I. Difference Between a Weight and Freq Statements
   - **FREQ** specifies a numeric variable whose value represents the frequency of the observation. If you use the FREQ statement, the procedure assumes that each observation represents n observations, where n is the value of variable. If n is not an integer, the SAS System truncates it. If n is less than 1 or is missing, the procedure does not use that observation to calculate statistics.
   - The sum of the frequency variable represents the total number of observations. (Therefore should be used in calculations of variance and standard deviation)
   - **WEIGHT** specifies a numeric variable whose values weight the values of the analysis variables. The values of the variable do not have to be integers. If the value of the weight variable is
     - Weight value PROC MEANS
     - 0 .... counts the observation in the total number of observations
     - less than 0 .... converts the value to zero and counts the observation in the total number of observations
     - Missing .... excludes the observation

II. Distributional Functions
   - SAS can generate random observations from discrete and continuous distributions.
     - Binomial (n,p) \( \rightarrow \) ranbin(see,n,p)
     - Exponential (\( \lambda=1 \)) \( \rightarrow \) ranexp(seed)
     - Standard Normal (\( \mu=0; \sigma=1 \)) \( \rightarrow \) rannor(seed)
     - Poisson (mean > 0) \( \rightarrow \) norpoi(seed, mean)
     - Uniform (interval (0,1)) \( \rightarrow \) ranuni(seed)
   - **SEED** is a number used by the random number generator to start the algorithm
     - Can be any **POSITIVE NUMBER**
   - **REMEMBER** these are the STANDARD distributions. Therefore:
     - X \~ Exponential (\( \lambda =4 \))
       - Create a random vector E = exponential (\( \lambda=1 \))
       - \( X = E / 4 \)
     - X \~ Uniform – Interval (3, 4)
       - Create a random vector U = uniform – interval (0,1)
       - \( X = (4 – 3) * U + 3 \)
     - X \~ Random Normal (\( \mu=5, \sigma=8 \))
       - Create a random vector \( N = \) Standard Normal (\( \mu=0, \sigma=1 \))
       - \( X = N * \sigma + \mu \)

III. Do Loops
   - Are a way to repeat a command over and over again, while only having to type it in once.
   - Used in the **DATA STEP** in conjunction with other SAS statements
   - Allow you to create random vectors
   - Begin with **DO**
   - End with **END**
   - Use an **INDEX** variable and the DO LOOP is repeated the number of times as indicated by this variable.

```sas
DATA one;
  X = rannor(2435); *CREATE "X" as a Normal(0,1) variable;
proc print data=one;
run;
```

<table>
<thead>
<tr>
<th>Obs</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.69676</td>
</tr>
</tbody>
</table>
Biometry 700 – Introduction to Clinical Biostatistics

Lab 5

DATA Ex_2;
DO i = 1 to 10; OUTPUT; /* Create INDEX VARIABLE "i" */
END;
PROC PRINT DATA=Ex_2;
RUN;

DATA Ex_3;
DO i = 1 to 10;
x = rannor(2435); /* Creates a variable "X" for each value of "i" */
OUTPUT;
END;
PROC PRINT DATA=Ex_3;
RUN;

• DO – starts the loops and creates ‘i’ as the index variables. Therefore SAS will run through the LOOP for each value created for ‘i’
• OUTPUT – outputs values for the functions that are stated in the loop each time the loop is complete.
• END – Ends the loop … without it SAS will continue in a never ending loop.

• SAS cycles through the index variables (1 → 2 → … 10) at each cycle through the loop it creates a random value from the standard normal distribution.
• Many functions to be carried out on each observation can be placed in the DO LOOP.

Wednesday, September 20, 2000

Obs  i  x
1  1  0.69676
2  2 -0.69530
3  3  0.78178
4  4 -0.93837
5  5  1.41155
6  6 -1.79230
7  7 -2.39990
8  8 -0.71647
9  9  0.64391
10 10 -1.25879

DATA Ex_2;
DO i = 1 to 10; OUTPUT;
END;
PROC PRINT DATA=Ex_2;
RUN;

Wednesday, September 20, 2000

Obs  i
1  1
2  2
3  3
4  4
5  5
6  6
7  7
8  8
9  9
10 10

IV. Proc Plot
• Generates X-Y plots, allowing you to see if 2 variables are related in some way.

BASIC CODE:

PROC PLOT DATA=<data set name>;
BY <variable list>;
PLOT <Y-variable>*<X-variable> <options1> / <options2>;
RUN;

• BY → prepares a graph for each value of the variable listed here
• <Y-variable> → variable to be plotted on the Y-axis (vertical axis)
**Biometry 700 – Introduction to Clinical Biostatistics**

**Lab 5**

- `<X-variable>` → variable to be plotted on the X-axis (Horizontal axis)
- PLOT Y1*(X1 X2 X3) → will produce 3 graphs
  - Y1*X1 ; Y1*X2 and Y1*X3
- OPTIONS1:
  - `variable` → Uses the first character in the variable as the symbol on the graph
  - `character` → Allows you to determine the symbol on the plot
- OPTIONS2:
  - Overlay → allows you to make 2 (or more) graphs and then places them one on top of the other

**EXAMPLES**

```plaintext
PROC PLOT DATA = utility;
   PLOT total*year;
RUN;

PROC PLOT DATA = utility;
   PLOT total*(phone fuel electric);
RUN;

PROC PLOT DATA = utility;
   PLOT total*year='*';
RUN;

PROC PLOT DATA = utility;
   PLOT total*year = quarter;
RUN;

PROC PLOT DATA = utility;
   PLOT phone*year='P' fuel*year='F' electric*year='E' / overlay;
RUN;
```