

Use SUMIF with OR in Excel

Authored by
Mohammed looti

October 31, 2025

RECOMMENDED CITATION

Mohammed looti (2025). *Use SUMIF with OR in Excel*. PSYCHOLOGICAL STATISTICS.
Retrieved from <https://statistics.arabpsychology.com/?p=6636>

In the realm of [Microsoft Excel](#), mastering conditional functions such as [SUMIF](#) is essential for effective [data analysis](#). While [SUMIF](#) efficiently sums values based on a single condition, real-world reporting often requires more complex logical tests involving the [OR](#) operator. This necessitates calculating a sum if a value meets condition A [OR](#) condition B, a capability not natively built into the standard [SUMIF](#) syntax.

Fortunately, [Excel](#) offers sophisticated workarounds to integrate the power of [SUMIF](#) or its multi-criteria sibling, [SUMIFS](#), with [OR](#) logic. This tutorial will detail two distinct and powerful strategies for achieving conditional summation using alternative criteria, enabling you to manage diverse data aggregation challenges within your spreadsheets effectively.

Understanding SUMIF and the Need for OR Logic

The [SUMIF](#) function serves as the foundation for conditional summation in Excel. Its design is strictly focused on summing numerical values within a designated range based on a singular, predefined [criteria](#). For example, you might effortlessly calculate total revenue for a specific department or sum expenses associated with a single category, utilizing its straightforward syntax of (Range, Criterion, Sum_Range).

However, its utility quickly diminishes when the requirements involve multiple, non-mutually exclusive conditions. Consider a scenario where you must sum all transactions categorized as "Retail" [OR](#) "Wholesale." Since standard [SUMIF](#) only handles one logical test, it cannot satisfy this requirement. Furthermore, while [SUMIFS](#) accommodates multiple criteria, it is fundamentally built upon [AND logic](#), meaning all conditions must be met simultaneously, rather than offering the necessary alternative, [OR](#) functionality.

To successfully implement true [OR](#) conditions for summation, we must employ creative array processing or additive function structures. The following two methods represent the most robust and common techniques used by Excel professionals to bypass the built-in limitations of conditional functions, allowing for precise data aggregation across complex logical sets.

Strategy 1: Using SUM and Array Constants for Single-Column OR

This first methodology is designed specifically for scenarios where the [OR](#) conditions all relate to a single column range. It ingeniously combines the flexibility of the [SUMIFS](#) function with an [array constant](#), wrapping the whole structure within an outer [SUM](#) function.

Instead of manually writing separate summation functions for each alternative criterion, this technique allows you to supply a list of [criteria](#) directly to a single instance of [SUMIFS](#). Excel treats the [array constant](#) as multiple criteria simultaneously, calculating a distinct sum for every item in the array. The mandatory outer [SUM](#) function then aggregates these calculated partial

sums into the definitive grand total.

=SUM(SUMIFS(B2:B13, A2:A13,{"Value1","Value2", "Value3"}))

In this generic example, `B2:B13` is the [sum range](#), containing the values intended for aggregation. `A2:A13` is the criteria range being evaluated. The crucial component is the curly-braced list `{"Value1","Value2", "Value3"}`, which acts as the [array constant](#), serving as the set of alternative [criteria](#). This clean solution ensures that values in the sum range are included if the corresponding cell in the criteria range matches any one of the specified values.

Strategy 2: Adding Multiple SUMIF Functions for Cross-Column OR

When your summation criteria must evaluate conditions across different columns, or if you prefer a method that explicitly breaks down each logical test, the additive structure becomes the preferred solution. This technique involves creating a separate, standard [SUMIF](#) function for every condition required, and then combining their resulting sums using the addition operator.

This methodology is invaluable when the goal is to sum data if a criterion is met in Column A [OR](#) if a completely separate criterion is met in Column B. Since each [SUMIF](#) functions independently, it calculates a partial sum specific to its condition. By adding these partial sums together, the desired [OR](#) effect is achieved, providing a clear and traceable calculation path.

=SUMIF(A2:A13,"Value1", C2:C13)+SUMIF(B2:B13,"Value2", C2:C13)

In this formula, the first `SUMIF` aggregates values from `C2:C13` based on a condition found in `A2:A13`, while the second `SUMIF` aggregates values from the same sum range (`C2:C13`) based on a condition found in a different range, `B2:B13`. The total calculated sum therefore includes any row that satisfies the condition in Column A [OR](#) the condition in Column B. A key consideration with this method is the potential for double-counting, which occurs if a single row satisfies both criteria simultaneously. If overlaps are expected, a more complex formula involving [SUMIFS](#) subtraction may be necessary to ensure accuracy.

Practical Demonstration: Setting Up the Dataset

To fully grasp the mechanics of these powerful techniques, we will now apply them to a practical, illustrative dataset. This hands-on demonstration is crucial for visualizing how each method operates in a real-world context and understanding the specific scenarios best suited for each implementation. Our example data simulates player statistics, detailing critical information such as the player's Team, their Position, and the Points they have scored.

This consistent dataset will be used across both upcoming examples, offering a clear comparison of how different [OR](#) logic strategies yield results based on varying conditional requirements. We encourage readers to pay close attention to the specific column references and [criteria](#) used in each formula to reinforce learning.

	A	B	C	D	E	F
1	Team	Position	Points			
2	Mavs	Guard	12			
3	Rockets	Forward	8			
4	Cavs	Guard	29			
5	Spurs	Guard	20			
6	Mavs	Forward	22			
7	Nets	Center	23			
8	Thunder	Forward	25			
9	Heat	Guard	19			
10	Rockets	Guard	11			
11	Magic	Center	30			
12	Cavs	Guard	24			
13	Hornets	Forward	28			
14						
15						
16						
17						
18						
19						

The table above serves as our foundation. The columns are clearly defined: `Team` (e.g., Mavs, Rockets), `Position` (e.g., Guard, Forward, Center), and the numerical column, `Points`. Our goal in the subsequent sections is to conditionally sum the `Points` column based on complex logical requirements using the two advanced summation techniques we have just outlined.

Example 1: Implementing Single-Column OR using Array Constants

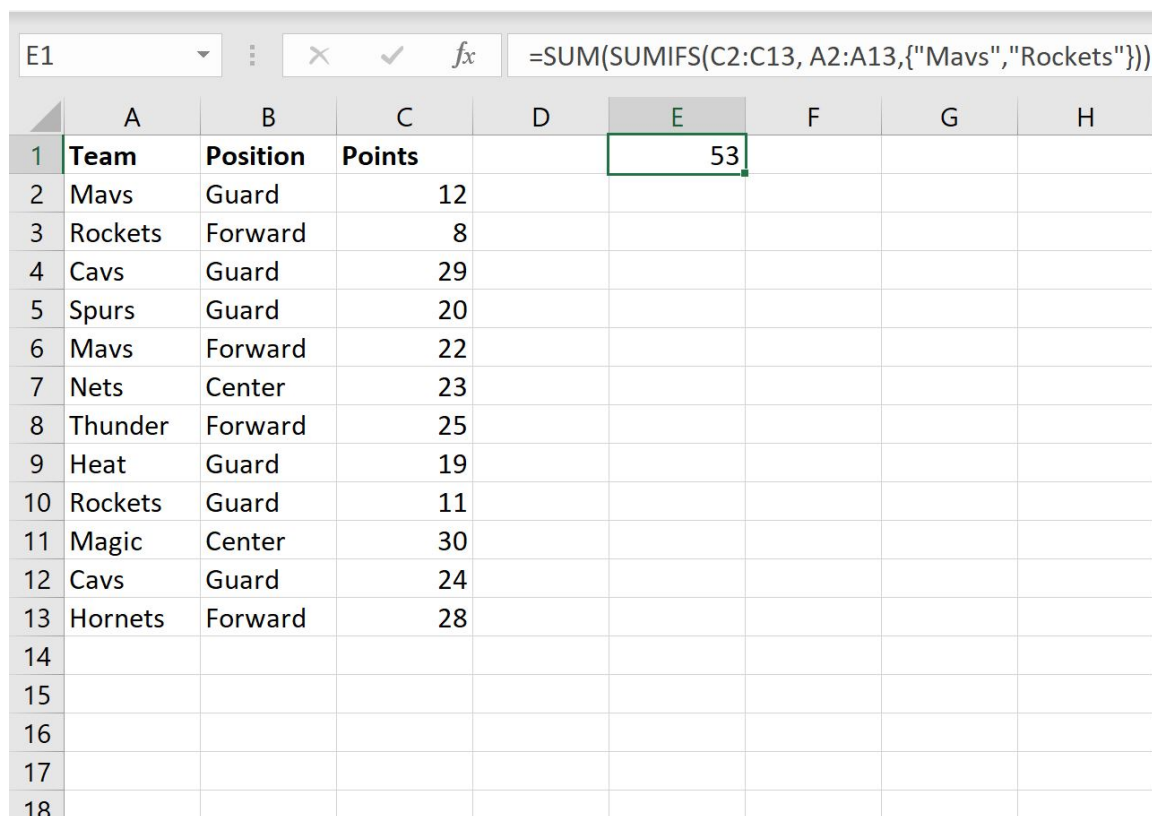
For our first application, we will deploy Strategy 1: the combination of [SUM](#) and [SUMIFS](#) with an [array constant](#). The objective is to calculate the total `Points` scored exclusively by players belonging to either the "Mavs" [OR](#) the "Rockets" team. Since both conditional tests apply to the same column (`Team`), this method offers the most concise and powerful solution.

The formula is structured to instruct [SUMIFS](#) to simultaneously evaluate the `Team` column (A2:A13) for matches against both "Mavs" and "Rockets." For every successful match, the corresponding value from the `Points` column (C2:C13) is summed internally. The wrapping [SUM](#)

function then takes the two intermediate sums (Mavs' total and Rockets' total) and produces the final aggregate.

=SUM(SUMIFS(C2:C13, A2:A13,{"Mavs","Rockets"}))

The effectiveness of the [array constant](#) `{"Mavs","Rockets"}` is clear here. It forces Excel to execute dual calculations internally, one for each team. The output of these two calculations is then seamlessly combined by the outer [SUM](#) function.



	A	B	C	D	E	F	G	H
1	Team	Position	Points		53			
2	Mavs	Guard	12					
3	Rockets	Forward	8					
4	Cavs	Guard	29					
5	Spurs	Guard	20					
6	Mavs	Forward	22					
7	Nets	Center	23					
8	Thunder	Forward	25					
9	Heat	Guard	19					
10	Rockets	Guard	11					
11	Magic	Center	30					
12	Cavs	Guard	24					
13	Hornets	Forward	28					
14								
15								
16								
17								
18								

As depicted in the accompanying screenshot, executing this formula yields a precise total of **53**. This numerical result confirms the sum of all points for players meeting the "Mavs" criterion [OR](#) the "Rockets" criterion, confirming the efficiency of the array constant approach for single-column [OR](#) logic.

Example 2: Implementing Cross-Column OR using Multiple SUMIFs

Moving to Strategy 2, we will now use multiple additive [SUMIF](#) functions. Our specific goal is to aggregate the total `Points` for players who are either members of the "Mavs" team [OR](#) whose `Position` is "Center." This scenario explicitly demands checking conditions across two distinct column ranges (`Team` in Column A and `Position` in Column B).

Given that the conditions span separate columns, the single [array constant](#) method is unsuitable. Instead, we rely on the transparency and flexibility of summing two individual [SUMIF](#) calculations. Each part of the formula is responsible for isolating and summing data based on its unique condition.

=SUMIF(A2:A13,"Mavs",C2:C13)+SUMIF(B2:B13,"Center",C2:C13)

The first portion, `SUMIF(A2:A13,"Mavs",C2:C13)`, computes the total points accumulated by players on the "Mavs." The second segment, `SUMIF(B2:B13,"Center",C2:C13)`, determines the total points for all players classified as "Center." The final addition operation unites these two independent sums. A crucial detail for this specific dataset and query is the lack of overlap, guaranteeing that no points are counted twice.

	A	B	C	D	E	F	G	H	I
1	Team	Position	Points		87				
2	Mavs	Guard	12						
3	Rockets	Forward	8						
4	Cavs	Guard	29						
5	Spurs	Guard	20						
6	Mavs	Forward	22						
7	Nets	Center	23						
8	Thunder	Forward	25						
9	Heat	Guard	19						
10	Rockets	Guard	11						
11	Magic	Center	30						
12	Cavs	Guard	24						
13	Hornets	Forward	28						
14									
15									
16									
17									
18									
19									
20									

The resulting total, as clearly visible in the image, is **87**. This figure represents the combined points from players who satisfy the "Mavs" team [criteria](#) **OR** the "Center" position [criteria](#). This outcome powerfully illustrates the adaptability of using multiple [SUMIF](#) functions to manage **OR** conditions spanning diverse columns.

Conclusion: Mastering Conditional Summation

Successfully integrating [OR](#) logic into your conditional summation formulas is a significant step toward advanced [data analysis](#) in [Excel](#). By mastering the two methods outlined--utilizing [SUM](#) with [SUMIFS](#) and an [array constant](#) for single-column criteria, or summing multiple individual [SUMIF](#) functions for cross-column criteria--you gain the ability to handle highly complex conditional summation tasks with efficiency and accuracy.

The decision of which method to use should be guided entirely by the spatial arrangement of your [criteria](#). When all alternative conditions reside within the same range, the array constant approach offers unmatched conciseness. Conversely, when conditions are distributed across disparate columns, the method of adding multiple [SUMIF](#) results provides essential flexibility and clear logic.

Consistent practice with these advanced formulas using your own worksheets is the best way to develop proficiency. A deep understanding of these strategies will significantly enhance your ability to extract meaningful and nuanced insights from large datasets, thereby elevating your overall [Excel](#) proficiency.

Additional Resources

For those dedicated to deepening their expertise in [Excel](#), the following tutorials explain how to perform other common [SUMIF](#) operations: