

How to Easily Subset Data in SAS: A Step-by-Step Guide

Authored by
stats writer

December 1, 2025

RECOMMENDED CITATION

stats writer (2025). *How to Easily Subset Data in SAS: A Step-by-Step Guide*.

PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=103103>

Subsetting data in SAS is the process of extracting a subset of data from a larger dataset. This can be done using a variety of methods, such as the where clause, subsetting if statements, and the data step. Each method has its own advantages, depending on the data and the desired outcome. For example, the where clause is a simple and efficient way to subset data, while subsetting if statements are more flexible and allow for more complex subsetting. The data step is also useful for creating new datasets from the existing data.

Here are the three most common ways to subset a dataset in SAS:

Method 1: Choose Which Columns to Keep

```
data new_data;  
set original_data;  
keep var1 var3;  
run;
```

Method 2: Choose Which Columns to Drop

```
data new_data;  
set original_data;  
drop var4;  
run;
```

Method 3: Choose Which Rows to Keep Based on Condition

```
data new_data;  
set original_data;  
if var1 < 25 then delete;  
run;
```

The following examples show how to use each method with the following dataset in SAS:

```
/*create dataset*/  
data original_data;  
input team $ points rebounds;  
datalines;  
Warriors 25 8  
Wizards 18 12  
Rockets 22 6
```

```
Celtics 24 11
Thunder 27 14
Spurs 33 19
Nets 31 20
;
run;
```

```
/*view dataset*/
proc print data=original_data;
```

Obs	team	points	rebounds
1	Warriors	25	8
2	Wizards	18	12
3	Rockets	22	6
4	Celtics	24	11
5	Thunder	27	14
6	Spurs	33	19
7	Nets	31	20

Example 1: Choose Which Columns to Keep

The following code shows how to subset a dataset by using the **KEEP** statement to keep only certain columns:

```
/*create new dataset*/
data new_data;
set original_data;
keep team points;
run;
```

```
/*view new dataset*/
proc print data=new_data;
```

Obs	team	points
1	Warriors	25
2	Wizards	18
3	Rockets	22
4	Celtics	24
5	Thunder	27
6	Spurs	33
7	Nets	31

Example 2: Choose Which Columns to Drop

The following code shows how to subset a dataset by using the **DROP** statement to drop specific columns:

```
/*create new dataset*/  
data new_data;  
set original_data;  
drop points;  
run;  
  
/*view new dataset*/  
proc print data=new_data;
```

Obs	team	rebounds
1	Warriors	8
2	Wizards	12
3	Rockets	6
4	Celtics	11
5	Thunder	14
6	Spurs	19
7	Nets	20

Example 3: Choose Which Rows to Keep Based on Condition

The following code shows how to subset a dataset by using the **DELETE** statement to drop

specific rows from the dataset where the value in the **points** column is less than 25:

```
/*create new dataset*/  
data new_data;  
set original_data;  
if points < 25 then delete;  
run;
```

```
/*view new dataset*/  
proc print data=new_data;
```

Obs	team	points	rebounds
1	Warriors	25	8
2	Thunder	27	14
3	Spurs	33	19
4	Nets	31	20

You can also use the **OR** "|" operator to drop the rows where **points** is less than 25 *or* **rebounds** is less than 10:

```
/*create new dataset*/  
data new_data;  
set original_data;  
if points < 25 | rebounds < 10 then delete;  
run;
```

```
/*view new dataset*/  
proc print data=new_data;
```

Obs	team	points	rebounds
1	Thunder	27	14
2	Spurs	33	19
3	Nets	31	20

You can also use the **AND** "&" operator to drop the rows where **points** is less than 25 and **rebounds** is less than 10:

```
/*create new dataset*/  
data new_data;  
set original_data;  
if points < 25 & rebounds < 10 then delete;  
run;  
  
/*view new dataset*/  
proc print data=new_data;
```

Obs	team	points	rebounds
1	Warriors	25	8
2	Wizards	18	12
3	Celtics	24	11
4	Thunder	27	14
5	Spurs	33	19
6	Nets	31	20

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