Econ C142 Final Project

Robbie Netzke

May 14, 2021

1 Part 1

1.1 Analytical Exercise

The casual model is defined as:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 D_i + u_i$$

with associated first stage model:

$$x_i = \pi_0 + \pi_1 z_i + \pi_2 D_i + \eta_i$$

and associated reduced form model:

$$y_i = \delta_0 + \delta_1 z_i + \delta_2 D_i + \nu_i$$

1.1.1 Proof of A

Using the above equations, by substituting the expression for x_i in the first stage model into the casual model, then:

$$\begin{aligned} y_i &= \beta_0 + \beta_1(\pi_0 + \pi_1 z_i + \pi_2 D_i + \eta_i) + \beta_2 D_i + u_i \\ &= (\beta_0 + \beta_1 \pi_0) + (\beta_1 \pi_1) z_i + (\beta_2 + \beta_1 \pi_2) D_i + (\beta_1 \eta_i + u_i) \end{aligned}$$

Using the third equation above we can now form the equations:

$$\beta_0 + \beta_1 \pi_0 = \delta_0$$

$$\beta_1 \pi_1 = \delta_1$$

$$\beta_2 + \beta_1 \pi_2 = \delta_2$$

If we let $\pi_1 = 1$, then $\beta_1 = \delta_1$, which completes the exercise.

1.1.2 Proof of B

Considering the models defined above, it is required to show that the model, fit only to observations where $D_i = 1$:

$$y_i = \delta_0' + \delta_1' z_i + \nu_i$$

Satisfies $\delta_1' = \delta_1$, where δ_1 is the coefficient on z_i in the reduced form model.

Define $N_1 = \sum_i^N D_i$. Let $\bar{z}_0, \bar{z}_1$ be defined as they are in the prompt.

Consider the model:

$$z_i = \lambda_0 + \lambda_1 D_i + \xi_i$$

From the first order conditions, $\lambda_0 = \bar{z}_0$ and $\lambda_1 = \bar{z}_1 - \bar{z}_0$. However $z_i = 0$ if $D_i = 0$, which implies $\bar{z}_0 = 0$, since this defines the mean of z_i when $D_i = 0$.

Then the model above can be reduced to:

$$z_i = \bar{z}_1 D_i + \xi_i$$

Implying:

$$\xi_i = z_i - \bar{z}_1 D_i$$

Referring back to the reduced form model:

$$y_i = \delta_0 + \delta_1 z_i + \delta_2 D_i + \nu_i$$

From F-W,

$$\hat{\delta_1} = (\sum_i^N \hat{\xi_i}^2)^{-1} \sum_i^N \hat{\xi_i} y_i$$

Substituting the expression for ξ_i into this equation:

$$\hat{\delta_1} = (\sum_i^N (z_i - \bar{z}_1 D_i)^2)^{-1} \sum_i^N (z_i - \bar{z}_1 D_i) y_i$$

Now, because $z_i = 0$ if $D_i = 0$,

If $D_i = 0$:

$$z_i - \bar{z}_1 D_i = 0$$

and if $D_i = 1$:

$$z_i - \bar{z}_1 D_i = z_i - \bar{z}_1$$

Therefore:

$$\delta_1 = (\sum_i^N (z_i - \bar{z}_1 D_i)^2)^{-1} \sum_i^N (z_i - \bar{z}_1 D_i) y_i = (\sum_i^{N_1} (z_i - \bar{z}_1)^2)^{-1} \sum_i^{N_1} (z_i - \bar{z}_1) y_i = \delta_1'$$

Completing the proof.

1.2 Examining the Data

The data analysis begins by comparing the means of wages and employment in New Jersey and Pennsylvania before and after a minimum wage increase in New Jersey.

Variable	New Jersey	Pennsylvania	Difference
WAGE_ST	4.613	4.654	-0.041
WAGE_ST2	5.082	4.619	0.463
PCHWAGE	0.107	-0.004	0.111
EMPTOT	20.678	23.705	-3.026
EMPTOT2	21.0763	21.826	-0.749
PCHEMP	0.022	-0.033	0.055

Table 1: NJ and PA Characteristics

1.2.1 Narrative 1: Table of Means

A. Examining the results of Table 1 below, the difference in differences alludes to an increase in wages of around 11 percent in New Jersey relative to Pennsylvania. Certainly, it appears wages have increased in New Jersey and stagnated in Pennsylvania, and, on observation of the starting wages in wave 1 and starting wages in wave 2, New Jersey starting wages have increased to roughly the new minimum wage while Pennsylvania wages have actually decreased. Similarly, there is around a 5 percent increase in arc percent employment, so there is an increase in employment relative to Pennsylvania as well.

1.2.2 Preliminary Regression Analysis

Note: all coefficients and confidence intervals can be verified in the appendix (pages 3, 4, 5) and the calculations are omitted for brevity.

B. Beginning with the model:

$$PCHWAGE_i = \gamma_0 + \gamma_1 NJ_i + \epsilon_i$$

We have an estimate of $\gamma_1 = 0.111$, which is precisely the same estimate as the entry in Table 1 row 3, column 3. The confidence interval for this estimate is: [0.090, 0.133].

C. Proceeding to the next model:

$$PCHEMP_i = \rho_0 + \rho_1 NJ_i + \phi_i$$

We have an estimate of $\rho_1 = 0.055$, which is precisely the same estimate as the entry in Table 1 row 6, column 3. The confidence interval for this estimate is: [-0.040, 0.150]

D. Lastly, the casual model, to be estimated by IV:

$$PCHEMP_i = \beta_0 + \beta_1 PCHWAGE_i + u_i$$

We obtain a coefficient $\hat{\beta}_1 = 0.493$, which is equal to $\hat{\rho_1}/\hat{\gamma_1} = 0.055/0.111 = 0.493$

1.2.3 Narrative 2: Examining the Effect of Minimum Wage Changes

The effect of the minimum wage change has increased both starting wages and employment in New Jersey relative to Pennsylvania: there are positive estimated coefficients on both ρ_1 and γ_1, the coefficients on NJ. The model in part B identifies the first stage model, the model in part C identifies the reduced form model, and the model in part D identifies the casual model. From the many derivations of IV, it is known that the ratio of ρ_1 and γ_1 will estimate $\hat{\beta_1}$ when using NJ as an instrumental variable. It is important to note, when using NJ as an instrumental variable, it is assumed that NJ only effects the changes in employment through the changes in wages. We can reason that this may not be true, and that working in New Jersey (or not) effects both the changes in employment and wages directly. Therefore, the exclusion restriction, which is an assumption of IV, would be violated in this case.

Table 2: GAP as an Instrument

	OLS	First Stage	Reduced Form	IV
	(1)	(2)	(3)	(4)
const	-0.024	-0.002	-0.032	-0.031
	(0.026)	(0.003)	(0.028)	(0.042)
pchwage	0.418**			0.493
	(0.208)			(0.431)
gap		1.041***	0.514**	
		(0.031)	(0.247)	
RMSE	6.572	0.813	6.569	6.573
R^2	0.011	0.768	0.012	0.011
Residual Std. Error	0.352(df = 349)	0.044(df = 349)	0.352(df = 349)	0.352(df = 349)
Note:				*p<0.1; **p<0.05; ***p<0.01

1.2.4 Narrative 3: Examination of Table 2

The OLS model and the IV model do not differ in coefficients by more than two standard errors, but the models are slightly different. The IV model predicts a higher increase in employment per each increase in $PCHWAGE$ compared to the OLS model, after controlling for the minimum wage changes. The first stage model has the largest R^2 out of any of the models, and π_1 is very close to 1. I think the first stage model does indeed seem to provide a good description of wage changes, as I do not expect starting wages to typically go above minimum wage: this is reflected in the coefficient on GAP that is close 1. Keeping in mind that when $\pi_1 = 1$, then $\delta_1 = \beta_1$, we see that the reduced form coefficient $\hat{\delta_1}$ is close to the IV estimate for $\hat{\beta_1}$. The reduced form estimate is a less precise estimate of the casual effect β_1, however, because $\hat{\pi_1}$ is not quite 1. Finally, we can verify directly that $\hat{\beta_1} = \hat{\delta_1}/\hat{\pi_1} = 0.514/1.041 = 0.493$. Analysis continues on the following pages.

Table 3: Adding NJ_i to the Model

	OLS	First Stage	Reduced Form	IV
	(1)	(2)	(3)	(4)
const	-0.031	-0.004	-0.033	-0.031
	(0.043)	(0.005)	(0.043)	(0.043)
pchwage	0.396*			0.494*
	(0.238)			(0.285)
gap		1.031***	0.510*	
		(0.036)	(0.294)	
nj	0.011	0.004	0.002	-0.000
	(0.055)	(0.007)	(0.057)	(0.058)
RMSE	6.571	0.8124	6.569	6.573
R^2	0.012	0.769	0.012	0.011
Residual Std. Error	0.352(df = 348)	0.044(df = 348)	0.352(df = 348)	0.352(df = 348)
Note:				*p<0.1; **p<0.05; ***p<0.01

1.2.5 Narrative 4: Examination of Table 3

The models in this table look quite similar. In fact, the IV model is virtually the same, only with tighter confidence intervals around each coefficient when adding NJ. The first stage and reduced form models are quite similar as well, with the order of magnitude in coefficient changes on GAP being quite small. The model that has changed most significantly is the OLS. After controlling for the impacts of minimum wage, I would conclude that it is definitely OK to assume New Jersey and Pennsylvania are similar.

Table 4: Only $NJ_i = 1$ Observations

	OLS	First Stage	Reduced Form	IV
	(1)	(2)	(3)	(4)
const	-0.043	-0.001	-0.031	-0.031
	(0.035)	(0.003)	(0.036)	(0.036)
pchwage	0.607**			0.494*
	(0.263)			(0.278)
gap		1.031***	0.510*	
		(0.022)	(0.288)	
RMSE	5.784	0.434	5.806	5.786
R^2	0.019	0.890	0.011	0.018
Residual Std. Error	0.344(df = 283)	0.026(df = 283)	0.345(df = 283)	0.344(df = 283)
Note:				*p<0.1; **p<0.05; ***p<0.01

1.2.6 Narrative 5: Examining Why the Key Estimates are the Same in Table 4

In the proof of 1.1 B, it was shown that the reduced form model of $PCHEMP$ onto GAP and NJ could be estimated by simply using only the observations such that $NJ = 1$. We observe that when $NJ_i = 0$, then $GAP = 0$. Likewise, GAP is nonzero when $NJ_i = 1$. In the proof, from F-W, this resulted in observations where $NJ_i = 0$ not contributing to the coefficient on GAP in the reduced form. My best intuition behind this is, since we know the outcome of D_i through GAP_i, we only need to use observations where $D_i = 1$. Because the revised reduced form and first stage models in this question have coefficients on GAP that are equivalent to those in table 3, the results are the same as table 3, and the resulting IV estimate of β_1 remains the same as well.

Table 5: Regional Controls

	OLS	First Stage	Reduced Form	IV
	(1)	(2)	(3)	(4)
const	-0.110*	0.005	-0.108	-0.110*
	(0.067)	(0.008)	(0.067)	(0.067)
pchwage	0.402*			0.489*
	(0.239)			(0.286)
gap		1.031***	0.504*	
		(0.036)	(0.295)	
southj	0.105	-0.006	0.093	0.096
	(0.083)	(0.011)	(0.085)	(0.084)
centralj	0.049	-0.005	0.037	0.040
	(0.086)	(0.011)	(0.088)	(0.088)
northj	0.104	-0.004	0.094	0.095
	(0.076)	(0.010)	(0.078)	(0.078)
shore	-0.049	-0.006	-0.051	-0.048
	(0.069)	(0.009)	(0.069)	(0.069)
pa2	0.137	-0.016	0.130	0.138
	(0.088)	(0.011)	(0.088)	(0.088)
RMSE	6.532	0.808	6.531	6.533
R^2	0.023	0.771	0.024	0.023
Residual Std. Error	0.352(df = 344)	0.044(df = 344)	0.352(df = 344)	0.352(df = 344)
Note:				*p<0.1; **p<0.05; ***p<0.01

1.2.7 Narrative 6: Examination of Table 5

Similar to using NJ as a control variable, the regional dummies do not alter the first stage, reduced form, and IV estimates much. The key estimates remain relatively the same, so we can conclude that it is a reasonable assumption to ignore the regional demand shocks once the impacts of minimum wage are accounted for.

Table 6: All Controls Added to the Model

	OLS	First Stage	Reduced Form	IV
	(1)	(2)	(3)	(4)
const	-1.272**	-0.002	-1.264**	-0.526
	(0.506)	(0.060)	(0.506)	(1887702.369)
pchwage	0.212			0.525
	(0.331)			(0.469)
gap		0.951***	0.499	
		(0.052)	(0.444)	
RMSE	6.325	0.745	6.317	6.334
R^2	0.084	0.805	0.087	0.082
Residual Std. Error	0.349(df = 328)	0.041(df = 328)	0.349(df = 328)	0.350(df = 327)
Note:				*p<0.1; **p<0.05; ***p<0.01

1.2.8 Narrative 7: Adding All Controls

Indeed, adding all the controls effects the regression. Interestingly, by comparing to table 4, the RMSE of the models with all controls are higher than the simple model fitting only to observations in New

Jersey. The coefficient on GAP, π_1, has deviated from the values in tables 3, 4, and 5. Similarly, the reduced form coefficient GAP has also changed slightly, which has caused the coefficient β_1 to increase in the IV estimate. When using all the controls, the model estimates π_1 to be less than 1. As the variables are defined, this would suggest that the starting wages in New Jersey have not reached the new minimum wage. Although I am not well versed on the following concepts, there is a potential that, when adding many controls, we introduce both colliders and confounders into the regression. Furthermore, outliers have a large impact on regression outcomes when using L_2 as the loss function, so we must be wary of the introduction of so many controls without accounting for these outliers. Perhaps one of these concepts could explain why our π_1 has deviated from 1.

An extra comment on Table 7: After the DoubleML procedure, the coefficients on GAP in the reduced form and first stage models as well as the coefficient on $PCHWAGE$ seem to have slightly converged towards the coefficients seen in Tables 3, 4, and 5. Still, the coefficients resemble those in Table 6. The DoubleML model has done a reasonable job of dealing with so many co-variates, but I am still surprised by the similarities between this model and the simple model using all co-variates. Of course, the results of the DoubleML change each time it is run due to randomness of the design.

Table 7:

	OLS	First Stage	Reduced Form
	(1)	(2)	(3)
pchwage	0.258		
	(0.295)		
gap		0.962	0.510
		(0.073)	(0.408)

2 Part 2

Part 2 begins with a visual exploratory data analysis of the relationship between the running variable, age, and health outcomes such as health insurance coverage and doctor visits. Figures 2.1-2.4 follow

2.1 Figures

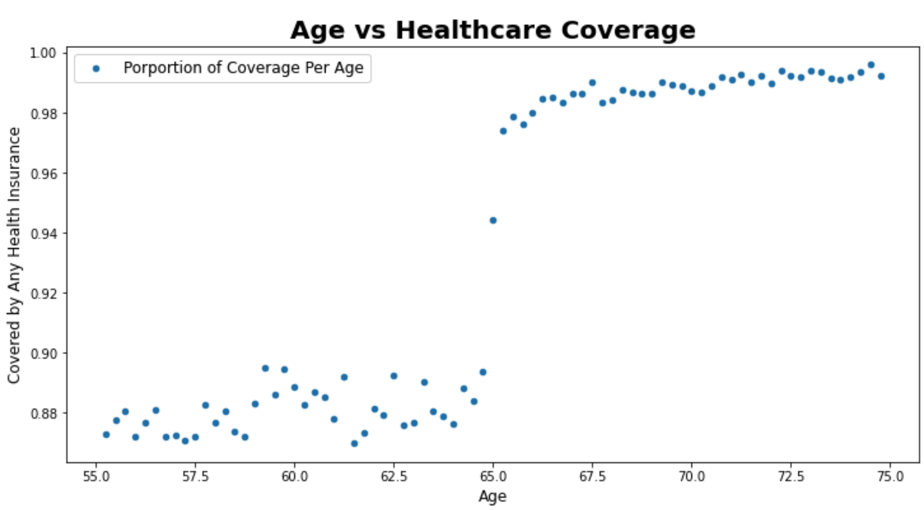


Figure 1: 2.1

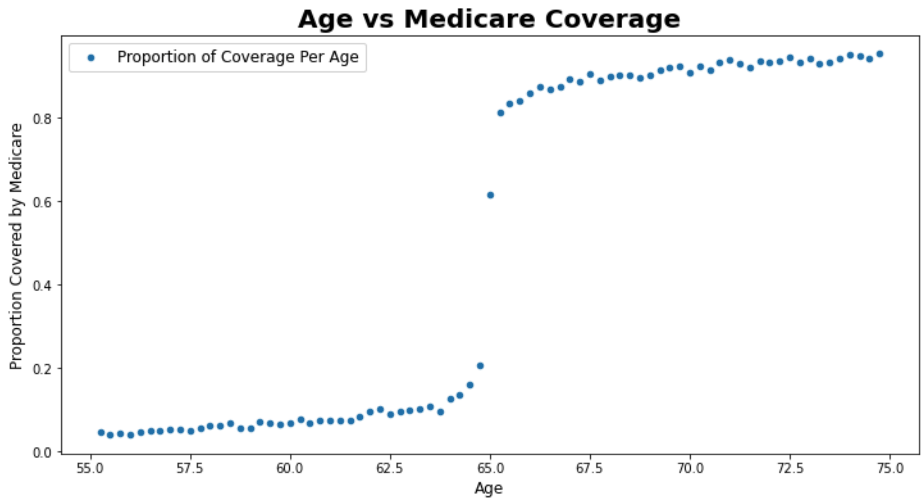


Figure 2: 2.2

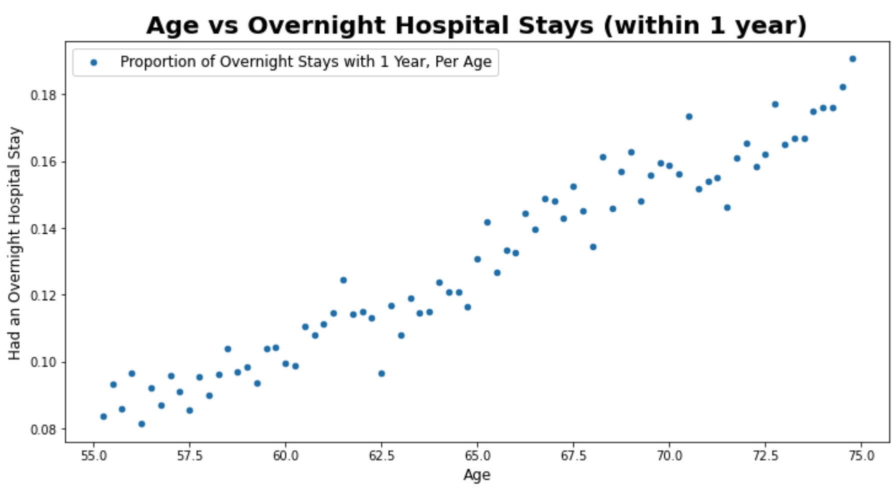


Figure 3: 2.3

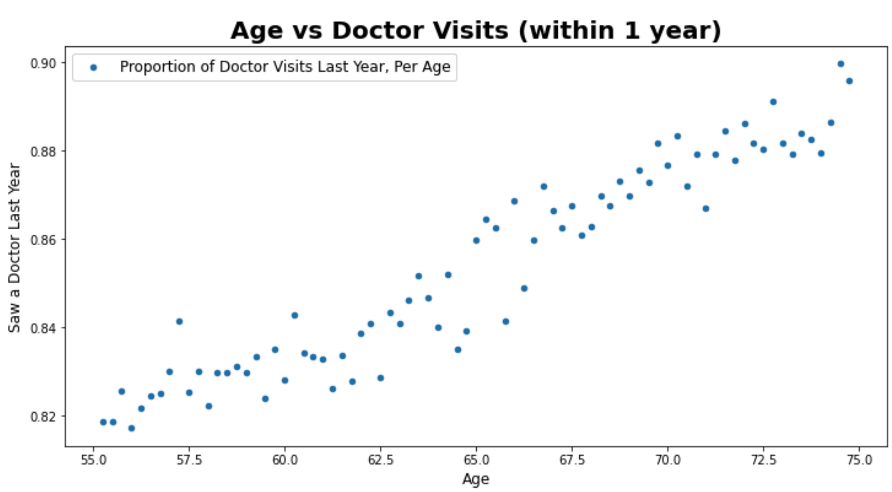


Figure 4: 2.4

2.1.1 Narrative 8: Comments on the Figures

Once an individual reaches age 65, the probability they are covered by any health insurance converges towards 1. Similarly, the probability of having Medicare jumps from nearly 20 percent to approximately 80 percent after an individual turns 65. We can see in the graphs, however, that these transitions are not smooth. There is not a clear, sharp jump for the proportion of covered individuals. Rather, in both the graphs of Medicare and any health care coverage, there is a point that straddles the probabilities of coverage well before and well after 65. This is likely due to a sign-up or transition period, where 65 year old individuals must still enroll in Medicare, and it does not take effect immediately. The probability of having seen a doctor in the past year and the probability of having an overnight hospital stay seem to be increasing functions of age with no significant jumps as far as I can tell. The conditions I expect to see a jump at 65 in visiting the doctor or staying at a hospital are when most individuals will never see a doctor or never stay in a hospital, unless they are covered by health insurance. If this relationship is strict, then it could be possible to see jumps in doctor visits and hospital stays at 65, where more individuals get health insurance coverage.

2.2 Regression in Part i

Table 8: Local. Linear Model for Coverage

	Dependent variable: covered
	(1)
const	0.886***
	(0.002)
r	0.001***
	(0.000)
r_z	0.001**
	(0.000)
z	0.091***
	(0.003)
Observations	153,782
R^2	0.043
Adjusted R^2	0.043
Residual Std. Error	0.252(df = 153778)
Note:	*p<0.1; **p<0.05; ***p<0.01

2.3 Part ii and iii

2.4 Figure 2.5 and Regression Results After Removing Age 65

2.4.1 Narrative 9: Comments on Robustness and Removing $age4_i = 65$

As seen on figure 2.5 on the following page, the estimate of the effect of reaching age 65 on the probability of having any form of health insurance appears to be robust to the choice of bandwidth. The 95 percent confidence intervals for π_1 become thinner as the bandwidth increases, but the estimate of π_1 stays within a reasonable range for each choice of bandwidth. The estimate of the increase in probability of having health insurance once reaching age 65 is roughly 9 percent in the first model, but when we exclude people at age 65, the estimate increases to around 9.5 percent, with the same standard error as the previous model. As was discussed before, Medicare likely uses a sign-up or enrollment period such that some 65 year old individuals are not yet covered. By excluding these individuals that have delayed the sign-up process, the regression decomposition provides a better representation of the jump in health care coverage before and after 65.

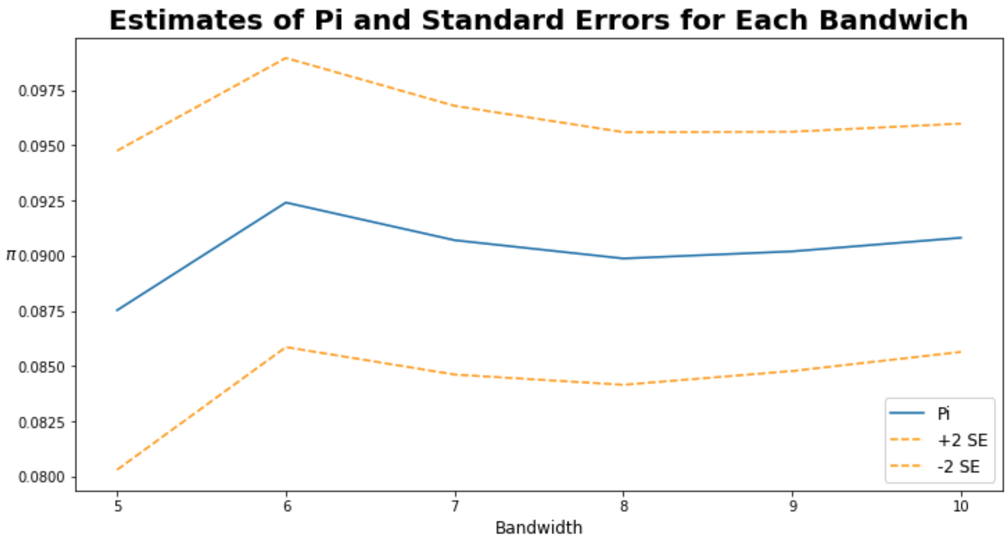


Figure 5: (2.5)

Table 9: Age 65 Removed from Observations

	Dependent variable: covered
	(1)
const	0.886***
	(0.002)
r	0.001***
	(0.000)
r_z	0.000
	(0.000)
z	0.094***
	(0.003)
Observations	151,842
R^2	0.043
Adjusted R^2	0.043
Residual Std. Error	0.252(df = 151838)
F Statistic	2281.952*** (df = 3.0; 151838.0)
Note:	*p<0.1; **p<0.05; ***p<0.01

2.5 Regression in Part C: The Local Quadratic Model

The results of the local quadratic estimation are shown below. The estimates of π_1 are larger in Figure 2.5, the local quadratic model predicts a lower jump in probability of having medicare coverage at age 65. The quadratic model is arguably more precise, as it captures nonlinear trends in probability with respect to the running variable, but considering that both the linear and quadratic model use the same bandwidth, I believe the local linear model is sufficient in estimating the jump and that the local quadratic has a danger of overfitting to noise in the outcome variable.

Table 10: The Local Quadratic Model

	Dependent variable: covered
	(1)
const	0.883***
	(0.003)
r	-0.000
	(0.001)
r^2	-0.000
	(0.000)
r_z	0.007***
	(0.002)
w^2	-0.000
	(0.000)
z	0.087***
	(0.004)
Observations	153,782
R^2	0.043
Residual Std. Error	0.252(df = 153776)
Note:	*p<0.1; **p<0.05; ***p<0.01

2.6 Regressions in Part D: Checking The Validity of the RD

Table 11: Validity of the RD (Table 2.1)

	college	wnh	bnh	hispanic	minority
	(1)	(2)	(3)	(4)	(5)
const	0.166***	0.749***	0.113***	0.108***	0.221***
	(0.003)	(0.003)	(0.002)	(0.002)	(0.003)
r	-0.007***	0.003***	-0.001*	-0.001***	-0.002***
	(0.000)	(0.001)	(0.000)	(0.000)	(0.001)
r_z	0.003***	0.002***	-0.001	-0.002***	-0.002***
	(0.001)	(0.001)	(0.001)	(0.001)	(0.001)
z	0.002	-0.002	0.002	0.000	0.002
	(0.004)	(0.004)	(0.003)	(0.003)	(0.004)
Observations	153,782	153,782	153,782	153,782	153,782
R^2	0.006	0.002	0.000	0.002	0.002
Adjusted R^2	0.006	0.002	0.000	0.002	0.002
Note:					*p<0.1; **p<0.05; ***p<0.01

2.6.1 Narrative 10: Validity of the RD

Observing Table 11 above, there does not appear to be any discontinuities in these variables at age 65. All coefficients are near zero. This should be expected, as the exogenous characteristics of the population should not change with age. This supports the validity of the RD model, as it strengthens the argument that increases in healthcare coverage can be quantified by a local linear or quadratic model, and that the jump in probability does not change with the exogenous characteristics of the population.

Table 12: (AKA Table 2.2)

	Linear: sawdr	Quadratic: sawdr	Linear: inhosp	Quadratic: inhosp
	(1)	(2)	(3)	(4)
covered	0.136^{***}	0.141^{***}	0.126^{***}	0.145^{**}
	(0.033)	(0.052)	(0.037)	(0.059)
R^2	0.022	0.022	0.005	0.002
Adjusted R^2	0.022	0.022	0.005	0.002
Note:				*p<0.1; **p<0.05; ***p<0.01

2.6.2 Narrative 11: Examining Table 12 (2.2 in the Prompt)

In each one of these models, the probability of seeing a doctor in the last year or having an overnight stay in a hospital increases when individuals have health insurance. Insurance coverage appears to increase doctor visits by around 13-14 percent and overnight hospital stays by around 12-14.5 percent. These results appear to be robust to fitting a local or quadratic model, as the estimates of β_1 for each model are nearly equal for each outcome variable. Although, in the case of hospital visits, the quadratic model estimates a more significant jump in overnight stays with insurance when compared to the linear model. This relationship might be worth exploring.

2.7 Open Ended Question

As explored previously, the transition into being covered by Medicare may not immediately occur for an individual after turning 65. I have decided to omit individuals that have $age4 = 65$. With this omission, I anticipate the probability of seeing a doctor or staying overnight in a hospital will increase, as age will better explain D_i. In addition, I added dummy variables for health and included the indicator on employment. This set of controls should account for variations in the exogenous characteristics of the population and result in a robust estimate of the casual effect of having health insurance as it relates to doctor visits and overnight stays. The results of these modifications are shown in the table below.

	Linear: sawdr	Quadratic: sawdr	Linear: inhosp	Quadratic: inhosp
	(1)	(2)	(3)	(4)
covered	0.129^{***}	0.126^{**}	0.141^{***}	0.155^{***}
	(0.032)	(0.050)	(0.035)	(0.055)
R^2	0.038	0.038	0.068	0.067
Adjusted R^2	0.038	0.038	0.068	0.067
Note:				*p<0.1; **p<0.05; ***p<0.01

2.7.1 Results

After adding these specifications, the original models appear to overestimate the casual effect of insurance on the probability of seeing a doctor and underestimated that of having an overnight hospital stay. The revised models seem reasonable: Overnight stays in a hospital are expensive, so I would expect the impact of insurance to cause the probability of overnight stays in a hospital to increase, as an individual no longer has to assume the majority of the cost of the stay. Considering 0 is not in the confidence interval for the coefficients on *covered* in the *inhosp* regressions, we can confirm there is some nonzero jump in probability of overnight stays with the introduction of insurance. By adjusting the bandwidths, the figures support these results. In conclusion, the casual effect of insurance results in both increased doctor visits and overnight stays in the neighborhood of 12 percent and increases in doctor visits and 15 percent increases in overnight stays (+/- 2SE).

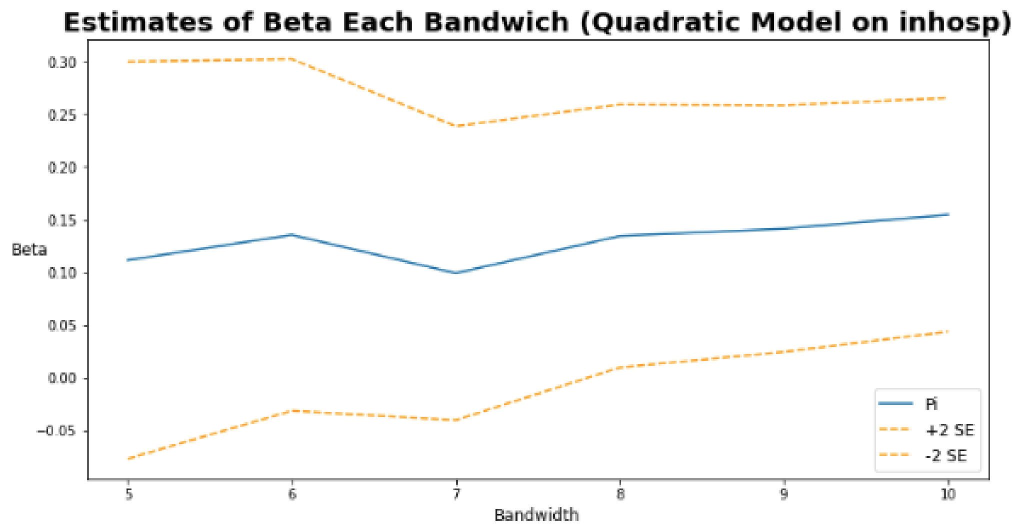


Figure 6: 2.6

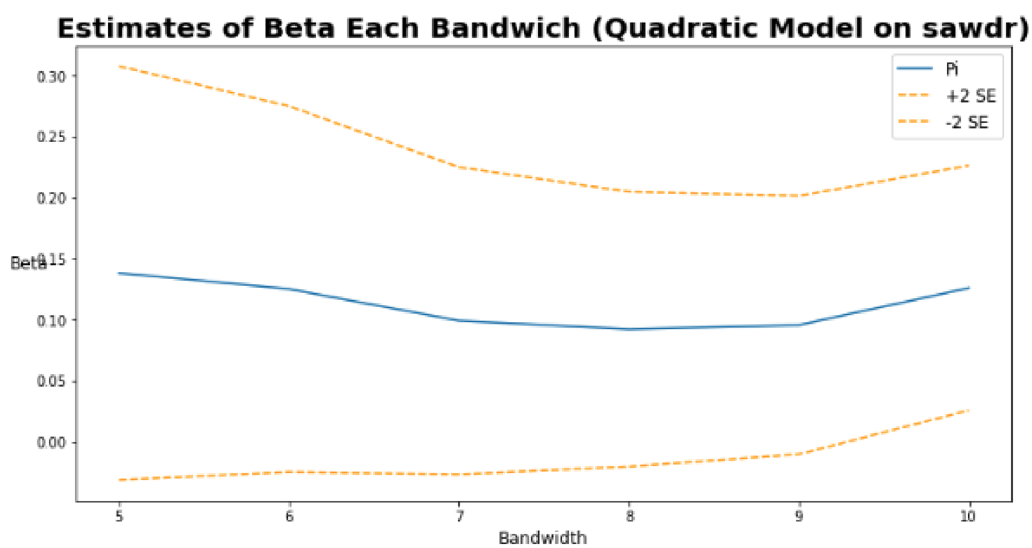


Figure 7: 2.7

3 Appendix

3.1 Appendix 1: Part 1, Code Up to DoubleML

3.2 Appendix 2: Part 1, DoubleML Code

3.3 Appendix 3: Part 2, Required Analysis

3.4 Appendix 4: Part 2, Open Ended Analysis

Final Project 1

May 13, 2021

1 code appendix 1

[1]:
```
import sys
!{sys.executable} -m pip install stargazer
import pandas as pd
import statsmodels.api as sm
from patsy import dmatrices
import numpy as np
from statsmodels.sandbox.regression.gmm import IV2SLS
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LassoCV
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from stargazer.stargazer import Stargazer, LineLocation
from IPython.core.display import HTML


import seaborn as sns
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (16,8)

import warnings
warnings.filterwarnings('ignore')
```

```
Collecting stargazer
  Using cached stargazer-0.0.5-py3-none-any.whl (9.7 kB)
Installing collected packages: stargazer
Successfully installed stargazer-0.0.5
```

1.1 read in tables

[2]:
```
bal = pd.read_csv('balanced.csv')
bal.head()
```

[2]:
```
   co_owned  southj  centralj  northj  pa1  pa2  shore  ncalls  wage_st  \
0         1       0         1       0    0    0      0       2     5.00
```

```
1          0        0          1        0     0     0       0        0      5.12
2          0        0          0        1     0     0       0        3      5.56
3          1        0          1        0     0     0       0        2      5.00
4          0        0          0        1     0     0       0        4      5.00

   bonus  …   pchwage   gap  nj  bk  kfc  roys  wendys  atmin  atnewmin2  \
0      0  …  0.010000  0.01   1   0    0     1       0      0          1
1      0  … -0.013672  0.00   1   0    1     0       0      0          1
2      1  … -0.091727  0.00   1   1    0     0       0      0          1
3      0  …  0.010000  0.01   1   0    1     0       0      0          1
4      0  …  0.010000  0.01   1   0    1     0       0      0          1

   freemeal
0         1
1         3
2         2
3         1
4         1

[5 rows x 33 columns]
```

[3]:
```
prd = pd.read_csv('projectrd.csv')
prd.head()
```

[3]:
```
   REGION  EDUC  female   age4  hispanic  wnh  bnh  onh  inhosp  sawdr  …  \
0       2     8       0  56.50         0    1    0    0     0.0    1.0  …
1       4     6       0  65.25         1    0    0    0     0.0    0.0  …
2       4     9       1  59.75         1    0    0    0     0.0    0.0  …
3       4    18       1  61.25         1    0    0    0     0.0    NaN  …
4       4    12       0  71.00         1    0    0    0     0.0    0.0  …

   mcare  health     r  z   r_z  dropout  somecoll  college  covered  vghealth
0      0     3.0 -8.50  0 -0.00        1         0        0        1         0
1      1     3.0  0.25  1  0.25        1         0        0        1         0
2      1     3.0 -5.25  0 -0.00        1         0        0        1         0
3      0     5.0 -3.75  0 -0.00        0         0        1        1         0
4      1     3.0  6.00  1  6.00        0         0        0        1         0

[5 rows x 21 columns]
```

1.2 part 1

1.2.1 1.2 a/

1.2.2 table 1

[4]:
```
bal_nj_only = bal[bal['nj'] == 1]
bal_pa_only = bal[bal['nj'] == 0]
# bal.columns
```

[5]:
```
def compute_means(column):
    nj_mean = np.mean(bal_nj_only[column])
    pa_mean = np.mean(bal_pa_only[column])
    diff = nj_mean - pa_mean

    return [nj_mean,
            pa_mean,
            diff]
```

[6]:
```
col_of_interest = ['wage_st', 'wage_st2', 'pchwage',
                   'emptot','emptot2','pchemp']

rows_table1 = []

for column in col_of_interest:
    rows_table1.append(compute_means(column))

table1 = pd.DataFrame(columns=['NJ', 'PA', 'Difference'],
                      data=np.array(rows_table1))
table1
```

[6]:

	NJ	PA	Difference
0	4.612982	4.653636	-0.040654
1	5.082140	4.618788	0.463352
2	0.107230	-0.004168	0.111399
3	20.678246	23.704545	-3.026300
4	21.076316	21.825758	-0.749442
5	0.022006	-0.032929	0.054935

1.2.3 Can be cross-checked with my results on page 3

1.2.4 1.2 b/

[14]:
```
reg1_2b = sm.OLS(endog=bal['pchwage'], exog=sm.add_constant(bal[['nj']])).fit()
reg1_2b.summary()
```

[14]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
```

```
==============================================================================
Dep. Variable:                pchwage   R-squared:                       0.233
Model:                            OLS   Adj. R-squared:                  0.231
Method:                 Least Squares   F-statistic:                     106.1
Date:                Thu, 13 May 2021   Prob (F-statistic):           6.63e-22
Time:                        21:38:01   Log-Likelihood:                 393.19
No. Observations:                 351   AIC:                            -782.4
Df Residuals:                     349   BIC:                            -774.7
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0042      0.010     -0.428      0.669      -0.023       0.015
nj             0.1114      0.011     10.302      0.000       0.090       0.133
==============================================================================
Omnibus:                       31.190   Durbin-Watson:                   0.693
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              113.395
Skew:                           0.255   Prob(JB):                     2.38e-25
Kurtosis:                       5.737   Cond. No.                         4.41
==============================================================================

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

1.2.5 1.2 c/

[15]:
```
reg1_2c = sm.OLS(endog=bal['pchemp'], exog=sm.add_constant(bal[['nj']])).fit()
reg1_2c.summary()
```

[15]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                 pchemp   R-squared:                       0.004
Model:                            OLS   Adj. R-squared:                  0.001
Method:                 Least Squares   F-statistic:                     1.297
Date:                Thu, 13 May 2021   Prob (F-statistic):              0.256
Time:                        21:38:01   Log-Likelihood:                -131.72
No. Observations:                 351   AIC:                             267.4
Df Residuals:                     349   BIC:                             275.2
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
```

```
------------------------------------------------------------------------------
const         -0.0329      0.043     -0.757      0.449      -0.118       0.053
nj             0.0549      0.048      1.139      0.256      -0.040       0.150
==============================================================================
Omnibus:                        1.239   Durbin-Watson:                   2.046
Prob(Omnibus):                  0.538   Jarque-Bera (JB):                0.994
Skew:                          -0.065   Prob(JB):                        0.608
Kurtosis:                       3.226   Cond. No.                         4.41
==============================================================================

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

1.2.6 1.2 d/

[16]:
```
# y3a, X3a = dmatrices("lwage76 ~ ed76 + exp76 + black + momdad14 + smsa66r +
↪reg662 + reg663 + reg664 + reg665 + reg666 + reg667 + reg668 + reg669",
#                      data=edu, return_type = "dataframe")
# y3a, InsV3a = dmatrices("lwage76 ~ nearc4 + exp76 + black + momdad14 +
↪smsa66r + reg662 + reg663 + reg664 + reg665 + reg666 + reg667 + reg668 +
↪reg669",
#                      data=edu, return_type = "dataframe")
# result3a = IV2SLS(exog = X3a, endog = y3a, instrument = InsV3a)
# result3a.fit().summary()

exog_1_2 = sm.add_constant(bal['pchwage'])
ins_1_2 = sm.add_constant(bal[['nj']])
result_1_2_iv = IV2SLS(endog = bal['pchemp'],
                    exog = exog_1_2, instrument = ins_1_2).fit()
result_1_2_iv.summary();
```

[17]:
```
reg1_2c.params[1]/reg1_2b.params[1]
```

[17]: 0.4931359457410413

[18]:
```
result_1_2_iv.params[1]
```

[18]: 0.4931359457410509

1.2.7 1.2 e/

[19]:
```
## first stage

reg1_2e_1 = sm.OLS(endog=bal['pchwage'], exog=sm.add_constant(bal[['gap']])).
↪fit()
reg1_2e_1.summary();
```

[20]:
```
## reduced form

reg1_2e_2 = sm.OLS(endog=bal['pchemp'], exog=sm.add_constant(bal[['gap']])).
↪fit()
reg1_2e_2.summary();
```

1.2.8 table 2

[21]:
```
ols_casual_1_2 = sm.OLS(endog=bal['pchemp'], exog=sm.
↪add_constant(bal[['pchwage']])).fit()
ols_casual_1_2.summary();
```

[22]:
```
table2_row1 = [ols_casual_1_2.params[0], reg1_2e_1.params[0],
               reg1_2e_2.params[0], result_1_2_iv.params[0]]
table2_row2 = [ols_casual_1_2.bse[0], reg1_2e_1.bse[0],
               reg1_2e_2.bse[0], result_1_2_iv.bse[0]]
table2_row3 = [ols_casual_1_2.params[1], np.nan,
               np.nan, result_1_2_iv.params[1]]
table2_row4 = [ols_casual_1_2.bse[1], np.nan,
               np.nan, result_1_2_iv.bse[1]]
table2_row5 = [np.nan, reg1_2e_1.params[1],
               reg1_2e_2.params[1], np.nan]
table2_row6 = [np.nan, reg1_2e_1.bse[1],
               reg1_2e_2.bse[1], np.nan]
table2_row7 = [np.sqrt(sum((ols_casual_1_2.predict() -␣
↪list(bal['pchemp']))**2)),
               np.sqrt(sum((reg1_2e_1.predict() - list(bal['pchwage']))**2)),
               np.sqrt(sum((reg1_2e_2.predict() - list(bal['pchemp']))**2)),
               np.sqrt(sum((result_1_2_iv.predict() - list(bal['pchemp']))**2))]

table2_row8 = [ols_casual_1_2.rsquared, reg1_2e_1.rsquared,
               reg1_2e_2.rsquared, result_1_2_iv.rsquared]
```

[23]:
```
rows_table2 = [table2_row1, table2_row2, table2_row3, table2_row4,
               table2_row5, table2_row6, table2_row7, table2_row8]
```

[24]:
```
table2 = pd.DataFrame(columns=['OLS Estimate', 'First Stage',
                               'Reduced Form', 'IV Estimate'],
                      data = np.array(rows_table2))
```

```
table2
```

[24]:

	OLS Estimate	First Stage	Reduced Form	IV Estimate
0	-0.024414	-0.002067	-0.031975	-0.030873
1	0.025994	0.003483	0.028156	0.041698
2	0.418274	NaN	NaN	0.493136
3	0.208315	NaN	NaN	0.431474
4	NaN	1.040648	0.514154	NaN
5	NaN	0.030583	0.247203	NaN
6	6.572155	0.812740	6.569417	6.573371
7	0.011420	0.768391	0.012243	0.011054

1.2.9 stargazer table 2

[26]:
```
sg_table2 = Stargazer([ols_casual_1_2, reg1_2e_1, reg1_2e_2, result_1_2_iv])
sg_table2.title('Table 2')
sg_table2.custom_columns(['OLS', 'First Stage', 'Reduced Form', 'IV'], [1, 1,␣
↪1, 1])
sg_table2.covariate_order(['const', 'pchwage', 'gap'])
sg_table2.add_line('RMSE', table2_row7, LineLocation.FOOTER_TOP)
print(sg_table2.render_latex())
```

```
\begin{table}[!htbp] \centering
  \caption{Table 2}
\begin{tabular}{@{\extracolsep{5pt}}lcccc}
\\[-1.8ex]\hline
\hline \\[-1.8ex]
\\[-1.8ex] & \multicolumn{1}{c}{OLS} & \multicolumn{1}{c}{First Stage} &
\multicolumn{1}{c}{Reduced Form} & \multicolumn{1}{c}{IV}  \\
\\[-1.8ex] & (1) & (2) & (3) & (4) \\
\hline \\[-1.8ex]
 const & -0.024$^{}$ & -0.002$^{}$ & -0.032$^{}$ & -0.031$^{}$ \\
  & (0.026) & (0.003) & (0.028) & (0.042) \\
 pchwage & 0.418$^{**}$ & & & 0.493$^{}$ \\
  & (0.208) & & & (0.431) \\
 gap & & 1.041$^{***}$ & 0.514$^{**}$ & \\
  & & (0.031) & (0.247) & \\
\hline \\[-1.8ex]
 RMSE & 6.572154999975606 & 0.812739923041351 & 6.56941711704484 &
6.573370867237839 \\
 Observations & 351 & 351 & 351 & 351 \\
 $R^2$ & 0.011 & 0.768 & 0.012 & 0.011 \\
 Adjusted $R^2$ & 0.009 & 0.768 & 0.009 & 0.008 \\
 Residual Std. Error & 0.352(df = 349) & 0.044(df = 349) & 0.352(df = 349) &
0.352(df = 349)  \\
 F Statistic & 4.032$^{**}$ (df = 1.0; 349.0) & 1157.848$^{***}$ (df = 1.0;
349.0) & 4.326$^{**}$ (df = 1.0; 349.0) & 1.306$^{}$ (df = 1.0; 349) \\
```

```
\hline
\hline \\[-1.8ex]
\textit{Note:} & \multicolumn{4}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05;
$^{***}$p$<$0.01} \\
\end{tabular}
\end{table}
```

verfication

```
[27]: 0.514154 / 1.040648
```

```
[27]: 0.4940710019141919
```

1.2.10 f

```
[28]: reg1_2f_casual = sm.OLS(endog=bal['pchemp'], exog=sm.
       ↪add_constant(bal[['pchwage','nj']])).fit()
      reg1_2f_casual.summary();
```

```
[29]: reg1_2f_fs = sm.OLS(endog=bal['pchwage'], exog=sm.
       ↪add_constant(bal[['gap','nj']])).fit()
      reg1_2f_fs.summary();
```

```
[30]: reg1_2f_rf = sm.OLS(endog=bal['pchemp'], exog=sm.
       ↪add_constant(bal[['gap','nj']])).fit()
      reg1_2f_rf.summary();
```

```
[31]: exog_1_2_f = sm.add_constant(bal[['pchwage', 'nj']])
      ins_1_2_f = sm.add_constant(bal[['gap','nj']])
      result_1_2_iv_f = IV2SLS(endog = bal['pchemp'],
                               exog = exog_1_2_f, instrument = ins_1_2_f).fit()
      result_1_2_iv_f.summary();
```

1.2.11 table 3

```
[32]: ## reg1_2f_casual   reg1_2f_fs   reg1_2f_rf   result_1_2_iv_f

      table3_row1 = [reg1_2f_casual.params[0], reg1_2f_fs.params[0],
                     reg1_2f_rf.params[0], result_1_2_iv_f.params[0]]
      table3_row2 = [reg1_2f_casual.bse[0], reg1_2f_fs.bse[0],
                     reg1_2f_rf.bse[0], result_1_2_iv_f.bse[0]]
      table3_row3 = [reg1_2f_casual.params[1], np.nan,
                     np.nan, result_1_2_iv_f.params[1]]
      table3_row4 = [reg1_2f_casual.bse[1], np.nan,
                     np.nan, result_1_2_iv_f.bse[1]]
      table3_row5 = [np.nan, reg1_2f_fs.params[1],
                     reg1_2f_rf.params[1], np.nan]
```

```
table3_row6 = [np.nan, reg1_2f_fs.bse[1],
               reg1_2f_rf.bse[1], np.nan]
table3_row7 = [reg1_2f_casual.params[2], reg1_2f_fs.params[2],
               reg1_2f_rf.params[2], result_1_2_iv_f.params[2]]

table3_row8 = [reg1_2f_casual.bse[2], reg1_2f_fs.bse[2],
               reg1_2f_rf.bse[2], result_1_2_iv_f.bse[2]]


table3_row9 = [np.sqrt(sum((reg1_2f_casual.predict() -␣
 ↪list(bal['pchemp']))**2)),
               np.sqrt(sum((reg1_2f_fs.predict() - list(bal['pchwage']))**2)),
               np.sqrt(sum((reg1_2f_rf.predict() - list(bal['pchemp']))**2)),
               np.sqrt(sum((result_1_2_iv_f.predict() -␣
 ↪list(bal['pchemp']))**2))]

table3_row10 = [reg1_2f_casual.rsquared, reg1_2f_fs.rsquared,
               reg1_2f_rf.rsquared, result_1_2_iv_f.rsquared]
```

[33]:
```
rows_table3 = [table3_row1, table3_row2, table3_row3, table3_row4,
               table3_row5, table3_row6, table3_row7, table3_row8,
               table3_row9, table3_row10]
```

[34]:
```
table3 = pd.DataFrame(columns=['OLS Estimate', 'First Stage',
                               'Reduced Form', 'IV Estimate'],
                      data = np.array(rows_table3))
table3
```

[34]:

	OLS Estimate	First Stage	Reduced Form	IV Estimate
0	-0.031280	-0.004168	-0.032929	-0.030868
1	0.043375	0.005361	0.043348	0.043390
2	0.395510	NaN	NaN	0.494466
3	0.238218	NaN	NaN	0.285164
4	NaN	1.030562	0.509578	NaN
5	NaN	0.036321	0.293700	NaN
6	0.010875	0.003642	0.001653	-0.000148
7	0.054955	0.007058	0.057072	0.057672
8	6.571785	0.812429	6.569409	6.573414
9	0.011531	0.768568	0.012246	0.011041

1.2.12 stargazer table 3

[35]:
```
sg_table3 = Stargazer([reg1_2f_casual, reg1_2f_fs, reg1_2f_rf, result_1_2_iv_f])
sg_table3.title('Table 3')
sg_table3.custom_columns(['OLS', 'First Stage', 'Reduced Form', 'IV'], [1, 1,␣
 ↪1, 1])
```

```python
sg_table3.add_line('RMSE', table3_row9, LineLocation.FOOTER_TOP)
sg_table3.covariate_order(['const', 'pchwage', 'gap', 'nj'])
print(sg_table3.render_latex())
```

```
\begin{table}[!htbp] \centering
  \caption{Table 3}
\begin{tabular}{@{\extracolsep{5pt}}lcccc}
\\[-1.8ex]\hline
\hline \\[-1.8ex]
\\[-1.8ex] & \multicolumn{1}{c}{OLS} & \multicolumn{1}{c}{First Stage} &
\multicolumn{1}{c}{Reduced Form} & \multicolumn{1}{c}{IV}  \\
\\[-1.8ex] & (1) & (2) & (3) & (4) \\
\hline \\[-1.8ex]
 const & -0.031$^{}$ & -0.004$^{}$ & -0.033$^{}$ & -0.031$^{}$ \\
  & (0.043) & (0.005) & (0.043) & (0.043) \\
 pchwage & 0.396$^{*}$ & & & 0.494$^{*}$ \\
  & (0.238) & & & (0.285) \\
 gap & & 1.031$^{***}$ & 0.510$^{*}$ & \\
  & & (0.036) & (0.294) & \\
 nj & 0.011$^{}$ & 0.004$^{}$ & 0.002$^{}$ & -0.000$^{}$ \\
  & (0.055) & (0.007) & (0.057) & (0.058) \\
\hline \\[-1.8ex]
 RMSE & 6.571785222475173 & 0.8124291550961136 & 6.569409201265866 &
6.573414352535168 \\
 Observations & 351 & 351 & 351 & 351 \\
 $R^2$ & 0.012 & 0.769 & 0.012 & 0.011 \\
 Adjusted $R^2$ & 0.006 & 0.767 & 0.007 & 0.005 \\
 Residual Std. Error & 0.352(df = 348) & 0.044(df = 348) & 0.352(df = 348) &
0.352(df = 348)  \\
 F Statistic & 2.030$^{}$ (df = 2.0; 348.0) & 577.840$^{***}$ (df = 2.0; 348.0)
& 2.157$^{}$ (df = 2.0; 348.0) & 2.155$^{}$ (df = 2.0; 348) \\
\hline
\hline \\[-1.8ex]
\textit{Note:} & \multicolumn{4}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05;
$^{***}$p$<$0.01} \\
\end{tabular}
\end{table}
```

[36]:
```python
bal_nj_only.head()
```

[36]:

	co_owned	southj	centralj	northj	pa1	pa2	shore	ncalls	wage_st	\
0	1	0	1	0	0	0	0	2	5.00	
1	0	0	1	0	0	0	0	0	5.12	
2	0	0	0	1	0	0	0	3	5.56	
3	1	0	1	0	0	0	0	2	5.00	
4	0	0	0	1	0	0	0	4	5.00	

```
   bonus  …   pchwage   gap  nj  bk  kfc  roys  wendys  atmin  atnewmin2  \
0      0  …  0.010000  0.01   1   0    0     1       0      0          1
1      0  … -0.013672  0.00   1   0    1     0       0      0          1
2      1  … -0.091727  0.00   1   1    0     0       0      0          1
3      0  …  0.010000  0.01   1   0    1     0       0      0          1
4      0  …  0.010000  0.01   1   0    1     0       0      0          1

   freemeal
0         1
1         3
2         2
3         1
4         1

[5 rows x 33 columns]
```

1.2.13 g

[37]:
```
ols_casual_1_2_g = sm.OLS(endog=bal_nj_only['pchemp'], exog=sm.
 ↪add_constant(bal_nj_only[['pchwage']])).fit()
ols_casual_1_2_g.summary();
```

[38]:
```
reg1_2e_1_g = sm.OLS(endog=bal_nj_only['pchwage'], exog=sm.
 ↪add_constant(bal_nj_only[['gap']])).fit()
reg1_2e_1_g.summary();
```

[39]:
```
reg1_2e_2_g = sm.OLS(endog=bal_nj_only['pchemp'], exog=sm.
 ↪add_constant(bal_nj_only[['gap']])).fit()
reg1_2e_2_g.summary();
```

[40]:
```
exog_1_2_g = sm.add_constant(bal_nj_only['pchwage'])
ins_1_2_g = sm.add_constant(bal_nj_only[['gap']])
result_1_2_iv_g = IV2SLS(endog = bal_nj_only['pchemp'],
                    exog = exog_1_2_g, instrument = ins_1_2_g).fit()
result_1_2_iv_g.summary();
```

[41]:
```
table4_row1 = [ols_casual_1_2_g.params[0], reg1_2e_1_g.params[0],
               reg1_2e_2_g.params[0], result_1_2_iv_g.params[0]]
table4_row2 = [ols_casual_1_2_g.bse[0], reg1_2e_1_g.bse[0],
               reg1_2e_2_g.bse[0], result_1_2_iv_g.bse[0]]
table4_row3 = [ols_casual_1_2_g.params[1], np.nan,
               np.nan, result_1_2_iv_g.params[1]]
table4_row4 = [ols_casual_1_2_g.bse[1], np.nan,
               np.nan, result_1_2_iv_g.bse[1]]
table4_row5 = [np.nan, reg1_2e_1_g.params[1],
               reg1_2e_2_g.params[1], np.nan]
```

```
table4_row6 = [np.nan, reg1_2e_1_g.bse[1],
               reg1_2e_2_g.bse[1], np.nan]
table4_row7 = [np.sqrt(sum((ols_casual_1_2_g.predict() -␣
 ↪list(bal_nj_only['pchemp']))**2)),
               np.sqrt(sum((reg1_2e_1_g.predict() -␣
 ↪list(bal_nj_only['pchwage']))**2)),
               np.sqrt(sum((reg1_2e_2_g.predict() -␣
 ↪list(bal_nj_only['pchemp']))**2)),
               np.sqrt(sum((result_1_2_iv_g.predict() -␣
 ↪list(bal_nj_only['pchemp']))**2))]

table4_row8 = [ols_casual_1_2_g.rsquared, reg1_2e_1_g.rsquared,
               reg1_2e_2_g.rsquared, result_1_2_iv_g.rsquared]
```

[42]:
```
rows_table4 = [table4_row1, table4_row2, table4_row3, table4_row4,
               table4_row5, table4_row6, table4_row7, table4_row8]
```

[43]:
```
table4 = pd.DataFrame(columns=['OLS Estimate', 'First Stage',
                               'Reduced Form', 'IV Estimate'],
                      data = np.array(rows_table4))
table4
```

[43]:

	OLS Estimate	First Stage	Reduced Form	IV Estimate
0	-0.043076	-0.000526	-0.031276	-0.031016
1	0.034746	0.002721	0.036387	0.036138
2	0.606932	NaN	NaN	0.494466
3	0.262526	NaN	NaN	0.278349
4	NaN	1.030562	0.509578	NaN
5	NaN	0.021524	0.287869	NaN
6	5.784286	0.434168	5.806594	5.786161
7	0.018536	0.890113	0.010951	0.017900

[44]:
```
table3
```

[44]:

	OLS Estimate	First Stage	Reduced Form	IV Estimate
0	-0.031280	-0.004168	-0.032929	-0.030868
1	0.043375	0.005361	0.043348	0.043390
2	0.395510	NaN	NaN	0.494466
3	0.238218	NaN	NaN	0.285164
4	NaN	1.030562	0.509578	NaN
5	NaN	0.036321	0.293700	NaN
6	0.010875	0.003642	0.001653	-0.000148
7	0.054955	0.007058	0.057072	0.057672
8	6.571785	0.812429	6.569409	6.573414
9	0.011531	0.768568	0.012246	0.011041

1.2.14 stargazer table 4

[45]:
```
sg_table4 = Stargazer([ols_casual_1_2_g, reg1_2e_1_g, reg1_2e_2_g,␣
↪result_1_2_iv_g])
sg_table4.title('Table 4')
sg_table4.custom_columns(['OLS', 'First Stage', 'Reduced Form', 'IV'], [1, 1,␣
↪1, 1])
sg_table4.covariate_order(['const','pchwage','gap'])
sg_table4.add_line('RMSE', table4_row7, LineLocation.FOOTER_TOP)
print(sg_table4.render_latex())
```

```
\begin{table}[!htbp] \centering
  \caption{Table 4}
\begin{tabular}{@{\extracolsep{5pt}}lcccc}
\\[-1.8ex]\hline
\hline \\[-1.8ex]
\\[-1.8ex] & \multicolumn{1}{c}{OLS} & \multicolumn{1}{c}{First Stage} &
\multicolumn{1}{c}{Reduced Form} & \multicolumn{1}{c}{IV}  \\
\\[-1.8ex] & (1) & (2) & (3) & (4) \\
\hline \\[-1.8ex]
 const & -0.043$^{}$ & -0.001$^{}$ & -0.031$^{}$ & -0.031$^{}$ \\
  & (0.035) & (0.003) & (0.036) & (0.036) \\
 pchwage & 0.607$^{**}$ & & & 0.494$^{*}$ \\
  & (0.263) & & & (0.278) \\
 gap & & 1.031$^{***}$ & 0.510$^{*}$ & \\
  & & (0.022) & (0.288) & \\
\hline \\[-1.8ex]
 RMSE & 5.784285730662954 & 0.4341681370695096 & 5.806594350055371 &
5.786160997522124 \\
 Observations & 285 & 285 & 285 & 285 \\
 $R^2$ & 0.019 & 0.890 & 0.011 & 0.018 \\
 Adjusted $R^2$ & 0.015 & 0.890 & 0.007 & 0.014 \\
 Residual Std. Error & 0.344(df = 283) & 0.026(df = 283) & 0.345(df = 283) &
0.344(df = 283)  \\
 F Statistic & 5.345$^{**}$ (df = 1.0; 283.0) & 2292.371$^{***}$ (df = 1.0;
283.0) & 3.134$^{*}$ (df = 1.0; 283.0) & 3.156$^{*}$ (df = 1.0; 283) \\
\hline
\hline \\[-1.8ex]
\textit{Note:} & \multicolumn{4}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05;
$^{***}$p$<$0.01} \\
\end{tabular}
\end{table}
```

1.2.15 narrative

1.2.16 h

[46]:
```python
bal.columns
```

[46]:
```
Index(['co_owned', 'southj', 'centralj', 'northj', 'pa1', 'pa2', 'shore',
       'ncalls', 'wage_st', 'bonus', 'open', 'hrsopen', 'psoda', 'pfry',
       'pentree', 'nregs', 'nregs11', 'wage_st2', 'emptot', 'emptot2', 'demp',
       'pchemp', 'dwage', 'pchwage', 'gap', 'nj', 'bk', 'kfc', 'roys',
       'wendys', 'atmin', 'atnewmin2', 'freemeal'],
      dtype='object')
```

[47]:
```python
reg1_2h_casual = sm.OLS(endog=bal['pchemp'], exog=sm.
 ↪add_constant(bal[['pchwage','southj', 'centralj', 'northj', 'shore',␣
 ↪'pa2']])).fit()
reg1_2h_casual.summary();
```

[48]:
```python
reg1_2h_fs = sm.OLS(endog=bal['pchwage'], exog=sm.
 ↪add_constant(bal[['gap','southj', 'centralj', 'northj', 'shore', 'pa2']])).
 ↪fit()
reg1_2h_fs.summary();
```

[49]:
```python
reg1_2h_rf = sm.OLS(endog=bal['pchemp'], exog=sm.
 ↪add_constant(bal[['gap','southj', 'centralj', 'northj', 'shore', 'pa2']])).
 ↪fit()
reg1_2h_rf.summary();
```

[50]:
```python
exog_1_2_h = sm.add_constant(bal[['pchwage','southj', 'centralj', 'northj',␣
 ↪'shore', 'pa2']])
ins_1_2_h = sm.add_constant(bal[['gap','southj', 'centralj', 'northj', 'shore',␣
 ↪'pa2']])
result_1_2_iv_h = IV2SLS(endog = bal['pchemp'],
                    exog = exog_1_2_h, instrument = ins_1_2_h).fit()
result_1_2_iv_h.summary();
```

[51]:
```python
## constant
table5_row1 = [reg1_2h_casual.params[0], reg1_2h_fs.params[0],
               reg1_2h_rf.params[0], result_1_2_iv_h.params[0]]
table5_row2 = [reg1_2h_casual.bse[0], reg1_2h_fs.bse[0],
               reg1_2h_rf.bse[0], result_1_2_iv_h.bse[0]]
## pchwage

table5_row3 = [reg1_2h_casual.params[1], np.nan,
               np.nan, result_1_2_iv_f.params[1]]
table5_row4 = [reg1_2h_casual.bse[1], np.nan,
               np.nan, result_1_2_iv_f.bse[1]]
## gap
```

```
table5_row5 = [np.nan, reg1_2h_fs.params[1],
               reg1_2h_rf.params[1], np.nan]
table5_row6 = [np.nan, reg1_2h_fs.bse[1],
               reg1_2h_rf.bse[1], np.nan]
## region 1

table5_row7 = [reg1_2h_casual.params[2], reg1_2h_fs.params[2],
               reg1_2h_rf.params[2], result_1_2_iv_h.params[2]]

table5_row8 = [reg1_2h_casual.bse[2], reg1_2h_fs.bse[2],
               reg1_2h_rf.bse[2], result_1_2_iv_h.bse[2]]

## region2

table5_row9 = [reg1_2h_casual.params[3], reg1_2h_fs.params[3],
               reg1_2h_rf.params[3], result_1_2_iv_h.params[3]]

table5_row10 = [reg1_2h_casual.bse[3], reg1_2h_fs.bse[3],
               reg1_2h_rf.bse[3], result_1_2_iv_h.bse[3]]

## region3

table5_row11 = [reg1_2h_casual.params[4], reg1_2h_fs.params[4],
               reg1_2h_rf.params[4], result_1_2_iv_h.params[4]]

table5_row12 = [reg1_2h_casual.bse[4], reg1_2h_fs.bse[4],
               reg1_2h_rf.bse[4], result_1_2_iv_h.bse[4]]

## region4

table5_row13 = [reg1_2h_casual.params[5], reg1_2h_fs.params[5],
               reg1_2h_rf.params[5], result_1_2_iv_h.params[5]]

table5_row14 = [reg1_2h_casual.bse[5], reg1_2h_fs.bse[5],
               reg1_2h_rf.bse[5], result_1_2_iv_h.bse[5]]

## region5

table5_row15 = [reg1_2h_casual.params[6], reg1_2h_fs.params[6],
               reg1_2h_rf.params[6], result_1_2_iv_h.params[6]]

table5_row16 = [reg1_2h_casual.bse[6], reg1_2h_fs.bse[6],
               reg1_2h_rf.bse[6], result_1_2_iv_h.bse[6]]
```

```
table5_row17 = [np.sqrt(sum((reg1_2h_casual.predict() -␣
↪list(bal['pchemp']))**2)),
                np.sqrt(sum((reg1_2h_fs.predict() - list(bal['pchwage']))**2)),
                np.sqrt(sum((reg1_2h_rf.predict() - list(bal['pchemp']))**2)),
                np.sqrt(sum((result_1_2_iv_h.predict() -␣
↪list(bal['pchemp']))**2))]

table5_row18 = [reg1_2h_casual.rsquared, reg1_2h_fs.rsquared,
                reg1_2h_rf.rsquared, result_1_2_iv_h.rsquared]
```

[52]:
```
table5_rows = [table5_row1, table5_row2, table5_row3, table5_row4, table5_row5,
               table5_row6, table5_row7, table5_row8, table5_row9, table5_row10,
               table5_row11, table5_row12, table5_row13, table5_row14,␣
↪table5_row15,
               table5_row16, table5_row17, table5_row18]
```

[53]:
```
table5 = pd.DataFrame(columns=['OLS Estimate', 'First Stage',
                               'Reduced Form', 'IV Estimate'],
                      data = np.array(table5_rows))
table5
```

[53]:

	OLS Estimate	First Stage	Reduced Form	IV Estimate
0	-0.109883	0.004871	-0.107927	-0.110309
1	0.066573	0.008243	0.066553	0.066590
2	0.401544	NaN	NaN	0.494466
3	0.238895	NaN	NaN	0.285164
4	NaN	1.031081	0.504130	NaN
5	NaN	0.036487	0.294592	NaN
6	0.105308	-0.006004	0.092645	0.095581
7	0.082643	0.010542	0.085114	0.084482
8	0.048735	-0.005471	0.037146	0.039821
9	0.086443	0.010952	0.088427	0.087927
10	0.104144	-0.003838	0.093501	0.095378
11	0.076434	0.009714	0.078433	0.078051
12	-0.049127	-0.006310	-0.051060	-0.047975
13	0.068695	0.008502	0.068641	0.068740
14	0.136565	-0.015700	0.130261	0.137937
15	0.087802	0.010863	0.087709	0.087854
16	6.532629	0.808979	6.531656	6.533899
17	0.023275	0.770529	0.023566	0.022895

[54]:
```
sg_table5 = Stargazer([reg1_2h_casual, reg1_2h_fs, reg1_2h_rf, result_1_2_iv_h])
sg_table5.title('Table 5')
sg_table5.custom_columns(['OLS', 'First Stage', 'Reduced Form', 'IV'], [1, 1,␣
↪1, 1])
sg_table5.add_line('RMSE', table5_row17, LineLocation.FOOTER_TOP)
sg_table5.covariate_order(['const','pchwage','gap','southj',
```

```python
                              'centralj','northj','shore','pa2'])
print(sg_table5.render_latex())
```

```
\begin{table}[!htbp] \centering
  \caption{Table 5}
\begin{tabular}{@{\extracolsep{5pt}}lcccc}
\\[-1.8ex]\hline
\hline \\[-1.8ex]
\\[-1.8ex] & \multicolumn{1}{c}{OLS} & \multicolumn{1}{c}{First Stage} &
\multicolumn{1}{c}{Reduced Form} & \multicolumn{1}{c}{IV}  \\
\\[-1.8ex] & (1) & (2) & (3) & (4) \\
\hline \\[-1.8ex]
 const & -0.110$^{*}$ & 0.005$^{}$ & -0.108$^{}$ & -0.110$^{*}$ \\
  & (0.067) & (0.008) & (0.067) & (0.067) \\
 pchwage & 0.402$^{*}$ & & & 0.489$^{*}$ \\
  & (0.239) & & & (0.286) \\
 gap & & 1.031$^{***}$ & 0.504$^{*}$ & \\
  & & (0.036) & (0.295) & \\
 southj & 0.105$^{}$ & -0.006$^{}$ & 0.093$^{}$ & 0.096$^{}$ \\
  & (0.083) & (0.011) & (0.085) & (0.084) \\
 centralj & 0.049$^{}$ & -0.005$^{}$ & 0.037$^{}$ & 0.040$^{}$ \\
  & (0.086) & (0.011) & (0.088) & (0.088) \\
 northj & 0.104$^{}$ & -0.004$^{}$ & 0.094$^{}$ & 0.095$^{}$ \\
  & (0.076) & (0.010) & (0.078) & (0.078) \\
 shore & -0.049$^{}$ & -0.006$^{}$ & -0.051$^{}$ & -0.048$^{}$ \\
  & (0.069) & (0.009) & (0.069) & (0.069) \\
 pa2 & 0.137$^{}$ & -0.016$^{}$ & 0.130$^{}$ & 0.138$^{}$ \\
  & (0.088) & (0.011) & (0.088) & (0.088) \\
\hline \\[-1.8ex]
 RMSE & 6.532628977855861 & 0.8089792364494209 & 6.531656446655876 &
6.5338994398400185 \\
 Observations & 351 & 351 & 351 & 351 \\
 $R^2$ & 0.023 & 0.771 & 0.024 & 0.023 \\
 Adjusted $R^2$ & 0.006 & 0.767 & 0.007 & 0.006 \\
 Residual Std. Error & 0.352(df = 344) & 0.044(df = 344) & 0.352(df = 344) &
0.352(df = 344)  \\
 F Statistic & 1.366$^{}$ (df = 6.0; 344.0) & 192.517$^{***}$ (df = 6.0; 344.0)
& 1.384$^{}$ (df = 6.0; 344.0) & 1.383$^{}$ (df = 6.0; 344) \\
\hline
\hline \\[-1.8ex]
\textit{Note:} & \multicolumn{4}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05;
$^{***}$p$<$0.01} \\
\end{tabular}
\end{table}
```

1.2.17 h

step 1

[55]:
```
### co_owned bk kfc roys wendys nj southj centralj northj shore pa2, ncalls,␣
↪bonus,
### open, hrsopen, psoda, pfrt, pentree, nregs, nregs11, freemeal, wage_st,␣
↪wage_st,
### dwage, pchwage, gap, emptot, emptot2, demp, pchemp, atmin, atnewmin2,␣
↪highwage

bal.columns
```

[55]:
```
Index(['co_owned', 'southj', 'centralj', 'northj', 'pa1', 'pa2', 'shore',
       'ncalls', 'wage_st', 'bonus', 'open', 'hrsopen', 'psoda', 'pfry',
       'pentree', 'nregs', 'nregs11', 'wage_st2', 'emptot', 'emptot2', 'demp',
       'pchemp', 'dwage', 'pchwage', 'gap', 'nj', 'bk', 'kfc', 'roys',
       'wendys', 'atmin', 'atnewmin2', 'freemeal'],
      dtype='object')
```

[56]:
```
bal = bal.fillna(bal.mean())
controls = ['co_owned', 'southj', 'centralj', 'northj', 'pa2', 'shore',
       'ncalls', 'bonus', 'open', 'hrsopen', 'psoda', 'pfry',
       'pentree', 'nregs', 'nregs11', 'bk', 'kfc', 'roys',
       'wendys', 'atmin', 'atnewmin2', 'freemeal']


## Casual

reg1_2h2_casual = sm.OLS(endog=bal['pchemp'],
                         exog=sm.add_constant(bal[['pchwage'] + controls])).
↪fit()
## First Stage

reg1_2h2_fs = sm.OLS(endog=bal['pchwage'], exog=sm.add_constant(bal[['gap'] +␣
↪controls])).fit()

## Reduced Form

reg1_2h2_rf = sm.OLS(endog=bal['pchemp'], exog=sm.add_constant(bal[['gap'] +␣
↪controls])).fit()

## IV

exog_1_2_h2 = sm.add_constant(bal[['pchwage'] + controls])
ins_1_2_h2 = sm.add_constant(bal[['gap'] + controls])
result_1_2_iv_h2 = IV2SLS(endog = bal['pchemp'],
                    exog = exog_1_2_h2, instrument = ins_1_2_h2).fit()
```

```
[57]: table6_row17 = [np.sqrt(sum((reg1_2h2_casual.predict() -
       ↪list(bal['pchemp']))**2)),
                   np.sqrt(sum((reg1_2h2_fs.predict() - list(bal['pchwage']))**2)),
                   np.sqrt(sum((reg1_2h2_rf.predict() - list(bal['pchemp']))**2)),
                   np.sqrt(sum((result_1_2_iv_h2.predict() -
       ↪list(bal['pchemp']))**2))]
```

```
[60]: sg_table6 = Stargazer([reg1_2h2_casual, reg1_2h2_fs, reg1_2h2_rf,
       ↪result_1_2_iv_h2])
      sg_table6.title('Table 6')
      sg_table6.custom_columns(['OLS', 'First Stage', 'Reduced Form', 'IV'], [1, 1,
       ↪1, 1])
      sg_table6.add_line('RMSE', table6_row17, LineLocation.FOOTER_TOP)
      sg_table6.covariate_order(['const','pchwage','gap'])
      sg_table6
```

```
[60]: <stargazer.stargazer.Stargazer at 0x7f63ff1b6490>
```

```
[61]: print(sg_table6.render_latex())
```

```
\begin{table}[!htbp] \centering
  \caption{Table 6}
\begin{tabular}{@{\extracolsep{5pt}}lcccc}
\\[-1.8ex]\hline
\hline \\[-1.8ex]
\\[-1.8ex] & \multicolumn{1}{c}{OLS} & \multicolumn{1}{c}{First Stage} &
\multicolumn{1}{c}{Reduced Form} & \multicolumn{1}{c}{IV}  \\
\\[-1.8ex] & (1) & (2) & (3) & (4) \\
\hline \\[-1.8ex]
 const & -1.272$^{**}$ & -0.002$^{}$ & -1.264$^{**}$ & -0.526$^{}$ \\
  & (0.506) & (0.060) & (0.506) & (1887702.369) \\
 pchwage & 0.212$^{}$ & & & 0.525$^{}$ \\
  & (0.331) & & & (0.469) \\
 gap & & 0.951$^{***}$ & 0.499$^{}$ & \\
  & & (0.052) & (0.444) & \\
\hline \\[-1.8ex]
 RMSE & 6.325397987239532 & 0.745039175819829 & 6.317194407008422 &
6.334016210697419 \\
 Observations & 351 & 351 & 351 & 351 \\
 $R^2$ & 0.084 & 0.805 & 0.087 & 0.082 \\
 Adjusted $R^2$ & 0.023 & 0.792 & 0.025 & 0.017 \\
 Residual Std. Error & 0.349(df = 328) & 0.041(df = 328) & 0.349(df = 328) &
0.350(df = 327)  \\
 F Statistic & 1.372$^{}$ (df = 22.0; 328.0) & 61.693$^{***}$ (df = 22.0; 328.0)
& 1.414$^{}$ (df = 22.0; 328.0) & 2.168$^{***}$ (df = 23.0; 327) \\
\hline
\hline \\[-1.8ex]
```

```
\textit{Note:} & \multicolumn{4}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05;
$^{***}$p$<$0.01} \\
\end{tabular}
\end{table}
```

Final Project 4

May 13, 2021

0.1 code appendix 2: double ML

[22]:
```
import sys
!{sys.executable} -m pip install stargazer
!{sys.executable} -m pip install -U DoubleML
import pandas as pd
import statsmodels.api as sm
from patsy import dmatrices
import numpy as np
from statsmodels.sandbox.regression.gmm import IV2SLS
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LassoCV
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from stargazer.stargazer import Stargazer, LineLocation
from IPython.core.display import HTML


import seaborn as sns
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (16,8)

import warnings
warnings.filterwarnings('ignore')
```

```
Requirement already satisfied: stargazer in /opt/conda/lib/python3.8/site-
packages (0.0.5)
Requirement already up-to-date: DoubleML in /opt/conda/lib/python3.8/site-
packages (0.2.2)
Requirement already satisfied, skipping upgrade: numpy in
/opt/conda/lib/python3.8/site-packages (from DoubleML) (1.19.5)
Requirement already satisfied, skipping upgrade: scipy in
/opt/conda/lib/python3.8/site-packages (from DoubleML) (1.6.0)
Requirement already satisfied, skipping upgrade: joblib in
/opt/conda/lib/python3.8/site-packages (from DoubleML) (1.0.0)
Requirement already satisfied, skipping upgrade: statsmodels in
/opt/conda/lib/python3.8/site-packages (from DoubleML) (0.11.1)
```

```
Requirement already satisfied, skipping upgrade: sklearn in
/opt/conda/lib/python3.8/site-packages (from DoubleML) (0.0)
Requirement already satisfied, skipping upgrade: pandas in
/opt/conda/lib/python3.8/site-packages (from DoubleML) (1.2.0)
Requirement already satisfied, skipping upgrade: patsy>=0.5 in
/opt/conda/lib/python3.8/site-packages (from statsmodels->DoubleML) (0.5.1)
Requirement already satisfied, skipping upgrade: scikit-learn in
/opt/conda/lib/python3.8/site-packages (from sklearn->DoubleML) (0.24.0)
Requirement already satisfied, skipping upgrade: pytz>=2017.3 in
/opt/conda/lib/python3.8/site-packages (from pandas->DoubleML) (2020.5)
Requirement already satisfied, skipping upgrade: python-dateutil>=2.7.3 in
/opt/conda/lib/python3.8/site-packages (from pandas->DoubleML) (2.8.1)
Requirement already satisfied, skipping upgrade: six in
/opt/conda/lib/python3.8/site-packages (from patsy>=0.5->statsmodels->DoubleML)
(1.15.0)
Requirement already satisfied, skipping upgrade: threadpoolctl>=2.0.0 in
/opt/conda/lib/python3.8/site-packages (from scikit-learn->sklearn->DoubleML)
(2.1.0)
```

[23]:
```
from doubleml import DoubleMLData
from doubleml import DoubleMLPLR
from sklearn.base import clone
from sklearn.linear_model import LassoCV
```

[24]:
```
learner = LassoCV(cv=10)
ml_g = clone(learner)
ml_m = clone(learner)
```

[25]:
```
bal = pd.read_csv('balanced.csv')
bal = bal.fillna(bal.mean())
```

[26]:
```
data = sm.add_constant(bal)
controls = ['co_owned', 'southj', 'centralj', 'northj', 'pa2', 'shore',
       'ncalls', 'bonus', 'open', 'hrsopen', 'psoda', 'pfry',
       'pentree', 'nregs', 'nregs11', 'bk', 'kfc', 'roys',
       'wendys', 'atmin', 'atnewmin2', 'freemeal','const']


# ## Casual

# reg1_2h2_casual = sm.OLS(endog=bal['pchemp'],
#                           exog=sm.add_constant(bal[['pchwage'] + controls])).
 ↪fit()
# ## First Stage

# reg1_2h2_fs = sm.OLS(endog=bal['pchwage'], exog=sm.add_constant(bal[['gap'] +␣
 ↪controls])).fit()
```

```python
# ## Reduced Form

# reg1_2h2_rf = sm.OLS(endog=bal['pchemp'], exog=sm.add_constant(bal[['gap'] +
↪controls])).fit()

# ## IV

# exog_1_2_h2 = sm.add_constant(bal[['pchwage'] + controls])
# ins_1_2_h2 = sm.add_constant(bal[['gap'] + controls])
# result_1_2_iv_h2 = IV2SLS(endog = bal['pchemp'],
#                      exog = exog_1_2_h2, instrument = ins_1_2_h2).fit()
```

0.1.1 OLS estimate

[27]:
```python
dml_data = DoubleMLData(data, y_col = 'pchemp', d_cols = ['pchwage'], x_cols =
↪controls)
```

[28]:
```python
print(dml_data)
```

```
=== DoubleMLData Object ===
y_col: pchemp
d_cols: ['pchwage']
x_cols: ['co_owned', 'southj', 'centralj', 'northj', 'pa2', 'shore', 'ncalls',
'bonus', 'open', 'hrsopen', 'psoda', 'pfry', 'pentree', 'nregs', 'nregs11',
'bk', 'kfc', 'roys', 'wendys', 'atmin', 'atnewmin2', 'freemeal', 'const']
z_cols: None
data:
 <class 'pandas.core.frame.DataFrame'>
RangeIndex: 351 entries, 0 to 350
Columns: 34 entries, const to freemeal
dtypes: float64(17), int64(17)
memory usage: 93.4 KB
```

[42]:
```python
plr = DoubleMLPLR(dml_data, ml_g, ml_m, n_folds = 10, dml_procedure = 'dml2',
↪score = 'partialling out')
```

[43]:
```python
plr.fit()
```

[43]:
```
<doubleml.double_ml_plr.DoubleMLPLR at 0x7fc6b3663a90>
```

[44]:
```python
print(plr)
```

```
================== DoubleMLPLR Object ==================

------------------ Data summary      ------------------
```

```
Outcome variable: pchemp
Treatment variable(s): ['pchwage']
Covariates: ['co_owned', 'southj', 'centralj', 'northj', 'pa2', 'shore',
'ncalls', 'bonus', 'open', 'hrsopen', 'psoda', 'pfry', 'pentree', 'nregs',
'nregs11', 'bk', 'kfc', 'roys', 'wendys', 'atmin', 'atnewmin2', 'freemeal',
'const']
Instrument variable(s): None
No. Observations: 351

------------------ Score & algorithm ------------------
Score function: partialling out
DML algorithm: dml2

------------------ Machine learner   ------------------
Learner ml_g: LassoCV(cv=10)
Learner ml_m: LassoCV(cv=10)

------------------ Resampling        ------------------
No. folds: 10
No. repeated sample splits: 1
Apply cross-fitting: True

------------------ Fit summary       ------------------
             coef   std err         t     P>|t|     2.5 %    97.5 %
pchwage  0.258859  0.295362  0.876412  0.380806 -0.320041  0.837758
```

0.1.2 Reduced Form

[32]:
```
learner_2 = LassoCV(cv=10)
ml_g_2 = clone(learner_2)
ml_m_2 = clone(learner_2)
```

[33]:
```
dml_data_2 = DoubleMLData(data, y_col = 'pchemp', d_cols = ['gap'], x_cols =␣
↪controls)
```

[34]:
```
plr2 = DoubleMLPLR(dml_data_2, ml_g_2, ml_m_2, n_folds = 10, dml_procedure =␣
↪'dml2', score = 'partialling out')
```

[35]:
```
plr2.fit()
```

[35]:
```
<doubleml.double_ml_plr.DoubleMLPLR at 0x7fc6b3663790>
```

[36]:
```
print(plr2)
```

```
================== DoubleMLPLR Object ==================

------------------ Data summary      ------------------
Outcome variable: pchemp
```

```
Treatment variable(s): ['gap']
Covariates: ['co_owned', 'southj', 'centralj', 'northj', 'pa2', 'shore',
'ncalls', 'bonus', 'open', 'hrsopen', 'psoda', 'pfry', 'pentree', 'nregs',
'nregs11', 'bk', 'kfc', 'roys', 'wendys', 'atmin', 'atnewmin2', 'freemeal',
'const']
Instrument variable(s): None
No. Observations: 351

------------------ Score & algorithm ------------------
Score function: partialling out
DML algorithm: dml2

------------------ Machine learner   ------------------
Learner ml_g: LassoCV(cv=10)
Learner ml_m: LassoCV(cv=10)

------------------ Resampling        ------------------
No. folds: 10
No. repeated sample splits: 1
Apply cross-fitting: True

------------------ Fit summary       ------------------
         coef   std err         t     P>|t|     2.5 %    97.5 %
gap  0.510172  0.408197  1.249819  0.211366 -0.289879  1.310224
```

0.2 First Stage

[37]:
```python
learner_3 = LassoCV(cv=10)
ml_g_3 = clone(learner_3)
ml_m_3 = clone(learner_3)
```

[38]:
```python
dml_data_3 = DoubleMLData(data, y_col = 'pchwage', d_cols = ['gap'], x_cols =␣
 ↪controls)
```

[39]:
```python
plr3 = DoubleMLPLR(dml_data_3, ml_g_3, ml_m_3, n_folds = 10, dml_procedure =␣
 ↪'dml2', score = 'partialling out')
```

[40]:
```python
plr3.fit()
```

[40]: <doubleml.double_ml_plr.DoubleMLPLR at 0x7fc6b3663df0>

[41]:
```python
print(plr3)
```

```
================== DoubleMLPLR Object ==================

------------------ Data summary      ------------------
Outcome variable: pchwage
Treatment variable(s): ['gap']
```

```
Covariates: ['co_owned', 'southj', 'centralj', 'northj', 'pa2', 'shore',
'ncalls', 'bonus', 'open', 'hrsopen', 'psoda', 'pfry', 'pentree', 'nregs',
'nregs11', 'bk', 'kfc', 'roys', 'wendys', 'atmin', 'atnewmin2', 'freemeal',
'const']
Instrument variable(s): None
No. Observations: 351

------------------ Score & algorithm ------------------
Score function: partialling out
DML algorithm: dml2

------------------ Machine learner   ------------------
Learner ml_g: LassoCV(cv=10)
Learner ml_m: LassoCV(cv=10)

------------------ Resampling        ------------------
No. folds: 10
No. repeated sample splits: 1
Apply cross-fitting: True

------------------ Fit summary       ------------------
         coef   std err          t         P>|t|     2.5 %    97.5 %
gap  0.962823  0.073703  13.063528  5.320368e-39  0.818368  1.107279
```

[ ]:

Final Project 2

May 13, 2021

0.1 code appendix 3

[1]:
```
import sys
!{sys.executable} -m pip install stargazer
import pandas as pd
import statsmodels.api as sm
from patsy import dmatrices
import numpy as np
from statsmodels.sandbox.regression.gmm import IV2SLS
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LassoCV
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from stargazer.stargazer import Stargazer, LineLocation
from IPython.core.display import HTML


import seaborn as sns
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (12,6)

import warnings
warnings.filterwarnings('ignore')
```

```
Requirement already satisfied: stargazer in /opt/conda/lib/python3.8/site-
packages (0.0.5)
```

[2]:
```
prd = pd.read_csv('projectrd.csv')
# prd.head()
```

[3]:
```
rd = prd.fillna(prd.mean())
rd.describe()
```

[3]:
```
              REGION           EDUC         female           age4  \
count  153782.000000  153782.000000  153782.000000  153782.000000
mean        2.565333      12.036279       0.541130      64.241429
std         1.024521       3.493988       0.498307       5.695231
```

```
min          1.000000       0.000000       0.000000      55.250000
25%          2.000000      11.000000       0.000000      59.250000
50%          3.000000      12.000000       1.000000      64.000000
75%          3.000000      14.000000       1.000000      69.000000
max          4.000000      20.000000       1.000000      74.750000

            hispanic            wnh            bnh            onh  \
count  153782.000000  153782.000000  153782.000000  153782.000000
mean        0.105643       0.750732       0.113323       0.030303
std         0.307381       0.432591       0.316988       0.171419
min         0.000000       0.000000       0.000000       0.000000
25%         0.000000       1.000000       0.000000       0.000000
50%         0.000000       1.000000       0.000000       0.000000
75%         0.000000       1.000000       0.000000       0.000000
max         1.000000       1.000000       1.000000       1.000000

              inhosp          sawdr  …          mcare         health  \
count  153782.000000  153782.000000  …  153782.000000  153782.000000
mean        0.126250       0.850757  …       0.450423       2.646071
std         0.332038       0.295136  …       0.497538       1.157518
min         0.000000       0.000000  …       0.000000       1.000000
25%         0.000000       0.850757  …       0.000000       2.000000
50%         0.000000       1.000000  …       0.000000       3.000000
75%         0.000000       1.000000  …       1.000000       3.000000
max         1.000000       1.000000  …       1.000000       5.000000

                   r              z            r_z        dropout  \
count  153782.000000  153782.000000  153782.000000  153782.000000
mean       -0.758571       0.453486       2.113147       0.281769
std         5.695231       0.497833       3.015870       0.449863
min        -9.750000       0.000000      -0.000000       0.000000
25%        -5.750000       0.000000      -0.000000       0.000000
50%        -1.000000       0.000000      -0.000000       0.000000
75%         4.000000       1.000000       4.000000       1.000000
max         9.750000       1.000000       9.750000       1.000000

            somecoll        college        covered       vghealth
count  153782.000000  153782.000000  153782.000000  153782.000000
mean        0.185015       0.180028       0.928633       0.454351
std         0.388311       0.384212       0.257438       0.497913
min         0.000000       0.000000       0.000000       0.000000
25%         0.000000       0.000000       1.000000       0.000000
50%         0.000000       0.000000       1.000000       0.000000
75%         0.000000       0.000000       1.000000       1.000000
max         1.000000       1.000000       1.000000       1.000000

[8 rows x 21 columns]
```

[4]:
```
collapsed = rd.groupby(by = 'age4').mean()
```

[5]:
```
collapsed = collapsed.reset_index()
collapsed
```

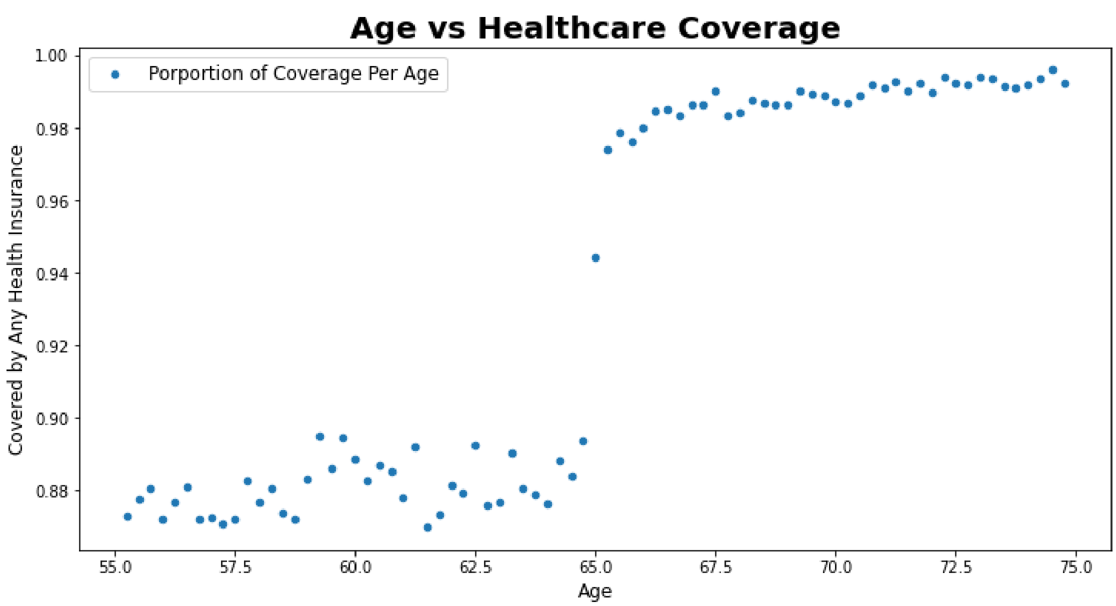
[5]:
```
     age4    REGION       EDUC    female  hispanic       wnh       bnh  \
0   55.25  2.576291  12.644757  0.526213  0.132629  0.716354  0.110720
1   55.50  2.601950  12.739643  0.520715  0.118197  0.724208  0.127945
2   55.75  2.586288  12.866036  0.510244  0.113475  0.728920  0.120567
3   56.00  2.603692  12.793401  0.516104  0.122152  0.723488  0.119010
4   56.25  2.592502  12.665037  0.517930  0.117359  0.730236  0.116952
..    …         …          …         …         …         …         …
74  73.75  2.571702  11.507967  0.574251  0.082218  0.788400  0.101976
75  74.00  2.511692  11.389959  0.585970  0.086657  0.786107  0.100413
76  74.25  2.549020  11.236601  0.556209  0.086928  0.771895  0.108497
77  74.50  2.574526  11.467480  0.569783  0.094173  0.775068  0.107046
78  74.75  2.584595  11.514154  0.591178  0.082949  0.812377  0.086899

         onh    inhosp     sawdr  …     mcare    health     r    z   r_z  \
0   0.040297  0.083823  0.818540  …  0.047731  2.453647 -9.75  0.0  0.00
1   0.029651  0.093219  0.818578  …  0.041430  2.477610 -9.50  0.0  0.00
2   0.037037  0.085994  0.825475  …  0.044917  2.437952 -9.25  0.0  0.00
3   0.035350  0.096622  0.817265  …  0.042419  2.446951 -9.00  0.0  0.00
4   0.035452  0.081602  0.821647  …  0.048492  2.464522 -8.75  0.0  0.00
..       …         …         …  …       …         …    …    …    …
74  0.027406  0.174714  0.882578  …  0.942001  2.832564  8.75  1.0  8.75
75  0.026823  0.176153  0.879471  …  0.951169  2.797757  9.00  1.0  9.00
76  0.032680  0.175900  0.886365  …  0.949673  2.885540  9.25  1.0  9.25
77  0.023713  0.182335  0.899702  …  0.942412  2.828465  9.50  1.0  9.50
78  0.017775  0.190506  0.896026  …  0.953917  2.855553  9.75  1.0  9.75

     dropout  somecoll   college   covered  vghealth
0   0.207746  0.213224  0.235524  0.872848  0.535211
1   0.199431  0.233550  0.225833  0.877742  0.518684
2   0.200552  0.215524  0.249409  0.880615  0.544917
3   0.197958  0.225844  0.241948  0.871956  0.541241
4   0.206601  0.211899  0.236349  0.876936  0.517930
..       …         …         …         …         …
74  0.363926  0.147228  0.152964  0.991077  0.373486
75  0.359697  0.162311  0.143741  0.991747  0.386520
76  0.364706  0.152288  0.122222  0.993464  0.358824
77  0.341463  0.164634  0.132114  0.995935  0.376694
78  0.350230  0.169190  0.147465  0.992100  0.372614

[79 rows x 21 columns]
```

[6]:
```
# plt.plot(bws, pi_1s, label = 'Pi')
# plt.plot(bws, pi_1s + 2 * std_ers, '--', c = 'orange', label = '+2 SE')
# plt.plot(bws, pi_1s - 2 * std_ers, '--', c = 'orange', label = '-2 SE')
# plt.title('Estimates of Beta Each Bandwich (Linear Model on sawdr)', size =␣
 ↪20, fontweight="bold")
# plt.xlabel('Bandwidth', size = 12)
# plt.ylabel('Beta', size = 12, rotation = 0)
# plt.legend(prop={"size":12})
```

0.1.1 plots

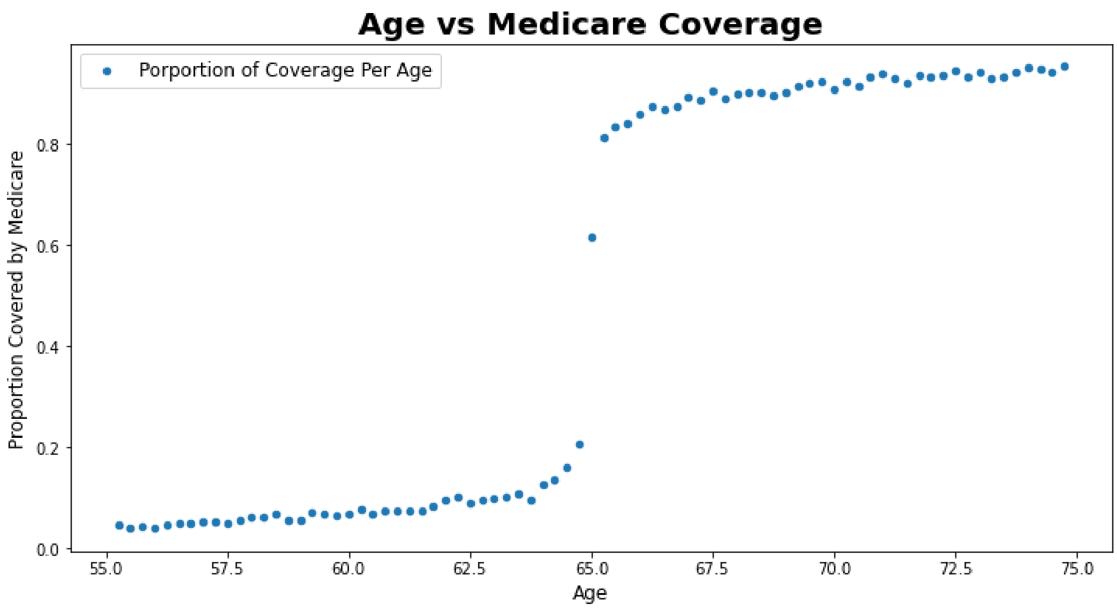
[7]:
```
collapsed.plot('age4', 'covered', kind = 'scatter')
plt.title('Age vs Healthcare Coverage', size = 20, fontweight="bold")
plt.ylabel('Covered by Any Health Insurance', size = 12)
plt.xlabel('Age', size = 12)
plt.legend(['Porportion of Coverage Per Age'], prop={"size":12})
```

[7]:
```
<matplotlib.legend.Legend at 0x7fefc026a1c0>
```



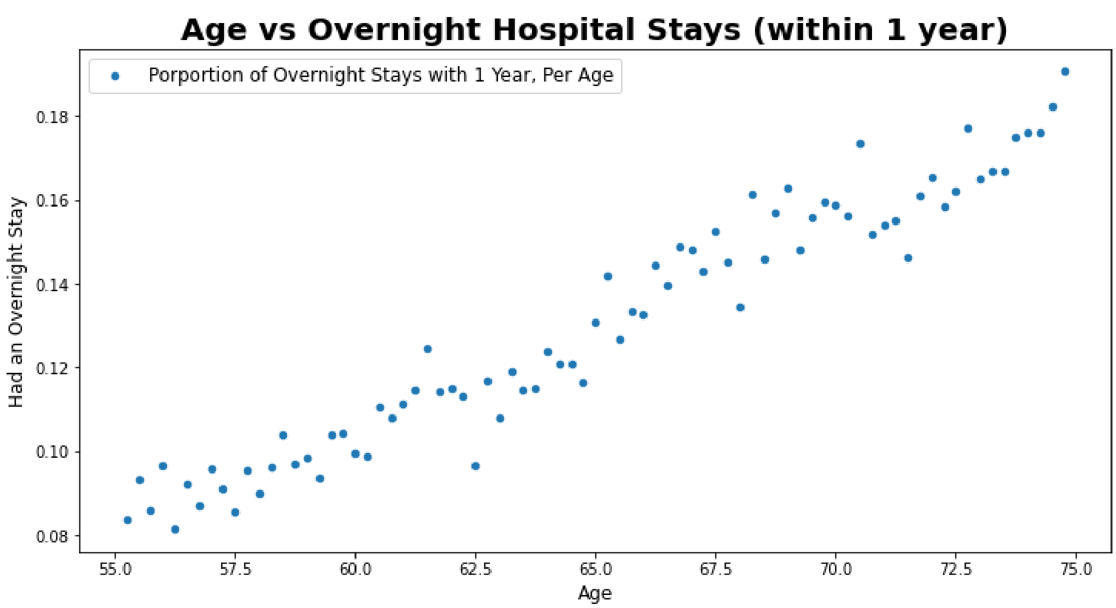
[8]:
```
collapsed.plot('age4', 'mcare', kind = 'scatter')
plt.title('Age vs Medicare Coverage', size = 20, fontweight="bold")
plt.ylabel('Proportion Covered by Medicare', size = 12)
plt.xlabel('Age', size = 12)
plt.legend(['Porportion of Coverage Per Age'], prop={"size":12})
```

[8]:
```
<matplotlib.legend.Legend at 0x7fef38049ca0>
```



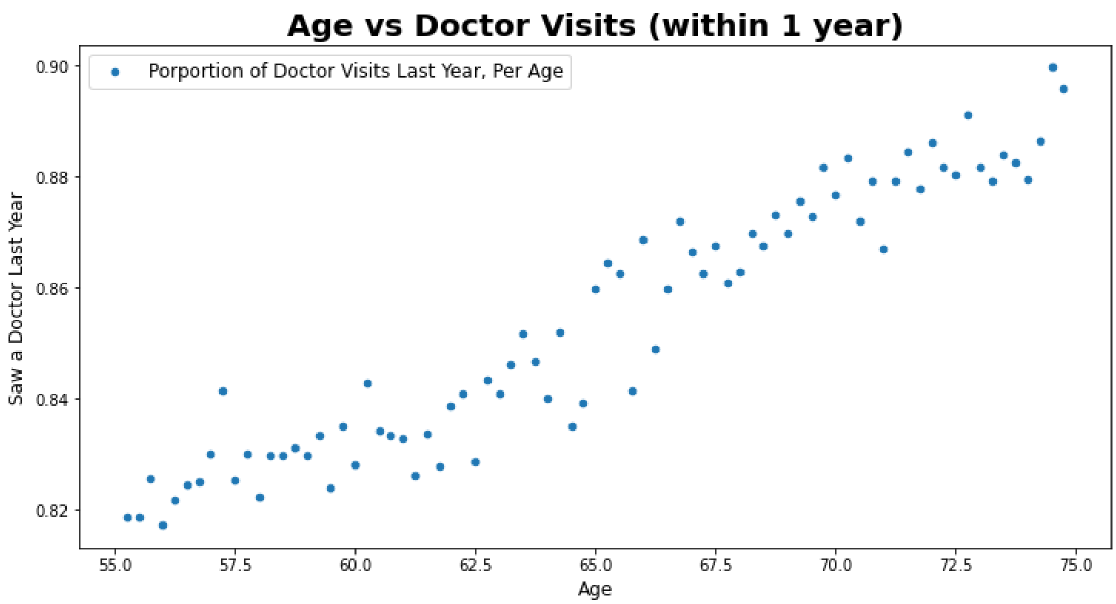

```
[9]: collapsed.plot('age4', 'inhosp', kind = 'scatter')
plt.title('Age vs Overnight Hospital Stays (within 1 year)', size = 20,␣
 ↪fontweight="bold")
plt.ylabel('Had an Overnight Stay', size = 12)
plt.xlabel('Age', size = 12)
plt.legend(['Porportion of Overnight Stays with 1 Year, Per Age'], prop={"size":
 ↪12})
```

```
[9]: <matplotlib.legend.Legend at 0x7fef3804f610>
```



[10]:
```
collapsed.plot('age4', 'sawdr', kind = 'scatter')
plt.title('Age vs Doctor Visits (within 1 year)', size = 20, fontweight="bold")
plt.ylabel('Saw a Doctor Last Year', size = 12)
plt.xlabel('Age', size = 12)
plt.legend(['Porportion of Doctor Visits Last Year, Per Age'], prop={"size":12})
```

[10]: <matplotlib.legend.Legend at 0x7fef36d94df0>



[11]:
```
reg_i = sm.OLS(endog=rd['covered'], exog=sm.add_constant(rd[['z', 'r',␣
↪'r_z']])).fit()
results_i = Stargazer([reg_i])
```

0.1.2 regression i results

[12]:
```
results_i
```

[12]: <stargazer.stargazer.Stargazer at 0x7fef36cf7f10>

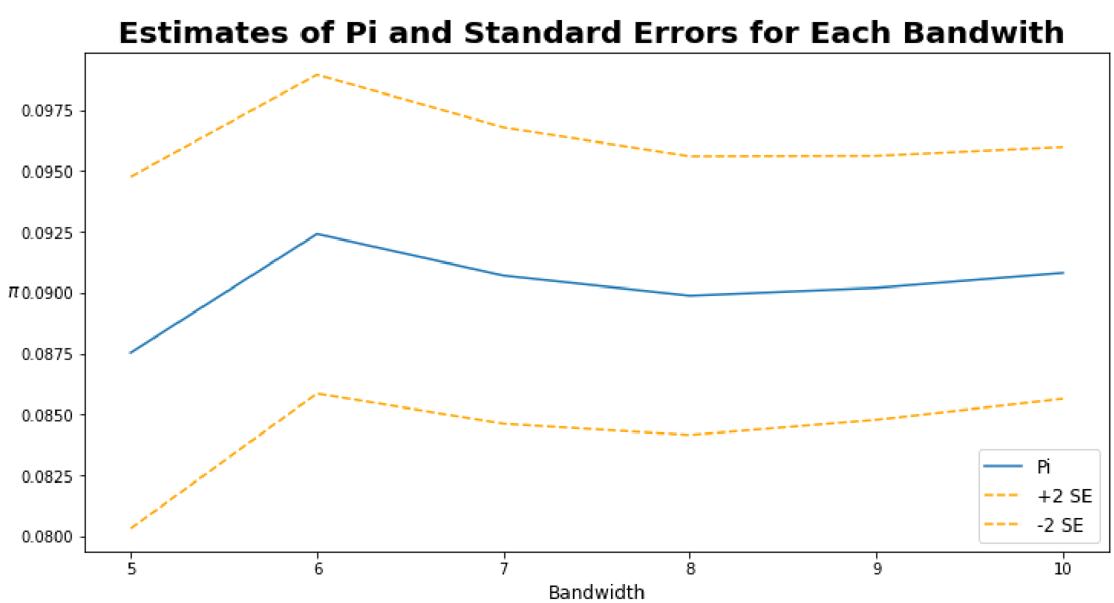
0.1.3 figure

[13]:
```
pi_1s = []
std_ers = []
bws = np.arange(5, 11, 1)
```

```
for interval in bws:

    data = rd.copy()
    bw_data = data[(data['age4'] >= 65 - interval + 0.25) & (data['age4'] <= 65␣
↪+ interval - 0.25)]

    reg = sm.OLS(endog=bw_data['covered'], exog=sm.add_constant(bw_data[['z',␣
↪'r', 'r_z']])).fit()

    pi_1s.append(reg.params[1])
    std_ers.append(reg.bse[1])
```

[14]:
```
pi_1s = np.array(pi_1s)
std_ers = np.array(std_ers)

plt.plot(bws, pi_1s, label = 'Pi')
plt.plot(bws, pi_1s + 2 * std_ers, '--', c = 'orange', label = '+2 SE')
plt.plot(bws, pi_1s - 2 * std_ers, '--', c = 'orange', label = '-2 SE')
plt.title('Estimates of Pi and Standard Errors for Each Bandwith ', size = 20,␣
↪fontweight="bold")
plt.xlabel('Bandwidth', size = 12)
plt.ylabel('$\pi$', size = 12, rotation = 0)
plt.legend(prop={"size":12})
```

[14]: <matplotlib.legend.Legend at 0x7fef362393d0>



0.1.4 regression c

[15]:
```
rd['r_2'] = rd['r']**2
rd['w_2'] = rd['r_2']*rd['z']
```

[16]:
```
reg_c = sm.OLS(endog=rd['covered'], exog=sm.add_constant(rd[['z', 'r', 'r_z',␣
↪'r_2', 'w_2']])).fit()
results_c = Stargazer([reg_c])
results_c
```

[16]: <stargazer.stargazer.Stargazer at 0x7fef36177b50>

0.1.5 validity

[17]:
```
rd['minority'] = np.array([(rd['bnh'] == 1) | (rd['hispanic'] == 1)]).T

results = []
features = ['college', 'wnh', 'bnh', 'hispanic', 'minority']

for feature in features:
    results.append(sm.OLS(endog=rd[feature],
                          exog=sm.add_constant(rd[['z', 'r', 'r_z']])).fit())
```

[18]:
```
results_v = Stargazer(results)
results_v
```

[18]: <stargazer.stargazer.Stargazer at 0x7fef36d0b160>

0.1.6 IV models

[19]:
```
## linear IV model on sawdr

exog_o_ll = sm.add_constant(rd[['covered', 'r', 'r_z']])
ins_o_ll = sm.add_constant(rd[['z','r', 'r_z']])
result_o_ll = IV2SLS(endog = rd['sawdr'],
                    exog = exog_o_ll, instrument = ins_o_ll).fit()

## linear IV model on inhosp

exog_o_ll_2 = sm.add_constant(rd[['covered', 'r', 'r_z']])
ins_o_ll_2 = sm.add_constant(rd[['z','r', 'r_z']])
result_o_ll_2 = IV2SLS(endog = rd['inhosp'],
                    exog = exog_o_ll_2, instrument = ins_o_ll_2).fit()


## quad IV model on sawdr
```

```
exog_o_lq = sm.add_constant(rd[['covered', 'r', 'r_z', 'r_2', 'w_2']])
ins_o_lq = sm.add_constant(rd[['z','r', 'r_z', 'r_2', 'w_2']])
result_o_lq = IV2SLS(endog = rd['sawdr'],
                     exog = exog_o_lq, instrument = ins_o_lq).fit()

## quad IV model on inhosp

exog_o_lq_2 = sm.add_constant(rd[['covered', 'r', 'r_z', 'r_2', 'w_2']])
ins_o_lq_2 = sm.add_constant(rd[['z','r', 'r_z', 'r_2', 'w_2']])
result_o_lq_2 = IV2SLS(endog = rd['inhosp'],
                     exog = exog_o_lq_2, instrument = ins_o_lq_2).fit()
```

[20]:
```
results_iv = Stargazer([result_o_ll, result_o_lq, result_o_ll_2, result_o_lq_2])
results_iv
```

[20]:
```
<stargazer.stargazer.Stargazer at 0x7fef361cf1c0>
```

Final Project 3

May 13, 2021

0.1 code appendix 4: open ended analysis

[1]:
```
import sys
!{sys.executable} -m pip install stargazer
import pandas as pd
import statsmodels.api as sm
from patsy import dmatrices
import numpy as np
from statsmodels.sandbox.regression.gmm import IV2SLS
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LassoCV
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import confusion_matrix
from stargazer.stargazer import Stargazer

import seaborn as sns
import matplotlib.pyplot as plt
plt.rcParams["figure.figsize"] = (12,6)

import warnings
warnings.filterwarnings('ignore')
```

```
Requirement already satisfied: stargazer in /opt/conda/lib/python3.8/site-
packages (0.0.5)
```

[2]:
```
prd = pd.read_csv('projectrd.csv')
rd = prd.fillna(prd.mean())
```

[3]:
```
rd['r_2'] = rd['r']**2
rd['w_2'] = rd['r_2']*rd['z']
```

[4]:
```
health_dummies = pd.get_dummies(rd['health'].astype(int)).rename(columns={1:␣
↪"Excellent",
                                                                 2: "Very Good",
                                                                 3: "Good",
                                                                 4: "Fair",
                                                                 5: "Poor"})
```

```python
w_dummies = rd.join(health_dummies)
w_dummies.head()
```

[4]:

```
   REGION  EDUC  female   age4  hispanic  wnh  bnh  onh  inhosp     sawdr  \
0       2     8       0  56.50         0    1    0    0     0.0  1.000000
1       4     6       0  65.25         1    0    0    0     0.0  0.000000
2       4     9       1  59.75         1    0    0    0     0.0  0.000000
3       4    18       1  61.25         1    0    0    0     0.0  0.850757
4       4    12       0  71.00         1    0    0    0     0.0  0.000000

   …  college  covered  vghealth      r_2      w_2  Excellent  Very Good  \
0  …        0        1         0  72.2500   0.0000          0          0
1  …        0        1         0   0.0625   0.0625          0          0
2  …        0        1         0  27.5625   0.0000          0          0
3  …        1        1         0  14.0625   0.0000          0          0
4  …        0        1         0  36.0000  36.0000          0          0

   Good  Fair  Poor
0     1     0     0
1     1     0     0
2     1     0     0
3     0     0     1
4     1     0     0

[5 rows x 28 columns]
```

[5]:

```python
w_dummies = w_dummies[~(w_dummies['age4'] == 65)]
```

[6]:

```python
conditions = ['Excellent', 'Very Good', 'Good', 'Fair']
## linear IV model on sawdr

exog_o_ll = sm.add_constant(w_dummies[['covered', 'r', 'r_z','emp'] +␣
 ↪conditions])
ins_o_ll = sm.add_constant(w_dummies[['z','r', 'r_z','emp'] + conditions])
result_o_ll = IV2SLS(endog = w_dummies['sawdr'],
                     exog = exog_o_ll, instrument = ins_o_ll).fit()

## linear IV model on inhosp

exog_o_ll_2 = sm.add_constant(w_dummies[['covered', 'r', 'r_z','emp'] +␣
 ↪conditions])
ins_o_ll_2 = sm.add_constant(w_dummies[['z','r', 'r_z','emp'] + conditions])
result_o_ll_2 = IV2SLS(endog = w_dummies['inhosp'],
                       exog = exog_o_ll_2, instrument = ins_o_ll_2).fit()
```

```python
## quad IV model on sawdr

exog_o_lq = sm.add_constant(w_dummies[['covered', 'r', 'r_z', 'r_2',␣
↪'w_2','emp'] + conditions])
ins_o_lq = sm.add_constant(w_dummies[['z','r', 'r_z', 'r_2', 'w_2','emp'] +␣
↪conditions])
result_o_lq = IV2SLS(endog = w_dummies['sawdr'],
                     exog = exog_o_lq, instrument = ins_o_lq).fit()

## quad IV model on inhosp

exog_o_lq_2 = sm.add_constant(w_dummies[['covered', 'r', 'r_z', 'r_2',␣
↪'w_2','emp'] + conditions])
ins_o_lq_2 = sm.add_constant(w_dummies[['z','r', 'r_z', 'r_2', 'w_2','emp'] +␣
↪conditions])
result_o_lq_2 = IV2SLS(endog = w_dummies['inhosp'],
                     exog = exog_o_lq_2, instrument = ins_o_lq_2).fit()
result_o_lq_2.summary()
```

[6]:

```
<class 'statsmodels.iolib.summary.Summary'>
"""
                          IV2SLS Regression Results
==============================================================================
Dep. Variable:                 inhosp   R-squared:                       0.067
Model:                         IV2SLS   Adj. R-squared:                  0.067
Method:                     Two Stage   F-statistic:                     1139.
                        Least Squares   Prob (F-statistic):               0.00
Date:                Thu, 13 May 2021
Time:                        17:52:28
No. Observations:              151842
Df Residuals:                  151831
Df Model:                          10
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.2326      0.051      4.565      0.000       0.133       0.332
covered        0.1545      0.055      2.788      0.005       0.046       0.263
r              0.0012      0.002      0.715      0.475      -0.002       0.004
r_z            0.0005      0.002      0.210      0.834      -0.004       0.005
r_2         4.413e-06      0.000      0.029      0.977      -0.000       0.000
w_2         7.753e-05      0.000      0.326      0.744      -0.000       0.001
emp           -0.0262      0.002    -12.208      0.000      -0.030      -0.022
Excellent     -0.3111      0.004    -76.788      0.000      -0.319      -0.303
Very Good     -0.2917      0.004    -76.158      0.000      -0.299      -0.284
Good          -0.2490      0.004    -70.467      0.000      -0.256      -0.242
Fair          -0.1582      0.004    -40.347      0.000      -0.166      -0.150
==============================================================================
```

```
Omnibus:                    54717.114   Durbin-Watson:                   1.989
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           150393.109
Skew:                           2.002   Prob(JB):                         0.00
Kurtosis:                       5.783   Cond. No.                         449.
==============================================================================
"""
```

[7]:
```
sg_table23 = Stargazer([result_o_ll, result_o_lq, result_o_ll_2, result_o_lq_2])
sg_table23.covariate_order(['covered'])
sg_table23.custom_columns(['Linear: sawdr', 'Quadratic: sawdr', 'Linear:␣
↪inhosp', 'Quadratic: inhosp'],[1,1,1,1])
sg_table23
```

[7]: <stargazer.stargazer.Stargazer at 0x7f0113000310>

[8]:
```
print(sg_table23.render_latex())
```

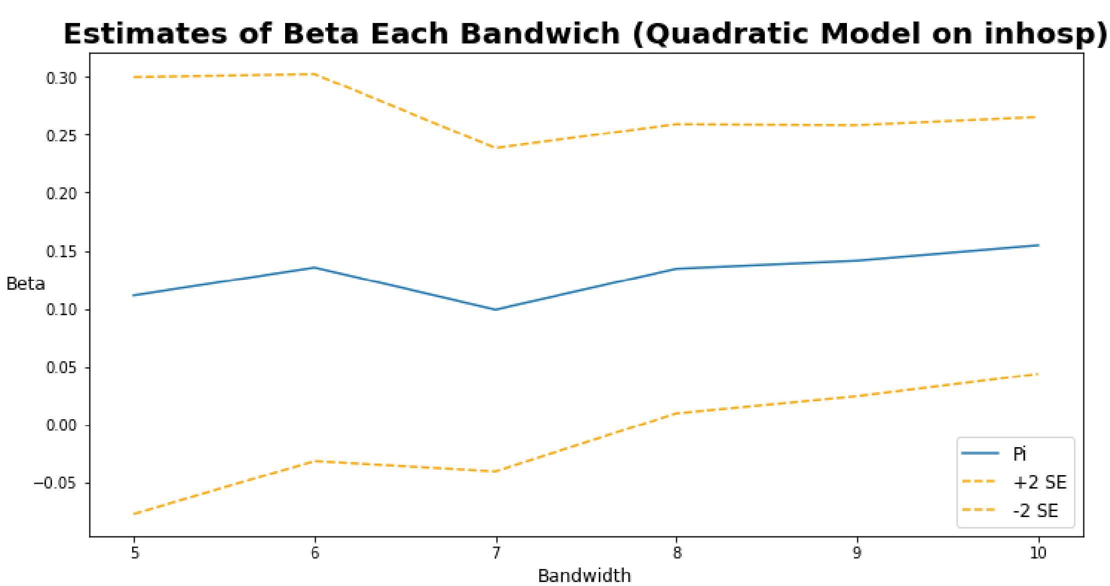
```
\begin{table}[!htbp] \centering
\begin{tabular}{@{\extracolsep{5pt}}lcccc}
\\[-1.8ex]\hline
\hline \\[-1.8ex]
\\[-1.8ex] & \multicolumn{1}{c}{Linear: sawdr} & \multicolumn{1}{c}{Quadratic:
sawdr} & \multicolumn{1}{c}{Linear: inhosp} & \multicolumn{1}{c}{Quadratic:
inhosp}  \\
\\[-1.8ex] & (1) & (2) & (3) & (4) \\
\hline \\[-1.8ex]
 covered & 0.129$^{***}$ & 0.126$^{**}$ & 0.141$^{***}$ & 0.155$^{***}$ \\
  & (0.032) & (0.050) & (0.035) & (0.055) \\
\hline \\[-1.8ex]
 Observations & 151,842 & 151,842 & 151,842 & 151,842 \\
 $R^2$ & 0.038 & 0.038 & 0.068 & 0.067 \\
 Adjusted $R^2$ & 0.038 & 0.038 & 0.068 & 0.067 \\
 Residual Std. Error & 0.290(df = 151833) & 0.290(df = 151831) & 0.320(df =
151833) & 0.321(df = 151831)  \\
 F Statistic & 390.866$^{***}$ (df = 8.0; 151833) & 312.662$^{***}$ (df = 10.0;
151831) & 1425.253$^{***}$ (df = 8.0; 151833) & 1138.588$^{***}$ (df = 10.0;
151831) \\
\hline
\hline \\[-1.8ex]
\textit{Note:} & \multicolumn{4}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05;
$^{***}$p$<$0.01} \\
\end{tabular}
\end{table}
```

[9]:
```
pi_1s = []
std_ers = []
bws = np.arange(5, 11, 1)
```

```
for interval in bws:

    data = w_dummies.copy()
    bw_data = data[(data['age4'] >= 65 - interval + 0.25) & (data['age4'] <= 65␣
 ↪+ interval - 0.25)]

    exog = sm.add_constant(bw_data[['covered', 'r', 'r_z', 'r_2', 'w_2','emp']␣
 ↪+ conditions])
    ins = sm.add_constant(bw_data[['z','r', 'r_z', 'r_2', 'w_2','emp'] +␣
 ↪conditions])
    result = IV2SLS(endog = bw_data['inhosp'],
                    exog = exog, instrument = ins).fit()

    pi_1s.append(result.params[1])
    std_ers.append(result.bse[1])
```

[10]:
```
pi_1s = np.array(pi_1s)
std_ers = np.array(std_ers)

plt.plot(bws, pi_1s, label = 'Pi')
plt.plot(bws, pi_1s + 2 * std_ers, '--', c = 'orange', label = '+2 SE')
plt.plot(bws, pi_1s - 2 * std_ers, '--', c = 'orange', label = '-2 SE')
plt.title('Estimates of Beta Each Bandwich (Quadratic Model on inhosp)', size =␣
 ↪20, fontweight="bold")
plt.xlabel('Bandwidth', size = 12)
plt.ylabel('Beta', size = 12, rotation = 0)
plt.legend(prop={"size":12})
```

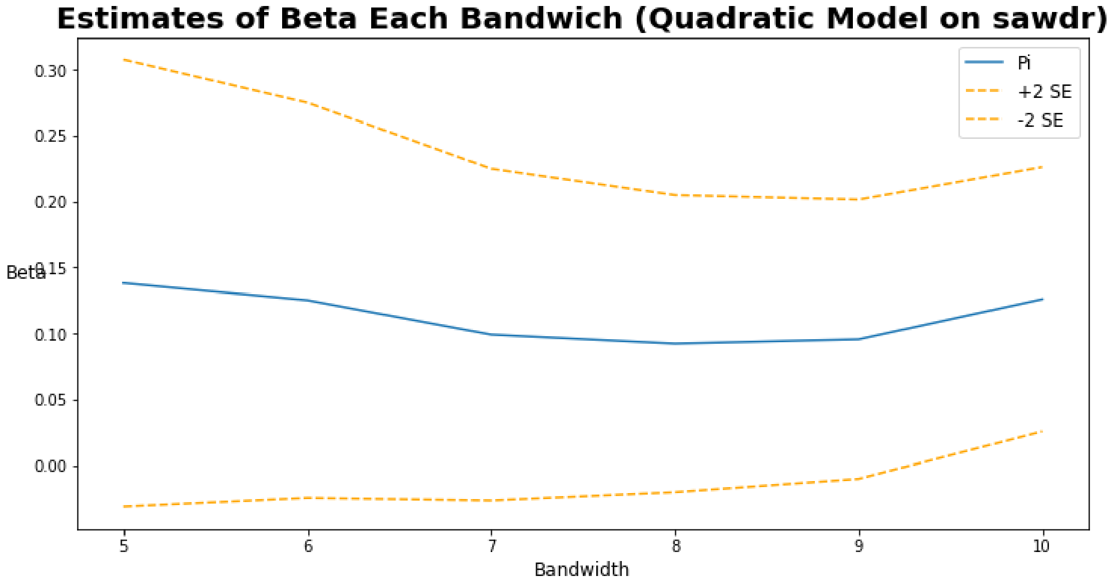
[10]: <matplotlib.legend.Legend at 0x7f0110c09490>



[11]:
```
pi_1s = []
std_ers = []
bws = np.arange(5, 11, 1)

for interval in bws:

    data = w_dummies.copy()
    bw_data = data[(data['age4'] >= 65 - interval + 0.25) & (data['age4'] <= 65␣
 ↪+ interval - 0.25)]

    exog = sm.add_constant(bw_data[['covered', 'r', 'r_z', 'r_2', 'w_2','emp']␣
 ↪+ conditions])
    ins = sm.add_constant(bw_data[['z','r', 'r_z', 'r_2', 'w_2','emp'] +␣
 ↪conditions])
    result = IV2SLS(endog = bw_data['sawdr'],
                    exog = exog, instrument = ins).fit()

    pi_1s.append(result.params[1])
    std_ers.append(result.bse[1])
```

[12]:
```
pi_1s = np.array(pi_1s)
std_ers = np.array(std_ers)

plt.plot(bws, pi_1s, label = 'Pi')
plt.plot(bws, pi_1s + 2 * std_ers, '--', c = 'orange', label = '+2 SE')
plt.plot(bws, pi_1s - 2 * std_ers, '--', c = 'orange', label = '-2 SE')
plt.title('Estimates of Beta Each Bandwich (Quadratic Model on sawdr)', size =␣
 ↪20, fontweight="bold")
plt.xlabel('Bandwidth', size = 12)
plt.ylabel('Beta', size = 12, rotation = 0)
plt.legend(prop={"size":12})
```

[12]:
```
<matplotlib.legend.Legend at 0x7f0110b28e20>
```


Estimates of Beta Each Bandwich (Quadratic Model on sawdr)
Beta
Bandwidth
Pi
+2 SE
-2 SE
0.30
0.25
0.20
0.15
0.10
0.05
0.00
5
6
7
8
9
10

```
[ ]:
```