Industry Picking Strategy

In [1]:
```
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import openpyxl as p
import yfinance as yf
from statsmodels.formula.api import ols
import statsmodels.formula.api as smf
import pickle
import statsmodels.stats.api as sms
from statsmodels.compat import lzip
```

In [2]:
```
indus = pd.read_csv('11 sectors.csv')
indus = indus.set_index('Date')
indus5 = indus.loc['201701':'201912']
x = indus5.std()
x = x.sort_values()
most_stable_indus = x.index[0:5]

print('The 1st stable industry from Jan. 2017 to Dec. 2019 is ' + most_stable_ir
print('The 2nd stable industry from Jan. 2017 to Dec. 2019 is ' + most_stable_ir
print('The 3rd stable industry from Jan. 2017 to Dec. 2019 is ' + most_stable_ir
print('The 4th stable industry from Jan. 2017 to Dec. 2019 is ' + most_stable_ir
print('The 5th stable industry from Jan. 2017 to Dec. 2019 is ' + most_stable_ir
```

```
The 1st stable industry from Jan. 2017 to Dec. 2019 is Utilities Industry.
The 2nd stable industry from Jan. 2017 to Dec. 2019 is Communication Services In
dustry.
The 3rd stable industry from Jan. 2017 to Dec. 2019 is Financials Industry.
The 4th stable industry from Jan. 2017 to Dec. 2019 is Information Technology In
dustry.
The 5th stable industry from Jan. 2017 to Dec. 2019 is Health Care Industry.
```

Industry Picking Strategy

In [3]:
```
# Get S&P500 stocks symbols from Wikipedia and set column 'Symbol' as tickers.
sp500url = 'https://en.wikipedia.org/wiki/List_of_S%26P_500_companies'
data_table = pd.read_html(sp500url)
tickers = data_table[0]

# Use for loop to adjust the format of the stock code in the 'Symbol' column.
for i in range(len(tickers)):
    if tickers['Symbol'][i] == 'BRK.B':
        tickers['Symbol'][i] == 'BRK-B'
    elif tickers['Symbol'][i] == 'BF.B':
        tickers['Symbol'][i] == 'BF-B'

display(tickers)
```

	Symbol	Security	SEC filings	GICS Sector	GICS Sub-Industry	Headquarters Location	Date first added	CIK
0	MMM	3M	reports	Industrials	Industrial Conglomerates	Saint Paul, Minnesota	1976-08-09	6674(
1	AOS	A. O. Smith	reports	Industrials	Building Products	Milwaukee, Wisconsin	2017-07-26	91142
2	ABT	Abbott	reports	Health Care	Health Care Equipment	North Chicago, Illinois	1964-03-31	180(
3	ABBV	AbbVie	reports	Health Care	Pharmaceuticals	North Chicago, Illinois	2012-12-31	155115
4	ACN	Accenture	reports	Information Technology	IT Consulting & Other Services	Dublin, Ireland	2011-07-06	146737
...	...	...	...	...	...	...	...	..
498	YUM	Yum! Brands	reports	Consumer Discretionary	Restaurants	Louisville, Kentucky	1997-10-06	104106
499	ZBRA	Zebra Technologies	reports	Information Technology	Electronic Equipment & Instruments	Lincolnshire, Illinois	2019-12-23	87721
500	ZBH	Zimmer Biomet	reports	Health Care	Health Care Equipment	Warsaw, Indiana	2001-08-07	113686
501	ZION	Zions Bancorporation	reports	Financials	Regional Banks	Salt Lake City, Utah	2001-06-22	10938(
502	ZTS	Zoetis	reports	Health Care	Pharmaceuticals	Parsippany, New Jersey	2013-06-21	155528(

503 rows × 9 columns

In [4]:
```
df = pd.read_csv('daily.csv')
df['Date'] = pd.to_datetime(df['Date'])
df = df.set_index('Date')
df = df.loc['2017-01-01':'2019-12-31']
```

In [5]:
```
chosen_stock = []

for j in range(len(most_stable_indus)):
    sector = tickers.loc[tickers['GICS Sector'] == most_stable_indus[int('{}'.fo
    prices = yf.download(sector, start = '2017-01-01', end = '2019-12-31')['Adj
    prices = prices.dropna(axis = 1)
    prices.index = pd.to_datetime(prices.index)
    sector = prices.columns.values
    result = df.merge(prices, on = ['Date'], how = 'inner')

    alphas_list = []
    std0_list = []
    beta1_list = []
    std1_list = []
    beta2_list = []
```

```
    std2_list = []
    beta3_list = []
    std3_list = []
    beta4_list = []
    std4_list = []
    f_test_result = []

    for i in sector:
        result['ret_{}'.format(i)] = np.log(result['{}'.format(i)]/result['{}'.f
        result['retrf_{}'.format(i)] = result['ret_{}'.format(i)] - np.log(resul
        result = result.drop(columns = ['{}'.format(i)])

        # Fama-French Model
        formula = 'retrf_{} ~ mktrf + smb + hml + mom'.format(i)
        lr = ols(formula, result).fit()
        alphas_list.append(lr.params['Intercept'])
        std0_list.append(lr.bse['Intercept'])
        beta1_list.append(lr.params['mktrf'])
        std1_list.append(lr.bse['mktrf'])
        beta2_list.append(lr.params['smb'])
        std2_list.append(lr.bse['smb'])
        beta3_list.append(lr.params['hml'])
        std3_list.append(lr.bse['smb'])
        beta4_list.append(lr.params['mom'])
        std4_list.append(lr.bse['mom'])
        hypotheses = '(smb=0), (hml=0), (mom=0)'
        f_test_result.append(lr.f_test(hypotheses))

    final_result = pd.DataFrame(index=["Portfolio_{}".format(i) for i in sector]
                          data={
                            "alpha":alphas_list,
                            "alpha_standard_error": std0_list,
                            "beta1":beta1_list,
                            "beta1_standard_error": std1_list,
                            "beta2":beta2_list,
                            "beta2_standard_error": std2_list,
                            "beta3":beta3_list,
                            "beta3_standard_error": std3_list,
                            "beta4":beta4_list,
                            "beta4_standard_error": std4_list})

    final_result['predict'] = final_result['alpha'] + final_result['beta1'] * re
    final_result = final_result['predict'].sort_values()
    chosen_stock.append(final_result.index[-1].split('_')[1])
    chosen_stock.append(final_result.index[-2].split('_')[1])
```

```
[**********************100%**************************]  30 of 30 completed

1 Failed download:
- CEG: Data doesn't exist for startDate = 1483246800, endDate = 1577768400
[**********************100%**************************]  25 of 25 completed
[**********************100%**************************]  67 of 67 completed

1 Failed download:
- BRK.B: No data found for this date range, symbol may be delisted
[**********************100%**************************]  76 of 76 completed
[**********************100%**************************]  63 of 63 completed

1 Failed download:
- OGN: Data doesn't exist for startDate = 1483246800, endDate = 1577768400
```

In [6]:
```
chosen_stock
```

Out[6]:
```
['NRG', 'NEE', 'MTCH', 'LYV', 'MSCI', 'SPGI', 'ENPH', 'SEDG', 'DXCM', 'VRTX']
```

In [7]:
```
download_stock = yf.download(chosen_stock, start = '2017-01-01', end = '2019-12-
chosen_result = df.merge(download_stock, on = ['Date'], how = 'inner')

for i in chosen_stock:
    chosen_result['ret_{}'.format(i)] = np.log(chosen_result['{}'.format(i)]/cho
    chosen_result['retrf_{}'.format(i)] = chosen_result['ret_{}'.format(i)] - np
    formula = 'retrf_{}~ mktrf + smb + hml + mom'.format(i)
    results = smf.ols(formula, chosen_result).fit()
    print()
    print('****************The Fama-French Model Regression of {}:**************
    print(results.summary())
```

```
[**********************100%**********************]  10 of 10 completed

****************The Fama-French Model Regression of NRG:*****************
                            OLS Regression Results
==============================================================================
Dep. Variable:              retrf_NRG   R-squared:                       0.115
Model:                            OLS   Adj. R-squared:                  0.110
Method:                 Least Squares   F-statistic:                     24.26
Date:                Fri, 23 Dec 2022   Prob (F-statistic):           6.78e-19
Time:                        16:48:54   Log-Likelihood:                 1944.8
No. Observations:                 752   AIC:                            -3880.
Df Residuals:                     747   BIC:                            -3857.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0052      0.001     -7.709      0.000      -0.006      -0.004
mktrf          0.0074      0.001      9.049      0.000       0.006       0.009
smb            0.0009      0.001      0.655      0.513      -0.002       0.004
hml           -0.0017      0.001     -1.267      0.205      -0.004       0.001
mom            0.0006      0.001      0.508      0.612      -0.002       0.003
==============================================================================
Omnibus:                      699.180   Durbin-Watson:                   1.658
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            74765.471
Skew:                           3.727   Prob(JB):                         0.00
Kurtosis:                      51.276   Cond. No.                         2.42
==============================================================================

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

****************The Fama-French Model Regression of NEE:*****************
                            OLS Regression Results
==============================================================================
Dep. Variable:              retrf_NEE   R-squared:                       0.117
Model:                            OLS   Adj. R-squared:                  0.112
Method:                 Least Squares   F-statistic:                     24.69
Date:                Fri, 23 Dec 2022   Prob (F-statistic):           3.20e-19
Time:                        16:48:54   Log-Likelihood:                 2484.5
```

```
No. Observations:                 752   AIC:                            -4959.
Df Residuals:                     747   BIC:                            -4936.
Df Model:                           4
Covariance Type:            nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept     -0.0055      0.000    -16.868      0.000      -0.006      -0.005
mktrf          0.0015      0.000      3.673      0.000       0.001       0.002
smb           -0.0044      0.001     -6.508      0.000      -0.006      -0.003
hml           -0.0031      0.001     -4.656      0.000      -0.004      -0.002
mom            0.0008      0.001      1.370      0.171      -0.000       0.002
==============================================================================
Omnibus:                       46.382   Durbin-Watson:                   1.889
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               86.415
Skew:                          -0.415   Prob(JB):                     1.72e-19
Kurtosis:                       4.439   Cond. No.                         2.42
==============================================================================

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

**********************The Fama-French Model Regression of MTCH:**********************
                            OLS Regression Results
==============================================================================
Dep. Variable:             retrf_MTCH   R-squared:                       0.197
Model:                            OLS   Adj. R-squared:                  0.192
Method:                 Least Squares   F-statistic:                     45.69
Date:                Fri, 23 Dec 2022   Prob (F-statistic):           2.34e-34
Time:                        16:48:54   Log-Likelihood:                 1686.9
No. Observations:                 752   AIC:                            -3364.
Df Residuals:                     747   BIC:                            -3341.
Df Model:                           4
Covariance Type:            nonrobust
================================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------
Intercept     -0.0051      0.001     -5.415      0.000      -0.007      -0.003
mktrf          0.0097      0.001      8.369      0.000       0.007       0.012
smb            0.0040      0.002      2.046      0.041       0.000       0.008
hml           -0.0096      0.002     -4.937      0.000      -0.013      -0.006
mom            0.0064      0.002      4.051      0.000       0.003       0.010
==============================================================================
Omnibus:                      343.976   Durbin-Watson:                   2.085
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            26666.858
Skew:                          -1.153   Prob(JB):                         0.00
Kurtosis:                      32.082   Cond. No.                         2.42
==============================================================================

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

**********************The Fama-French Model Regression of LYV:**********************
                            OLS Regression Results
==============================================================================
Dep. Variable:              retrf_LYV   R-squared:                       0.302
Model:                            OLS   Adj. R-squared:                  0.299
Method:                 Least Squares   F-statistic:                     80.98
```

```
Date:                Fri, 23 Dec 2022   Prob (F-statistic):           4.25e-57
Time:                        16:48:54   Log-Likelihood:                 2110.1
No. Observations:                 752   AIC:                            -4210.
Df Residuals:                     747   BIC:                            -4187.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0057      0.001    -10.578      0.000      -0.007      -0.005
mktrf          0.0102      0.001     15.380      0.000       0.009       0.011
smb            0.0026      0.001      2.324      0.020       0.000       0.005
hml           -0.0038      0.001     -3.444      0.001      -0.006      -0.002
mom            0.0020      0.001      2.258      0.024       0.000       0.004
==============================================================================
Omnibus:                      161.098   Durbin-Watson:                   1.819
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3199.846
Skew:                           0.379   Prob(JB):                         0.00
Kurtosis:                      13.077   Cond. No.                         2.42
==============================================================================

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

****************The Fama-French Model Regression of MSCI:****************
                            OLS Regression Results
==============================================================================
Dep. Variable:             retrf_MSCI   R-squared:                       0.463
Model:                            OLS   Adj. R-squared:                  0.460
Method:                 Least Squares   F-statistic:                     161.2
Date:                Fri, 23 Dec 2022   Prob (F-statistic):           2.06e-99
Time:                        16:48:54   Log-Likelihood:                 2305.1
No. Observations:                 752   AIC:                            -4600.
Df Residuals:                     747   BIC:                            -4577.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0054      0.000    -13.104      0.000      -0.006      -0.005
mktrf          0.0117      0.001     22.950      0.000       0.011       0.013
smb           -0.0004      0.001     -0.459      0.646      -0.002       0.001
hml           -0.0029      0.001     -3.436      0.001      -0.005      -0.001
mom            0.0025      0.001      3.597      0.000       0.001       0.004
==============================================================================
Omnibus:                       82.733   Durbin-Watson:                   1.934
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              412.817
Skew:                           0.348   Prob(JB):                     2.28e-90
Kurtosis:                       6.563   Cond. No.                         2.42
==============================================================================

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

****************The Fama-French Model Regression of SPGI:****************
                            OLS Regression Results
==============================================================================
Dep. Variable:             retrf_SPGI   R-squared:                       0.507
```

```
Model:                            OLS   Adj. R-squared:                  0.505
Method:                 Least Squares   F-statistic:                     192.4
Date:                Fri, 23 Dec 2022   Prob (F-statistic):          2.51e-113
Time:                        16:48:54   Log-Likelihood:                 2484.0
No. Observations:                 752   AIC:                            -4958.
Df Residuals:                     747   BIC:                            -4935.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0057      0.000    -17.433      0.000      -0.006      -0.005
mktrf          0.0103      0.000     25.595      0.000       0.009       0.011
smb           -0.0031      0.001     -4.594      0.000      -0.004      -0.002
hml           -0.0013      0.001     -1.979      0.048      -0.003   -1.07e-05
mom            0.0027      0.001      4.835      0.000       0.002       0.004
==============================================================================
Omnibus:                       90.309   Durbin-Watson:                   1.806
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              710.661
Skew:                           0.168   Prob(JB):                    4.81e-155
Kurtosis:                       7.751   Cond. No.                         2.42
==============================================================================

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

*****************The Fama-French Model Regression of ENPH:*****************
                            OLS Regression Results
==============================================================================
Dep. Variable:             retrf_ENPH   R-squared:                       0.092
Model:                            OLS   Adj. R-squared:                  0.088
Method:                 Least Squares   F-statistic:                     19.02
Date:                Fri, 23 Dec 2022   Prob (F-statistic):           6.59e-15
Time:                        16:48:54   Log-Likelihood:                 1223.0
No. Observations:                 752   AIC:                            -2436.
Df Residuals:                     747   BIC:                            -2413.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0025      0.002     -1.456      0.146      -0.006       0.001
mktrf          0.0098      0.002      4.556      0.000       0.006       0.014
smb            0.0208      0.004      5.699      0.000       0.014       0.028
hml           -0.0033      0.004     -0.906      0.365      -0.010       0.004
mom            0.0074      0.003      2.523      0.012       0.002       0.013
==============================================================================
Omnibus:                      180.307   Durbin-Watson:                   1.944
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1685.577
Skew:                           0.793   Prob(JB):                         0.00
Kurtosis:                      10.161   Cond. No.                         2.42
==============================================================================

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

*****************The Fama-French Model Regression of SEDG:*****************
                            OLS Regression Results
```

```
==============================================================================
Dep. Variable:             retrf_SEDG   R-squared:                       0.099
Model:                            OLS   Adj. R-squared:                  0.094
Method:                 Least Squares   F-statistic:                     20.46
Date:                Fri, 23 Dec 2022   Prob (F-statistic):           5.22e-16
Time:                        16:48:54   Log-Likelihood:                 1552.2
No. Observations:                 752   AIC:                            -3094.
Df Residuals:                     747   BIC:                            -3071.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0043      0.001     -3.807      0.000      -0.007      -0.002
mktrf          0.0096      0.001      6.908      0.000       0.007       0.012
smb            0.0067      0.002      2.857      0.004       0.002       0.011
hml           -0.0050      0.002     -2.162      0.031      -0.010      -0.000
mom            0.0018      0.002      0.964      0.335      -0.002       0.006
==============================================================================
Omnibus:                      368.674   Durbin-Watson:                   2.185
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             6003.694
Skew:                           1.795   Prob(JB):                         0.00
Kurtosis:                      16.369   Cond. No.                         2.42
==============================================================================

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

****************The Fama-French Model Regression of DXCM:*****************
                            OLS Regression Results
==============================================================================
Dep. Variable:             retrf_DXCM   R-squared:                       0.129
Model:                            OLS   Adj. R-squared:                  0.125
Method:                 Least Squares   F-statistic:                     27.73
Date:                Fri, 23 Dec 2022   Prob (F-statistic):           1.71e-21
Time:                        16:48:54   Log-Likelihood:                 1564.5
No. Observations:                 752   AIC:                            -3119.
Df Residuals:                     747   BIC:                            -3096.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0053      0.001     -4.749      0.000      -0.007      -0.003
mktrf          0.0094      0.001      6.892      0.000       0.007       0.012
smb            0.0088      0.002      3.814      0.000       0.004       0.013
hml           -0.0080      0.002     -3.516      0.000      -0.013      -0.004
mom            0.0029      0.002      1.576      0.116      -0.001       0.007
==============================================================================
Omnibus:                      381.261   Durbin-Watson:                   2.007
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            96692.664
Skew:                          -1.060   Prob(JB):                         0.00
Kurtosis:                      58.511   Cond. No.                         2.42
==============================================================================

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

```
**********The Fama-French Model Regression of VRTX:**********
                            OLS Regression Results
==============================================================================
Dep. Variable:              retrf_VRTX   R-squared:                       0.290
Model:                             OLS   Adj. R-squared:                  0.286
Method:                  Least Squares   F-statistic:                     76.34
Date:                 Fri, 23 Dec 2022   Prob (F-statistic):           2.81e-54
Time:                         16:48:54   Log-Likelihood:                 1970.6
No. Observations:                  752   AIC:                            -3931.
Df Residuals:                      747   BIC:                            -3908.
Df Model:                            4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0057      0.001     -8.839      0.000      -0.007      -0.004
mktrf          0.0111      0.001     13.982      0.000       0.010       0.013
smb            0.0021      0.001      1.584      0.114      -0.001       0.005
hml           -0.0086      0.001     -6.433      0.000      -0.011      -0.006
mom           -0.0001      0.001     -0.136      0.892      -0.002       0.002
==============================================================================
Omnibus:                      689.767   Durbin-Watson:                   1.948
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            43975.765
Skew:                           3.844   Prob(JB):                         0.00
Kurtosis:                      39.666   Cond. No.                         2.42
==============================================================================

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

Portfolio Chosen

In [8]:
```
def total_return(weights, calculated_stocks):
    return np.sum((weights * calculated_stocks.mean())*252)

def total_vol(weights, calculated_stocks):
    return math.sqrt(np.dot(weights.T, np.dot(calculated_stocks.cov()*252, weigh
```

In [9]:
```
for i in chosen_stock:
    download_stock['ret_{}'.format(i)] = np.log(download_stock['{}'.format(i)]/d
    download_stock = download_stock.drop(columns = ['{}'.format(i)])
```

In [10]:
```
rfrate = chosen_result['rf'].mean()

best_weight = []
best_sharpe = 0
stock_return = []
stock_vol = []

for j in range(0,100000):
    # Generate 10 weights for 10 stocks randomly
    weight = np.random.random(10)
    weight /= np.sum(weight)

    # Calculate the expected return and standard deviation of the portfolio
```

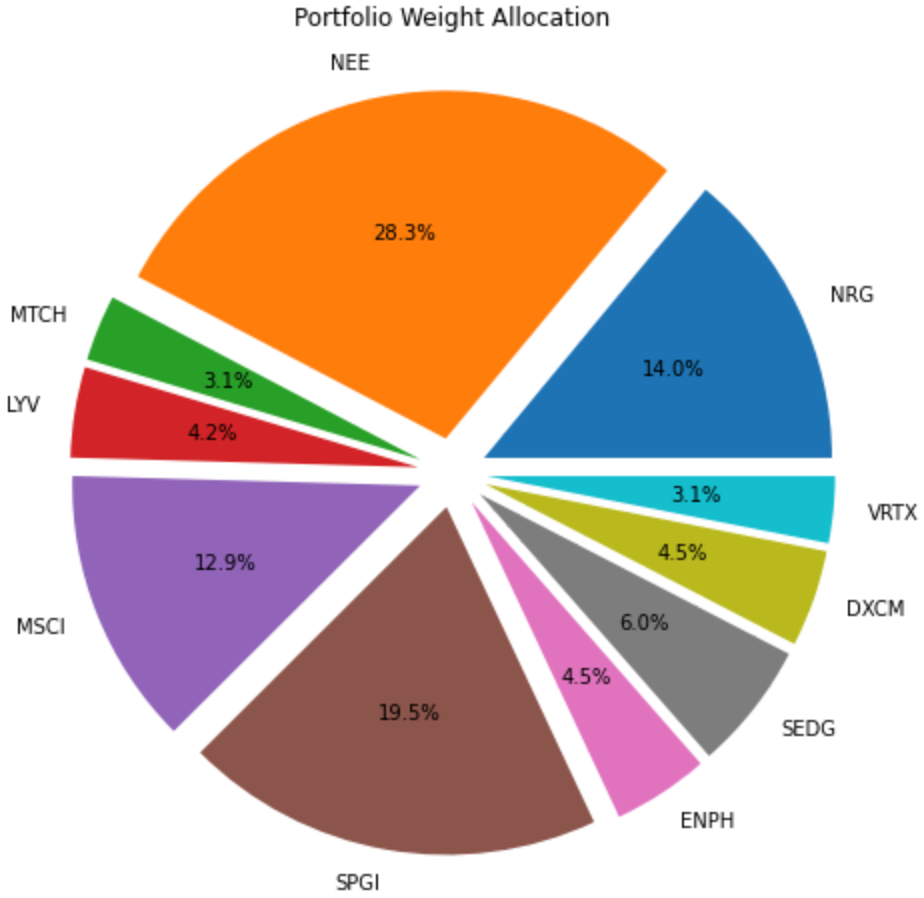
```
    stock_return.append(total_return(weight, download_stock))
    stock_vol.append(total_vol(weight, download_stock))

    # Calculate the sharpe ratio of each stock and find the highest portfolio
    sharpe_ratio = (total_return(weight, download_stock) - rfrate) / total_vol(w
    if sharpe_ratio >= best_sharpe:
        best_sharpe = sharpe_ratio
        best_weight = weight

best_return = total_return(best_weight, download_stock)
best_vol = total_vol(best_weight, download_stock)
best_sharpe_ratio = (best_return - rfrate) / best_vol

# Plot the pie charts
plt.rcParams['figure.figsize'] = [12,8]
plt.pie(best_weight, explode = np.ones(10)*0.1, labels = chosen_stock, autopct =
plt.title('Portfolio Weight Allocation')
plt.show()

# Print out the final result
print(f'The tangency portfolio invests {best_weight[0]:.1%} into {chosen_stock[0
```



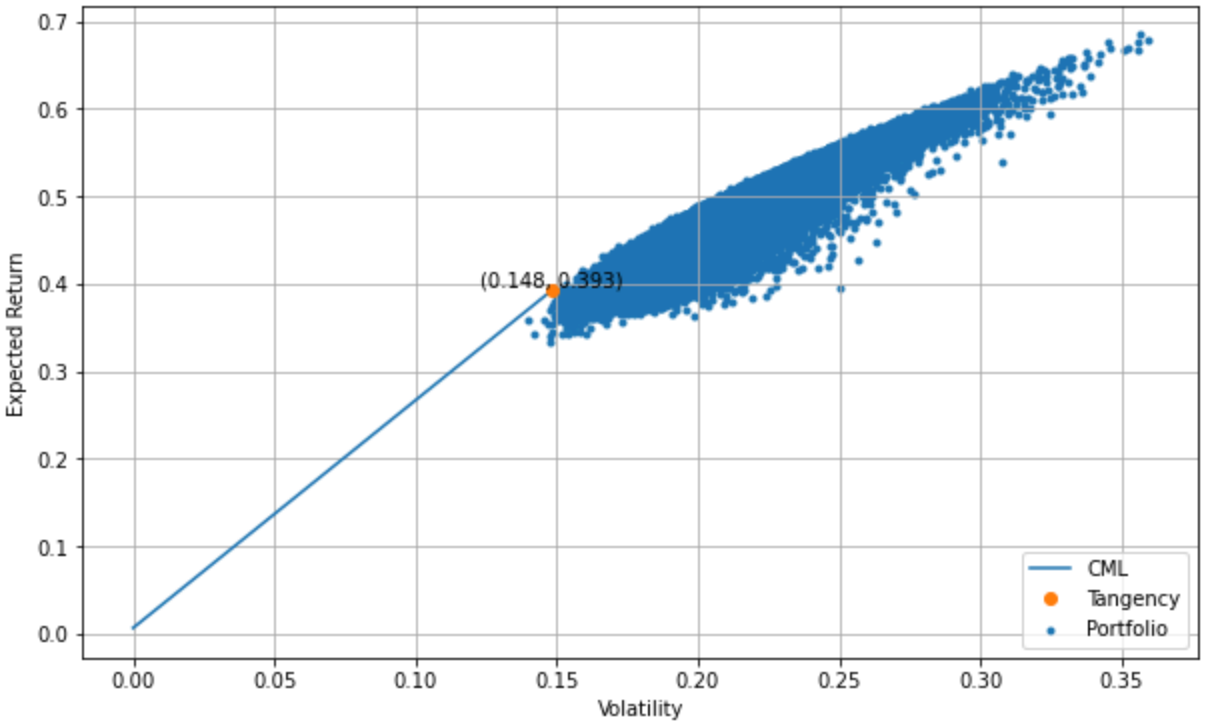

```
The tangency portfolio invests 14.0% into NRG, 28.3% into NEE, 3.1% into MTCH,
4.2% into LYV, 12.9% into MSCI, 19.5% into SPGI, 4.5% into ENPH, 6.0% into SEDG,
4.5% into DXCM, and 3.1% into VRTX achieving a volatility 14.83% and expected re
turn of 39.25% for a Sharpe Ratio of 2.604.
```

In [11]:

```
result_shown = pd.DataFrame([[best_return, best_vol, best_sharpe_ratio]],columns
result_shown.index = ['Result']
display(result_shown)
```

```
plt.rcParams['figure.figsize'] = [10,6]
plt.scatter(stock_vol , stock_return, marker = '.', label='Portfolio')
plt.plot(np.array([0, best_vol]), np.array([rfrate, best_return]), label='CML')
plt.plot(best_vol,best_return,'o',label='Tangency')
plt.text(best_vol,best_return,(round(best_vol,3),round(best_return,3)), ha = 'ce
plt.grid('on')
plt.legend(loc='lower right')
plt.xlabel('Volatility')
plt.ylabel('Expected Return')
plt.show()
```

	Return	Volatility	Sharpe Ratio
Result	0.392534	0.148322	2.604093



Fama_French Regression

In [12]:

```
a = pd.DataFrame(np.sum(download_stock*best_weight, axis = 1), columns = ['total
final_fama = chosen_result.merge(a, on = ['Date'], how = 'inner')
final_fama.head()
```

Out[12]:

Date	mktrf	smb	hml	rf	mom	DXCM	ENPH	LYV	MSCI	MTCH	...	ret
2017-01-03	0.83	-0.13	0.05	0.002	-0.61	14.5625	1.05	27.400000	74.041428	16.494013	...	
2017-01-04	0.79	0.95	-0.16	0.002	-0.37	15.1800	1.15	27.629999	75.922829	16.455944	...	C

Date	mktrf	smb	hml	rf	mom	DXCM	ENPH	LYV	MSCI	MTCH	...	ret
2017-01-05	-0.21	-0.88	-0.79	0.002	-0.60	15.6650	1.12	27.500000	76.797646	17.122175	...	0
2017-01-06	0.29	-0.66	-0.31	0.002	-0.18	15.8075	1.11	27.680000	78.453247	17.160246	...	(
2017-01-09	-0.37	-0.29	-1.04	0.002	-0.37	15.6225	1.11	27.290001	77.775955	17.274456	...	-(

5 rows × 36 columns

In [13]:
```
final_fama['total_returnrf'] = final_fama['total_return'] - np.log(final_fama['r
formula = 'total_returnrf~ mktrf + smb + hml + mom'
results = smf.ols(formula, final_fama).fit()
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         total_returnrf   R-squared:                       0.534
Model:                            OLS   Adj. R-squared:                  0.532
Method:                 Least Squares   F-statistic:                     214.5
Date:                Fri, 23 Dec 2022   Prob (F-statistic):          1.61e-122
Time:                        16:52:13   Log-Likelihood:                 2702.1
No. Observations:                 753   AIC:                            -5394.
Df Residuals:                     748   BIC:                            -5371.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0053      0.000    -21.481      0.000      -0.006      -0.005
mktrf          0.0075      0.000     24.763      0.000       0.007       0.008
smb            0.0002      0.001      0.472      0.637      -0.001       0.001
hml           -0.0033      0.001     -6.512      0.000      -0.004      -0.002
mom            0.0020      0.000      4.819      0.000       0.001       0.003
==============================================================================
Omnibus:                       56.390   Durbin-Watson:                   1.640
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              100.853
Skew:                           0.507   Prob(JB):                     1.26e-22
Kurtosis:                       4.479   Cond. No.                         2.42
==============================================================================

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

In [14]:
```
alphas_listfama = []
beta_listfama = []
formula = 'total_returnrf ~ mktrf'
lr = ols(formula, final_fama).fit()
alphas_listfama.append(lr.params['Intercept'])
beta_listfama.append(lr.params['mktrf'])
```

```
resultfama = pd.DataFrame(index=['resultfama'],
                          data={
                            "alpha":alphas_listfama,
                            "beta":beta_listfama})
resultfama
```

Out[14]:

	alpha	beta
resultfama	-0.005126	0.007972

In [15]:

```
TIfama = final_fama['total_returnrf'].mean() / resultfama['beta']
JAfama = final_fama['total_returnrf'].mean() - resultfama['beta'] * final_fama['
print(TIfama)
print(JAfama)
```

```
resultfama   -0.590952
Name: beta, dtype: float64
resultfama   -0.005126
Name: beta, dtype: float64
```

Test

In [16]:

```
# breusch-pagan heteroskedasticity test
name = ['Lagrange multiplier statistic', 'p-value',
        'f-value', 'f p-value']
test = sms.het_breuschpagan(results.resid, results.model.exog)
lzip(name, test)
```

Out[16]:

```
[('Lagrange multiplier statistic', 5.165879301311405),
 ('p-value', 0.27069675883162475),
 ('f-value', 1.2917562900749995),
 ('f p-value', 0.27164885209296025)]
```

In [17]:

```
#White Standard Errors
formula = 'total_returnrf~ mktrf + smb + hml + mom'
WSE_results = smf.ols(formula, final_fama).fit(cov_type='HC1')
print(results.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:         total_returnrf   R-squared:                       0.534
Model:                            OLS   Adj. R-squared:                  0.532
Method:                 Least Squares   F-statistic:                     214.5
Date:                Fri, 23 Dec 2022   Prob (F-statistic):          1.61e-122
Time:                        16:52:13   Log-Likelihood:                 2702.1
No. Observations:                 753   AIC:                            -5394.
Df Residuals:                     748   BIC:                            -5371.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0053      0.000    -21.481      0.000      -0.006      -0.005
mktrf          0.0075      0.000     24.763      0.000       0.007       0.008
smb            0.0002      0.001      0.472      0.637      -0.001       0.001
```

```
hml           -0.0033      0.001     -6.512      0.000      -0.004      -0.002
mom            0.0020      0.000      4.819      0.000       0.001       0.003
==============================================================================
Omnibus:                       56.390   Durbin-Watson:                   1.640
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              100.853
Skew:                           0.507   Prob(JB):                     1.26e-22
Kurtosis:                       4.479   Cond. No.                         2.42
==============================================================================

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

In [18]:

```
#Newey West Standard Errors
formula = 'total_returnrf~ mktrf + smb + hml + mom'
NWSE_results = smf.ols(formula, final_fama).fit(cov_type='HAC',
                                    cov_kwds={'maxlags':6,'use_correction':True}
print(results.summary())
```

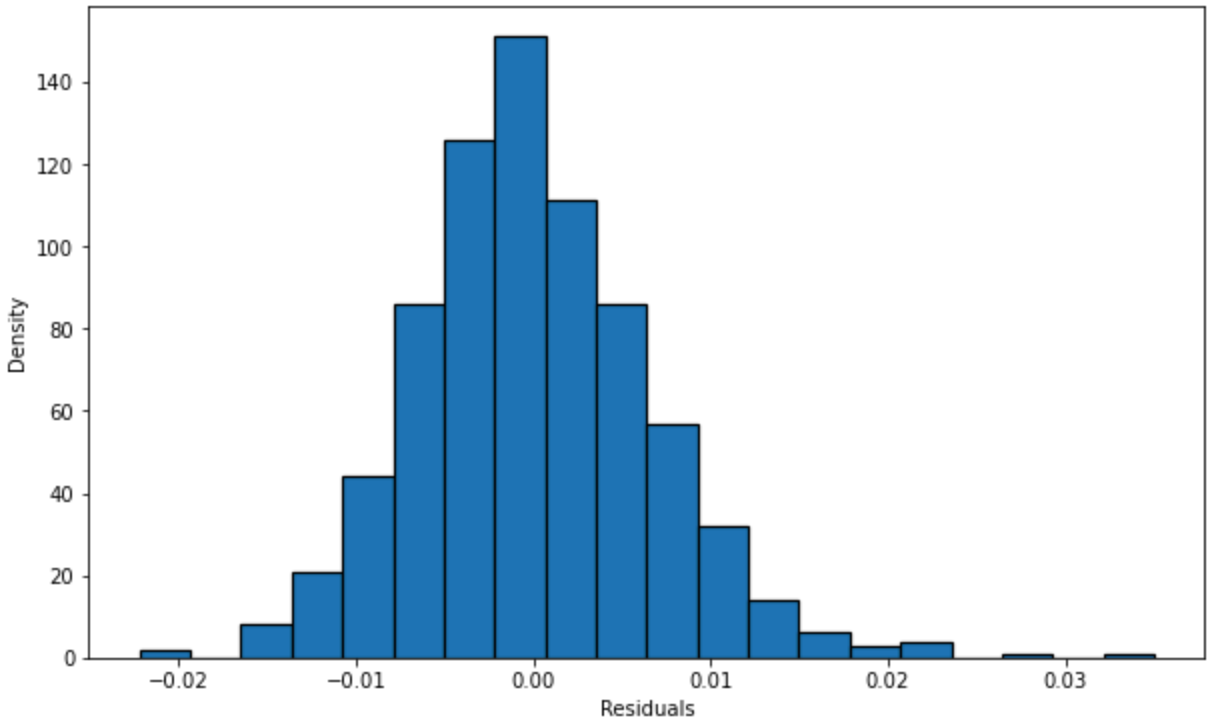
```
                            OLS Regression Results
==============================================================================
Dep. Variable:         total_returnrf   R-squared:                       0.534
Model:                            OLS   Adj. R-squared:                  0.532
Method:                 Least Squares   F-statistic:                     214.5
Date:                Fri, 23 Dec 2022   Prob (F-statistic):          1.61e-122
Time:                        16:52:13   Log-Likelihood:                 2702.1
No. Observations:                 753   AIC:                            -5394.
Df Residuals:                     748   BIC:                            -5371.
Df Model:                           4
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept     -0.0053      0.000    -21.481      0.000      -0.006      -0.005
mktrf          0.0075      0.000     24.763      0.000       0.007       0.008
smb            0.0002      0.001      0.472      0.637      -0.001       0.001
hml           -0.0033      0.001     -6.512      0.000      -0.004      -0.002
mom            0.0020      0.000      4.819      0.000       0.001       0.003
==============================================================================
Omnibus:                       56.390   Durbin-Watson:                   1.640
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              100.853
Skew:                           0.507   Prob(JB):                     1.26e-22
Kurtosis:                       4.479   Cond. No.                         2.42
==============================================================================

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

In [19]:

```
#Testing for Normality
residuals = results.resid

plt.figure(1)
plt.hist(residuals,20,edgecolor='black',linewidth=1.2)
plt.xlabel('Residuals')
plt.ylabel('Density')
plt.show()
```



In [20]:
```
#Testing for Normality
name = ['Jarque-Bera', 'Chi^2 two-tail prob.', 'Skew', 'Kurtosis']
test = sms.jarque_bera(residuals)
lzip(name, test)
```

Out[20]:
```
[('Jarque-Bera', 100.8526200607653),
 ('Chi^2 two-tail prob.', 1.2593075614350123e-22),
 ('Skew', 0.5070054528850473),
 ('Kurtosis', 4.478583112433732)]
```

In [21]:
```
#Testing for autocorrelation
sms.durbin_watson(residuals)
```

Out[21]:
```
1.6401397179974317
```

In [22]:
```
#Autocorrelation Test: Breusch Godfrey
name = ['Lagrange multiplier statistic', 'p-value',
'f-value', 'f p-value']
results1 = sms.acorr_breusch_godfrey(results, 10)
lzip(name, results1)
```

Out[22]:
```
[('Lagrange multiplier statistic', 96.91687594897245),
 ('p-value', 2.252222955239477e-16),
 ('f-value', 10.90176714327113),
 ('f p-value', 1.953717904688144e-17)]
```

In [23]:
```
#F-test
hypotheses = 'mktrf = smb = hml = mom = 0'
f_test = results.f_test(hypotheses)
print(f_test)
```

```
<F test: F=array([[214.46791597]]), p=1.6072991762046611e-122, df_denom=748, df_
num=4>
```

In [24]:
```
#Assumption 1
residuals.mean()
```

Out[24]:
```
-2.8336120523923942e-18
```

Analysis

In [25]:
```
# Analysis1: 2021
df_2021 = pd.read_csv('daily.csv')
df_2021['Date'] = pd.to_datetime(df_2021['Date'])
df_2021 = df_2021.set_index('Date')
df_2021 = df_2021.loc['2021-01-01':'2021-12-31']

stock_21 = yf.download(chosen_stock, start = '2021-01-01', end = '2021-12-31')['

for i in chosen_stock:
    stock_21['ret_{}'.format(i)] = np.log(stock_21['{}'.format(i)] / stock_21['{
    stock_21 = stock_21.drop(columns = ['{}'.format(i)])

return_21 = total_return(best_weight, stock_21)
vol_21 = total_vol(best_weight, stock_21)
sharpe_ratio_21 = (return_21 - df_2021['rf'].mean()) / vol_21
print(f'The portfolio return from Jan. 2021 to Dec. 2021 is {return_21:.2%}.')
print(f'The portfolio volatility from Jan. 2021 to Dec. 2021 is {vol_21:.2f}.')
print(f'The portfolio sharpe ratio from Jan. 2021 to Dec. 2021 is {sharpe_ratio_
```

```
[**********************100%**********************]  10 of 10 completed
The portfolio return from Jan. 2021 to Dec. 2021 is 24.55%.
The portfolio volatility from Jan. 2021 to Dec. 2021 is 0.19.
The portfolio sharpe ratio from Jan. 2021 to Dec. 2021 is 1.28.
```

In [26]:
```
# Analysis2: Compare 2021
df_2021 = pd.read_csv('daily.csv')
df_2021['Date'] = pd.to_datetime(df_2021['Date'])
df_2021 = df_2021.set_index('Date')
df_2021 = df_2021.loc['2021-01-01':'2021-12-31']

stock_2021 = yf.download(chosen_stock, start = '2021-01-01', end = '2021-12-31')

for i in chosen_stock:
    stock_2021['ret_{}'.format(i)] = np.log(stock_2021['{}'.format(i)] / stock_2
    stock_2021 = stock_2021.drop(columns = ['{}'.format(i)])

total_return_2021 = pd.DataFrame(np.sum(stock_2021*best_weight, axis = 1), colum
capm_2021 = df_2021.merge(total_return_2021, on = ['Date'], how = 'inner')
capm_2021['total_returnrf'] = capm_2021['total_return'] - np.log(capm_2021['rf']

alphas_list2021 = []
beta_list2021 = []
formula = 'total_returnrf ~ mktrf'
lr = ols(formula, capm_2021).fit()
alphas_list2021.append(lr.params['Intercept'])
beta_list2021.append(lr.params['mktrf'])
```

```
result2021 = pd.DataFrame(index=['result2021'],
                          data={
                            "alpha":alphas_list2021,
                            "beta":beta_list2021})
result2021
```

```
[**********100%**********]  10 of 10 completed
```

Out[26]:

	alpha	beta
result2021	0.000057	0.010258

In [27]:

```
TI2021 = capm_2021['total_returnrf'].mean() / result2021['beta']
JA2021 = capm_2021['total_returnrf'].mean() - result2021['beta'] * capm_2021['mk
print(TI2021)
print(JA2021)
```

```
result2021    0.094599
Name: beta, dtype: float64
result2021    0.000057
Name: beta, dtype: float64
```

In [28]:

```
# SP500
sp500 = pd.read_csv('HistoricalPrices.csv')
sp500 = sp500.set_index('Date')
sp500.index = pd.to_datetime(sp500.index)
sp500['return'] = np.log(sp500['close'] / sp500['close'].shift(1))*252
sp500.head()
```

Out[28]:

	close	return
Date		
2021-01-04	3700.65	NaN
2021-01-05	3726.86	1.778509
2021-01-06	3748.14	1.434803
2021-01-07	3803.79	3.714032
2021-01-08	3824.68	1.380170

In [29]:

```
sp500_vol = sp500['return'].std()
sp500_er = sp500['return'].mean()
sp500_sharpe = (sp500_er - np.log(df_2021['rf'].mean()+1)) / sp500_vol

print(f'The return of S&P500 from Jan. 2021 to Dec. 2021 is {sp500_er:.2%}.')
print(f'The volatility of S&P500 from Jan. 2021 to Dec. 2021 is {sp500_vol:.2f}.
print(f'The Sharpe ratio of S&P500 from Jan. 2021 to Dec. 2021 is {sp500_sharpe:
```

```
The return of S&P500 from Jan. 2021 to Dec. 2021 is 25.40%.
The volatility of S&P500 from Jan. 2021 to Dec. 2021 is 2.07.
The Sharpe ratio of S&P500 from Jan. 2021 to Dec. 2021 is 0.12.
```

In [30]:

```
capm_sp500 = df_2021.merge(sp500, on = ['Date'], how = 'inner')
capm_sp500['total_returnrf'] = capm_sp500['return'] - np.log(capm_2021['rf']+1)
```

```
alphas_listsp500 = []
beta_listsp500 = []
formula = 'total_returnrf ~ mktrf'
lr = ols(formula, capm_sp500).fit()
alphas_listsp500.append(lr.params['Intercept'])
beta_listsp500.append(lr.params['mktrf'])

resultsp500 = pd.DataFrame(index=['resultsp500'],
                      data={
                        "alpha":alphas_listsp500,
                        "beta":beta_listsp500})
resultsp500
```

Out[30]:

	alpha	beta
resultsp500	0.038618	2.305282

In [31]:

```
TIsp500 = capm_sp500['total_returnrf'].mean() / resultsp500['beta']
JAsp500 = capm_sp500['total_returnrf'].mean() - resultsp500['beta'] * capm_sp500
print(TIsp500)
print(JAsp500)
```

```
resultsp500    0.111792
Name: beta, dtype: float64
resultsp500    0.056091
Name: beta, dtype: float64
```

In [32]:

```
# Analysis3: Covid 19
stock_1719 = yf.download(chosen_stock, start = '2017-03-01', end = '2019-03-31')
stock_2022 = yf.download(chosen_stock, start = '2020-03-01', end = '2022-03-31')
```

```
[**********************100%**************************]  10 of 10 completed
[**********************100%**************************]  10 of 10 completed
```

In [33]:

```
for i in chosen_stock:
    stock_1719['ret_{}'.format(i)] = np.log(stock_1719['{}'.format(i)] / stock_1
    stock_1719 = stock_1719.drop(columns = ['{}'.format(i)])

for i in chosen_stock:
    stock_2022['ret_{}'.format(i)] = np.log(stock_2022['{}'.format(i)] / stock_2
    stock_2022 = stock_2022.drop(columns = ['{}'.format(i)])

return_1719 = total_return(best_weight, stock_1719)
return_2022 = total_return(best_weight, stock_2022)
vol_1719 = total_vol(best_weight, stock_1719)
vol_2022 = total_vol(best_weight, stock_2022)
sharpe_ratio_1719 = (return_1719 - rfrate) / vol_1719
sharpe_ratio_2022 = (return_2022 - rfrate) / vol_2022

print(f'The portfolio actual return from Mar. 2017 to Mar. 2019 is {return_1719:
print(f'The portfolio actual return from Mar. 2020 to Mar. 2022 is {return_2022:
print(f'The portfolio volatility from Mar. 2017 to Mar. 2019 is {vol_1719:.2f}.'
print(f'The portfolio volatility from Mar. 2020 to Mar. 2022 is {vol_2022:.2f}.'
print(f'The portfolio sharpe ratio from Mar. 2017 to Mar. 2019 is {sharpe_ratio_
print(f'The portfolio sharpe ratio from Mar. 2020 to Mar. 2022 is {sharpe_ratio_
```

```
The portfolio actual return from Mar. 2017 to Mar. 2019 is 33.89%.
The portfolio actual return from Mar. 2020 to Mar. 2022 is 20.72%.
The portfolio volatility from Mar. 2017 to Mar. 2019 is 0.15.
The portfolio volatility from Mar. 2020 to Mar. 2022 is 0.32.
The portfolio sharpe ratio from Mar. 2017 to Mar. 2019 is 2.21.
The portfolio sharpe ratio from Mar. 2020 to Mar. 2022 is 0.64.
```

In [34]:

```
df_1719 = pd.read_csv('daily.csv')
df_1719['Date'] = pd.to_datetime(df_1719['Date'])
df_1719 = df_1719.set_index('Date')
df_1719 = df_1719.loc['2017-03-01':'2019-3-31']

total_return_1719 = pd.DataFrame(np.sum(stock_1719 * best_weight, axis = 1), col
capm_1719 = df_1719.merge(total_return_1719, on = ['Date'], how = 'inner')
capm_1719['total_returnrf'] = capm_1719['total_return'] - np.log(capm_1719['rf']

alphas_list1719 = []
beta_list1719 = []
formula = 'total_returnrf ~ mktrf'
lr = ols(formula, capm_1719).fit()
alphas_list1719.append(lr.params['Intercept'])
beta_list1719.append(lr.params['mktrf'])

result1719 = pd.DataFrame(index=['result1719'],
                      data={
                        "alpha":alphas_list1719,
                        "beta":beta_list1719})
result1719
```

Out[34]:

	alpha	beta
result1719	-0.004923	0.008268

In [35]:

```
df_2022 = pd.read_csv('daily.csv')
df_2022['Date'] = pd.to_datetime(df_2022['Date'])
df_2022 = df_2022.set_index('Date')
df_2022 = df_2022.loc['2020-03-01':'2022-3-31']

total_return_2022 = pd.DataFrame(np.sum(stock_2022 * best_weight, axis = 1), col
capm_2022 = df_2022.merge(total_return_2022, on = ['Date'], how = 'inner')
capm_2022['total_returnrf'] = capm_2022['total_return'] - np.log(capm_2022['rf']

alphas_list2022 = []
beta_list2022 = []
formula = 'total_returnrf ~ mktrf'
lr = ols(formula, capm_2022).fit()
alphas_list2022.append(lr.params['Intercept'])
beta_list2022.append(lr.params['mktrf'])

result2022 = pd.DataFrame(index=['result2022'],
                      data={
                        "alpha":alphas_list2022,
                        "beta":beta_list2022})
result2022
```

Out[35]:

	alpha	beta
result2022	-0.000557	0.010574

In [36]:
```
TI1719 = capm_1719['total_returnrf'].mean() / result1719['beta']
JA1719 = capm_1719['total_returnrf'].mean() - result1719['beta'] * capm_1719['mk
print(TI1719)
print(JA1719)
```

```
result1719   -0.55465
Name: beta, dtype: float64
result1719   -0.004923
Name: beta, dtype: float64
```

In [37]:
```
TI2022 = capm_2022['total_returnrf'].mean() / result2022['beta']
JA2022 = capm_2022['total_returnrf'].mean() - result2022['beta'] * capm_2022['mk
print(TI2022)
print(JA2022)
```

```
result2022    0.049991
Name: beta, dtype: float64
result2022   -0.000557
Name: beta, dtype: float64
```