CSC 423 Data Analysis and Regression

An Introduction to SAS

SAS is a sophisticated computer package containing many components. The capabilities of the entire package extend far beyond the scope of this course. We will limit our discussions only to those aspects of SAS relevant to carrying out the statistical analyses discussed in class.

The goal of this document is to get you started with SAS and to explain the basic commands of this statistical package.

Description of a PC Windows SAS session

When you start SAS, the *Display Manager Window* should pop out on your screen.

The **Log window** (below the bars) contains the listing of the SAS program that you run. Details concerning the running of the program are printed here, e.g. how much time it took. If the program does not work, SAS prints messages along with the program, that indicate where it founds errors. <u>The Log window should always be examined after running a SAS program.</u>

The **Enhanced Editor window** appears below the Log window. The SAS program can be typed in this window and then be submitted for execution. You can submit a program for processing in several ways:

1) by clicking on the “Submit button” on the task bar;
2) by clicking on “Submit” in the list under the menu command RUN;
3) by using the function key F8 on your keyboard.

(If you are using the Program Editor, once a program is submitted, the Program Editor window empties. If you want to recall the originally submitted program back to this window, you can either: 1) click on “Recall Last Submit” in the list under the menu command RUN; or 2) use the function key F4.)

The **Output window** displays the output from the program, if the program runs successfully.

The **Help window** provides a convenient online help manual. You can open it by clicking on the help button or using the F1 key. Remember this is a great way to get more information about a command or to remind yourself of some detail that you may have partially forgotten.

Computing with SAS Software

A SAS program consists of SAS statements. The statements are organized into groups or **steps**. There are only two types of steps:

1. The DATA step consists of SAS statements that define your data and create a SAS data set.
2. The PROC steps are groups of SAS statements that indicate what kind of statistical analysis to perform.

A few basic rules about SAS:

- <u>All SAS statements end with a semicolon</u> ";". NOTE: Probably the most common error found in SAS programs is the omission of the semicolon.
- SAS statements can be entered in free-format. That is, they can begin in any column; there may be multiple statements per line; you may split a statement over several lines (as long as no word is split.).
- Uppercase and lowercase are equivalent, except inside quote marks (lang = 'c'; is not the same as lang = 'C';).
- Naming Convention:
    1. 32 characters in length (*Note*: v6.12 users: 8 characters in length)
    2. begin with A-Z or _(i.e. underscore)
    3. cannot contain blanks or special symbols (e.g., &, %, $, #, etc.)
- SAS Types:
    1. Character variables (followed by $)
    2. Numeric variables
- *Note:* Missing data: represented by '.' for numeric variables; by ' ' (i.e. space) for character variables.
- Commented text is enclosed by /* and */.

SAS DATA steps:

The general form of the data step is

```
    DATA name;
    Statements;
    DATALINES;
```

Where `name` corresponds to the name we give to the SAS data set.
Suppose we want to enter the following simple table of data into SAS.

Age	Gender	Exam grade	Homework grade
19	F	90	94
20	M	89	90
20	F	78	86
19	M	95	90
21	M	83	85

We have 4 variables (AGE, GENDER, EXAM, HWORK) collected for each of five subjects. The following SAS program will create the SAS data set "GRADES" for the table above.

```
/* This is an example on how to input data into SAS */

TITLE 'Example of data input in SAS';

DATA GRADES;
INPUT AGE GENDER $ EXAM HWORK;
DATALINES;
19 F 90 94
20 M 89 90
20 F 78 86
19 M 95 90
21 M 83 85
;
/* Statements for displaying the dataset GRADES to check if the data
entries are correct */

PROC PRINT DATA=GRADES;
TITLE2 'Listing of data';
TITLE3 'Exam and Homework grades';
    RUN;
```

The output of this statements (generated by PROC PRINT) is

```
              Example of data input in SAS
                    Listing of data
                 Exam and Homework grades

        Obs     AGE     GENDER     EXAM     HWORK
         1       19       F         90        94
         2       20       M         89        90
         3       20       F         78        86
         4       19       M         95        90
         5       21       M         83        85
```

1. **Program Title.** Give the program a title. A SAS program can have up to 10 titles. The second title begins with a Title2 statement, while the third title begins with a Title3 statement, etc. The title text may be enclosed in single or double quotes.

   ```
   TITLE 'Data Cleaning Preview';
   TITLE2  'Your Name, Your Class';
   TITLE3 'Project Name';
   ```

2. **SAS Data set:** Name the data set

   ```
   DATA Grades;
   ```

3. **INPUT statement:** to input data and define variables in SAS. Alphanumeric variables must be followed by $.
   a. If the data are not separated by common delimiters, such as blanks, then "column input" should be used. An example of column input is given above.

      You can also specify the location

      ```
      DATA GRADES;
      INPUT AGE 1-2 GENDER $ 4 EXAM 6-7 HWORK 9-10 ;
      DATALINES;
      19 F 90 94
      20 M 89 90
      20 F 78 86
      19 M 95 90
      21 M 83 85
      ;
      ```

   b. If there is at least one blank space between the variables, then you may use list input. A list of variables in the order that the data are contained in the data file may be used using the syntax:
      ```
      INPUT <var1 var2 ...> @@;
      ```
      For example:
      ```
      DATA Grades;
      INPUT AGE GENDER $ EXAM HWORK @@;
      ```

```
DATALINES;
19 F 90 94 20 M 89 90 20 F 78 86 19 M 95 90 21 M 83 85
;
```

c. To insert data from a file use INFILE. For example:

```
DATA GRADES;
INFILE ‘c:\tmp\grades.dat’;
INPUT AGE 1-2 GENDER $ 4 EXAM 6-7 HWORK 9-10;
```

Options of the INFILE statement are:
The **delimiter** option defines the separator. By default SAS assumes that data are separated by blank spaces. If data are separated by tabs use the option **`delimiter = ‘09’x`**, for commas use **`delimiter = ‘,’`**
MISSOVER = to allow missing data – SAS will skip missing data
FIRSTOBS = 2 declares that data start from second row of the file. By default data will be read starting from the first row.

For example:

```
DATA GRADES;
INFILE ‘c:\tmp\grades.dat’ missover delimiter=’09’x
firstobs=1;
INPUT AGE GENDER $ EXAM HWORK;
```

4. **Labeling variables:** to describe the variables, you can attach labels to variable names.

```
DATA Grades;
INPUT AGE GENDER $ EXAM HWORK @@;
LABEL AGE=’Student’s age’ GENDER=’Student’s sex’ EXAM = ‘Final
exam grade’
     HWORK=’Homework grade’;
DATALINES;
...
```

5. **Data Transformations:** Assignment statements create new variables. The usual arithmetic operations and transformations are available.

```
DATA GRADES;
INPUT AGE GENDER $ EXAM HWORK;
MEANGRADE=(EXAM+HWORK)/2;
AGE_LOG=LOG(AGE);
DATALINES;
...
```

6. **Data manipulation:** You can add or drop variables and/or observations from a dataset.

a. For instance, if you only wanted to consider the data with age less than 21, you could create a dataset GRADES1 from the dataset GRADES previously created as follows:

```
DATA GRADES; SET GRADES1;
IF AGE<21;
```

1. The "DROP" and "KEEP" statements are used to remove variables and all associated values from a dataset:

```
DATA GRADES; SET GRADES1;
DROP AGE;
```

removes age from the dataset and keeps remaining variables.

```
KEEP <variable_name>...;
```

keeps named variables and drops unnamed variables from the dataset.

2. The "IF" statement is used for conditional processing:

```
   IF <expression> THEN <statement1>;
   ELSE <statement2>;
```

Examples:

```
   EX=0;           /* Create EX=0 for all values of EXAM
smaller than 90 */
   IF EXAM >90 EX=1;     /* EX=1 for EXAM>90*/
   IF AGE<21 THEN AGE=.; /* Recode AGE=21 as missing value*/
   IF AGE<21 THEN DELETE;  /* Delete observations for AGE=21*/
```

SAS logical operators are in the table below. You can use either the symbol or the two-letter abbreviation.

Expression	**Symbol**	**Meaning**
EQ NE	= ^=	Equal, Not equal
LT LE	< <=	Less than, Less than or equal
GT GE	> >=	Greater than, Greater than or equal
NOT	^	Negation

7. **Formatting values of variables:** Value labels may be formed as formats. They are listed in a PROC FORMAT. An example of the Proc Format may be found in the program in the following type of syntax:

```
PROC FORMAT;
    VALUE $SX ‘M’=’Male’ ‘F’=’Female’;
  RUN;
```

```
DATA GRADES;
INPUT AGE GENDER $ EXAM HWORK;
FORMAT GENDER $SX.;
DATALINES;
...
```

8. **Assignment statements:** New variables may be constructed with assignment statements:

```
A = 100-EXAM;
```

SAS PROC Steps

The analysis of SAS data sets is performed through PROC steps. A PROC step takes the form

```
PROC <proc_name> DATA=dataset1 [options1];
[Statements / options2]
```

where `proc_name` gives the name of the PROC being used, DATA=dataset1 gives the name of the SAS data set to be analyzed by the procedure. If the DATA=dataset1 is omitted, the most recently created SAS data set is used.

There are many different proc's and each one has different statements and options. Here is a list of procedures that you should be familiar with:

- Correlations among a set of variables:
  ```
  PROC CORR [options];
    [VAR <variable_name>;]
  ```

- Means, standard deviations, and other univariate statistics (N; MEAN; STD; MIN; MAX; SUM; VAR):
  ```
  PROC MEANS [options];
    [VAR <variable_name>...;]
  ```

- Univariate statistics. That is means, standard deviations, median etc. Also provides options to generate a p-value for a normality test and to produce the box plot, stem & leaf and normal plots:
  ```
  PROC UNIVARIATE [options];
   [VAR <variable_name>...;]
  ```

- Print a SAS data set:
  ```
  PROC PRINT [options];
   [VAR <variable_name>...;]
  ```

- Sort a SAS data set according to one or more variables:
  ```
  PROC SORT [options];
   BY <variable_name>...;
  ```

- Plot y aginst x. May be used to create a scatter plot or a residual plot:
  ```
  PROC PLOT [options];
   PLOT <dep_var_name>*<indep_var_name>='*' [options];
  ```

Example of SAS Program

```
/* Example of Program: Data on students’ grades*/

OPTIONS NODATE;     /* Suppress the date that is normally printed in the output*/
TITLE 'Example of SAS Program for CSC323';
PROC FORMAT;
       VALUE $SX 'M'='Male' 'F'='Female';
        RUN;
/* It inputs the data; computes the final score as the average of the exam score and the
homework grade and it assigns a letter grade */

DATA GRADES;
        INPUT AGE GENDER $ EXAM HWORK;
FORMAT GENDER $SX.;
FINAL= (EXAM + HWORK)/2;
IF FINAL < 75 THEN GRADE ='C';
        ELSE IF FINAL >=75 AND FINAL <=85 THEN GRADE = 'B';
        ELSE IF FINAL >=85 THEN GRADE = 'A';
LABEL AGE='Student's age' GENDER='Student's sex' EXAM = 'Final exam grade'
          HWORK='Homework grade';
DATALINES;
19 F 90 94
20 M 89 90
20 F 78 86
19 M 95 90
21 M 83 85
;

/* It lists the student's grades in student's age order*/

PROC SORT DATA=GRADES;
BY AGE;
RUN;

PROC PRINT DATA=GRADES;
TITLE2 "Students' grades in Age order";
ID AGE;
VAR GENDER EXAM HWORK FINAL GRADE;
RUN;

/* It computes the class average for the exam, the homework and the final score */

PROC MEANS DATA=GRADES N MEAN STDDEV STDERR MAXDEC=1;
TITLE2 "Descriptive Statistics";
VAR EXAM HWORK FINAL;
RUN;

/*It computes a frequency count for gender and final grade. */

PROC FREQ DATA=GRADES;
TABLES GENDER GRADE;
RUN;
```

Program Output:

```
                         Example of SAS Program for CSC323
                           Students' grades in Age order

                   AGE    GENDER    EXAM    HWORK    FINAL    GRADE

                    19    Female     90       94      92.0      A
                    19    Male       95       90      92.5      A
                    20    Male       89       90      89.5      A
                    20    Female     78       86      82.0      B
                    21    Male       83       85      84.0      B

                          Example of SAS Program for CSC323
                                Descriptive Statistics

                                 The MEANS Procedure

 Variable    Label               N            Mean         Std Dev      Std Error
 ────────────────────────────────────────────────────────────────────────────────
 EXAM        Final exam grade    5            87.0             6.6            2.9
 HWORK       Homework grade      5            89.0             3.6            1.6
 FINAL                           5            88.0             4.8            2.1
 ────────────────────────────────────────────────────────────────────────────────

                          Example of SAS Program for CSC323
                                Descriptive Statistics

                                  The FREQ Procedure

                                                Cumulative    Cumulative
             GENDER    Frequency     Percent     Frequency      Percent
             ───────────────────────────────────────────────────────────
             Female           2        40.00             2        40.00
             Male             3        60.00             5       100.00


                                                Cumulative    Cumulative
              GRADE    Frequency     Percent     Frequency      Percent
              ──────────────────────────────────────────────────────────
              A               3        60.00             3        60.00
              B               2        40.00             5       100.00
```