STAT 207 Homework 5 [50 points] - Solutions

Probability, Sampling Distributions, and Random Variables

Due: Friday, February 17 by noon (11:59 am) CST

Package Imports

We'll import three Python packages that we've used so far and will need for this assignment. Those packages are pandas, matplotlib.pyplot, and seaborn.

Run the cell provided below to access the functions in these packages.

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
```

Case Study: County Housing Information

The provided county.csv file contains various statistics about each county in the United States. You can read more about this dataset and its variables here:

https://www.openintro.org/data/index.php?data=county

Observational Unit: Each row contains data recorded for a county.

Population: The population of all counties in the United States is included in the county.csv data set.

Variable: We will focus on the multi_unit variable in the data, which reports the percent of housing units in each county that are in multi-unit structures (e.g. apartments).

1. Champaign County [9 points]

a) First, read in the county.csv file.

Note: The csv file does have some missing values. Our variable of interest, multi_unit, does not have any missing values. For this assignment and because our variable of interest is complete, we do not need to clean the data or drop any rows with missing values.

In [2]:
```
df = pd.read_csv('county.csv')
df.shape
```

Out[2]:
```
(3142, 15)
```

In [3]:
```
df.head()
```

Out[3]:

	name	state	pop2000	pop2010	pop2017	pop_change	poverty	homeownership	multi_unit	unem
0	Autauga County	Alabama	43671	54571	55504	1.48	13.7	77.5	7.2	
1	Baldwin County	Alabama	140415	182265	212628	9.19	11.8	76.7	22.6	
2	Barbour County	Alabama	29038	27457	25270	-6.22	27.2	68.0	11.1	
3	Bibb County	Alabama	20826	22915	22668	0.73	15.2	82.9	6.6	
4	Blount County	Alabama	51024	57322	58013	0.68	15.6	82.0	3.7	

Sample 1: We will start by focusing on our county (Champaign County in Illinois). Because our county is home to UIUC, in which a lot of students and young adults live in apartments, we believe that our county will have a high percent of housing units in multi-unit structures.

b) Display just the row for Champaign County, Illinois.

Tip: We can use the **&** ("and") operator to indicate that we want **both** conditions on either side of the operator to be met. We can use the **|** ("or") operator to indicate that we want **at least one** of the conditions to be met. We can also chain these operators together if we need to represent more complex operations.

In [4]:
```
df[(df['name'] == 'Champaign County') & (df['state'] == 'Illinois')]
```

Out[4]:

	name	state	pop2000	pop2010	pop2017	pop_change	poverty	homeownership	multi_unit	un
604	Champaign County	Illinois	179669	201081	209399	1.83	21.4	55.9	35.2	

c) Calculate the probability that a randomly selected county in the United States would have a multi-unit rate that is at least as high as Champaign county's rate.

In [5]:
```
sum(df['multi_unit'] >= 35.2)/df.shape[0]
```

Out[5]:
```
0.02864417568427753
```

In [6]:
```
# OR
champaign_rate = df[(df['name'] == 'Champaign County') &
                    (df['state'] == 'Illinois')]['multi_unit'].item()
print(champaign_rate)
(df['multi_unit'] >= champaign_rate).mean()
```

Out[6]:
```
35.2
0.02864417568427753
```

d) Calculate the probability that a randomly selected county in Illinois would have a multi-unit rate that is at least as high as Champaign county's rate.

In [7]:
```
il = df[df['state'] == 'Illinois']
(il['multi_unit'] >= champaign_rate).mean()
```

Out[7]: 0.0196078431372549

e) Are the events "has a multi-unit rate at least as high as Champaign County, Illinois" and "county is in the state of Illinois" independent for the counties in the United States? Explain.

No, these events are not independent. If events are independent, then $P(A) = P(A|B)$. In part **c**, I calculated P("has a multi-unit rate at least as high as Champaign County, Illinois") as 2.86%. In part **d**, I calculated P("has a multi-unit rate at least as high as Champaign County, Illinois" | "county is in the state of Illinois") as 1.96%. These two probabilities are not equivalent, which indicates that these two events are not independent.

2. Describing Champaign + Neighboring Counties [8 points]

Sample 2: What if we extend from our county to include four neighboring counties (Vermilion, Ford, Piatt, and Douglas)? will the average multi-unit rate for these five counties be higher than the average rate for five randomly selected counties from the population of U.S. counties?

a) To start, create a data frame that is composed of just the five following Illinois counties:

- Champaign
- Vermilion
- Ford
- Piatt
- Douglas

Print the resulting data frame.

In [8]:
```
df5 = df[((df['name'] == 'Champaign County') | (df['name'] == 'Vermilion County') |
          (df['name'] == 'Ford County') | (df['name'] == 'Piatt County') |
          (df['name'] == 'Douglas County'))  & (df['state'] == 'Illinois')]
df5
```

Out[8]:

	name	state	pop2000	pop2010	pop2017	pop_change	poverty	homeownership	multi_unit	un
604	Champaign County	Illinois	179669	201081	209399	1.83	21.4	55.9	35.2	
615	Douglas County	Illinois	19922	19980	19748	-0.57	12.1	78.3	11.2	
621	Ford County	Illinois	14241	14081	13280	-2.86	15.7	79.1	7.9	
668	Piatt County	Illinois	16365	16729	16445	0.09	5.4	81.7	7.5	
686	Vermilion County	Illinois	83919	81625	77909	-3.31	19.8	71.3	14.5	

b) Calculate the mean multi-unit rate for these five counties.

In [9]:
```
df5['multi_unit'].mean()
```

Out[9]: 15.260000000000002

c) Calculate the mean and standard deviation for the multi-unit rate for the population of counties in the United States.

In [10]:
```
df['multi_unit'].mean()
```

Out[10]:
```
12.321896880967572
```

In [11]:
```
df['multi_unit'].std()
```

Out[11]:
```
9.290204854486714
```

d) Generate a random sample with replacement of size $n = 5$ counties from the United States, and calculate the mean multi-unit rate for these 5 counties. What symbol should you use to describe the value you just calculated? Compare your calculated value to the values previously calculated in **2b** and **2c**.

In [12]:
```
onesample = df.sample(5, replace = True, random_state = 2102023)
onesample
```

Out[12]:

	name	state	pop2000	pop2010	pop2017	pop_change	poverty	homeownership	multi_unit
503	Putnam County	Georgia	18812	21218	21730	1.9	16.2	79.9	3.8
914	Ellsworth County	Kansas	6525	6497	6330	-0.52	11.4	78.4	6.5
788	Whitley County	Indiana	30707	33292	33756	1.43	9.5	83.2	9.8
1531	Jasper County	Missouri	104686	117404	120217	2.96	17.3	65.7	15.5
2197	Seminole County	Oklahoma	24894	25482	24878	-2.21	22.7	72.4	7.8

In [13]:
```
onesample['multi_unit'].mean()
```

Out[13]:
```
8.68
```

The symbol for the value I just calculated is $\bar{x}$, or a sample mean, because it's a mean calculated from a random sample of size 5.

I found a mean multi-unit rate of 8.68, which is much lower than the multi-unit rate for the population (12.3) and the rate for the 5 counties of interest in Illinois (15.26). Is it meaningful that it is so much lower? It's hard to tell (yet).

3. Comparing Champaign + Neighboring Counties [10 points]

a) The following MCmeans function has been provided for you. Run the code in the first cell to load this function into Python.

Then, using the MCmeans function, geneerate output according to the following specifications:

- Specify the argument (input) for df to reference your data frame of counties
- Specify the argument for var to reference the variable name for multi-unit housing rates in the counties data frame
- Specify the argument to generate samples of size 5
- Specify the argument to generate 5000 repeated samples

- Save the output of this function to a Python object

In [14]:
```
def MCmeans(df, x='', replace=True, n=1, M=1):
    #INPUT:
    # df is a data frame
    # x is a text-valued name for a variable in the data frame
    # replace = True or False depending on whether
    #    draws are with or without replacement
    # n = number of draws per sample
    # M = number of samples to draw
    MCstats = []
    for i in range(M):
        #1. Collect a random sample of size n with replacement
        #2. Take the mean of this random sample
        #3. Append this random sample mean to the SampleMeans list
        # (which is our SAMPLING DISTRIBUTION OF SAMPLE MEANS!)
        MCstats.append(df[x].sample(n, replace=replace).mean())
    #4. returns the sampling distribution in a dataframe format
    return pd.DataFrame({x: MCstats})
```

In [15]:
```
county_sampdist = MCmeans(df, 'multi_unit', n = 5, M = 5000)
county_sampdist.shape
```

Out[15]:
```
(5000, 1)
```

b) Calculate the mean and standard deviation for the object that is created in part **3a**. Compare these values to those from **2c**.

In [16]:
```
county_sampdist.mean().item()
```

Out[16]:
```
12.348924000000002
```

In [17]:
```
county_sampdist.std().item()
```

Out[17]:
```
4.177938414589582
```

The mean from the estimated sampling distribution is 12.39, which is very close to the population mean of 12.32. This is especially evident when comparing to the other sample means calculated throughout Question **2**.

The standard deviation is 4.2, which is smaller (less than half) the standard deviation of 9.29 calculated for the population in Question **2c**.

c) What proportion of samples of size 5 have a mean multi-unit rate at least as large as the mean multi-unit rate of Champaign and four of its neighboring counties (from **2b**)?

In [18]:
```
(county_sampdist >= 15.26).mean()
```

Out[18]:
```
multi_unit    0.2178
dtype: float64
```

In [19]:
```
# OR
(county_sampdist >= df5['multi_unit'].mean()).mean()
```

Out[19]:
```
multi_unit    0.2166
dtype: float64
```

Note: The above averages are slightly different due to rounding differences in the mean calculation from `df5`.

d) Which is more unusual of the multi-unit rate of Champaign county or the average multi-unit rate of Champaign county and its four neighboring counties. Explain, including numerical support in your justification.

We saw that for the population of all counties, only 2.86% of the counties have a multi-unit rate of Champaign county, Illinois or higher (Question **1c**). Above, we see that approximately 21.5% of random samples of 5 counties have a multi-unit rate of Champaign county and its for neighboring counties or higher (Question **3d**). This indicates that Champaign county itself is more unusual, since it has a fewer proportion of counties that are as or more unusual than it is.

Case Study: Valentine's Day Spending

4. How much do college students spend on Valentine's Day? [13 points]

A resident advisor aims to answer this question, so he surveys all residents who live in his dorm at the fictitious College University. He saves this information a data file.

He has a few friends who claim that they do not spend very much on Valentine's Day. In fact, the average spending of these friends is only $6.

The resident advisor, always a skeptic, wants to assess the claim of his friends: do they actually spend a small amount on Valentine's Day compared to his residents? He knows that the $6 average of his friends shouldn't be compared to the population distribution for all of his residents, so he simulates a sampling distribution for the average spending of his residents on Valentine's Day. He saves the results of this simulation to another data file.

a) Read in the two data files: vday1.csv and vday2.csv. Report the number of observations for each file, and preview the first few rows of each file.

In [20]:
```
vday1 = pd.read_csv('vday1.csv')
vday1.shape
```

Out[20]:
```
(214, 2)
```

In [21]:
```
vday1.head()
```

Out[21]:

	Unnamed: 0	spending
0	0	2.95
1	1	8.14
2	2	5.36
3	3	2.12
4	4	1.79

In [22]:
```
vday2 = pd.read_csv('vday2.csv')
vday2.shape
```

Out[22]:
```
(214, 2)
```

In [23]:
```
vday2.head()
```

Out[23]:

	Unnamed: 0	spending
0	0	4.17
1	1	8.05
2	2	0.00
3	3	0.00
4	4	16.15

b) Unfortunately, the resident advisor realizes that he did not use the best naming system for his two files. He forgets which file contains the population data and which file contains the simulated sampling distribution. Help the resident advisor out by determining which data file contains which distribution. Explain to him how you know that you are correct.

In [24]:
```
vday1['spending'].mean()
```

Out[24]:
```
5.015233644859816
```

In [25]:
```
vday1['spending'].std()
```

Out[25]:
```
2.4546473488315224
```

In [26]:
```
vday2['spending'].mean()
```

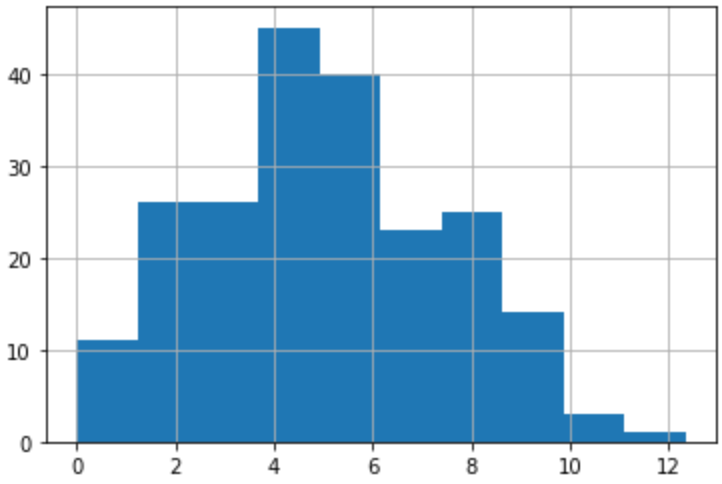
Out[26]:
```
5.028551401869158
```

In [27]:
```
vday2['spending'].std()
```

Out[27]:
```
5.050620494085818
```

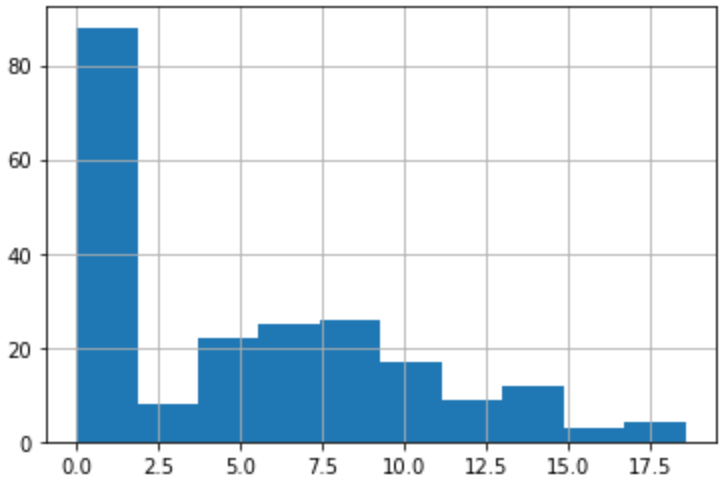
In [28]:
```
vday1['spending'].hist()
```

Out[28]:
```
<AxesSubplot:>
```



In [29]:
```
vday2['spending'].hist()
```

Out[29]:
```
<AxesSubplot:>
```



Based on the above analysis, I see that vday2 contains the population distribution and vday1 contains the simulated sampling distribution. I can identify these differences by:

- the shape of the histograms (vday1 has a histogram that appears more symmetric and bell-shaped)
- the range of the histograms (vday2 has a larger maximum)
- the standard deviations (vday1 has a smaller standard deviation)

c) Calculate the parameter values for the mean and standard deviation.

In [30]:
```
vday2['spending'].mean()
```

Out[30]:
```
5.028551401869158
```

In [31]:
```
vday2['spending'].std()
```

Out[31]:
```
5.050620494085818
```

Note: The **parameter** values refers to the **population** values, so I calculate the values from my population distribution.

d) The resident advisor has once again been a little careless with his record keeping. He forgot to record how many friends he surveyed in order to calculate the $6 average spent on Valentine's Day. Based on the available information, help him identify how many friends he surveyed.

In [32]:
```
(vday2['spending'].std() / vday1['spending'].std()) ** 2
```

Out[32]:
```
4.233614001002797
```

We know that the standard deviation for a sampling distribution should equal $\frac{\sigma}{\sqrt{n}}$. Using that information combined with our earlier calculations, we can estimate n according to $(\frac{\sigma}{\sigma_{\bar{x}}})^2$

From this equation, we would estimate a sample of size 4.23, which we can round to 4.

e) Finally, are his friends especially stingy in terms of Valentine's Day spending, relative to the residents in his dorm? Explain.

In [33]:
```
(vday1['spending'] <= 6).mean()
```

Out[33]:
```
0.6728971962616822
```

If we were to take repeated random samples of the same size of students from the dorm in question, we would find that roughly 67% of these samples (~2/3) actually spent less on average than this group of 4 friends. Therefore, I would not say that this group is especially stingy.

However, you could make an argument that the group of friends might be substantially different from the residents in this dorm that makes this comparison incompatible. For example, if there are differences related to how many friends are in relationships, or the general spending of residents who live in dorms compared to these friends, then you might say that this calculation is not an accurate representation of how much (or little) his friends are spending.

Case Study: Super Bowl

5. Random Variables [10 points]

Suppose that the number of touchdowns scored by a team in the Super Bowl follows a Poisson random variable with parameter $\mu = 2.4$ (μ is also sometimes called λ).

Information about Poisson random variables is contained within the scipy.stats package. The code to import this is included below.

In [34]:
```
from scipy.stats import poisson
```

a) One important aspect of working with a programming language is the ability to get information about the packages.

To start, find the documentation for this package. You may consider performing a web search where you include the package name (scipy.stats) and the specific distribution name.

Provide the url to the documentation that you find.

From reading through this documentation, is a Poisson random variable discrete or continuous?

One url you might find is: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.poisson.html

A Poisson random variable is discrete, as one of the first lines on this page says "A Poisson discrete random variable.". You might also notice that the page refers to a probability mass function, which is calculated for discrete random variables.

b) Continuing to look through the documentation, what are the first and second methods listed in the table of available methods?

The first two methods are `rvs` and `pmf`.

c) As of this homework's writing, you can bet on whether you think the Chiefs will score more than 2.5 touchdowns, with the bets favoring the Chiefs will score less than this number of touchdowns.

If the number of touchdowns for the Chiefs follows a Poisson distribution with parameter $\mu = 2.4$, what is the probability that the Chiefs will score 2 touchdowns or less?

In [35]:
```
poisson.pmf(0, 2.4) + poisson.pmf(1, 2.4) + poisson.pmf(2, 2.4)
```

Out[35]:
```
0.5697087466575105
```

In [36]:
```
poisson.cdf(2, 2.4)
```

Out[36]:
```
0.5697087466575106
```

To calculate the probability that the Chiefs will score 2 touchdowns or less, we would need to consider the probability that the Chiefs score 0 touchdowns, 1 touchdown, or 2 touchdowns. We can calculate this directly using three `pmf` calculations, or we can take a shortcut and use the `cdf` (cumulative density function), which already calculates $P(X \leq x)$.

d) Using the underlying Poisson distribution described above, randomly simulate the number of touchdowns scored for each team (two teams play). You may use random states if you'd like, but you don't need to.

In [37]:
```
poisson.rvs(2.4, size = 2, random_state = 2122023)
```

Out[37]:
```
array([0, 5])
```

e) You can also bet on whether you think the total score of the game will be above or below 51 points.

We'll make a simplifying assumption that each touchdown will be worth 7 points.

Based on the simulation in part **d**, how many touchdowns would be scored in your simulated game? How many points do those touchdowns correspond to? Based only on the number of touchdowns scored, would the score of the game be above or below 51 points?

In [38]:
```
poisson.rvs(2.4, size = 2, random_state = 2122023).sum() * 7
```

Out[38]:
```
35
```

Above we randomly simulated that one team would score 0 touchdowns and the other team would score 5 touchdowns. Therefore, we would estimate a total score (from touchdowns only) of 35 points. Therefore, the score of the game would be below 51 points (from this simulation only).

Note: Your results will likely vary, as you'll have a different simulation.

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.