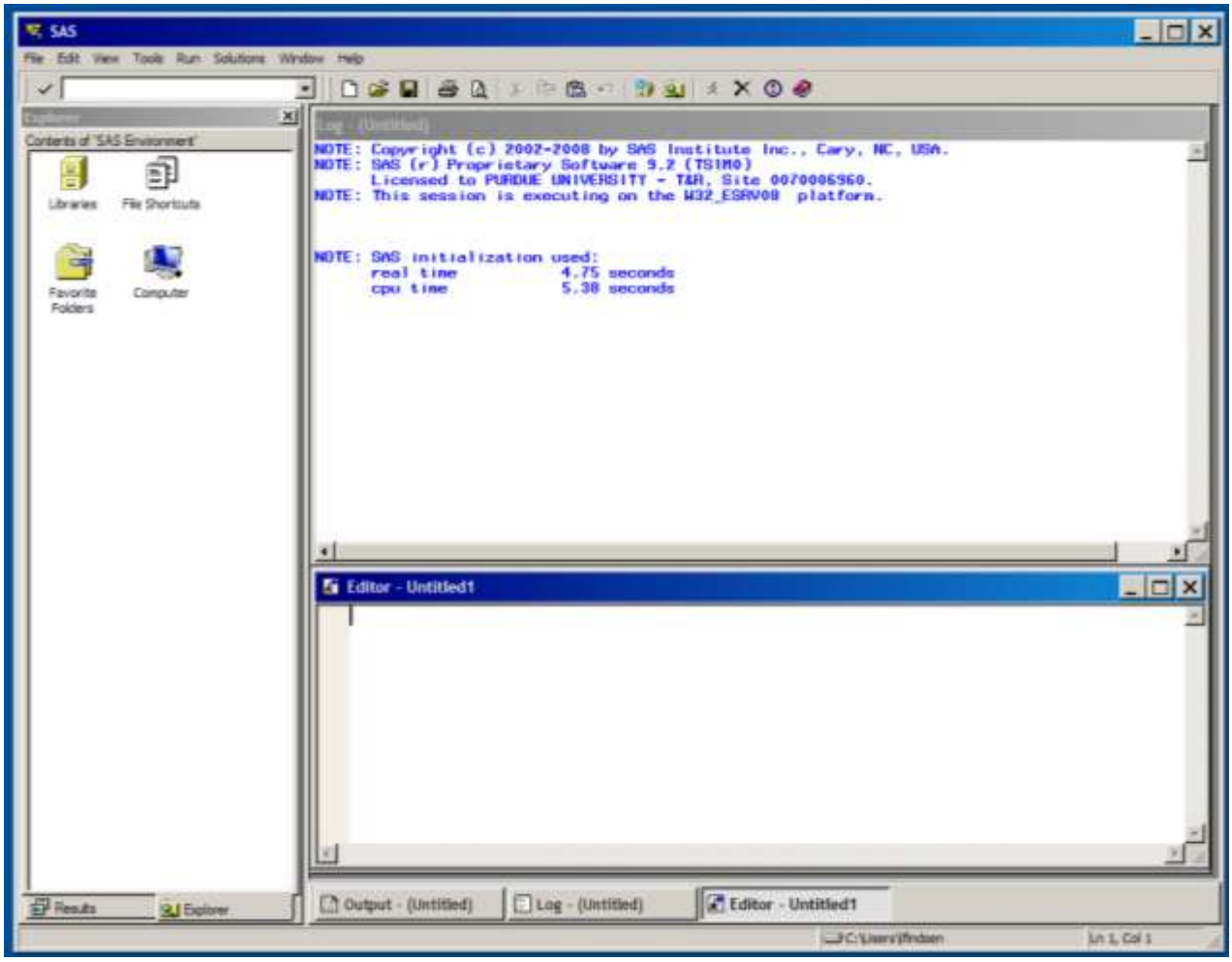
Lab 1: Introduction to SAS/Descriptive Statistics (10 pts.)

Objectives

- **Part 1: Introduction to SAS**
  - **1.1) Get familiar with SAS**
  - **1.2) DATA statement and proc print**
  - **1.3) infiling and log files**
- **Part 2: Descriptive Statistics**
  - **2.1) Histograms**
  - **2.2) Boxplot**
  - **2.3) Numerical Summaries**

Remember:
a) Please put your name, STAT 503 and the lab # on the front of the lab
b) Label each part and put them in logical order.
c) ALWAYS include your SAS code for each problem.

When you first open SAS you will see this window (after you click on the Editor window):



One way to execute the code is go to run → submit. Another way is to click on the  icon when the Editor window is activated.

The program editor window is where you will spend most of your time entering SAS code. In each lab I will provide template SAS coding or "learning code" which can be downloaded from the web site. By studying the code provided and running it in SAS you will see how different pieces of code work and the resulting output they provide in SAS. Below is the first set of learning code.

1.2 DATALINES:

Creating datasets in SAS. Remember to always check to see if you have one or two variables.

SAS Learning code: (a1.sas)
Note: From the SAS editor, the commands are color coded:
blue: commands black: responses green: comments blue green: numbers

```
data a1; /*creates a data set called a1 */
    *Note: data sets have to start with a letter;
input x y @@; /* define two variables x and y, @@ is used here to
  allow more than one observation per line */
    /*Note: If you only wanted to use one variable, the code would be
      input x @@; */
datalines;
1 1 0 8 1 6 0 1 0 1 2 5
0 3 1 0 1 0 1 4 2 4 1 0
0 0 0 1 1 2 1 1 0 4 1 0
1 4 1 0 1 3 0 0 0 1 0 1
1 0 1 1 2 3 0 2 1 4 2 6
2 6 1 0 1 1 0 1 2 8 1 3
1 3 0 5 1 0 5 5 0 2 3 3
0 1 1 0 1 0 0 0 0 3
; /* data input completes here, notice that the semicolon must occupy
  one line by itself */
run; /* indicates that when program is submitted, this is the boundary
  of the data step */

proc print data=a1; run;
 /* This actually sends the dataset to the output window */

quit; /*stops the program */
```

SAS Learning Output

```
The SAS System          10:12 Tuesday, June 5, 2012   2
 Obs    x    y
   1    1    1
   2    0    8
   3    1    6
   4    0    1
   5    0    1
   6    2    5
       ...
  45    1    0
  46    0    0
  47    0    3
```

Problem 1 (1 pts.)
Modify the datalines code (a1.sas) to create a new dataset in SAS using the data from Example 2.2.4 (Table 2.2.4, p. 31) in section 2.2:

```
5 1 6 0 7 2 8 3 9 3 10 9 11 8 12 5 13 3 14 2
```

Name the dataset and variables appropriately for the dataset (see page 31). Points will be taken off if you name the dataset a1 and the variables x and y.
Your submission should consist of your code and dataset (output window).

Problem 2 (1 pts.)
Modify the datalines code (a1.sas) to create a new dataset in SAS using the data from Example 2.2.3 (Table 2.2.3, p. 30) in section 2.2:

```
11.4 44.7 22.6 7.7 18.9 20.9 30.0 24.7 28.6 18.8 11.3 26.5
```

Name the dataset and variable appropriately for the dataset (see page 30). Points will be taken off if you name the dataset a1 and the variable x.
Your submission should consist of your code and dataset (output window).

1.3. INFILING:

An alternate way of creating a dataset is to have SAS copy an existing data file on **your** computer, such as a text or data file. To do this you use an infile statement and direct SAS to the location of the file on your computer. For convenience, I suggest you use your H: drive for storing the datasets. If you don't use your H: drive, use the built in 'Explorer' to determine what the directory your files are in. Remember to always check to see if your inflie has one or two variables.

Additionally, SAS keeps a log of what you have submitted thru the SAS system, all of this is shown in the LOG window in SAS. Here SAS will direct you to errors in your SAS code and often it tries to indicate what it thinks you have done wrong if anything. If your SAS code does not run, your first step is to look at the log file to see what happened.

SAS Learning code: (a2.sas)

```
data a2; /* This creates the same dataset as above but using an infile
  statement */
infile 'H:\a2.dat';
input x y;
run;

proc print data=a2;
run;
```

Example SAS Learning log file:

```
1    data a2; /* This creates the same dataset as above but using an infile statement */
2    infile 'H:\a2.dat';
3    input x y;
4    run;

NOTE: The infile 'H:\a2.dat' is:
      Filename=H:\a2.dat,
      RECFM=V,LRECL=256,File Size (bytes)=272,
      Last Modified=24Aug2009:15:25:26,
      Create Time=24Aug2009:15:25:26

NOTE: 47 records were read from the infile 'H:\a2.txt'.
      The minimum record length was 3.
      The maximum record length was 4.
NOTE: The data set WORK.A1 has 47 observations and 2 variables.
NOTE: DATA statement used (Total process time):
      real time           0.67 seconds
      cpu time            0.07 seconds
```

Problem 3 (2 pts.)

Consider the Serum CK values of 36 men presented in Example 2.2.6 (Table 2.2.6, p. 32) of our text. (dataset Ex2.2.6.dat). Modify the SAS infiling code (a2.sas) to infile this dataset into SAS.

Your submission should consist of your code and the log file. Note: your log file should only be for problem 2 and contain no errors in it.

2.1. HISTOGRAMS:

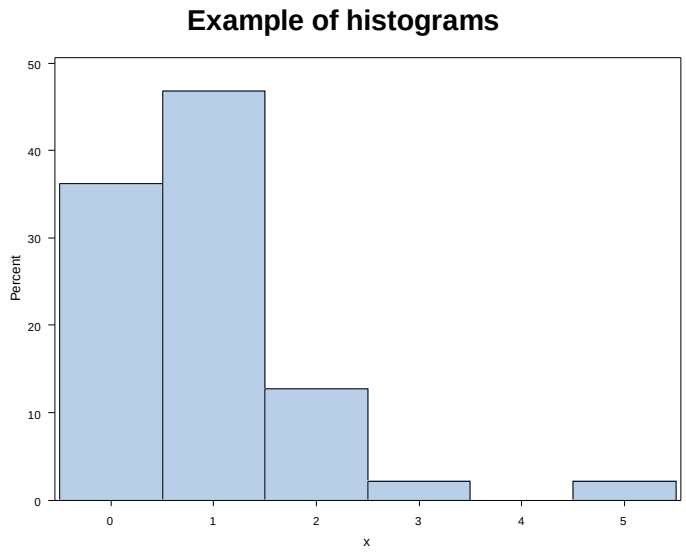
To produce histograms in SAS we use the univariate procedure or “proc univariate”. Procedures are inherent SAS commands which produce specific statistical analysis or output. The univariate procedure is one that handles univariate data, or single variables at a time. Univariate, by default, produces lots of numerical statistics but here we’d only like to use it just to create some histograms. I have added a title to the plot. This should always be done in your labs or homework. Be sure that the title is appropriate for the problem; not just copied from the sample code.

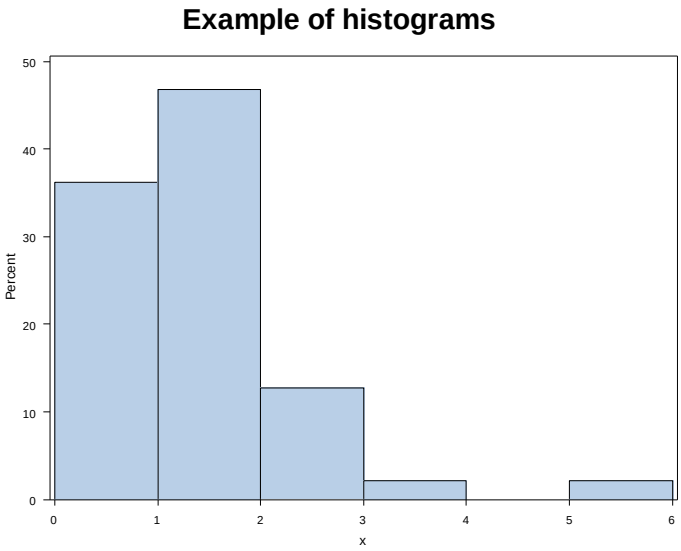
SAS Learning code: (a2a.sas)

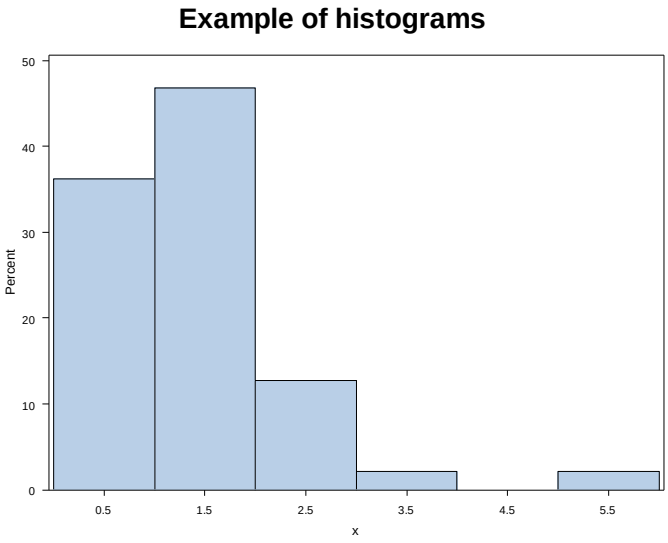
```
data a2;
infile 'H:\a2.dat';
input x y;
run;
proc print data=a2;
run;

title1 'Example of histograms';
proc univariate data=a2 noprint; /* noprint is used to suppress
   summary statistics*/
histogram x; /* produces a histogram for variable x */
histogram x / endpoints = 0 to 6 by 1; /* another histogram for x; by
   1, gives the class width: 0 to 6 gives range for the x-axis  */
histogram x / midpoints = 0.5 to 5.5 by 1; /* yet another histogram
   for x; by 1, gives the class width: -0.5 to 5.5 gives the range for
   the midpoints of each class  */
run;
quit;
```

SAS Learning Output







Problem 4 (2 pts.)

Produce a histogram similar to Fig. 2.2.7 for the same dataset as in Problem 3 (Example 2.2.6).
Note: I am only asking for ONE histogram here. Therefore, you will need to delete (comment out) 2 of the lines in the learning code. Be sure to change the title to something appropriate.
Your submission should consist of your code and histogram.
Note: If you can match the scale in the x-axis of the Figure, you will get 1 bonus point.

2.2. BOXPLOTS:

Creating boxplots in SAS.

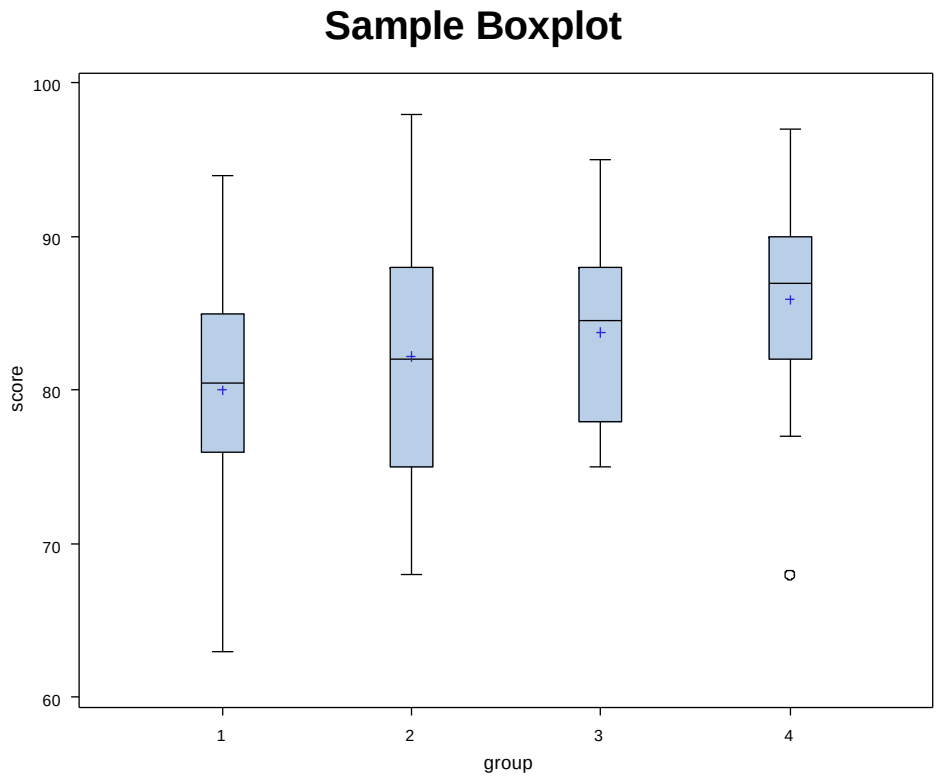
The "proc boxplot" code requires 2 variables. For side-by-side boxplots, the group variable indicates which of the groups the boxplot is in. However, even if you only have one “group", you need to specify a variable for "group". In the output, the ‘+’ indicates the location of the mean. If you want to read in a text variable, place a dollar sign ($) after the variable name. The space before the $ is optional. Always check to see what the order of the variables are when they are read in the ‘input’ line. This does not have to be the same as the order of the variables in the proc boxplot.

SAS Learning code: (a4.sas)

```
data a4;
infile 'H:\a4.dat';
input group $ score;
run;

title1 'Sample Boxplot';
proc boxplot data=a4 ;
plot score*group/boxstyle=schematic idsymbol=circle; /* This creates a
   modified boxplot(s) of the score variable for each group in the
   group variable. Note, if there is only one group it will produce a
   single boxplot, if there are multiple groups it will create side-by-
   side boxplots */
run;
quit;
```

SAS Learning Output



Problem 5 (2 pts.)

Consider the Radish Growth using three different light treatments presented in Example 2.5.3 (p. 55) of our text. (dataset Ex2.5.3.dat). Modify the SAS infiling code code (a4.sas) to produce side-by-side modified boxplots as in Figure 2.5.3. In the data set for this problem, the second variable is text so be sure to include the $ after it. Again, please change the title of the plot.

Your submission should consist of your code and the one figure of the side-by-side boxplots.

BONUS: Problem B1 (1 pts.)

Modify the data file Ex2.5.3.dat so that you produce only a modified boxplot for the radishes grown in light. The plot should be similar to Fig. 2.4.3 (p.51) except it will be vertical instead of horizontal.

Your submission should consist of your code and the boxplot.

2.3 Numerical Summaries

Numerical summaries are often used to describe a set of data. In SAS, they are calculated in proc univariate (which we didn't print out before). In this section, we will be looking at the following values: mean, median, mode, max, min, standard deviation, variance, various percentiles, quartiles, sum of observations, sum of squares (uncorrected SS) and sum of the deviations squared (corrected SS). Note that there is a lot more data printed out then just those values. Some of the other values that are listed, we will be talking about later in the semester. You can also include a title if the output is all text. This will label all of the output after the title. This is useful when you include more than one statement in your code.

SAS Learning code: (a5.sas)

```
data a5;
infile 'H:\a5.dat';
input x;
run;

title1 'Numerical Summaries';
proc univariate data=a5;
run;
quit;
```

SAS Learning Output (complete)

```
                          Numerical Summaries        12:12 Friday, June 8, 2012   2

                               The SAS System
                          The UNIVARIATE Procedure
                                Variable:  x

                                  Moments

N                          39    Sum Weights                39
Mean                 1.44461538    Sum Observations        56.34
Std Deviation        0.18312846    Variance           0.03353603
Skewness             0.69343857    Kurtosis           -0.0753082
Uncorrected SS           82.664    Corrected SS       1.27436923
Coeff Variation      12.6766239    Std Error Mean     0.02932402

                        Basic Statistical Measures

              Location                     Variability

          Mean     1.444615     Std Deviation            0.18313
          Median   1.410000     Variance                 0.03354
          Mode     1.230000     Range                    0.74000
                                Interquartile Range      0.29000
NOTE: The mode displayed is the smallest of 3 modes with a count of 3.
                          Tests for Location: Mu0=0
```

Test	-Statistic-		-----p Value------	
Student's t	t	49.26389	Pr > |t|	<.0001
Sign	M	19.5	Pr >= |M|	<.0001
Signed Rank	S	390	Pr >= |S|	<.0001

Quantiles (Definition 5)

Quantile	Estimate
100% Max	1.92
99%	1.92
95%	1.83
90%	1.72
75% Q3	1.58
50% Median	1.41
25% Q1	1.29
10%	1.23
5%	1.20
1%	1.18
0% Min	1.18

The UNIVARIATE Procedure
Variable: x

Extreme Observations

----Lowest----		----Highest---	
Value	Obs	Value	Obs
1.18	1	1.65	35
1.20	2	1.72	36
1.23	5	1.76	37
1.23	4	1.83	38
1.23	3	1.92	39

Problem 6 (2 pts.)

Consider the Serum CK values of 36 men presented in Example 2.2.6 (Table 2.2.6, p. 32) of our text. (dataset Ex2.2.6.dat). What are the mean, median, standard deviation, range, Q1 and Q3?

Your submission should consist of your code, the relevant output with the values required clearly labeled. Points will be taken off if you submit the complete SAS output and/or don't clearly label the appropriate values.