

How to Select SPSS Cases Containing Specific Text

Authored by
stats writer

January 23, 2026

RECOMMENDED CITATION

stats writer (2026). *How to Select SPSS Cases Containing Specific Text*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=127284>

Selecting cases in SPSS based on specific text contained within a string variable is a fundamental requirement for targeted data analysis. This filtering process allows researchers and analysts to isolate subgroups that meet highly specific textual criteria, enabling focused examination without altering the integrity of the overall dataset. Unlike numerical filtering, which relies on simple equality or range checks, filtering based on partial text requires specialized functions that can search within the variable's values. The primary tool for this operation is the powerful Select Cases function, which facilitates the creation of a temporary filter variable. This filter systematically identifies cases where the desired text is present within the specified string, thereby including only those relevant observations in subsequent statistical procedures. Mastering this technique, particularly by utilizing functions like char.index, is crucial for effective data management and advanced analytical workflows in SPSS.

The ability to filter observations based on textual content significantly enhances the flexibility of data management within SPSS. By employing the "IF" command embedded within the Select Cases dialogue, users can define complex logical conditions that specifically target text patterns. Once the criteria--the exact text contained within the string--is defined, the selected cases are marked for inclusion in the active analysis session. This method is exceptionally valuable for routine tasks such as data cleansing, subset generation, and performing comparative analyses on distinct textual categories within a larger, often complex, dataset. The following detailed guide outlines the most reliable and efficient process for executing this text-based case selection.

SPSS: Selecting Cases When a String Contains Specific Text

Introduction to Case Selection in SPSS

Data analysis often necessitates working with subsets of observations rather than the entire dataset. This segmentation is particularly true when dealing with large volumes of data where only specific groups or categories are relevant to a particular research question. The primary mechanism for achieving this necessary filtering in SPSS is the Select Cases function, found under the Data menu. While simple filtering using numerical comparisons (e.g., selecting cases where Age > 30) is straightforward, selecting cases based on complex text patterns within a string variable requires a more advanced approach.

A string variable stores text or categorical information, and often, the meaningful criteria for selection are not the entire string value, but rather a specific substring embedded within it. For example, in a column listing product codes, we might only want to analyze products whose codes contain the identifier "HQ" regardless of the surrounding characters. Identifying this partial text match is critical for accurate filtering, and fortunately, SPSS provides the necessary tools to handle

this requirement efficiently, primarily through dedicated string manipulation functions accessible within the Select Cases dialogue.

The subsequent steps will guide you through using the Select Cases dialogue in conjunction with the powerful char.index function. This combination allows the user to establish a conditional expression that evaluates to true only when the targeted substring exists within the specified variable, successfully isolating the desired cases for analysis.

Understanding the char.index Function for Substring Matching

When attempting to select cases based on partial text, standard equality operators (like `=`) are insufficient because they require an exact match of the entire cell content. To search for a substring within a larger string, we must employ a function specifically designed for pattern detection. In SPSS, this function is **char.index** (or its counterpart **char.index.r** for right-to-left searches).



The core utility of the char.index function is to search a target string (the variable) for the occurrence of a specified substring (the text we are looking for). If the substring is successfully located, the function returns the starting position (the character count) where that substring begins within the target string. Crucially, if the substring is not found anywhere within the target string for a given case, the function returns a value of **0**.

This behavior--returning a positive integer upon success and 0 upon failure--makes char.index perfectly suited for conditional filtering within the Select Cases dialogue. By simply checking if the result of the function is greater than zero, we can create a powerful Boolean condition that evaluates to **True** for all cases containing the text and **False** for all cases that do not.

Practical Example: Filtering a Dataset

To illustrate this process, consider a scenario involving a sports dataset containing information about basketball players. Suppose this dataset includes a column named **Team** which contains the full names of various teams, and we are tasked with selecting only those cases belonging to teams whose names contain the abbreviation "avs".

The initial dataset, which we will use as our starting point for this demonstration, is structured with columns for Player, Team, and Points Scored, as shown below:

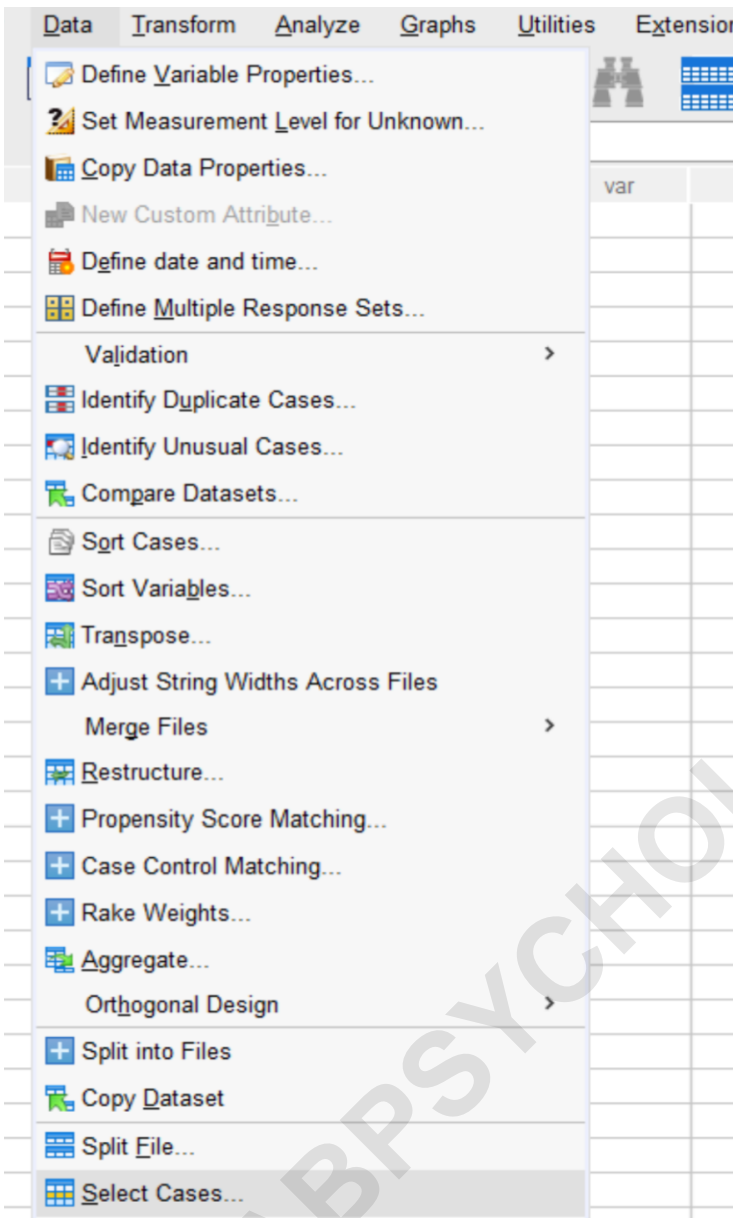
	 Team	 Points	var	var	
1	Mavs	22.00			
2	Mavs	14.00			
3	Rockets	37.00			
4	Nets	19.00			
5	Lakers	11.00			
6	Cavs	12.00			
7	Cavs	15.00			
8	Warriors	23.00			
9	Spurs	40.00			
10	Celtics	30.00			
11	Celtics	14.00			
12	Hawks	11.00			
13	Mavs	24.00			
14	Rockets	28.00			
15	Thunder	10.00			
16					
17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

Our goal is to apply a filter condition that searches the **Team** column specifically for the presence of the substring "avs". This method ensures that we capture teams like "Cavaliers" or any other team name where "avs" is embedded, irrespective of its position within the string. This focused selection process is vital for targeted statistical summaries or visualizations related only to that specific subset of teams.

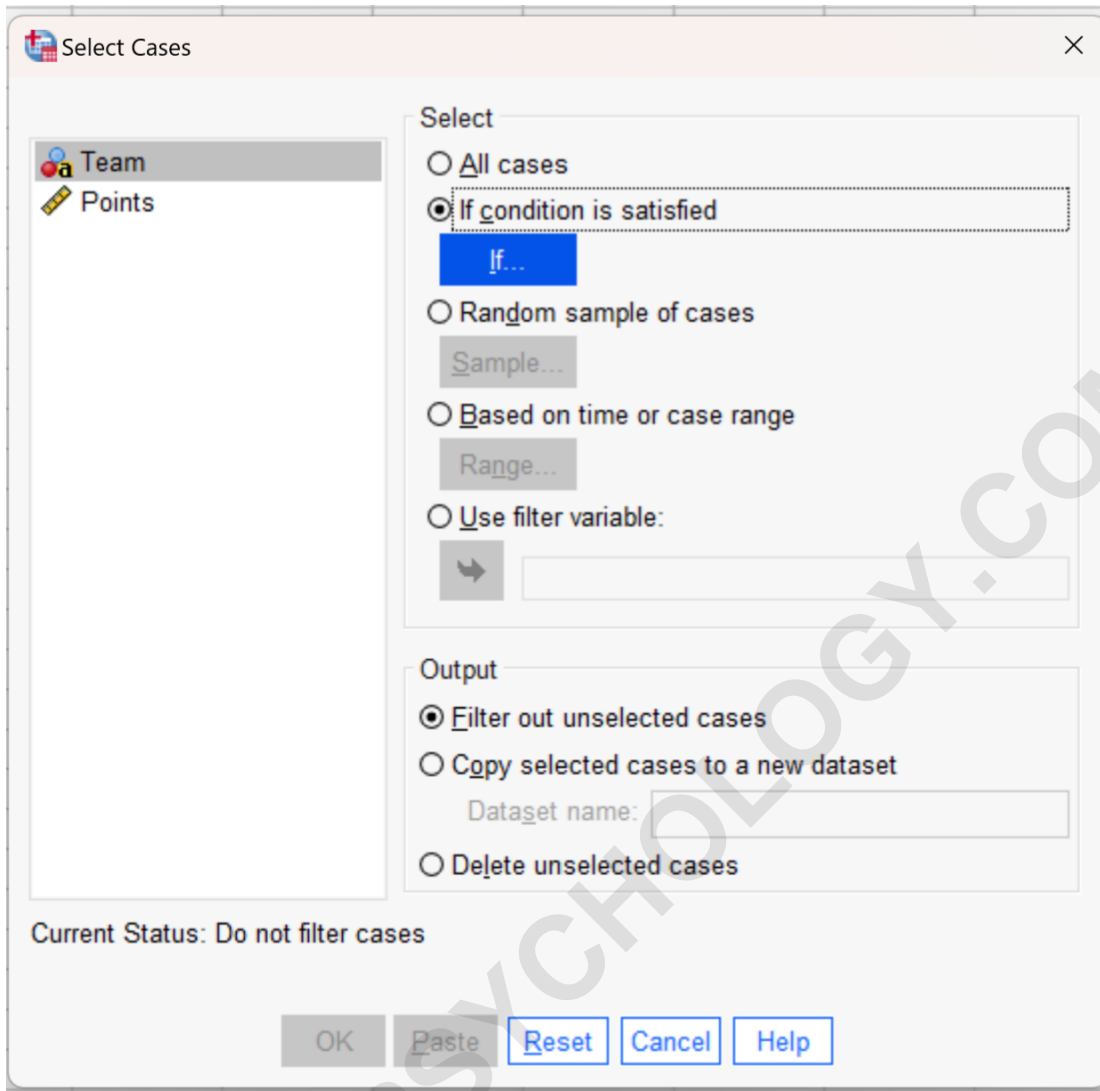
Step-by-Step Guide: Utilizing the Select Cases Dialogue

The following steps outline the exact procedure within the SPSS application to implement this text-based filtering criteria.

First, locate and click the **Data** tab in the main menu bar of SPSS, and then select the **Select Cases** option from the dropdown menu. This action opens the primary dialogue box for case filtering:



Within the Select Cases window, you must choose the option to apply a conditional filter. Click the radio button next to **If condition is satisfied**. Once this option is activated, the **If** button (located on the lower left of the dialogue box) becomes available. Click the **If** button to open the central calculator interface where the filtering formula will be defined:



This new window is where the logical expression using the `char.index` function must be input. The formula must check whether the position returned by the substring search is a positive value (indicating detection) or zero (indicating absence).

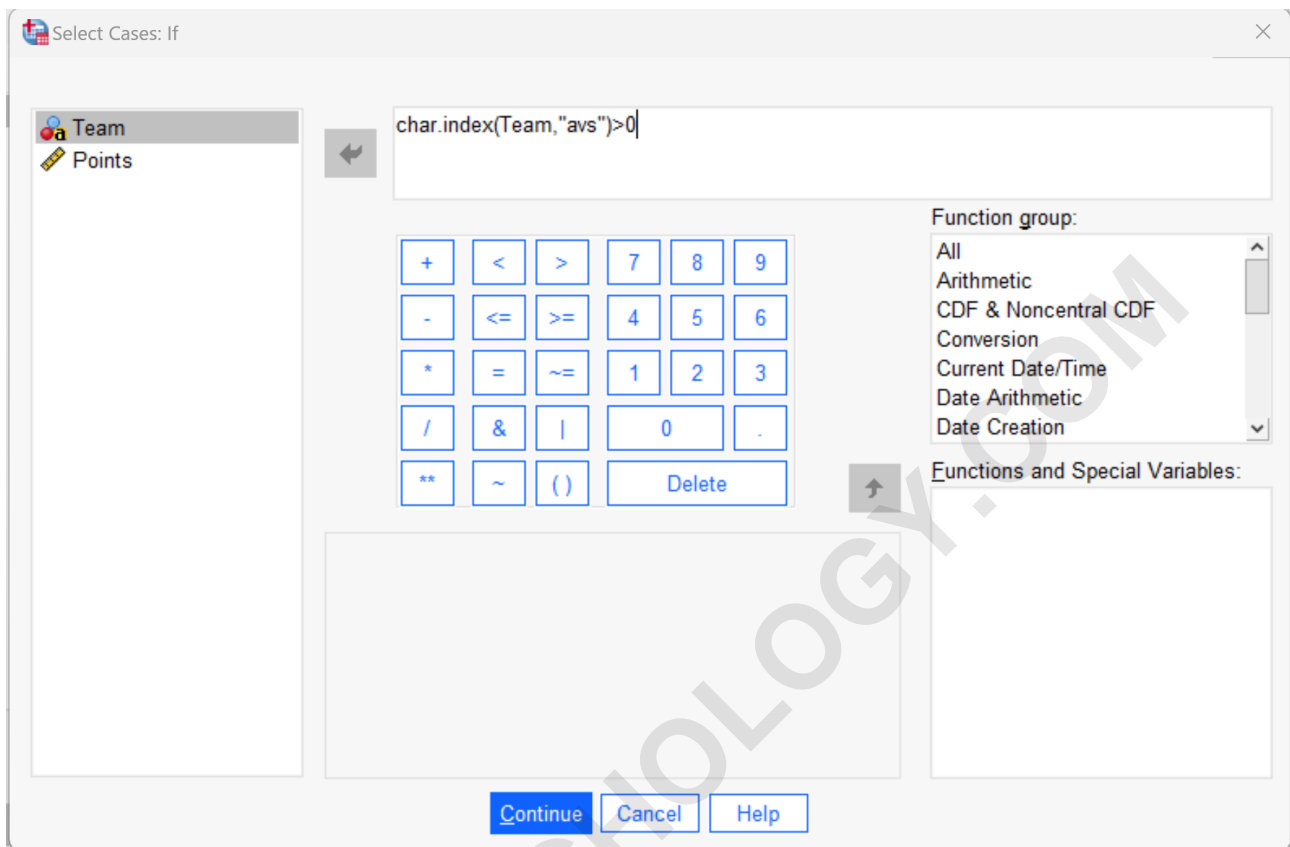
Implementing the `char.index` Formula

In the conditional expression dialogue box, input the following formula. This command tells SPSS to execute the `char.index` function, searching the **Team** variable for the substring "avs", and then to only select the case if the result is greater than zero.

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

The variable name (**Team**) must be entered exactly as it appears in the dataset, and the target text ("avs") must be enclosed in quotation marks, as required for all string literals in SPSS syntax. Once

the formula is correctly entered, the dialogue box should resemble the following visualization:






After verifying the accuracy of the formula, click **Continue** to return to the main Select Cases dialogue. Finally, click **OK** to execute the selection command. SPSS will now process the dataset, identifying and applying a filter to all cases that meet the established criteria.

Reviewing the Filtered Output

Upon successful execution of the Select Cases command, the Data View window will visually confirm the filtering operation. SPSS does not delete the unselected cases; instead, it visually marks them as filtered out by placing a diagonal strike-through across the corresponding row numbers.

In our example, only cases where the **Team** column explicitly contains the substring "avs" remain active for analysis. All other cases that do not contain this text pattern in the **Team string** will be crossed out and temporarily excluded from any subsequent statistical procedures, reporting, or charting operations.

The filtered dataset view demonstrates the success of the applied logic:

	 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				

It is important to note that a new system variable, typically named **filter_\$**, is automatically created in the SPSS Variable View. This variable holds the Boolean result (1 for selected, 0 for filtered out) of the condition defined in the Select Cases dialogue, confirming which observations are currently active.

Detailed Explanation of the Conditional Logic

Let us revisit the conditional expression used for the filtering operation to understand its precise mechanism:

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

This formula leverages the intrinsic behavior of the char.index function, which is designed for robust text searching. For every case in the dataset, SPSS evaluates this expression:

Search Execution: The char.index function attempts to find the specified substring ("avs") within the content of the **Team** string variable for that particular case.

Positive Result: If "avs" is found (e.g., in "Cavaliers"), the function returns the starting character

position, which is always a value greater than 0 (e.g., 3). Since $3 > 0$, the overall conditional statement evaluates to **True**, and the case is selected.

Negative Result: If "avs" is not found (e.g., in "Raptors"), the function returns **0**. Since the conditional check $0 > 0$ is **False**, the case is filtered out and excluded from the analysis.

By combining the positional search result with a simple greater-than inequality, we effectively transform the specialized `char.index` output into a standard Boolean criterion suitable for the `Select Cases` function, thereby facilitating highly precise filtering based on partial `string` matches.

Advanced Considerations for String Matching

While the `char.index` function is highly effective, analysts must be aware of certain advanced considerations, particularly regarding case sensitivity. By default, `SPSS` string functions are often case-sensitive. This means that if the data contains "Avs" (capital A) but the search criteria is "avs" (lowercase), the case may not be selected, and `char.index` might return 0.

To ensure robustness and avoid errors due to capitalization differences, it is best practice to first convert the target variable to a consistent case (either upper or lower) before executing the search. This is achieved by nesting the `char.index` function within a case conversion function, such as **LOWER**. The enhanced, case-insensitive syntax would look like this: `char.index(LOWER(Team), "avs") > 0`. By applying **LOWER(Team)**, `SPSS` temporarily converts all team names to lowercase before searching, guaranteeing that the search for "avs" will find matches regardless of the original capitalization (e.g., "CAVALIERS", "Cavaliers", or "cavaliers").

Furthermore, analysts should consider the implications of missing data. If the target string variable contains system-missing values for certain cases, the `char.index` function will typically treat these as non-matches, returning 0 and thus filtering them out unless specific instructions are given to handle them otherwise. Understanding these nuances allows for the creation of precise and reliable filtering criteria crucial for high-quality statistical work.

Related SPSS Operations and Further Reading

The technique of using the `Select Cases` function with `char.index` is a cornerstone of data preparation in `SPSS`. Once you have mastered selecting cases based on a single condition involving a substring, you may wish to explore more complex filtering scenarios. This includes applying multiple criteria simultaneously (e.g., Team contains "avs" AND Points > 10) or performing other common data transformation tasks.

For those interested in extending their data management skills in `SPSS`, the following resources provide guidance on related operations that complement the functionality discussed here:

The following tutorials explain how to perform other common operations in SPSS:

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

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

ARABPSYCHOLOGY.COM