Foundations for statistical inference - Sampling distributions

Your reproducible lab report: Before you get started, download the R Markdown template for this lab. Remember all of your code and answers go in this document:

```
download.file("https://dyurovsky.github.io/85309/post/rmd/lab4.Rmd",
              destfile = "lab4.Rmd")
```

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.

Since the markdown file will run the code, and generate a new sample each time you Knit it, you should also "set a seed" **before** you sample. Read more about setting a seed below.

> **Setting a seed:** Setting a seed will cause `R` to select the same sample each time you knit your document. This will make sure your results don't change each time you knit, and it will also ensure reproducibility of your work(by setting the same seed it will be possible to reproduce your results). You can set a seed like this:
>
> ```
> set.seed(85309)
> ```
>
> The number you use is completely arbitrary. If you need inspiration, you can use your ID, birthday, or just a random string of numbers. The important thing is that you use each seed only once. Remember to do this **before** you sample.

Getting started

As usual, we're going to load the `tidyverse` package for data manipulation. We'll also be reading in a dataset to work with just like we usually do.

```
ames <- read_csv("https://dyurovsky.github.io/85309/data/lab4/ames.csv")
```

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.

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`).

```
ggplot(ames, aes(x = area)) +
  geom_histogram(binwidth = 250)
```

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 combine 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 is 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*".

```
ames %>%
  summarise(mu = mean(area),
            pop_med = median(area),
            sigma = sd(area),
            pop_iqr = IQR(area),
            pop_min = min(area),
            pop_max = max(area),
            pop_q1 = quantile(area, 0.25),  # first quartile, 25th percentile
            pop_q3 = quantile(area, 0.75))  # third quartile, 75th percentile
```

Exercise 1

Describe this population distribution using the visualization and the summary statistics. 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 and the summary statistics output 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 `sample_n` command to survey the population.

```
samp1 <- ames %>%
  sample_n(50)
```

This command collects a simple random sample of size 50 from the `ames` dataset, and assigns the result 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? **Hint:** `sample_n` function takes a random sample of observations (i.e. rows) from the dataset, you can still refer to the variables in the dataset with the same names. Code you used in the previous exercise will also be helpful for visualizing and summarizing the sample, however be careful to not label values `mu` and `sigma` anymore since these are sample statistics, not population parameters. You can customize the labels of any of the statistics to indicate that these come from the sample.

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.

```
samp1 %>%
  summarise(x_bar = mean(area))
```

Depending on which 50 homes you selected, your estimate could be a bit above or a bit below the true population mean of 1499.6904437 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

Take a second sample, also of size 50, and call it `samp2`. How does the mean of `samp2` compare with the mean of `samp1`? 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?

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 of the mean*, 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 1,000 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.

```
sample_50 <- function() {
  ames %>%
    sample_n(50, replace = TRUE) %>%
    summarise(x_bar = mean(area)) %>%
    pull()
}

sample_means50 <- tibble(sample = 1:1000,
                         mean = replicate(1000, sample_50()))

ggplot(sample_means50, aes(x = mean)) +
  geom_histogram()
```

Here we use `R` to take 1,000 samples of size 50 from the population, calculate the mean of each sample, and store all of the results together in a tibble called `sample_means50`. Next, we review how this set of code works.

Exercise 4

How many elements are there in `sample_means50`? Describe the sampling distribution, and be sure to specifically note its center. Make sure to include a plot of the distribution in your answer.

Interlude: The `replicate` function

Let's take a break from the statistics for a moment to let that last block of code sink in. The idea behind the replicate function is *repetition*: it allows you to execute a line of code as many times as you want and put the results in an array. 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 the `sample_means50` tibble.

With the `replicate` function, we can do all of this one line of code. To make things more modular, we wrote the `sample_50` function to draw a conceptual distinction between what happens for each sample, and the process of drawing many samples. But, we could have written this all in one command:

```
sample_means50 <- tibble(sample = 1:1000,
                         mean = replicate(1000, ames %>%
                                            sample_n(50, replace = TRUE) %>%
                                            summarise(x_bar = mean(area)) %>%
                                            pull()))
```

Note that for each of the 1000 times we computed a mean, we did so from a **different** sample!

Exercise 5

To make sure you understand how sampling distributions are built, and exactly what the `sample_n` and `replicate` function do, try modifying the code to create a sampling distribution of **25 sample means** from **samples of size 10**, and put them in a tibble named `sample_means_small`. Print the output. How many observations are there in this object called `sample_means_small` ? What does each observation represent?

Sample size and the sampling distribution

Mechanics aside, let’s return to the reason we used the `replicate` function: to compute a sampling distribution, specifically, this one.

```
ggplot(sample_means50, aes(x = mean)) +
  geom_histogram()
```

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 population, and the spread of the distribution indicates how much variability is induced by sampling only 50 home sales.

In the remainder of this section we will work on getting a sense of the effect that sample size has on our sampling distribution. First, let’s make the sampling function more general so that we look at changes in our estimates of mean area for samples of size 10, 50, and 100. The new `sample_varying` function takes a number `n` as input, and produces the mean for a sample of that size. We can then pass it different values of `n` and see how the sampling distribution changes.

```
sample_varying <- function(n) {
  ames %>%
    sample_n(n, replace = TRUE) %>%
    summarise(x_bar = mean(area)) %>%
    pull()
}

sample_varying(50)
```

```
## [1] 1432.32
```

Exercise 6

Try using `replicate` to see what happens to the sampling distribution when you try these three different values. Use 1000 replications. What does each observation in the sampling distribution represent? How does the mean, standard error, and shape of the sampling distribution change as the sample size increases? How (if at all) do these values change if you increase the number of simulations? (You do not need to include plots in your answer.)

More Practice

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 7

Take a sample of size 15 from the population and calculate the mean `price` of the homes in this sample. Using this sample, what is your best point estimate of the population mean of prices of homes?

Exercise 8

Since you have access to the population, simulate the sampling distribution for $\bar{x}_{price}$ by taking 1000 samples from the population of size 15 and computing 1000 sample means. Store these means in a vector called `sample_means15`. 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? Finally, calculate and report the population mean.

Exercise 9

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`. Describe the shape of this sampling distribution, and compare it to the sampling distribution for a sample size of 15. Based on this sampling distribution, what would you guess to be the mean sale price of homes in Ames?

Exercise 10

Of the sampling distributions from Exercises 8 and 9, which has a smaller spread? 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?