

FINC 780: Mean-Variance Optimal Risky Portfolio Using Simulation

B. Umesh, B.L. Dang, M. Hsu, S. Rajesh Panicker, J. Sarun

November 2nd, 2024

1. Introduction

In finance, portfolios can be summed up to be any collection of investments. These generally include stocks, bonds, cash, but can include other assets. However, these portfolios are designed to generate some form of income: the degree of which is dependent on how risk one is willing to take. In order to provide an objective, we are going to set out to calculate our portfolio risks by determining Variance, as well as the Sharpe ratio. The Sharpe ratio is a particularly strong measurement as it measures the performance in comparison to a risk free security: therefore providing us the amount of return we may see from a high risk asset.

In this assignment, we are tasked to build R functions to analyse stock return data, to output mean-variance metrics from simulated portfolios. Furthermore, some degree of visualization is included, in order to make it easier to see what kind of data we are working with.

2. Building the Function myMeanVarPort

In order to calculate any of the information from the stocks, we need to build our function to do the actual calculation to get the metrics from the stocks. In order to do so, we are going to pull our data, and calculate returns, as these are the metrics we will be evaluating the stocks on. To do so, we're going to be calculating the risk ratios and Sharpe ratios.

While generally we would include the loading of libraries, we can include that in the setup of the markdown file. The libraries we are using are; quantmod, dplyr, ggplot2, and ggrepel.

```
myMeanVarPort <- function(tickers, start_date, end_date, risk_free_rate) {  
  
  # Data acquisition: get stock data from Yahoo  
  getSymbols(tickers, src = 'yahoo', from = start_date, to = end_date, periodicity = 'monthly')  
  
  # Calculate monthly returns for each ticker  
  stock_returns <- lapply(tickers, function(ticker) {  
    monthly_prices <- Ad(get(ticker))  
    monthly_returns <- periodReturn(monthly_prices, period = 'monthly', type = 'log')  
    colnames(monthly_returns) <- ticker  
    return(monthly_returns)  
  })  
  
  # Combine monthly returns into a single data frame  
  combined_returns <- do.call(merge, stock_returns) %>% na.locf() %>% na.omit()  
  
  # Calculate mean returns and covariance matrix
```

```

mean_returns <- colMeans(combined_returns, na.rm = TRUE)
cov_matrix <- cov(combined_returns, use = "complete.obs")

# Simulate portfolios
set.seed(12)
num_assets <- length(tickers)
num_portfolios <- 100 * num_assets
simulated_portfolios <- replicate(num_portfolios, {
  weights <- runif(num_assets)
  weights <- weights / sum(weights) # Normalize weights to sum to 1
  portfolio_return <- sum(weights * mean_returns)
  portfolio_risk <- sqrt(t(weights) %*% cov_matrix %*% weights)
  sharpe_ratio <- (portfolio_return - risk_free_rate) / portfolio_risk
  c(weights, portfolio_return, portfolio_risk, sharpe_ratio)
})

# Convert simulated portfolios to a data frame
portfolio_data <- as.data.frame(t(simulated_portfolios))
colnames(portfolio_data) <- c(paste0("Weight_", tickers), "Mean_Return", "Risk", "Sharpe_Ratio")

# Return the results as a list
result <- list(
  mean_returns = mean_returns,
  cov_matrix = cov_matrix,
  portfolio_data = portfolio_data
)

return(result)
}

```

Within this function, our goal is to provide the inputs of: - Ticker (or the stock name) - Start Date - End Date - Risk Free Rate. In return, the output of “result” should generate a list that includes the mean returns, the covariance matrix, as well as additional data for us to visualize and plot the security.

Additionally, we need to simulate a portfolio. By using a set seed of 12, we can continuously reproduce a generic data set to use our functions on. We’re going to repeat this 100 times so we have enough data to put onto our plot, and then calculate the risk.

2. Building our Plot Function

In this part, we want to take all of our portfolio and plot it into a graph so we can see where each asset sits in regards to the amount of expected return, as well as the risk of the asset. We also want to include the optimal portfolio and minimum variance points, so we will include those as points that we calculate using the risk compared to the return. These points will be highlighted in red and green once we generate the plot.

```

plot_efficient_frontier <- function(portfolio_data) {
  # Find the optimal portfolio (highest Sharpe ratio)
  optimal_portfolio <- portfolio_data[which.max(portfolio_data$Sharpe_Ratio), ]

  # Find the minimum variance portfolio
  min_var_portfolio <- portfolio_data[which.min(portfolio_data$Risk), ]
}

```

```

# Create the plot
p <- ggplot(portfolio_data, aes(x = Risk, y = Mean_Return)) +
  # Plot all portfolios
  geom_point(alpha = 0.5, color = "darkblue", size = 1) +

  # Add optimal portfolio point
  geom_point(data = data.frame(Risk = optimal_portfolio$Risk, Mean_Return = optimal_portfolio$Mean_Return),
            alpha = 0.5, color = "red", size = 1) +

  # Add minimum variance portfolio point
  geom_point(data = data.frame(Risk = min_var_portfolio$Risk, Mean_Return = min_var_portfolio$Mean_Return),
            alpha = 0.5, color = "green", size = 1) +

  geom_label_repel(
    data = data.frame(
      Risk = c(optimal_portfolio$Risk, min_var_portfolio$Risk),
      Mean_Return = c(optimal_portfolio$Mean_Return, min_var_portfolio$Mean_Return),
      Label = c(
        sprintf("Optimal Portfolio\nSharpe Ratio: %.4f", optimal_portfolio$Sharpe_Ratio),
        sprintf("Minimum Variance Portfolio\nRisk: %.4f", min_var_portfolio$Risk)
      )
    ),
    aes(label = Label),
    box.padding = 0.5,
    force = 3
  ) +
  theme_minimal() +

  labs(
    title = "Portfolio Optimization Results",
    subtitle = "Efficient Frontier with Optimal and Minimum Variance Portfolios",
    x = "Portfolio Risk (Standard Deviation)",
    y = "Expected Return",
    caption = "Note: Red point indicates optimal portfolio, green point indicates minimum variance portfolio"
  ) +
  theme(
    plot.title = element_text(face = "bold"),
    plot.subtitle = element_text(size = 10),
    axis.title = element_text(face = "bold")
  )

return(p)
}

```

3. Portfolio Details Function

In this function, we want to output the information about the ideal portfolios. This means that we have to generate two tables that are going to correspond with being the most optimal (that of the highest Sharpe ratio), and the one with the least amount of risk.

```

print_portfolio_details <- function(portfolio_data, tickers) {
  # Find optimal portfolio
  optimal_idx <- which.max(portfolio_data$Sharpe_Ratio)
  optimal_portfolio <- portfolio_data[optimal_idx, ]
}

```

```

# Find minimum variance portfolio
min_var_idx <- which.min(portfolio_data$Risk)
min_var_portfolio <- portfolio_data[min_var_idx, ]

# Print optimal portfolio details
cat("\nOptimal Portfolio Details:\n")
cat("-----\n")
cat("Expected Return:", round(optimal_portfolio$Mean_Return * 100, 2), "%\n")
cat("Risk:", round(optimal_portfolio$Risk * 100, 2), "%\n")
cat("Sharpe Ratio:", round(optimal_portfolio$Sharpe_Ratio, 4), "\n")
cat("Weights:\n")
for(i in 1:length(tickers)) {
  cat(sprintf("%s: %.2f%\n", tickers[i], optimal_portfolio[[paste0("Weight_", tickers[i])]] * 100))
}

# Print minimum variance portfolio details
cat("\nMinimum Variance Portfolio Details:\n")
cat("-----\n")
cat("Expected Return:", round(min_var_portfolio$Mean_Return * 100, 2), "%\n")
cat("Risk:", round(min_var_portfolio$Risk * 100, 2), "%\n")
cat("Sharpe Ratio:", round(min_var_portfolio$Sharpe_Ratio, 4), "\n")
cat("Weights:\n")
for(i in 1:length(tickers)) {
  cat(sprintf("%s: %.2f%\n", tickers[i], min_var_portfolio[[paste0("Weight_", tickers[i])]] * 100))
}
}

```

4. Results

Now that we have all of our functions ready, all that's left is to input our theoretical stocks, their respective dates, and our risk free rate. As given in the assignment, we will be inputting: Tickers: GE, XOM, GBX, SBUX, PFE, HMC, and NVDA Start Date: 20140101 (Which we will convert to 2014-01-01) End Date: 20171231 (Which we will convert to 2017-12-31) Risk Free Rate: .02

Once we have all this information input, all that's left is for us to visualize our plot and output our tables.

```

#Part 4: Execution Code
# Set parameters
start_date <- as.Date("2014-01-01")
end_date <- as.Date("2017-12-31")
risk_free_rate <- 0.02
tickers <- c("GE", "XOM", "GBX", "SBUX", "PFE", "HMC", "NVDA")

# Pre-download the data
for(ticker in tickers) {
  getSymbols(ticker, src = 'yahoo', from = start_date, to = end_date)
}

# Run the analysis
results <- myMeanVarPort(tickers, start_date, end_date, risk_free_rate)

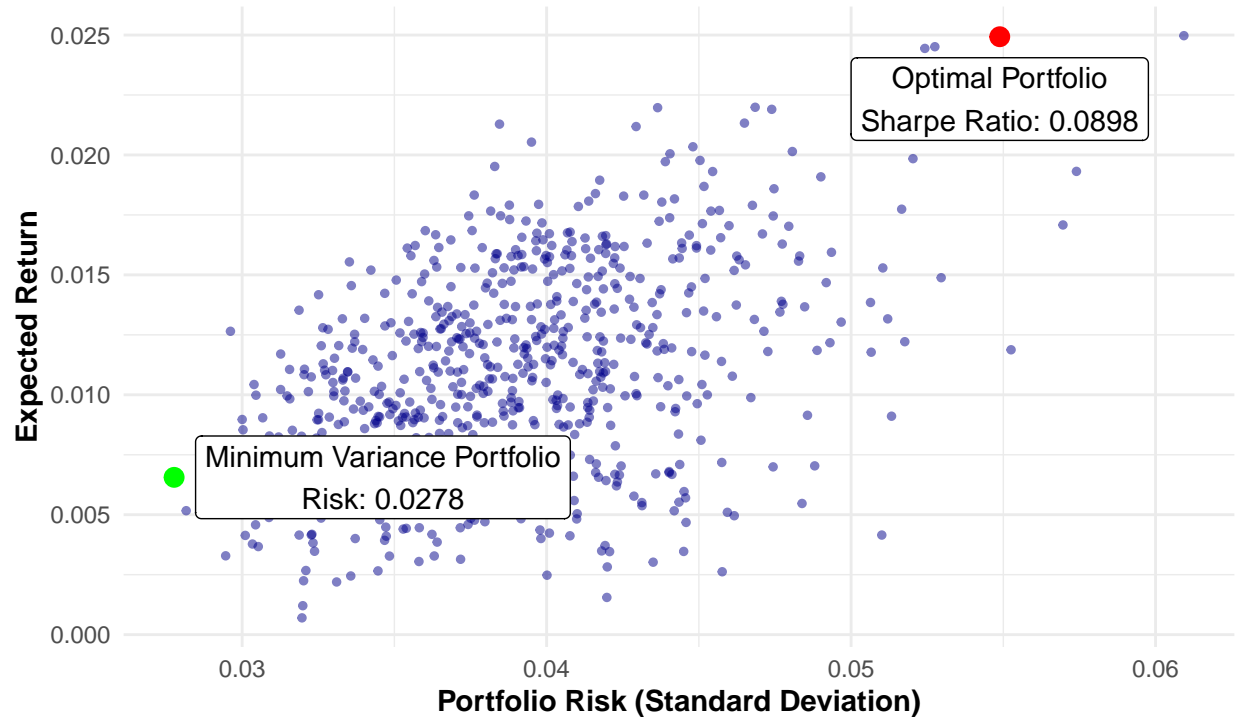
# Create and display the plot
plot <- plot_efficient_frontier(results$portfolio_data)

```

```
print(plot)
```

Portfolio Optimization Results

Efficient Frontier with Optimal and Minimum Variance Portfolios



Note: Red point indicates optimal portfolio, green point indicates minimum variance portfolio

```
# Print portfolio details
print_portfolio_details(results$portfolio_data, tickers)
```

```
##
## Optimal Portfolio Details:
## -----
## Expected Return: 2.49 %
## Risk: 5.49 %
## Sharpe Ratio: 0.0898
## Weights:
## GE: 1.31%
## XOM: 0.40%
## GBX: 17.39%
## SBUX: 0.24%
## PFE: 24.68%
## HMC: 15.00%
## NVDA: 40.98%
##
## Minimum Variance Portfolio Details:
## -----
## Expected Return: 0.66 %
## Risk: 2.78 %
## Sharpe Ratio: -0.4841
```

```
## Weights:  
## GE: 2.45%  
## XOM: 24.89%  
## GBX: 0.86%  
## SBUX: 27.44%  
## PFE: 17.52%  
## HMC: 22.45%  
## NVDA: 4.39%
```

Now that we have our options, we can clearly see what our portfolio should look like if we're optimizing either based on variance or Sharpe ration. Some of the strengths that come with the Sharpe ratio is the consideration of diversity. By evaluating assets in comparison to one another, it's possible for us to diversify our portfolio in order to reduce risk. An additional strength of the Sharpe ratio is that we can focus on total return, allowing us to gain a clear big picture of the portfolio.

However, one of the glaring weaknesses of using this method is that it doesn't account for futures: this works because we are looking at previous (albeit simulated) portfolios. Another weakness is the reliance on the risk-free rate. If this rate changes drastically, the Sharpe ratio is a less reliable source to base judgement from.

A different method used to evaluate portfolios is that Mean-Variance Optimization. This strategy has a strength in which we can more effectively allow for an investor to determine how much risk they would like: as opposed to the Sharpe ratio which must include the risk AND the returns.

Regardless of what method we choose to use, portfolio optimization allows for an analyst to consider many different factors, with the ultimate goal of trying to make more money, or make a portfolio more secure. Regardless of what technique may be implemented, the overall common denominator is that diversification is a core part of portfolios; by diversifying more, it is possible to reduce risk, improve returns, or ideally, a combination of both.