

How to Average Multiple Rows with VLOOKUP in Excel

Authored by
stats writer

January 3, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Average Multiple Rows with VLOOKUP in Excel*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=111363>

While the primary purpose of the VLOOKUP function is to retrieve a single value corresponding to a specified lookup value, advanced users can leverage its capabilities in conjunction with other functions to perform powerful aggregation tasks, such as calculating an average across multiple data points or rows. Achieving this requires a clear understanding of VLOOKUP's behavior, particularly when the optional range_lookup argument is set to FALSE, which forces an exact match. However, VLOOKUP alone cannot handle conditional aggregation across multiple matching rows; this requires the integration of functions like AVERAGE and IF, often resulting in an array formula structure. This combination allows a single cell to return a comprehensive statistical measure, such as the mean value, derived from multiple records that satisfy the lookup criteria.

The Power of VLOOKUP and Conditional Averaging

In data analysis using Excel, calculating the average of a dataset is straightforward, but finding the average of specific data points conditionally looked up across multiple columns or rows presents a greater challenge. The methods presented below demonstrate how to leverage the retrieval capabilities of VLOOKUP--or a related, more flexible formula structure--to accomplish complex conditional averaging tasks within a single formula cell. We will explore two distinct approaches tailored for different aggregation needs.

The core difficulty when attempting to average multiple rows using VLOOKUP stems from its inherent design: it is engineered to retrieve only the value from the very first match it finds in the lookup column. To overcome this, advanced techniques are required, utilizing array constants or implementing genuine array formulas that process entire ranges simultaneously. Understanding these distinctions is paramount for successful data manipulation.

Method 1: Averaging Values in the First Matched Row

This approach uses VLOOKUP to target a single row--the first exact match found for the criteria--and then averages multiple columns within that specific row. This is highly useful when your data structure treats columns 2, 3, and 4 as different metrics belonging to the same single record (e.g., scores from Game 1, Game 2, and Game 3 for a player's first appearance in the list). It does not, however, average multiple occurrences of the lookup value across different rows.

The syntax below illustrates how to compel VLOOKUP to return an array of values from a single matched row, which the outer AVERAGE function then processes:

=AVERAGE(VLOOKUP(A14, \$A\$2:\$D\$11, {2,3,4}, FALSE))

This specific formula is structured to calculate the average of the numerical values residing in columns 2, 3, and 4 (relative to the lookup range A2:D11). Crucially, this aggregation only occurs

within the very first row where the corresponding value in the first column (column A) exactly matches the criteria specified in cell A14. The use of the curly braces {2, 3, 4} is essential, as it instructs VLOOKUP to retrieve an array of values rather than a single column result.

Decoding the VLOOKUP Array Constant Syntax

The array constant, represented by the structure {2, 3, 4} in the col_index_num argument of VLOOKUP, is the key to retrieving multiple columns simultaneously. When VLOOKUP encounters an array constant here, instead of returning a single value, it returns a horizontal array of values corresponding to the specified column indices from the matched row. This internal array is then passed directly to the outer function, which, in this case, is AVERAGE.

It is important to note that while this structure looks like an array formula, in modern versions of Excel (Excel 365, 2021), it often does not require the traditional Ctrl+Shift+Enter (CSE) confirmation because the array constant processing is handled internally by VLOOKUP and the resulting array is immediately consumed by the AVERAGE function. In older versions of Excel, depending on the exact implementation, explicit CSE confirmation might still be necessary for some variations of this technique.

The formula operates in a sequence: first, VLOOKUP locates the criteria; second, it pulls the data from columns 2, 3, and 4 of that single row into a temporary array (e.g., {10, 15, 15}); and third, the AVERAGE function calculates the arithmetic mean of the elements within that temporary array. This is a highly efficient way to compute row-wise statistics based on a single lookup criterion.

Method 2: Averaging Values Across All Matched Rows

If your goal is to calculate the average of scores belonging to a specific entity (like a player) who appears multiple times in the dataset (i.e., across multiple rows), the standard VLOOKUP approach is insufficient. This scenario requires a true conditional aggregation, typically achieved using a combination of IF and AVERAGE, implemented as an array formula. This technique effectively filters the entire data range based on the lookup criteria before performing the calculation.

The following formula structure calculates the average of values across all matched rows:

=AVERAGE(IF(A2:A11=\$A\$14,B2:D11))

This formula performs a detailed check across the entire lookup column range (A2:A11) to determine which rows match the specified lookup value in cell **A14**. Where a match is found, the corresponding range of values from columns B, C, and D (B2:D11) is included in an intermediate array. Non-matching rows result in a FALSE value. The outer AVERAGE function is designed to

ignore these FALSE values, successfully calculating the mean only of the numerical values passed to it from the matched rows.

It is crucial to understand that `A2:A11=A14` creates a Boolean array (an array of TRUEs and FALSEs). The IF function then uses this array to filter the data range B2:D11. If the condition is TRUE, the corresponding score is included; if FALSE, a logical FALSE is returned. By passing this mixed array to `AVERAGE`, we ensure that every single score associated with the target player across all games and all rows is factored into the final calculation, providing a true cumulative average.

Prerequisites: Entering Array Formulas in Excel

Method 2, the conditional averaging technique, relies on the function acting as an array formula, especially in older versions of Excel (pre-Excel 365). Array formulas must be confirmed using a special key combination known as **Control+Shift+Enter** (CSE) rather than just Enter. When correctly entered, Excel automatically surrounds the formula with curly braces (`{ }`) in the formula bar, indicating that it is being processed as an array. Users of modern Excel versions (using Dynamic Arrays) may find that this CSE confirmation is no longer strictly necessary, as the array processing is handled implicitly.

Steps for Entering a Traditional Array Formula (CSE):

Type the complete formula (e.g., `=AVERAGE( IF( A2:A11=$A$14, B2:D11 ) )`) into the destination cell. Instead of pressing Enter, hold down the **Control** key and the **Shift** key simultaneously, and then press **Enter**.

Verify that the formula bar now shows the curly brackets surrounding your formula: `{=AVERAGE( IF( A2:A11=$A$14, B2:D11 ) )}`.

Failure to use CSE confirmation (in older versions) will result in the formula only processing the first element of the range, leading to an incorrect result or an error value. This distinction is critical when applying Method 2 to corporate or legacy data systems.

Practical Dataset Overview

To illustrate the practical differences between these two powerful methods, we will apply them to a simple dataset. This example simulates basketball scores, where a single player may have multiple records representing different games or instances of scoring. The objective is to calculate the player's average points scored, first focusing only on the scores in their initial appearance, and then factoring in all their appearances.

The dataset structure includes four columns:

Column A: Player Name (The lookup column)

Column B: Game 1 Points

Column C: Game 2 Points

Column D: Game 3 Points

Note that the player "Chad" appears twice, which is the key reason we need two separate methods for aggregation. The calculation cells (lookup value and results) are typically placed outside the primary data table to avoid confusion.

	A	B	C	D	E	F	G
1	Player	Game 1	Game 2	Game 3			
2	Andy	10	8	31			
3	Bernard	12	17	9			
4	Chad	16	15	9			
5	Derrick	9	15	14			
6	Eric	14	14	15			
7	Frank	34	17	13			
8	George	29	18	12			
9	Harry	18	10	22			
10	Chad	14	20	28			
11	John	20	22	27			
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							

Let's now proceed to apply these formulas to the dataset, using cell **A14** to hold the target player's name (our lookup value).

Example 1: Calculating the First Match Average (VLOOKUP Array)

Our first objective is to determine the average points scored by the player "Chad" based only on the first row where "Chad" appears in the dataset (rows 2-11). This demonstrates Method 1, which utilizes the array constant within VLOOKUP to retrieve multiple column values at once.

We input the formula into cell **B14**, where we anticipate the average result to be displayed:

=AVERAGE(VLOOKUP(A14, \$A\$2:\$D\$11, {2,3,4}, FALSE))

Upon pressing **Enter** (or Ctrl+Shift+Enter, depending on the Excel version), the result is immediately calculated and displayed:

B14		=AVERAGE(VLOOKUP(A14, \$A\$2:\$D\$11, {2,3,4}, FALSE))						
	A	B	C	D	E	F	G	H
1	Player	Game 1	Game 2	Game 3				
2	Andy	10	8	31				
3	Bernard	12	17	9				
4	Chad	16	15	9				
5	Derrick	9	15	14				
6	Eric	14	14	15				
7	Frank	34	17	13				
8	George	29	18	12				
9	Harry	18	10	22				
10	Chad	14	20	28				
11	John	20	22	27				
12								
13	Player							
14	Chad	13.33333						
15								
16								
17								

The interpretation of this outcome is precise: the VLOOKUP component successfully located the first instance of "Chad" (in Row 2) and extracted the points from Game 1 (10), Game 2 (15), and Game 3 (15). The AVERAGE function then processed this internal array {10, 15, 15}. Consequently, the formula returns a value of **13.33** (rounded), which is the arithmetic mean of the points scored by Chad in his first recorded instance within the data range.

Example 2: Calculating the Average Across All Matches (Array Formula)

If we want to capture the true overall performance of "Chad," we must incorporate the scores from his second appearance in the data set as well. This requires utilizing Method 2, the conditional AVERAGE(IF) array formula. This approach filters all rows that meet the criteria regardless of their position in the dataset.

The formula is entered into a designated cell (e.g., **C14**):

=AVERAGE(IF(A2:A11=\$A\$14,B2:D11))

After confirming the formula using the appropriate key stroke (Enter for modern Excel or Ctrl+Shift+Enter for older versions), the aggregate results are displayed:

	A	B	C	D	E	F	G
1	Player	Game 1	Game 2	Game 3			
2	Andy	10	8	31			
3	Bernard	12	17	9			
4	Chad	16	15	9			
5	Derrick	9	15	14			
6	Eric	14	14	15			
7	Frank	34	17	13			
8	George	29	18	12			
9	Harry	18	10	22			
10	Chad	14	20	28			
11	John	20	22	27			
12							
13	Player						
14	Chad	17					
15							
16							

The logic of this formula searches the entire Player column (A2:A11), identifying all instances of "Chad." The resulting array passed to AVERAGE includes the 6 scores associated with Chad (10, 15, 15 from Row 2, and 20, 19, 23 from Row 8), while ignoring all other scores using the FALSE logical values. This means the formula looks up "Chad" in the Player column and then returns the average of the points values for each game in each row that matches "Chad."

We can clearly observe that Chad scored an overall average of **17** points per game across the two rows (six scores total) he appeared in. This sophisticated array formula structure is the definitive method for achieving conditional aggregation across multiple matching rows in Excel, providing a significant step up from the limitations of the standard VLOOKUP function.