python_5

February 27, 2024

1 Python Excersie #5: FF 3 Factor Model w/ Factset Data

1.1 Step 1: Load in Facset Data.

[ ]:
```
import numpy as np
import pandas as pd
import statsmodels.formula.api as smf
import matplotlib.pyplot as plt
import seaborn as sns
import requests
import time
import zipfile
import os
```

[ ]:
```
data_raw = pd.read_excel('factset_data.xlsx', skiprows=2, index_col=0).iloc[::
    ↪-1]
```

Clean up the data on compute log returns.

[ ]:
```
data = data_raw.dropna()
data.drop(columns='Composite', inplace=True)
data = data.resample('ME').last().dropna()
securities = ['SP500', 'VFIAX', 'TRBCX', 'CFR']
```

[ ]:
```
nav_data = data
nav_data.columns = securities
```

[ ]:
```
data.columns = securities

for sec in securities:
    data[f'{sec}_log_return'] = np.log(data[sec] / data[sec].shift(1))
data.dropna(inplace=True)
```

[ ]:
```
data.head()
```

[ ]:
```
                SP500   VFIAX  TRBCX    CFR  SP500_log_return  \
Date
2004-03-31  1126.2115  104.01  28.98  42.76         -0.016495
2004-04-30  1107.3048  102.38  28.34  43.30         -0.016930
```

```
2004-05-31  1120.6831  103.78  28.81  43.82          0.012009
2004-06-30  1140.8356  105.41  29.28  44.75          0.017823
2004-07-31  1101.7195  101.92  27.89  43.02         -0.034889

            VFIAX_log_return  TRBCX_log_return  CFR_log_return
Date
2004-03-31         -0.018952         -0.009957        0.007983
2004-04-30         -0.015796         -0.022332        0.012550
2004-05-31          0.013582          0.016448        0.011938
2004-06-30          0.015584          0.016182        0.021001
2004-07-31         -0.033669         -0.048636       -0.039426
```

[ ]:
```
data = data.drop(columns=securities).dropna()
```

[ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 240 entries, 2004-03-31 to 2024-02-29
Freq: ME
Data columns (total 4 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   SP500_log_return  240 non-null    float64
 1   VFIAX_log_return  240 non-null    float64
 2   TRBCX_log_return  240 non-null    float64
 3   CFR_log_return    240 non-null    float64
dtypes: float64(4)
memory usage: 9.4 KB
```

[ ]:
```
data.head()
```

[ ]:
```
            SP500_log_return  VFIAX_log_return  TRBCX_log_return  \
Date
2004-03-31         -0.016495         -0.018952         -0.009957
2004-04-30         -0.016930         -0.015796         -0.022332
2004-05-31          0.012009          0.013582          0.016448
2004-06-30          0.017823          0.015584          0.016182
2004-07-31         -0.034889         -0.033669         -0.048636

            CFR_log_return
Date
2004-03-31        0.007983
2004-04-30        0.012550
2004-05-31        0.011938
2004-06-30        0.021001
2004-07-31       -0.039426
```

[ ]:
```
data.describe()
```

```
[ ]:        SP500_log_return  VFIAX_log_return  TRBCX_log_return  CFR_log_return
count          240.000000        240.000000        240.000000      240.000000
mean             0.006200          0.006193          0.007233        0.003874
std              0.043424          0.043590          0.051237        0.068196
min             -0.185634         -0.183720         -0.207890       -0.340102
25%             -0.017585         -0.018488         -0.015318       -0.034005
50%              0.012115          0.013398          0.013986        0.009644
75%              0.032280          0.032025          0.037897        0.045315
max              0.119421          0.120611          0.139681        0.253125
```

1.2 Step 2: Net Asset Value.

The Net Asset Value (NAV) is calculated differently depending on whether or not the mutual fund is an open or closed end fund. If the fun is open, the NAV is calculated by dviding the total value of the fund by the number of outstanding shares. This is done at the end of the day. The open-end funds issue new shares whenever an investor wants to purchase one, then redeems those shares for cash when the investor wants to sell those shares. This is the most common of the two funds.

The close-end fund behaves more like a traditional stock does. They IPO, and issue an initial amount of shares that are then traded at varying amounts based on a few factors. Because the price of these shares can fluctuate, it is possible to get them at discounts, or have to pay premiums, above the NAV.

```
[ ]: print("First and last 5 NAV data points for T. Rowe Blue Chip Growth Fund")
     print(nav_data['TRBCX'].head())
     print(nav_data['TRBCX'].tail())
```

```
First and last 5 NAV data points for T. Rowe Blue Chip Growth Fund
Date
2004-03-31    28.98
2004-04-30    28.34
2004-05-31    28.81
2004-06-30    29.28
2004-07-31    27.89
Freq: ME, Name: TRBCX, dtype: float64
Date
2023-10-31    134.89
2023-11-30    149.54
2023-12-31    149.34
2024-01-31    154.66
2024-02-29    166.08
Freq: ME, Name: TRBCX, dtype: float64
```

1.3 Step 3: Input FactSet Data into lecture code.

FF dataset download and clean.

```
[ ]: url = 'http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/ftp/
      ↪F-F_Research_Data_Factors_CSV.zip'
     local_zip_file = 'F-F_Research_Data_Factors_CSV.zip'
```

```
local_path = 'C:/Users/Brady/Dropbox/School Files/FIN 5322 - Investment␣
↪Analysis/Python 5 - FF 3 Factor Model'

response = requests.get(url)
with open(local_zip_file, 'wb') as file:
    file.write(response.content)

with zipfile.ZipFile(local_zip_file, 'r') as zip_ref:
    zip_ref.extractall(local_path)

time.sleep(0.5)
os.remove(local_zip_file)
```

[ ]:
```
ff_data_raw = pd.read_csv('F-F_Research_Data_Factors.CSV', names=['date',␣
↪'mkt', 'smb', 'hml', 'rf'], skiprows=4)
ff_data = ff_data_raw.iloc[0:1170, :]
```

[ ]:
```
# Clean up the data and convert to log returns.
ff_data.set_index('date', drop=True, inplace=True)
ff_data.index = pd.to_datetime(ff_data.index, format='%Y%m')
ff_data = ff_data.apply(pd.to_numeric, errors='coerce').div(100).map(
    lambda x: np.log(1 + x))
```

[ ]:
```
ff_data.tail()
```

[ ]:

	mkt	smb	hml	rf
date				
2023-08-01	-0.024190	-0.032110	-0.010657	0.004490
2023-09-01	-0.053823	-0.025420	0.015086	0.004291
2023-10-01	-0.032420	-0.039469	0.001898	0.004689
2023-11-01	0.084709	-0.000200	0.016267	0.004390
2023-12-01	0.047361	0.061565	0.048219	0.004291

[ ]:
```
# Merge the two datasets.

# Align date formats.
data.index = data.index.to_period('M').to_timestamp()
ff_data.index = ff_data.index.to_period('M').to_timestamp()
m_data = pd.merge(data, ff_data, how='outer', left_index=True, right_index=True)
m_data.dropna(inplace=True)
m_data = m_data.loc['2004-03-01':'2023-12-01', :]

# Cleaning up some memory.
del data_raw, data, ff_data_raw, ff_data
```

[ ]:
```
print(' Merge Data Info '.center(75, '-'))
print(m_data.info())
```

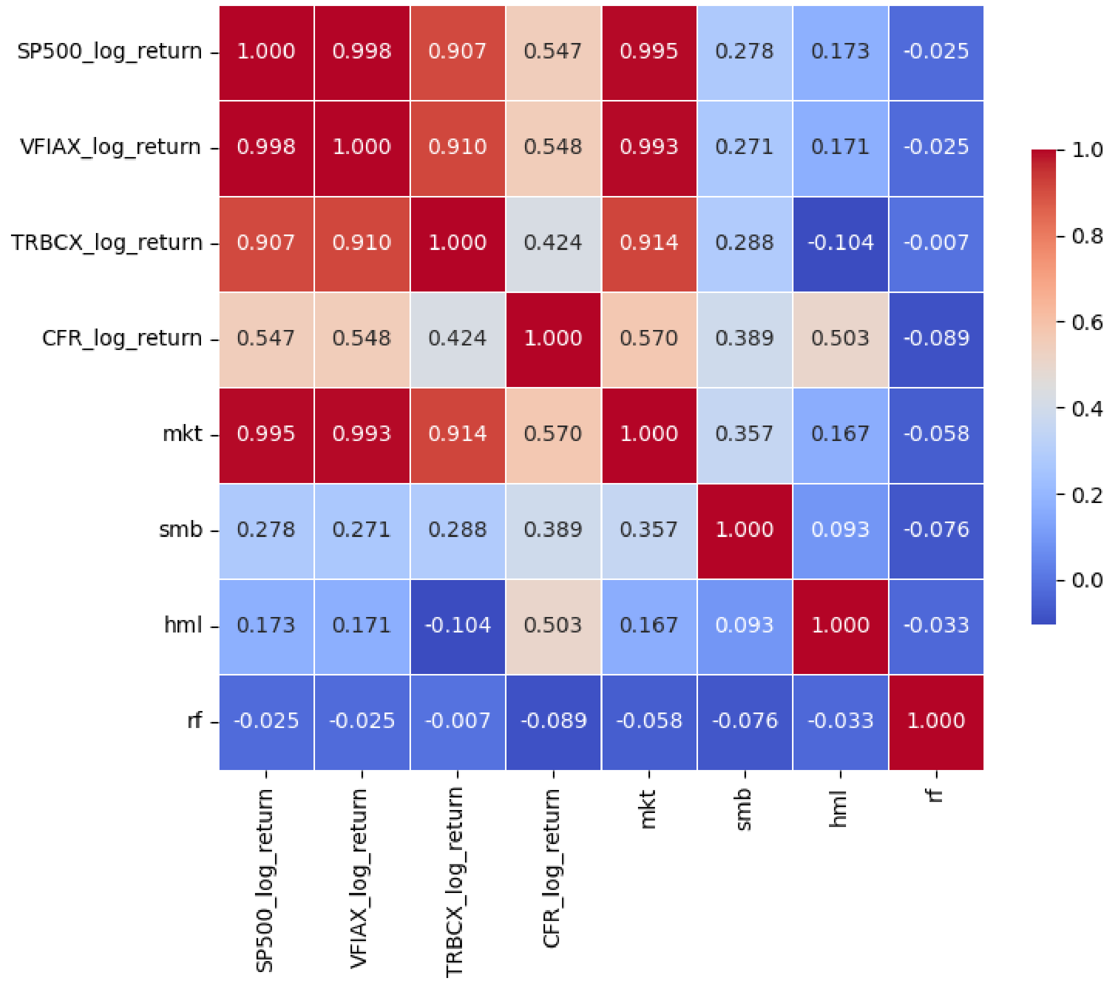
```python
print(' Descriptive Stats '.center(75, '-'))
print(m_data.describe())
```

```
----------------------------- Merge Data Info -----------------------------
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 238 entries, 2004-03-01 to 2023-12-01
Freq: MS
Data columns (total 8 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   SP500_log_return   238 non-null    float64
 1   VFIAX_log_return   238 non-null    float64
 2   TRBCX_log_return   238 non-null    float64
 3   CFR_log_return     238 non-null    float64
 4   mkt                238 non-null    float64
 5   smb                238 non-null    float64
 6   hml                238 non-null    float64
 7   rf                 238 non-null    float64
dtypes: float64(8)
memory usage: 16.7 KB
None
---------------------------- Descriptive Stats ----------------------------
       SP500_log_return  VFIAX_log_return  TRBCX_log_return  CFR_log_return  \
count        238.000000        238.000000        238.000000      238.000000
mean           0.005996          0.005980          0.006847        0.003946
std            0.043528          0.043690          0.051251        0.068460
min           -0.185634         -0.183720         -0.207890       -0.340102
25%           -0.017652         -0.018565         -0.015448       -0.034220
50%            0.012062          0.013341          0.013589        0.009644
75%            0.032009          0.032012          0.037848        0.045476
max            0.119421          0.120611          0.139681        0.253125

              mkt         smb         hml          rf
count  238.000000  238.000000  238.000000  238.000000
mean     0.006585    0.000056   -0.001055    0.001125
std      0.044968    0.024632    0.031944    0.001423
min     -0.189105   -0.061237   -0.149312    0.000000
25%     -0.018291   -0.018088   -0.017808    0.000000
50%      0.011780    0.000650   -0.002202    0.000200
75%      0.032540    0.015258    0.013928    0.001798
max      0.127953    0.071017    0.120003    0.004689
```

[ ]:

```python
corr_matrix = m_data.corr()

plt.figure(figsize=(8,8))
sns.heatmap(corr_matrix, annot=True, fmt=".3f", cmap='coolwarm',
            square=True, linewidths=0.5, cbar_kws={'shrink': 0.5})
```

```
plt.show()
```



Compute betas.

```
[ ]: def y_data(risky_asset, dataframe):
         y = dataframe[risky_asset]
         y.name = f'{risky_asset}'
         return y
```

```
[ ]: m_data.index = m_data.index.strftime('%Y-%m')
```

```
[ ]: risky_assets = ['VFIAX_log_return', 'TRBCX_log_return', 'CFR_log_return']

     for risk in risky_assets:
         m_data[f'Ex_{risk}'] = m_data[risk] - m_data['rf']
     m_data.columns = m_data.columns.str.replace('_log_return', '_rtn')
     m_data.columns = m_data.columns.str.lower()
```

[ ]:
```
m_data.info()
```

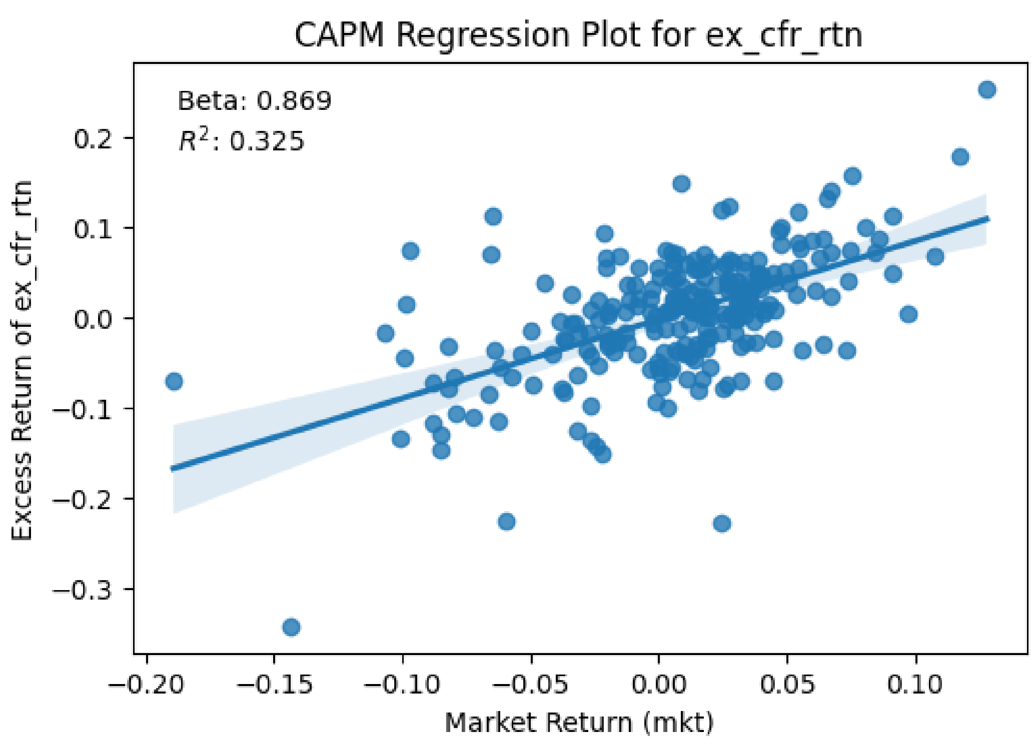
```
<class 'pandas.core.frame.DataFrame'>
Index: 238 entries, 2004-03 to 2023-12
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   sp500_rtn     238 non-null    float64
 1   vfiax_rtn     238 non-null    float64
 2   trbcx_rtn     238 non-null    float64
 3   cfr_rtn       238 non-null    float64
 4   mkt           238 non-null    float64
 5   smb           238 non-null    float64
 6   hml           238 non-null    float64
 7   rf            238 non-null    float64
 8   ex_vfiax_rtn  238 non-null    float64
 9   ex_trbcx_rtn  238 non-null    float64
 10  ex_cfr_rtn    238 non-null    float64
dtypes: float64(11)
memory usage: 22.3+ KB
```

CAPM Model

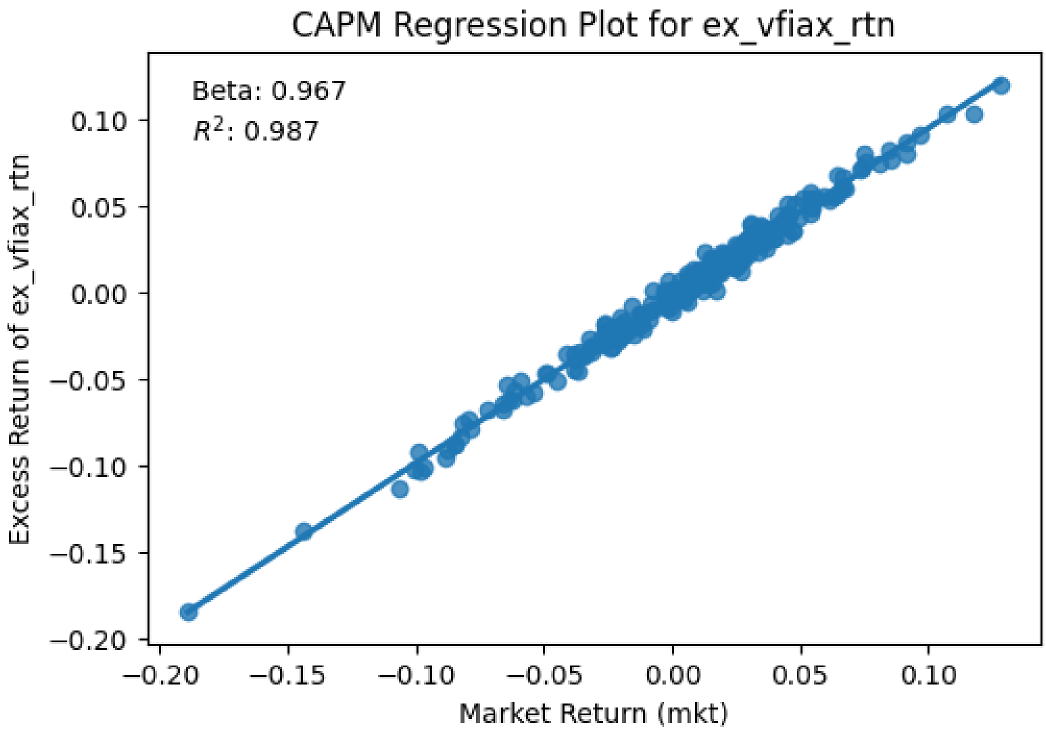
[ ]:
```
# Full regression results.

asset_analysis = ['ex_cfr_rtn', 'ex_vfiax_rtn', 'ex_trbcx_rtn']

for column in asset_analysis:
    Y = column
    X = 'mkt'
    capm_model = smf.ols(formula=f'{Y} ~ {X}', data=m_data).fit()
    print(f'CAPM Regression Results for {column}'.center(80))
    print(capm_model.summary())

    beta = capm_model.params['mkt']
    r_squared = capm_model.rsquared

    plt.figure(figsize=(6, 4))
    sns.regplot(x='mkt', y=column, data=m_data)
    plt.title(f'CAPM Regression Plot for {column}')
    plt.xlabel('Market Return (mkt)')
    plt.ylabel(f'Excess Return of {column}')
    plt.annotate(f'Beta: {beta:.3f}\n$R^2$: {r_squared:.3f}', xy=(0.05,0.85),
                 xycoords='axes fraction', fontsize=10)
    plt.show()
```

```
                     CAPM Regression Results for ex_cfr_rtn
                            OLS Regression Results
==============================================================================
```

```
Dep. Variable:             ex_cfr_rtn   R-squared:                       0.325
Model:                            OLS   Adj. R-squared:                  0.322
Method:                 Least Squares   F-statistic:                     113.4
Date:                Tue, 27 Feb 2024   Prob (F-statistic):           7.00e-22
Time:                        10:57:55   Log-Likelihood:                 347.20
No. Observations:                 238   AIC:                            -690.4
Df Residuals:                     236   BIC:                            -683.5
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0029      0.004     -0.784      0.434      -0.010       0.004
mkt            0.8691      0.082     10.650      0.000       0.708       1.030
==============================================================================
Omnibus:                       23.154   Durbin-Watson:                   1.887
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               57.445
Skew:                          -0.412   Prob(JB):                     3.36e-13
Kurtosis:                       5.262   Cond. No.                         22.3
==============================================================================

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

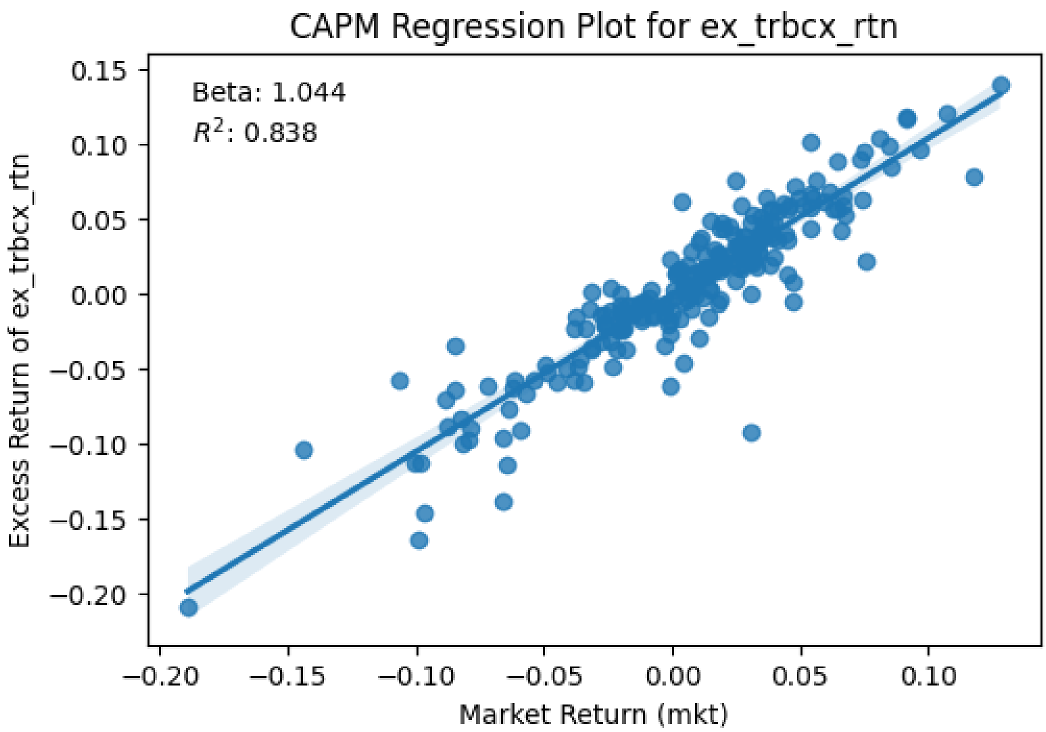


```
                  CAPM Regression Results for ex_vfiax_rtn
                            OLS Regression Results
==============================================================================
Dep. Variable:           ex_vfiax_rtn   R-squared:                       0.987
Model:                            OLS   Adj. R-squared:                  0.987
Method:                 Least Squares   F-statistic:                 1.854e+04
Date:                Tue, 27 Feb 2024   Prob (F-statistic):         2.68e-226
Time:                        10:57:55   Log-Likelihood:                 928.39
No. Observations:                 238   AIC:                            -1853.
Df Residuals:                     236   BIC:                            -1846.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0015      0.000     -4.691      0.000      -0.002      -0.001
mkt            0.9668      0.007    136.174      0.000       0.953       0.981
==============================================================================
Omnibus:                        0.807   Durbin-Watson:                   1.992
Prob(Omnibus):                  0.668   Jarque-Bera (JB):                0.919
Skew:                           0.090   Prob(JB):                        0.632
Kurtosis:                       2.754   Cond. No.                         22.3
==============================================================================

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




```
                    CAPM Regression Results for ex_trbcx_rtn
                            OLS Regression Results
==============================================================================
Dep. Variable:           ex_trbcx_rtn   R-squared:                       0.838
Model:                            OLS   Adj. R-squared:                  0.837
Method:                 Least Squares   F-statistic:                     1222.
Date:                Tue, 27 Feb 2024   Prob (F-statistic):           2.77e-95
Time:                        10:57:55   Log-Likelihood:                 586.43
No. Observations:                 238   AIC:                            -1169.
Df Residuals:                     236   BIC:                            -1162.
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0012      0.001     -0.851      0.396      -0.004       0.002
mkt            1.0440      0.030     34.953      0.000       0.985       1.103
==============================================================================
Omnibus:                       77.364   Durbin-Watson:                   1.628
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              411.476
Skew:                          -1.169   Prob(JB):                     4.46e-90
Kurtosis:                       9.003   Cond. No.                         22.3
==============================================================================
```

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



[ ]:
```python
# Betas from regression.

for column in asset_analysis:
    Y = column
    X = 'mkt'
    capm_model = smf.ols(formula=f'{Y} ~ {X}', data=m_data).fit()
    beta = capm_model.params['mkt']
    r_squared = capm_model.rsquared
    alpha = capm_model.params['Intercept']
    print(f'  FF Beta Stats {column}  '.center(40, '=') + '\n')
    print(f'Beta: {beta:.3f}')
    print(f'R-Squared: {r_squared:.3f}')
    print(f'Alpha: {alpha:.3f}\n')
    print(''.center(40, '=') + '\n')
```

```
======  FF Beta Stats ex_cfr_rtn  ======

Beta: 0.869
R-Squared: 0.325
```

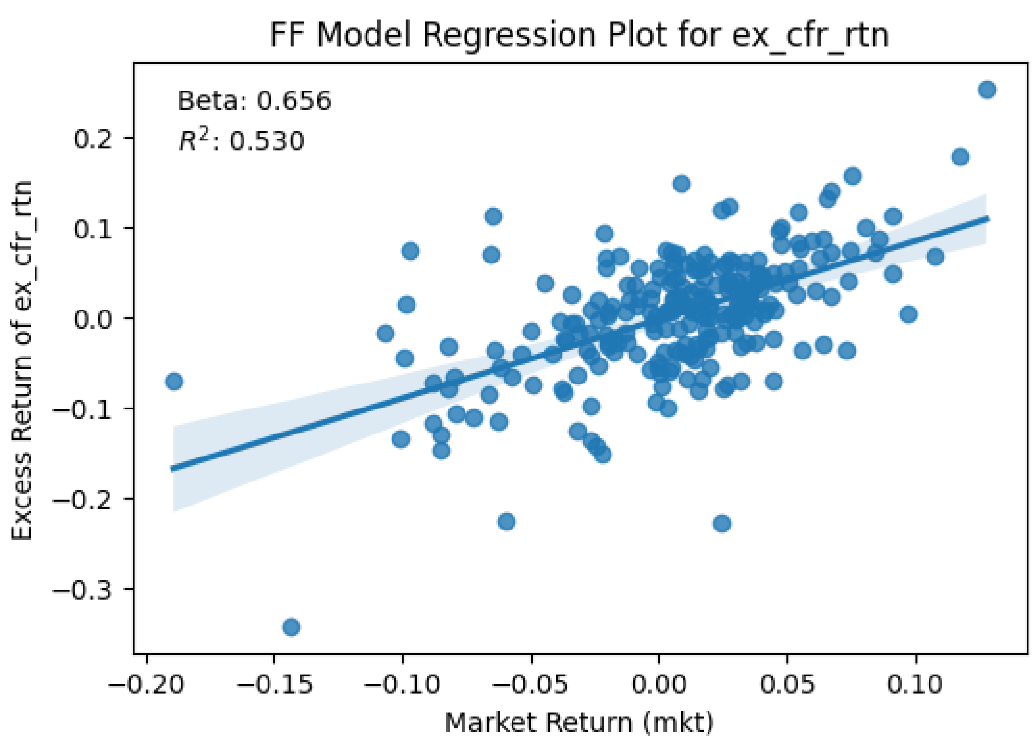
```
Alpha: -0.003

========================================

=====  FF Beta Stats ex_vfiax_rtn  =====

Beta: 0.967
R-Squared: 0.987
Alpha: -0.002

========================================

=====  FF Beta Stats ex_trbcx_rtn  =====

Beta: 1.044
R-Squared: 0.838
Alpha: -0.001

========================================
```

Fama and French 3 factor model.

[ ]:
```python
for column in asset_analysis:
    Y = column
    X = 'mkt + smb + hml'
    ff_model = smf.ols(formula=f'{Y} ~ {X}', data=m_data).fit()
    print(f'3 Factor Model Regression Results for {column}'.center(80))
    print(ff_model.summary())

    beta = ff_model.params['mkt']
    r_squared = ff_model.rsquared

    plt.figure(figsize=(6, 4))
    sns.regplot(x='mkt', y=column, data=m_data)
    plt.title(f'FF Model Regression Plot for {column}')
    plt.xlabel('Market Return (mkt)')
    plt.ylabel(f'Excess Return of {column}')
    plt.annotate(f'Beta: {beta:.3f}\n$R^2$: {r_squared:.3f}', xy=(0.05,0.85),
                 xycoords='axes fraction', fontsize=10)
    plt.show()
```

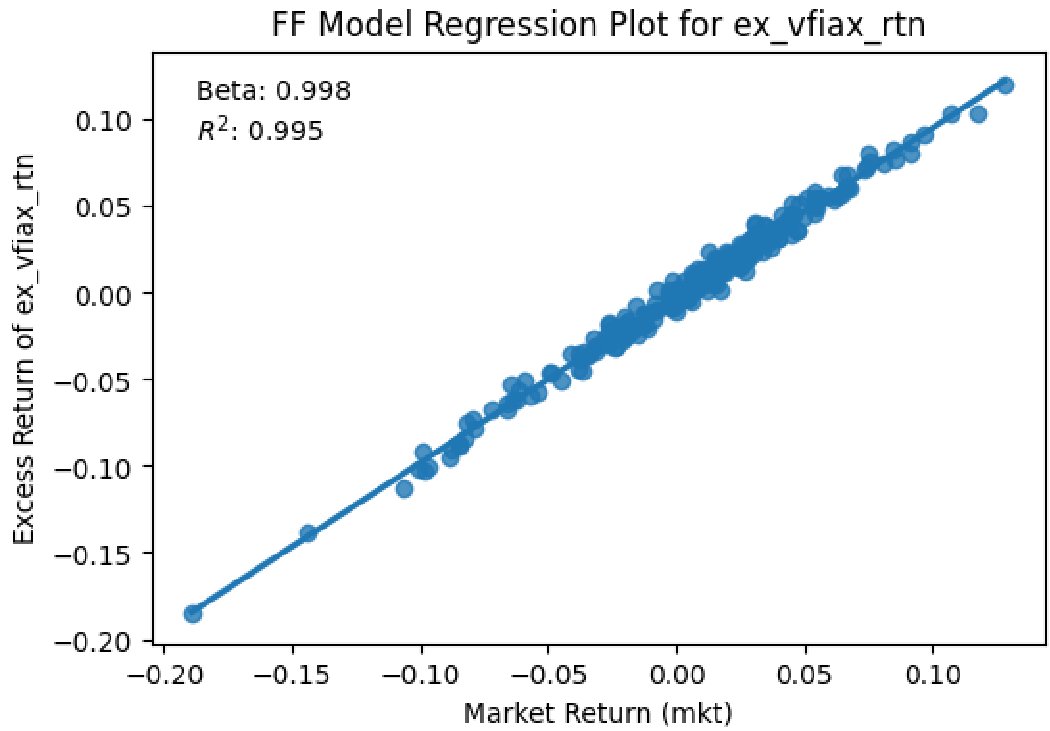
```
                3 Factor Model Regression Results for ex_cfr_rtn
                            OLS Regression Results
==============================================================================
Dep. Variable:             ex_cfr_rtn   R-squared:                       0.530
Model:                            OLS   Adj. R-squared:                  0.524
Method:                 Least Squares   F-statistic:                     87.82
Date:                Tue, 27 Feb 2024   Prob (F-statistic):           4.25e-38
```

```
Time:                        10:57:55   Log-Likelihood:                 390.25
No. Observations:                 238   AIC:                            -772.5
Df Residuals:                     234   BIC:                            -758.6
Df Model:                           3
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0006      0.003     -0.192      0.848      -0.007       0.006
mkt            0.6561      0.074      8.867      0.000       0.510       0.802
smb            0.5521      0.134      4.127      0.000       0.289       0.816
hml            0.8856      0.098      9.062      0.000       0.693       1.078
==============================================================================
Omnibus:                       10.709   Durbin-Watson:                   1.897
Prob(Omnibus):                  0.005   Jarque-Bera (JB):               20.312
Skew:                          -0.167   Prob(JB):                     3.88e-05
Kurtosis:                       4.391   Cond. No.                         44.7
==============================================================================

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



3 Factor Model Regression Results for ex_vfiax_rtn

```
                            OLS Regression Results
==============================================================================
Dep. Variable:            ex_vfiax_rtn   R-squared:                       0.995
Model:                             OLS   Adj. R-squared:                  0.995
Method:                  Least Squares   F-statistic:                 1.609e+04
Date:                 Tue, 27 Feb 2024   Prob (F-statistic):          1.13e-270
Time:                         10:57:55   Log-Likelihood:                 1042.3
No. Observations:                  238   AIC:                            -2077.
Df Residuals:                      234   BIC:                            -2063.
Df Model:                            3
Covariance Type:             nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0017      0.000     -8.431      0.000      -0.002      -0.001
mkt            0.9980      0.005    208.827      0.000       0.989       1.007
smb           -0.1671      0.009    -19.340      0.000      -0.184      -0.150
hml            0.0120      0.006      1.905      0.058      -0.000       0.024
==============================================================================
Omnibus:                        0.024   Durbin-Watson:                   2.168
Prob(Omnibus):                  0.988   Jarque-Bera (JB):                0.052
Skew:                          -0.023   Prob(JB):                        0.974
Kurtosis:                       2.945   Cond. No.                         44.7
==============================================================================

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



3 Factor Model Regression Results for ex_trbcx_rtn

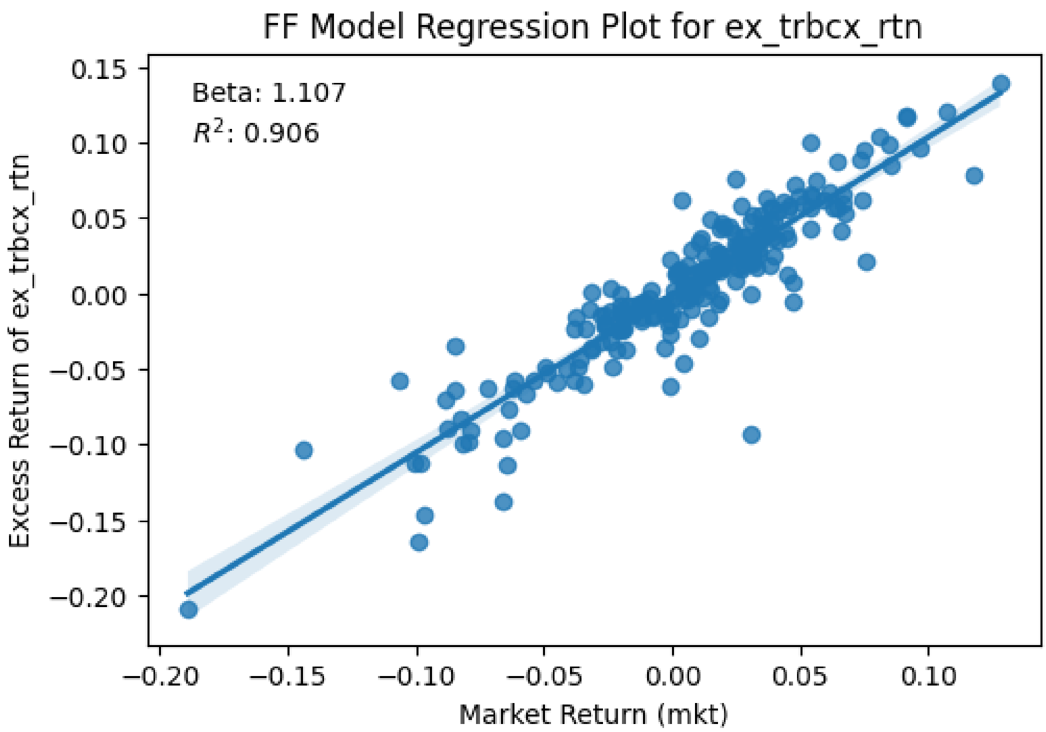
OLS Regression Results

Dep. Variable:	ex_trbcx_rtn	R-squared:	0.906
Model:	OLS	Adj. R-squared:	0.905
Method:	Least Squares	F-statistic:	755.9
Date:	Tue, 27 Feb 2024	Prob (F-statistic):	4.69e-120
Time:	10:57:56	Log-Likelihood:	651.72
No. Observations:	238	AIC:	-1295.
Df Residuals:	234	BIC:	-1282.
Df Model:	3		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	-0.0020	0.001	-1.936	0.054	-0.004	3.52e-05
mkt	1.1070	0.025	44.883	0.000	1.058	1.156
smb	-0.0669	0.045	-1.500	0.135	-0.155	0.021
hml	-0.4211	0.033	-12.928	0.000	-0.485	-0.357

Omnibus:	133.229	Durbin-Watson:	1.891
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1313.679
Skew:	-1.990	Prob(JB):	5.47e-286

```
Kurtosis:                      13.799   Cond. No.                         44.7
==============================================================================

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



[ ]:
```
for column in asset_analysis:
    Y = column
    X = 'mkt + smb + hml'
    ff_model = smf.ols(formula=f'{Y} ~ {X}', data=m_data).fit()
    beta = ff_model.params['mkt']
    r_squared = ff_model.rsquared
    alpha = ff_model.params['Intercept']
    print(f'  FF Beta Stats {column}  '.center(40, '=') + '\n')
    print(f'Beta: {beta:.3f}')
    print(f'R-Squared: {r_squared:.3f}')
    print(f'Alpha: {alpha:.3f}\n')
    print(''.center(40, '=') + '\n')
```

```
======  FF Beta Stats ex_cfr_rtn  ======

Beta: 0.656
R-Squared: 0.530
```

```
Alpha: -0.001

=========================================

=====  FF Beta Stats ex_vfiax_rtn  =====

Beta: 0.998
R-Squared: 0.995
Alpha: -0.002

=========================================

=====  FF Beta Stats ex_trbcx_rtn  =====

Beta: 1.107
R-Squared: 0.906
Alpha: -0.002

=========================================
```

Results interpretation. The assets analyzed were Frost banks stock, the Vanguard 500 Index Fund Admiral Shares, and the T. Rowe Price Blue Chip Growth mutual fund.

After running a CAPM regression and a Fama and French 3 factor regression, it is apparent that the index and mutual fund mimick the market relatively close, but all underperformed the market when reviwing their alpha values. I would like to note that the alpha values are in some instances not significant at the 5% level, and should not be considred strong indicators.

Because the index and mutual fund nore closely track the market, their beta's show they are subject to similar sytematic risk of the market. Frost bank on the other hand has a beta less than one, indicating that its vaulation is not tied so closely to the market. This makes sense when considering portfolio construction, especially for the index fund, is trying to mimick the market as best it can.

The R^2 values of each asset are also shown to increase when moving from the CAPM model to the 3 Factor model. This also make sense, as the Fama-French model includes additional parameters, which will boost the R^2 value due to the reduction in unexplained residuals.