How can I use the %LET statement in SAS?

Authored by

stats writer

June 23, 2024

The %LET statement in SAS is used to assign a value to a macro variable. This allows for the creation of dynamic and reusable code by substituting the value of the macro variable in various parts of a program. By using the %LET statement, the programmer can easily modify the value of the macro variable in one place, instead of manually changing it in multiple locations throughout the code. This helps to increase efficiency and reduce the chances of errors. Additionally, the %LET statement can also be used to create new macro variables or update existing ones, providing flexibility in programming. Overall, the %LET statement is a useful tool for managing and manipulating macro variables in SAS programs.

Use the %LET Statement in SAS (With Examples)

You can use the %LET statement in SAS to create macro variables that can store values that you can then use later on in your program.

The following examples show two common ways to use the %LET statement in practice.

Example 1: Use %LET Statement to Store Numeric Value in SAS

Suppose we have the following dataset in SAS that contains information about various basketball players:

```
/*create dataset*/
data my_data;
input team $ points;
datalines;
A 22
A 14
```

```
A 23
B 30
B 18
B 20
C 13
C 12
C 26
;
run;

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

Obs	team	points
1	A	22
2	A	14
3	A	23
4	B	30
5	B	18
6	B	20
7	C	13
8	C	12
9	C	26

We can use the %LET statement to create a macro variable called points_cutoff that has a value of 20.

We can then reference this variable later on by using an ampersand (&) in front of it to create a new dataset that contains a new column indicating if each player scored more than 20 points or not:

```
/*assign value of 20 to macro variable*/
%let points_cutoff = 20;

/*use macro variable to create new column called good_player*/
data new_data;
set my_data;
good_player = points > &points_cutoff;
run;

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

Obs	team	points	good_player
1	A	22	1
2	A	14	0
3	A	23	1
4	B	30	1
5	B	18	0
6	B	20	0
7	C	13	0
8	C	12	0
9	C	26	1

Notice that the new column called good_player contains a value of 1 if the value in the points column is greater than 20 or a value of 0 if the value in the points column is not greater than 20.

Example 2: Use %LET Statement to Store String Value in SAS

We can also use the %LET statement in SAS to create a macro variable that stores a string value.

The following code shows how to use the %LET statement to create a variable called table_title that contains a value of "Basketball Data" which we can then reference later on to print a title for the dataset:

```
/*create dataset*/
data my_data;
```

```
input team $ points;
datalines;
A 22
A 14
A 23
B 30
B 18
B 20
C 13
C 12
C 26
;
run;

/*assign string to macro variable*/
%let table_title = "Basketball Data";

/*print dataset with title*/
proc printdata=my_data;
title &table_title;
run;
```

Basketball Data

Obs	team	points
1	A	22
2	A	14
3	A	23
4	B	30
5	B	18
6	B	20
7	C	13
8	C	12
9	C	26

The dataset contains the title that we specified in the macro variable.

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