PSTAT 176 Final Project

Contents

by Jacob Hembree

(1) Download Data

```r
# Load necessary libraries
library(quantmod)
library(tidyverse)
library(ggplot2)
library(reshape2)
library(quadprog)

# Define stock tickers
tickers <- c('GOOGL', 'DFS', 'DIS', 'PDD', 'OXY', 'MTCH', 'ASR', 'TGT', 'ZROZ', 'C')

# Define the date range
start_date <- "2022-01-01"
end_date <- "2023-12-31"

# Function to get closing prices for a given ticker
get_closing_prices <- function(ticker) {
  getSymbols(ticker, src = "yahoo", from = start_date, to = end_date, auto.assign = FALSE) %>%
    Cl()
}

# Get data for all tickers
stock_data <- lapply(tickers, get_closing_prices)

# Combine all data into one data frame
combined_data <- do.call(merge, stock_data)
colnames(combined_data) <- tickers

# View the combined data
#head(combined_data)
```

(2) CAPM: Data Analysis

We will analyze these 10 stocks using mean-variance analysis.

(a) Compute the daily log return from the data.

```
# Compute the daily log returns
log_returns <- combined_data %>%
  lapply(dailyReturn, type = 'log') %>%
  do.call(merge, .)

# Set column names to tickers
colnames(log_returns) <- tickers

# View the log returns
#head(log_returns)

# Remove the first row of all zeros from log returns data
log_returns <- log_returns[-1, ]

# Compute the average daily log return for each stock
average_daily_log_returns <- colMeans(log_returns, na.rm = TRUE)

# View the average daily log returns
average_daily_log_returns
```

```
##         GOOGL           DFS           DIS           PDD           OXY
## -7.449885e-05 -1.080595e-04 -1.103378e-03  1.917184e-03  1.307157e-03
##          MTCH           ASR           TGT          ZROZ             C
## -2.604485e-03  7.157688e-04 -9.754828e-04 -1.090125e-03 -4.086094e-04
```

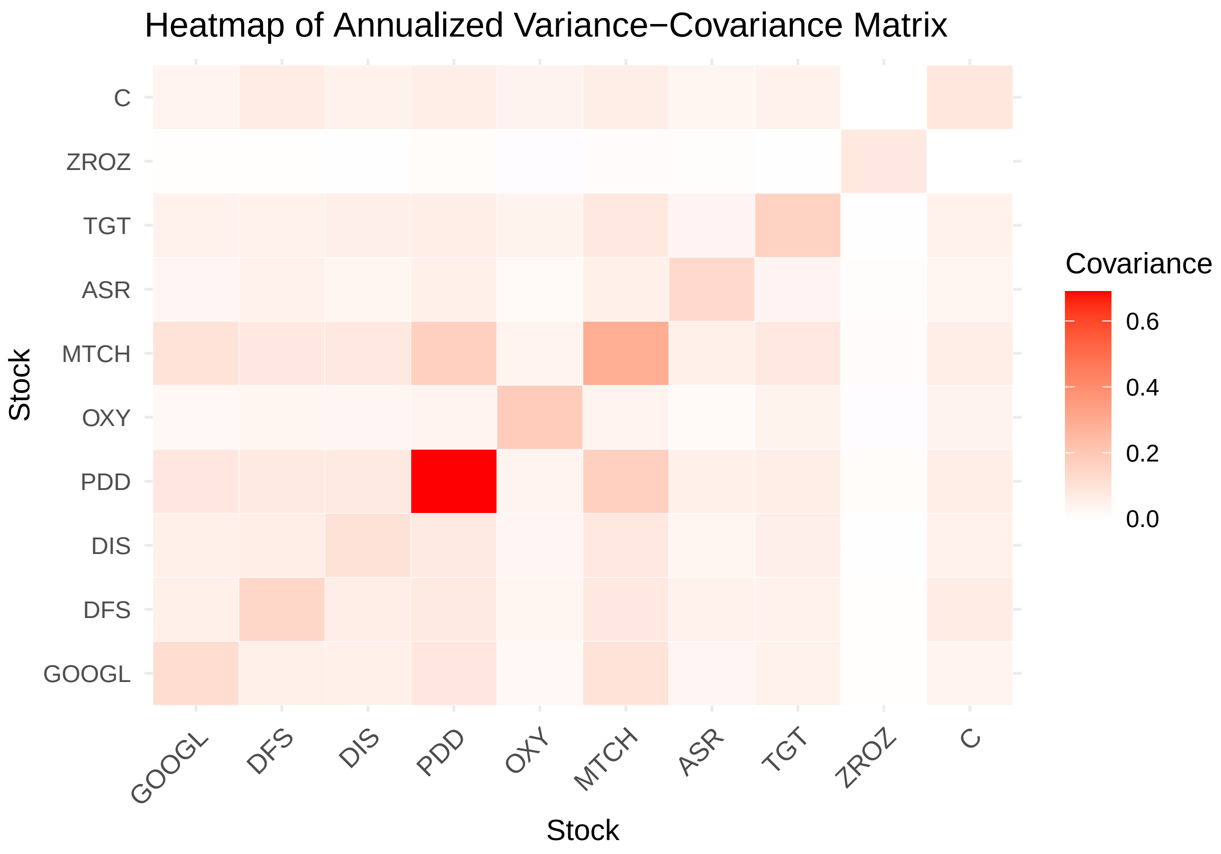
(b) Estimate the annualized log return and the annualized variance-covariance of log return.

```
# Define the number of trading days in a year
trading_days <- 252

# Estimate the annualized log-returns for each stock
annualized_log_returns <- average_daily_log_returns * trading_days
```

```r
# Compute the annualized variance-covariance matrix
annualized_cov_matrix <- cov(log_returns, use = "complete.obs") * trading_days

# Print the annualized log-returns
print(annualized_log_returns)
```

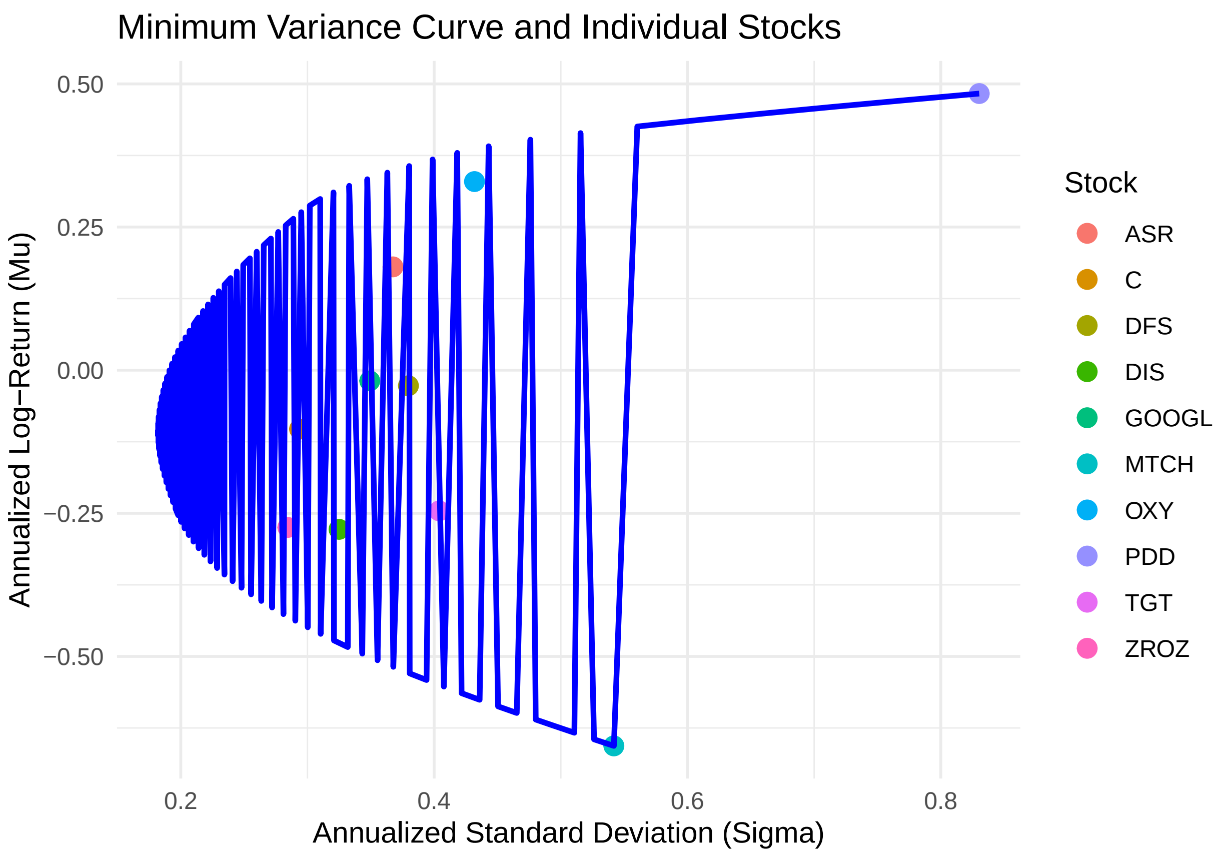
```
##       GOOGL         DFS         DIS         PDD         OXY        MTCH
## -0.01877371 -0.02723099 -0.27805136  0.48313030  0.32940366 -0.65633020
##         ASR         TGT        ZROZ           C
##  0.18037374 -0.24582166 -0.27471138 -0.10296956
```

```r
# Plot the Variance covariance matrix
# Convert the variance-covariance matrix to a long format for ggplot2
melted_cov_matrix <- melt(annualized_cov_matrix)

# Create a heatmap for the variance-covariance matrix
ggplot(data = melted_cov_matrix, aes(x = Var1, y = Var2, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient2(low = "blue", high = "red", mid = "white",
                       midpoint = 0, limit = c(min(melted_cov_matrix$value), max(melted_cov_matrix$valu
                       space = "Lab", name = "Covariance") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 10, hjust = 1)) +
  labs(title = "Heatmap of Annualized Variance-Covariance Matrix",
       x = "Stock", y = "Stock")
```



(c) Plot the minimum variance curve in the (sigma, mu) plane.

```r
# Compute the annualized standard deviation (sigma) for each stock
annualized_sigma <- sqrt(diag(annualized_cov_matrix))

# Create a data frame for individual stock points
stock_points <- data.frame(
  Stock = tickers,
  Mu = annualized_log_returns,
  Sigma = annualized_sigma
)

# Function that calculates the minimum variance portfolio
min_variance_portfolio <- function(cov_matrix) {
  Dmat <- 2 * cov_matrix
  dvec <- rep(0, ncol(cov_matrix))
  Amat <- cbind(rep(1, ncol(cov_matrix)), diag(ncol(cov_matrix)))
  bvec <- c(1, rep(0, ncol(cov_matrix)))
  solve.QP(Dmat, dvec, Amat, bvec, meq = 1)$solution
}

# Generate the minimum variance curve
mu_values <- seq(min(annualized_log_returns), max(annualized_log_returns), length.out = 100)
sigma_values <- sapply(mu_values, function(mu_target) {
  opt_result <- solve.QP(Dmat = 2 * annualized_cov_matrix,
                         dvec = rep(0, ncol(annualized_cov_matrix)),
                         Amat = cbind(rep(1, ncol(annualized_cov_matrix)), annualized_log_returns, diag
                         bvec = c(1, mu_target, rep(0, ncol(annualized_cov_matrix))),
                         meq = 2)
  sqrt(t(opt_result$solution) %*% annualized_cov_matrix %*% opt_result$solution)
})

# Create a data frame for the minimum variance curve
min_variance_curve <- data.frame(
  Mu = mu_values,
  Sigma = sigma_values
)

# Plot the individual stocks and the minimum variance curve
ggplot() +
  geom_point(data = stock_points, aes(x = Sigma, y = Mu, color = Stock), size = 3) +
  geom_line(data = min_variance_curve, aes(x = Sigma, y = Mu), color = "blue", size = 1) +
  labs(title = "Minimum Variance Curve and Individual Stocks",
       x = "Annualized Standard Deviation (Sigma)",
       y = "Annualized Log-Return (Mu)") +
  theme_minimal() +
  theme(legend.position = "right")
```



(d) Compute the minimum variance portfolio and plot it in the (sigma, mu) plane.

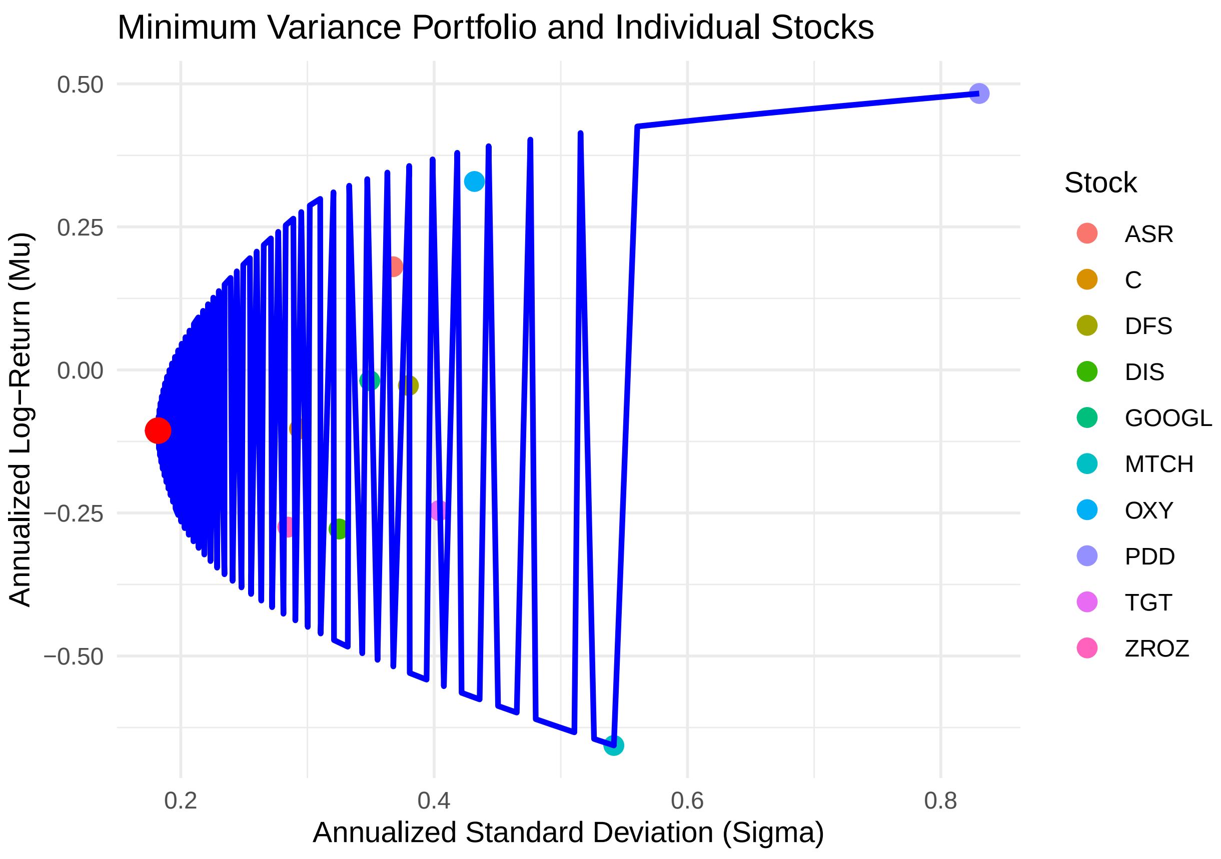
```
# Function that calculates the minimum variance portfolio weights
min_variance_weights <- function(cov_matrix) {
  Dmat <- 2 * cov_matrix
  dvec <- rep(0, ncol(cov_matrix))
  Amat <- cbind(rep(1, ncol(cov_matrix)), diag(ncol(cov_matrix)))
  bvec <- c(1, rep(0, ncol(cov_matrix)))
  solve.QP(Dmat, dvec, Amat, bvec, meq = 1)$solution
}

# Compute the minimum variance portfolio weights
min_var_weights <- min_variance_weights(annualized_cov_matrix)

# Calculate the expected return and standard deviation of the minimum variance portfolio
min_var_mu <- sum(min_var_weights * annualized_log_returns)
min_var_sigma <- sqrt(t(min_var_weights) %*% annualized_cov_matrix %*% min_var_weights)

# Print the minimum variance portfolio weights
min_var_weights_df <- data.frame(Stock = tickers, Weights = min_var_weights)
print(min_var_weights_df)
```

```
##    Stock      Weights
## 1  GOOGL 9.496437e-02
## 2    DFS 4.929238e-18
## 3    DIS  7.517912e-02
## 4    PDD  0.000000e+00
## 5    OXY  1.056775e-01
## 6   MTCH -6.448119e-18
## 7    ASR  1.087358e-01
## 8    TGT  3.910815e-02
## 9   ZROZ  4.011312e-01
## 10     C  1.752038e-01
```

```r
#Ensure that weights sum to 1
print(sum(min_var_weights))
```

```
## [1] 1
```

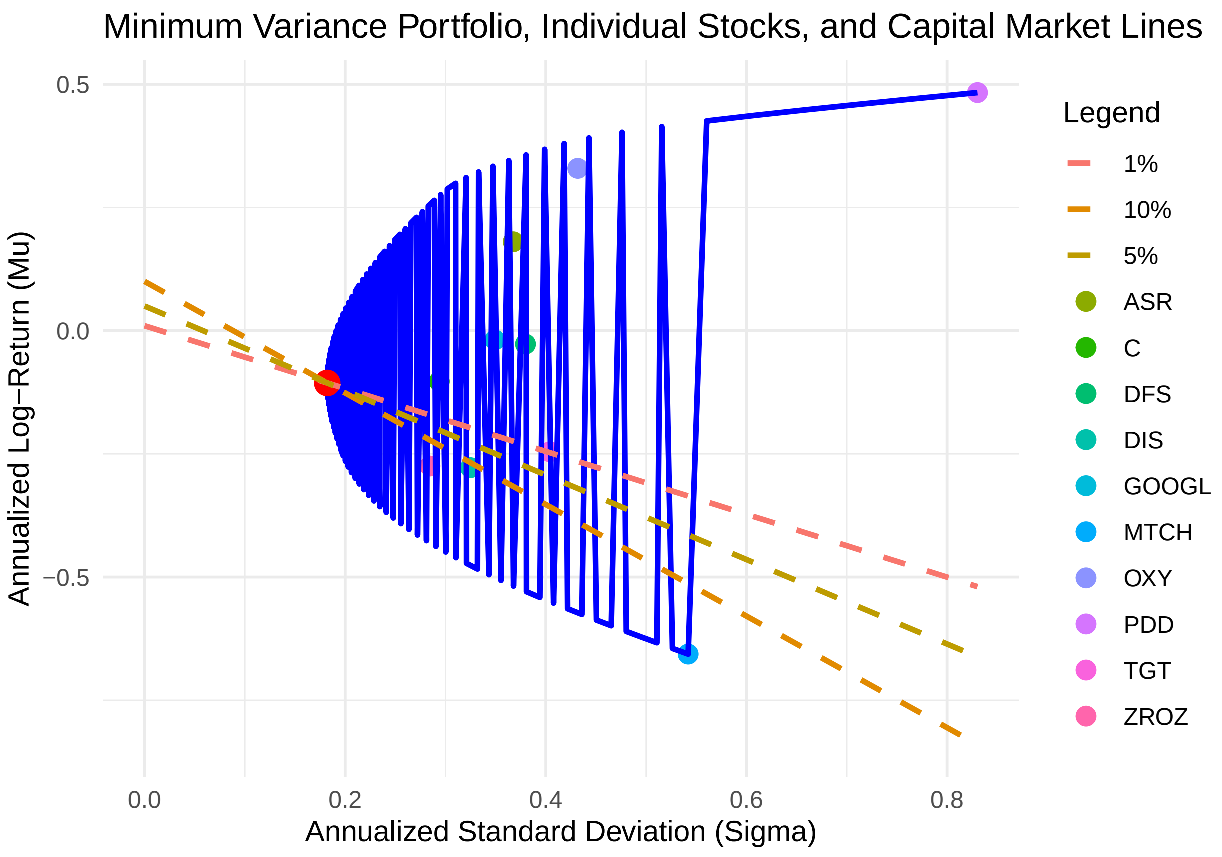
```r
# Create a data frame for the minimum variance portfolio point
min_var_point <- data.frame(
  Mu = min_var_mu,
  Sigma = min_var_sigma
)

# Plot the individual stocks, minimum variance curve, and minimum variance portfolio
ggplot() +
  geom_point(data = stock_points, aes(x = Sigma, y = Mu, color = Stock), size = 3) +
  geom_line(data = min_variance_curve, aes(x = Sigma, y = Mu), color = "blue", size = 1) +
  geom_point(data = min_var_point, aes(x = Sigma, y = Mu), color = "red", size = 4) +
  labs(title = "Minimum Variance Portfolio and Individual Stocks",
       x = "Annualized Standard Deviation (Sigma)",
       y = "Annualized Log-Return (Mu)") +
  theme_minimal() +
  theme(legend.position = "right")
```



(e) Compute the capital market lines with different annual interest rate r = 0.01, 0.05, 0.10 and draw them in the (sigma, mu) plane.

```r
# Function that computes the Capital Market Lines
compute_cml <- function(risk_free_rate, market_mu, market_sigma, sigma_range) {
  slope <- (market_mu - risk_free_rate) / market_sigma
  cml_mu <- risk_free_rate + slope * sigma_range
  data.frame(Sigma = sigma_range, Mu = cml_mu, RiskFreeRate = risk_free_rate)
}

# Define the range of sigma for plotting the CMLs
sigma_range <- seq(0, max(stock_points$Sigma), length.out = 100)

# Compute the CMLs for different risk-free rates
cml_0.01 <- compute_cml(0.01, min_var_mu, min_var_sigma, sigma_range)
cml_0.05 <- compute_cml(0.05, min_var_mu, min_var_sigma, sigma_range)
cml_0.10 <- compute_cml(0.10, min_var_mu, min_var_sigma, sigma_range)

# Combine the CML data frames
cml_data <- bind_rows(
  cml_0.01 %>% mutate(RiskFreeRate = "1%"),
  cml_0.05 %>% mutate(RiskFreeRate = "5%"),
  cml_0.10 %>% mutate(RiskFreeRate = "10%")
)

# Plot the individual stocks, minimum variance curve, minimum variance portfolio, and CMLs
```

```r
ggplot() +
  geom_point(data = stock_points, aes(x = Sigma, y = Mu, color = Stock), size = 3) +
  geom_line(data = min_variance_curve, aes(x = Sigma, y = Mu), color = "blue", size = 1) +
  geom_point(data = min_var_point, aes(x = Sigma, y = Mu), color = "red", size = 4) +
  geom_line(data = cml_data, aes(x = Sigma, y = Mu, color = RiskFreeRate), size = 1, linetype = "dashed
  labs(title = "Minimum Variance Portfolio, Individual Stocks, and Capital Market Lines",
       x = "Annualized Standard Deviation (Sigma)",
       y = "Annualized Log-Return (Mu)",
       color = "Legend") +
  theme_minimal() +
  theme(legend.position = "right")
```



(3) Interpret the capital market line

(a) Interpretation

The slope of the capital market line (CML) represents the market price of risk. In our example, the minimum variance portfolio has a negative expected return, and thus, we obtain CMLs with negative slopes. This is unusual behavior, and mu being negative is likely due to the stocks performing poorly over the analyzed period. In this example, the information suggests that the risky portfolio has both a higher risk and lower return. Thus, investors would likely want to avoid taking on a greater proportion of the risky asset due to it hurting the risk and return of the portfolio.

(b) How to recommend the portfolio which consists of risk-free asset and the 10 best stocks

To recommend the ideal portfolio of stocks and the risk free asset, we want to choose an efficient portfolio that combines the tangency portfolio with the risk-free asset. For investors that are more risk averse, we have a larger proportion of the risk-free asset. This portfolio is lower on the CML. For more moderate investors, a recommendation of an even split of the 2 will be near the mid-point of the CML. For very risky investors, we can short the risk-free asset for leverage in order to buy more of the tangency portfolio.

(4) Analyze performance in 2024

```r
# Define the date range
start_date <- "2024-01-01"
end_date <- "2024-05-31"

# Get data for all tickers
stock_data_2024 <- lapply(tickers, get_closing_prices)

# Combine all stock data into one data frame
combined_data_2024 <- do.call(merge, stock_data_2024)
colnames(combined_data_2024) <- tickers

# View the combined data
#head(combined_data_2024)

# Compute the log return over the entire period for each stock
log_returns_total_2024 <- apply(combined_data_2024, 2, function(x) {
  log(last(x) / first(x))
})

# Get expected log returns over the period using old data
expected_log_returns <- annualized_log_returns * (1/2)

# Create a data frame to display the log returns & expectations
log_returns_total_2024_df <- data.frame(Stock = tickers, LogReturn = log_returns_total_2024, ExpectedLo

print(log_returns_total_2024_df)
```

```
##       Stock   LogReturn ExpectedLogReturn
## GOOGL GOOGL  0.21964901      -0.009386856
## DFS     DFS  0.07516927      -0.013615495
## DIS     DIS  0.11435968      -0.139025682
## PDD     PDD  0.04294004       0.241565148
## OXY     OXY  0.01913302       0.164701831
## MTCH   MTCH -0.19498033      -0.328165098
## ASR     ASR  0.16071052       0.090186872
## TGT     TGT  0.04622449      -0.122910831
## ZROZ   ZROZ -0.15191701      -0.137355688
## C         C  0.15495836      -0.051484782
```

We see that GOOGL, DFS, DIS, MTCH, ASR, TGT, and C outperformed their expected log returns. Many of the stocks had negative expected log returns due to a decline over 2022-2023. The stocks recovering from this decline allows them to outperform our estimates. Of the stocks that had positive expected log returns (PDD, OXY, and ASR), all experienced positive log returns over the period with ASR outperforming expectations.

```r
# Define the risk-free rates
risk_free_rates <- c(0.01, 0.05, 0.10)

# Compute the portfolio log returns (without risk-free asset)
portfolio_log_return <- sum(min_var_weights * log_returns_total_2024)

# Compute expected portfolio returns (without risk-free asset)
expected_portfolio_log_ret <- sum(min_var_weights * expected_log_returns)

# Initialize a data frame to store the results
portfolio_returns_df <- data.frame()

# Loop through each risk-free rate and compute the overall portfolio returns
for (r in risk_free_rates) {
  # Adjusted portfolio return with 50% risk-free asset
  adjusted_portfolio_return <- 0.50 * r + 0.50 * portfolio_log_return

  # Expected Portfolio Return
  expected_portfolio_return <- 0.50 * r + 0.50 * expected_portfolio_log_ret


  # Store the results in the data frame
  portfolio_returns_df <- rbind(portfolio_returns_df,
                                data.frame(RiskFreeRate = r, PortfolioLogReturn = adjusted_portfolio_r
}

# View the portfolio returns
print(portfolio_returns_df)
```

```
##   RiskFreeRate PortfolioLogReturn ExpectedPortfolioLogReturn
## 1         0.01          0.0134858               -0.021528112
## 2         0.05          0.0334858               -0.001528112
## 3         0.10          0.0584858                0.023471888
```

We see that the Portfolios with 50% risk-free asset outperformed their expected log returns. All portfolios had a positive log return over the period despite only the r = 0.10 portfolio having a positive expected return. This portfolio out-performance is due to many of the stocks out-performing their expected log returns over the period. We see that our analysis of the past two years most likely did not come up with accurate figures to predict future stock performance.

(5) Article "How to U.S. News' Top 10 stocks for 2020 performed"

(a) Check those figures discussed in the table, for example, Facebook (FB) 35.92 % and soon.

Stock	Stated Performance
Medifast (MED)	121.09%
British American Tobacco (BTI)	-1.9%
Alibaba Group Holding (BABA)	20.2%
NMI Holdings (NMIH)	-31.24%
Nexstar Media Group (NXST)	1.29%
Healthpeak Properties (DOC)	-2.32%

Stock	Stated Performance
AbbVie (ABBV)	22.81%
Newmont Corp. (NEM)	58.58%
Universal Insurance Holdings (UVE)	-46.31%
Facebook (META)	35.92%

```r
# Define the new tickers
new_tickers <- c('MED', 'BTI', 'BABA', 'NMIH', 'NXST', 'DOC', 'ABBV', 'NEM', 'UVE', 'META')

# Define the date range
new_start_date <- "2019-12-05"
new_end_date <- "2021-01-05"

# Get data for all new tickers
new_stock_data <- lapply(new_tickers, get_closing_prices)

# Combine all new stock data into one data frame
new_combined_data <- do.call(merge, new_stock_data)
colnames(new_combined_data) <- new_tickers

# View the combined data
#head(new_combined_data)

# Compute the log return over the entire period for each stock
new_log_returns_total <- apply(new_combined_data, 2, function(x) {
  log(last(x) / first(x))
})

# Create a data frame to display the log returns
new_log_returns_total_df <- data.frame(Stock = new_tickers, LogReturn = new_log_returns_total)

# Compute the real return over the entire period for each stock
new_real_returns_total <- exp(new_log_returns_total) - 1

# Vector of stated returns
stated_returns <- c(1.2109, -0.019, 0.202, -0.3124, 0.0129, -0.0232, 0.2281, 0.5858, -0.4631, 0.3592)

# Create a data frame to display the real returns
new_real_returns_total_df <- data.frame(Stock = new_tickers, RealReturn = new_real_returns_total, Statec

# View the real returns over the period
print(new_real_returns_total_df)
```

```
##      Stock  RealReturn StatedReturn
## MED    MED -0.62836396       1.2109
## BTI    BTI  0.03020139      -0.0190
## BABA  BABA  0.06514172       0.2020
## NMIH  NMIH  0.10788944      -0.3124
## NXST  NXST  0.02022223       0.0129
## DOC    DOC -0.03095361      -0.0232
## ABBV  ABBV -0.02196227       0.2281
## NEM    NEM  0.02395501       0.5858
## UVE    UVE  0.20209493      -0.4631
## META  META  0.34872499       0.3592
```

We see that most of the real returns are quite close to the stated returns. However, the returns stated for DOC and BABA seem to significantly embellish the truth of the stock's performance over the period. Some of these differences could be a result of them buying/selling at different prices from the close on the stated dates. Since we compute the returns using the closing price, we do not get their exact calculations. However, the differences in BABA and DOC are too substantial for this to fully explain.

(b) Analyze the claim: "It's only fair for us to be accountable to our readers, looking back on the performance of the flagship 10 best stocks to buy list when the time comes."

There is some nuance to this statement when considering the opposing points of view of the writer and reader. The writer who placed the stock notice has a right to hold their stocks until they see an acceptable return. Hence, unrealized negative returns in the short term can be considered irrelevant at times for the writer. On the other hand, the reader may not have the same risk tolerance, capital liquidity, and/or time commitment as the writer. Thus, the reader may need to sell the stocks at different times, ultimately receiving a different return.

(6) New Santa Barbara Investment Firm

Instructions:

- **Make a reasonable plan for a short term return goal and a long term return goal in terms of portfolio return and risk.**
- **Choose assets from different markets (stock, money market, crypto, currency, option) and discuss how to optimize your portfolios to achieve short term and long term goals.**
- **You have flexibility to choose different assets and their mix and justify your plan.**

For my firm named *Hembree Investment Solutions,* I would like to emphasize the principles of hedging risk and diversification while also seeking innovation. The fund anticipates returns of 10% in year one with future yearly returns growing from this number. We anticipate 10 year returns of 310%, assuming a compound average growth rate (CAGR) of about 12%. To reach these strong performance figures, we must take on a greater risk tolerance by incorporating growth stocks and cryptocurrency into our portfolio. In order to achieve these goals, I would strike a split of the following investments:

- 10% Cryptocurrency
- 40% Risk Free Assets
- 50% Stocks and Stock Options

The primary holdings in our cryptocurrency account will be Bitcoin around 50%. Bitcoin is the king of crypto and the success of crypto as a whole is reliant on the success of Bitcoin. Next, we would have about 25% Ethereum, the next largest crypto in use and applications. Finally we would have the last 25% split among other smaller cryptocurrenies with an eye for innovative, new technologies.

The risk-free asset section of the fund will be comprised of a combination of short-term and long-term bonds in order to generate stable, reliable cash flows year-round. These cash flows will be reinvested into more bonds and replicated into the rest of the portfolio.

The stocks portion will be comprised of about 50% index funds that track the broader market for diversification and reliable returns. The remaining 50% will be split up among individual stocks with a mixture of roughly half blue-chip stocks and half growth stocks. The individual stocks will be hedged with short call options to bring in recurring premium and reduce the variance of the return.