Investment portfolio Optimization model using Monte Carlo Simulation and Optimization algorithm in Python

(BANA 7030)

Author:

Anupam Shukla, M13469377

April 2020

Table of Contents

Executive Summary

Purpose -

The purpose of this study is to compare risk and returns of an investment portfolio by adjusting asset weights using Monte Carlo Simulation and optimization algorithm. I have tried to answer the below questions through this study –

- How to calculate expected return for a portfolio of stocks by fetching its adjusted closing prices

- How to calculate standard deviation or volatility for daily returns

- Introduce the concept of Sharpe Ratio to determine what weight of assets to allocate for maximum returns and minimum risk in a portfolio

Approach -

Two approaches are used to determine the optimum allocation of weights in a portfolio

i) Monte Carlo Simulation – 25000 random portfolios are simulated and risk, returns and sharpe ratio are calculated for each. The portfolios having maximum sharpe ratio and minimum volatility are compared.

ii) Optimization using Scipy - The portfolios having maximum sharpe ratio and minimum volatility are compared and the set of portfolios that provide a target return with minimum volatility are found by iteratively applying the SciPy optimizer.

Data – A portfolio comprising of 10 S&P stocks is used in this analysis (this is a real investment portfolio of a friend of mine) – **Facebook, Netflix, Salesforce, Budweiser, Axon, Microsoft, GW Pharma, Square, Chevron and Apple**. Daily Adjusted Closing prices for all stocks for the period Jan 4,2016 – March 31, 2019 are used (downloaded from yahoo finance in python). Risk Free Rate of 0.11% is used (as sourced from Department of Treasury website on March 31, 2020)

Result -

Monte Carlo simulation:

	Max Sharpe Ratio portfolio	Minimum volatility portfolio
Expected Annual Return	26%	6%
Volatility	27%	22%

Optimization:

	Max Sharpe Ratio portfolio	Minimum volatility portfolio
Expected Annual Return	28%	2.7%
Volatility	26%	21.5%

The optimization gives better expected annual returns in case of Maximum Sharpe portfolio while Monte Carlo Simulation gives better returns for minimum volatility portfolio at almost similar risk.

Background

To understand investment optimization, we derive concepts from Modern Portfolio theory and Capital Asset Pricing Model. We will later discuss the methods that use these concepts to give an optimized portfolio.

A. Modern Portfolio Theory

Modern Portfolio Theory is a theory about how investors (who are risk averse) construct portfolios that maximize their expected returns for given levels of risk.

Assumptions of the theory are –

- Investors are rational and avoid risks whenever possible

- Investors aim for the maximum returns for their investment

- All investors share the aim maximizing their expected returns

- Commissions and taxes on the market are left out of consideration

- All investors have access to the same sources and level of all necessary information about investment decisions

- Investors have unlimited access to borrow and lend money at the risk free rate

The fundamental tenet of this theory is the possibility for investors to construct an **"efficient set of portfolios — Efficient Frontier**" that offers the maximum expected returns for a given level of risk

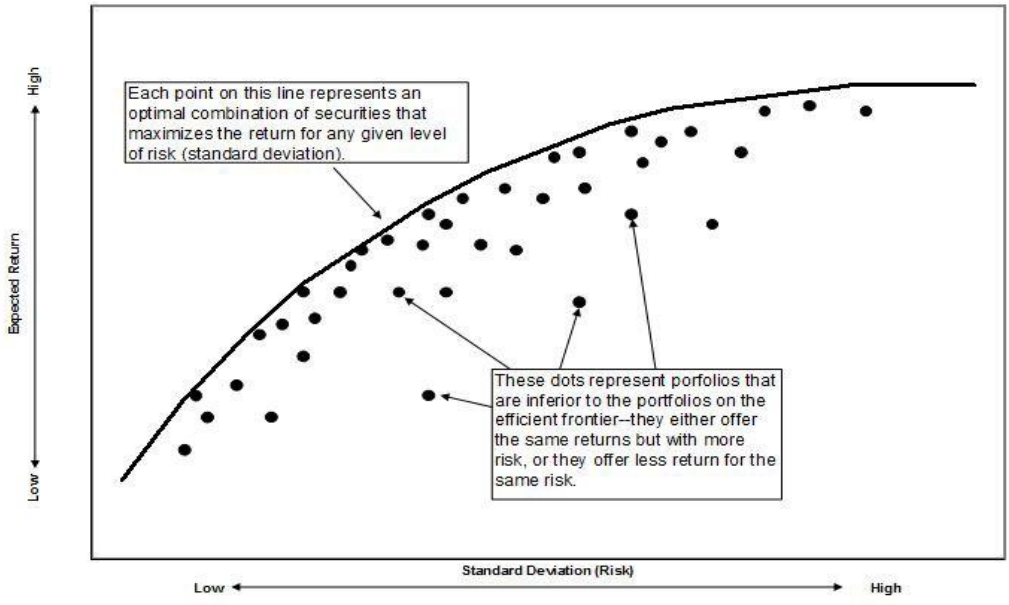


Figure 2.3: Reward versus risk, a selection of risky assets and the efficient frontier (bold black)

For a portfolio of assets that are potentially correlated to one another, the weighted average of the mean returns, and the volatility is calculated using the below formula:

$$Mean(Portfolio\ Returns) = \sum_{i=1}^{n} w_i r_i$$

$$Var(Portfolio\ Returns) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_i w_j \rho_{i,j} \sigma_i \sigma_j = \mathbf{w}^T \bullet (\mathbf{cov} \bullet \mathbf{w})$$

B. Sharpe Ratio (Derived from Capital Asset Pricing Model) -

Developed by Nobel Laureate William F. Sharpe, the Sharpe Ratio is a measure for calculating risk-adjusted return and has been the industry standard for such calculations.

The Sharpe Ratio allows us to quantify the relationship the average return earned in excess of the risk-free rate per unit of volatility or total risk.

$$Sharpe\ Ratio = \frac{R_p - R_{rf}}{\sigma_p}$$

R_p = Expected portfolio/asset return

R_{rf} = Risk-free rate of return

σ_p = Portfolio/asset standard deviation

(src: Investopedia)

In this study, we will study two types of portfolios -

Optimal portfolio with the highest Sharpe ratio for investors with higher risk appetite

Minimum volatility portfolio with for the most risk-averse investors

Approach

I have used two methods for portfolio optimization

A. Monte Carlo simulation
B. Optimization algorithm using Scipy in python

A. Monte Carlo method

One approach to optimizing a portfolio is application of the Monte Carlo Method. It incorporates the idea of carrying out repeated trials using randomly generated inputs and observing the outcomes. The methodology revolves around the concept that as the number of portfolios are increased, we get closer to the actual optimum portfolio.

Steps for optimization:

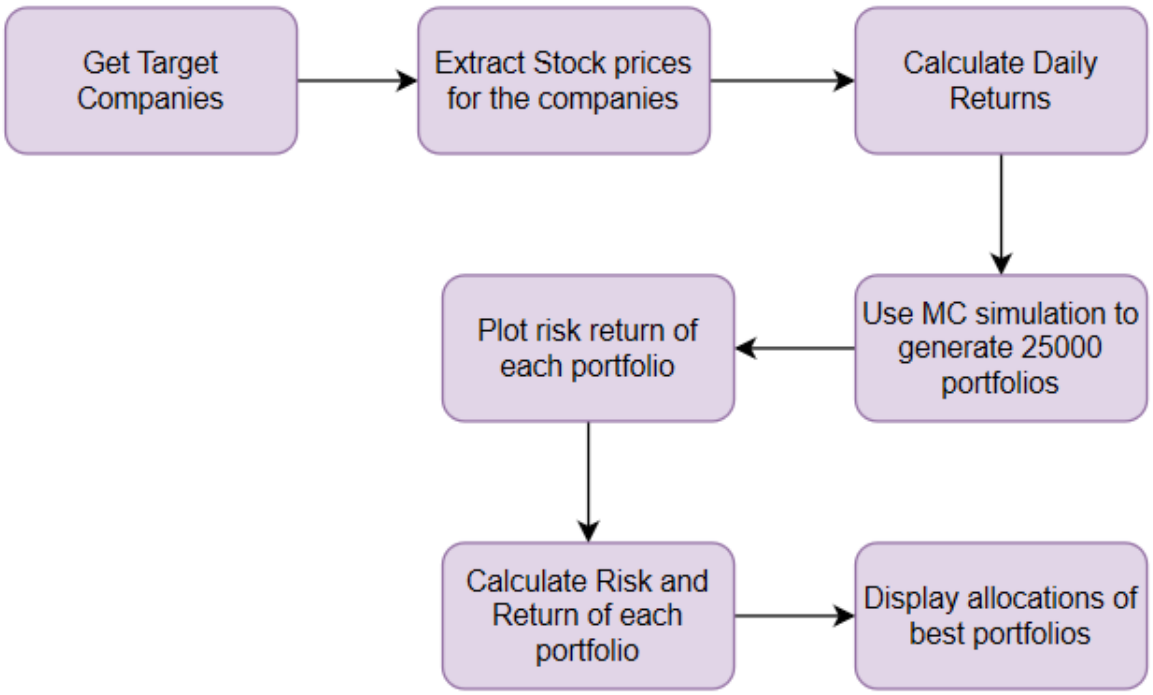


Figure: Investment optimization using Monte Carlo method

B. Optimization algorithm using scipy in python

This is a mathematical optimization algorithm. I have used minimize function from scipy.optimize library from python. A constraint of sum of weight allocations should equal to 1 is applied and initially equal weight allocations are made for all assets in the portfolio

Portfolio Analysis

Adjusted closing price for the period Jan 4,2016- March 31,2020 of a portfolio comprising of 10 S&P stocks is used in this analysis – Facebook, Netflix, Salesforce, Budweiser, Axon, Microsoft, GW Pharma, Square, Chevron and Apple.

Glimpse of the data:

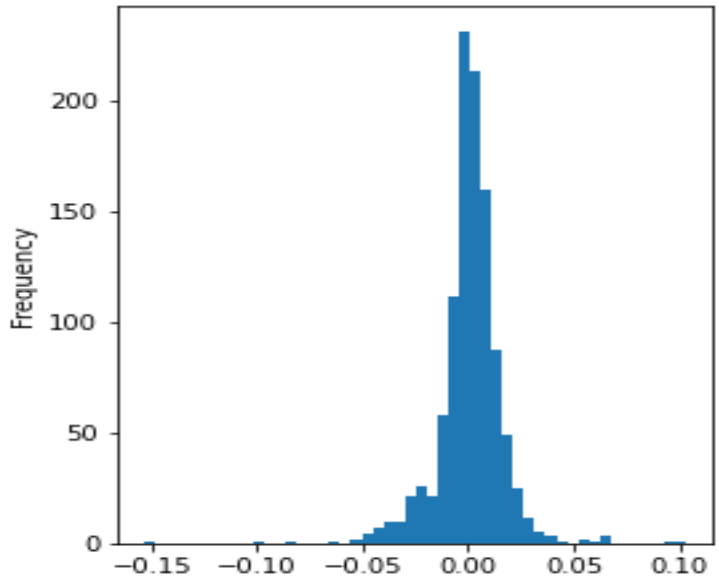
Symbols	FB	NFLX	CRM	BUD	AAXN	MSFT	GWPH	SQ	CVX	AAPL
Date										
2016-01-04	102.220001	109.959999	76.709999	107.600929	16.830000	50.398834	68.459999	12.160000	74.822914	98.213585
2016-01-05	102.730003	107.660004	77.050003	107.671211	17.020000	50.628761	68.370003	11.510000	75.462914	95.752419
2016-01-06	102.970001	117.680000	76.290001	105.632889	17.639999	49.709072	63.860001	11.520000	72.481781	93.878586
2016-01-07	97.919998	114.559998	74.300003	104.965157	16.590000	47.980057	61.810001	11.160000	69.913307	89.916473
2016-01-08	97.330002	111.389999	73.230003	102.909256	16.280001	48.127216	59.860001	11.310000	69.163826	90.391907
...	...	...	...	...	...	...	...	...	...	...
2020-03-25	156.210007	342.390015	147.059998	45.590000	71.150002	146.919998	88.910004	52.389999	69.269997	245.520004
2020-03-26	163.339996	362.989990	154.729996	46.299999	76.519997	156.110001	86.559998	56.029999	76.379997	258.440002
2020-03-27	156.789993	357.119995	146.000000	42.990002	74.410004	149.699997	85.750000	53.340000	68.779999	247.740005
2020-03-30	165.949997	370.959991	149.850006	42.869999	69.459999	160.229996	85.180000	55.000000	71.949997	254.809998
2020-03-31	166.800003	375.500000	143.979996	44.119999	70.769997	157.710007	87.570000	52.380001	72.459999	254.289993

Average Daily Return: 0.09%

Average daily volatility: 1.58%

Sharpe Ratio: 0.91

Histogram of Daily Returns:



The distribution is close to a normal distribution

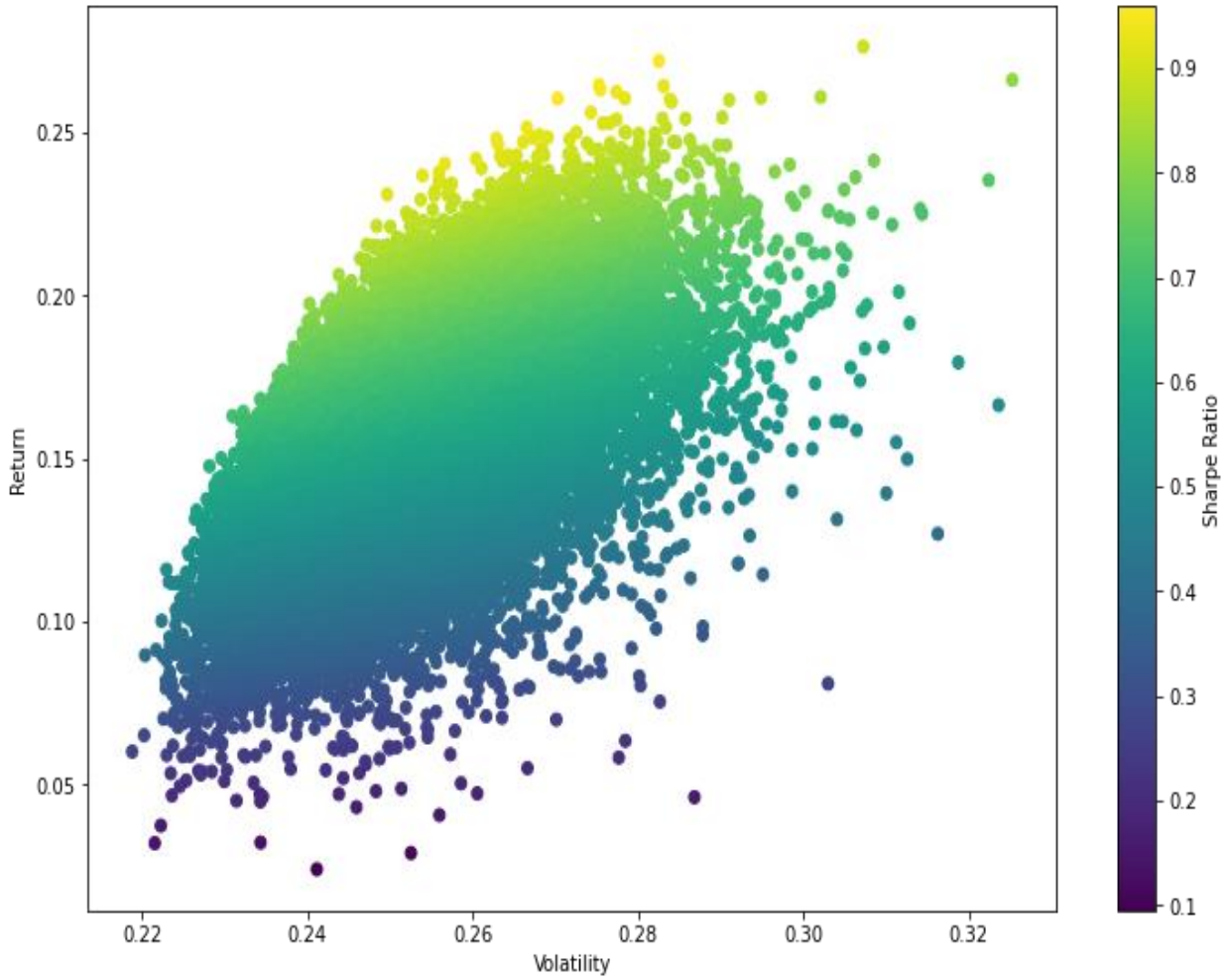
Portfolio Optimization

Monte Carlo Simulation

- In the plot below, 25000 portfolios with randomly varying weights of the assets were generated and evaluated.(code in appendix)

- Their expected annual return is then plotted versus the historical volatility of the portfolio.

- Further, each point representing a portfolio has been shaded according to the Sharpe Ratio



The above figure reveals interesting outcomes. It can be said that changing the weight of each asset in the portfolio will have a dramatic effect on the expected return and the level of risk that the investor is exposed to. For example if the investor is targeting a 20% return, volatility varies from 24% to almost 32%. Thus choosing how much weight of each asset should be considered carefully

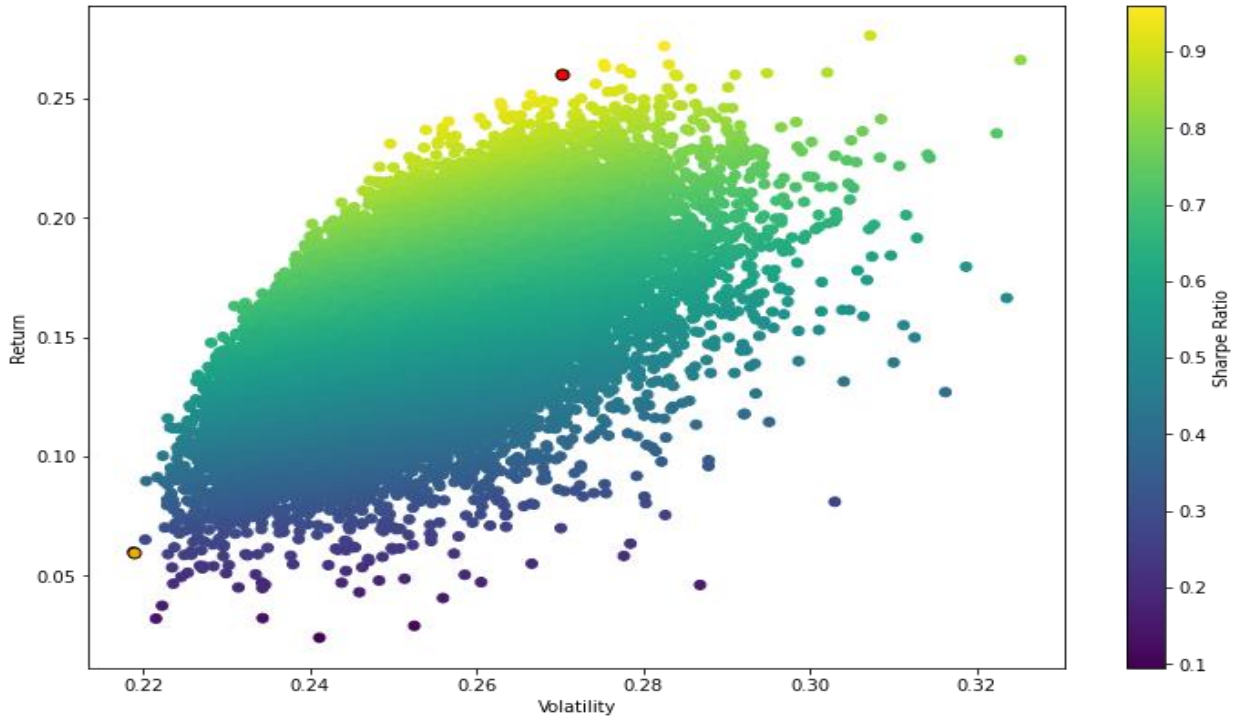
	Max Sharpe Ratio portfolio	Minimum volatility portfolio
Expected Annual Return	26%	6%
Volatility	27%	22%
Sharpe Ratio	0.95	0.27
Asset weights	{'FB':0.03,'NFLX':0.068, 'CRM':0.012,'BUD':0.004, 'AAXN':0.262,'MSFT':0.192, 'GWPH':0.023,'SQ':0.125, 'CVX':0.054,'AAPL':0.228}	{'FB':0.036,'NFLX':0.044, 'CRM':0.002,'BUD':0.287, 'AAXN':0.059,'MSFT':0.196, 'GWPH':0.045,'SQ':0.003, 'CVX':0.202,'AAPL':0.124}

Table: Comparison of Minimum volatility and Maximum Sharpe Ratio portfolios(Monte Carlo Simulation)

Observations:

Investors seeking the maximum risk-adjusted return would opt for portfolio with the maximum Sharpe Ratio which has an expected return of 26% with expected volatility pegged at 27%. The most risk-averse investor would construct the minimum variance portfolio which has an expected return of 6% with an accompanying expected volatility of 22%. Clearly, the increase in returns is much higher compared to increase in risk for Maximum Sharpe portfolio over Minimum volatility portfolio

Below plot shows the point (red dot) for maximum sharpe ratio portfolio and (orange dot) for minimum volatility portfolio



Optimization Algorithm

From the MC simulation, it is evident that a target return can be achieved with a wide range of risk levels. This introduces the concept of the "Efficient Frontier." The Efficient Frontier is the set of portfolios that achieve a given return with the minimum amount of risk for that return. In the above case, these were the portfolios furthest to the left for each expected return.

Keeping this concept in mind, a better approach can be applied to the selection of asset weights such that we consider only these efficient portfolios which meet a criteria important to the investor. First, we could optimize the weights to target a metric such as the Sharpe Ratio. Second, we could opt to find the minimum volatility portfolio and accept the return that it provides.

I have used the optimization algorithm from scipy.optimize.

Few constraints were added for the optimization objective –

i) The allocations needs to add up to one

ii) An initial guess to start with, which is nothing but equal distribution of weight allocations

	Max Sharpe Ratio portfolio	Minimum volatility portfolio
Annual Return	28%	2.7%
Volatility	26%	21.5%
Sharpe Ratio	1.08	0.12
Asset weights	{'FB':0.00,'NFLX':0.133, 'CRM':0.00,'BUD':0.00, 'AAXN':0.205,'MSFT':0.595, 'GWPH':0.00,'SQ':0.017, 'CVX':0.00,'AAPL':0.05}	{'FB':0.084,'NFLX':0.038, 'CRM':0.029,'BUD':0.376, 'AAXN':0.038,'MSFT':0.111, 'GWPH':0.03,'SQ':0.00, 'CVX':0.124,'AAPL':0.169}

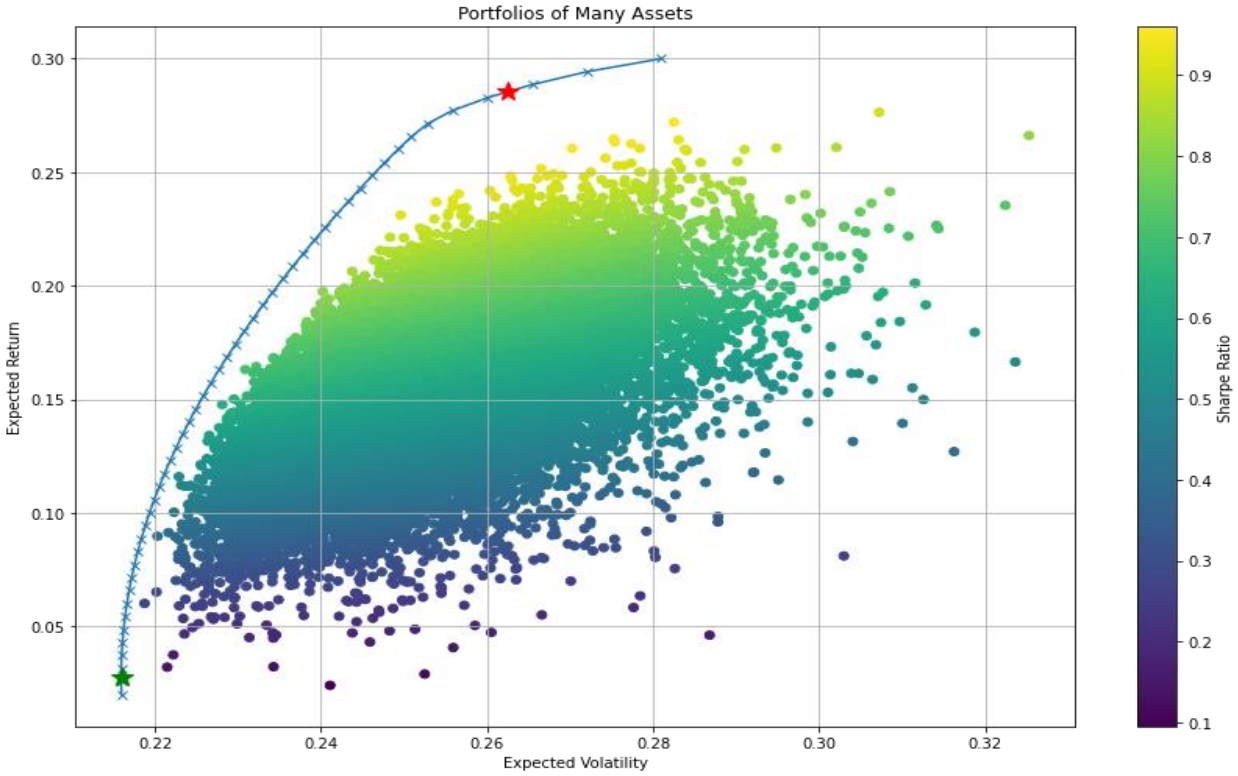
Table: Comparison of Minimum volatility and Maximum Sharpe Ratio portfolios(Optimization Algorithm)

Observations:

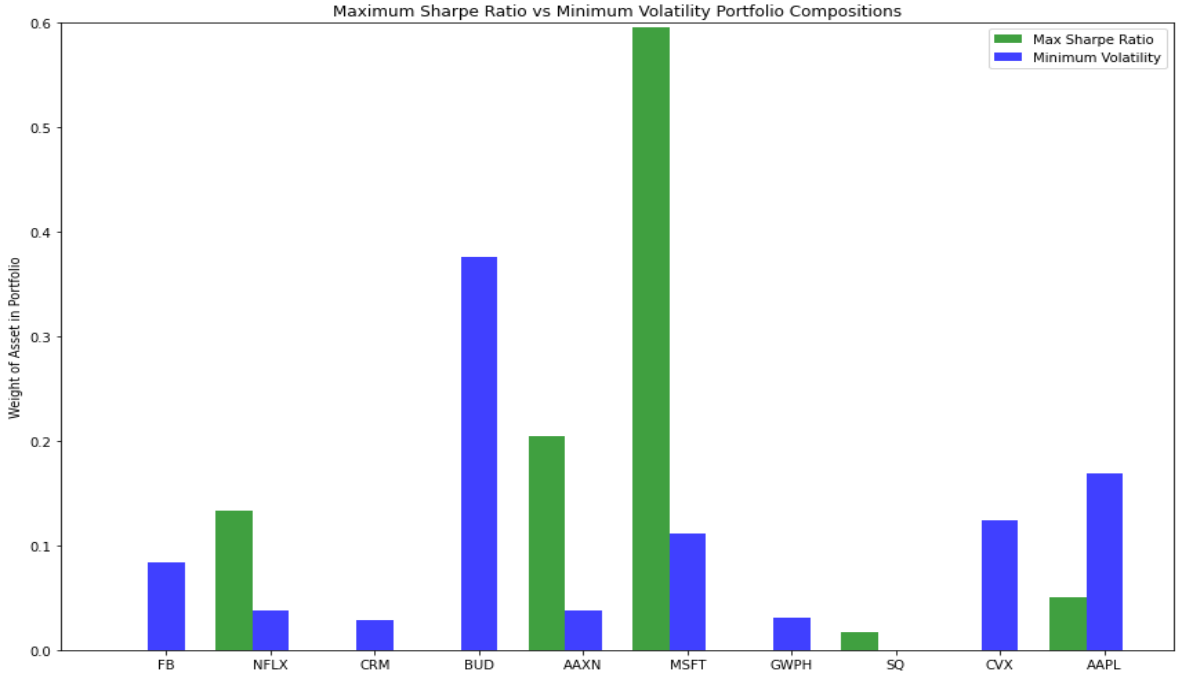
Investors seeking the maximum risk-adjusted return would opt for portfolio with the maximum Sharpe Ratio which has an expected return of 28% with expected volatility pegged at 26%. The most risk-averse investor would construct the minimum variance portfolio which has an expected return of 2.7% with an accompanying expected volatility of 21.5%.

It is interesting to note that using the maximum sharpe portfolio the weights for stocks like FB, CRM, BUD, GWPH and CVX is zero. It means the optimization objective suggests that we do not include these stocks when building the portfolio.

Below plot shows the point (red dot) for maximum sharpe ratio portfolio and (orange dot) for minimum volatility portfolio. Blue Line represents the Efficient Frontier which are the portfolios that give maximum expected returns for a given level of risk



The below plot shows the comparison of weights of assets to be used for Maximum Sharpe and Minimum volatility portfolios



Results and Recommendations

We were introduced to the concept of Modern portfolio theory and Sharpe ratio in this study. Both these concepts were used by monte carlo simulation and optimization algorithm methods to optimize the asset weights in the investment portfolio.

Monte Carlo simulation:

	Max Sharpe Ratio portfolio	Minimum volatility portfolio
Expected Annual Return	26%	6%
Volatility	27%	22%

Optimization:

	Max Sharpe Ratio portfolio	Minimum volatility portfolio
Expected Annual Return	28%	2.7%
Volatility	26%	21.5%

The optimization gives better expected annual returns in case of Maximum Sharpe portfolio while Monte Carlo Simulation gives better returns for minimum volatility portfolio at almost similar risk.

In real life, investing decision really depends on the risk appetite of the investor. Few investors are risk averse and they should always choose portfolios which are created using minimum volatility while for risk taking investors, maximum sharpe ratio should be the guiding metric. Also, using the efficient frontier, the expected returns can be maximized for a given level of risk which investors should use if they know how much risk they can bear

References and Appendix

References

https://www.linkedin.com/pulse/optimizing-portfolio-mpt-capm-using-python-antti-v%C3%A4is%C3%A4nen/

https://ebrary.net/7079/business_finance/what_modern_portfolio_theory

Code:

To run the below code, upload the attached file in Google Colab jupyter notebook and run.

StockPortfolio_Optimization.ipynb

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from pandas_datareader import data

# Set the start and end date

start_date = '2016-01-01'

end_date = '2020-03-31'

# Set the ticker

tickers = ['FB', 'NFLX','CRM','BUD', 'AAXN','MSFT','GWPH', 'SQ','CVX','AAPL']

weights = np.array(np.random.random(len(tickers)))

weights = weights/np.sum(weights)

ticker_allocs =
{'FB':0.13,'NFLX':0.06,'CRM':0.07,'BUD':0.09,'AAXN':0.05,'MSFT':0.12,'GWPH':0.09,'SQ':0.06,'CVX':0.07,'AAPL':0.32}

total_inv_amount = 20228.18

df = data.get_data_yahoo(tickers, start_date, end_date)['Adj Close']

df1 = df.copy()

for k,v in ticker_allocs.items():

    df1[k] = df1[k]/df1.iloc[0][k]

    df1[k] = df1[k]*v
```

```
    df1[k] = df1[k]*total_inv_amount
df1['Total'] = df1.sum(axis=1)
df1['Daily Return'] = df1['Total'].pct_change(1)

# average daily return
print(df1['Daily Return'].mean())
# standard deviation
print(df1['Daily Return'].std())
df1['Daily Return'].plot(kind='hist', bins=50, figsize=(4,5))
#Sharpe Ratio
sharpe_ratio = df1['Daily Return'].mean() / df1['Daily Return'].std()
sharpe_ratio
ASR = (252**0.5) * sharpe_ratio
ASR
log_return = np.log(df/df.shift(1))
log_return.head()
meanReturns = log_return.mean()
covMatrix = log_return.cov()
rfr = 0.0011
# Monte Carlo Simulation
num_portfolios = 25000
all_weights = np.zeros((num_portfolios, len(df.columns)))
ret_arr = np.zeros(num_portfolios)
vol_arr = np.zeros(num_portfolios)
sharpe_arr = np.zeros(num_portfolios)
for ind in range(num_portfolios):
    # weights
    weights = np.array(np.random.random(len(tickers)))
    weights = weights/np.sum(weights)
```

```
    # save the weights
    all_weights[ind,:] = weights
    # expected return
    ret_arr[ind] = np.sum((log_return.mean()*weights)*252)
    # expected volatility
    vol_arr[ind] = np.sqrt(np.dot(weights.T,np.dot(log_return.cov()*252, weights)))
    # Sharpe Ratio
    sharpe_arr[ind] = (ret_arr[ind]-rfr)/vol_arr[ind]
# maximum sharpe ratio
sharpe_arr.max()
#get index of maximum sharpe ratio to calculate weights
argmax = sharpe_arr.argmax()
all_weights[argmax,:]
# get returns and volatility coreesponding to maximum sharpe ratio
max_sr_ret = ret_arr[argmax]
max_sr_vol = vol_arr[argmax]
print(max_sr_ret)
print(max_sr_vol)
#get minimum volatility
min_vol = vol_arr.min()
#get index of minimum volatility to calculate weights
argmin = vol_arr.argmin()
all_weights[argmin,:]
# get returns and sharpe ratio coreesponding to minimum volatility
min_vol_ret = ret_arr[argmin]
min_vol_sr = sharpe_arr[argmin]
print(min_vol_ret)
print(min_vol_sr)
#plot volatility vs return
```

```
plt.figure(figsize=(12,8))
plt.scatter(vol_arr,ret_arr,c=sharpe_arr,marker='o')
plt.colorbar(label='Sharpe Ratio')
plt.xlabel('Volatility')
plt.ylabel('Return');
# plot the maximumsharpe ratio and minimum volatility points
plt.figure(figsize=(12,8))
plt.scatter(vol_arr,ret_arr,c=sharpe_arr,marker ='o')
plt.colorbar(label='Sharpe Ratio')
plt.xlabel('Volatility')
plt.ylabel('Return')
# add a red dot for max_sr_vol & max_sr_ret
plt.scatter(max_sr_vol, max_sr_ret, c='red', s=50, edgecolors='black');
plt.scatter(min_vol, min_vol_ret, c='orange', s=50, edgecolors='black');
# function that calculates and returns the annual returns, volatility and sharpe ratio
def ret_vol_sr(weights):
   weights = np.array(weights)
   ret = np.sum(log_return.mean()*weights)*252
   vol = np.sqrt(np.dot(weights.T,np.dot(log_return.cov()*252,weights)))
   sr = (ret-rfr)/vol
   return np.array([ret,vol,sr])
from scipy.optimize import minimize
# minimize negative Sharpe Ratio
def neg_sharpe(weights):
        return ret_vol_sr(weights)[2] * -1
# minimize portfolio volatility
def portfolio_vol(weights):
        return ret_vol_sr(weights)[1]
# check allocation sums to 1
```

```
def check_sum(weights):
        return np.sum(weights) - 1



# annual returns
def portfolio_ret(weights):
    return ret_vol_sr(weights)[0]
# create constraint variable
cons1 = ({'type':'eq','fun':check_sum})
# bounds
bounds = tuple( (0,1) for asset in range(len(tickers)))
# initial guess
percent_alloc = 1/len(tickers)
init_guess = []
for i in range(len(tickers)):
  init_guess.append(percent_alloc)
# get the maximum sharpe portfolio by minimizing the negative sharpe ratio
max_sharpe_portfolio = minimize(neg_sharpe, init_guess, method='SLSQP', bounds=bounds,
constraints=cons1)
# get the minimum volatility portfolio
min_vol_portfolio = minimize(portfolio_vol, init_guess, method='SLSQP', bounds=bounds,
constraints=cons1)
#annual returns, volatility, sharpe ratio of maximum sharpe portfolio
et_vol_sr(max_sharpe_portfolio.x)
#annual returns, volatility, sharpe ratio of minimum volatility portfolio
ret_vol_sr(min_vol_portfolio.x)
# weights of assets in maximum sharpe portfolio
np.round(max_sharpe_portfolio.x,3)
# weights of assets in minimum volatility portfolio
```

```
np.round(min_vol_portfolio.x,3)

# target returns for efficient frontier

targetReturns = np.linspace(0.02, 0.30, 50)

# efficient portfolios with minimu volatility

efficientPortfolios=[]

for target in targetReturns:

    cons2 = ({'type':'eq','fun':lambda x: portfolio_ret(x)-target},{'type':'eq','fun':check_sum})

    efficientPortfolios.append(minimize(portfolio_vol, init_guess, method='SLSQP', bounds=bounds,
constraints=cons2))

# plot the data

plt.figure(figsize=(12,8))

plt.scatter(vol_arr,ret_arr,c=sharpe_arr,marker ='o')

plt.xlabel('Volatility')

plt.ylabel('Return')

plt.plot([p['fun'] for p in efficientPortfolios], targetReturns, marker='x')

rp = ret_vol_sr(max_sharpe_portfolio['x'])[0]

sdp = ret_vol_sr(max_sharpe_portfolio['x'])[1]

plt.plot(sdp, rp, 'r*', markersize=15.0)

rp1 = ret_vol_sr(min_vol_portfolio['x'])[0]

sdp1 = ret_vol_sr(min_vol_portfolio['x'])[1]

plt.plot(sdp1, rp1, 'g*', markersize=15.0)

plt.grid(True)

plt.xlabel('Expected Volatility')

plt.ylabel('Expected Return')

plt.colorbar(label='Sharpe Ratio')

plt.title('Portfolios of Many Assets')

plt.tight_layout()

plt.savefig('Monte Carlo Simulation for Portfolio', dpi=100)

# plot weights comparison for maximum sharpe vs minimum volatility portfolios
```

```
ind = np.arange(len(tickers))
width = 0.35
fig, ax = plt.subplots(figsize=(12,8))
rects1 = ax.bar(ind, max_sharpe_portfolio['x'], width, color='g', alpha=0.75)
rects2 = ax.bar(ind + width, min_vol_portfolio['x'], width, color='b', alpha=0.75)
ax.set_ylabel('Weight of Asset in Portfolio')
ax.set_ylim(0,0.6)
ax.set_title('Maximum Sharpe Ratio vs Minimum Volatility Portfolio Compositions')
ax.set_xticks(ind + width)
ax.set_xticklabels(tickers)
plt.tight_layout()
ax.legend((rects1[0], rects2[0]), ('Max Sharpe Ratio', 'Minimum Volatility'))
#plt.savefig('Portfolio Compositions', dpi=100)
plt.show()
```