Statistical Methods and Computing, 22834/105
Spring 2004, Lab 3
Getting more meaningful output from SAS
Feb. 18, 2004

1 Datasets to download

Please download the datasets billion.dat and gulanick.dat from the course web page after reading their associated info file.

2 Using formats to get SAS to print something other than the values a variable actually contains

We will be using the billionaire dataset again today. Its variable called region contains abbreviations ("A" for Asia, "E" for Europe, etc.). If we want SAS to print out the complete words instead of the abbreviations, so that tables and graphs are more understandable, we need to run a "proc format" before the data step. The data step then refers to the formats defined in the format procedure.

```
options linesize = 75 ;

proc format;
  value $region 'A' = 'Asia' 'E' = 'Europe' 'M' = 'Middle East'
    'O' = 'Other' 'U' = 'Unsure';
run ;
```

Because the original data values are characters rather than numbers, we have to use a dollar sign as the first character in the name of the format.

Note the format statement in the data step below. It tells SAS to apply the format you have defined here to a particular variable. When you use the format statement in a data step, you must put a period at the end of the format name.

3 Using labels to get SAS to print more descriptive variable names

```
data billion ;
  infile 'c:\temp\billion.dat' ;
  input wealth age region $ ;
  format region $region . ;
  label wealth = 'Wealth in Billion $'
    age = 'Age in Years' ;
run ;
```

Now enter and run the following code to see how the formats and labels affect the output of the "print" and "freq" procedures,

```
proc print data = billion (obs = 20);
run ;
```

```
proc print label data = billion (obs = 20);
run ;
```

```
proc freq data = billion ;
tables region ;
run ;
```

4 Formats for numeric variables

Formats can also be used to group numeric data. Add a line to your format procedure and change one line in the data step as follows:

```
options linesize = 75 ;

proc format;
  value $region 'A' = 'Asia' 'E' = 'Europe' 'M' = 'Middle East'
    'O' = 'Other' 'U' = 'Unsure';
value = label 10 <= C <= 50 <= C <= 100 <= C <= 500 <= C <= 1000 <= C <= 10000 <= C <= 100000 <= C <= 1000000 ;
run ;
```

```
data billion ;
infile 'c:\temp\billion.dat' ;
input wealth age region $ ;
  format region $region , wealth $ ;
  label wealth = 'Wealth in Billion $'
    age = 'Age in Years' ;
run ;
```

To see the effect of adding this format:

```
proc print data = billion ;
run ;
```

```
proc freq data = billion ;
tables wealth ;
run ;
```
Using proc tabulate to summarize the distributions of quantitative variables in different groups

Galatrick (Heart and Lung, 1991) studied patients who were recovering from heart surgery. She was interested in whether different combinations of supervised exercise or teaching would affect patients' self-efficacy (or confidence) to perform physical activity.

Patients were randomly assigned to one of three groups: Group 1 received teaching, treadmill exercise testing, and exercise training three times per week. Group 2 received only teaching and exercise testing. Group 3 received only routine care without supervised exercise or teaching. After 4 weeks, each patient was scored on self-efficacy.

Self-efficacy was measured on a continuous scale and scores were assumed to be distributed normally in each of the populations of interest. Her results are in the dataset "galatrick.dat." We wish to produce a table that shows the number of observations and the mean and standard deviation of scores within each of the three groups.

```sas
proc format;
    value genfmt 1 = 'Teaching and Training' 2 = 'Teaching' 3 = 'Neither';
run;

data galan;
    infile 'c:\temp\galatrick.dat';
    input score group;
    format group genfmt.;
run;

proc tabulate data = galan;
    class group; * class statement identifies qualitative variables;
    var score; * var statement identifies quantitative variables;
    tables group , score * (n mean std);
run;
```

Getting SAS to draw a simple random sample from a dataset

The following code will draw a simple random sample of approximately 10% of the observations in the dataset. Putting different numbers in the uniform() function will cause different samples to be drawn. Try this with some number other than 17. Look at your output and record how many observations were in your sample and what sample mean was computed for each of the two variables.

```sas
data billion2;
    infile 'c:\temp\billion.dat';
    input wealth age region $;
    select = uniform(17);
    if select < 0.1;
        format region billion.;
        label wealth = 'Wealth in Billion $';
        age = 'Age in Years';
    run;

    proc univariate data = billion2;
    var wealth age;
```