“How can I use the ‘Subset Data’ function in SAS? Can you provide some examples of how to apply this function?”

Authored by

stats writer

July 1, 2024

The 'Subset Data' function in SAS allows users to filter and extract specific data from a larger dataset based on certain criteria. This function is especially useful when working with large amounts of data, as it allows for easier manipulation and analysis of relevant information. To use this function, users can specify the variables and conditions they want to include or exclude from their data. For example, a user can subset data to only include records for a specific time period or for a particular group of individuals. This function can also be used to create new datasets for further analysis. Overall, the 'Subset Data' function in SAS provides a convenient and efficient way to manage and extract data for various purposes.

Subset Data in SAS (3 Examples)

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 < 25then 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 printdata=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 printdata=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 printdata=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 < 25then delete;
run;

/*view new dataset*/proc printdata=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 < 10then delete;
run;

/*view new dataset*/proc printdata=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 < 10then delete;
run;

/*view new dataset*/proc printdata=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

Additional Resources

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