Chapter 29

The RANK Procedure

ABSTRACT 493
INTRODUCTION 493
SPECIFICATIONS 494
  PROC RANK Statement 494
  BY Statement 497
  RANKS Statement 497
  VAR Statement 498
DETAILS 498
  Missing Values 498
  Output Data Set 498
  Nonparametric Statistics 498
EXAMPLES 499
  Example 1: Ranking of Gain Values 499
  Example 2: Ranking of Swim Scores 500
REFERENCES 502

ABSTRACT

By default, the RANK procedure computes ranks for one or more numeric variables across the observations of a SAS data set. The ranks are output to a new SAS data set. Alternatively, PROC RANK produces normal scores or other rank scores.

INTRODUCTION

The RANK procedure ranks values from smallest to largest, assigning the rank 1 to the smallest number, 2 to the next largest, and so on up to rank n, the number of nonmissing observations. Tied values are given averaged ranks. Several options are available to request other ranking and tie-handling rules.

Many nonparametric statistical methods use ranks rather than the original values of a variable. For example, a set of data can be passed through PROC RANK to obtain the ranks for a response variable that could then be fit to an analysis-of-variance model using the ANOVA or GLM procedures.

Ranks are also useful for investigating the distribution of values for a variable. The ranks divided by n or n+1 form values in the range 0 to 1, and these values estimate the cumulative distribution function. Inverse cumulative distribution functions can be applied to these fractional ranks to obtain probability quantile scores, which can be compared to the original values to judge the fit to the distribution. For example, if a set of data has a normal distribution, the normal scores should be a linear function of the original values, and a plot of scores versus original values should be a straight line.
PROC RANK is also useful for grouping continuous data. The GROUPS= option can break a population into approximate groups.

**SPECIFICATIONS**

The RANK procedure is controlled by the following statements:

```
PROC RANK <option-list>;
   BY variable-list;
   RANKS new-variable-list;
   VAR variable-list;
```

**PROC RANK Statement**

```
PROC RANK <option-list>;
```

The options listed in Table 29.1 can appear in the PROC RANK statement. They are described in alphabetic order following the table.

<table>
<thead>
<tr>
<th>Task</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify data sets</td>
<td>DATA=</td>
</tr>
<tr>
<td></td>
<td>OUT=</td>
</tr>
<tr>
<td>Select ranking method</td>
<td>FRACTION</td>
</tr>
<tr>
<td></td>
<td>GROUPS=</td>
</tr>
<tr>
<td></td>
<td>NORMAL=</td>
</tr>
<tr>
<td></td>
<td>NPLUS1</td>
</tr>
<tr>
<td></td>
<td>PERCENT</td>
</tr>
<tr>
<td></td>
<td>SAVAGE</td>
</tr>
<tr>
<td>Reverse ranking</td>
<td>DESCENDING</td>
</tr>
<tr>
<td>Report tied values</td>
<td>TIES=</td>
</tr>
</tbody>
</table>

The PROC RANK statement starts the procedure. You may specify only one ranking method (FRACTION, GROUPS=, NORMAL=, NPLUS1, PERCENT, or SAVAGE) in a given execution of PROC RANK. If you request more than one of these options on the same PROC RANK statement, you will get the following message:

**ERROR:** More than one method has been specified. No ranks will be computed.

You can specify the following options in the PROC RANK statement:

**DATA=SAS-data-set**

names the SAS data set to be used by PROC RANK. If the DATA= option is omitted, the most recently created SAS data set is used.
DESCENDING
reverses the ranking to be from largest to smallest. The largest value is
given a rank of 1, the next smallest a rank of 2, and so on. When the
DESCENDING option is omitted, values are ranked from smallest to
largest.

FRACTION
F
requests fractional ranks. The RANK procedure divides each rank by the
number of observations having nonmissing values of the ranking variable
and expresses the ranks as fractions. If you use the FRACTION option
and have tied values, TIES=HIGH is the default. If the TIES= option is
omitted or if TIES=HIGH is specified, these fractional ranks can be
considered values of a right-continuous empirical cumulative distribution
function.

GROUPS=n
requests grouping scores, where n is the number of groups. The
grouping scores are the integers 0 to (n−1). In the absence of ties, the
groups have equal or nearly equal numbers of observations. The lowest
values are in the first group; the highest values are in the last group. The
following are common specifications:
GROUPS=100 produces percentile ranks.
GROUPS=10 produces deciles.
GROUPS=4 produces quartiles.

For example, if you specify GROUPS=4, then PROC RANK separates
the values of the ranking variable into four groups according to size. The
values in the group containing the smallest values receive a quartile
value of 0, the values in the next group receive a value of 1, the values
in the next group receive a value of 2, and the largest values receive the
value 3.

The formula used to calculate the quantile rank of a value is

\[ FLOOR \left( \frac{rank \times k}{n + 1} \right) \]

where FLOOR is the floor function, rank is the value's rank, k is the
number of groups specified with the GROUPS= option, and n is the
number of observations having nonmissing values of the ranking
variable. Use the GROUP= option only once with a single PROC RANK
statement.

NORMAL=BLOM | TUKEY | VW
requests normal scores to be computed from the ranks. The resulting
variables appear normally distributed. The formulas are as follows:

BLOM \[ y_i = \Phi^{-1}(r_i - 3/8) / (n + 1/4) \]
TUKEY \[ y_i = \Phi^{-1}(r_i - 1/3) / (n + 1/3) \]
VW \[ y_i = \Phi^{-1}(r_i) / (n + 1) \]

where \( \Phi^{-1} \) is the inverse cumulative normal (PROBIT) function, \( r_i \) is the
rank, of the \( i \)th observation, and \( n \) is the number of nonmissing
observations for the ranking variable. VW stands for van der Waerden,
whose scores are used for a nonparametric location test.
These three normal scores are approximations to the exact expected order statistics for the normal distribution, also called normal scores. The BLOM version appears to fit slightly better than the others (Blom 1958; Tukey 1962).

NPLUS1
FN1
N1
requests fractional ranks where the denominator is $n+1$ rather than $n$ (as in the FRACTION option), where $n$ is the number of observations having nonmissing values of the ranking variable.

OUT=SAS-data-set
names the output data set that will contain the resulting ranks. If you do not specify the OUT= option, the data set is named using the DATA$n$ naming convention. If you want to create a permanent SAS data set, you must specify a two-level name (see Chapter 6, “SAS Files,” in SAS Language: Reference, Version 6, First Edition for more information on permanent SAS data sets). For details on the data set created by PROC RANK, see Output Data Set later in this chapter.

PERCENT
P
asks the RANK procedure to divide each rank by the number of observations having nonmissing values of the ranking variable and then to multiply the result by 100 to get a percentage. Like the FRACTION option, the PERCENT option implies TIES=HIGH unless another TIES= value is specified.

Note: The PERCENT option does not give what are usually called percentile ranks. These are produced by specifying GROUPS=100.

SAVAGE
requests Savage (or exponential) scores be computed from the ranks. The scores are computed by this formula (Lehman 1975):

$$y_i = \left( \sum_{r_i=n-r_i+1}^{\infty} (1/j) \right) - 1 .$$

TIES=MEAN | HIGH | LOW
specifies which rank to report for tied values. TIES=MEAN requests that tied values receive the mean of the corresponding ranks (midranks). The specification TIES=HIGH requests that the largest of the corresponding ranks be used. TIES=LOW requests that the smallest of the corresponding ranks be used. The default is TIES=MEAN.
Table 29.2 illustrates the three options available for handling tied values. Consider the values of the variable WEIGHT and the ranks that are assigned for each TIES value.

<table>
<thead>
<tr>
<th>WEIGHT</th>
<th>TIES=MEAN</th>
<th>TIES=HIGH</th>
<th>TIES=LOW</th>
</tr>
</thead>
<tbody>
<tr>
<td>107</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>110</td>
<td>2.5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>110</td>
<td>2.5</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>121</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>125</td>
<td>6</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>125</td>
<td>6</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>125</td>
<td>6</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>132</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

**BY Statement**

**BY variable-list;**

A BY statement can be used with PROC RANK to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables or to have an appropriate index. If your input data set is not sorted in ascending order, you can do one of the following:

- Use the SORT procedure with a similar BY statement to sort the data.
- If appropriate, use the BY statement options NOTSORTED or DESCENDING.
- Use the DATASETS procedure to create an index on the BY variables you want to use. For more information on using the BY statement with indexed data sets, see Chapter 17, “The DATASETS Procedure.”

**RANKS Statement**

**RANKS new-variable-list;**

If you want the original variables included in the output data set in addition to the ranks, use the RANKS statement to assign variable names to the ranks. First, name the rank corresponding to the first variable in the VAR statement, next name the rank corresponding to the second variable in the VAR statement, and so on. If the RANKS statement is omitted, the rank values replace the original variable values in the output data set. See Example 1 for an example using the RANKS statement and Example 2 for an example without a RANKS statement.
VAR Statement

\textit{VAR variable-list;}

The RANK procedure computes ranks for the variables given in the VAR statement. These variables must be numeric. If the VAR statement is omitted, ranks are computed for all numeric variables in the data set. The VAR statement must be included if a RANKS statement is used.

DETAILS

Missing Values

Missing values are not ranked and are left missing when ranks or rank scores replace the other values of the ranking variable.

Output Data Set

The RANK procedure creates a new SAS data set containing the ranks or rank scores but does not create any printed output.

The new output data set contains all the variables from the input data set plus the variables named in the RANKS statement, if one is specified. If a RANKS statement is used, a VAR statement must also be included. If you do not use the RANKS statement, the procedure stores the ranks in the output data set using the names of the original variables, changing the values to rankings. See Example 2 for an illustration. If no VAR statement is included, the procedure ranks all numeric variables.

Nonparametric Statistics

Many nonparametric methods are based on taking the ranks of a variable and analyzing these ranks instead of the original values:

- A two-sample \( t \)-test applied to the ranks is equivalent to a Wilcoxon rank sum test using the \( t \) approximation for the significance level. If the \( t \)-test is applied to the normal scores rather than to the ranks, the test becomes equivalent to the van der Waerden test. If the \( t \)-test is applied to median scores (GROUPS=2), the test becomes the median test.
- A one-way analysis of variance applied to ranks is equivalent to the Kruskal-Wallis \( k \)-sample test; the \( F \) test generated by the parametric procedure applied to the ranks is often better than the \( \chi^2 \) approximation used by Kruskal-Wallis. This test can be extended to other rank scores (Quade 1966).
- Friedman's two-way analysis for block designs can be obtained by ranking within blocks (using a BY statement with PROC RANK) and then performing a main-effects analysis of variance on these ranks (Conover 1980).
- Regression relationships can be investigated using rank transformations with a method described by Iman and Conover (1979).