Analysing and Assessing Financial Models of Investments Using Linear Algebra

1 Introduction

Matrix algebra provides a powerful framework for effectively representing and analyzing various financial concepts and models. This project aims to utilize the tools and techniques of matrix algebra to gain valuable insights into the realm of investments.
By utilizing matrix operations such as multiplication, inversion, and eigenvalue analysis, the project seeks to elaborate on the essential aspects of finance, including portfolio optimization, risk assessment, asset pricing, and financial forecasting.
In the concept of Risk Assessment, an integral part of financial analysis, risk assessment can be enhanced through matrix algebra. By representing risk factors and their correlations in matrix form, it becomes possible to assess the overall risk exposure of a portfolio. In this project, we describe the usage of matrices and the subsequent linear algebra concepts for risk assessment:

- Factor Exposures Matrices
- Covariance and Correlation Matrices
- Eigenvalues and Eigenvectors

In the concept of Financial Forecasting, it explores two main applications of linear algebra in the field of financial forecasting. Essentially, financial forecasting is the study of predicting the future of a financial model (be it a stock, or a company) based on previous data. Here, we will be primarily focusing on stock data and will emphasize on the usage of matrices, vectors and linear regression in financial forecasting. In the concept of Portfolio optimization, it refers to the process of strategically allocating investments within a portfolio to achieve an optimal balance between risk and return. The primary objective of portfolio optimization is to maximize returns while minimizing the level of risk, considering an investor's specific objectives and constraints.
In this project we can explores the application of linear algebra and it tools to Optimize the portfolio. We will look into these objectives to show that how the Linear Algebra used in the concept.

- Portfolio Weights

- Mean-Variance Optimization
- Matrix Operations(Multiplication and Inverse)

2 Related Works

- "Risk-Based Portfolios with Large Dynamic Covariance Matrices" (2018) by Dr. Kei Nakagawa, Mr. Mitsuyoshi Imamura, Kenichi Yoshida
- "Stock Price Prediction Using Regression Analysis" (2016) by Dr. Md. Uzzal Hossain, Md. Shakhawat Hossain, and Md. Monirul Islam.

3 Problem Statement

We aim at illustrating the use of various linear algebra concepts, primarily matrices, in analyzing and assessing financial models of investments. These concepts include matrices and their operations, eigenvalues and eigenvectors, linear regression, etc.

4 Target

The aim of this project is to:

Enhance Financial Decision-Making: The project aims to improve the decision-making processes in investments by using the properties of matrix algebra. It seeks to provide valuable insights and tools that can assist financial professionals in making informed choices regarding portfolio optimization, risk assessment, asset pricing, and financial forecasting.
Improve Risk Assessment: The project seeks to enhance the assessment of risks on the investment front. By utilizing matrix operations, it aims to quantify and analyze various risk factors and their interrelationships. The goal is to develop comprehensive risk models that enable investors to better understand and mitigate risks associated with their investment portfolios.
Balancing Risk and Returns: Portfolio optimization seeks to strike a balance between risk and return. It aims to find the optimal asset allocation that offers the highest potential returns for a given level of risk or the lowest level of risk for a desired level of returns. The goal is to achieve an efficient trade-off between risk and return that aligns with the investor's risk preferences and investment goals.

5 Application of Linear Algebra in Analysing Financial Models

Risk Assessment

Introduction

Risk assessment is a systematic process of evaluating the potential risks that may be involved in an investment. It involves understanding and quantifying the impact of different risk factors on the overall risk of a portfolio. Risk assessment can be done using multiple concepts of linear algebra such as variance, covariance, and correlation matrices. Some usage of linear algebra topics to assess and analyze risks in investments include factor exposures matrices, covariance and portfolio variance, and weights of eigenvalues and eigenvectors.

1 Factor Exposures Matrices

Factors are characteristics that explain the risk of stock and equity portfolio returns. In risk factor analysis, a matrix can be constructed to represent the factor exposures of a portfolio to various risk factors. Each row of the matrix represents an asset, in this case, an investment, and each column represents a risk factor. The values in the matrix indicate the sensitivity or exposure of each security to the corresponding risk factor so it is a very convenient way to calculate the total risk of an asset towards various factors. The factor exposures matrix provides a quantitative representation of how each investment in the portfolio is influenced by the different risk factors.

$$\begin{bmatrix} Stocks & MarketRisk & InterestRates & Inflation & OilPrices & ExchangeRates \\ Apple & 0.85 & 0.02 & 0.10 & 0.05 & -0.03 \\ Amazon & 0.92 & 0.04 & -0.06 & 0.12 & -0.08 \\ Microsoft & 0.78 & 0.03 & 0.08 & 0.01 & -0.05 \\ Google & 0.88 & 0.01 & -0.04 & 0.09 & -0.02 \end{bmatrix}$$

The matrix above is a factor exposures matrix of various risk factors for the stocks of some popular companies. Each column represents a different risk factor and each row shows the sensitivity of that stock to that particular risk factor. We can use this matrix to calculate the overall risk of a stock, by taking the sum of the squared factor exposures and then taking the square root of the sum. This will give us the magnitude of the risk associated with each stock based on the factors considered.

For example:

$Apple : \sqrt{0.85^2 + 0.02^2 + 0.10^2 + 0.05^2 + (-0.03)^2} = 0.858$

$Amazon : \sqrt{0.92^2 + 0.04^2 + (-0.06)^2 + 0.12^2 + (-0.08)^2} = 0.934$

This shows that the magnitude of the overall risk factor of both apple and amazon are quite high. A negative risk factor implies that there will be a negative correlation between the risk factor and the stock, aka an increase in the value of the stock or return with a decrease in the risk factor.

2 Covariance and Correlation Matrices

Variance is the deviation of a stock's return with its own average returns. Co variance on the other hand is the variance of a stock's return with respect to another stocks' return. The covariance between two assets affects the portfolio's variance, which is a measure of the portfolio's total risk. Positive covariance between assets implies that they tend to move together, and this can increase the portfolio's overall risk. On the other hand, negative covariance indicates that assets move in opposite directions, potentially reducing the portfolio's risk through diversification. Positive covariance implies that the two assets have a tendency to move together, which can result in increased volatility of the portfolio. When the assets experience simultaneous fluctuations, the portfolio's value may experience larger swings, reflecting higher volatility and potentially higher risk. Positive covariance between assets indicates that their returns are dependent on similar factors or market conditions. If these factors or conditions turn unfavorable, both assets may experience losses simultaneously, leading to concentrated risk in the portfolio.
In general, if there are 'k' stocks in the portfolio, then the size of the variance covariance matrix will be k x k. The formula to create a variance covariance matrix is as follows:

$$\sum_{kxk} = \frac{1}{n} X^T X$$

where

k = number of stocks in the portfolio
n = number of observations
X = this is the n × k excess return matrix
X^T = transpose matrix of X

An excess return matrix is the the difference between stock's daily return over its average return. The return of a stock can be calculated using the following formula:

$$Return = \frac{(EndingStockPrice\ -\ BeginningStockPrice)}{BeginningStockPrice} \times 100$$

Taking the example of a matrix of excess returns of 5 stocks (A 5 x 5 matrix) and multiplying is by its transpose to get X^TX:

	$Cipla$	$Idea$	$Wonderla$	PVR	$Alkem$
$Cipla$	0.02788	0.00679	0.00425	0.00515	0.00804
$Idea$	0.00679	0.14084	0.00497	0.00289	0.00475
$Wonderla$	0.00425	0.00497	0.03055	0.00500	0.00351
PVR	0.00515	0.00289	0.00500	0.05109	0.0038
$Alkem$	0.00804	0.00475	0.00351	0.00338	0.04310

For this matrix, if we take 127 observations, the variance-covariance matrix will be:

	$Cipla$	$Idea$	$Wonderla$	PVR	$Alkem$
$Cipla$	0.0002195	0.0000535	0.0000335	0.0000405	0.0000633
$Idea$	0.0000535	0.001109	0.0000391	0.0000227	0.0000374
$Wonderla$	0.0000335	0.0000391	0.0002405	0.0000394	0.0000277
PVR	0.0000405	0.0000227	0.0000394	0.0004022	0.0000266
$Alkem$	0.0000633	0.0000374	0.0000277	0.0000266	0.0003393

The matrix suggests that the covariance between Wonderla and PVR is 0.000034. This is the same as the covariance between PVR and Wonderla. The first value of the first column corresponds to Cipla and Cipla. This represents the covariance between Cipla and Cipla, the covariance of a stock with itself, which is variance. This is exactly why this matrix is called 'Variance – Covariance Matrix', because it gives us both the values. It tells us the extent to which two assets move together, which can result in increased risk of the portfolio.

Correlation matrices are often preferred when comparing variables on different scales or when focusing on the strength of the linear relationship. We can calculate correlation as $Correlation = \frac{Cov(x,y)}{\sigma_x \times \sigma_y}$ where
Cov(x,y) = covariance between the two stocks
σ_x= Standard deviation of stock x
σ_y = Standard deviation of stock y
For example, using the covariance matrix from above, we can calculate the standard deviations as:

$Cipla$	1.49
$Idea$	3.34
$Wonderla$	1.56
PVR	2.02
$Alkem$	1.86

Given that we have the stock-specific standard deviations, we now need to get the product of the standard deviation of all possible portfolio combination. We resort to matrix multiplication for this. This can be easily achieved by multiply the standard deviation array with the transpose of itself. The corresponding product is:

	$Cipla$	$Idea$	$Wonderla$	PVR	$Alkem$
$Cipla$	0.0002229	0.000499	0.0002328	0.0003017	0.000277
$Idea$	0.0004990	0.0011173	0.0005214	0.0006756	0.0006203
$Wonderla$	0.0002328	0.0005214	0.0002433	0.0003153	0.0002895
PVR	0.0003017	0.0006756	0.0003153	0.0004086	0.0003751
$Alkem$	0.0002770	0.0006203	0.0002895	0.0003751	0.0003444

Dividing the variance-covariance matrix by the product of the standard deviations should result in the correlation matrix. The resulting correlation matrix looks like this:

	$Cipla$	$Idea$	$Wonderla$	PVR	$Alkem$
$Cipla$	1.0	0.10715052	0.14368425	0.13434778	0.22858556
$Idea$	0.10715052	1.0	0.07499042	0.03363225	0.06024903
$Wonderla$	0.14368425	0.07499042	1.0	0.1248625	0.09557120
PVR	0.13434778	0.03363225	0.1248625	1.0	0.07085759
$Alkem$	0.22858556	0.06024903	0.09557120	0.07085759	1.0

The correlation matrix gives us the correlation between any two stocks. For example, if I have to know the correlation between Cipla and Alkem, I simply have to look under the intersecting cell between Cipla and Alkem. Both these should reflect the same result i.e 0.2285. This is quite obvious since the correlation between stock A with Stock B is similar to the correlation of Stock B with Stock A. For this reason, the matrix displays symmetrically similar values above and below the diagonal. The correlations along the diagonal represent the correlation of certain stock with itself.

In a correlation matrix, the value in each cell represents the correlation coefficient between two variables. The correlation coefficient measures the strength and direction of the linear relationship between the variables. The values range from -1 to +1, where -1 indicates a perfect negative linear relationship, +1 indicates a perfect positive linear relationship, and 0 indicates no linear relationship.

```
import numpy as np

rows = int(input("Enter the number of rows: "))
cols = int(input("Enter the number of columns: "))
```

```
print("Enter the matrix elements:")
matrix = []
for i in range(rows):
    row = []
    for j in range(cols):
        element = float(input(f"Enter element at position ({i+1},{j+1}): "))
        row.append(element)
    matrix.append(row)
matrix = np.array(matrix)
n = int(input("Enter number of observations: "))
covariance_matrix =  np.cov(matrix, rowvar=False)
covariance_matrix = covariance_matrix / (n-1)
print("The resulting Covariance Matrix is: ")
print(covariance_matrix)

correlation_matrix = np.corrcoef(matrix, rowvar=False)
print("The resulting Correlation Matrix is: ")
print(correlation_matrix)
```

python The above code in python can be used to calculate the covariance and correlation matrices for a given returns matrix.

3 Eigenvalues and Eigenvectors in Risk Assessment

3.1 Eigenvalues

In risk assessment, eigenvalues represent the variability or importance of different factors or components within a given dataset. Larger eigenvalues correspond to more significant factors that explain a greater proportion of the total variation in the data. By examining the magnitudes of eigenvalues, you can assess the relative importance of different risk factors and determine how much they contribute to the overall risk. For example, if we consider a covariance matrix of 3 stocks:

$$\begin{bmatrix} & Cipla & Idea & Wonderla \\ Cipla & 0.05 & 0.02 & 0.03 \\ Idea & 0.02 & 0.03 & 0.01 \\ Wonderla & 0.03 & 0.01 & 0.01 \end{bmatrix}$$

The corresponding eigenvalues would be
$\lambda_1 = 0.05$

$\lambda_2 = 0.03$
$\lambda_3 = 0.01$
By examining the magnitudes of the eigenvalues, we can assess the relative importance of the risk factors. In this case, $\lambda_1 = 0.05$ is the largest eigenvalue, followed by $\lambda_2 = 0.03$ and $\lambda_3 = 0.01$. This indicates that the first risk factor (associated with the largest eigenvalue) explains the most significant proportion of the total variation in the data, followed by the second and third risk factors. Since the sum of all eigenvalues represents the total variation in the data, we can calculate the proportion of variation explained by each eigenvalue (Assuming sum of the eigenvalues is 1 for simplicity):
Proportion of Variation Explained by $\lambda_1 = \frac{\lambda_1}{(\lambda_1+\lambda_2+\lambda_3)} = \frac{0.05}{(0.05+0.03+0.01)} = 0.5$
Proportion of Variation Explained by $\lambda_2 = \frac{\lambda_2}{(\lambda_1+\lambda_2+\lambda_3)} = \frac{0.03}{(0.05+0.03+0.01)} = 0.3$
Proportion of Variation Explained by $\lambda_3 = \frac{\lambda_3}{(\lambda_1+\lambda_2+\lambda_3)} = \frac{0.01}{(0.05+0.03+0.01)} = 0.1$
These proportions indicate that the first risk factor explains 50% of the total variation in the data, the second risk factor explains 30%, and the third risk factor explains 20%.

Therefore, in this example, the larger eigenvalue (λ_1= 0.05) corresponds to the most significant risk factor that explains the highest proportion of the total variation in the data. The magnitudes of the eigenvalues provide a quantitative measure to assess the relative importance of different risk factors and their contributions to the overall risk.
Eigenvalues can be used to evaluate the stability and sensitivity of risk models. Changes in the eigenvalues can indicate shifts in the importance or significance of risk factors over time or in response to different market conditions. Monitoring the eigenvalues allows for ongoing assessment and adjustment of risk models to capture changing risk dynamics.

3.2 Eigenvectors

Eigenvectors assist in assessing the contribution of individual variables or assets to each risk factor. The eigenvector elements represent the weights or loadings of the variables in the corresponding risk factor. By examining these weights, you can determine which variables have a higher impact on a specific risk factor. This information is valuable for risk management decisions, such as diversification strategies or identifying key drivers of portfolio risk. Let us consider a matrix:

$$\begin{bmatrix} & RiskFactor1 & RiskFactor2 & RiskFactor3 \\ StockA & 0.6 & 0.2 & 0.3 \\ StockB & 0.3 & 0.7 & 0.5 \\ StockC & 0.4 & 0.4 & 0.8 \end{bmatrix}$$

On calculating the eigenvectors, we get:

Eigenvector for Risk Factor 1: $\begin{bmatrix} 0.6 \\ 0.3 \\ 0.4 \end{bmatrix}$

Eigenvector for Risk Factor 2: $\begin{bmatrix} 0.2 \\ 0.7 \\ 0.5 \end{bmatrix}$

Eigenvector for Risk Factor 3: $\begin{bmatrix} 0.3 \\ 0.5 \\ 0.8 \end{bmatrix}$

In this example, the eigenvectors represent the factor loadings or weights that indicate the contribution of each stock to each risk factor. The magnitude and sign of the elements in the eigenvectors determine the strength and direction of the relationship. By examining the eigenvectors, we can assess the contribution of each stock to the different risk factors. For Risk Factor 1, the eigenvector V1 indicates the contribution of each stock. Here, Stock A has a weight of 0.6, Stock B has a weight of 0.3, and Stock C has a weight of 0.4. This suggests that Stock A has a relatively higher contribution to Risk Factor 1 compared to Stock B and Stock C. Similarly, by analyzing the eigenvectors for other risk factors, we can determine the contribution of stocks to those factors. The eigenvector V2 represents the contribution of each stock to Risk Factor 2. Here, Stock B has a weight of 0.7, indicating it has a relatively higher contribution to Risk Factor 2 compared to Stock A and Stock C. Contribution also refers to the fact that a higher contribution of a stock to a risk factor indicates that that particular factor holds a higher risk weight for that stock.
By analyzing all the eigenvectors, we gain insights into how each stock contributes to different risk factors in the portfolio. This information can help us understand the diversification benefits, risk exposures, and potential sources of risk within the portfolio. It can guide portfolio construction decisions, such as balancing exposures to different risk factors or identifying stocks with higher contributions to specific risk factors.

4 State of The Art

We are taking the research paper **"Risk-Based Portfolios with Large Dynamic Covariance Matrices" (2018) by Dr. Kei Nakagawa, Mr. Mitsuyoshi Imamura, Kenichi Yoshida** into consideration to describe the state of the art applications of linear algebra in the field of risk assessment. In the field of portfolio management, practitioners are focusing increasingly on risk-based portfolios rather than on mean-variance portfolios. Risk-based portfolios are constructed based solely on covariance matrices. In this research, the authors compare the performances of risk-based portfolios for several estimation

methods of covariance matrices. The portfolios are as follows:
1. Risk Based Portfolios - In risk assessment, a weighing vector refers to a vector of weights assigned to different factors or variables to determine their relative importance or contribution in the overall risk calculation. It is used to quantify the impact or influence of each factor on the risk assessment process. Another method to calculate the covariance matrix based on the different weights of the risk factors is by using the formula

$$\sum = E[(R - \mu)(R - \mu)^T]$$

where there are 'n' risky assets, with rates of returns given by the random variables R = $(R_1, ..., R_n)^T$, w = $(w_1, ..., w_n)^T$ is the weighing vector for each asset and $\mu = (\mu_1, ..., \mu_n)^T$ is the expected return vector. E represents the expected value of this matrix. It quantifies the average covariance or second-moment properties of the random variable vector R, reflecting how its values vary together. Also, a portfolio's variance and risk (standard deviation) can be expressed as ${\sigma_p}^2 = w^T w$

2. Minimum Variance Portfolios - The minimum variance portfolio is constructed by considering the covariance matrix of the assets in the portfolio. The covariance matrix provides a measure of how the returns of different assets move together. By analyzing the covariance matrix, it is possible to determine the optimal weights to assign to each asset in order to achieve the lowest possible portfolio variance:
$\min_{\text{w}} {\sigma_p}^2 = w^T w$
such that $w^T 1 = 1 \, and \, w > 0$

3. Maximum Diversification Portfolio - The maximum diversification portfolio is an investment strategy that aims to construct a portfolio with the highest level of diversification, thereby reducing concentration risk and capturing the benefits of diversification across different assets.
The key principle behind the maximum diversification portfolio is to allocate capital in a way that maximizes the portfolio's diversification ratio or the Diversification Ratio (DR). The diversification ratio measures the extent to which the portfolio's risk is diversified across its constituent assets.
Here, they define the diversification ratio of any portfolio as DR(w), the ratio of the weighted average of the volatilities divided by the portfolio volatility. The MD portfolio maximizes the diversification ratio and tries to minimize the volatility and the correlation, rather than the expected return.
$\max_{\text{w}} = \sum_{i=1}^{N} w_i \sigma_i {\sigma_p}^{-1}$
In order to maximize DR(w), it is necessary to reduce the inverse term or to increase the non-inverse term. The numerator and the denominator are linked through the volatility term. Because the denominator has a correlation term, a portfolio with a low correlation is constructed as a result. In order to check whether the performance of risk-based portfolios depends on the estimation

accuracy of the covariance matrices, it is necessary to confirm (1) the estimation accuracy of each method, and (2) the effect of the estimation accuracy on performance. In this section, we first compare the estimation accuracy of the covariance matrix using various DCC models. Next, we examine the effect of the difference in the estimation accuracy on the performance of the risk-based portfolios. The DDC (Data-Driven Concept) model is an approach used in risk assessment that focuses on analyzing and understanding data to assess risks. The DDC model leverages data to improve the accuracy and reliability of risk assessments. It allows for a data-driven understanding of risks by identifying patterns and relationships that may not be apparent through traditional risk assessment methods.
The authors performed a Monte Carlo simulation to compare the estimation accuracy of the parameters of various DCC models: the DCC, cDCC, DCC-LS, cDCC-LS, DCC-NLS, and cDCC-NLS models. Monte Carlo simulation is a computational technique used to model and analyze uncertainties in a wide range of fields, including finance, engineering, physics, and risk assessment. It involves generating a large number of random samples or scenarios to estimate the behavior or outcomes of a complex system or process.
They evaluate the accuracy of the covariance matrix estimation using the loss function and found that the average loss of the cDCC-NLS method is the lowest of the various methods for any number of assets and so the cDCC-NLS method shows the best estimation accuracy.They have also examined the effect of the difference in the estimation accuracy on the performance of risk-based portfolios. Here, they used the minimum variance (MV), minimum variance without short constraint (MVS), risk parity (RP), and maximum diversification (MD). Risk parity portfolio is an investment strategy that aims to allocate assets in a way that each asset contributes an equal amount of risk to the overall portfolio. In a risk parity portfolio, the focus is on balancing risk.
Conclusion:
All risk-based portfolios show improved performance as the number of stocks 'N' increases and performance does not depend on the estimation accuracy. There are no statistically significant differences between the DCC and cDCC-NLS or between the cDCC and cDCC-NLS. Also, the weights of the RP and MD are similar across the models which explains why the performance of the RP and MD are similar. They have compared the performance of risk-based portfolios under several estimation methods for covariance matrices: the DCC, cDCC, DCC-LS, DCC-NLS, cDCC-LS, and cDCC-NLS.

Financial Forecasting

Introduction

Financial forecasting is the process of estimating or predicting the future financial outcomes and performance based on historical data, trends and various other assumptions. It involves using quantitative and qualitative methods to project

future revenues, expenses, cash flows and other financial metrics. Financial forecasting in the context of stocks involves predicting the future performance and value of individual stocks or the stock market as a whole. We will be looking at individual stocks, and using 'Tesla' stock (TSLA) as our primary example.

Linear Algebraic Models

Using matrices and vectors to gather historical financial data:

Matrices and vectors provide a powerful mathematical framework for organizing, manipulating and extracting insights from historical data. Matrices are particularly useful for storing data related to multiple variables over a period of time, while we use vectors to represent and analyse data for a single variable. Let us assume we have historical data for a stock over a given time period. We can organise this data into a matrix where each row corresponds to a specific time point and each column represents a particular variable or attribute of the stock. The variables can be the date, opening price, closing price, high price, low price, trading volume, risk factor, etc.

We can the matrix of historical financial data for one stock as follows:

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & x_{m3} & \dots & x_{mn} \end{bmatrix}$$

Here, each element x_{ij} represents the value of the j-th variable at the i-th time point. The matrix has m rows and n columns; where m denotes the number of time points and n represents the number of variables. By utilising matrix operations, we can perform various analyses on this financial data. One analysis we can perform is financial forecasting using linear regression.

Using Linear Regression to perform financial forecasting:

Linear regression is a statistical technique used to model the relationship between a dependant variable and one or more independent variables by fitting a linear equation to observe data. In the context of predicting stock prices, we can use linear regression to estimate the relationship between the closing prices of a stock and the corresponding number of days.

Lets consider a data set where we have historical data for the closing prices of a stock (y) and the number of days (X). Then, we can represent this data set as a matrix where each row corresponds to a specific day and has two columns: one for the closing price (y) and one for the number of days (X) This matrix

can be written as below:

$$\mathbf{X} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_n & y_n \end{bmatrix}$$

Here, x_i represents the number of days for the i-th data point and y_j represents the corresponding closing prices. The matrix X has n rows indicating the number of data points, and 2 columns.

To apply linear regression, we aim to find the best fit line that minimizes the overall difference between the predicted closing prices ($\hat{y}$) and the actual closing prices (y) for the given dataset. The linear equation representing the relationship can be defined as:

$$\hat{y} = \beta_0 + \beta_1 x$$

Here, $\hat{y}$ is the predicted closing price, x is the number of days, β_0 is the intercept (y-axis intercept), and β_1 is the slope of the line (representing the change in closing price with respect to the number of days). The goal of linear regression is to estimate the optimal value of β_0 and β_1 that minimize the sum of squared difference between the predicted and actual closing prices.

Once the coefficients β_0 and β_1 are determined, we can use the linear equation to make predictions for future closing prices based on our data considering number of days.

Here, we have taken a case study of TSLA stock.

Using the python module yfinances, we have imported the data containing Days and Closing Prices. If we have specified our time frame to be "10y". for example:

- Days contains all values from day 0 (that is 2013-06-XX) to present day (Where XX is the exact same Date as present day.)
- Closing Prices represents the price of the Stock at the end of each day, for these 10y worth of data.

The code:

```
import yfinance as yf

def get_data(symbol):
    stock = yf.Ticker(symbol)
    data = stock.history(period="10y")
    return data
def get_data2(symbol):
    stock = yf.Ticker(symbol)
    data = stock.history(period="1mo")
    return data
def get_data3(symbol):
    stock = yf.Ticker(symbol)
    data = stock.history(period="30mo")
    return data
```

Here, "data" stores the past 10y worth of data of a particular stock symbol you input, "data2" stores the past 1 month, and "data3" stores the past 30 months. Now we create a model to plot historical data along with the linear regression. This model also calculates mean square error. We call this function *regression analysis*.

Here is the code:

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
import matplotlib.pyplot as plt
import numpy as np

def regression_analysis(data):
    data['Day'] = range(1, len(data) + 1)

    X = data['Day'].values.reshape(-1, 1)
    y = data['Close'].values.reshape(-1, 1)

    X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.2, random_state=0)
    regressor = LinearRegression()
    regressor.fit(X_train, y_train)

    plt.scatter(X, y, color='blue')
    plt.plot(X, regressor.predict(X), color='red')
    plt.title('Stock Price vs Time (Train set)')
    plt.xlabel('Time (days)')
    plt.ylabel('Stock Price')
    plt.show()

    y_pred = regressor.predict(X_test)
    mse = metrics.mean_squared_error(y_test, y_pred)
    print("Mean Squared Error:", mse)

    return regressor
```

In this code, we import functions from the sklearn library. Some of the primary functions we use are *train test split* which splits the given data X and y into training data (data used to train the model) and testing data (data used to test the model).

We use the *LinearRegression*() to find the linear regression of the given data. Here, $fit()$ finds the β value. Here, *predict*() is used to find the predicted y.

Next, we use the given data, and try to find the closing prices for the next few days. The below picture is the code for the following:

```
def predict_and_adjust_next_5_days(regressor, data):
    # Predicting the next 5 days
    next_5_days = np.array(range(len(data) + 1, len(data) + 6)).
    reshape(-1, 1)
    predicted_prices = regressor.predict(next_5_days)

    # Generate random risk values for each day
    risks = np.random.rand(5)
    affect = np.random.randint(50)
    adjusted_predicted_prices = predicted_prices.flatten() - (0.5 -
     risks) * affect
    # Plotting the existing data
    plt.scatter(data['Day'], data['Close'], color='green')
    plt.plot(data['Day'], regressor.predict(data['Day'].values.
    reshape(-1,1)), color='red')

    # Plotting the original predicted data
    plt.scatter(next_5_days.flatten(), predicted_prices.flatten(),
    color='blue', s=1)

    # Plotting the risk adjusted predicted data
    plt.scatter(next_5_days.flatten(), adjusted_predicted_prices,
    color='purple')

    plt.title('Stock Price vs Time')
    plt.xlabel('Time (days)')
    plt.ylabel('Stock Price')
    plt.legend(['Linear Regression', 'Historical data', 'Predicted
    data', 'Risk Adjusted data'])
    plt.show()

    return predicted_prices, adjusted_predicted_prices
```

In our model, we are employing linear regression to generate a trend line that's based on the historical data of stock prices.

This trend line can be viewed as the average behavior of the stock over the observed time period, and is represented mathematically by our β matrix. Specifically, the components of this matrix, β_0 and β_1, stand for the intercept and the slope of our line, respectively.

However, it's crucial to understand that stock prices do not strictly adhere to this average trend. In reality, they oscillate around it, influenced by a variety

of factors. In our model, we've condensed these influences into two variables: risk rate and affect.

Let's delve deeper into what these variables represent.

Risk rate is a random value between 0 and 1, serving as an abstract representation of diverse market risks. It gives us an estimate of the potential deviation of the stock price from the average trend. When the risk rate is high, it suggests a greater likelihood of significant deviation from the trend line.

On the other hand, **affect,** a semi-random value between 0 and 50, quantifies the extent to which the stock price will react to the risk rate. A high affect value implies that a given risk factor will significantly impact the stock price.

Using these two variables, we construct a function to predict how much the stock price will change - either increase or decrease. When the risk rate is above 0.5, we adjust the stock price upwards; when it's below 0.5, the price is adjusted downwards.

Now, here's the crucial part - instead of changing our β matrix (which would essentially change our overall trend), we're introducing fluctuations to our predicted stock prices around the trend line. This is achieved by adjusting the y-values, which represent the stock prices, based on the risk rate and the affect. Therefore, our β matrix remains constant, while the resulting stock prices (y-values) are modulated around the linear regression line.

To put this in linear algebra terms, we can view this modulation as an addition operation performed on the vector of y-values, adding a random vector influenced by the risk rate and affect to our original y-vector.

In conclusion, we're not altering the linear regression itself. Instead, we're introducing a level of variability to the stock prices that oscillate around the regression line. This variability, injected into our model by the risk rate and affect variables, allows us to create a more realistic portrayal of stock prices that includes both an overall trend (given by the linear regression) and individual fluctuation.

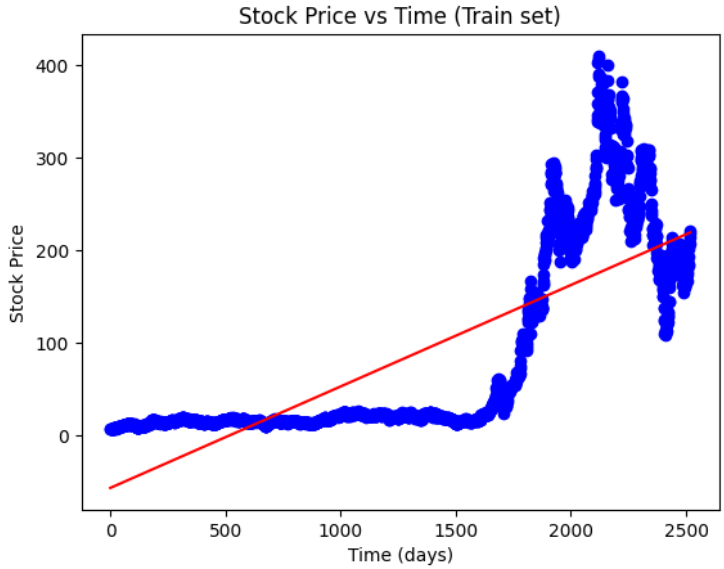
Let risk rate be r, and affect be a:

$$\text{stock\_price\_change}(r, a) = \begin{cases} (0.5 - \text{r}) \times \text{a} & \text{if r} > 0.5 \\ -(0.5 - \text{r}) \times \text{a} & \text{otherwise} \end{cases}$$

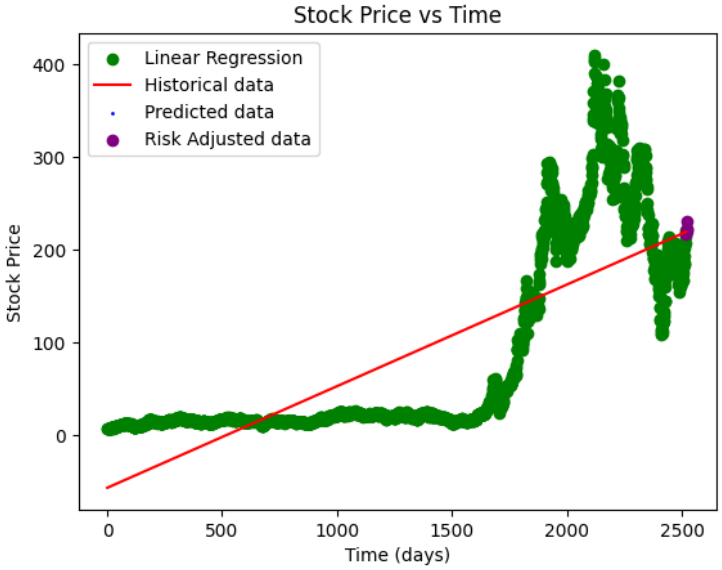
The upcoming code is used to run our previously defined functions. This code collects data about the Tesla Stock. Then we run a regression analysis of the past 10 years, the past 30 months, and the past 1 month. We also, respectively, plot this regression analysis. We then run a program to predict the next five days of stock prices for each of the time duration. We plot this graph as well.

```
data = get_data("TSLA")
data2 = get_data2("TSLA")
data3 = get_data3("TSLA")
regressor = regression_analysis(data)
predicted_prices, adjusted_predicted_prices =
    predict_and_adjust_next_5_days(regressor, data)
regressor3 = regression_analysis(data3)
predicted_prices3, adjusted_predicted_prices3 =
    predict_and_adjust_next_5_days(regressor3, data3)
regressor2 = regression_analysis(data2)
predicted_prices2, adjusted_predicted_prices2 =
    predict_and_adjust_next_5_days(regressor2, data2)
```

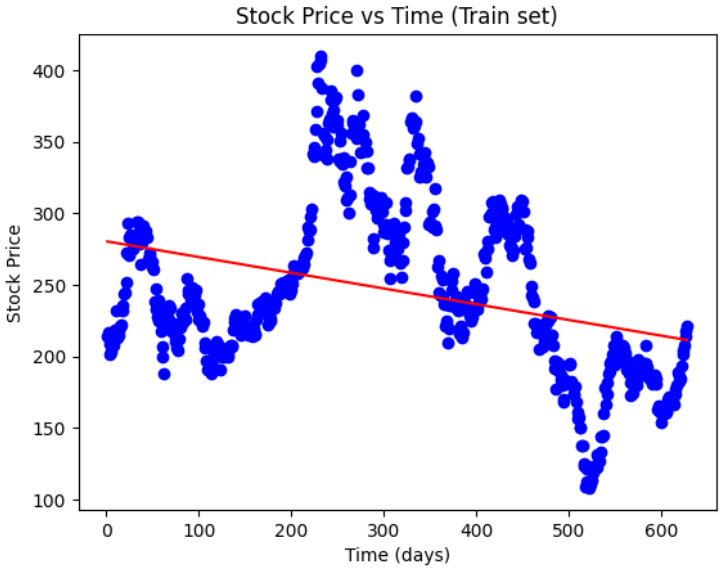
The following are the output graphs along with the printed outputs:



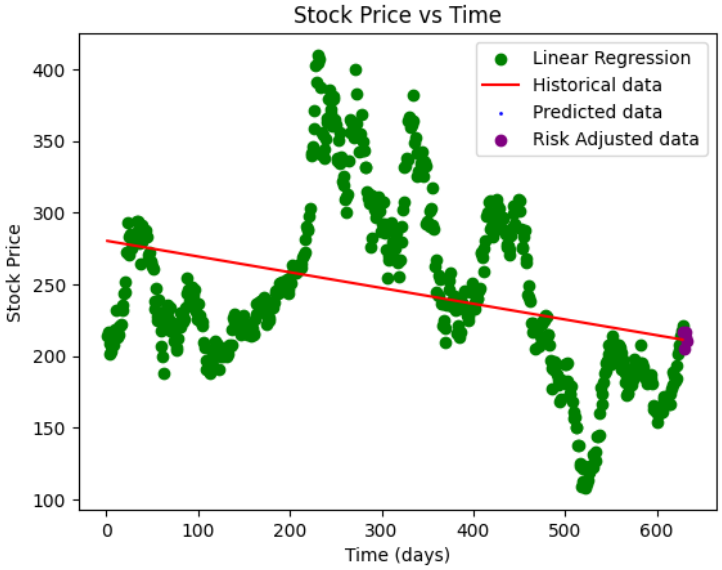

```
regression analysis of the past 10years of Tesla stock
Mean Squared Error: 4182.644777829221
```



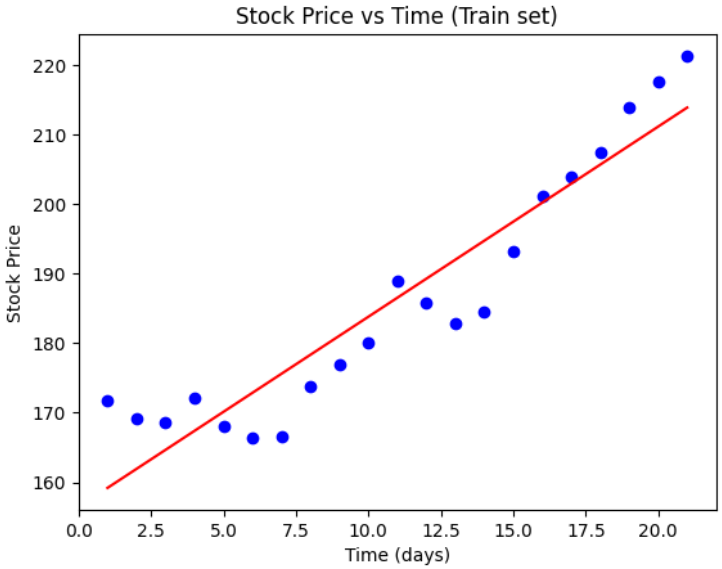

```
prediction of the Next 5 days
```



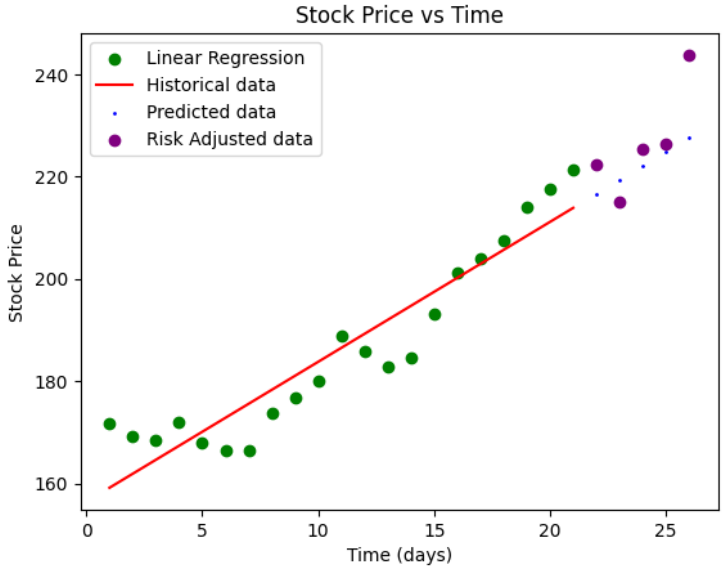

```
regression analysis of tesla stock based on part 30 months of daya
Mean Squared Error: 3037.8822520623617
```




```
prediction of the Next 5 days
```




```
regression analysis of tesla stock based on the past month of data:
Mean Squared Error: 48.50854694176982
```




```
prediction of the Next 5 days
```

We can see that our error is lesser when we take a shorter time period.

State Of The Art

We are taking the research paper **"Stock Price Prediction Using Regression Analysis" (2016) by Dr. Md. Uzzal Hossain, Md. Shakhawat Hossain, and Md. Monirul Islam."** into consideration to describe the state of the art applications of linear algebra (in specific, linear regression) in the field of financial forecasting.

Using mathematical and technological methods and tools to predict stocks is not something unique to this decade. But, the recent advances in machine learning and big data have added more dimension to this research area.

The authors of this paper use multiple linear regression method in the prediction of stock prices, taking 6 different organisations enlisted in the Dhaka Stock Exchange. They used 3 years of historical stock market data.

Multiple linear regression is a statistical method used to model the relationship between a dependant variable and two or more independent variables. In simple linear regression, we use only one linear variable.

The goal of multiple linear regression is to find a linear equation that best fits the data by estimating the coefficients of each independent variable. The follow is the equation

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ...\beta_n x_n + \epsilon$$

Where $\hat{y}$ is the dependent variable, and $x_1, x_2, x_3....x_n$ are the independent variables. Here, $\beta_0, \beta_1, ...\beta_n$ are the coefficients (called regression coefficients or parameters) representing the effect of each independent variable on the dependent variable. ϵ here represents error.

This method assumes a linear relationship between dependent and independent variables. The coefficients $\beta_0, \beta_1, ...\beta_n$ are estimated using a technique called **ordinary least squares (OLS)** , which, in essence, minimizes the sum of the squared differences between the observed and the predicted values of the dependent variable. Here, they compute the **mean squared error** (which is average square difference between predicted values and actual values of a dependent variable.) The goal is to minimize the mean squared error during training; ie, find model parameters that result in the smallest overall square errors.

Finally, the authors proposed to use an algorithm called **Stochastic Gradient Descent Algorithm (SGD** which is an iterative optimization algorithm which is commonly used to train machine learning models. SGD randomly selects a subset of training data (what we can call a mini-batch) to compute the **gradient of the MSE (also know as the loss function)**. It then updates the model parameters in the direction of the negative gradient in order to minimize the loss.

In summary, for a particular set of stocks :

- Multiple Regression:
  - A statistical technique to model the relationship between a dependent variable and multiple independent variables.
  - Estimation of coefficients for each independent variable to find a best-fit linear equation.
- Squared Loss Function:
  - Also known as mean squared error (MSE), a commonly used loss function in regression problems.
  - Measures the average squared difference between predicted and actual values.
  - Minimization of this loss function during training to achieve the smallest overall squared errors.
- Stochastic Gradient Descent (SGD) Algorithm:
  - An iterative optimization algorithm for training machine learning models.
  - Suitable for large datasets due to efficient and incremental parameter updates.
  - Random selection of mini-batches to compute gradients and update model parameters in the direction of the negative gradient.
- Prediction Process:

- Definition of multiple regression model with dependent and independent variables.
- Application of squared loss function to calculate the difference between predicted and actual values.
- Utilization of SGD algorithm to optimize model parameters by iteratively adjusting them based on computed gradients.
- Training continues until convergence or a specified number of iterations, reducing the loss and improving predictive performance.
- Trained model used for predictions on new data by applying learned coefficients to corresponding independent variables.

Conclusion

In this study, we've utilized concepts of linear algebra to perform financial forecasting, specifically with respect to stock investments. The integration of matrices and vectors provided us with a robust tool set for organizing, manipulating, and extracting insights from historical data. Moreover, our model for financial forecasting relied heavily on linear regression to establish relationships between independent and dependent variables, leading to the prediction of future stock prices.

However, it is important to note that while linear regression provides an approachable and intuitive model for forecasting, it also comes with limitations. The main assumption is that the relationship between variables is linear, which may not be the case in many scenarios, especially in complex and volatile markets. Also, the risk rate and affect in our model are only rudimentary representations of the many complexities in the stock market.

Despite these limitations, the methods and concepts discussed in this report remain relevant for financial forecasting. They can serve as a solid foundation for more sophisticated models and techniques, like polynomial regression, machine learning models, or even deep learning models for financial forecasting. The blend of linear algebra, statistical methods, and computational tools forms a versatile platform that can help investors make informed decisions.

Portfolio Optimisation

Introduction

Portfolio optimization is the process of strategically allocating investments within a portfolio to achieve an optimal balance between risk and return. It involves utilizing quantitative techniques, statistical analysis, and mathematical models to construct an investment portfolio that maximizes returns while minimizing risk, based on an investor's specific objectives, constraints, and risk tolerance.

The ultimate goal of portfolio optimization is to construct a well-diversified portfolio that aligns with the investor's objectives, risk tolerance, and investment horizon. The optimal portfolio aims to maximize returns given the level of risk that the investor is willing to accept or minimize risk while achieving the desired level of returns. It is a dynamic process that may require periodic adjustments as market conditions, investment goals, and risk preferences evolve.

Using Matrix operations like Inverse , Transpose to gather the portfolio Background:

- The portfolio optimization was performed using linear algebra approach. The result is a formula used to determine the optimum composition of the portfolio weights. The resulting formula is very useful for the analysis of the investment portfolio optimization.

1 Portfolio Weights

- To understand the portfolio's value that is invested in each asset.

It refer to the allocation or percentage of the total investment that is assigned to each asset within a portfolio. These weights determine the proportion of the portfolio's value that is invested in each asset. The portfolio weights are typically represented as a vector of values, with each value indicating the weight of the corresponding asset.
Suppose we have a portfolio with four assets: stocks (S), bonds (B), real estate (RE), and commodities (C). We have historical data on the returns of these assets over a specific time period. We can represent these returns as a vector:

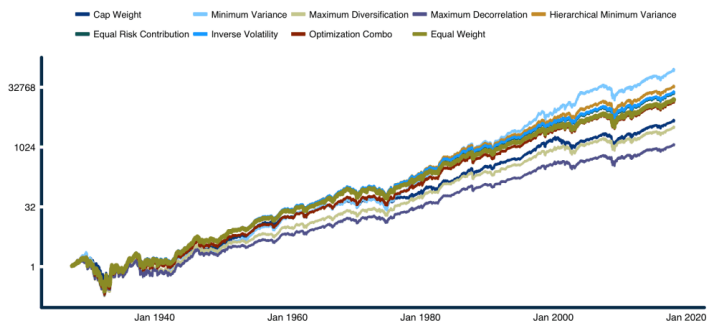
$$R = \begin{bmatrix} R_S \\ R_B \\ R_{RE} \\ R_C \end{bmatrix}$$

To calculate the portfolio weights using linear algebra, we can use the following formula:

$$w = (Cov(R)^{-1} \cdot \mu)/(1^T \cdot Cov(R)^{-1} \cdot \mu)$$

Where:

- w is the vector of portfolio weights.
- $Cov(R)$ is the covariance matrix of the asset returns.
- μ is the vector of expected returns for each asset.



The inverse of the covariance matrix, $Cov(R)^{-1}$, is used to calculate the weights. This allows us to capture the relationships and diversification benefits among the assets.

Let's assume the following values for expected returns and covariance matrix:

Expected returns:

$$\mu = \begin{bmatrix} 0.08 \\ 0.04 \\ 0.10 \\ 0.06 \end{bmatrix}$$

Covariance matrix:

$$Cov(R) = \begin{bmatrix} 0.04 & 0.02 & 0.01 & 0.03 \\ 0.02 & 0.03 & 0.01 & 0.02 \\ 0.01 & 0.01 & 0.06 & 0.01 \\ 0.03 & 0.02 & 0.01 & 0.05 \end{bmatrix}$$

By substituting the values into the formula, we can calculate the portfolio weights using linear algebraic operations. The resulting weights will indicate the optimal allocation for the portfolio.

Additionally, matrices and vectors play a vital role in other aspects of portfolio optimization, such as calculating portfolio risk, constructing the efficient frontier, and solving optimization problems.

2 Mean - Variance Optimization

- Maximum returns and minimum risk optimisation

Mean-Variance Optimization is a popular approach used in portfolio optimization to determine the optimal asset allocation that balances risk and return. It aims to maximize the expected portfolio return while minimizing the portfolio's variance or standard deviation, considering an investor's risk tolerance.

The ultimate goal of portfolio optimization is to construct a well-diversified portfolio that aligns with the investor's objectives, risk tolerance, and investment horizon.

Suppose we have a portfolio with three fictional assets: Asset A, Asset B, and Asset C. We want to determine the optimal weights to allocate to each asset based on their expected returns, the covariance matrix, and a given risk tolerance.

First, we define the following information:

Expected Returns:

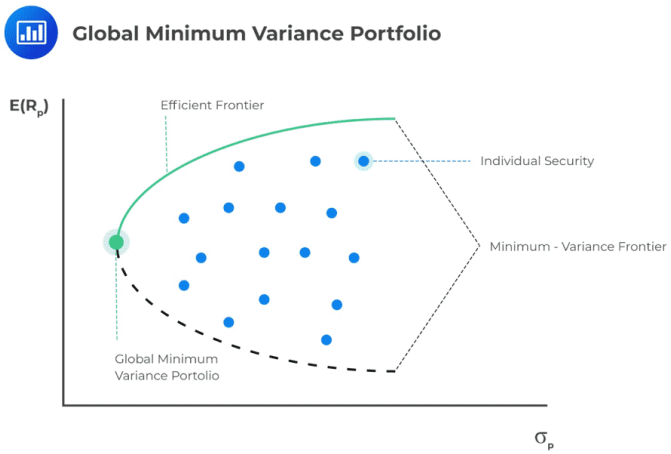
$$\begin{aligned}&\text{Asset A} :8\%\\&\text{Asset B} :10\%\\&\text{Asset C} :12\%\end{aligned}$$

Covariance Matrix:

$$C = \begin{bmatrix}0.04 & 0.02 & 0.01\\0.02 & 0.09 & 0.03\\0.01 & 0.03 & 0.16\end{bmatrix}$$

Risk Tolerance: 0.08 (this represents the maximum acceptable portfolio volatility)

To solve for the optimal weights, we can use the mean-variance optimization technique:



Step 1: Calculate the expected returns vector (R) and the covariance matrix (C):

$$R = \begin{bmatrix}0.08\\0.10\\0.12\end{bmatrix}$$

$$C = \begin{bmatrix}0.04 & 0.02 & 0.01\\0.02 & 0.09 & 0.03\\0.01 & 0.03 & 0.16\end{bmatrix}$$

Step 2: Set up the mean-variance optimization problem:

Let w be the weight vector representing the allocation to each asset. We want to find the weights w that minimize the portfolio variance while achieving the desired expected return. Mathematically, this can be represented as:

$$\text{minimize } w^T C w$$
$$\text{subject to } w^T R = r$$
$$w^T 1 = 1$$

where w^T represents the transpose of the weight vector w, C is the covariance matrix, R is the expected returns vector, r is the desired expected return, and 1 is a vector of ones.

Step 3: Solve the quadratic optimization problem:

Using linear algebra, we can solve the above quadratic optimization problem. By introducing Lagrange multipliers, we can transform it into a linear system:

$$2Cw - \lambda R - \mu 1 = 0$$
$$w^T R - r = 0$$
$$w^T 1 - 1 = 0$$

where λ and μ are the Lagrange multipliers.

Step 4: Solve the linear system of equations:

Solving the above linear system of equations will give us the optimal weights. This can be done using various numerical methods, such as matrix inversion or optimization algorithms.

Step 5: Interpret the solution:

Once the linear system is solved, the resulting weight vector w will represent the optimal allocation to each asset. The values in the weight vector will indicate the percentage of the portfolio allocated to each asset, ensuring that the portfolio achieves the desired expected return while minimizing the variance.

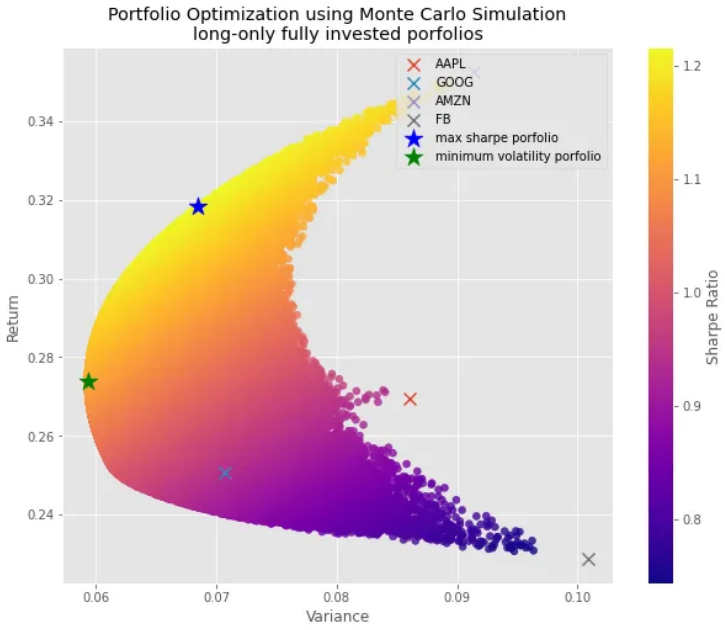
Next, we use the given data, After defining the core objectives, we lay out the optimization problem as functions, being optimize sharp ratio $(R, \sum, rf = 0, wbounds = (0,1))$ and $\underline{minimize\ portfolio\ variance}\,(R, \sum bounds = (0,1))$. This is the python code for the following optimization :

```
def minimize_portfolio_variance(mean_returns, cov_matrix, w_bounds
    =(0,1)):
    "This function finds the portfolio weights which minimize the
    portfolio volatility(variance)"

    init_guess = np.array([1/len(mean_returns) for _ in range(len(
    mean_returns))])
    args = (mean_returns, cov_matrix)
    constraints = ({'type': 'eq', 'fun': lambda x: np.sum(x) - 1})
    result = opt.minimize(fun=portfolio_variance,
                          x0=init_guess,
                          args=args,
                          method='SLSQP',
```

```
                        bounds=tuple(w_bounds for _ in range(len(
    mean_returns))),
                        constraints=constraints,
                        )

    if result['success']:
        print(result['message'])
        min_var = result['fun']
        min_var_weights = result['x']
        min_var_return, min_var_variance, min_var_std = portfolio(
    min_var_weights, mean_returns, cov_matrix)
        min_var_sharpe = (min_var_return/min_var_std)*(252**0.5)
        return(min_var_sharpe, min_var_weights, min_var_return.item
    ()*252, min_var_variance.item()*252, min_var_std.item()*(252**
    0.5))
    else:
        print("Optimization operation was not succesfull!")
        print(result['message'])
        return(None)
```



In our model, we are employing best way of optimisation to generate a portfolio line(maximize the returns) that's based on the calculations of data of the stock prices.

State Of The Art

We are taking the research paper **"Linear Algebra on investment portfolio optimization model By Dr B Basuki, S Sukono, D Sofyan, S S Madio and N Puspitasari"** into consideration to describe the state of the art applications of linear algebra in Portfolio Optimisation.

-To cite this article: B Basuki et al 2019 J. Phys.: Conf. Ser. 1402 077089

Abstract: In this paper we discuss the issue of linear algebra on the investment portfolio optimization models. It was assumed that stock returns are analyzed have a certain distribution, so that the mean and variance and covariance between the separation can be determined. Return of some stock used to form a vector averaging, and the number of shares used as the basis to form a unit vector. While the variance of each stock as well as the covariance between stocks, is used to form a covariance matrix. The investment portfolio was formed consisting of several stocks, in order to maximize the expected return and minimize risk. The portfolio optimization was performed using linear algebra approach. The result is a formula used to determine the optimum composition of the portfolio weights. The resulting formula is very useful for the analysis of the investment portfolio optimization.

Key Role: In linear algebra, portfolio objective function is expressed as equations involving weight vector, the mean vector, unit vector, and the covariance matrix. Wherein the weight vectors transpose times the unit vector must be equal to one. Both in engineering and the use of ordinary algebra linear algebra, optimization of the portfolio can be done by using Lagrange multiplier and Kuhn-Tucker method.

Mathematical model: In this part of the mathematical model is discussed about the equation for determining stock returns, and formulating weight optimization of portfolio investment.

Conclusion: In this paper has discussed the issue of linear algebra on the investment portfolio optimization models. Discussion has formulated the optimum weight vector expressed as a function of linear algebra, as given in equation. Furthermore, the resulting formula is used to analyze five stocks in the formation of an investment portfolio.

Bibliography

- `https://iopscience.iop.org/article/10.1088/1742-6596/1212/1/012031`
- `https://ajse.aiub.edu/index.php/ajse/article/view/490`

- https://www.investopedia.com/articles/trading/09/linear-regression-time-price.asp
- https://www.alpharithms.com/predicting-stock-prices-with-linear-regression-214618/
- https://pypi.org/project/yfinance/
- https://www.cmi.ac.in/~ksutar/NLA2013/stockportfolio.pdf
- https://blog.quantinsti.com/calculating-covariance-matrix-portfolio-variance/
- https://zerodha.com/varsity/chapter/risk-part-3-_variance-covariance-matrix/
- https://www.researchgate.net/publication/325150874_Risk-Based_Portfolios_with_Large_Dynamic_Covariance_Matrices
- https://iopscience.iop.org/article/10.1088/1742-6596/1402/7/077089