Homework 3 (Intro to DS)

Colab Link:

Student Info: **NCU MIS 109403019** 鄒翔宇

Table of contents

Question 1.

- Are these samples representative of the population of earthquakes in the original table (that is, should we expect the mean to be close to the population mean)? (20 points)

Answer:

- sample1 is *not* representative of the population of earthquakes in the original table because it only takes the average of the largest earthquakes. It has a higher average than the population's.
- sample2 is more representative of the population of earthquakes in the original table than sample1 because it takes the average of the last 100 earthquakes, not only the largest ones.

Question 2.

- Write code to produce a sample of size 200 that is representative of the population. Then, take the mean of the magnitudes of the earthquakes in this sample. Assign these to representative_sample and representative_mean respectively. (20 points)

Answer:

```
representative_sample=earthquakes.sample(200,  with_replacement=False)
representative_mean=np.mean(representative_sample.column('mag'))
print("representative_mean:",  representative_mean)
```

```
representative_mean: 5.3231
```

Question 3.

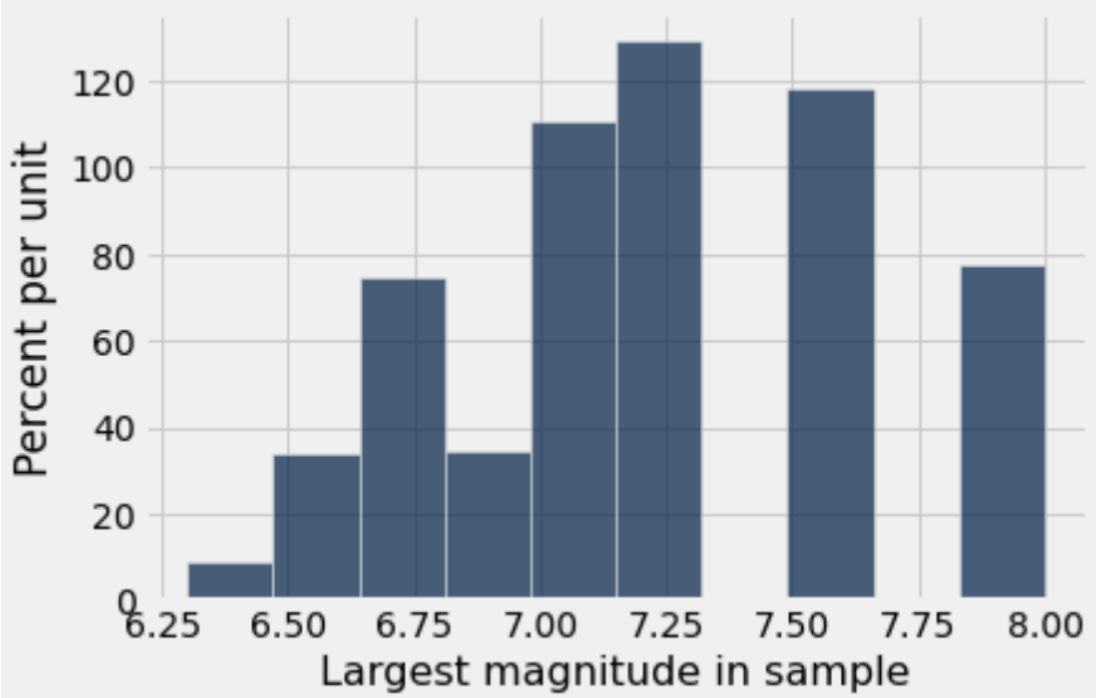
- Suppose we want to figure out what the biggest magnitude earthquake was in 2019, but we only have our representative sample of 200. Let's see if trying to find the biggest magnitude in the population from a random sample of 200 is a reasonable idea!
- Write code that takes many random samples from the earthquakes table and finds the maximum of each sample. You should take a random sample of size 200 and do this 5000 times. Assign the array of maximum magnitudes you find to maximums. (20 points)

Answer:

```
maximums = []
for i in np.arange(5000):
    representative_sample = earthquakes.sample(200, with_replacement=False)
    maximums.append(max(representative_sample.column('mag')))
```

Question 4.

- Run the following code to plot its histogram. (20 points)



Question 5.

- Now find the magnitude of the actual strongest earthquake in 2019 (not the maximum of a sample).
- This will help us determine whether a random sample of size 200 is likely to help you determine the largest magnitude earthquake in the population. (20 points)

Answer:

The magnitude of the actual strongest earthquake in 2019 is: 8.0

```
strongest_earthquake_magnitude_in_2019 = max(earthquakes.column('mag'))
print("The magnitude of the actual strongest earthquake in 2019 is: ", strongest_earthquake_magnitude_in_2019)
```

```
The magnitude of the actual strongest earthquake in 2019 is: 8.0
```

Question 6.

- Explain whether you believe you can accurately use a sample size of 200 to determine the maximum.
- What is one problem with using the maximum as your estimator? Use the histogram above to help answer.

Answer:

I don't believe that I can accurately use a sample size of 200 to determine the maximum. The histogram above shows that the probability of we get the maximum of the population from sampling with the size of 200 is not high. We can count that every value in the *maximums* array is equal to the max value of the population, and calculate the probability, it is 0.*1198*, which is pretty low.

And the problem with using the maximum as an estimator is that the result can be sensitive to the choice of starting values if confidence intervals for the parameters are desired.

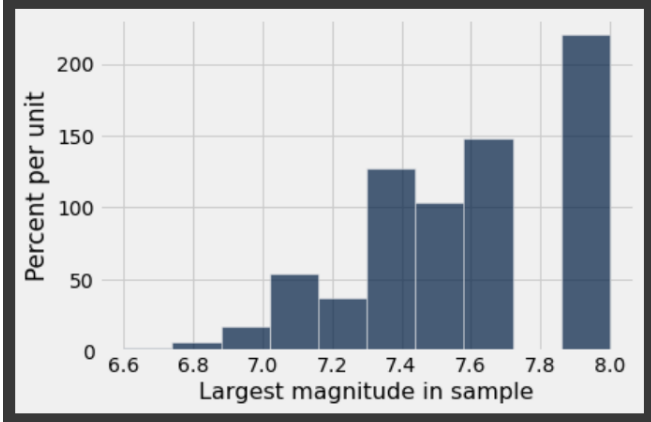
```
count_equal_to_population_max = 0
for i in range(len(maximums)):
    if maximums[i] == strongest_earthquake_magnitude_in_2019:
        count_equal_to_population_max += 1

count_equal_to_population_max / len(maximums)
```

```
0.1198
```

By the way, in order to increase the probability, we can set our sample size as more as possible. I set the sample size as 500 for example. The histogram clearly shows that the

probability is more than the front one, while it's still not accurate for determining the maximum of the population.

```
maximums_with_sample_500 = []
for i in np.arange(5000):
    representative_sample = earthquakes.sample(500, with_replacement=False)
    maximums_with_sample_500.append(max(representative_sample.column('mag')))
Table().with_column('Largest magnitude in sample', maximums_with_sample_500).hist('Largest magnitude in sample')
```




```
count_equal_to_population_max = 0
for i in range(len(maximums)):
    if maximums_with_sample_500[i] == strongest_earthquake_magnitude_in_2019:
        count_equal_to_population_max += 1

count_equal_to_population_max / len(maximums)
```

```
0.3084
```