Quadratics

Chap3_C10

C10 Use the data in HTV to answer this question. The data set includes information on wages, education, parents' education, and several other variables for 1,230 working men in 1991.

(i) What is the range of the *educ* variable in the sample? What percentage of men completed twelfth grade but no higher grade? Do the men or their parents have, on average, higher levels of education?

(ii) Estimate the regression model

$$educ = \beta_0 + \beta_1 motheduc + \beta_2 fatheduc + u$$

by OLS and report the results in the usual form. How much sample variation in *educ* is explained by parents' education? Interpret the coefficient on *motheduc*.

(iii) Add the variable *abil* (a measure of cognitive ability) to the regression from part (ii), and report the results in equation form. Does "ability" help to explain variations in education, even after controlling for parents' education? Explain.

(iv) (Requires calculus) Now estimate an equation where *abil* appears in quadratic form:

$$educ = \beta_0 + \beta_1 motheduc + \beta_2 fatheduc + \beta_3 abil + \beta_4 abil^2 + u.$$

Using the estimates $\hat{\beta}_3$ and $\hat{\beta}_4$, use calculus to find the value of *abil*, call it $abil^*$, where *educ* is minimized. (The other coefficients and values of parents' education variables have no effect; we are holding parents' education fixed.) Notice that *abil* is measured so that negative values are permissible. You might also verify that the second derivative is positive so that you do indeed have a minimum.

(v) Argue that only a small fraction of men in the sample have "ability" less than the value calculated in part (iv). Why is this important?

(vi) If you have access to a statistical program that includes graphing capabilities, use the estimates in part (iv) to graph the relationship between the predicted education and *abil*. Set *motheduc* and *fatheduc* at their average values in the sample, 12.18 and 12.45, respectively.

讀入資料

```python
#讀入HTV資料
import pandas as pd
import numpy as np
HTV= pd.read_csv("HTV.csv")
HTV.head()
```

	wage	abil	educ	ne	nc	west	south	exper	motheduc	fatheduc	...	ne18	nc18	south
0	12.019231	5.027738	15	0	0	1	0	9	12	12	...	1	0	
1	8.912656	2.037170	13	1	0	0	0	8	12	10	...	1	0	
2	15.514334	2.475895	15	1	0	0	0	11	12	16	...	1	0	
3	13.333333	3.609240	15	1	0	0	0	6	12	12	...	1	0	
4	11.070110	2.636546	13	1	0	0	0	15	12	15	...	1	0	

5 rows × 23 columns

Chap3_C10(1)變數educ涵蓋範圍

```python
from scipy import stats
def descriptive_statistics(x) :
    return pd.Series([x.count(),x.min(),x.max(),x.mean()],index=['count','min','max','mean'])
descriptive_statistics(HTV.educ)
```

```
count    1230.000000
min         6.000000
max        20.000000
mean       13.037398
dtype: float64
```

educ涵蓋範圍6~20，樣本數1230，平均數13.04

Chap3_C10(1)教育水準剛好12年級之百分比為何?

- Step1:篩選教育水準剛好為12年級

```
fliter = (HTV["educ"] == 12)
HTV[fliter]
```

7	11.667099	-0.133598	12	0	0	0	1	14	12
9	11.538462	-0.338460	12	1	0	0	0	9	14
10	14.814815	1.380710	12	1	0	0	0	13	9
11	20.699173	3.412799	12	1	0	0	0	14	12
25	11.057693	1.112235	12	1	0	0	0	8	9
...	...	...	...	...	...	...	...	...	...
1217	6.726458	3.715002	12	0	0	0	1	16	12
1218	3.301321	2.630618	12	1	0	0	0	9	12
1222	4.656578	1.757988	12	0	0	1	0	15	12
1224	9.615385	1.726616	12	0	1	0	0	9	12
1225	7.735584	2.803173	12	0	0	0	1	9	12

512 rows × 23 columns

Chap3_C10(1)教育水準剛好12年級之百分比為何?

- Step2:兩種算法算出百分比

```
#寫法1
print("The percentage of tewlfth grade",512/1230)
```

```
The percentage of tewlfth grade 0.4162601626016260
```

```
#寫法2
import statistics
mean = statistics.mean(fliter)
print("The percentage of tewlfth grade",mean)
```

```
The percentage of tewlfth grade 0.4162601626016260
```

Chap3_C10(1)平均而言這些工作者或是父母誰有較高教育水準?

- educ平均數為13.04
- 大於motheduc平均數12.18
- 大於fatheduc平均數12.45

```python
import statistics
mean = statistics.mean(HTV.motheduc)
mean1 = statistics.mean(HTV.fatheduc)
print("The average of the motheduc",mean,
      "The average of the fatheduc",mean1)
```

```
The average of the motheduc 12.178048780487805 The average of the fatheduc 12.447154471544716
```

Chap3_C10(2)估計迴歸模型

$$\widehat{educ} = 6.96 + .304\ motheduc + .190\ fatheduc$$
$$n = 1{,}230 \quad R^2 = .249.$$

```
import statsmodels.api as sm
# 迴歸分析 應變數是educ 自變數是motheduc fatheduc
pairf=pd.concat([HTV.motheduc,HTV.fatheduc],axis = 1)
model=sm.OLS(HTV.educ,sm.add_constant(pairf)).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   educ   R-squared:                       0.249
Model:                            OLS   Adj. R-squared:                  0.248
Method:                 Least Squares   F-statistic:                     203.7
Date:                Sun, 16 May 2021   Prob (F-statistic):           4.13e-77
Time:                        16:06:30   Log-Likelihood:                -2621.7
No. Observations:                1230   AIC:                             5249.
Df Residuals:                    1227   BIC:                             5265.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          6.9644      0.320     21.776      0.000       6.337       7.592
motheduc       0.3042      0.032      9.528      0.000       0.242       0.367
fatheduc       0.1903      0.022      8.539      0.000       0.147       0.234
==============================================================================
Omnibus:                       60.519   Durbin-Watson:                   1.748
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               82.103
```

Chap3_C10(3)加入abil

$$\widehat{educ} = 8.45 + .189\ motheduc + .111\ fatheduc + .502\ abil$$
$$n = 1{,}230 \quad R^2 = .428$$

```python
import statsmodels.api as sm
# 迴歸分析 應變數是educ 自變數是motheduc fatheduc abil
pairf=pd.concat([HTV.motheduc,HTV.fatheduc,HTV.abil],axis = 1)
model=sm.OLS(HTV.educ,sm.add_constant(pairf)).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   educ   R-squared:                       0.428
Model:                            OLS   Adj. R-squared:                  0.426
Method:                 Least Squares   F-statistic:                     305.2
Date:                Sun, 16 May 2021   Prob (F-statistic):          5.95e-148
Time:                        16:06:39   Log-Likelihood:                -2455.0
No. Observations:                1230   AIC:                             4918.
Df Residuals:                    1226   BIC:                             4938.
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          8.4487      0.290     29.180      0.000       7.881       9.017
motheduc       0.1891      0.029      6.635      0.000       0.133       0.245
fatheduc       0.1111      0.020      5.586      0.000       0.072       0.150
abil           0.5025      0.026     19.538      0.000       0.452       0.553
==============================================================================
Omnibus:                       52.055   Durbin-Watson:                   1.821
```

Chap3_C10(4)加入abil^2

$$\widehat{educ} = 8.24 + .190\, motheduc + .109\, fatheduc + .401\, abil + .051\, abil^2$$
$$n = 1{,}230 \quad R^2 = .444$$

```
abil=pd.concat([HTV.abil])
abilsqu=abil*abil
# 迴歸分析 應變數是educ 自變數是motheduc fatheduc abil abil^2
pairf=pd.concat([HTV.motheduc,HTV.fatheduc,HTV.abil,abilsqu],axis = 1)
model=sm.OLS(HTV.educ,sm.add_constant(pairf)).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   educ   R-squared:                       0.444
Model:                            OLS   Adj. R-squared:                  0.443
Method:                 Least Squares   F-statistic:                     244.9
Date:                Sun, 16 May 2021   Prob (F-statistic):          1.34e-154
Time:                        16:06:47   Log-Likelihood:                -2436.6
No. Observations:                1230   AIC:                             4883.
Df Residuals:                    1225   BIC:                             4909.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          8.2402      0.287     28.671      0.000       7.676       8.804
motheduc       0.1901      0.028      6.767      0.000       0.135       0.245
fatheduc       0.1089      0.020      5.558      0.000       0.070       0.147
abil           0.4015      0.030     13.255      0.000       0.342       0.461
abil           0.0506      0.008      6.093      0.000       0.034       0.067
==============================================================================
Omnibus:                       45.933   Durbin-Watson:                   1.820
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               56.769
Skew:                           0.404   Prob(JB):                     4.71e-13
Kurtosis:                       3.674   Cond. No.                         115.
==============================================================================
```

The derivative with respect to *abil* is $.401 + .102\, abil$. Setting equal to zero and solving gives

$$abil^* = -\frac{.401}{.102} \approx -3.93,$$

so about -4. The second derivative is $.102$, and so we know we have found the global minimum.

Chap3_C10(5)

(v) Argue that only a small fraction of men in the sample have "ability" less than the value calculated in part (iv). Why is this important?

(v) Out of 1,230 men, only 15 have $abil < -3.93$, or only about 1.2 percent of the sample. This is reassuring because it means we can effectively ignore what is happening to the left of -3.93. The important story is that the level of education increases with ability at an increasing rate.

Chap3_C10(6)畫出預測教育和abil之關係

- 將motheduc和fatheduc設成樣本平均值，分別為12.18和12.45

```
import statistics
mean_motheduc=statistics.mean(HTV.motheduc)
mean_fatheduc=statistics.mean(HTV.fatheduc)
print("motheduc平均數",round(mean_motheduc,2),"fatheduc平均數",round(mean_fatheduc,2)) #算到小數點第二位
```

```
motheduc平均數 12.18 fatheduc平均數 12.45
```

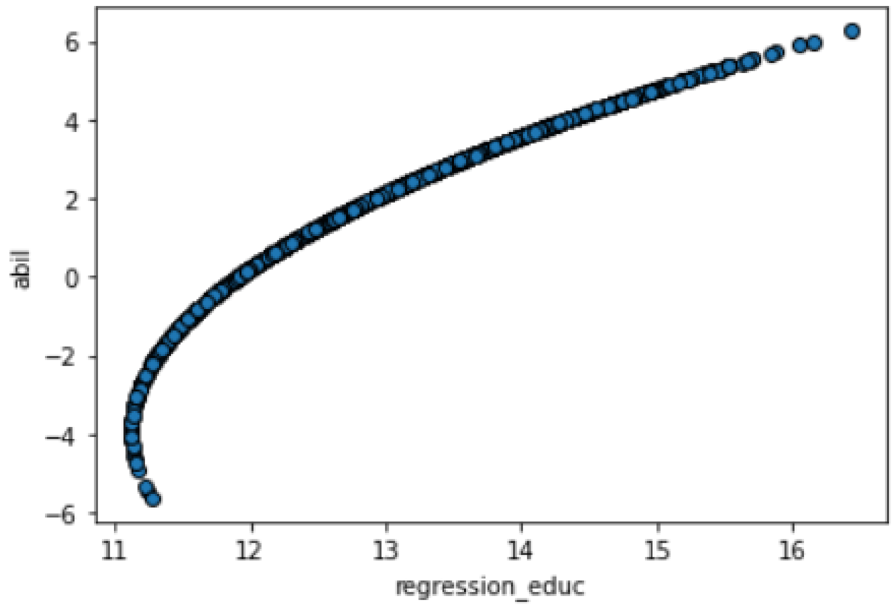
- 算出預測教育

```
regression_educ=8.24 +0.190*12.18+0.109*12.45+0.401*abil+0.051*abilsqu
regression_educ
```

```
0       15.216559
1       12.939809
2       13.216717
3       14.022912
4       13.323025
          ...
1225    13.436069
1226    14.465762
1227    12.310069
1228    11.677832
1229    12.482126
Name: abil, Length: 1230, dtype: float64
```

Chap3_C10(6)畫出預測教育和abil之關係

```
import matplotlib.pyplot as plt
fig = plt.figure
_ = plt.plot(regression_educ, abil, linestyle = "None", marker = "o", markeredgecolor = "black")
_ = plt.xlabel("regression_educ")
_ = plt.ylabel("abil")
plt.show()
```



Example6.2

EXAMPLE 6.2 **Effects of Pollution on Housing Prices**

We modify the housing price model from Example 4.5 to include a quadratic term in *rooms*:

$$\log(price) = \beta_0 + \beta_1\log(nox) + \beta_2\log(dist) + \beta_3 rooms + \beta_4 rooms^2 + \beta_5 stratio + u. \quad [6.14]$$

The model estimated using the data in HPRICE2 is

$$\begin{aligned}\widehat{\log(price)} = \underset{(.57)}{13.39} - \underset{(.115)}{.902}\log(nox) - \underset{(.043)}{.087}\log(dist) \\ - \underset{(.165)}{.545}\, rooms + \underset{(.013)}{.062}\, rooms^2 - \underset{(.006)}{.048}\, stratio \\ n = 506, R^2 = .603.\end{aligned}$$

The quadratic term $rooms^2$ has a t statistic of about 4.77, and so it is very statistically significant. But what about interpreting the effect of *rooms* on log(*price*)? Initially, the effect appears to be strange. Because the coefficient on *rooms* is negative and the coefficient on $rooms^2$ is positive, this equation literally implies that, at low values of *rooms*, an additional room has a *negative* effect on log(*price*). At some point, the effect becomes positive, and the quadratic shape means that the semi-elasticity of *price* with respect to *rooms* is increasing as *rooms* increases. This situation is shown in Figure 6.2.

We obtain the turnaround value of *rooms* using equation (6.13) (even though $\hat{\beta}_1$ is negative and $\hat{\beta}_2$ is positive). The absolute value of the coefficient on *rooms*, .545, divided by twice the coefficient on $rooms^2$, .062, gives $rooms^* = .545/[2(.062)] \approx 4.4$; this point is labeled in Figure 6.2.

Do we really believe that starting at three rooms and increasing to four rooms actually reduces a house's expected value? Probably not. It turns out that only five of the 506 communities in the sample

讀入資料

```
#讀入hprice2資料
import pandas as pd
import numpy as np
hprice= pd.read_csv("hprice2.csv")
hprice.head()
```

	price	crime	nox	rooms	dist	radial	proptax	stratio	lowstat	lprice	lnox	lproptax
0	24000	0.006	5.38	6.57	4.09	1	29.600000	15.300000	4.98	10.085809	1.682688	5.690360
1	21599	0.027	4.69	6.42	4.97	2	24.200001	17.799999	9.14	9.980402	1.545433	5.488938
2	34700	0.027	4.69	7.18	4.97	2	24.200001	17.799999	4.03	10.454495	1.545433	5.488938
3	33400	0.032	4.58	7.00	6.06	3	22.200001	18.700001	2.94	10.416311	1.521699	5.402678
4	36199	0.069	4.58	7.15	6.06	3	22.200001	18.700001	5.33	10.496787	1.521699	5.402678

```
#呼叫DataFrame內的price、nox、dist、rooms、stratio
price=pd.concat([hprice.price])
nox=pd.concat([hprice.nox])
dist=pd.concat([hprice.dist])
rooms=pd.concat([hprice.rooms])
stratio=pd.concat([hprice.stratio])
log_price=np.log(price)
log_nox=np.log(nox)
log_dist=np.log(dist)
rsqr=rooms*rooms
```

跑迴歸6.14

$$\Delta\widehat{\log(price)} \approx \{[-.545 + 2(.062)]rooms\}\Delta rooms$$

$$\%\Delta\widehat{price} \approx 100\{[-.545 + 2(.062)]rooms\}\Delta rooms$$
$$= (-54.5 + 12.4\ rooms)\Delta rooms.$$

```
# 迴歸分析 應變數是log_price 自變數是log_nox,log_dist,rooms,rsqr,stratio
pairf=pd.concat([log_nox,log_dist,rooms,rsqr,stratio],axis = 1)
model=sm.OLS(log_price,sm.add_constant(pairf)).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.603
Model:                            OLS   Adj. R-squared:                  0.599
Method:                 Least Squares   F-statistic:                     151.8
Date:                Fri, 23 Apr 2021   Prob (F-statistic):           7.89e-98
Time:                        08:36:03   Log-Likelihood:                -31.806
No. Observations:                 506   AIC:                             75.61
Df Residuals:                     500   BIC:                             101.0
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         13.3855      0.566     23.630      0.000      12.273      14.498
nox           -0.9017      0.115     -7.862      0.000      -1.127      -0.676
dist          -0.0868      0.043     -2.005      0.045      -0.172      -0.002
rooms         -0.5451      0.165     -3.295      0.001      -0.870      -0.220
rooms          0.0623      0.013      4.862      0.000       0.037       0.087
stratio       -0.0476      0.006     -8.129      0.000      -0.059      -0.036
==============================================================================
Omnibus:                       56.649   Durbin-Watson:                   0.691
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              384.168
Skew:                          -0.100   Prob(JB):                     3.79e-84
Kurtosis:                       7.264   Cond. No.                     2.30e+03
==============================================================================
```

```
#x=abs/ (beta_1/2*beta_2) /
x=abs(0.545/(2*0.062))
round(x,2)
```

```
4.4
```

```
#當rooms=6.45，效果為25.5%
delta_price=-54.5+12.4*6.45
#當rooms=7，效果為32.3%
delta_price_1=-54.5+12.4*7
print("delta_price",round(delta_price,1),"delta_price_1",delta_price_1)
```

```
delta_price 25.5 delta_price_1 32.3
```

Chap 6.C2

C2 Use the data in WAGE1 for this exercise.

(i) Use OLS to estimate the equation

$$\log(wage) = \beta_0 + \beta_1 educ + \beta_2 exper + \beta_3 exper^2 + u$$

and report the results using the usual format.

(ii) Is $exper^2$ statistically significant at the 1% level?

(iii) Using the approximation

$$\%\Delta\widehat{wage} \approx 100(\hat{\beta}_2 + 2\hat{\beta}_3 exper)\Delta exper,$$

find the approximate return to the fifth year of experience. What is the approximate return to the twentieth year of experience?

(iv) At what value of *exper* does additional experience actually lower predicted log(*wage*)? How many people have more experience in this sample?

讀入資料

```python
#讀入Wage1資料
import pandas as pd
import numpy as np
wage1= pd.read_csv("wage1.csv")
wage1.head()
```

	wage	educ	exper	tenure	nonwhite	female	married	numdep	smsa	northcen	...	trcommpu
0	3.10	11	2	0	0	1	0	2	1	0	...	0
1	3.24	12	22	2	0	1	1	3	1	0	...	0
2	3.00	11	2	0	0	0	0	2	0	0	...	0
3	6.00	8	44	28	0	0	1	0	1	0	...	0
4	5.30	12	7	2	0	0	1	1	0	0	...	0

5 rows × 24 columns

Chap 6.C2(1)估計迴歸

$$\widehat{\log(wage)} = \underset{(.106)}{.128} + \underset{(.0075)}{.0904}\, educ + \underset{(.0052)}{.0410}\, exper - \underset{(.000116)}{.000714}\, exper^2$$

$$n = 526,\quad R^2 = .300,\quad \bar{R}^2 = .296.$$

```
# 迴歸分析 應變數是log_wage 自變數是educ,exper,esqur
wage=pd.concat([wage1.wage])
exper=pd.concat([wage1.exper])
esqur=exper*exper
log_wage=np.log(wage)
pairf=pd.concat([wage1.educ,exper,esqur],axis = 1)
model=sm.OLS(log_wage,sm.add_constant(pairf)).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                   wage   R-squared:                       0.300
Model:                            OLS   Adj. R-squared:                  0.296
Method:                 Least Squares   F-statistic:                     74.67
Date:                Fri, 23 Apr 2021   Prob (F-statistic):           3.38e-40
Time:                        09:12:49   Log-Likelihood:                -319.53
No. Observations:                 526   AIC:                             647.1
Df Residuals:                     522   BIC:                             664.1
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.1280      0.106      1.208      0.227      -0.080       0.336
educ           0.0904      0.007     12.100      0.000       0.076       0.105
exper          0.0410      0.005      7.892      0.000       0.031       0.051
exper         -0.0007      0.000     -6.164      0.000      -0.001      -0.000
==============================================================================
Omnibus:                        5.379   Durbin-Watson:                   1.785
Prob(Omnibus):                  0.068   Jarque-Bera (JB):                7.152
Skew:                           0.028   Prob(JB):                       0.0280
Kurtosis:                       3.568   Cond. No.                     4.24e+03
==============================================================================
```

在1%水準下，顯著

Chap 6.C2(3)利用近似公式求5年與20年經驗

Using the approximation

$$\%\widehat{\Delta wage} \approx 100(\hat{\beta}_2 + 2\hat{\beta}_3 exper)\Delta exper,$$

```
#近似公式，%delta_wage=100(beta_2+2*beta_3*exper)*delta_exper
#%delta_wage=100(0.0410-2*0.000714*exper)*delta_exper
#求第5年經驗近似報酬，exper=4，增加exper=1
delta_wage_5year=100*(0.0410-2*0.000714*4)*1
delta_wage_5year
#求第20年經驗近似報酬，exper=19，增加exper=1
delta_wage_20year=100*(0.0410-2*0.000714*19)*1
print('delta_wage_5year',round(delta_wage_5year,2),'delta_wage_20year',round(delta_wage_20year,2))
```

```
delta_wage_5year 3.53 delta_wage_20year 1.39
```

Chap 6.C2(4)

```
#多1年經驗會降低預測log(wage)的exper值
#x=abs/ (beta_1/2*beta_2) /
x=abs(0.041/(2*0.000714))
print('reduce_log_wage_1_year',round(x,2))
```

```
reduce_log_wage_1_year 28.71
```

Chap 6.C2(4)

```python
fliter_wage = (wage1['exper'] >28.71)
wage1[fliter_wage]
#樣本中共121個人經驗比28.71高
```

	wage	educ	exper	tenure	nonwhite	female	married	numdep	smsa
3	6.000000	8	44	28	0	0	1	0	1
14	22.200001	12	31	15	0	0	1	1	1
19	4.500000	12	36	6	0	1	1	0	1
21	8.480000	12	29	13	0	0	1	3	1
24	6.000000	11	37	8	1	1	0	0	1
...	...	...	...	...	...	...	...	...	...
502	2.890000	0	42	0	0	1	1	2	0
503	2.900000	5	34	0	0	1	1	5	0
508	3.500000	12	31	3	1	1	0	1	1
510	3.000000	12	36	1	1	1	0	0	1
519	4.750000	13	47	1	0	0	1	0	0

121 rows × 24 columns

Chap 6.C4

C4 Use the data in GPA2 for this exercise.

(i) Estimate the model

$$sat = \beta_0 + \beta_1 hsize + \beta_2 hsize^2 + u,$$

where *hsize* is the size of the graduating class (in hundreds), and write the results in the usual form. Is the quadratic term statistically significant?

(ii) Using the estimated equation from part (i), what is the "optimal" high school size? Justify your answer.

(iii) Is this analysis representative of the academic performance of *all* high school seniors? Explain.

(iv) Find the estimated optimal high school size, using log(*sat*) as the dependent variable. Is it much different from what you obtained in part (ii)?

讀入資料

```python
#讀入gpa2資料
import pandas as pd
import numpy as np
gpa2= pd.read_csv("gpa2.csv")
gpa2.head()
```

	sat	tothrs	colgpa	athlete	verbmath	hsize	hsrank	hsperc	female	white
0	920	43	2.04	1	0.48387	0.10	4	40.000000	1	0
1	1170	18	4.00	0	0.82813	9.40	191	20.319149	0	1
2	810	14	1.78	1	0.88372	1.19	42	35.294117	0	1
3	940	40	2.42	0	0.80769	5.71	252	44.133099	0	1
4	1180	18	2.61	0	0.73529	2.14	86	40.186916	0	1

Chap 6.C4(1)估計迴歸式

```
# 迴歸分析 應變數是sat 自變數是hsize,hsqr
hsize=pd.concat([gpa2.hsize])
hsqr=hsize*hsize
pairf=pd.concat([hsize,hsqr],axis = 1)
model=sm.OLS(gpa2.sat,sm.add_constant(pairf)).fit()
print(model.summary())
```

$$\widehat{sat} = \underset{(6.20)}{997.98} + \underset{(3.99)}{19.81}\, hsize - \underset{(0.55)}{2.13}\, hsize^2$$

$$n = 4{,}137, \quad R^2 = .0076.$$

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    sat   R-squared:                       0.008
Model:                            OLS   Adj. R-squared:                  0.007
Method:                 Least Squares   F-statistic:                     15.93
Date:                Fri, 23 Apr 2021   Prob (F-statistic):           1.28e-07
Time:                        11:00:49   Log-Likelihood:                -26280.
No. Observations:                4137   AIC:                         5.257e+04
Df Residuals:                    4134   BIC:                         5.258e+04
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        997.9805      6.203    160.875      0.000     985.818    1010.143
hsize         19.8145      3.991      4.965      0.000      11.991      27.638
hsize         -2.1306      0.549     -3.881      0.000      -3.207      -1.054
==============================================================================
Omnibus:                        9.736   Durbin-Watson:                   1.956
Prob(Omnibus):                  0.008   Jarque-Bera (JB):               10.476
Skew:                           0.078   Prob(JB):                      0.00531
Kurtosis:                       3.191   Cond. No.                         56.6
==============================================================================
```

Chap 6.C4(2)何為最適高中大小
Chap 6.C4(3)此分析是否代表全部高中高年級生之學業表現?

```
#最適之高中大小
#x=abs/ (beta_1/2*beta_2) /
hsize=19.81/(2*2.13)
round(hsize,2)
```

```
4.65
```

- 樣本中僅顯示實際參加SAT考試的學生，因此它並不代表所有高中生。

Chap 6.C4(4)用log(sat)當應變數，找出最適高中大小

```
# 迴歸分析 應變數是log_sat 自變數是hsize,hsqr
hsize=pd.concat([gpa2.hsize])
sat=pd.concat([gpa2.sat])
log_sat=np.log(sat)
hsqr=hsize*hsize
pairf=pd.concat([hsize,hsqr],axis = 1)
model=sm.OLS(log_sat,sm.add_constant(pairf)).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                    sat   R-squared:                       0.008
Model:                            OLS   Adj. R-squared:                  0.007
Method:                 Least Squares   F-statistic:                     16.19
Date:                Fri, 23 Apr 2021   Prob (F-statistic):           9.89e-08
Time:                        13:12:19   Log-Likelihood:                 2332.6
No. Observations:                4137   AIC:                            -4659.
Df Residuals:                    4134   BIC:                            -4640.
Df Model:                           2
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const          6.8960      0.006   1121.032      0.000       6.884       6.908
hsize          0.0196      0.004      4.954      0.000       0.012       0.027
hsize         -0.0021      0.001     -3.834      0.000      -0.003      -0.001
==============================================================================
Omnibus:                      189.839   Durbin-Watson:                   1.952
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              277.089
Skew:                          -0.424   Prob(JB):                     6.78e-61
Kurtosis:                       3.942   Cond. No.                         56.6
==============================================================================
```

```
#最適之高中大小
#x=abs/ (beta_1/2*beta_2) /
hsize1=0.0196/(2*0.0021)
round(hsize1,2)
```

```
4.67
```

Example9.2

EXAMPLE 9.2 **Housing Price Equation**

We estimate two models for housing prices. The first one has all variables in level form:

$$price = \beta_0 + \beta_1 lotsize + \beta_2 sqrft + \beta_3 bdrms + u. \qquad [9.4]$$

The second one uses the logarithms of all variables except *bdrms*:

$$lprice = \beta_0 + \beta_1 llotsize + \beta_2 lsqrft + \beta_3 bdrms + u. \qquad [9.5]$$

Using $n = 88$ houses in HPRICE1, the RESET statistic for equation (9.4) turns out to be 4.67; this is the value of an $F_{2,82}$ random variable $(n = 88, k = 3)$, and the associated p-value is .012. This is evidence of functional form misspecification in (9.4).

The RESET statistic in (9.5) is 2.56, with p-value $= .084$. Thus, we do not reject (9.5) at the 5% significance level (although we would at the 10% level). On the basis of RESET, the log-log model in (9.5) is preferred.

讀入資料

```python
#讀入hpice1資料
import pandas as pd
import numpy as np
hprice= pd.read_csv("hprice1.csv")
hprice.head()
```

	price	assess	bdrms	lotsize	sqrft	colonial	lprice	lassess	llotsize	lsqrft
0	300.0	349.100006	4	6126	2438	1	5.703783	5.855359	8.720297	7.798934
1	370.0	351.500000	3	9903	2076	1	5.913503	5.862210	9.200593	7.638198
2	191.0	217.699997	3	5200	1374	0	5.252274	5.383118	8.556414	7.225482
3	195.0	231.800003	3	4600	1448	1	5.273000	5.445875	8.433811	7.277938
4	373.0	319.100006	4	6095	2514	1	5.921578	5.765504	8.715224	7.829630

跑迴歸

$$price = \beta_0 + \beta_1 lotsize + \beta_2 sqrft + \beta_3 bdrms + u.$$

```
# 迴歸分析 應變數price 自變數是lotsize,sqrft,bdrms
pairf=pd.concat([hprice.lotsize,hprice.sqrft,hprice.bdrms],axis = 1)
model=sm.OLS(hprice.price,sm.add_constant(pairf)).fit()
res1 = model
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.672
Model:                            OLS   Adj. R-squared:                  0.661
Method:                 Least Squares   F-statistic:                     57.46
Date:                Fri, 23 Apr 2021   Prob (F-statistic):           2.70e-20
Time:                        12:05:18   Log-Likelihood:                -482.88
No. Observations:                  88   AIC:                             973.8
Df Residuals:                      84   BIC:                             983.7
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        -21.7703     29.475     -0.739      0.462     -80.385      36.844
lotsize        0.0021      0.001      3.220      0.002       0.001       0.003
sqrft          0.1228      0.013      9.275      0.000       0.096       0.149
bdrms         13.8525      9.010      1.537      0.128      -4.065      31.770
==============================================================================
Omnibus:                       20.398   Durbin-Watson:                   2.110
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               32.278
Skew:                           0.961   Prob(JB):                     9.79e-08
Kurtosis:                       5.261   Cond. No.                     6.41e+04
==============================================================================
```

跑迴歸

$$lprice = \beta_0 + \beta_1 llotsize + \beta_2 lsqrft + \beta_3 bdrms + u.$$

```
# 迴歸分析 應變數log_price 自變數是log_lotsize,log_sqrft,bdrms
price=pd.concat([hprice.price])
lotsize=pd.concat([hprice.lotsize])
sqrft=pd.concat([hprice.sqrft])
bdrms=pd.concat([hprice.bdrms])
log_price=np.log(price)
log_lotsize=np.log(lotsize)
log_sqrft=np.log(sqrft)
pairf=pd.concat([log_lotsize,log_sqrft,bdrms],axis = 1)
model2=sm.OLS(log_price,sm.add_constant(pairf)).fit()
res2 = model2
print(model2.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  price   R-squared:                       0.643
Model:                            OLS   Adj. R-squared:                  0.630
Method:                 Least Squares   F-statistic:                     50.42
Date:                Fri, 23 Apr 2021   Prob (F-statistic):           9.74e-19
Time:                        12:13:29   Log-Likelihood:                 25.861
No. Observations:                  88   AIC:                            -43.72
Df Residuals:                      84   BIC:                            -33.81
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -1.2970      0.651     -1.992      0.050      -2.592      -0.002
lotsize        0.1680      0.038      4.388      0.000       0.092       0.244
sqrft          0.7002      0.093      7.540      0.000       0.516       0.885
bdrms          0.0370      0.028      1.342      0.183      -0.018       0.092
==============================================================================
```

Ramsey's RESET test

```
import statsmodels
statsmodels.stats.diagnostic.linear_reset(res1,power=3,
                                          test_type='fitted',use_f=bool,cov_type='nonrobust',cov_kwargs=None)
```

```
<class 'statsmodels.stats.contrast.ContrastResults'>
<F test: F=array([[4.66820553]]), p=0.012021711443144014, df_denom=82, df_num=2>
```

```
import statsmodels
statsmodels.stats.diagnostic.linear_reset(res2,power=3,
                                          test_type='fitted',use_f=bool,cov_type='nonrobust',cov_kwargs=None)
```

```
<class 'statsmodels.stats.contrast.ContrastResults'>
<F test: F=array([[2.56504079]]), p=0.08307588975322601, df_denom=82, df_num=2>
```

其他RESET作法

https://www.aptech.com/resources/tutorials/econometrics/ols-diagnostics-model-specification/#reset