STAT 207 Homework 7 [50 points] - Solutions

Central Limit Theorem, Confidence Intervals, and Hypotheses

Due: Friday, March 10 by noon (11:59 am) CST

Note: For this assignment, you should not use any built-in functions for calculating confidence intervals. You will not earn credit if you use the built-in confidence interval functions on this assignment.

Package Imports

Run the cell provided below to import packages needed for this assignment.

You may also need to read in additional packages below.

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import norm
from scipy.stats import t
```

Case Study 1: Preparing to Eat

We have a random sample of time spent preparing food and drink (in minutes) by American adults in the last 24 hours contained in the food_prep.csv file. Using this random sample, we will construct and understand a confidence interval in this Case Study.

1. Read in the data [1 point]

Read in the data for the food_prep.csv file below. The data is already cleaned, so you don't need to worry about cleaning the data. However, you should try to learn a little bit about the data below.

In [2]:
```
df = pd.read_csv('food_prep.csv')
df.head()
```

Out[2]:

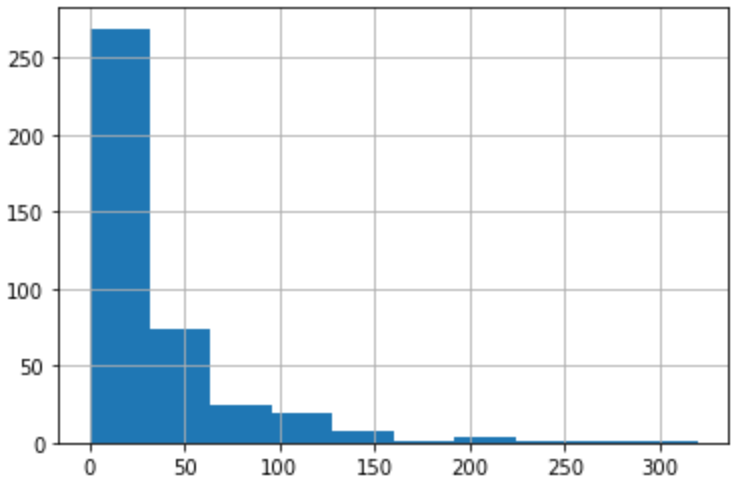
	Unnamed: 0	food_prep
0	1	15
1	2	2
2	3	60
3	4	45
4	5	0

In [3]:
```
df.shape
```

Out[3]:
```
(400, 2)
```

In [4]:
```
df['food_prep'].hist()
```

Out[4]:
```
<AxesSubplot:>
```



2. Construct a Confidence Interval [15 points]

We will construct an 88% confidence interval for the average time spent preparing food or drink in the last 24 hours.

a) Define the parameter of interest for our confidence interval.

Note: the parameter definition should be provided in words and include a symbol.

The parameter in this case is μ, the population mean time spent preparing food and drink (in minutes) by all American adults in the last 24 hours.

b) Construct an 88% confidence interval for the average time spent preparing food or drink in the last 24 hours.

In [5]:
```
df = pd.read_csv('food_prep.csv')
samp_mean = df['food_prep'].mean()
samp_std = df['food_prep'].std()
samp_size = df.shape[0]
multiplier = t.ppf(0.94, 399)
```

In [6]:
```
lower = samp_mean - multiplier * samp_std / samp_size ** (1/2)
upper = samp_mean + multiplier * samp_std / samp_size ** (1/2)
print("88% Confidence interval: (", lower, ", ", upper, ")")
```

```
88% Confidence interval: ( 27.94671414818182 ,  34.893285851818185 )
```

c) Put your confidence interval into words. That is, interpret your confidence interval.

I am 88% confident that the true population mean time spent preparing food or drink in the last 24 hours for all American adults falls inside the confidence interval of (27.95, 34.89) minutes.

d) Interpret the confidence level for your interval calculation.

If I were to repeat this process many times, taking repeated random samples of size 400 and creating an 88% confidence interval for each of the random samples, then I would expect 88% of the resulting intervals

to contain the true population mean time spent preparing food and drink (in minutes) by all American adults in the last 24 hours.

e) State & check the conditions (assumptions) for creating your confidence interval.

In [7]:
```
df.shape[0]
```

Out[7]:
```
400
```

Conditions are:

- Random sample of observations from the population
- Sample size is less than 10% of our population size
- Sample size is at least 30 OR population is normally distributed

Check:

- We are told that we have a random sample from the population
- Our sample size (400) is less than 10% of the population size of all American adults
- Our sample size (400) is at least 30. This is helpful, since there is not evidence that the population is normally distributed, at least based on our sample distribution.

f) What distribution did you use to find your multiplier above? Explain why you used this distribution.

To find the multiplier, I used the t distribution with $n - 1 = 399$ degrees of freedom.

I used this distribution because I also had to estimate the standard deviation for the distribution using another statistic (s to estimate σ), which means that I want to add a little extra wiggle room to my confidence interval by using the t distribution.

Note: with a proper argument about the fact that the degrees of freedom is very large and the t(399) and standard normal distributions have only minute differences, the standard normal distribution can be accepted as a valid solution.

Case Study 2: Colleges and Universities

We will use a random sample of post-secondary education facilities (colleges & universities) from the United States, including Puerto Rico & other US territories. The **colleges.csv** file contains a random sample of 135 post-secondary education facilities from the US. This data comes from: https://www.kaggle.com/yamqwe/colleges-and-universitiese

We will focus on two variables in particular:

- the total dorm capacity of the college or university (`DORM_CAP`)
- the total number of students enrolled (`TOT_ENROLL`)

3. Read and prepare the data [3 points]

You should read in the data from the **colleges.csv** file, perform any cleaning that needs to take place, and create the following variable:

- the variable `dorms`, a logical (Boolean) variable that indicates if the college or university offers dorms (dorm capacity is larger than 0)

Hint: For the data cleaning, we only care about the 2 variables defined above. Consider reasonable values for the `TOT_ENROLL` variable in particular, and remove any colleges or universities that do not have reasonable values recorded.

In [8]:
```
df = pd.read_csv('colleges.csv')
df.shape
```

Out[8]:
```
(135, 47)
```

In [9]:
```
df = df[["DORM_CAP", "TOT_ENROLL"]]
```

In [10]:
```
df.dtypes
```

Out[10]:
```
DORM_CAP      int64
TOT_ENROLL    int64
dtype: object
```

The two variables are both of type int64, so a numeric variable. Let's look at some values that the "TOT_ENROLL" variable takes. In particular, notice the values of 0 for TOT_ENROLL. We do need to remove those colleges that don't have any students.

In [11]:
```
df["TOT_ENROLL"].describe()
```

Out[11]:
```
count      135.000000
mean      2820.822222
std       5822.892201
min          0.000000
25%        154.000000
50%        507.000000
75%       2443.000000
max      40695.000000
Name: TOT_ENROLL, dtype: float64
```

In [12]:
```
df = df[df["TOT_ENROLL"] > 0]
df.shape
```

Out[12]:
```
(128, 2)
```

Notice that 7 colleges were removed from the data frame. Now, let's make the additional variable requested.

In [13]:
```
df["dorms"] = df["DORM_CAP"] > 0
```

4. Do Colleges Offer Dorms? [18 points]

Based on our remaining random sample of colleges and universities, we will estimate the proportion of colleges and universities that offer dorms as housing with 75% confidence.

a) Define our parameter of interest in the context of the problem.

p is the population proportion of all colleges and universities that offer dorms as housing.

b) State & check the conditions before generating your confidence interval.

1. We have a random sample of colleges and universities. [CHECK, as defined by the background described above.]
2. Our sample size (n) of 128 is less than 10% of the population of all colleges & universities. While we don't know our population size, it is reasonable to say that there are more than 1280 colleges & universities in the US. [CHECK]
3. We have a large enough sample size for a Normal approximation of the sampling distribution. [CHECK]

- $n\hat{p} \geq 10 \Rightarrow 40 \geq 10$
- $n(1-\hat{p}) \geq 10 \Rightarrow 88 \geq 10$

In [14]:
```
phat = df["dorms"].mean()
print(phat)
n = df["dorms"].shape[0]
print(n)
```

```
0.3125
128
```

In [15]:
```
print(n*phat)
print(n*(1-phat))
```

```
40.0
88.0
```

c) Calculate the 75% confidence interval that estimates the proportion of colleges and universities that offer dorms as housing.

The formula for the 75% confidence interval is: $\hat{p} \pm z^* \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$

In [16]:
```
z_star = norm.ppf(.875)
print(z_star)
lowerbound = phat - z_star * (phat * (1 - phat) / n) ** (1/2)
upperbound = phat + z_star * (phat * (1 - phat) / n) ** (1/2)
print("The 75% confidence interval is (", lowerbound, ", ", upperbound, ").")
```

```
1.1503493803760079
The 75% confidence interval is ( 0.26537122622556997 ,  0.35962877377443003 ).
```

d) Interpret your confidence interval. That is, put your interval into words.

I am 75% confident that the true population proportion of all colleges & universities that offer dorms as a housing option for students is contained in the interval (0.2654, 0.3596).

e) Based on your confidence interval, respond to the following statement.

Is it reasonable to say that the proportion of all colleges and universities that offer dorms as housing options for students is different from one third (1/3)?

Be sure to explain your answer, and write out the corresponding hypotheses that we could be testing.

The value of 0.33, representing 1/3, is contained in the interval, so it's a reasonable value for our parameter of interest.

For the hypotheses described here:

$H_0 : p = 0.33$ vs. $H_a : p \neq 0.33$, I would fail to reject the null hypothesis at a signficance level of $\alpha = 0.25$.

f) Suppose that we gather a second random sample of 128 colleges, and record the proportion of colleges and univerisities that offer dorms as housing for this sample. We generate a second confidence interval for our proportion from this sample, and find that we have a different confidence interval calculated than from part **c** above. Based on the fact that our confidence interval calculation is different, did we do something wrong during the confidence interval generation? Explain.

We did not necessarily do something incorrect. We have a sample of the same size (128). Our z* multiplier should be the same. However, since we have a different sample, it would be reasonable to have a different sample mean and sample standard deviation that is used in the confidence interval equation:

$$\hat{p} \pm z^* \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

I would be surprised to have extremely different statistics across the different samples, but I would also be suprised to have the exact same statistics from two different samples.

g) Now suppose that we gather 200 random samples of 128 colleges, and record the proportion of colleges and universities that offer dorms as housing for each sample. We use our sample data for each sample to generate a new 75% confidence interval. How many of the 200 random samples do you expect to miss the true proportion of colleges and universities that offer dorms as housing for each sample?

In [17]:
```
200*.25
```

Out[17]:
```
50.0
```

The confidence level interpretation explains that I would expect 75% of the 200 confidence intervals to contain/cover the true proportion.

That means that I would expect 25% of the intervals to miss the parameter, so 25% of 200, or 50 intervals.

5. Average enrollment? [13 points]

The University of Illinois is a large university, with a total enrollment of approximately 44,000. There are many smaller colleges in the state, including Parkland College with a total enrollment of 9,715.

We know that there are many more smaller colleges than large universities, so we'd like to test if the average college enrollment is smaller than the enrollment of Parkland College.

We'd like to perform a hypothesis test using a 10% significance level based on our sample of colleges and universities.

a) Write out your hypotheses. Be sure to use appropriate notation and to define the parameter of interest.

$H_0 : \mu = 9715$ vs. $H_a : \mu < 9715$

where μ is the population mean enrollment for the population of all colleges and universities.

b) While we could follow the standard hypothesis testing procedure, we will instead make use of our simulation procedures that we have developed so far this semester.

While we only have our one sample available, we will use it as a stand in for the population. From our sample, gather a random sample with replacement of the same size as our original sample data. We will

then repeat this process to generate many random samples. For each of our random samples, calculate and record our statistic of interest. We will repeat this process 5000 times, and use the 5000 observations of our statistic of interest as a sampling distribution.

Since we are using sample data to generate this sampling distribution, we can refer to our estimated sampling distribution as a **bootstrapped** distribution.

In [18]:
```
bootstrap = []
for i in range(5000):
    bootstrap.append((df['TOT_ENROLL'].sample(128, replace = True)).mean())

bootstrap = pd.DataFrame({'xbar': bootstrap})
```

In [19]:
```
bootstrap.head()
```

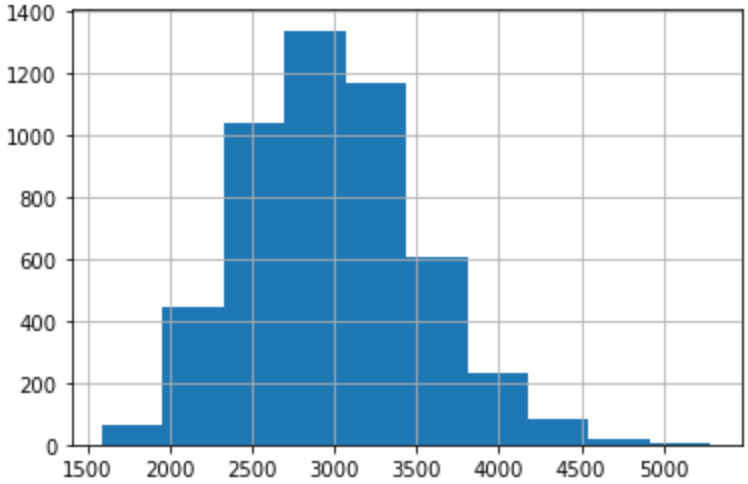
Out[19]:

	xbar
0	2498.257812
1	2090.476562
2	2421.460938
3	3296.453125
4	3545.750000

c) Generate a histogram of the **bootstrapped** distribution.

In [20]:
```
bootstrap['xbar'].hist()
```

Out[20]:
```
<AxesSubplot:>
```



d) Find the middle 80% of the bootstrapped distribution.

We can use this as an approximation for an 80% confidence interval.

Hint: Recall that we previously found percentiles for a set of data in Case Study 3 (about Quantitative Variables) when we were finding the IQR. We can make adjustments to the arguments for the code to help us find the middle 80% of our bootstrapped distribution.

In [21]:
```
lower = bootstrap['xbar'].quantile(0.1)
upper = bootstrap['xbar'].quantile(0.9)
```

```
print("The estimated 80% confidence interval based on the bootstrapping procedure is (", l
```

```
The estimated 80% confidence interval based on the bootstrapping procedure is ( 2325.44140
625 ,  3672.48671875 ).
```

The middle 80% corresponds to 20% in the tails and 10% in each tail. This means that we are looking for the 10th and 90th percentiles, which we can find from our simulated bootstrapped distribution.

e) Based on the interval from part **d**, assess the theories in the hypotheses in part **a**. Which is more reasonable? Explain.

Based on the interval from part **d**, 9,715 is not a reasonable value for the total enrollment of a college. In fact, the entire confidence interval is less than that value, which corresponds to the alternative hypothesis. Therefore, I would reject my null hypothesis and say that my alternative is more reasonable.

How does this compare to what I would have gotten if I had used a traditional confidence interval? Let's check it out:

In [22]:

```
xbar = df["TOT_ENROLL"].mean()
s = df["TOT_ENROLL"].std()
mu_0 = 9715
n = df["TOT_ENROLL"].shape[0]
t_star = t.ppf(0.9, n-1)
lower = xbar - t_star * s / n ** (1/2)
upper = xbar + t_star * s / n ** (1/2)
print("80% confidence interval is (", lower, ", ", upper, ").")
```

```
80% confidence interval is ( 2298.443037754975 ,  3651.728837245025 ).
```

These are pretty similar results, although my bootstrapped confidence interval is a little narrower. The bootstrapped confidence interval results will differ based on your exact simulations.

Remember to keep all your cells and hit the save icon above periodically to checkpoint (save) your results on your local computer. Once you are satisified with your results restart the kernel and run all (Kernel -> Restart & Run All). **Make sure nothing has changed**. Checkpoint and exit (File -> Save and Checkpoint + File -> Close and Halt). Follow the instructions on the Homework 4 Canvas Assignment to submit your notebook to GitHub.