Project 1

March 4, 2023

1 Purpose

I have two goals for this project:

1. To help you master data manipulation and visualization
2. To help you understand the risk-return tradeoff for several measures of risk

2 Tasks

2.1 Packages and Settings

[1]:
```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
```

[2]:
```
%config InlineBackend.figure_format = 'retina'
%precision 4
pd.options.display.float_format = '{:.4f}'.format
```

2.2 Data

I used the following code cell to download the data for this project. Leave this code cell commented out and use the CSV files I provided with this notebook.

[3]:
```
import yfinance as yf
import pandas_datareader as pdr
import requests_cache
session = requests_cache.CachedSession(expire_after=1)
```

[4]:
```
# wiki = pd.read_html('https://en.wikipedia.org/wiki/Russell_1000_Index')
```

[5]:
```
# (
#     yf.Tickers(
#         tickers=wiki[2]['Ticker'].str.replace(pat='.', repl='-', regex=False).
 ↪to_list(),
#         session=session
#     )
#     .history(period='max', auto_adjust=False)
#     .assign(Date = lambda x: x.index.tz_localize(None))
#     .set_index('Date')
#     .rename_axis(columns=['Variable', 'Ticker'])
#     ['Adj Close']
#     .pct_change()
#     .loc['1962':'2022']
#     .to_csv('returns.csv')
# )
```

[6]:

```
# (
#     pdr.DataReader(
#         name='F-F_Research_Data_Factors_daily',
#         data_source='famafrench',
#         start='1900',
#         session=session
#     )
#     [0]
#     .rename_axis(columns='Variable')
#     .div(100)
#     .loc['1962':'2022']
#     .to_csv('ff.csv')
# )
```

Run the following code cell to read the data for this project. The `returns.csv` file contains daily returns for the Russell 1000 stocks from 1962 through 2022, and the `ff.csv` contains daily Fama and French factors from 1962 through 2022.

[7]:

```
returns = pd.read_csv('returns.csv', index_col='Date', parse_dates=True).
 ↪mul(100)
ff = pd.read_csv('ff.csv', index_col='Date', parse_dates=True)
```

2.3 Single Stocks

For this section, use the single stock returns in `returns`. You may select years t and $t+1$, but only use stocks with complete returns data for years t and $t+1$.

2.3.1 Task 1: Do mean returns in year t predict mean returns in year $t+1$?

[8]:

```
returnst = returns.loc['2000':].dropna(axis=1)
```

[9]:

```
def Plot(a,b):
    plt.scatter(a,b)
    corr_returns=np.corrcoef(a, b)
    sns.regplot(x=a, y=b)
    c = corr_returns**2
    print(f'The correlation coefficient is: {corr_returns[0,1].round(4)}')
    print(f'The R Squared is: {c[0,1].round(4)}')
```

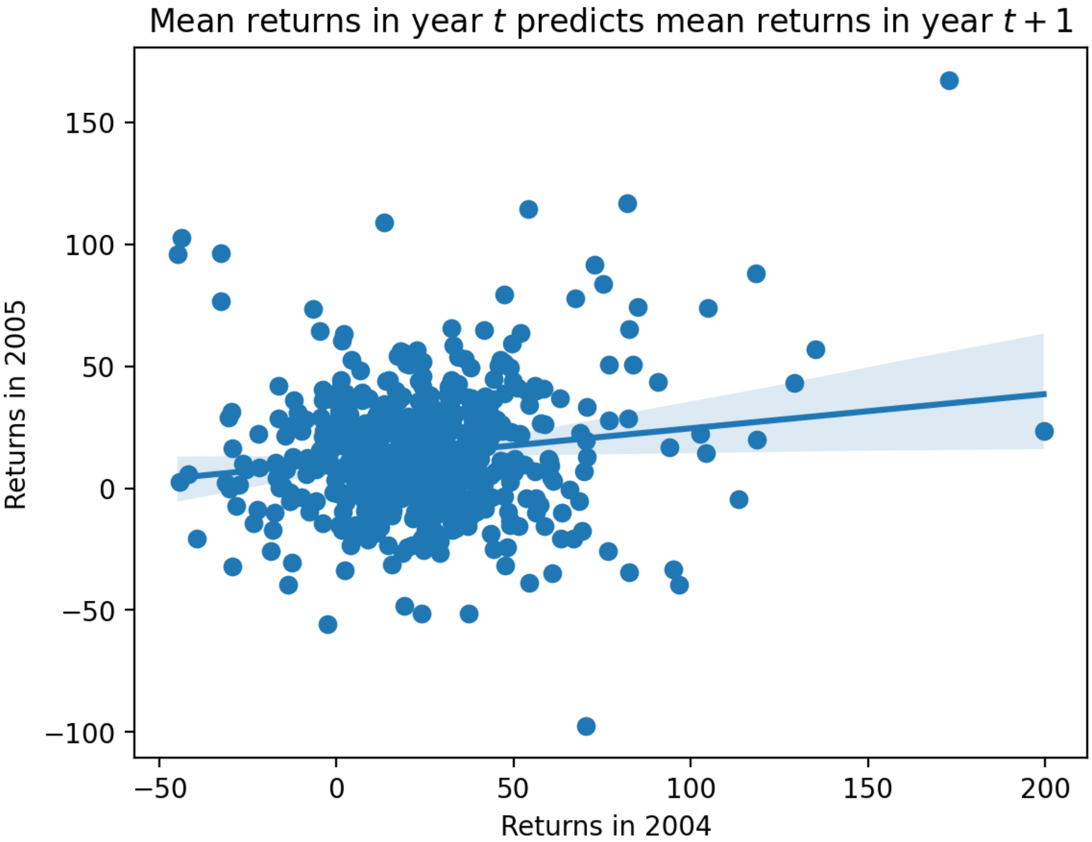
[10]:

```
returns_2004 = returnst.loc['2004'].mean().mul(252)
returns_2005 = returnst.loc['2005'].mean().mul(252)
Plot (returns_2004,returns_2005)
plt.title ('Mean returns in year $t$ predicts mean returns in year $t+1$')
plt.xlabel ('Returns in 2004 ')
plt.ylabel ('Returns in 2005')
```

```
The correlation coefficient is: 0.1517
The R Squared is: 0.023
```

[10]:

```
Text(0, 0.5, 'Returns in 2005')
```



We have compared the mean returns for 2004 with the mean returns from 2005. From the above code we have calculated the correlation coeffecient which is 0.1517. This implies that the relationship between the two mean returns over our chosen time period is positive. When it comes to analysing whether the mean returns in 2004 can predict the mean returns in 2005, we have calucalted the R^2 which is 0.023. This implies that mean returns in 2004 does have minimal strength in predicting the mean returns in 2005.The scatter plot clearly highlights the relationship between the mean returns over the two periods. We need to conduct further statistical analysis to determine statistical significance of this relationship.

2.3.2 Task 2: Does volatility in year t predict volatility in year $t + 1$?

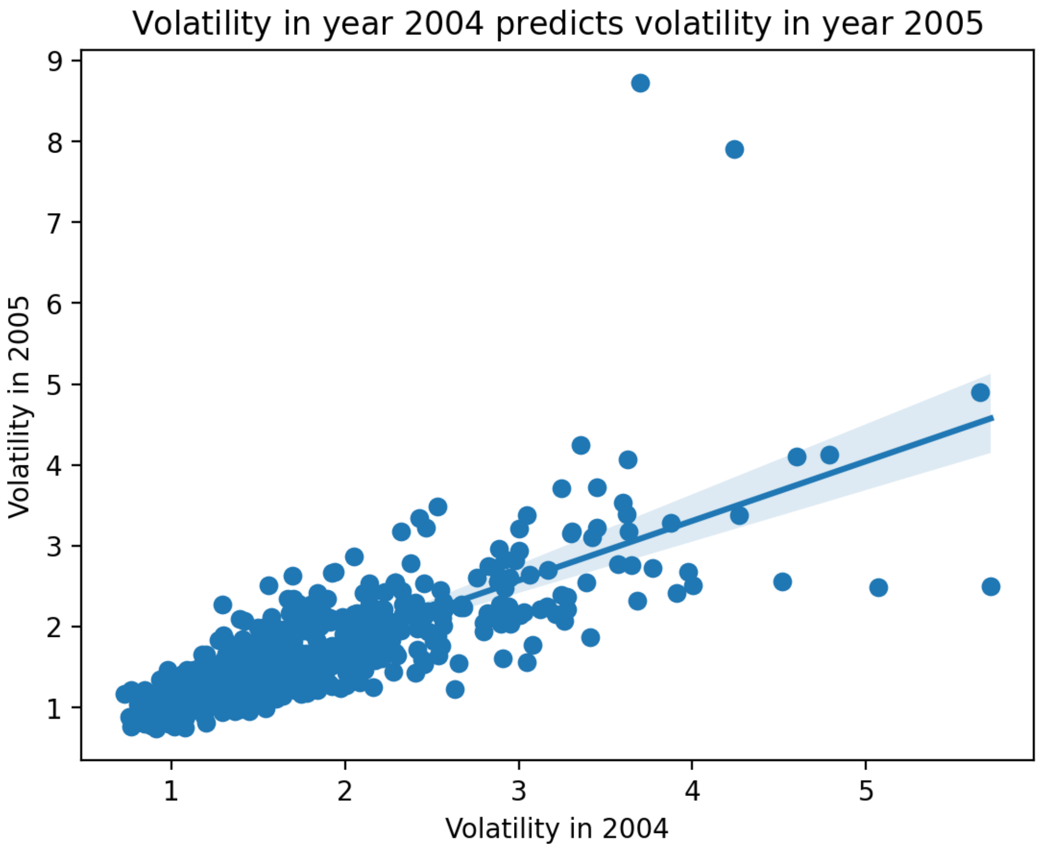
[11]:

```
vol_2004 = returnst.loc['2004'].std()
vol_2005 = returnst.loc['2005'].std()
Plot (vol_2004,vol_2005)
plt.title ('Volatility in year $2004$ predicts volatility in year $2005$')
plt.xlabel ('Volatility in 2004')
plt.ylabel ('Volatility in 2005')
```

```
The correlation coefficient is: 0.7711
The R Squared is: 0.5946
```

[11]:

```
Text(0, 0.5, 'Volatility in 2005')
```



We have compared the volatility(Std) for 2004 with the volatility for 2005. From the above code we have calculated the correlation coeffecient which is 0.7711. This implies that the relationship between the two volatility's over our chosen time period is strongly positive. When it comes to analysing whether volatility in 2004 can predict the volatility in 2005, we have calucalted the R^2 which is 0.5964. This implies that volatility in 2004 does have strength in predicting the volatility in 2005. The scatter plot clearly highlights the relationship between the volatility's over the two periods which is linear.

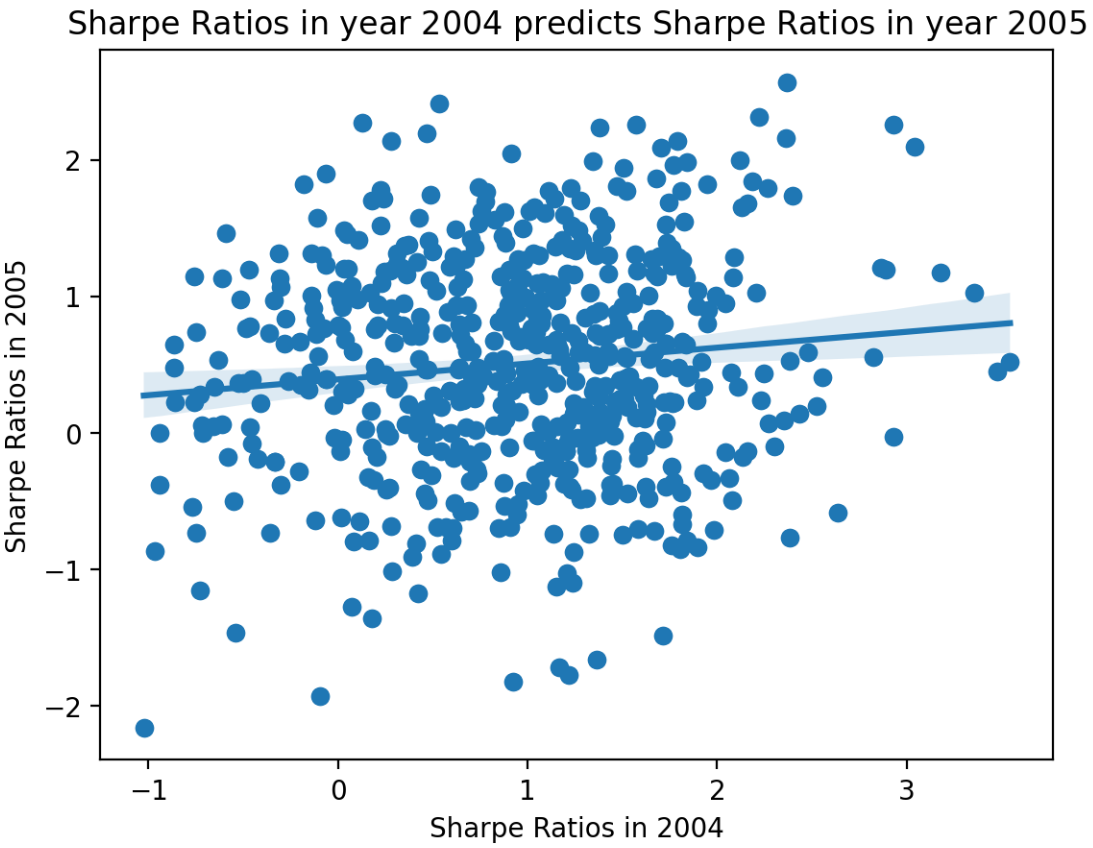
2.3.3 Task 3: Do Sharpe Ratios in year t predict Sharpe Ratios in year $t+1$?

[12]:
```
def sharpe(ri, rf, ann_fac=np.sqrt(252)):
    ri_rf = ri.sub(rf.loc[ri.index], axis=0).dropna()
    return ann_fac * ri_rf.mean() / ri_rf.std()

sharpe_2004 = sharpe(returnst.loc['2004'], ff['RF'])
sharpe_2005 = sharpe(returnst.loc['2005'], ff['RF'])
Plot (sharpe_2004,sharpe_2005)
plt.title ('Sharpe Ratios in year $2004$ predicts Sharpe Ratios in year $2005$')
plt.xlabel ('Sharpe Ratios in 2004')
plt.ylabel ('Sharpe Ratios in 2005')
```

```
The correlation coefficient is: 0.1179
The R Squared is: 0.0139
```

[12]:
```
Text(0, 0.5, 'Sharpe Ratios in 2005')
```



We have compared the sharpe ratio for 2004 with the sharpe ratio for 2005. From the above code we have calculated the correlation coeffecient which is 0.1179. This implies that the relationship between the two sharpe ratios over our chosen time period is positive. When it comes to analysing whether sharpe ratio in 2004 can predict the sharpe ratio in 2005, we have calucalted the R^2 which is 0.0139. This implies that sharpe ratio in 2004 does have strength in predicting the sharpe ratio in 2005. The scatter plot clearly highlights the relationship between the sharpe ratios over the two periods is linear.

2.3.4 Task 4: Do CAPM betas in year t predict CAPM betas in year $t + 1$?

[13]:

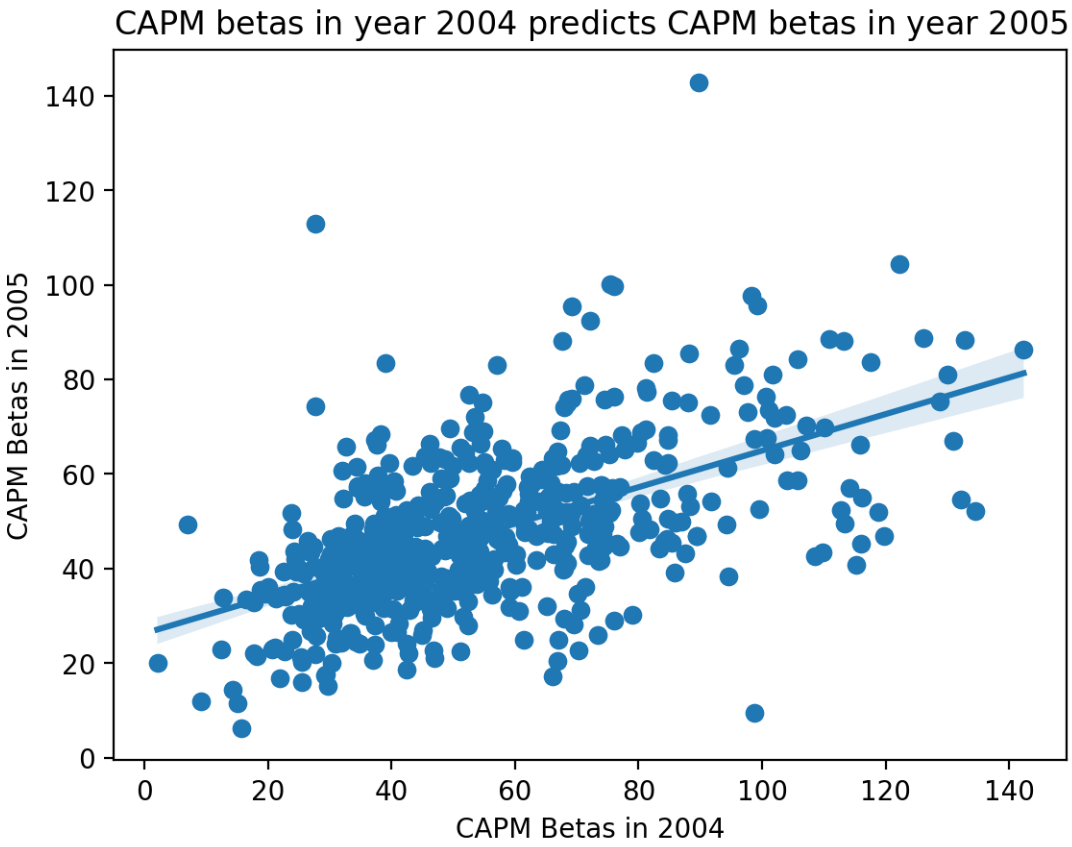
```
def beta(ri, rf = ff['RF'], rm_rf = ff['Mkt-RF']):
    ri_rf = ri.sub(rf)
    rm_rf = rm_rf.loc[ri_rf.index]
    return ri_rf.cov(rm_rf) / rm_rf.var()

beta_2004 = returnst.loc['2004'].apply(beta)
beta_2005 = returnst.loc['2005'].apply(beta)
Plot (beta_2004,beta_2005)
plt.title ('CAPM betas in year $2004$ predicts CAPM betas in year $2005$')
plt.xlabel ('CAPM Betas in 2004')
plt.ylabel ('CAPM Betas in 2005')
```

```
The correlation coefficient is: 0.5669
The R Squared is: 0.3214
```

[13]:

```
Text(0, 0.5, 'CAPM Betas in 2005')
```



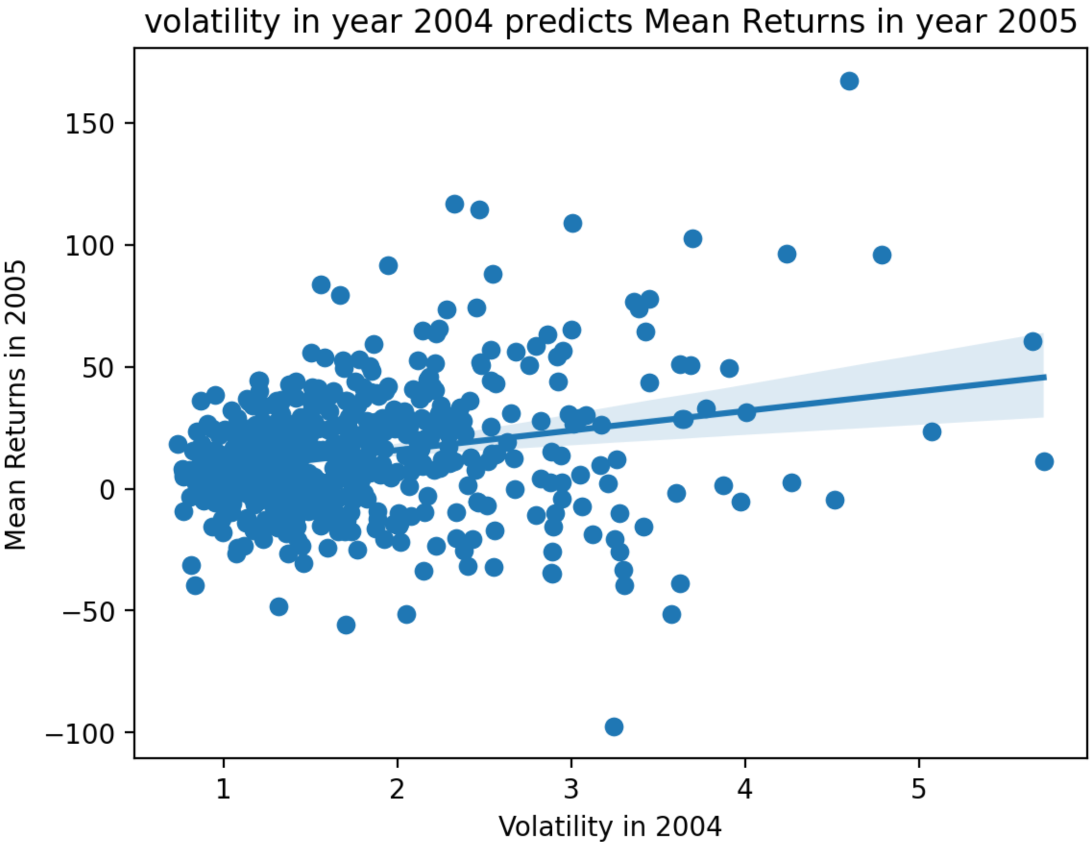
We have compared the beta for 2004 with the beta for 2005. From the above code we have calculated the correlation coeffecient which is 0.5669. This implies that the relationship between the two betas over our chosen time period is positive. When it comes to analysing whether beta in 2004 can predict the beta in 2005, we have calucalted the R^2 which is 0.3214. This implies that beta in 2004 does have strength in predicting the beta in 2005. The scatter plot clearly highlights the relationship between the betas over the two periods is linear.

2.3.5 Task 5: Does volatility in year t predict *mean returns* in year $t + 1$?

[14]:
```
Plot (vol_2004, returns_2005)
plt.title ('volatility in year $2004$ predicts Mean Returns in year $2005$')
plt.xlabel ('Volatility in 2004')
plt.ylabel ('Mean Returns in 2005')
```

```
The correlation coefficient is: 0.2438
The R Squared is: 0.0594
```

```
[14]: Text(0, 0.5, 'Mean Returns in 2005')
```



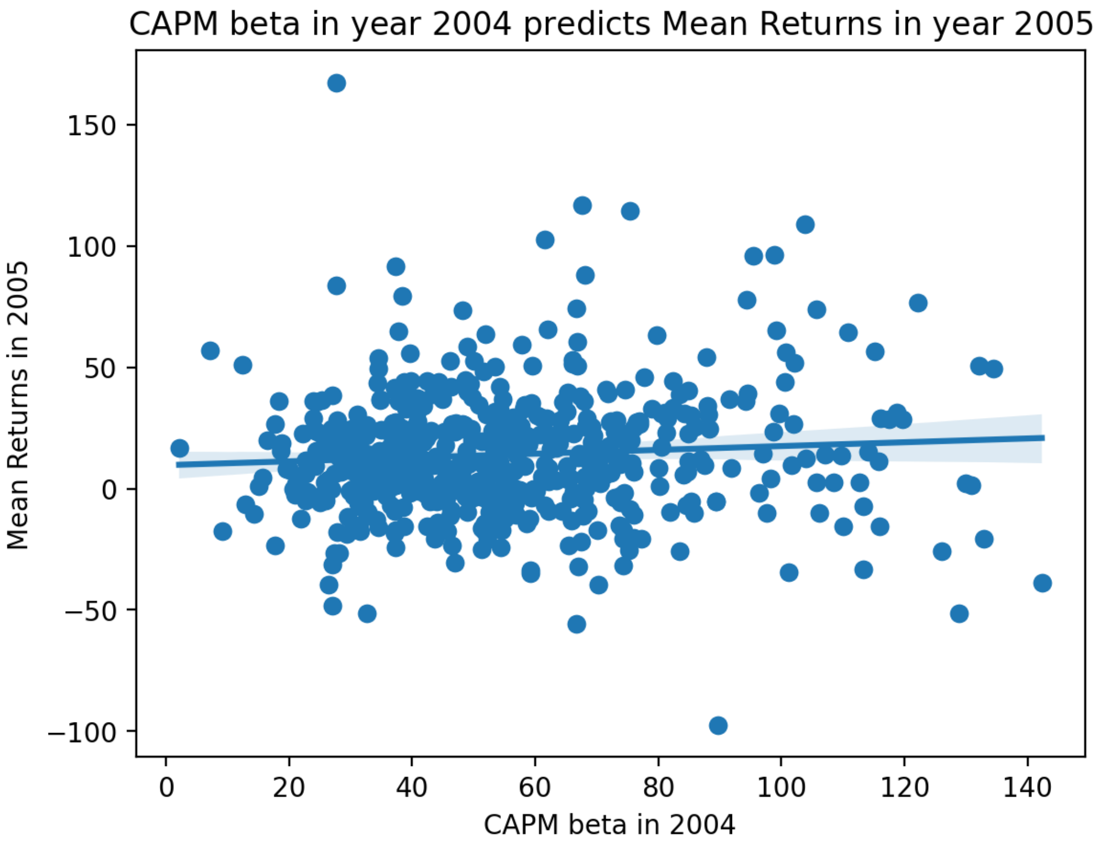
We have compared the volatility for 2004 with the mean return for 2005. From the above code we have calculated the correlation coeffecient which is 0.2438. This implies that the relationship between the volatility and mean return over our chosen time period is positive. When it comes to analysing whether volatility in 2004 can predict the mean returns in 2005, we have calucalted the R^2 which is 0.0594. This implies that volatility in 2004 does have strength in predicting the mean returns in 2005. The scatter plot clearly highlights the relationship is linear.

2.3.6 Task 6: Does CAPM beta in year t predict *mean returns* in year $t + 1$?

[15]:
```
Plot (beta_2004, returns_2005)
plt.title ('CAPM beta in year $2004$ predicts Mean Returns in year $2005$')
plt.xlabel ('CAPM beta in 2004')
plt.ylabel ('Mean Returns in 2005')
```

```
The correlation coefficient is: 0.0763
The R Squared is: 0.0058
```

[15]:
```
Text(0, 0.5, 'Mean Returns in 2005')
```



We have compared the beta for 2004 with the mean return for 2005. From the above code we have calculated the correlation coeffecient which is 0.0763. This implies that the relationship between the beta and mean return over our chosen time period is positive. When it comes to analysing whether beta in 2004 can predict the mean returns in 2005, we have calucalted the R^2 which is 0.0058. This implies that beta in 2004 does have very minimal strength in predicting the mean returns in 2005. The scatter plot clearly highlights the relationship is horizontal.

2.4 Portfolios I

For this section, create 100 random portfolios of 50 stocks each from the daily returns in `returns`. Equally weight these portfolios and rebalance them daily. Use the same stocks and years t and $t+1$ as the previous section.

2.4.1 Task 7: Does volatility in year t predict *mean returns* in year $t+1$?

[16]:

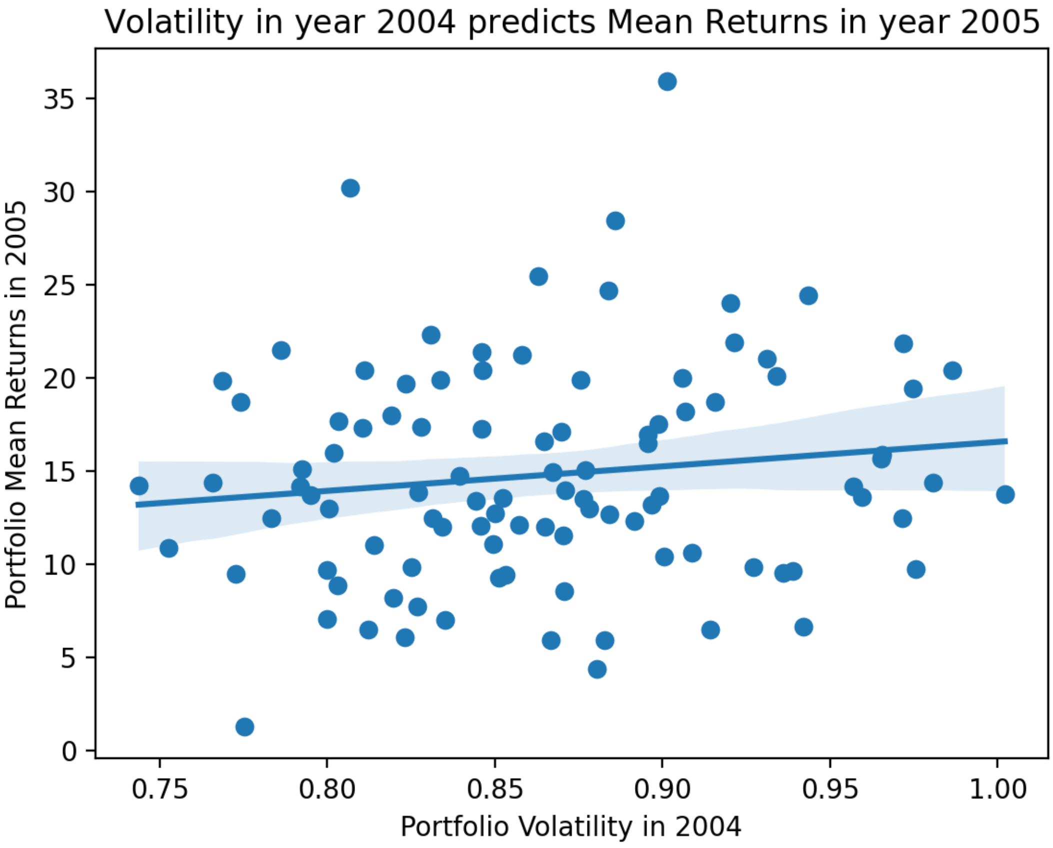
```
portfolios = []
for i in range(100):
    portfolio = np.random.choice(returns.columns, size=50, replace=False)
    portfolios.append(portfolio)
```

```
portfolio_returns = []
for portfolio in portfolios:
    portfolio_return = returns[portfolio].mean(axis=1)
    portfolio_returns.append(portfolio_return)
portfolio_returns
portfolio_returns = pd.DataFrame(portfolio_returns).T

portfolio_vol_t=portfolio_returns.loc['2004'].std()
portfolio_mean_t_plus_1=portfolio_returns.loc['2005'].mean().mul(252)
Plot (portfolio_vol_t, portfolio_mean_t_plus_1)
plt.title ('Volatility in year $2004$ predicts Mean Returns in year $2005$')
plt.xlabel ('Portfolio Volatility in 2004')
plt.ylabel ('Portfolio Mean Returns in 2005')
```

```
The correlation coefficient is: 0.1344
The R Squared is: 0.0181
```

```
[16]: Text(0, 0.5, 'Portfolio Mean Returns in 2005')
```



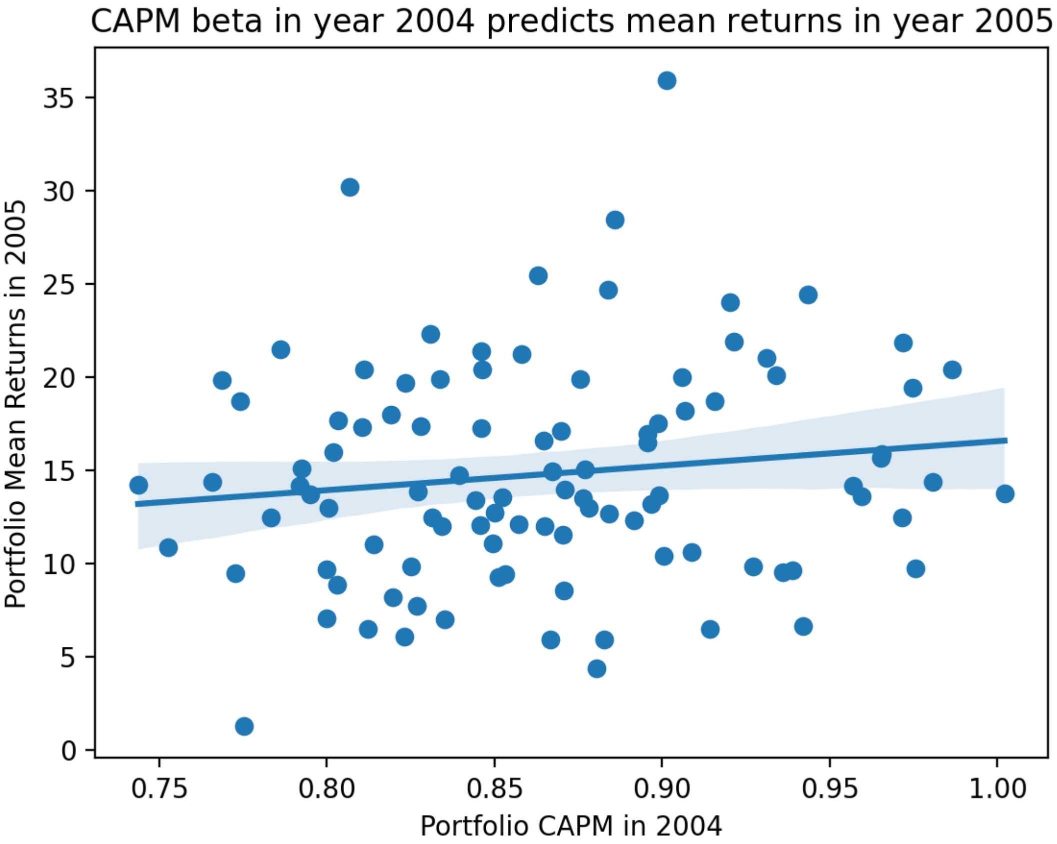
We have compared the volatility for 100 portfolios with 50 stocks which were randomly assigned for 2004 with the mean return for the same in 2005. From the above code we have calculated the correlation coeffecient which is 0.1344. This implies that the relationship between the volatility and mean return over our chosen time period for the portfolios is negative. When it comes to analysing whether volatility in 2004 can predict the mean returns in 2005, we have calucalted the R^2 which is 0.0181. This implies that the volatility in 2004 does have very minimal strength in predicting the mean returns in 2005. The scatter plot clearly highlights the relationship is random.

2.4.2 Task 8: Does CAPM beta in year t predict *mean returns* in year $t + 1$?

```
[17]: portfolio_beta_t= portfolio_returns.loc['2004'].apply(beta)
      portfolio_mean_t_plus_1
      Plot (portfolio_vol_t, portfolio_mean_t_plus_1)
      plt.title ('CAPM beta in year $2004$ predicts mean returns in year $2005$')
      plt.xlabel ('Portfolio CAPM in 2004')
      plt.ylabel ('Portfolio Mean Returns in 2005')
```

```
The correlation coefficient is: 0.1344
The R Squared is: 0.0181
```

```
[17]: Text(0, 0.5, 'Portfolio Mean Returns in 2005')
```



We have compared the beta for 100 portfolios with 50 stocks which were randomly assigned for 2004 with the mean return for the same in 2005. From the above code we have calculated the correlation coeffecient which is 0.1344. This implies that the relationship between the beta and mean return over our chosen time period is barely positive. When it comes to analysing whether beta in 2004 can predict the mean returns in 2005, we have calucalted the R^2 which is close to 0.0181. This implies that CAPM beta in 2004 does has no strength in predicting the mean returns in 2005. The scatter plot clearly highlights the relationship is random.

2.5 Portfolios II

Calculate monthly volatility and total return for *every stock* and *every month* in `returns`. Drop stock-months with fewer than 15 returns. Each month, assign these stocks to one of five portfolios based on their volatility during the previous month. Equally weight these portfolios and rebalance them monthly.

```
[18]: #Calculating and creating a Dataframe of Total Returns
portfolio_r = (
    np.log(1+(returns/100))
    .resample("M")
    .agg(sum)
    .pipe(np.exp)
    .sub(1)
    .mul(100)
    .replace(0,np.nan)
)
portfolio_r.columns = pd.MultiIndex.from_tuples([('Total Returns', col) for col␣
  ↪in returns.columns])
```

```
[19]: #Calculating and creating a Dataframe of Volatility
portfolio_v = (
    returns
    .resample("M")
    .agg("std")
)
portfolio_v.columns = pd.MultiIndex.from_tuples([('Volatility', col) for col in␣
  ↪returns.columns])
```

```
[20]: #Calculating and creating a Dataframe with every assigned asset to a portfolio␣
  ↪based on volatilty
portfolio_p = portfolio_v.apply(pd.qcut, q=5, labels=False, axis=1).add(1)
portfolio_p.columns = pd.MultiIndex.from_tuples([('Portfolio', col) for col in␣
  ↪returns.columns])
```

```
[21]: portfolio_2 = pd.concat([portfolio_r, portfolio_v,portfolio_p], axis=1)
portfolio_2.head()
```

```
[21]:           Total Returns                                                 …  \
                           A       AA AAL AAP AAPL ABBV ABC ABNB ABT ACGL  …
      Date                                                                 …
      1962-01-31         NaN  -8.6042 NaN NaN  NaN  NaN NaN  NaN NaN  NaN  …
      1962-02-28         NaN   1.7683 NaN NaN  NaN  NaN NaN  NaN NaN  NaN  …
      1962-03-31         NaN   5.5786 NaN NaN  NaN  NaN NaN  NaN NaN  NaN  …
      1962-04-30         NaN  -3.9139 NaN NaN  NaN  NaN NaN  NaN NaN  NaN  …
      1962-05-31         NaN -10.1133 NaN NaN  NaN  NaN NaN  NaN NaN  NaN  …

                 Portfolio
                       YUM   Z ZBH ZBRA  ZG  ZI ZION  ZM  ZS ZTS
      Date
      1962-01-31       NaN NaN NaN  NaN NaN NaN  NaN NaN NaN NaN
      1962-02-28       NaN NaN NaN  NaN NaN NaN  NaN NaN NaN NaN
      1962-03-31       NaN NaN NaN  NaN NaN NaN  NaN NaN NaN NaN
      1962-04-30       NaN NaN NaN  NaN NaN NaN  NaN NaN NaN NaN
      1962-05-31       NaN NaN NaN  NaN NaN NaN  NaN NaN NaN NaN

      [5 rows x 3030 columns]
```

2.5.1 Task 9: Do high volatility portfolios have high mean returns and Sharpe Ratios?

```
[22]: #Calculating average Monthly returns for each Portfolio and then annualizing it
      portfolio_mean = (portfolio_2
          .stack()
          .groupby('Portfolio')
          ["Total Returns"]
          .mean().mul(12)
      )
      portfolio_mean
```

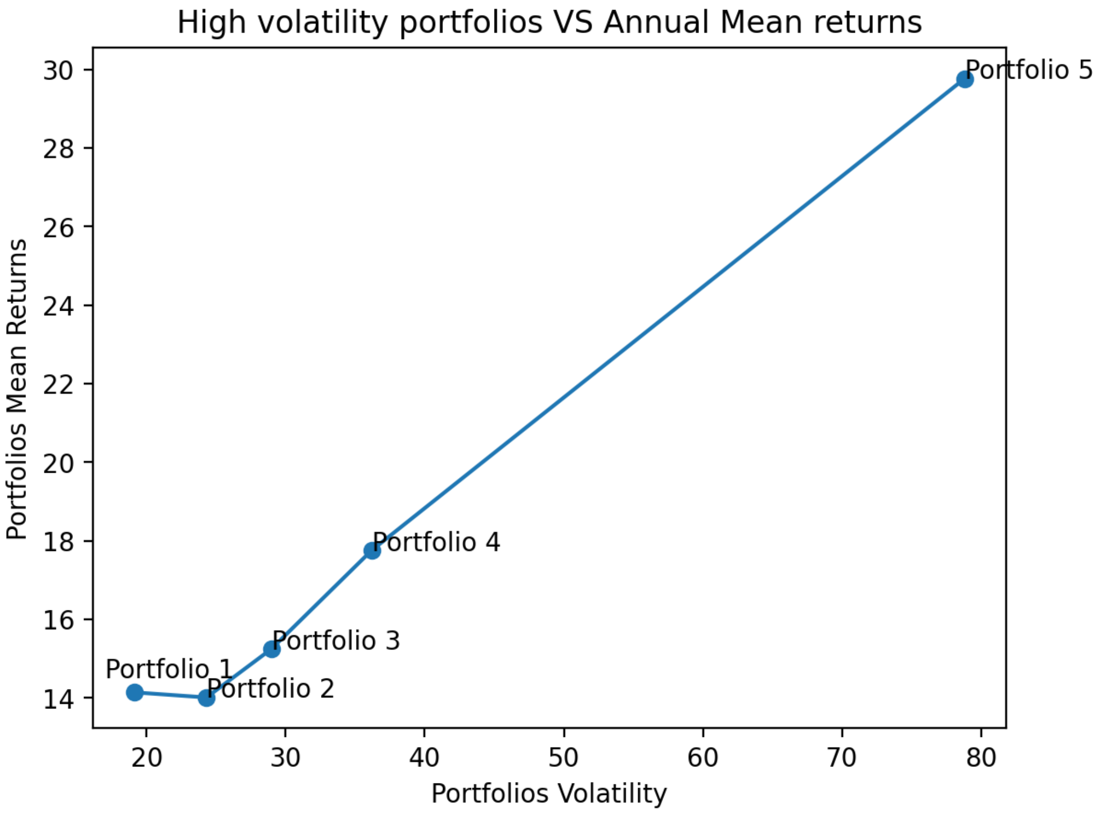
```
[22]: Portfolio
      1.0000   14.1414
      2.0000   14.0117
      3.0000   15.2448
      4.0000   17.7539
      5.0000   29.7604
      Name: Total Returns, dtype: float64
```

```
[23]: #Calculating average Monthly returns for each Portfolio and then annualizing it
      portfolio_vol = (portfolio_2
          .stack()
          .groupby('Portfolio')
          ["Total Returns"]
          .std()
          .mul(np.sqrt(12))
      )
```

```
portfolio_vol
```

```
[23]: Portfolio
      1.0000    19.1118
      2.0000    24.3038
      3.0000    28.9647
      4.0000    36.1744
      5.0000    78.7991
      Name: Total Returns, dtype: float64
```

```
[24]: plt.plot(portfolio_vol, portfolio_mean)
      plt.scatter(portfolio_vol, portfolio_mean)
      plt.title ('High volatility portfolios VS Annual Mean returns')
      plt.xlabel ('Portfolios Volatility')
      plt.ylabel ('Portfolios Mean Returns')
      plt.text( 17, 14.5,s = f"Portfolio {1}")
      for i in range (2,6):
          plt.text( portfolio_vol[i], portfolio_mean[i],s = f"Portfolio {i}")
```



The data reinforces the observation that higher volatility is generally associated with higher returns, as seen in the graph. The fact that this observation is based on a large sample of data adds further support to the idea that the relationship between volatility and returns is statistically significant. However, it is important to keep in mind that correlation does not necessarily imply causation. Just because we observe a positive correlation between portfolio volatility and returns, it does not necessarily mean that higher volatility is the direct cause of higher returns. There may be other factors at play that contribute to both higher volatility and higher returns, or there may be other factors that are driving the returns that are not related to volatility.

[25]:
```
# Calculating monthly RF
rf = (ff["RF"]
     .add(1)
     .resample("M")
     .agg("prod")
     .sub(1)
     .mul(100)
      )
```

[26]:
```
#Creating a Data frame with excess returns
portfolio_excess = portfolio_r.sub(rf , axis=0).rename(columns={'Total Returns':
  ↪ 'Excess Returns'})
portfolio_ri_rf = pd.concat([portfolio_excess, portfolio_v,portfolio_p], axis=1)
portfolio_ri_rf.head()
```

[26]:
```
           Excess Returns                                                ...  \
                        A       AA AAL AAP AAPL ABBV ABC ABNB ABT ACGL   ...
Date                                                                     ...
1962-01-31            NaN  -8.8465 NaN NaN  NaN  NaN NaN  NaN NaN  NaN   ...
1962-02-28            NaN   1.5591 NaN NaN  NaN  NaN NaN  NaN NaN  NaN   ...
1962-03-31            NaN   5.3804 NaN NaN  NaN  NaN NaN  NaN NaN  NaN   ...
1962-04-30            NaN  -4.1342 NaN NaN  NaN  NaN NaN  NaN NaN  NaN   ...
1962-05-31            NaN -10.3556 NaN NaN  NaN  NaN NaN  NaN NaN  NaN   ...

           Portfolio
                 YUM   Z ZBH ZBRA  ZG  ZI ZION  ZM  ZS ZTS
Date
1962-01-31       NaN NaN NaN  NaN NaN NaN  NaN NaN NaN NaN
1962-02-28       NaN NaN NaN  NaN NaN NaN  NaN NaN NaN NaN
1962-03-31       NaN NaN NaN  NaN NaN NaN  NaN NaN NaN NaN
1962-04-30       NaN NaN NaN  NaN NaN NaN  NaN NaN NaN NaN
1962-05-31       NaN NaN NaN  NaN NaN NaN  NaN NaN NaN NaN

[5 rows x 3030 columns]
```

[27]:
```
# Calculating the mean of excess returns
sharpe_r = (portfolio_ri_rf
        .stack()
        .groupby (["Portfolio"])["Excess Returns"]
         .mean()
        .mul (12)
)
```

[28]:

```
# Calculating the Volatility of excess returns
sharpe_v = (portfolio_ri_rf
       .stack()
       .groupby (["Portfolio"])["Excess Returns"]
        .std()
        .mul(np.sqrt(12))
)
print(sharpe_v)
```

```
Portfolio
1.0000   19.1861
2.0000   24.3402
3.0000   28.9825
4.0000   36.1822
5.0000   78.8025
Name: Excess Returns, dtype: float64
```

[29]:

```
#Calculating Ratios
Sharpe = sharpe_r.div(sharpe_v).mul(np.sqrt(12))
Sharpe.to_frame()
```

[29]:

```
           Excess Returns
Portfolio
1.0000             2.0296
2.0000             1.5776
3.0000             1.4726
4.0000             1.4194
5.0000             1.1794
```

[30]:

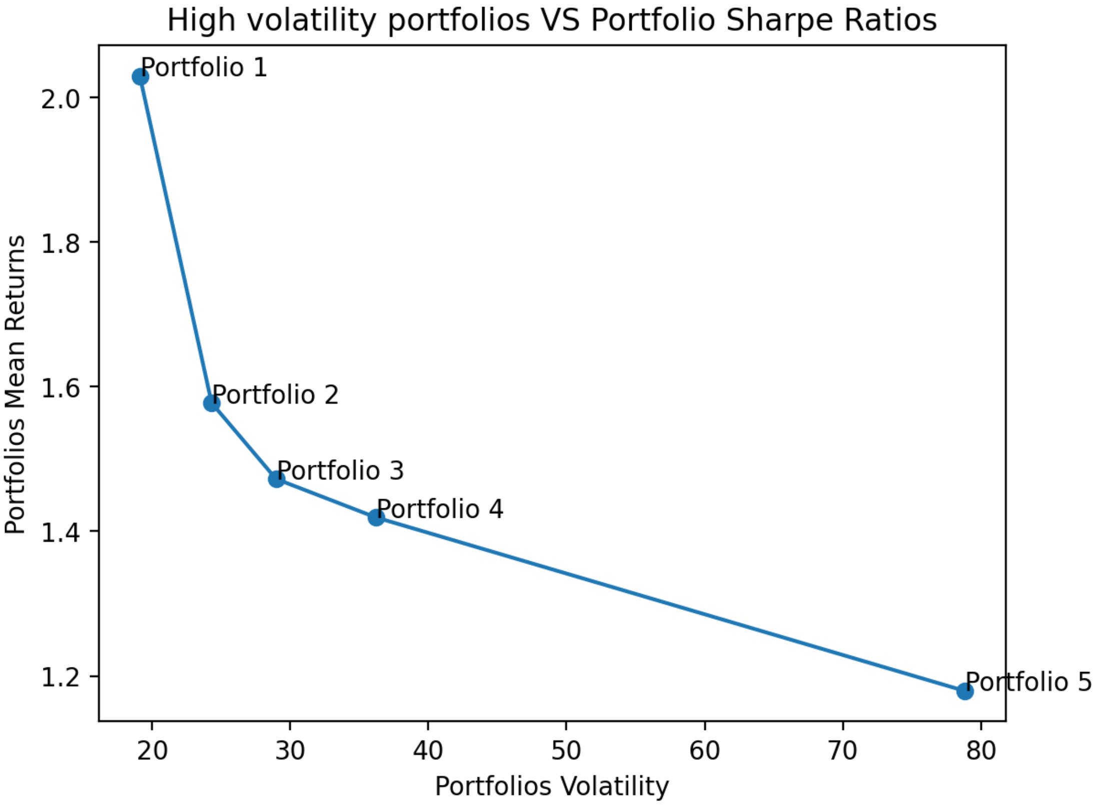
```
Sharpe.index[0]
```

[30]:

```
1.0
```

[31]:

```
plt.plot(portfolio_vol, Sharpe)
plt.scatter(portfolio_vol, Sharpe)
plt.title ('High volatility portfolios VS Portfolio Sharpe Ratios')
plt.xlabel ('Portfolios Volatility')
plt.ylabel ('Portfolios Mean Returns')
for i in range (1,6):
    plt.text( portfolio_vol[i], Sharpe[i],s = f"Portfolio {i}")
```



The data highlights the concept that as the volatility of a portfolio increases, the additional return generated per unit of risk taken decreases. This means that at some point, taking on additional risk by increasing the volatility of the portfolio may not be worth the potential return.

The observation that portfolios with higher volatility have lower Sharpe ratios supports this idea. The Sharpe ratio is a measure of risk-adjusted return, which considers both the total return of a portfolio and the volatility of the returns. Therefore, a lower Sharpe ratio suggests that the portfolio is generating less return per unit of risk taken.

The plot above clearly demonstrates that portfolios with higher volatility have lower Sharpe ratios, indicating a lower return per unit of risk taken. In contrast, portfolios with less volatility have higher Sharpe ratios, indicating a higher return per unit of risk taken. This observation supports the idea of diminishing marginal returns, which implies that increasing the risk level of the portfolio beyond a certain point may not result in a proportional increase in returns.

2.6 Discussion

2.6.1 Task 10: Discuss and explain any limitations of your analysis above

1. Our project is limited to only two successive years, which are 2004 and 2005. This approach is biased because the performance of an economy can be influenced by several factors beyond the two-year period, and a recession or recovery may continue to affect the economy for a longer duration. Therefore, our analysis based on only two years can lead to biased conclusions.

2. Mean returns can be a good predictor for the future, but it is important to note that several other factors can affect the returns or prices of assets, making it noisy and unreliable. For example, geopolitical events, regulatory changes, and market sentiment can impact the performance of assets, making it difficult to predict their future performance based solely on historical data.

3. The use of R-squared as a measure of variance can be biased in cases where there are outliers or extreme observations in the data set. R-squared only measures the proportion of the variance that is explained by the model and may not capture the entire picture of the relationship between the variables.

4. Another limitation of our project is the small sample size. Except for Question 9, we have a limited amount of data to work with, which can lead to flawed interpretations. A larger sample size can provide a more accurate representation of the population, leading to more reliable insights.

5. In contrast to the small sample size in most questions, Question 9 has a large amount of data, which can also have limitations. The data is old, and the market conditions have significantly changed since then. Therefore, the relevance of the data to current market conditions may be limited.

6. Finally, it is important to note that statistical significance does not always imply economic significance. Our reliance on statistical inferences may overlook the practical implications of the results, which may be too small to be of any real importance in economic decision-making. Therefore, it is essential to consider the economic implications of our findings alongside statistical measures of significance.

3 Criteria

1. All tasks are worth ten points
2. Discuss and explain your findings for all ten tasks
3. Here are a few more tips
    1. ***Your goal is to convince me of your calculations and conclusions***
    2. I typically find figures most convincing
    3. If you use correlations, consider how a handful of outliers may affect your findings
    4. Remove unnecessary code, outputs, and print statements
    5. Write functions for calculations that you expect to use more than once
    6. ***I will not penalize code style, but I will penalize submissions that are difficult to follow or do not follow these instructions***
4. How to submit your project
    1. Restart your kernel, run all cells, and save your notebook
    2. Export your notebook to PDF (`File > Save And Export Notebook As ... > PDF` in JupyterLab)
        1. If this export does not work, you can either (1) Install MikTeX on your laptop with the default settings or (2) use DataCamp Workspace to export your notebook to PDF
        2. You do not need to re-run your notebook to export it because notebooks store output cells
    3. Upload your notebook and PDF to Canvas

4. Upload your PDF only to Gradescope and tag your teammates
5. Gradescope helps me give better feedback more quickly, but I do not consider it reliable for sharing and storing your submission files