Chapter 3

Procedures in SAS

Procedures in SAS have the following format:

    PROC proname <options>;

The most common option is DATA=some_dataset. If you omit the DATA= option SAS uses the last dataset created. Do not omit the DATA statement as it is easy to make mistakes this way.

3.1 Printing datasets using PROC PRINT

Let us return to the FEV dataset that we looked at previously in this class. Download the file 'small_fev.txt' from the website and store it in the usual location. It contains a random subsample of 30 children from the original dataset. Here is the format of the file:

- seqno = case number
- subjid = subject identification number
- age = subject age at time of measurement (years)
- fev = measured FEV (liters per second)
- height = subject height at time of measurement (inches)
- sex = subject sex (1 = male, 2 = female)
- smoke = smoking habits (1 = yes, 2 = no)

Here is the DATA statement to read in the dataset.

    DATA fev_data;
    INFILE "C:\My SAS files\small_fev.txt" FIRSTOBS=2;
INPUT seqno subjid age fev height sex smoke;
RUN;

Now add some lines to the program to print out the dataset

PROC PRINT DATA=fev_data NOOBS;
   TITLE "FEV dataset";

Here DATA=fev_data and NOOBS are options to the procedure PRINT. Remember we can put TITLE anywhere in the program. For this example, it makes most sense to put it here with the PRINT procedure. We will now supply some other statements we can use with the PRINT procedure.

The WHERE statement

The WHERE statement can be used in any procedure when we only want to work with a subset of a dataset (without altering the dataset). The format of the command is

   WHERE condition;

Here condition defines that subset of the data to take. For our example here is the PRINT command to show only the smokers in the dataset (smoke=1)

PROC PRINT DATA=fev_data NOOBS;
   WHERE smoke=1;

The ID Statement

The ID statement replaces the observation number with identifier variables of your choice. Identifier variables are written first on the output window. The format of the command is:

   ID variable-list;

Here is a program to print out the FEV data using the identifier variable subjid;

PROC PRINT DATA=fev_data NOOBS;
   ID subjid;

Note how the subjid variable comes first. Now try
PROC PRINT DATA=fev_data NOOBS;
   ID subjid age;

Here the subjid then age variables appear first in the output window.

**The VAR Statement**

We use the VAR statement to select the variables we wish to PRINT. The format of the command is:

   VAR variable-list;

For example to print only the subjid, fev and age variables

PROC PRINT DATA=fev_data NOOBS;
   VAR subjid fev age;

Note the variables appear in order stated in the VAR statement.

**The SUM Statement**

To produce a columns sum for the variables in the dataset we use the SUM command. The format of the command is:

   SUM variable-list;

Consider this SAS program.

DATA fev_data;
   INFILE "C:\My SAS files\small_fev.txt" FIRSTOBS=2;
   INPUT seqno subjid age fev height sex smoke;

   /* does_smoke is 1 if the child smokes, 0 otherwise. */
   IF (smoke = 1) THEN does_smoke = 1;
   ELSE does_smoke = 0;

   /* is_female is 1 if the child is female, 0 otherwise. */
   IF (sex = 2) THEN is_female = 1;
   ELSE is_female = 0;
Submit the program. What does the SUM command do here?

The BY Statement

Using the BY statement, here is a program to separately print out the smokers and nonsmokers in the dataset.

PROC PRINT DATA=fev_data NOOBS;
   BY smoke;
RUN;

Look at the log window after you have submitted the program. There is an error!

ERROR: Data set WORK.FEV_DATA is not sorted in ascending sequence. The current by-group has smoke = 2 and the next by-group has smoke = 1.
NOTE: The SAS System stopped processing this step because of errors.
NOTE: There were 4 observations read from the data set WORK.FEV_DATA.
NOTE: PROCEDURE PRINT used:
   real time 0.01 seconds
   cpu time 0.01 seconds

The BY statement assumes that the data is sorted in order of the smoke variable. We need to use the PROC SORT command first to do this before the command will work.

3.2 Sorting datasets

We use the procedure SORT to sort the data in increasing or descending order. The format of the procedure is
PROC SORT DATA=data_in OUT=data_out;
   BY variable;

The procedure sorts the dataset 'data_in' in ascending order (smallest to largest) of the variable and stores the sorted dataset as 'data_out'. If you omit the OUT= command SAS stores the sorted dataset in 'data_in'. Hint: Do not omit the OUT= statement – you will make less mistakes this way. To sort the values in descending order (largest to smallest) use the command DESCENDING before the variable:

PROC SORT DATA=data_in OUT=data_out;
   BY DESCENDING variable;

If you use the BY DESCENDING command in PROC SORT, you also need to use BY DESCENDING in the PROC PRINT command.

Here is the full program to summarize the fev dataset by smoke.

DATA fev_data;
   INFILE "C:\My SAS files\small_fev.txt" FIRSTOBS=2;
   INPUT seqno subjid age fev height sex smoke;
PROC SORT DATA=fev_data OUT=fev_data_sorted_by_smoke;
   BY smoke;
PROC PRINT DATA=fev_data_sorted_by_smoke NOOBS;
   BY smoke;
RUN;

Here is another program to sort the data by combinations of the variables smoke and sex.

DATA fev_data;
   INFILE "C:\My SAS files\small_fev.txt" FIRSTOBS=2;
   INPUT seqno subjid age fev height sex smoke;
PROC SORT DATA=fev_data OUT=fev_data_sorted;
   BY smoke sex;
PROC PRINT DATA=fev_data_sorted NOOBS;
   BY smoke sex;
RUN;
3.3 Calculating summary statistics

We use the procedure `MEANS` to produce the summary statistics for the observations in a dataset (i.e., to summarize the rows). The format of the command is

```
PROC MEANS options;
```

Here, the options include the `DATA=` statement as well as statements that determine what summary statistics to show on the output window, as we shall demonstrate. Try this program:

```
DATA fev_data;
   INFILE "C:\My SAS files\small_fev.txt" FIRSTOBS=2;
   INPUT seqno subjid age fev height sex smoke;
PROC MEANS DATA=fev_data;
RUN;
```

By default we get the number of observations (N), the sample mean, the sample standard deviation and minimum and maximum values for each variable in the dataset. As with `PRINT`, we can specify a subset of variables to summarize using the `VAR` command, e.g., to summarize just the `fev` values

```
DATA fev_data;
   INFILE "C:\My SAS files\small_fev.txt" FIRSTOBS=2;
   INPUT seqno subjid age fev height sex smoke;
PROC MEANS DATA=fev_data;
   VAR fev;
RUN;
```

To summarize the data by smoking we use the `BY` command. As with `PRINT` we need to make sure that the data is sorted by the variables we use in the `BY` command e.g., to summarize `fev` by smoking

```
DATA fev_data;
   INFILE "C:\My SAS files\small_fev.txt" FIRSTOBS=2;
   INPUT seqno subjid age fev height sex smoke;
PROC SORT DATA=fev_data OUT=fev_data_smoke;
   BY smoke;
```
PROC MEANS DATA=fev_data_smoke;
   VAR fev;
   BY smoke;
RUN;

We get a more compact summary using the CLASS statement rather than the BY statement. We do not need to sort the data for CLASS to work correctly:

DATA fev_data;
   INFILE "C:\My SAS files\small_fev.txt" FIRSTOBS=2;
   INPUT seqno subjid age fev height sex smoke;
PROC MEANS DATA=fev_data;
   VAR fev;
   CLASS smoke;
RUN;

What about if we want other summary statistics? We can enter the statistics we want to calculate after the DATA= command in PROC MEANS. In this case SAS calculates only the summary statistic that were requested. Suppose we want to calculate the number of observations 'N', the sample mean, the sample standard deviation, the minimum, Q1, median, Q3 and the maximum for the fev values. Here is the program.

DATA fev_data;
   INFILE "C:\My SAS files\small_fev.txt" FIRSTOBS=2;
   INPUT seqno subjid age fev height sex smoke;
PROC MEANS DATA=fev_data N MEAN STDDEV MIN Q1 MEDIAN Q3 MAX;
   VAR fev;
RUN;

Table 3.3 shows some of the summary statistics you can calculate in the MEANS procedure. For a complete list see the online help at http://support.sas.com/onlinedoc/913/getDoc/en/proc.hlp/a000146728.htm.

(Note the spelling of the sample standard deviation command, STDDEV).

For the CLM command we can specify the confidence level 100(1 - α)% by using the ALPHA= option (the default ALPHA is 0.05). Here is a program to calculate a 95% confidence interval for the mean.

DATA fev_data;
Stat 673, Autumn 2008

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Number of observations</td>
</tr>
<tr>
<td>NMISS</td>
<td>Number of missing values (',')</td>
</tr>
<tr>
<td>MIN</td>
<td>Minimum value</td>
</tr>
<tr>
<td>Q1 or P25</td>
<td>First quartile</td>
</tr>
<tr>
<td>MEDIAN or P50</td>
<td>Median</td>
</tr>
<tr>
<td>Q3 or P75</td>
<td>Third quartile</td>
</tr>
<tr>
<td>MAX</td>
<td>Maximum value</td>
</tr>
<tr>
<td>SUM</td>
<td>Sum</td>
</tr>
<tr>
<td>MEAN</td>
<td>Sample mean</td>
</tr>
<tr>
<td>STDDEV</td>
<td>Sample standard deviation</td>
</tr>
<tr>
<td>VAR</td>
<td>Sample variance</td>
</tr>
<tr>
<td>STDERR</td>
<td>Standard error for the mean</td>
</tr>
<tr>
<td>CLM</td>
<td>Two sided confidence interval for the mean</td>
</tr>
<tr>
<td>T</td>
<td>Student’s t statistic</td>
</tr>
<tr>
<td>PROBT</td>
<td>Probability for student’s t distribution</td>
</tr>
</tbody>
</table>

Table 3.1: Some summary statistics available in the MEANS procedure.

```
INFILE "C:\My SAS files\small_fev.txt" FIRSTOBS=2;
INPUT seqno subjid age fev height sex smoke;
PROC MEANS DATA=fev_data N MEAN STDDEV STDERR CLM ALPHA=0.05;
VAR fev;
RUN;
```

Here is the output from the program

```
The MEANS Procedure

Analysis Variable : fev

N  Mean  Std Dev  Std Error  Lower 95% CL for Mean  Upper 95% CL for Mean
---  ------  --------  ---------  ----------------  ----------------
30  2.6443667  0.8170492  0.1491721  2.3392755  2.9494579
```

Check the confidence interval is correct using the other summary statistics.

**Exercise:**
Write a program to summarize all the heights of the children in the small FEV dataset. Then write a program to summarize the female and male children individually using the BY command. What do you conclude? Now try to summarize the heights of the children by their gender AND by whether they smoke or not. Are there any problems in making this summary?

(Hint: you will need to use PROC SORT here).

### 3.4 Using libraries

We have already seen how we can use quotes round a name to create a permanent dataset that is stored on the hard drive (see Section 2.6). A more efficient way especially with long programs is to use libraries. The command LIBNAME declares a folder on the computer where we will store our library of datasets. Each library is given a name, 'libname' (a name of between 1 and 8 characters long), that is easier to refer to that the location of the folder on the computer:

```sas
LIBNAME libref "folder_on_the_computer";
```

For example suppose we wish to create a library for all fev data we create. Using the Windows Explorer create a new folder 'fev' in the folder My Computer → Drive C: → My SAS files. We create a library reference as follows:

```sas
LIBNAME fev "C:\My SAS Files\fev";
```

We can now create the permanent 'fev_data' dataset that shall be stored in the 'fev' library. To refer to a permanent dataset in the 'fev' library use 'fev.name_of_dataset'. Here is a program to create the 'fev_data' dataset that is stored in the 'fev' library.

```sas
/* declare a library called 'fev' that is stored in "C:\My SAS Files\fev" */
LIBNAME fev "C:\My SAS Files\fev";

/* Load the 'fev_data' dataset */
DATA fev.fev_data;
   INFILE "C:\My SAS files\small_fev.txt" FIRSTOBS=2;
   INPUT seqno subjid age fev height sex smoke;

/* Print out contents of the 'fev_data' dataset, stored in library 'fev'. */
PROC PRINT data=fev.fev_data NOOBS;
RUN;
```

Check that the permanent dataset has been created (look in the 'fev' folder in the 'My SAS files' folder on drive C:).