**ST512 Assignment 1**

**Due Date:** Wednesday, January 22. (Note the unusual due date. Labs will typically be due on Mondays.)

**Assignment Goal:** The primary goal of this assignment is to introduce R. The exercises are fairly simple so that you can keep your focus on the software. Exercises will become more difficult in future labs.

**Reminders about the mechanics of completing and handing in labs:**

Working together on lab assignments to overcome obstacles is permitted. However, each student must compose and write his or her own programs, analyses, and reports. Any work that you hand in must be your own work, and must reflect your own understanding. Labs are due at the *beginning* of class on the day the homework is due. If you do not plan to attend class on the due date, you may turn the lab in by sliding it under the instructor's office door (251 David Clark Labs) before class on the due date.

**R:**

R is state‐of‐the‐art data analysis software. It can be freely downloaded at www.r‐project.org.

**RStudio:**

RStudio is a freely available integrated development environment for R. Using RStudio requires that you also have R installed on your machine. For ST512, you are welcome to work in whichever environment (R or RStudio) you prefer. Most R users use RStudio.

**The R Environment:**

When you first open R or RStudio, you will see a command window called the R Console. The R prompt “>” is the location where you enter R commands. We refer to this location as the “command line.”

**Some basics**

The internet abounds with tools to help you learn R. The Torfs & Brauer document, “A (very) short introduction to R” is especially helpful. It can be found on the ST512 homepage. If you are completely new to R, read sections 3‐4 of that document now (sections 1‐2 cover installation). Read sections 5‐9 before next week. (Sections 10‐12 are more advanced than we’ll need for ST 512, but do read them if you are so motivated.) If you have experience with R, browse sections 1‐9 to see if it contains any new information that you didn’t already know.

**Reading in data:**

R works by holding objects in memory. There are many types of R objects, including vectors, arrays, and functions. The basic R object for storing data is a data frame. A data frame is very similar to a matrix, with one row for each subject and one (named) column for each observation.

It is possible to construct small data frames directly from the command line. However, it is usually more efficient to build the data set in another program (e.g., Excel), save it as a text file, and then read the text file in directly to R.

In this analysis we will consider a study in which researcher took measurements on 247 men and 260 women. These were primarily individuals in their twenties and early thirties, with a scattering of older men and women, all physically active (several hours of exercise a week). The measurements were taken at San Jose State University and at the U.S. Naval Postgraduate School in Monterey, California and in dozens of California health and fitness clubs. The data set “body” contains the following variables:

# *Age (years) Weight‐ body weight (kg) Height (cm) Gender*

This data set (like all data sets used in labs) is stored in a public directory that can be accessed by pointing your web browser to

https://www4.stat.ncsu.edu/~gross/courses/ST512/data/. You may have to authenticate using your UNITY credentials.

In RStudio, there are two basic strategies for importing data. Both of these strategies begin with saving the data set to a text file. Instructions for completing these tasks follow below:

* Open a web browser. Point your web browser to https://www4.stat.ncsu.edu/~gross/courses/ST512/data/.
* Right‐click on the file that you want to save. Click 'Save Link As' (or 'Save Target As', depending on your browser).
* Save the file to a suitable destination.

Once you have downloaded the data file and saved it in an appropriate place, there are two options for reading the data into memory. Both are described below.

**Option 1:** Read the data into RStudio by using the 'read.table' command at the command line. That is, in the “Console” window of RStudio, enter the command below. Be sure to replace ‘*mypath*’ with the path of the file where you have saved the data.

# > body <‐ read.table("*mypath*/body.txt", header = TRUE)

The first row of the data file contains the column names “age”, “weight”, etc. The optional argument header = TRUE is included in the read.table command above to tell R that the first line of the file is a “header” line that contains column names.

**Option 2:** Use the Import Dataset wizard. The latest release of RStudio gives you two options for reading data into memory: via a ‘base’ program, and via a more sophisticated program called ‘readr’. I find the ‘base’ program to be easier to use, and ‘readr’ to be a bit clunky, but you are welcome to use whichever program you prefer. In any event, look for the 'Import Dataset' button on the Workspace tab (usually in the upper right) to launch the import wizard. Choose either ‘From Text (base)…’ or ‘From Text (readr)…’ to launch a dialog box based on either the ‘base’ or ‘readr’ program, respectively. If you use the ‘base’ program, the program should recognize the different features of the data file correctly. If you use the ‘readr’ program, you’ll need to change the delimiter from Comma to Tab. Note also that the ‘readr’ program will not correctly recognize the gender variable as a factor, so you’ll have to use the graphical interface to change the variable type from ‘character’ to ‘factor’. When you do so, RStudio will prompt you to enter the ‘factors’. This is a typo in the dialog box. The dialog box should prompt you to enter the ‘levels’ (a level is a unique value of a factor). In this case, the two levels are ‘Male’ and ‘Female’.

R now has a data frame in memory called “body” that contains these data. Confirm this by using the ls() command (for 'list') to ask for a list of objects in memory. You should see something similar to:

# > ls() [1] "body"

In RStudio, you should see the data set listed in the Workspace window.

To confirm that the data were entered correctly, ask for a summary of “body”. Try

# > summary(body)

“Summary” is a useful, all‐purpose command for learning about a data frame. If you have read the data into memory correctly, the output from the “summary” command should look like this:

> summary(body)

age weight height gender Min. :18.00 Min. : 42.00 Min. :147.2 Male :247

1st Qu.:23.00 1st Qu.: 58.40 1st Qu.:163.8 Female:260

Median :27.00 Median : 68.20 Median :170.3

Mean :30.18 Mean : 69.15 Mean :171.1

3rd Qu.:36.00 3rd Qu.: 78.85 3rd Qu.:177.8

Max. :67.00 Max. :116.40 Max. :198.1

Use the output of 'summary' to answer question 1.

## Accessing individual variables in a data frame

The data frame 'body' contains four separate variables. The syntax for accessing each of these variables is

## data.frame$variable

For example, if we want to find the average age of individuals in this data set, we can do so with the command

# > mean(body$age)

Use what you've learned about R so far to answer questions 2‐5.

## Comparing heights and weights of men vs. women

We would like to use these data to compare the heights and weights of males and females. We will do this using a two sample *t*‐test. You should know how to do a t‐test from ST511 or a previous course. If you need a refresher, consult section 5.3 of the OpenIntro statistics text available from the course website.

First, we want to plot the data using the function “stripchart”. Use the command

# > stripchart(body$height ~ body$gender, method = "jitter")

There are two “arguments” supplied to the stripchart function. The first argument is “body$height~body$gender”, which is a formula that tells R to plot height for different genders. The second argument is method=”jitter” which tells R to “jitter” the data points so that they don’t lie on top of one another. (If you’re curious, try the above command without the second argument and see what happens. Note that you can use the up‐ and down‐ arrow keys to cycle through recently issued commands.)

R has a function called “t.test” that performs a two‐sample t‐test automatically.

Perform a *t*‐test comparing the heights of men to women by using the command

# > t.test(height ~ gender, data = body)

Use the output from the above *t*‐test to answer question 6.

## Simple regression: Studying the relationship between height and weight

Now we'll use a simple linear regression to characterize the relationship between body height and body weight. We will treat height as the "predictor" and weight as the "response". First make a scatter plot of these data with the commands

> plot(weight ~ height, data = body)

Now fit a simple linear regression using height to predict weight. Use the code

# > slr1 <‐ lm(weight ~ height, data = body)

"lm" is the R command that we use to fit a regression. ("lm" is an abbreviation for [l]inear [m]odel.) As a result of the command above, there is a regression model stored in memory under the name 'slr1'. Ask for a summary of the regression model with the command

> summary(slr1)

Add the regression line to your plot with the command

# > abline(slr1)

(Note that the above command only works if you already have a plot window open. If you've closed the scatter plot from the 'plot' command above, you'll need to re‐create it before the 'abline' command works.) Use the regression model to answer questions 7 –

10.

We'll do much more with regression models in subsequent labs.

**ST512 Assignment 1. Questions**

**Name:**

1. What is the median age of the individuals in this data set? What is the range?

Median: 27 years

Range: (18, 67) years

1. The R command for calculating a variance is var(). (One must supply the name of the variable within the parentheses.) What is the variance of the heights of individuals in this data set?

Variance: 88.5 cm2

1. The R command for square root is sqrt(). What is the standard deviation of the weights of individuals in this data set?

Standard Deviation: 13.35 kg

1. The R command to find the correlation between two variables (call them a and b) is cor(a,b). What is the correlation between the height and weight of individuals in this data set?

Cor: 0.717

1. Suppose we view the individuals in this study as a random sample from a population of healthy individuals, and we want to draw inferences about this population. Let μ be the average height of this population. Treat the sample mean from this sample as an estimate of μ. What is the standard error of your estimate?

Standard Error: 0.593

1. Does the t‐test provide evidence that the average heights of males and females are significantly different? Explain.

1. Report the estimated slope and intercept from the simple linear regression of weight vs. height, as well as their estimated standard errors.

β0 = -105.01 cm β1 = 1.02 cm/kg sβ0 =

1. If you've fit the regression model correctly, the estimate of the intercept should be negative. We know that people cannot have negative weights. Therefore, does the negative intercept suggest that the regression line provides a poor fit to the data? Why or why not?

1. Find each of the following quantities for the simple linear regression of weight vs. height: (Hint: You will have to use the *R*2 value at some point.)

*s*2 

*SSE* 

*SSTotal* 

1. Find a 90% CI for the slope,1: