**Your name: Matt Bass Your lecture instructor: Professor Bret Zeldow**

Objectives:

**Part1**

1. Estimate population parameters using sample data.
2. Assess the bias (i.e. accuracy) of your estimate.
3. Use R-studio to take a sample and estimate a parameter.

**Part2**

1. Generate basic descriptive statistics in R-studio.

2. Create basic histograms and boxplots

3. Create categorical variables and summarize them with the table command

4. Create a graph of side-by-side boxplots (x-var is categorical, y-var is numerical)

The full text of the Gettysburg Address is below:

Four score and seven years ago our fathers brought forth on this continent a new nation, conceived in liberty, and dedicated to the proposition that all men are created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation so conceived and so dedicated, can long endure. We are met on a great battlefield of that war.

We have come to dedicate a portion of that field as a final resting- place for those who here gave their lives that that nation might live. It is altogether fitting and proper that we should do this.

But, in a larger sense, we cannot dedicate, we cannot consecrate, we cannot hallow this ground. The brave men, living and dead, who struggled here, have consecrated it far above our poor power to add or detract. The world will little note nor long remember what we say here, but it can never forget what they did here.

It is for us, the living, rather, to be dedicated here to the unfinished work which they who fought here have thus far so nobly advanced. It is rather for us to be here dedicated to the great task remaining before us, that from these honored dead we take increased devotion to that cause for which they gave the last full measure of devotion, that we here highly resolve that these dead shall not have died in vain, that this nation, under God, shall have a new birth of freedom, and that government of the people, by the people, for the people, shall not perish from the earth.

**1.** Examine the Gettysburg Address and then circle 10 representative words from the passage. After you’ve made your selections, go on to the next page.

**2.** Consider the following variables and identify the type of each variable

Length of word (number of letters)

Variable type: \_\_\_\_\_\_\_Quantitative\_\_\_\_\_\_\_\_\_\_\_

Whether or not the word contains *more* than four letters

Variable type: \_\_\_\_\_\_\_Categorical\_\_\_\_\_\_\_\_\_\_\_

**3.** Record the words in your sample and the data for these two variables in the table below

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| Word | Liberty | Civil | Nation | Battlefield | Consecrated |
| Length | 6 | 5 | 6 | 11 | 11 |
| More than 4 characters? | yes | yes | yes | yes | yes |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 6 | 7 | 8 | 9 | 10 |
| Word | Power | God | Freedom | Government | People |
| Length | 5 | 3 | 7 | 10 | 6 |
| More than 4 characters? | yes | no | yes | yes | yes |

**4.** Do you think your sample is representative of the population of 268 words in the Gettysburg Address? Explain?

I do not think that this is the most representative sample of the 268 words because

The words were chosen by myself (a human) and humans are notorious for being unable to truly choose randomly. There is also the case that when I was choosing words I didn’t chose stop words (words that don’t have much meaning but appear a lot in the English language) and I could assume that in general the text will be comprised mainly of those stop words

**5a.** What is the average word length in your sample?

You should use R-Studio to answer this question. Open it up and enter your data using the c() command. The “c” is short for concatenate. It’s used to combine a set of numbers into a vector. For example, to create a vector called my.data that contains the values 5, 10, 15, and 20, you could type: my.data <- c(5, 10, 15, 20). Once you’ve done that, you can determine the average by typing mean(my.data).

**\*\*\*IMPORTANT NOTE\*\*\***

Any time you attempt to run an R command, I strongly recommend that you DO NOT copy and paste it. Instead, type the command into an R-script and then execute it (using Ctrl+Enter, or highlighting the command and clicking the Run button). Copying and pasting can sometimes produce unexpected results when going between word documents/pdfs/etc. and R, especially when unusual characters are used. Also, typing the command into a script will allow you to easily modify it if you want to use it later.

**5b.** What proportion of words in your sample is longer than four characters? (Hint: do something similar to what you did in 5a).

0.9

Next, load the entire Gettysburg Address into RStudio. You can do this a number of ways, but the simplest way would be to download the Gettysburg file from our Moodle page (under today’s Moodle heading), then upload it to your local directory (click on the ‘Files’ tab in the lower right window in R-studio, then click ‘Upload’, and then click the ‘choose file’ button. You can then navigate to where the file is stored and click on it). Once you’ve uploaded the data to your directory, you can load it into R-studio with the following command:

GB.data <- read.csv(file=file.choose())

This command will open a window and let you navigate to where the file is stored. Once you click on the file, it will be stored as a new dataframe in RStudio called GB.data. Note that the ‘GB.data’ part is arbitrary – you could call it anything you want.

**5c.** Determine the average word length for the whole passage.

You should use the mean() function. The most direct way to calculate the average word length is to type: mean(GB.data$Length) Note that this is a population parameter since it has been calculated using all the words in the entire speech (i.e., the population).

4.294776

**6a.** How close was your sample mean to the true mean word length? Do you think your way of sampling words was representative? Was it biased? Explain.

The sample mean of 4.294776 was much less than the “representative sample” that I chose which averaged out to a word length of 7. I would say that I as a human, and other students in my class based off of their samples mean word lengths, are bias to thinking that longer words hold more meaning of the text (which I would argue to a point they do because many stop words such as a ,an ,and the appear a lot and have little meaning in true English language) However true representative sample in statistics would mean that a truly random sample was taken so the average word length of my samples should have been much closer to 4.

**6b.** What if we used everyone else’s sample data too? Using information from everyone in the class, enter a dataset in R-studio that contains the sample means from your classmates. What is the overall average of all the sample means? Use the c() function and the mean() function again. For example:

classdata <- c(value1, value2,...)

mean(classdata)

**6c.** Was the overall average of the samples close to the true average word length?

It was much closer at 6.277273 versus mine at 7 when comparing to the true average word length of all the data at 4.294776. This makes sense because we are not all likely going chose the same words leading to more variations in our samples with a min word mean of 4 and a max word sample length mean of 9

**7.** Think of an alternate method for selecting the words that would create an accurate (unbiased) estimate. What would you do? State the steps you’d take.

I would use a random number generator to generate a truly random and unique pick for each word in my sample. If I’m using deplyr I would just use sample\_n. Normal R I would go rand\_word\_samp = sample(GB.data$Word, size = 10) for a 10 word sample

**8.** We can use R-studio to take a sample for us. Try running the following:

my.sample <- sample(GB.data$Length, 10)

mean(my.sample)

Is this sample mean closer to the true average word length? Try rerunning the above commands a few times to see how sample mean changes.

**Part 2**

Download the surveydata.csv file from the moodle page – it’s located under today’s heading.

You’ll see the following variables in the file:

classyr: year and semester course was taken

sex: 1=Female, 2=Male

tv: number of hours of TV watched per week

salary: expected annual salary 10 years after graduating from Colby

interest: initial interest in course (1=no interest, 10=very interested)

politics: political views (1=very liberal, 10=very conservative)

class: number of years at colby (1 = First Year, 4 = Senior)

dining: favorite dining hall (1=Dana, 2=Foss, 3=Bobs)

biology: 1 for bio majors, 0 for others

ne: 1 for from New England, 0 for from someplace else

number: random number between 1 and 100

mother: mother’s height in inches

father: father’s height in inches

you: respondent’s height in inches

money: how much money respondent was carrying

hometown: how far away respondent’s hometown is from Colby

Load these data into R-studio (see above if you forgot how to load the data).  
**Name your dataframe surveydata.**

Take a look at the first few records in the dataframe by typing: head(surveydata). You can view the whole dataframe by typing: View(surveydata)

The basic commands that you need to summarize and view data are listed in Chapter 2 of the R User’s Guide that came with your book (posted on Moodle). Check them out! We’ll go over a few of them below.

**Basic Descriptive Statistics – quantitative variables**

**1.** You can find out how many rows (# of observations) and columns (# of variables) are in your dataframe by typing: dim(surveydata). What did you find?

I found that there are 134 samples (cases) with 6 variables (observations a sample)

**2.** You can determine the mean, median, and standard deviation of a variable by using the mean(), median(), and sd() commands respectively. We’ll discuss these much more in class, but for now, let’s try calculating the mean of salary. Remember this variable is the student’s best guess for what their annual salary will be 10 years after they graduate from Colby. Try typing mean(surveydata$salary)

What happened?

It results in NA because their were some NA entries in the column meaning there is not a response for that sample (NA is a special value in R standing for Not Available and is used to represent missing values)

**3.** R doesn’t always know what to do when there are missing values in your dataset. You need to tell it how you want to handle missing data. If you type surveydata$salary you’ll see that there are quite a few ‘NA’s in the data. You can still calculate the mean of the non-missing observations, but you need to tell R to omit the missing values. Do this by typing:

mean(surveydata$salary, na.rm = TRUE) == 94560.16

The na.rm = TRUE option tells R that you want to remove the missing values (i.e. the NA’s) before trying to calculate the mean. You can see the other options/syntax for mean() by typing ?mean.

**4.** Some functions automatically handle missing data. For example, you can produce a brief summary for a quantitative variable with the summary() command. Try generating a summary for the salary variable. R-studio should return the min, mean, median, max, 25th, and 75th percentile. You can also see the number of NA’s (i.e. missing values).

**Basic Graphs – quantitative variables**

**5a.** You should use a histogram or a boxplot to show the distribution of a quantitative variable. The corresponding commands are hist() and boxplot(). We’ll discuss both of these much more in class, but for now, try generating a histogram for the salary variable.

**5b.** You can add additional options (separated in the command by commas) by inserting them into the command. Add a title and change the axis label by adding the options main=” “ and xlab = “ “ into the hist command. Your command should look something like this:

hist(surveydata$salary, main=“mytitle”, xlab=“mylabel”)

**5c.** Boxplots are similar. The command: boxplot(surveydata$salary) produces a basic boxplot. Try it out.

**Categorical Variables**

Categorical variables can be summarized with a table. For example, let’s investigate dining hall preference. Try table(surveydata$dining). By itself this isn’t very helpful. R-studio doesn’t know that the dining variable is categorical and it doesn’t know that 1=Dana, 2=Foss, and 3=Bobs. We should fix this. First, let’s see what type of variable R-studio thinks dining is – this is accomplished with the class() command:

> class(surveydata$dining)

[1] "integer"

**6a.** We can change this into a categorical variable using the factor() command. Categorical variables are “factor” variables in R-studio. Try typing:

surveydata$dining <- factor(surveydata$dining)

To see if the command worked, try checking the class of the dining variable now. R should return “Factor”. If you just display the variable, you should see something like this:

> surveydata$dining

[1] 2 2 1 1 1 1 1 2 2 1 1 2 2 2 1 3 2

[18] 2 1 2 3 3 3 3 2 2 1 3 1 2 1 3 3 2

[35] 2 3 3 2 2 2 3 1 3 2 1 2 3 3 1 2 1

[52] 2 1 2 2 2 1 3 1 3 1 <NA> 2 2 2 2 1 2

[69] 2 3 2 3 3 2 2 2 3 3 3 3 3 3 2 3 1

[86] 3 2 1 3 2 3 1 2 2 3 2 1 1 1 2 3 1

[103] 2 3 2 1 3 1 2 3 1 1 2 1 1 1 2 1 2

[120] 1 1 3 1 2 2 2 2 1 2 1 3 2 3 1

Levels: 1 2 3

**6b.** All factor variables have levels. The levels are the different values the factor variable can take on. Right now, they are still set to 1, 2, and 3…but we can change them. Try typing:

levels(surveydata$dining)

And R-studio should return 1, 2, and 3. We can change these values to whatever we want.

**6c.** Try typing the following command (TYPE it in…don’t copy and paste it – the quotes don’t directly translate into R-studio when pasted in!).

levels(surveydata$dining) <- c(“Dana”, “Foss”, “Bobs”)

You can check out your new variable by just typing: surveydata$dining

All the numbers should be replaced with the correct names. Now, retry that table command!

> table(surveydata$dining)

Dana Foss Bobs

42 54 37

Much better. You could get proportions instead of counts by dividing by the total number of observations.

**6d.** You can add more categorical variables to the table command to create 2-way tables. Does dining hall preference differ based on sex? Try:

table(surveydata$dining, surveydata$sex)

It sure looks like it! (Note: if you want, convert sex to a factor variable and rename the levels just like we did with the dining variable, 1=Female and 2=Male).

**Comparing distributions**

**7.** One advantage of boxplots over histograms is that it’s relatively easy to put multiple boxplots on the same axes, which allows you to easily compare distributions. For example, the following command can be used to compare distribution of political views by dining hall preference (recall that 1=more liberal and 10=more conservative).

boxplot(surveydata$politics ~ surveydata$dining,  
 main=“Distribution of Politics by Dining Hall Preference”)

This is a great graph to produce when the explanatory variable is categorical and the response is numeric.

All the graphics commands in this lab are basic versions (e.g. hist(), boxplot(), etc.). They’re great for exploring and doing preliminary analyses. In future labs we’ll be learning how to produce high-quality graphics using the ggplot2 package. You should refer back to this lab and to Chapter 2 of the posted R Users Guide that came with your book if you need help with the basic commands.

**ALL OF MY CODE FOR THE LAB CAN BE FOUND HERE:**

https://github.com/mattjax16/SC212-Data-Science/blob/main/Labs/Lab2/LAB02.R