CHAPTER 35

The STANDARD Procedure

Overview

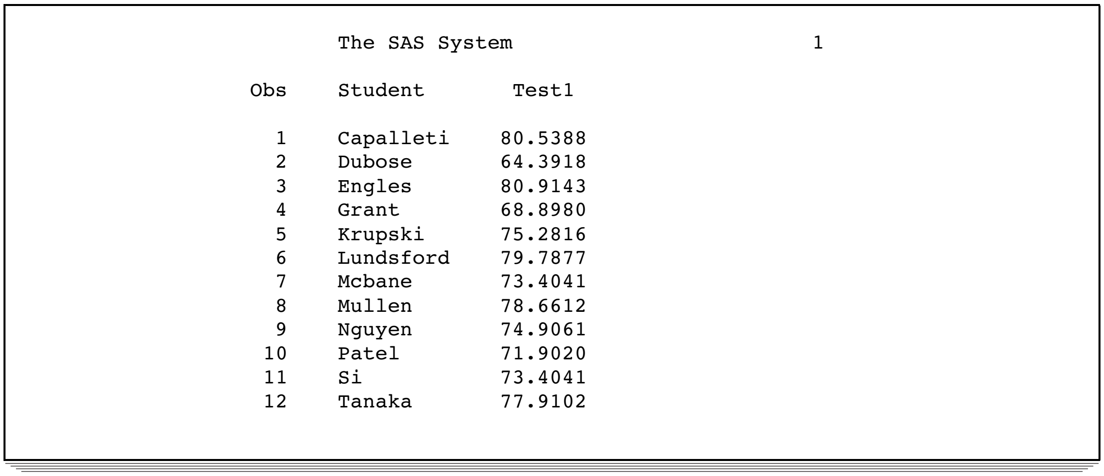
The STANDARD procedure standardizes variables in a SAS data set to a given mean and standard deviation, and it creates a new SAS data set containing the standardized values.

Output 35.1 on page 1135 shows a simple standardization where the output data set contains standardized student exam scores. The statements that produce the output follow:

```
proc standard data=score mean=75 std=5
              out=stndtest;
run;

proc print data=stndtest;
run;
```

Output 35.1 Standardized Test Scores Using PROC STANDARD



```
                           The SAS System                        1

                    Obs    Student       Test1

                      1    Capalleti    80.5388
                      2    Dubose       64.3918
                      3    Engles       80.9143
                      4    Grant        68.8980
                      5    Krupski      75.2816
                      6    Lundsford    79.7877
                      7    Mcbane       73.4041
                      8    Mullen       78.6612
                      9    Nguyen       74.9061
                     10    Patel        71.9020
                     11    Si           73.4041
                     12    Tanaka       77.9102
```

Output 35.2 on page 1136 shows a more complex example that uses BY-group processing. PROC STANDARD computes Z scores separately for two BY groups by standardizing life-expectancy data to a mean of 0 and a standard deviation of 1. The data are 1950 and 1993 life expectancies at birth for 16 countries. The birth rates for each country, classified as stable or rapid, form the two BY groups. The statements that produce the analysis also

□ print statistics for each variable to standardize

□ replace missing values with the given mean

□ calculate standardized values using a given mean and standard deviation

□ print the data set with the standardized values.

For an explanation of the program that produces this output, see Example 2 on page 1145.

Output 35.2 Z Scores for Each BY Group Using PROC STANDARD

```
                       Life Expectancies by Birth Rate                        1

---------------------------- PopulationRate=Stable ----------------------------

                              Standard
 Name              Mean       Deviation              N    Label

 Life50       67.400000        1.854724              5    1950 life expectancy
 Life93       74.500000        4.888763              6    1993 life expectancy


----------------------------- PopulationRate=Rapid ----------------------------

                              Standard
 Name              Mean       Deviation              N    Label

 Life50       42.000000        5.033223              8    1950 life expectancy
 Life93       59.100000        8.225300             10    1993 life expectancy
                   Standardized Life Expectancies at Birth                    2
                          by a Country's Birth Rate

              Population
                 Rate       Country            Life50      Life93

                Stable      France           -0.21567     0.51138
                Stable      Germany           0.32350     0.10228
                Stable      Japan            -1.83316     0.92048
                Stable      Russia            0.00000    -1.94323
                Stable      United Kingdom    0.86266     0.30683
                Stable      United States     0.86266     0.10228
                Rapid       Bangladesh        0.00000    -0.74161
                Rapid       Brazil            1.78812     0.96045
                Rapid       China            -0.19868     1.32518
                Rapid       Egypt             0.00000     0.10942
                Rapid       Ethiopia         -1.78812    -1.59265
                Rapid       India            -0.59604    -0.01216
                Rapid       Indonesia        -0.79472    -0.01216
                Rapid       Mozambique        0.00000    -1.47107
                Rapid       Philippines       1.19208     0.59572
                Rapid       Turkey            0.39736     0.83888
```

Procedure Syntax

Tip: Supports the Output Delivery System (see Chapter 2, "Fundamental Concepts for Using Base SAS Procedures")

Reminder: You can use the ATTRIB, FORMAT, LABEL, and WHERE statements. See Chapter 3, "Statements with the Same Function in Multiple Procedures," for details. You can also use any global statements as well. See Chapter 2, "Fundamental Concepts for Using Base SAS Procedures," for a list.

PROC STANDARD <*option(s)*>;

BY <DESCENDING> *variable-1* <...<DESCENDING> *variable-n*>
<NOTSORTED>;

FREQ *variable*;

VAR *variable(s)*;

WEIGHT *variable*;

To do this	Use this statement
Calculate separate standardized values for each BY group	BY
Identify a variable whose values represent the frequency of each observation	FREQ
Select the variables to standardize and determine the order they appear in the printed output	VAR
Identify a variable whose values weight each observation in the statistical calculations	WEIGHT

PROC STANDARD Statement

PROC STANDARD <*option(s)*>;

To do this	Use this option
Specify the input data set	DATA=
Specify the output data set	OUT=
Computational options	
Exclude observations with nonpositive weights	EXCLNPWGT
Specify the mean value	MEAN=
Replace missing values with a variable mean or MEAN= value	REPLACE
Specify the standard deviation value	STD=
Specify the divisor for variance calculations	VARDEF=
Control printed output	
Print statistics for each variable to standardize	PRINT

Without Options

If you do not specify MEAN=, REPLACE, or STD=, the output data set is an identical copy of the input data set.

Options

DATA=*SAS-data-set*
identifies the input SAS data set.

Main discussion: "Input Data Sets" on page 18

Restriction: You cannot use PROC STANDARD with an engine that supports concurrent access if another user is updating the data set at the same time.

EXCLNPWGT

excludes observations with nonpositive weight values (zero or negative). The procedure does not use the observation to calculate the mean and standard deviation, but the observation is still standardized. By default, the procedure treats observations with negative weights like those with zero weights and counts them in the total number of observations.

MEAN=*mean-value*

standardizes variables to a mean of *mean-value*.

Alias: M=

Default: mean of the input values

Featured in: Example 1 on page 1143

OUT=*SAS-data-set*

identifies the output data set. If *SAS-data-set* does not exist, PROC STANDARD creates it. If you omit OUT=, the data set is named DATA*n*, where *n* is the smallest integer that makes the name unique.

Default: DATA*n*

Featured in: Example 1 on page 1143

PRINT

prints the original frequency, mean, and standard deviation for each variable to standardize.

Featured in: Example 2 on page 1145

REPLACE

replaces missing values with the variable mean.

Interaction: If you use MEAN=, PROC STANDARD replaces missing values with the given mean.

Featured in: Example 2 on page 1145

STD=*std-value*

standardizes variables to a standard deviation of *std-value*.

Alias: S=

Default: standard deviation of the input values

Featured in: Example 1 on page 1143

VARDEF=*divisor*

specifies the divisor to use in the calculation of variances and standard deviation. Table 35.1 on page 1139 shows the possible values for *divisor* and the associated divisors.

Table 35.1 Possible Values for VARDEF=

Value	Divisor	Formula for Divisor
DF	degrees of freedom	$n - 1$
N	number of observations	n
WDF	sum of weights minus one	$(\Sigma_i w_i) - 1$
WEIGHT |WGT	sum of weights	$\Sigma_i w_i$

The procedure computes the variance as $CSS/divisor$, where CSS is the corrected sums of squares and equals $\sum\left(x_i - \overline{x}\right)^2$. When you weight the analysis variables, CSS equals $\sum w_i\left(x_i - \overline{x}_w\right)^2$ where $\overline{x}_w$ is the weighted mean.

Default: DF

Tip: When you use the WEIGHT statement and VARDEF=DF, the variance is an estimate of σ^2, where the variance of the ith observation is $var\left(x_i\right) = \sigma^2/w_i$ and w_i is the weight for the ith observation. This yields an estimate of the variance of an observation with unit weight.

Tip: When you use the WEIGHT statement and VARDEF=WGT, the computed variance is asymptotically (for large n) an estimate of $\sigma^2/\overline{w}$, where $\overline{w}$ is the average weight. This yields an asymptotic estimate of the variance of an observation with average weight.

See also: “WEIGHT” on page 73

Main discussion: “Keywords and Formulas” on page 1458

BY Statement

Calculates standardized values separately for each BY group.

Main discussion: “BY” on page 68

Featured in: Example 2 on page 1145

BY <DESCENDING> *variable-1* <...<DESCENDING> *variable-n*><NOTSORTED>;

Required Arguments

variable

specifies the variable that the procedure uses to form BY groups. You can specify more than one variable. If you do not use the NOTSORTED option in the BY statement, the observations in the data set must either be sorted by all the variables that you specify, or they must be indexed appropriately. These variables are called *BY variables*.

Options

DESCENDING

specifies that the data set is sorted in descending order by the variable that immediately follows the word DESCENDING in the BY statement.

NOTSORTED

specifies that observations are not necessarily sorted in alphabetic or numeric order. The data are grouped in another way, for example, chronological order.

The requirement for ordering or indexing observations according to the values of BY variables is suspended for BY-group processing when you use the NOTSORTED option. In fact, the procedure does not use an index if you specify NOTSORTED. The procedure defines a BY group as a set of contiguous observations that have the same values for all BY variables. If observations with the same values for the BY variables are not contiguous, the procedure treats each contiguous set as a separate BY group.

FREQ Statement

Specifies a numeric variable whose values represent the frequency of the observation.

Tip: The effects of the FREQ and WEIGHT statements are similar except when calculating degrees of freedom.

See also: For an example that uses the FREQ statement, see “FREQ” on page 70

FREQ *variable*;

Required Arguments

variable

specifies a numeric variable whose value represents the frequency of the observation. If you use the FREQ statement, the procedure assumes that each observation represents *n* observations, where *n* is the value of *variable*. If *n* is not an integer, the SAS System truncates it. If *n* is less than 1 or is missing, the procedure does not use that observation to calculate statistics but the observation is still standardized.

The sum of the frequency variable represents the total number of observations.

VAR Statement

Specifies the variables to standardize and their order in the printed output.

Default: If you omit the VAR statement, PROC STANDARD standardizes all numeric variables not listed in the other statements.

Featured in: Example 1 on page 1143

VAR *variable(s)*;

Required Arguments

variable(s)

identifies one or more variables to standardize.

WEIGHT Statement

Specifies weights for analysis variables in the statistical calculations.

See also: For information on calculating weighted statistics and for an example that uses the WEIGHT statement, see "WEIGHT" on page 73

WEIGHT *variable*;

Required Arguments

variable

specifies a numeric variable whose values weight the values of the analysis variables. The values of the variable do not have to be integers. If the value of the weight variable is

Weight value…	PROC STANDARD…
0	counts the observation in the total number of observations
less than 0	converts the weight value to zero and counts the observation in the total number of observations
missing	excludes the observation from the calculation of mean and standard deviation

To exclude observations that contain negative and zero weights from the calculation of mean and standard deviation, use EXCLNPWGT. Note that most SAS/STAT procedures, such as PROC GLM, exclude negative and zero weights by default.

Tip: When you use the WEIGHT statement, consider which value of the VARDEF= option is appropriate. See VARDEF= on page 1139 and the calculation of weighted statistics in "Keywords and Formulas" on page 1458 for more information.

Note: Prior to Version 7 of the SAS System, the procedure did not exclude the observations with missing weights from the count of observations. △

Results

Missing Values

By default, PROC STANDARD excludes missing values for the analysis variables from the standardization process, and the values remain missing in the output data set. When you specify the REPLACE option, the procedure replaces missing values with the variable's mean or the MEAN= value.

If the value of the WEIGHT variable or the FREQ variable is missing then the procedure does not use the observation to calculate the mean and the standard deviation. However, the observation is standardized.

Output Data Set

PROC STANDARD always creates an output data set that stores the standardized values in the VAR statement variables, regardless of whether you specify the OUT= option. The output data set contains all the input data set variables, including those not standardized. PROC STANDARD does not print the output data set. Use PROC PRINT, PROC REPORT, or another SAS reporting tool to print the output data set.

Statistical Computations

Standardizing values removes the location and scale attributes from a set of data. The formula to compute standardized values is

$$x_i' = \frac{S * (x_i - \overline{x})}{s_x} + M$$

where

x_i'	is a new standardized value
S	is the value of STD=
M	is the value of MEAN=
x_i	is an observation's value
$\overline{x}$	is a variable's mean
s_x	is a variable's standard deviation.

PROC STANDARD calculates the mean ($\overline{x}$) and standard deviation (s_x) from the input data set. The resulting standardized variable has a mean of *M* and a standard deviation of *S*.

If the data are normally distributed, standardizing is also studentizing since the resulting data have a Student's *t* distribution.

Examples

Example 1: Standardizing to a Given Mean and Standard Deviation

Procedure features:

PROC STANDARD statement options:

- MEAN=
- OUT=
- STD=

VAR statement

Other features:

PRINT procedure

This example

- □ standardizes two variables to a mean of 75 and a standard deviation of 5
- □ specifies the output data set
- □ combines standardized variables with original variables
- □ prints the output data set.

Program

The data set SCORE contains test scores for students who took two tests and a final exam. The FORMAT statement assigns the Z*w.d* format to StudentNumber. This format pads right-justified output with 0s instead of blanks. The LENGTH statement specifies the number of bytes to use to store values of Student.

```
options nodate pageno=1 linesize=80 pagesize=60;
data score;
   length Student $ 9;
   input Student $ StudentNumber Section $
         Test1 Test2 Final @@;
   format studentnumber z4.;
   datalines;
Capalleti 0545 1 94 91 87  Dubose    1252 2 51 65 91
Engles    1167 1 95 97 97  Grant     1230 2 63 75 80
Krupski   2527 2 80 69 71  Lundsford 4860 1 92 40 86
Mcbane    0674 1 75 78 72  Mullen    6445 2 89 82 93
Nguyen    0886 1 79 76 80  Patel     9164 2 71 77 83
Si        4915 1 75 71 73  Tanaka    8534 2 87 73 76
;
```

PROC STANDARD uses a mean of 75 and a standard deviation of 5 to standardize the values. OUT= identifies STNDTEST as the data set to contain the standardized values.

```
proc standard data=score mean=75 std=5
              out=stndtest;
```

The VAR statement specifies the variables to standardize.

```
   var test1 test2;
run;
```

PROC SQL joins SCORE and STNDTEST to create a table (COMBINED) that contains standardized and original test scores for each student. Using AS to rename the standardized variables NEW.TEST1 to StdTest1 and NEW.TEST2 to StdTest2 makes the variable names unique.

```
proc sql;
   create table combined as
   select old.student, old.studentnumber,
          old.section,
          old.test1, new.test1 as StdTest1,
          old.test2, new.test2 as StdTest2,
          old.final
   from score as old, stndtest as new
   where old.student=new.student;
```

PROC PRINT prints the COMBINED table. ROUND rounds the standardized values to two decimal places. The TITLE statement specifies a title.

```
proc print data=combined noobs round;
   title 'Standardized Test Scores for a College Course';
run;
```

Output

The data set contains variables with both standardized and original values. StdTest1 and StdTest2 store the standardized test scores that PROC STANDARD computes.

```
                 Standardized Test Scores for a College Course                1

              Student                          Std                Std
   Student    Number     Section    Test1     Test1     Test2    Test2    Final

   Capalleti   0545         1         94      80.54       91     80.86      87
   Dubose      1252         2         51      64.39       65     71.63      91
   Engles      1167         1         95      80.91       97     82.99      97
   Grant       1230         2         63      68.90       75     75.18      80
   Krupski     2527         2         80      75.28       69     73.05      71
   Lundsford   4860         1         92      79.79       40     62.75      86
   Mcbane      0674         1         75      73.40       78     76.24      72
   Mullen      6445         2         89      78.66       82     77.66      93
   Nguyen      0886         1         79      74.91       76     75.53      80
   Patel       9164         2         71      71.90       77     75.89      83
   Si          4915         1         75      73.40       71     73.76      73
   Tanaka      8534         2         87      77.91       73     74.47      76
```

Example 2: Standardizing BY Groups and Replacing Missing Values

Procedure features:

PROC STANDARD statement options:

PRINT
REPLACE

BY statement

Other features:

FORMAT procedure
PRINT procedure
SORT procedure

This example

- calculates Z scores separately for each BY group using a mean of 1 and standard deviation of 0
- replaces missing values with the given mean
- prints the mean and standard deviation for the variables to standardize
- prints the output data set.

Program

PROC FORMAT creates a format to identify birth rates with a character value.

```
options nodate pageno=1 linesize=80 pagesize=60;
proc format;
   value popfmt 1='Stable'
                2='Rapid';
run;
```

Each observation in the LIFEXP data set contains information on 1950 and 1993 life expectancies at birth for 16 nations*. The birth rate for each nation is classified as stable (1) or rapid (2). The nations with missing data obtained independent status after 1950.

```
data lifexp;
   input PopulationRate Country $char14. Life50 Life93 @@;
   label life50='1950 life expectancy'
         life93='1993 life expectancy';
   datalines;
2 Bangladesh     .  53 2 Brazil         51 67
2 China          41 70 2 Egypt          42 60
2 Ethiopia       33 46 1 France         67 77
1 Germany        68 75 2 India          39 59
2 Indonesia      38 59 1 Japan          64 79
2 Mozambique      . 47 2 Philippines    48 64
1 Russia          . 65 2 Turkey         44 66
1 United Kingdom 69 76 1 United States  69 75
;
```

PROC SORT sorts the observations by the birth rate.

* Data are from VITAL SIGNS 1994: The Trends That Are Shaping Our Future by Lester R. Brown, Hal Kane, and David Malin Roodman, eds. Copyright (c) 1994 by Worldwatch Institute. Reprinted by permission of W.W. Norton & Company, Inc.

```
proc sort data=lifexp;
   by populationrate;
run;
```

PROC STANDARD standardizes all numeric variables to a mean of 1 and a standard deviation of 0. REPLACE replaces missing values. PRINT prints statistics.

```
proc standard data=lifexp mean=0 std=1 replace
              print out=zscore;
```

The BY statement standardizes the values separately by birth rate.

```
   by populationrate;
```

The FORMAT statement assigns a format to PopulationRate. The output data set contains formatted values. The TITLE statement specifies a title.

```
   format populationrate popfmt.;
   title1 ’Life Expectancies by Birth Rate’;
run;
```

PROC PRINT prints the standardized values.

```
proc print data=zscore noobs;
   title ’Standardized Life Expectancies at Birth’;
   title2 ’by a Country’’s Birth Rate’;
run;
```

Output

PROC STANDARD prints the variable name, mean, standard deviation, input frequency, and label of each variable to standardize for each BY group.

Life expectancies for Bangladesh, Mozambique, and Russia are no longer missing. The missing values are replaced with the given mean (0).

```
                        Life Expectancies by Birth Rate                        1

---------------------------- PopulationRate=Stable -----------------------------

                              Standard
 Name              Mean       Deviation              N    Label

 Life50       67.400000        1.854724              5    1950 life expectancy
 Life93       74.500000        4.888763              6    1993 life expectancy


----------------------------- PopulationRate=Rapid -----------------------------

                              Standard
 Name              Mean       Deviation              N    Label

 Life50       42.000000        5.033223              8    1950 life expectancy
 Life93       59.100000        8.225300             10    1993 life expectancy
                    Standardized Life Expectancies at Birth                    2
                           by a Country's Birth Rate

              Population
                 Rate       Country           Life50      Life93

                Stable      France           -0.21567     0.51138
                Stable      Germany           0.32350     0.10228
                Stable      Japan            -1.83316     0.92048
                Stable      Russia            0.00000    -1.94323
                Stable      United Kingdom    0.86266     0.30683
                Stable      United States     0.86266     0.10228
                Rapid       Bangladesh        0.00000    -0.74161
                Rapid       Brazil            1.78812     0.96045
                Rapid       China            -0.19868     1.32518
                Rapid       Egypt             0.00000     0.10942
                Rapid       Ethiopia         -1.78812    -1.59265
                Rapid       India            -0.59604    -0.01216
                Rapid       Indonesia        -0.79472    -0.01216
                Rapid       Mozambique        0.00000    -1.47107
                Rapid       Philippines       1.19208     0.59572
                Rapid       Turkey            0.39736     0.83888
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS* ® *Procedures Guide, Version 8,* Cary, NC: SAS Institute Inc., 1999. 1729 pp.

SAS® Procedures Guide, Version 8

ISBN 1–58025–482–9

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, October 1999

The Institute is a private company devoted to the support and further development of its software and related services.