Complete all ***Exercises***, and submit answers to **Questions** on the Coursera platform. Note that the order of the choices in multiple choice questions may be different on the Coursera platform than the order in this document.

Lab 3A - Foundations for inference: Sampling distributions

In Part A of 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.

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 (which is rare to have access to), but we will also work with smaller samples from this population. Let's load the data.

```
load(url("http://www.openintro.org/stat/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 (`Gr.Liv.Area`) and the sale price (`SalePrice`). To save some effort throughout the lab, create two variables with short names that represent these two variables.

```
area <- ames$Gr.Liv.Area
price <- ames$SalePrice
```

Let's look at the distribution of area in our population of home sales by calculating a few summary statistics and making a histogram.

```
summary(area)
hist(area)
```

Question 1 [MULTIPLE CHOICE] Which of the following is *false*?

(a) The distribution of areas of houses in Ames is unimodal and right-skewed.

(b) 50% of houses in Ames are smaller than 1,500 square feet.

(c) The middle 50% of the houses range between approximately 1,130 square feet and 1,740 square feet.

(d) The IQR is approximately 610 square feet.

(e) The smallest house is 334 square feet and the largest is 5,642 square feet.

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 `sample` function to sample from the population.

```
samp0 <- sample(area, 50)
```

This command collects a simple random sample of size 50 from the vector `area`, which is assigned to `samp0`. This is like going into the City Assessor's database and pulling up the files on 50 random home sales. If we didn't have access to the population data, working with these 50 files would be considerably simpler than having to go through all 2930 home sales.

Now that you've taken a sample, take another sample and compare the two. Are the samples the same? Why?

Now we're ready to sample:

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

Exercise Describe the distribution of this sample? How does it compare to the distribution of the population?

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

```
mean(samp1)
```

Depending on which 50 homes you selected, your estimate could be a bit above or a bit below the true population mean of approximately 1,500 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.

Question 2 [MULTIPLE CHOICE] Suppose we took two more samples, one of size 100 and one of size 1000. Which would you think would provide a more accurate estimate of the population mean?

(a) Sample size of 50

(b) Sample size of 100

(c) Sample size of 1000

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 5000 samples and compute the sample mean of each.

```
sample_means50 <- rep(NA, 5000)
for (i in 1:5000) {
    samp <- sample(area, 50)
    sample_means50[i] <- mean(samp)
}
hist(sample_means50)
```

If you would like to adjust the bin width of your histogram to show a little more detail, you can do so by changing the `breaks` argument.

```
hist(sample_means50, breaks = 25)
```

Here we use R to take 5000 samples of size 50 from the population, calculate the mean of each sample, and store each result in a vector called `sample_means50`, using what we call a *for loop*. If this is completely new to you, do not fear, on the next page we'll review in detail how this set of code works.

Exercise Describe the sampling distribution (the distribution of the sample means that you just created), and be sure to specifically note its center.

Interlude: The `for` loop

Let's take a break from the statistics for a moment to let that last block of code sink in. You have just run your first for loop, a cornerstone of computer programming. The idea behind the for loop is *iteration*: it allows you to execute code as many times as you want without having to type out every iteration. In the case above, we wanted to iterate the two lines of code inside the curly braces that take a random sample of size 50 from `area` then save the mean of that sample into the `sample_means50` vector. Without the for loop, this would be painful:

```
sample_means50 <- rep(NA, 5000)
samp <- sample(area, 50)
sample_means50[1] <- mean(samp)
samp <- sample(area, 50)
sample_means50[2] <- mean(samp)
samp <- sample(area, 50)
sample_means50[3] <- mean(samp)
samp <- sample(area, 50)
```

```
sample_means50[4] <- mean(samp)
```

and so on...

With the for loop, these thousands of lines of code are compressed into a handful of lines. We've added one extra line to the code below, which prints the variable `i` during each iteration of the for loop. Run this code.

```
sample_means50 <- rep(NA, 5000)
for (i in 1:5000) {
    samp <- sample(area, 50)
    sample_means50[i] <- mean(samp)
    print(i)
}
```

Let's consider this code line by line to figure out what it does. In the first line we *initialized a vector*. In this case, we created a vector of 5000 NAs called `sample_means50`. This vector will store values generated within the for loop. NA means *not available*, and in this case they're used as placeholders until we fill in the values with actual sample means. NA is also often used for missing data in R.

The second line calls the for loop itself. The syntax can be loosely read as, "for every element `i` from 1 to 5000, run the following lines of code". You can think of `i` as the counter that keeps track of which loop you're on. Therefore, more precisely, the loop will run once when `i=1`, then once when `i=2`, and so on up to `i=5000`.

The body of the for loop is the part inside the curly braces, and this set of code is run for each value of `i`. Here, on every loop, we take a random sample of size 50 from `area`, take its mean, and store it as the i^{th} element of `sample_means50`.

In order to display that this is really happening, we asked R to print `i` at each iteration. This line of code is optional and is only used for displaying what's going on while the for loop is running.

The for loop allows us to not just run the code 5000 times, but to neatly package the results, element by element, into the empty vector that we initialized at the outset.

> *Exercise* To make sure you understand what you've done in this loop, try running a smaller version. Initialize a vector of 100 NAs called `sample_means_small`. Run a loop that takes a sample of size 50 from `area` and stores the sample mean in `sample_means_small`. Print the output to your screen (type `sample_means_small` into the console and press enter).

> **Question 3** [MULTIPLE CHOICE] How many elements are there in this object called `sample_means_small`?
>
> (a) 0
>
> (b) 30
>
> (c) 50
>
> (d) 100
>
> (e) 5,000

Question 4 [MULTIPLE CHOICE] Which of the following is *true* about the elements in the sampling distributions you created?

(a) Each element represents a mean square footage from a simple random sample of 50 houses.

(b) Each element represents the square footage of a house.

(c) Each element represents the true population mean of square footage of houses.

Sample size and the sampling distribution

Mechanics aside, let's return to the reason we used a for loop: to compute a sampling distribution, specifically, this one.

```
hist(sample_means50)
```

The sampling distribution that we computed tells us much about estimating the average living area in homes in Ames. Because the sample mean is an unbiased estimator, the sampling distribution is centered at the true average living area of the the population, and the spread of the distribution indicates how much variability is induced by sampling only 50 home sales.

To get a sense of the effect that sample size has on our sampling distribution, let's build up two more sampling distributions: one based on a sample size of 10 and another based on a sample size of 100.

```
sample_means10 <- rep(NA, 5000)
sample_means100 <- rep(NA, 5000)
for (i in 1:5000) {
    samp <- sample(area, 10)
    sample_means10[i] <- mean(samp)
    samp <- sample(area, 100)
    sample_means100[i] <- mean(samp)
}
```

Here we're able to use a single for loop to build two distributions by adding additional lines inside the curly braces. Don't worry about the fact that `samp` is used for the name of two different objects. In the second command of the for loop, the mean of `samp` is saved to the relevant place in the vector `sample_means10`. With the mean saved, we're now free to overwrite the object `samp` with a new sample, this time of size 100. In general, anytime you create an object using a name that is already in use, the old object will get replaced with the new one, i.e. R will *write over* the existing object with the new one, which is something you want to be careful about if you don't intend to do so.

To see the effect that different sample sizes have on the sampling distribution, plot the three distributions on top of one another.

```
par(mfrow = c(3, 1))
xlimits = range(sample_means10)
hist(sample_means10, breaks = 20, xlim = xlimits)
hist(sample_means50, breaks = 20, xlim = xlimits)
hist(sample_means100, breaks = 20, xlim = xlimits)
```

The first command specifies that you'd like to divide the plotting area into 3 rows and 1 column of plots[†].

[†]You may need to stretch your plotting window to accommodate the extra plots. To return to the default setting of plotting one plot at a time, run the following command:

The `breaks` argument specifies the number of bins used in constructing the histogram. The `xlim` argument specifies the range of the x-axis of the histogram, and by setting it equal to `xlimits` for each histogram, we ensure that all three histograms will be plotted with the same limits on the x-axis.

Question 5 [MULTIPLE CHOICE] It makes intuitive sense that as the sample size increases, the center of the sampling distribution becomes a more reliable estimate for the true population mean. Also as the sample size increases, the variability of the sampling distribution _____.

(a) decreases

(b) increases

(c) stays the same

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.

Exercise Take a random sample of size 50 from `price`. Using this sample, what is your best point estimate of the population mean?

Exercise Since you have access to the population, simulate the sampling distribution for $\bar{x}_{price}$ by taking 5000 samples from the population of size 50 and computing 5000 sample means. Store these means in a vector called `sample_means50`. Plot the data, then describe the shape of this sampling distribution. Based on this sampling distribution, what would you guess the mean home price of the population to be?

Exercise Change your sample size from 50 to 150, then compute the sampling distribution using the same method as above, and store these means in a new vector called `sample_means150`. Describe the shape of this sampling distribution, and compare it to the sampling distribution for a sample size of 50. Based on this sampling distribution, what would you guess to be the mean sale price of homes in Ames?

Question 6 [MULTIPLE CHOICE] Which of the following is false?

(a) The variability of the sampling distribution with the smaller sample size (`sample_means50`) is smaller than the variability of the sampling distribution with the larger sample size (`sample_means150`).

(b) The means for the two sampling distribtuions are roughly similar.

(c) Both sampling distributions are symmetric.

```
par(mfrow = c(1, 1))
```

End of Lab Survey

The following questions are not graded, but your feedback is very much appreciated and immensely useful for the development of the course.

Question 7 [MULTIPLE CHOICE] This lab covered material that is covered in the class.

(a) Strongly Disagree

(b) Disagree

(c) Neutral

(d) Agree

(e) Strongly Agree

Question 8 [MULTIPLE CHOICE] The lab improved your understanding of these topics.

(a) Strongly Disagree

(b) Disagree

(c) Neutral

(d) Agree

(e) Strongly Agree

Question 9 [MULTIPLE CHOICE] The instructions were clear and it was easy to understand what was wanted.

(a) Strongly Disagree

(b) Disagree

(c) Neutral

(d) Agree

(e) Strongly Agree

Question 10 [MULTIPLE CHOICE] The data were relevant and interesting to me.

(a) Strongly Disagree

(b) Disagree

(c) Neutral

(d) Agree

(e) Strongly Agree

Question 11 [MULTIPLE CHOICE] The length of time took to complete lab.

(a) Less than 30 minutes

(b) Between 30 minutes and 1 hour

(c) Between 1 hour and 2 hours

(d) More than 2 hours