

Google Sheets: Count Cells Between Two Values

Authored by
Mohammed loot

October 31, 2025

RECOMMENDED CITATION

Mohammed loot (2025). *Google Sheets: Count Cells Between Two Values*.
PSYCHOLOGICAL STATISTICS. Retrieved from
<https://statistics.arabpsychology.com/?p=6657>

Mastering Conditional Data Analysis in Google Sheets

Effective data analysis within [Google Sheets](#) frequently demands precise methods for isolating and quantifying data points. A common, yet critical, requirement is the ability to accurately count the number of numerical values that fall within a specific, predefined [range](#). Whether you are tasked with tracking sales performance, grading student scores, or managing inventory thresholds, knowing how to precisely count items between two boundary values offers indispensable insights into the distribution and characteristics of your dataset.

This comprehensive guide focuses on the definitive solution for this task: leveraging the powerful and flexible [COUNTIFS function](#). Unlike simpler counting methods, **COUNTIFS** is specifically engineered to handle multiple [criteria](#) simultaneously. By applying two conditions--one for the lower limit and one for the upper limit--to the same column, we can define a counting zone, ensuring only relevant data points are included in the final tally.

Through mastering this technique, you gain the ability to quickly summarize large datasets, identify underlying trends, and make far more informed decisions without the tedious process of manual data inspection. We will explore how to set up formulas for both exclusive ranges (excluding the boundaries) and inclusive ranges (including the boundaries), ensuring the resulting count perfectly aligns with your analytical needs.

Deconstructing the COUNTIFS Function for Range Counting

The [COUNTIFS function](#) stands as a cornerstone of conditional calculations in spreadsheet software. It is significantly more powerful than its counterpart, **COUNTIF**, because it processes multiple criteria across various ranges, or as needed for this scenario, multiple criteria applied to a single range. Understanding its fundamental [syntax](#) is the first step toward implementation:

```
=COUNTIFS(range1, criterion1, , ...)
```

range1: This parameter specifies the primary [range](#) containing the numerical data you wish to evaluate. For counting between two values in a single column, this will be your entire target column (e.g., `A:A`) or a specific subset of cells (e.g., `B1:B100`).

criterion1: This is the first condition applied to **range1**, typically defining either the lower or upper boundary. Criteria are always expressed as strings (within quotation marks) and incorporate [logical operators](#) such as `>` (greater than), `<` (less than), `>=` (greater than or equal to), or `<=` (less than or equal to), coupled with the boundary value (e.g., `">50"`).

: These optional pairs allow for the application of additional conditions. Critically, when counting between two values, you will repeat the target range (as **range2**) and apply a second,

complementary criterion (as **criterion2**) to define the other boundary.

To count [cells](#) that fall between two numbers, say 10 and 15, you are asking Google Sheets to check two conditions simultaneously: is the value greater than 10, AND is the value less than 15? The **COUNTIFS** function handles this "AND" logic seamlessly by evaluating both criteria against every cell in the specified range.

For instance, to count all values in column A that are strictly greater than 10 and strictly less than 15, the formula construction is straightforward and highly efficient:

=COUNTIFS(A:A,">10",A:A,"<15")

This formula ensures that only numerical entries that satisfy both the lower boundary condition and the upper boundary condition are included in the final count, providing an exact and precise summary of the data within the defined parameters.

Counting Cells within an Exclusive Range (Strictly Between Boundaries)

An exclusive range count requires that the numerical values be strictly greater than the lower bound and strictly less than the upper bound; the boundary values themselves are excluded from the result. This is often necessary when defining tiers or buckets of data where the endpoints belong to adjacent groups. Let us consider a practical scenario where we analyze a dataset of "Points" and need to count how many entries are greater than 10 but less than 20.

Imagine your data is arranged in a [spreadsheet](#) similar to the visualization below, with the points listed in column B:

	A	B	C	D	E
1	Player	Points			
2	A	10			
3	B	12			
4	C	14			
5	D	19			
6	E	22			
7	F	27			
8	G	25			
9	H	20			
10	I	13			
11	J	30			
12	K	23			
13	L	29			
14	M	25			
15					
16					
17					
18					
19					

To execute this exclusive count, we utilize the [COUNTIFS function](#) with the strict [logical operators](#): `>` for the lower bound and `<` for the upper bound. Specifically, we define two criteria applied to column **B**: one requiring the value to be greater than **10**, and the second requiring it to be less than **20**.

The precise formula required to achieve this exclusive count is:

```
=COUNTIFS(B:B,">10",B:B,"<20")
```

Upon entering this formula into any available [cell](#) within your spreadsheet, the function evaluates the column and returns the count of entries that satisfy both conditions simultaneously. The resulting output, as depicted in the screenshot below, clearly illustrates the calculated result:

D2		<i>fx</i>	=COUNTIFS(B:B, ">10", B:B, "<20")		
	A	B	C	D	
1	Player	Points		Count	
2	A	10		4	
3	B	12			
4	C	14			
5	D	19			
6	E	22			
7	F	27			
8	G	25			
9	H	20			
10	I	13			
11	J	30			
12	K	23			
13	L	29			
14	M	25			
15					
16					

The calculation yields a result of **4**. This is because only the values 12, 14, 16, and 18 meet the strict requirement of being greater than 10 and less than 20. The boundary values (10 and 20) are thus correctly excluded. The visual confirmation below highlights these four values within the specified [range](#):

D2 fx =COUNTIFS(B:B, ">10", B:B, "<20")				
	A	B	C	D
1	Player	Points		Count
2	A	10		4
3	B	12		
4	C	14		
5	D	19		
6	E	22		
7	F	27		
8	G	25		
9	H	20		
10	I	13		
11	J	30		
12	K	23		
13	L	29		
14	M	25		
15				
16				

Counting Cells within an Inclusive Range (Including Boundaries)

Conversely, many analytical tasks require the inclusion of the boundary numbers in the final count, defining an inclusive [range](#). To achieve this, we must count all [cells](#) that are greater than or equal to the lower bound and less than or equal to the upper bound. Fortunately, the robust structure of the [COUNTIFS function](#) is easily adapted for this purpose.

Using the identical dataset of "Points" in column **B**, let us now adjust our goal to count entries that are greater than or equal to **10** and simultaneously less than or equal to **20**. The fundamental principle remains the same--applying two criteria to the same range--but the choice of [logical operators](#) changes. We replace **>** and **<** with their inclusive counterparts: **>=** (greater than or equal to) and **<=** (less than or equal to).

The modified formula necessary for an inclusive count is constructed as follows:

=COUNTIFS(B:B,">=10",B:B,"<=20")

When this formula is applied in your [Google Sheets](#) document, the resulting count will naturally be higher than the exclusive count (assuming the boundary values exist in the data), as it now includes 10 and 20. The following illustration confirms the expected output:

D2	<i>fx</i>	=COUNTIFS(B:B, ">=10", B:B, "<=20")		
	A	B	C	D
1	Player	Points		Count
2	A	10		6
3	B	12		
4	C	14		
5	D	19		
6	E	22		
7	F	27		
8	G	25		
9	H	20		
10	I	13		
11	J	30		
12	K	23		
13	L	29		
14	M	25		
15				
16				

The function correctly identifies **6** values that satisfy the inclusive criteria. These values comprise the original four (12, 14, 16, 18) plus the boundary numbers (10 and 20). This adjustment from 4 to 6 clearly demonstrates the importance of choosing the correct logical operators based on whether the boundaries should be included in the analytical scope. For verification, the values included in this inclusive count are highlighted below:

	A	B	C	D
D2				<code>=COUNTIFS(B:B, ">=10", B:B, "<=20")</code>
1	Player	Points		Count
2	A	10		6
3	B	12		
4	C	14		
5	D	19		
6	E	22		
7	F	27		
8	G	25		
9	H	20		
10	I	13		
11	J	30		
12	K	23		
13	L	29		
14	M	25		
15				
16				
17				

Implementing Robust Formulas: Best Practices and Flexibility

While the basic application of [COUNTIFS](#) is simple, adopting best practices ensures your formulas are robust, flexible, and accurate, particularly when dealing with dynamic or large datasets in [Google Sheets](#). These considerations are vital for preventing errors and simplifying future maintenance of your spreadsheets.

Dynamic Criteria Referencing: Hardcoding boundary values (like ">10") directly into the formula limits flexibility. A superior practice involves referencing separate [cells](#) that contain your lower and upper bounds. For example, if the value 10 is in cell C1 and 20 is in cell C2, you construct the criteria using the concatenation operator (&): `=COUNTIFS(B:B, ">"&C1, B:B, "<"&C2)`. This allows analysts to modify the boundaries simply by changing the values in C1 and C2, without ever touching the core formula.

Data Integrity Check: The **COUNTIFS** function operates on the assumption that the data being evaluated is numerical. If your column mistakenly contains numbers formatted as text, or if there are invisible leading/trailing spaces, the function may fail to identify them correctly. It is essential to ensure your target column uses the correct numerical formatting. If necessary, data cleaning functions like **VALUE()** or **TRIM()** should be employed before running the count.

Range Optimization: For most applications, referencing entire columns (e.g., `B:B`) is convenient and efficient in Google Sheets. However, for exceptionally large spreadsheets where performance is paramount, defining a limited [range](#) corresponding exactly to the occupied rows (e.g., `B1:B5000`) can sometimes slightly improve calculation speed, though this is often unnecessary for modern spreadsheet engines.

The "AND" Logic Principle: Always remember that **COUNTIFS** operates under strict "AND" logic. Every single [criterion](#) provided must be satisfied simultaneously for a specific [cell](#) to be included in the final tally. For counting between two values, this "AND" requirement--must be greater than X AND less than Y--is precisely what makes the function suitable.

Summary and Continuous Improvement of Spreadsheet Skills

Successfully implementing the [COUNTIFS function](#) to count cells between two values is a foundational skill that significantly enhances your data manipulation capabilities in [Google Sheets](#). This technique provides immediate, accurate summaries of data distribution within specified numerical boundaries, which is essential for reporting and deep-dive analysis.

To consolidate your understanding and explore the full potential of this function, always consult the official Google Sheets documentation. This reliable resource offers extensive details regarding [syntax](#), examples, and advanced usage scenarios for **COUNTIFS** and other related functions.

Expanding beyond simple counting, the logic employed here is transferable to other conditional functions. We encourage you to explore the following related operations to further transform your raw data into meaningful, actionable insights:

Using **SUMIFS** for conditional summation (e.g., summing values between 10 and 20).

Applying **AVERAGEIFS** for calculating conditional averages.

Mastering array formulas for complex data processing across a [range](#).

Advanced applications of [Logical Operators](#) in combined formulas.

By continuously integrating these powerful conditional functions, you ensure that your work in [spreadsheets](#) remains efficient, accurate, and impactful, positioning you as a proficient data analyst.