

Excel: Use AVERAGEIFS with OR Condition

Authored by
stats writer

November 17, 2025

RECOMMENDED CITATION

stats writer (2025). *Excel: Use AVERAGEIFS with OR Condition*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=94702>

Excel stands as a cornerstone in the world of business intelligence, offering unmatched capabilities for financial modeling, statistical calculations, and comprehensive data analysis. Its versatility allows professionals across various sectors--from finance and engineering to marketing and logistics--to efficiently manage, manipulate, and visualize vast datasets. Among its suite of sophisticated functions, the ability to calculate averages based on conditional criteria is paramount for deriving meaningful insights. Understanding how to apply functions like AVERAGEIFS is essential for anyone seeking to move beyond basic spreadsheet operations and perform precise, segmented data assessments over time or across different data subsets. This function allows for rapid and accurate calculation of averages only for records that meet specified requirements, dramatically streamlining the analytical process and enhancing decision-making accuracy in complex environments.

While the standard implementation of the AVERAGEIFS function is highly intuitive when dealing with conjunctions (where multiple criteria must all be true, known as AND logic), the requirement for disjunctions--where the average should include values meeting one condition OR another--presents a common analytical challenge for many users. The default function structure is inherently built around AND logic, meaning that if you list multiple criteria, a data point must satisfy all of them simultaneously to be included in the calculation. However, sophisticated data analysis frequently demands flexibility, such as finding the average metric for items categorized as 'High Priority' OR 'Medium Priority.' This frequent scenario necessitates a different approach, one that bypasses the built-in conjunctive constraints of the standard function structure by employing advanced array techniques.

This detailed guide aims to demystify the process of applying the AVERAGEIFS functionality with OR condition logic in Excel. We will explore the limitations of the standard function, introduce the robust technique required to achieve the desired disjunctive calculation, and provide a clear, practical example using relevant data. Mastering this technique--which involves integrating the **AVERAGE** function with an **IF** statement and leveraging array formula processing capabilities--is a crucial step for advanced data handling in spreadsheets, enabling users to perform sophisticated filtering and reporting that accurately reflects complex business requirements.

Understanding the Standard AVERAGEIFS Function and AND Logic

The native AVERAGEIFS function is explicitly designed to calculate the arithmetic mean of a range of cells that satisfy all criteria defined by the user. Its fundamental operational mechanism is inherently based on AND logic. This means that if you specify two or more criteria, say, Criterion 1 (e.g., Department = Finance) and Criterion 2 (e.g., Status = Active), only those numerical values where both conditions are simultaneously TRUE will be included in the final calculation. This conjunction model is highly effective for narrowing down results based on a strict intersection of defined parameters, ensuring high precision in standard conditional reporting tasks.

The syntax of the standard function requires the user to first define the range of values to be averaged, followed by pairs of criteria ranges and their corresponding criteria values. This mandatory pairing defines the filtering rule. For instance, to find the average profit in the Central region for products sold in January, you would specify the profit range, followed by the region range and the criterion "Central," and then the month range and the criterion "January." Excel processes these inputs sequentially, generating a temporary logical array where only cells meeting the conjunctive criteria evaluate as TRUE, allowing the average to be computed accurately against that filtered subset.

By default, the **AVERAGEIFS** function in Excel utilizes AND logic to calculate the average value of cells that meet multiple conditions. This robust built-in behavior ensures that when users define multiple constraints, the calculation is strictly limited to the intersection of those criteria, guaranteeing precision in typical data segmentation tasks.

This strong reliance on AND logic clarifies why attempting to directly input multiple criteria for the same field (e.g., Position="Guard", Position="Forward") into the standard structure fails to function as an OR condition. The function attempts to check if a single cell can simultaneously equal both "Guard" AND "Forward," which is a logical impossibility, invariably resulting in a calculated average of zero or an error. Overcoming this fundamental structural limitation requires adopting a specialized method that allows Excel to evaluate multiple criteria against the same range disjunctively.

The Challenge: Implementing OR Logic for Multiple Criteria

The most significant analytical gap encountered by professionals leveraging conditional functions is how to correctly introduce the OR condition when complex averaging is required. When working with multifaceted datasets, it is frequently essential to calculate a metric based on records matching Condition A or Condition B, rather than demanding that both conditions be true. This disjunctive requirement is common in various analytical scenarios, such as calculating the average transaction value for customers in State X OR customers in State Y, or finding the average duration of tasks marked as 'Critical' OR 'High Priority.'

Since the standard **AVERAGEIFS** function is incapable of natively handling this disjunction across a single criteria range, the solution lies in constructing a logical test that processes the entire criteria range simultaneously. This test must return a single, combined TRUE/FALSE status for each row, indicating whether that row successfully meets at least one of the multiple OR condition criteria. This is achieved by converting the logical tests into numerical arrays and utilizing the mathematical addition operator (+) as a stand-in for the Boolean OR operation within the array environment.

The technique relies on Excel's internal handling of logical values during array evaluation, where

TRUE is coerced into 1 and FALSE into 0. By adding the results of two separate criteria checks--for example, (Range = Criterion 1) + (Range = Criterion 2)--if either test returns TRUE (1), the resulting sum will be 1 or greater, successfully signaling a match based on the OR condition. Conversely, if a row fails both tests, the sum will be 0 (0 + 0). This numerically derived logical array is the key to enabling the disjunctive filtering required for complex conditional averages.

Leveraging Array Formulas for OR Conditions

To successfully implement the OR condition logic within conditional averaging, we must utilize the power of the Array Formula structure, which involves nesting the **IF** function inside the **AVERAGE** function. The fundamental pattern used to achieve this is: **=AVERAGE(IF((Condition 1) + (Condition 2), Range_to_Average))**. This structure compels Excel to evaluate the logical conditions across entire ranges simultaneously, treating them as matrices rather than processing data cell by cell, which is essential for determining the overall match status for every row in the dataset.

The core mechanism is the addition operation within the logical test: **(A2:A11="Guard") + (A2:A11="Forward")**. This expression generates two separate arrays of TRUEs and FALSEs. When these arrays are added, Excel performs coercion, converting the logical values into numerical counterparts (1s and 0s). If a cell in range A2:A11 contains "Guard," the first array returns 1 for that position, and the second returns 0, summing to 1. If it contains "Forward," the sum is 0 + 1 = 1. Any result greater than zero (1 or higher) signifies that the row has met at least one of the conditions, thus fulfilling the requirement of the logical OR operation.

This resulting numerical array (comprising 1s and 0s) is then passed to the **IF** function. The **IF** function evaluates this array; if the value is 1 (or TRUE), the corresponding value from the target range (e.g., B2:B11) is returned. If the value is 0 (or FALSE), the **IF** function returns **FALSE** (because we omitted the *value_if_false* argument). The outer **AVERAGE** function then executes the final average calculation, which is programmed to automatically ignore logical FALSE values, ensuring that only the numerical values corresponding to the successful OR condition matches are included in the final mean.

If you instead would like to use the conditional averaging capability with OR condition logic to calculate the average value of cells that meet at least one of multiple conditions, you can use the following syntax, utilizing a powerful array formula structure:

=AVERAGE(IF((A2:A11="Guard")+(A2:A11="Forward"),B2:B11))

This formula finds the average of the values in the range **B2:B11** where the corresponding value in the range **A2:A11** is equal to Guard or Forward. This sophisticated approach effectively

overcomes the inherent AND logic barrier present in functions like AVERAGEIFS.

Practical Example: Calculating Averages Based on Multiple Criteria

To demonstrate the practical application of this array technique, we will analyze a simple dataset of basketball player statistics. This example highlights the necessity of the OR condition, as sports analysis often requires aggregating metrics for players who fall into related but distinct functional groups. Our objective is to calculate the mean scoring output solely for players designated as "Guard" or "Forward," deliberately excluding Centers and any other specialized positions to focus the analysis on perimeter players.

Imagine a dataset organized with player positions in Column A and their total points scored over a season in Column B. The requirement is to determine the average scoring metric specifically for those players occupying the primary perimeter roles--the Guards and the Forwards. This type of targeted aggregation allows analysts to assess the offensive efficiency of specific functional units within the team roster without being skewed by data from other positions.

The following visual representation shows the dataset we will be working with. Column A, labeled **Position**, contains the categorical criteria, and Column B, labeled **Points**, contains the numerical values to be averaged. We will use the combined **AVERAGE(IF())** structure to evaluate Column A against the two criteria and average the corresponding scores in Column B only when a match is found for either criterion.

Suppose we have the following dataset in Excel that shows the position and points scored by various basketball players:

	A	B	C	D	E	F
1	Position	Points				
2	Guard	22				
3	Guard	14				
4	Forward	17				
5	Center	30				
6	Forward	35				
7	Center	18				
8	Guard	11				
9	Forward	19				
10	Forward	24				
11	Center	16				
12						
13						
14						
15						
16						

Executing the AVERAGE(IF()) Array Formula

The execution of this formula requires precision in defining the ranges and ensuring the logical conditions are correctly grouped. Our goal is to calculate the average of the values in the **Points** column (Range B2:B11) specifically for all players who have a value of either "Guard" or "Forward" in the **Position** column (Range A2:A11). This segmentation of data will provide the precise average score for the designated positional group.

We begin by defining the two criteria checks and linking them using the addition operator, which implements the OR logic: **(A2:A11="Guard") + (A2:A11="Forward")**. This operation generates the numerical array of 1s and 0s, where 1 signifies a successful match (Guard OR Forward) and 0 signifies a non-match.

The **IF** function then filters the results. If the logical test returns 1, the **IF** function returns the corresponding point total from **B2:B11**; if the test returns 0, the **IF** function defaults to returning FALSE. The outer **AVERAGE** function then processes this resulting array, calculating the mean of the numbers while automatically excluding the FALSE values, thus ensuring only the scores of Guards and Forwards contribute to the final average.

We can use the following formula to execute this calculation efficiently and accurately:

=AVERAGE(IF((A2:A11="Guard")+(A2:A11="Forward"),B2:B11))

In versions of Excel that do not automatically handle array formula entry, it is mandatory to finalize the formula entry by pressing **Ctrl+Shift+Enter**. This action signals to Excel that the formula must be processed as an array. The following screenshot illustrates the result of applying this formula within the spreadsheet environment, confirming the effectiveness of the OR condition logic.

The following screenshot shows how to use this formula in practice, yielding the average points for the selected positions:

D2		=AVERAGE(IF((A2:A11="Guard")+(A2:A11="Forward"),B2:B11))				
	A	B	C	D	E	F
1	Position	Points		Avg. Points for Guards or Forwards		
2	Guard	22		20.28571429		
3	Guard	14				
4	Forward	17				
5	Center	30				
6	Forward	35				
7	Center	18				
8	Guard	11				
9	Forward	19				
10	Forward	24				
11	Center	16				
12						
13						
14						
15						

Verifying the Calculated Average

The calculated result, obtained through the sophisticated **AVERAGE(IF())** structure, provides the precise average for all players categorized as either Guard or Forward. As observed in the result cell, the average points scored by players fulfilling the criteria of Guard or Forward is calculated as **20.2857**. To ensure complete confidence in this advanced calculation method, it is highly recommended to perform a manual verification process, confirming that the formula correctly identified and averaged the target data points.

The manual verification process begins by isolating the records that satisfy the OR condition. Examining the dataset, the point totals corresponding to the positions of Guard and Forward are: 22, 14, 17, 35, 11, 19, and 24. We must confirm the count of included records, which totals 7 players. Next, we sum these scores: $22 + 14 + 17 + 35 + 11 + 19 + 24$, resulting in a total of 142 points. The final step of the average calculation is dividing the total sum by the count of records.

We can verify this is correct by manually calculating the average of all points values for players who are Guards or Forwards:

Average Points: $(22 + 14 + 17 + 35 + 11 + 19 + 24) / 7 = 142 / 7 = 20.2857$

Since the result of the manual calculation, **20.2857**, is an exact match to the value derived from the **AVERAGE(IF())** array formula, we have successfully validated the implementation of conditional averaging with OR logic. Mastery of this technique is essential for any serious Excel user, enabling the circumvention of native function limitations and providing the ability to perform highly customized and complex data analysis across expansive datasets.