Portfolio Optimization

MSDS460 SEC58 Final Project
Winter 2019
Dr. Carrie Dugan
Aamer Khan, Annie Reber, Bo Hong

Abstract:

This paper explores the foundations of portfolio optimization in order to find the optimal allocations of resources to invest in a sample of four different stocks. The two primary goals when selecting the portfolio weights are to minimize risk and maximize reward. Research shows that Markowitz's Mean-Variance Optimization is the foundation of portfolio optimization to achieve the goals successfully. Building off Markowitz's, Sharpe's Ratio was used to measure the maximum returns of the various portfolio weights with the standard deviation and variance measured the volatility which best represents a risk. In order to visualize the different choices of the portfolio and select the portfolios, an Efficient Frontier was built and leveraged to pick the options for best maximum return and minimized risk. As a result, we were able to see which stocks appeared to be the safest bet to gain on investment versus which were riskier with a potential higher reward.

Keywords: Portfolio Optimization, Mean-Variance, Sharpe's Ratio, Volatility, Efficient Frontier

Introduction:

Portfolio optimization is a stock allocation strategy that is applied to determine the best distribution of assets by either minimizing the overall portfolio risk at a specific rate of return or maximizing the return at a set level of risk. As beginner investors, we analyzed if we can find the optimal distribution of assets with minimal overall risk and maximum portfolio return by diversifying our portfolio using different industries with limited research. Stocks of four companies operating in different industries were purposely chosen to evaluate if portfolio diversification is achievable by merely combining assets from different trades. Therefore, we chose the following stocks operating in distinct sectors: Nvidia Corporation, Bank of America, Starbucks and Nike.

To construct and evaluate the portfolio, we applied Harry Markowitz's Modern Portfolio Theory (MPT) and specifically his work on Mean-Variance Optimization. While variations of MPT exist, this project sought to construct an optimal portfolio using the mean-absolute deviation model. We build an efficient frontier to show the variations of the optimal portfolio with the highest return based on different levels of risk. We then used the Sharpe Ratio and standard deviation as return and risk measures to evaluate our model and findings. While doing so, we explored other variations and improvements of MPT and highlighted the advantages and disadvantages of those methods.

Literature Review:

Mean-Variance Optimization (MVO) from Markowitz's Modern Portfolio Theory (MPT) has been the basis for most portfolio optimization since 1952. (Krokhmal, Uryasev & Palmquist, 2001) MVO is based on expected utility theory where mean and variance are balanced to the relative expected return. These expected returns can be used to build an efficient frontier where investors can choose a portfolio on this mean-variance efficient frontier based on their risk level. The goal of portfolio optimization is to determine the weights of the stocks with minimum portfolio risk at a specific rate of return.

In 1998, Konno and Yamazaki introduced a similar model, the mean absolute deviation, to minimize portfolio risk within the Tokyo Stock Market. The absolute deviation being used as a risk measure, and the mean return is used as an expected return in the mean-absolute deviation model. In their paper, they deal with using a linear underestimating function for a concave cost function to calculate a good bound, and demonstrate that a reasonably large scale problem can be solved in an efficient manner using the real stock data and transaction cost table in the Tokyo Stock Exchange. This development meant that they can now handle the transaction costs as long as it is a concave function of the amount of transaction. Thus, the mean-absolute deviation model has been applied to the portfolio optimization problem in different markets such as Japan, Germany, and France. (Konno & Yamazaki, 1998)

MVO has its limitations. For example, the founder Markowitz once quoted MVO works "as long as we are talking about portfolios that rarely lose much more than 30 percent or 40 percent or gain much more than 40 percent or 50 percent." (Kaplan, 2017) However, extreme returns happen more frequently than Markowitz's original assumption. Therefore, since its inception in 1952, different methods have also been introduced such as Post Modern Portfolio Theory (PMPT) by Rom and Ferguson and what Kaplan (2017) dubbed as Markowitz 2.0 that uses Value-at-Risk (VaR) and Conditional-Value-at-Risk (CVaR) among others as risk measures. Rom and Ferguson argued that MVO is flawed because it takes into consideration the overall volatility, which includes both losses and gains. Rom and Ferguson argued that PMPT is superior because it focuses only on the downside deviation rather than standard deviation and only seeks to minimize losses than the overall volatility. Markowitz also recognized the shortfall of using the overall standard deviation but used it in place for ease of calculations.

However, Kaplan (2017) that disadvantage of PMPT is the measure of reward is the expected arithmetic and fails to factor in performance over time. Instead, Kaplan suggested that variations of efficient frontier should be created using a variety of risk and reward measures based on the scenario. He coined this general framework as Markowitz 2.0. In his article, Kaplan concluded that "the efficient frontier that is most relevant to investors is the one that has the expected geometric mean as the measure of reward and CVaR as the measure of risk because it shows the tradeoff between reward and risk that is most meaningful to investors, namely, long term potential growth versus potential short-term loss."

VaR is the upper percentile of the loss distribution which can be efficiently estimated and managed when the underlying risk factors are normally distributed. CVaR is the conditional expected loss exceeding VaR for continuous distributions and a weighted average of VaR and losses exceeding VaR for general distributions. CVaR simultaneously calculated VaR to optimize CVaR to keep risk low. In a paper by Krokhmal, Palmquist, and Uryasev (2001) they compared running portfolio optimization on S&P100 stocks compared to Mean-Variance. By using efficient frontier graph analysis, it was found that the differences to not be very significant and both returned near-optimal results for each other.

In our paper, we ran a portfolio optimization utilizing a mean-variance model with efficient frontier to select the best portfolio. Since we are beginner investors and want to start a basis of understanding portfolio optimization, the foundations are the best place to start. Our approach is not the state-of-the-art approach and could probably be improved upon. However, since there are downsides and no significant deviations between specific state-of-the-art approaches, using a mean-variance approach with efficient frontier graphs is a great way to start portfolio optimization. We did incorporate the use of Sharpe Ratio to our mean-variance analysis, which was founded by William F. Sharpe and has been used to evaluate the return of an investment to its risk.

The formula for Sharpe Ratio is:

$$Sharpe\ Ratio = \frac{R_p - R_f}{\sigma_p}$$

Figure 1

where R_p , R_f , and σ_p equal the return of portfolio, risk-free rate and the standard deviation of the portfolio's excess return. Since its inception, the Sharpe Ratio is used to measure the risk-adjusted rate. Combining assets with low correlations should diversify portfolio and can potentially decrease the overall portfolio risk without decreasing the expected return. The Sharpe Ratio would increase, in this case, as it is a reward-to-variability ratio. By adjusting the expected rate by the risk, it captures a more true view of the portfolio's expected performance.

Methodology:

Our approach will involve looking at portfolio optimization with Python. The critical aspect of our decision analytics modeling stems from initially looking at the past for historical data on our stocks. We will leverage ideas from the mean-variance model to solve our problem. The objective of portfolio optimization problem is diversification and essentially boils down to the idea that blending risky stocks, such that the combined expected return is aligned with an investor's target, will provide a lower risk-level than carrying just one or two stocks by themselves. The technique is even more effective when the stocks are not well correlated with one another.

The primary methodology will focus on looking into the mean-variance of a portfolio of stocks that are potentially correlated to one another. We will gather the weighted average of the mean returns and the sum of the terms in the covariance matrix for the stocks. We will utilize the

Python package NumPy as it provides us with all of the functions necessary to calculate the mean-variance of returns for individual assets and portfolios. Using this package and we will be able to retrieve historical quotes, calculate the mean-variance of the assets, and return the expected portfolio performance in the end. In term of model formulation, we will employ the Monte Carlo Method to optimize a portfolio via creating a simulation of many randomly generated portfolios with varying weighted stocks and observing the outcomes.

As an approach, we intend to generate randomly varying weights of our stocks and evaluate them. Their expected annual return will be plotted versus the historical volatility of the portfolio. Lastly, each point in the plot shall represent a generated portfolio and will be shaded according to the Sharpe's Ratio. The ratio itself is simply used to understand the return of an investment compared to its risk and will act as an indicator for us to evaluate or measure the performance of our randomly generated portfolios.

Some of the mean-variance assumptions are important to be aware of and include the following. Starting with the idea that investors are risk averse in that they prefer higher return for a given level of risk (variance, standard deviation), or they want to minimize risk for a given level of returns. The degree of risk aversion may vary from investor to investor. Secondly, investment returns are normally distributed, so only returns, variances, and covariances are needed to derive the optimal portfolio.

For sourcing, our data we will be working with Yahoo Finance in order to access financial information about stocks including their historical data. The Python library Pandas will allow us an easy way to pull stock quotes this source, which is great since we want to estimate portfolio performance using past pricing data. For validating our model, we will leverage concepts from Efficient Frontier to see how we did. To recap, the Efficient Frontier is the set of optimal portfolios that offers the highest expected return for a defined level of risk or the lowest risk for a given level of expected return. Portfolios that lie below the efficient frontier are sub-optimal because they do not provide enough return for the level of risk. Portfolios that cluster to the right of the efficient frontier are sub-optimal because they have a higher level of risk for the defined rate of return. Thus, keeping this concept of Efficient Frontier in mind, a more structured approach can be applied to the selection of stock weights such that we consider only these efficient portfolios which meet criteria relevant to the investor. First, we could optimize the weights to target a metric such as the Sharpe Ratio as mentioned earlier. Or, we could opt to find the minimum volatility portfolio and accept the return that that provides. We will leverage various python optimization libraries to solve these problems in the end.

As far as sensitivity analysis is concerned, when looking at a mean-variance portfolio, if C0 is raised by one, the investment in each asset of the minimum variance portfolio is increased with the derivative, so with

$$\frac{\partial \theta_{mv}}{\partial C_0} = \frac{\partial \Sigma^{-1} \bar{1} \frac{C_0}{c}}{\partial C_0} = \Sigma^{-1} \bar{1} \frac{1}{c}$$

Figure 2

Which is independent of the parameter C0. So if C0 is multiplied with a factor x, the optimal solution also raises with factor x. In other words, it doesn't matter how much money an investor can invest; the proportions invested in each asset always stay the same. This can be verified by the fact that the invested fractions are given by:

$$\frac{\theta_{mv}}{C_0} = \frac{\Sigma^{-1}\bar{1}\frac{C_0}{c}}{C_0} = \Sigma^{-1}\bar{1}\frac{1}{c}$$

Figure 3

Computational Experiment and Results:

The experiment we wanted to carry out required us to realize that we are not interested in expected return and volatility or standard deviation of individual stocks, but instead, we want that information for the portfolio of stocks all together. As a result, we will capture the benefits of diversification of the correlation between the stocks in the portfolio. Our portfolio of 4 stocks consists of, Bank of America, Nike, Nvidia, and Starbucks. Ultimately we want to calculate the expected return and volatility of that portfolio.Well, to begin that process we need as our first input, the weights of the stocks in the portfolio or other words, how much of each stock do we hold as a percentage of the entire portfolio holdings. First, we need a baseline scenario. Thus, let's suppose our portfolio is made up of 50% Bank of America stock, 20% Nike stock, 20% Nvidia stock and 10% Starbucks. We can calculate the portfolio expected return and volatility as follows. First, we downloaded the daily price data for each of the stocks in the portfolio. Secondly, we converted daily stock prices into daily returns. This will allow us to calculate mean daily return and covariance of daily returns efficiently using the .mean() and .cov() methods in Python. We can then set an array called 'weights' holding portfolio weights of each stock.

Finally, we calculated the annualized portfolio return and volatility. We used the Python package NumPy to calculate the mean variance of returns for individual assets and portfolios. So what we learned was that we could retrieve historical data, calculate the mean-variance of the stocks, and return the expected portfolio performance if we have a set of assets and their weights within a portfolio. We created a baseline test portfolio, to begin with, because we now needed to use randomly generated inputs and observe their outcomes in the portfolio. We ended up doing a simulation of this using the Monte Carlo method to create 25000 randomly weighted portfolios.

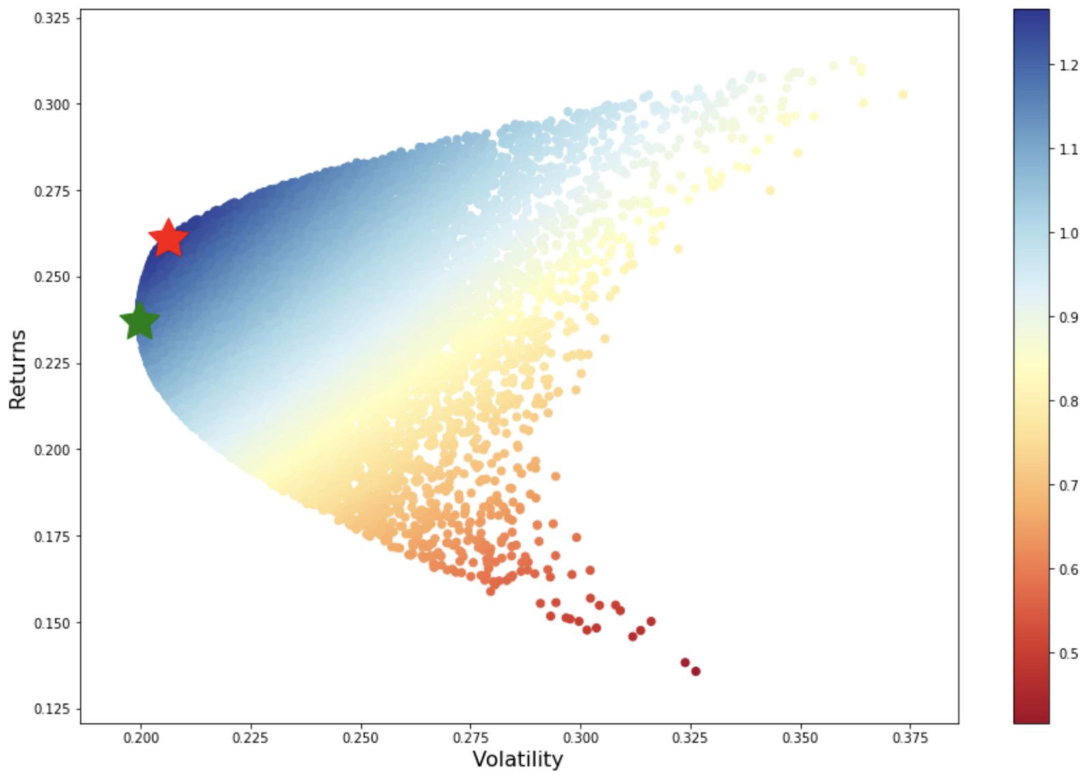

Figure 4

You can tell that changing the weight of each stock in the portfolio can have a different effects on the expected return and level of risk involved. We also wanted to highlight two portfolios that are important to note. Firstly, the portfolio with the highest Sharpe Ratio which is the highest risk-adjusted returns (Red Star). Secondly, the minimum variance portfolio, which is the portfolio with the lowest volatility (Green Star).

Discussion and Conclusions:

Our model recommended two portfolios with the maximum Sharpe Ratio (Portfolio A) and minimum volatility (Portfolio B). The estimated return, standard deviation, the Sharpe Ratio as well as the weights of the stocks are summarized below.

Portfolio	Portfolio A: Maximum Sharpe Ratio	Portfolio B: Minimum Volatility
Return	0.2577	0.2385
Standard Deviation	0.2039	0.1997
Sharpe Ratio	1.2637	1.1941
Weights of the Stocks		
Bank of America	0.51%	10.49%
Nike	40.93%	42.52%
Nvidia	14.45%	4.72%
Starbucks	44.11%	42.28%

The weights of the two portfolios are compared in the graph below. Interestingly, the asset distribution of the two portfolios is somewhat similar in that Starbucks and Nike comprise 85% of the portfolio. The difference in the two portfolios is whether to allocate the majority of the remaining 15% to Bank of America or Nvidia. To maximize the Sharpe Ratio, we should invest 14.45% and 0.51% of the investment to Nvidia and Bank of America, respectively. This process is the inverse for portfolio seeking to minimize volatility as this portfolio suggests that we invest 4.72% and 10.49% of the investment to Nvidia and Bank of America, respectively. It appears that Nvidia has a higher chance of introducing risk to the overall portfolio as Portfolio B only suggests that we invest less than 5% of the overall assets of this company. However, Portfolio A suggests that we invest almost all of the remaining 15% to Nvidia and invest only the remaining 0.51% to Bank of America. We suspect that Nvidia could be a risky investment but with a potential to yield a higher return-to-risk than Bank of America.

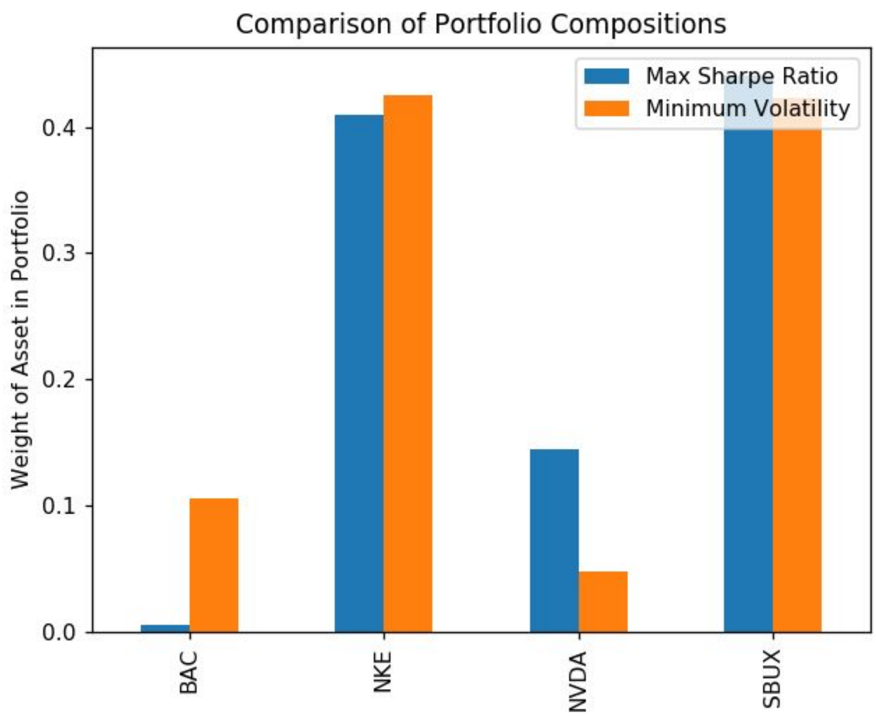

Figure 5

It is also important to note that the expected return and volatility of the overall portfolio are relatively close in both recommendations. Portfolio A has a higher return of 0.0192 and Portfolio B has a smaller standard deviation by 0.0042. As discussed previously, we sought to focus on the foundations of MPT and MVO. We utilized the standard deviation as Markowitz also did for the ease of calculation and because it should not deviate too much from other methodologies. However, we do acknowledge that limitations exist in our method as the standard deviation seeks to minimize the overall volatility and could have suggested asset distributions that sought to minimize upward deviation (gains) while trying to minimize downward deviation (losses). For future works, we suggest that a component of Rom and Ferguson's PMPT be applied and use the downward standard deviation for the volatility minimization analysis. As Portfolio A and Portfolio B have the inverse recommendations in allocating the remaining 15% to Bank of America and Nvidia, this detailed analysis of the losses could have helped decide which portfolio would yield the best returns and the least volatility.

Besides, our literature review showed that variations of risk measures could be applied such as the VaR and CVaR. VaR factors in time duration and attempts to evaluate the financial risk of a portfolio during a specific time frame. VaR looks at the worst case scenario during this time. However, CVaR looks at possible losses beyond the VaR and tries to find the average risk of extreme losses beyond VaR's estimation. Applying VaR or CVaR as the measure of risk could potentially be more meaningful in the interpretation of the actual risk.

As for the selection of portfolios, no correct option exists as the choice of the portfolio is subjective based on individual goals. If the goal is to maximize returns, the investor should choose Portfolio A that seeks to maximize Sharpe Ratio as it produces the highest return while taking on additional risk. For investors wanting to minimize risk, portfolio B should be chosen to minimize the volatility even if it means sacrificing potential expected returns. For our analysis, we choose the first portfolio since it has a higher Sharpe Ratio and thus indicates that diversification is stronger in this asset mix.

References:

Hargrave, Marshall. (2019). Sharpe Ratio Definition. *Investopedia.* Retrieved from https://www.investopedia.com/terms/s/sharperatio.asp.

Krokhmal, P., Uryasev, T., & Palmquist, J. (2001). Portfolio optimization with conditional value-at-risk objective and constraints. *The Journal of Risk,4*(2), 43-68. doi:10.21314/jor.2002.057

Konno, H., & Yamazaki, H. (1991). Mean-Absolute Deviation Portfolio Optimization Model and Its Applications to Tokyo Stock Market. *Management Science,37*(5), 519-531. Retrieved from https://www.jstor.org/stable/2632458.

Kaplan, P. D. (2017). From Markowitz 1.0 to Markowitz 2.0 with a Detour to Postmodern Portfolio Theory and Back. *The Journal of Investing,26*(1), 122-130. doi:10.3905/joi.2017.26.1.122

Marnix Engels. "." Math.leidenuniv.nl. 13 Jan. 2004. Web. 4 Mar. 2019. <https://www.math.leidenuniv.nl/scripties/Engels.pdf>

Appendix:

Code:

```
import numpy as np
import pandas as pd
import pandas_datareader.data as web
import matplotlib.pyplot as plt

#list of stocks in portfolio
stocks = ['BAC','NKE','NVDA','SBUX']

#download daily price data for each of the stocks in the portfolio
data = web.DataReader(stocks,data_source='yahoo',start='01/01/2010')['Adj Close']

data.sort_index(inplace=True)

#convert daily stock prices into daily returns
returns = data.pct_change()

#calculate mean daily return and covariance of daily returns
mean_daily_returns = returns.mean()
cov_matrix = returns.cov()

#set number of runs of random portfolio weights
num_portfolios = 25000

#set up array to hold results
#We have increased the size of the array to hold the weight values for each stock
results = np.zeros((4+len(stocks)-1,num_portfolios))

for i in range(num_portfolios):
    #select random weights for portfolio holdings
    weights = np.array(np.random.random(4))
    #rebalance weights to sum to 1
    weights /= np.sum(weights)

    #calculate portfolio return and volatility
    portfolio_return = np.sum(mean_daily_returns * weights) * 252
```

```
    portfolio_std_dev = np.sqrt(np.dot(weights.T,np.dot(cov_matrix, weights))) * np.sqrt(252)

    #store results in results array
    results[0,i] = portfolio_return
    results[1,i] = portfolio_std_dev
    #store Sharpe Ratio (return / volatility) - risk free rate element excluded for simplicity
    results[2,i] = results[0,i] / results[1,i]
    #iterate through the weight vector and add data to results array
    for j in range(len(weights)):
        results[j+3,i] = weights[j]

#convert results array to Pandas DataFrame
results_frame =
pd.DataFrame(results.T,columns=['ret','stdev','sharpe',stocks[0],stocks[1],stocks[2],stocks[3]])

#locate position of portfolio with highest Sharpe Ratio
max_sharpe_port = results_frame.iloc[results_frame['sharpe'].idxmax()]
#locate positon of portfolio with minimum standard deviation
min_vol_port = results_frame.iloc[results_frame['stdev'].idxmin()]

#create scatter plot coloured by Sharpe Ratio
plt.figure(figsize=(15,10))
plt.scatter(results_frame.stdev,results_frame.ret,c=results_frame.sharpe,cmap='RdYlBu')
plt.xlabel('Volatility', fontsize=16)
plt.ylabel('Returns', fontsize=16)
plt.colorbar()
#plot red star to highlight position of portfolio with highest Sharpe Ratio
plt.scatter(max_sharpe_port[1],max_sharpe_port[0],marker=(5,1,0),color='r',s=1000)
#plot green star to highlight position of minimum variance portfolio
plt.scatter(min_vol_port[1],min_vol_port[0],marker=(5,1,0),color='g',s=1000)
plt.show()

print(max_sharpe_port)

print(min_vol_port)

sharpelist = []
volatility = []
for i in range(7):
```

```
    sharpelist.append(max_sharpe_port[i])
    volatility.append(min_vol_port[i])

col = ['ret', 'stdev', 'sharpe', 'BAC', 'NKE', 'NVDA', 'SBUX']

sharpedf = pd.DataFrame(sharpelist, index = col, columns = ['Value'])
volatilitydf = pd.DataFrame(volatility, index = col, columns = ['Value'])
sharpedf = sharpedf.T
volatilitydf = volatilitydf.T
df = sharpedf.append(volatilitydf)

df.index = (['Max Sharpe Ratio', 'Minimum Volatility'])

df = df.T

df = df.drop(['ret', 'stdev', 'sharpe'])

plt.figure(figsize=(15,10))
df.plot.bar(title = 'Comparison of Portfolio Compositions')
plt.ylabel('Weight of Asset in Portfolio')
```