Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore
for additional books and resources.
Overview: SCORE Procedure

The SCORE procedure multiplies values from two SAS data sets, one containing coefficients (for example, factor-scoring coefficients or regression coefficients) and the other containing raw data to be scored using the coefficients from the first data set. The result of this multiplication is a SAS data set containing linear combinations of the coefficients and the raw data values.

Many statistical procedures output coefficients that PROC SCORE can apply to raw data to produce scores. The new score variable is formed as a linear combination of raw data and scoring coefficients. For each observation in the raw data set, PROC SCORE multiplies the value of a variable in the raw data set by the matching scoring coefficient from the data set of scoring coefficients. This multiplication process is repeated for each variable in the VAR statement. The resulting products are then summed to produce the value of the new score variable. This entire process is repeated for each observation in the raw data set. In other words, PROC SCORE cross multiplies part of one data set with another.
Raw Data Set

The raw data set can contain the original data used to calculate the scoring coefficients, or it can contain an entirely different data set. The raw data set must contain all the variables needed to produce scores. In addition, the scoring coefficients and the variables in the raw data set that are used in scoring must have the same names. See the section “Getting Started: SCORE Procedure” on page 7295 for more information.

Scoring Coefficients Data Set

The data set containing scoring coefficients must contain two special variables: the _TYPE_ variable and the _NAME_ or _MODEL_ variable.

- The _TYPE_ variable identifies the observations that contain scoring coefficients.
- The _NAME_ or _MODEL_ variable provides a SAS name for the new score variable.

PROC SCORE first looks for a _NAME_ variable in the SCORE= input data set. If there is such a variable, the variable’s value is what SCORE uses to name the new score variable. If the SCORE= data set does not have a _NAME_ variable, then PROC SCORE looks for a _MODEL_ variable.

For example, PROC FACTOR produces an output data set that contains factor-scoring coefficients. In this output data set, the scoring coefficients are identified by _TYPE_='SCORE’. For _TYPE_='SCORE’, the _NAME_ variable has values of ’Factor1’, ’Factor2’, and so forth. PROC SCORE gives the new score variables the names Factor1, Factor2, and so forth.

As another example, the REG procedure produces an output data set that contains parameter estimates. In this output data set, the parameter estimates are identified by _TYPE_='PARMS’. The _MODEL_ variable contains the label used in the MODEL statement in PROC REG, or it uses MODELn if no label is specified. This label is the name PROC SCORE gives to the new score variable.

Standardization of Raw Data

PROC SCORE automatically standardizes or centers the DATA= variables for you, based on information from the original variables and analysis from the SCORE= data set.

If the SCORE= scoring coefficients data set contains observations with _TYPE_ =’MEAN’ and _TYPE_ =’STD’, then PROC SCORE standardizes the raw data before scoring. For example, this type of SCORE= data set can come from PROC PRINCOMP without the COV option.

If the SCORE= scoring coefficients data set contains observations with _TYPE_ =’MEAN’ but _TYPE_ =’STD’ is absent, then PROC SCORE centers the raw data (the means are subtracted) before scoring. For example, this type of SCORE= data set can come from PROC PRINCOMP with the COV option.

If the SCORE= scoring coefficients data set does not contain observations with _TYPE_ =’MEAN’ and _TYPE_ =’STD’, or if you use the NOSTD option, then PROC SCORE does not center or standardize the raw data.
If the SCORE= scoring coefficients are obtained from observations with _TYPE_='USCORE', then PROC SCORE “standardizes” the raw data by using the uncorrected standard deviations identified by _TYPE_='USTD', and the means are not subtracted from the raw data. For example, this type of SCORE= data set can come from PROC PRINCOMP with the NOINT option. For more information about _TYPE_='USCORE’ scoring coefficients in TYPE=UCORR or TYPE=UCOV output data sets, see Appendix A, “Special SAS Data Sets.”

You can use PROC SCORE to score the data that were also used to generate the scoring coefficients, although more typically, scoring results are directly obtained from the OUT= data set in a procedure that computes scoring coefficients. When scoring new data, it is important to realize that PROC SCORE assumes that the new data have approximately the same scales as the original data. For example, if you specify the COV option with PROC PRINCOMP for the original analysis, the scoring coefficients in the PROC PRINCOMP OUTSTAT= data set are not appropriate for standardized data. With the COV option, PROC PRINCOMP will not output _TYPE_='STD' observations to the OUTSTAT= data set, and PROC SCORE will only subtract the means of the original (not new) variables from the new variables before multiplying. Without the COV option in PROC PRINCOMP, both the original variable means and standard deviations will be in the OUTSTAT= data set, and PROC SCORE will subtract the original variable means from the new variables and divide them by the original variable standard deviations before multiplying.

In general, procedures that output scoring coefficients in their OUTSTAT= data sets provide the necessary information for PROC SCORE to determine the appropriate standardization. However, if you use PROC SCORE with a scoring coefficients data set that you constructed without _TYPE_='MEAN’ and _TYPE_='STD’ observations, you might have to do the relevant centering or standardization of the new data first. If you do this, you must use the means and standard deviations of the original variables—that is, the variables that were used to generate the coefficients—not the means and standard deviations of the variables to be scored.

See the section “Getting Started: SCORE Procedure” on page 7295 for further illustration.
data Schools;
  input Type $ English Math Biology @@;
datalines;
p 52 55 45 p 42 49 40 p 63 64 54
p 47 50 51 p 64 69 47 p 63 67 54
p 59 63 42 p 56 61 41 p 41 44 72
p 39 42 45 p 56 63 44 p 63 73 42

... more lines ...

r 50 47 49 r 55 48 46 r 38 36 51;

The data set Schools contains the character variable Type, which represents the type of school. Valid values are p (private schools), r (state-run rural schools), and u (state-run urban schools).

The three numeric variables in the data set are English, Math, and Biology, which represent the student scores for English, mathematics, and biology, respectively. The double trailing at sign (@@) in the INPUT statement specifies that observations are input from each line until all values are read.

The following statements invoke the FACTOR procedure to compute the data set of factor scoring coefficients. The statements perform a principal components factor analysis that uses all three numeric variables in the SAS data set Schools. The OUTSTAT= option requests that PROC FACTOR output the factor scores to the data set Scores. The NOPRINT option suppresses display of the output.

    proc factor data=Schools score outstat=Scores noprint;
      var english math biology;
    run;

    proc score data=schools score=Scores out=New;
      var english math biology;
      id type;
    run;

The SCORE procedure is then invoked using Schools as the raw data set to be scored and Scores as the scoring data set. The OUT= option creates the SAS data set New to contain the linear combinations.

The VAR statement specifies that the variables English, Math, and Biology are used in computing scores. The ID statement copies the variable Type from the Schools data set to the output data set New.

The following statements print the SAS output data set Scores, the first two observations from the original data set Schools, and the first two observations of the resulting data set New.

    title 'OUTSTAT= Data Set from PROC FACTOR';
    proc print data=Scores;
    run;

    title 'First Two Observations of the DATA= Data Set from PROC SCORE';
    proc print data=Schools(obs=2);
    run;

    title 'First Two Observations of the OUT= Data Set from PROC SCORE';
    proc print data=New(obs=2);
    run;
Figure 88.1 displays the output data set Scores produced by the FACTOR procedure. The last observation (number 11) contains the scoring coefficients (_TYPE_= ’SCORE’). Only one factor has been retained.

```
Figure 88.1  Listing of the Data Set Created by PROC FACTOR

OUTSTAT= Data Set from PROC FACTOR

<table>
<thead>
<tr>
<th>Obs</th>
<th><em>TYPE</em></th>
<th><em>NAME</em></th>
<th>English</th>
<th>Math</th>
<th>Biology</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MEAN</td>
<td></td>
<td>55.525</td>
<td>52.325</td>
<td>50.350</td>
</tr>
<tr>
<td>2</td>
<td>STD</td>
<td></td>
<td>12.949</td>
<td>12.356</td>
<td>12.239</td>
</tr>
<tr>
<td>3</td>
<td>N</td>
<td></td>
<td>120.000</td>
<td>120.000</td>
<td>120.000</td>
</tr>
<tr>
<td>4</td>
<td>CORR</td>
<td>English</td>
<td>1.000</td>
<td>0.833</td>
<td>0.672</td>
</tr>
<tr>
<td>5</td>
<td>CORR</td>
<td>Math</td>
<td>0.833</td>
<td>1.000</td>
<td>0.594</td>
</tr>
<tr>
<td>6</td>
<td>CORR</td>
<td>Biology</td>
<td>0.672</td>
<td>0.594</td>
<td>1.000</td>
</tr>
<tr>
<td>7</td>
<td>COMMUNAL</td>
<td></td>
<td>0.881</td>
<td>0.827</td>
<td>0.696</td>
</tr>
<tr>
<td>8</td>
<td>PRIORS</td>
<td></td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>9</td>
<td>EIGENVAL</td>
<td></td>
<td>2.405</td>
<td>0.437</td>
<td>0.159</td>
</tr>
<tr>
<td>10</td>
<td>PATTERN</td>
<td>Factor1</td>
<td>0.939</td>
<td>0.910</td>
<td>0.834</td>
</tr>
<tr>
<td>11</td>
<td>SCORE</td>
<td>Factor1</td>
<td>0.390</td>
<td>0.378</td>
<td>0.347</td>
</tr>
</tbody>
</table>
```

Figure 88.2 lists the first two observations of the original SAS data set (Schools).

```
Figure 88.2  First Two Observations of the Schools Data Set

First Two Observations of the DATA= Data Set from PROC SCORE

<table>
<thead>
<tr>
<th>Obs</th>
<th>Type</th>
<th>English</th>
<th>Math</th>
<th>Biology</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 p</td>
<td></td>
<td>52</td>
<td>55</td>
<td>45</td>
</tr>
<tr>
<td>2 p</td>
<td></td>
<td>42</td>
<td>49</td>
<td>40</td>
</tr>
</tbody>
</table>
```

Figure 88.3 lists the first two observations of the output data set New created by PROC SCORE.

```
Figure 88.3  Listing of the New Data Set

First Two Observations of the OUT= Data Set from PROC SCORE

<table>
<thead>
<tr>
<th>Obs</th>
<th>Type</th>
<th>Factor1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 p</td>
<td></td>
<td>-0.17604</td>
</tr>
<tr>
<td>2 p</td>
<td></td>
<td>-0.80294</td>
</tr>
</tbody>
</table>
```

The score variable Factor1 in the New data set is named according to the value of the _NAME_ variable in the Scores data set. The values of the variable Factor1 are computed as follows: the DATA= data set variables are standardized using the same means and standard deviations that PROC FACTOR used when extracting the factors because the Scores data set contains observations with _TYPE_ = ’MEAN’ and _TYPE_ = ’STD’.

Note that in order to correctly use standardized scoring coefficients created by other procedures such as PROC FACTOR in this example, the data to be scored must be standardized in the same way that the data were standardized when the scoring coefficients were computed. Otherwise, the resulting scores might be incorrect. PROC SCORE does this automatically if the SCORE= data set is the original OUTSTAT= data set output from the procedure creating the scoring coefficients.
These standardized variables are then multiplied by their respective standardized scoring coefficients from
the data set Scores. These products are summed over all three variables, and the sum is the value of the new
variable Factor1. The first two values of the scored variable Factor1 are obtained as follows:

\[
\left( \frac{52 - 55.525}{12.949} \times 0.390 \right) + \left( \frac{55 - 52.325}{12.356} \times 0.378 \right) + \left( \frac{45 - 50.350}{12.239} \times 0.347 \right) = -0.17604
\]

\[
\left( \frac{42 - 55.525}{12.949} \times 0.390 \right) + \left( \frac{49 - 52.325}{12.356} \times 0.378 \right) + \left( \frac{40 - 50.350}{12.239} \times 0.347 \right) = -0.80294
\]

The following statements request that the SG PLOT procedure produce a horizontal bar chart of the variable
Type. The length of each bar represents the mean of the variable Factor1.

```plaintext
    title 'Mean Score of Variable Factor1 by Each Type of Schools';
    proc sgplot data=New;
        hbar type / stat = mean response=Factor1;
    run;
```

Figure 88.4 displays the mean score of the variable Factor1 for each of the three school types. For private
schools (Type=p), the average value of the variable Factor1 is 0.384, while for state-run schools the average
values are much lower. The state-run urban schools (Type=u) have the lowest mean value of –0.202, and the
state-run rural schools (Type=r) have a mean value of –0.183.
Syntax: SCORE Procedure

The following statements are available in the SCORE procedure:

```
PROC SCORE DATA=SAS-data-set <options> ;
  BY variables ;
  ID variables ;
  VAR variables ;
```

The only required statement is the PROC SCORE statement. The BY, ID, and VAR statements are described following the PROC SCORE statement.
The PROC SCORE statement invokes the SCORE procedure. Table 88.1 summarizes the options available in the PROC SCORE statement.

### Table 88.1 PROC SCORE Statement Options

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA=</td>
<td>Names the input SAS data set containing the raw data to score</td>
</tr>
<tr>
<td>NOSTD</td>
<td>Suppresses centering and scaling of the raw data</td>
</tr>
<tr>
<td>OUT=</td>
<td>Specifies the name of the SAS data set</td>
</tr>
<tr>
<td>PREDICT</td>
<td>Treats coefficients of –1 as 0</td>
</tr>
<tr>
<td>RESIDUAL</td>
<td>Reverses the sign of each score</td>
</tr>
<tr>
<td>SCORE=</td>
<td>Names the data set containing the scoring coefficients</td>
</tr>
<tr>
<td>TYPE=</td>
<td>Specifies the observations that contain scoring coefficients</td>
</tr>
</tbody>
</table>

You can specify the following options in the PROC SCORE statement.

**DATA=SAS-data-set**

names the input SAS data set containing the raw data to score. This option is required.

**NOSTD**

suppresses centering and scaling of the raw data. Ordinarily, if PROC SCORE finds _TYPE_=`MEAN`, _TYPE_=`USCORE`, _TYPE_=`USTD`, or _TYPE_=`STD` observations in the SCORE= data set, the procedure uses these to standardize the raw data before scoring.

**OUT=SAS-data-set**

specifies the name of the SAS data set created by PROC SCORE. If you want to create a SAS data set in a permanent library, you must specify a two-level name. For more information about permanent libraries and SAS data sets, see *SAS Language Reference: Concepts*. If the OUT= option is omitted, PROC SCORE still creates an output data set and automatically names it according to the DATAn convention, as if you omitted a data set name in a DATA statement.

**PREDICT**

specifies that PROC SCORE should treat coefficients of –1 in the SCORE= data set as 0. In regression applications, the dependent variable is coded with a coefficient of –1. Applied directly to regression results, PROC SCORE produces negative residuals (see the description of the RESIDUAL option, which follows); the PREDICT option produces predicted values instead.

**RESIDUAL**

reverses the sign of each score. Applied directly to regression results, PROC SCORE produces negative residuals (PREDICT–ACTUAL); the RESIDUAL option produces positive residuals (ACTUAL–PREDICT) instead.

**SCORE=SAS-data-set**

names the data set containing the scoring coefficients. If you omit the SCORE= option, the most recently created SAS data set is used. This data set must have two special variables: _TYPE_ and either _NAME_ or _MODEL_.


**TYPE=name or 'string'**

specifies the observations in the SCORE= data set that contain scoring coefficients. The TYPE= procedure option is unrelated to the data set option that has the same name. PROC SCORE examines the values of the special variable _TYPE_ in the SCORE= data set. When the value of _TYPE_ matches TYPE=name, the observation in the SCORE= data set is used to score the raw data in the DATA= data set.

If you omit the TYPE= option, scoring coefficients are read from observations with either _TYPE_='SCORE' or _TYPE_='USCORE'. Because the default for PROC SCORE is TYPE=SCORE, you need not specify the TYPE= option for factor scoring or for computing scores from OUTSTAT= data sets from the CANCORR, CANDISC, PRINCOMP, or VARCLUS procedure. When you use regression coefficients from PROC REG, specify TYPE=PARMS.

The maximum length of the argument specified in the TYPE= option depends on the length defined by the VALIDVARNAME= SAS system option. For additional information, see *SAS System Options: Reference*.

Note that the TYPE= option setting is not case sensitive. For example, the two option settings TYPE='MyScore' and TYPE='myscore' are equivalent.

---

**BY Statement**

**BY variables ;**

You can specify a BY statement with PROC SCORE to obtain separate analyses of observations in groups that are 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. If you specify more than one BY statement, only the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the NOTSORTED or DESCENDING option in the BY statement for the SCORE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about BY-group processing, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

You can specify a BY statement to apply separate groups of scoring coefficients to the entire DATA= data set. If the DATA= data set does not contain any of the BY variables, the entire DATA= data set is scored by each BY group of scoring coefficients in the SCORE= data set.

If the DATA= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the DATA= data set as in the SCORE= data set, then PROC SCORE prints an error message and stops.
If all the BY variables appear in the DATA= data set with the same type and length as in the SCORE= data set, then each BY group in the DATA= data set is scored using scoring coefficients from the corresponding BY group in the SCORE= data set. The BY groups in the DATA= data set must be in the same order as in the SCORE= data set. All BY groups in the DATA= data set must also appear in the SCORE= data set. If you do not specify the NOTSORTED option, some BY groups can appear in the SCORE= data set but not in the DATA= data set; such BY groups are not used in computing scores.

**ID Statement**

```
ID variables;
```

The ID statement identifies variables from the DATA= data set to be included in the OUT= data set. If there is no ID statement, all variables from the DATA= data set are included in the OUT= data set. The ID variables can be character or numeric.

**VAR Statement**

```
VAR variables;
```

The VAR statement specifies the variables to be used in computing scores. These variables must be in both the DATA= and SCORE= input data sets and must be numeric. If you do not specify a VAR statement, the procedure uses all numeric variables in the SCORE= data set. You should almost always specify a VAR statement with PROC SCORE because you would rarely use all the numeric variables in your data set to compute scores.

**Details: SCORE Procedure**

**Missing Values**

If one of the scoring variables in the DATA= data set has a missing value for an observation, all the scores have missing values for that observation. The exception to this criterion is that if the PREDICT option is specified, the variable with a coefficient of –1 can tolerate a missing value and still produce a prediction score. Also, a variable with a coefficient of 0 can tolerate a missing value.

If a scoring coefficient in the SCORE= data set has a missing value for an observation, the coefficient is not used in creating the new score variable for the observation. In other words, missing values of scoring coefficients are treated as zeros. This treatment affects only the observation in which the missing value occurs.
Regression Parameter Estimates from PROC REG

If the SCORE= data set is an OUTEST= data set produced by PROC REG and if you specify TYPE=PARMS, the interpretation of the new score variables depends on the PROC SCORE options chosen and the variables listed in the VAR statement. If the VAR statement contains only the independent variables used in a model in PROC REG, the new score variables give the predicted values. If the VAR statement contains the dependent variables and the independent variables used in a model in PROC REG, the interpretation of the new score variables depends on the PROC SCORE options chosen. If you omit both the PREDICT and the RESIDUAL options, the new score variables give negative residuals (PREDICT–ACTUAL). If you specify the RESIDUAL option, the new score variables give positive residuals (ACTUAL–PREDICT). If you specify the PREDICT option, the new score variables give predicted values.

Unless you specify the NOINT option for PROC REG, the OUTEST= data set contains the variable Intercept. The SCORE procedure uses the intercept value in computing the scores.

Output Data Set

PROC SCORE produces an output data set but displays no output. The output OUT= data set contains the following variables:

- the ID variables, if any
- all variables from the DATA= data set, if no ID variables are specified
- the BY variables, if any
- the new score variables, named from the _NAME_ or _MODEL_ values in the SCORE= data set

Computational Resources

Let

\[ v = \text{number of variables used in computing scores} \]
\[ s = \text{number of new score variables} \]
\[ b = \text{maximum number of new score variables in a BY group} \]
\[ n = \text{original input value} \]

Memory

The array storage required is approximately \(8(4v + (3 + v)b + s)\) bytes. When you do not use BY processing, the array storage required is approximately \(8(4v + (4 + v)s)\) bytes.

Time

The time required to construct the scoring matrix is roughly proportional to \(vs\), and the time needed to compute the scores is roughly proportional to \(nvs\).
Examples: SCORE Procedure

The following three examples use a subset of the Fitness data set. The complete data set is given in Chapter 85, “The REG Procedure.”

Example 88.1: Factor Scoring Coefficients

This example shows how to use PROC SCORE with factor scoring coefficients. First, the FACTOR procedure produces an output data set containing scoring coefficients in observations identified by _TYPE_='SCORE'. These data, together with the original data set Fitness, are supplied to PROC SCORE, resulting in a data set containing scores Factor1 and Factor2. The following statements produce Output 88.1.1 through Output 88.1.3:

```sas
/* This data set contains only the first 12 observations */
/* from the full data set used in the chapter on PROC REG. */
data Fitness;
  input Age Weight Oxygen RunTime RestPulse RunPulse @@;
datalines;
44 89.47 44.609 11.37 62 178 40 75.07 45.313 10.07 62 185
44 85.84 54.297 8.65 45 156 42 68.15 59.571 8.17 40 166
38 89.02 49.874 9.22 55 178 47 77.45 44.811 11.63 58 176
40 75.98 45.681 11.95 70 176 43 81.19 49.091 10.85 64 162
44 81.42 39.442 13.08 63 174 38 81.87 60.055 8.63 48 170
44 73.03 50.541 10.13 45 168 45 87.66 37.388 14.03 56 186
;
proc factor data=Fitness outstat=FactOut
  method=prin rotate=varimax score;
  var Age Weight RunTime RunPulse RestPulse RunPulse @@;
  title 'Factor Scoring Example';
run;
proc print data=FactOut;
  title2 'Data Set from PROC FACTOR';
run;
proc score data=Fitness score=FactOut out=Fscore;
  var Age Weight RunTime RunPulse RestPulse RunPulse;
run;
proc print data=Fscore;
  title2 'Data Set from PROC SCORE';
run;
```

Output 88.1.1 shows the PROC FACTOR output. The scoring coefficients for the two factors are shown at the end of the PROC FACTOR output.
Output 88.1.1 Creating an OUTSTAT= Data Set with PROC FACTOR

Factor Scoring Example

The FACTOR Procedure

Input Data Type: Raw Data
Number of Records Read: 12
Number of Records Used: 12
N for Significance Tests: 12

Factor Scoring Example

The FACTOR Procedure
Initial Factor Method: Principal Components

Prior Communality Estimates: ONE

Eigenvalues of the Correlation Matrix:
Total = 5  Average = 1

<table>
<thead>
<tr>
<th>Eigenvalue</th>
<th>Difference</th>
<th>Proportion</th>
<th>Cumulative</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2.30930638</td>
<td>0.4619</td>
<td>0.4619</td>
</tr>
<tr>
<td>2</td>
<td>1.19219952</td>
<td>0.30997249</td>
<td>0.7003</td>
</tr>
<tr>
<td>3</td>
<td>0.88222702</td>
<td>0.37965990</td>
<td>0.8767</td>
</tr>
<tr>
<td>4</td>
<td>0.50256713</td>
<td>0.38886717</td>
<td>0.9773</td>
</tr>
<tr>
<td>5</td>
<td>0.11369996</td>
<td>0.0227</td>
<td>1.0000</td>
</tr>
</tbody>
</table>

2 factors will be retained by the MINEIGEN criterion.

<table>
<thead>
<tr>
<th>Factor Pattern</th>
<th>Factor1</th>
<th>Factor2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>0.29795</td>
<td>0.93675</td>
</tr>
<tr>
<td>Weight</td>
<td>0.43282</td>
<td>-0.17750</td>
</tr>
<tr>
<td>RunTime</td>
<td>0.91983</td>
<td>0.28782</td>
</tr>
<tr>
<td>RunPulse</td>
<td>0.72671</td>
<td>-0.38191</td>
</tr>
<tr>
<td>RestPulse</td>
<td>0.81179</td>
<td>-0.23344</td>
</tr>
</tbody>
</table>

Variance Explained
by Each Factor

<table>
<thead>
<tr>
<th>Factor1</th>
<th>Factor2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3093064</td>
<td>1.1921995</td>
</tr>
</tbody>
</table>

Final Communality Estimates: Total = 3.501506

<table>
<thead>
<tr>
<th>Age</th>
<th>Weight</th>
<th>RunTime</th>
<th>RunPulse</th>
<th>RestPulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.96628351</td>
<td>0.21883401</td>
<td>0.92893333</td>
<td>0.67396207</td>
<td>0.71349297</td>
</tr>
</tbody>
</table>
Output 88.1.1 continued

Factor Scoring Example

The FACTOR Procedure
Rotation Method: Varimax

Orthogonal Transformation Matrix

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.92536 0.37908</td>
</tr>
<tr>
<td>2</td>
<td>-0.37908 0.92536</td>
</tr>
</tbody>
</table>

Rotated Factor Pattern

<table>
<thead>
<tr>
<th>Factor1</th>
<th>Factor2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>-0.07939 0.97979</td>
</tr>
<tr>
<td>Weight</td>
<td>0.46780 -0.00018</td>
</tr>
<tr>
<td>RunTime</td>
<td>0.74207 0.61503</td>
</tr>
<tr>
<td>RunPulse</td>
<td>0.81725 -0.07792</td>
</tr>
<tr>
<td>RestPulse</td>
<td>0.83969 0.09172</td>
</tr>
</tbody>
</table>

Variance Explained by Each Factor

<table>
<thead>
<tr>
<th>Factor1</th>
<th>Factor2</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1487753</td>
<td>1.3527306</td>
</tr>
</tbody>
</table>

Final Communality Estimates: Total = 3.501506

<table>
<thead>
<tr>
<th>Age</th>
<th>Weight</th>
<th>RunTime</th>
<th>RunPulse</th>
<th>RestPulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.96628351</td>
<td>0.21883401</td>
<td>0.92893333</td>
<td>0.67396207</td>
<td>0.71349297</td>
</tr>
</tbody>
</table>

Factor Scoring Example

The FACTOR Procedure
Rotation Method: Varimax

Scoring Coefficients Estimated by Regression

<table>
<thead>
<tr>
<th>Squared Multiple Correlations of the Variables with Each Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Factor1</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>1.0000000</td>
</tr>
</tbody>
</table>

Standardized Scoring Coefficients

<table>
<thead>
<tr>
<th>Factor1</th>
<th>Factor2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Age</td>
<td>-0.17846 0.77600</td>
</tr>
<tr>
<td>Weight</td>
<td>0.22987 -0.06672</td>
</tr>
<tr>
<td>RunTime</td>
<td>0.27707 0.37440</td>
</tr>
<tr>
<td>RunPulse</td>
<td>0.41263 -0.17714</td>
</tr>
<tr>
<td>RestPulse</td>
<td>0.39952 -0.04793</td>
</tr>
</tbody>
</table>
Output 88.1.2 lists the OUTSTAT= data set from PROC FACTOR. Note that observations 18 and 19 have _TYPE_ = 'SCORE'. Observations 1 and 2 have _TYPE_ = 'MEAN' and _TYPE_ = 'STD', respectively. These four observations are used by PROC SCORE.

**Output 88.1.2**  OUTSTAT= Data Set from PROC FACTOR Reproduced with PROC PRINT

```
Factor Scoring Example
Data Set from PROC FACTOR

<table>
<thead>
<tr>
<th>Obs</th>
<th><em>TYPE</em></th>
<th><em>NAME</em></th>
<th>Age</th>
<th>Weight</th>
<th>RunTime</th>
<th>RunPulse</th>
<th>RestPulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MEAN</td>
<td></td>
<td>42.4167</td>
<td>80.5125</td>
<td>10.6483</td>
<td>172.917</td>
<td>55.6667</td>
</tr>
<tr>
<td>2</td>
<td>STD</td>
<td></td>
<td>2.8431</td>
<td>6.7660</td>
<td>1.8444</td>
<td>8.918</td>
<td>9.2769</td>
</tr>
<tr>
<td>3</td>
<td>N</td>
<td></td>
<td>12.0000</td>
<td>12.0000</td>
<td>12.0000</td>
<td>12.0000</td>
<td>12.0000</td>
</tr>
<tr>
<td>4</td>
<td>CORR</td>
<td>Age</td>
<td>1.0000</td>
<td>0.0128</td>
<td>0.5005</td>
<td>-0.095</td>
<td>-0.0080</td>
</tr>
<tr>
<td>5</td>
<td>CORR</td>
<td>Weight</td>
<td>0.0128</td>
<td>1.0000</td>
<td>0.2637</td>
<td>0.173</td>
<td>0.2396</td>
</tr>
<tr>
<td>6</td>
<td>CORR</td>
<td>RunTime</td>
<td>0.5005</td>
<td>0.2637</td>
<td>1.0000</td>
<td>0.556</td>
<td>0.6620</td>
</tr>
<tr>
<td>7</td>
<td>CORR</td>
<td>RunPulse</td>
<td>-0.0953</td>
<td>0.1731</td>
<td>0.5555</td>
<td>1.000</td>
<td>0.4853</td>
</tr>
<tr>
<td>8</td>
<td>CORR</td>
<td>RestPulse</td>
<td>-0.0080</td>
<td>0.2396</td>
<td>0.6620</td>
<td>0.485</td>
<td>1.0000</td>
</tr>
<tr>
<td>9</td>
<td>COMMUNAL</td>
<td></td>
<td>0.9663</td>
<td>0.2188</td>
<td>0.9289</td>
<td>0.674</td>
<td>0.7135</td>
</tr>
<tr>
<td>10</td>
<td>PRIORS</td>
<td></td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
<td>1.0000</td>
</tr>
<tr>
<td>11</td>
<td>EIGENVAL</td>
<td></td>
<td>2.3093</td>
<td>1.1922</td>
<td>0.8822</td>
<td>0.503</td>
<td>0.1137</td>
</tr>
<tr>
<td>12</td>
<td>UNROTATE</td>
<td>Factor1</td>
<td>0.2980</td>
<td>0.4328</td>
<td>0.9198</td>
<td>0.727</td>
<td>0.8118</td>
</tr>
<tr>
<td>13</td>
<td>UNROTATE</td>
<td>Factor2</td>
<td>0.9368</td>
<td>-0.1775</td>
<td>0.2878</td>
<td>-0.382</td>
<td>-0.2334</td>
</tr>
<tr>
<td>14</td>
<td>TRANSFOR</td>
<td>Factor1</td>
<td>0.9254</td>
<td>-0.3791</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>TRANSFOR</td>
<td>Factor2</td>
<td>0.3791</td>
<td>0.9254</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>PATTERN</td>
<td>Factor1</td>
<td>-0.0794</td>
<td>0.4678</td>
<td>0.7421</td>
<td>0.817</td>
<td>0.8397</td>
</tr>
<tr>
<td>17</td>
<td>PATTERN</td>
<td>Factor2</td>
<td>0.9798</td>
<td>-0.0002</td>
<td>0.6150</td>
<td>-0.078</td>
<td>0.0917</td>
</tr>
<tr>
<td>18</td>
<td>SCORE</td>
<td>Factor1</td>
<td>-0.1785</td>
<td>0.2299</td>
<td>0.2771</td>
<td>0.413</td>
<td>0.3995</td>
</tr>
<tr>
<td>19</td>
<td>SCORE</td>
<td>Factor2</td>
<td>0.7760</td>
<td>-0.0667</td>
<td>0.3744</td>
<td>-0.177</td>
<td>0.0479</td>
</tr>
</tbody>
</table>
```

Since the PROC SCORE statement does not contain the NOSTD option, the data in the Fitness data set are standardized before scoring. For each variable specified in the VAR statement, the mean and standard deviation are obtained from the FactOut data set. For each observation in the Fitness data set, the variables are then standardized. For example, for observation 1 in the Fitness data set, the variable Age is standardized to $0.5569 = [(44 - 42.4167)/2.8431]$. After the data in the Fitness data set are standardized, the standardized values of the variables in the VAR statement are multiplied by the matching coefficients in the FactOut data set, and the resulting products are summed. This sum is output as a value of the new score variable.

**Output 88.1.3** displays the FScore data set produced by PROC SCORE. This data set contains the variables Age, Weight, Oxygen, RunTime, RestPulse, and RunPulse from the Fitness data set. It also contains Factor1 and Factor2, the two new score variables.
Example 88.2: Regression Parameter Estimates

In this example, PROC REG computes regression parameter estimates for the Fitness data. (See Example 88.1 to for more information about how to create the Fitness data set.) The parameter estimates are output to a data set and used as scoring coefficients. For the first part of this example, PROC SCORE is used to score the Fitness data, which are the same data used in the regression.

In the second part of this example, PROC SCORE is used to score a new data set, Fitness2. For PROC SCORE, the TYPE= specification is PARMS, and the names of the score variables are found in the variable _MODEL_, which gets its values from the model label. The following code produces Output 88.2.1 through Output 88.2.3:

```plaintext
proc reg data=Fitness outest=RegOut;
  OxyHat: model Oxygen=Age Weight RunTime RunPulse RestPulse;
  title 'Regression Scoring Example';
run;
proc print data=RegOut;
  title2 'OUTEST= Data Set from PROC REG';
run;
proc score data=Fitness score=RegOut out=RScoreP type=parms;
  var Age Weight RunTime RunPulse RestPulse;
run;
proc print data=RScoreP;
  title2 'Predicted Scores for Regression';
run;
.proc score data=Fitness score=RegOut out=RScoreR type=parms;
  var Oxygen Age Weight RunTime RunPulse RestPulse;
run;
```
Example 88.2: Regression Parameter Estimates

```
proc print data=RScoreR;
  title2 'Negative Residual Scores for Regression';
run;
```

Output 88.2.1 shows the PROC REG output. The column labeled “Parameter Estimates” lists the parameter estimates. These estimates are output to the RegOut data set.

**Output 88.2.1 Creating an OUTEST= Data Set with PROC REG**

**Regression Scoring Example**

**The REG Procedure**

**Model: OxyHat**

**Dependent Variable: Oxygen**

<table>
<thead>
<tr>
<th>Source</th>
<th>DF</th>
<th>Sum of Squares</th>
<th>Mean Square</th>
<th>F Value</th>
<th>Pr &gt; F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>5</td>
<td>509.62201</td>
<td>101.92440</td>
<td>15.80</td>
<td>0.0021</td>
</tr>
<tr>
<td>Error</td>
<td>6</td>
<td>38.70060</td>
<td>6.45010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corrected Total</td>
<td>11</td>
<td>548.32261</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Root MSE       | 2.53970 |
| Dependent Mean | 48.38942 |
| Coeff Var      | 5.24847 |

**Output 88.2.2 lists the RegOut data set. Note that _TYPE_='PARMS' and _MODEL_='OXYHAT', which are from the label in the MODEL statement in PROC REG.**

**Output 88.2.2 OUTEST= Data Set from PROC REG Reproduced with PROC PRINT**

<table>
<thead>
<tr>
<th>Obs</th>
<th><em>MODEL</em></th>
<th><em>TYPE</em></th>
<th><em>DEPVAR</em></th>
<th><em>RMSE</em></th>
<th>Intercept</th>
<th>Age</th>
<th>Weight</th>
<th>RunTime</th>
<th>RunPulse</th>
<th>RestPulse</th>
<th>Oxygen</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>OxyHat</td>
<td>PARMS</td>
<td>Oxygen</td>
<td>2.53970</td>
<td>151.91550</td>
<td>-0.63045</td>
<td>-0.10586</td>
<td>-1.75698</td>
<td>-0.22891</td>
<td>-0.17910</td>
<td>-1</td>
</tr>
</tbody>
</table>
Output 88.2.3 lists the data sets created by PROC SCORE. Since the SCORE= data set does not contain observations with _TYPE_ = ’MEAN’ or _TYPE_ = ’STD’, the data in the Fitness data set are not standardized before scoring. The SCORE= data set contains the variable Intercept, so this intercept value is used in computing the score. To produce the RScoreP data set, the VAR statement in PROC SCORE includes only the independent variables from the model in PROC REG. As a result, the OxyHat variable contains predicted values. To produce the RScoreR data set, the VAR statement in PROC SCORE includes both the dependent variables and the independent variables from the model in PROC REG. As a result, the OxyHat variable contains negative residuals (PREDICT–ACTUAL) as shown in Output 88.2.4. If the RESIDUAL option is specified, the variable OxyHat contains positive residuals (ACTUAL–PREDICT). If the PREDICT option is specified, the OxyHat variable contains predicted values.

### Output 88.2.3 Predicted Scores from the OUT= Data Set Created by PROC SCORE

**Regression Scoring Example**

**Predicted Scores for Regression**

<table>
<thead>
<tr>
<th>Obs</th>
<th>Age</th>
<th>Weight</th>
<th>Oxygen</th>
<th>RunTime</th>
<th>RestPulse</th>
<th>RunPulse</th>
<th>OxyHat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>44</td>
<td>89.47</td>
<td>44.609</td>
<td>11.37</td>
<td>62</td>
<td>178</td>
<td>42.8771</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>75.07</td>
<td>45.313</td>
<td>10.07</td>
<td>62</td>
<td>185</td>
<td>47.6050</td>
</tr>
<tr>
<td>3</td>
<td>44</td>
<td>85.84</td>
<td>54.297</td>
<td>8.65</td>
<td>45</td>
<td>156</td>
<td>56.1211</td>
</tr>
<tr>
<td>4</td>
<td>42</td>
<td>68.15</td>
<td>59.571</td>
<td>8.17</td>
<td>40</td>
<td>166</td>
<td>58.7044</td>
</tr>
<tr>
<td>5</td>
<td>38</td>
<td>89.02</td>
<td>49.874</td>
<td>9.22</td>
<td>55</td>
<td>178</td>
<td>51.7386</td>
</tr>
<tr>
<td>6</td>
<td>47</td>
<td>77.45</td>
<td>44.811</td>
<td>11.63</td>
<td>58</td>
<td>176</td>
<td>42.9756</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>75.98</td>
<td>45.681</td>
<td>11.95</td>
<td>70</td>
<td>176</td>
<td>44.8329</td>
</tr>
<tr>
<td>8</td>
<td>43</td>
<td>81.19</td>
<td>49.091</td>
<td>10.85</td>
<td>64</td>
<td>162</td>
<td>48.6020</td>
</tr>
<tr>
<td>9</td>
<td>44</td>
<td>81.42</td>
<td>39.442</td>
<td>13.08</td>
<td>63</td>
<td>174</td>
<td>41.4613</td>
</tr>
<tr>
<td>10</td>
<td>38</td>
<td>81.87</td>
<td>60.055</td>
<td>8.63</td>
<td>48</td>
<td>170</td>
<td>56.6171</td>
</tr>
<tr>
<td>11</td>
<td>44</td>
<td>73.03</td>
<td>50.541</td>
<td>10.13</td>
<td>45</td>
<td>168</td>
<td>52.1299</td>
</tr>
<tr>
<td>12</td>
<td>45</td>
<td>87.66</td>
<td>37.388</td>
<td>14.03</td>
<td>56</td>
<td>186</td>
<td>37.0080</td>
</tr>
</tbody>
</table>

### Output 88.2.4 Residual Scores from the OUT= Data Set Created by PROC SCORE

**Regression Scoring Example**

**Negative Residual Scores for Regression**

<table>
<thead>
<tr>
<th>Obs</th>
<th>Age</th>
<th>Weight</th>
<th>Oxygen</th>
<th>RunTime</th>
<th>RestPulse</th>
<th>RunPulse</th>
<th>OxyHat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>44</td>
<td>89.47</td>
<td>44.609</td>
<td>11.37</td>
<td>62</td>
<td>178</td>
<td>-1.73195</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>75.07</td>
<td>45.313</td>
<td>10.07</td>
<td>62</td>
<td>185</td>
<td>2.29197</td>
</tr>
<tr>
<td>3</td>
<td>44</td>
<td>85.84</td>
<td>54.297</td>
<td>8.65</td>
<td>45</td>
<td>156</td>
<td>1.82407</td>
</tr>
<tr>
<td>4</td>
<td>42</td>
<td>68.15</td>
<td>59.571</td>
<td>8.17</td>
<td>40</td>
<td>166</td>
<td>-0.86657</td>
</tr>
<tr>
<td>5</td>
<td>38</td>
<td>89.02</td>
<td>49.874</td>
<td>9.22</td>
<td>55</td>
<td>178</td>
<td>1.86460</td>
</tr>
<tr>
<td>6</td>
<td>47</td>
<td>77.45</td>
<td>44.811</td>
<td>11.63</td>
<td>58</td>
<td>176</td>
<td>-1.83542</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>75.98</td>
<td>45.681</td>
<td>11.95</td>
<td>70</td>
<td>176</td>
<td>-0.84811</td>
</tr>
<tr>
<td>8</td>
<td>43</td>
<td>81.19</td>
<td>49.091</td>
<td>10.85</td>
<td>64</td>
<td>162</td>
<td>-0.48897</td>
</tr>
<tr>
<td>9</td>
<td>44</td>
<td>81.42</td>
<td>39.442</td>
<td>13.08</td>
<td>63</td>
<td>174</td>
<td>2.01935</td>
</tr>
<tr>
<td>10</td>
<td>38</td>
<td>81.87</td>
<td>60.055</td>
<td>8.63</td>
<td>48</td>
<td>170</td>
<td>-3.43787</td>
</tr>
<tr>
<td>11</td>
<td>44</td>
<td>73.03</td>
<td>50.541</td>
<td>10.13</td>
<td>45</td>
<td>168</td>
<td>1.58892</td>
</tr>
<tr>
<td>12</td>
<td>45</td>
<td>87.66</td>
<td>37.388</td>
<td>14.03</td>
<td>56</td>
<td>186</td>
<td>-0.38002</td>
</tr>
</tbody>
</table>
The second part of this example uses the parameter estimates to score a new data set. The following statements produce Output 88.2.5 and Output 88.2.6:

```sas
/* The FITNESS2 data set contains observations 13-16 from */
/* the FITNESS data set used in EXAMPLE 2 in the PROC REG */
/* chapter. */
data Fitness2;
  input Age Weight Oxygen RunTime RestPulse RunPulse;
datalines;
45 66.45 44.754 11.12 51 176
47 79.15 47.273 10.60 47 162
54 83.12 51.855 10.33 50 166
49 81.42 49.156 8.95 44 180;
proc print data=Fitness2;
  title 'Regression Scoring Example';
  title2 'New Raw Data Set to be Scored';
run;
proc score data=Fitness2 score=RegOut out=NewPred type=parms
  nostd predict;
  var Oxygen Age Weight RunTime RunPulse RestPulse;
run;
proc print data=NewPred;
  title2 'Predicted Scores for Regression';
  title3 'for Additional Data from FITNESS2';
run;
```

Output 88.2.5 lists the Fitness2 data set.

**Output 88.2.5** Listing of the Fitness2 Data Set

<table>
<thead>
<tr>
<th>Obs</th>
<th>Age</th>
<th>Weight</th>
<th>Oxygen</th>
<th>RunTime</th>
<th>RestPulse</th>
<th>RunPulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>45</td>
<td>66.45</td>
<td>44.754</td>
<td>11.12</td>
<td>51</td>
<td>176</td>
</tr>
<tr>
<td>2</td>
<td>47</td>
<td>79.15</td>
<td>47.273</td>
<td>10.60</td>
<td>47</td>
<td>162</td>
</tr>
<tr>
<td>3</td>
<td>54</td>
<td>83.12</td>
<td>51.855</td>
<td>10.33</td>
<td>50</td>
<td>166</td>
</tr>
<tr>
<td>4</td>
<td>49</td>
<td>81.42</td>
<td>49.156</td>
<td>8.95</td>
<td>44</td>
<td>180</td>
</tr>
</tbody>
</table>

PROC SCORE scores the Fitness2 data set by using the parameter estimates in the RegOut data set. These parameter estimates result from fitting a regression equation to the Fitness data set. The NOSTD option is specified, so the raw data are not standardized before scoring. (However, the NOSTD option is not necessary here. The SCORE= data set does not contain observations with _TYPE_='MEAN' or _TYPE_='STD', so standardization is not performed.) The VAR statement contains the dependent variables and the independent variables used in PROC REG. In addition, the PREDICT option is specified. This combination gives predicted values for the new score variable. The name of the new score variable is OxyHat, from the value of the _MODEL_ variable in the SCORE= data set. Output 88.2.6 shows the data set produced by PROC SCORE.
**Output 88.2.6** Predicted Scores from the OUT= Data Set Created by PROC SCORE and Reproduced Using PROC PRINT

**Regression Scoring Example**  
**Predicted Scores for Regression for Additional Data from FITNESS2**

<table>
<thead>
<tr>
<th>Obs</th>
<th>Age</th>
<th>Weight</th>
<th>Oxygen</th>
<th>RunTime</th>
<th>RestPulse</th>
<th>RunPulse</th>
<th>OxyHat</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>45</td>
<td>66.45</td>
<td>44.754</td>
<td>11.12</td>
<td>51</td>
<td>176</td>
<td>47.5507</td>
</tr>
<tr>
<td>2</td>
<td>47</td>
<td>79.15</td>
<td>47.273</td>
<td>10.60</td>
<td>47</td>
<td>162</td>
<td>49.7802</td>
</tr>
<tr>
<td>3</td>
<td>54</td>
<td>83.12</td>
<td>51.855</td>
<td>10.33</td>
<td>50</td>
<td>166</td>
<td>43.9682</td>
</tr>
<tr>
<td>4</td>
<td>49</td>
<td>81.42</td>
<td>49.156</td>
<td>8.95</td>
<td>44</td>
<td>180</td>
<td>47.5949</td>
</tr>
</tbody>
</table>

**Example 88.3: Custom Scoring Coefficients**

This example uses a specially created custom scoring data set and produces Output 88.3.1 and Output 88.3.2. The first scoring coefficient creates a variable that is Age–Weight; the second scoring coefficient evaluates the variable RunPulse–RstPulse; and the third scoring coefficient totals all six variables. Since the scoring coefficients data set (data set A) does not contain any observations with _TYPE_='MEAN' or _TYPE_='STD', the data in the Fitness data set (see Example 88.1) are not standardized before scoring.

The following statements produce Output 88.3.1 and Output 88.3.2:

```plaintext
data A;
   input _type_ $ _name_ $;
   Age Weight RunTime RunPulse RestPulse;
   datalines;
   SCORE AGE_WGT 1 -1 0 0 0
   SCORE RUN_RST 0 0 0 1 -1
   SCORE TOTAL 1 1 1 1 1
;
proc print data=A;
   title 'Constructed Scoring Example';
   title2 'Scoring Coefficients';
run;

proc score data=Fitness score=A out=B;
   var Age Weight RunTime RunPulse RestPulse;
run;

proc print data=B;
   title2 'Scored Data';
run;
```
Output 88.3.1 Custom Scoring Data Set and Scored Fitness Data: PROC PRINT

**Constructed Scoring Example**

**Scoring Coefficients**

<table>
<thead>
<tr>
<th>Obs</th>
<th>type</th>
<th>name</th>
<th>Age</th>
<th>Weight</th>
<th>RunTime</th>
<th>RunPulse</th>
<th>RestPulse</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>SCORE</td>
<td>AGE_WGT</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>SCORE</td>
<td>RUN_RST</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td>3</td>
<td>SCORE</td>
<td>TOTAL</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Output 88.3.2 Custom Scored Fitness Data: PROC PRINT

**Constructed Scoring Example**

**Scored Data**

<table>
<thead>
<tr>
<th>Obs</th>
<th>Age</th>
<th>Weight</th>
<th>Oxygen</th>
<th>RunTime</th>
<th>RestPulse</th>
<th>RunPulse</th>
<th>AGE_WGT</th>
<th>RUN_RST</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>44</td>
<td>89.47</td>
<td>44.609</td>
<td>11.37</td>
<td>62</td>
<td>178</td>
<td>-45.47</td>
<td>116</td>
<td>384.84</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>75.07</td>
<td>45.313</td>
<td>10.07</td>
<td>62</td>
<td>185</td>
<td>-35.07</td>
<td>123</td>
<td>372.14</td>
</tr>
<tr>
<td>3</td>
<td>44</td>
<td>85.84</td>
<td>54.297</td>
<td>8.65</td>
<td>45</td>
<td>156</td>
<td>-41.84</td>
<td>111</td>
<td>339.49</td>
</tr>
<tr>
<td>4</td>
<td>42</td>
<td>68.15</td>
<td>59.571</td>
<td>8.17</td>
<td>40</td>
<td>166</td>
<td>-26.15</td>
<td>126</td>
<td>324.32</td>
</tr>
<tr>
<td>5</td>
<td>38</td>
<td>89.02</td>
<td>49.874</td>
<td>9.22</td>
<td>55</td>
<td>178</td>
<td>-51.02</td>
<td>123</td>
<td>369.24</td>
</tr>
<tr>
<td>6</td>
<td>47</td>
<td>77.45</td>
<td>44.811</td>
<td>11.63</td>
<td>58</td>
<td>176</td>
<td>-30.45</td>
<td>118</td>
<td>370.08</td>
</tr>
<tr>
<td>7</td>
<td>40</td>
<td>75.98</td>
<td>45.681</td>
<td>11.95</td>
<td>70</td>
<td>176</td>
<td>-35.98</td>
<td>106</td>
<td>373.93</td>
</tr>
<tr>
<td>8</td>
<td>43</td>
<td>81.19</td>
<td>49.091</td>
<td>10.85</td>
<td>64</td>
<td>162</td>
<td>-38.19</td>
<td>98</td>
<td>361.04</td>
</tr>
<tr>
<td>9</td>
<td>44</td>
<td>81.42</td>
<td>39.442</td>
<td>13.08</td>
<td>63</td>
<td>174</td>
<td>-37.42</td>
<td>111</td>
<td>375.50</td>
</tr>
<tr>
<td>10</td>
<td>38</td>
<td>81.87</td>
<td>60.055</td>
<td>8.63</td>
<td>48</td>
<td>170</td>
<td>-43.87</td>
<td>122</td>
<td>346.50</td>
</tr>
<tr>
<td>11</td>
<td>44</td>
<td>73.03</td>
<td>50.541</td>
<td>10.13</td>
<td>45</td>
<td>168</td>
<td>-29.03</td>
<td>123</td>
<td>340.16</td>
</tr>
<tr>
<td>12</td>
<td>45</td>
<td>87.66</td>
<td>37.388</td>
<td>14.03</td>
<td>56</td>
<td>186</td>
<td>-42.66</td>
<td>130</td>
<td>388.69</td>
</tr>
</tbody>
</table>

References

Chaseling, J. (1996), “Standard Test Results of Students at Three Types of Schools,” Sample data, Faculty of Environmental Sciences, Griffith University, Queensland, Australia.
Subject Index

analyzing data in groups
   SCORE procedure, 7301

custom scoring coefficients, example
   SCORE procedure, 7312

DOT product (SCORE), 7293

factor scoring coefficients
   FACTOR procedure, 7294
   SCORE procedure, 7294, 7304

matrix
   multiplication (SCORE), 7293

missing values
   SCORE procedure, 7302

OUT= data sets
   SCORE procedure, 7303

output data set
   SCORE procedure, 7300, 7303

regression coefficients
   using with SCORE procedure, 7294

regression parameter estimates, example
   SCORE procedure, 7308

SCORE procedure
   computational resources, 7303
   examples, 7295, 7304
   input data set, 7300
   OUT= data sets, 7303
   output data set, 7300, 7303
   regression parameter estimates from REG
      procedure, 7303
   scoring coefficients, 7294

score variables
   interpretation (SCORE), 7303

scoring coefficients (SCORE), 7293

standardizing
   raw data (SCORE), 7294
Syntax Index

BY statement
   SCORE procedure, 7301

DATA= option
   PROC SCORE statement, 7300

NOSTD option
   PROC SCORE statement, 7300

OUT= option
   PROC SCORE statement, 7300

PREDICT option
   PROC SCORE statement, 7300
   PROC SCORE statement, see SCORE procedure

RESIDUAL option
   PROC SCORE statement, 7300

SCORE procedure
   syntax, 7299
   SCORE procedure, BY statement, 7301
   SCORE procedure, ID statement, 7302
   SCORE procedure, PROC SCORE statement, 7299
      DATA= option, 7300
      NOSTD option, 7300
      OUT= option, 7300
      PREDICT option, 7300
      RESIDUAL option, 7300
      SCORE= option, 7300
      TYPE= option, 7301
   SCORE procedure, VAR statement, 7302
   SCORE= option
      PROC SCORE statement, 7300

TYPE= option
   PROC SCORE statement, 7301