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 34.1 on page 1148 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 34.1  Standardized Test Scores Using PROC STANDARD

<table>
<thead>
<tr>
<th>The SAS System</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Obs</td>
<td>Student</td>
</tr>
<tr>
<td>1</td>
<td>Capalleti</td>
</tr>
<tr>
<td>2</td>
<td>Dubose</td>
</tr>
<tr>
<td>3</td>
<td>Engles</td>
</tr>
<tr>
<td>4</td>
<td>Grant</td>
</tr>
<tr>
<td>5</td>
<td>Krupski</td>
</tr>
<tr>
<td>6</td>
<td>Lundsford</td>
</tr>
<tr>
<td>7</td>
<td>Mcbane</td>
</tr>
<tr>
<td>8</td>
<td>Mullen</td>
</tr>
<tr>
<td>9</td>
<td>Nguyen</td>
</tr>
<tr>
<td>10</td>
<td>Patel</td>
</tr>
<tr>
<td>11</td>
<td>Si</td>
</tr>
<tr>
<td>12</td>
<td>Tanaka</td>
</tr>
</tbody>
</table>

Output 34.2 on page 1149 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 1159.
### 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> variable1 <...<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	EXCLNPWGTT
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.
**EXCLNPWG**

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 1156

**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 1156

**PRINT**

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

*Featured in:* Example 2 on page 1159

**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 1159

**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 1156

**VARDEF=divisor**

specifies the divisor to use in the calculation of variances and standard deviation. Table 34.1 on page 1152 shows the possible values for divisor and the associated divisors.
Table 34.1 Possible Values for VARDEF=

<table>
<thead>
<tr>
<th>Value</th>
<th>Divisor</th>
<th>Formula for Divisor</th>
</tr>
</thead>
<tbody>
<tr>
<td>DF</td>
<td>degrees of freedom</td>
<td>n - 1</td>
</tr>
<tr>
<td>N</td>
<td>number of observations</td>
<td>n</td>
</tr>
<tr>
<td>WDF</td>
<td>sum of weights minus one</td>
<td>((\sum w_i)) - 1</td>
</tr>
<tr>
<td>WEIGHT</td>
<td>sum of weights</td>
<td>(\sum w_i)</td>
</tr>
<tr>
<td>WGT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The procedure computes the variance as \(CSS/\text{divisor}\), where \(CSS\) is the corrected sums of squares and equals \(\sum (x_i - \overline{x})^2\). When you weight the analysis variables, \(CSS\) equals \(\sum w_i (x_i - \overline{x}_w)^2\) 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 \(\sigma^2\), where the variance of the \(i\)th observation is \(\text{var}(x_i) = \sigma^2/w_i\) and \(w_i\) is the weight for the \(i\)th 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 67

**Main discussion:** “Keywords and Formulas” on page 1397

---

**BY Statement**

Calculates standardized values separately for each BY group.

**Main discussion:** “BY” on page 62

**Featured in:** Example 2 on page 1159

BY <DESCENDING> variable1 <...<DESCENDING> variablen><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 64

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 1156
**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 67

**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

<table>
<thead>
<tr>
<th>Weight value...</th>
<th>PROC STANDARD...</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>counts the observation in the total number of observations</td>
</tr>
<tr>
<td>less than 0</td>
<td>converts the weight value to zero and counts the observation in the total number of observations</td>
</tr>
<tr>
<td>missing</td>
<td>excludes the observation from the calculation of mean and standard deviation</td>
</tr>
</tbody>
</table>

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 1151 and the calculation of weighted statistics in “Keywords and Formulas” on page 1397 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.
**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

\[
x_i' = \frac{S \times (x_i - \bar{x})}{s_x} + M
\]

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
- \(\bar{x}\) is a variable's mean
- \(s_x\) is a variable's standard deviation.

PROC STANDARD calculates the mean (\(\bar{x}\)) 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.
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 Zw.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.

```
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.
<table>
<thead>
<tr>
<th>Student</th>
<th>Student Number</th>
<th>Section</th>
<th>Test1</th>
<th>Std Test1</th>
<th>Test2</th>
<th>Std Test2</th>
<th>Final</th>
</tr>
</thead>
<tbody>
<tr>
<td>Capalleti</td>
<td>0545</td>
<td>1</td>
<td>94</td>
<td>80.54</td>
<td>91</td>
<td>80.86</td>
<td>87</td>
</tr>
<tr>
<td>Dubose</td>
<td>1252</td>
<td>2</td>
<td>51</td>
<td>64.39</td>
<td>65</td>
<td>71.63</td>
<td>91</td>
</tr>
<tr>
<td>Engles</td>
<td>1167</td>
<td>1</td>
<td>95</td>
<td>80.91</td>
<td>97</td>
<td>82.99</td>
<td>97</td>
</tr>
<tr>
<td>Grant</td>
<td>1230</td>
<td>2</td>
<td>63</td>
<td>68.90</td>
<td>75</td>
<td>75.18</td>
<td>80</td>
</tr>
<tr>
<td>Krupski</td>
<td>2527</td>
<td>2</td>
<td>80</td>
<td>75.28</td>
<td>69</td>
<td>73.05</td>
<td>71</td>
</tr>
<tr>
<td>Lundsford</td>
<td>4860</td>
<td>1</td>
<td>92</td>
<td>79.79</td>
<td>40</td>
<td>62.75</td>
<td>86</td>
</tr>
<tr>
<td>McBane</td>
<td>0674</td>
<td>1</td>
<td>75</td>
<td>73.40</td>
<td>78</td>
<td>76.24</td>
<td>72</td>
</tr>
<tr>
<td>Mullen</td>
<td>6445</td>
<td>2</td>
<td>89</td>
<td>78.66</td>
<td>82</td>
<td>77.66</td>
<td>93</td>
</tr>
<tr>
<td>Nguyen</td>
<td>0886</td>
<td>1</td>
<td>79</td>
<td>74.91</td>
<td>76</td>
<td>75.53</td>
<td>80</td>
</tr>
<tr>
<td>Patel</td>
<td>9164</td>
<td>2</td>
<td>71</td>
<td>71.90</td>
<td>77</td>
<td>75.89</td>
<td>83</td>
</tr>
<tr>
<td>Si</td>
<td>4915</td>
<td>1</td>
<td>75</td>
<td>73.40</td>
<td>71</td>
<td>73.76</td>
<td>73</td>
</tr>
<tr>
<td>Tanaka</td>
<td>8534</td>
<td>2</td>
<td>87</td>
<td>77.91</td>
<td>73</td>
<td>74.47</td>
<td>76</td>
</tr>
</tbody>
</table>
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.

```plaintext
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.

```plaintext
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
```

* 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 sorts the observations by the birth rate.

```plaintext
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.

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

The BY statement standardizes the values separately by birth rate.

```plaintext
by populationrate;
```

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

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

PROC PRINT prints the standardized values.

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>N</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Life50</td>
<td>67.4000000</td>
<td>1.854724</td>
<td>5</td>
<td>1950 life expectancy</td>
</tr>
<tr>
<td>Life93</td>
<td>74.5000000</td>
<td>4.888763</td>
<td>6</td>
<td>1993 life expectancy</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Mean</th>
<th>Standard Deviation</th>
<th>N</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Life50</td>
<td>42.0000000</td>
<td>5.033223</td>
<td>8</td>
<td>1950 life expectancy</td>
</tr>
<tr>
<td>Life93</td>
<td>59.1000000</td>
<td>8.2253000</td>
<td>10</td>
<td>1993 life expectancy</td>
</tr>
</tbody>
</table>

Standardized Life Expectancies at Birth by a Country's Birth Rate

<table>
<thead>
<tr>
<th>Population Rate</th>
<th>Country</th>
<th>Life50</th>
<th>Life93</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stable</td>
<td>France</td>
<td>-0.21567</td>
<td>0.51138</td>
</tr>
<tr>
<td>Stable</td>
<td>Germany</td>
<td>0.32350</td>
<td>0.10228</td>
</tr>
<tr>
<td>Stable</td>
<td>Japan</td>
<td>-1.83316</td>
<td>0.92048</td>
</tr>
<tr>
<td>Stable</td>
<td>Russia</td>
<td>0.000000</td>
<td>1.94323</td>
</tr>
<tr>
<td>Stable</td>
<td>United Kingdom</td>
<td>0.86266</td>
<td>0.30683</td>
</tr>
<tr>
<td>Stable</td>
<td>United States</td>
<td>0.86266</td>
<td>0.10228</td>
</tr>
<tr>
<td>Rapid</td>
<td>Bangladesh</td>
<td>0.000000</td>
<td>-0.74161</td>
</tr>
<tr>
<td>Rapid</td>
<td>Brazil</td>
<td>1.78812</td>
<td>0.96045</td>
</tr>
<tr>
<td>Rapid</td>
<td>China</td>
<td>-0.19866</td>
<td>1.32518</td>
</tr>
<tr>
<td>Rapid</td>
<td>Egypt</td>
<td>0.000000</td>
<td>0.10942</td>
</tr>
<tr>
<td>Rapid</td>
<td>Ethiopia</td>
<td>-1.78812</td>
<td>-1.59265</td>
</tr>
<tr>
<td>Rapid</td>
<td>India</td>
<td>-0.59604</td>
<td>-0.01216</td>
</tr>
<tr>
<td>Rapid</td>
<td>Indonesia</td>
<td>-0.79472</td>
<td>-0.01216</td>
</tr>
<tr>
<td>Rapid</td>
<td>Mozambique</td>
<td>0.000000</td>
<td>-1.47107</td>
</tr>
<tr>
<td>Rapid</td>
<td>Philippines</td>
<td>1.19208</td>
<td>0.59572</td>
</tr>
<tr>
<td>Rapid</td>
<td>Turkey</td>
<td>0.39736</td>
<td>0.83888</td>
</tr>
</tbody>
</table>