ds4e_hw3

November 10, 2022

1 DS4E: Homework 3

[20]:

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import statsmodels.formula.api as smf
```

Q1(a): Support for cancelling student debt

Q1(b): There is no independent variable

Q1(c): The researcher is conceptualizing the research by standing in the wsp and asking passerby of how supportive they are to student debt relief plan.

Q1(d): The researcher is operationalising the support by giving numbers to their support from 1 to 5.

Q1(e): One of the strenghths of the support is that by assigning values, the graduate student would be easily able to convert the data collected onto something that he can use to make predictions and make inferences about the support.

Q1(f): There is high chances, that people can make mistakes and assign a wrong number when they convert their support towards this program into numeric form. It very difficult to convert a support into numeric form, which can bias the study. Furthermore, the researcher just asks numbers and there is no place to further elaborate why a value was assigned.

Q1(g): One source of random error, is if either the researcher makes a mistake when recording data or mishears someone during the interview.

Q1(h): Response bias, as people are more likely to show support to the program due to social desirability as people usually want the student loans to be cancelled, so more people would result in favoring cancelling which would make the bias the values in more people supporting cancellation of student debts.

Q1(i): Only people who volunteer would be asking the questions so would cause selection bias. furthermore, only people at washington park are being interviewed, which might represent students from NYU only and this does not represents all people in US and would be a selection bias.

Q1(j):erros of validity, as we might not be measuring what we think we actually are, with in this case the critic says that we might be measuring support for a specific policy and not student loans.

```
[21]: ##Q2(A)
      data_set = pd.read_csv("forbes_athletes.csv")
      data_set.head(15)
```

```
[21]:                   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
```

Q2(B) Highest paid atheltes.

```
[22]: #Q2(c)
      data_set=((data_set.rename(columns={data_set.columns[5]: 'earnings',data_set.
       ↪columns[2]:'Current_Rank'})).rename(columns=str.lower))
      data_set.head(5)
```

```
## Citation for the code. I used https://stackoverflow.com/questions/11346283/
 ↪renaming-column-names-in-pandas this website to remember and confirm how a␣
 ↪column is renamed.
```

[22]:

```
                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
```

[23]:

```
#Q2(D)
data_set['sport'].replace(['NFL'],'American Football',inplace=True)

#Citations: Used the week6 jupyter notebook in the recitation slides, to␣
 ↪understand inplace=True and False.
```

[24]:

```
#Q2(e)
data_set.year.value_counts()
##the year 2002 had the 11 atheltes in the data while other years had 10␣
 ↪athelets for each year.  There are no values for the year 2001.
```

[24]:

```
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
```

[25]:

```
#Q2(f)
earning_dataset=(data_set[['name', 'year','earnings']].
 ↪sort_values(by="earnings",ascending=False).head(5))
earning_dataset
```

[25]:

```
                  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
```

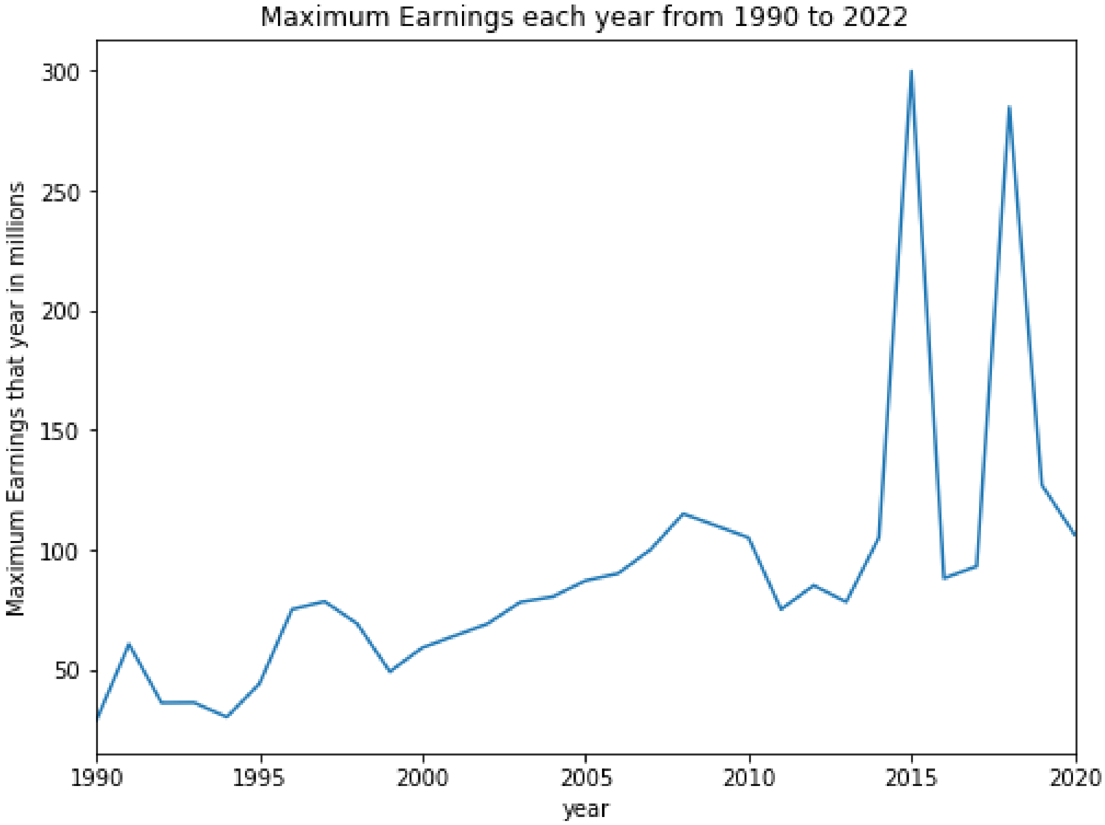
[26]:

```
##Q2(G)
max_data_set=data_set.groupby('year')['earnings'].max()
print(max_data_set)
max_data_set.plot(x="year", y='earnings', figsize=(8, 6))
plt.ylabel('Maximum Earnings that year in millions')
plt.xlabel('year')
plt.title('Maximum Earnings each year from 1990 to 2022')
plt.show()


##Q2(g) the earnings have increased massively throughout the years except it␣
 ↪has two outliers on the years .
##Used this https://www.geeksforgeeks.org/python-pandas-dataframe-groupby/ to␣
 ↪understand how can I use groupby, for my problem.
```

```
year
1990     28.6
1991     60.5
1992     35.9
1993     36.0
1994     30.0
1995     43.9
1996     75.0
1997     78.3
1998     69.0
1999     49.0
2000     59.0
2002     69.0
```

```
2003     78.0
2004     80.3
2005     87.0
2006     90.0
2007    100.0
2008    115.0
2009    110.0
2010    105.0
2011     75.0
2012     85.0
2013     78.1
2014    105.0
2015    300.0
2016     88.0
2017     93.0
2018    285.0
2019    127.0
2020    106.3
Name: earnings, dtype: float64
```



[27]:

```
##Q2(H)
data_set.groupby("nationality")["earnings"].sum().sort_values(ascending=False)
```

[27]:

```
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
Name: earnings, dtype: float64
```

[28]:

```
#Q3(a)
data_frame = pd.read_csv("chicago_salary_sample.csv")
data_frame.annual_salary.mean()
```

[28]:

```
99217.66344
```

[29]:

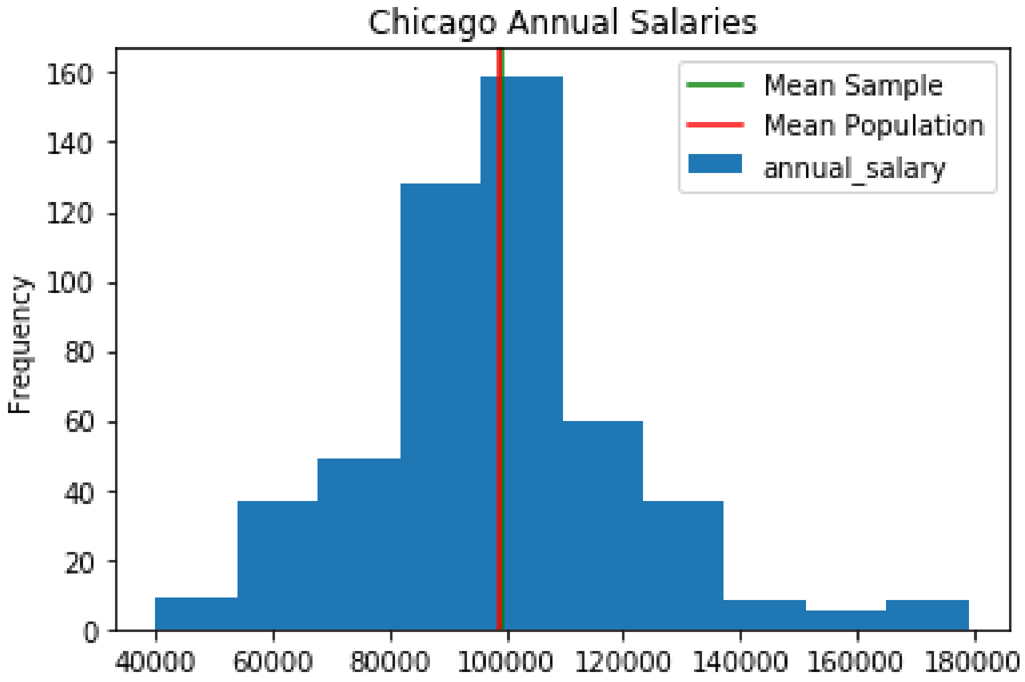
```
#Q3(b)
pop_data_frame = pd.read_csv("chicago_salary_full.csv")
pop_data_frame.annual_salary.mean()
```

[29]:

```
98915.8253718593
```

[30]:

```
#Q3(c)
data_frame['annual_salary'].plot(kind='hist')
plt.axvline(x=99217.66344, color='g',label="Mean Sample")
plt.axvline(x=98915.8253718593, color='r',label="Mean Population")
plt.title("Chicago Annual Salaries")
plt.legend(loc="best")
plt.show()
```



```
[31]: ##Q3(d)
      annual_salary = data_frame['annual_salary'].tolist()
      array_salary=np.array(annual_salary)
      output =np.random.choice(array_salary,size=len(array_salary),replace = True)
      mean=np.mean(output)
      mean

      #use this website to understand how can i convert a dataframe values to a list.␣
       ↪https://www.geeksforgeeks.org/how-to-convert-pandas-dataframe-into-a-list/
```

```
[31]: 99605.95992000001
```

```
[32]: #Q3(e)
      mean_sample=[]
      for i in range(1000):
          output =np.random.choice(array_salary,size=len(array_salary),replace = True)
          mean=np.mean(output)
          mean_sample.append(mean)
      sorted_mean = np.sort(mean_sample)
      upper_confidence= np.percentile(sorted_mean,97.5)
      lower_confidence = np.percentile(sorted_mean,2.5)
      print(upper_confidence)
      print(lower_confidence)
```

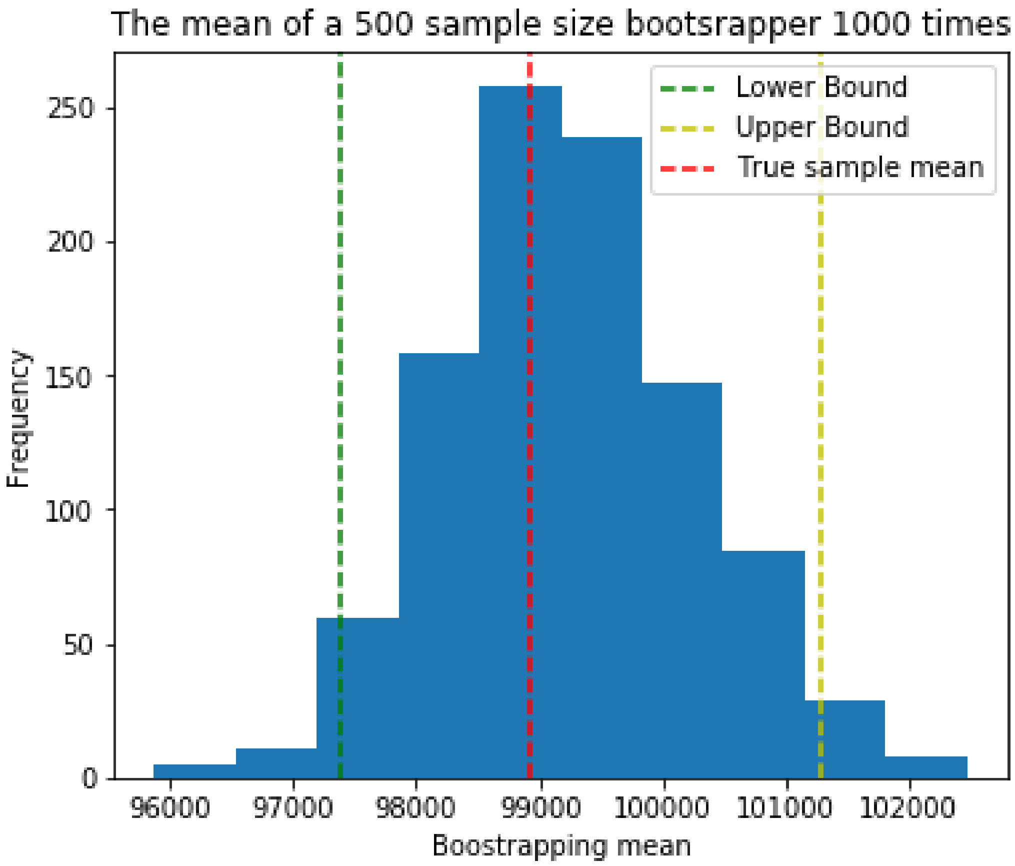
```
101288.19826199999
```

```
97372.872738
```

Q3(e): The value shows that the true mean would be between in the 95% confidence interval in this scenario.

```
[33]: ##Q3(f)
plt.figure(figsize=(6, 5))
plt.subplots_adjust(hspace=0.4)
plt.hist(mean_sample)
plt.xlabel("Boostrapping mean")
plt.ylabel("Frequency")
plt.title("The mean of a 500 sample size bootsrapper 1000 times")
plt.axvline(x=lower_confidence, color='g',linestyle='--',label="Lower Bound")
plt.axvline(x=upper_confidence, color='y',linestyle='--',label="Upper Bound")
plt.axvline(x=98915.8253718593, color='r',linestyle='--',label="True sample␣
  ↪mean")
plt.legend(loc="best")
plt.show
print("Interval:", lower_confidence, ",", upper_confidence)
```

```
Interval: 97372.872738 , 101288.19826199999
```



[34]:
```
##Q4(A)
df =pop_data_frame [(pop_data_frame.department == 'POLICE') | (pop_data_frame.
↪department == 'FIRE')]
df.loc[:,['department', 'annual_salary']].head(5)

##used this https://www.w3resource.com/python-exercises/pandas/practice-set1/
↪pandas-practice-set1-exercise-18.php to learn how can I display only police␣
↪and fire department in this case.
```

[34]:
```
  department  annual_salary
0     POLICE       122568.0
1     POLICE       110796.0
3     POLICE        86730.0
4       FIRE       118830.0
5     POLICE       109236.0
```

[35]:
```
##Q4(b)
police_df= df[df.department == "POLICE"]
print("mean salary for the police department is",police_df.mean())
fire_df= df[df.department == "FIRE"]
print("mean salary for the fire department is",fire_df.mean())
```

```
mean salary for the police department is annual_salary    101170.563985
dtype: float64
mean salary for the fire department is annual_salary    106580.967191
dtype: float64
```

[36]:
```
#Q4(c)
regression = smf.ols('annual_salary ~ department', data=df).fit()    # simple␣
↪linear regression
regression.summary()

#use the lecture notes 9B codebook, to make this regression.
```

[36]:
```
<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:                        17:57:38   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.
"""
```

Q4(c): The coefficient of police department is -5410.4032. From the mean values we found on in part (b) the difference between the mean salary of police department and fire department is -5410.4032.

#Citations: To do this homeowork, I alongside these websites made use of code on the lecture slides, codebooks attached as well as the week 6 lab book attached by the recitation leader Doshi.