Optimizing Portfolios: A Simulation-Driven Approach to Risk and Return

Parth Chaudhari (Group 1)

Contents

1 Introduction

Portfolio optimization is a critical task for portfolio managers, aimed at finding the optimal weights of selected securities in a way that the resulting portfolio is mean-variance efficient. Unlike portfolio selection, which is concerned with choosing specific assets, portfolio optimization focuses on the optimal structuring of those assets to achieve the best risk-adjusted returns.

This report explores a simulation-based approach to portfolio optimization. The simulation method generates random portfolio weights and evaluates them in mean-variance space. The optimal portfolio is identified as the one with the highest Sharpe ratio, representing the best return per unit of risk.

1.1 Objectives

1. Calculate mean-variance metrics for a collection of selected stocks.
2. Simulate multiple portfolios and identify the optimal and minimum variance portfolios.
3. Plot simulated portfolios with the efficient frontier and capital market line (CML).
4. Provide insights into the optimal portfolio configuration and its risk-return profile.

2 Data and Methodology

2.0.1 Stock Data

The analysis uses monthly adjusted stock returns for the following tickers:

- **General Electric (GE)**
- **Exxon Mobil (XOM)**
- **Greenbrier Companies (GBX)**
- **Starbucks (SBUX)**
- **Pfizer (PFE)**
- **Honda Motor Company (HMC)**
- **NVIDIA (NVDA)**

The data spans from January 2014 to December 2017. Monthly returns were annualized to calculate mean returns and standard deviations.

2.0.2 Simulation Methodology

We simulated **700 portfolios**, where each portfolio's weights were randomly generated and normalized to sum to 1. Key metrics were computed for each portfolio:

- **Mean Return**: The weighted average of stock returns in each portfolio.
- **Risk (Standard Deviation)**: The portfolio's overall volatility, derived from the covariance matrix of stock returns.
- **Sharpe Ratio**: The risk-adjusted return of the portfolio, calculated as `(Mean Return - Risk-Free Rate) / Standard Deviation`.

The risk-free rate for this analysis is set at **0.02 (2%)**.

The function myMeanVarPort:

- Retrieves historical stock returns for each selected stock.
- Calculates annualized average returns, standard deviations, and covariance between assets.
- Simulates multiple portfolios with random weights and computes mean return, risk, and Sharpe Ratio for each.
- Identifies the optimal and minimum variance portfolios.
- Plots the Efficient Frontier, Capital Market Line, and highlights optimal and minimum variance portfolios.

```r
myMeanVarPort <- function(tickers, start_date, end_date, rf_rate) {
    set.seed(12)  # Ensures reproducibility

    # Step 1: Download stock data and calculate returns
    stock_returns <- lapply(tickers, function(ticker) {
        data <- getSymbols(ticker, src = "yahoo", from = start_date, to = end_date,
            auto.assign = FALSE)
        monthly_data <- to.monthly(data, indexAt = "lastof", OHLC = FALSE)
        monthly_returns <- na.omit(periodReturn(Ad(monthly_data), period = "monthly",
            type = "log"))
        colnames(monthly_returns) <- ticker
        return(monthly_returns)
    })

    # Combine returns and calculate average and std dev (annualized)
    returns_data <- do.call(merge, stock_returns)
    avg_returns <- colMeans(returns_data) * 12  # Annualized mean returns
    std_devs <- apply(returns_data, 2, sd) * sqrt(12)  # Annualized std dev
```

```r
    # Create a summary table for stocks
    stock_summary <- data.frame(Ticker = tickers, Avg_Annual_Return = avg_returns,
        Annual_Std_Dev = std_devs)

    # Calculate annualized covariance matrix
    cov_matrix <- cov(returns_data) * 12

    # Step 2: Simulate portfolios with random weights
    num_assets <- length(tickers)
    num_portfolios <- 100 * num_assets
    portfolio_means <- numeric(num_portfolios)
    portfolio_risks <- numeric(num_portfolios)
    portfolio_sharpe <- numeric(num_portfolios)
    weights_matrix <- matrix(nrow = num_portfolios, ncol = num_assets)

    for (i in 1:num_portfolios) {
        weights <- runif(num_assets)
        weights <- weights/sum(weights)

        portfolio_means[i] <- sum(weights * avg_returns)
        portfolio_risks[i] <- sqrt(t(weights) %*% cov_matrix %*% weights)
        portfolio_sharpe[i] <- (portfolio_means[i] - rf_rate)/portfolio_risks[i]
        weights_matrix[i, ] <- weights
    }

    # Store portfolio data in a data frame
    portfolio_data <- data.frame(Mean_Return = portfolio_means, Risk = portfolio_risks,
        Sharpe_Ratio = portfolio_sharpe)
    portfolio_data <- cbind(portfolio_data, weights_matrix)
    colnames(portfolio_data)[-(1:3)] <- tickers

    # Find the optimal and minimum variance portfolios
    optimal_idx <- which.max(portfolio_data$Sharpe_Ratio)
    min_var_idx <- which.min(portfolio_data$Risk)
    optimal_portfolio <- portfolio_data[optimal_idx, ]
    min_variance_portfolio <- portfolio_data[min_var_idx, ]

    # Efficient frontier
    efficient_frontier <- portfolio_data %>%
        arrange(Risk) %>%
        filter(cummax(Mean_Return) == Mean_Return)

    # Capital Market Line (CML)
    cml_slope <- (optimal_portfolio$Mean_Return - rf_rate)/optimal_portfolio$Risk
    cml_x <- seq(0, max(portfolio_data$Risk), length.out = 100)
    cml_y <- rf_rate + cml_slope * cml_x

    # Return results
    return(list(stock_summary = stock_summary, optimal_portfolio = optimal_portfolio,
        min_variance_portfolio = min_variance_portfolio, efficient_frontier = efficient_frontier,
        portfolio_data = portfolio_data, cml_x = cml_x, cml_y = cml_y))
}
```

3 Run Portfolio Optimization

The stock tickers, risk-free rate, and date range are defined as inputs for the myMeanVarPort function. This initiates the process of gathering historical data, calculating risk-return metrics, and simulating various portfolio configurations. The function subsequently identifies the optimal and minimum variance portfolios, providing a foundation for analyzing portfolio choices along the efficient frontier.

```
tickers <- c("GE", "XOM", "GBX", "SBUX", "PFE", "HMC", "NVDA")
risk_free_rate <- 0.02
results <- myMeanVarPort(tickers, "2014-01-01", "2017-12-31", risk_free_rate)
```

4 Results

4.1 Stock Summary

Below is a summary of the average annual returns and annualized standard deviations for each stock:

Table 1: Stock Summary: Average Annual Returns and Standard Deviations

	Ticker	Avg_Annual_Return	Annual_Std_Dev
GE	GE	-0.0580214	0.1847388
XOM	XOM	0.0091890	0.1332663
GBX	GBX	0.1101966	0.3560724
SBUX	SBUX	0.1350825	0.1510016
PFE	PFE	0.0795770	0.1504818
HMC	HMC	-0.0081623	0.1885273
NVDA	NVDA	0.6398149	0.3340339

Insights:

NVIDIA (NVDA) had the highest average annual return of **63.98%** with an annualized standard deviation of **33.40%**, indicating high return but also high volatility. **General Electric (GE)** showed a negative average annual return of **-5.80%** and had an annualized standard deviation of **18.47%**.

4.2 Optimal Portfolio

The optimal portfolio is the one with the highest Sharpe ratio, indicating the best risk-adjusted return. The table below provides details of the optimal portfolio's configuration, including its weights, mean return, risk, and Sharpe ratio.

Table 2: Optimal Portfolio: Highest Sharpe Ratio

	Mean_Return	Risk	Sharpe_Ratio	GE	XOM	GBX	SBUX	PFE	HMC	NVDA
327	0.2554934	0.1372822	1.715396	0.0074949	0.2088997	0.018732	0.3352189	0.076409	0.0392451	0.3140005

Optimal Portfolio Statistics:

- Mean Return: 25.55%
- Risk (Standard Deviation): 13.73%
- Sharpe Ratio: 1.7154

Asset Weights:

- GE: 0.75%
- XOM: 20.89%
- GBX: 1.87%
- SBUX: 33.52%
- PFE: 7.64%
- HMC: 3.92%
- NVDA: 31.40%

Interpretation:

The optimal portfolio provides a mean return of approximately **25.55%** with a risk of **13.73%**, achieving a Sharpe ratio of **1.7154**. This portfolio configuration offers a balanced trade-off between risk and return, suitable for investors aiming for high returns with moderate risk exposure.

4.3 Minimum Variance Portfolio

The minimum variance portfolio minimizes risk. Its configuration, mean return, risk, and Sharpe ratio are shown below:

Table 3: Minimum Variance Portfolio: Lowest Risk

	Mean_Return	Risk	Sharpe_Ratio	GE	XOM	GBX	SBUX	PFE	HMC	NVDA
524	0.0790642	0.0980129	0.6026167	0.0245192	0.2488828	0.0086057	0.2743911	0.1752096	0.2245103	0.0438814

Minimum Variance Portfolio Statistics:

- Mean Return: 7.91%
- Risk (Standard Deviation): 9.80%
- Sharpe Ratio: 0.6026

Asset Weights:

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

Interpretation:

The minimum variance portfolio has the lowest risk at **9.80%** with a mean return of **7.91%**. Although it has a lower Sharpe ratio compared to the optimal portfolio, it is better suited for risk-averse investors who prioritize stability over high returns.

4.4 Efficient Frontier Portfolios

The efficient frontier represents the set of portfolios that offer the highest return for a given level of risk. Portfolios on the efficient frontier are considered Pareto optimal in the mean-variance space.

Table 4: Efficient Frontier Portfolios

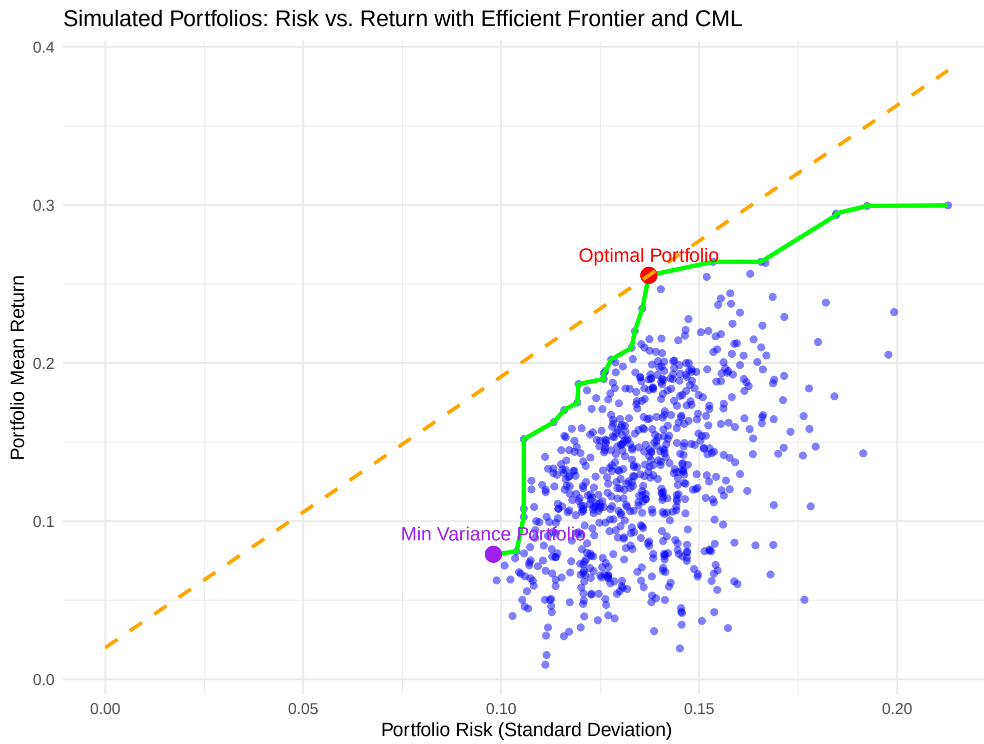
Mean_Return	Risk	Sharpe_Ratio	GE	XOM	GBX	SBUX	PFE	HMC	NVDA
0.0790642	0.0980129	0.6026167	0.0245192	0.2488828	0.0086057	0.2743911	0.1752096	0.2245103	0.0438814
0.0807897	0.1038491	0.5853651	0.1499496	0.2011693	0.0527720	0.2391955	0.2350545	0.0727761	0.0490830

Mean_Return	Risk	Sharpe_Ratio	GE	XOM	GBX	SBUX	PFE	HMC	NVDA
0.1026448	0.1056825	0.7820105	0.0339766	0.2635086	0.0700616	0.2371301	0.2617501	0.0676708	0.0659022
0.1518981	0.1056936	1.2479292	0.0046594	0.3548928	0.0053890	0.2966364	0.1036928	0.0774593	0.1572702
0.1626243	0.1132427	1.2594576	0.0685487	0.1753187	0.0352291	0.2926498	0.1209804	0.1306353	0.1766380
0.1702555	0.1159267	1.2961250	0.0086349	0.0926399	0.0391559	0.2659044	0.2938505	0.1320078	0.1678065
0.1750083	0.1192103	1.3002927	0.1128025	0.1259921	0.0201625	0.2200769	0.1756196	0.1334737	0.2118727
0.1867420	0.1195171	1.3951305	0.0962663	0.1755136	0.0057227	0.2024597	0.2297766	0.0636796	0.2265814
0.1899542	0.1258901	1.3500208	0.0676996	0.1775577	0.0324553	0.2353038	0.0622687	0.1848912	0.2398236
0.1936079	0.1259205	1.3787107	0.0209188	0.1853861	0.0588050	0.1945640	0.1949695	0.1174787	0.2278778
0.1949653	0.1263247	1.3850446	0.1016269	0.0802838	0.0746819	0.2968770	0.1543204	0.0732267	0.2189834
0.2023052	0.1277853	1.4266523	0.0988234	0.1483927	0.0727443	0.2674664	0.1770007	0.0035167	0.2320558
0.2097150	0.1329348	1.4271281	0.0012820	0.1093415	0.0803209	0.2025120	0.2394740	0.1255220	0.2415475
0.2201990	0.1337377	1.4969520	0.1452933	0.0001331	0.0235418	0.2025681	0.3504419	0.0109562	0.2670656
0.2344993	0.1356434	1.5813477	0.1009519	0.0784117	0.0481199	0.3451882	0.0846225	0.0591033	0.2836024
0.2554934	0.1372822	1.7153963	0.0074949	0.2088997	0.0187320	0.3352189	0.0764090	0.0392451	0.3140005
0.2640400	0.1535117	1.5897165	0.1421359	0.0514780	0.0475195	0.3008794	0.0048313	0.0993654	0.3537905
0.2640771	0.1656168	1.4737461	0.0707012	0.2306484	0.0620970	0.0167092	0.1670880	0.0710159	0.3817402
0.2934753	0.1845376	1.4819487	0.0329537	0.2401019	0.1194645	0.0349656	0.0596647	0.0888667	0.4239830
0.2946337	0.1846012	1.4877132	0.2307768	0.0248342	0.0980868	0.1070849	0.0450197	0.0574961	0.4367015
0.2993790	0.1924172	1.4519437	0.0130994	0.0039821	0.1738845	0.0023762	0.2468098	0.1500354	0.4098126
0.2997343	0.2128559	1.3141956	0.0209738	0.1003786	0.3113236	0.0173611	0.1414314	0.0142942	0.3942372

5 Visualization of Portfolios

The plot below shows all simulated portfolios, the efficient frontier, the Capital Market Line (CML), and highlights the optimal and minimum variance portfolios.

```r
ggplot(results$portfolio_data, aes(x = Risk, y = Mean_Return)) + geom_point(color = "blue",
    alpha = 0.5) + geom_line(data = results$efficient_frontier, aes(x = Risk, y = Mean_Return),
    color = "green", linewidth = 1.2) + annotate("point", x = results$optimal_portfolio$Risk,
    y = results$optimal_portfolio$Mean_Return, color = "red", size = 4) + annotate("point",
    x = results$min_variance_portfolio$Risk, y = results$min_variance_portfolio$Mean_Return,
    color = "purple", size = 4) + geom_line(data = data.frame(x = results$cml_x,
    y = results$cml_y), aes(x = x, y = y), color = "orange", linetype = "dashed",
    linewidth = 1) + labs(title = "Simulated Portfolios: Risk vs. Return with Efficient Frontier and CM
    x = "Portfolio Risk (Standard Deviation)", y = "Portfolio Mean Return") + annotate("text",
    x = results$optimal_portfolio$Risk, y = results$optimal_portfolio$Mean_Return,
    label = "Optimal Portfolio", color = "red", vjust = -1) + annotate("text", x = results$min_variance_
    y = results$min_variance_portfolio$Mean_Return, label = "Min Variance Portfolio",
    color = "purple", vjust = -1) + theme_minimal()
```



Plot Analysis:

- **Blue Points:** Represent the simulated portfolios.
- **Green Line:** The efficient frontier, showing the optimal portfolios for each risk level.
- **Orange Dashed Line:** The Capital Market Line (CML), tangent to the efficient frontier at the optimal portfolio.
- **Red Point:** Indicates the optimal portfolio, with the highest Sharpe ratio.
- **Purple Point:** Indicates the minimum variance portfolio, with the lowest risk.

Discussion

The optimal portfolio identified in this analysis offers a balance between risk and return, maximizing the Sharpe ratio. Meanwhile, the minimum variance portfolio serves as a low-risk alternative. Investors can select an appropriate portfolio along the efficient frontier depending on their risk tolerance.

The simulation-based approach to portfolio optimization is effective for generating insights into mean-variance efficiency without requiring closed-form solutions. However, it has limitations in precision and may need fine-tuning in real-world applications.

6 Conclusion

This analysis demonstrates the application of portfolio optimization principles using a simulation approach. By evaluating portfolios in mean-variance space, we identified optimal and efficient portfolio configurations, which can serve as foundational strategies for investors with different risk preferences.

Future analyses could extend this work by including additional constraints, such as transaction costs or minimum/maximum weights, to better reflect real-world conditions.

7 References

The code used in this report is available on GitHub: https://github.com/parthchaudhari18/PortfolioOptimiz
ation.