

How can we use the Datalines statement in SAS to create a dataset using the SAS Assignment Statement (SAS)?

Authored by
stats writer

July 1, 2024

RECOMMENDED CITATION

stats writer (2024). *How can we use the Datalines statement in SAS to create a dataset using the SAS Assignment Statement (SAS)?*. PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=163996>

The Datalines statement in SAS is used to create a dataset by inputting data into the program. This statement allows the user to specify the variables and their values directly in the code, rather than importing them from an external source. By using the SAS Assignment Statement, the user can assign labels and formats to the variables, as well as manipulate the data before it is added to the dataset. This method of creating a dataset is efficient and convenient, especially for small datasets, as it eliminates the need for manually inputting each data point. Overall, the Datalines statement and SAS Assignment Statement provide an effective way to quickly and accurately create datasets in SAS.

SAS: Use Datalines Statement to Create a Dataset

You can use the datalines statement in SAS to quickly create a new dataset from scratch.

You can use the following basic syntax to do so:

```
data original_data;
```

```
input var1 $ var2;
```

```
datalines;
```

```
A 12
```

```
B 19
```

```
C 23
```

```
D 40
```

```
;
```

```
run;
```

Here's what each statement does:

data: The name of the dataset
input: The name and type of each variable in the dataset
datalines: The actual values in the dataset

Note that a dollar sign "\$" following a variable name tells SAS that the variable is a character variable.

The following examples show how to use the datalines statement in practice.

Example 1: Create Dataset with All Numeric Variables

The following code shows how to create a dataset with three numeric variables: points, assists, and rebounds:

```
/*create dataset*/  
data original_data;  
input points assists rebounds;  
datalines;  
22 8 4  
29 5 4  
31 12 8  
30 9 14  
22 7 1  
24 9 2  
18 6 4
```

```
20 5 5
```

```
25 1 4
```

```
;
```

```
run;
```

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

```
proc printdata=original_data;
```

Obs	points	assists	rebounds
1	22	8	4
2	29	5	4
3	31	12	8
4	30	9	14
5	22	7	1
6	24	9	2
7	18	6	4
8	20	5	5
9	25	1	4

The result is a dataset with three numeric variables.

Example 2: Create Dataset with Character & Numeric Variables

The following code shows how to create a dataset with both character and numeric variables:

```
/*create dataset*/
```

```
data original_data;
```

```
input team $ position $ points assists;  
datalines;  
A Guard 8 4  
A Guard 5 4  
A Forward 12 8  
A Forward 9 14  
A Forward 7 1  
B Guard 9 2  
B Guard 14 9  
B Forward 15 8  
B Forward 11 4  
;  
run;  
  
/*view dataset*/proc printdata=original_data;
```

Obs	team	position	points	assists
1	A	Guard	8	4
2	A	Guard	5	4
3	A	Forward	12	8
4	A	Forward	9	14
5	A	Forward	7	1
6	B	Guard	9	2
7	B	Guard	14	9
8	B	Forward	15	8
9	B	Forward	11	4

We can use the proc contents function to check the type of each variable:

```
proc contents data=original_data;
```

#	Variable	Type	Len
4	assists	Num	8
3	points	Num	8
2	position	Char	8
1	team	Char	8

From the output we can see that team and position are character variables while points and assists are numeric variables.

Additional Resources

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