

# How to Select Cases in SPSS Based on Text Within a String

Authored by  
**mohammed looti**

January 7, 2026

## RECOMMENDED CITATION

mohammed looti (2026). *How to Select Cases in SPSS Based on Text Within a String*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=124824>

The capability to filter data based on textual content is fundamental in modern data analysis. The Statistical Package for the Social Sciences (SPSS) is fully equipped to handle this type of complex selection. Specifically, SPSS allows users to perform targeted filtering where cases are selected contingent upon whether a textual string variable contains a specific sequence of characters or a predefined substring. This powerful feature enables researchers and analysts to efficiently narrow down large datasets, focusing only on records that meet precise textual criteria.

This process involves utilizing specialized SPSS transformation functions within the dedicated Select Cases dialogue box. By employing functions designed for string manipulation--chief among them the `char.index` function--users can construct a logical condition that evaluates to true only when the target substring is present in the specified variable. This method is far more efficient than manual inspection, especially when dealing with variables containing lengthy descriptive text or coded information.

## The Necessity of Filtering String Data

In many research and business contexts, datasets often contain rich descriptive variables, such as comments, product descriptions, demographic categories, or organization names, all stored as strings. Traditional numerical filtering (e.g., age > 30) is insufficient when the analytical goal requires identifying patterns or instances based on these textual elements. For example, a quality control analyst might need to isolate all customer complaints that contain the word "faulty" or "broken," regardless of where those words appear in the full comment field, making string searching capabilities essential.

The SPSS Select Cases functionality serves as a critical gateway for performing this precise type of textual data analysis. By specifying an explicit condition using string functions, the entire dataset is evaluated row by row. This conditional filtering process effectively creates a temporary subset of the data, allowing subsequent analyses (like frequency counts, summaries, or statistical tests) to be run exclusively on the selected records, thereby sharpening the focus of the research.

## Leveraging the char.index Function for Case Selection

The core mechanism used for textual matching within SPSS conditional expressions is the `char.index` function. This function is specifically designed to perform positional searches within string variables. It takes two primary arguments: the target string (the variable being searched) and the desired substring (the text you are looking for).

When `char.index(Target_String, Substring_to_Find)` is executed, it returns an integer value. This value represents the starting position of the first occurrence of the specified substring within the target variable. If the substring is found at the very beginning of the string, the function returns 1. If it is found later in the string, it returns the corresponding index number (e.g., 5, 12,



etc.).

Crucially, if the specified substring is **not** present in the target string for a given case, the `char.index` function returns the value **0**. This binary outcome (a positive integer indicating presence, or zero indicating absence) allows us to construct a simple yet powerful logical condition for filtering, transforming the complex task of text searching into a straightforward numerical comparison.

## Practical Example: Setting Up the Dataset

To illustrate this powerful procedure, let us consider a hypothetical dataset utilized in sports data analysis. Suppose our dataset tracks performance metrics for various basketball players across different teams. The relevant information includes the player's name, the points they scored, and the name of their team. In this scenario, we are interested in isolating only those players who belong to teams whose names contain a specific identifying abbreviation.

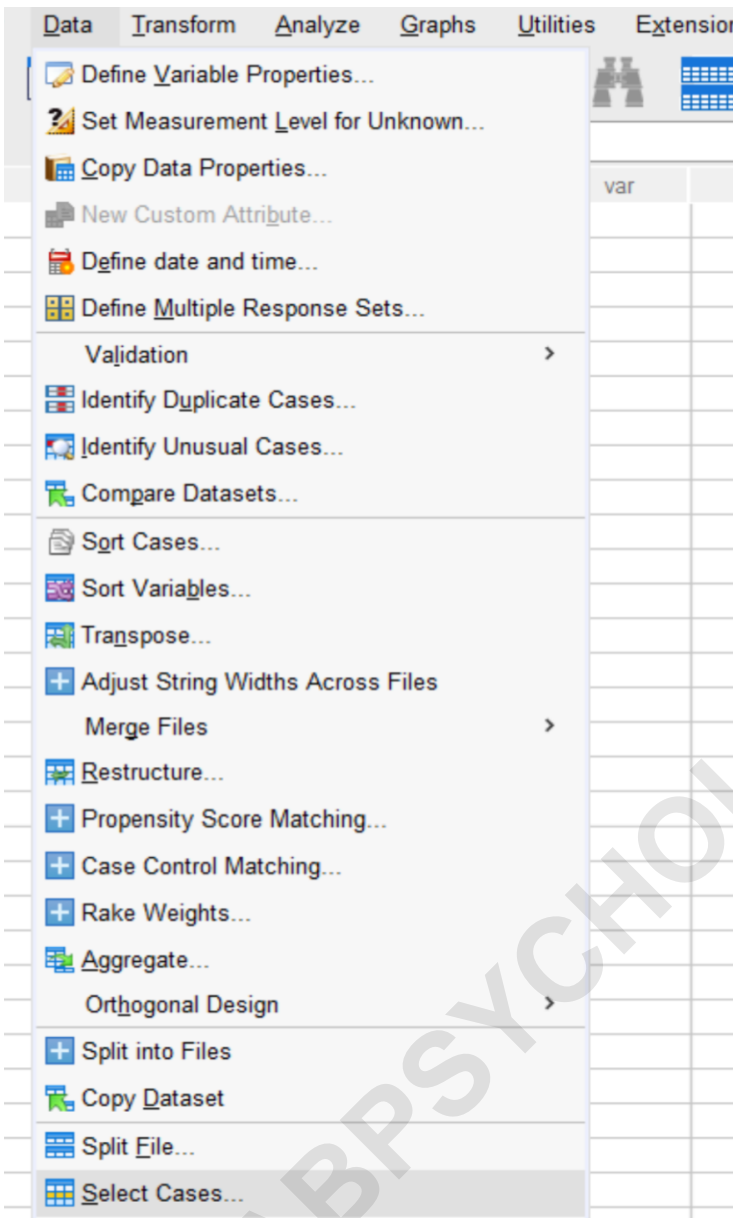
For our example, we will focus on selecting all cases that contain the specific substring "avs" within the **Team** column. The dataset structure, before applying any filtering criteria, would appear similar to the visualization below, showcasing various entries in the **Team** column:

|    |  Team |  Points | var | var |  |
|----|--|--|-----|-----|--|
| 1  | Mavs   | 22.00  |     |     |  |
| 2  | Mavs   | 14.00  |     |     |  |
| 3  | Rockets  | 37.00  |     |     |  |
| 4  | Nets   | 19.00  |     |     |  |
| 5  | Lakers   | 11.00  |     |     |  |
| 6  | Cavs   | 12.00  |     |     |  |
| 7  | Cavs   | 15.00  |     |     |  |
| 8  | Warriors   | 23.00  |     |     |  |
| 9  | Spurs  | 40.00  |     |     |  |
| 10 | Celtics  | 30.00  |     |     |  |
| 11 | Celtics  | 14.00  |     |     |  |
| 12 | Hawks  | 11.00  |     |     |  |
| 13 | Mavs   | 24.00  |     |     |  |
| 14 | Rockets  | 28.00  |     |     |  |
| 15 | Thunder  | 10.00  |     |     |  |
| 16 |  |  |     |     |  |
| 17 |  |  |     |     |  |
| 18 |  |  |     |     |  |
| 19 |  |  |     |     |  |

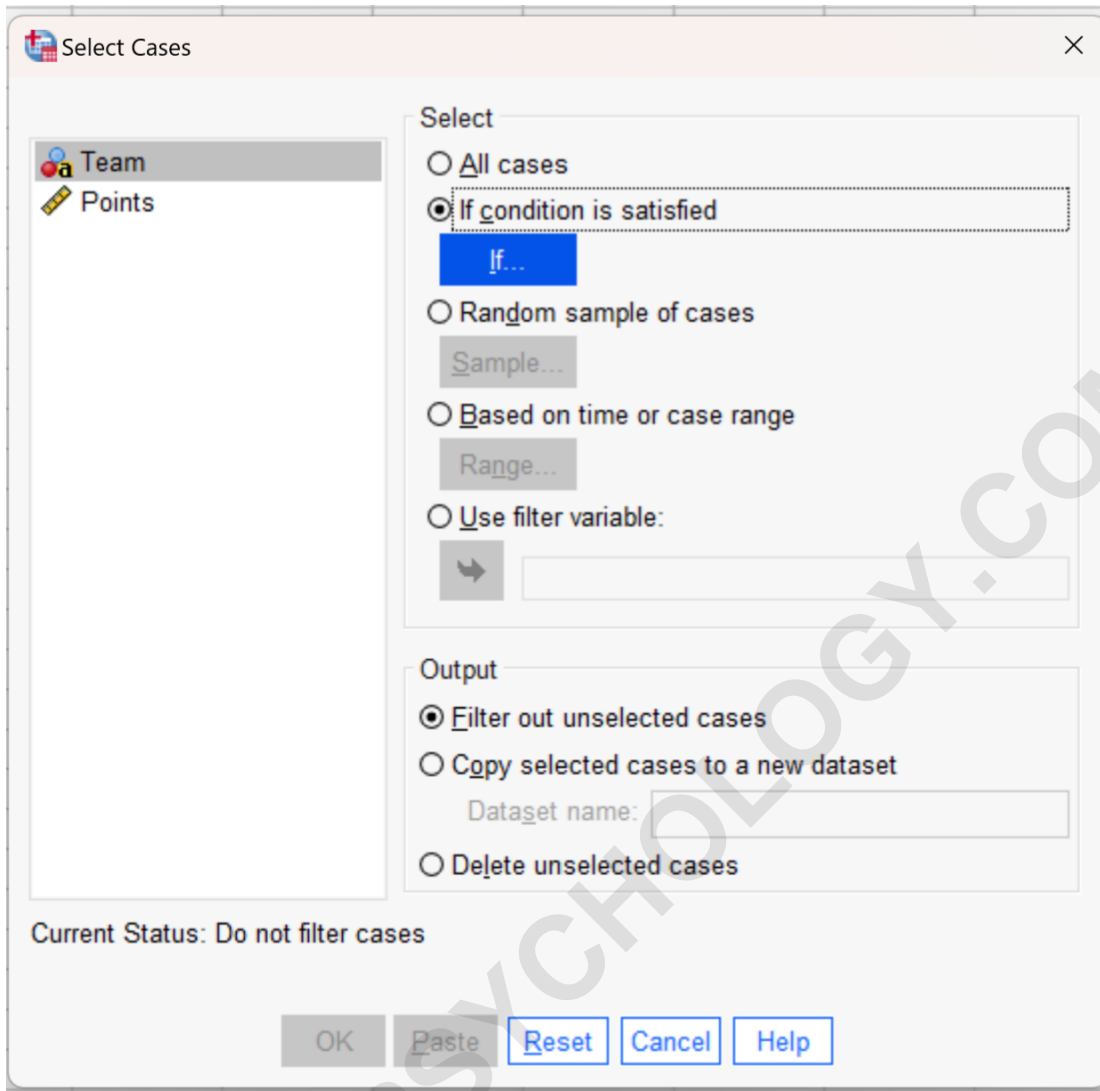
Our objective is clear: we must execute a case selection operation in SPSS that retains only the rows where the string value in the **Team** variable contains the target sequence "avs". This methodology ensures that our subsequent statistical calculations are only performed on the specific subset of data meeting this stringent textual criterion.

### Step-by-Step Guide to Using the Select Cases Dialogue

The process begins by accessing the primary data manipulation tools within the SPSS interface. Navigate to the **Data** tab located in the main menu bar. Within the dropdown options, locate and select the **Select Cases** entry. This action initiates the filtering process and opens the corresponding dialogue box where the conditional logic must be defined.



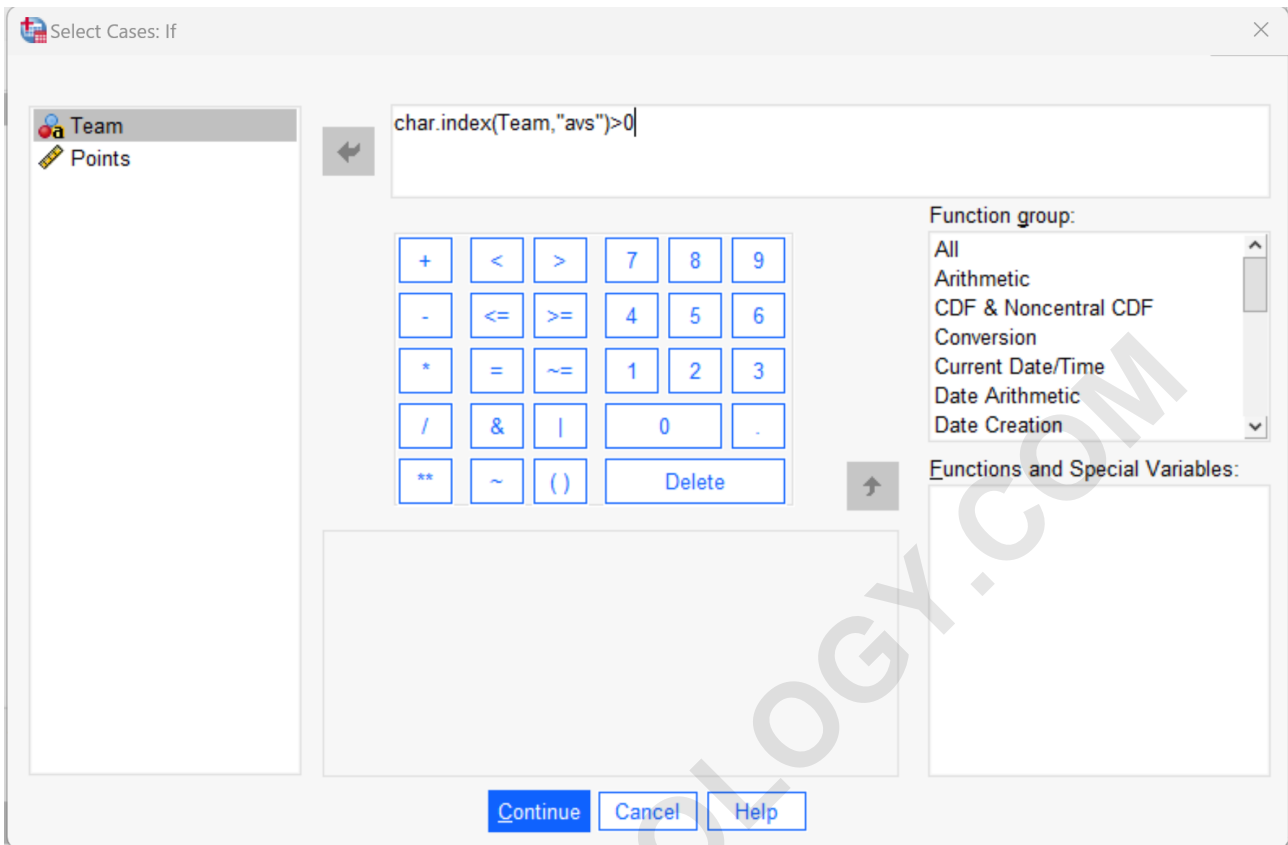
In the initial **Select Cases** window, you must specify the nature of your selection criteria. Choose the radio button labeled **If condition is satisfied**. This tells SPSS that the selection will be based on a logical test applied to each case. Subsequently, click the **If...** button to open the detailed condition builder interface.






Within the condition builder, we must now input the logical expression using the `char.index` function. Recall that we are searching the **Team** variable for the substring "avs." The condition is formulated as follows: we check if the function returns any positive index number. This means we are testing if the result is greater than zero.

Enter the following precise formula into the numerical expression box:

**`char.index(Team,"avs")>0`**



After verifying the formula, click **Continue** to exit the condition builder, and then click **OK** in the main **Select Cases** dialogue box. SPSS immediately processes the instruction, applying the filter across all records. The result is a filtered dataset where cases that do not satisfy the condition (i.e., those whose **Team** name does not contain "avs") are visually marked, typically by being crossed out in the Data View, signifying their exclusion from subsequent analyses.

|    |  Team |  Points |  filter_\$( | var |
|----|--|--|--|-----|
| 1  | Mavs   | 22.00  | 1  |     |
| 2  | Mavs   | 14.00  | 1  |     |
| 3  | Rockets  | 37.00  | 0  |     |
| 4  | Nets   | 19.00  | 0  |     |
| 5  | Lakers   | 11.00  | 0  |     |
| 6  | Cavs   | 12.00  | 1  |     |
| 7  | Cavs   | 15.00  | 1  |     |
| 8  | Warriors   | 23.00  | 0  |     |
| 9  | Spurs  | 40.00  | 0  |     |
| 10 | Celtics  | 30.00  | 0  |     |
| 11 | Celtics  | 14.00  | 0  |     |
| 12 | Hawks  | 11.00  | 0  |     |
| 13 | Mavs   | 24.00  | 1  |     |
| 14 | Rockets  | 28.00  | 0  |     |
| 15 | Thunder  | 10.00  | 0  |     |
| 16 |  |  |  |     |
| 17 |  |  |  |     |
| 18 |  |  |  |     |

## Understanding the Logic: How `char.index` Generates Results

To fully appreciate the robustness of this filtering technique, it is essential to understand the underlying Boolean logic governing the `char.index(Team, "avs") > 0` expression. This single line of code effectively acts as a truth test for every row in the dataset, returning either **True** (1) or **False** (0).

The `char.index` function first calculates the starting character position of the substring "avs" within the string stored in the **Team** variable. If the search is successful (meaning "avs" exists), it returns a positive integer (1, 2, 3, etc.). If the search fails (meaning "avs" does not exist), it returns the integer **0**.

The second part of the expression, `> 0`, compares the result of the `char.index` function to zero. When `char.index` returns a positive integer (presence of the substring), the inequality `> 0` evaluates to **True**, and the case is selected. Conversely, when `char.index` returns **0** (absence of the substring), the inequality `0 > 0` evaluates to **False**, and the case is excluded from the analysis. This mechanism provides a precise and reliable way to filter large volumes of text data rapidly.

## Conclusion and Further Data Manipulation Techniques

The ability to conditionally select cases based on the content of a string variable is indispensable for meticulous data analysis in SPSS. By mastering the application of the `char.index` function within the **Select Cases** dialogue, analysts gain significant control over their data subsets, enabling highly targeted research outcomes and reports.

This technique is easily extensible to more complex criteria. For instance, you could combine this string selection with other numeric or categorical conditions using standard Boolean logic operators (AND, OR, NOT) to refine your selection even further. Whether you are cleaning data, segmenting customers, or preparing highly specific statistical models, textual filtering remains a cornerstone skill in statistical computation.

For users looking to expand their skills in data manipulation and case selection within SPSS, the following tutorials explain how to perform other common operations:

[How to Select Cases Based on Multiple Conditions in SPSS](#)

[How to Replace Missing Values with Zero in SPSS](#)