ECONC142_FINAL

June 18, 2020

1 Econ C142: Final Project

Submitted by: Aileen Gui

SID#: 3032164825

[321]:
```
#Importing libraries. The same will be used throughout the article.
import numpy as np
import pandas as pd
import random
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib.pylab import rcParams
import matplotlib
#misc packages
import itertools
#plotting
import seaborn as sns
from mpl_toolkits import mplot3d
plt.rcParams['figure.figsize'] = (5,5)
#3d plotting at the end
import plotly.express as px
from IPython.display import Image
#stat analysis
from scipy import stats
from scipy import special
import statsmodels.api as sm
import statsmodels.formula.api as smf
import statsmodels
#Reminder on how to install packages
import sys
!{sys.executable} -m pip install Stargazer
from stargazer.stargazer import Stargazer
from IPython.core.display import HTML
from statsmodels.iolib.summary2 import summary_col
from Oaxaca import Oaxaca
```

```
Requirement already satisfied: Stargazer in /srv/app/venv/lib/python3.6/site-
```

```
packages
You are using pip version 9.0.3, however version 20.1 is available.

You should consider upgrading via the 'pip install --upgrade pip' command.
```

Stargazer - python compatible sources https://github.com/mwburke/stargazer

https://pypi.org/project/stargazer/

[322]:
```
#importing given datasets
rd = pd.read_csv("project2020_rd.csv") #dataset for problem 2
dd = pd.read_csv("project2020_dd.csv") #dataset for problem 1
```

2 1.1 Table 1 and Figure 1

Goal: compare the characteristics and wages of male and female workers, focusing on period 0

Format: - column 1 = characteristics for all workers - column 2 = characteristics for female workers - column 3 = characteristics for male workers - column 4 = test statistic comparing females and males (t test)

Main characteristics of interest are: - education - age - log of real hourly wage (y) - mean log real hour wage for co-workers as of period 0 (owage2)

[323]:
```
#Table 1
#cleaning datasets
ddsubset = dd[['y', 'educ', 'age', 'owage2']]
dd_subset = pd.DataFrame(ddsubset.mean()).rename(columns = {0: "All Workers"})

dd_allfemale = dd[dd["female"] == 1][['y', 'educ', 'age', 'owage2']]
ddallfemale = pd.DataFrame(dd_allfemale.mean()).rename(columns = {0: "Female␣
↪Only"})

dd_allmale = dd[dd["female"] == 0][['y', 'educ', 'age', 'owage2']]
ddallmale = pd.DataFrame(dd_allmale.mean()).rename(columns = {0: "Male Only"})
```

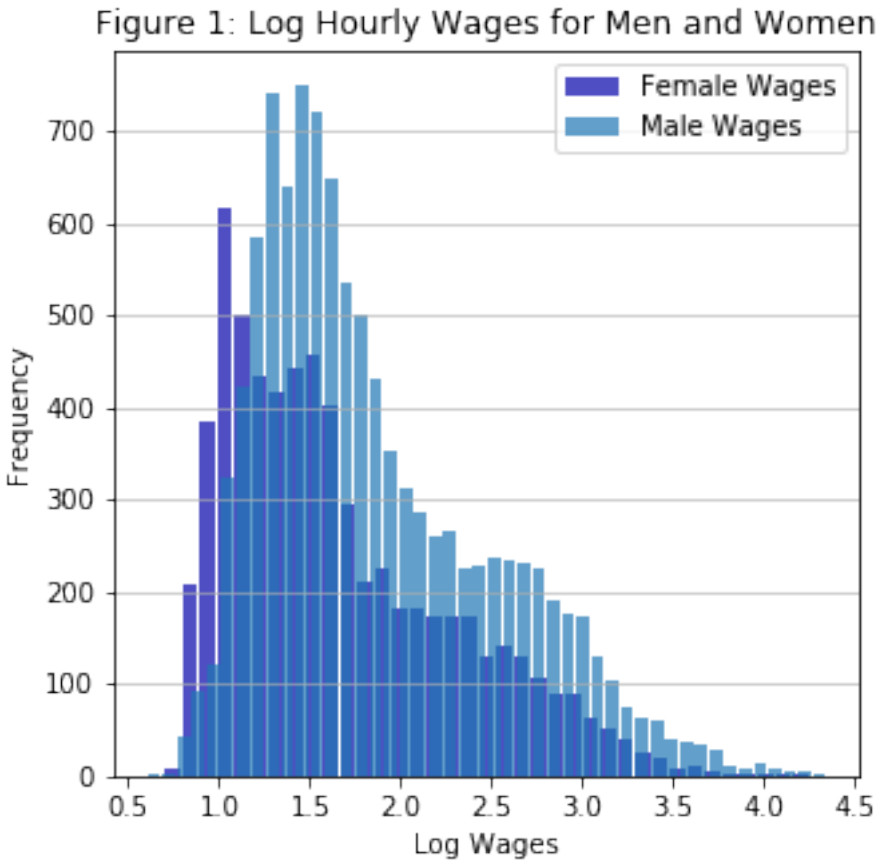
[324]:
```
#Table 1
#adding t-test column
horizontal_stack = pd.concat([dd_subset, ddallfemale, ddallmale], axis=1, sort␣
↪= False)
eductstat = stats.ttest_ind(dd_allfemale['educ'], dd_allmale['educ'], axis = 0,␣
↪equal_var = True)
agetstat = stats.ttest_ind(dd_allfemale['age'], dd_allmale['age'], axis = 0,␣
↪equal_var = True, nan_policy = 'omit')[0]
ytstat = stats.ttest_ind(dd_allfemale['y'], dd_allmale['y'], axis = 0,␣
↪equal_var = True, nan_policy = 'omit')[0]
owage2tstat = stats.ttest_ind(dd_allfemale['owage2'], dd_allmale['owage2'],␣
↪axis = 0, equal_var = True, nan_policy = 'omit')[0]
```

```
compare = [ytstat, eductstat, agetstat, owage2tstat]
horizontal_stack['Male-Female Comparison'] = compare
horizontal_stack.style.set_caption("Table 1: Worker Characteristics")
```

[324]: <pandas.io.formats.style.Styler at 0x7f1b570332b0>

[325]:
```
#Figure 1
#plot the smoothed histograms of log hourly wages for men and women on one␣
 ↪figure

plt.grid(axis='y', alpha=0.75)
plt.hist(x = dd_allfemale['y'], bins = 'auto', color='#0504aa', alpha=0.7,␣
 ↪rwidth = 0.85, label='Female Wages')
plt.hist(x = dd_allmale['y'], bins = 'auto', alpha=0.7, rwidth = 0.85,␣
 ↪label='Male Wages')
plt.legend(loc='upper right')
plt.xlabel('Log Wages')
plt.ylabel('Frequency')
plt.title('Figure 1: Log Hourly Wages for Men and Women')
```

[325]: Text(0.5, 1.0, 'Figure 1: Log Hourly Wages for Men and Women')



Narrative: Briefly discuss the main differences between men and women using the table and figure to make your points

MY ANSWER: The main difference between men and women is in mean log hourly wages. Shown in the figure "Log Hourly Wages for Men and Women", we see that the histogram for men looks almost like a copy of the histogram for women but shifted to the right, meaning that, on average, men make higher hourly wages than women in this dataset. We see this belief confirmed in Table 1 as well, where the coefficient relating gender to log wages is high for men than it is for women.

3 1.2 Table 2

Goal: fit a series of standard wage models for wages in period 0 and construct Oaxaca decompositions of the wage gap between men and women

3.1 Part A

Fit 2 models using the pooled data for men and women - including only a constant and a female dummy - including a constant, education, a cubic in experience, and a female dummy

[326]:
```
#first model
q2data1 = dd[['female']]
q2data1 = sm.add_constant(q2data1)
yvar = dd['y']
q2model1 = sm.OLS(yvar, q2data1).fit()
#second model
dd['exp_cubed'] = pow(dd['exp'], 3)
q2data2 = dd[['educ', 'exp_cubed', 'female']]
q2data2 = sm.add_constant(q2data2)
q2model2 = sm.OLS(yvar, q2data2).fit()
```

3.2 Part B

Fit separate models for men and women that include a constant, education, and a cubic in experience

Use the models to construct standard Oaxaca decompositions as in Lecture 7 - construct BOTH of the 2 alternatives

[327]:
```
#third model
femaleonly = dd[dd["female"] == 1][['educ', 'exp_cubed']]
maleonly = dd[dd["female"] == 0][['educ', 'exp_cubed']]
femaleonly = sm.add_constant(femaleonly)
maleonly = sm.add_constant(maleonly)
```

```
yvarfemale = dd[dd["female"] ==1]['y']
femaleonlymodel = sm.OLS(yvarfemale, femaleonly).fit()
#fourth model
yvarmale = dd[dd["female"] ==0]['y']
maleonlymodel = sm.OLS(yvarmale, maleonly).fit()
```

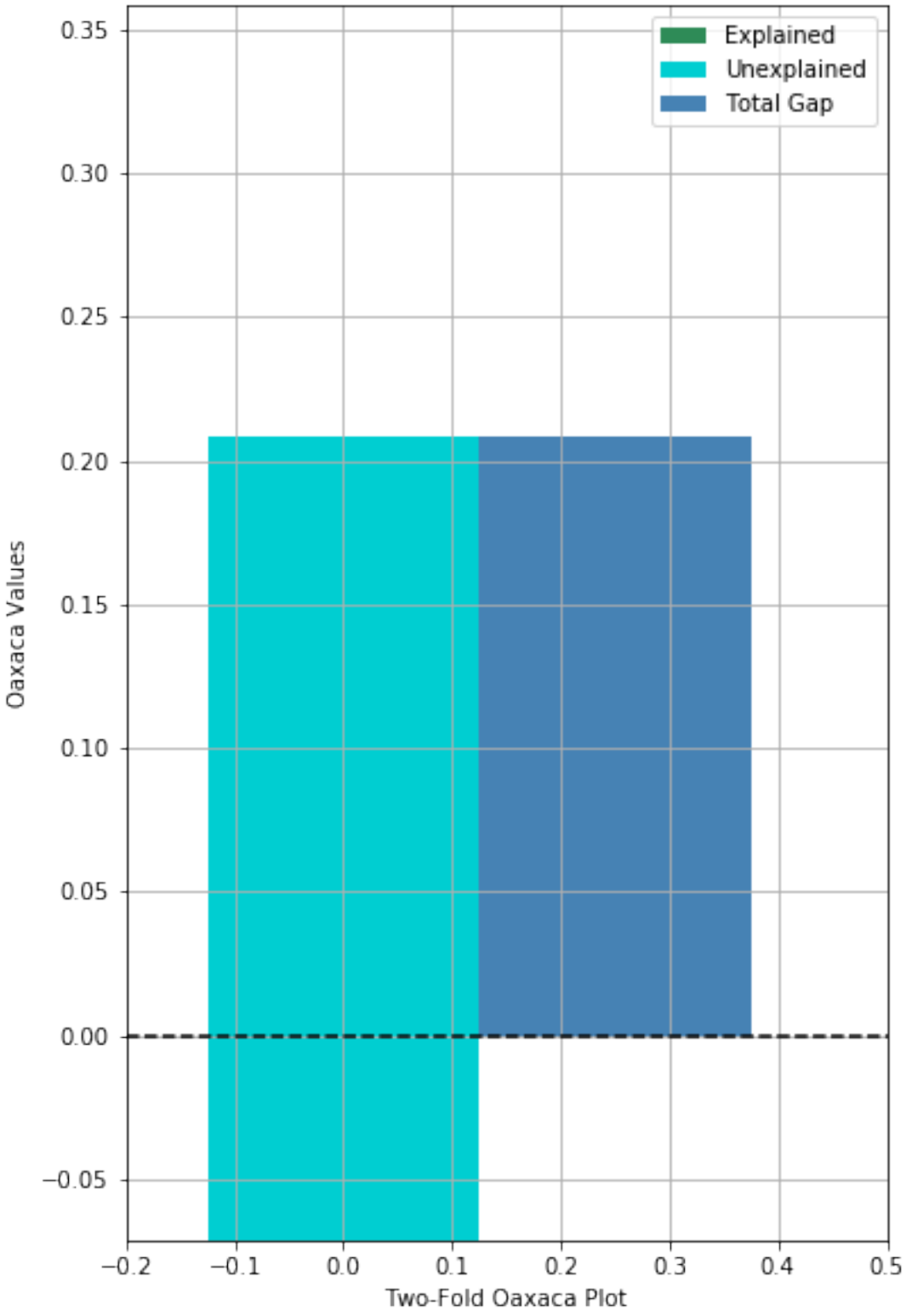
[328]:
```
#Table representation of results
stargazer = Stargazer([q2model1, q2model2, femaleonlymodel, maleonlymodel])
stargazer.title("Table 2: Examining Mean Log Wage Characteristics")
stargazer.custom_columns(["Entire Sample1", "Entire Sample2", "Female Only",␣
↪"Male Only"], [1,1,1,1])
HTML(stargazer.render_html())
```

[328]:
```
<IPython.core.display.HTML object>
```

3.3 Two-Fold Oaxaca Decomposition

[329]:
```
#Two-fold oaxaca decomposition
oaxacadata = dd[['y', 'educ', 'exp_cubed', 'female']]
model = Oaxaca(oaxacadata, by = 'female', endo = 'y')
model.fit(two_fold = True, three_fold = False)
model.plot(plt_type = 2)
```

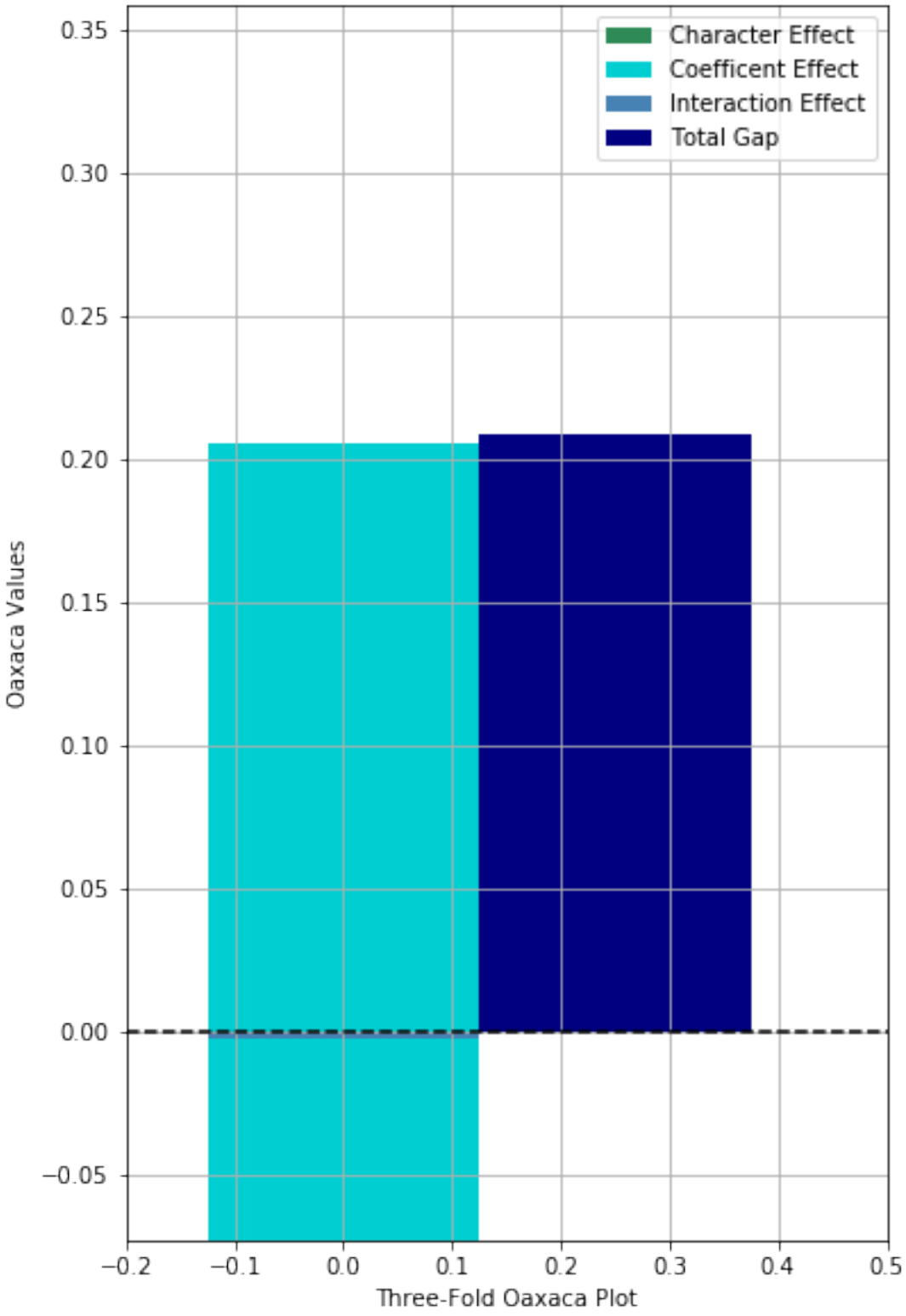
```
These are the attempted split values: Int64Index([0, 1], dtype='int64')
Unexplained Effect: 0.27977
Explained Effect: -0.07136
Gap: 0.2084
```


Explained
Unexplained
Total Gap
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.05
Oaxaca Values
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
Two-Fold Oaxaca Plot

3.4 Three-Fold Oaxaca Decomposition

[330]:
```
#Three-fold oaxaca decomposition
model1 = Oaxaca(oaxacadata, "female", 'y')
model1.fit(two_fold = False, three_fold = True)
model1.plot(3)
```

```
These are the attempted split values: Int64Index([0, 1], dtype='int64')
Characteristic Effect: -0.0731
Coefficent Effect: 0.2786
Interaction Effect: 0.00291
Gap: 0.2084
```



Narrative: Briefly discuss the decomposition. HINT: you will see that females are better educated

than males so the models do not "explain" the gender gap. In fact, they suggest the observed raw wage difference between men and women understates the gender gap. What do you think of this?

MY ANSWER: We can see from Table 1 that, on average, females are better educated than males. Furthermore, the education coefficient for females is higher than the education coefficient for males, as seen from Table 2. This would suggest that females would have higher average earnings than men, but this is not the case. Seen in the 2-fold oaxaca decomposition, there is a 0.27977 unexplained effect still present in wages. Characteristic effect accounts for -0.731 of the gap, which means the coefficient effect of .2786 can be interpreted as a measure of labor market discrimination.

4 1.3 Table 3

Goal: examine the effect of a new control variable, which is the mean log wage of a person's co-workers (the variable owage2)

4.1 Part A

Fit 2 models using the pooled data for men and women - including only a constant, female dummy, and owage2 - including a constant, education, a cubic in experience, a female dummy, and owage2

[331]:
```
#first model
data3 = dd[['female', 'owage2']]
data3 = sm.add_constant(data3)
yvar = dd['y']
model3 = sm.OLS(yvar, data3).fit()
```

[332]:
```
#second model
data4 = dd[['educ', 'exp_cubed', 'female', 'owage2']]
data4 = sm.add_constant(data4)
model4 = sm.OLS(yvar, data4).fit()
```

4.2 Part B

Fit separate models for men and women that include a constant, education, and a cubic in experience, and owage2

Use these models to conduct a pair of new decompositions that accounts for the effect of higher-wage coworkers (as above - construct BOTH decompositions)

[333]:
```
#men only
maleonly3 = dd[dd["female"] == 0][['educ', 'exp_cubed', 'owage2']]
maleonly3 = sm.add_constant(maleonly3)
maleonlymodel3 = sm.OLS(yvarmale, maleonly3).fit()
```

[334]:
```
#women only
femaleonly3 = dd[dd["female"] == 1][['educ', 'exp_cubed', 'owage2']]
femaleonly3 = sm.add_constant(femaleonly3)
yvarfemale = dd[dd["female"] == 1]['y']
femaleonlymodel3 = sm.OLS(yvarfemale, femaleonly3).fit()
```

[335]:
```
#Table representation of results
stargazer3 = Stargazer([model3, model4, femaleonlymodel3, maleonlymodel3])
stargazer3.custom_columns(["Entire Sample1", "Entire Sample2", "Female Only",␣
↪"Male Only"], [1,1,1,1])
stargazer3.title("Table 3: Examining Mean Log Wage Characteristics")
HTML(stargazer3.render_html())
```

[335]:
```
<IPython.core.display.HTML object>
```

4.3 Two-Fold Oaxaca Decomposition

[336]:
```
#Two-fold decomposition
oaxaca2data = dd[['y', 'educ', 'exp_cubed', 'female', 'owage2']]
oaxacamodel2 = Oaxaca(oaxaca2data, by = 'female', endo = 'y')
oaxacamodel2.fit(two_fold = True, three_fold = False)
```

```
These are the attempted split values: Int64Index([0, 1], dtype='int64')
Unexplained Effect: 0.19582
Explained Effect: 0.01259
Gap: 0.2084
```

4.4 Three-Fold Oaxaca Decomposition

[337]:
```
#Three-fold decomposition
oaxacamodel2.fit(two_fold = False, three_fold = True)
```

```
Characteristic Effect: 0.00918
Coefficent Effect: 0.1932
Interaction Effect: 0.00602
Gap: 0.2084
```

Narrative: In your narrative you will discuss alternative interpretations of the effect of working with highly-paid co-workers. Think about two possible explanations for why people who work with higher-paid co-workers earn more. - Model 1: getting a job with high paid co-workers is largely a matter of good luck or connections, and men have better connections, or search harder to find higher paid coworkers jobs. - Model 2: getting a job with highly paid co-workers is only possible for workers who have high levels of cognitive skills or ambition, which is not measured in our data but potentially varies by gender.

MY ANSWER: In this oaxaca decomposition, we've included another variable from the one done in question 1.2. We've included the pay of co-workers, and this has led to a significant dent in explaining the difference in wages between men and women. Now, the unexplained effect has decreased to 0.19582

One reason having higher paid coworkers may affect the gender pay gap is given in model 1: higher paid coworkers are a result of good connections, and men may have better connections or search harder to find jobs with higher paid coworkers. This might be plausible if men are on average more involved in extracurricular activities that create connections, network more, or if male-dominated industries coincide with higher-paying industries (such as investment banking or software engineering).

Another reason having higher paid coworkers may affect the gender pay gap is due to omitted variable bias as mentioned in model 2. Perhaps men are more likely to have higher paid co-workers because those jobs are reserved for those with high levels of cognitive ability or ambition, assuming these traits vary by gender. A common argument in favor of this model can be seen with negotiation confidence. For instance, if men are more likely to be assertive when asking for raises or negotiating work-contract terms. If men are, on average, more ambitious than women, then men might ask for promotions more frequently or engage in strategic networking more frequently.

5 1.4 Table 4 and Figure 2

Use the fact that we have job changers in the data to conduct some event studies. Do an analysis of wage changes as people move between jobs with higher and lower paid co-workers.

Setup

- Find the terciles of owage1
- Classify all the first jobs (held in periods -3, -2, and -1) into 3 groups based on the tercile of owage1
- Find the terciles of owage2
- Classify all second jobs (held in periods 0, 1, 2) into 3 groups
- Classify workers into 9 groups based on tercile of owage1 x tercile of owage2

[338]:
```
#Dividing data into terciles according to owage1
owage1_quantile = dd['owage1'].quantile([0, (1/3), (2/3)]) #everything above␣
↪until the next = in that tercile
#Dividing data into terciles according to owage2
owage2_quantile = dd['owage2'].quantile([0, (1/3), (2/3)])
```

[339]:
```
#taking the means of all the first jobs
first_job = []
for i in np.arange(len(dd['yl1'])):
    value = np.mean([dd['yl1'].loc[i], dd['yl2'].loc[i], dd['yl3'].loc[i]])
    first_job.append(value)
dd['first_job'] = first_job
```

```
#taking the means of alll the second jobs
second_job = [np.mean([dd['y'].loc[i], dd['yp1'].loc[i], dd['yp2'].loc[i]]) for␣
 ↪i in np.arange(len(dd['yp1']))]
dd['second_job'] = second_job

#classifying first jobs into owage1
first_job_tercile = []
for value in dd['first_job']:
    if value < owage1_quantile.iloc[1]:
        first_job_tercile.append(0)
    elif value >= owage1_quantile.iloc[1] and value < owage1_quantile.iloc[2]:
        first_job_tercile.append(1)
    elif value >= owage1_quantile.iloc[2]:
        first_job_tercile.append(2)

dd['first_job_tercile'] = first_job_tercile


#classifying second jobs into owage2
second_job_tercile = []
for value in dd['second_job']:
    if value < owage2_quantile.iloc[1]:
        second_job_tercile.append(0)
    elif value >= owage2_quantile.iloc[1] and value < owage2_quantile.iloc[2]:
        second_job_tercile.append(1)
    elif value >= owage2_quantile.iloc[2]:
        second_job_tercile.append(2)

dd['second_job_tercile'] = second_job_tercile
```

[340]:
```
#Classify workers into 9 groups based on tercile of owage1 x tercile of owage2
# 0 = 00
# 1 = 01
# 2 = 02
# 3 = 10
# 4 = 11
# 5 = 12
# 6 = 20
# 7 = 21
# 8 = 22
combo_groups = []
for i in np.arange(len(dd['first_job_tercile'])):
    if dd['first_job_tercile'].iloc[i] == 0 and dd['second_job_tercile'].
 ↪iloc[i] == 0:
        combo_groups.append(0)
    elif dd['first_job_tercile'].iloc[i] == 0 and dd['second_job_tercile'].
 ↪iloc[i] == 1:
```

```
        combo_groups.append(1)
    elif dd['first_job_tercile'].iloc[i] == 0 and dd['second_job_tercile'].
↪iloc[i] == 2:
        combo_groups.append(2)
    elif dd['first_job_tercile'].iloc[i] == 1 and dd['second_job_tercile'].
↪iloc[i] == 0:
        combo_groups.append(3)
    elif dd['first_job_tercile'].iloc[i] == 1 and dd['second_job_tercile'].
↪iloc[i] == 1:
        combo_groups.append(4)
    elif dd['first_job_tercile'].iloc[i] == 1 and dd['second_job_tercile'].
↪iloc[i] == 2:
        combo_groups.append(5)
    elif dd['first_job_tercile'].iloc[i] == 2 and dd['second_job_tercile'].
↪iloc[i] == 0:
        combo_groups.append(6)
    elif dd['first_job_tercile'].iloc[i] == 2 and dd['second_job_tercile'].
↪iloc[i] == 1:
        combo_groups.append(7)
    elif dd['first_job_tercile'].iloc[i] ==2 and dd['second_job_tercile'].
↪iloc[i]== 2:
        combo_groups.append(8)

dd['combo_groups'] = combo_groups
```

6 1.4 Figure 2

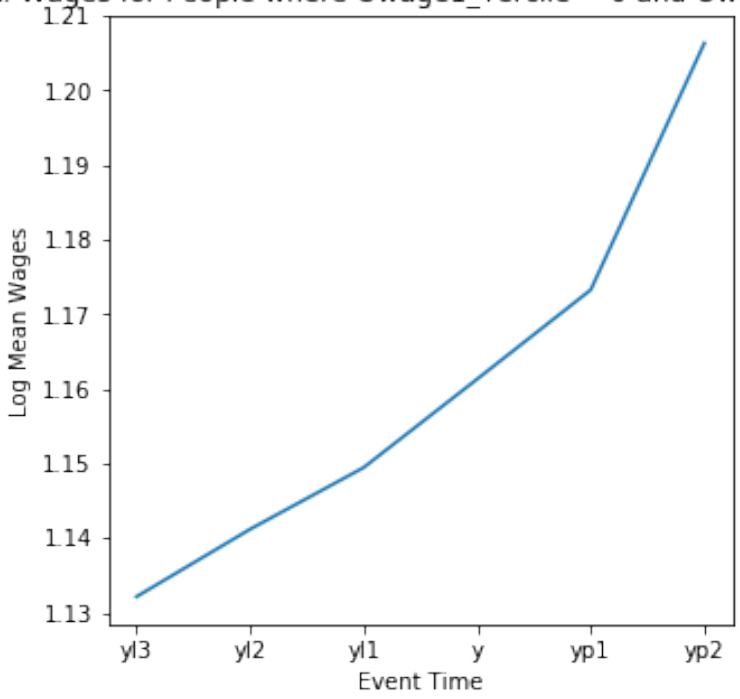
Show 9 separate plots of mean wages over time for people who start in each tercile of owage1 and go to each tercile of owage2 - x-axis = ”event time” which ranges from -3-+2

[341]:
```
#combo group 0
data0 = dd[dd['combo_groups'] == 0][['yl3', 'yl2', 'yl1', 'y', 'yp1', 'yp2']]
y0 = np.mean(data0)
plt.plot(y0)
plt.title("Figure 2A: Wages for People where Owage1_Tercile = 0 and␣
 ↪Owage2_Tercile = 0")
plt.xlabel("Event Time")
plt.ylabel("Log Mean Wages")
plt.figure(figsize = (1, 1))
plt.show()
#combo group 1
data1 = dd[dd['combo_groups'] == 1][['yl3', 'yl2', 'yl1', 'y', 'yp1', 'yp2']]
y1 = np.mean(data1)
plt.plot(y1)
```

```
plt.title("Figure 2B: Wages for People where Owage1_Tercile = 0 and␣
 ↪Owage2_Tercile = 1")
plt.xlabel("Event Time")
plt.ylabel("Log Mean Wages")
plt.figure(figsize = (1, 1))
plt.show()
#combo group 2
data2 = dd[dd['combo_groups'] == 2][['yl3', 'yl2', 'yl1', 'y', 'yp1', 'yp2']]
y2 = np.mean(data2)
plt.plot(y2)
plt.title("Figure 2C: Wages for People where Owage1_Tercile = 0 and␣
 ↪Owage2_Tercile = 2")
plt.xlabel("Event Time")
plt.ylabel("Log Mean Wages")
plt.figure(figsize = (1, 1))
plt.show()
#combo group 3
data3 = dd[dd['combo_groups'] == 3][['yl3', 'yl2', 'yl1', 'y', 'yp1', 'yp2']]
y3 = np.mean(data3)
plt.plot(y3)
plt.title("Figure 2D: Wages for People where Owage1_Tercile = 1 and␣
 ↪Owage2_Tercile = 0")
plt.xlabel("Event Time")
plt.ylabel("Log Mean Wages")
plt.figure(figsize = (1, 1))
plt.show()
#combo group 4
data4 = dd[dd['combo_groups'] == 4][['yl3', 'yl2', 'yl1', 'y', 'yp1', 'yp2']]
y4 = np.mean(data4)
plt.plot(y4)
plt.title("Figure 2E: Wages for People where Owage1_Tercile = 1 and␣
 ↪Owage2_Tercile = 1")
plt.xlabel("Event Time")
plt.ylabel("Log Mean Wages")
plt.figure(figsize = (1, 1))
plt.show()
#combo group 5
data5 = dd[dd['combo_groups'] == 5][['yl3', 'yl2', 'yl1', 'y', 'yp1', 'yp2']]
y5 = np.mean(data5)
plt.plot(y5)
plt.title("Figure 2F: Wages for People where Owage1_Tercile = 1 and␣
 ↪Owage2_Tercile = 2")
plt.xlabel("Event Time")
plt.ylabel("Log Mean Wages")
plt.figure(figsize = (1, 1))
plt.show()
#combo group 6
```

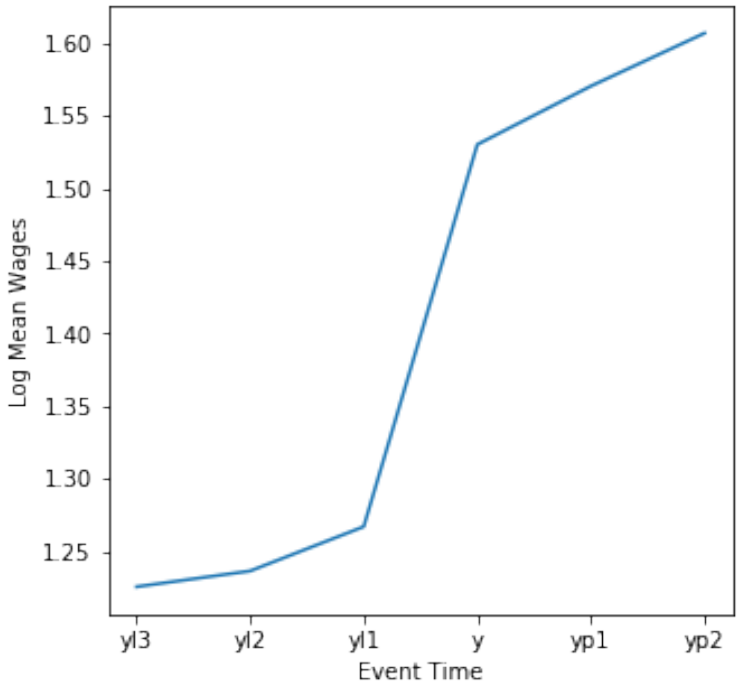
```
data6 = dd[dd['combo_groups'] == 0][['yl3', 'yl2', 'yl1', 'y', 'yp1', 'yp2']]
y6 = np.mean(data6)
plt.plot(y6)
plt.title("Figure 2G: Wages for People where Owage1_Tercile = 2 and␣
 ↪Owage2_Tercile = 0")
plt.xlabel("Event Time")
plt.ylabel("Log Mean Wages")
plt.figure(figsize = (1, 1))
plt.show()
#combo group 7
data7 = dd[dd['combo_groups'] == 7][['yl3', 'yl2', 'yl1', 'y', 'yp1', 'yp2']]
y7 = np.mean(data7)
plt.plot(y7)
plt.title("Figure 2H: Wages for People where Owage1_Tercile = 2 and␣
 ↪Owage2_Tercile = 1")
plt.xlabel("Event Time")
plt.ylabel("Log Mean Wages")
plt.figure(figsize = (1, 1))
plt.show()
#combo group 8
data8 = dd[dd['combo_groups'] == 8][['yl3', 'yl2', 'yl1', 'y', 'yp1', 'yp2']]
y8 = np.mean(data8)
plt.plot(y8)
plt.title("Figure 2I: Wages for People where Owage1_Tercile = 2 and␣
 ↪Owage2_Tercile = 2")
plt.xlabel("Event Time")
plt.ylabel("Log Mean Wages")
plt.figure(figsize = (1, 1))
plt.show()
```

Figure 2A: Wages for People where Owage1_Tercile = 0 and Owage2_Tercile = 0



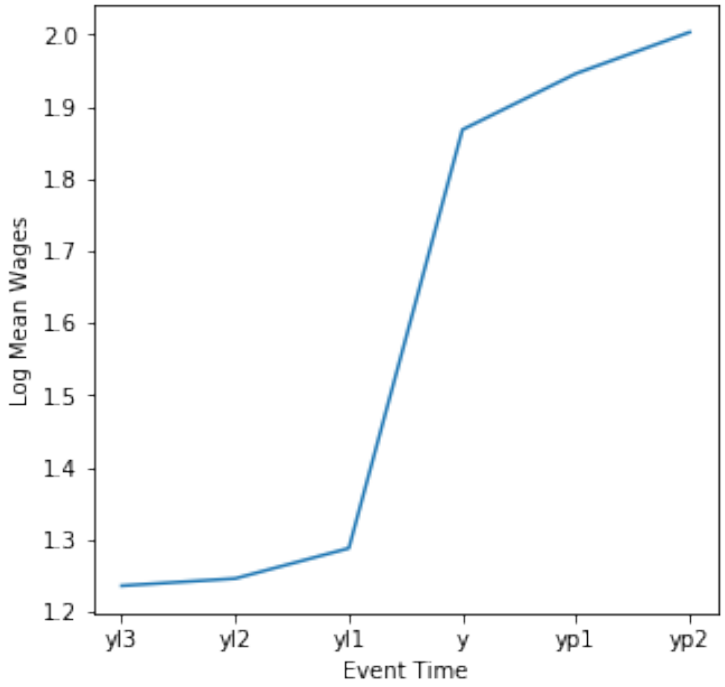
<Figure size 72x72 with 0 Axes>

Figure 2B: Wages for People where Owage1_Tercile = 0 and Owage2_Tercile = 1



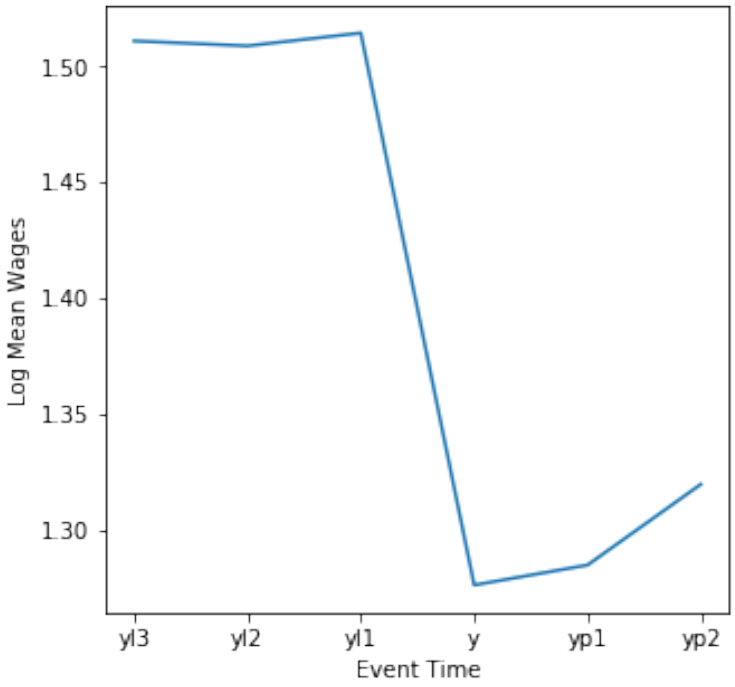
<Figure size 72x72 with 0 Axes>

Figure 2C: Wages for People where Owage1_Tercile = 0 and Owage2_Tercile = 2



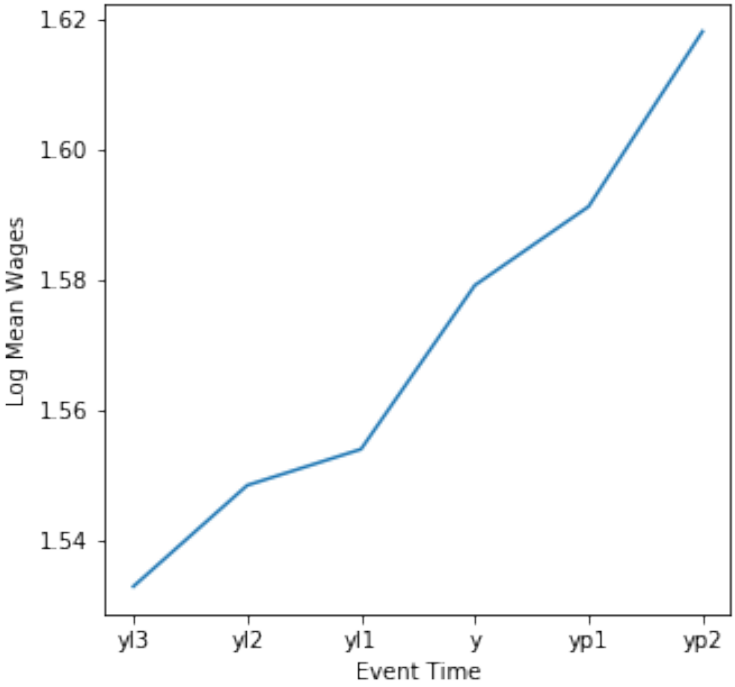
<Figure size 72x72 with 0 Axes>

Figure 2D: Wages for People where Owage1_Tercile = 1 and Owage2_Tercile = 0



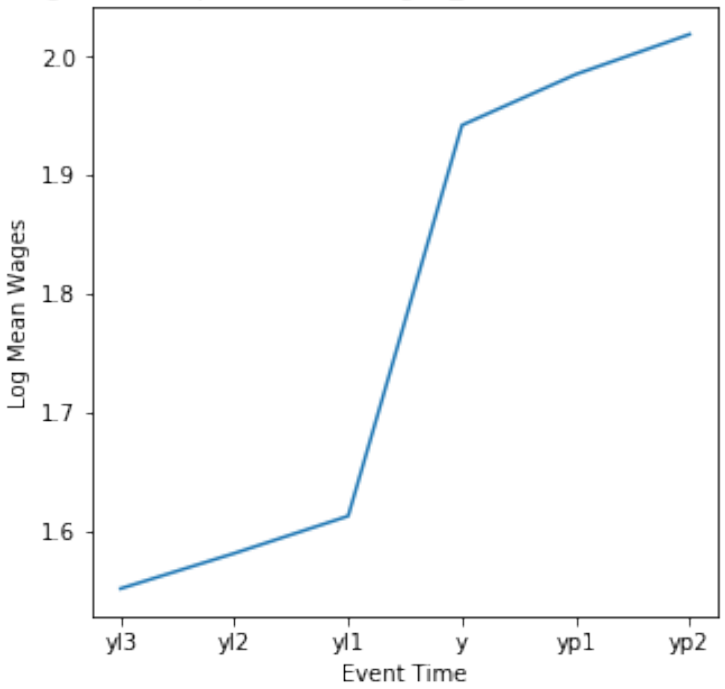

```
<Figure size 72x72 with 0 Axes>
```

Figure 2E: Wages for People where Owage1_Tercile = 1 and Owage2_Tercile = 1



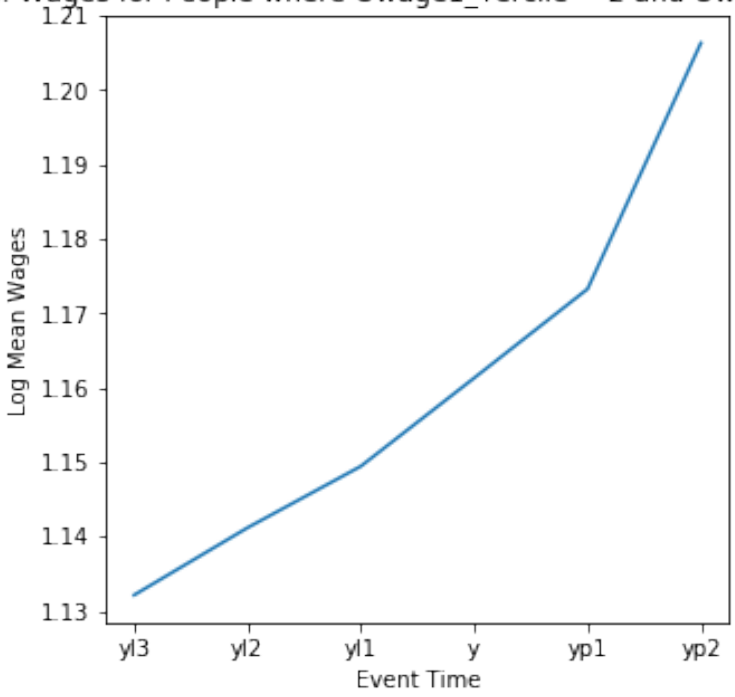
<Figure size 72x72 with 0 Axes>

Figure 2F: Wages for People where Owage1_Tercile = 1 and Owage2_Tercile = 2



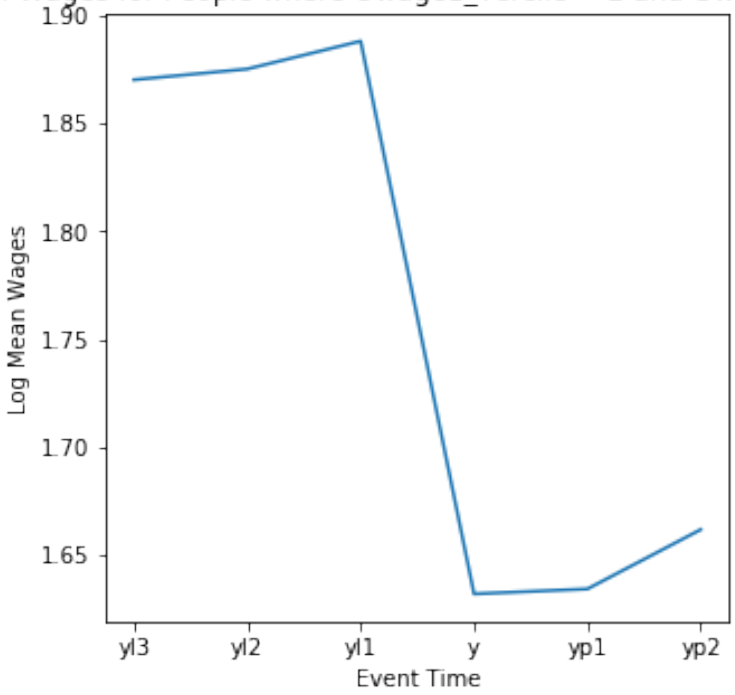
<Figure size 72x72 with 0 Axes>

Figure 2G: Wages for People where Owage1_Tercile = 2 and Owage2_Tercile = 0

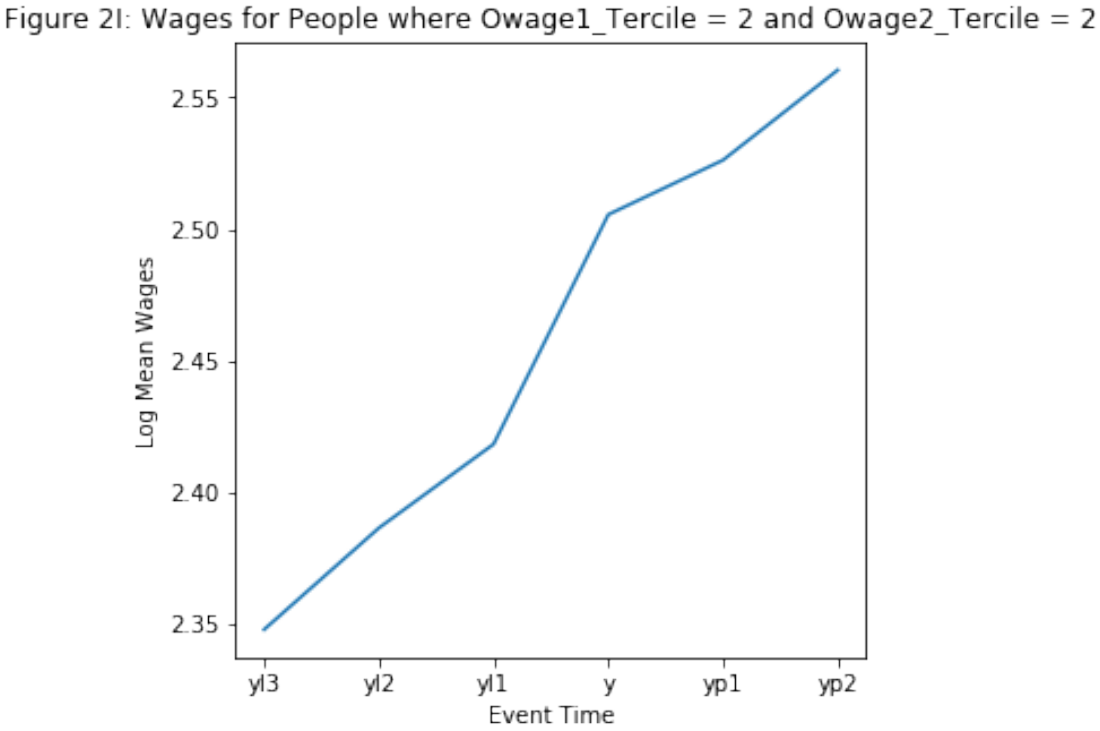



```
<Figure size 72x72 with 0 Axes>
```

Figure 2H: Wages for People where Owage1_Tercile = 2 and Owage2_Tercile = 1



```
<Figure size 72x72 with 0 Axes>
```




```
<Figure size 72x72 with 0 Axes>
```

Narrative: Think carefully about the alternative models (Model 1 and Model 2) and why co-worker wages matter. Then discuss the event study graphs. Do these graphs provide more support for Model 1 or Model 2? Also, do you see any pattern of wage movements before a job change that lead you to be concerned?

MY ANSWER: Co-worker wages matter because we can see from the above figures that all combinations of co-worker wages for the first job versus second job who either stay in the same tercile or jump to a higher tercile in their second job experience a steady increase in log wages over time. Note that this is not the case for those whose co-workers' wages decrease from first job to second job. This suggests strongly that co-worker wages are important because they indicate how the individual's wages will shift over time. Those who start out in the second tercile will continue experience increased log wages even if they do not explicitly jump terciles. This result is true across all figures above where coworkers' wages stay in the same tercile from first job to second job, or jump to the tercile above from first job to second job.

This strongly suggests that model 2 (omitted variables related to individual propensity towards ambition) is not accurate. If we were to encounter omitted variable bias related to individual skill,

we might expect less consistent results than the ones seen in the above figures.

6.1 1.4 Table Part A

Fit a set of models for the change in wages using the pooled data for men and women - including only a constant, a female dummy, and Dwage = owage2-owage1 - including a constant, a quadratic in experience as of period -1, a female dummy, and Dwage

[342]:
```
#model5: constant, female dummy, Dwage
dd5 = dd
dd5['Dwage'] = dd['owage2'] - dd['owage1']
dd5 = dd[["female", "Dwage"]]
dd5 = sm.add_constant(dd5)
yvar5 = dd['y']
dd5model = sm.OLS(yvar5, dd5).fit()
```

[343]:
```
#model6: constant, quadratic in experience as of period -1, female, Dwage
dd6 = dd
dd6['Dwage'] = dd['owage2'] - dd['owage1']
dd6['exp_quad'] = pow(dd6['exp'] - 1, 4)
dd6 = dd[['exp_quad', 'female', 'Dwage']]
dd6 = sm.add_constant(dd6)
yvar6 = dd['y']
dd6model = sm.OLS(yvar6, dd6).fit()
```

6.2 1.4 Table Part B

Fit separate models for men and women that include a constant, quadratic in experience as of period -1, and Dwage

**Note: experience in period -1 is just experience in period 0 minus 1

[344]:
```
#men only
dd7 = dd
dd7['exp_quad'] = pow(dd7['exp'] - 1, 4)
dd7 = dd7[['exp_quad', 'female', 'Dwage']]
maleonly4 = dd7[dd7["female"] == 0][['exp_quad', 'Dwage']]
maleonly4 = sm.add_constant(maleonly4)
yvarmaleonly4 = dd[dd["female"] == 0][['y']]
maleonlymodel4 = sm.OLS(yvarmaleonly4, maleonly4).fit()
```

[345]:
```
#female only
dd8 = dd
dd8['exp_quad'] = pow(dd8['exp'] - 1, 4)
dd8 = dd8[['exp_quad', 'female', 'Dwage']]
femaleonly4 = dd8[dd8["female"] == 1][['exp_quad', 'Dwage']]
```

```
femaleonly4 = sm.add_constant(femaleonly4)
yvarfemaleonly4 = dd[dd["female"] == 1][['y']]
femaleonlymodel4 = sm.OLS(yvarfemaleonly4, femaleonly4).fit()
```

[346]:
```
#Table representation of results
q4results = Stargazer([dd5model, dd6model, femaleonlymodel4, maleonlymodel4])
q4results.custom_columns(["Entire Sample1", "Entire Sample2", "Female Only",␣
↪"Male Only"], [1,1,1,1])
q4results.title("Table 4: Examining Wage Differences Coworkers for between␣
↪First and Second Jobs")

HTML(q4results.render_html())
```

[346]:
```
<IPython.core.display.HTML object>
```

Narrative: The main issue in this part of the narrative is the comparison between the effect of coworker average wages in OLS models (Table 3) and first-differenced models that control for all unobserved characteristics of people (Table 4).

One way to summarize the two sets of results is to ask: what fraction of the OLS effect of co-worker wages do we see in the first-differenced models? If, for example, the OLS model for males gives a coefficient on coworker wages of 0.66, but the differenced model gives a coefficient of 0.33, then you might conclude that one half of the OLS effect is a causal effect and the other half reflects differences in the unobserved skills of people who tend to work at high-coworker wage jobs.

If the true causal effect of coworker wages for men (from the differenced model) is `lambda^m` and the true causal effect of coworker wages for women (from the differenced model) is `lambda^f` explain how you would modify the decompositions you developed from Table 3 to adjust for the true effects of co-worker wages. What does it imply?

MY ANSWER: In order to capture the true causal effect of coworker wages for men and women, we should adjust the table 3 models by changing the dependent variable to subtract the owage2 to account for the difference between groups over time from first job to second job. This effectively seeks to control for any baseline bias between men and women. From this decomposition, we can see the "treatment effect" of coworkers' wages on the individual's wages. This implies that in order to decrease the gender-pay gap, perhaps policy should be aimed towards increasing the proportion of women in industries with highly paid coworkers that are traditionally male-dominated.

7 2.1 See Proofs

[347]:
```
%%latex

(i) Prove that:
$$E\left(w_{i} | C(0)\right)=\frac{E\left(w_{i} | AT(0) \text { or }␣
↪C(0)\right) \times P(A T(0) or C(0))-E[w_i | A T(0)] \times␣
↪P(AT(0))}{P(C(0))}$$
$$$$
```

```
Given:
$$
E\left(w_{i} | AT(0)\right)=E\left(w_{i} | D_{i}=1, x_{i} \rightarrow 0,
 ↪z_{i}=0\right)
$$

$$E\left(w_{i} | AT(0) \text { or } C(0)\right)=E\left(w_{i} | D_i=1, x_i
 ↪\rightarrow 0, z_i=0\right)$$

$$\Rightarrow E\left(w_{i} | AT(0) \text { or } C(0)\right)=\frac{\left.
 ↪E\left(w_{i} | AT(0)\right)\right) P(AT(0)+E\left(w_{i} | C(0)\right)
 ↪P(C(0))}{P(AT(0) \text { or } C(0))}$$

$$\left.\Rightarrow E\left(w_{i} | AT(0) \text { or } C(0)\right)(P(AT(0) \text
 ↪{ or } C(0))=E\left(w_{i} | AT(0)\right) P(AT(0)\right)+E\left(w_{i} |
 ↪C(0)\right)P(C(0))$$

$$\Rightarrow \frac{E\left[w_{i}\left|AT(0) \text { or } C(0)\right] [P(AT(0) \text {
 ↪or } C(0)] -E[w_i | AT(0)] P(AT(0))\right.\right.}{P(C(0))}=E\left(w_{i}
 ↪|(C(0)\right)$$
```

(i) Prove that:

$$E\left(w_i|C(0)\right)=\frac{E\left(w_i|AT(0) \text{ or } C(0)\right) \times P(AT(0) or C(0))-E[w_i|AT(0)] \times P(AT(0))}{P(C(0))}$$

Given:

$$E\left(w_i|AT(0)\right)=E\left(w_i|D_i=1, x_i \rightarrow 0, z_i=0\right)$$

$$E\left(w_i|AT(0) \text{ or } C(0)\right)=E\left(w_i|D_i=1, x_i \rightarrow 0, z_i=0\right)$$

$$\Rightarrow E\left(w_i|AT(0) \text{ or } C(0)\right)=\frac{E\left(w_i|AT(0)\right)\right) P(AT(0)+E\left(w_i|C(0)\right) P(C(0))}{P(AT(0) \text{ or } C(0))}$$

$$\Rightarrow E\left(w_i|AT(0) \text{ or } C(0)\right)\left(P(AT(0) \text{ or } C(0)\right)=E\left(w_i|AT(0)\right) P(AT(0))+E\left(w_i|C(0)\right) P(C(0))$$

$$\Rightarrow \frac{E\left[w_i\left|AT(0) \text{ or } C(0)\right][P(AT(0) \text{ or } C(0)\right]-E[w_i|AT(0)]P(AT(0))}{P(C(0))}=E\left(w_i|(C(0)\right)$$

[348]:
```
%%latex
(ii) Prove that
$$E\left[w_{i} D_{i} | x_{i} \rightarrow 0, z_{i}=1\right)=E\left(w_{1} | A
 ↪T(0) \text { or } C(0)\right) \times P(A T(0) \text { or }(C(0))$$
```

```
$$$$
Start with the law of iterated expectation:
$$$$
$$\begin{aligned} E\left(w_{i} | z_{i}=1\right) &=E(w_{i}|D_{i}= 1, z_{i}=1)\times P\left(D_{i}=1 | z_{i}=1\right) \\ &=E\left[w_{i} | A T(0) \text { or } C(0)\right] \times P(A T(0) \text { or } C(0)) \end{aligned}$$
```

(ii) Prove that

$$E\left[w_i D_i | x_i \rightarrow 0, z_i = 1\right) = E\left(w_1 | AT(0) \text{ or } C(0)\right) \times P(AT(0) \text{ or } (C(0))$$

Start with the law of iterated expectation:

$$\begin{aligned} E\left(w_i | z_i = 1\right) &= E(w_i | D_i = 1, z_i = 1) \times P\left(D_i = 1 | z_i = 1\right) \\ &= E\left[w_i | AT(0) \text{ or } C(0)\right] \times P(AT(0) \text{ or } C(0)) \end{aligned}$$

[349]:
```
%%latex
(iii) Prove that
$$E\left(w_{i} D_{i} | x_{i} \rightarrow 0, z_{i}=0\right)=E\left(w_{i} | AT(0)\right) \times P(AT(0))$$
$$$$
Start with the law of iterated expectation
$$$$
$\begin{aligned} E\left(w_{i} D_{i} |x_{1} \rightarrow 0, z_{i}=0\right)&=E\left(w_{i} | D_{i}=1, z_{i}=0\right) \times \operatorname{P(D_{i} = 1|z_{i} = 1)} \\ &=E\left(w_{i} | AT(0)\right) \times P(AT(0)) \end{aligned}$
```

(iii) Prove that

$$E\left(w_i D_i | x_i \rightarrow 0, z_i = 0\right) = E\left(w_i | AT(0)\right) \times P(AT(0))$$

Start with the law of iterated expectation

$$\begin{aligned} E\left(w_i D_i | x_1 \rightarrow 0, z_i = 0\right) &= E\left(w_i | D_i = 1, z_i = 0\right) \times \mathrm{P(D_i = 1|z_i = 1)} \\ &= E\left(w_i | AT(0)\right) \times P(AT(0)) \end{aligned}$$

[350]:
```
%%latex

(iv) Prove that the 2SLS estimate of $B$ is an estimate of $E(w_i | C(0))$
$$$$
```

```
$$Given:$$
$D_i=\pi_{0}+\pi_{1} z_{i}+\pi_{2} x_{i}+\pi_{3} x_{i} z_{i}+\epsilon_{i}␣
 ↪\quad$ 1st stage
$$$$
$w_{i} D_i=\beta_{0}+\beta_{1} D_{i}+\beta_{2} x_{i}+\beta_{3} x_{i}␣
 ↪z_{i}+v_{i} \quad$ structural model
$$$$
$w_{i}D_i =\delta_{0}+\delta_{1} z_{i}+\delta_{2} x_{i}+\delta_{3} x_{i}␣
 ↪z_{i}+v_i$ reduced form
$$$$


$$Note:$$
$\hat{\beta}_{1}=\frac{\hat{\delta}_{1}}{\hat{\pi}_{1}}$
$$$$
$$Start:$$
$$E\left(w_{i} | C(0)\right)=\frac{E\left(w_{i} D_{i} | x_{i} \rightarrow 0,␣
 ↪z_{i}=1\right)-E\left(w_{i} D_{i} | x_{i} \rightarrow 0,␣
 ↪z_{i}=0\right)}{P(C(0))}$$
$$\Rightarrow\frac{\hat{\delta}_{1}+\hat{\delta}_{0}-\hat{\delta}_{0}}{E(D_i|x␣
 ↪\rightarrow 0)}$$
$$\Rightarrow\frac{\hat{\delta}_{1}}{\hat{\pi}_{0}+\hat{\pi}_{1}-\hat{\pi}_{0}}$$
$$\Rightarrow\frac{\hat{\delta}_{1}}{\pi_{1}}$$
$$\Rightarrow\hat{\beta}_{1}$$
```

(iv) Prove that the 2SLS estimate of B is an estimate of $E(w_i|C(0))$

Given :

$D_i = \pi_0 + \pi_1 z_i + \pi_2 x_i + \pi_3 x_i z_i + \epsilon_i$ 1st stage

$w_i D_i = \beta_0 + \beta_1 D_i + \beta_2 x_i + \beta_3 x_i z_i + v_i$ structural model

$w_i D_i = \delta_0 + \delta_1 z_i + \delta_2 x_i + \delta_3 x_i z_i + v_i$ reduced form

Note :

$\hat{\beta}_1 = \frac{\hat{\delta}_1}{\hat{\pi}_1}$

$$Start:$$

$$E\left(w_i|C(0)\right) = \frac{E\left(w_i D_i|x_i \to 0, z_i = 1\right) - E\left(w_i D_i|x_i \to 0, z_i = 0\right)}{P(C(0))}$$

$$\Rightarrow \frac{\hat{\delta}_1 + \hat{\delta}_0 - \hat{\delta}_0}{E(D_i|x \to 0)}$$

$$\Rightarrow \frac{\hat{\delta}_1}{\hat{\pi}_0 + \hat{\pi}_1 - \hat{\pi}_0}$$

$$\Rightarrow \frac{\hat{\delta}_1}{\pi_1}$$

$$\Rightarrow \hat{\beta}_1$$

8 2.2 Estimating Characteristics of Compliers

8.1 Table 5 and Figures 3, 4

Goal: show the relationship between PSU and the probability of entering college

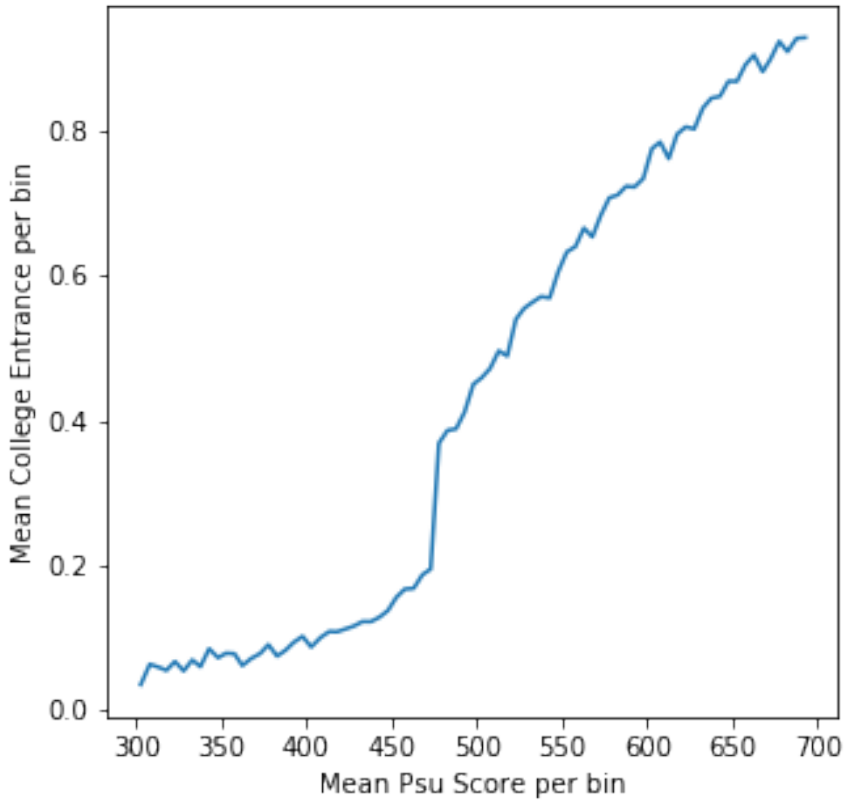
[351]:
```
#adding bins to dataset
binsarray = np.arange(min(rd['psu']), max(rd['psu']), 5)
rd['bins'] = pd.cut(x=rd['psu'], bins = binsarray)
```

8.2 Figure 3

[352]:
```
psugraphdata = rd.groupby('bins').agg(np.mean)
psugraphdata
plt.plot(psugraphdata['psu'], psugraphdata['entercollege'])
plt.xlabel('Mean Psu Score per bin')
plt.ylabel('Mean College Entrance per bin')
plt.title('Figure 3: Psu versus College Entrance Rate by 5 point Psu-score␣
 ↪bins')
```

[352]:
```
Text(0.5, 1.0, 'Figure 3: Psu versus College Entrance Rate by 5 point Psu-score
bins')
```

Figure 3: Psu versus College Entrance Rate by 5 point Psu-score bins



8.3 Table 5

Goal: estimate local linear first stage models for the probability of attending college - constant - zi = indicator if PSU >= 475 - xi = PSU - 475 - zi*xi - Dependent variable = Di = entercollege

[353]:
```
#creating new variables
table5data = rd
indicator475 = []
for value in table5data['psu']:
    if value >= 475:
        indicator475.append(0)
    else:
        indicator475.append(1)
table5data['psu>=475'] = indicator475
table5data['psu-475'] = table5data['psu']-475
table5data['interaction'] = table5data['psu>=475']*table5data['psu-475']

#selecting relevant columns
table5subsetdata = table5data[['psu>=475', 'psu-475', 'interaction']]
```

```
table5subsetdata = sm.add_constant(table5subsetdata)
dvar = rd[['entercollege']]

#fitting model
table5model = sm.OLS(dvar, table5subsetdata).fit()
```

[354]:
```
#results for bandwidth = 25
band25 = table5data[table5data['psu'] <= 500]
band25 = band25[band25['psu'] >= 450]
band25 = band25[['psu>=475', 'psu-475', 'interaction']]
band25 = sm.add_constant(band25)
dvar25 = rd[rd['psu'] <= 500]
dvar25 = dvar25[dvar25['psu'] >= 450][['entercollege']]


#fitting model
band25model = sm.OLS(dvar25, band25).fit()
```

[355]:
```
#results for bandwidth = 50
band50 = table5data[table5data['psu'] <= 525]
band50 = band50[band50['psu'] >= 425]
band50 = band50[['psu>=475', 'psu-475', 'interaction']]
band50 = sm.add_constant(band50)
dvar50 = rd[rd['psu'] <= 525]
dvar50 = dvar50[dvar50['psu'] >= 425][['entercollege']]


#fitting model
band50model = sm.OLS(dvar50, band50).fit()
```

[356]:
```
#results for bandwidth = 75
band75 = table5data[table5data['psu'] <= 550]
band75 = band75[band75['psu'] >= 400]
band75 = band75[['psu>=475', 'psu-475', 'interaction']]
band75 = sm.add_constant(band75)
dvar75 = rd[rd['psu'] <= 550]
dvar75 = dvar75[dvar75['psu'] >= 400][['entercollege']]


#fitting model
band75model = sm.OLS(dvar75, band75).fit()
```

[357]:
```
#results for bandwidth = 100
band100 = table5data[table5data['psu'] <= 575]
band100 = band100[band100['psu'] >= 375]
band100 = band100[['psu>=475', 'psu-475', 'interaction']]
band100 = sm.add_constant(band100)
```

```
dvar100 = rd[rd['psu'] <= 575]
dvar100 = dvar100[dvar100['psu'] >= 375][['entercollege']]


#fitting model
band100model = sm.OLS(dvar100, band100).fit()
```

[358]:
```
#Table representation of results
q5results = Stargazer([band25model, band50model, band75model, band100model])
q5results.custom_columns(["Bandwidth = 25", "Bandwidth = 50", "Bandwidth = 75",␣
↪"Bandwidth = 100"], [1,1,1,1])
q5results.title("Table 5: Examining College Entrance Characteristics")
HTML(q5results.render_html())
```

[358]:
```
<IPython.core.display.HTML object>
```

8.4 Figure 4

Estimate model 4 using a range of bandwidths

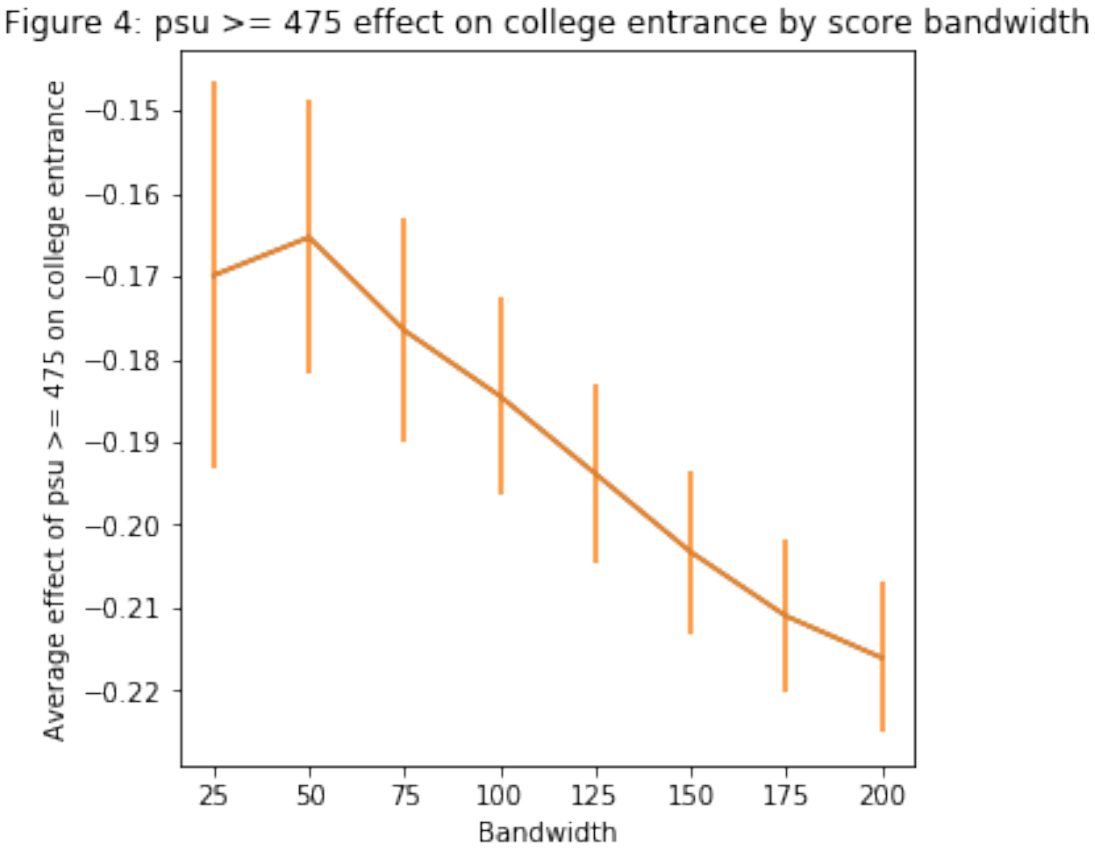
Plot the estimates of pi1 and the +- 2 standard error confidence bands for each estimate against the bandwidth choice

[359]:
```
x = np.arange(25, 225, 25)
y_vals = []
se_vals = []
for band in x:
    upper = 475 + band
    lower = 475 - band
    band = table5data[table5data['psu'] <= upper]
    band = band[band['psu'] >= lower]
    band = band[['psu>=475', 'psu-475', 'interaction']]
    band = sm.add_constant(band)
    dvar_gen = rd[rd['psu'] <= upper]
    dvar_gen = dvar_gen[dvar_gen['psu'] >= lower][['entercollege']]
    #fitting model

    bandmodel = sm.OLS(dvar_gen, band).fit()
    y_vals.append(bandmodel.params[1])
    se_vals.append(bandmodel.bse[1])

se_vals = [val*2 for val in se_vals]
```

[360]:
```
plt.plot(x, y_vals)
plt.errorbar(
x = x,
y = y_vals,
```

```
yerr = se_vals)
plt.xlabel("Bandwidth")
plt.ylabel("Average effect of psu >= 475 on college entrance")
plt.title("Figure 4: psu >= 475 effect on college entrance by score bandwidth")
```

[360]: `Text(0.5, 1.0, 'Figure 4: psu >= 475 effect on college entrance by score bandwidth')`



Narrative: Using Figures 3 and 4 and the estimates in Table 5, discuss what you think is a reasonable bandwidth choice. Discuss how a bigger bandwidth may give a more biased but more precise estimate of `pi_1`

MY ANSWER: For selection of bandwidth, a high value leads to high bias and low variance, while a low value leads to low bias but high variance. We see that reflected in Figure 4. According to Figure 3, we see a jump around psu = 475, which is to be expected, as 475 is the cutoff for loan eligibility. The idea with regression discontinuity is to zoom into the data close enough that there is no substantial confounding difference between the group that gets just above psu = 475 and below psu = 475. With a bandwidth of 50, we can achieve slightly lower variance than we would when using bandwidth = 25, but retain the "zoom in" qualities that remaining close to 475 gives us. Seen in Table 5 as well as Figure 4, the bigger the bandwidth, the lower the standard error. While adding a wider range of datapoints to our results might bias answers if there was an omitted variable which

biased the data, we achieve a lower standard error precisely because there are more data points. For instance, if bandwidth = 200, we would capture most of the datapoints available and achieve a lower standard error. However, there may also be substantial differences between students which score significantly above the cutoff threshold and students who score significantly below the cutoff threshold. Students who score much higher than the cutoff score may perhaps be wealthier and able to afford exam preparation courses and tutoring. They may have more time available to study if they come from smaller, wealthier families, whereas students who score substantially lower than 475 might come from low income backgrounds. These students might not be able to afford exam tutoring or be able to spend as much time studying if they work jobs to support their families. These biasing variables would otherwise be assumed away with a smaller bandwidth.

8.5 Table 6

[361]:
```
#initialize with dataset and empty dataframe
def choicedataframe(data, frame):
    quint1 = np.mean([data['quintile'] == 1])
    quint2 = np.mean([data['quintile'] == 2])
    quint3 = np.mean([data['quintile'] == 3])
    quint4 = np.mean([data['quintile'] == 4])
    sharefemale = np.mean([data['female'] == 1])
    sharegpa6070 = np.mean([(data['gpa'] >= 60) & (data['gpa'] <= 70)])
    sharegpa5060 = np.mean([(data['gpa'] >= 50) & (data['gpa'] <= 60)])
    sharegpa50less = np.mean([data['gpa'] <50])
    motheredhsabove = np.mean([data['himom'] == 1])
    fatheredhsabove = np.mean([data['hidad'] == 1])

    frame['quint1'] = [quint1]
    frame['quint2'] = quint2
    frame['quint3'] = quint3
    frame['quint4'] = quint4
    frame['sharefemale'] = sharefemale
    frame['gpa between 60 and 70'] = sharegpa6070
    frame['gpa between 50 and 60'] = sharegpa5060
    frame['gpa less than 50'] = sharegpa50less
    frame['mother with education >= HS'] = motheredhsabove
    frame['father with education >= HS'] = fatheredhsabove

    return frame
```

[362]:
```
#bandwidth choice = 50
bandwidthchoice = pd.DataFrame()

table6data = table5data[table5data['psu'] <= 525]
table6data = table6data[table6data['psu'] >= 425]
```

```
#run function to generate dataframe with mean values according to bandwidth =␣
↪50 restriction
choicedataframe(table6data, bandwidthchoice)
bandwidthchoice = bandwidthchoice.rename(index = {0: "bandwidth sample"})
```

[363]:
```
#entire dataset
entireset = pd.DataFrame()
choicedataframe(table5data, entireset)
entireset = entireset.rename(index = {0: "entire sample"})
```

[364]:
```
#complier dataset
complierdata = table5data
complierdata = complierdata[(complierdata['psu'] >= 475) &␣
↪(complierdata['quintile'] != 4) & (complierdata['entercollege'] ==1)]
compliers = pd.DataFrame()
choicedataframe(complierdata, compliers)
compliers = compliers.rename(index = {0: 'compliers only'})
```

[365]:
```
#ratio of the mean of each characteristic for the compliers versus the entire␣
↪sample
compliersvsentire = pd.DataFrame()
intermediate = []
for value in np.arange(10):
    intermediate.append(compliers.iloc[0][value]/entireset.iloc[0][value])
compliersvsentire['quint1'] = [intermediate[0]]
compliersvsentire['quint2'] = intermediate[1]
compliersvsentire['quint3'] = intermediate[2]
compliersvsentire['quint4'] = intermediate[3]
compliersvsentire['sharefemale'] = intermediate[4]
compliersvsentire['gpa between 60 and 70'] = intermediate[5]
compliersvsentire['gpa between 50 and 60'] = intermediate[6]
compliersvsentire['gpa less than 50'] = intermediate[7]
compliersvsentire['mother with education >= HS'] = intermediate[8]
compliersvsentire['father with education >= HS'] = intermediate[9]
compliersvsentire = compliersvsentire.rename(index = {0: "Ratio of mean of␣
↪compliers vs entire sample"})
```

[366]:
```
table6 = pd.concat([bandwidthchoice, entireset, compliers, compliersvsentire])
table6.index = ["bandwidth choice", "entire sample", "compliers only", "ratio␣
↪of compliers vs entire sample"]
table6.style.set_caption("Table 6: Mean Characteristics of PSU Test Takers ")
```

[366]:
```
<pandas.io.formats.style.Styler at 0x7f1b5c578080>
```

Narrative: Discuss the claim that the loan program extends college access to more economically disadvantaged students. What else can you say about the compliers?

MY ANSWER: The claim that the loan program extends college access to more economically disadvantaged students seems accurate according to Table 6. If we look at the characteristics for only compliers, we see that students in quintile 1 (the lowest income quintile) make up the largest proportion of students. The next largest proportion of students come from quintile 2, and then quintile 3, respectively. None come from quintile 4, as expected because the loan program is only eligible for those with income under quintile 4's threshold.

Compliers are also comprised primarily of high-gpa students. While the share of students in the unfiltered dataset who hold a gpa between 60 and 70 is 0.3189, the share of students in the complier group who fall in that same category is 0.5113. These are extremely high performing students.

Compared to the unfiltered dataset, mothers and fathers of the compliers have higher rates of high school graduation and higher education.

[ ]:

[367]:
```
from IPython.display import display
from IPython.display import HTML
import IPython.core.display as di # Example: di.display_html('<h3>%s:</h3>' %␣
↪str, raw=True)

# This line will hide code by default when the notebook is exported as HTML
di.display_html('<script>jQuery(function() {if (jQuery("body.notebook_app").
↪length == 0) { jQuery(".input_area").toggle(); jQuery(".prompt").toggle();
↪}});</script>', raw=True)

# This line will add a button to toggle visibility of code blocks, for use with␣
↪the HTML export version
di.display_html('''<button onclick="jQuery('.input_area').toggle(); jQuery('.
↪prompt').toggle();">Toggle code</button>''', raw=True)

#Source: https://stackoverflow.com/questions/27934885/
↪how-to-hide-code-from-cells-in-ipython-notebook-visualized-with-nbviewer
```

[ ]:

[ ]: