), which provides a clean and consistent basis for applying complex criteria:

	A	B	C	D	E	F
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						

The refined objective is to filter for rows where the "Region" column satisfies two simultaneous conditions: it must not contain "East" **AND** it must not contain "West." This robust exclusion ensures that all rows, whether labeled "East," "Eastern," "West," or "Western," are completely removed from the resulting view, leaving only truly neutral regions.

Step-by-Step Guide for Example 2

Define the Criteria Range for Multiple Exclusions: To implement [AND logic](#) for multiple conditions applied to the same data field, each condition must be placed in a separate column within the same row of your [criteria range](#). Start by ensuring that both columns have the exact matching header name ("Region"). For instance, place "Region" in cell F1 and "Region" again in cell G1. In the row directly beneath the headers, enter the first exclusion formula `="<>*East*"` (in F2), and the second exclusion formula `="<>*West*"` (in G2). This horizontal arrangement within the [criteria range](#) mandates that a row must satisfy both conditions to be included in the final filtered dataset.

	A	B	C	D	E	F	G
1	Region	Product	Revenue			Region	Region
2	East	A	10			<>*East*	<>*West*
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							

Access the Advanced Filter Utility: As demonstrated previously, access the **Data** tab on the ribbon and click the **Advanced** button. This action opens the [Advanced Filter](#) dialog box, preparing for the range input.

Configure the Filter Settings: Within the dialog box, maintain the "Filter the list, in-place" option. Define the **List range** as your full dataset (e.g., `A1:C17`). Crucially, for the **Criteria range**, select the expanded two-column range that encompasses both sets of exclusion criteria (e.g., `F1:G2`). This selection ensures that both the "not East" and "not West" conditions are utilized simultaneously.

	A	B	C	D	E	F	G	H
1	Region	Product	Revenue			Region	Region	
2	East	A	10			<>*East*	<>*West*	
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								

Advanced Filter ? X

Action

Filter the list, in-place

Copy to another location

List range: ↑

Criteria range: ↑

Copy to: ↑

Unique records only

OK Cancel

Apply the Filter and Review Results: Click **OK** to execute the filter. Your dataset will now be precisely filtered, displaying only the rows where the "Region" column meets the compound condition: it does **not** contain "**East**" **AND** it does **not** contain "**West**." This results in a clean isolation of regions such as "North" and "South," providing the exact data subset required for the focused analysis.

	A	B	C	D	E	F	G
1	Region	Product	Revenue			Region	Region
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							
22							
23							
24							

Best Practices for Advanced Filtering

To guarantee the accuracy, reliability, and long-term efficiency of your data processing when utilizing [Excel's Advanced Filter](#), especially when deploying complex "does not contain" logic, adhering to established best practices is vital. These guidelines prevent common errors and ensure seamless operation:

Exact Header Matching is Mandatory: A fundamental requirement for the [criteria range](#) is that the column headers used must precisely duplicate the headers found in the list range. Even minor discrepancies--such as an unintended space, a punctuation difference, or incorrect capitalization--will cause the filter mechanism to fail, returning unexpected or empty results.

Strategic Criteria Range Placement: Always designate an area on your worksheet for the [criteria range](#) that is completely separate and sufficiently distant from your main list range. Placing criteria adjacent to or within the dataset risks accidental data overwriting or misinterpreting the boundaries of the list range during selection.

Understanding Case Sensitivity Defaults: By design, the [Advanced Filter](#) treats text criteria as case-insensitive. Consequently, the criterion `*east*` will successfully match "East," "east," or "EAST." If the filtering requirement explicitly demands case-sensitive exclusion, you must transition to formula-based criteria, incorporating case-sensitive functions such as `FIND` or `EXACT`.

Filtering In-Place vs. Copying to Another Location: The [Advanced Filter](#) dialog offers the choice between "Filter the list, in-place" (modifying the current view) and "Copy to another location." For robust, non-destructive analysis, or when the goal is to retain the original dataset untouched, it is highly recommended to choose the "Copy to another location" option and specify a new, empty destination range for the filtered results.

Clearing Filters Effectively: After concluding your analysis using the [Advanced Filter](#), it is necessary to revert the display. This is achieved by navigating back to the **Data** tab and clicking the **Clear** button located within the "Sort & Filter" group, which restores the dataset to its initial, unfiltered state.

Conclusion

Achieving mastery over the "does not contain" feature within [Excel's Advanced Filter](#) represents a significant enhancement to any analyst's data manipulation toolkit. Whether the requirement is to exclude a lone keyword or to enforce complex, multi-string exclusions using rigorous [AND logic](#), the systematic application of the `<>*sometext*` syntax delivers exceptional control over data visibility. By diligently following the step-by-step procedures detailed in this guide and consistently applying the recommended best practices, you can efficiently refine your [datasets](#), leading directly to more targeted analysis, clearer insights, and ultimately, more informed decision-making processes.

Additional Resources

To further expand your proficiency in advanced filtering techniques and explore other complex data operations in [Excel](#), the following authoritative resources are recommended:

[Microsoft Support: Filter data by using an Advanced Filter](#)

[Microsoft Support: Criteria range rules](#)

[Wikipedia: Wildcard character](#)