

How can I use the SET statement with multiple datasets in SAS to implement the SAS programming concept of “Single Assignment (SAS)”?

Authored by
stats writer

June 26, 2024

RECOMMENDED CITATION

stats writer (2024). *How can I use the SET statement with multiple datasets in SAS to implement the SAS programming concept of “Single Assignment (SAS)”?*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=153275>

The SET statement in SAS allows for the merging of multiple datasets into one, implementing the concept of "Single Assignment." This concept ensures that each observation in the output dataset is assigned only once, even if it appears in multiple input datasets. By using the SET statement with multiple datasets, SAS efficiently combines data and avoids duplication, improving data accuracy and reducing processing time. This feature is particularly useful when working with large datasets and performing complex data manipulations. Overall, utilizing the SET statement with multiple datasets in SAS is a powerful tool for implementing the programming concept of "Single Assignment" and improving data management in SAS.

SAS: Use SET Statement with Multiple Datasets

You can use the following basic syntax to include multiple datasets in the set statement in SAS:

```
data new_data;  
set data1 data2 data3;  
run;
```

The following example shows how to use this syntax in practice.

Example: Use SET Statement with Multiple Datasets in SAS

Suppose we have the following dataset in SAS that shows the points scored by various basketball players on a team called A:

```
/*create first dataset*/  
data data1;
```

```
input team $ points;
```

```
datalines;
```

```
A 12
```

```
A 15
```

```
A 16
```

```
A 21
```

```
A 22
```

```
;
```

```
run;
```

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

```
proc printdata=data1;
```

Obs	team	points
1	A	12
2	A	15
3	A	16
4	A	21
5	A	22

And suppose we have another dataset that shows the points scored by various basketball players on a team called B:

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

```
data data2;  
input team $ points;  
datalines;  
B 16  
B 22  
B 25  
B 29  
B 30  
;  
run;  
  
/*view dataset*/  
proc printdata=data2;
```

Obs	team	points
1	B	16
2	B	22
3	B	25
4	B	29
5	B	30

We can use the set statement with multiple datasets to combine these two datasets into one:

```
/*create new dataset that combines two datasets*/
```

```
data data3;  
set data1 data2;  
run;  
  
/*view new dataset*/  
proc printdata=data3;
```

Obs	team	points
1	A	12
2	A	15
3	A	16
4	A	21
5	A	22
6	B	16
7	B	22
8	B	25
9	B	29
10	B	30

The result is a third dataset called data3 that combines the rows from data1 and data2.

Note: Even if the two datasets didn't share the same column names, the set statement would still combine the datasets into one and simply leave empty spaces in the cells where the columns don't match.

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

ARABPSYCHOLOGY.COM