Lab 7: Introducing the Central Limit Theorem and Confidence Intervals

- Due date: Friday, October 16 10:00pm.
- Submission process: Follow the submission instructions on the final page. Make sure you do not remove any `\newpage` tags or rename this file, as this will break the submission.

Helpful hints:

- Every function you need to use was taught during lecture! So you may need to revisit the lecture code to help you along by opening the relevant files on Datahub. Alternatively, you may wish to view the code in the condensed PDFs posted on the course website. Good luck!
- Knit your file early and often to minimize knitting errors! If you copy and paste code for the slides, you are bound to get an error that is hard to diagnose. Typing out the code is the way to smooth knitting! We recommend knitting your file each time after you write a few sentences/add a new code chunk, so you can detect the source of knitting errors more easily. This will save you and the GSIs from frustration! **You must knit correctly before submitting.**
- If your code runs off the page of the knitted PDF then you will LOSE POINTS! To avoid this, have a look at your knitted PDF and ensure all the code fits in the file (you can easily view it on Gradescope via the provided link after submitting). If it doesn't look right, go back to your .Rmd file and add spaces (new lines) using the return or enter key so that the code runs onto the next line.

(Optional) Relevant Textbook Exercises

The following questions in your textbook are helpful practice for understanding today's material:

Baldi and Moore: Ex. 13.5, 13.8, 13.9, 13. 10, 13.12, 13.14

Introduction

You will use a central data source (a Google sheet) attached to the lab to develop a very concrete idea of sampling distributions, and to see the central limit theorem in action.

The underlying population

Suppose you had a data frame containing the **entire population** of all residents of Alameda County. You have data on three variables:

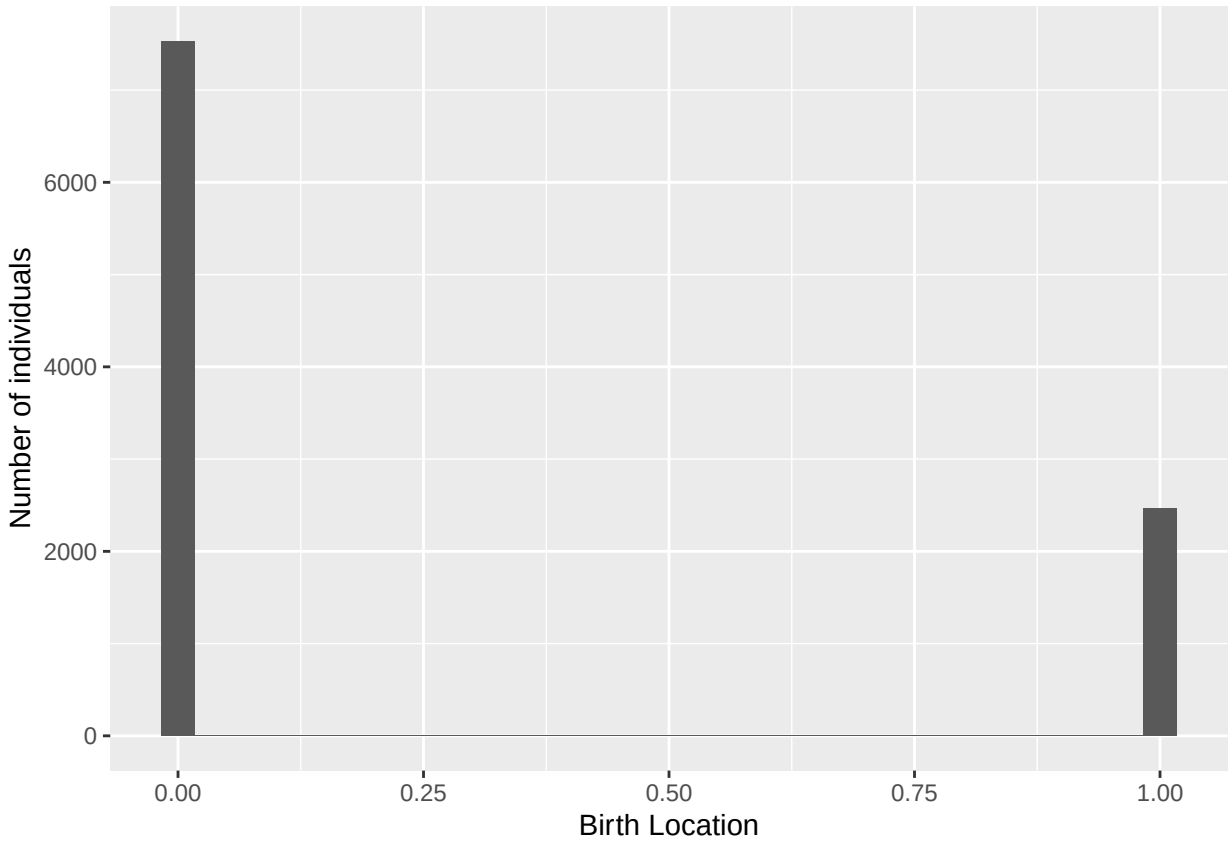
1. Born either out (=1) versus in (=0) the county.
2. Number of siblings (integer)
3. Number of visits to the hospital last year
4. Read in the data, L06_Alameda.csv, and assign it to the name `alameda_pop`. Calculate the true (population) mean, and make histograms or bar charts of the distribution for each variable.

```
library(dplyr)
library(ggplot2)
library(readr)

alameda_pop <- read.csv("L06_Alameda.csv")
alameda_pop %>% summarize(mean_birthloc = mean(birth_loc))
```

```
##   mean_birthloc
## 1        0.2468
```

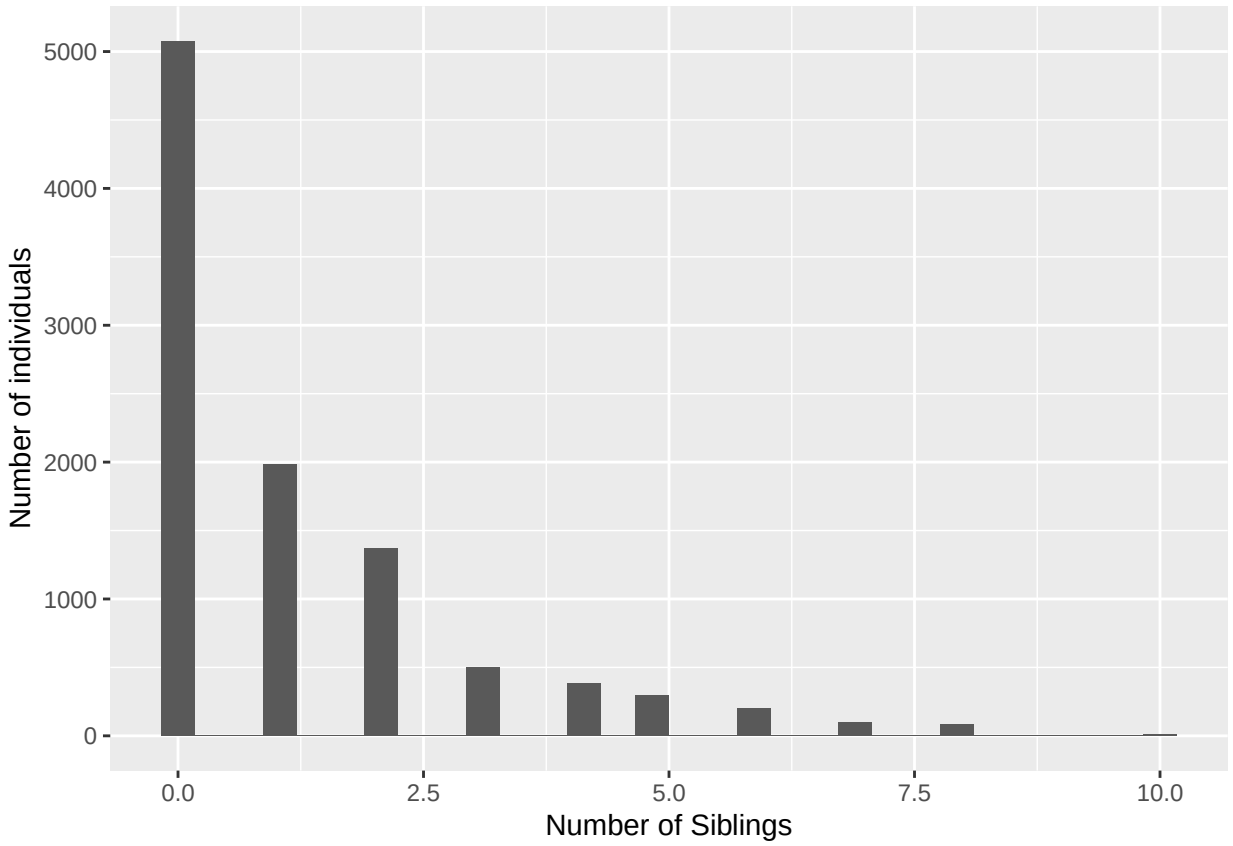
```
ggplot(alameda_pop, aes(x = birth_loc)) + geom_histogram() +
  labs(x = "Birth Location", y = "Number of individuals")
```




```
alameda_pop %>% summarize(mean_numsibs = mean(num_sibs))
```

```
##   mean_numsibs
## 1       1.1899
```

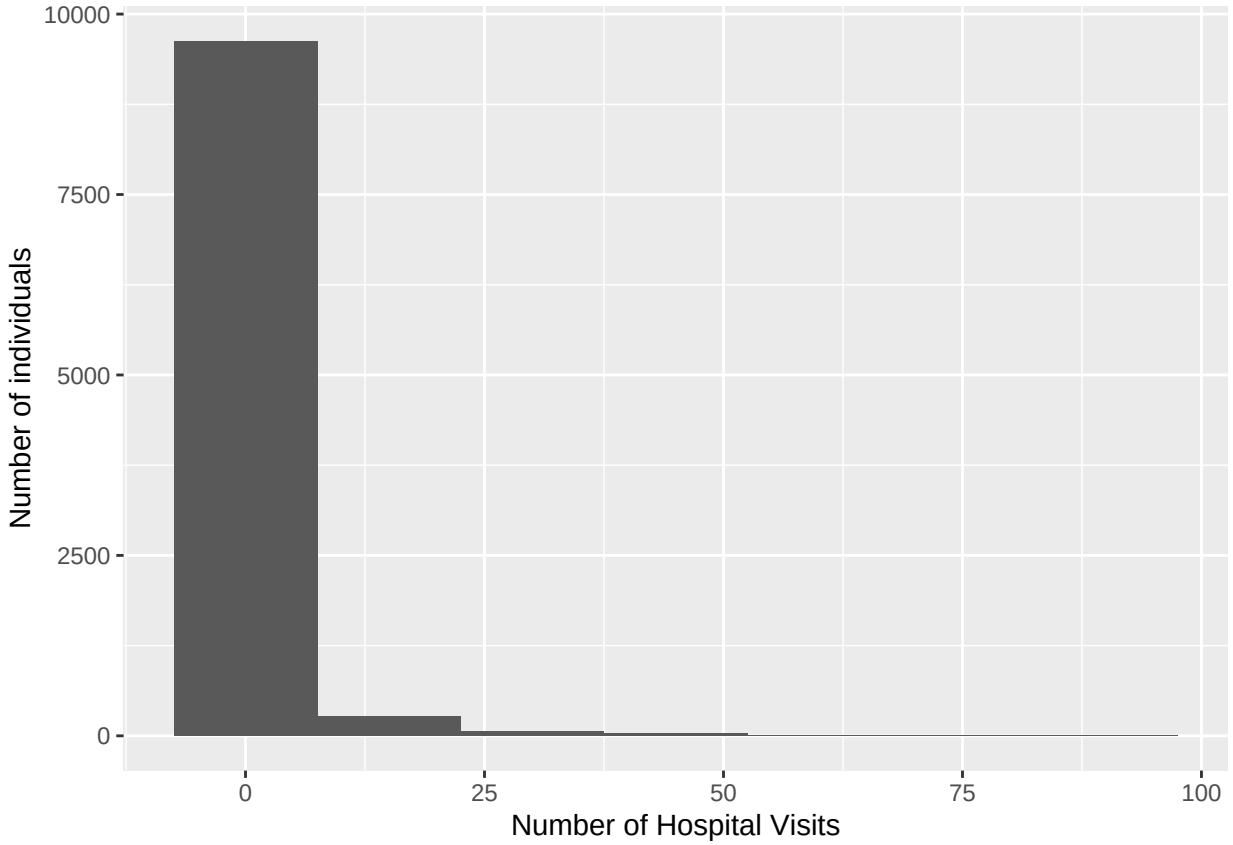
```
ggplot(alameda_pop, aes(x = num_sibs)) + geom_histogram() +
  labs(x = "Number of Siblings", y = "Number of individuals")
```




```
alameda_pop %>% summarize(mean_hospvisit = mean(hosp_visit))
```

```
##   mean_hospvisit
## 1         1.0287
```

```
ggplot(alameda_pop, aes(x = hosp_visit)) + geom_histogram(binwidth = 15) +
  labs(x = "Number of Hospital Visits", y = "Number of individuals")
```



Notice that the distribution of number of siblings and number of hospital visits are discrete distributions and both skewed right. Today, we will focus on the distribution of the number of siblings `num_sibs`. Remember, we know the population mean exactly, because we have all the data. You calculated the underlying population mean in the code chunk above.

Calculating the sampling distribution

We are now going to look at the approximate **sampling distribution for the sample mean** of the `num_sibs` in (live) action. Remember from earlier lectures that a **sampling distribution** is a distribution for a statistic, like the sample proportion or the sample mean.

Each student will be tasked with repeatedly taking a random sample of the population. Once you take your sample you will compute the sample mean and add it to a shared google sheet. As the data is added to the google sheet, notice how the graph to the right of the data changes; this illustrate how the sampling distribution varies for increasingly larger sample sizes.

The GSIs will provide you the link to the communal google sheet. The columns in the sheets are `n (Sample_size)` `mean(numSibs)` `Name (your sign in)`.

Your task

1. Randomly generate 10 simple random samples of size $n = 5$ from the population. Calculate the average number of siblings for each of your samples. We wrote code to start you off, which you can simply copy and paste 10 times to generate ten randomly drawn samples and their sample means.

One sample has been provided to you. Try it yourself 10 times and record your results in the vector below.

```
size_5 <- sample_n(alameda_pop, 5, replace = FALSE)
size_5 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          2.4
```

```
size_5 <- sample_n(alameda_pop, 5, replace = FALSE)
size_5 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          0.6
```

```
size_5 <- sample_n(alameda_pop, 5, replace = FALSE)
size_5 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          2.6
```

```
size_5 <- sample_n(alameda_pop, 5, replace = FALSE)
size_5 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          1.4
```

```
size_5 <- sample_n(alameda_pop, 5, replace = FALSE)
size_5 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          0.4
```

```
size_5 <- sample_n(alameda_pop, 5, replace = FALSE)
size_5 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          1.4
```

```
size_5 <- sample_n(alameda_pop, 5, replace = FALSE)
size_5 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          1.4
```

```
size_5 <- sample_n(alameda_pop, 5, replace = FALSE)
size_5 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          0.8
```

```r
size_5 <- sample_n(alameda_pop, 5, replace = FALSE)
size_5 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          0.4
```

```r
#repeat this code 10 times.
size_5 <- sample_n(alameda_pop, 5, replace = FALSE)
size_5 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          1.2
```

After you've calculated 10 sample means using the above code, copy and paste your data into the google sheet for your lab section. The links to all the google sheets are:

- 101B (Thursday 5-7pm): https://docs.google.com/spreadsheets/d/130FOigRrcdzyMpxSmzxEcdx2TOthUeKZFBpfJgH edit?usp=sharing
- 102B (Wednesday 2-4pm): https://docs.google.com/spreadsheets/d/1sVx2Vd_57DWNHRvQ1UP_ BhaMuRUnc-H-L-f5Ujuv-fs/edit?usp=sharing
- 103B (Friday 9-11am): https://docs.google.com/spreadsheets/d/1sJpPRIyl83CcFwOyfwJCgyM1LubDZx-RvAfYwG1v8iA/edit?usp=sharing
- 104B (Wednesday 9-11am): https://docs.google.com/spreadsheets/d/1iOUF5bohUL3_tEHkjOjXC_ bueFVLm-sA-lWtxHFLgS0/edit?usp=sharing
- 105B (Wednesday 4-6pm): https://docs.google.com/spreadsheets/d/1PwcGfK4dKbybxFfrwXm8-H4yCzKVBzCenDaf902Btlg/edit?usp=sharing
- 106B (Wednesday 5-7pm): https://docs.google.com/spreadsheets/d/1AP3oewsuzDDpG6oPnMEqXXjzYNY1Q2EBQjy edit?usp=sharing
- 107B (Thursday 8-10am): https://docs.google.com/spreadsheets/d/15JdSP4V3-K5BY76sxRXL3wQcDWhNkuAXxPX edit?usp=sharing
- 108B (Wednesday 1-3pm): https://docs.google.com/spreadsheets/d/151RrhBvC33sb_4mJDl5ibNOPJRKvKQK4LlCC 1pDA/edit?usp=sharing
- 109B (Wednesday 9-11am): https://docs.google.com/spreadsheets/d/1bR2eXPNrdKIi_Cxw5nllQFPwR3-KWcvhA93a6IfRxhc/edit?usp=sharing
- 110B (Friday 9-11am): https://docs.google.com/spreadsheets/d/17Td3U8t4MoH0nCOTevhJjsiENPaYAWT_ egRclsxM9IM/edit?usp=sharing

Once the sheet is full, look at the plot of the **sampling distribution** for the mean number of siblings when $n = 5$.

- What is the range of the sampling distribution of the mean?

The range is from 0 to 4.5.

- What is the shape of the sampling distribution of the mean?

The sampling distribution is unimodal and skewed to the right.

- What is the mean of the sampling distribution of the mean?

The mean of the sampling distribution of the mean is 1.18625.

Once the class has examined the sampling distribution when $n = 5$, repeat the same steps for n=50.

2. Repeat the above process for a sample size of $n = 50$.

```
#### YOUR CODE GOES HERE ####
size_50 <- sample_n(alameda_pop, 50, replace = FALSE)
size_50 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1         1.54
```

```
size_50 <- sample_n(alameda_pop, 50, replace = FALSE)
size_50 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1         0.86
```

```
size_50 <- sample_n(alameda_pop, 50, replace = FALSE)
size_50 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          0.8
```

```
size_50 <- sample_n(alameda_pop, 50, replace = FALSE)
size_50 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1         0.98
```

```
size_50 <- sample_n(alameda_pop, 50, replace = FALSE)
size_50 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          1.4
```

```
size_50 <- sample_n(alameda_pop, 50, replace = FALSE)
size_50 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1         1.14
```

```
size_50 <- sample_n(alameda_pop, 50, replace = FALSE)
size_50 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1         0.92
```

```
size_50 <- sample_n(alameda_pop, 50, replace = FALSE)
size_50 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1         1.42
```

```
size_50 <- sample_n(alameda_pop, 50, replace = FALSE)
size_50 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1         1.34
```

```
size_50 <- sample_n(alameda_pop, 50, replace = FALSE)
size_50 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1         1.26
```

```
size_50_samples <- c("YOUR VALUES HERE")
```

After you calculated your 10 sample means, navigate to the google sheet from before, but switch to the sheet with $n = 50$ (you can switch tabs in the bottom left). Add your data for $n = 50$.

Look at the plot to the right on the google sheet; now with $n = 50$

- What is the range of the sampling distribution of the mean? How does it compare to when $n = 5$?

The range is from 0.6 to 2, which is smaller than the range of when n=5.

- What is the shape of the sampling distribution of the mean? How does it compare to when $n = 5$?

The shape is unimodal and approximately normal, without much skew. This graph looks more normal than the distribution for n=5.

- What is the mean of the sampling distribution of the mean?

The mean of the sampling distribution of the mean is 1.190857143.

3. Repeat the entire process for sample size $n = 500$.

```
#### YOUR CODE GOES HERE ####
size_500 <- sample_n(alameda_pop, 500, replace = FALSE)
size_500 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1        1.282
```

```
size_500 <- sample_n(alameda_pop, 500, replace = FALSE)
size_5 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1          1.2
```

```
size_500 <- sample_n(alameda_pop, 500, replace = FALSE)
size_500 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1        1.014
```

```
size_500 <- sample_n(alameda_pop, 500, replace = FALSE)
size_500 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1        1.094
```

```
size_500 <- sample_n(alameda_pop, 500, replace = FALSE)
size_500 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1        1.324
```

```
size_500 <- sample_n(alameda_pop, 500, replace = FALSE)
size_500 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1        1.208
```

```
size_500 <- sample_n(alameda_pop, 500, replace = FALSE)
size_500 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1         1.25
```

```
size_500 <- sample_n(alameda_pop, 500, replace = FALSE)
size_500 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1        1.172
```

```
size_500 <- sample_n(alameda_pop, 500, replace = FALSE)
size_500 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1         1.19
```

```r
size_500 <- sample_n(alameda_pop, 500, replace = FALSE)
size_500 %>% summarise(mean_numSibs=mean(num_sibs))
```

```
##   mean_numSibs
## 1        1.282
```

```r
size_500_samples <- c("YOUR VALUES HERE")
```

- What is the range of the sampling distribution of the mean? How does it compare to when $n = 5$ and $n = 50$?

The range is 0.85 to 1.85, which is even smaller than the ranges for both n=5 and n=50.

- What is the shape of the sampling distribution of the mean? How does it compare to when $n = 5$ and $n = 50$?

The shape will be approximately normal, with more variance. It is more normal than when n=5 and n=50.

- What is the mean of the sampling distribution of the mean?

The mean of the sampling distribution of the mean is 1.2024.

Suppose you have 500 classmates, and they have done this lab and added their data to this sheet: https://docs.google.com/spreadsheets/d/1AXStOd618raoWvrBequxOh5CDwgisFJoHo50fmcKb_E/edit?usp=sharing

Open the link, and look at the resulting sampling distributions for $n = 5$, $n = 50$, and $n = 500$. This is what happens when you repeat the sampling 5,000 times.

4. For which sample size should the sampling distribution of the mean be most normal?

(a) n=5
(b) n=50
(c) n=500

Assign your letter choice as a string. Example: sampleSize_answer<-“b”

```r
sampleSize_answer<- "c"
sampleSize_answer
```

```
## [1] "c"
```

```r
check_problem4()
```

```
## [1] "Checkpoint 1 Passed: Correct"
##
## Problem 4
## Checkpoints Passed: 1
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

Part 2: Confidence intervals

Calculate the true (population) mean of the variable `height`, and the population standard deviation. To do this, use a dplyr function to make a dataframe object called `height_mean_sd` with the first column called `mean_height` and the second column called `sd_height`.

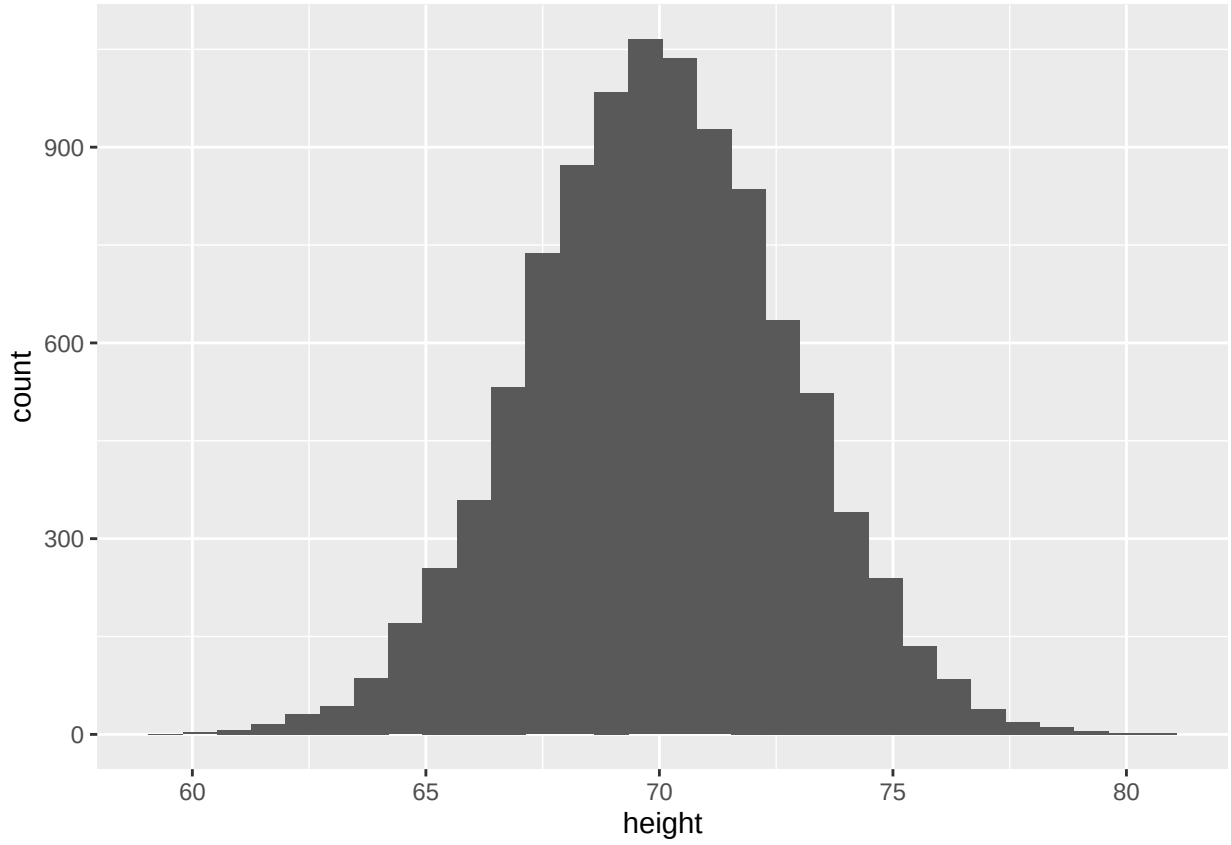
Make a histogram called `plot1` of `height` and comment its distribution. Use ggplot() to quickly make a ggplot. Does it look Normally distributed?

```
height_mean_sd <- alameda_pop %>% summarize(mean_height = mean(height), sd_height = sd(height))
plot1 <- ggplot(alameda_pop, aes(x = height)) + geom_histogram()

height_mean_sd
```

```
##   mean_height sd_height
## 1    69.97705  2.786314
```

```
plot1
```




```
check_problem5()
```

```
## [1] "Checkpoint 1 Passed: You used alameda_pop"
## [1] "Checkpoint 2 Passed: height is on the x axis"
## [1] "Checkpoint 3 Passed: A histogram has been defined in ggplot"
## [1] "Checkpoint 4 Passed: height_mean_sd has 1 row and 2 columns."
```

```
## [1] "Checkpoint 5 Passed: You did the corret manipulation!"
##
## Problem 5
## Checkpoints Passed: 5
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

Yes, the histogram does look normally distributed.

Suppose that you know the population standard deviation. It is what you calculated in the previous code chunk. Use the following code chunk to save its value. Replace the word `NULL` with your calculation of the population standard deviation.:

```
known_pop_sd <- 2.786314


check_problem6()
```

```
## [1] "Checkpoint 1 Passed: You found the correct standard deviation!"
##
## Problem 6
## Checkpoints Passed: 1
## Checkpoints Errored: 0
## 100% passed
## --------
## Test: PASSED
```

Now, suppose you *do not know* the population mean, but wanted to estimate it based on a sample. In this lab, we actually know the true value because we calculated it above. This way, we can see how well any one sample does at estimating the population mean and see how often the confidence intervals contain the mean across several repeated samples.

Calculating the CI and looking at its performance

We are now going to compute (and enter into our master google sheet) 95% confidence intervals (CI) for sampling means of different sizes. For this lab we:

- Have a variable with an underlying Normal distribution
- Will take simple random samples (SRS) from this distribution
- Know the value of the population mean (from your calculation in the first code chunk)

Thus, we satisfy the three conditions discussed in Wednesday's lecture for calculating a confidence interval when the underlying SD is known.

Recall the format for the 95% confidence interval in this setting (Hover with your mouse within the double-dollar signs to see the formula or knit the file to read them easily):

$$\bar{x} \pm 1.96 \times \frac{\sigma}{\sqrt{n}}$$

Where:

- The sample mean is the estimated mean, based on your sample
- σ is the known standard deviation `known_pop_sd`, that you saved earlier for the distribution of `height`
- $\sigma/\sqrt{n}$ is the standard error of the sampling distribution for $\bar{x}$
- 1.96 is the critical value for a 95% CI.

Like Part 1, each student will take a few samples from the distribution. This time, you will calculate the mean of your sampled heights and its confidence interval using the above formula. We will then record this information into the google sheet and plot all the CIs when we have at least 20 of them.

Here are the links to the communal Google sheets. Open the one for your lab. The columns in the sheets are `sample_mean_heights lower_CI upper_CI sample_size student`.

Lab 101B: https://docs.google.com/spreadsheets/d/1bWatXMlpJLvOR0ch4Hox-5kMHcsit-32nAuPXhYgXMI/edit?usp=sharing Lab 102B: https://docs.google.com/spreadsheets/d/1V8TtSV6uv7prZIYcGQTn9QQ482IajVYLX6E4LZjw edit?usp=sharing Lab 103B: https://docs.google.com/spreadsheets/d/1diY17Q78CVczyLFOSwlySRwF0Ef9T96o8IYO6lHEj edit?usp=sharing Lab 104B: https://docs.google.com/spreadsheets/d/1gVGxDOPrlk3jvKtI5raezOj17WF1IhJaxt3gs1KR4Ds edit?usp=sharing Lab 105B: https://docs.google.com/spreadsheets/d/1sYpg1MkTcIR2ViHdQZrtVJ7_oR5OUwVJT-f_LwmBjQo/edit?usp=sharing Lab 106B: https://docs.google.com/spreadsheets/d/1TMpkxrOXvfp7-u6KxnQiIRFvUuIdrFDNDl6hHwSqtS0/edit?usp=sharing Lab 107B: https://docs.google.com/spreadsheets/d/15V6qR37-pPW0t3DAAuH_ElyqToTBgtFp0DYLu3iR0Og/edit?usp=sharing Lab 108B: https://docs.google.com/spreadsheets/d/1TFQYLia1HG1GA6hPGy1WwoFYS-qav5fvcbshwrqD7zs/edit?usp=sharing Lab 109B: https://docs.google.com/spreadsheets/d/1Axn4186utxbIUGcKu_SNp_u2pd7rtTPNsSlv4HTmvps/edit?usp=sharing Lab 110B: https://docs.google.com/spreadsheets/d/1pwv2WJbpy_oZZT7D5TN0Gb8-vh-sFx5SkK60Fkr51AI/edit?usp=sharing

Your task

1. Randomly generate 3 simple random samples of size $n = 10$ from the population. Calculate the average number of siblings for each of your samples. We wrote code to start you off, **but you need to replace the three instances of NULL with calculations** to compute the sample mean (`sample_mean_heights`), the lower confidence interval (`lower_CI`) and the upper confidence interval (`upper_CI`). Hint: Review the above section for tips on how to calculate the CI if you forget. Once you do this, you can simply copy and paste 3 times to generate three randomly-drawn samples and their sample means.

```
sample_size <- 10
critical_value <- 1.96
size_10 <- sample_n(alameda_pop, sample_size, replace = FALSE)
size_10 %>% summarise(mean_heights = mean(height)) %>%
mutate(lower_CI = mean_heights - critical_value*known_pop_sd/sqrt(sample_size),
       upper_CI = mean_heights + critical_value*known_pop_sd/sqrt(sample_size)
)
```

```
##   mean_heights lower_CI upper_CI
## 1     70.69339 68.96641 72.42037
```

Navigate to the google sheet for your lab and add the mean and its CI for your three samples. Once this is done enough times, the GSI can make a plot of the CIs and see how many contain the true value for the mean. Based on this plot:

- What proportion of the confidence intervals contain the mean?

For the lab section I attended, 96% of the confidence intervals contain the mean.

2. Repeat for a sample size of $n = 50$. In the code chunk below, generalize your code from the previous chunk to create three samples, this time of size 50:

```
sample_size <- 50
critical_value <- 1.96
size_50 <- sample_n(alameda_pop, sample_size, replace = FALSE)
size_50 %>% summarise(mean_heights = mean(height)) %>%
mutate(lower_CI = mean_heights - critical_value*known_pop_sd/sqrt(sample_size),
       upper_CI = mean_heights + critical_value*known_pop_sd/sqrt(sample_size))
```

```
##   mean_heights lower_CI upper_CI
## 1     69.54817 68.77585  70.3205
```

After you calculated your 3 sample means, navigate to the Google sheet from before, but switch to the sheet with $n = 50$ (you can switch tabs in the bottom left). Add your data for $n = 50$.

Once this is done, the GSI will plot these data, now with $n = 50$

- What proportion of the confidence intervals contain the mean?

For our lab section, 97.5% of the confidence intervals contain the mean.

- How do the average widths of the CI's compare for $n = 50$ versus $n = 10$?

The average width of the CI for n=50 is smaller than that of n=10.

- What would happen to the confidence intervals if $n = 500$?

If n=500, the confidence interval would be even smaller than both n=10 and n=50.

Check your score

Click on the middle icon on the top right of this code chunk (with the downwards gray arrow and green bar) to run all your code in order. Then, run this chunk to check your score.

```
# Just run this chunk.
total_score()
```

```
##                     Test Points_Possible          Type
## Problem 1 NOT YET GRADED               1 free-response
## Problem 2 NOT YET GRADED               1 free-response
## Problem 3 NOT YET GRADED               1 free-response
## Problem 4         PASSED               1    autograded
## Problem 5         PASSED               1    autograded
## Problem 6         PASSED               1    autograded
```

Submission

For assignments in this class, you'll be submitting using the **Terminal** tab in the pane below. In order for the submission to work properly, make sure that:

1. Any image files you add that are needed to knit the file are in the `src` folder and file paths are specified accordingly.
2. You **have not changed the file name** of the assignment.
3. The file is saved (the file name in the tab should be **black**, not red with an asterisk).
4. The file knits properly.

Once you have checked these items, you can proceed to submit your assignment.

1. Click on the **Terminal** tab in the pane below.
2. Copy-paste the following line of code into the terminal and press enter.

cd; cd ph142-fa20/lab/lab07; python3 turn_in.py

3. Follow the prompts to enter your Gradescope username and password. When entering your password, you won't see anything come up on the screen–don't worry! This is just for security purposes–just keep typing and hit enter.
4. If the submission is successful, you should see "Submission successful!" appear as output.
5. If the submission fails, try to diagnose the issue using the error messages–if you have problems, post on Piazza.

The late policy will be strictly enforced, **no matter the reason**, including submission issues, so be sure to submit early enough to have time to diagnose issues if problems arise.