STA108 Fall 2002 **Prof. Hans-Georg Mueller**

SAS Handout-1

Revised by Ping-Shi Wu

BASIC SAS RULE

- SAS program names end with ".sas"
- SAS statements end with a semicolon
- SAS statements are case insensitive (lower case and Upper case letters are the same)
- Any number of SAS statements can appear on one line
- A SAS statement can be continued from one line to the next, as long as no word is split
- Variables can be a combination of letters, numbers or underscores. (Must start with a letter or underscore)
- Variable names cannot be longer than 32 characters

WRITING AND RUNNING SAS PROGRAM (PC version)

From PC SAS version 6 on, you can start typing in your program in the Program Editor window. When you are finished with the typing, simply click on "Submit" under "Run" (or the little running man icon at the icon bar ); your program will be compiled. The Log window will tell you what SAS has done; if there is any error, it will be displayed in the Log window. Any output will be displayed in the output window.

If there are errors in the program, go back to the program Editor window, and make any necessary corrections.

BASIC STRUCTURE OF A SAS PROGRAM

To help you understand what your codes are doing, you can add comment statements anywhere in SAS. Comments in SAS are enclosed by "/*" and "*/" like the following:

```
/* This is a comment. SAS will not execute this line */
```

or starts with "*", ends with ";"

```
* This is a comment. SAS will not execute this line;
```

In general, there are two parts to a SAS program : the DATA step and the PROC step(s). The data step creates/manipulates the data you will use in the program, the PROC step(s) carry out the desired calculations. We'll talk about that a bit more in the following sections.

Preceding the 2 parts mentioned above, there can be a "option" statement, which mainly affected the output of your program. Some common option statements are :

CENTER|NOCENTER : CENTER centers your output on the page
NOCENTER left justifies your output
DATE|NODATE : with DATE, today's date appear on top of each page;
with NODATE it will not
NUMBER|NONUMBER : controls whether or not a page number appears on each output
LINESIZE=n : controls the maximum length of output size. Usually set at 80

PAGESIZE=n : controls the maximum number of lines per page of output.

For example, you might start your SAS program with an option statement like this :

```
OPTIONS LINESIZE=80 NODATE;
```

THE DATA STEP

The data step usually begins with the statement "data (name)", where (name) is the name you give to the data set (no longer than 32 characters long).

Next there is the "**input** statement", where you indicate how many variables there are in your data set, and how you wan tot name them. If the variable is of *character* type instead of numerical type, put a "**$**" sign after the variable name.

There are two ways to enter data into SAS. You can either type it in yourself, or you can read it from an external file.

If you are typing in the data yourself, type in each observation one line at a time. By default, SAS reads data as columns, not as rows. Before you start typing in your data, add the line "cards;". You won't need a separate semi-colon at the end of each line of observation, one semi-colon at the end of the data set will do the job.

If you are reading the data from an external file, your command will be: "**INFILE** 'full path of filename';" See the tutoring on course website

As an example, here is a how SAS will read a small data set :

```
OPTIONS LINESIZE=80 NODATE;

DATA church;
   INPUT type $ height length;
CARDS;
G 100 519
G 75 225
G 52 300
G 62 418
G 68 409
R 83 407
R 80 451
R 70 551
R 76 530
R 74 547
;

RUN;
```

THE PROCEDURE STEPS

There are a few procedures that you'll find useful in this course. The most simple one is "proc print;". it will print out the entire data set. In our example, the statement "**PROC PRINT** DATA=church; **RUN;**" will print out the entire data set "church". One can also specify which variable(s) to be printed. For example:

```
PROC PRINT DATA=church;
     VAR type height;
RUN;
```

will print only the variables *type* and *height*.

Another common procedure is "**PROC MEANS;**". It prints descriptive statistics of the data set. Like "proc print;", you can specify which variable you want the proc to process. You can also calculate the descriptive statistics by groups. For example:

```
PROC MEANS DATA=church;
     VAR length;
     BY type;
RUN;
```

will print out (separate) descriptive statistics of length for each of the 2 types (G and R).

Next example is the "**PROC CORR;**" procedure. It will print out the Pearson correlation coefficient along with the descriptive statistics for the input numeric variable.

```
PROC CORR DATA=church;
     VAR length height;
RUN;
```

Last, but definitely not least, is the procedure "**PROC REG;**". This is the procedure that calculates linear regression. In its simplest form, it looks something like this :

```
PROC REG DATA=church;
   MODEL length=height;
   PLOT length*height;
RUN;
```

These statements will run a regression analysis, with *length* as the "Y variable" and *height* as the "X variable". A plot of length vs height will also be produced.

There are other more "advanced" feature in **PROC REG;** we will talk about them later in the course.

EXAMPLE

We combine all the codes in the previous section into the SAS program below :
Here, bold uppercased words denotes system keywords; uppercase denotes syntax corresponding to each section (DATA step, or PROCEDURE)

```
/* This is a comment. SAS will not execute this line*/
/* Specify System options*/
OPTIONS LINESIZE=80 NODATE;
/* Tell SAS you will start a DATA section, and the generated dataset */
/* is named church                                                   */
DATA church;
/* Specify input variables and its attribute: type is character var  */
/* while height and weight are numeric var.                          */
   INPUT type $ height length;
/* CARDS tell SAS you will start entering the data values            */
/* each row refers to one observation; each column refers to one var.*/
/* values are separated by space. One space means same as 100 spaces.*/
CARDS;
G 100 519
G 75 225
G 52 300
G 62 418
G 68 409
R 83 407
R 80 451
R 70 551
R 76 530
R 74 547
;
/* This semicolon is required to tell SAS you finish entering data  */
/* RUN tells SAS to start compiling codes and execution.             */
RUN;

/* Tell SAS you would start a printing procedure for data set church*/
PROC PRINT DATA=church;
/* Specify which variables you want to print. If you skip this, SAS */
/* just print all the variables.                                     */
     VAR type height;
RUN;

/* Tell SAS you would start a MEANS procedure for data set church   */
PROC MEANS DATA=church;
/* Specify which NUMERIC variables you want to explore.             */
     VAR length;
/* Specify pivot variable for subgrouping                           */
     BY type;
RUN;

/* Tell SAS you would start a CORR procedure for data set church    */
PROC CORR DATA=church;
/* Specify which set of NUMERIC variables you want to investigate.  */
     VAR length height;
RUN;
```

```
/* Tell SAS you would start a REG procedure for data set church    */
PROC REG DATA=church;
/* Specify your model: response = predictors */
     MODEL length=height;
/* Specify pair of variables you want to plot*/
     PLOT length*height;
/* Tell SAS to plot residuals against corresponding predicted values*/
     PLOT r.*p.;
RUN;
QUIT;
```

The output are printed as follows :

```
/* Result for PROC PRINT; */
```

```
                                The SAS System                                1

                            OBS    TYPE    HEIGHT

                              1     G        100
                              2     G         75
                              3     G         52
                              4     G         62
                              5     G         68
                              6     R         83
                              7     R         80
                              8     R         70
                              9     R         76
                             10     R         74
```

```
/* Result for PROC MEANS; */
```

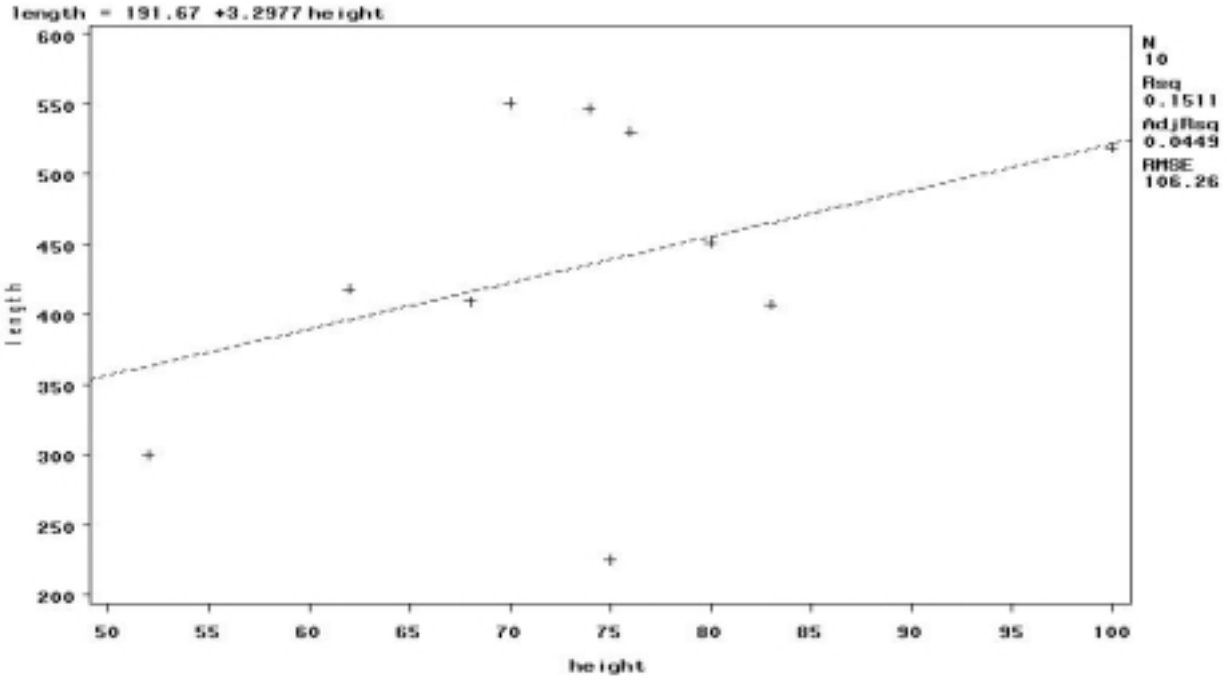
```
                          The SAS System                                2


           Analysis Variable : LENGTH


------------------------------------ TYPE=G -------------------------------------


            N          Mean       Std Dev       Minimum       Maximum
           ------------------------------------------------------------
            5   374.2000000   113.8670277   225.0000000   519.0000000
           ------------------------------------------------------------

------------------------------------ TYPE=R -------------------------------------


            N          Mean       Std Dev       Minimum       Maximum
           ------------------------------------------------------------
            5   497.2000000    64.6544662   407.0000000   551.0000000
           ------------------------------------------------------------
```

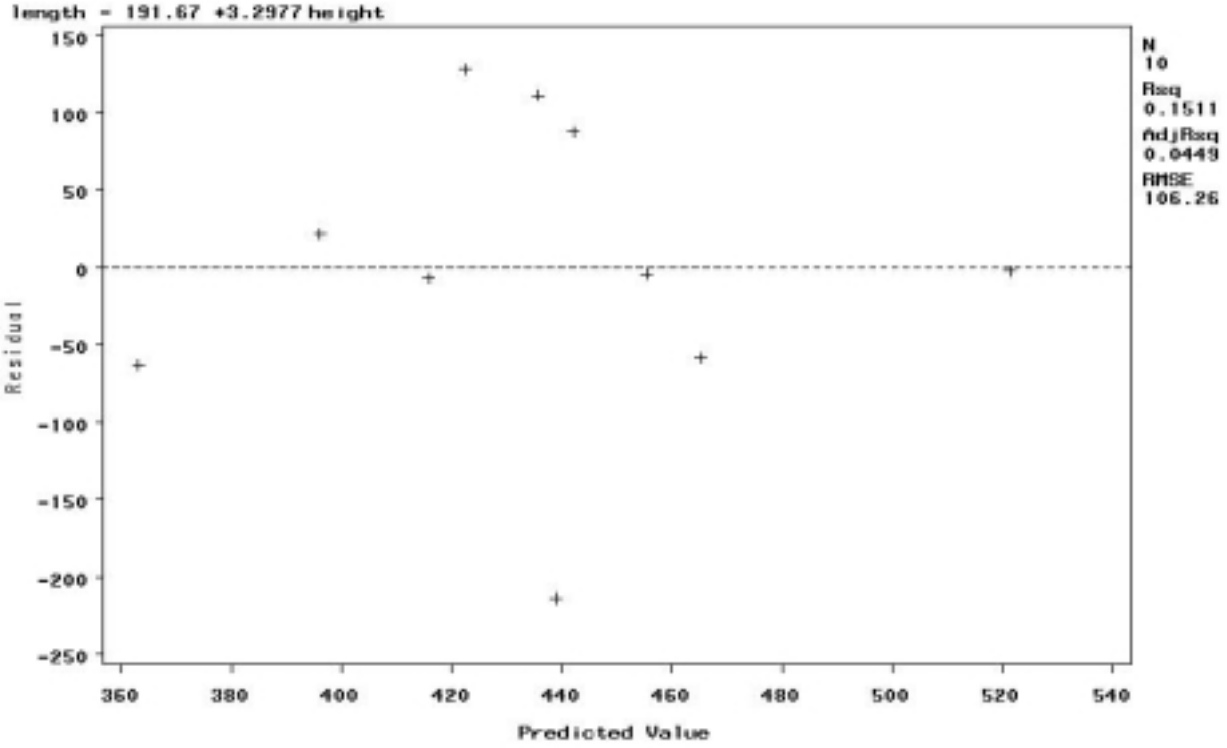
```
/* Result for PROC CORR; */
```

```
                                The SAS System                                 3

                                The CORR Procedure

                         2  Variables:    length   height


                                Simple Statistics

 Variable          N        Mean     Std Dev         Sum     Minimum     Maximum

 length           10   435.70000   108.73316        4357   225.00000   551.00000
 height           10    74.00000    12.81492   740.00000    52.00000   100.00000


                     Pearson Correlation Coefficients, N = 10
                           Prob > |r| under H0: Rho=0

                                      length        height

                         length       1.00000       0.38866
                                                    0.2670

                         height       0.38866       1.00000
                                      0.2670
```

```
/* Result for PROC REG; */
```

```
                                The SAS System                                 4

Model: MODEL1
Dependent Variable: LENGTH

                              Analysis of Variance

                                   Sum of          Mean
          Source           DF     Squares        Square      F Value       Prob>F

          Model             1  16072.98782  16072.98782        1.423       0.2670
          Error             8  90333.11218  11291.63902
          C Total           9 106406.10000

              Root MSE     106.26212     R-square       0.1511
              Dep Mean     435.70000     Adj R-sq       0.0449
              C.V.          24.38883


                              Parameter Estimates

                        Parameter      Standard    T for H0:
        Variable  DF      Estimate         Error   Parameter=0    Prob > |T|

        INTERCEP   1    191.670230  207.27943187         0.925        0.3822
        HEIGHT     1      3.297700    2.76402060         1.193        0.2670
```


length = 191.67 +3.2977 height
N
10
Rsq
0.1511
AdjRsq
0.0449
RMSE
106.26
length
height
600
550
500
450
400
350
300
250
200
50
55
60
65
70
75
80
85
90
95
100


length = 191.67 +3.2977 height
N
10
Rsq
0.1511
AdjRsq
0.0449
RMSE
106.26
Residual
Predicted Value
150
100
50
0
-50
-100
-150
-200
-250
360
380
400
420
440
460
480
500
520
540