

Learning to Find Asterisks in Excel Cells: A Comprehensive Guide

Authored by
Mohammed loot

January 24, 2026

RECOMMENDED CITATION

Mohammed loot (2026). *Learning to Find Asterisks in Excel Cells: A Comprehensive Guide*. PSYCHOLOGICAL STATISTICS. Retrieved from <https://statistics.arabpsychology.com/?p=2939>

When managing extensive datasets within [Excel](#), it is common to encounter specialized characters used for signaling specific attributes, classifications, or flags. Chief among these is the [asterisk](#) (*), which typically functions as a powerful [wildcard character](#) across many software environments, including Excel's native search capabilities. This inherent wildcard behavior, while useful for pattern matching, presents a significant hurdle when the objective is to locate cells containing the asterisk as a literal symbol, rather than as a substitute for any sequence of characters.

The ability to accurately search for and identify specific special characters, such as the literal asterisk, is fundamental to effective [data cleaning](#), validation, and analysis workflows. Whether your goal is to track distinct data flags, pinpoint data entry anomalies, or systematically categorize information based on unique markers, mastering this precise detection technique ensures both accuracy and efficiency in spreadsheet management. This comprehensive guide details a reliable, function-based method to isolate and confirm the presence of a literal asterisk in any cell, explaining the logic behind the solution and providing practical, step-by-step examples.

Overcoming the Wildcard Challenge: The Escape Character

The challenge stems from the default behavior of text functions in [Excel](#), where the [asterisk](#) (*) is interpreted as a [wildcard character](#). When used in functions such as [SEARCH](#) or [MATCH](#), the asterisk does not represent itself; instead, it matches any string of characters of any length. For example, executing a search for "data*" would successfully locate entries like "data", "database", and "datamart". This powerful pattern-matching feature is essential for flexible queries, but it is precisely this behavior that must be bypassed when looking for the symbol itself.

To resolve this conflict, [Excel](#) offers a specific mechanism: the [tilde symbol](#) (~), which acts as an escape character. When the tilde is placed immediately preceding a wildcard character in a search pattern, it signals to Excel that the following character should be treated as a literal character, stripping away its special function. Therefore, to correctly search for a literal asterisk, the pattern must be specified as "~*". This fundamental concept is the cornerstone of reliably identifying cells that contain the asterisk symbol.

The robust technique presented here integrates several native Excel functions to achieve precise detection. We utilize the [SEARCH function](#) to pinpoint the location of the escaped asterisk, the [ISNUMBER function](#) to logically confirm whether the search was successful (i.e., returned a position number), and finally, the [IF function](#) to translate the technical result into a clear, user-friendly output. This layered approach ensures both high precision and immediate clarity in your results, proving highly valuable across various data analysis scenarios.

Implementing the Core Detection Formula

To definitively ascertain whether a specific cell in [Excel](#) contains a literal [asterisk](#), we rely on a

specialized formula structure explicitly designed to manage the symbol's [wildcard](#) nature. This formula efficiently scans the target cell for the presence of the escaped sequence "~*" and delivers a simple, unambiguous "Yes" or "No" verdict.

The standard formula used to check if a target cell contains an asterisk anywhere within its contents is structured as follows:

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

In this configuration, the formula examines the content of cell **A2**. If the cell contains a literal asterisk, the function returns the string "Yes"; otherwise, if the asterisk is not found, it returns "No."

A detailed understanding of each function within this formula is essential for effective data manipulation:

SEARCH("~*", A2): The [SEARCH function](#) attempts to find the starting index of the specified text string within another. Here, the search string is "~*". The [tilde symbol](#) (~) acts as the escape character, ensuring the subsequent [asterisk](#) is treated as a literal character. If the string is found in cell **A2**, **SEARCH** outputs a numerical value corresponding to the starting position. If the asterisk is absent, **SEARCH** returns the **#VALUE!** error.

ISNUMBER(...): This function evaluates whether the output of its argument is a number. When it encloses the [SEARCH function](#), **ISNUMBER** returns **TRUE** if the asterisk was successfully located (because **SEARCH** returned a number). Conversely, if **SEARCH** results in a **#VALUE!** error, indicating failure to locate the character, **ISNUMBER** outputs **FALSE**.

IF(..., "Yes", "No"): The [IF function](#) is a logical test that returns one value if the condition is **TRUE** and another if it is **FALSE**. In this context, if the **ISNUMBER** result is **TRUE** (asterisk present), the [IF function](#) returns "Yes". If the result is **FALSE** (asterisk absent), it returns "No".

It is important to remember that the [tilde symbol](#) (~) is the specific escape sequence in Excel used to treat the subsequent wildcard character as a literal character.

Practical Application: Identifying Data Flags

Suppose you are tasked with analyzing a dataset containing information about various basketball players, as shown in the image below:

	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						

In this scenario, the dataset lists players and their corresponding teams. A critical piece of information is encoded within the team names themselves: an **asterisk** (*) appended to a team name serves as a flag, indicating that the player from that team is designated as an **All-Star**. This is a common and efficient data entry technique used to instantly categorize specific attributes without the necessity of creating an auxiliary column.

Our primary objective is to process this raw data efficiently to confirm, for every player listed, whether their team name contains this distinguishing **asterisk**, thereby systematically identifying all **All-Star** players. For large datasets, attempting to verify each cell manually would be tedious and highly susceptible to human error. This is precisely where the specialized Excel formula proves indispensable, automating the rigorous detection process across the entire range.

To execute this verification, we will apply the asterisk detection formula to the first player's team name, located in cell **A2**, and then rapidly extend this formula down the column. This action will generate a new column of data providing a clear "Yes" or "No" status for each player's potential All-Star designation. To begin the implementation, we enter the following formula into cell **C2**:

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

After typing the formula into cell **C2** and pressing **Enter**, the cell will instantly display either "Yes" or "No," reflecting whether the text string in **A2** contains a literal [asterisk](#). We then propagate this logic to the rest of the column by clicking and dragging the fill handle down column C, as illustrated in the following result:

The screenshot shows an Excel spreadsheet with the following data:

	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 formula bar for cell C2 shows the formula: `=IF(ISNUMBER(SEARCH("~*", A2)), "Yes", "No")`

By utilizing the fill handle (the small square located at the bottom-right corner of cell **C2**), the formula is automatically adjusted for each subsequent row, sequentially checking cells **A3**, **A4**, and so forth. This efficient process ensures the entire column is populated with the correct All-Star status for every player in the dataset, based solely on the presence of the asterisk flag. The output clearly differentiates the data points:

Mavs* is correctly flagged as containing an asterisk, returning "Yes".

Nets is confirmed not to contain an asterisk, returning "No".

Warriors is also confirmed not to contain an asterisk, returning "No".

This automated and visual output provides an immediate, accurate overview of which players have been designated as **All-Stars**, significantly streamlining subsequent data analysis and reporting

requirements. This method is highly scalable, reliable, and serves as an excellent foundational technique for any Excel task that requires precise detection of specific special characters.

Customizing Results and Advanced Search Techniques

The inherent flexibility of the [IF function](#) grants users complete control over the format and content of the output values. While using "Yes" and "No" offers maximum clarity, you may require alternative labels or even wish to integrate the detection result directly into complex calculations based on the presence of the [asterisk](#).

If you need results other than the default "Yes" and "No" strings, you simply substitute these values within the final two arguments of the [IF function](#). For example, you could opt to return "All-Star" and "Regular Player," or even utilize numerical flags like 1 and 0. Numerical outputs are particularly beneficial for subsequent quantitative analysis, aggregation, or for applying advanced [conditional formatting](#) rules.

Alternatively, for scenarios where a simple boolean output is preferred, you can bypass the [IF function](#) entirely, allowing the core logical test to return **TRUE** or **FALSE** directly. This is accomplished using the simplified formula:

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

This concise version leverages the native boolean output of the [ISNUMBER function](#), which returns **TRUE** if the [asterisk](#) is successfully found, and **FALSE** if it is not. This direct boolean result is ideal when the output is intended to serve as a logical test for other complex functions or for driving automatic styling via [conditional formatting](#). The implementation of this formula is shown below:

The screenshot shows an Excel spreadsheet with the following data:

	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						

The formula bar at the top shows the formula: `=ISNUMBER(SEARCH("~*", A2))`

The resulting screenshot clearly demonstrates how this streamlined formula returns **TRUE** or **FALSE** to precisely indicate whether each team name contains a literal asterisk. Using native boolean outputs streamlines complex spreadsheet models, enabling cleaner integration with further logical operations. For instance, this TRUE/FALSE value can be nested efficiently within functions such as [AND](#) or [OR](#) to construct more intricate conditional rules.

Refining Searches: Other Wildcards and Case Sensitivity

While the primary focus of this discussion has been the reliable detection of the literal [asterisk](#), the same underlying principle involving the [tilde escape character](#) (~) is universally applicable to other [wildcard characters](#) utilized by Excel. Notably, the [question mark](#) (?), which is typically used to match any single character, can be searched for literally by simply specifying the pattern "~?". Furthermore, should you ever need to locate a literal [tilde](#) character itself, the required search pattern is "~~".

For more granular pattern matching, such as verifying if an asterisk appears exclusively at the beginning or end of a text string, you can combine the standard [SEARCH function](#) with text

manipulation functions. For example, to confirm an asterisk is the very last character in cell A2, you could use a comparison such as `SEARCH("~*", A2)=LEN(A2)`. This approach leverages functions like [LEFT](#) or [RIGHT](#), depending on the required position check.

It is also critical to recall a key characteristic of the [SEARCH function](#): it is inherently **not case-sensitive**. If your data analysis demands case sensitivity--for example, if you must distinguish between "FLAG*" and "flag*"--you must substitute the [SEARCH function](#) with the [FIND function](#). The [FIND function](#) is case-sensitive but still requires the [tilde escape character](#) (~) to treat wildcards literally. Understanding these subtle yet powerful functional differences allows for comprehensive control over data manipulation in [Excel](#).

Conclusion

Effective data management and rigorous analysis in [Excel](#) fundamentally depend on employing precise methods for identifying specific characters, particularly those that carry special functional weight. The capability to accurately search for a literal [asterisk](#), overriding its default [wildcard behavior](#), is an indispensable skill for any serious data professional.

By mastering the application of the [tilde escape character](#) (~) in concert with core functions like [SEARCH](#), [ISNUMBER](#), and [IF](#), you gain enhanced control over your spreadsheet data classification. This technique not only guarantees reliable data validation but also enables you to extract meaningful and precise insights from complex datasets quickly. Developing proficiency in these seemingly minor yet robust Excel features is paramount to excelling as a data analyst.

Additional Resources

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