STAT100B Lab2 Sampling Distributions

First Name __________ Last Name ___________SID (last 4 digits only)_____

In this lab, we investigate the ways in which the statistics from a random sample of data can serve as point estimates for population parameters. We're interested in formulating a *sampling distribution* of our estimate in order to learn about the properties of the estimate, such as its distribution.

Setting a seed: We will take some random samples and build sampling distributions in this lab, which means you should set a seed on top of your lab. If this concept is new to you, search it online or ask your TA.

```
require(stats)
set.seed(60)
```

The data

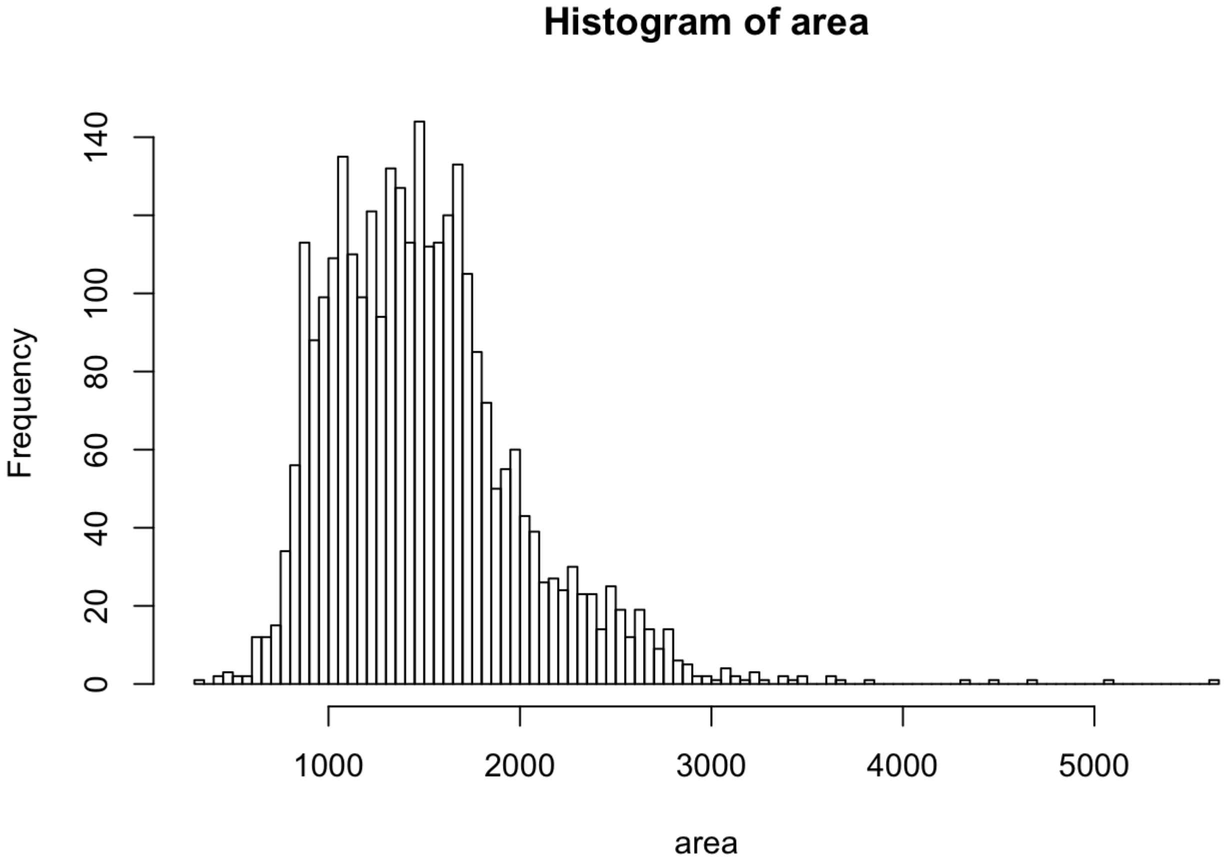
We consider real estate data from the city of Ames, Iowa. The details of every real estate transaction in Ames is recorded by the City Assessor's office. Our particular focus for this lab will be all residential home sales in Ames between 2006 and 2010. This collection represents our population of interest. In this lab we would like to learn about these home sales by taking smaller samples from the full population. Let's load the data.

```
load(url("https://stat.duke.edu/~mc301/data/ames.RData"))
```

We see that there are quite a few variables in the data set, enough to do a very in-depth analysis. For this lab, we'll restrict our attention to just two of the variables: the above ground living area of the house in square feet (`area`) and the sale price (`price`).

We can explore the distribution of areas of homes in the population of home sales visually and with summary statistics. Let's first create a visualization, a histogram:

```
area <- ames$area
hist(area, breaks = 100)
```



Let’s also obtain some summary statistics. Note that we can do this using the `summarise` function. We can calculate as many statistics as we want using this function, and just string along the results. Some of the functions below should be self explanatory (like `mean` , `median` , `sd` , `IQR` , `min` , and `max`). A new function here is the `quantile` function which we can use to calculate values corresponding to specific percentile cutoffs in the distribution. For example `quantile(x, 0.25)` will yield the cutoff value for the 25th percentile (Q1) in the distribution of x. Finding these values are useful for describing the distribution, as we can use them for descriptions like *“the middle 50% of the homes have areas between such and such square feet”*.

```
summary(area)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334    1126    1442    1500    1743    5642
```

Exercise 1 Describe this population distribution using the visualization and the summary statistics. (1pt) You don’t have to use all of the summary statistics in your description, you will need to decide which ones are relevant based on the shape of the distribution. Make sure to include the plot (2pts) and the summary statistics output (2pts) in your report along with your narrative.

The unknown sampling distribution

In this lab we have access to the entire population, but this is rarely the case in real life. Gathering information on an entire population is often extremely costly or impossible. Because of this, we often take a sample of the population and use that to understand the properties of the population.

If we were interested in estimating the mean living area in Ames based on a sample, we can use the following command to survey the population.

```
samp1 <- sample(area, 50)
```

This command collects a simple random sample of size 50 from the `ames` dataset `area` , which is assigned to `samp1` . This is like going into the City Assessor's database and pulling up the files on 50 random home sales. Working with these 50 files would be considerably simpler than working with all 2930 home sales.

Exercise 2

Describe the distribution of area in this sample. How does it compare to the distribution of the population? Show your code to find the mean and median of variable area in this sample.(1pt) Show your code to make a histogram of area (1pt) Distribe the distribution of area in this sample.(1pt)

If we're interested in estimating the average living area in homes in Ames using the sample, our best single guess is the sample mean.

Depending on which 50 homes you selected, your estimate could be a bit above or a bit below the true population mean of 1499.69 square feet. In general, though, the sample mean turns out to be a pretty good estimate of the average living area, and we were able to get it by sampling less than 3% of the population.

Exercise 3

Would you expect the mean of your sample to match the mean of another team's sample? (1pt) Why, or why not? (1pt) If the answer is no, would you expect the means to just be somewhat different or very different?

Exercise 4

Take a second sample, also of size 50, and call it `samp2` . How does the mean of `samp2` compare with the mean of `samp1` ? (1pt) Show your code to build samp2 and find the mean of `samp2` . (2pts)

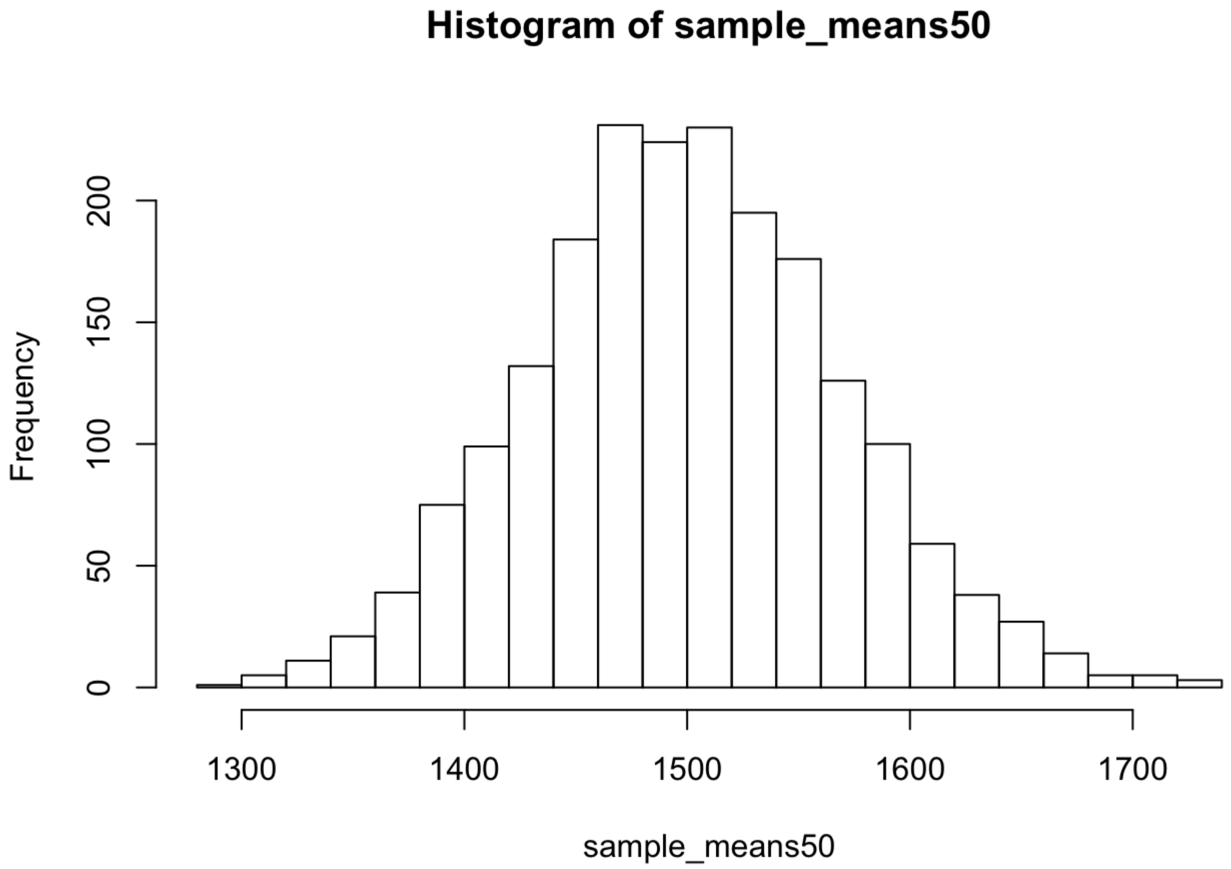
Exercise 5

Suppose we took two more samples, one of size 100 and one of size 1000. Which would you think would provide a more accurate estimation of the population mean? (1pt)

Not surprisingly, every time we take another random sample, we get a different sample mean. It's useful to get a sense of just how much variability we should expect when estimating the population mean this way. The distribution of sample means, called the *sampling distribution*, can help us understand this variability. In this lab, because we have access to the population, we can build up the sampling distribution for the sample mean by repeating the above steps many times. Here we will generate 2000 samples and compute the sample mean of each. Note that we are sampling with replacement, `replace = TRUE` since sampling distributions are constructed with sampling with replacement.

```
# build an empty vector to hold sample means
num_samples <- 2000
sample_means50 <- rep(0, num_samples)

# generate 2000 samples of size 50
# calculate sample means and store them in vector sample_mean50
for (i in 1:num_samples){
  temp_samp <- sample(area, 50)
  sample_means50[i] <- mean(temp_samp)
}
# visualize the sampling distribution
hist(sample_means50, breaks = 20)
```



Here we use R to take 2000 samples of size 50 from the population, calculate the mean of each sample, and store each result in a vector called `sample_means50` . On the next page, we'll review how this set of code works.

Exercise 6

How many elements are there in `sample_means50` ? (1pt) Describe the shape of the sampling distribution (1pt) The sampling distribution is centered at ___________. (1pt)

Interlude: The `for` loop

Let’s take a break from the statistics for a moment to let that last block of code sink in. The idea behind the `for` loop is *repetition*: it allows you to execute a line of code as many times as you want and put the results in a data frame. In the case above, we wanted to repeatedly take a random sample of size 50 from `area` and then save the mean of that sample into the `sample_means50` vector.

Without the `for` loop, this would be painful. First, we’d have to create an empty vector filled with 0s to hold the 2000 sample means. Then, we’d have to compute each of the 2000 sample means one line at a time, putting them individually into the slots of the `sample_means50` vector:

```
sample_means50 <- rep(0, 2000)

# generate the first sample
temp_samp <- sample(area, 50)
sample_means50[1] <- mean(temp_samp)

# generate the second sample
temp_samp <- sample(area, 50)
sample_means50[2] <- mean(temp_samp)

# generate the third sample
temp_samp <- sample(area, 50)
sample_means50[3] <- mean(temp_samp)
###
#...
# until you generate 2000 samples
```

With the `for` loop, these thousands of lines of code are compressed into one short chunck:

```
sample_means50 <- rep(0, 2000)
for (i in 1:num_samples){
  temp_samp <- sample(area, 50)
  sample_means50[i] <- mean(temp_samp)
}
```

On your own (Lab B Exercises)

So far, we have only focused on estimating the mean living area in homes in Ames. Now you’ll try to estimate the mean home price.

Note that while you might be able to answer some of these questions using the app you are expected to write the required code and produce the necessary plots and summary statistics. You are welcomed to use the app for exploration.

1. Take a sample of size 15 from the population and calculate the mean `price` of the homes in this sample. Show your code to obtain the mean of this sample (1pt) Using this sample, what is your best point estimate of the population mean of prices of homes? (1pt)
2. Since you have access to the population, simulate the sampling distribution for $\bar{x}_{price}$ by taking 2000 samples from the population of size 15 and computing 2000 sample means. Store these means in a vector called `sample_means15` . Show your code of the sampling process.(2pts) Show your code of plotting

sampling distribution of size 15. (1pt) Describe the shape of this sampling distribution (1pt) Finally, show your code to calculate and report the population mean.(2pts)

3. Change your sample size from 15 to 150, then compute the sampling distribution using the same method as above, and store these means in a new vector called `sample_means150` . Show your code here for sampling and ploting (4pts) Comparing to the sampling distribution from a sample size of 15, the spread of this sampling distribution is _____________ (smaller/larger) when sample size increased to 150.(1pt) The shape of this sampling distribution is __________(1pt) A. roughly bell shaped B.not close to bell shaped
4. If we’re concerned with making estimates that are more often close to the true value, would we prefer a sampling distribution with a large or small spread? (1pt)