Learning to Calculate Conditional Averages with AVERAGEIF Using Multiple Ranges in Excel

Authored by

Mohammed looti

October 28, 2025

The Limitation of Conditional Averaging in Excel

In the expansive and crucial domain of Microsoft Excel, the capability to execute sophisticated conditional calculations forms the backbone of advanced data analysis and reporting. The dedicated function, **AVERAGEIF()**, serves as an essential utility for calculating the conditional average of a set of numerical values based on a single, user-defined criterion. For example, a data analyst can effortlessly utilize this function to compute the average expenditure for a specific project category or the mean performance score for employees residing in a particular geographical region, provided all relevant data resides within a single, unified block.

However, the inherent architecture of **AVERAGEIF()** introduces a significant structural constraint. This function is fundamentally designed to operate on a single, contiguous block of data, meaning both the range containing the criteria and the range containing the values to be averaged must be part of one connected data set. When dealing with complex business or academic datasets, information is frequently organized into multiple, distinct tables or spread across non-contiguous ranges within the same worksheet. This fragmentation immediately renders the standard **AVERAGEIF()** function inadequate, as its syntax does not permit the inclusion of multiple, independent value ranges.

This limitation poses a genuine hurdle when analysts need to generate a consolidated average from disparate data sources without physically restructuring the worksheet--a task often impractical or undesirable due to data integrity concerns or reporting requirements. Imagine needing the average sales figures from Quarter 1 and Quarter 3, where those figures exist in completely separate columns far removed from each other. A direct application of the conditional average function simply cannot bridge this gap. Consequently, overcoming this constraint requires transitioning from simple, built-in conditional functions to a more sophisticated, composite formula structure that can intelligently aggregate data across these separated blocks.

Why Standard Functions Fail with Non-Contiguous Data

To fully appreciate the necessity of an advanced technique, we must first solidify our understanding of the native conditional averaging functions. The syntax for the **AVERAGEIF function** is structured as: `AVERAGEIF(range, criteria, )`. The `range` parameter specifies where the condition is checked, and the `average_range` specifies the values to be included in the calculation. This rigid, three-part structure mandates that all components--criteria range and average range--must belong to a single, continuous data matrix. Crucially, Excel does not allow an array of ranges (e.g., `(A1:A10, C1:C10)`) to be entered into the single `range` argument.

A common misstep for users facing this multi-range challenge is attempting to utilize the highly versatile **AVERAGEIFS function**, which is specifically designed to handle scenarios involving

multiple simultaneous conditions. While **AVERAGEIFS()** is a powerful tool for complex filtering, it does not address the issue of non-contiguous data blocks. Its primary requirement is that the average range and all subsequent criteria ranges must be of identical size and orientation, and must align perfectly row-by-row. It is intended for filtering a single large table based on several columns, not for consolidating results from two entirely separate tables.

Consider a practical example where data set 1 uses columns A and B, and data set 2 uses columns D and E. We want the average of values in B and E, where the corresponding entries in A and D match a criterion. Since **AVERAGEIFS()** demands a unified structure, attempting to input range B and range E into a single function call will result in an error or, at best, an incorrect calculation based on mismatched cell counts. Therefore, when data is genuinely fragmented--existing in separate, independent blocks on the spreadsheet--neither **AVERAGEIF()** nor **AVERAGEIFS()** can provide a native, straightforward solution. This gap necessitates the adoption of a clever workaround that bypasses the single-range limitation by treating the fundamental components of averaging independently.

The Advanced Solution: Leveraging the SUMIF and COUNTIF Pair

The core mathematical definition of the average is straightforward: the total sum of all relevant values divided by the total count of those values. Since standard conditional averaging functions cannot handle multiple ranges simultaneously, we must build a custom function that performs the conditional summation and conditional counting operations separately across each data block, and then aggregate the results. This robust strategy relies on combining the strengths of the **SUMIF function** and the **COUNTIF function**, unified by the versatile **SUM function**.

The role of **SUMIF()** is to conditionally accumulate the values. In our scenario, we will use one **SUMIF()** instance for each data block that requires averaging. Each instance will check its respective criteria range and sum up the corresponding values from its respective value range. This ensures that we generate multiple sub-totals, each representing the conditional sum for one data segment. These sub-totals are then pooled together using the outer **SUM()** function to derive the grand total sum of all values that meet the specified condition across all non-contiguous ranges.

Similarly, the **COUNTIF()** function is employed to accurately tally the number of items that meet the condition in each data block. Just as with summation, we execute one **COUNTIF()** instance per criteria range. This step is critical because simply counting the rows in the data ranges would be incorrect; we only want to count the rows where the specified condition (e.g., "Product X") is actually present. By performing a conditional count, we ensure that the denominator in our average calculation precisely reflects the number of items that contributed to the numerator. The results of these individual conditional counts are then combined using a second **SUM()** function, yielding the

accurate total count of occurrences across the entire fragmented dataset.

Constructing the Multi-Range Conditional Average Formula

The composite formula is an elegant expression of the fundamental average calculation, adapted for fragmented data. It is structured entirely around the principle of **(Total Conditional Sum) / (Total Conditional Count)**. This technique requires meticulous attention to parentheses to ensure proper order of operations, as we are aggregating intermediate results before performing the final division.

The structure for combining two distinct data blocks (Data Block 1: Criteria in A, Values in B; Data Block 2: Criteria in D, Values in E) is as follows, assuming the criterion itself is stored in cell **G2**:

=(SUM(SUMIF(A2:A11,G2,B2:B11),SUMIF(D2:D11,G2,E2:E11))/SUM(COUNTIF(A2:A11,G2),COUNTIF(D2:D11,G2)))

This single line of code effectively consolidates the analysis of two separate data segments. The numerator aggregates the conditional sums (one **SUMIF** for A/B, one **SUMIF** for D/E), resulting in the total value of all items matching the criterion in **G2**. The denominator similarly aggregates the conditional counts (one **COUNTIF** for A, one **COUNTIF** for D), yielding the total number of items that contributed to the sum. The final division then calculates the true, consolidated conditional average across the entire virtual dataset.

This approach demonstrates the flexibility of Excel when its built-in functions fall short. By understanding the core mathematical requirements of the average and matching them with the conditional aggregation capabilities of **SUMIF()** and **COUNTIF()**, we construct a powerful, resilient formula that eliminates the need for data consolidation or complex array methods for simple conditional averaging across fragmented tables. The key advantage is that the ranges **B2:B11** and **E2:E11** do not need to be adjacent or aligned in any specific way, only that their corresponding criteria ranges (A2:A11 and D2:D11, respectively) must match their size and orientation.

Practical Application: Step-by-Step Sales Data Analysis

To move from theory to practical implementation, let us apply this technique to a real-world scenario involving fragmented sales tracking. Assume we manage sales data for various fruit products. Due to legacy systems or differing collection periods, the daily sales figures are segregated into two distinct tables on the same worksheet. Our objective is to calculate the **average daily sales for Mangos** across the entirety of our recorded history, irrespective of which table the entries reside in.

Our data set is structured as depicted below, clearly illustrating the two separate blocks of

information. The first block occupies columns A and B, detailing the product type and its sales volume, while the second block occupies columns D and E, maintaining the same structure but covering a different set of transactions or period. The criterion we are searching for, "Mangos," is conveniently placed in cell **G2**.

	A	B	C	D	E	F
1	**Product**	**Sales**		**Product**	**Sales**	
2	Apple	10		Mango	4	
3	Banana	4		Kiwi	12	
4	Mango	8		Kiwi	9	
5	Kiwi	9		Apple	9	
6	Apple	10		Banana	4	
7	Kiwi	14		Mango	8	
8	Kiwi	13		Banana	5	
9	Mango	6		Apple	10	
10	Banana	5		Mango	8	
11	Mango	5		Kiwi	12	
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						

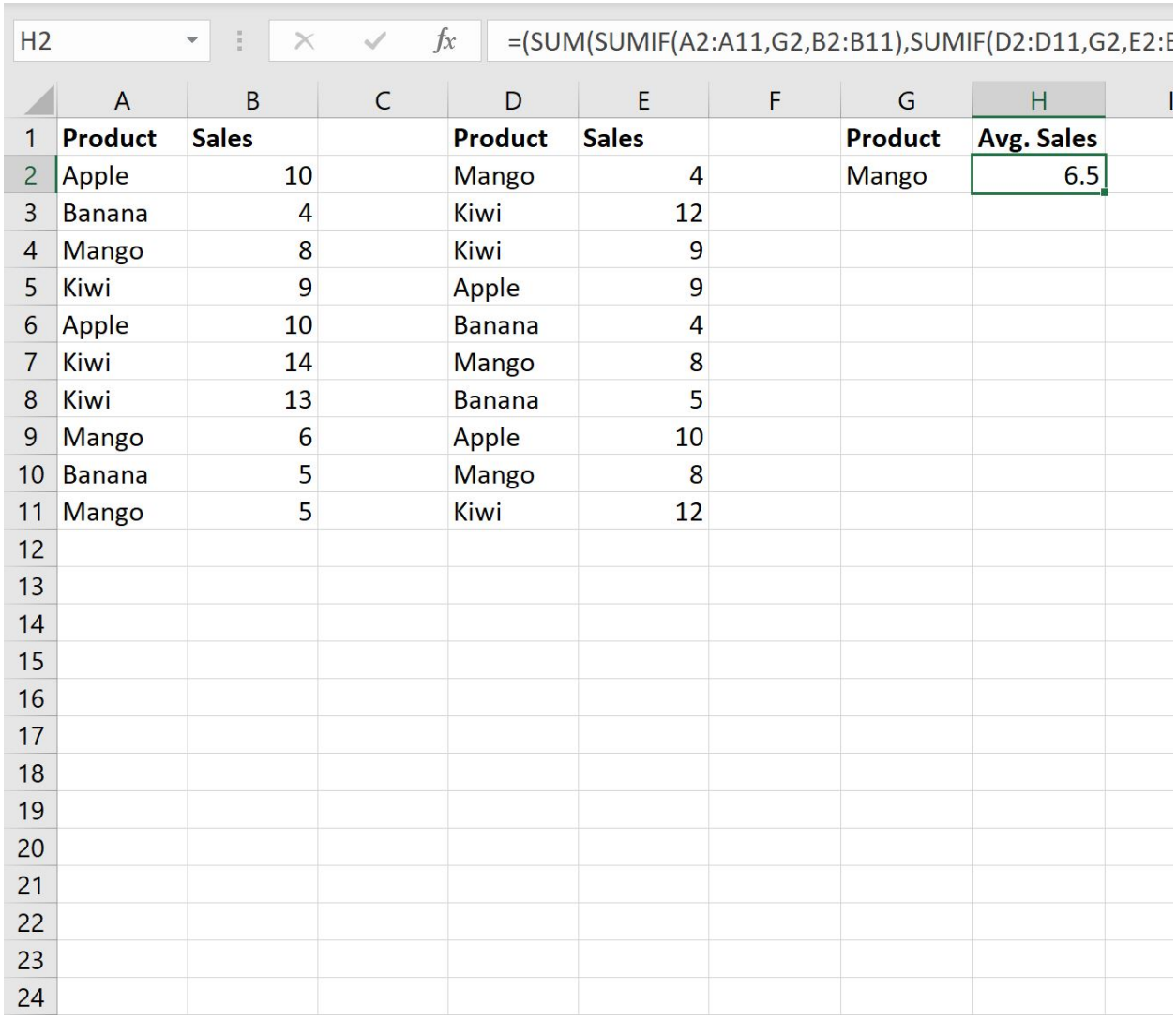
The implementation involves carefully entering the constructed composite formula into a designated output cell, such as **H2**. This action tells Excel to initiate a two-pronged process: first, calculating the sum of all Mango sales from both data blocks; and second, calculating the total count of Mango entries from both data blocks. This simultaneous conditional aggregation ensures the integrity of the averaging process.

The exact formula to be entered into cell **H2** is:

=(SUM(SUMIF(A2:A11,G2,B2:B11),SUMIF(D2:D11,G2,E2:E11))/SUM(COUNTIF(A2:A11,G2),COUNTIF(D2:D11,G2)))

Upon execution (by pressing **Enter**), Excel processes the formula, meticulously evaluating the criteria in A2:A11 and D2:D11 against G2. It then sums the corresponding values in B2:B11 and

E2:E11, divides this grand sum by the total count derived from A2:A11 and D2:D11, and presents the final, unified average. As demonstrated in the resulting spreadsheet view, the formula successfully yields the consolidated average sales figure.



	A	B	C	D	E	F	G	H
1	**Product**	**Sales**		**Product**	**Sales**		**Product**	**Avg. Sales**
2	Apple	10		Mango	4		Mango	6.5
3	Banana	4		Kiwi	12			
4	Mango	8		Kiwi	9			
5	Kiwi	9		Apple	9			
6	Apple	10		Banana	4			
7	Kiwi	14		Mango	8			
8	Kiwi	13		Banana	5			
9	Mango	6		Apple	10			
10	Banana	5		Mango	8			
11	Mango	5		Kiwi	12			
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								

The resulting value, **6.5**, is the accurate, collective average daily sales for **Mangos** across all designated data segments. This example decisively proves that complex conditional averages across non-contiguous data blocks can be managed efficiently without altering the source data structure, using only standard Excel functions combined in this specific, powerful arrangement.

Detailed Formula Dissection and Logic Flow

Achieving mastery over this technique requires a clear understanding of how the components interact within the overarching structure **(Numerator / Denominator)**. The robust logic ensures that only data points that satisfy the specified criterion contribute to both the summation and the counting operations, thereby guaranteeing an accurate conditional average.

The **Numerator: Calculating the Total Conditional Sum**. This section is responsible for summing the sales figures (values) only where the product name (criterion) matches **G2**. The structure is **SUM(SUMIF_Block_1, SUMIF_Block_2)**.

The expression **SUMIF(A2:A11,G2,B2:B11)** processes the first data block. It checks the criteria range **A2:A11** against the value in **G2** ("Mangos"). When a match occurs, the corresponding number from the average range **B2:B11** is included in a subtotal.
The subsequent expression, **SUMIF(D2:D11,G2,E2:E11)**, operates identically on the second data block, aggregating the conditional sum from **E2:E11** based on matches in **D2:D11**.
The outer **SUM()** function is essential; it takes the two independent subtotals generated by the **SUMIF()** calls and combines them into a single, comprehensive total conditional sum. This successfully aggregates the relevant data across the two separate ranges.

The **Denominator: Calculating the Total Conditional Count**. This section ensures we divide by the exact number of matching entries. The structure is **SUM(COUNTIF_Block_1, COUNTIF_Block_2)**.

The function **COUNTIF(A2:A11,G2)** accurately tallies the number of "Mango" occurrences within the first criteria range, **A2:A11**. This provides the count for the first data block.
Similarly, **COUNTIF(D2:D11,G2)** calculates the count of matching entries in the second criteria range, **D2:D11**.
The final **SUM()** wrapper adds these two counts together, producing the precise total count of entries that fulfilled the condition. This ensures that the denominator is not simply the total number of rows, but the actual number of data points included in the numerator's sum.

By dividing the combined conditional sum by the combined conditional count, this composite formula achieves what native functions cannot: a precise, aggregated conditional average over multiple, non-aligned data sets within Microsoft Excel. This methodology is a cornerstone of advanced spreadsheet modeling, providing flexibility when dealing with real-world data fragmentation.

Verifying the Results

To establish absolute confidence in our advanced formula, a manual verification of the results is highly recommended. This process confirms that every relevant data point was correctly included in the calculation. Based on the sales data example provided, we manually extract all sales figures corresponding to "Mangos" from both the primary (A/B) and secondary (D/E) data blocks.

From the first data block (Columns A and B), the sales values identified for Mangos are **8, 6, and 5**.
From the second data block (Columns D and E), the sales values identified for Mangos are **4, 8,**

and 8.

The complete collection of sales figures for Mangos across the entire sheet is therefore: 8, 6, 5, 4, 8, 8. We can now manually compute the average using the standard mathematical procedure:

Average Mango Sales Calculation:

Total Sum = 8 + 6 + 5 + 4 + 8 + 8 = **39**
Total Count = **6**
Average = 39 / 6 = **6.5**

This rigorous, step-by-step verification confirms that the result obtained by our composite **SUM(SUMIF, COUNTIF)** formula--which was **6.5**--is mathematically precise and accurate. The verification process ensures that the advanced formula is reliable for aggregating data from multiple, distinct source areas, providing a trustworthy solution for complex conditional queries.

Conclusion: Mastering Complex Conditional Aggregations

While the native **AVERAGEIF()** function excels in simple, single-range conditional calculations, the realities of enterprise data management often necessitate querying information spread across multiple, non-contiguous ranges. In these complex scenarios, relying solely on built-in functions proves insufficient.

The powerful technique of combining **SUMIF()** and **COUNTIF()** functions within a **SUM()** framework offers an elegant and indispensable workaround. This approach not only overcomes the structural limitations of standard conditional functions but also provides the flexibility needed to perform accurate data aggregation regardless of how fragmented the source data may be. Mastering this method is a hallmark of advanced spreadsheet proficiency, significantly extending the analytical reach of Microsoft Excel.

We encourage data professionals to practice implementing this technique, as the underlying principles--conditional summation and conditional counting--can be adapted for various other types of conditional aggregation beyond just calculating the average. Understanding how these functions interact is crucial for building robust models that handle real-world data complexities effectively.

Additional Resources

The following tutorials explain how to perform other common tasks in Excel:

How to Use VLOOKUP
Conditional Formatting Tips
Working with Pivot Tables