Chapter 7
OUTPUT Statement

Chapter Table of Contents

Chapter 7

OUTPUT Statement

Overview

You can use the OUTPUT statement to save summary statistics in a SAS data set. This information can then be used to create customized reports or to save historical information about a process.

You can use options in the OUTPUT statement to

- specify the statistics to save in the output data set
- specify the name of the output data set
- compute and save percentiles not automatically computed by the CAPABILITY procedure

Getting Started

This section introduces the OUTPUT statement with simple examples that illustrate commonly used options. Complete syntax for the OUTPUT statement is presented in the “Syntax” section on page 237, and advanced examples are given in the “Examples” section on page 243.

Saving Summary Statistics in an Output Data Set

See CAPOUT1
in the SAS/QC
Sample Library

An automobile manufacturer producing seat belts saves summary information in an output data set using the CAPABILITY procedure. The following statements create the data set BELTS, which contains the breaking strengths (STRENGTH) and widths (WIDTH) of a sample of 50 belts:

```
data belts;
   label strength = ’Breaking Strength (lb/in)’
         width    = ’Width in Inches’;
   input strength width @@;
datalines;
1243.51 3.036  1221.95 2.995  1131.67 2.983  1129.70 3.019
1198.08 3.106  1273.31 2.947  1250.24 3.018  1225.47 2.980
1126.78 2.965  1174.62 3.033  1250.79 2.941  1216.75 3.037
1285.30 2.893  1214.14 3.035  1270.24 2.957  1249.55 2.958
1166.02 3.067  1278.85 3.037  1280.74 2.984  1201.96 3.002
1101.73 2.961  1165.79 3.075  1186.19 3.058  1124.46 2.929
1213.62 2.984  1213.93 3.029  1289.59 2.956  1208.27 3.029
1247.48 3.027  1284.34 3.073  1209.09 3.004  1146.78 3.061
1224.03 2.915  1200.43 2.974  1183.42 3.033  1195.66 2.995
1258.31 2.958  1136.05 3.022  1177.44 3.090  1246.13 3.022
1183.67 3.045  1206.50 3.024  1195.69 3.005  1223.49 2.971
1147.47 2.944  1171.76 3.005  1207.28 3.065  1131.33 2.984
1215.92 3.003  1202.17 3.058
;
```

The following statements produce two output data sets containing summary statistics:

```
proc capability data=belts;
   var strength width;
   output out=means    mean=smean wmean;
   output out=strstats mean=smean std=sstd max=smax min=smin;
run;
```

Note that if you specify an OUTPUT statement, you must also specify a VAR statement. You can use multiple OUTPUT statements with a single procedure statement. Each OUTPUT statement creates a new data set. The OUT= option specifies the name of the output data set. In this case, two data sets, MEANS and STRSTATS, are created. See Figure 7.1 for a listing of MEANS and Figure 7.2 for a listing of STRSTATS.

Summary statistics are saved in an output data set by specifying *keyword=names* after the OUT= option. In the preceding statements, the first OUTPUT statement specifies the *keyword* MEAN followed by the *names* SMEAN and WMEAN. The second OUTPUT statement specifies the *keywords* MEAN, STD, MAX, and MIN, for which the *names* SMEAN, SSTD, SMAX, and SMIN are given.

The *keyword* specifies the statistic to be saved in the output data set, and the *names* determine the names for the new variables. The first *name* listed after a keyword contains that statistic for the first variable listed in the VAR statement; the second *name* contains that statistic for the second variable in the VAR statement, and so on.

Thus, the data set MEANS contains the mean of STRENGTH in a variable named SMEAN and the mean of WIDTH in a variable named WMEAN. The data set STRSTATS contains the mean, standard deviation, maximum value, and minimum value of STRENGTH in the variables SMEAN, SSTD, SMAX, and SMIN, respectively.

```
                     Obs      smean       wmean

                      1     1205.75    3.00584
```

Figure 7.1. Listing of the Output Data Set MEANS

```
             Obs      smean        sstd        smax        smin

              1     1205.75    48.3290    1289.59    1101.73
```

Figure 7.2. Listing of the Output Data Set STRSTATS

Saving Percentiles in an Output Data Set

The CAPABILITY procedure automatically computes the 1st, 5th, 10th, 25th, 75th, 90th, 95th, and 99th percentiles for each variable. You can save these percentiles in an output data set by specifying the appropriate keywords. For example, the following statements create an output data set named PCTLSTR containing the 5th and 95th percentiles of the variable STRENGTH:

See CAPOUT1
in the SAS/QC
Sample Library

```
proc capability data=belts;
   var strength;
   output out=pctlstr p5=p5str p95=p95str;
run;
```

The output data set PCTLSTR is listed in Figure 7.3.

```
                     Obs      p5str      p95str

                      1     1126.78    1284.34
```

Figure 7.3. Listing of the Output Data Set PCTLSTR

You can use the PCTLPTS=, PCTLPRE=, and PCTLNAME= options to save percentiles not automatically computed by the CAPABILITY procedure. For example, the following statements create an output data set named PCTLS containing the 20th and 40th percentiles of the variables STRENGTH and WIDTH:

```
proc capability data=belts;
   var strength width;
   output out=pctls pctlpts  = 20 40
                    pctlpre  = S W
                    pctlname = pct20 pct40;
run;
```

The PCTLPTS= option specifies the percentiles to compute (in this case, the 20th and 40th percentiles). The PCTLPRE= and PCTLNAME= options build the names for the variables containing the percentiles. The PCTLPRE= option gives prefixes for the new variables, and the PCTLNAME= option gives a suffix to add to the prefix. Note that if you use the PCTLPTS= specification, you must also use the PCTLPRE= specification. For details on these options, see the "Syntax" section on page 237.

The preceding OUTPUT statement saves the 20th and 40th percentiles of STRENGTH and WIDTH in the variables SPCT20, WPCT20, SPCT40, and WPCT40. The output data set PCTLS is listed in Figure 7.4.

```
Obs     Spct20     Spct40    Wpct20    Wpct40

 1     1165.91    1199.26    2.9595     2.995
```

Figure 7.4. Listing of the Output Data Set PCTLS

Syntax

The syntax for the OUTPUT statement is as follows:

OUTPUT <**OUT=***SAS-data-set*> *keyword=names* < . . . *keyword=names* >
PCTLPTS=*percentiles* **PCTLPRE=** *prefixes* <**PCTLNAME=***suffixes*>**;**

You can use any number of OUTPUT statements in the CAPABILITY procedure. Each OUTPUT statement creates a new data set containing the statistics specified in that statement. When you use the OUTPUT statement, you must also use the VAR statement. In addition, the OUTPUT statement must contain at least one of the following:

- a specification of the form *keyword=names*
- the PCTLPTS= and PCTLPRE= specifications

The components of the OUTPUT statement are described as follows.

keyword=names

specifies the statistics to include in the output data set and gives names to the new variables that contain the statistics. Specify a *keyword* for each desired statistic, an equal sign, and the *names* of the variables to contain the statistic.

In the output data set, the first variable listed after a keyword in the OUTPUT statement contains the statistic for the first variable listed in the VAR statement; the second variable contains the statistic for the second variable in the VAR statement, and so on. The list of *names* following the equal sign can be shorter than the list of variables in the VAR statement. In this case, the procedure uses the *names* in the order in which the variables are listed in the VAR statement. Consider the following example:

```
proc capability noprint;
   var length width height;
   output out=summary mean=mlength mwidth;
run;
```

The variables MLENGTH and MWIDTH contain the means for LENGTH and WIDTH. The mean for HEIGHT is computed by the procedure but is not saved in the output data set. See “Summary of Keywords” on page 240 for tables of available keywords and the statistics they represent. Formulas for selected statistics are provided in the “Details” section on page 242.

OUT=*SAS-data-set*

specifies the name of the output data set. To create a permanent SAS data set, specify a two-level name. See *SAS Language Reference: Dictionary* for more information on permanent SAS data sets. For example, the previous statements create an output data set named SUMMARY. If the OUT= option is omitted, then by default the new data set is named using the DATAn convention.

PCTLPTS=*percentiles*

specifies *percentiles* that are not automatically computed by the procedure. The CAPABILITY procedure automatically computes the 1st, 5th, 10th, 25th, 50th, 75th, 90th, 95th, and 99th percentiles for the data. These can be saved in an output data set using *keyword=names* specifications. The PCTLPTS= option generates additional percentiles and outputs them to a data set; these additional percentiles are not printed.

If you use the PCTLPTS= option, you must also use the PCTLPRE= option to provide a prefix for the new variable names. For example, to create variables that contain the 20th, 40th, 60th, and 80th percentiles of LENGTH, use the following statements:

```
proc capability noprint;
   var length;
   output pctlpts=20 40 60 80 pctlpre=plen;
run;
```

This creates the variables PLEN20, PLEN40, PLEN60, and PLEN80, whose values are the corresponding percentiles of LENGTH. In addition to specifying name prefixes with the PCTLPRE= option, you can also use the PCTLNAME= option to create name suffixes for the new variables created by the PCTLPTS= option.

PCTLPRE=*prefixes*

specifies prefixes used to create variable names for percentiles requested with the PCTLPTS= option. The PCTLPRE= and PCTLPTS= options must be used together.

The procedure generates new variable names using the *prefix* and the percentile values. If the specified percentile is an integer, the variable name is simply the *prefix* followed by the value. For noninteger percentiles, an underscore replaces the decimal point in the variable name, and decimal values are truncated to one decimal place. For example, the following statements create the variables PWID20, PWID33_3, PWID66_6, and PWID80 for the 20th, 33.33rd, 66.67th, and 80th percentiles of WIDTH, respectively:

```
proc capability noprint;
   var width;
   output pctlpts=20 33.33 66.67 80 pctlpre=pwid;
run;
```

If you request percentiles for more than one variable, you should list prefixes in the same order in which the variables appear in the VAR statement. If combining the *prefix* and percentile value results in a name longer than 8 characters, the prefix is truncated so that the variable name is 8 characters. For example, the following statements compute the 80th and 87.5th percentiles for LENGTH and WIDTH and save the new variables PLENGT80, PLEN87_5, PWIDTH80, and PWID87_5 in the output data set:

```
proc capability noprint;
   var length width;
   output pctlpts=80 87.5 pctlpre=plength pwidth;
run;
```

PCTLNAME=*suffixes*

provides name *suffixes* for the new variables created by the PCTLPTS= option. These *suffixes* are appended to the *prefixes* you specify with the PCTLPRE= option, replacing the percentile values that are used as suffixes by default. List the *suffixes* in the same order in which you specify the percentiles. If you specify n *suffixes* with the PCTLNAME= option and m percentile values with the PCTLPTS= option, where $m > n$, the *suffixes* are used to name the first n percentiles, and the default names are used for the remaining $m - n$ percentiles. For example, consider the following statements:

```
proc capability;
   var length width height;
   output pctlpts  = 20 40
          pctlpre  = pl pw ph
          pctlname = twenty;
run;
```

The value TWENTY in the PCTLNAME= option is used for only the first percentile in the PCTLPTS= list. This suffix is appended to the values in the PCTLPRE= option to generate the new variable names PLTWENTY, PWTWENTY, and PHTWENTY, which contain the 20th percentiles for LENGTH, WIDTH, and HEIGHT, respectively. Since a second PCTLNAME= suffix is not specified, variable names for the 40th percentiles for LENGTH, WIDTH, and HEIGHT are generated using the prefixes and percentile values. Thus, the output data set contains the variables PLTWENTY, PL40, PWTWENTY, PW40, PHTWENTY, and PH40.

If combining the prefix you specify with the PCTLPRE= option and the suffix you specify with the PCTLNAME= option results in a name longer than eight characters, the prefix is truncated from the right so that the variable name is exactly eight characters. For example, the following statements add the variables PLENGMED and PWIDTMED to the output data set:

```
proc capability;
   var length width;
   output pctlpts  = 50
          pctlpre  = plength pwidth
          pctlname = med;
run;
```

Summary of Keywords

The following tables list all keywords available in the OUTPUT statement grouped by type. Formulas for selected statistics are given in the “Details” section on page 242.

Table 7.1. Descriptive Statistics

Keyword	Description
KURTOSIS	kurtosis
MAX	largest (maximum) value
MEAN	mean
MEDIAN	median (50th percentile)
MIN	smallest (minimum) value
MODE	most frequent value (if not unique, the smallest mode is used)
N	number of observations on which calculations are based
NMISS	number of missing values
NOBS	number of observations
RANGE	range
SKEWNESS	skewness
STD	standard deviation
SUM	sum
SUMWGT	sum of weights
VAR	variance

Table 7.2. Specification Limits and Related Statistics

Keyword	Description
LSL	lower specification limit
PCTGTR	percent of nonmissing observations greater than the upper specification limit
PCTLSS	percent of nonmissing observations less than the lower specification limit
TARGET	target value
USL	upper specification limit

Table 7.3. Capability Indices and Related Statistics

Keyword	Description
CP	capability index C_p
CPLCL	lower confidence limit for C_p
CPUCL	upper confidence limit for C_p
CPK	capability index C_{pk} (also denoted *CPK*)
CPKLCL	lower confidence limit for C_{pk}
CPKUCL	upper confidence limit for C_{pk}
CPL	capability index *CPL*
CPLLCL	lower confidence limit for CPL
CPLUCL	upper confidence limit for CPL
CPM	capability index C_{pm}
CPMLCL	lower confidence limit for C_{pm}
CPMUCL	upper confidence limit for C_{pm}
CPU	capability index *CPU*
CPULCL	lower confidence limit for CPU
CPUUCL	upper confidence limit for CPU
K	capability index k (also denoted K)

Table 7.4. Quantile Statistics

Keyword	Description
MEDIAN	median (50th percentile)
P1	1st percentile
P5	5th percentile
P10	10th percentile
P90	90th percentile
P95	95th percentile
P99	99th percentile
Q1	lower quartile (25th percentile)
Q3	upper quartile (75th percentile)
QRANGE	interquartile range (Q3−Q1)

Table 7.5. Normality and Signed Rank Test Statistics

Keyword	Description
NORMAL	test statistic for normality
PNORMAL	p-value for normality test
SIGNRANK	signed rank statistic

Details

The CAPABILITY procedure creates an OUT= data set for each OUTPUT statement. The new data set contains an observation for each combination of levels of the variables in the BY statement, or a single observation if you do not specify a BY statement. Thus, the number of observations in the new data set corresponds to the number of groups for which statistics are calculated. The variables in the new data set are as follows:

- variables in the BY statement. The values of these variables match the values in the corresponding BY group in the DATA= data set.
- variables in the ID statement. The values of these variables match those for the first observation in each BY group, or for the first observation in the data set if you do not specify a BY statement.
- variables created by selecting statistics in the OUTPUT statement. The values of the statistics are computed using all the nonmissing data, or statistics are computed for each BY group if you use a BY statement.
- variables created by requesting new percentiles with the PCTLPTS= option. The names of these new variables depend on the values of the PCTLPRE= and PCTLNAME= options.

If the output data set contains a percentile variable or a quartile variable, the percentile definition assigned with the PCTLDEF= option in the PROC CAPABILITY statement is recorded on the output data set label.

The values of variables requested with the statistics keywords CP, CPK, CPL, CPM, CPU, K, PCTGTR, and PCTLSS are missing unless you identify specification limits in a SPEC statement or in a SPEC= data set.

As an alternative to OUT= data sets, you can create an OUTTABLE= data set. The structure of the OUTTABLE= data set may be more appropriate when you are computing summary statistics and capability indices for multiple process variables. See “OUTTABLE= Data Set” on page 32.

Examples

This section provides additional examples of the OUTPUT statement.

Example 7.1. Computing Nonstandard Capability Indices

See CPCPMK in the SAS/QC Sample Library

In recent years, a number of process capability indices that have been proposed in the research literature are gradually being introduced in applications. As shown in this example, you can compute such indices in the DATA step after using the OUTPUT statement in the CAPABILITY procedure to save various summary statistics.

Hardness measurements (in scaled units) for 50 titanium samples are saved as values of the variable HARDNESS in the following SAS data set:

```
data titanium;
   label hardness = 'Hardness Measurement';
   input hardness @@;
datalines;
1.38  1.49  1.43  1.60  1.59
1.34  1.44  1.64  1.83  1.57
1.45  1.74  1.61  1.39  1.63
1.73  1.61  1.35  1.51  1.47
1.46  1.41  1.56  1.40  1.58
1.43  1.53  1.53  1.58  1.62
1.58  1.46  1.26  1.57  1.41
1.53  1.36  1.63  1.36  1.66
1.49  1.55  1.67  1.41  1.39
1.75  1.37  1.36  1.86  1.49
;
```

The target value for hardness is 1.6, and the lower and upper specification limits are 0.8 and 2.4, respectively. The samples are produced by an in-control process, and the measurements are assumed to be normally distributed.

The following statements use the OUTPUT statement to save various descriptive statistics and an estimate of the index C_{pm} in a data set named INDICES:

```
proc capability data=titanium noprint;
   var hardness;
   specs lsl=0.8 target=1.6 usl=2.4;
   output out=indices
      n       = n
      mean    = avg
      std     = std
      var     = var
      lsl     = lsl
      target  = t
      usl     = usl
      pnormal = pnormal
      cpm     = cpm ;
run;
```

In addition to C_{pm}, you want to report an estimate for the index C_{pmk}, which is defined as follows:

$$C_{pmk} = \frac{d - |\mu - m|}{3\sqrt{\sigma^2 + (\mu - T)^2}}$$

where $d = (\mathrm{USL} - \mathrm{LSL})/2$, $m = (\mathrm{USL} + \mathrm{LSL})/2$, and μ and σ are the mean and standard deviation of the normal distribution. Refer to Section 3.6 of Kotz and Johnson (1993). A natural estimator for C_{pmk} is

$$\hat{C}_{pmk} = \frac{d - |\overline{X} - m|}{3\sqrt{\frac{1}{n}\sum_{i=1}^{n}(X_i - T)^2}}$$

The following statements compute this estimate:

```
data indices;
   set indices;
   d    = 0.5*( USL - LSL );
   m    = 0.5*( USL + LSL );
   num  = d - abs( avg - m );
   den  = 3 * sqrt( (n-1)*var/n + (avg-t)*(avg-t) );
   cpmk = num/den;

title 'Capability Analysis of Titanium Hardness';
proc print data=indices noobs;
   var n avg std lsl t usl cpm cpmk pnormal;
run;
```

The results are listed in Output 7.1.1.

Output 7.1.1. Computation of C_{pmk}

```
                          Capability Analysis of Titanium Hardness

 n      avg        std       lsl      t      usl       cpm        cpmk      pnormal

50    1.5212    0.13295     0.8     1.6     2.4     1.72545    1.56713    0.25111
```

Note that the p-value for the Kolmogorov-Smirnov test of normality is 0.27693, indicating that the assumption of normality is justified.

The following statements also compute an estimate of the index C_{pm} using the SPECIALINDICES option:

```
proc capability data=titanium specialindices;
   var hardness;
   specs lsl=0.8 target=1.6 usl=2.4;
run;
```

Output 7.1.2. Computation of C_{pmk} using the SPECIALINDICES option

```
                          Process Capability Indices

                  Index       Value     95% Confidence Limits

                  Cp       2.005745     1.609575     2.401129
                  CPL      1.808179     1.438675     2.175864
                  CPU      2.203311     1.757916     2.646912
                  Cpk      1.808179     1.438454     2.177904
                  Cpm      1.725446     1.410047     2.066027
```

Example 7.2. Approximate Confidence Limits for Cpk

This example illustrates how you can use the OUTPUT statement to compute confidence limits for the capability index C_{pk}.

See CPKCON3 in the SAS/QC Sample Library

You can request the approximate confidence limits given by Bissell (1990) with the keywords CPKLCL and CPKUCL in the OUTPUT statement. However, this is not the only method that has been proposed for computing confidence limits for C_{pk}. Zhang, Stenback, and Wardrop (1990), referred to here as ZSW, proposed approximate confidence limits of the form

$$\widehat{C}_{pk} \pm k\widehat{\sigma}_{pk}$$

where $\widehat{\sigma}_{pk}$ is an estimator of the standard deviation of $\widehat{C}_{pk}$. Equation (8) of ZSW provides an approximation to the variance of $\widehat{C}_{pk}$ from which one can obtain 100γ% confidence limits for C_{pk} as

$$\begin{aligned}
\text{LCL} &= \widehat{C}_{pk}\left[1 - \Phi^{-1}((1-\gamma)/2)\sqrt{\frac{n-1}{n-3} - \frac{(n-1)\Gamma^2((n-2)/2)}{2\Gamma^2((n-1)/2)}}\;\right] \\
\text{UCL} &= \widehat{C}_{pk}\left[1 + \Phi^{-1}(1-(1-\gamma)/2)\sqrt{\frac{n-1}{n-3} - \frac{(n-1)\Gamma^2((n-2)/2)}{2\Gamma^2((n-1)/2)}}\;\right]
\end{aligned}$$

This assumes that $\widehat{C}_{pk}$ is normally distributed. You can also compute approximate confidence limits based on equation (6) of ZSW, which provides an exact expression for the variance of $\widehat{C}_{pk}$.

The following program uses the methods of Bissell (1990) and ZSW to compute approximate confidence limits for C_{pk} for the variable HARDNESS in the data set TITANIUM (see page 243).

```
proc capability data=titanium noprint;
   var hardness;
   specs lsl=0.8 usl=2.4 gamma=0.95;
   output out=summary
       n    = n
       mean = mean
       std  = std
       lsl  = lsl
       usl  = usl
       cpk  = cpk
       cpklcl = cpklcl
       cpkucl = cpkucl
       cpl  = cpl
       cpu  = cpu;

data summary;
   set summary;
   length method $ 16;

method = ’Bissell’;
lcl = cpklcl;
ucl = cpkucl;
output;

method = ’ZSW Equation 6’;
level = 0.95;
aux   = probit( 1 - (1-level)/2 );
zsw = log(0.5*n-0.5)
      + ( 2*(lgamma(0.5*n-1)-lgamma(0.5*n-0.5)) );
zsw = sqrt((n-1)/(n-3)-exp(zsw));
lcl = cpk*(1-aux*zsw);
ucl = cpk*(1+aux*zsw);
output;

method = ’ZSW Equation 8’;
ds = 3*(cpu+cpl)/2;
ms = 3*(cpl-cpu)/2;
f1 = (1/3)*sqrt((n-1)/2)*gamma((n-2)/2)*(1/gamma((n-1)/2));
f2 = sqrt(2/n)*(1/gamma(0.5))*exp(-n*0.5*ms*ms);
f3 = ms*(1-(2*probnorm(-sqrt(n)*ms)));
ex = f1*(ds-f2-f3);
sd = ((n-1)/(9*(n-3)))*(ds**2-(2*ds*(f2+f3))+ms**2+(1/n));
sd = sd-(ex*ex);
sd = sqrt(sd);
lcl = cpk-aux*sd;
ucl = cpk+aux*sd;
output;
run;

title ’Approximate 95% Confidence Limits for Cpk’;
proc print data = summary noobs;
   var method lcl cpk ucl;
run;
```

The results are shown in Output 7.2.1.

Output 7.2.1. Approximate Confidence Limits for C_{pk}

```
                   Approximate 95% Confidence Limits for Cpk

                method              lcl        cpk        ucl

                Bissell           1.43845    1.80818    2.17790
                ZSW Equation 6    1.43596    1.80818    2.18040
                ZSW Equation 8    1.42419    1.80818    2.19217
```

Note that there is fairly close agreement in the three methods.

You can display the confidence limits computed using Bissell's approach on plots produced by the CAPABILITY procedure by specifying the keywords CPKLCL and CPKUCL in the INSET statement.

The following statements also compute an estimate of the index C_{pk} along with approximate limits using the SPECIALINDICES option:

```
proc capability data=titanium specialindices;
   var hardness;
   specs lsl=0.8 usl=2.4 gamma=0.95;
run;
```

Output 7.2.2. Approximate Confidence Limits for C_{pk} using the SPECIALINDICES option

```
                          Process Capability Indices

                  Index       Value     95% Confidence Limits

                  Cp       2.005745     1.609575     2.401129
                  CPL      1.808179     1.438675     2.175864
                  CPU      2.203311     1.757916     2.646912
                  Cpk      1.808179     1.438454     2.177904
                  Cpm      1.725446     1.410047     2.066027
```

The correct bibliographic citation for this manual is as follows: SAS Institute Inc., *SAS/QC® User's Guide, Version 8*, Cary, NC: SAS Institute Inc., 1999. 1994 pp.

SAS/QC® User's Guide, Version 8

ISBN 1–58025–493–4

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.