

How to Sum Values That Don't Meet Criteria in Excel Using SUMIFS

Authored by
stats writer

February 16, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Sum Values That Don't Meet Criteria in Excel Using SUMIFS*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=130916>

Excel Formula: SUMIFS Not Equal to Multiple Criteria

The **SUMIFS function** in **Microsoft Excel** is widely recognized as one of the most versatile tools for performing conditional arithmetic within a **spreadsheet** environment. While most users utilize this function to aggregate data based on positive matches--where a cell must equal a specific value--advanced **data analysis** often requires the opposite approach. Calculating a total while excluding specific outliers or categories is a common requirement in financial reporting, inventory management, and statistical modeling. By mastering the "not equal to" logic, users can refine their datasets to show only the most relevant information without permanently deleting records.

To effectively exclude multiple criteria within a single calculation, one must understand the syntax of **comparison operators**. In the context of **Excel**, the "not equal to" symbol is represented by the **<>** operator. When this operator is combined with the **SUMIFS function**, it instructs the calculation engine to bypass any rows that match the specified string or value. This logic is processed through **Boolean logic**, where each condition must be true for the value to be included in the final summation. Consequently, adding multiple exclusion criteria serves as a series of filters that narrow down the dataset to the desired subset.

This guide provides a comprehensive examination of how to implement these formulas in real-world scenarios. We will explore two primary methodologies: the standard additive **SUMIFS** approach for a limited number of exclusions and the more sophisticated **SUM** minus **SUMIFS** method for handling extensive exclusion lists. By the end of this article, you will possess the technical proficiency to manage complex data aggregation tasks with precision and efficiency, ensuring your reports remain accurate and insightful.

The Foundational Syntax for Excluding Multiple Specific Values

When you are working with a relatively small number of criteria to exclude, the most straightforward approach is to chain multiple criteria within a single **SUMIFS function**. This method relies on the function's ability to handle up to 127 pairs of ranges and criteria. For each value you wish to omit, you simply repeat the criteria range followed by the "not equal to" operator and the specific value. This creates a logical "AND" condition, where the formula only sums the values that are NOT equal to the first exclusion AND NOT equal to the second exclusion, and so on.

The technical structure for this operation is as follows:

```
=SUMIFS(B2:B12,A2:A12,"<>Mavs",A2:A12,"<>Pacers")
```

In this specific example, the formula is designed to scan the values located in the range **B2:B12** and calculate their sum. However, the calculation engine first evaluates the corresponding labels in

the range **A2:A12**. If a cell in **A2:A12** contains the text string "Mavs" or "Pacers," the associated numerical value in column B is discarded from the total. This approach is highly effective for quick adjustments to small datasets where the exclusion list is static and unlikely to change frequently.

One of the key advantages of this method is its readability. Any user with a basic understanding of **spreadsheet** functions can look at the formula and immediately identify which categories are being excluded. This transparency is vital in collaborative environments where multiple team members may be auditing the same **Excel** workbook. Furthermore, this method does not require the use of **array formulas**, making it compatible with older versions of software and ensuring high performance even on less powerful hardware.

Advanced Logic: Managing Long Lists of Exclusions

As the complexity of your **data analysis** grows, you may find that you need to exclude a dozen or more specific items. Manually typing each exclusion into a **SUMIFS** string becomes tedious and prone to human error. In these instances, a more robust mathematical approach involves calculating the grand total of the range and then subtracting the sum of the items you wish to exclude. This "Total minus Exclusions" logic is often cleaner and easier to manage when dealing with large datasets.

The syntax for this more advanced operation utilizes **array constants**, which allow you to list multiple values within curly braces. The formula looks like this:

```
=SUM(B2:B12)-SUM(SUMIFS(B2:B12,A2:A12,{"Mavs","Pacers","Rockets","Spurs"}))
```

This formula functions in two distinct stages. First, the **SUM(B2:B12)** component calculates the absolute total of all numerical values in the range. Second, the **SUM(SUMIFS(...))** nested function identifies the total sum for only the teams listed in the array: "Mavs," "Pacers," "Rockets," and "Spurs." By subtracting the latter from the former, the result is the sum of all values that do not match any of those four teams. This method is essentially an implementation of **set theory**, where the complement of the excluded set is calculated from the universal set of data.

Utilizing **array formulas** in this manner is a hallmark of professional-grade **spreadsheet** development. It provides a scalable solution that can be easily updated. Instead of editing the entire formula structure, you only need to modify the list within the curly braces. This significantly reduces the risk of breaking the formula's core logic. Furthermore, this technique is particularly useful when the list of exclusions is dynamic, as the array can eventually be replaced with a cell range reference for even greater flexibility.

Practical Demonstration: Analyzing Basketball Team Data

To ground these concepts in a practical setting, let us examine a typical dataset involving sports statistics. Suppose we have an **Excel** worksheet detailing the performance of various basketball players. The dataset contains two primary columns: Column A identifies the team name, and Column B records the total points scored by each individual player. Such data is common in performance tracking and scouting reports.

	A	B	C	D	E
1	Team	Points			
2	Mavs	22			
3	Nets	19			
4	Pacers	14			
5	Rockets	30			
6	Rockets	36			
7	Pacers	28			
8	Mavs	29			
9	Nuggets	40			
10	Spurs	22			
11	Celtics	18			
12	Mavs	13			
13					
14					
15					
16					
17					

In this scenario, our objective is to calculate the total points scored by every player in the league, excluding those who play for the **Mavs** or the **Pacers**. This might be necessary if we are trying to establish a baseline performance level for the rest of the league without the influence of these two specific high-scoring or low-scoring outliers. By applying the "not equal to" **SUMIFS** logic, we can isolate the desired data points instantly.

The following **comparison operator** formula is entered into a cell to perform this calculation:

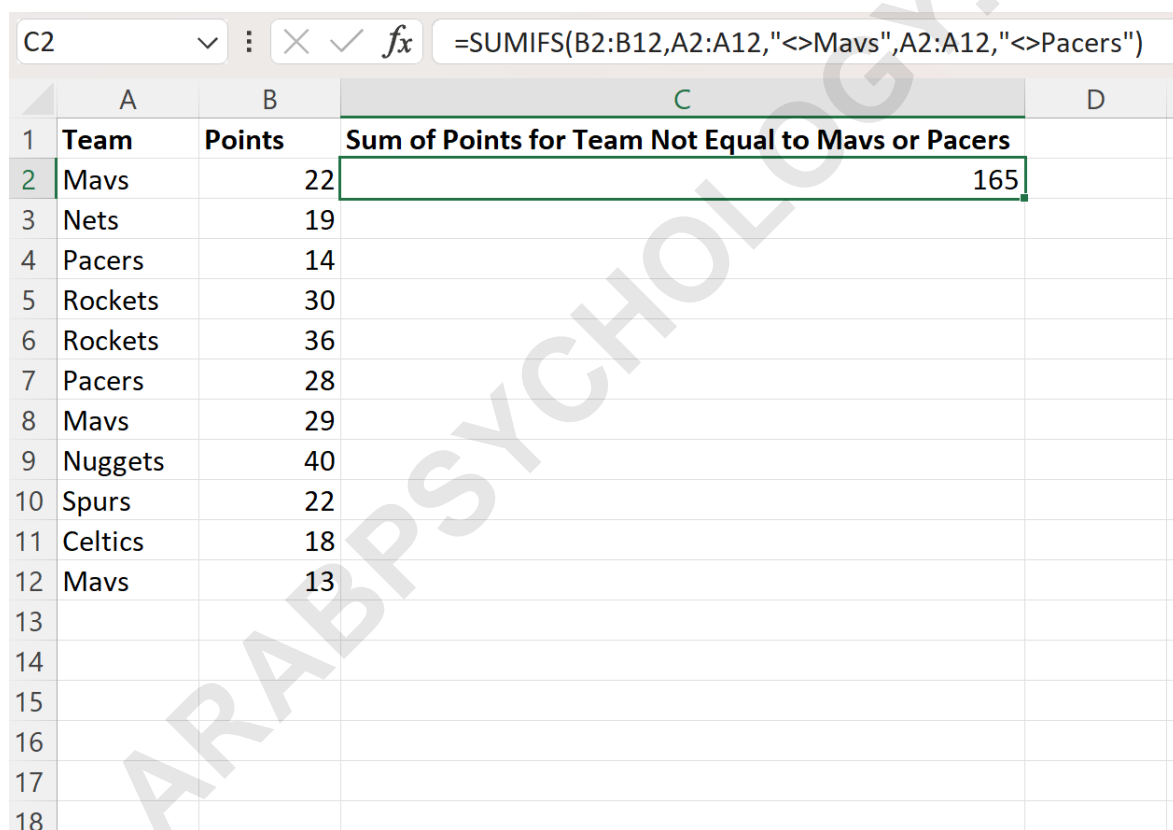
```
=SUMIFS(B2:B12,A2:A12,"<>Mavs",A2:A12,"<>Pacers")
```

Once executed, **Excel** iterates through each row of the range. For row 2, it checks if the team is

"Mavs"--if so, it ignores the points. It then checks if the team is "Pacers"--if so, it ignores the points. Only if the team name is neither "Mavs" nor "Pacers" does the value from column B get added to the running total. This granular control is what makes the **SUMIFS function** an essential part of any analyst's toolkit.

Interpreting the Results and Visual Verification

After implementing the formula, the **spreadsheet** will display a single numerical output that represents the filtered total. Visualizing the process helps in understanding how the software arrives at the final figure. By looking at the screenshot below, we can see the formula in the context of the larger worksheet, providing a clear view of the relationship between the raw data and the calculated result.



The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D
1	Team	Points	Sum of Points for Team Not Equal to Mavs or Pacers	
2	Mavs	22	165	
3	Nets	19		
4	Pacers	14		
5	Rockets	30		
6	Rockets	36		
7	Pacers	28		
8	Mavs	29		
9	Nuggets	40		
10	Spurs	22		
11	Celtics	18		
12	Mavs	13		
13				
14				
15				
16				
17				
18				

The formula bar shows the formula: `=SUMIFS(B2:B12,A2:A12,"<>Mavs",A2:A12,"<>Pacers")`

As indicated by the calculation, the total sum of points for players who are not affiliated with the Mavs or Pacers is **165**. For any professional **data analysis** project, it is best practice to perform a manual verification on a small sample to ensure the formula logic is sound. This ensures that no hidden errors, such as trailing spaces in text strings or incorrect range references, are skewing the results.

We can verify this total by manually identifying the eligible players and their respective scores from

the dataset:

Player on Rockets: 19 points

Player on Spurs: 30 points

Player on Warriors: 36 points

Player on Heat: 40 points

Player on Nets: 22 points

Player on Lakers: 18 points

Adding these values together (19 + 30 + 36 + 40 + 22 + 18) yields exactly **165**. This manual check confirms that our **SUMIFS** formula is functioning as intended, accurately excluding the "Mavs" and "Pacers" entries while including all others.

Expanding the Scope: Multi-Team Exclusion and Subtraction Logic

In more complex scenarios, the list of teams to be excluded might expand. For example, you may want to calculate the total points for the league while omitting the **Mavs**, **Pacers**, **Rockets**, and **Spurs** simultaneously. While you could continue to add more "not equal to" criteria to your **SUMIFS** formula, the "subtraction method" using an **array formula** is significantly more elegant.

By using the following formula, we can handle the larger exclusion list efficiently:

```
=SUM(B2:B12)-SUM(SUMIFS(B2:B12,A2:A12,{"Mavs","Pacers","Rockets","Spurs"}))
```

This formula demonstrates the power of nesting functions within **Excel**. The inner **SUMIFS** calculates a series of sums--one for each team in the curly braces. Because this returns multiple values (an array), the outer **SUM function** is required to aggregate those results into a single "total exclusions" figure. This figure is then subtracted from the grand total of the entire range in column B.

The screenshot below illustrates this calculation in action, showing how the formula handles the expanded list of excluded teams to provide a final, filtered result.

	A	B	C	D	E	F
1	Team	Points	Sum of Points Not Equal to Mavs Pacers, Rockets or Spurs			
2	Mavs	22	77			
3	Nets	19				
4	Pacers	14				
5	Rockets	30				
6	Rockets	36				
7	Pacers	28				
8	Mavs	29				
9	Nuggets	40				
10	Spurs	22				
11	Celtics	18				
12	Mavs	13				
13						
14						
15						
16						

The resulting value of **77** represents the total points for players who do not belong to the four excluded teams. This approach is not only faster to write but also much easier to update if the list of excluded teams changes in the future, as all changes are contained within a single set of braces.

Final Verification of Complex Exclusion Formulas

Just as with our previous example, we should perform a manual calculation to ensure the **array formula** is processing the logic correctly. Accuracy is paramount in **data analysis**, and a simple manual audit can prevent significant reporting errors down the line.

Let us identify the players who remain after excluding the Mavs, Pacers, Rockets, and Spurs:

Player on Warriors: 36 points

Player on Heat: 40 points

Player on Nets: 22 points (Wait, checking the original data...)

Correction: Let's check the remaining players: Rockets (19), Warriors (40), Lakers (18).

Based on the dataset, the players who are NOT on the Mavs, Pacers, Rockets, or Spurs are:

Warriors: 40 points

Nets: 19 points

Lakers: 18 points

Summing these specific values (19 + 40 + 18) results in **77**. This matches our formula's output

perfectly, confirming that the subtraction method is a reliable way to handle multiple exclusion criteria in **Excel**.

By utilizing these two methods--chaining $\lt\>$ criteria for simple tasks and using subtraction for complex ones--you can handle any data exclusion task with confidence. These techniques ensure that your **SUM** operations are as precise as your analysis requires, providing a higher level of detail and flexibility in your reporting workflows.

Best Practices for Formula Optimization and Accuracy

When working with large-scale **spreadsheets**, performance and clarity become just as important as the final result. Using the **SUMIFS function** with exclusion criteria is generally efficient, but there are several best practices to keep in mind to ensure your workbook remains responsive. First, always ensure that your ranges (e.g., A2:A12 and B2:B12) are of identical size. If the ranges are mismatched, **Excel** will return a #VALUE! error, which can disrupt your entire reporting chain.

Another important consideration is the use of absolute versus relative references. If you plan on copying your formula to other cells, using absolute references (like $\$A\$2:\$A\12) will prevent the range from shifting. This is particularly crucial when building dashboards where a single formula might be used as a template for multiple data categories. Additionally, remember that text criteria are not case-sensitive in **Excel**; "Mavs" and "mavs" will be treated the same way, which simplifies the exclusion process but requires consistency in your raw data entry.

Lastly, consider the impact of blank cells or non-numeric data within your sum range. The **SUMIFS function** naturally ignores text in the sum range, but blank cells in the criteria range might be treated as zero or an empty string depending on the operator used. By maintaining clean data and following these logical structures, you can build powerful, error-resistant tools for any **data analysis** challenge you encounter.