How can the PRXCHANGE function be used in SAS, and what are some examples of its application?

Authored by

stats writer

June 23, 2024

The PRXCHANGE function in SAS is a powerful tool for pattern matching and string manipulation. It allows users to search and replace a specific pattern within a given string, and can be used in a variety of applications.

One example of its application is data cleaning and formatting. PRXCHANGE can be applied to a dataset to replace specific characters or patterns with desired values, making the data more consistent and suitable for analysis.

Another use case is in data validation, where PRXCHANGE can be used to identify and correct incorrect or missing data based on predefined patterns.

Additionally, the function can also be utilized in text mining and data extraction, where it can be used to extract specific information from unstructured text data based on user-defined patterns.

Overall, the PRXCHANGE function is a versatile tool that can greatly enhance data manipulation and analysis in SAS, making it a valuable asset for any programmer or analyst.

Use PRXCHANGE Function in SAS (With Examples)

You can use the PRXCHANGE function in SAS to replace a specific pattern in a string.

This function uses the following basic syntax:

PRXCHANGE(regular expression, times, source)

where:

regular expression: Regular expression that specifies the pattern to search fortimes: The number of times to replace to search for and replace the pattern (use -1 to continue to replace pattern until end of *source* is

reached)source: Name of the variable to search

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

```
/*create dataset*/
data my_data;
input phrase $char40.;
datalines;
This is a cool name
That is a cool cool zebra
Oh hey there
Oh cool it's a cool-looking dog
Well now that is COOL
;
run;

/*view dataset*/
proc printdata=my_data;
```

Obs	phrase
1	This is a cool name
2	That is a cool cool zebra
3	Oh hey there
4	Oh cool it's a cool-looking dog
5	Well now that is COOL

Example 1: Use PRXCHANGE to Replace Pattern in String with New Pattern

The following code shows how to use the PRXCHANGE function to create a new column called new_phrase that replaces each occurrence of "cool" with "fun" in the phrase column:

```
/*create new dataset*/
data new_data;
set my_data;
new_phrase = prxchange('s/cool/fun/i', -1, phrase);
run;

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

Obs	phrase	new_phrase
1	This is a cool name	This is a fun name
2	That is a cool cool zebra	That is a fun fun zebra
3	Oh hey there	Oh hey there
4	Oh cool it's a cool-looking dog	Oh fun it's a fun-looking dog
5	Well now that is COOL	Well now that is fun

Notice that each occurrence of "cool" has been replaced with "fun" instead.

Note that we used s in the regular expression to specify that we wanted to perform a substitution and we used i to specify that it should be case-insensitive.

Example 2: Use PRXCHANGE to Replace Pattern in String with Blank

The following code shows how to use the PRXCHANGE function to create a new column called new_phrase that replaces each occurrence of "cool" with a blank in the phrase column:

```
/*create new dataset*/
data new_data;
set my_data;
new_phrase = prxchange('s/cool//i', -1, phrase);
```

```
run;

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

Obs	phrase	new_phrase
1	This is a cool name	This is a name
2	That is a cool cool zebra	That is a zebra
3	Oh hey there	Oh hey there
4	Oh cool it's a cool-looking dog	Oh it's a -looking dog
5	Well now that is COOL	Well now that is

Notice that each occurrence of "cool" has been replaced with a blank.

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