SAS WORKSHOP

Yuexiao Dong

Department of Statistics
Fox School of Business & Management
Temple University

August 24, 2010
Outline

1. Introduction to SAS
2. Using SAS for Windows
3. SAS data handling: Data Step
   - Read data
   - Useful statements
4. SAS Procedures
Outline

1. Introduction to SAS
2. Using SAS for Windows
3. SAS data handling: Data Step
   - Read data
   - Useful statements
4. SAS Procedures
Objectives

This workshop is designed for beginning users who want to get started with the program, and experienced users who want to get to know the new generation of SAS system. Through this workshop, you may

- Get accustomed to the SAS system;
- Understand how SAS processes a program;
- Get familiar with the data handling process;
- Manage data sets in SAS;
- Perform simple statistical analysis.
What is SAS

SAS stands for Statistical Analysis System. It is both a statistical language and a system that performs sophisticated data management and statistical analysis.

SAS is available in multiple computing environments, including Windows, Mac, UNIX, etc.

SAS for Windows will do every task that other editions of SAS do, it is easy to use, and is more powerful.
Availability of SAS 9 for windows

At Temple, SAS 9 is available for...
1. Introduction to SAS
2. Using SAS for Windows
3. SAS data handling: Data Step
   - Read data
   - Useful statements
4. SAS Procedures
Windows display system

- Explorer/Result window, provide control and organization of the SAS objects and system output;
- Editor window, color coding, edit program;
- Log window, diagnostics of the programming syntax and hints for debugging;
- Output window, show results output by SAS procedures, blank if program does not go through correctly;
- Many other windows, such as Library window, Filename window, Options window, Graphic windows, etc;
- Keys window is a road map to SAS windows and gives you shortcuts to different windows;
- Type "keys" in the command box at the upper left corner.
Steps of using SAS

- Entering SAS statements.
  - Each statement can go on several lines, but it MUST end with a semi-colon;
  - Conventions on Windows operations are applicable, cut, paste and save for example;
  - Format of display, cascade or tile under Window option at the menu bar.

- Submitting SAS statements.
- Saving SAS statements.
- Retrieve SAS programs. File option at the menu bar.
- Ending a SAS session.
Processing of a SAS program

- SAS Program
  - Submit
  - SAS Processing
  - Log Output (Charts)
    - Error
    - No Error
- Further SAS Program
  - Re-submit
Outline

1. Introduction to SAS
2. Using SAS for Windows
3. SAS data handling: Data Step
   - Read data
   - Useful statements
4. SAS Procedures
A DATA step consists of a group of statements that reads ASCII text data (in a computer file) or existing SAS data sets in order to create a new SAS data set.

```sas
DATA one;
INPUT x y z @@;
CARDS;
1 2 3 4 5 6 7 8 9
;  
RUN;

PROC PRINT DATA=one;
RUN;
```
This sample SAS program creates a temporary data set using the CARDS statement to read in the in-line data.

- The DATA statement defines a temporary data set;
- The INPUT statement defines the variables and their formats;
- The CARDS statement gives instruction to start reading in the data that follow.
- The RUN statement ends this session of the program and submits for processing;
- Data manipulation must be done in a DATA step and cannot be done in a PROC step.
Read free formatted data

DATA cars1;
INPUT make $ model $ mpg weight price;
CARDS;
AMC Concord 22 2930 4099
AMC Pacer 17 3350 4749
Buick Electra 15 4080 7827
;
PROC PRINT DATA=cars1(obs=1);
RUN;

It is usually a good idea to print the first few cases of your dataset to check that things were read correctly. $ means the previous variable is characteristic.
DATA cars2;
  INPUT make $ 1-5 model $ 6-12 mpg 13-14
     weight 15-18 price 19-22;
CARDS;
AMC  Concord2229304099
AMC  Pacer  1733504749
AMC  Spirit 2226403799
BuickCentury2032504816
BuickElectra1540807827
;
RUN;
Here’s what the data in the file cars3.txt look like:

AMC Concord 22 2930 4099
AMC Pacer 17 3350 4749
AMC Spirit 22 2640 3799
Buick Century 20 3250 4816
Buick Electra 15 4080 7827

To read the data from cars3.txt into SAS, use the following syntax:

DATA cars3;
INFILE "c:\cars3.txt";
INPUT make $ model $ mpg weight price;
RUN;
Here’s what the data in the file cars4.txt look like:

AMC,Concord,22,2930,4099
AMC,Pacer,17,3350,4749
AMC,Spirit,22,2640,3799
Buick,Century,20,3250,4816
Buick,Electra,15,4080,7827

To read the data from cars4.txt into SAS, use the following syntax:

DATA cars4;
INFILE "c:\cars4.txt" delimiter=',';
INPUT make $ model $ mpg weight price;
RUN;
Here’s what the data in the file cars5.txt look like:

AMC  Concord  2229304099
AMC  Pacer    1733504749
AMC  Spirit   2226403799
BuickCentury2032504816
BuickElectra1540807827

To read the data from cars5.txt into SAS, use the following syntax:

DATA cars5;
INFILE "c:\cars5.txt";
INPUT make $ 1-5 model $ 6-12 mpg 13-14 weight 15-18 price 19-22;
RUN;
Previous examples use CARDS or INFILE to create temporary data set. Now we create a permanent data set and store under the library project1.

LIBNAME myproj 'c:\sasworkshop';
DATA myproj.cars1;
SET cars1;
PROC PRINT;
RUN;

A SAS library is like a special SAS pointer to a location where your SAS files are stored. To create a library, we use LIBNAME nickname ‘name-of-directory’; where nickname should begin with either a letter or an underscore and having no more than 8 characters.
DATA temp1;
SET cars1(DROP=make model);
RENAME price=newprice;
PROC PRINT;
RUN;

DATA temp2(KEEP=price);
SET cars1(DROP=make model);
PROC PRINT;
RUN;

DATA temp3;
SET cars1;
IF upcase(make) EQ 'BUICK';
PROC PRINT;
VAR make model;RUN;
LIBNAME myproj 'c:\sasworkshop';
DATA myproj.new;
SET cars1;
IF mpg GE 20 THEN fuelefficiency = 'high';
ELSE fuelefficiency='low';
RUN;

PROC PRINT;
RUN;

Here we create a new variable fuelefficiency. Then we can print the most recently created data set.
DATA high low;
SET myproj.new;
IF fuelefficiency = ’high’ THEN output high;
ELSE output low;
RUN;

The IF-THEN statements draw subsets from the data set by the fuelefficiency variable. We have created two temporary datasets named high and low. What is the print result next?

PROC PRINT;
RUN;
Outline

1. Introduction to SAS
2. Using SAS for Windows
3. SAS data handling: Data Step
   - Read data
   - Useful statements
4. SAS Procedures

Yuexiao Dong
SAS WORKSHOP
A PROCedure step calls a SAS procedure to analyze or process a SAS dataset. The PROC step begins with a PROC statement and ends with a RUN statement. All of the statistical procedures require the input of a SAS data set.

The general syntax for a PROC step is:

```sas
PROC name [DATA=dataset] [options];
[other PROC-specific statements;]
[BY varlist;]
RUN;
```
PROC name [DATA=dataset] [options];
[other PROC-specific statements;]
[BY varlist;]
RUN;

- SAS keywords are in uppercase;
- User-supplied words are in lowercase;
- Options are in brackets [ ]. Note that you do not type the brackets when programming.
A SAS program can contain any number of DATA and PROC steps.

- Once a dataset has been created, it can be processed by any subsequent DATA or PROC step;
- All SAS statements start with a keyword (DATA, INPUT, PROC, etc.) and end with a semicolon;
- Uppercase and lowercase are equivalent, except inside quote marks (sex = ’m’ is not the same as sex = ’M’).
Useful procedures

- **PROC PRINT;**
  check the data being read by SAS;

- **PROC CONTENTS [data = _all_];**
  descriptions of the contents of one or more files;

- **PROC SORT; BY var; RUN;**
  sort the data by the specified variable var;

- **PROC MEANS;**
  compute statistics for an entire SAS data set or for groups of observations in the data set.
PROC UNIVARIATE provides detail on the distribution of one variable.

DATA Measures;
INPUT Diameter @@;
CARDS;
5.501 5.251 5.404 5.366 5.445 5.576 5.607 5.200 5.977 ...;
RUN;
title 'Normal Q-Q Plot for Diameters';
PROC UNIVARIATE data=Measures noprint;
QQPLOT Diameter / normal;
RUN;
PROC UNIVARIATE

Normal Q-Q Plot

Diameter vs. Normal Quantiles
Some references

- http://www.uc.edu/sashtml/stat/index.htm