

How to Find Cells Containing Asterisks in Excel

Authored by
stats writer

November 21, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Find Cells Containing Asterisks in Excel*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=99052>

The asterisk symbol (*) holds a dual role within Microsoft Excel. While it is frequently used as the standard mathematical operator for multiplication, its primary function in search operations is that of a wildcard character. This designation means that when attempting to locate a literal asterisk within a spreadsheet, standard search methods often fail, returning results for nearly every cell instead of just those containing the specific symbol.

To successfully locate the literal asterisk character, we must employ specific techniques, ranging from the built-in Find and Replace utility to advanced logical formulas utilizing escape characters. Understanding how Excel interprets these special symbols is crucial for accurate data analysis and manipulation. This guide details both quick search methods and robust formulaic solutions for locating asterisks within your datasets, ensuring precision in identifying cells containing this specific character.

Initial Search Method: Using the Find and Replace Dialogue

The most intuitive way to begin searching for content in Excel is by accessing the Find and Replace dialogue box, usually initiated by pressing the keyboard shortcut **CTRL+F**. If you simply type an asterisk (*) into the Find What field and click Find All, Excel will interpret the asterisk as a wildcard representing any sequence of characters, effectively returning every non-empty cell in the sheet. This behavior, while useful for general pattern matching, completely undermines the goal of finding the literal symbol.

Fortunately, the Find and Replace dialogue box allows for the use of an escape character to neutralize the wildcard function of the asterisk. By typing the tilde symbol (~) immediately before the asterisk (i.e., typing ~*), you instruct Excel to treat the subsequent asterisk as a literal character rather than a wildcard. When you execute the search with ~* in the Find What field, Excel will accurately search the entire sheet and list only the cells that contain the specific, literal asterisk character, providing a quick solution for manual verification.

This method is highly effective for one-off searches or quick checks across a workbook. After entering ~*, clicking the **Find All** button generates a list at the bottom of the dialogue box, displaying the location (cell reference) and content of every matching cell. You can then navigate directly to those cells by clicking on their respective entries in the list. This technique is fundamentally important because the concept of escaping the wildcard character using the tilde is also necessary when writing formulas, which provides a much more dynamic and permanent solution for data validation.

Developing a Formulaic Approach Using SEARCH and ISNUMBER

While the quick Find method is useful, analyzing large datasets often requires a dynamic solution that can flag cells containing the symbol based on a formula. We can achieve this by combining

several powerful functions: the [SEARCH](#) function, the [ISNUMBER](#) function, and the [IF](#) function. The primary goal is to determine if the search returns a starting position (a number) for the asterisk. If it does, the cell contains the symbol; otherwise, it does not.

The core of this formula lies in the [SEARCH](#) function: `SEARCH("~*", A2)`. As previously discussed, we must use "~*" within the quotes. The [SEARCH](#) function attempts to find the specified text ("~*") within the target cell (A2). If the asterisk is found, the function returns a number indicating its starting position (e.g., 5 if it starts at the fifth character). If the asterisk is not found, the function returns an error value (**#VALUE!**). Since we are only interested in whether the symbol exists, we need to convert this result into a simple logical check.

This is where the [ISNUMBER](#) function becomes invaluable. Wrapping `SEARCH("~*", A2)` inside [ISNUMBER](#) allows us to check if the result of the search is a number. If the [SEARCH](#) function returns a position (a number), [ISNUMBER](#) returns **TRUE**. If [SEARCH](#) returns an error (meaning no asterisk was found), [ISNUMBER](#) returns **FALSE**. Finally, we nest this entire logic inside the [IF](#) function to output human-readable responses like "Yes" or "No." The resulting comprehensive formula is shown below, ready for deployment in your spreadsheet:

=IF(ISNUMBER(SEARCH("~*", A2)), "Yes", "No")

This particular formula checks if the cell referenced, **A2**, contains a literal [asterisk](#). If the asterisk is successfully located, the formula returns the text string "Yes," providing clear confirmation. Conversely, if the search fails to find the literal symbol, the formula returns "No." This structure provides a reliable and scalable method for identifying specific characters across extensive lists of data.

Practical Example: Identifying All-Star Players in a Dataset

To illustrate the efficiency of this formula, consider a common scenario in data analysis involving sports statistics. Suppose we possess a dataset containing detailed information about various professional basketball players. In this hypothetical setup, team names are listed in column A, and a convention is used where an [asterisk](#) placed at the end of a team name signifies that the corresponding player has achieved **All-Star** status during their career.

Our objective is to quickly scan column A and generate a parallel column that explicitly labels whether each player is an All-Star or not, based solely on the presence of the asterisk. Manually checking hundreds or thousands of rows would be impractical and error-prone. Instead, we can apply our newly constructed formula to automate this identification process. The initial dataset might look like the representation below, requiring us to identify which entries in Column A contain the distinguishing symbol:

	A	B	C	D	E	F
1	Team	Points				
2	Mavs*	34				
3	Nets	15				
4	Warriors	9				
5	Hawks*	29				
6	Celtics*	20				
7	Lakers	12				
8	Cavs	14				
9	Spurs	7				
10	Rockets*	41				
11	Hornets	13				
12						
13						
14						
15						
16						
17						
18						
19						
20						

To initiate the identification process, we begin by typing the powerful formula into the first cell of our results column, which is **C2** in this specific example. This application starts the logical evaluation against the corresponding data point in cell **A2**. The formula ensures that the SEARCH function correctly interprets the asterisk as a literal character using the tilde symbol (~), thereby guaranteeing that we are only capturing the desired marker and not misinterpreting general cell content.

The formula we enter into cell **C2**, which dictates the logic for identifying the All-Star status, is precisely the robust structure we developed earlier. This setup allows for immediate, accurate flagging based on the presence of the specific character. This single implementation sets the stage for rapid analysis across the entire dataset by leveraging Excel's ability to efficiently propagate formulas.

=IF(ISNUMBER(SEARCH("~*", A2)), "Yes", "No")

Scaling the Formula and Interpreting Results

Once the formula is correctly entered into the starting cell (**C2**), the next crucial step is scaling this

calculation across all remaining rows in column C. This is efficiently accomplished by using the fill handle--the small square at the bottom-right corner of the selected cell. By clicking and dragging this handle downwards, the formula automatically adjusts its cell references (from A2 to A3, A4, and so on) for each subsequent row, a feature known as relative referencing in Excel.

As the formula propagates down column C, it instantaneously processes the team name in the adjacent row of column A, determining if the entry contains the literal asterisk. The results provide a clear, binary status for each player. The resulting output, after successfully applying the drag-and-fill operation to the entire column, will clearly demarcate the All-Star players from the rest of the roster, as visualized in the screenshot below:

	A	B	C	D	E	F	G
1	Team	Points	All-Star?				
2	Mavs*	34	Yes				
3	Nets	15	No				
4	Warriors	9	No				
5	Hawks*	29	Yes				
6	Celtics*	20	Yes				
7	Lakers	12	No				
8	Cavs	14	No				
9	Spurs	7	No				
10	Rockets*	41	Yes				
11	Hornets	13	No				
12							
13							
14							
15							
16							
17							
18							
19							

The interpretation of the results column is straightforward: the formula returns "Yes" if the team name entry contains the identifying asterisk, confirming the player's All-Star status. Conversely, it returns "No" if no asterisk is found, indicating that the player has not been designated as an All-Star under this specific data convention. This structured output facilitates immediate filtering, sorting, or subsequent complex analysis based on this important attribute. Consider the following examples from the dataset:

Mavs* contains a literal asterisk, resulting in a "Yes" output.

Nets does not contain an asterisk, resulting in a "No" output.

Warriors does not contain an asterisk, resulting in a "No" output.

This automated process highlights how essential the proper handling of wildcards is in spreadsheet operations. Utilizing the escape character within the SEARCH function ensures that our logical checks are based on the intended symbol, delivering precise and reliable classification across all entries in the dataset.

Customizing Output Values Beyond Yes/No

The default output values of "Yes" and "No" in the primary formula `=IF(ISNUMBER(SEARCH("~*", A2)), "Yes", "No")` are practical for simple binary classification. However, the true strength of the IF function lies in its flexibility to return any specified value, text string, or even another calculation based on the condition being met or unmet. Data analysts frequently need to replace these simple strings with more descriptive labels, numerical flags, or calculated values that feed into subsequent analytical models.

If, for instance, you required the output to be "All-Star" instead of "Yes" and "Standard Player" instead of "No," you would simply modify the `value_if_true` and `value_if_false` arguments within the IF function. The revised structure would look like this: `=IF(ISNUMBER(SEARCH("~*", A2)), "All-Star", "Standard Player")`. This customization is critical for generating reports that are immediately understandable to non-technical stakeholders or for integrating results directly into dashboards that rely on specific textual inputs.

Furthermore, the output values do not have to be text strings. You could return a numerical value, such as 1 for a match and 0 for no match, which is extremely useful for generating scores or performing summations. For example, `=IF(ISNUMBER(SEARCH("~*", A2)), 1, 0)` provides a numerical flag that can easily be totaled using the SUM function to count the total number of All-Star players in the list. This adaptability ensures the formula serves not just as a validator but as an integral component of a larger computational workflow within Excel.

Returning Boolean Values (TRUE or FALSE) for Logic Operations

For advanced logical operations, conditional formatting, or integration with array formulas, it is often more efficient to rely on native boolean outputs (**TRUE** or **FALSE**) rather than converting them into "Yes" or "No" strings. Since the middle section of our original formula--`ISNUMBER(SEARCH("~*", A2))`--already produces a definitive boolean result, we can simplify the expression dramatically by removing the outer IF function wrapper.

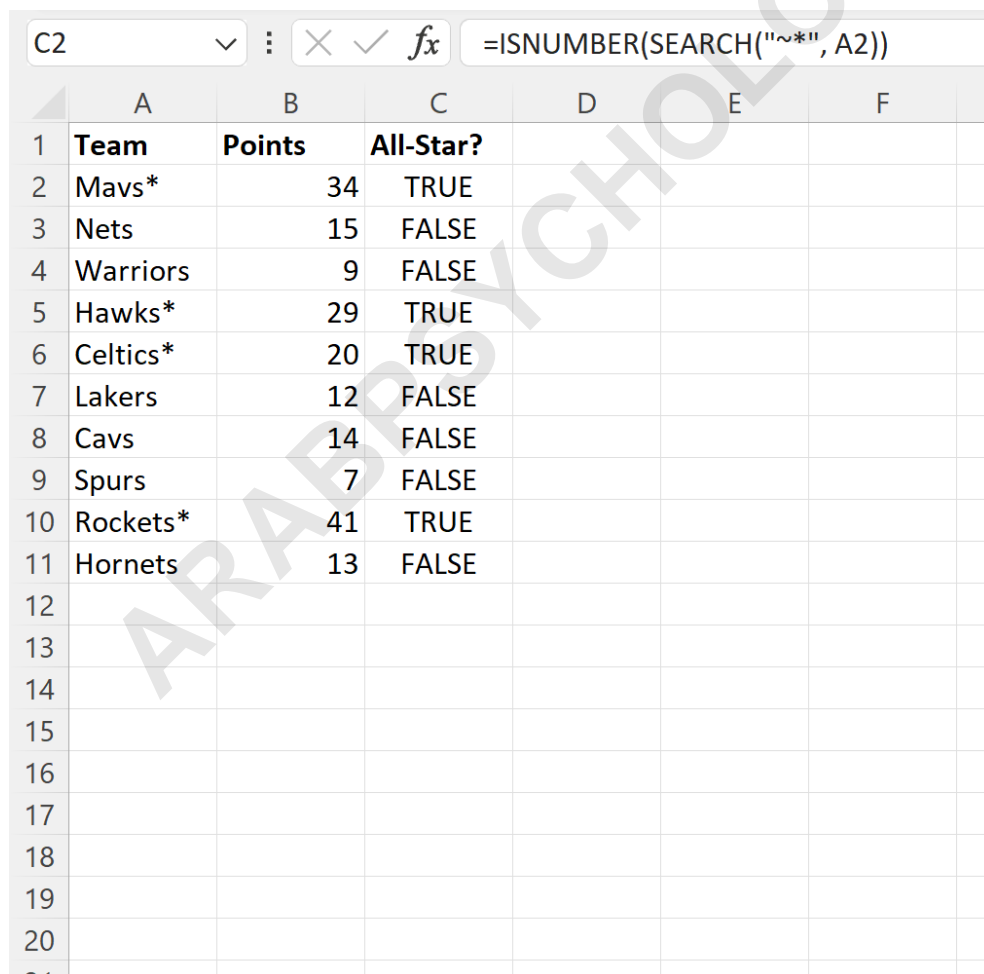
By using the simplified formula, we directly leverage the output of the ISNUMBER function. If the

search successfully finds the literal asterisk, `ISNUMBER` detects a numeric position and returns **TRUE**. If the search fails, resulting in an error, `ISNUMBER` returns **FALSE**. This approach provides the most computationally efficient method for determining the presence of the special character.

The following formula represents this optimized boolean approach, which is ideal when the result needs to feed directly into another logical test (e.g., within an AND or OR statement, or as the condition for conditional formatting):

`=ISNUMBER(SEARCH("~*", A2))`

Applying this formula to our existing dataset demonstrates its utility. The result is a column populated strictly with the boolean values **TRUE** or **FALSE**, providing a foundation for subsequent automation. The screenshot below illustrates how this streamlined formula appears in practice, confirming its ability to accurately identify the presence of the literal asterisk without the overhead of text conversion:



	A	B	C	D	E	F
1	Team	Points	All-Star?			
2	Mavs*	34	TRUE			
3	Nets	15	FALSE			
4	Warriors	9	FALSE			
5	Hawks*	29	TRUE			
6	Celtics*	20	TRUE			
7	Lakers	12	FALSE			
8	Cavs	14	FALSE			
9	Spurs	7	FALSE			
10	Rockets*	41	TRUE			
11	Hornets	13	FALSE			
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						

As evident in the output, the formula returns **TRUE** when the team name contains the literal

asterisk, signifying an All-Star designation. It returns **FALSE** for all other entries. This raw boolean output is preferable when the results are intended for programmatic use within the spreadsheet environment, providing the most direct link to Excel's underlying logical architecture.

Summary of Techniques for Asterisk Searching

Successfully searching for a literal asterisk in Excel requires moving past the default interpretation of the symbol as a wildcard character. By consistently applying the tilde symbol (~) as the essential escape character, users can compel Excel to treat the asterisk as a piece of text to be located, rather than a command to match any sequence of characters.

The choice of method depends entirely on the user's needs. For a quick, manual identification of asterisk-containing cells, the **CTRL+F Find and Replace dialog box** using the search string ~* is the fastest solution. For creating dynamic reports, conditional logic, or derived columns that must automatically update with source data, utilizing the comprehensive formula structure based on SEARCH and ISNUMBER is superior. This formula can be tailored to return descriptive text, numerical flags, or pure boolean values, offering maximum analytical flexibility.

Mastering the use of escape characters in Excel is a fundamental skill that extends beyond just the asterisk; it applies to searching for other wildcards like the question mark (?) as well. By consistently applying these principles, you ensure data integrity and unlock advanced capabilities in pattern matching and data validation across complex spreadsheet operations.