Introduction to SAS: Lecture Two

Shang Xue @ Purdue Statistics

Getting Your Data Set Ready and Checking Your Data

- DATA Step: Getting Your Data Set Ready
- The DATA Statement
- The Input Statement
- Direct Data Entry
- Loading Data From Raw Data Files
- PROC Step: Checking Your Data
- PROC PRINT
- PROC MEANS
- PROC GPLOT

The DATA Statement

The DATA statement tells SAS that you want to read a data file and store it in a SAS data set with a name you specify. The DATA statement starts with the keyword **DATA**:

```
DATA libref.data-set-name;
```

If the library is the temporary library, **libref** could be omitted. For example, to create a temporary dataset named "temp", we use:

```
DATA temp;
```

Creating SAS Data Sets

Data set iteself could be created by:

- Direct data entry (INPUT + DATALINES/CARDs).
- Loading data from raw data files (INPUT + INFILE).
- Modifying existing SAS data sets (not covered in this intro course).

The Inpute Statement

The INPUT statement tells SAS the names of the variables and the column numbers read on a specified line. General form of the INPUT statement:

```
INPUT input-specifications;
```

Input specifications mainly:

- names SAS variables
- identies the variables to as character or numeric
- A $ sign right after the variable name indicates a character variable, otherwise it is numeric.
- Indicate the column numbers for each variable, input statements need not contain column numbers provided there is a space between each variable value on the data line.

Direct Data Entry

To create a SAS data set by direct data entry, we need to do the following:

- Start a DATA step and name the SAS data set being created by a DATA statement.
- Describle how to read the data fields by the INPUT statement.
- Add the DATALINES/CARDS statement, indicating next line contains real data.
- Enter data inline.

General form of the DATALINES/CARDS statement:

```
DATALINES/CARDS;
DATA ENTRY HERE
```

In the data entry part, the lines of data do not end in a semicolon. DATALINES and CARDS are equivalent.

An Example

```
DATA revenue;
INPUT City $ State $ Revenue;
CARDS;
LA           CA  5000
Chicago      IL  3000
Indianapolis IN  2000
Dallas       TX  2500
;
PROC PRINT DATA=revenue;
RUN;
```

PROC PRINT here is used to check the contents of the data set "revenue":

Loading Data From Raw Data Files

To read raw data files into a SAS data set, we need to do the following:

- Start a DATA step and name the SAS data set being created by a DATA statement.
- Add the INFILE statement and specify the location of the raw data file
- Describle how to read the data fields by the INPUT statement

General form of the INFILE statement:

```
INFILE 'location of raw data file';
```

The INFILE command is usually entered immediately after the DATA statement.

Two Examples

```
DATA revenue1;
INFILE 'location of revenue1.dat';
INPUT City $ State $ Revenue;
RUN;
PROC PRINT DATA=revenue1;
RUN;

DATA revenue2;
INFILE 'location of revenue2.dat';
INPUT City $ 1-12 State $ 14-15  Revenue 17-20;
RUN;

PROC PRINT DATA=revenue2;
RUN;
```

PROC PRINT

PROC PRINT can be used to list the contents of a SAS data set. The general form of PROC PRINT:

```
PROC PRINT DATA = data-set-name;
VAR Variable-list
```

Exmaple:

```
PROC PRINT DATA=revenue1;
Var Revenue City;
RUN;
```

Output of PROC PRINT Example:

The SAS System

Obs	Revenue	City
1	5000	LA
2	3000	Chicago
3	2000	Indianap
4	2500	Dallas

PROC MEANS

PROC MEANS computes descriptive statistics for specified variables. The general form of PROC MEANS:

```
PROC MEANS DATA=data-set-name;
VAR Variable list;
```

If the VAR statement is omitted, descriptive statistics for each variable in the data set will be calculated. Example:

```
PROC MEANS DATA=revenue1;
VAR Revenue;
RUN;
```

Output of PROC MEANS Example:

The SAS System

The MEANS Procedure

Analysis Variable : Revenue

N	Mean	Std Dev	Minimum	Maximum
4	3125.00	1314.98	2000.00	5000.00

PROC GPLOT

PROC GPLOT is used to make presentation quality graphs. Here we will see how to make a simple scatter plot using PROG GPLOT. The general form of PROC GPLOT:

```
PROC GPLOT DATA=data-set-name;
PLOT y*x;
```

SYMBOL statement can be added to specifies the interpolation method, the line style specifies, the type of line to draw, and the width option defines the thickness etc. SYMBOL STATEMENTS ARE GLOBAL and will be used by all subsequent graphs. General form of the SYMBOL statement:

```
SYMBOL v=symbol
       cv=symbol color
       i=interpolation type
       l=line style
       ci=line color
       w=thickness;
```

Example of PROG GPLOT

```
PROC GPLOT DATA=revenue1; PLOT Revenue*City; RUN;
DATA HTWT;
INPUT Height Weight;
CARDS;
157 56
165 60
168 62
170 71
171 73
172 75
175 81
181 88
;
PROC GPLOT DATA=HTWT; PLOT Height*Weight;
SYMBOL v=dot cv=red ci=black l=1 i=j w=0.5;
RUN;
```

Output of PROC GPLOT Example