

How can variables be renamed in SAS, and what are some examples of doing so?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can variables be renamed in SAS, and what are some examples of doing so?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=164909>

Variables in SAS can be renamed using the RENAME statement or the RENAME= option in a PROC or DATA step. This allows for the creation of new variable names or the modification of existing ones. An example of renaming variables using the RENAME statement is as follows:

```
DATA new_data;  
SET old_data;  
RENAME old_var = new_var;  
RUN;
```

This statement will rename the variable "old_var" to "new_var" in the new dataset "new_data". An alternative way to achieve the same result is by using the RENAME= option in a PROC or DATA step, for example:

```
DATA new_data;  
SET old_data (RENAME = (old_var = new_var));  
RUN;
```

In this case, the variable "old_var" will also be renamed to "new_var" in the new dataset "new_data". It is important to note that when renaming variables, the new name must be unique and cannot be the same as any existing variable names in the dataset. Additionally, multiple variables can be renamed using either the RENAME statement or the RENAME= option by specifying each variable and its new name separated by an equal sign.

Overall, renaming variables in SAS allows for the flexibility and customization of variable names, making it easier to understand and analyze data. This can be particularly useful when working with large datasets or when merging datasets with different variable names.

Rename Variables in SAS (With Examples)

You can use the rename function to rename one or more variables in a SAS dataset.

This function uses the following basic syntax:

```
data new_data;  
set original_data (rename=(old_name=new_name));
```

run;

The following examples show how to use this function in practice with the following dataset:

/*create dataset*/

data original_data;

input x y z;

datalines;

1 4 76

2 3 49

2 3 85

4 5 88

2 2 90

;

run;

/*view dataset*/

proc printdata=original_data;

Obs	x	y	z
1	1	4	76
2	2	3	49
3	2	3	85
4	4	5	88
5	2	2	90

Example 1: Rename One Variable

The following code shows how to rename just the x variable in the dataset:

```
/*rename one variable*/  
data new_data;  
set original_data (rename=(x=new_x));  
run;  
  
/*view new dataset*/  
proc printdata=new_data;
```

Obs	new_x	y	z
1	1	4	76
2	2	3	49
3	2	3	85
4	4	5	88
5	2	2	90

Notice that x has been renamed to new_x, but every other variable name remained the same.

Example 2: Rename Multiple Variables

The following code shows how to rename both the x and y variables in the dataset.

Note that you don't need to include commas in between the new variable names.

```
/*rename multiple variables*/  
data new_data;  
set original_data (rename=(x=new_x y=new_y));  
run;  
  
/*view new dataset*/  
proc print data=new_data;
```

Obs	new_x	new_y	z
1	1	4	76
2	2	3	49
3	2	3	85
4	4	5	88
5	2	2	90

Example 3: Add Prefix to All Variables

```
/*define prefix to append to each variable*/  
proc sql noprint;  
select cats(name, '=', '_NEW', name)  
into :list  
separated by ' '  
from dictionary.columns  
where libname = 'WORK' and memname =  
'ORIGINAL_DATA';  
quit;  
  
/*add prefix to each variable in dataset*/  
proc datasets library = work;  
modify original_data;  
rename &list;  
quit;  
  
/*view updated dataset*/  
proc print data=original_data;
```

Obs	NEW_x	NEW_y	NEW_z
1	1	4	76
2	2	3	49
3	2	3	85
4	4	5	88
5	2	2	90

Example 4: Add Suffix to All Variables

The following code shows how to add a suffix of `_NEW` to all variables in the dataset:

```
/*define suffix to append to each variable*/  
proc sqlnoprint;  
select cats(name, '=', name, '_NEW')  
into :list  
separated by ' '  
from dictionary.columns  
where libname = 'WORK' and memname =  
'ORIGINAL_DATA';  
quit;
```

```
/*add suffix to each variable in dataset*/  
proc datasetslibrary = work;  
modify original_data;  
rename &list;
```

```
quit;
```

```
/*view updated dataset*/
```

```
proc printdata=original_data;
```

Obs	x_NEW	y_NEW	z_NEW
1	1	4	76
2	2	3	49
3	2	3	85
4	4	5	88
5	2	2	90

Additional Resources

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