

How to Use MINIFS in Google Sheets to Find Minimum Values in Multiple Ranges

Authored by
stats writer

November 20, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Use MINIFS in Google Sheets to Find Minimum Values in Multiple Ranges*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=98495>

The MINIFS function represents a significant advancement in data analysis capabilities within Google Sheets, offering users a dynamic way to extract minimum values based on conditional logic. Unlike the standard MIN function, which simply finds the smallest number in a specified range, MINIFS allows for the application of one or more specific criteria across parallel ranges. This ability to conditionally filter data before calculating the minimum makes MINIFS an indispensable tool for complex data summary and reporting tasks, moving beyond simple aggregation to nuanced selection based on specific business rules or data categories.

Before the introduction of MINIFS, users often had to rely on complex array formulas involving combinations of `MIN`, `IF`, and `ARRAYFORMULA`, which were notoriously difficult to write, debug, and maintain. MINIFS simplifies this entire process by consolidating conditional minimum calculation into a single, intuitive function. This streamlines workflows, reduces the likelihood of errors, and significantly enhances the readability of complex spreadsheets. Whether you are filtering sales data by region, production times by machine type, or player statistics by team, MINIFS provides the precision required for accurate analysis.

While the name suggests its focus is on finding the minimum, the conditional logic inherent in MINIFS is parallel to other modern conditional functions like SUMIFS, COUNTIFS, and AVERAGEIFS. This consistent naming convention helps users who are already familiar with conditional summation or counting to quickly adapt to calculating conditional minimums. It is crucial to understand that MINIFS is designed to work exclusively with numerical data for the minimum range, while the criteria ranges can contain any data type (text, dates, or numbers) that supports the specified conditions.

Understanding the Core Syntax of MINIFS

The foundation of effective utilization of any spreadsheet function lies in a thorough grasp of its structure. The MINIFS function follows a logical and expandable syntax that allows for flexible application across diverse datasets. The function requires at least three arguments: the range from which the minimum value is sought, the first range to be checked against a condition, and the corresponding condition itself. The structure is built to handle an indefinite number of additional criterion pairs, making it highly adaptable.

The official syntax structure is formally defined as follows:

=MINIFS(range, criteria_range1, criteria1,)

A crucial rule when working with MINIFS is ensuring that all specified ranges--the primary minimum range and all subsequent criteria ranges--must have the exact same number of rows and columns. If there is a mismatch in dimensions, the function will inevitably return an error, typically indicating

an incorrect range size. Adherence to this structural consistency is paramount for the function to correctly map the minimum values to their associated criteria rows during the calculation process. Furthermore, while the first three arguments are mandatory, the capacity to include optional pairs of `criteria_range` and its corresponding `criteria` allows for the construction of highly complex, multi-layered conditional queries.

In practice, the function executes by first identifying all rows within the dataset that satisfy **ALL** the specified conditions simultaneously. Once this filtered subset of data is isolated, the function then proceeds to locate the smallest numerical value exclusively within the corresponding cells of the designated minimum range. This powerful mechanism ensures that the resultant minimum value is not simply the lowest number in the entire column, but rather the lowest number that meets every single requirement stipulated by the user's criteria. The following examples show how to use this syntax in practice.

Key Components of the MINIFS Formula

To master the implementation of MINIFS, it is essential to clearly define the role of each required and optional argument. Understanding these components ensures that data integrity is maintained and that the desired analytical outcome is achieved. The arguments are distinctly categorized based on their purpose: the target range for calculation, and the range and condition pairs used for filtering the data.

range (The Minimum Range): This is the single, essential range of cells from which the smallest numerical value will be ultimately returned. This range must exclusively contain numbers or cell references that resolve to numbers. If this range contains text or non-numerical values, those cells will be ignored during the calculation, or the function may produce an error if the entire range is non-numeric. This is the only range that provides the output value.

criteria_range1 (The Filtering Range): This refers to the range of cells that will be evaluated against the first condition, `criteria1`. This range can contain any data type--numbers, text strings, dates, or boolean values. It serves as the primary filter column, running parallel to the minimum range.

criteria1 (The Condition): This is the specific pattern, value, or logical expression used to test the cells in `criteria_range1`. It determines which rows are included in the minimum calculation. Criteria can be static values (e.g., "Team A", 50), cell references, or expressions incorporating comparison operators (e.g., ">100", "<=DATE(2024,1,1)").

The optional arguments, denoted by the ellipsis (), allow users to layer additional conditions onto the filtering process. For instance, if you are analyzing sales, the first criterion might filter by Region ("West"), and the subsequent pair might filter by Product Type ("Electronics"). The function operates under an inherent **AND** logic, meaning that a row is only considered valid for the

minimum calculation if it satisfies the condition of `criteria1` **AND** the condition of `criteria2`, and so on, for all criteria pairs provided. The practical limit on the number of criteria pairs is extensive, allowing for granular control over the data selection.

When constructing criteria involving text, it is important to enclose the text string in double quotation marks, such as `"Guard"`. When dealing with numerical comparison operators, such as greater than or less than, the operator and the value must also be enclosed in quotes, for example, `">50"`. If the criterion itself is a cell reference (e.g., cell `D1` contains the text `"Manager"`), you do not need quotation marks. However, if you combine an operator with a cell reference (e.g., finding values greater than the number in cell `D1`), you must use concatenation: `">"&D1`. Mastery of these small criteria formatting rules is key to writing robust and error-free MINIFS formulas.

Prerequisites and Data Setup for Examples

To illustrate the practical application of the MINIFS function, we will utilize a sample dataset commonly found in sports analytics. This dataset represents scoring information for various players, allowing us to filter the minimum points based on attributes such as team assignment and player position. This scenario perfectly highlights the function's utility in isolating specific subsets of numerical data within a larger table. Before executing the formulas, it is vital to correctly structure the source data within your Google Sheets environment.

Our sample data includes three core columns: Column A contains the **Team** designation (A or B), Column B lists the **Position** (Guard, Forward, Center), and Column C records the total **Points** scored. The data spans rows 2 through 16. The goal in the subsequent examples will be to determine the minimum value from the 'Points' column (C2:C16) only for those rows that meet our specific criteria in the 'Team' (A2:A16) or 'Position' (B2:B16) columns. This setup clearly delineates the Minimum Range (C2:C16) from the Criteria Ranges (A2:A16 and B2:B16).

Here is a visualization of the dataset we will be referencing throughout the following demonstrations:

	A	B	C	D	E
1	Team	Position	Points		
2	A	Guard	29		
3	A	Guard	30		
4	A	Forward	32		
5	A	Forward	35		
6	A	Forward	28		
7	B	Guard	22		
8	B	Guard	17		
9	B	Guard	19		
10	B	Forward	23		
11	B	Forward	26		
12	C	Guard	19		
13	C	Guard	15		
14	C	Forward	14		
15	C	Forward	19		
16	C	Forward	21		
17					
18					
19					
20					
21					

The consistent alignment of the ranges is the most critical setup requirement. For instance, the value in cell C2 (Points) must correspond exactly to the values in A2 (Team) and B2 (Position). If the dataset were sorted improperly, the calculated minimum value would be based on mismatched player attributes, rendering the analysis invalid. Always confirm that all ranges used in the MINIFS formula are aligned vertically and span the exact same row count.

Example 1: Applying MINIFS with a Single Criterion

The simplest application of the MINIFS function involves filtering the data based on just one condition. In this scenario, we aim to calculate the minimum number of points scored, specifically restricted to players who belong to **Team A**. This requires specifying the minimum range, the criteria range related to the team, and the text string representing the criterion itself. This single condition acts as an immediate filter, drastically reducing the dataset before the minimum calculation takes place.

To achieve this specific conditional minimum, we construct the formula as follows:

=MINIFS(C2:C16, A2:A16, "A")

Let's break down the arguments used in this formula: `C2:C16` is the range containing the numerical values (Points) from which the minimum will be drawn. `A2:A16` is the first criteria range, corresponding to the 'Team' column. Finally, `"A"` is the specific criterion; only rows where the corresponding cell in A2:A16 equals the text "A" are considered. The function first isolates all point values associated with Team A, and then it finds the smallest number within that specific subset. This efficient filtering mechanism is the core strength of MINIFS.

When this formula is executed in Google Sheets, the result clearly demonstrates the conditional minimum calculation. The function processes all 15 rows, rejects those belonging to Team B, and then evaluates the remaining point values for Team A. As shown in the following output, the minimum score for players on Team A is **28**. This result is significantly different from the absolute minimum of the entire C2:C16 range, illustrating the necessity of conditional filtering for accurate, segment-specific data aggregate values.

	A	B	C	D	E
E1	=MINIFS(C2:C16, A2:A16, "A")				
1	Team	Position	Points		
2	A	Guard	29		
3	A	Guard	30		
4	A	Forward	32		
5	A	Forward	35		
6	A	Forward	28		
7	B	Guard	22		
8	B	Guard	17		
9	B	Guard	19		
10	B	Forward	23		
11	B	Forward	26		
12	C	Guard	19		
13	C	Guard	15		
14	C	Forward	14		
15	C	Forward	19		
16	C	Forward	21		
17					
18					
19					
20					

Example 2: Leveraging MINIFS with Multiple Criteria

The true power of the MINIFS function becomes apparent when dealing with multiple, compounding conditions. This capability allows analysts to drill down into highly specific subsets of data, applying multiple filters simultaneously using inherent AND logic. For this demonstration, we will narrow our search further: we want to find the minimum points scored only by players who are on **Team A** and who hold the **Guard** position. This requires linking the criteria for two separate columns (Team and Position) to the Points column.

We start again with our original dataset, which we can reference here:

	A	B	C	D	E
1	Team	Position	Points		
2	A	Guard	29		
3	A	Guard	30		
4	A	Forward	32		
5	A	Forward	35		
6	A	Forward	28		
7	B	Guard	22		
8	B	Guard	17		
9	B	Guard	19		
10	B	Forward	23		
11	B	Forward	26		
12	C	Guard	19		
13	C	Guard	15		
14	C	Forward	14		
15	C	Forward	19		
16	C	Forward	21		
17					
18					
19					
20					
21					

The resulting formula structure must incorporate two full criteria pairs after the initial minimum range:

=MINIFS(C2:C16, A2:A16, "A", B2:B16, "Guard")

In this expanded formula, C2:C16 remains the range of points. The first pair, A2:A16, "A", filters

for Team A. The second pair, B2:B16, "Guard", introduces the constraint that the position must be Guard. The function evaluates each row, and only if both the Team column (A) meets the first criterion AND the Position column (B) meets the second criterion is the corresponding Point value (C) included in the minimum calculation pool. This dual filtering significantly restricts the data, providing a highly specific result.

The execution of this formula isolates the Guard players on Team A and correctly identifies the minimum score among them. As visualized below, the result is **29**. Note that this value differs from the minimum for all of Team A (28, found in Example 1), highlighting how adding more criteria refines the data pool and changes the resulting minimum. This capability is essential for generating detailed reports where intersectional filtering is required, such as finding the lowest cost of a specific product sold in a specific geographic zone.

E1 fx =MINIFS(C2:C16, A2:A16, "A", B2:B16, "Guard")

	A	B	C	D	E
1	Team	Position	Points		29
2	A	Guard	29		
3	A	Guard	30		
4	A	Forward	32		
5	A	Forward	35		
6	A	Forward	28		
7	B	Guard	22		
8	B	Guard	17		
9	B	Guard	19		
10	B	Forward	23		
11	B	Forward	26		
12	C	Guard	19		
13	C	Guard	15		
14	C	Forward	14		
15	C	Forward	19		
16	C	Forward	21		
17					
18					
19					
20					

Handling Different Data Types and Operators

While the previous examples focused on exact match text criteria, the MINIFS function is highly versatile and supports various data types and comparison operators, allowing for much more

complex and dynamic filtering. It is not limited to text equality; users can employ numerical comparisons, date filtering, and even wildcard characters for partial text matching. Understanding how to format these advanced criteria correctly is crucial for maximizing the function's analytical power within Google Sheets.

When dealing with numerical data, criteria often involve comparison operators. These must always be enclosed in double quotes, even if the result is a number. For instance, to find the minimum points scored by players who scored more than 40 points, you would use the criterion "`>40`". If you wanted to include the value 40, you would use "`>=40`". Similarly, if you are filtering date ranges, you might look for minimum values associated with transactions that occurred before January 1, 2024, using a criterion like "`<&DATE(2024, 1, 1)`", where the operator is quoted and concatenated with the date value generated by the DATE function.

MINIFS also supports powerful wildcard characters for flexible text matching within its criteria arguments. The asterisk (*) represents any sequence of zero or more characters, and the question mark (?) represents any single character. For example, if you wanted to find the minimum value associated with any team name starting with the letter 'C', the criterion would be "`C*`". If you were looking for minimum scores from players whose position had exactly five letters (like "Guard"), you could use "`?????`" as the criterion. These wildcards enable robust pattern matching, which is especially useful when dealing with inconsistent or user-generated text data.

It is important to remember the inherent challenges when dealing with empty cells or zero values. If the range for the minimum calculation contains zeros, the function will correctly return 0 if that value meets all the specified conditions. If a criteria range contains blank cells, you can specifically target these using the criterion "" (empty quotes) to include rows where the cell is blank, or use "`<>&`" to exclude blank cells. Careful planning regarding how nulls and zeros should be handled in your specific dataset is vital for avoiding misleading aggregate values.

Advanced Applications and Performance Considerations

Beyond simple filtering, the MINIFS function can be strategically combined with other Google Sheets functionalities to achieve highly sophisticated data manipulation. For instance, it can be nested within functions like `IFERROR` to provide graceful error handling, or used in conjunction with dynamic array functions to generate conditional summaries that automatically update as the source data changes. Furthermore, the ability to reference criteria from calculated cells (rather than hardcoded text) allows the formula to be driven by user input or intermediate calculations, offering unparalleled interactive reporting capabilities.

One common advanced application involves using dynamic named ranges or the `INDIRECT` function within the MINIFS syntax. While this can complicate the readability, it provides a powerful way to automate reports that pull data from dynamically selected tabs or datasets without

constantly rewriting the formula references. For example, if a user selects a month name from a dropdown list, the MINIFS function can be designed to target the sheet corresponding to that month, calculating the minimum cost based on specific criteria within the selected range.

However, users should remain mindful of performance, especially when dealing with extremely large datasets (tens of thousands of rows). Although MINIFS is generally optimized compared to older array-based conditional formulas, incorporating a large number of criteria (e.g., more than five pairs) or referencing entire columns (e.g., `A:A`) across a massive spreadsheet can slow down recalculation times. Best practice dictates using explicit ranges (e.g., `A2:A5000`) rather than open-ended columns whenever possible, and ensuring that criteria ranges are tightly defined to minimize the computational burden and maintain optimal spreadsheet speed when calculating aggregate values.

Conclusion and Summary of Best Practices

The MINIFS function stands out as a critical tool for conditional data analysis in Google Sheets, enabling users to efficiently determine the minimum value within a range subject to multiple, specific criteria. Its straightforward syntax and high performance make it superior to legacy array formula solutions, providing cleaner, more maintainable code for conditional summaries and reporting dashboards. Mastery of this function is essential for anyone conducting serious analytical work within the Google ecosystem.

To ensure successful and error-free implementation of MINIFS, adhere to these key best practices:

Ensure Range Consistency: Always verify that the minimum range and all criteria ranges are perfectly aligned and cover the exact same number of rows. Mismatched ranges are the most frequent cause of errors.

Format Criteria Correctly: Enclose text criteria and comparison operators (like `>50`) in double quotation marks. Use concatenation (`"<"&D1`) when combining operators with cell references.

Utilize Explicit Ranges: Avoid referencing entire columns (e.g., `A:A`) in very large spreadsheets to improve calculation speed; define specific, bounded ranges instead.

Understand AND Logic: Remember that MINIFS operates strictly on AND logic; all specified conditions must be true for a row to be included in the minimum calculation.

By leveraging the power of conditional minimum calculation, analysts can transform raw data into targeted, actionable insights. The flexibility to incorporate numerous criteria means that MINIFS is powerful enough to handle almost any filtering requirement, whether you are finding the cheapest component under specific supplier constraints or the fastest time recorded under controlled experimental conditions.