**X <- 2Lab Assignment 01**

Every week, datasets and a worksheet like this one will be posted on the Blackboard page for POLS 6480.

**I. Objectives**: To introduce the procedures we will use this semester, and to familiarize students with R and RStudio. Students will use the calculator functions of R, create a data frame, load a dataset, and create some plots.

**II. Datasets**: “palmbeach.csv” contains votes for every presidential candidate in the 2000 election, for every county in Florida

**III. Packages**: none

**IV. Preparation**

1) Download data: “palmbeach.csv”

2)

**V. Instructions for Lab 1**

0. R and RStudio are loaded on all the lab computers, but if you wish to install it on your own computer, begin by visiting <http://www.r-project.org/> and following the download link. You should use the program RStudio, a working environment that allows you to see datasets, variables, installed packages, and plots in a single window. You can download the RStudio program from [http://www.rstudio.com](http://www.rstudio.com/) (after a few clicks, you’ll choose the appropriate installer package for your operating system).

1. Double-click the icon for R or select R from the Start menu in Windows. When R opens, you will see a window called “R Console.” This is a *command line interface*, meaning that you type commands that you want to execute and press **Enter**. Some examples:

Multiply 2 times 3 by typing > 2\*3

Find 2 cubed (23) by typing > 2^3

R can be used as a simple calculator for addition, subtraction, multiplication, and division using the standard operators +, -, \*, and /, respectively. You can use parentheses to group operations, and use the ^ symbol to raise a number to a higher power. R also calculates for natural exponent (e.g., > exp(1)) and the natural logarithm (e.g., > log(2.718)). It is straightforward to change the base of a logarithm in R (e.g., > log(50, base = 10)).

2. Using the R console effectively will require storing data by associating values with names, which can then be used in place of numbers to complete calculations. Values can be assigned to variables using one of three operators: <-, =, and ->. You can assign the number 2 to the variable x with any of the following commands (choose one!)

> x <- 2

> x = 2

> 2 -> x

Confirm the assigned value by typing > x

Then find 2 cubed by typing > x^3

Find the square root of 2 by typing > sqrt(x)

3. R stores data in vectors, which are simply ordered lists of numbers stored under a single name. A vector is created by using the c() command; in between the parentheses you will list the values of the variable. For instance, you can create a vector of the first six unique terms of the fibonnacci sequence (each number is the sum of the two numbers preceding it) by typing > fib <- c(1, 2, 3, 5, 8, 13)

Verify that you typed this correctly by typing > fib

Create another vector of the first six odd prime numbers by typing the following

> pri <- c(1, 3, 5, 7, 11, 13)

Suppose you wish to extract just the sixth element of either variable. You can do so by typing:

> fib[6]

> pri[6]

Suppose you wish to know the sum of either list of values. You can add up the values by typing:

> sum(fib)

> sum(pri)

4. R has a more complex data structure, which you will use often, called a data frame. A data frame can be thought of as a spreadsheet: different columns represent the different variables in the data frame, and each row represents the values for a particular unit. Combine your two variables (fib and pri) into a data frame by “binding” the vectors together by typing: > data <- cbind(fib, pri)

Verify that you typed this correctly by typing > data

As soon as you are done playing with R, type > q()

R will shut down after you respond to its prompt (whether to save an image of the workspace).

5. Either double-click the icon for RStudio or select RStudio from the Start menu in Windows. When RStudio opens, you will see three panes:

**Environment, History** lists all the open data frames (under Environment) and the most recent lines that you typed in the Console pane (under History)

**Console** is where you enter commands and can view results, as when you used R earlier.

**Files, Plots, Packages, Help, …** is where any plots you create show up, and where you can quickly install packages.

You can change the layout using the Tools drop-down menu. Select **Options**, and then select **Pane Layout** along the left hand side. This will allow you to decide which of the four quadrants will contain which pane. Note that there is a **Source** pane listed; this will open (pushing **Environment, History** down) when you open a script – which you will do shortly.

6. Now we’re going to work with a dataset. Click on the **Import Dataset** pull down menu on the **Environment** tab. Choose the “From Text File” option, navigate to the directory where you saved “palmbeach.csv”, double-click on that file, and then import it. Be sure to click Yes next to Heading, since the top row in the text file lists the names of the variables.

If I create a directory on my jump drive for all of the POLS 6480 labs, then I can guide RStudio to a particular file within the directory as follows:

palmbeach <- read.csv("J:/POLS 6480/Labs/palmbeach.csv")

The dataset, which should open for you automatically. If it does not, then click on the little icon that looks like a spreadsheet, or type > view(palmbeach)

Examine the variable names and descriptions in the variables window. If you go to the Environment tab, it will tell you that there are 67 obs. (for the 67 counties in Florida) for 27 variables. To see the variable names type > names(palmbeach)

In typing the previous command, I created an R data frame named *palmbeach*, and I also used a built-in command to read the file (which is a comma-delineated file). If the file has a .txt extension instead of a .csv extension, then you will need to use the read.table command instead.

R allows you to have multiple data frames open simultaneously, which is an advantage compared to Stata. However, you will have to tell R which data frame to which you are referring. You can make a data frame active by typing > attach(palmbeach)

Alternatively, you can type *[dataframe]$[variable]* to select a particular variable from a particular data frame.

6. If you remember your history, in 2000 there were concerns that the “Butterfly Ballot” led to voter confusion in Palm Beach county, and many residents of that county incorrectly cast votes for Pat Buchanan rather than Al Gore. To create the percentage of votes cast for Pat Buchanan in each county, go to the **Console** window and type the following:

palmbeach$bs <- 100\*palmbeach$BUCHANAN/palmbeach$VOTES

You have created a new variable, named *bs*, inside the *palmbeach* dataframe. If you look at the Environment tab, it should now tell you that there are 67 obs. for 28 variables.

Suppose you wanted to examine the new variable. If you simply type the name of the variable (*bs*), RStudio might give you an error message. You need to either attach the data frame or type the full name of the variable > palmbeach$bs

If you treat the data frame like a spreadsheet, you can find Palm Beach County (it’s the 50th observation down), and then you can find that its value of *bs* is approximately 0.4%.

7. The next step is to examine the distribution of Buchanan’s share of the vote, which you can do with summary statistics or various plots.

To see the summary statistics of *bs*: > summary(palmbeach$bs)

To see a box-and-whisker plot: > boxplot(palmbeach$bs,horizontal=T)

To see a stem-and-leaf plot: > stem(palmbeach$bs,scale=2)

To see a histogram: > hist(palmbeach$bs,prob=T)

You can export the box-and-whisker plot or the histogram using the Export tab in the **Plots** tab.

8. You have done a lot of typing so far; suppose you wanted to save your work for the future. The best way to do this is with a script. If you click on the icon with the green plus-sign, the first option it gives you is an R script. (You can also type Ctrl-Shift-N.) If you click on R script, it will create a new script, most likely named Untitled1, into which you can paste commands from the **Console** window. I advise you to create a script containing the commands for opening the data set, creating the *bs* variable, and plotting its distribution.

The last thing you should do before closing is clear out what you’ve done. In the **Plots** tab and the **Environment** tab, the icon that looks like a broom will clear all plots and data frames, respectively. In the **Console** window, type Ctrl-L. Close out RStudio and take a mental break.