

# How to Count Cells with Criteria Using COUNTA IF in Google Sheets

Authored by  
**stats writer**

January 31, 2026

## RECOMMENDED CITATION

stats writer (2026). *How to Count Cells with Criteria Using COUNTA IF in Google Sheets*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=128799>

The concept of a combined **COUNTA IF** function frequently arises when users seek to count the number of non-empty cells in a range, subject to conditions defined in a corresponding column. In advanced

Google Sheets, this specialized counting ability allows users to perform sophisticated conditional calculations. By implementing this methodology, one can efficiently filter data and calculate the total number of entries that meet specific, dual criteria. This capability transforms raw data into actionable insights, proving to be an invaluable tool for complex data analysis and rigorous data management.

While the syntax "COUNTA IF" might suggest a straightforward combination of two distinct operations--checking for non-empty content (COUNTA) and applying a condition (IF)--the functionality is actually handled by a more robust, single function designed specifically for multiple criteria. Understanding this distinction is vital for anyone aiming to master advanced counting techniques within the Google Sheets environment. This guide will clarify the appropriate function to use and provide a detailed, practical example demonstrating how to effectively count non-blank entries based on defined criteria.

## Using Conditional Non-Blank Counting in Google Sheets

### The Misconception of COUNTA IF

Many users migrating from other spreadsheet applications, or those familiar with basic functions like the

COUNTA function and the

IF function, instinctively search for a function named **COUNTA IF**. They correctly hypothesize the need: to count values in Column B, but only for rows where Column A meets a specified condition.

If we were to use **COUNTA** alone, it would simply tally all non-empty cells across an entire specified range, disregarding any external filters or conditional requirements defined in other cells.

The inherent limitation of structuring the logic as a direct combination of **COUNTA** and **IF** is that **IF** typically returns a value based on a single TRUE or FALSE outcome, whereas **COUNTA** operates across a range. While array formulas can sometimes link these functions, the resulting complexity and computational load often make them impractical for standard conditional counting tasks within a large

spreadsheet model. The goal is to apply a condition row-by-row and then aggregate the count of those rows that satisfy the criteria.

The underlying goal--counting based on multiple, simultaneous requirements, including the

requirement that a cell must contain data (i.e., not be empty)--is achievable, but requires a different approach. Recognizing that this need is common, developers created a dedicated function that natively handles these composite conditions, simplifying the syntax significantly and improving calculation efficiency. This function streamlines the process of data evaluation across complex relational datasets, providing a clean solution to conditional counting problems.

## Introducing the COUNTIFS Solution

Fortunately, the developers of [Google Sheets](#) anticipated the requirement for conditional counting across multiple criteria ranges. The functionality that users seek when they look for **COUNTA IF** is fully encompassed within the powerful

[COUNTIFS function](#). The **COUNTIFS** function is specifically designed to count the number of rows that satisfy all criteria provided across one or more ranges, making it the perfect tool for implementing complex filtering logic that requires simultaneous fulfillment of multiple conditions.

The key advantage of using **COUNTIFS** is its ability to handle multiple criteria seamlessly. Instead of nesting or combining functions, you simply list the criteria range and its corresponding condition sequentially within the function arguments. To simulate the "COUNTA" aspect (i.e., counting only non-empty cells), we utilize a specific logical operator within one of the criteria arguments of **COUNTIFS**. This method eliminates the need to manually build an array-based solution or rely on convoluted formula structures that can be difficult to audit and maintain.

By mastering the **COUNTIFS** syntax, users can easily determine metrics such as "How many sales transactions over \$100 were completed by Agent Smith this month?" or, in the context of this specific problem, "How many players categorized as 'Guard' actually have a recorded score?" This function is fundamental for accurate reporting and focused [data analysis](#) where precise filtering based on multiple data columns is mandatory for deriving meaningful conclusions.

## Setting Up the Practical Example

To demonstrate the functionality of using **COUNTIFS** to mimic **COUNTA IF** behavior, consider a practical dataset related to basketball statistics. This dataset includes player names, their assigned position, and the points they scored during a recent game. Our objective is to perform a count based on two conditions simultaneously: first, the player must hold a specific position (e.g., "Guard"); and second, the cell recording their points must not be blank, ensuring we only count players who actively participated and for whom data is recorded.

Suppose we have the following sample data structure that shows the points scored by various basketball players. Notice that some players might have a position assigned but their points

column is intentionally left blank, perhaps indicating they did not play or their statistics were not yet entered, representing true non-entries:

	A	B	C	D
1	<b>Position</b>	<b>Points</b>		
2	Guard	22		
3	Guard			
4	Forward	19		
5	Forward	30		
6	Guard			
7	Forward	29		
8	Forward	32		
9	Guard	30		
10	Guard	28		
11	Forward	24		
12	Forward			
13	Guard	19		
14	Forward	12		
15	Forward	15		
16	Guard	18		
17				
18				

Our specific task is to count the total number of players who are listed as "Guard" in the **Position** column (Column A) but only if the corresponding cell in the **Points** column (Column B) is not empty. This effectively counts Guards who have recorded points, distinguishing them from Guards whose score is unrecorded. This task is a classic application of conditional non-blank counting, demanding the precision that the **COUNTIFS** function provides.

## Applying the COUNTIFS Formula

To achieve this precise count, we must construct a formula using the **COUNTIFS** function that incorporates both the positional criteria and the non-blank criteria. The syntax requires pairing a range with its specific condition. In our example, we are using the range A2:A16 for the position check and the range B2:B16 for the points check, ensuring that the ranges are of the same length to maintain accurate row alignment.

We can type the following highly specific formula into an empty cell, such as **D2**, to calculate the

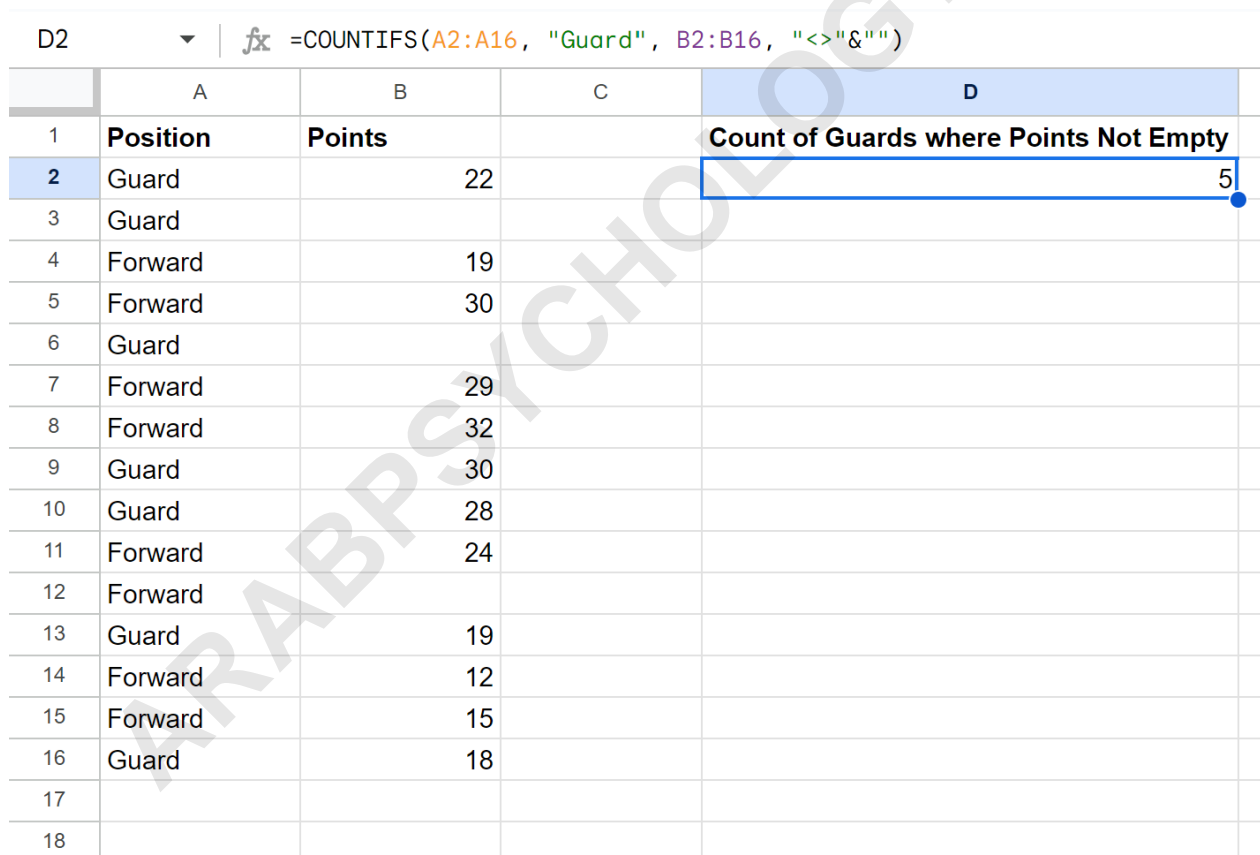
required number:

**=COUNTIFS(A2:A16, "Guard", B2:B16, "<>"&"")**

This formula instructs

Google Sheets to look through the range A2:A16 and ensure the value is exactly "Guard". Simultaneously, for the same row, it checks the range B2:B16 to ensure that the value meets the second criterion, which is that it must not be blank. The specific syntax for "not blank" or "not empty" is represented by the concatenation of the "not equal to" operator (<>) and an empty string ("").

The following screenshot illustrates the implementation of this formula within the live spreadsheet environment, showing exactly where the formula is placed and the resulting calculation:



D2    fx =COUNTIFS(A2:A16, "Guard", B2:B16, "<>"&"")

	A	B	C	D
1	<b>Position</b>	<b>Points</b>		<b>Count of Guards where Points Not Empty</b>
2	Guard	22		5
3	Guard			
4	Forward	19		
5	Forward	30		
6	Guard			
7	Forward	29		
8	Forward	32		
9	Guard	30		
10	Guard	28		
11	Forward	24		
12	Forward			
13	Guard	19		
14	Forward	12		
15	Forward	15		
16	Guard	18		
17				
18				

## Analyzing the Results and Verification

Upon entering the formula, the cell returns a value of **5**. This result signifies that there are exactly five rows in the dataset where both conditions are met: the player's position is explicitly designated as "Guard," AND the corresponding value in the **Points** column is recorded and is not empty. This

calculation provides instant quantification of a specific, filtered subset of the data, which is highly useful for performance metric tracking.

To ensure the formula is functioning as expected and to verify the accuracy of the **COUNTIFS** function, we can manually inspect the dataset. The manual verification process involves systematically going through each row and checking if it satisfies the dual conditions: Position equals "Guard" and Points is non-blank. This step is crucial in high-stakes **data analysis** to build confidence in the automated calculation, especially when designing new conditional metrics.

The following visual confirmation highlights the five specific rows that satisfy the criteria applied in our formula:

	A	B	C	D
1	<b>Position</b>	<b>Points</b>		<b>Count of Guards where Points Not Empty</b>
2	Guard	22		5
3	Guard			
4	Forward	19		
5	Forward	30		
6	Guard			
7	Forward	29		
8	Forward	32		
9	Guard	30		
10	Guard	28		
11	Forward	24		
12	Forward			
13	Guard	19		
14	Forward	12		
15	Forward	15		
16	Guard	18		
17				
18				

As shown by the highlighted rows, each player identified has "Guard" listed as their position, and critically, the cell in the **Points** column associated with them contains a numerical value, confirming that the formula successfully excluded any rows where the points cell was blank, even if the position was correctly listed as "Guard." This validates the effectiveness of using the specific non-blank criteria within **COUNTIFS**.

## Understanding Criteria Syntax for Non-Blank Cells

A key element of achieving the **COUNTA IF** functionality using **COUNTIFS** lies in correctly specifying the criteria for non-empty cells. In the formula provided, the crucial criterion is "`<>`"&" " applied to the points range (B2:B16). This syntax is standard across many spreadsheet platforms for defining a non-blank condition that functions reliably within criteria-based summary functions.

The `<>` operator is the logical symbol for "not equal to." When used in criteria arguments for functions like **COUNTIFS** or **SUMIFS**, it is enclosed in quotation marks. The "" represents an empty string--meaning a cell that contains absolutely no data. The ampersand symbol (&) is used here as a concatenation operator, combining the "not equal to" symbol with the empty string, effectively telling

Google Sheets to count cells that are "not equal to nothing." This construction is necessary because criteria arguments often require logical operators to be treated as text strings.

It is important to differentiate between cells that are genuinely empty and cells that contain a zero (0) or a space character. The formula "`<>`"&" " specifically targets true blank cells. If a cell contains a space, the

COUNTA function would count it, and this **COUNTIFS** criteria would also treat it as non-blank because a space character is not equivalent to an empty string. This precision in defining criteria ensures that the count reflects only those records where data truly exists in the target column.

## Broader Applications of Conditional Counting

While this example focused on basketball data, the technique of combining a category criterion with a non-blank criterion using the

COUNTIFS function is universally applicable across various sectors. Whether managing inventory (counting items sold only if their delivery status is confirmed), tracking sales leads (counting leads categorized as 'High Priority' only if a follow-up date is assigned), or managing financial records, conditional non-blank counting is essential for generating accurate reports based on completeness of data.

This approach ensures that your metrics are based on complete records, preventing skewed results that might arise from including rows where critical data points are missing. For instance, if you were relying on the **Points** column for an average calculation, counting only non-blank entries ensures that you are only considering players for whom the score is actually known, thus improving the integrity of subsequent statistical calculations and reporting.

Mastering the **COUNTIFS** function allows users to transcend simple counting operations and implement complex filtering logic that would otherwise require multiple steps or advanced scripting.

For those engaging in deep [data analysis](#), this powerful function remains a cornerstone of efficient and accurate [Google Sheets](#) usage, enabling sophisticated data modeling directly within the spreadsheet environment.

## Further Exploration in Google Sheets

Having successfully implemented conditional non-blank counting, users often look to solve related problems involving data aggregation and manipulation. The principles utilized in **COUNTIFS**--defining multiple criteria across different ranges--are applicable to several other critical spreadsheet functions, such as **SUMIFS** (for conditional summation) and **AVERAGEIFS** (for conditional averaging).

Understanding the core mechanics of conditional functions is the gateway to automating complex reporting tasks. Whether you need to conditionally count text, numbers, or dates, the approach remains largely the same: define the range, define the criterion, and repeat the pair for every condition that must be met simultaneously within the corresponding row.

The following tutorials explain how to perform other common and advanced tasks in [spreadsheets](#), helping you expand your proficiency beyond basic data entry and into sophisticated data modeling:

Learn how to use **SUMIFS** for conditional summing based on multiple criteria.

Explore techniques for using the [COUNTA function](#) in conjunction with array formulas for dynamic range counting.

Discover methods for counting unique values conditionally using a combination of **ARRAYFORMULA** and **QUERY**.

**Note:** You can find the complete documentation for the **COUNTIFS** function in [Google Sheets documentation](#).