exercise3_109301060

May 14, 2024

1 Exercises for Lecture 3

109301060

1.0.1 Q1

a.

[ ]:
```
import numpy as np
import pandas as pd
import statsmodels.formula.api as smf

data = pd.read_csv('data_exercise_3c.csv')
```

b.

[ ]:
```
data['DATE'] = pd.to_datetime(data['DATE'])

# Calculate simple return
AAPL_sr = np.diff(data['AAPL_Adj_Close'])/data['AAPL_Adj_Close'][:-1]

data['AAPL_sr'] = np.append(np.nan, AAPL_sr)

# Calculate excess return
data['AAPL_rp'] = data['AAPL_sr'] - data['RF']
```

c.

[ ]:
```
results_sst = data.loc[:,'AAPL_sr':'AAPL_rp'].describe()
print(results_sst)
```

```
          AAPL_sr     AAPL_rp
count  130.000000  130.000000
mean     0.024280   -0.017643
std      0.078486    0.102524
min     -0.184045   -0.364045
25%     -0.026441   -0.074593
50%      0.027021   -0.011113
75%      0.076592    0.060982
max      0.214380    0.204380
```

d.

```
[ ]: data = data.drop(data.index[0])

     capm = smf.ols(formula = 'AAPL_rp ~ Mkt_RF', data = data)
     result_capm = capm.fit()
     print(f'result_capm.summary():\n{result_capm.summary()}\n')
```

```
result_capm.summary():
                            OLS Regression Results
==============================================================================
Dep. Variable:                AAPL_rp   R-squared:                       0.239
Model:                            OLS   Adj. R-squared:                  0.233
Method:                 Least Squares   F-statistic:                     40.15
Date:                Tue, 14 May 2024   Prob (F-statistic):           3.67e-09
Time:                        01:21:14   Log-Likelihood:                 129.87
No. Observations:                 130   AIC:                            -255.7
Df Residuals:                     128   BIC:                            -250.0
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0310      0.008     -3.801      0.000      -0.047      -0.015
Mkt_RF         0.0122      0.002      6.336      0.000       0.008       0.016
==============================================================================
Omnibus:                       15.494   Durbin-Watson:                   1.015
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               18.915
Skew:                          -0.712   Prob(JB):                     7.81e-05
Kurtosis:                       4.210   Cond. No.                         4.41
==============================================================================

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

$\alpha = -0.0310$

$\beta_{MKT} = 0.0122$

e.

```
[ ]: ff3 = smf.ols(formula = 'AAPL_rp ~ Mkt_RF + SMB + HML', data = data)
     result_ff3 = ff3.fit()
     print(f'result_ff3.summary():\n{result_ff3.summary()}\n')
```

```
result_ff3.summary():
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:                AAPL_rp   R-squared:                       0.255
Model:                            OLS   Adj. R-squared:                  0.237
Method:                 Least Squares   F-statistic:                     14.37
Date:                Tue, 14 May 2024   Prob (F-statistic):           4.13e-08
Time:                        01:21:14   Log-Likelihood:                 131.26
No. Observations:                 130   AIC:                            -254.5
Df Residuals:                     126   BIC:                            -243.1
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0329      0.008     -4.008      0.000      -0.049      -0.017
Mkt_RF         0.0134      0.002      6.458      0.000       0.009       0.018
SMB           -0.0047      0.003     -1.377      0.171      -0.011       0.002
HML           -0.0013      0.003     -0.490      0.625      -0.006       0.004
==============================================================================
Omnibus:                       18.784   Durbin-Watson:                   0.959
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               24.644
Skew:                          -0.802   Prob(JB):                     4.45e-06
Kurtosis:                       4.406   Cond. No.                         4.65
==============================================================================

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

$\alpha = -0.0329$

$\beta_{MKT} = 0.0134$

$\beta_{SMB} = -0.0047$

$\beta_{HML} = -0.0013$

f.

```
[ ]: ff5 = smf.ols(formula = 'AAPL_rp ~ Mkt_RF + SMB + HML + RMW + CMA', data = data)
     result_ff5 = ff5.fit()
     print(f'result_ff5.summary():\n{result_ff5.summary()}\n')
```

```
result_ff5.summary():
                            OLS Regression Results
==============================================================================
Dep. Variable:                AAPL_rp   R-squared:                       0.307
Model:                            OLS   Adj. R-squared:                  0.279
Method:                 Least Squares   F-statistic:                     10.98
Date:                Tue, 14 May 2024   Prob (F-statistic):           9.20e-09
Time:                        01:21:15   Log-Likelihood:                 135.96
No. Observations:                 130   AIC:                            -259.9
```

```
Df Residuals:                     124   BIC:                            -242.7
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0348      0.008     -4.300      0.000      -0.051      -0.019
Mkt_RF         0.0122      0.002      5.724      0.000       0.008       0.016
SMB            0.0006      0.004      0.160      0.873      -0.007       0.008
HML           -0.0018      0.003     -0.562      0.575      -0.008       0.005
RMW            0.0140      0.005      3.011      0.003       0.005       0.023
CMA           -0.0034      0.006     -0.606      0.545      -0.015       0.008
==============================================================================
Omnibus:                       16.992   Durbin-Watson:                   0.951
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               21.040
Skew:                          -0.766   Prob(JB):                     2.70e-05
Kurtosis:                       4.239   Cond. No.                         4.78
==============================================================================

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

$\alpha = -0.0348$

$\beta_{MKT} = 0.0122$

$\beta_{SMB} = 0.0006$

$\beta_{HML} = -0.0018$

$\beta_{RMW} = 0.0140$

$\beta_{CMA} = -0.0034$

g.

[ ]:
```
ff6 = smf.ols(formula = 'AAPL_rp ~ Mkt_RF + SMB + HML + RMW + CMA + MOM', data␣
 ↪= data)
result_ff6 = ff6.fit()
print(f'result_ff6.summary():\n{result_ff6.summary()}\n')
```

```
result_ff6.summary():
                            OLS Regression Results
==============================================================================
Dep. Variable:                AAPL_rp   R-squared:                       0.312
Model:                            OLS   Adj. R-squared:                  0.279
Method:                 Least Squares   F-statistic:                     9.309
Date:                Tue, 14 May 2024   Prob (F-statistic):           2.08e-08
Time:                        01:21:15   Log-Likelihood:                 136.47
```

```
No. Observations:                 130   AIC:                            -258.9
Df Residuals:                     123   BIC:                            -238.9
Df Model:                           6
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0358      0.008     -4.391      0.000      -0.052      -0.020
Mkt_RF         0.0128      0.002      5.771      0.000       0.008       0.017
SMB            0.0012      0.004      0.314      0.754      -0.006       0.009
HML           -0.0005      0.004     -0.156      0.876      -0.008       0.006
RMW            0.0146      0.005      3.117      0.002       0.005       0.024
CMA           -0.0037      0.006     -0.651      0.516      -0.015       0.008
MOM            0.0027      0.003      0.989      0.325      -0.003       0.008
==============================================================================
Omnibus:                       15.389   Durbin-Watson:                   0.958
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               18.536
Skew:                          -0.716   Prob(JB):                     9.44e-05
Kurtosis:                       4.170   Cond. No.                         5.32
==============================================================================

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

$\alpha = -0.0358$

$\beta_{MKT} = 0.0128$

$\beta_{SMB} = 0.0012$

$\beta_{HML} = -0.005$

$\beta_{RMW} = 0.0146$

$\beta_{CMA} = -0.0037$

$\beta_{MOM} = 0.0027$

h.

[ ]:
```
capm_alpha = round(result_capm.params['Intercept'], 4)
ff3_alpha = round(result_ff3.params['Intercept'], 4)
ff5_alpha = round(result_ff5.params['Intercept'],4)
ff6_alpha = round(result_ff6.params['Intercept'], 4)

print('CAPM Alpha :',capm_alpha)
print('FF3 Alpha :',ff3_alpha)
print('FF5 Alpha :',ff5_alpha)
print('FF6 Alpha :',ff6_alpha)
```

```
CAPM Alpha : -0.031
FF3 Alpha : -0.0329
FF5 Alpha : -0.0348
FF6 Alpha : -0.0358
```

α α α

1.0.2 Q2

a.

```
[ ]: import wooldridge as woo
     import numpy as np
     import statsmodels.formula.api as smf

     wage2 = woo.data('WAGE2')

     reg_IQ = smf.ols(formula = 'IQ ~ educ', data = wage2)
     results_IQ = reg_IQ.fit()
     delta_educ_hat = results_IQ.params['educ']

     print(round(results_IQ.params, 4))
```

```
Intercept    53.6872
educ          3.5338
dtype: float64
```

b.

```
[ ]: reg1 = smf.ols(formula = 'np.log(wage) ~ educ', data = wage2)
     results1 = reg1.fit()
     beta_educ_tilde = results1.params['educ']

     print(round(results1.params, 4))
```

```
Intercept    5.9731
educ         0.0598
dtype: float64
```

c.

```
[ ]: reg = smf.ols(formula = 'np.log(wage) ~ educ + IQ', data = wage2)
     results = reg.fit()
     beta_hat = results.params

     print(beta_hat)
```

```
Intercept    5.658288
educ         0.039120
IQ           0.005863
dtype: float64
```

d.

```
[ ]: print('Value by regression:',beta_educ_tilde)
     beta_educ_tilde1 = beta_hat['educ']+beta_hat['IQ']*delta_educ_hat
     print('Value by equation :',round(beta_educ_tilde1, 4))
```

```
Value by regression: 0.05983920788637073
Value by equation : 0.0598
```

e.

```
[ ]: delta_int_hat = results_IQ.params['Intercept']
     beta_int_tilde = results1.params['Intercept']
     print('Value by regression:',beta_int_tilde)

     beta_int_tilde1 = results.params['Intercept'] + results.
      ↪params['IQ']*delta_int_hat
     print('Value by equation :',round(beta_int_tilde1, 4))
```

```
Value by regression: 5.973062450264955
Value by equation : 5.9731
```

1.0.3 Q3

```
[ ]: import statsmodels.api as sm
     import numpy as np
     import statsmodels.formula.api as smf

     # Redo the OLS estimation
     wage1 = woo.data('wage1')

     reg = smf.ols(formula='np.log(wage) ~ educ + exper + tenure', data = wage1)
     results = reg.fit()
     beta_educ_hat = results.params['educ']
     beta_educ_hat = round(beta_educ_hat, 4)

     print(f'beta_educ_hat: \n{beta_educ_hat}')

     # The first step: regress educ on exper and tenure with the OLS
     # and save the residuals
     reg_educ = smf.ols(formula='educ ~ exper + tenure', data = wage1)
     results_educ = reg_educ.fit()
     wage1['resid_educ'] = results_educ.resid

     # The second step: regress log(wage) on resid_educ with the OLS
     # and obtain the estimated slop parameter
     reg1 = smf.ols(formula='np.log(wage) ~ resid_educ', data = wage1)
     results1 = reg1.fit()
     beta_educ_hat1 = results1.params['resid_educ']
```

```python
beta_educ_hat1 = round(beta_educ_hat1, 4)

print(f'beta_educ_hat1: \n{beta_educ_hat1}')

# Is this result applied to the estimated intercept?
# The first step: regress a vector of ones on educ, exper and tenure with the␣
↪OLS
# and save the residuals
# note that NO INTERCEPT HERE!
wage1['ones'] = 1                          # add a vector of ones into the data set
reg_int = smf.ols(formula='ones ~ educ + exper + tenure -1', data = wage1)
results_int = reg_int.fit()
wage1['resid_int'] = results_int.resid

# The second step: regress y on resid_educ with the OLS
# and obtain the estimated slop parameter
reg_int1 = smf.ols(formula='np.log(wage) ~ resid_int', data = wage1)
results_int1 = reg_int1.fit()
beta_int_hat1 = results_int1.params['resid_int']
beta_int_hat1 = round(beta_int_hat1, 4)

# The OLS intercept from original regression
beta_int_hat = results.params['Intercept']
beta_int_hat = round(beta_int_hat, 4)

print(f'beta_int_hat: \n{beta_int_hat}')
print(f'beta_int_hat1: \n{beta_int_hat1}')
```

```
beta_educ_hat:
0.092
beta_educ_hat1:
0.092
beta_int_hat:
0.2844
beta_int_hat1:
-1.3861
```

$\hat{r_1}$ and β_1 is the same, but the result not applied to estimating the intercept term β_0.