MAT 128: Lab 5: Sample Mean and Variance

Motivating question: How do the mean and variance of a sample differ from the mean and variance of the population.

For this lab, we will use the 2016 Green Taxi Trip Data set from Lab 3. You can either use the CSV file you already downloaded, or follow the instructions at the beginning of Lab 3 to download it again. You only need taxi trip data for one day.

The *population* is all subjects possessing the characteristics we want to study. In this case, we will consider all green taxi trips taken on Feb. 8, 2016 as the population.

The *sample* is a group of subjects selected from the population. In this case, we will randomly choose some green taxi trips from our dataset.

Because we are studying how the sample mean and variance differ from the population mean and variance, we are using our entire dataset as the population. Usually, the entire dataset is the sample, and we care about a larger population. For example, we might be interested in the average length of green taxi trips for all of 2016, but the only data available is trips from January to June.

Part A: Sample mean

In a new Python file:

1) Write code to import the pandas library.

2) Write code to read the CSV file into a dataframe/variable called `taxi`.

3) Write code to compute and print out the mean of the Trip_distance column in taxi:

```
mean = taxi['Trip_distance'].mean()
print("The population mean is",mean)
```

4) We will now create a sample of 100 rows of the taxi dataframe. This is a new dataframe made up of 100 randomly chosen rows of the taxi dataframe.
taxi_sample = taxi.sample(100)

5) Compute and print the mean of the sample (the *sample mean*):

```
sample_mean = taxi_sample['Trip_distance'].mean()
print("The sample mean is",sample_mean)
```

How does the sample mean compare to the population mean?

Re-run your code several times. Do the sample mean and population mean always have the same relation?

To fully understand the relation of the sample mean with the population mean, we are going to use a *loop* to repeatedly take new samples and compute their means. We will then plot these. A *loop* repeats a section of code multiple times.

For example:

```
for i in range(10):
      print("Hello")
```

is a loop that prints Hello 10 times.

6) Our loop will look like:

```
for i in range(500):
      taxi_sample = taxi.sample(100)
      sample_mean = taxi_sample['Trip_distance'].mean()
      print("The sample mean is",sample_mean)
```

It takes a sample, computes its mean, and prints it out 500 times. Notice that we indent the code in the loop.

7) It is still hard to understand how the sample means and population mean relate, so we will plot the sample means. We first have to save them in a list.

Change your code to look like:

```
sample_means_list = []
for i in range(500):
      taxi_sample = taxi.sample(100)
      sample_mean = taxi_sample['Trip_distance'].mean()
      #print("The sample mean is",sample_mean)
      sample_means_list.append(sample_mean)
```

This creates a new, empty list called `sample_means_list` and adds the new sample mean to it each time with the line `sample_means_list.append(sample_mean)`

8) To plot the sample means as a histogram add this code after the loop, without an indent:

```
pd.Series(sample_means_list).plot.hist(title = "Sample means")
```

Notice we first make the list into a Series, and then plot it as usual.

9) Write code to add a vertical line to the histogram showing where the population mean is (see Lab 3).

How do the sample means relate to the population mean?
What happens if you increase the number of samples?
What happens if you decrease the number of samples?

What happens if you increase the size of the sample?
What happens if you decrease the size of the sample?

We will see later in the course that this is a visualization of something called the Central Limit Theorem.

Part B: Sample Variance

As we saw in Part A, as the sample size increases, the sample mean becomes a better and better estimator of the population mean. This is not true if we compute the variance of the sample using the definition of variance from last class:

Dataset: $x_1, x_2, x_3, \ldots, x_n$
Mean = μ
Variance = $\frac{(x_1-\mu)^2 + (x_2-\mu)^2 + \ldots + (x_n-\mu)^2}{n}$

1) To compute this variance using column operations:

```
num_trips = taxi['Trip_distance'].count()
population_var = ((taxi['Trip_distance'] -
mean)**2).sum()/num_trips
```

`**2` squares each entry in the column and `.sum()` adds up all entries once we have subtracted the mean from each row in `Trip_distance`, and squared the results.

Print the population variance out:

```
print("The population variance is",population_var)
```

2) What happens if we use the .var() command to compute the variance?

```
print("The population variance using .var() is", taxi['Trip_distance'].var())
```

The variance numbers are different!

3) .var() computes what is called the *sample variance*. The only difference is dividing by n-1 instead of n. We will see why shortly. You can compute the population variance using `.var(ddof=0)`:

```
population_var2 = taxi['Trip_distance'].var(ddof=0)
print("The correct population variance using .var() is",population_var2)
```

4) Add code to your loop from Part A to compute both the sample variance and the population variance (of the sample):

```
sample_var = taxi_sample['Trip_distance'].var()
pop_var_of_sample = taxi_sample['Trip_distance'].var(ddof=0)
```

5) Make two new lists to store the sample variance. Add the following above your loop:

```
sample_var_list = []
```

```
pop_var_of_sample_list = []
```
6) In your loop, add the sample variance and population variance of the sample into the lists:
```
sample_var_list.append(sample_var)
pop_var_of_sample_list.append(pop_var_of_sample)
```

7) Plot 2 histograms: one of `sample_var_list` and one of `pop_var_of_sample_list`

8) Compute the mean of each list, and add a vertical line at that point to the appropriate histogram:
```
mean_sample_var_list = pd.Series(mean_sample_var_list).mean()
mean_pop_var_of_sample_list =
pd.Series(pop_var_of_sample_list).mean()
```

9) Add a vertical line at the true population variance.

10) As you increase the sample size, what happens with the histograms?