Chapter 21

The MEANS Procedure

ABSTRACT 365
INTRODUCTION 365
SPECIFICATIONS 366
PROC MEANS Statement 366
   Statistic Keywords Available in the PROC MEANS Statement 368
BY Statement 370
CLASS Statement 370
FREQ Statement 371
ID Statement 371
OUTPUT Statement 371
   Using the MAXID and MINID Options 373
   Example 374
VAR Statement 375
WEIGHT Statement 375
DETAILS 376
   Missing Values 376
   Computer Resources 376
Output Data Set 376
   Variables 376
   Observations 377
Creating the Default _STAT_ Data Set 380
Usage Note: SAS Bit-Testing Feature 381
Printed Output 382
EXAMPLES 382
   Example 1: Univariate Statistics 382
   Example 2: Comparison of MEANS and SUMMARY Output 386

ABSTRACT

The MEANS procedure produces simple univariate descriptive statistics for numeric variables. You can use the OUTPUT statement to request that MEANS output statistics to a SAS data set.

INTRODUCTION

PROC MEANS computes statistics for an entire SAS data set or for groups of observations in the data set. PROC MEANS is nearly identical to PROC SUMMARY. A major difference between the MEANS and SUMMARY procedures is that, by default, PROC SUMMARY produces no printed output; PROC MEANS does.

If you use the CLASS statement, PROC MEANS calculates descriptive statistics separately for groups of observations. If you want to subdivide your data into
groups, use the CLASS statement. Your input data set does not need to be sorted by the class variables. If you have a large data set, you may have to use the BY statement to subdivide your data. However, the BY statement requires that the input data set be sorted according to the BY variables or have an appropriate index. No such restriction is imposed by the CLASS statement.

The MEANS procedure can optionally create one or more SAS data sets containing the statistics calculated. If you want descriptive statistics in a data set and do not require printed output, use the NOPRINT option in the PROC MEANS statement.

Other SAS procedures also compute univariate statistics. Although PROC MEANS is the easiest and most direct descriptive procedure, other procedures provide additional features. See Chapter 1, "SAS Elementary Statistics Procedures," for an overview of statistics procedures contained in this book.

Note: In this discussion, names of statistics refer to the sample estimates of the true parameters. Thus, the term mean refers to the sample mean, variance refers to the sample variance, and so forth.

SPECIFICATIONS

The MEANS procedure is controlled by the following statements:

```plaintext
PROC MEANS <option-list> <statistic-keyword-list>;
   VAR variable-list;
   CLASS variable-list;
   FREQ variable;
   WEIGHT variable;
   ID variable-list;
   BY variable-list;
   OUTPUT <OUT = SAS-data-set> <output-statistic-list>
      <MAXID <(var-1<(id-list-1) >>...var-n<(id-list-n)>>) = name-list>;
      <MINID <(var-1<(id-list-1) >>...var-n<(id-list-n)>>) = name-list>;
```

There is no limit to the number of OUTPUT statements that can accompany a PROC MEANS statement. The options given in PROC MEANS Statement also apply to the PROC SUMMARY statement, and the statements that follow can also be used with the SUMMARY procedure.

PROC MEANS Statement

```plaintext
PROC MEANS <option-list> <statistic-keyword-list>;
```

The PROC MEANS statement starts the procedure. The options shown in Table 21.1 can be specified in the PROC MEANS statement. The options are described in alphabetic order following the table.
Table 21.1 PROC MEANS Statement Options

<table>
<thead>
<tr>
<th>Task</th>
<th>Options</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specify input data set</td>
<td>DATA=</td>
</tr>
<tr>
<td>Select printing options</td>
<td>DESCENDING</td>
</tr>
<tr>
<td></td>
<td>FW=</td>
</tr>
<tr>
<td></td>
<td>MAXDEC=</td>
</tr>
<tr>
<td></td>
<td>NOPRINT</td>
</tr>
<tr>
<td></td>
<td>PRINT</td>
</tr>
<tr>
<td>Specify the minimum ID value</td>
<td>IDMIN</td>
</tr>
<tr>
<td>Treat missing values as valid</td>
<td>MISSING</td>
</tr>
<tr>
<td>Indicate order</td>
<td>ORDER=</td>
</tr>
<tr>
<td></td>
<td>DESCENDING</td>
</tr>
<tr>
<td>Output statistics with highest <em>TYPE</em> value</td>
<td>NWAY</td>
</tr>
<tr>
<td>Specify variance divisor</td>
<td>VARDEF=</td>
</tr>
</tbody>
</table>

DATA=SAS-data-set

names the SAS data set to be analyzed by PROC MEANS. If the DATA= option is omitted, the most recently created SAS data set is used.

DESCENDING

orders the output data set by descending _TYPE_ value (ASCENDING is the default). This causes the overall totals (_TYPE_=0) to be placed at the end of each BY group. This option has no effect if the NWAY option is also specified.

FW=field-width

specifies the field width for PROC MEANS to use in printing each statistic. The default is FW=12.

IDMIN

specifies that the value of the ID variable should be its minimum (rather than its maximum) value for the corresponding observations of the input data set.

MAXDEC=number

gives the maximum number of decimal places (0 to 8) for PROC MEANS to use in printing results. The default is the BEST. width for columnar format, typically about 7. This applies to all statistics except PROBT (labeled Prob>|T| on the printed output), which is determined by the PROBSIG SAS system option.

MISSING

requests that PROC MEANS treat missing values as valid subgroup values for the CLASS variables.

NOPRINT

suppresses printing of all the descriptive statistics. Use the NOPRINT option when the only purpose of using the procedure is to create a new SAS data set.
NWAY specifies that statistics be output for only the observation with the highest TYPE value (highest level of interaction among CLASS variables).

ORDER=DATA
  | EXTERNAL | FORMATTED
  | FREQ
  | INTERNAL
specifies the sorting order for the levels of the classification variables (specified in the CLASS statement). The table below shows how PROC MEANS interprets values of the ORDER= option.

<table>
<thead>
<tr>
<th>Value of ORDER=</th>
<th>Levels Sorted By</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATA</td>
<td>order of appearance in the input data set</td>
</tr>
<tr>
<td>EXTERNAL</td>
<td>FORMATTED</td>
</tr>
<tr>
<td>FREQ</td>
<td>descending frequency count; levels with the most observations come first</td>
</tr>
<tr>
<td>INTERNAL</td>
<td>unformatted value</td>
</tr>
</tbody>
</table>

By default, ORDER=INTERNAL. For FORMATTED and INTERNAL, the sort order is machine-dependent. The ORDER= option does not apply to missing values, which are always ordered first.

PRINT use with PROC SUMMARY to print any of the descriptive statistics. By default, PROC SUMMARY produces no printed output.

VARDEF=DF
  | N
  | WDF
  | WEIGHT | WGT
specifies the divisor to be used in the calculation of variances and covariances. The default value is VARDEF=DF. The values and associated divisors are shown below.

<table>
<thead>
<tr>
<th>Value</th>
<th>Divisor</th>
<th>Formula</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>WGT</td>
<td>sum of weights</td>
</tr>
</tbody>
</table>

**Statistic Keywords Available in the PROC MEANS Statement**

You can request the statistics below by giving their keywords in a PROC MEANS statement. These keywords are also used in the OUTPUT statement, described later in this chapter. The statistics are defined and formulas are given in Chapter 1, "SAS Elementary Statistics Procedures."
The valid keywords and the statistics they represent are as follows:

- **N** the number of observations in the subgroup having nonmissing values for the variable
- **NMISS** the number of observations in the subgroup having missing values for the variable
- **MIN** the minimum value
- **MAX** the maximum value
- **RANGE** the range
- **SUM** the sum
- **SUMWG** the sum of the WEIGHT variable values
- **MEAN** the mean
- **CSS** the corrected sum of squares
- **USS** the uncorrected sum of squares
- **VAR** the variance
- **STD** the standard deviation
- **STDEV** the standard error of the mean
- **CV** the coefficient of variation
- **SKEWNESS** skewness
- **KURTOSIS** kurtosis
- **T** Student's t for testing the hypothesis that the population mean is 0
- **PR** the probability of a greater absolute value for the t-value above.

If no statistics are specifically requested, PROC MEANS prints the variable name, N, MEAN, STD, MIN, and MAX.

An additional statistic, **N Obs**, is printed when you use PROC MEANS with the CLASS statement or PROC SUMMARY with the PRINT option. The **N Obs** statistic is the total number of observations in a given group or subgroup. N Obs is the sum of N and NMISS. The statements below produce **Output 21.1**.

```plaintext
data gains;
  input name $ team $ age $;
cards;
  Alfred blue 6
  Alicia red 5
  Barbara . 5
  Bennett red .
  Carol blue 5
  Carlos blue 6
;run;

proc means nmiss n;
  class team;
run;
```
PROC MEANS needs the CLASS statement to activate the N Obs statistic; PROC SUMMARY does not. In PROC SUMMARY, N Obs is activated by the PRINT option.

**BY Statement**

**BY variable-list;**

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

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

**CLASS Statement**

**CLASS variable-list;**

The CLASS statement assigns the variables used to form subgroups. Class variables may be either numeric or character, but normally each variable has a small number of discrete values or unique levels. The class variable used to produce Output 21.1 is an example. The CLASS statement has an effect on the statistics computed similar to the effect of the BY statement. The differences are in the format of the printed output and in the BY statement sorting requirement.

If you use the CLASS statement with the OUTPUT statement, the additional variables _TYPE_ and _FREQ_ are generated. The _TYPE_ variable indicates which CLASS variables are being taken into account for the observations. The CLASS statement produces all intermediate _TYPE_ observations as well as a single observation showing overall grouping without any class variables being taken into account. The _TYPE_ and _FREQ_ variables are discussed more fully in DETAILS later in this chapter.

You can do analysis on larger data sets using the BY statement because PROC MEANS need not hold all the groups in memory. **Example 1** illustrates the differences in printed output. You can use both the CLASS and BY statements together, each defining a different set of variables.
FREQ Statement

FREQ variable;

When a FREQ statement appears with PROC MEANS, each observation in the input data set is assumed to represent \( n \) observations in the calculation of statistics, where \( n \) is the value of the FREQ variable. If the value of the FREQ variable is less than 1 or is missing, the observation is not used in the calculations. If the value is not an integer, only the integer portion is used.

ID Statement

ID variable-list;

An ID statement can be used with PROC MEANS to include additional variables in the output data set. If your ID statement names only one variable, the value of the ID variable for a given observation in the output data set is the maximum value it has in the corresponding observations of the input data set, unless the IDMIN option is specified in the PROC MEANS statement. When your ID statement includes two or more variables, the maximum value is chosen as if the values of the ID variables were concatenated into one value for each observation. Thus, the maximum value comes from only one of the corresponding observations in the input data set.

OUTPUT Statement

OUTPUT <OUT=SAS-data-set> <output-statistic-list>
   <MAXID<(var-1<(id-list-1)> <...var-n<(id-list-n)>>)>> = name-list;
   <MINID<(var-1<(id-list-1)> <...var-n<(id-list-n)>>)>> = name-list;

The OUTPUT statement requests that PROC MEANS output statistics to a new SAS data set. There is no limit to the number of OUTPUT statements you can use in a single PROC MEANS step. One SAS data set is created for each valid OUTPUT statement. Use options in the OUTPUT statement to name the data set and to include specific variables.

The following options can be specified in the OUTPUT statement:

- MAXID<(var-1<(id-list-1)> <...var-n<(id-list-n)>>)>> = name-list;
- MINID<(var-1<(id-list-1)> <...var-n<(id-list-n)>>)>> = name-list;

allow you to associate lists of identifying variables with the maximum or minimum of different analysis variables, where:

- \( var \) is a variable to be minimized or maximized.
- \( id-list \) is the identifying variable for the minima or maxima.

Here is an example of a valid specification:

\[
\text{minid (height (name) ) = shortest;}
\]

If you use an ID statement, you do not need to specifically identify the \( id-list \) variables. For example, the specification

\[
\text{minid=shortest}
\]

is sufficient.

For a full description of specifying these options, see the next section, Using the MAXID and MINID Options.

OUT=SAS-data-set

specifies the name of an output data set. If you want to create a permanent SAS data set, you must specify a two-level name (see SAS
Language: Reference for more information on permanent data sets). If the
OUT= option is omitted, the new data set is named using the DATAn
naming convention.

output-statistic-list
specifies the statistics you want in the new data set and also names the
variable or variables to contain these statistics. Any of the statistics
available in the PROC MEANS statement can be output by using the
keyword for that statistic in the OUTPUT statement. The keywords are
listed below. Their definitions are listed in Statistic Keywords Available
in the PROC MEANS Statement earlier in this chapter, and detailed
descriptions are given in Chapter 1, “SAS Elementary Statistics
Procedures.”

<table>
<thead>
<tr>
<th>Statistic</th>
<th>New Variable</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>SUMWGT</td>
</tr>
<tr>
<td>NMISS</td>
<td>MEAN</td>
</tr>
<tr>
<td>MIN</td>
<td>CV</td>
</tr>
<tr>
<td>MAX</td>
<td>SKEWNESS</td>
</tr>
<tr>
<td>RANGE</td>
<td>KURTOSIS</td>
</tr>
<tr>
<td>SUM</td>
<td>T</td>
</tr>
<tr>
<td></td>
<td>PRT</td>
</tr>
</tbody>
</table>

The form of the output request determines the name of the new variable con-
taining the statistic. There are three forms for specifying statistics for the output
data set. Either the MAXID or MINID option can be used with any of the following
forms. The various forms are described below.

statistic-keyword=

supplies one (and only one) statistic to be output to the data set. The
statistic in the new data set has the same name as the corresponding
variable in the input data set if you simply follow the keyword with an
equal sign and a semicolon.

For example, say you want the output data set to contain means for
the variables PRE and POST, and the variables containing the means are
also to be called PRE and POST. You can specify the following:

    proc means;
    class teacher;
    var pre post;
    output out=results mean=;
    run;

You cannot use this form for more than one statistic keyword in the
OUTPUT statement. Requests for multiple statistics using this form cause
the output data set to contain two or more variables with the same
name, which is not allowed.

statistic-keyword=name-list;

requests names that differ from the original variable names. These names
are used for the variables in the new data set that contain the statistics.

The first variable name following the equal sign is given to the
corresponding statistic for the first variable in the VAR statement; the
second name is given to the statistic for the second variable in the VAR
statement; and so on.

For example, suppose that you still want to calculate means for the
variables PRE and POST, but this time you want to use different names
for the variables in the output data set. This time you want to list the
values as MEANEX1 and MEANEX2, referring to the first exam and
second exam. In the output data set named RESULTS, the variable containing the PRE mean is MEANEX1; the variable containing the POST mean is MEANEX2.

```plaintext
proc means;
  class teacher;
  var pre post;
  output out=results mean=meanex1 meanex2;
run;
```

`statistic-keyword(variable-list)=name-list;`
requests a statistic for only certain variables in the VAR statement, and gives those statistics new names in the output data set. After the keyword, list in parentheses the variables for which you want the statistic. Then place the names you assign to the variables in the new data set after the equal sign.

For example, suppose you want to output standard deviations for only the POST variable, but you want means for both the PRE and POST variables. The following statements do this:

```plaintext
proc means;
  class teacher;
  var pre post;
  output out=results mean=premean postmean
       std(post)=stdpost;
run;
```

**Caution:** When no new variable names follow the equal sign, the statistics will have the same names as the original variables, as shown in the first example, `statistic-keyword=;`.

Again, omitting the `name-list` for more than one statistic per variable causes duplicate variable names and is not allowed.

The statements in the following example assign new names to the standard deviations for PRE and POST, name the minimum for POST MINPOST, and leave the mean for PRE named PRE.

```plaintext
proc means;
  var pre post;
  output out=results mean(pre)=
       std=stdpre stdpost
       min(post)=minpost;
run;
```

**Using the MAXID and MINID Options**

The MAXID and MINID options allow you to associate lists of identifying variables with the maximum or minimum of different analysis variables. For example, you could rewrite the example from the previous section as

```plaintext
proc means;
  class teacher;
  var pre post;
  output out=results mean=
          maxid(pre(student))=hipre  minid(post(student))=lopost;
run;
```
You do not have to specify an ID statement to specify an id-list, and the variables in id-list can be different from the variables in an ID statement. However, if you omit the id-list variables, PROC MEANS uses all the variables specified in the ID statement.

The name-list is the list of output variable names for all the associated IDs defined by the different var and id-list combinations.

Names from the name-list are assigned to variables in the order in which they appear in the id-list. If there are fewer names in the name-list than there are variables defined (implicitly or explicitly) by the different var and id-list combinations, the remaining output variables use the corresponding names from the input file once the name-list is exhausted. If, for example, you specify the name-list but do not specify a var and id-list combination, such as

minid=(name-list)

then PROC MEANS associates all the variables in the ID statement with each analysis variable. Thus, for each analysis variable, the number of variables defined for the output data set is the same as the number of variables mentioned in the ID statement.

Caution: The flexibility of the syntax makes it possible to mistakenly create several output variables with the same name. If this occurs, only the first of these variables is accessible in the output data set.

Example Suppose you have a data set containing the WEIGHT and HEIGHT of a group of middle school students. You want to find the largest value for HEIGHT and WEIGHT, the name of the person who is the tallest, and the name of the person who is the heaviest. The complete data set is given in Example 1 later in this chapter. The statements below produce Output 21.2.

```plaintext
proc means data=gains;
  var height weight;
  class sex;
  output out=test
    max=maxht maxwht
    maxid(height(name) weight(name))=tallest heaviest;
run;

proc print data=test;
run;
```

Output 21.2 Using PROC MEANS with the MAXID Option

<table>
<thead>
<tr>
<th>SEX</th>
<th>N Obs</th>
<th>Variable</th>
<th>N</th>
<th>Mean</th>
<th>Std Dev</th>
<th>Minimum</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>14</td>
<td>HEIGHT</td>
<td>14</td>
<td>63.8642857</td>
<td>3.8508953</td>
<td>56.5000000</td>
<td>69.5000000</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>WEIGHT</td>
<td>14</td>
<td>97.8642857</td>
<td>9.6857236</td>
<td>84.0000000</td>
<td>114.3000000</td>
</tr>
<tr>
<td>N</td>
<td>24</td>
<td>HEIGHT</td>
<td>23</td>
<td>65.0391304</td>
<td>5.4584996</td>
<td>51.3000000</td>
<td>71.0000000</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>WEIGHT</td>
<td>23</td>
<td>101.1681818</td>
<td>21.8181966</td>
<td>50.5000000</td>
<td>130.5000000</td>
</tr>
</tbody>
</table>
In this example, the output data set has one observation for each sex and the new variables MAXHT (the maximum height), MAXWT (the maximum weight), TALLEST (the name of the tallest person), and HEAVIEST (the name of the heaviest person). Note that only one name is saved for each sex. If multiple observations have the maximum height or weight, only the first one found is saved.

**VAR Statement**

**VAR variable-list;**

Statistics are calculated for each numeric variable listed in the VAR statement. If a VAR statement is not used, all numeric variables in the input data set, except for those listed in BY, CLASS, FREQ, ID, or WEIGHT statements, are analyzed. The results are printed in the order of the variables in the VAR statement.

Note: If you omit the VAR statement with PROC SUMMARY, a simple count of observations is produced.

**WEIGHT Statement**

**WEIGHT variable;**

The WEIGHT statement specifies a numeric variable in the input SAS data set whose values are used to weight each observation. Only one variable can be specified. Both the FREQ and WEIGHT statements can be used. When a WEIGHT statement is specified, PROC MEANS uses the value of the WEIGHT variable, \(w_i\), to calculate a weighted mean \(\bar{x}_w\), a weighted variance \(s_w^2\), and a weighted sum \(\sum w_i x_i\). The sample mean and sample variance are then represented as

\[
\bar{x}_w = \frac{\sum w_i x_i}{\sum w_i}
\]

and

\[
s_w^2 = \frac{\sum w_i (x_i - \bar{x}_w)^2}{d}
\]

where the \(x_i\) values are the variable values and the divisor \(d\) is controlled by the VARDEF= option. The divisor can be \(n-1\) (when VARDEF=DF), \(\sum w_i\) (when VARDEF=WEIGHT), \(n\) (when VARDEF=N), or \(\sum w_i - 1\) (when VARDEF=WDF), where \(n\) is the number of values. VARDEF=DF is the default.

The WEIGHT variable values can be nonintegers. If the value of the WEIGHT variable is less than zero or is missing, a value of zero is assumed for the weight. If you use more than one WEIGHT statement, only the last one specified is used.