John Dennis, Sec 003

Mansi Pahade, Sec 002

Taimoor Sajid, Sec 003

12 December, 2024

Portfolio Optimization

The objective of this report is to optimize a portfolio under the Constant Expected Return (CER) Model. We will be using Modern Portfolio Theory, revolutionized by Harry Markowitz which showed that the best portfolios balanced high returns per unit of risk. We will first consider how one can minimize the variance in their portfolio without regard to expected return, and then how one can minimize variance with a desired return.

I. Attribution

Throughout this project, John and Mansi were the primary contributors to the MATLAB code for generating the plots and calculations. Taimoor contributed to the research for financial concepts and mathematical applications, as well as being the primary writer for the report.

II. Introduction

To address the complexities of managing large portfolios, we will utilize matrix algebra to simplify the calculations associated with portfolio optimization, expected returns, and variances. In the subsequent sections, we will define key concepts, present mathematical formulations, and apply computational methods to demonstrate the efficacy of our approach.

III. Mathematical Formulation

In statistics, a random variable is a variable whose outcomes follow an unknown pattern. They are usually represented by capital letters such as *X, Y, Z.* This is relevant to our project because the daily return on a stock can be treated as a random variable because of its unpredictability.

The expected value of a random variable is its long-term average. It is generally represented as E(X) or μ(X). For our project the expected value of the daily return will be an average of 255 days, which is the total number of trading days.

Variance is a measurement of the spread numbers in a data set and how far they are from the mean. In the investment world, it is used to help determine the volatility/riskiness, of an investment. The square root of variance is the standard deviation, which helps determine how consistent an investment's return is over time. To optimize our portfolio, we want to minimize variance, so we can lower risk while balancing returns.

Covariance measures the relationship between the returns on two assets. A positive covariance indicates that two variables trend in the same direction together. A negative covariance indicates that they have an inverse relationship.

The returns of an asset can be put into a random variable vector

$$\boldsymbol{R} = \begin{pmatrix} R_A \\ R_B \\ \vdots \\ R_n \end{pmatrix}$$

Therefore, the expected return of **R** is

$$E(\boldsymbol{R}) = \mu(\boldsymbol{R}) = \begin{pmatrix} \mu_A \\ \mu_B \\ \vdots \\ \mu_n \end{pmatrix}$$

The weights of each asset in a portfolio can be represented as a vector **x.**

$$\boldsymbol{x} = \begin{pmatrix} x_A \\ x_B \\ \vdots \\ x_n \end{pmatrix}$$

The weights of the portfolio must add up to one meaning

$$\boldsymbol{x}^T \boldsymbol{1} = \begin{pmatrix} x_A & x_B & \cdots & x_N \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix} = 1$$

The expected return of the portfolio is then given as the dot product between **x** and μ(**R**)

$$\mu_{p.x} = \boldsymbol{x^T} E[\boldsymbol{R}] = \begin{pmatrix} x_1 & x_2 & \cdots & x_n \end{pmatrix} \begin{pmatrix} \mu_A \\ \mu_B \\ \vdots \\ \mu_n \end{pmatrix} = x_1\mu_A + x_2\mu_B \dots x_n\mu_n \qquad (1)$$

Variance is calculated as the covariance of an asset with itself

$$\sigma_x^2 = var(x) = cov(x,x) = \sum_i^n \sum_j^n x_i \times x_j \times cov(R_i, R_j)$$

Where the covariance is

$$\sigma_{x,y} = cov_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

N = Number of data values, x_i = Data value of x, y_i = Data value of y, $\bar{x}$ = Mean of x data

$\bar{y}$ = Mean of y data.

With large portfolios, this calculation becomes tedious which is why matrix algebra is used to simplify the process. We can represent the covariance of the portfolio returns as a matrix Σ

$$\Sigma = \begin{pmatrix} Var(R) & Cov(R, R_2) & \cdots & Cov(R, R_n) \\ Cov(R_1, R_2) & Var(X_2) & \dots & Cov(R_2, R_n) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(R_1, R_n) & Cov(R_2, R_n) & \dots & Var(R_n) \end{pmatrix}$$

$$= \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \dots & \sigma_n^2 \end{pmatrix}$$

The variance of the portfolio is then calculated as the inner product of the weight vector with the covariance matrix.

$$\sigma^2 = \boldsymbol{x^T \Sigma x} = \begin{pmatrix} x_1 & x_2 & \dots & x_n \end{pmatrix} \begin{pmatrix} \sigma_1^2 & \sigma_{12} & \dots & \sigma_{1n} \\ \sigma_{12} & \sigma_2^2 & \dots & \sigma_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1n} & \sigma_{2n} & \dots & \sigma_n^2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \tag{2}$$

IV. Finding Global Minimum Portfolio

Let $\boldsymbol{m} = \begin{pmatrix} m_1 & m_2 & \dots & m_3 \end{pmatrix}$ be the minimum variance portfolio. To find a global minimum for the portfolio we want to solve the constrained problem

$$min_{m_1, m_2, \dots\, m_n}\ \sigma_m^2 = \boldsymbol{m^T}\Sigma\mathbf{m} \text{ s.t. } \boldsymbol{m^T}\mathbf{1} = 1$$

To do this, we will set up the Lagrangian function

$$L(\boldsymbol{m}, \lambda) = \boldsymbol{m^T}\Sigma\mathbf{m} + \lambda(\boldsymbol{m^T}\mathbf{1} - 1) \tag{3}$$

$$\frac{\partial L}{\partial \boldsymbol{m}} = 2\Sigma\boldsymbol{m} + \mathbf{1}\lambda = 0 \tag{4}$$

$$\frac{\partial L}{\partial \lambda} = \mathbf{1}^T\boldsymbol{m} - 1 = 0 \tag{5}$$

This system of equations can be represented in matrix form as

$$\begin{pmatrix} 2\Sigma & \mathbf{1} \\ \mathbf{1}^{\boldsymbol{T}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{m} \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix} \tag{7}$$

Since this system is in the form of

$$\boldsymbol{A}_{min}\boldsymbol{z}_{min} = \boldsymbol{b}$$

The solution to the system is simply

$$\boldsymbol{z}_{min} = \begin{pmatrix} \boldsymbol{m} \\ \lambda \end{pmatrix} = \boldsymbol{A}_{min}^{-1}\boldsymbol{b} \tag{8}$$

V. Minimum Variance

Now we want to create a portfolio with a desired return and then minimize the variance. Our constrained minimization problem becomes

$$min_x\, \sigma_x^2 = \boldsymbol{x}^{\boldsymbol{T}}\Sigma\boldsymbol{x}\ \ s.t.\ \mu_0 = \boldsymbol{x}^T\boldsymbol{\mu}, and\ \boldsymbol{x}^{\boldsymbol{T}}\mathbf{1} = 1$$

The Lagrangian becomes

$$L(x, \lambda_1, \lambda_2) = \boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{\Sigma x} + \lambda_1(\boldsymbol{x}^T\boldsymbol{\mu} - \mu_0) + \lambda_2(\boldsymbol{x}^T\mathbf{1} - 1) \tag{9}$$

$$\frac{\partial L}{\partial \boldsymbol{x}} = 2\boldsymbol{\Sigma x} + \lambda_1\boldsymbol{\mu} + \lambda_2 \tag{10}$$

$$\frac{\partial L}{\partial \lambda_1} = \boldsymbol{x}^T\boldsymbol{\mu} - \mu_0 = 0 \tag{11}$$

$$\frac{\partial L}{\partial \lambda_2} = \boldsymbol{x}^{\boldsymbol{T}}\mathbf{1} - 1 = 0 \tag{12}$$

The resulting matrix form for the system is

$$\begin{pmatrix} 2\Sigma & \boldsymbol{\mu} & \mathbf{1} \\ \boldsymbol{u}^{\boldsymbol{T}} & 0 & 0 \\ \mathbf{1}^{\boldsymbol{T}} & 0 & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mu_0 \\ 1 \end{pmatrix} \tag{13}$$

We will still use eq. 8 to solve for **x.**

The efficient frontier represents the set of optimal portfolios that offer the lowest risk for an expected return. Any portfolio that lies on the efficient frontier is considered efficient. The efficient frontier can be thought of as a convex combination between any two minimized portfolios. The plot of an efficient frontier is hyperbolic. Let **x** and **y** be two portfolios with different targeted returns but minimized variance. The convex combination of them will be

$$\boldsymbol{z} = \alpha\boldsymbol{x} + (1 - \alpha)\boldsymbol{y}$$

VI. Example and Numerical Results

We decided to create a mock portfolio to test out this model. We will create a five-stock portfolio comprised of Taiwan Semiconductors (TSM), Nvidia (NVDA), Home Depot (HD), Wells Fargo (WFC), and Coca-Cola(KO). We are looking at the 2018 market assuming we have one share of each. Using data from Yahoo finance,

Ticker	Expected Return (μ)	Weight (x)
TSM	-0.11	0.125
NVDA	-0.403	0.016
HD	-0.1	0.55
WFC	-0.245	0.18
KO	0.03	0.13

(Table 1)

With a corresponding covariance matrix

$$\Sigma = \begin{pmatrix} 3.61E-04 & 3.68E-04 & 7.04E-05 & 9.12E-05 & 1.38E-06 \\ 3.68E-04 & 8.19E-04 & 8.62E-05 & 1.12E-04 & 6.12E-07 \\ 7.04E-05 & 8.62E-05 & 1.86E-04 & 1.02E-04 & 2.85E-05 \\ 9.12E-05 & 1.12E-04 & 1.02E-04 & 3.08E-04 & 2.19E-05 \\ 1.38E-06 & 6.12E-07 & 2.854E-05 & 2.19E-05 & 7.13E-05 \end{pmatrix}$$

Solving the global minimization problem (eq. 3), our optimal weights $\boldsymbol{x}_0$ for the global minimum are

$$\boldsymbol{x}_0 = \begin{pmatrix} 0.1218 \\ -0.0075 \\ 0.1139 \\ 0.0586 \\ 0.7132 \end{pmatrix}$$

Interestingly we see that our optimal weight for NVDA is negative. A negative weight can be considered as holding a short position on a stock. Shorting is an investment technique where an investor looks to profit from an asset's decline in price.

Plugging $\boldsymbol{x}_0$ into eq. 1 and eq. 2, we get an expected return of -1.4%, and a variance of 5.5E-05. A negative expected return on the global minimum variance indicates that although we have the lowest possible risk, our portfolio is expected to lose value over time. Since the global minimum variance is solely focused on minimizing risk without regard to returns, this can lead to a portfolio with very low or negative expected returns depending on how the asset performs. 2018 was an especially bad year for the stock market which is reflected in our expected return.

For the experiment's sake, we will say that we want a rather aggressive return of 10% for the year. We now want to solve the minimization problem (eq.9), with $\mu_0 = 0.1$. Solving this leads to the optimal weights ($\boldsymbol{x}_1$)

$$\boldsymbol{x}_1 = \begin{pmatrix} 0.282 \\ -0.1823 \\ 0.075 \\ -0.143 \\ 0.9710 \end{pmatrix}$$

Our variance using eq 2, is 8E-05, which is higher than the global minimum variance which means a 10% return is riskier.

Using the global minimum variance portfolio, and the 10% return portfolio, we can generate the efficient frontier (figure 1).

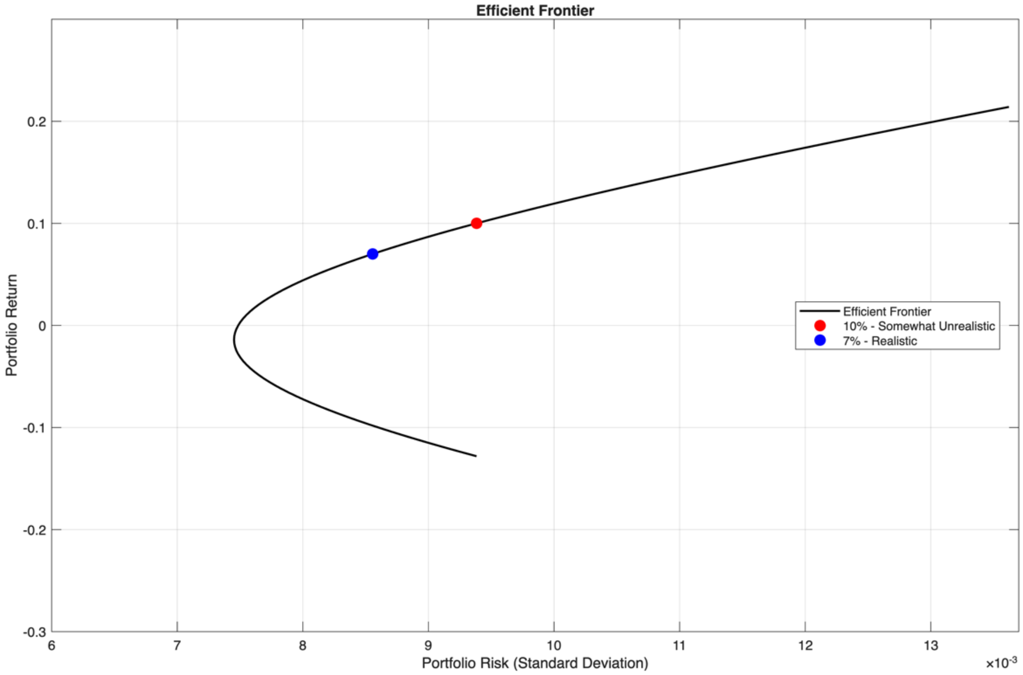


Figure 1

This graph shows that any portfolio that lies on that hyperbolic curve is considered an efficient portfolio. Any portfolio above the curve is not possible, and any portfolio beneath the curve is inefficient.

VII. Discussion, Conclusion, and Interesting Findings

To further test our results, we looked at the returns from 2023, which was our back-testing period. For 2023, our portfolio's performance varied across assets. Taiwan Semiconductors (TSM) had a high variance of 0.0853 and a negative expected return of -9.58%, indicating a large risk with a loss. Nvidia (NVDA) showed a high variance of 0.0853 with a strong positive return of 24.94%, suggesting a high-risk but high-reward asset. Home Depot (HD) had a low variance of 0.0161 and a slightly negative return of -0.72%, representing a low-risk, low-reward investment. Wells Fargo (WFC) had a moderate variance of 0.0399 and a positive return of 2.37%, reflecting a good balance between risk and return. Coca-Cola (KO) had the highest variance at 0.1297 with a return of 6.06%, indicating a high risk but moderate reward.

Our model used data from 2018 which was a rather bearish market for the year. In contrast, the 2023 market was more bullish, with the S&P500, a common metric for the state of the American market, climbing up 24.31%. The fact that there was such a difference in the back-testing data from 2023 versus the optimizing data from 2018 highlights a limit on our model. It fails to capture long-term bullish trends. This comparison shows the need to adapt portfolio strategies that align with changing market conditions and economic trends.

Our project successfully optimized a portfolio by minimizing variance using the CER model. However, the biggest drawback to using the CER model is that it assumes that every asset has a constant expected return over time. The market is highly dynamic, and therefore, an asset is not likely to have constant returns. There are model limitations based on the data that's used to optimize. Models optimized for a bear market like 2018 may fail to capture opportunities in bullish or recovery phases, as they emphasize minimizing losses over maximizing returns. On the other hand, testing this model in 2023, a year with positive returns and high growth, might result in underperformance or missed opportunities. Our minimization problem also assumes that our portfolio is comprised solely of assets that have some level of risk. Assets such as bonds which are considered risk free would change the minimization equations. To further improve the model, we can introduce more constraints such as transaction costs, taxes, and minimum investment thresholds.

Principal Component Analysis (PCA) is another tool that we can further investigate to get a deeper understanding of portfolio optimization. This is because it pulls out the dominant factors that contribute the most variance to the portfolio. We took the initial steps in finding the principal components of our mock portfolio. We first found the eigenvectors of our covariance matrix.

$$\Sigma = \begin{pmatrix} 3.61E-04 & 3.68E-04 & 7.04E-05 & 9.12E-05 & 1.38E-06 \\ 3.68E-04 & 8.19E-04 & 8.62E-05 & 1.12E-04 & 6.12E-07 \\ 7.04E-05 & 8.62E-05 & 1.86E-04 & 1.02E-04 & 2.85E-05 \\ 9.12E-05 & 1.12E-04 & 1.02E-04 & 3.08E-04 & 2.19E-05 \\ 1.38E-06 & 6.12E-07 & 2.854E-05 & 2.19E-05 & 7.13E-05 \end{pmatrix}$$

Which gave us the eigenvectors (in descending order):

$$\begin{pmatrix} 0.48 \\ 0.84 \\ 0.14 \\ 0.2 \\ 0.01 \end{pmatrix}, \begin{pmatrix} 0.03 \\ -0.3 \\ 0.45 \\ 0.83 \\ 0.12 \end{pmatrix}, \begin{pmatrix} -0.9 \\ 0.44 \\ -.01 \\ 0.24 \\ 0.02 \end{pmatrix}, \begin{pmatrix} 0.19 \\ -0.07 \\ -0.84 \\ 0.45 \\ -0.23 \end{pmatrix}, \begin{pmatrix} 0.05 \\ 0.003 \\ 0.255 \\ -0.0007 \\ 0.96 \end{pmatrix}$$

The principal components plotted gives us figure 2.

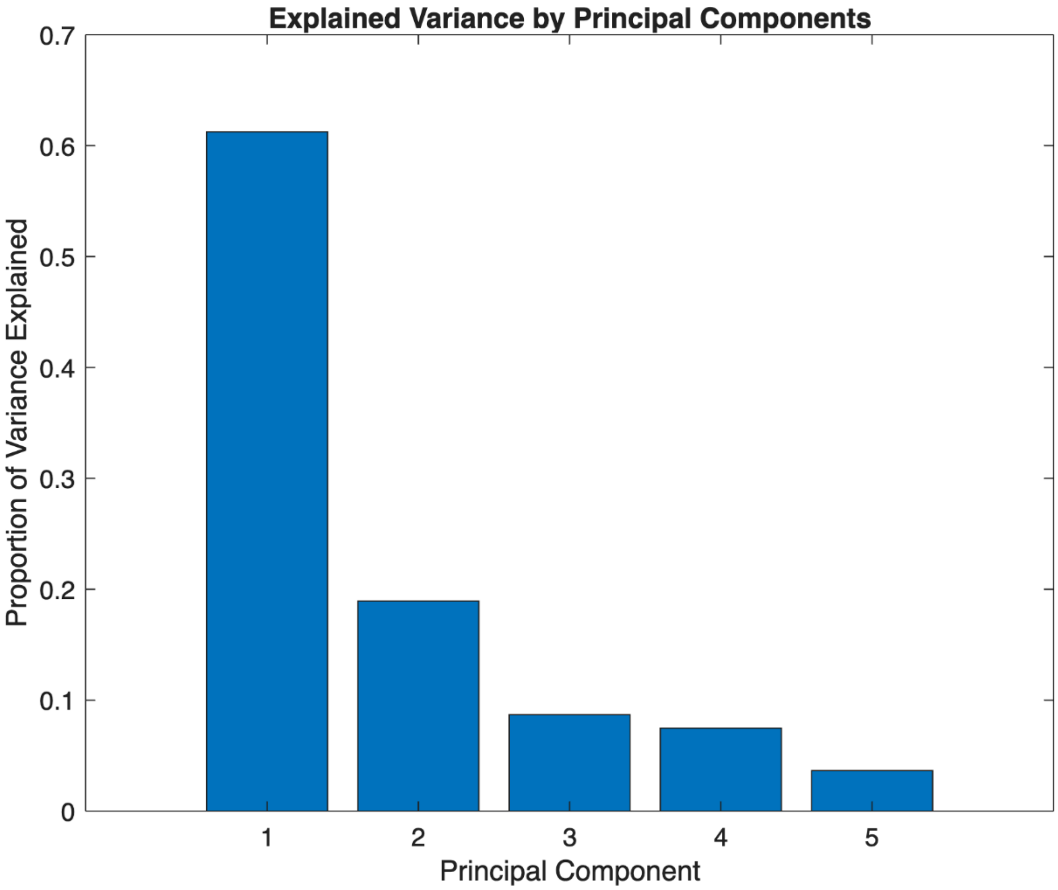


Figure 2

Figure 2 shows that 3 components, or assets account for close to 90% of the variance in our portfolio. This means that we can reduce the size/dimension of our portfolio, while still maintaining a good risk to return ratio. However, further analysis will be needed to determine which assets specifically contribute to the most variance.

References

[1] *Chapter 1 Portfolio Theory with Matrix Algebra*, University of Washington, 7 Aug. 2013, faculty.washington.edu/ezivot/econ424/portfolioTheoryMatrix.pdf.

[2] *The Coca-Cola Company (KO) Stock Historical Prices & Data - Yahoo Finance*, finance.yahoo.com/quote/KO/history. Accessed 12 Dec. 2024.

[3] Frazee, Gretchen. “6 Factors That Fueled the Stock Market Dive in 2018.” *PBS*, Public Broadcasting Service, 27 Dec. 2018, www.pbs.org/newshour/economy/making-sense/6-factors-that-fueled-the-stock-market-dive-in-2018.

[4] Hayes, Adam. “Covariance: Definition, Formula, Types, and Examples.” *Investopedia*, Investopedia, 31 Aug. 2024, www.investopedia.com/terms/c/covariance.asp.

[5] Hayes, Adam. “Short Selling: Your Step-by-Step Guide for Shorting Stocks.” *Investopedia*, Investopedia, www.investopedia.com/terms/s/shortselling.asp. Accessed 11 Dec. 2024.

[6] Hayes, Adam. “What Is Variance in Statistics? Definition, Formula, and Example.” *Investopedia*, Investopedia, 20 Sept. 2024, www.investopedia.com/terms/v/variance.asp.

[7] *The Home Depot, Inc. (HD) Stock Historical Prices & Data - Yahoo Finance*, finance.yahoo.com/quote/HD/history/. Accessed 12 Dec. 2024.

[8] *Nvidia Corporation (NVDA) Stock Historical Prices & Data - Yahoo Finance*, finance.yahoo.com/quote/NVDA/history/. Accessed 12 Dec. 2024.

[9] *Portfolios - Principal Component Analysis - Reduce Number of Assets Based on Risk Contribution*, YouTube, www.youtube.com/watch?v=HSh-96fF6Bo. Accessed 11 Dec. 2024.

[10] “Principal Component Analysis (PCA) Explained.” *Built In*, builtin.com/data-science/step-step-explanation-principal-component-analysis. Accessed 11 Dec. 2024.

[11] *Taiwan Semiconductor Manufacturing Company Limited (TSM) Stock Historical Prices & Data - Yahoo Finance*, finance.yahoo.com/quote/TSM/history/. Accessed 12 Dec. 2024.

[12] *Wells Fargo & Company (WFC) Stock Historical Prices & Data - Yahoo Finance*, finance.yahoo.com/quote/WFC/history/. Accessed 12 Dec. 2024.

[13] Written by Bahr, Kevin. “2023 Global Stock Market Review.” *University of Wisconsin - Stevens Point*, blog.uwsp.edu/cps/2024/01/04/2023-global-stock-market-review/. Accessed 11 Dec. 2024.

[14] Wyss-Gallifent, Justin. *Portfolio Optimization*, University of Maryland, www.math.umd.edu/~immortal/MATH401/book/ch_portfolio.pdf. Accessed 11 Dec. 2024.

Appendix

Table of Contents

```
clc; clear close all; %Optimizing For Jan 3 2018 -
Jan 2019 (Dec 29 2018)
```

Creating File names

```
file1 = "Mktdata_TSM.xlsx";
file2 = "Mktdata_NVDA.xlsx";
file3 = "Mktdata_HD.xlsx";
file4 = "Mktdata_WFC.xlsx";
file5 = "Mktdata_KO.xlsx";
% These files are for back-testing
fileTSM_2018 = "TSM_2018.xlsx";
fileNVDA_2018 = "NVDA_2018.xlsx";
fileHD_2018 = "HD_2018.xlsx";
fileWFC_2018 = "WFC_2018.xlsx";
fileKO_2018 = "KO_2018.xlsx";
%creating vectors for use in functions files = [file1 file2 file3 file4
file5]; files2 = [fileTSM_2018 fileNVDA_2018 fileHD_2018 fileWFC_2018
fileKO_2018];
```

Setting up the portfolio with basic elements such as tickers, intial opening values, the RoR, weight, and the average close

```
stock1 = struct('name', 'TSM', 'value_i', 0, 'mu', 0, 'weight_i',
0.1205297629, 'mean_close', 0, "wt_opt", 0); %Taiwan Semiconductors stock2 =
struct('name', 'NVDA', 'value_i', 0, 'mu', 0, 'weight_i',
0.0162000811, 'mean_close', 0, "wt_opt", 0); %Nvidia
stock3 = struct('name', 'HD', 'value_i', 0, 'mu', 0, 'weight_i', 0.553324059,
```

'mean_close', 0, "wt_opt", 0); %Home Depot

```
stock4 = struct('name', 'WFC', 'value_i', 0, 'mu', 0, 'weight_i',
0.1768967716, 'mean_close', 0, "wt_opt", 0); % Wells Fargo and Co.
stock5 = struct('name', 'KO', 'value_i', 0, 'mu', 0, 'weight_i',
0.1330493248, 'mean_close', 0, "wt_opt", 0); %Coca-cola
```

%These functions simply read in historical market data and retrieve certain
%values such as intial stock value at the beginning of the year, the mean
%closing value for each asset, and it's RoR

```
[stock1.value_i, stock1.mean_close, stock1.mu] = getValues(fileTSM_2018);
[stock2.value_i, stock2.mean_close, stock2.mu] = getValues(fileNVDA_2018);
[stock3.value_i, stock3.mean_close, stock3.mu] = getValues(fileHD_2018);
[stock4.value_i, stock4.mean_close, stock4.mu] = getValues(fileWFC_2018);
[stock5.value_i, stock5.mean_close, stock5.mu] = getValues(fileKO_2018);
```

Calculating the Covariance Matrix, weight vector, portfolio expected return, and portfolio var-

iance

```
mu = [stock1.mu stock2.mu stock3.mu stock4.mu stock5.mu]';
%Generating the expected returns vector containing each asset's
expected return [sigma, returns] = getCov(files); %Calculating the
covariance matrix
weights = [stock1.weight_i stock2.weight_i stock3.weight_i stock4.weight_i
stock5.weight_i]';%Creating a weight vector to contain all the weights of the
assets muPortfolio = weights' * mu;%Expected returns of the Portfolio
muPortfolio_annual = (1 + muPortfolio / 100)^252 - 1;%Annualizing it for 252
days, cumulative Returns variancePortfolio = weights' * sigma *
weights;%Variance of the portfolio
```

Global Minimum Variance

```
w_GMV = GlobalOptimal(sigma, mu);% Calculating the weights for the
Global Minimum Variance
mu_GMV = dot(w_GMV,mu);%Expected return from the GMV
var_GMV = w_GMV' * sigma * w_GMV;%The value of the GMV
```

Solving Constrained optimization problems for minimizing variance with target return

```
mu_target_1 = 0.10;% Target for 10% return
mu_target_2 = .07;% Target for 7% return
X = minVar_TargetReturn(mu_target_1, mu, sigma); % The optimal weights for
10% return
w_target1 = minVar_TargetReturn(mu_target_1, mu, sigma);
mu_target1 = mu' * w_target1; % return of X/w_target1, they're the same
var_target1 = w_target1' * sigma * w_target1; % variance of X
Y = minVar_TargetReturn(mu_target_2, mu, sigma); % Optimal weights for target
2
w_target2 = minVar_TargetReturn(mu_target_2, mu, sigma);
mu_target2 = mu' * w_target2; % return of Y
```

```
var_target2 = w_target2' * sigma * w_target2; % variance of Y
stock1.wt_opt = X(1);
stock2.wt_opt = X(2);
stock3.wt_opt = X(3);
stock4.wt_opt = X(4);
stock5.wt_opt = X(5);
```

Portfolio

```
Portfolio = [stock1 stock2 stock3 stock4 stock5];% Setting up the portfolio
```

Efficient Frontier

alpha = linspace(-1,2,100);%Creating spacing for later use

```
mu_frontier = zeros(size(alpha));
sigma_frontier = zeros(size(alpha));

% z matrix for i =
1:length(alpha)
    w = (alpha(i) * w_GMV) + ((1 - alpha(i)) * X);
    mu_frontier(i) = mu' * w;
    sigma_frontier(i) = sqrt(w' * sigma * w);
end
```

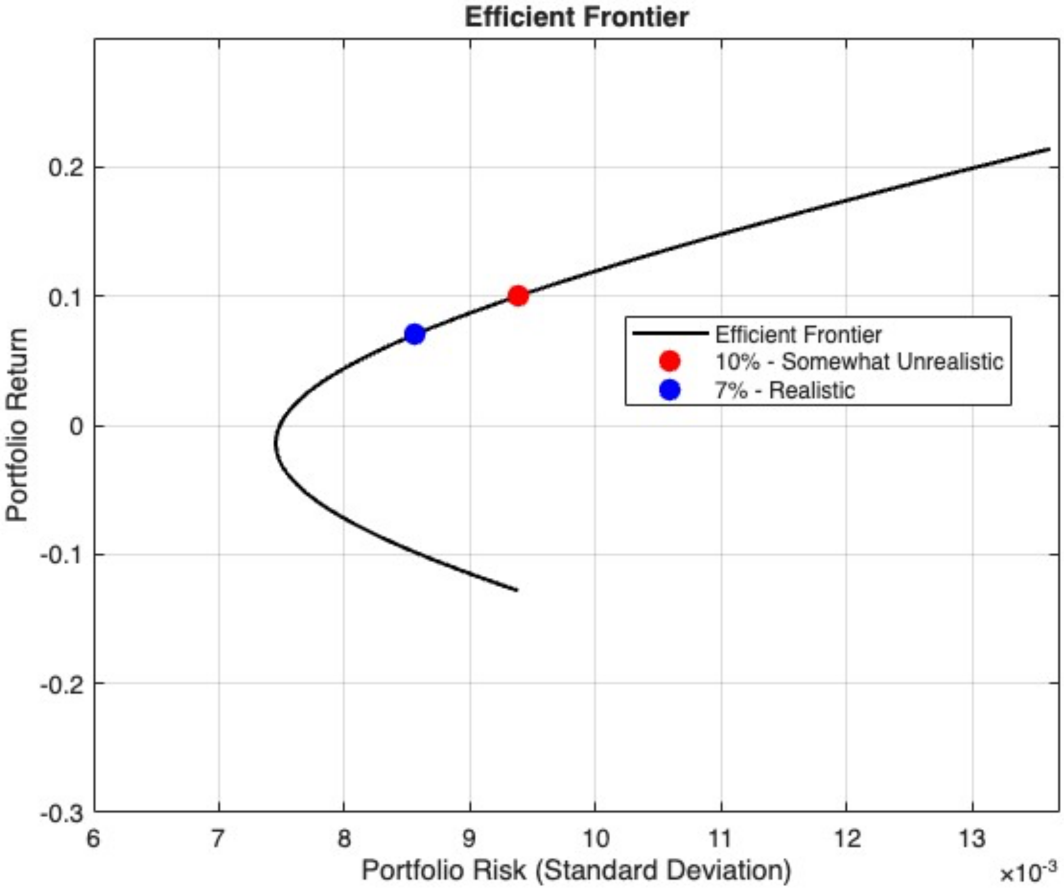
% plotting the efficient frontier

```
figure;
plot(sigma_frontier, mu_frontier, 'k-', 'LineWidth', 1.5);
hold on;
plot(sqrt(var_target1), mu_target1, 'ro', 'MarkerSize', 8, 'MarkerFaceColor',
'r'); %Plotting the 10% portfolio
plot(sqrt(var_target2), mu_target2, 'bo', 'MarkerSize', 8, 'MarkerFaceColor',
'b'); %Plotting the 7% portfolio %Formatting
```

xlabel('Portfolio Risk (Standard Deviation)') ylabel('Portfolio Return') title('Efficient Frontier') legend('Efficient Frontier', '10% - Somewhat Unrealistic', ... '7% - Realistic', 'Location', 'Best');

```
ylim([-0.3, 0.3]);
xlim([0.006, .0137]);
grid on;
```

```
%print('Efficient_Frontier', '-dpng', '-r300') hold off;
```

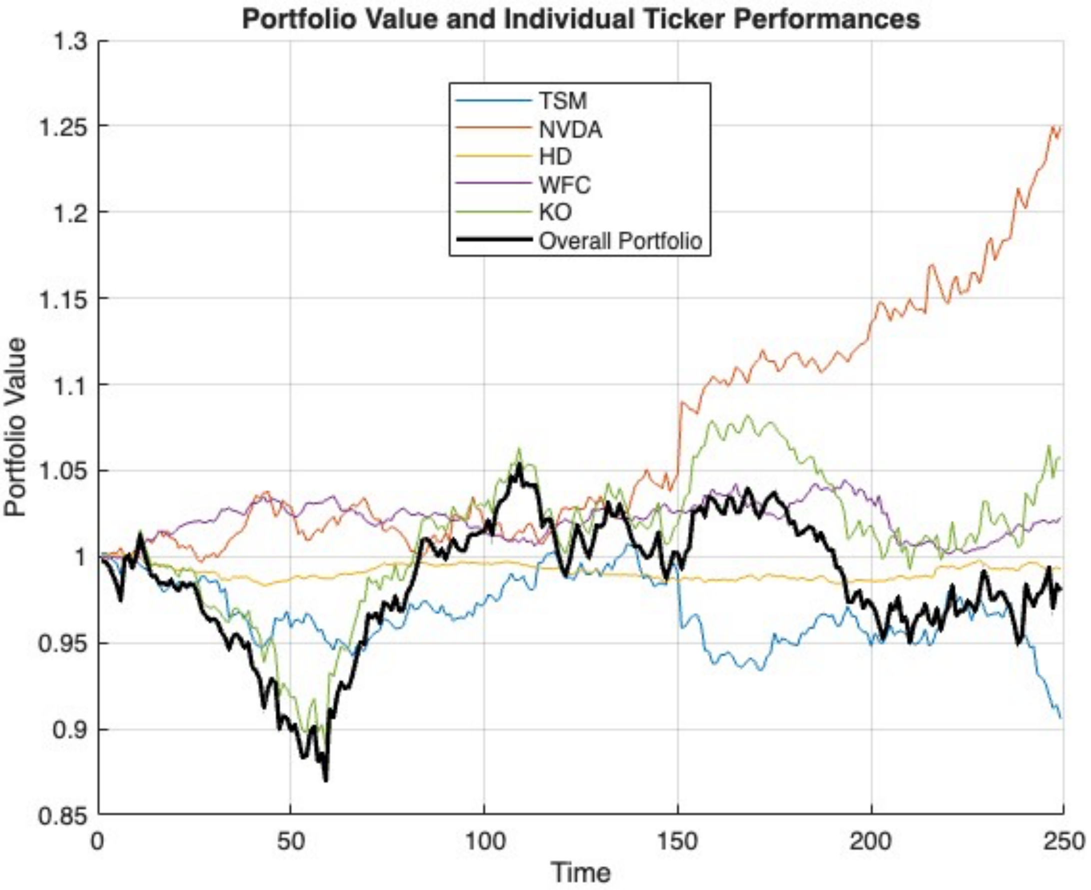


Back-Test for the year 2023

```
Rebalance_Freq = 252; %daily
[portfolio_value_2023, cumulative_return_2023, portfolio_variance_2023] =
Backtest(files, X, Rebalance_Freq);% Back testing
```

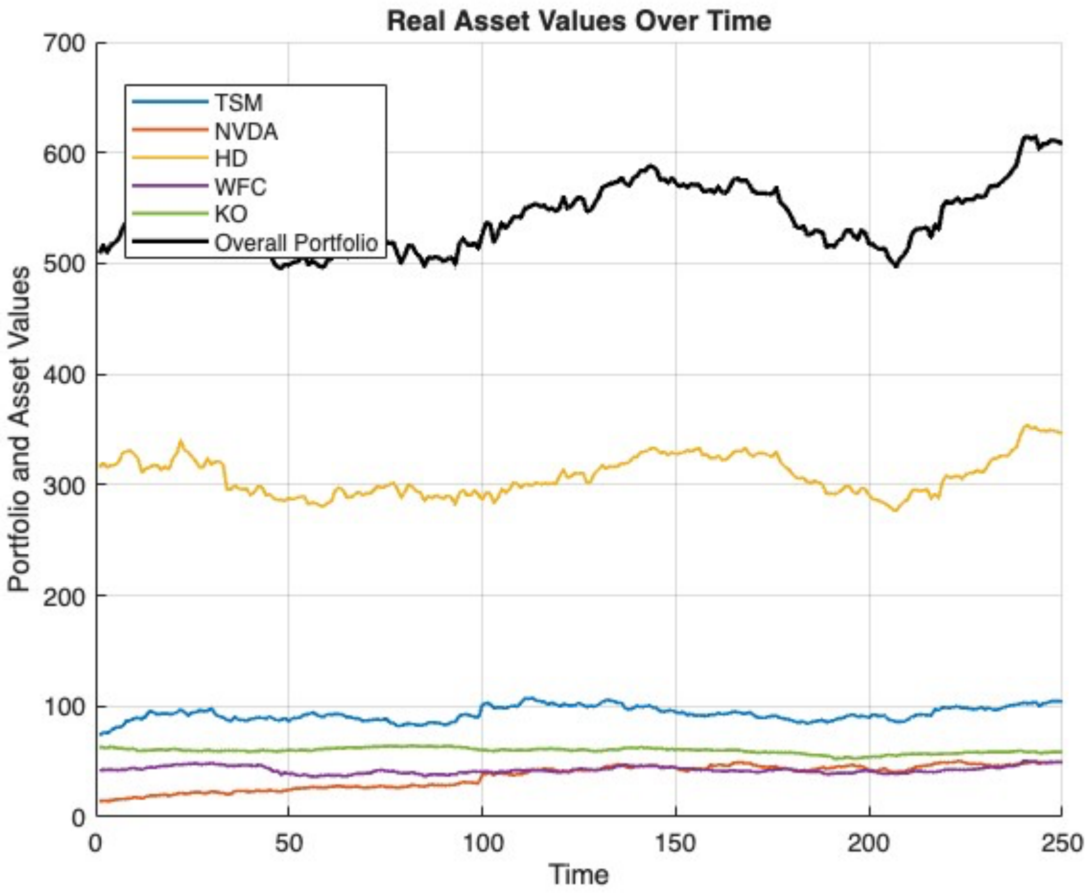
Visualizing Results

```
dispPortfolioValues(portfolio_value_2023, X);
```



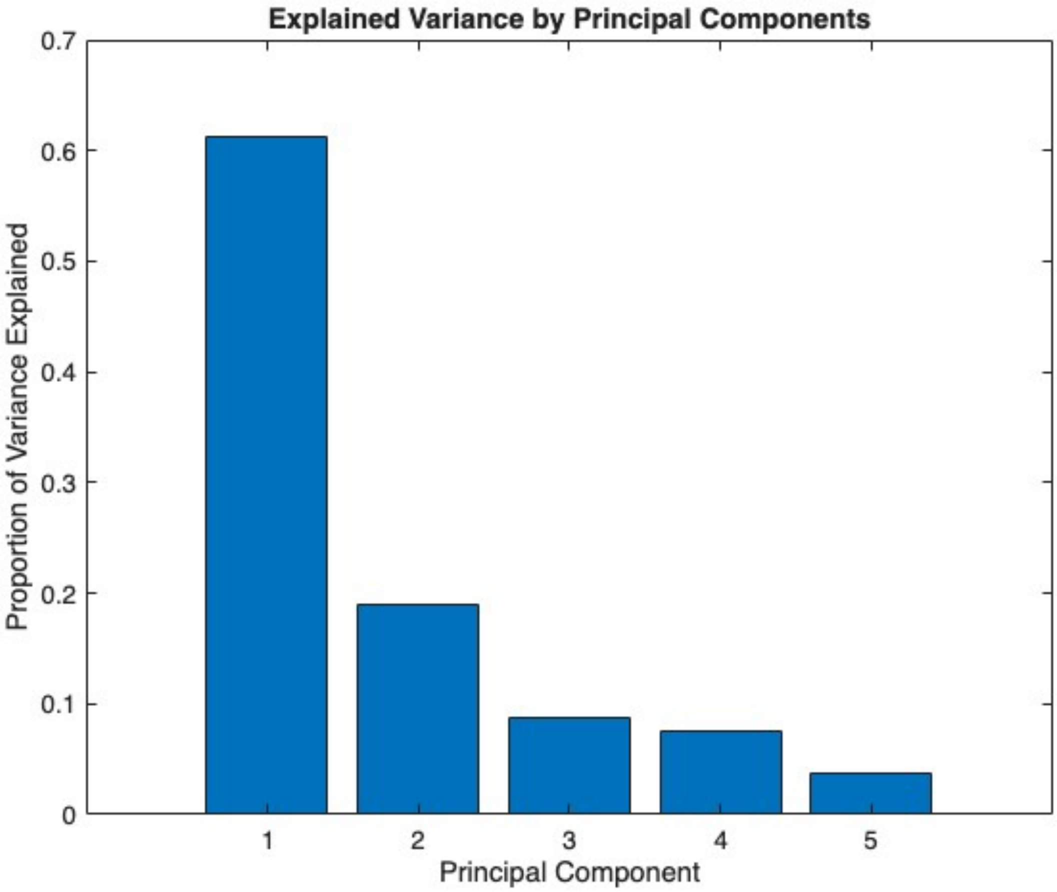
Displaying Real Market Values for 2023

```
dispMktValues(files);
```



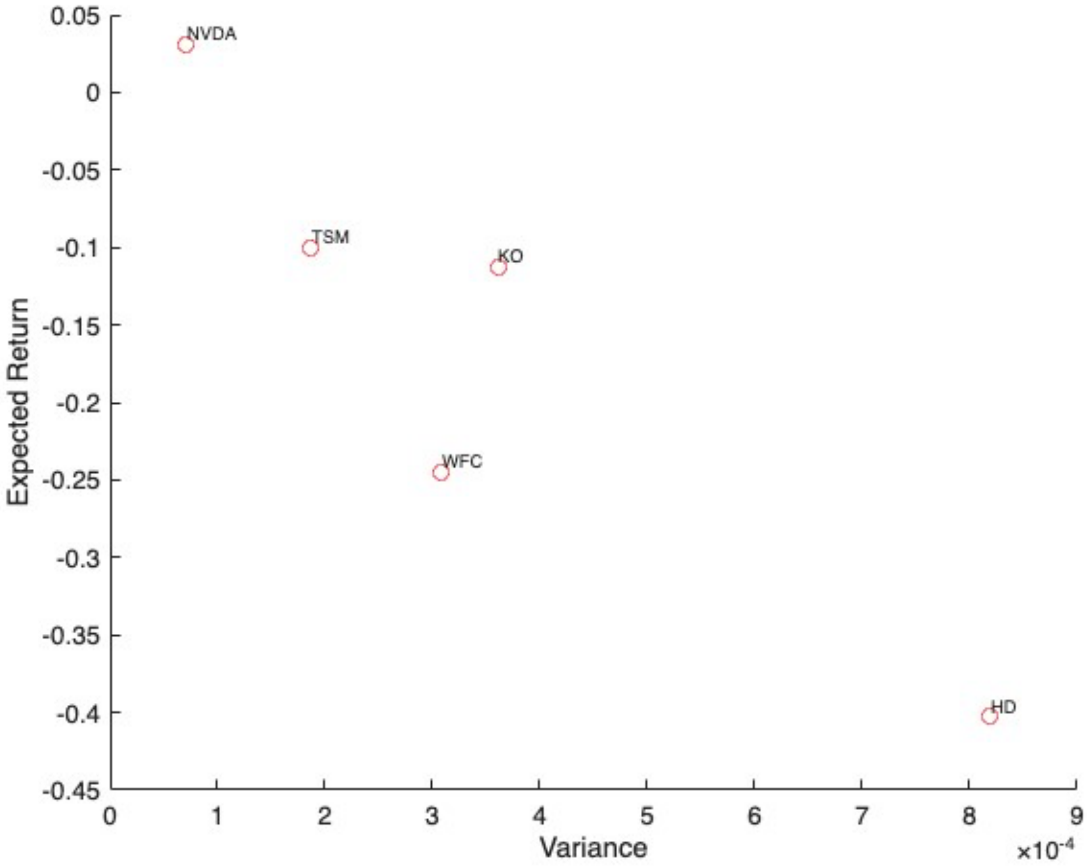
PCA of variance for the problem

```
[eVecs, eVecs_raw, D, Percent_Impact, Feature, Var] = getPCA(sigma);
```



Scatter Plot of Returns vs Variance

```
dispMuVar(mu,Var);
```



Functions

```
function [open_i, mean_close, RoR] = getValues(filename) data =

readmatrix(filename);%Read in the Files open_i = data(end, 2);%The opening

value at the beginning of the year fin_close = data(1,5);%The closing value

at the end of the year mean_close = mean(data(:,5));%The average closing

values throughout the year RoR = (fin_close-open_i)/open_i;%THe rate of

return, expected return, or mu end

function [sigma, returns] = getCov(files)
```

%Creating the variables containing the files

```
f1 = files(1);
f2 = files(2);
f3 = files(3);
```

```
f4 = files(4);
f5 = files(5);
%reading the files
data1 = readmatrix(f1);
data2 = readmatrix(f2);
data3 = readmatrix(f3);
data4 = readmatrix(f4);
data5 = readmatrix(f5);
%Gathering the closing values
close_asset1 = data1(:,5);
close_asset2 = data2(:,5);
close_asset3 = data3(:,5);
close_asset4 = data4(:,5);
close_asset5 = data5(:,5);
```

```
%Creating a vector of all the closing values
```

```
tot_close = [close_asset1 close_asset2 close_asset3 close_asset4
close_asset5];
%Calculating Percentage Returns
returns = diff(tot_close) ./ tot_close(1:end-1, :);
%Creating the covariance matrix with the returns

sigma = cov(returns); end

function w_optimal = minVar_TargetReturn(mu_target, mu, sigma)
n = length(mu); % Number of assets
A = [2*sigma, mu, ones(n,1); mu', 0, 0; ones(1,n), 0, 0];%Creating the matrix
to solve the system of equations
b = [zeros(n,1); mu_target; 1];%Defining the solution
```

```
solution = A \ b;%Calculating the solution w_optimal = solution(1:n); %
Optimal weights exlcuding the lagrange multipliers end
```

```
function [portfolio_values, cumulative_returns, portfolio_variance] =
Backtest(files, wt_opt, Rebalance_Freq)
```

```
% Backtesting function to compute portfolio values, cumulative
returns, and
```

```
volatilities % for
multiple tickers.
```

```
% Initialize output variables
num_files = length(files);
```

```
portfolio_values = cell(num_files, 1);     %For the overall portfolio values
cumulative_returns = zeros(num_files, 1); %For the cumulative returns
portfolio_variance = zeros(num_files, 1); %Storing the volatilites/variances
of each asset
```

```
% Loop through each file and perform calculations
for i = 1:num_files

  %Read data

    data = readmatrix(files(i));

    %Grabs the closing
closing_prices = data(:, 5);
returns_backtest =
diff(log(closing_prices)); %
Logarithmic returns

  % Computing portfolio returns for the asset considering its weight
portfolio_returns = returns_backtest * wt_opt(i);

    % Computing the cumulative portfolio value in total
portfolio_value = cumprod(1 + portfolio_returns);

  %Storing the values for later use

    portfolio_values{i} = portfolio_value; % Store the cumulative value series
cumulative_returns(i) = portfolio_value(end) / portfolio_value(1) - 1; %
Cumulative return
    portfolio_variance(i) = std(portfolio_returns) * sqrt(Rebalance_Freq); %
Annualized variance

end end

function dispPortfolioValues(portfolio_values, wt_opt)% Helps to visualize

results of the portfolio_values

figure;
hold on;

for i = 1:length(portfolio_values)%Goes through the cells to plot it

    plot(portfolio_values{i});
end

% This calculates overall portfolio value

num_assets = length(portfolio_values); %used for the for loop%
num_days = length(portfolio_values{1});%Used for length
overall_portfolio_value = zeros(num_days, 1);

for i = 1:num_assets
    overall_portfolio_value = overall_portfolio_value + wt_opt(i) *
portfolio_values{i};%Calculating the value of the portfolio end
```

```
plot(overall_portfolio_value, 'k-', 'LineWidth', 1.75); % Black line for
overall portfolio

% Formatting

title('Portfolio Value and Individual Ticker Performances'); xlabel('Time');
ylabel('Portfolio Value'); legend({'TSM', 'NVDA', 'HD', 'WFC', 'KO', 'Overall
Portfolio'}, 'Location',
'Best');
grid on;

%print('MktValues_OptWts', '-dpng', '-r300')
end


function dispMktValues(files)
    % read data from files
data1 = readmatrix(files(1));
data2 = readmatrix(files(2));
data3 = readmatrix(files(3));
data4 = readmatrix(files(4));
data5 = readmatrix(files(5));

    % Getting the close price data
close_asset1 = flipud(data1(:, 5));
close_asset2 = flipud(data2(:, 5));
close_asset3 = flipud(data3(:, 5));
close_asset4 = flipud(data4(:, 5));
close_asset5 = flipud(data5(:, 5));

  % Making it into a vector

    portfolio_values = [close_asset1, close_asset2, close_asset3,
close_asset4, close_asset5];

    % Calculating the portfolio value over all the values
overall_portfolio_value = sum(portfolio_values, 2);

  %Plotting the values of each asset and then putting the black line for
%the overall portfolio value

    figure;
hold on;
    plot(close_asset1, 'LineWidth', 1.25);
plot(close_asset2, 'LineWidth', 1.25);
plot(close_asset3, 'LineWidth', 1.25);
plot(close_asset4, 'LineWidth', 1.25);
plot(close_asset5, 'LineWidth', 1.25);
```

```
    plot(overall_portfolio_value, 'k-', 'LineWidth', 1.75); % Black line for
overall portfolio value

    %More Formatting
    title('Real Asset Values Over Time');
    xlabel('Time');

  ylabel('Portfolio and Asset Values');    legend('TSM', 'NVDA', 'HD', 'WFC',
'KO', 'Overall Portfolio', 'Location',

'Best');
grid on;
hold off;

  print('MktValues', '-dpng', '-r300') end

function [eVecs, eVecs_raw, D, Percent_Impact, Feature, Var] = getPCA(sigma)
    % eigenvalues and eigenvectors
    [eVecs_raw, e_raw] = eig(sigma);
    Var = diag(sigma); % Variances of each asset

    %Getting and thensort eigenvalues in descending order
    [D, idx] = sort(diag(e_raw), 'descend');
    eVecs = eVecs_raw(:, idx); % Reordering the eigenvectors to match sorted
eigenvalues

  % Figuring out the percentage of impact

    sumD = sum(D);
    Percent_Impact = D / sumD;
    % Plotting
figure;
bar(Percent_Impact);
%Some more formatting

  title('Explained Variance by Principal Components');    xlabel('Principal
Component');

    ylabel('Proportion of Variance Explained');
%print('PCA_PLOT', '-dpng', '-r300');

  % Creating the feature matrix

    Feature = eVecs(:, 1:3);

end function

dispMuVar(mu,Var)
```

```
names = ["KO", "HD", "TSM", "WFC", "NVDA"];
```

```
figure;
scatter(Var, mu, 'red');
xlabel("Variance");
ylabel("Expected Return")
```

```
text(Var, mu, names, 'Vert','bottom', 'Horiz','left', 'FontSize',7)
```

```
%print('Scatter', '-dpng', '-r300') end
```

Published with MATLAB® R2024b