

How to Count Unique Values in Google Sheets with Multiple Criteria

Authored by
stats writer

January 25, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Count Unique Values in Google Sheets with Multiple Criteria*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=127523>

Mastering Google Sheets involves moving beyond simple calculations to perform sophisticated data analysis. A common, yet critical, challenge is determining the count of unique values within a dataset that simultaneously satisfies several specified conditions. This process moves beyond the basic capabilities of functions like COUNTIFS, requiring a dedicated tool for uniqueness combined with criteria filtering.

The solution lies in the powerful **COUNTUNIQUEIFS** function. This function is specifically engineered to allow users to apply multiple filtering criteria to a range and then calculate the exact number of distinct entries remaining after the filtration process. Utilizing **COUNTUNIQUEIFS** ensures both high accuracy and efficiency when dealing with complex datasets where duplication is a factor, providing valuable insights into segmented data groups based on specific conditions, such as regions, dates, or performance metrics.

By meticulously following the prescribed syntax and defining precise ranges and criteria, analysts can streamline their workflow. This capability is indispensable in scenarios like inventory management, tracking sales performance across different markets, or analyzing survey responses, where it is paramount to identify the number of truly unique entities that conform to a specific set of rules. This article will provide a comprehensive guide to implementing this function effectively.

Google Sheets: Counting Unique Values Based on Multiple Criteria

Understanding the COUNTUNIQUEIFS Function Syntax

The core principle of counting unique values based on conditions relies on the COUNTUNIQUEIFS function. Unlike its predecessor, which only counts instances, this function ensures that duplicate entries within the counting range are only tabulated once, provided they satisfy all associated criteria. Understanding the proper structure, or syntax, is the first step toward successful implementation.

The generalized syntax of the function is defined as follows: `=COUNTUNIQUEIFS(count_range, criteria_range1, criterion1,)`. The initial parameter, **count_range**, specifies the column or range containing the values you wish to count uniquely. Subsequent pairs of arguments define the filters: **criteria_rangeN** specifies the range where the condition must be met, and **criterionN** specifies the actual condition itself (e.g., a specific text string, a number, or a comparison operator).

For instance, to count unique IDs in column A only if column B equals "West" and column C is greater than 20, the formula is structured logically, linking the counting range (A2:A11) to the two independent criteria ranges (B2:B11 and C2:C11) and their respective conditions. This structure is highly flexible, allowing for the application of many criteria pairs, significantly enhancing data segmentation capabilities within **Google Sheets**.

Basic Syntax Implementation

You can use the following basic syntax in [Google Sheets](#) to count unique values based on multiple conditions, focusing on clarity and range definition. Note that text criteria must be enclosed in quotation marks, as must any criteria involving comparison operators like greater than (>) or less than (<).

```
=COUNTUNIQUEIFS(A2:A11,B2:B11,"West",C2:C11,">20")
```

This specific formula is designed to calculate the number of unique entries found within the range **A2:A11**. The uniqueness count is conditional, requiring that two specific criteria are satisfied simultaneously across the corresponding rows of the dataset. This ensures that only data points that meet the required segmentation rules are included in the final tally.

The criteria applied are robust and distinct. The first criterion mandates that the value in the parallel range **B2:B11** must be the exact string "West". The second criterion imposes a numerical constraint, requiring that the value in the range **C2:C11** must be strictly greater than 20. Both conditions must evaluate to true for a row's value in A2:A11 to be considered for the unique count.

Why Standard COUNTIFS Falls Short for Unique Counting

It is essential to understand the functional distinction between the standard COUNTIFS function and **COUNTUNIQUEIFS**. While COUNTIFS is excellent for tallying the total number of cells that meet multiple conditions, it fails when the requirement is to count only the distinct occurrences. If an item appears three times in the counting range, and all three instances satisfy the criteria, COUNTIFS will return 3, whereas **COUNTUNIQUEIFS** will correctly return 1.

Imagine a scenario where a specific product ID (the unique value we are counting) appears multiple times in a sales log due to repeat purchases, but we only want to know how many **unique products** were sold in the West region during Q1. Using **COUNTIFS** would inflate the result by counting every transaction. **COUNTUNIQUEIFS** provides the necessary de-duplication step after the criteria filtering has occurred, yielding an accurate count of distinct entities.

Prior to the introduction of **COUNTUNIQUEIFS**, achieving this required complex array formulas involving ARRAYFORMULA, QUERY, and UNIQUE, often leading to slow performance and complicated debugging. The specialized function consolidates this complexity into a straightforward, single-function solution, significantly improving ease of use and computational efficiency for analysts dealing with large datasets in **Google Sheets**.

Detailed Example: Basketball Player Analysis

To illustrate the practical application of COUNTUNIQUEIFS, consider a typical sports data analysis scenario involving basketball players. Suppose we maintain a spreadsheet tracking various player attributes, including their name, conference, and points scored. We want to identify how many distinct athletes meet a high-performance threshold within a specific geographical division.

Below is a hypothetical dataset structured in Google Sheets, illustrating player performance across different conferences. Note that some players, such as Player E, appear multiple times, representing different game performances or tracking periods.

	A	B	C	D
1	Team	Conference	Points	
2	Mavs	West	22	
3	Spurs	West	18	
4	Rockets	West	13	
5	Celtics	East	40	
6	Heat	East	23	
7	Nets	East	30	
8	Lakers	West	29	
9	Knicks	East	18	
10	Blazers	West	36	
11	Jazz	West	15	
12				
13				
14				
15				
16				
17				

Our objective is twofold, requiring the application of two simultaneous criteria to the player list (Column A). Specifically, we aim to count the number of unique players who fulfill the following rigorous conditions:

The player must belong to the **West** conference (Criteria Range: Column B).

The player must have scored **greater than 20** points (Criteria Range: Column C).

Applying the Formula in Practice

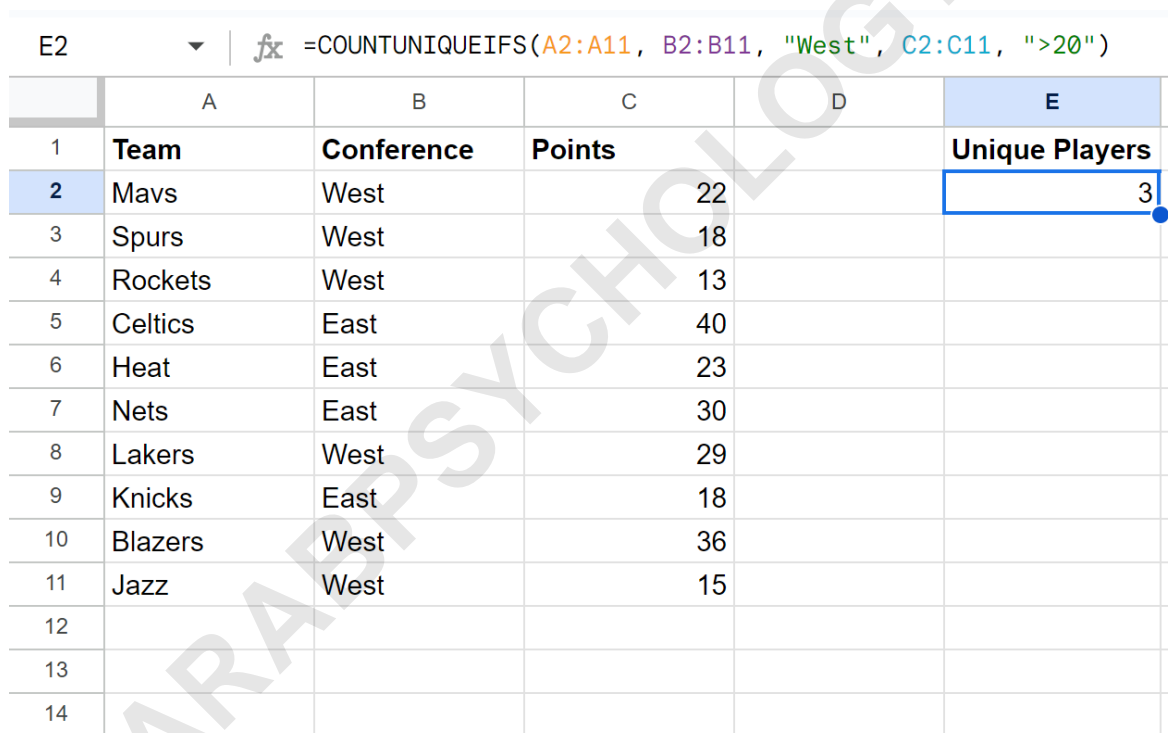
To successfully identify the count of unique players meeting both criteria, we must map our ranges

to the **COUNTUNIQUEIFS** syntax. Column A contains the player names, which is our unique counting range (A2:A11). Column B contains the conference (criteria range 1), and Column C contains the points (criteria range 2).

The resulting formula, structured precisely to reference these ranges and criteria, is as follows. Note the careful use of quotation marks around the text string ("West") and the comparison operator with the numerical value (">20") to ensure correct interpretation by the function.

=COUNTUNIQUEIFS(A2:A11,B2:B11,"West",C2:C11,">20")

We insert this formula into an empty cell, such as cell **E2**, within the spreadsheet. Upon execution, the function processes the ten rows of data, filters them based on the combined criteria, and finally counts how many distinct player names remain in the filtered subset.



	A	B	C	D	E
1	Team	Conference	Points		Unique Players
2	Mavs	West	22		3
3	Spurs	West	18		
4	Rockets	West	13		
5	Celtics	East	40		
6	Heat	East	23		
7	Nets	East	30		
8	Lakers	West	29		
9	Knicks	East	18		
10	Blazers	West	36		
11	Jazz	West	15		
12					
13					
14					

Interpreting the Result and Verification

Upon calculating the formula in cell E2, the output reveals the numerical result. In this example, the formula returns the value **3**. This outcome signifies that there are exactly three unique players within the dataset who are associated with the West conference AND have achieved a score exceeding 20 points in any recorded instance.

To ensure the accuracy of the **COUNTUNIQUEIFS** function, manual verification is highly

recommended, especially when working with complex criteria or new datasets. Verification involves systematically reviewing each row and checking if it satisfies both criteria: Conference is "West" AND Points are greater than 20. We then identify the unique players associated with those qualifying rows.

By manually cross-referencing the rows, we can confirm the distinct individuals who meet the performance standards. The rows that satisfy both conditions are highlighted below. Notice how Player E appears twice in the qualifying rows, but the function correctly counts them only once due to the unique constraint.

E2 **fx** =COUNTUNIQUEIFS(A2:A11, B2:B11, "West", C2:C11, ">20")

	A	B	C	D	E
1	Team	Conference	Points		Unique Players
2	Mavs	West	22		3
3	Spurs	West	18		
4	Rockets	West	13		
5	Celtics	East	40		
6	Heat	East	23		
7	Nets	East	30		
8	Lakers	West	29		
9	Knicks	East	18		
10	Blazers	West	36		
11	Jazz	West	15		
12					
13					
14					

The qualifying unique players are: Player B (Row 3), Player D (Row 6), and Player E (Rows 8 and 9). Since Player E is counted only once, the total unique count is indeed **3**, validating the formula's output.

Advanced Uses and Criteria Flexibility

The power of COUNTUNIQUEIFS extends beyond simple equality and inequality checks. The function supports a wide array of criteria types, allowing for highly nuanced data analysis. For example, you can use comparison operators like less than or equal to (\leq), or even use wildcards for partial text matching.

When using wildcards, the asterisk (*) represents any sequence of zero or more characters, and the question mark (?) represents any single character. For instance, a criterion of "S*" would

match any text string that begins with the letter 'S', allowing analysts to count unique items based on categorical prefixes. Furthermore, the criterion can reference a cell value, such as `A1`, instead of being hardcoded, which makes formulas dynamic and easier to update.

Moreover, this function is frequently paired with date-based criteria. By formatting dates as serial numbers or using functions like `TODAY()`, you can count unique transactions that occurred between two specific dates (e.g., `">="&DATE(2023,1,1)` and `"<"&DATE(2024,1,1)`). This versatility makes **COUNTUNIQUEIFS** a fundamental tool for complex reporting in **Google Sheets**.

Troubleshooting Common Implementation Errors

While **COUNTUNIQUEIFS** is straightforward, several common errors can occur during implementation. The most frequent issue is incorrect referencing of ranges. All criteria ranges must have the exact same number of rows as the counting range to ensure accurate row-by-row mapping. Mismatched range sizes will typically result in a `#VALUE!` error or inaccurate counts.

Another pitfall involves criterion formatting. Numerical criteria using comparison operators (like `">20"`) must always be enclosed in double quotation marks. Text criteria, such as `"West"`, must also be quoted. If the criterion refers to a cell, only the concatenation operator (`&`) is used outside the cell reference (e.g., `">="&E1`, assuming `E1` holds the numerical threshold). Failure to adhere to these quoting rules often leads to incorrect results or function errors.

Finally, ensure that the data types match the expected criteria. If a column contains numerical data formatted as text, the comparison operators might fail to work correctly. Always verify that numbers are stored as numbers and dates are stored as date serial values to guarantee the reliable filtering capabilities of the **COUNTUNIQUEIFS** function.

Note: You can find the complete official documentation for the **COUNTUNIQUEIFS** function in Google Sheets by visiting the official Google Docs Help Center.

Further Learning Resources

To expand your proficiency in [Google Sheets](#) and advanced data analysis techniques, consider exploring tutorials on related conditional functions and array manipulation.

The following tutorials explain how to perform other common tasks in Google Sheets:

Using `SUMIFS` and `AVERAGEIFS` for conditional aggregation.

Implementing array formulas for complex data transformations.

Mastering the `QUERY` function for SQL-like data manipulation.