In [30]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import yfinance as yf
import statsmodels.api as sm
from pandas_datareader import data as pdr

yf.pdr_override()

# Read the data
FF3F = pd.read_csv('FF_RDF_daily-2015-2019.csv', index_col=0, parse_dates=['date'])
pIUSA = pdr.get_data_yahoo("IUSA.AS", start="2015-01-01", end="2019-05-31").Close.rename('IUSA')
```

```
[**********************100%**********************]  1 of 1 downloaded
```

In [31]:
```
# Calculate the daily returns
pIUSA = pIUSA.pct_change() * 100
pIUSA = pIUSA.dropna()

# Combine the datasets
data=pd.merge(FF3F, pIUSA, left_on = 'date', right_index=True, how = 'left').dropna()
data.tail()
```

Out[31]:

date	market_excess_returns	SMB	HML	risk_free_rate	IUSA
2019-05-24	0.23	0.51	0.29	0.009	-0.158667
2019-05-28	-0.78	0.26	-0.60	0.009	0.000000
2019-05-29	-0.71	-0.29	0.44	0.009	-1.268331
2019-05-30	0.14	-0.28	-0.78	0.009	0.200723
2019-05-31	-1.37	-0.15	-0.22	0.009	-0.721154

In [32]:
```
# Import the statsmodels library
import statsmodels.api as sm

# Independent variables
X = data[['market_excess_returns', 'SMB', 'HML']]

# Dependent variable
y = data['IUSA']

# Create a regression model
reg = sm.OLS(y, X).fit()

# Print beta of market_excess_returns
print('Beta of market_excess_returns= ', reg.params[0])

# Print beta of SMB
print('Beta of SMB= ', reg.params[1])

# Print beta of HML
print('Beta of HML= ', reg.params[2])
```

```
Beta of market_excess_returns=  0.5713500004273582
Beta of SMB=  0.05814048394876629
Beta of HML=  0.13234240840837946
```

In [33]:
```
#Annualize returns
annualised_market_excess_returns = (
    (data.market_excess_returns/100 + 1).cumprod()[-1]) ** (252/len(data)) - 1

annualised_SMB_returns = ((data.SMB/100 + 1).cumprod()
                          [-1]) ** (252/len(data)) - 1

annualised_HML_returns = ((data.HML/100 + 1).cumprod()
                          [-1]) ** (252/len(data)) - 1

annualised_risk_free_rate_returns = (
    data.risk_free_rate/100 + 1).cumprod()[-1] ** (252/len(data)) - 1
```

In [34]:
```
# Calculate the expected returns of IUSA
expected_returns_IUSA = annualised_risk_free_rate_returns + \
    reg.params[0] * annualised_market_excess_returns + reg.params[1] * \
    annualised_SMB_returns + reg.params[2] * annualised_HML_returns

# Print expected returns of IUSA
print('Expected Returns of IUSA: %.2f' %(expected_returns_IUSA * 100))
```

```
Expected Returns of IUSA: 2.81
```

In [ ]: