

# Excel Formula: COUNTIF Does Not Contain

Authored by  
**stats writer**

November 17, 2025

## RECOMMENDED CITATION

stats writer (2025). *Excel Formula: COUNTIF Does Not Contain*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=92672>

## Introduction to Conditional Counting in Microsoft Excel

Conditional counting is a fundamental task for data analysis within Microsoft Excel. While the primary use of functions like COUNTIF function is to tally cells that meet specific criteria (e.g., equals "X" or is greater than 5), data analysts often encounter the inverse requirement: counting cells that expressly **do not contain** certain text strings. Mastering this negative conditional counting is essential for cleaning datasets, identifying exceptions, and performing complex filtering operations efficiently across large volumes of data.

The core challenge in formulating a "does not contain" rule lies in correctly employing wildcard characters in conjunction with the standard inequality **logical operators**. Wildcards allow Excel to search for patterns rather than exact matches, and the logical operator is necessary to specify that any cell matching that pattern must be excluded from the final count. This article will guide you through two primary methods: utilizing the standard COUNTIF function for single exclusion conditions and employing the powerful COUNTIFS function for scenarios requiring multiple, simultaneous exclusions.

## Understanding the COUNTIF Function and Criteria

The COUNTIF function is designed to count cells within a specified range that meet a single, predefined criterion. Its basic syntax is simple: **COUNTIF(range, criteria)**. When dealing with complex criteria, especially those involving partial text matches, the criteria argument must be carefully constructed to ensure accurate results. If we were looking for an exact match, the criterion would be straightforward, but when we search for text that might be embedded within a longer string, standard text matching is insufficient.

To instruct Excel to look for a text string that is only a portion of the cell's content, we must employ the asterisk (\*) wildcard character. The asterisk acts as a placeholder for any number of characters, including zero characters, both before and after the text string in question. For example, if we want to check whether the letter "A" exists anywhere in a cell, the positive criterion is **"\*A"**. This tells Excel: find "A" preceded by anything and followed by anything. To achieve the necessary negation--counting what **is not present**--we must use the "not equal to" operator, which is universally represented in Excel criteria as **"<>"**.

## Formulating the "Does Not Contain" Criterion (Single Condition)

The core of the single-condition "does not contain" formula involves merging the negation operator with the wildcard pattern. If our objective is to count every cell that does not contain the specified text, say the letter "A," the complete criterion string must be constructed within quotation marks as **"<>\*A"**. This structure forms a single logical instruction for the COUNTIF function: "Only count the

cell if its content is not equivalent to a string that contains 'A' anywhere within it."

This method is exceptionally useful for primary filtering tasks, such as excluding all records that are flagged with a specific identifier or keyword. It provides a quick and robust way to segregate data based on the absence of a characteristic. It is vital to remember that the COUNTIF function is strictly limited to evaluating only one criterion at a time. If your analytical requirements mandate the exclusion of two or more distinct text strings simultaneously, you must utilize the expanded capabilities of the COUNTIFS function, which allows for compounded negative logical operators.

The structure for executing a count based on a single exclusion criteria is demonstrated below, using the common text "A" as the target for exclusion:

### Formula 1: Count Cells that Do Not Contain One Specific Text

**=COUNTIF(A2:A11,"<>\*A\*")**

This implementation counts the number of cells within the range **A2:A11** that are completely devoid of the specified text "A." The placement of the wildcards ensures that the search includes partial matches for exclusion, meaning any cell with "A" embedded within it will be ignored by the count.

### Practical Implementation: Counting Cells Excluding One Term (Example 1)

To demonstrate the effectiveness of this negated wildcard search, let us apply it to a practical scenario involving a list of Product Identification (Product ID) codes. Imagine you are managing inventory and need to identify all products that are not associated with a specific legacy manufacturing process, which is flagged by the letter "A" appearing anywhere in its code. Our initial data array, containing ten Product IDs, spans the range **A2:A11**.

First, we visualize the dataset we are working with:

	A	B	C	D	E
1	<b>Product ID</b>				
2	AA049B				
3	A500C9				
4	AA3440				
5	BD3450				
6	CC5600				
7	CR9320				
8	CE3490				
9	AC7834				
10	AA78R4				
11	AG0978				
12					
13					
14					

Our goal is to execute a count of the cells that **do not contain** the letter "A" anywhere within the Product ID string. To achieve this, we will input the appropriate COUNTIF formula into an adjacent, empty cell, such as **C2**, where the result will be displayed.

The formula utilized is the single-criterion COUNTIF approach:

**=COUNTIF(A2:A11,"<>\*A\*")**

The successful application of this formula within the worksheet is visible in the subsequent screenshot, showing the calculation in practice:

	A	B	C	D
1	Product ID		Count of Cells that Do Not Contain "A"	
2	AA049B		4	
3	A500C9			
4	AA3440			
5	BD3450			
6	CC5600			
7	CR9320			
8	CE3490			
9	AC7834			
10	AA78R4			
11	AG0978			
12				
13				
14				

The output reveals the result of **4**. This confirms that there are four distinct Product IDs within the defined range that satisfy the complex condition of not containing the character "A." This method ensures efficient isolation of data points that fail to match a specific negative criterion.

## Leveraging COUNTIFS for Multiple Exclusion Criteria

For analytical requirements that necessitate the exclusion of cells based on **two or more** independent criteria simultaneously, the standard COUNTIF function is insufficient. Instead, the analyst must transition to the more advanced COUNTIFS function. COUNTIFS is designed to evaluate multiple range and criteria pairs, meaning it can check a single cell against numerous conditions. Critically, for a cell to be counted, it must satisfy **all** criteria provided to the function. This is based on the implicit AND logical operator that governs the function's logic.

This implicit AND logic is precisely what makes COUNTIFS suitable for multiple exclusions. If the objective is to count cells that do **not contain "A" AND do not contain "C"**, we define two separate exclusion criteria within the same COUNTIFS structure. Each criterion must maintain the negated wildcard pattern (e.g., "<>\*A\*" and "<>\*C\*"). Since COUNTIFS requires all conditions to be true, listing both exclusions ensures that only cells free of both "A" and "C" are included in the final tally.

The structure for counting cells that do not contain multiple specific text strings requires repeating the evaluation range for every criterion. If we aim to exclude both "A" and "C," the formula must reference the range **A2:A11** twice, pairing it first with the exclusion criteria for "A" and then with the

exclusion criteria for "C." This repeated pairing is non-negotiable for multi-criteria evaluation in this function.

### Formula 2: Count Cells that Do Not Contain Multiple Specific Text

**=COUNTIFS(A2:A11,"<>\*A\*", A2:A11, "<>\*C\*")**

This compound formula systematically checks every cell in the defined range. It counts a cell only if it meets the first negative condition (does not contain "A") **and** the second negative condition (does not contain "C"). If a cell contains "A" but not "C," or vice versa, it is excluded from the final count because it fails one of the required conditions.

### Handling Multiple Exclusion Criteria (Example 2)

Building upon our previous Product ID dataset, let us introduce a more selective requirement: we need to isolate products that are associated with neither the legacy system "A" nor the specialized category "C." This task mandates the use of the COUNTIFS function to impose two simultaneous negative constraints on the data.

We will utilize the identical dataset spanning **A2:A11** and input the dual-exclusion COUNTIFS formula into cell **C2**, overwriting the previous calculation:

**=COUNTIFS(A2:A11,"<>\*A\*", A2:A11, "<>\*C\*")**

The outcome of this operation, which filters the list to exclude any product ID containing either "A" or "C," is demonstrated in the screenshot below:

	A	B	C	D
1	Product ID		Count of Cells that Do Not Contain "A" or "C"	
2	AA049B		1	
3	A500C9			
4	AA3440			
5	BD3450			
6	CC5600			
7	CR9320			
8	CE3490			
9	AC7834			
10	AA78R4			
11	AG0978			
12				
13				

By imposing the constraint that the Product ID must not contain "A" **AND** must not contain "C," the result set is significantly reduced. The output clearly shows that there is only **1** cell that successfully meets both stringent, negative requirements. This single result represents the unique product ID that is completely free from both the "A" and "C" classifications, confirming the precise nature of the COUNTIFS function when handling compounded exclusions.

## Key Considerations and Limitations of COUNTIF/COUNTIFS

While the use of negated wildcards in conditional counting is highly effective, users must be aware of inherent function behaviors, particularly regarding case sensitivity and correct syntax implementation. By default, both the COUNTIFS function and the COUNTIF function in Microsoft Excel operate in a **case-insensitive** manner when processing text criteria. This means that a criterion designed to exclude "\*a\*" will successfully exclude cells containing "apple," "Apple," or "APPLE" equally. If the requirement demands case-sensitive exclusion, these native functions are inadequate, and the solution typically requires shifting to more advanced array formulas involving functions such as **SUMPRODUCT** in combination with string manipulation functions like **FIND** or **SEARCH**, which drastically increases complexity.

Furthermore, proper syntax is absolutely critical for successful execution. It must be reinforced that all criteria, including the "not equal to" operator (" $\neq$ ") and the wildcard characters (\*), must be completely enclosed within double quotation marks (e.g., " $\neq$ \*text\*"). If the quotation marks are omitted or misplaced, Excel will incorrectly interpret the inequality symbols as calculation operators, inevitably leading to a **#VALUE!** error or a complete failure of the logical test. Attention

to these minute syntactical details is paramount for reliable data analysis.

**Note:** For users who require comprehensive detail on the syntax, structure, and constraints related to multi-conditional counting, you can find the complete official documentation for the [COUNTIF function](#) and related formulas in official Microsoft Excel online resources. Mastering these nuanced functions is vital for advanced data manipulation tasks and ensures that your conditional counts are accurate and reliable, regardless of whether you are including or excluding specific text strings.

ARABPSYCHOLOGY.COM