Lab#2 – Wed, Feb 1, 2012.

Due by the beginning of Friday (Feb 3)’s class.
When turning in labs: (Please hand in the hard copy)
BE SURE TO PUT YOUR NAME, Sec # / CLASS TIME, AND THE LAB # ON YOUR
SUBMISSION. BE SURE TO LABEL EACH PART WITH THE PROBLEM NUMBER AND
PUT THEM IN LOGICAL ORDER. STAPLE ALL PAGES TOGETHER. 1 POINT WILL
BE CUT OFF FOR NOT DOING SO.

Purposes: 1) Creating new variables in a SAS dataset
           2) Boxplots
           3) QQplots

Note: From now on, I will assume you can use a data statement to create SAS datasets. For each
new dataset we work with you will have to infile them into the SAS system before the following
code will work for your datasets. In other words, the code below assumes you've already put
your dataset into SAS.

Creating New Variables in an existing dataset:
You can use the set command in a data statement to copy an existing dataset and create new
variables. The variables can be completely new in which you assign values to them or can be
based off existing variables in your dataset and you can manipulate them using mathematical
functions.

SAS Learning code:

data ex1_17;
infile 'H:\ex1_17.txt';
input lbs;
run;
data new;
set ex1_17; /* This copies the dataset ex1_17 */
group = 1; /* This creates a variable called group in the new
dataset with all values 1 */
kg = lbs/2.2; /* This creates a variable called kg in the new
dataset by dividing the old variable called lbs by 2.2*/
run;
proc print data=new;
run;

Problem 1

Using the dataset from last week, ex1_11.txt, modify the code above to create a new dataset in
SAS that copies last week's dataset and does two things:
1) Creates a group variable for the dataset with the value '3' for all observations.

2) Converts the value in the dataset from miles to kilometers. 1 kilometer = 0.62137 miles. Call the new variable 'km'.

Submit the dataset when you are done.

**BOXPLOTS:**
Creating boxplots in SAS.

Be aware that inherently the boxplot code requires 2 variables. Even if you only have "1 group", you need to specify a variable for "group" even if the value is the same for all observations.

**SAS Learning code:**

```sas
data tests;
infile 'H:\tests.txt';
input group score;
run;
proc boxplot data=tests;
plot score*group; /* This creates 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;
```

**Problem 2**

Using the dataset from last week, modify the code above to create 2 boxplots for the dataset.

1) Make a boxplot from the original values (miles)

2) Make a boxplot for the new values measured in kilometers

Submit both boxplots.

**Problem 3** Using the new dataset for this week called ex9_39.txt, create side by side boxplots for the dataset. Also, write directly on it any similarities or differences between the 3 groups. (symmetry/skewness, variability, outliers)

**QQplots:**
Creating QQplots in SAS.
The QQplot code is a simple command line within proc univariate, which we've already seen. Here we are using it with a by command.

**SAS Learning code:**

```sas
proc univariate data=tests;
by group; /* This breaks up the results from the proc univariate by group, you will get numerical summaries for each group seperately */
qqplot score; /* This creates QQplot(s) for the variable score, note that if you also used the by command then it will give you a QQplot for EACH group */
run;
```

**Problem 4** Using the new dataset for this week called ex9_39.txt, modify the code above to do several things:

1) Get the mean and standard deviation for each group (DO NOT SPAM ME WITH OUTPUT!)

2) Generate a QQplot for each group seperately.

3) Interpret each QQplot seperately indicating to what extent you think each group is normally distributed. You may write directly on each of the 3 QQplots.

**Problem 5** Use the same ex9_39.txt dataset. Assume that the groups are irrelevant and all values represent the same single group. Make one last QQplot for the entire dataset as a single group and interpret it. (HINT: How can I change the code above to ignore groups?)