Untitled

July 8, 2019

[50]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import math
```

[51]:
```
stock = pd.read_stata('StockRetAcct_insample.dta')
stock.head()
```

[51]:

	FirmID	year	lnAnnRet	lnRf	MEwt	lnIssue	lnMom	lnME
0	6.0	1980	0.363631	0.078944	0.000281	0.031344	0.075355	12.581472
1	6.0	1981	-0.290409	0.130199	0.000321	0.044213	0.512652	12.907996
2	6.0	1982	0.186630	0.130703	0.000266	-0.068195	-0.220505	12.557775
3	6.0	1983	0.489819	0.089830	0.000170	-0.071780	0.046218	12.561954
4	10.0	1991	-0.508005	0.061216	0.000033	0.115204	1.341053	11.565831

	lnProf	lnEP	lnInv	lnLever	lnROE	rv	lnBM
0	0.201767	0.146411	0.093626	0.696001	0.095294	0.084134	0.633391
1	0.215661	0.102555	0.087242	0.709843	0.082180	0.056381	0.356723
2	0.184087	0.119548	0.111663	0.730972	0.079516	0.062072	0.779405
3	0.165531	0.115924	-0.033117	0.710885	0.055374	0.076955	0.702113
4	0.239788	0.023147	0.300051	0.418764	0.146828	0.374368	-2.160942

	ff_ind
0	3.0
1	3.0
2	3.0
3	3.0
4	10.0

[52]:
```
ff3factors = pd.read_csv('ff3.csv')
ff3factors.columns = ['year', 'Mkt-RF', 'SMB', 'HML', 'RF']
ff3factors.iloc[:, 1:5] = ff3factors.iloc[:, 1:5]/100
ff3factors.head()
```

[52]:

	year	Mkt-RF	SMB	HML	RF
0	1928	0.2947	-0.0246	-0.0375	0.0312
1	1929	0.3539	0.0420	-0.0615	0.0356
2	1930	-0.1954	-0.3080	0.1181	0.0475
3	1931	-0.3123	-0.0513	-0.1228	0.0241

```
4  1932 -0.4511  0.0353 -0.1429  0.0107
```

[53]:
```
stock = pd.DataFrame(stock)
ff3factors = pd.DataFrame(ff3factors)
merged = pd.merge(stock, ff3factors, on = 'year')
merged.head()
```

[53]:
```
   FirmID  year  lnAnnRet      lnRf      MEwt   lnIssue     lnMom       lnME  \
0     6.0  1980  0.363631  0.078944  0.000281  0.031344  0.075355  12.581472
1    50.0  1980  0.160067  0.078944  0.000100 -0.020156  0.306288  11.546848
2   120.0  1980 -0.005239  0.078944  0.000645  0.157939 -0.001933  13.410748
3   128.0  1980  0.159110  0.078944  0.001573  0.172605  0.545400  14.302121
4   135.0  1980  0.124829  0.078944  0.000309  0.059166 -0.297930  12.675659

     lnProf      lnEP     lnInv   lnLever     lnROE        rv      lnBM  \
0  0.201767  0.146411  0.093626  0.696001  0.095294  0.084134  0.633391
1  0.293823  0.162321  0.174245  0.666893  0.182228  0.104305  0.161259
2  0.169987  0.157097  0.045986  0.957038  0.096519  0.046953  0.553104
3  0.444678  0.007621  0.265526  1.057882  0.007154  0.111016  0.133317
4  0.235848  0.173496  0.003995  0.785183  0.121454  0.063248  0.844075

   ff_ind  Mkt-RF     SMB     HML      RF
0     3.0  0.1309  0.2169 -0.0128  0.1038
1     3.0  0.1309  0.2169 -0.0128  0.1038
2     8.0  0.1309  0.2169 -0.0128  0.1038
3     5.0  0.1309  0.2169 -0.0128  0.1038
4     3.0  0.1309  0.2169 -0.0128  0.1038
```

[54]:
```
merged['ExRet'] = merged['lnAnnRet'].apply(math.exp) - merged['lnRf'].
 ↪apply(math.exp)
merged.head()
```

[54]:
```
   FirmID  year  lnAnnRet      lnRf      MEwt   lnIssue     lnMom       lnME  \
0     6.0  1980  0.363631  0.078944  0.000281  0.031344  0.075355  12.581472
1    50.0  1980  0.160067  0.078944  0.000100 -0.020156  0.306288  11.546848
2   120.0  1980 -0.005239  0.078944  0.000645  0.157939 -0.001933  13.410748
3   128.0  1980  0.159110  0.078944  0.001573  0.172605  0.545400  14.302121
4   135.0  1980  0.124829  0.078944  0.000309  0.059166 -0.297930  12.675659

     lnProf      lnEP  ...   lnLever     lnROE        rv      lnBM  ff_ind  \
0  0.201767  0.146411  ...  0.696001  0.095294  0.084134  0.633391     3.0
1  0.293823  0.162321  ...  0.666893  0.182228  0.104305  0.161259     3.0
2  0.169987  0.157097  ...  0.957038  0.096519  0.046953  0.553104     8.0
3  0.444678  0.007621  ...  1.057882  0.007154  0.111016  0.133317     5.0
4  0.235848  0.173496  ...  0.785183  0.121454  0.063248  0.844075     3.0

   Mkt-RF     SMB     HML      RF     ExRet
0  0.1309  0.2169 -0.0128  0.1038  0.356400
1  0.1309  0.2169 -0.0128  0.1038  0.091445
```

```
2  0.1309  0.2169 -0.0128  0.1038 -0.087369
3  0.1309  0.2169 -0.0128  0.1038  0.090323
4  0.1309  0.2169 -0.0128  0.1038  0.050811

[5 rows x 21 columns]
```

[55]:
```
import statsmodels.api as sm
```

[72]:
```
temp = merged[merged['year'] == 1980]
out = sm.formula.ols(formula = "ExRet ~ lnMom", data = temp, missing = 'drop').
 ↪fit()
out.params
```

[72]:
```
Intercept    0.276947
lnMom        0.153960
dtype: float64
```

[57]:
```
out.summary()
```

[57]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                  ExRet   R-squared:                       0.011
Model:                            OLS   Adj. R-squared:                  0.011
Method:                 Least Squares   F-statistic:                     18.55
Date:                Mon, 08 Jul 2019   Prob (F-statistic):           1.75e-05
Time:                        16:15:35   Log-Likelihood:                -977.42
No. Observations:                1615   AIC:                             1959.
Df Residuals:                    1613   BIC:                             1970.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.2769      0.012     22.344      0.000       0.253       0.301
lnMom          0.1540      0.036      4.308      0.000       0.084       0.224
==============================================================================
Omnibus:                      482.721   Durbin-Watson:                   1.968
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1933.918
Skew:                           1.397   Prob(JB):                         0.00
Kurtosis:                       7.576   Cond. No.                         3.33
==============================================================================

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

0.1 Fama Macbeth Regression

[74]:
```
years = merged['year'].unique()
lambdas = [0] * len(years)
for idx, val in enumerate(years):
    out = sm.formula.ols(formula = "ExRet ~ lnProf", data =␣
 ↪merged[merged['year'] == val], missing = 'drop').fit()
    lambdas[idx] = out.params[-1]
mean = np.mean(lambdas)
std = np.std(lambdas)
SR = mean/std
tstats = SR * math.sqrt(len(years))
print('Mean return: ' + str(mean))
print('std: ' + str(std))
print('Sharpe Ratio: ' + str(SR))
print('t stats: ' + str(tstats))
```

```
Mean return: 0.11177208303926002
std: 0.20174605549840272
Sharpe Ratio: 0.5540236351245289
t stats: 3.277648027119584
```

[75]:
```
# fixed effect
years = merged['year'].unique()
lambdas = [0] * len(years)
for idx, val in enumerate(years):
    out = sm.formula.ols(formula = "ExRet ~ lnProf + C(ff_ind)", data =␣
 ↪merged[merged['year'] == val], missing = 'drop').fit()
    lambdas[idx] = out.params[-1]
mean = np.mean(lambdas)
std = np.std(lambdas)
SR = mean/std
tstats = SR * math.sqrt(len(years))
print('Mean return: ' + str(mean))
print('std: ' + str(std))
print('Sharpe Ratio: ' + str(SR))
print('t stats: ' + str(tstats))
```

```
Mean return: 0.10385104780678882
std: 0.15723618858183794
Sharpe Ratio: 0.6604780282672437
t stats: 3.907440710213337
```

[64]:
```
def rolling_regression(temp2):
    temp2 = temp2.sort_values(by = 'year')
    Y = temp2['ExRet'].values
    X = temp2['Mkt-RF'].values
```

```
    # Add intercept
    X = sm.add_constant(X)
    betas = [None] * len(Y)
    for i in range(1 , len(Y)):
        y = Y[0:i]
        x = X[0:i]
        model = sm.OLS(y,x)
        results = model.fit()
        betas[i] = results.params[1]
    temp2['betas'] = betas
    return temp2
```

[65]:
```
signals = merged.groupby('FirmID').apply(rolling_regression)
```

[66]:
```
years = signals['year'].unique()
years = np.sort(years)
years = np.delete(years, 0)
lambdas = [0] * len(years)
for idx, val in enumerate(years):
    out = sm.formula.ols(formula = "ExRet ~ betas", data =␣
 ↪signals[signals['year'] == val], missing = 'drop').fit()
    lambdas[idx] = out.params[1]
mean = np.mean(lambdas)
std = np.std(lambdas)
SR = mean/std
tstats = SR * math.sqrt(len(years))
print('Mean return: ' + str(mean))
print('std: ' + str(std))
print('Sharpe Ratio: ' + str(SR))
print('t stats: ' + str(tstats))
```

```
Mean return: -0.008080010574560014
std: 0.051918682340334914
Sharpe Ratio: -0.15562819028407363
t stats: -0.9074604910282641
```

[ ]:

[ ]: