Assignment 2_Final

June 5, 2022

1 Preparation

[1]:
```
import pandas as pd
import statsmodels.api as sm
import numpy as np
from statsmodels.tools.eval_measures import mse
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
```

[2]:
```
temp1 = pd.read_stata("Assignment_2(StockReturn).dta")
temp2 = pd.read_stata("Assignment_2(Factors).dta")
```

[3]:
```
df = temp1.set_index('ym').join(temp2.set_index('ym'))
df.head()
```

[3]:

ym	permno	ticker	comnam	date	prc	vol	ret	shrout	marketcap	turnover	mktrf	smb	hml	rmw	cma	rf
2001-01-01	10107.0	MSFT	MICROSOFT CORP	2001-01-31	61.062500	10208871.0	0.407781	5335391.0	325792320.0	1.913425	0.0313	0.0545	-0.0511	-0.0546	-0.0505	0.0054
2001-01-01	11850.0	XOM	EXXON MOBIL CORP	2001-01-31	84.150002	1558784.0	-0.032063	3476189.0	292521312.0	0.448418	0.0313	0.0545	-0.0511	-0.0546	-0.0505	0.0054
2001-01-01	12490.0	IBM	INTERNATIONAL BUSINESS MACHS COR	2001-01-31	112.000000	1989212.0	0.317647	1754380.0	196490560.0	1.133855	0.0313	0.0545	-0.0511	-0.0546	-0.0505	0.0054
2001-01-01	14593.0	AAPL	APPLE COMPUTER INC	2001-01-31	21.625000	2482727.0	0.453782	346029.0	7482877.0	7.174910	0.0313	0.0545	-0.0511	-0.0546	-0.0505	0.0054
2001-01-01	55976.0	WMT	WAL MART STORES INC	2001-01-31	56.799999	1799279.0	0.069176	4466336.0	253687888.0							

```
2001-01-01  0.402853  0.0313  0.0545 -0.0511 -0.0546 -0.0505  0.0054
```

[4]:
```
# check if null exists
df.isnull().values.any()
```

[4]:
```
False
```

2 First Part: Seeking Alpha

2.1 IBM

Times-series regression is implemented to IBM. The outcome of the regressions for the return on market premium factor, Fama-French 3 factors & 5 factors, and the Newey-West estimator are as stated. Accordingly, the market premium factor plays a critical rule in all the models, while other factors are insignificant in the two Fama-French models.

2.1.1 Market premium factor

[5]:
```
df_IBM = df[df["ticker"]=="IBM"]
Y_IBM = df_IBM["ret"]
X_IBM = df_IBM["mktrf"]
X_IBM = sm.add_constant(X_IBM)
model_IBM = sm.OLS(Y_IBM, X_IBM)
results_IBM = model_IBM.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[6]:
```
results_IBM.summary()
```

[6]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.385
Model:                            OLS   Adj. R-squared:                  0.383
Method:                 Least Squares   F-statistic:                     59.93
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           2.44e-13
Time:                        22:07:07   Log-Likelihood:                 371.80
No. Observations:                 252   AIC:                            -739.6
Df Residuals:                     250   BIC:                            -732.5
Df Model:                           1
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0005      0.004     -0.133      0.894      -0.008       0.007
mktrf          0.9848      0.127      7.741      0.000       0.735       1.234
==============================================================================
Omnibus:                       67.363   Durbin-Watson:                   1.977
```

```
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               331.897
Skew:                           0.962   Prob(JB):                     8.50e-73
Kurtosis:                       8.282   Cond. No.                         22.5
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

2.1.2 Fama-French 3 factors

[7]:
```
X_IBM_3 = df_IBM[["mktrf", "smb", "hml"]]
X_IBM_3 = sm.add_constant(X_IBM_3)
model_IBM_3 = sm.OLS(Y_IBM, X_IBM_3)
results_IBM_3 = model_IBM_3.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[8]:
```
results_IBM_3.summary()
```

[8]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.402
Model:                            OLS   Adj. R-squared:                  0.395
Method:                 Least Squares   F-statistic:                     31.11
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           4.17e-17
Time:                        22:07:07   Log-Likelihood:                 375.27
No. Observations:                 252   AIC:                            -742.5
Df Residuals:                     248   BIC:                            -728.4
Df Model:                           3
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.0005      0.004     -0.152      0.879      -0.008       0.007
mktrf          1.0133      0.140      7.231      0.000       0.739       1.288
smb           -0.0527      0.254     -0.208      0.835      -0.550       0.444
hml           -0.2904      0.214     -1.357      0.175      -0.710       0.129
==============================================================================
Omnibus:                       53.837   Durbin-Watson:                   1.935
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              223.909
Skew:                           0.789   Prob(JB):                     2.39e-49
Kurtosis:                       7.340   Cond. No.                         44.2
==============================================================================

Notes:
```

```
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

2.1.3 Fama-French 5 factors

[9]:
```
X_IBM_5 = df_IBM[["mktrf", "smb", "hml", "rmw", "cma"]]
X_IBM_5 = sm.add_constant(X_IBM_5)
model_IBM_5 = sm.OLS(Y_IBM, X_IBM_5)
results_IBM_5 = model_IBM_5.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[10]:
```
results_IBM_5.summary()
```

[10]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.413
Model:                            OLS   Adj. R-squared:                  0.401
Method:                 Least Squares   F-statistic:                     18.92
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           6.51e-16
Time:                        22:07:07   Log-Likelihood:                377.73
No. Observations:                 252   AIC:                            -743.5
Df Residuals:                     246   BIC:                            -722.3
Df Model:                           5
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0018      0.004      0.406      0.685      -0.007       0.011
mktrf          0.9379      0.132      7.106      0.000       0.679       1.197
smb           -0.1407      0.271     -0.520      0.603      -0.671       0.390
hml           -0.1762      0.184     -0.956      0.339      -0.538       0.185
rmw           -0.3754      0.209     -1.795      0.073      -0.785       0.034
cma           -0.1016      0.262     -0.387      0.698      -0.615       0.412
==============================================================================
Omnibus:                       49.300   Durbin-Watson:                   1.921
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              180.777
Skew:                           0.754   Prob(JB):                     5.56e-40
Kurtosis:                       6.865   Cond. No.                         71.4
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

2.2 TSM

Times-series regression is implemented to TSM. The outcome of the regressions for the return on market premium factor, Fama-French 3 factors & 5 factors, and the Newey-West estimator are as stated. Accordingly, the market premium factor plays a critical rule in all the models, while other factors are insignificant in the two Fama-French models. Nevertheless, the smb and hml results are significant in Fama-French 3 factors model, and hml and rmw factors are significant in five factors model, implying that hml is important for TSM's stock price.

2.2.1 Market premium factor

[11]:
```
df_TSM = df[df["ticker"]=="TSM"]
Y_TSM = df_TSM["ret"]
X_TSM = df_TSM["mktrf"]
X_TSM = sm.add_constant(X_TSM)
model_TSM = sm.OLS(Y_TSM, X_TSM)
results_TSM = model_TSM.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[12]:
```
results_TSM.summary()
```

[12]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.377
Model:                            OLS   Adj. R-squared:                  0.374
Method:                 Least Squares   F-statistic:                     40.84
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           8.04e-10
Time:                        22:07:07   Log-Likelihood:                294.42
No. Observations:                 252   AIC:                            -584.8
Df Residuals:                     250   BIC:                            -577.8
Df Model:                           1
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0084      0.004      1.910      0.056      -0.000       0.017
mktrf          1.3158      0.206      6.390      0.000       0.912       1.719
==============================================================================
Omnibus:                       44.443   Durbin-Watson:                   2.051
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              128.917
Skew:                           0.751   Prob(JB):                     1.01e-28
Kurtosis:                       6.166   Cond. No.                         22.5
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
```

```
"""
```

2.2.2 Fama-French 3 factors

[13]:
```
X_TSM_3 = df_TSM[["mktrf", "smb", "hml"]]
X_TSM_3 = sm.add_constant(X_TSM_3)
model_TSM_3 = sm.OLS(Y_TSM, X_TSM_3)
results_TSM_3 = model_TSM_3.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[14]:
```
results_TSM_3.summary()
```

[14]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.405
Model:                            OLS   Adj. R-squared:                  0.398
Method:                 Least Squares   F-statistic:                     17.82
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           1.66e-10
Time:                        22:07:07   Log-Likelihood:                 300.25
No. Observations:                 252   AIC:                            -592.5
Df Residuals:                     248   BIC:                            -578.4
Df Model:                           3
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0077      0.004      1.913      0.056      -0.000       0.016
mktrf          1.2817      0.186      6.878      0.000       0.916       1.647
smb            0.3514      0.245      1.435      0.151      -0.129       0.831
hml           -0.5360      0.232     -2.309      0.021      -0.991      -0.081
==============================================================================
Omnibus:                       29.319   Durbin-Watson:                   2.063
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               64.698
Skew:                           0.565   Prob(JB):                     8.93e-15
Kurtosis:                       5.210   Cond. No.                         44.2
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

2.2.3 Fama-French 5 factors

[15]:
```
X_TSM_5 = df_TSM[["mktrf", "smb", "hml", "rmw", "cma"]]
X_TSM_5 = sm.add_constant(X_TSM_5)
model_TSM_5 = sm.OLS(Y_TSM, X_TSM_5)
results_TSM_5 = model_TSM_5.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[16]:
```
results_TSM_5.summary()
```

[16]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.441
Model:                            OLS   Adj. R-squared:                  0.430
Method:                 Least Squares   F-statistic:                     19.11
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           4.66e-16
Time:                        22:07:07   Log-Likelihood:                308.12
No. Observations:                 252   AIC:                            -604.2
Df Residuals:                     246   BIC:                            -583.1
Df Model:                           5
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0131      0.005      2.711      0.007       0.004       0.023
mktrf          1.1125      0.138      8.086      0.000       0.843       1.382
smb            0.1352      0.237      0.572      0.568      -0.328       0.599
hml           -0.3100      0.156     -1.985      0.047      -0.616      -0.004
rmw           -0.9123      0.284     -3.216      0.001      -1.468      -0.356
cma           -0.0958      0.456     -0.210      0.834      -0.990       0.799
==============================================================================
Omnibus:                       29.927   Durbin-Watson:                   1.983
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               67.905
Skew:                           0.567   Prob(JB):                     1.80e-15
Kurtosis:                       5.276   Cond. No.                         71.4
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

3 Second Part: Turnover and Out-of-Sample Tests

Do stocks with weaker liquidity have to compensate investors with higher returns?

We present the result of time series regression of stock return on the turnover ratio in the previous

month by using the Newey-West estimator, which includes the coefficient estimated, the t-statistic, and the corresponding p-value. **With the negative coefficient, we can roughly draw a conclusion that stocks with weaker liquidity do have to compensate investors with higher returns; However, the p-value is insignificant**.

[17]:
```
df_MSFT = df[df["ticker"]=="MSFT"]
Y_MSFT = df_MSFT["ret"][1:]

x = df_MSFT["turnover"].copy()
X_MSFT = pd.DataFrame(x[:-1]).set_index(df.index.unique()[1:])  # reset index␣
↪for lag X

X_MSFT = sm.add_constant(X_MSFT)
model_MSFT = sm.OLS(Y_MSFT, X_MSFT)
results_MSFT = model_MSFT.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[18]:
```
results_MSFT.summary()
```

[18]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.012
Model:                            OLS   Adj. R-squared:                  0.008
Method:                 Least Squares   F-statistic:                     3.120
Date:                Sun, 05 Jun 2022   Prob (F-statistic):             0.0786
Time:                        22:07:07   Log-Likelihood:                 317.95
No. Observations:                 251   AIC:                            -631.9
Df Residuals:                     249   BIC:                            -624.8
Df Model:                           1
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0348      0.011      3.067      0.002       0.013       0.057
turnover      -0.0175      0.010     -1.766      0.077      -0.037       0.002
==============================================================================
Omnibus:                        9.417   Durbin-Watson:                   2.181
Prob(Omnibus):                  0.009   Jarque-Bera (JB):               11.728
Skew:                           0.309   Prob(JB):                      0.00284
Kurtosis:                       3.860   Cond. No.                         5.92
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

Model Comparison

With the comparison of turnover model and random walk model, we can see the result of MSE as below. For the turnover model, its MSE is slightly higher than the random walk model. **The result shows that turnover model cannot be a good mdoel for stock return since the model underperform the random walk model.**

MSE of Rolling Regression

[19]:
```
ret = df_MSFT["ret"]
turnover = df_MSFT["turnover"]
all_res = []
a = []
b = []

for i in range(df_MSFT.shape[0]-24):
    ret_i = ret.iloc[i:i+23]
    turnover_i = turnover.iloc[i:i+23]
    turnover_i = sm.add_constant(turnover_i)
    model = sm.OLS(ret_i, turnover_i).fit()
    residual = ret.iloc[i+24] - (model.params[0] + model.params[1]*turnover.
 ↪iloc[i+24])
    all_res.append(residual)

for i in range(len(all_res)):
    all_res[i] = all_res[i]**2

sum(all_res)/228
```

[19]:
```
0.004213175991491555
```

MSE of average of 24 month model

[20]:
```
past_average = []
for end_time in range(len(Y_MSFT)):
    avg = np.mean(Y_MSFT[end_time-24:end_time])
    if np.isnan(avg) == False:
        past_average.append(avg)

mse(past_average, Y_MSFT[24:])
```

[20]:
```
0.004087260674424726
```

4 Third Part: Portfolio Analysis

using the information between 2001 and 2010

[21]:
```
df.index.unique()[:120]
```

[21]:
```
DatetimeIndex(['2001-01-01', '2001-02-01', '2001-03-01', '2001-04-01',
               '2001-05-01', '2001-06-01', '2001-07-01', '2001-08-01',
               '2001-09-01', '2001-10-01',
               …
               '2010-03-01', '2010-04-01', '2010-05-01', '2010-06-01',
               '2010-07-01', '2010-08-01', '2010-09-01', '2010-10-01',
               '2010-11-01', '2010-12-01'],
              dtype='datetime64[ns]', name='ym', length=120, freq=None)
```

using the information between 2011 and 2021

[22]:
```
df.index.unique()[120:]
```

[22]:
```
DatetimeIndex(['2011-01-01', '2011-02-01', '2011-03-01', '2011-04-01',
               '2011-05-01', '2011-06-01', '2011-07-01', '2011-08-01',
               '2011-09-01', '2011-10-01',
               …
               '2021-03-01', '2021-04-01', '2021-05-01', '2021-06-01',
               '2021-07-01', '2021-08-01', '2021-09-01', '2021-10-01',
               '2021-11-01', '2021-12-01'],
              dtype='datetime64[ns]', name='ym', length=132, freq=None)
```

4.1 First Portfolio

Equal-weighted

[23]:
```
df_first = {
    "ret": [],
    "mktrf": [],
    "smb": [],
    "hml": [],
    "rmw": [],
    "cma": []
}
```

[24]:
```
for i in range(len(df.index.unique())):
    df_first["ret"].append(df.loc[df.index.unique()[i]]["ret"].mean())
    df_first["mktrf"].append(df.loc[df.index.unique()[i]]["mktrf"].mean())
    df_first["smb"].append(df.loc[df.index.unique()[i]]["smb"].mean())
    df_first["hml"].append(df.loc[df.index.unique()[i]]["hml"].mean())
    df_first["rmw"].append(df.loc[df.index.unique()[i]]["rmw"].mean())
    df_first["cma"].append(df.loc[df.index.unique()[i]]["cma"].mean())
```

[25]:
```
df_first = pd.DataFrame(df_first).set_index(df.index.unique())
```

[26]:
```
df_first.head()
```

[26]:
```
                 ret    mktrf     smb     hml     rmw     cma
ym
2001-01-01  0.255562  0.0313  0.0545 -0.0511 -0.0546 -0.0505
2001-02-01 -0.103990 -0.1005  0.0279  0.1248  0.0912  0.0906
2001-03-01  0.034233 -0.0726  0.0236  0.0643  0.0339  0.0390
2001-04-01  0.139320  0.0794 -0.0088 -0.0470 -0.0348 -0.0318
2001-05-01 -0.047119  0.0072  0.0359  0.0336  0.0025  0.0191
```

4.1.1 Market premium factor (2001-2010)

[27]:
```
Y1_first = df_first["ret"][df.index.unique()[:120]]
X_first = df_first["mktrf"][df.index.unique()[:120]]
X_first = sm.add_constant(X_first)
model_first = sm.OLS(Y1_first, X_first)
results_first = model_first.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[28]:
```
results_first.summary()
```

[28]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.700
Model:                            OLS   Adj. R-squared:                  0.698
Method:                 Least Squares   F-statistic:                     275.8
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           1.14e-32
Time:                        22:07:09   Log-Likelihood:                 234.98
No. Observations:                 120   AIC:                            -466.0
Df Residuals:                     118   BIC:                            -460.4
Df Model:                           1
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0108      0.004      2.839      0.005       0.003       0.018
mktrf          1.0833      0.065     16.608      0.000       0.955       1.211
==============================================================================
Omnibus:                       87.556   Durbin-Watson:                   1.578
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              743.461
Skew:                           2.387   Prob(JB):                    3.63e-162
Kurtosis:                      14.220   Cond. No.                         20.7
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.1.2 Market premium factor (2011-2021)

[29]:
```
Y2_first = df_first["ret"][df.index.unique()[120:]]
X_first = df_first["mktrf"][df.index.unique()[120:]]
X_first = sm.add_constant(X_first)
model_first = sm.OLS(Y2_first, X_first)
results_first = model_first.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[30]:
```
results_first.summary()
```

[30]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.790
Model:                            OLS   Adj. R-squared:                  0.788
Method:                 Least Squares   F-statistic:                     698.1
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           4.09e-54
Time:                        22:07:09   Log-Likelihood:                 330.04
No. Observations:                 132   AIC:                            -656.1
Df Residuals:                     130   BIC:                            -650.3
Df Model:                           1
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0032      0.001      2.277      0.023       0.000       0.006
mktrf          0.9595      0.036     26.421      0.000       0.888       1.031
==============================================================================
Omnibus:                        1.209   Durbin-Watson:                   2.156
Prob(Omnibus):                  0.546   Jarque-Bera (JB):                0.774
Skew:                           0.061   Prob(JB):                        0.679
Kurtosis:                       3.355   Cond. No.                         24.9
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.1.3 Fama-French 3 factors (2001-2010)

[31]:
```
X_first_3 = df_first[["mktrf", "smb", "hml"]]
X_first_3 = sm.add_constant(X_first_3)
model_first_3 = sm.OLS(Y1_first, X_first_3[:120])
results_first_3 = model_first_3.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[32]:
```
results_first_3.summary()
```

[32]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.706
Model:                            OLS   Adj. R-squared:                  0.698
Method:                 Least Squares   F-statistic:                     113.0
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           2.85e-34
Time:                        22:07:09   Log-Likelihood:                236.10
No. Observations:                 120   AIC:                            -464.2
Df Residuals:                     116   BIC:                            -453.0
Df Model:                           3
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0107      0.003      3.344      0.001       0.004       0.017
mktrf          1.0740      0.071     15.089      0.000       0.935       1.214
smb            0.0822      0.196      0.419      0.676      -0.303       0.467
hml           -0.1504      0.175     -0.858      0.391      -0.494       0.193
==============================================================================
Omnibus:                       77.770   Durbin-Watson:                   1.523
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              525.237
Skew:                           2.132   Prob(JB):                    8.84e-115
Kurtosis:                      12.320   Cond. No.                         41.4
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.1.4 Fama-French 3 factors (2011-2021)

[33]:
```
X_first_3 = df_first[["mktrf", "smb", "hml"]]
X_first_3 = sm.add_constant(X_first_3)
model_first_3 = sm.OLS(Y2_first, X_first_3[120:])
results_first_3 = model_first_3.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[34]:
```
results_first_3.summary()
```

[34]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:                    ret   R-squared:                       0.809
Model:                            OLS   Adj. R-squared:                  0.804
Method:                 Least Squares   F-statistic:                     270.7
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           3.18e-55
Time:                        22:07:09   Log-Likelihood:                336.29
No. Observations:                 132   AIC:                            -664.6
Df Residuals:                     128   BIC:                            -653.1
Df Model:                           3
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0029      0.001      1.937      0.053   -3.37e-05       0.006
mktrf          0.9998      0.039     25.633      0.000       0.923       1.076
smb           -0.2394      0.098     -2.445      0.014      -0.431      -0.048
hml            0.1498      0.052      2.873      0.004       0.048       0.252
==============================================================================
Omnibus:                        1.810   Durbin-Watson:                   2.157
Prob(Omnibus):                  0.405   Jarque-Bera (JB):                1.449
Skew:                          -0.031   Prob(JB):                        0.484
Kurtosis:                       3.510   Cond. No.                         48.1
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.1.5 Fama-French 5 factors (2001-2010)

[35]:
```
X_first_5 = df_first[["mktrf", "smb", "hml", "rmw", "cma"]]
X_first_5 = sm.add_constant(X_first_5)
model_first_5 = sm.OLS(Y1_first, X_first_5[:120])
results_first_5 = model_first_5.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[36]:
```
results_first_5.summary()
```

[36]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.746
Model:                            OLS   Adj. R-squared:                  0.735
Method:                 Least Squares   F-statistic:                     166.5
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           1.12e-50
Time:                        22:07:09   Log-Likelihood:                244.90
No. Observations:                 120   AIC:                            -477.8
```

```
Df Residuals:                      114   BIC:                            -461.1
Df Model:                            5
Covariance Type:                   HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0143      0.004      3.622      0.000       0.007       0.022
mktrf          0.8955      0.128      6.970      0.000       0.644       1.147
smb            0.0610      0.177      0.345      0.730      -0.286       0.408
hml            0.1742      0.153      1.142      0.253      -0.125       0.473
rmw           -0.3824      0.150     -2.545      0.011      -0.677      -0.088
cma           -0.6133      0.219     -2.797      0.005      -1.043      -0.184
==============================================================================
Omnibus:                       59.598   Durbin-Watson:                   1.406
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              257.119
Skew:                           1.696   Prob(JB):                     1.47e-56
Kurtosis:                       9.318   Cond. No.                         67.2
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.1.6 Fama-French 5 factors (2011-2021)

[37]:
```
X_first_5 = df_first[["mktrf", "smb", "hml", "rmw", "cma"]]
X_first_5 = sm.add_constant(X_first_5)
model_first_5 = sm.OLS(Y2_first, X_first_5[120:])
results_first_5 = model_first_5.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[38]:
```
results_first_5.summary()
```

[38]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.810
Model:                            OLS   Adj. R-squared:                  0.802
Method:                 Least Squares   F-statistic:                     176.3
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           4.21e-55
Time:                        22:07:09   Log-Likelihood:                 336.62
No. Observations:                 132   AIC:                            -661.2
Df Residuals:                     126   BIC:                            -643.9
Df Model:                           5
Covariance Type:                  HAC
==============================================================================
```

```
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0031      0.002      1.943      0.052    -2.6e-05       0.006
mktrf          0.9916      0.041     24.163      0.000       0.911       1.072
smb           -0.2496      0.087     -2.870      0.004      -0.420      -0.079
hml            0.1832      0.070      2.617      0.009       0.046       0.320
rmw           -0.0190      0.121     -0.157      0.875      -0.257       0.219
cma           -0.0962      0.227     -0.425      0.671      -0.540       0.348
==============================================================================
Omnibus:                        1.094   Durbin-Watson:                   2.122
Prob(Omnibus):                  0.579   Jarque-Bera (JB):                0.666
Skew:                          -0.048   Prob(JB):                        0.717
Kurtosis:                       3.334   Cond. No.                         81.2
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.2 Second portfolio

62% AAPL & 14% WMT & 24% XOM

[39]:
```
df_second = {
    "ret": [],
    "mktrf": [],
    "smb": [],
    "hml": [],
    "rmw": [],
    "cma": []
}
```

[40]:
```
## test: stock == "AAPL" & column = "ret" & specific date
df[df["ticker"]=="AAPL"]["ret"][df.index.unique()[0]]
```

[40]:
```
0.45378151535987854
```

[41]:
```
for i in range(len(df.index.unique())):
    df_second["ret"].append(df[df["ticker"]=="AAPL"]["ret"][df.index.
 ↪unique()[i]]*0.62 + df[df["ticker"]=="WMT"]["ret"][df.index.unique()[i]]*0.
 ↪14 + df[df["ticker"]=="XOM"]["ret"][df.index.unique()[i]]*0.24)
    df_second["mktrf"].append(df[df["ticker"]=="AAPL"]["mktrf"][df.index.
 ↪unique()[i]]*0.62 + df[df["ticker"]=="WMT"]["mktrf"][df.index.unique()[i]]*0.
 ↪14 + df[df["ticker"]=="XOM"]["mktrf"][df.index.unique()[i]]*0.24)
```

```
    df_second["smb"].append(df[df["ticker"]=="AAPL"]["smb"][df.index.
↪unique()[i]]*0.62 + df[df["ticker"]=="WMT"]["smb"][df.index.unique()[i]]*0.
↪14 + df[df["ticker"]=="XOM"]["smb"][df.index.unique()[i]]*0.24)
    df_second["hml"].append(df[df["ticker"]=="AAPL"]["hml"][df.index.
↪unique()[i]]*0.62 + df[df["ticker"]=="WMT"]["hml"][df.index.unique()[i]]*0.
↪14 + df[df["ticker"]=="XOM"]["hml"][df.index.unique()[i]]*0.24)
    df_second["rmw"].append(df[df["ticker"]=="AAPL"]["rmw"][df.index.
↪unique()[i]]*0.62 + df[df["ticker"]=="WMT"]["rmw"][df.index.unique()[i]]*0.
↪14 + df[df["ticker"]=="XOM"]["rmw"][df.index.unique()[i]]*0.24)
    df_second["cma"].append(df[df["ticker"]=="AAPL"]["cma"][df.index.
↪unique()[i]]*0.62 + df[df["ticker"]=="WMT"]["cma"][df.index.unique()[i]]*0.
↪14 + df[df["ticker"]=="XOM"]["cma"][df.index.unique()[i]]*0.24)
```

[42]:
```
df_second = pd.DataFrame(df_second).set_index(df.index.unique())
```

[43]:
```
df_second.head()
```

[43]:

ym	ret	mktrf	smb	hml	rmw	cma
2001-01-01	0.283334	0.0313	0.0545	-0.0511	-0.0546	-0.0505
2001-02-01	-0.120888	-0.1005	0.0279	0.1248	0.0912	0.0906
2001-03-01	0.130969	-0.0726	0.0236	0.0643	0.0339	0.0390
2001-04-01	0.122032	0.0794	-0.0088	-0.0470	-0.0348	-0.0318
2001-05-01	-0.133126	0.0072	0.0359	0.0336	0.0025	0.0191

4.2.1 Market premium factor (2001-2010)

[44]:
```
Y1_second = df_second["ret"][:120]
X_second = df_second["mktrf"][:120]
X_second = sm.add_constant(X_second)
model_second = sm.OLS(Y1_second, X_second)
results_second = model_second.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[45]:
```
results_second.summary()
```

[45]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.367
Model:                            OLS   Adj. R-squared:                  0.361
Method:                 Least Squares   F-statistic:                     88.65
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           4.86e-16
Time:                        22:07:12   Log-Likelihood:                 155.48
No. Observations:                 120   AIC:                            -307.0
Df Residuals:                     118   BIC:                            -301.4
```

```
Df Model:                           1
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0254      0.006      4.299      0.000       0.014       0.037
mktrf          1.0452      0.111      9.415      0.000       0.828       1.263
==============================================================================
Omnibus:                        6.858   Durbin-Watson:                   1.961
Prob(Omnibus):                  0.032   Jarque-Bera (JB):                7.550
Skew:                           0.376   Prob(JB):                       0.0229
Kurtosis:                       3.972   Cond. No.                         20.7
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.2.2 Market premium factor (2011-2021)

[46]:
```
Y2_second = df_second["ret"][120:]
X_second = df_second["mktrf"][120:]
X_second = sm.add_constant(X_second)
model_second = sm.OLS(Y2_second, X_second)
results_second = model_second.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[47]:
```
results_second.summary()
```

[47]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.488
Model:                            OLS   Adj. R-squared:                  0.484
Method:                 Least Squares   F-statistic:                     105.4
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           1.80e-18
Time:                        22:07:12   Log-Likelihood:                 237.85
No. Observations:                 132   AIC:                            -471.7
Df Residuals:                     130   BIC:                            -465.9
Df Model:                           1
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0063      0.004      1.600      0.110      -0.001       0.014
mktrf          0.9710      0.095     10.266      0.000       0.786       1.156
```

```
==============================================================================
Omnibus:                        9.596   Durbin-Watson:                   1.897
Prob(Omnibus):                  0.008   Jarque-Bera (JB):               11.716
Skew:                          -0.450   Prob(JB):                      0.00286
Kurtosis:                       4.149   Cond. No.                         24.9
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.2.3 Fama-French 3 factors (2001-2010)

[48]:
```
X_second_3 = df_second[["mktrf", "smb", "hml"]]
X_second_3 = sm.add_constant(X_second_3)
model_second_3 = sm.OLS(Y1_second, X_second_3[:120])
results_second_3 = model_second_3.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[49]:
```
results_second_3.summary()
```

[49]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.413
Model:                            OLS   Adj. R-squared:                  0.398
Method:                 Least Squares   F-statistic:                     66.29
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           4.86e-25
Time:                        22:07:12   Log-Likelihood:                 160.03
No. Observations:                 120   AIC:                            -312.1
Df Residuals:                     116   BIC:                            -300.9
Df Model:                           3
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0250      0.006      4.029      0.000       0.013       0.037
mktrf          1.0103      0.122      8.298      0.000       0.772       1.249
smb            0.3138      0.237      1.322      0.186      -0.152       0.779
hml           -0.5816      0.171     -3.409      0.001      -0.916      -0.247
==============================================================================
Omnibus:                        9.638   Durbin-Watson:                   1.842
Prob(Omnibus):                  0.008   Jarque-Bera (JB):               10.513
Skew:                           0.534   Prob(JB):                      0.00521
Kurtosis:                       3.980   Cond. No.                         41.4
==============================================================================
```

```
Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.2.4 Fama-French 3 factors (2011-2021)

[50]:
```
X_second_3 = df_second[["mktrf", "smb", "hml"]]
X_second_3 = sm.add_constant(X_second_3)
model_second_3 = sm.OLS(Y2_second, X_second_3[120:])
results_second_3 = model_second_3.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[51]:
```
results_second_3.summary()
```

[51]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.533
Model:                            OLS   Adj. R-squared:                  0.522
Method:                 Least Squares   F-statistic:                     58.36
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           7.65e-24
Time:                        22:07:12   Log-Likelihood:                 243.93
No. Observations:                 132   AIC:                            -479.9
Df Residuals:                     128   BIC:                            -468.3
Df Model:                           3
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0042      0.004      1.157      0.247      -0.003       0.011
mktrf          1.0795      0.104     10.349      0.000       0.875       1.284
smb           -0.3407      0.157     -2.173      0.030      -0.648      -0.033
hml           -0.2233      0.122     -1.827      0.068      -0.463       0.016
==============================================================================
Omnibus:                       13.847   Durbin-Watson:                   1.934
Prob(Omnibus):                  0.001   Jarque-Bera (JB):               20.645
Skew:                          -0.539   Prob(JB):                     3.29e-05
Kurtosis:                       4.609   Cond. No.                         48.1
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.2.5 Fama-French 5 factors (2001-2010)

[52]:
```
X_second_5 = df_second[["mktrf", "smb", "hml", "rmw", "cma"]]
X_second_5 = sm.add_constant(X_second_5)
model_second_5 = sm.OLS(Y1_second, X_second_5[:120])
results_second_5 = model_second_5.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[53]:
```
results_second_5.summary()
```

[53]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.444
Model:                            OLS   Adj. R-squared:                  0.419
Method:                 Least Squares   F-statistic:                     32.48
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           1.83e-20
Time:                        22:07:12   Log-Likelihood:                 163.27
No. Observations:                 120   AIC:                            -314.5
Df Residuals:                     114   BIC:                            -297.8
Df Model:                           5
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0268      0.005      4.979      0.000       0.016       0.037
mktrf          0.9492      0.199      4.759      0.000       0.558       1.340
smb            0.3612      0.216      1.671      0.095      -0.062       0.785
hml           -0.2784      0.274     -1.015      0.310      -0.816       0.259
rmw            0.0391      0.229      0.171      0.865      -0.410       0.489
cma           -0.8554      0.267     -3.203      0.001      -1.379      -0.332
==============================================================================
Omnibus:                        9.900   Durbin-Watson:                   1.868
Prob(Omnibus):                  0.007   Jarque-Bera (JB):               11.479
Skew:                           0.512   Prob(JB):                      0.00322
Kurtosis:                       4.116   Cond. No.                         67.2
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.2.6 Fama-French 5 factors (2011-2021)

[54]:
```
X_second_5 = df_second[["mktrf", "smb", "hml", "rmw", "cma"]]
X_second_5 = sm.add_constant(X_second_5)
model_second_5 = sm.OLS(Y2_second, X_second_5[120:])
results_second_5 = model_second_5.fit(cov_type='HAC',cov_kwds={'maxlags':12})
```

[55]:
```
results_second_5.summary()
```

[55]:
```
<class 'statsmodels.iolib.summary.Summary'>
"""
                            OLS Regression Results
==============================================================================
Dep. Variable:                    ret   R-squared:                       0.573
Model:                            OLS   Adj. R-squared:                  0.556
Method:                 Least Squares   F-statistic:                     75.53
Date:                Sun, 05 Jun 2022   Prob (F-statistic):           3.12e-36
Time:                        22:07:12   Log-Likelihood:                 249.84
No. Observations:                 132   AIC:                            -487.7
Df Residuals:                     126   BIC:                            -470.4
Df Model:                           5
Covariance Type:                  HAC
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          0.0027      0.003      0.892      0.373      -0.003       0.009
mktrf          1.0438      0.092     11.318      0.000       0.863       1.225
smb           -0.0852      0.162     -0.527      0.598      -0.402       0.231
hml           -0.3017      0.148     -2.042      0.041      -0.591      -0.012
rmw            0.6765      0.217      3.120      0.002       0.252       1.101
cma           -0.0691      0.364     -0.190      0.849      -0.782       0.644
==============================================================================
Omnibus:                        6.057   Durbin-Watson:                   1.967
Prob(Omnibus):                  0.048   Jarque-Bera (JB):                6.580
Skew:                          -0.322   Prob(JB):                       0.0373
Kurtosis:                       3.884   Cond. No.                         81.2
==============================================================================

Notes:
[1] Standard Errors are heteroscedasticity and autocorrelation robust (HAC)
using 12 lags and without small sample correction
"""
```

4.3 Summary of Alphas

[56]:
```
df = pd.DataFrame([("1.08%", "0.32%", "2.54%", "0.63%"),
                   ("1.07%", "0.29%", "2.50%", "0.42%"),
                   ("1.43%", "0.31%", "2.68%", "0.27%")],
                  index=["Market Premium", "Fama-French 3 Factors",␣
 ↪"Fama-French 5 Factors"],
                  columns=("Portflio1 (2001-10)", "Portflio1 (2011-21)",␣
 ↪"Portflio2 (2001-10)", "Portflio2 (2011-21)"))
df
```

[56]:

	Portflio1 (2001-10)	Portflio1 (2011-21)	\
Market Premium	1.08%	0.32%	
Fama-French 3 Factors	1.07%	0.29%	
Fama-French 5 Factors	1.43%	0.31%	

	Portflio2 (2001-10)	Portflio2 (2011-21)
Market Premium	2.54%	0.63%
Fama-French 3 Factors	2.50%	0.42%
Fama-French 5 Factors	2.68%	0.27%

4.4 Result

With the above coefficients and intercepts, we may know that the alphas between two periods (2001-2010 and 2011-2021) decrease for all three regression models whether it's equally weighted portfolio or tangency portfolio. We can interpret that the stock market may gradually become efficient with time passing, which makes a long position generates less alpha in a long run. However, the tangency portfolio generates more alpha than the equally weighted one. It may be resulted by AAPL since it generates more alpha than the average of all stocks, and AAPL takes up to 62% for the portfolio.