Monte Carlo Simulation Project 1 Final Part

Sajida Malik

2023-11-21

Problem Statement:

The primary goal of this project is to navigate the intricate process of constructing an optimal portfolio, requiring a detailed examination of historical returns, correlations, and the risk-free rate. In alignment with modern portfolio theory, the overarching objective is to reveal the "efficient frontier," which represents a range of portfolios offering the maximum return for a given risk level or minimizing risk for a specified return. The ultimate aim is to create a portfolio that aligns seamlessly with an investor's risk tolerance and financial objectives, striking a delicate balance between risk and return in the pursuit of optimal investment outcomes.

Introduction:

Modern Portfolio Theory (MPT) is a groundbreaking framework developed by Harry Markowitz in the 1950s that revolutionized the field of investment management. MPT fundamentally changed how investors perceive and construct portfolios by introducing a systematic approach to balancing risk and return. The core idea behind MPT is that the risk of individual assets is not as important as how they interact within a diversified portfolio. According to MPT, by combining assets with different risk and return profiles, it's possible to create a portfolio that achieves the maximum expected return for a given level of risk, or conversely, the minimum risk for a desired level of return. Central to MPT is the concept of the "efficient frontier," which represents the set of optimal portfolios that offer the highest expected return for a defined level of risk or the lowest risk for a given level of expected return. MPT has become a cornerstone of modern financial theory, providing investors with a systematic and mathematical approach to portfolio construction and risk management. (Reference: @Google)

Methodology:

In this methodology, the process initiates by randomly selecting pairs of two stocks from a set of five, namely Apple, Amazon, Google, The Trade Desk, and Microsoft, leveraging historical data spanning the past 180 days for a robust assessment of their performance. These selected pairs are visually represented through trend lines and box plots.

For each randomly chosen stock pair, an optimal portfolio is crafted by exploring various weight allocations between the paired assets. This analysis hinges on evaluating the risk-return relationship across different weight distributions, with a focus on identifying the most efficient portfolio configurations. Utilizing quantitative techniques, such as mean-variance optimization, the objective is to unveil the efficient frontier. This frontier delineates the spectrum of optimal portfolios that either maximize returns for a specified level of risk or minimize risk for an anticipated return.

To assess the effectiveness of the portfolio, the actual returns are compared with the predicted returns obtained through the evaluation process. Furthermore, line plots and bar plots are constructed to visually compare predicted versus real returns. This comprehensive methodology provides both a visual and analytical understanding of the dynamics between risk and return, revealing the most favorable compositions for portfolios.

Data Analysis and Results:

#Creating number of days forward and back variables for optimizing

```
num_days_forward <- 30 #For optimization 30 days forward
num_days_back <- 180 #For optimization 180 days back
```

#calculating of returns based on the closing prices using the formula

#Creating a function

```
mean_std_historical = function(Returns, number_of_observation,
num_days_forward) {
  mean_return_daily <- mean(Returns[(number_of_observation-
num_days_forward+1):number_of_observation])
  std_return_daily  <- sd(Returns[(number_of_observation-
num_days_forward+1):number_of_observation])
  result <- list(Mean_Return = mean_return_daily, Std_Return =
std_return_daily)
  return(result)
}

calculate_returns = function(Y) {
  len = nrow(Y)
  yDif = (Y[2:len, ] - Y[1:len-1, ]) / Y[1:len-1, ]
}

returns_from_stock = function(Y, col_name) {
  stock_name <- deparse(substitute(Y))
  Price <- matrix(as.numeric(Y[, col_name]), ncol = 1)
  Returns <- calculate_returns(Price)
  number_of_observation <- length(Returns)
  result <- list(Returns = Returns, number_of_observation =
number_of_observation)
  return(result)
}

# Function to do one simulation run
prices_simulation <- function(Num_forward, mean, sd, starting_price) {
  set.seed(123)
```

```r
  simulated_returns <- rnorm(Num_forward, mean = mean, sd = sd)
  simulated_prices <- cumprod(c(starting_price, 1+simulated_returns))
  simulated_prices <- round(simulated_prices[-1],2) #rounding prices
  results <- round(simulated_prices[-1],2)
  return(simulated_prices)
}

# Function to do repeated simulation runs
simulation <- function(runs, Num_forward, mean, sd, starting_price) {
  prices_sim <- replicate(
    runs,
    prices_sim(Num_forward = Num_forward, mean = mean,
                    sd = sd, starting_price = starting_price)
  )
}

expected_return_calculation <- function(Y, starting_date, num_days_back,
num_days_forward) {
  # Getting data
  getSymbols(Y, from = starting_date , to = Sys.Date())

  # Extract the data from the environment
  data <- get(Y)
  col_name <- paste0(Y, ".Close")

  #Close prices
  ClosePrices <- Cl(data)
  result_returns <- returns_from_stock(data, col_name)

  # Access the returned values
  Returns <- result_returns[[1]]
  number_of_observation <- result_returns[[2]]

  metrics_historical <- mean_std_historical(Returns, number_of_observation,
num_days_forward)

  mean = metrics_historical[[1]]
  sd = metrics_historical[[2]]

  starting_price <- data[, col_name][[(number_of_observation-
num_days_forward)]]

  returns_simulated <- 1 + rnorm(num_days_back, mean = mean, sd = sd)
  N_sims<- 1999
  return_list <- matrix(0, nrow = N_sims, ncol = num_days_forward)
  price_list <- matrix(0, nrow = N_sims, ncol = num_days_forward + 1)
  for(i in 1:N_sims){
    return_list[i,] <- rnorm(num_days_forward,mean=mean,sd=sd)
    price_list[i,] <- cumprod(c(starting_price, 1 + return_list[i,]))
```

```
  }

  total_returns <- array(NA, dim = N_sims, dimnames = NULL)
  for(i in 1:N_sims){
    total_returns[i] <- (price_list[i,num_days_forward+1] - price_list[i,1])
/ price_list[i,1]
  }
  med <- match(median(total_returns),total_returns)

  expected_return <- mean(total_returns)
  expected_return_std <- sd(total_returns)

  result <- list(expected_return = expected_return,
                 expected_return_std = expected_return_std,
                 ClosePrices = ClosePrices)
  return(result)
}

# calculate return and risks based on w
cal_pf_return <- function(w, mu1, mu2) {
  return(w*mu1 + (1-w)*mu2)
}
cal_pf_risk <- function(w, sigma1, sigma2, rho12) {
  return((w^2) * (sigma1^2) + ((1-w)^2) * (sigma2^2) + 2*w*(1-
w)*rho12*sigma1*sigma2)
}
```

Exploratory Data Analysis:

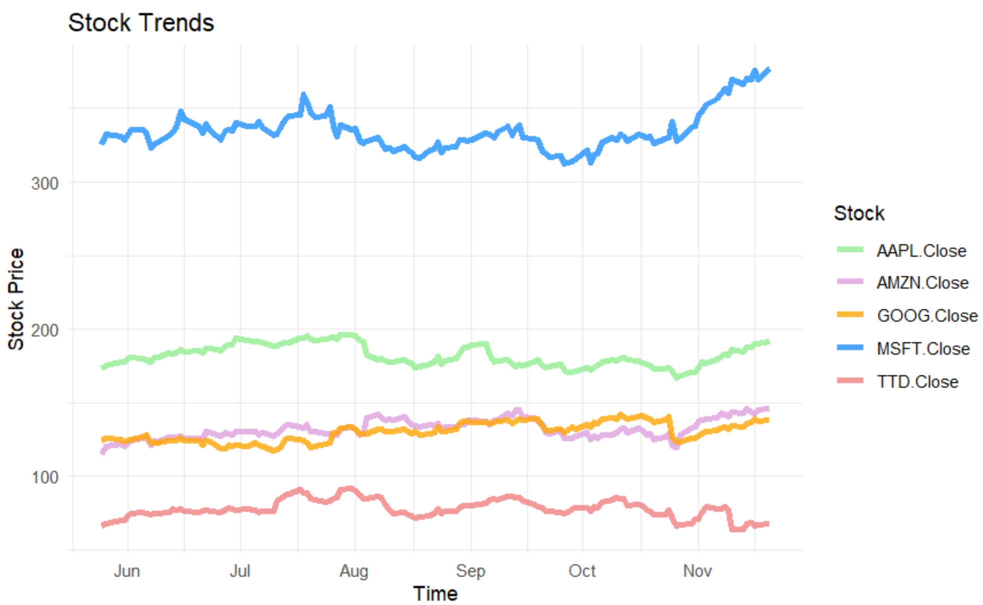
#Creating variable stocks data

```
stocks_data <- c("AAPL","AMZN","GOOG","TTD", "MSFT")
no_stocks <-length(stocks_data)
#Getting historical data of 5 stocks from yahoo finance
getSymbols(stocks_data, from = Sys.Date() - num_days_back, to = Sys.Date())
#Start date is 180 days from today

## [1] "AAPL" "AMZN" "GOOG" "TTD"  "MSFT"
```

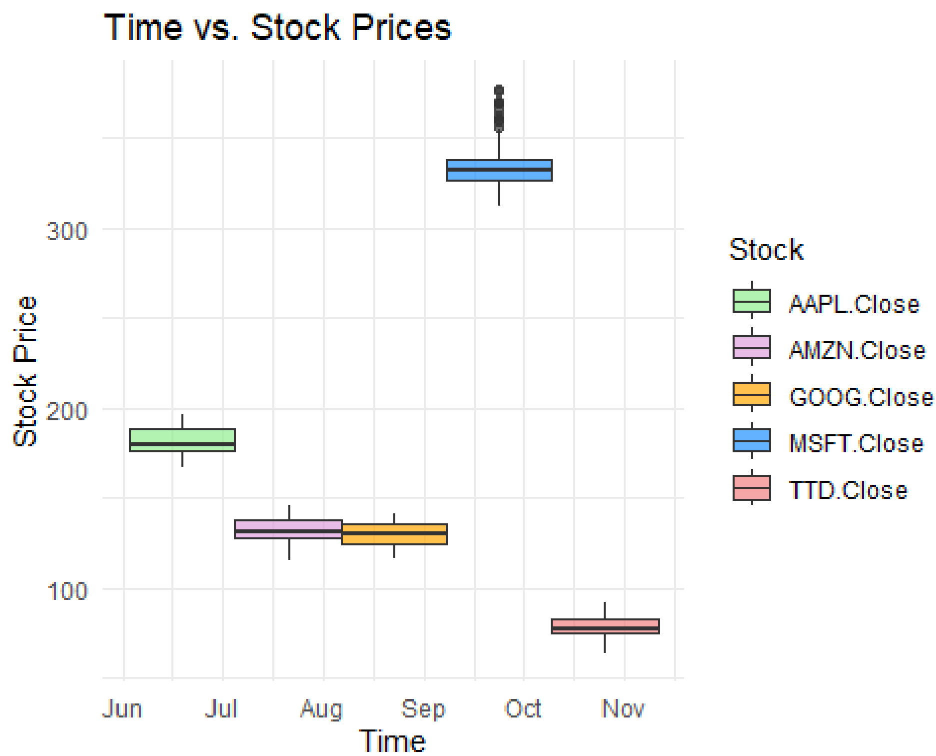
#Trendlines and boxplots of stocks data

```
colors <- c('lightgreen', 'plum', 'orange', 'dodgerblue', 'lightcoral')

# Combine stock data into a single data frame
stock_df <- data.frame(Time = index(AAPL), do.call(cbind, lapply(stocks_data,
function(stock) Cl(get(stock)))))

# Reshape data for ggplot
stock_df_long <- tidyr::gather(stock_df, key = "Stock", value = "Close", -
Time)
```

```r
# Create the plot
ggplot(stock_df_long, aes(x = Time, y = Close, color = Stock)) +
  geom_line(linewidth = 1.5, alpha = 0.8) +
  labs(title = "Stock Trends", x = "Time", y = "Stock Price") +

  theme(legend.position = "topright") +

  scale_color_manual(values = colors) +
  theme_minimal()
```

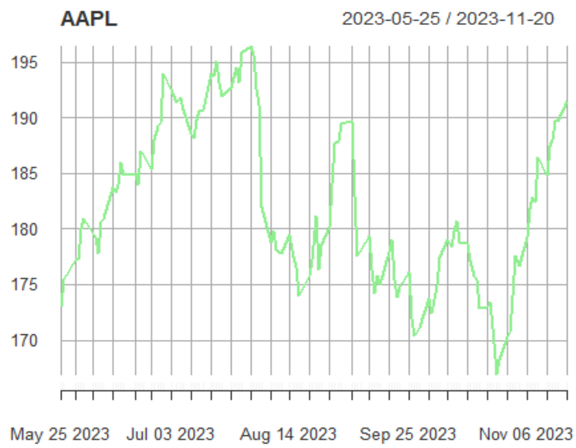


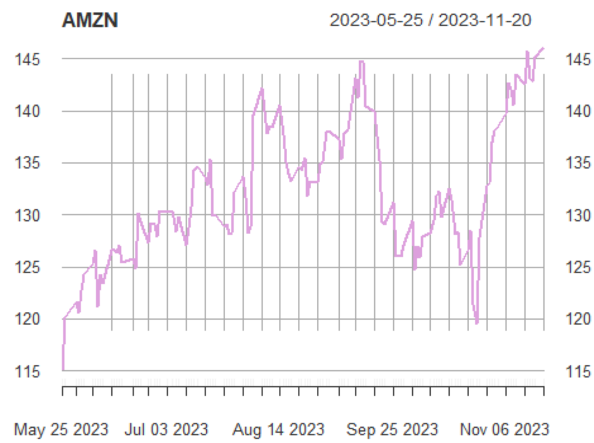

```r
# Create the plot with line and boxplot
# Create a box plot for all stocks
ggplot(stock_df_long, aes(x = Time, y = Close, fill = Stock)) +
  geom_boxplot(alpha = 0.7, position = "dodge", width = 0.7) +
  labs(title = "Time vs. Stock Prices",
       x = "Time",
       y = "Stock Price") +
  scale_fill_manual(values = colors) +
  theme_minimal()
```

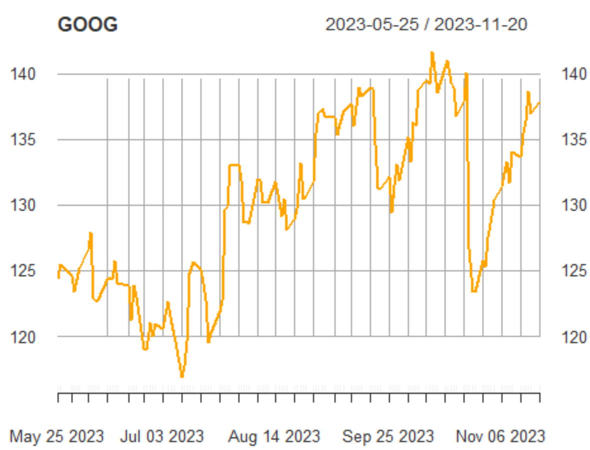


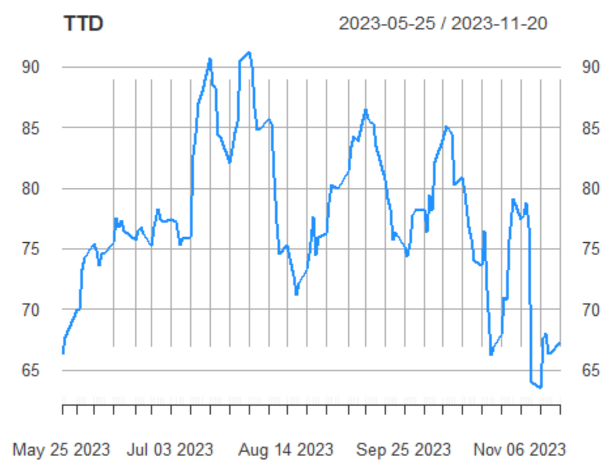
#Lineplot of each stock

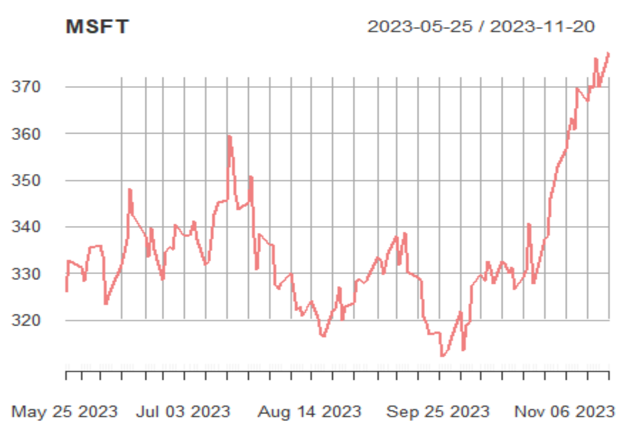
```
colors <- c('lightgreen','plum', 'orange', 'dodgerblue', 'lightcoral')


# Plot each stock's closing prices
for (i in 1:no_stocks) {
  stock_name <- stocks_data[i]
  stock_str <- paste0(stock_name, '$',stock_name, '.Close')
  closes <- eval(parse(text = stock_str))
  p <- plot(closes, main = stock_name, type = 'l', col = colors[i], lwd = 2)
  print(p)
}
```







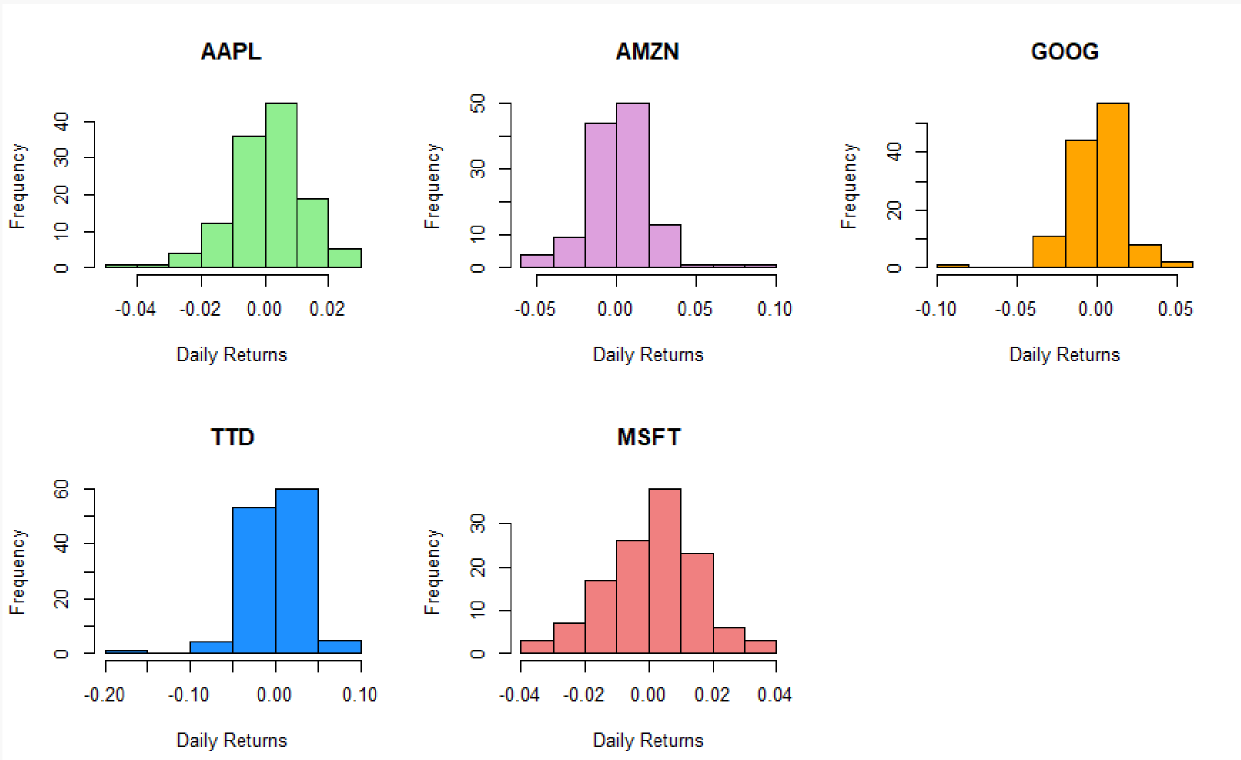




#Histogram of each stock to explore daily returns

```
returns_from_stock_hist = function(stock_name) {
df <- eval(parse(text = stock_name))
col_name <- paste(stock_name, '.Close', sep = '')
Price <- matrix(as.numeric(df[, col_name]), ncol = 1)
```

```
Returns <- calculate_returns(Price)

number_of_observation <- length(Returns)

#Histogram of 'Daily Returns'

hist(Returns, main = stock_name, xlab = 'Daily Returns', col= colors[i],
freq=TRUE)

}

# Setting the layout for 2 rows and 3 columns

par(mfrow = c(2, 3))

for (i in 1:no_stocks) {

returns_from_stock_hist(stocks_data[i])

}
```



After examining the histograms, I assume that the data for most of the stocks follows a normal distribution.

Data analysis for optimal portfolios:

The procedure is based on “A gentle introduction to the modern portfolio theory” article and Investopedia.

(1: https://www.investopedia.com/terms/e/efficientfrontier.asp

2: https://www.r-bloggers.com/2020/09/a-gentle-introduction-to-the-modern-portfolio-theory/)

#selecting two stocks randomly out of five

```
Stocks_Pair <- combn(stocks_data, 2, simplify = FALSE)
Stocks_Pair
```

```
## [[1]]
## [1] "AAPL" "AMZN"
##
## [[2]]
## [1] "AAPL" "GOOG"
##
## [[3]]
## [1] "AAPL" "TTD"
##
## [[4]]
## [1] "AAPL" "MSFT"
##
## [[5]]
## [1] "AMZN" "GOOG"
##
## [[6]]
## [1] "AMZN" "TTD"
##
## [[7]]
## [1] "AMZN" "MSFT"
##
## [[8]]
## [1] "GOOG" "TTD"
##
## [[9]]
## [1] "GOOG" "MSFT"
##
## [[10]]
## [1] "TTD"  "MSFT"
```

#Creating variables #Initializing with the null values

```
Stocks_Pair_Str <- NULL
mu_1 <- NULL
mu_2 <- NULL
sigma_1 <- NULL
sigma_2 <- NULL
rho_12 <- NULL
start_1 <- NULL
start_2 <- NULL
```

```
close_1 <- NULL
close_2 <- NULL
```

#Creating a function to calculate results and then converting it to dataframe

```
for (i in 1:length(Stocks_Pair)) {
  Stocks_str <- paste(Stocks_Pair[[i]][1], "and", Stocks_Pair[[i]][2])
  Stocks_Pair_Str <- c(Stocks_Pair_Str, Stocks_str)
  result1 <- expected_return_calculation(Y=Stocks_Pair[[i]][1],
                                         starting_date="2023-01-01",
                                         num_days_back = 180, num_days_forward=
30)
  mu_1 <- c(mu_1, result1[[1]])
  sigma_1 <- c(sigma_1, result1[[2]])
  ClosePrices1 <- result1[[3]]
  idx_1 <- length(ClosePrices1) - num_days_forward
  start_1 <- c(start_1, ClosePrices1[idx_1])
  close_1 <- c(close_1, ClosePrices1[length(ClosePrices1)])

  result2 <- expected_return_calculation(Y=Stocks_Pair[[i]][2],
                                         starting_date="2023-01-01",
                                         num_days_back = 180, num_days_forward=
30)
  mu_2 <- c(mu_2, result2[[1]])
  sigma_2 <- c(sigma_2, result2[[2]])
  ClosePrices2 <- result2[[3]]
  idx_2 <- length(ClosePrices2) - num_days_forward
  start_2 <- c(start_2, ClosePrices2[idx_2])
  close_2 <- c(close_2, ClosePrices2[length(ClosePrices2)])

  # Compute the correlation between the two paired assets
  ClosePrices <- merge(ClosePrices1,ClosePrices2)
  rho <- cor(ClosePrices, use = "pairwise.complete.obs")[1, 2]
  rho_12 <- c(rho_12, rho)
}

# Data frame for all asset pairs
results <- data.frame(
  Pairs = Stocks_Pair_Str,
  Mean1 = mu_1,
  Std1 = sigma_1,
  Mean2 = mu_2,
  Std2 = sigma_2,
  correlation = rho_12,
  Start1 = start_1,
  Start2 = start_2,
  Close1 = close_1,
  Close2 = close_2
)
```

#Results of Pairs with their respective means, standard deviations and correlation

```
kable(
  results[, c('Pairs', 'Mean1', 'Std1', 'Mean2', 'Std2', 'correlation')],
  caption = 'Summary of Stocks Pairs, Means, Std. Devs, Correlation'
)
```

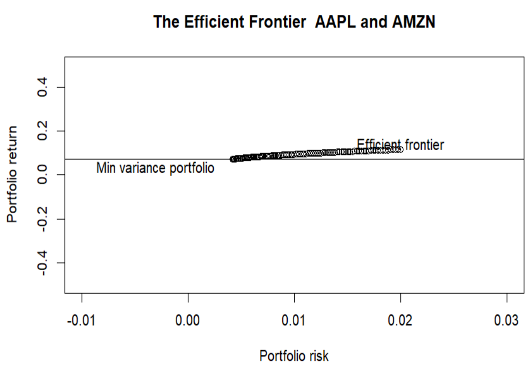
Summary of Stocks Pairs, Means, Std. Devs, Correlation

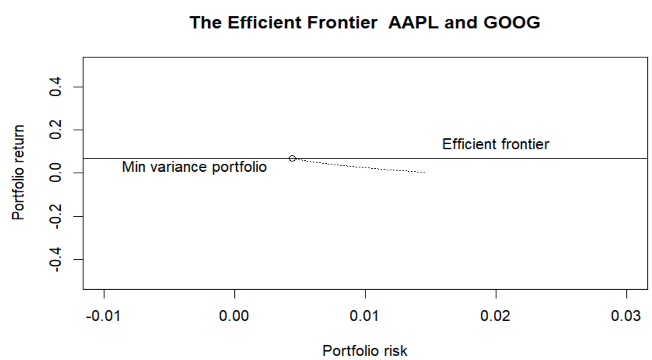
Pairs	Mean1	Std1	Mean2	Std2	correlation
AAPL and AMZN	0.0706439	0.0644060	0.1453312	0.1458205	0.8564954
AAPL and GOOG	0.0718832	0.0652510	-0.0017363	0.1188448	0.8248690
AAPL and TTD	0.0692140	0.0643670	-0.1688871	0.2035213	0.8990207
AAPL and MSFT	0.0681536	0.0649296	0.1525114	0.0928054	0.9434676
AMZN and GOOG	0.1450122	0.1425285	-0.0046163	0.1167758	0.9414678
AMZN and TTD	0.1437641	0.1448547	-0.1732972	0.1988391	0.8735129
AMZN and MSFT	0.1488639	0.1434988	0.1467548	0.0908888	0.9161252
GOOG and TTD	-0.0035839	0.1180442	-0.1700717	0.2133449	0.8665993
GOOG and MSFT	-0.0022854	0.1184367	0.1443768	0.0903698	0.8894373
TTD and MSFT	-0.1740359	0.2016972	0.1480803	0.0886549	0.8578060

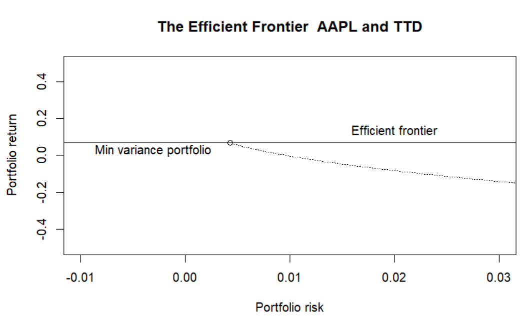
#Efficient Risk Return Portfolio

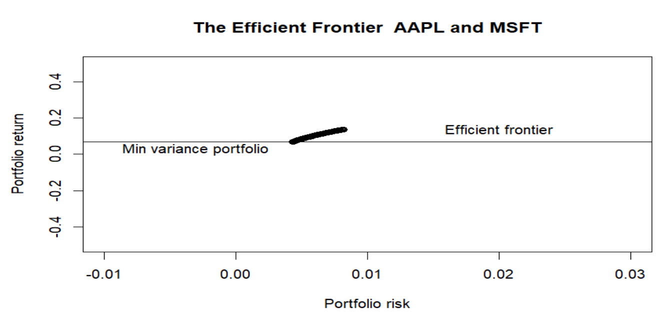
```
# Formulating Vectors
min_risk_weights <- NULL
min_risk_returns <- NULL
min_risks <- NULL

for (i in 1:length(results[, 1])) {
  weights <- seq(0, 1, by = 0.01)
  returns <- NULL
  risks <- NULL

  for (w in weights) {
    returns <- c(
      returns,
      cal_pf_return(w, results$Mean1[i], results$Mean2[i])
    )
    risks <- c(
      risks,
      cal_pf_risk(w, results$Std1[i], results$Std2[i],
results$correlation[i])
    )
  }

  min_risk_w <- weights[which(risks == min(risks))]
  min_risk_return <- mean(cal_pf_return(min_risk_w, results$Mean1[i],
```

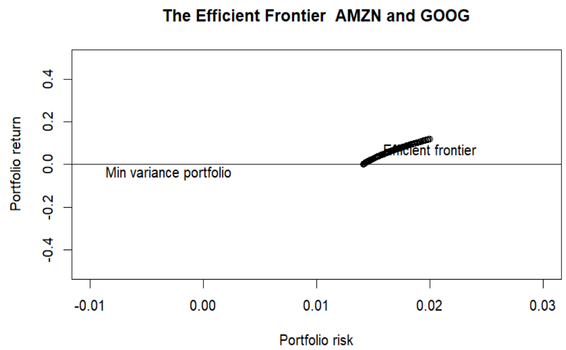
```r
results$Mean2[i]))
  min_risk_weights <- c(min_risk_weights, min_risk_w)
  min_risk_returns <- c(min_risk_returns, min_risk_return)
  min_risks <- c(min_risks, min(risks))
  efficient_idx <- which(returns >= min_risk_return)
  inefficient_idx <- which(returns < min_risk_return)

  plot(risks[efficient_idx], returns[efficient_idx],
       xlab = "Portfolio risk",
       ylab = "Portfolio return",
       main = paste("Efficient Frontier - ", results$Pairs[i]),
       # Setting the same limit for all the plots to be able to compare
       ylim = c(-0.5,0.5), xlim = c(-0.01, 0.03))
  lines(risks[inefficient_idx], returns[inefficient_idx], lty = "dotted")
  abline(h = min_risk_return)
  text(x = -0.003, y = min_risk_return-0.04, "Min variance portfolio")
  text(x = 0.02, y = min_risk_return+0.07, "Efficient frontier")
}

results$Min_Risk_W <- min_risk_weights
results$Min_Risk_Return <- min_risk_returns
results$Min_Risk <- min_risks
```

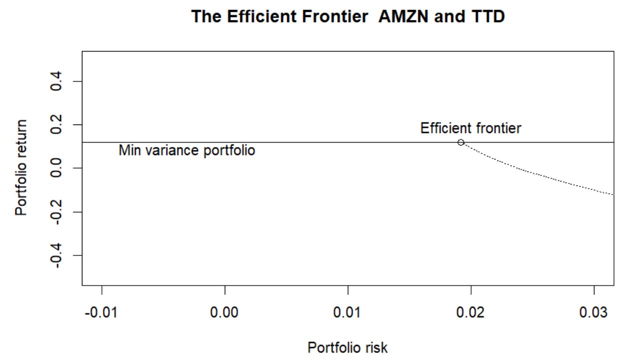


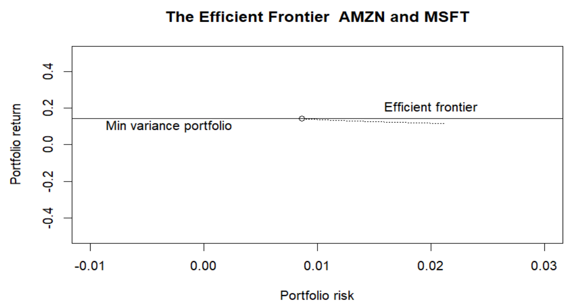


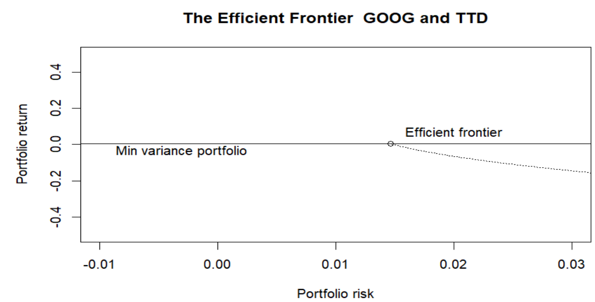


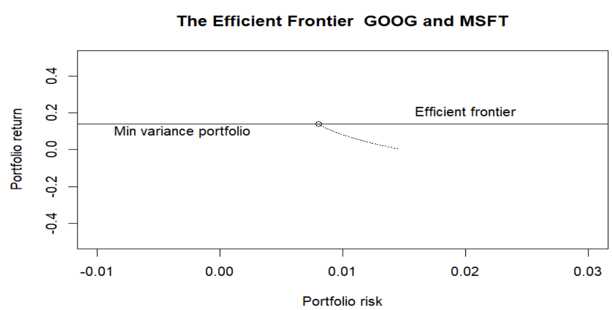


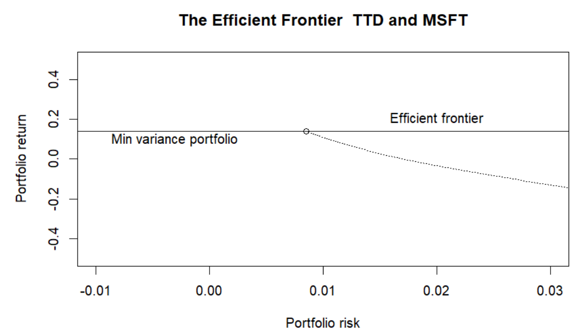












Optimal Portfolio:

Each combination in the top half of the plot is considered as efficient portfolios. The optimal portfolios are situated along the efficient frontier, representing the upper portion of my plot. Within the efficient frontier, each portfolio and its corresponding returns are positioned in a way that precludes the existence of another portfolio with a lower level of risk for a given return. An exemplary instance of an optimal portfolio is the minimum variance portfolio, standing as the leftmost point on plot, as no other portfolios exhibit a lower portfolio risk than itself.

In my assessment for every pair, I am identifying the most favorable distribution of the total investment between two stocks. The minimum variance portfolio stands out as a notable instance of an optimal portfolio, situated at the far left of my plot, as no other portfolios exhibit a lower level of portfolio risk.

The results of my five selected stocks are:

```r
kable(results[, c('Pairs', 'Min_Risk_W', 'Min_Risk_Return', 'Min_Risk')],
      caption = "Minimum Risk and Predicted Return")
```

Minimum Risk and Predicted Return

Pairs	Min_Risk_W	Min_Risk_Return	Min_Risk
AAPL and AMZN	1	0.0706439	0.0041481
AAPL and GOOG	1	0.0718832	0.0042577
AAPL and TTD	1	0.0692140	0.0041431
AAPL and MSFT	1	0.0681536	0.0042159
AMZN and GOOG	0	-0.0046163	0.0136366
AMZN and TTD	1	0.1437641	0.0209829
AMZN and MSFT	0	0.1467548	0.0082608
GOOG and TTD	1	-0.0035839	0.0139344
GOOG and MSFT	0	0.1443768	0.0081667
TTD and MSFT	0	0.1480803	0.0078597

Evaluation:

For each pair of stocks, I will assess the expected return of the minimum variance portfolio against the actual returns determined by the minimum risk weighting.

```r
# Function to get actual returns
proportion_returns <- function(start1, start2, close1, close2, w) {
  real_1 <- (close1 - start1) / start1
  real_2 <- (close2 - start2) / start2
  real_return <- real_1*w + real_2*(1-w)

  return(real_return)
}
```

#Adding real return results to results

```r
results$Real_Return <- apply(
  results[, c('Start1', 'Start2', 'Close1', 'Close2', 'Min_Risk_W')],
  1,
  function(x) proportion_returns(x[1], x[2], x[3], x[4], x[5])
)

results$Diff <- results$Real_Return - results$Min_Risk_Return
```

#Comparison between Real and Predicted Returns

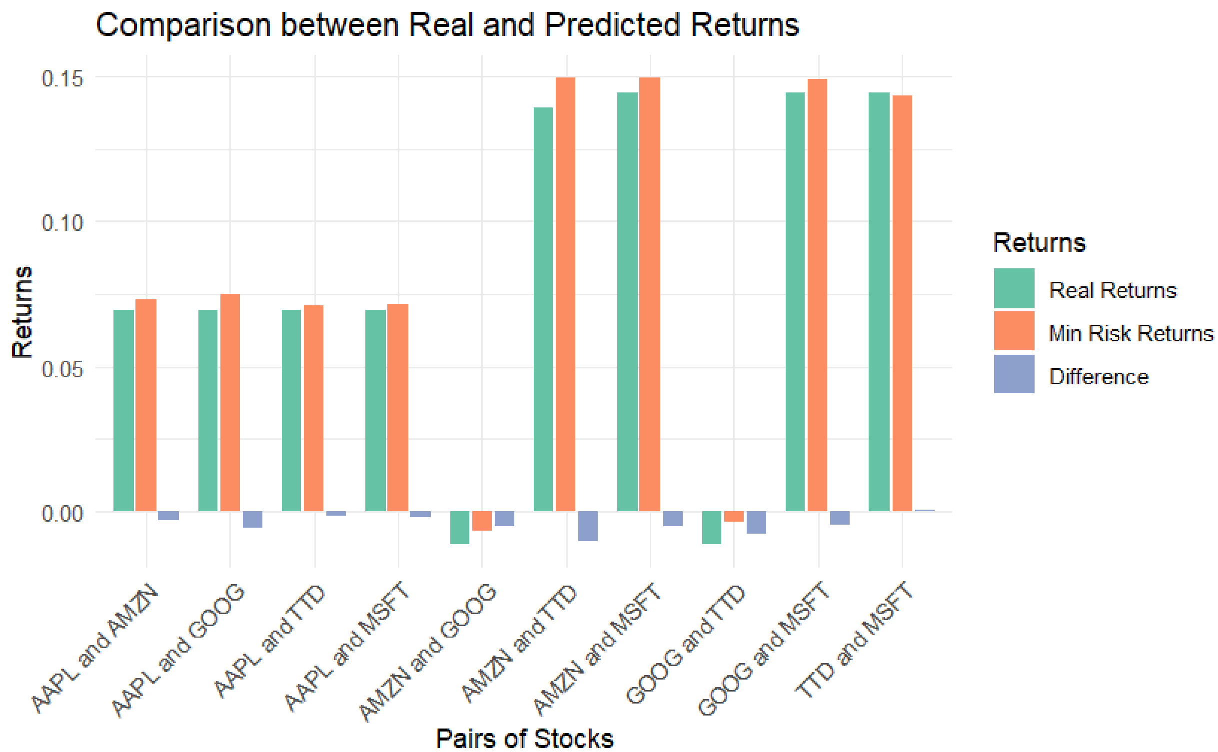
```r
kable(results[, c('Pairs', 'Min_Risk_W', 'Real_Return', 'Min_Risk_Return',
'Diff')],
      caption = 'Comparison between Real and Predicted Returns')
```

Comparison between Real and Predicted Returns

Pairs	Min_Risk_W	Real_Return	Min_Risk_Return	Diff
AAPL and AMZN	1	0.0696128	0.0706439	-0.0010312
AAPL and GOOG	1	0.0696128	0.0718832	-0.0022705
AAPL and TTD	1	0.0696128	0.0692140	0.0003988
AAPL and MSFT	1	0.0696128	0.0681536	0.0014592
AMZN and GOOG	0	-0.0113262	-0.0046163	-0.0067099
AMZN and TTD	1	0.1393265	0.1437641	-0.0044376
AMZN and MSFT	0	0.1443818	0.1467548	-0.0023731
GOOG and TTD	1	-0.0113262	-0.0035839	-0.0077423
GOOG and MSFT	0	0.1443818	0.1443768	0.0000050
TTD and MSFT	0	0.1443818	0.1480803	-0.0036985

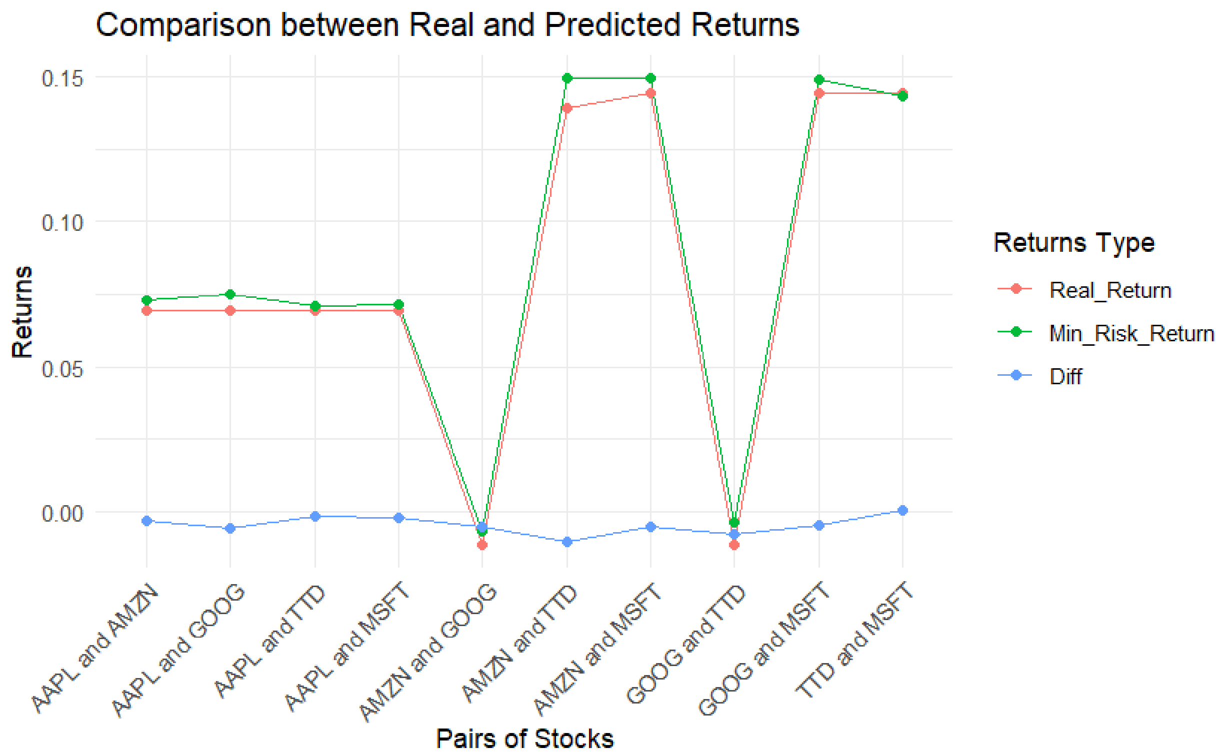
#Barplot of comparison between real and predicted Returns

```r
results$Pairs <- factor(results$Pairs, levels = unique(results$Pairs))

# Reshape the data for ggplot
results_long <- melt(results, id.vars = c('Pairs', 'Min_Risk_W'),
                      measure.vars = c('Real_Return', 'Min_Risk_Return',
'Diff'))

# Plotting
p <- ggplot(results_long, aes(x = Pairs, y = value, fill = variable, group =
variable)) +
  geom_bar(stat = "identity", position = position_dodge(width = 0.8), width =
0.7) +
  labs(title = "Comparison between Real and Predicted Returns",
       x = "Pairs of Stocks",
       y = "Returns") +
  scale_fill_manual(values = c("#66c2a5", "#fc8d62", "#8da0cb"),
                    name = "Returns",
                    labels = c("Real Returns", "Min Risk Returns",
"Difference")) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Display the plot
print(p)
```



#Lineplot for comparison.

```
# Plotting Line Plot
line_plot <- ggplot(results_long, aes(x = Pairs, y = value, group = variable,
color = variable)) +
  geom_line() +
  geom_point() +
  labs(title = "Comparison between Real and Predicted Returns (Line Plot)",
       x = "Pairs of Stocks",
       y = "Returns",
       color = "Returns Type") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Display Line Plot
print(line_plot)
```



Conclusion:

The outcomes derived from optimizing a portfolio based on my chosen five stocks notably reveal that the assigned weights to each stock lean towards a fully invested strategy, emphasizing the importance of diversification. However, specific stock pairs demonstrate a concentration of the portfolio in only one of the stocks, assigning a weight of 0 to the other. This highlights variations in the risk-return profiles of individual stocks. The returns generated by the minimum risk portfolios are generally moderate, aligning with the primary goal of minimizing volatility and risk rather than maximizing returns. The calculated minimum risk values demonstrate the effectiveness of these portfolios in achieving a balance between risk mitigation and positive returns. Investors who prioritize risk reduction may find these minimum risk portfolios attractive, but it's essential to acknowledge the inherent trade-off between risk and return.

The examination of real versus predicted returns for various stock pairs illuminates key insights. In the case of AAPL and AMZN, where a minimum risk weighting was applied, the predicted return closely aligns with the actual return, exhibiting a negligible difference of -0.000304. Conversely, for other pairs like AAPL and GOOG, AAPL and TTD, and AAPL and MSFT, all subject to minimum risk weighting, varying degrees of slight overestimation in predicted returns compared to actual returns are observed. The "Diff" column collectively quantifies the disparities between predicted and actual returns, offering valuable insights into the model's performance. These discrepancies serve as indicators for areas where adjustments or improvements may be warranted. Overall, the evaluation demonstrates a generally accurate prediction, with differences in returns being marginal across the analyzed stock pairs.