

How can I use the PRXMATCH function in SAS, and what are some examples of its usage?

Authored by
stats writer

June 23, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use the PRXMATCH function in SAS, and what are some examples of its usage?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=148167>

The PRXMATCH function in SAS is used to find the position of a pattern within a given string. It returns the starting position of the first occurrence of the pattern within the string. This function is commonly used in data manipulation and cleaning tasks, as well as in text processing and analysis.

For example, if we have a dataset containing email addresses and we want to extract the domain names from each address, we can use the PRXMATCH function to find the position of the "@" symbol and then use the SUBSTR function to extract the domain name from the string.

Another example is in data validation, where we can use the PRXMATCH function to check if a given string follows a specific pattern or format, such as a phone number or social security number.

In summary, the PRXMATCH function in SAS is a versatile tool that can be used for various purposes, including data manipulation, text processing, and data validation. Its usage can greatly enhance the efficiency and accuracy of data analysis and manipulation tasks.

Use PRXMATCH Function in SAS (With Examples)

You can use the PRXMATCH function in SAS to search for a specific pattern in a string and return the position at which the pattern is found.

This function uses the following basic syntax:

PRXMATCH(regular expression, source)

where:

regular expression: Regular expression that specifies the pattern to search for
source: Name of the variable to search

The following examples show three common ways to use this function in practice with the following dataset in SAS:

```
/*create dataset*/  
data my_data;  
input team $ points;  
datalines;  
Mavs 22  
mavs 14  
Warriors 23  
Mavs 19  
warriors 34  
MAVS 40  
WARRIORS 39  
;  
run;  
  
/*view dataset*/  
proc printdata=my_data;
```

| Obs | team | points |
|-----|----------|--------|
| 1 | Mavs | 22 |
| 2 | mavs | 14 |
| 3 | Warriors | 23 |
| 4 | Mavs | 19 |
| 5 | warriors | 34 |
| 6 | MAVS | 40 |
| 7 | WARRIORS | 39 |

Example 1: Use PRXMATCH to Find Position of Pattern in String

The following code shows how to use the PRXMATCH function to create a new column called position that contains the position where the string "avs" occurs in the team column:

```
/*create new dataset*/  
data new_data;  
set my_data;  
position = prxmatch("/avs/", team);  
run;
```

```
/*view new dataset*/  
proc printdata=new_data;
```

| Obs | team | points | position |
|-----|----------|--------|----------|
| 1 | Mavs | 22 | 2 |
| 2 | mavs | 14 | 2 |
| 3 | Warriors | 23 | 0 |
| 4 | Mavs | 19 | 2 |
| 5 | warriors | 34 | 0 |
| 6 | MAVS | 40 | 0 |
| 7 | WARRIORS | 39 | 0 |

From the output we can see:

The first row contains the pattern "avs" in the team column starting in position 2 of the string. The second row contains the pattern "avs" in the team column starting position 2 of the string. The third row does not contain the pattern "avs" in the team column so a value of 0 is returned.

And so on.

Example 2: Use PRXMATCH to Check if String Contains Pattern

The following code shows how to use the PRXMATCH function to create a new column called mavs_flag that contains a value of 1 if the string in the team column contains "avs" anywhere in the string and a value of 0 if it does not:

```
/*create new dataset*/  
data new_data;  
set my_data;  
if prxmatch("/Mavs/i", team) > 0 then mavs_flag = 1;  
else mavs_flag = 0;  
run;  
  
/*view new dataset*/  
proc printdata=new_data;
```

| Obs | team | points | mavs_flag |
|-----|----------|--------|-----------|
| 1 | Mavs | 22 | 1 |
| 2 | mavs | 14 | 1 |
| 3 | Warriors | 23 | 0 |
| 4 | Mavs | 19 | 1 |
| 5 | warriors | 34 | 0 |
| 6 | MAVS | 40 | 1 |
| 7 | WARRIORS | 39 | 0 |

Note that the *i* at the end of the regular expression specifies that SAS should perform a case-insensitive search.

Simply leave out the *i* if you'd like to perform a case-sensitive search instead.

Example 3: Use PRXMATCH to Filter Dataset for Rows that Contain Pattern

The following code shows how to use the PRXMATCH function to create a new dataset that only contains rows from my_data that have the string "Mavs" (case insensitive) in the team column:

```
/*create dataset*/  
data original_doriginal_data;
```

| Obs | team | points |
|-----|------|--------|
| 1 | Mavs | 22 |
| 2 | mavs | 14 |
| 3 | Mavs | 19 |
| 4 | MAVS | 40 |

Notice that each of the rows in the new dataset contain "Mavs" in the team column.

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