How can PROC RANK be used in SAS? Can you provide examples?

Authored by

stats writer

July 1, 2024

PROC RANK is a SAS procedure that can be used to rank data in ascending or descending order. This procedure is useful for identifying the highest or lowest values within a dataset, as well as for creating groups based on the relative position of values. This can be particularly useful for identifying outliers or detecting patterns within the data. For example, PROC RANK can be used to rank students based on their test scores, with the highest score being ranked as 1 and the lowest score being ranked as n (the number of students). Another example could be ranking countries based on their GDP, with the country with the highest GDP being ranked as 1 and the country with the lowest GDP being ranked as n. Overall, PROC RANK is a valuable tool for organizing and analyzing data in SAS.

Use PROC RANK in SAS (With Examples)

You can use PROC RANK in SAS to calculate the rank for one or more numeric variables.

Here are the four most common ways to use this procedure:

Method 1: Rank One Variable

```
proc rankdata=original_data out=ranked_data;
var var1;
ranks var1_rank;
run;
```

Method 2: Rank One Variable by Group

```
proc rankdata=original_data out=ranked_data;
var var1;
```

```
by var2;
ranks var1_rank;
run;
```

Method 3: Rank One Variable into Percentiles

```
proc rankdata=original_data groups=4
out=ranked_data;
var var1;
ranks var1_rank;
run;
```

Method 4: Rank Multiple Variables

```
proc rankdata=original_data out=ranked_data;
var var1 var2;
ranks var1_rank var2_rank;
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;
A 25 10
A 18 4
A 18 7
A 24 8
B 27 9
B 33 13
B 31 11
B 30 16
;
run;

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

Obs	team	points	rebounds
1	A	25	10
2	A	18	4
3	A	18	7
4	A	24	8
5	B	27	9
6	B	33	13
7	B	31	11
8	B	30	16

Example 1: Rank One Variable

The following code shows how to create a new variable called points_rank that ranks the points scored by each team:

```
/*rank points scored by team*/
proc rankdata=original_data out=ranked_data;
var points;
ranks points_rank;
run;

/*view ranks*/proc printdata=ranked_data;
```

Obs	team	points	rebounds	points_rank
1	A	25	10	4.0
2	A	18	4	1.5
3	A	18	7	1.5
4	A	24	8	3.0
5	B	27	9	5.0
6	B	33	13	8.0
7	B	31	11	7.0
8	B	30	16	6.0

Any ties in points scored are assigned a mean rank. For example, the rows with the first and second lowest points scored both receive a rank of 1.5, since this is

the average of 1 and 2.

Note that you can instead use the descending statement to assign the team with the most points the *lowest* rank:

```
/*rank points scored by team in descending order*/
proc rankdata=original_data descending
out=ranked_data;
var points;
ranks points_rank;
run;

/*view ranks*/proc printdata=ranked_data;
```

Obs	team	points	rebounds	points_rank
1	A	25	10	5.0
2	A	18	4	7.5
3	A	18	7	7.5
4	A	24	8	6.0
5	B	27	9	4.0
6	B	33	13	1.0
7	B	31	11	2.0
8	B	30	16	3.0

Example 2: Rank One Variable by Group

The following code shows how to create a new variable

called points_rank that ranks the points scored, grouped by team:

```
/*rank points scored, grouped by team*/
proc rankdata=original_data out=ranked_data;
var points;
by team;
ranks points_rank;
run;

/*view ranks*/proc printdata=ranked_data;
```

Obs	team	points	rebounds	points_rank
1	A	25	10	4.0
2	A	18	4	1.5
3	A	18	7	1.5
4	A	24	8	3.0
5	B	27	9	1.0
6	B	33	13	4.0
7	B	31	11	3.0
8	B	30	16	2.0

Example 3: Rank One Variable into Percentiles

We can use the groups statement to rank variables into percentile groups. For example, we can rank each value of points into a quartile (four groups):

```
/*rank points into quartiles*/
proc rankdata=original_data groups=4
out=ranked_data;
var points;
ranks points_rank;
run;

/*view ranks*/proc printdata=ranked_data;
```

Obs	team	points	rebounds	points_rank
1	A	25	10	1
2	A	18	4	0
3	A	18	7	0
4	A	24	8	1
5	B	27	9	2
6	B	33	13	3
7	B	31	11	3
8	B	30	16	2

The rows with the points values in the lowest quartile are assigned a group of 0, the rows with the points in the next lowest quartile are assigned a group of 1, and so on.

Note: To assign values into deciles instead, simply use groups=10.

Example 4: Rank Multiple Variables

The following code shows how to create multiple new variables to rank both points and rebounds:

```
proc rankdata=original_data out=ranked_data;
var points rebounds;
ranks points_rank rebounds_rank;
run;
```

Obs	team	points	rebounds	points_rank	rebounds_rank
1	A	25	10	4.0	5
2	A	18	4	1.5	1
3	A	18	7	1.5	2
4	A	24	8	3.0	3
5	B	27	9	5.0	4
6	B	33	13	8.0	7
7	B	31	11	7.0	6
8	B	30	16	6.0	8

Additional Resources

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