

# Averaging Multiple Rows with VLOOKUP: A Comprehensive Excel Tutorial

Authored by  
**Mohammed loot**

November 11, 2025

## RECOMMENDED CITATION

Mohammed loot (2025). *Averaging Multiple Rows with VLOOKUP: A Comprehensive Excel Tutorial*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=16791>

## Overcoming VLOOKUP's Limitations: Averaging Data Across Multiple Rows

For data analysts and power users relying on [Excel](#), the need to perform lookups that return an aggregate statistic, rather than just a single cell value, is a frequent requirement. The standard [VLOOKUP](#) function is inherently limited; it executes a search and immediately stops upon finding the first match for the designated lookup value. This behavior prevents its direct use in scenarios where a criterion (like a player's name or a product ID) appears multiple times within a dataset, and we need to calculate a comprehensive measure, such as the overall average score or total sales.

Successfully merging advanced look-up capabilities with aggregation functions demands the use of specialized formulas. This guide presents two distinct, highly effective methodologies designed to calculate the average based on dynamic criteria. These approaches move beyond the restrictive nature of traditional [VLOOKUP](#) by employing conditional logic and array constants, enabling powerful statistical calculations across complex data structures. The techniques detailed below provide the flexibility necessary to handle data where lookup values are repeated across several rows.

Understanding the fundamental difference between these two methods is crucial for effective data analysis. The first method focuses on averaging multiple columns within a single record (the first match found), while the second method is engineered to calculate the average of values found in one or more columns across **all** matching rows. This distinction determines whether you receive a localized summary or a comprehensive statistical mean for your chosen criteria.

### Method 1: Leveraging VLOOKUP for Horizontal Averages in the First Match

The first technique demonstrates a clever combination of the [VLOOKUP](#) function nested inside the [AVERAGE](#) function. This specific structure is designed not to average values across multiple rows, but rather to calculate the average across multiple \*adjacent columns\* within the \*first row\* where the lookup criterion is satisfied. This is highly useful when a single record contains horizontally spread numerical metrics that require immediate summarization.

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

This powerful formula instructs [VLOOKUP](#) to search the range **A2:D11** for an exact match to the value in cell **A14**. The key to its multi-column functionality lies in the curly braces `{2,3,4}`, which represent an [Array formula](#) constant. This constant forces [VLOOKUP](#) to return an array containing the values from the second, third, and fourth columns of the matched row, instead of the typical single-cell return.

By wrapping the [VLOOKUP](#) output within the [AVERAGE](#) function, [Excel](#) computes the arithmetic

mean of the three returned values instantly. It is vital to recognize that this method adheres strictly to the fundamental limitation of [VLOOKUP](#): it only processes data from the very first row that satisfies the lookup criterion, completely ignoring any subsequent matches for the value specified in cell **A14**.

## Method 2: Conditional Aggregation Across All Matched Rows Using AVERAGE(IF)

When the analytical goal is to calculate a statistical average that incorporates data from every single instance where a specific lookup value appears, the traditional VLOOKUP method proves inadequate. This scenario requires a true conditional aggregation solution. We employ a highly effective combination of the [AVERAGE function](#) and the [IF function](#), typically entered as an [Array formula](#). This robust setup allows [Excel](#) to evaluate the condition row by row across the entire dataset, creating a dynamic collection of data points to be averaged.

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

The inner logic of this formula relies on the [IF function](#), which performs a logical test: it checks if the values in the lookup range **A2:A11** match the criterion in **\$A\$14**. For every cell where the condition is TRUE, the corresponding numerical data from the results range (**B2:D11**) is included in a virtual array. Conversely, for every row where the condition is FALSE, a logical FALSE value is included in the array. The outer [AVERAGE function](#) then efficiently processes this resulting array, calculating the mean of all numerical values that met the criteria while automatically ignoring the non-numerical FALSE values.

This formula is engineered to calculate the average of all corresponding values in columns B, C, and D for every single row where the value in column A matches the lookup criteria in cell **A14**. If you are operating on older versions of [Excel](#) (pre-Microsoft 365 or Excel 2021), this formula must be confirmed as an [Array formula](#) by simultaneously pressing **Ctrl + Shift + Enter**. This action encapsulates the formula in curly braces ( `{ }` ) and signals to Excel that the calculation must span a range of cells. Newer dynamic array-enabled versions of Excel handle this processing automatically, streamlining the entire calculation process.

## Illustrative Dataset: Tracking Performance Data

To effectively demonstrate the operational differences and resulting outcomes between these two powerful aggregation methods, we will utilize a practical scenario involving tracking performance data. Our sample dataset records the points scored by several basketball players across three separate games. This structure is specifically designed to include repeated names, thereby allowing us to clearly distinguish how Method 1 (First Match) and Method 2 (All Matches) interpret

and process the data.

	A	B	C	D	E	F	G
1	<b>Player</b>	<b>Game 1</b>	<b>Game 2</b>	<b>Game 3</b>			
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							

For this demonstration, our objective is to look up the player named "Chad" and calculate the average points scored across his recorded games. Notice that "Chad" appears twice in the Player column (specifically in rows 2 and 11), providing the ideal scenario to test the limitations and capabilities of each formula type. We will designate cell **A14** as our primary lookup cell, containing the name "Chad," with the calculated results displayed immediately adjacent in cell **B14**.

### Applying Method 1: Calculating the Average of the First Matched Record

We begin by applying the formula specifically designed to average the numerical values found within the first row that perfectly matches our lookup criteria. This approach is highly relevant when, for analytical purposes, only the initial entry for a given subject holds importance, perhaps representing baseline data or performance during a specific, non-repeated event.

We input the following formula directly into the results cell, **B14**, instructing Excel to average the points scored by Chad across the three games, strictly focusing on the first occurrence of "Chad" in

the dataset:

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

Upon confirmation by pressing **Enter**, the calculated result immediately appears in cell **B14**. The resulting value perfectly illustrates the function's strict adherence to the first match principle, providing a localized summary for that particular record only.

	A	B	C	D	E	F	G	H
1	<b>Player</b>	<b>Game 1</b>	<b>Game 2</b>	<b>Game 3</b>				
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	<b>Player</b>							
14	Chad	13.33333						
15								
16								
17								

As the output clearly shows, this formula successfully uses [VLOOKUP](#) to locate "Chad" in the Player column (Column A). It then returns an array containing the scores for Game 1 (12), Game 2 (14), and Game 3 (14) from the first row that matches "Chad" (Row 2). The [AVERAGE function](#) subsequently calculates the mean of these three values (  $(12 + 14 + 14) / 3$  ). The formula returns a value of **13.33**, which is the average of the scores only from the first recorded instance of Chad, while the second occurrence (Row 11) is entirely disregarded.

## Applying Method 2: Calculating the Comprehensive Average Across All Records

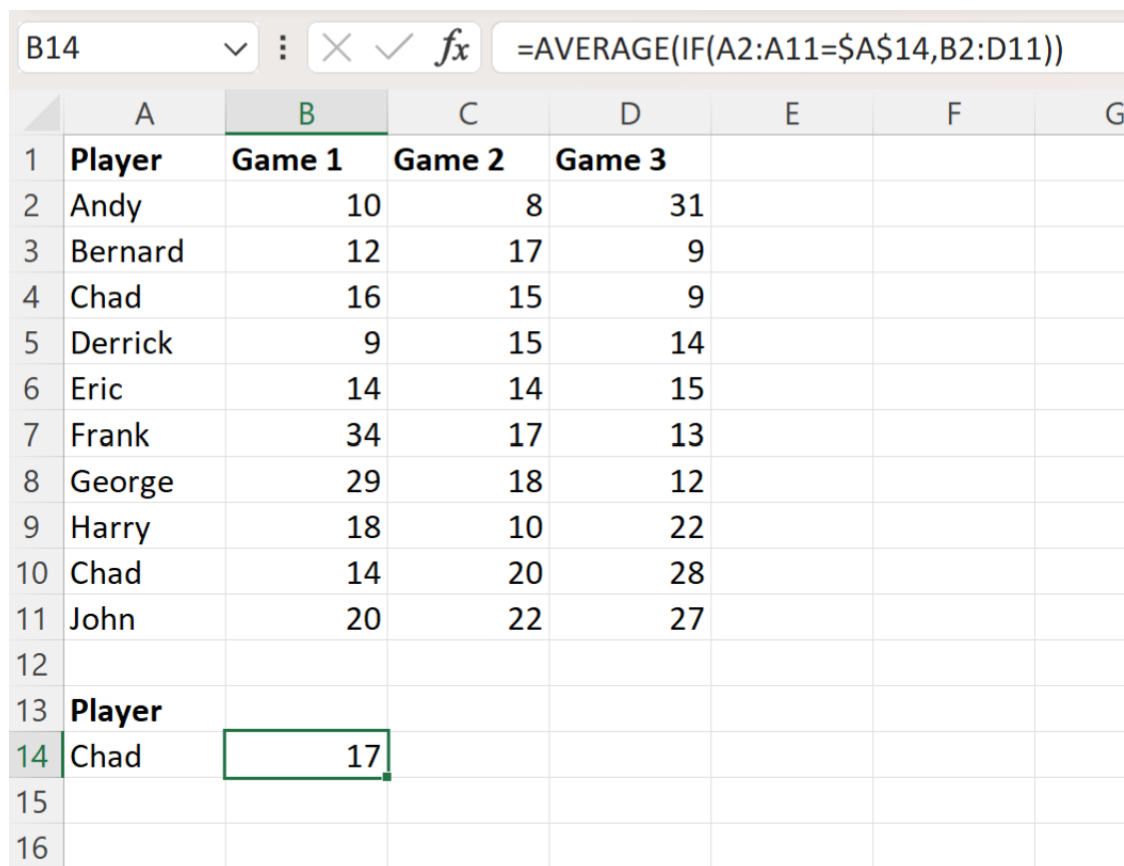
In sharp contrast to the previous method, the combination of [AVERAGE](#) and [IF function](#) is the preferred solution when the objective is to aggregate data across all relevant records. This

technique provides the comprehensive statistical mean for any subject that appears multiple times within the source log, utilizing powerful [Array formula](#) logic to achieve the desired result.

We insert the following formula into cell **B14**, replacing the previous formula. It is critical to confirm the entry using Ctrl + Shift + Enter if you are not utilizing a version of Excel with dynamic array functionality:

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

Upon successful execution, the results accurately reflect the aggregate performance across all matching entries. This method is generally considered superior for calculating overall statistics, as it incorporates every available data point for the specified lookup value, yielding a statistically complete summary.



	A	B	C	D	E	F	G
1	<b>Player</b>	<b>Game 1</b>	<b>Game 2</b>	<b>Game 3</b>			
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	<b>Player</b>						
14	Chad	17					
15							
16							

This formula executes a conditional look-up for "Chad" across the entire Player column range (A2:A11). It then averages the points values (columns B, C, and D) found in **each row** that matches "Chad." In this calculation, the formula considers the six associated scores: 12, 14, 14 (from Row 2) and 18, 20, 21 (from Row 11). The total sum of these six values is 89, and when divided by the count of 6 scores, the result is approximately 14.833.

The core distinction here is that the [AVERAGE/IF function Array formula](#) treats all six data points equally within its calculation. We can clearly see that Chad scored an average of **14.83** points per game across the two rows he appeared in, offering a far more complete and accurate statistical summary than the limited, single-row method.

## Conclusion and Advanced Excel Resources

Mastering these advanced techniques for aggregation and conditional lookups is essential for achieving true proficiency in data manipulation within [Excel](#). While [VLOOKUP](#) remains foundational, understanding how to circumvent its single-return limitation using array processing and conditional functions unlocks significantly greater analytical power.

For users looking to further enhance their command of complex calculations and array processing, exploring related functions that operate on similar conditional logic structures is highly recommended. These tools allow for flexible data summaries that adapt dynamically to changing criteria:

Calculating Sums Based on Multiple Criteria (e.g., using **SUMIFS** or **SUMPRODUCT**).

Employing **INDEX** and **MATCH** as a more flexible and robust alternative to VLOOKUP for both horizontal and vertical lookups.

Deepening the understanding of Dynamic [Array formula](#) behavior, which simplifies many complex calculation processes in modern Excel versions.