

Excel: Use Find and Replace with Wildcards

Authored by
stats writer

November 17, 2025

RECOMMENDED CITATION

stats writer (2025). *Excel: Use Find and Replace with Wildcards*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=95103>

Microsoft Excel stands as one of the most popular and critically acclaimed software programs for modern data analysis and spreadsheet management. Its robust capabilities allow users to rapidly process, organize, and analyze massive volumes of raw data, transforming complex figures into actionable, meaningful insights. Beyond standard calculations, Excel excels at automating repetitive tasks and performing intricate data manipulations. Central to its efficiency is the powerful Find and Replace feature, which becomes exponentially more useful when combined with the use of wildcards.

The integration of wildcards allows users to execute sophisticated searches that transcend simple exact-match queries. This functionality is essential for data cleaning, standardization, and large-scale editing, enabling users to identify and modify specific textual patterns even if the spelling, length, or punctuation varies across the dataset. For anyone managing substantial databases or needing precision and speed in document editing, understanding how to leverage Find and Replace with wildcards is an indispensable skill set, significantly improving accuracy and workflow efficiency.

Understanding Find and Replace with Wildcards

The primary goal of the **Find and Replace** dialogue box in Excel is to quickly locate specific character strings within selected cells and substitute them with alternative text or remove them entirely. While simple text searching is straightforward, the introduction of wildcard characters elevates this feature into a critical data management tool, allowing for partial matching and pattern recognition.

One significant advantage of utilizing the **Find and Replace** mechanism is the ability to apply the following standard wildcard characters to facilitate complex and targeted replacements:

The Essential Wildcard Characters in Excel

In Excel, the two most common wildcard characters used for text matching are the asterisk (`*`) and the question mark (`?`). For advanced scenarios, the tilde (`~`) is used to escape the meaning of the other two wildcards if you need to search for the literal character itself.

The asterisk (`*`) is perhaps the most powerful wildcard, as it represents any sequence of characters of any length. This means that using an asterisk allows you to match zero or more characters in that position. Conversely, the question mark (`?`) represents any single character. This distinction is crucial when defining precise search patterns.

The following three examples focus on using the asterisk to achieve common data cleaning goals:

Replace All Characters Between Two Specific Characters

This technique utilizes the asterisk placed between two defined boundary characters (like parentheses or dashes). It is highly effective for isolating and removing variable content that is bracketed by fixed markers. For example, searching for

th*t

would match "this is a test" if the goal was to find a word starting with 'th' and ending with 't'.

Replace All Characters Before Specific Characters

By placing the asterisk before the target string, we instruct Excel to match and replace any preceding characters. This is useful for removing prefixes, unique identifiers, or extraneous data that appears before a standard core keyword or phrase.

Replace All Characters After Specific Characters

Conversely, placing the asterisk after the target string tells Excel to match the target string itself plus any subsequent characters. This is ideal for trimming suffixes, comments, or variable data that occurs at the end of a standardized entry.

Visualizing Wildcard Replacement Patterns

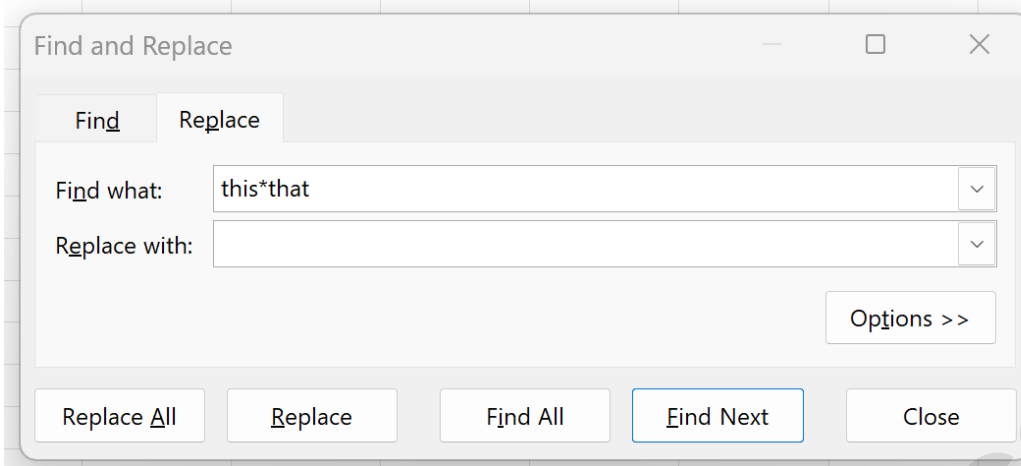
To better illustrate these concepts, consider the visual representation of how the search pattern interacts with the data:

Pattern 1: Replacing Content Between Two Markers

This example demonstrates replacing the defined strings "this" and "that" along with all characters **between** these two strings. The search pattern would look something like

this*that

in the Find what box.

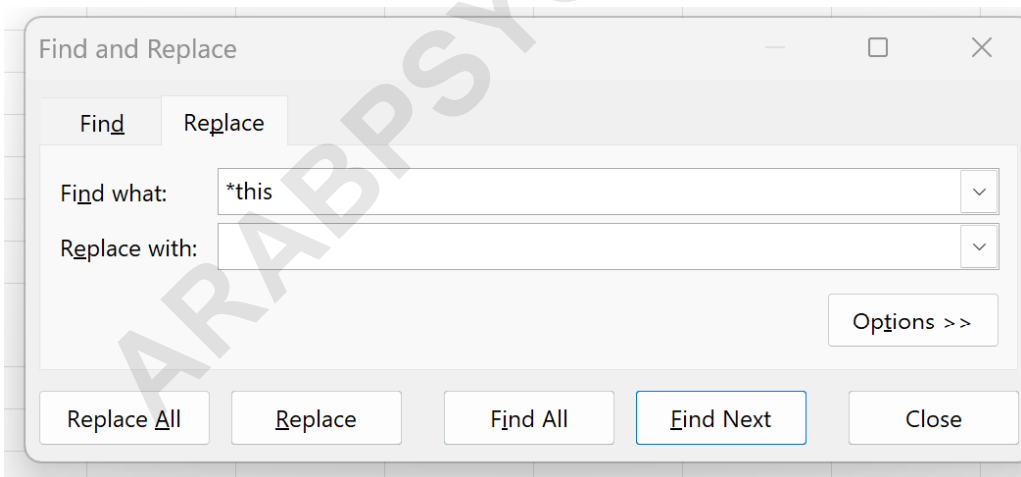


Pattern 2: Replacing Content Before a Marker

This particular example replaces the string "this" along with all characters that come **before** this string in the cell. The search pattern used here would be

*this

, effectively removing everything from the start of the cell up to and including the word "this."

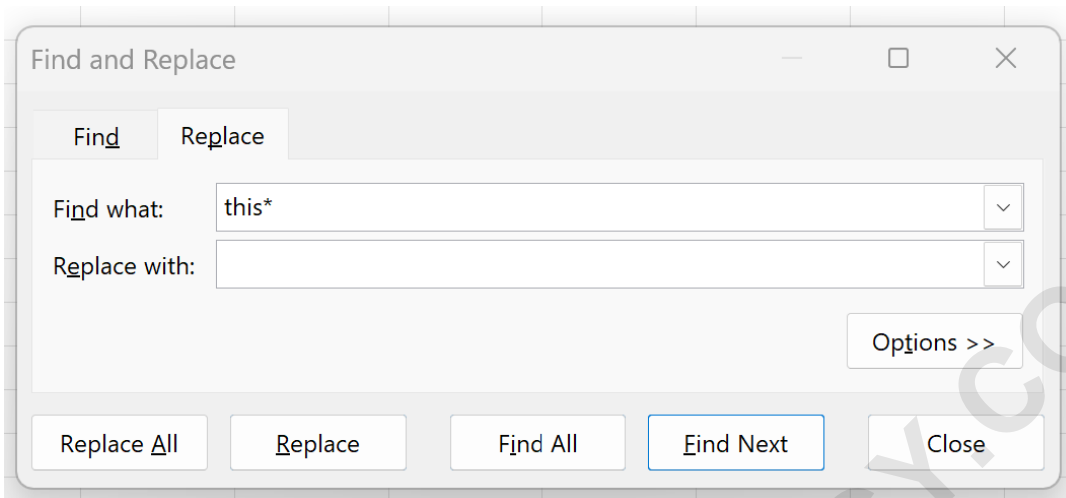


Pattern 3: Replacing Content After a Marker

In this scenario, we replace the string "this" along with all characters that come **after** this string in the cell. The search string would be

this*

. This is highly useful for cleaning up fields where the data structure is inconsistent after a fixed point.



The following practical examples demonstrate how to implement each of these powerful methods within a real-world dataset.

Method 1: Replacing Characters Between Specific Boundaries (Using Parentheses Example)

Consider a scenario involving a basketball team roster dataset. This dataset includes the player's name and their position, where the position field contains auxiliary information enclosed in parentheses that must be removed for standardization.

Example Dataset Structure:

	A	B	C	D
1	Name	Position		
2	Andy	(Starter) Guard		
3	Bob	(Backup) Guard		
4	Chad	(Backup) Guard		
5	Doug	(Starter) Forward		
6	Eric	(Backup) Forward		
7	Frank	(Backup) Forward		
8	Greg	(Starter) Center		
9	Henry	(Backup) Center		
10	Isaac	(Backup) Center		
11				
12				
13				
14				
15				

Our objective is to clean the **Position** column by replacing all characters, including the parentheses, that are found between the opening and closing bracket in each cell. This will leave only the primary position name (e.g., "Guard," "Forward").

The first step involves selecting the target range, which is **B2:B13** in this case. This ensures the operation is confined only to the position data. Next, press **Ctrl + H** (or navigate to the Find and Replace dialog) to initiate the replacement sequence.

In the **Find what** box, we enter the pattern **(*)**. Crucially, the asterisk is placed between the literal opening parenthesis

(

and closing parenthesis

)

. The **Replace with** box is left empty, as the intention is to delete the matched pattern.

Upon clicking **Replace All**, the parentheses and all characters contained within them are successfully removed from every cell in the selection, resulting in a clean, standardized column:

	A	B	C	D	E	F	G	H
1	Name	Position						
2	Andy	Guard						
3	Bob	Guard						
4	Chad	Guard						
5	Doug	Forward						
6	Eric	Forward						
7	Frank	Forward						
8	Greg	Center						
9	Henry	Center						
10	Isaac	Center						
11								
12								
13								
14								
15								
16								
17								
18								
19								
20								
21								
22								
23								
24								

Find and Replace

Find Replace

Find what: (*)

Replace with:

Replace All Replace Find All Find Next

Microsoft Excel

All done. We made 9 replacements.

OK

Method 2: Targeting and Removing Characters Before a Specific String

For the second scenario, we use the same basketball player dataset, but this time, the goal is to isolate the last part of the position string, eliminating all characters that precede a specific sequence.

Original Dataset:

	A	B	C	D	E
1	Name	Position			
2	Andy	Guard			
3	Bob	Guard			
4	Chad	Guard			
5	Doug	Forward			
6	Eric	Forward			
7	Frank	Forward			
8	Greg	Center			
9	Henry	Center			
10	Isaac	Center			
11					
12					
13					
14					
15					

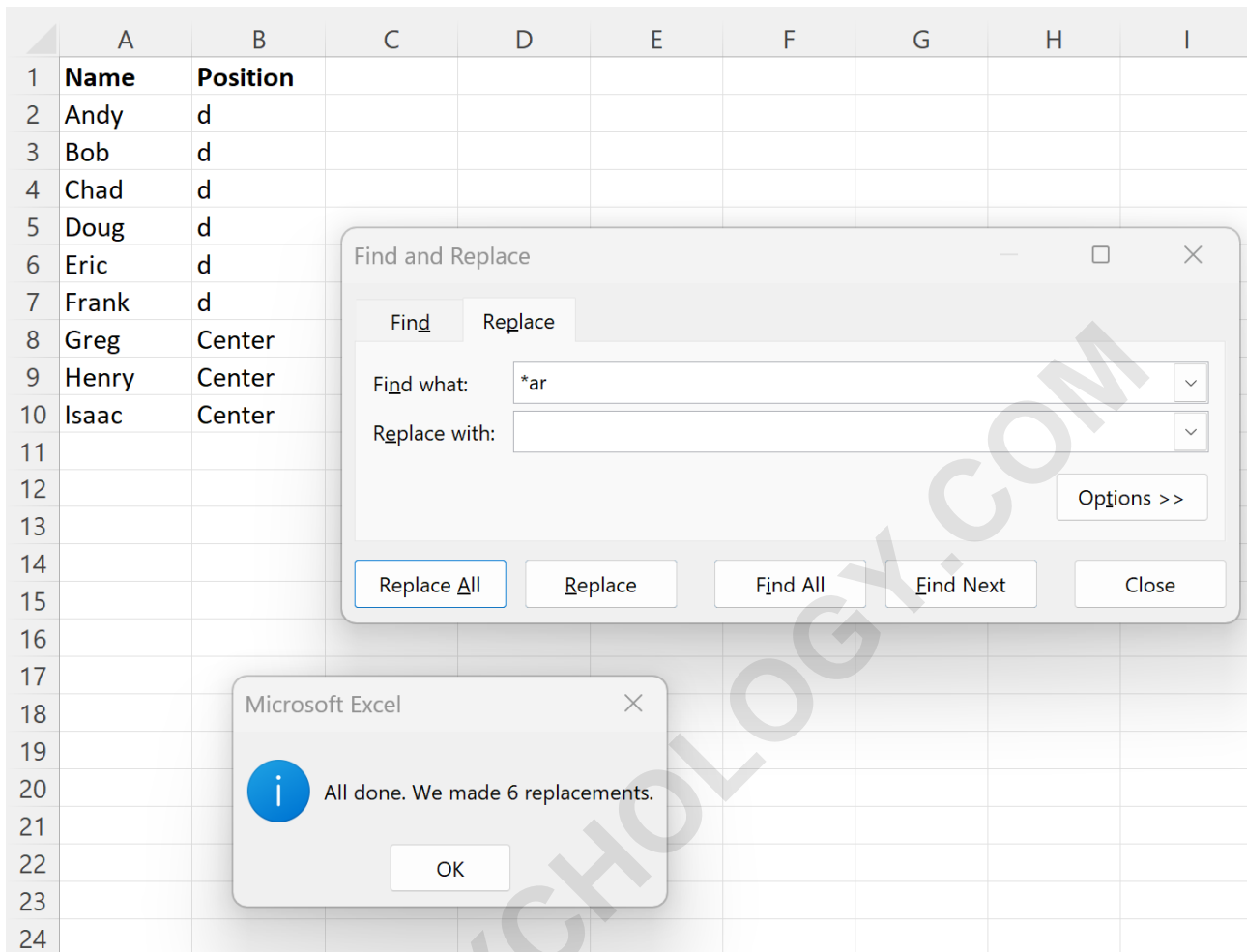
Suppose we wish to remove the string "ar" along with all preceding characters in each cell within the **Position** column. This might be necessary if we only want to retain positions that end in "ard" (like Guard or Forward).

Begin by highlighting the cell range **B2:B13**. Then, open the **Find and Replace** window by pressing **Ctrl + H**.

In the **Find what** box, the pattern should be set to ***ar**. The asterisk at the beginning indicates "match any number of characters" up to the literal sequence "ar," which is also included in the match. The **Replace with** box remains empty.

Executing **Replace All** removes the identified pattern, leaving only the characters that originally followed "ar" (or leaving the cell untouched if the sequence was not found):

	A	B	C	D	E	F	G	H	I
1	Name	Position							
2	Andy	d							
3	Bob	d							
4	Chad	d							
5	Doug	d							
6	Eric	d							
7	Frank	d							
8	Greg	Center							
9	Henry	Center							
10	Isaac	Center							
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									



The screenshot shows the 'Find and Replace' dialog box in Microsoft Excel. The 'Find what' field contains the wildcard pattern '*ar'. The 'Replace with' field is empty. The 'Replace All' button is highlighted. Below the dialog box is a message box that says 'All done. We made 6 replacements.' with an 'OK' button.

Note: Cells containing the position "Center" were unaffected because they did not contain the target string "ar." This illustrates how the wildcard feature performs highly targeted modifications based on specific character patterns.

Method 3: Eliminating Content Following a Defined Marker

Our final practical example demonstrates how to use the wildcard to remove data that appears after a specific marker. We return to our dataset showing the name and position of various basketball players:

	A	B	C	D	E
1	Name	Position			
2	Andy	Guard			
3	Bob	Guard			
4	Chad	Guard			
5	Doug	Forward			
6	Eric	Forward			
7	Frank	Forward			
8	Greg	Center			
9	Henry	Center			
10	Isaac	Center			
11					
12					
13					
14					
15					

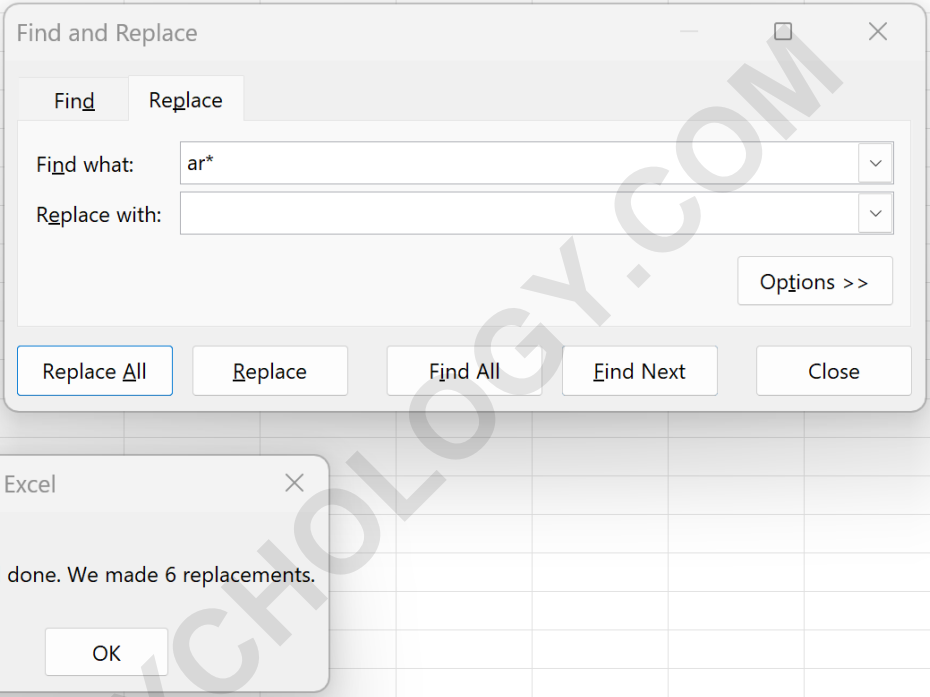
This time, we aim to replace the string "ar" and all characters that come **after** this string in each cell in the **Position** column. This action effectively truncates the cell content at the point where "ar" is found.

The procedure starts by highlighting the cell range **B2:B13** and activating the **Find and Replace** dialogue using **Ctrl + H**.

The critical search pattern is **ar***, placed in the **Find what** box. The asterisk following "ar" instructs Excel to match the literal sequence "ar" and any subsequent characters until the end of the cell content. The **Replace with** field is once again left blank for deletion.

Clicking **Replace All** executes the command, removing "ar" and all trailing characters from the selected cells:

	A	B	C	D	E	F	G	H	I
1	Name	Position							
2	Andy	Gu							
3	Bob	Gu							
4	Chad	Gu							
5	Doug	Forw							
6	Eric	Forw							
7	Frank	Forw							
8	Greg	Center							
9	Henry	Center							
10	Isaac	Center							
11									
12									
13									
14									
15									
16									
17									
18									
19									
20									
21									
22									
23									
24									
25									
26									



The screenshot shows the 'Find and Replace' dialog box in Microsoft Excel. The 'Find' tab is selected, and the 'Find what:' field contains the text 'ar*'. The 'Replace with:' field is empty. The 'Options >>' button is visible. Below the dialog box is a Microsoft Excel message box that says 'All done. We made 6 replacements.' with an 'OK' button.

Note: As with the previous example, any cell containing the position "Center" remains unchanged, as the search pattern **ar*** was not matched within that string. This confirms the precise, conditional nature of wildcard replacement.

Advanced Considerations and Best Practices

While the examples above focus on the powerful asterisk wildcard, users should be aware of several advanced considerations to maximize efficiency and avoid data loss. First, when opening the Find and Replace dialog, always check the 'Options' menu. Within options, you can specify whether the search should be **Case sensitive** (matching capitalization exactly) or if it should search within the entire workbook or just the selected cells. Being mindful of case sensitivity is vital when dealing with structured text data.

Secondly, if your goal is to search for a literal asterisk (

*

) or question mark (

?

) within the data--rather than using them as wildcards--you must use the tilde (~) character to escape their special meaning. For example, to find the string

10*

exactly, you would search for

10~*

in the **Find what** box. Forgetting this step will cause Excel to interpret the asterisk as a wildcard, potentially replacing unintended data.

Finally, always work on a copy of your dataset when performing large-scale wildcard replacements. Although the Find and Replace feature is efficient, improper pattern construction can lead to irreversible data corruption. A quick review of the results after clicking "Replace All" ensures data integrity before proceeding with further analysis.

In conclusion, the **Find and Replace** feature in Excel, particularly when augmented by wildcard characters, is a fundamental tool for streamlined data manipulation. By mastering the use of the asterisk and its placement--whether between two markers, before a specific string, or after a specific string--users gain unmatched precision in data cleaning and standardization. The practical examples outlined in this guide provide a solid foundation for implementing these methods to effectively manage and transform complex datasets. Utilizing these techniques ensures your data remains clean, consistent, and ready for advanced analysis.