DS4E HOMEWORK 3

January 25, 2023

1 DS4E: Homework 3

[2]:
```
# import libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.formula.api as smf
```

1.1 Question 1

1(a)

The dependent variable is "support for canceling student debt"

1(b)

There is no independent variable in this study; this is an observational study, and observational studies don't always need independent variables because there is no experinmental manipulation of them involved. One could make a case that WSP is independent, but it really has nothing to do with what the researcher is trying to measure/the relationship of it.

1(c)

The researcher is conceptualizing "support" for canceling student debt based on people's support of Biden's Student Debt Relief Plan. In order to conceptualize "support" for the Relief Plan, the researcher created a scale that measures support on a scale of 1 to 5, 1 being "strongly oppose the plan," or least support, and 5 being "strongly support the plan," or most support. In other words, level of support is quanitified on a scale of 1 to 5, 1 being no support (oppose), and 5 being strong support.

1(d)

The researcher operationalizes "support" by standing in Washington Square Park on a Saturday afternoon and asking random passerbys to indicate their level of support of the Student Debt Relief Plan, on a scale of 1-5.

1(e)

One strength of this measure of support is that the researcher quantifies support in a practical, straightforward way. Since every single participant is given the same scale with the same concepts of 1 ("strongly oppose the plan") and 5 ("strongly support the plan"), it allows for the researcher to not only efficiently understand but better compare the opinions of the participants in order to draw conclusions from the data.

1(f)

One weakness of this measure of support is that it's simplicity fails to capture other factors that might go into measuring "support". First of all, this measure of support only captures support for Biden's Relief Plan, not canceling sudent debt as a whole, which is a flaw in itself. People could support canceling student debt, but not Biden's plan, and vise versa. Furthermore, it is too black and white; this measure of support only really asks the participant if they support the plan or not… there are different aspects of the plan and of canceling student debt that are not mentioned, which may lead to a misleading representation of who actually supports the plan or not; ex. some individuals may support some parts of the plan, but not others, but they are forced to chose a number for support on a small scale, which likely contributes to the omission of important information from the study.

1(g)

One possible source of random error is if the researcher accidentally misheard and recorded the wrong answer of an individual. For example, the researcher could have misheard the participant's number (especially in a loud park), thinking they said a 1 instead of 2 for their rating, therefore recording the wrong answer into their data. This is a very minor error, though, and will likely cancel out over time.

1(h)

We call this type of bias response bias, where the participants aren't being honest, in a sometimes predictable way. Response bias is likely to occur when a participant has reason to conceal their true answer/opinion and instead lie to the researcher; oftentimes this happens when an individual is embarrased to give their true answer or may gain some social benefit from saying another. In this case, an individual may be too embarrased to say that they don't support Biden's Student Relief Plan, because it is a controversial topic, and many have strong opinions. Being in Washington Square Park, a very liberal area, it would be much more socially acceptable to say you are in support of the plan, and also because in general it seems morally right to support less debt for young adults becoming educated. This would skew the results in favor of Biden's Debt Relief Plan (aka left skew), because people wouldn't want to admit if they weren't in support of it due to embarrasment or social repercussions. In addition to that, the researcher is a student and is asking specifically about support for canceling student loan debt, therefore that may influence the participant to answer in favor of it.

1(i)

Sources of selection bias: 1. This is a volunteer based study, therefore it is not random; just anyone can come up and give their opinion. This will likely lead to those who are more enthusiastic about the question (ex. support the question being asked), or more bold/confident coming up to an interviewer. This may lead to a sampling issue and skew the results, likely to more extremes.

2. This study is being performed at a specific location, Washington Square Park, on a specific day, Saturday. Therefore, it is not random, and instead samples people that may have something in common. For example, those who go to Washington Square Park are likely students, and are likely more progressive, therefore leading to a sampling issue and skewing the results.

1(j)

This critic is identifying the possibility of an error of validity; measuring support for Biden's Student Debt Relief Plan may not be an accurate measurement or reflection of support for canceling student debt in general (in a broad sense); the study may not actually be measuring what it claims to be measuring.

1.2 Question 2

2(a)

[24]:
```
athlete_salary = pd.read_csv('forbes_athletes.csv')
athlete_salary.head(15)
```

[24]:
```
                 Name Nationality  Current Rank        Sport  Year  \
0          Mike Tyson         USA             1       boxing  1990
1      Buster Douglas         USA             2       boxing  1990
2   Sugar Ray Leonard         USA             3       boxing  1990
3        Ayrton Senna      Brazil             4  auto racing  1990
4         Alain Prost      France             5  auto racing  1990
5       Jack Nicklaus         USA             6         golf  1990
6         Greg Norman   Australia             7         golf  1990
7      Michael Jordan         USA             8   basketball  1990
8       Arnold Palmer         USA             8         golf  1990
9   Evander Holyfield         USA             8       boxing  1990
10  Evander Holyfield         USA             1       boxing  1991
11         Mike Tyson         USA             2       boxing  1991
12     Michael Jordan         USA             3   basketball  1991
13     George Foreman         USA             4       boxing  1991
14       Ayrton Senna      Brazil             5  auto racing  1991

    earnings ($ million)
0                   28.6
1                   26.0
2                   13.0
3                   10.0
4                    9.0
5                    8.6
6                    8.5
7                    8.1
8                    8.1
9                    8.1
10                  60.5
11                  31.5
12                  16.0
13                  14.5
14                  13.0
```

2(b)

The unit of analysis is the athlete

2(c)

[25]:
```
athlete_salary_renamed = athlete_salary.rename(columns={'Name': 'name',␣
↪'Nationality': 'nationality',
                              'Current Rank': 'current_rank', 'Sport': 'sport',
                              'Year': 'year', 'earnings ($ million)':␣
↪'earnings'})

athlete_salary_renamed.head()
```

[25]:

	name	nationality	current_rank	sport	year	earnings
0	Mike Tyson	USA	1	boxing	1990	28.6
1	Buster Douglas	USA	2	boxing	1990	26.0
2	Sugar Ray Leonard	USA	3	boxing	1990	13.0
3	Ayrton Senna	Brazil	4	auto racing	1990	10.0
4	Alain Prost	France	5	auto racing	1990	9.0

2(d)

[26]:
```
athlete_salary_renamed['sport'].replace(['NFL'], 'American Football',␣
↪inplace=True)

athlete_salary_renamed.head()
```

[26]:

	name	nationality	current_rank	sport	year	earnings
0	Mike Tyson	USA	1	boxing	1990	28.6
1	Buster Douglas	USA	2	boxing	1990	26.0
2	Sugar Ray Leonard	USA	3	boxing	1990	13.0
3	Ayrton Senna	Brazil	4	auto racing	1990	10.0
4	Alain Prost	France	5	auto racing	1990	9.0

2(e)

[36]:
```
athlete_salary_renamed['year'].value_counts()
```

[36]:
```
2002    11
2020    10
2019    10
1991    10
1992    10
1993    10
1994    10
1995    10
1996    10
1997    10
1998    10
```

```
1999    10
2000    10
2003    10
2004    10
2005    10
2006    10
2007    10
2008    10
2009    10
2010    10
2011    10
2012    10
2013    10
2014    10
2015    10
2016    10
2017    10
2018    10
1990    10
Name: year, dtype: int64
```

The unusual value is that the year 2002 has 11 conuts while all the rest have 10.

2(f)

[38]:
```
athlete_salary_earnings = athlete_salary_renamed.sort_values(by='earnings',␣
↪ascending=False)
athlete_salary_reordered = athlete_salary_earnings[['name', 'year', 'earnings']]
athlete_salary_reordered.head()
```

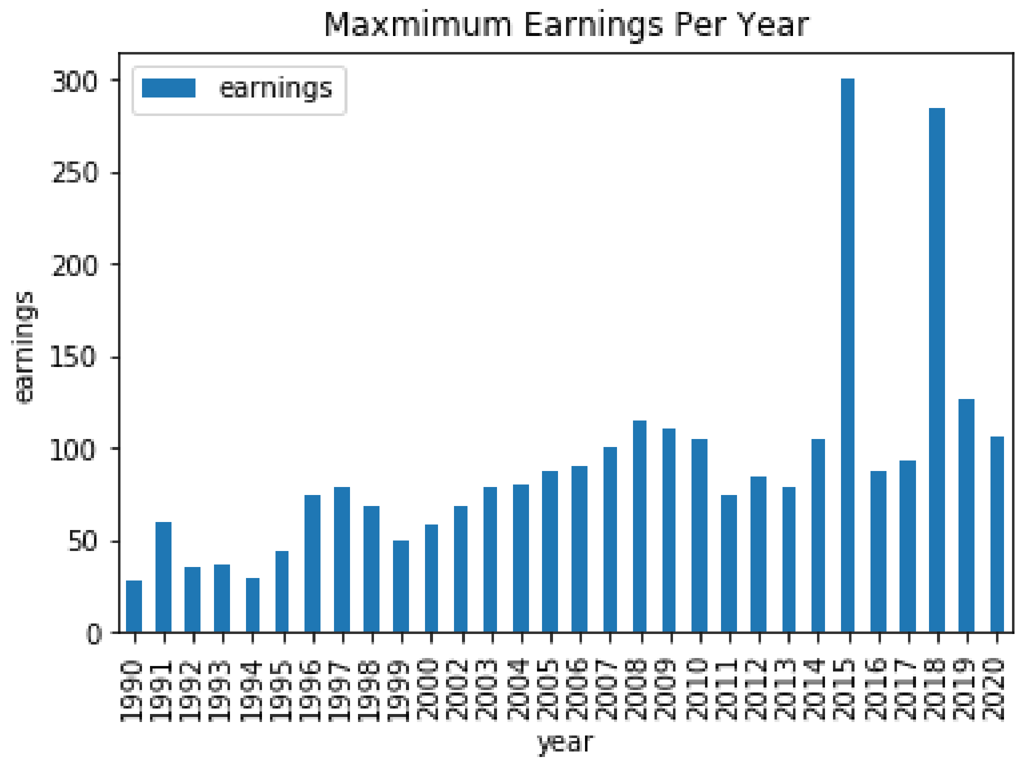
[38]:

	name	year	earnings
241	Floyd Mayweather	2015	300.0
271	Floyd Mayweather	2018	285.0
242	Manny Pacquiao	2015	160.0
281	Lionel Messi	2019	127.0
171	Tiger Woods	2008	115.0

2(g)

[40]:
```
athlete_max_earnings = athlete_salary_renamed[['year', 'earnings']]
athlete_max_earnings = athlete_max_earnings.groupby(['year']).max()
athlete_max_earnings_plot = athlete_max_earnings.plot.bar(title = 'Maxmimum␣
↪Earnings Per Year')
athlete_max_earnings_plot.set_ylabel('earnings')
athlete_max_earnings_plot
```

[40]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fcd187d6910>
```



This suggests that for prominent athletes, the maximum amount of earnings has been increasing steadily. There are two outliers, in 2015 and 2018, which also contribute to the trend that max earnings have been increasing more in recent years, as those values were over twice the size of every other value. In the 1990s, athletes with max earnings were only earning under $50 mil, while nowadays they are earning up to around $125 mil. This is basically showing how there is now a bigger opportunity in sports to make more as a prominent athlete.

One could speculate that this graph kind of displays a left skew, since the data is heavier towards the right, and dwindles out towards the left.

2(h)

[41]:
```
athlete_total_earnings = athlete_salary_renamed[['nationality', 'earnings']]
athlete_total_earnings = athlete_total_earnings.groupby(['nationality']).sum()
athlete_total_earnings = athlete_total_earnings.sort_values(by='earnings',␣
↪ascending=False)
athlete_total_earnings
```

[41]:
```
                    earnings
nationality
USA                   8786.3
Portugal               787.1
Switzerland            781.1
```

```
Argentina           715.5
Germany             639.0
UK                  443.2
Brazil              422.0
Philippines         242.0
Finland             129.0
Italy               128.0
Canada               99.1
Ireland              99.0
Mexico               94.0
Filipino             62.0
Serbia               55.8
Northern Ireland     50.0
Spain                44.5
France               36.0
Dominican            35.0
Russia               29.8
Austria              13.5
Australia             8.5
```

1.3 Question 3

3(a)

[8]:
```python
sample_salary = pd.read_csv('chicago_salary_sample.csv')
mean_sample_salary = sample_salary.mean()
print(mean_sample_salary)
```

```
annual_salary    99217.66344
dtype: float64
```

3(b)

[10]:
```python
population_salary = pd.read_csv('chicago_salary_full.csv')
mean_population_salary = population_salary.mean()
print(mean_population_salary)
```

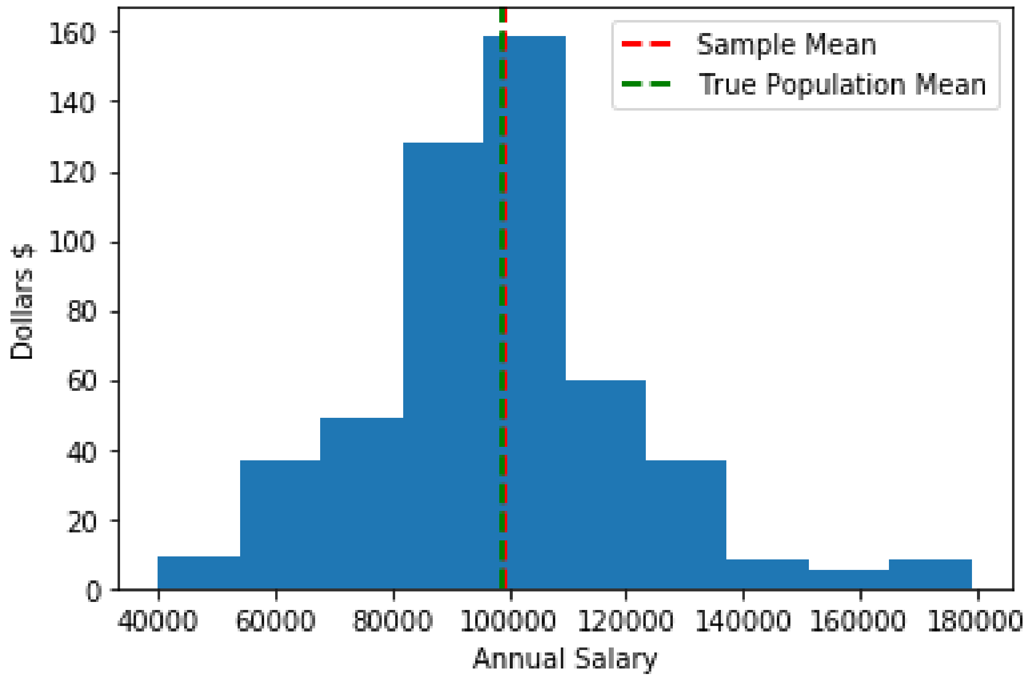
```
annual_salary    98915.825372
dtype: float64
```

3(c)

[11]:
```python
plt.hist(sample_salary['annual_salary'])
plt.xlabel("Annual Salary")
plt.ylabel("Dollars $")
plt.axvline(float(mean_sample_salary), color='red', lw = 2, ls = '--', label=
    'Sample Mean')
plt.axvline(float(mean_population_salary), color='green', lw = 2, ls = '--',
    label= 'True Population Mean')
```

```
plt.legend()
plt.show
```

[11]: <function matplotlib.pyplot.show(*args, **kw)>



3(d)

[12]:
```
len(sample_salary)

bootstrap_salary_500 = sample_salary.sample(n=500, replace=True)
print(np.mean(bootstrap_salary_500))
```

```
annual_salary    101712.21624
dtype: float64
```

3(e)

[15]:
```
sample_salary_means = np.array([])

for outcome in range(1000):
    sample_salary_means = np.append(sample_salary_means, (np.mean(sample_salary.
 ↪sample(n=500, replace=True))))

conf=95
lower_pct = (100-conf)/2
upper_pct = 100-((100-conf)/2)
```

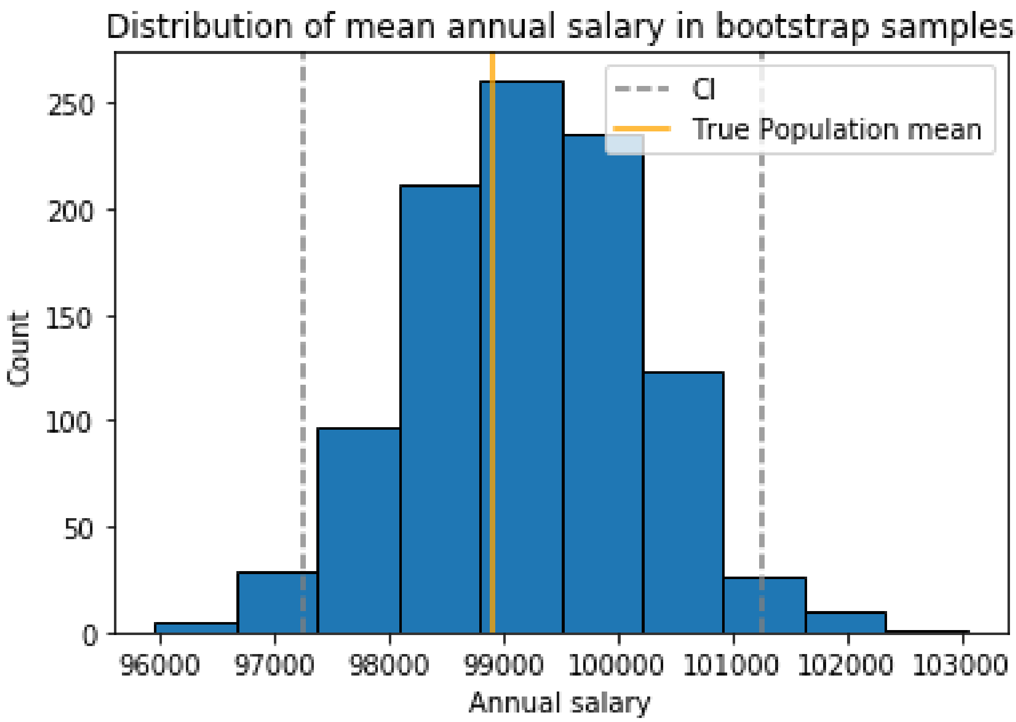
```
lower = np.percentile(sample_salary_means, lower_pct)
upper = np.percentile(sample_salary_means, upper_pct)

print("95% Confidence Interval:", lower, ",", upper)
```

```
95% Confidence Interval: 97151.46 , 100937.11799999999
```

Yes, this confidence interval does capture the true population mean of 98915.825372.

3(f)

[34]:
```
plt.hist(sample_salary_means, ec="black", bins=10)
plt.axvline(lower, color='grey', linestyle='--', label='CI')
plt.axvline(upper, color='grey', linestyle='--')
plt.axvline(float(mean_population_salary), color='orange', label='True␣
↪Population mean')
plt.title("Distribution of mean annual salary in bootstrap samples")
plt.xlabel("Annual salary")
plt.ylabel("Count")
plt.legend()
plt.show()
```



1.4 Question 4

4(a)

```
[3]: chicago_salaries = pd.read_csv('chicago_salary_full.csv')
cs_police_fire = chicago_salaries[chicago_salaries['department'].
 ↪isin(['POLICE', 'FIRE'])]
cs_police_fire.head()
```

```
[3]:                                job_titles department  annual_salary
0                                    SERGEANT     POLICE       122568.0
1  POLICE OFFICER (ASSIGNED AS DETECTIVE)     POLICE       110796.0
3                              POLICE OFFICER     POLICE        86730.0
4                           FIRE ENGINEER-EMT       FIRE       118830.0
5                              POLICE OFFICER     POLICE       109236.0
```

4(b)

```
[4]: print('Mean Annual Salary of Chicago Police = ' +␣
 ↪str(cs_police_fire[cs_police_fire['department'] == 'POLICE'].mean()))
print('Mean Annual Salary of Chicago Fire = ' +␣
 ↪str(cs_police_fire[cs_police_fire['department'] == 'FIRE'].mean()))
```

```
Mean Annual Salary of Chicago Police = annual_salary    101170.563985
dtype: float64
Mean Annual Salary of Chicago Fire = annual_salary    106580.967191
dtype: float64
```

4(c)

```
[17]: results = smf.ols('annual_salary ~ department-1', data=cs_police_fire).fit()
results.summary()
#remember the negative one in order to remove the intercept
```

```
[17]: <class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:          annual_salary   R-squared:                       0.014
Model:                            OLS   Adj. R-squared:                  0.014
Method:                 Least Squares   F-statistic:                     248.6
Date:                Thu, 10 Nov 2022   Prob (F-statistic):           1.29e-55
Time:                        19:36:29   Log-Likelihood:            -1.9215e+05
No. Observations:               16962   AIC:                         3.843e+05
Df Residuals:                   16960   BIC:                         3.843e+05
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
======
                         coef    std err          t      P>|t|      [0.025
0.975]
--------------------------------------------------------------------------------
```

```
------
department[FIRE]     1.066e+05    290.612    366.746      0.000     1.06e+05
1.07e+05
department[POLICE]   1.012e+05    182.439    554.546      0.000     1.01e+05
1.02e+05
==============================================================================
Omnibus:                     1268.984   Durbin-Watson:                   1.921
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             4084.504
Skew:                           0.366   Prob(JB):                         0.00
Kurtosis:                       5.290   Cond. No.                         1.59
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

According to this regression, the coefficient for the fire department is 1.066e+05, and the coefficient for the police department is 1.012e+05. The mean annual salary for the police department is 101170.6, while the mean annual salary for the fire departnment is 106580.9. This shows how the values of the coefficients and the values for the means are extremely similar, in appearance and what they're representing. For police department specifically, as the question asks, 101170.6 and 1.012e+05 are extremely close in feature, showing the relationship between police department and annual salary. It makes sense because regression shows the average for an increase in one unit, while the mean is the average.

[16]:
```
#only added for explanation of why i didn't do this
results = smf.ols('annual_salary ~ department', data=cs_police_fire).fit()
results.summary()
```

[16]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:          annual_salary   R-squared:                       0.014
Model:                            OLS   Adj. R-squared:                  0.014
Method:                 Least Squares   F-statistic:                     248.6
Date:                Thu, 10 Nov 2022   Prob (F-statistic):           1.29e-55
Time:                        19:36:03   Log-Likelihood:            -1.9215e+05
No. Observations:               16962   AIC:                         3.843e+05
Df Residuals:                   16960   BIC:                         3.843e+05
Df Model:                           1
Covariance Type:            nonrobust
================================================================================
========
                           coef    std err          t      P>|t|      [0.025
0.975]
```

```
--------------------------------------------------------------------------------
--------
Intercept             1.066e+05    290.612    366.746      0.000    1.06e+05
1.07e+05
department[T.POLICE] -5410.4032    343.132    -15.768      0.000   -6082.977
-4737.829
==============================================================================
Omnibus:                     1268.984   Durbin-Watson:                   1.921
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             4084.504
Skew:                           0.366   Prob(JB):                         0.00
Kurtosis:                       5.290   Cond. No.                         3.53
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
"""
```

Running the regression without the -1 only returns police department, and doesn't give an accurate coefficent that matches the means. The intercept matches the fire departnment