Econometrics I Homework 1

Eric Schulman

February 10, 2018

1 Part I

2 Part II

a) Below are the results of the regression described in part (a).

Dep. Variable:	lwage	**R-squared:**	0.233
Model:	OLS	**Adj. R-squared:**	0.233
Method:	Least Squares	**F-statistic:**	428.8
Date:	Sat, 10 Feb 2018	**Prob (F-statistic):**	3.57e-243
Time:	16:31:36	**Log-Likelihood:**	-3651.2
No. Observations:	4230	**AIC:**	7310.
Df Residuals:	4226	**BIC:**	7336.
Df Model:	3		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	1.1852	0.045	26.488	0.000	1.097	1.273
educ	0.0904	0.003	33.051	0.000	0.085	0.096
exp	0.0354	0.003	14.083	0.000	0.030	0.040
exp_sq	-0.0005	5.03e-05	-9.251	0.000	-0.001	-0.000

Omnibus:	1562.688	**Durbin-Watson:**	1.770
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	34837.483
Skew:	-1.229	**Prob(JB):**	0.00
Kurtosis:	16.843	**Cond. No.**	4.31e+03

b) Summing the residuals in Python effectively produces 0 with some floating point error. My last program output was $-1.56521018368e - 10$.

Again, they residuals multiplied with X is effectively 0 with some floating point error. $\langle -1.56560986e{-}10 \quad -1.82083681e{-}09 \quad -3.00428837e{-}09 \quad -7.82019924e{-}08\rangle$

c) Below are the results of the regression described in part (c).

Dep. Variable:	lwage	**R-squared:**	0.035
Model:	OLS	**Adj. R-squared:**	0.035
Method:	Least Squares	**F-statistic:**	77.07
Date:	Sat, 10 Feb 2018	**Prob (F-statistic):**	1.33e-33
Time:	16:31:36	**Log-Likelihood:**	-4137.4
No. Observations:	4230	**AIC:**	8281.
Df Residuals:	4227	**BIC:**	8300.
Df Model:	2		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	2.3471	0.031	75.594	0.000	2.286	2.408
exp	0.0344	0.003	12.195	0.000	0.029	0.040
exp_sq	-0.0006	5.62e-05	-10.986	0.000	-0.001	-0.001

Omnibus:	965.638	**Durbin-Watson:**	1.692
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	16245.042
Skew:	-0.633	**Prob(JB):**	0.00
Kurtosis:	12.517	**Cond. No.**	2.67e+03

The average product of the residuals and education is 0.940. Obviously, this is non-zero as education was not included in the regression.

d) Below are the results of the regression described in part (d).

Dep. Variable:	educ	**R-squared:**	0.112
Model:	OLS	**Adj. R-squared:**	0.112
Method:	Least Squares	**F-statistic:**	267.1
Date:	Sat, 10 Feb 2018	**Prob (F-statistic):**	6.00e-110
Time:	16:31:36	**Log-Likelihood:**	-10954.
No. Observations:	4230	**AIC:**	2.191e+04
Df Residuals:	4227	**BIC:**	2.193e+04
Df Model:	2		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
const	12.8461	0.156	82.572	0.000	12.541	13.151
exp	-0.0113	0.014	-0.797	0.425	-0.039	0.016
exp_sq	-0.0017	0.000	-5.967	0.000	-0.002	-0.001

Omnibus:	65.618	**Durbin-Watson:**	1.674
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	96.128
Skew:	-0.174	**Prob(JB):**	1.34e-21
Kurtosis:	3.652	**Cond. No.**	2.67e+03

The average product of the new residuals and education is 10.396. One would expect the product to be greater than zero based on the proof below.

$$y'e$$

$$= y'y - y'\hat{y}$$
$$= (X'\hat{\beta} + \hat{e})'(X'\hat{\beta} + \hat{e}) - (X'\hat{\beta} + \hat{e})'(X'\hat{\beta})$$

Since X and $\hat{e}$ are orthogonal,

$$= (X'\hat{\beta})'(X'\hat{\beta}) + \hat{e}'\hat{e} - (X'\hat{\beta})'(X'\hat{\beta})$$
$$= \hat{e}'\hat{e} \geq 0$$

Since the matrix has quadratic form.

e) Below are the results form (e). As you can see the coefficients on the residuals are the same.

Dep. Variable:	y	**R-squared:**	0.205
Model:	OLS	**Adj. R-squared:**	0.205
Method:	Least Squares	**F-statistic:**	1093.
Date:	Sat, 10 Feb 2018	**Prob (F-statistic):**	2.01e-213
Time:	16:31:36	**Log-Likelihood:**	-3651.2
No. Observations:	4230	**AIC:**	7304.
Df Residuals:	4229	**BIC:**	7311.
Df Model:	1		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
x1	0.0904	0.003	33.063	0.000	0.085	0.096

Omnibus:	1562.688	**Durbin-Watson:**	1.770
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	34837.483
Skew:	-1.229	**Prob(JB):**	0.00
Kurtosis:	16.843	**Cond. No.**	1.00

f) In (c) we calculated $M_{X_1}y$.

In (d) we calculated $M_{X_1}X_2$.

In (e) we calculated $((M_{X_1}X_2)'M_{X_1}X_2)^{-1}((M_{X_1}X_2)'M_{X_1}y)$.

Adding a column of ι as an intercept we have:

$$(\begin{bmatrix} M_{X_1}X_2 & \iota \end{bmatrix}' \begin{bmatrix} M_{X_1}X_2 & \iota \end{bmatrix})^{-1}((\begin{bmatrix} M_{X_1}X_2 & \iota \end{bmatrix})'M_{X_1}y)$$

$$= \begin{bmatrix} (M_{X_1}X_2)'M_{X_1}X_2 & (M_{X_1}X_2)'\iota \\ \iota'M_{X_1}X_2 & \iota'\iota \end{bmatrix}^{-1} ((\begin{bmatrix} M_{X_1}X_2 & \iota \end{bmatrix})'M_{X_1}y)$$

ι should be orthogonal to $M_{X_1}X_2$ as it was included in X_1 so

$$= \begin{bmatrix} (M_{X_1}X_2)'M_{X_1}X_2 & 0 \\ 0 & \iota'\iota \end{bmatrix}^{-1} \begin{bmatrix} (M_{X_1}X_2)'M_{X_1}y \\ 0 \end{bmatrix}$$

$$= \begin{bmatrix} ((M_{X_1}X_2)'M_{X_1}X_2)^{-1}((M_{X_1}X_2)'M_{X_1}y) \\ 0 \end{bmatrix}$$

Since, M_{X_1} is idempotent and symmetric, this is equivalent to

$$= \begin{bmatrix} (X_2'M_{X_1}X_2)^{-1}X_2'M_{X_1}y \\ 0 \end{bmatrix}$$

Finally, we proved in lecture that the top term is the coefficient on X_2

$$= \begin{bmatrix} \hat{\beta}_2 \\ 0 \end{bmatrix}$$

3 Python Code

```
import pandas
import math
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm

FNAME = '../cps09mar.dta'

def write_result(model,no):
        result_doc = open('tex/hw2_reg%s.tex'%no, 'w+')
        result_doc.write( model.summary().as_latex() )
        result_doc.close()

def main():
        #filter matrix and load it into memory
        df = pandas.read_stata(FNAME)
        X = df[ (df.female == 0) & (df.hisp == 1) & (df.race == 1) ]
        print X['educ'].count()
        y = X['lwage']

        #part a
        X1 = X.loc[:,('educ','exp')]
        X1['exp_sq'] = X1['exp']**2
        X1 = sm.add_constant(X1)
        model1 = sm.OLS(y,X1).fit()
        write_result(model1,1)

        #part b
        e1 = model1.resid
        print sum(e1)
        print np.matmul(e1,X1)

        #part c
        X2 = X1.loc[:,('exp','exp_sq')]
        X2 = sm.add_constant(X2)
        model2 = sm.OLS(y,X2).fit()
        write_result(model2,2)

        #actual test part c
```

```
        e2 = model2.resid

        print np.matmul(e2,X1['educ'])/X1['educ'].count()

        #part d
        model3 = sm.OLS(X1['educ'],X2).fit()
        write_result(model3,3)
        e3 = model3.resid
        print np.matmul(e3,X1['educ'])/X1['educ'].count()

        #part e
        model4 = sm.OLS(e2,e3).fit()
        write_result(model4,4)

        #part f

if __name__ == "__main__":
        main()
```