Using Wildcards with SUMIFS: A Guide to Partial Matching in Excel

Authored by

Mohammed looti

October 30, 2025

The **Excel** **SUMIFS** function stands as an indispensable feature for advanced data aggregation, enabling users to sum values contingent upon multiple specified criteria. While incredibly robust for exact matches, its full potential is truly unlocked when integrated with **wildcard characters**. This integration allows for flexible, powerful pattern matching, transforming the function from a simple conditional tool into a dynamic data analysis powerhouse. This comprehensive guide is dedicated to illustrating how to expertly deploy wildcards within your **SUMIFS** formulas, addressing common challenges related to partial-match criteria, such as finding values containing a specific string or those that adhere to particular starting or ending text patterns.

By incorporating **wildcard characters**, analysts can bypass the limitations of rigid, exact comparisons, constructing formulas that intelligently adapt to imperfect, inconsistent, or varied source data. This functionality is critical in numerous professional applications. Consider the task of categorizing sales data where product names vary slightly but share a common core description, or analyzing extensive inventory lists where items are categorized by partial identifiers. Mastering this technique ensures your formulas remain agile and accurate, regardless of minor variations in text strings within your datasets. This capability is invaluable for tasks such as categorizing sales data by partial product names, summing scores for teams with similar naming conventions, or analyzing inventory based on partial descriptions.

We will proceed by first defining the specific wildcard characters recognized by **Excel**, detailing the mandatory syntax for their usage within the **SUMIFS** framework. Following the foundational theory, we will engage in practical, step-by-step examples utilizing a consistent sample dataset. These demonstrations are designed to provide clear, actionable insights, enabling you to immediately apply these essential partial-match techniques to your own complex data analysis requirements.

Understanding Wildcard Characters and SUMIFS Syntax

In the context of **Excel**, **wildcard characters** function as special proxies, representing one or more unknown characters within a text string. This pattern-matching capability is fundamental to numerous lookup and conditional aggregation functions, including **SUMIFS**, **COUNTIFS**, and even older functions like **VLOOKUP** (though VLOOKUP's usage is more limited). Recognizing the roles of the two primary wildcards is the first step toward effective implementation:

Asterisk (*****): This is the most frequently used wildcard, designated to represent any arbitrary sequence of characters, encompassing zero, one, or multiple characters. For instance, constructing the criterion "app*****" will successfully match "apple", "application", and "approve", while "*****ply" would match "apply" and "comply". The asterisk is crucial for matching text strings that contain a specified sequence anywhere within the cell.
Question Mark (**?**): In contrast to the asterisk, this character represents exactly one single character. This is invaluable when dealing with minor, known variations in spelling or length. For

example, "b**?**t" would match "bat", "bet", and "bit", but it would strictly fail to match "boat" or "boot". This provides a highly specific control over the length of the matched string.

For the practical application of partial summation in conditional aggregation, we typically rely heavily on the **asterisk** due to its flexibility in handling variable-length text strings. The general syntax structure for the **SUMIFS** function mandates the following structure: `SUMIFS(sum_range, criteria_range1, criteria1, , ...)`. A critical rule when embedding wildcards is that the entire `criteria` argument--including the wildcard character itself--must be enclosed within double quotation marks. This rule applies even if the wildcard pattern is constructed by referencing a cell value using concatenation.

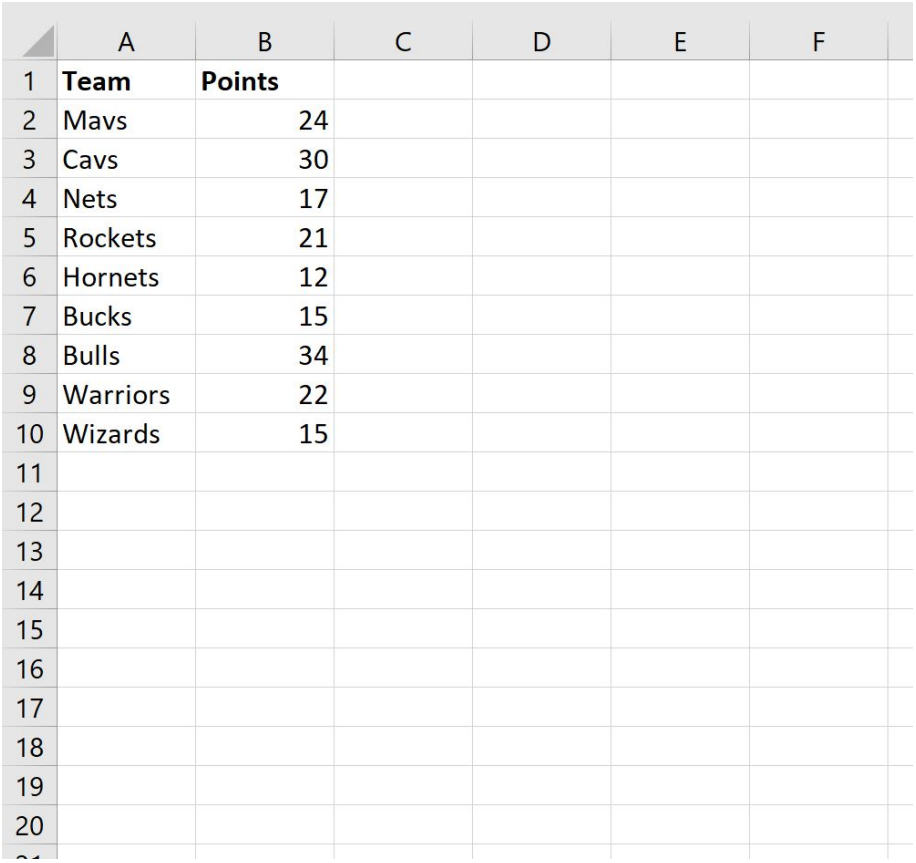
The ability to combine fixed text with the fluid nature of **wildcard characters** allows for the creation of exceptionally dynamic conditional formulas. Below are the foundational patterns used in Excel to integrate wildcards with **SUMIFS**, addressing the three most common pattern-matching requirements:

 # Sum values where cells in A2:A10 contains 'string' anywhere
=SUMIFS(B2:B10, A2:A10, "*string*")

Sum values where cells in A2:A10 start with 'string'
=SUMIFS(B2:B10, A2:A10, "string*")

Sum values where cells in A2:A10 end with 'string'
=SUMIFS(B2:B10, A2:A10, "*string")

Each pattern serves a distinct analytical purpose. The first pattern, using leading and trailing asterisks (`"*string*"`), is essential for comprehensive searches where the text may be buried within longer descriptions. The second pattern (`"string*"`) ensures the aggregation only occurs if the criterion initiates the cell content, making it perfect for prefix-based grouping. Conversely, the final pattern (`"*string"`) is tailored for identifying suffixes, summing values tied to items that conclude with the specified text sequence. To solidify these concepts, we will now examine a shared dataset used throughout the following examples.



	A	B	C	D	E	F
1	**Team**	**Points**				
2	Mavs	24				
3	Cavs	30				
4	Nets	17				
5	Rockets	21				
6	Hornets	12				
7	Bucks	15				
8	Bulls	34				
9	Warriors	22				
10	Wizards	15				
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						

Example 1: Summing Values Based on Contained Text (Anywhere Match)

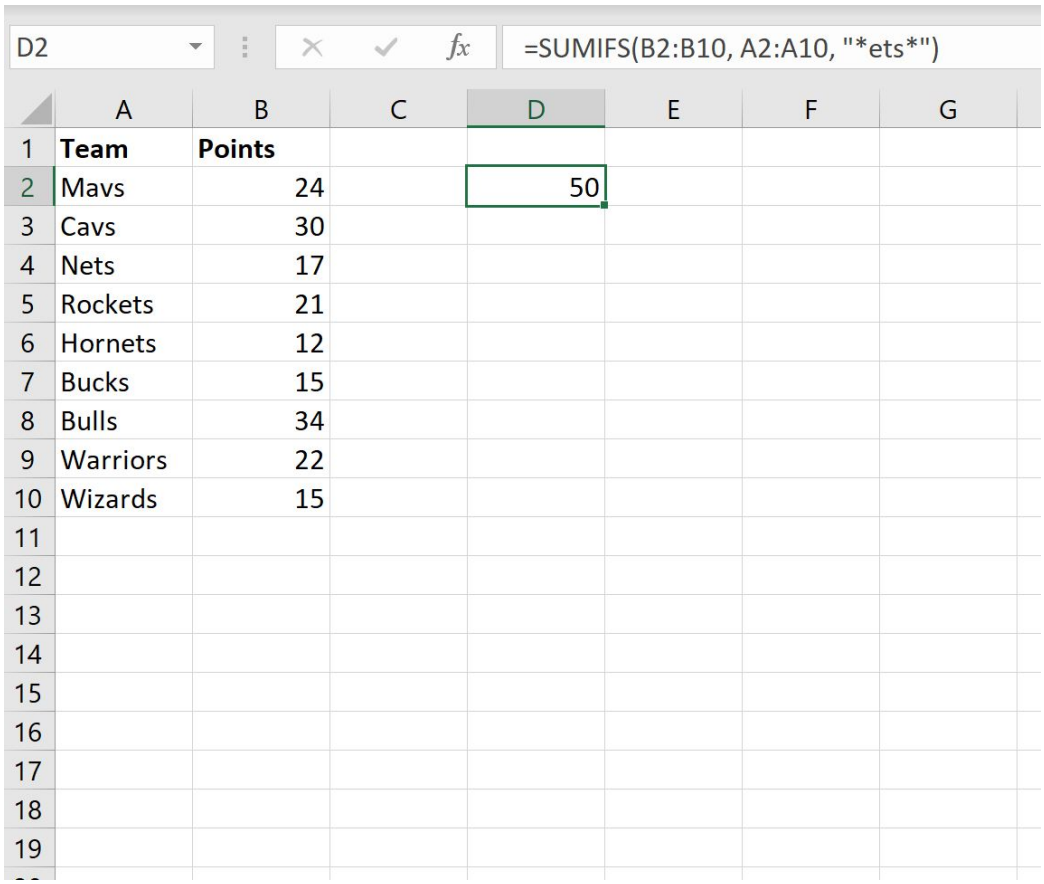
A frequent requirement in data analysis involves summing numerical values where the corresponding text description contains a particular keyword, irrespective of its position within the string. This scenario requires the use of the **asterisk** wildcard on both sides of the target text. This is a common requirement in data analysis, where exact matches are often not sufficient, forcing the analyst to search for substrings.

Using the provided sample dataset, which lists various teams and their accumulated points, our objective is to calculate the total points for all teams whose names include the substring "ets". This task demonstrates the power of a flexible partial match, enabling us to capture "Nets," "Rockets," and "Hornets" seamlessly. To execute this, we must configure the **SUMIFS** criteria to search for "ets" surrounded by any characters (or none) both before and after the term. The formula structure designed to achieve this universal containment match is as follows:

=SUMIFS(B2:B10, A2:A10, "*ets*")

In this specific implementation, `B2:B10` is designated as the `sum_range`, specifying the column containing the numerical values (Points) that we intend to aggregate. The `A2:A10` range serves as

the `criteria_range1`, which holds the text strings (Team names) against which the criterion will be evaluated. The core of this formula lies in the `criteria` argument, `"*ets*"`. This specific pattern directs **Excel** to include a row in the summation only if the corresponding cell in column A contains the sequence "ets" anywhere within its text. The following visual confirmation illustrates the result of applying this formula, validating its efficacy in complex data filtering:



	A	B	C	D	E	F	G
1	**Team**	**Points**					
2	Mavs	24		50			
3	Cavs	30					
4	Nets	17					
5	Rockets	21					
6	Hornets	12					
7	Bucks	15					
8	Bulls	34					
9	Warriors	22					
10	Wizards	15					
11							
12							
13							
14							
15							
16							
17							
18							
19							

As demonstrated by the output, the formula accurately computes the total points for all teams containing "ets" to be **50**. This confirms that the leading and trailing **asterisk** wildcards are highly effective for achieving flexible partial text matching. A manual verification confirms this result:

Teams with "ets" in name: Nets (17 points), Rockets (21 points), Hornets (12 points)
Sum of points: 17 + 21 + 12
Final Sum: 50

Example 2: Summing Values Based on Starting Text (Prefix Match)

In certain analytical scenarios, data aggregation must be focused solely on items that share a common prefix, allowing for categorization based on the beginning of a text string. For instance, you might need to sum sales figures for all products whose names start with a specific

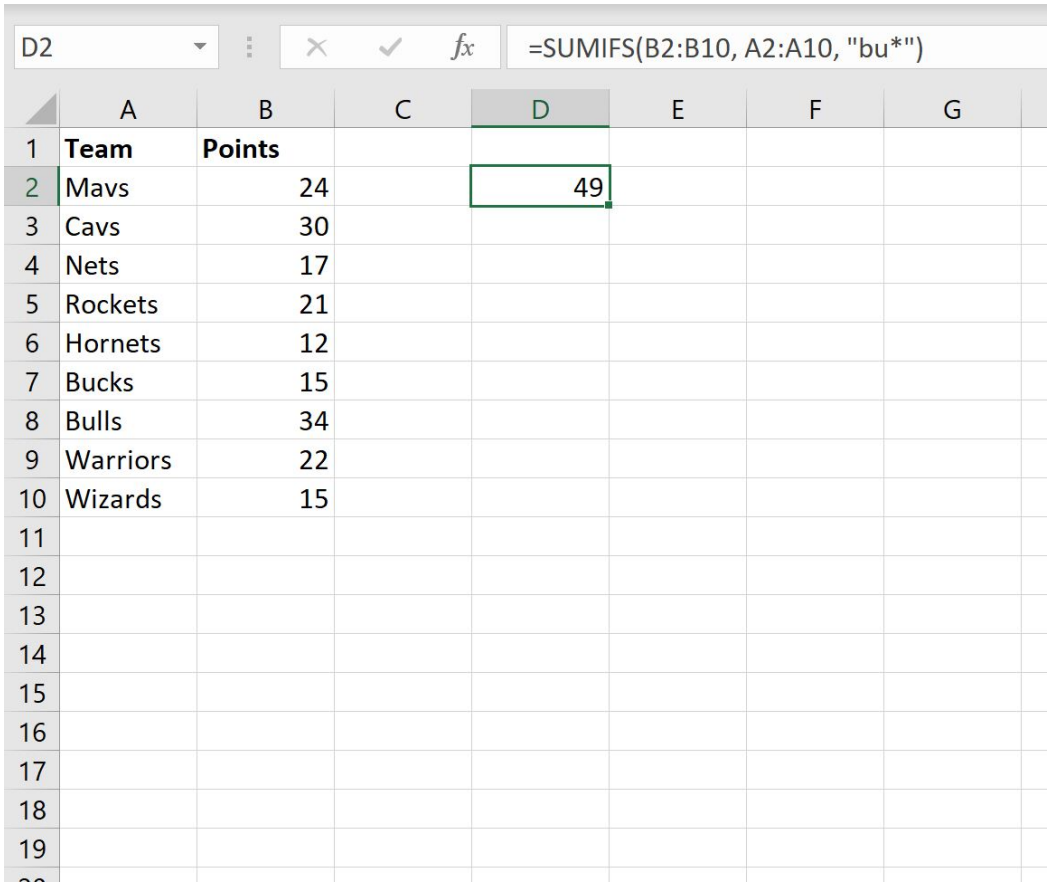
departmental code. In our dataset, we aim to sum the points exclusively for teams whose names commence with the characters "bu". This requires a precise application of the wildcard to ensure that "bu" is the fixed starting sequence.

To enforce this prefix requirement, the **wildcard characters**, specifically the asterisk, must be placed only at the end of the specified criterion. This configuration ensures that "bu" is the fixed starting point, followed by any sequence of zero or more characters. Crucially, this setup excludes any team names that might contain "bu" internally but do not begin with it. The formula implementing this start-of-string matching is:

=SUMIFS(B2:B10, A2:A10, "bu*")

Here, the `sum_range` is `B2:B10`, and the `criteria_range1` is `A2:A10`. The decisive factor is the criterion `"bu*"`. This specific instruction directs **Excel** to include only those team names in the sum that commence with "bu" and are followed by zero or more additional characters. This approach is highly valuable for generating summary statistics based on initial categorization labels within a dataset, ensuring that only items like "Bucks" or "Bulls" are included.

The result of applying this formula demonstrates the precise nature of the prefix match, as shown in the following screenshot:



	A	B	C	D	E	F	G
1	**Team**	**Points**					
2	Mavs	24		49			
3	Cavs	30					
4	Nets	17					
5	Rockets	21					
6	Hornets	12					
7	Bucks	15					
8	Bulls	34					
9	Warriors	22					
10	Wizards	15					
11							
12							
13							
14							
15							
16							
17							
18							
19							

Upon execution, Excel calculates the total points for teams starting with "bu" to be **49**. This result is confirmed by manually isolating the matching entries:

Team names that start with "bu": Bucks (15 points), Bulls (34 points)
Sum of points: 15 + 34
Final Sum: 49

This manual verification confirms the accuracy and targeted filtering provided by the starting text wildcard criteria using the **SUMIFS** function.

Example 3: Summing Values Based on Ending Text (Suffix Match)

Conversely, there are situations where you need to sum data based on a specific suffix or ending string. For example, if you're tracking product variants and want to sum all items ending with "Pro" or "Variant B." In our demonstration, our objective is to sum the values in the **Points** column exclusively for teams in the **Team** column whose names conclude with the text "avs". This scenario also leverages the **asterisk** wildcard, but positioned differently to enforce the end-of-string match.

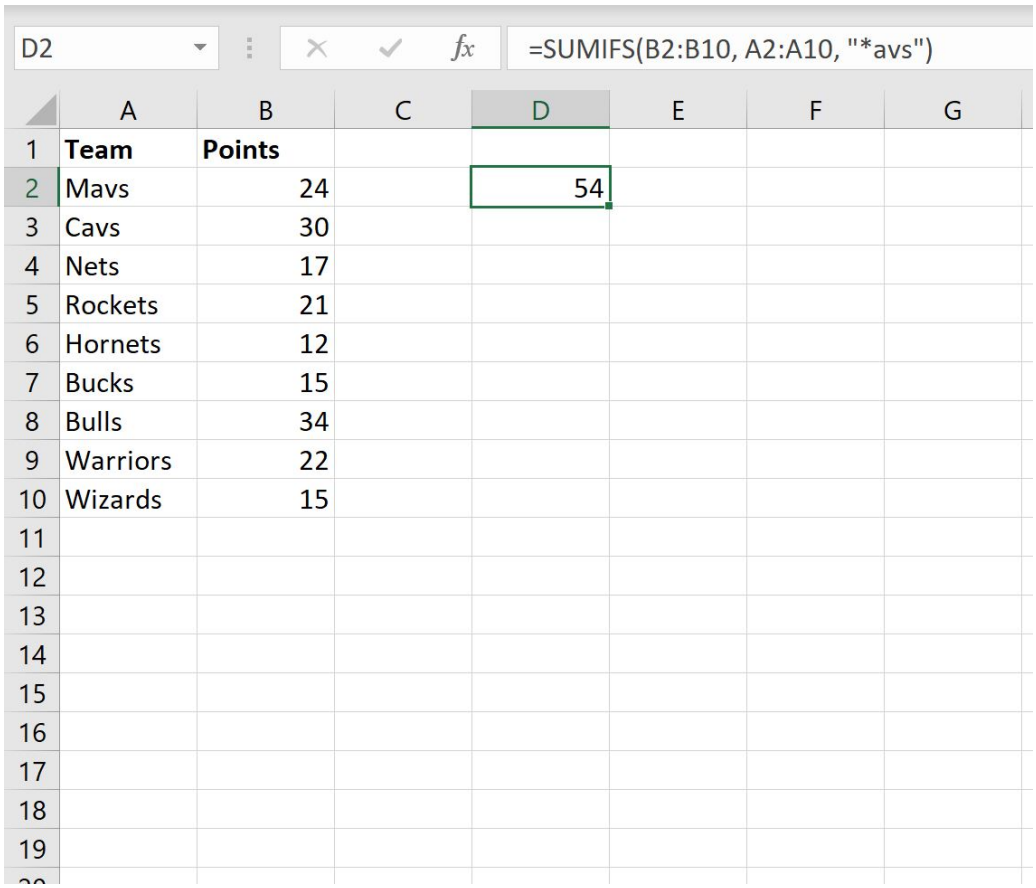
To achieve this end-of-string match, the **asterisk** wildcard must be positioned at the beginning of

the criterion. By using `"*avs"`, we instruct **Excel** to look for any sequence of characters followed by "avs", ensuring that "avs" is the definitive end to the team name. This effectively filters out any team names that might contain "avs" in the middle or beginning but do not conclude with it.

The formula constructed for this suffix-based summation is as follows:

=SUMIFS(B2:B10, A2:A10, "*avs")

Here, `B2:B10` is the `sum_range`, and `A2:A10` is the `criteria_range1`. The criterion `"*avs"` specifies that Excel should only sum points for teams whose names end precisely with "avs". This method provides a powerful solution for grouping data based on final identifiers or categories. The visual result of applying this formula is presented in the following screenshot.



The formula successfully calculates that the sum of points for the teams whose name ends with "avs" is **54**. This perfectly illustrates the utility of a trailing wildcard for end-of-string matching. Let's perform a manual verification to confirm the accuracy of our calculation:

Team names that end with "avs": Mavs (24 points), Cavs (30 points)
Sum of points: 24 + 30

Final Sum: 54

The manual verification confirms that the **SUMIFS** function, utilizing the `"*avs"` wildcard criterion, precisely identified and summed the points for the relevant teams, resulting in a correct total of 54.

Conclusion: Expanding Your Conditional Aggregation Skills

Mastering the integration of **wildcard characters**, particularly the versatile **asterisk**, within the **SUMIFS** function represents a significant advancement in data handling within **Excel**. By enabling partial and pattern-based text matching, analysts are no longer restricted to exact criteria, allowing for far more flexible and robust conditional summing across complex and imperfect datasets. Whether the goal is to aggregate figures based on keywords contained anywhere in a string, or precisely match items based on their prefixes or suffixes, wildcards provide the essential adaptability needed for advanced statistical analysis.

The practical examples provided herein--covering containment, starting text, and ending text matches--serve as a fundamental toolkit for conditional aggregation. It is important to note that this technique is highly transferrable; the same principles apply to other powerful conditional functions. For instance, you can use identical wildcard criteria in **COUNTIFS** to count occurrences based on partial text, or in **AVERAGEIFS** to calculate averages under similar flexible conditions. Incorporating these sophisticated methods into your routine data workflow will dramatically streamline processing time, minimize manual data cleaning efforts, and facilitate deeper, more accurate insights derived from your raw data.

Further Learning and Resources

To maximize your proficiency and continue enhancing your specialized skills in **Excel**, particularly concerning advanced formulas and dynamic data manipulation, we encourage exploring resources that delve into multi-criteria functions and array formulas. Expanding knowledge in these areas is crucial for becoming a truly masterful data analyst capable of tackling any data structure.

The following tutorials offer explanations and guidance on performing other common, yet complex, data tasks in Excel: