LavanyaKapoor_PortfolioAssignment

2025-11-02

```
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(xts)
library(zoo)
library(dplyr)
```

```
##
## ######################### Warning from 'xts' package #########################
## #                                                                            #
## # The dplyr lag() function breaks how base R's lag() function is supposed to #
## # work, which breaks lag(my_xts). Calls to lag(my_xts) that you type or      #
## # source() into this session won't work correctly.                           #
## #                                                                            #
## # Use stats::lag() to make sure you're not using dplyr::lag(), or you can add #
## # conflictRules('dplyr', exclude = 'lag') to your .Rprofile to stop          #
## # dplyr from breaking base R's lag() function.                               #
## #                                                                            #
## # Code in packages is not affected. It's protected by R's namespace mechanism #
## # Set `options(xts.warn_dplyr_breaks_lag = FALSE)` to suppress this warning. #
## #                                                                            #
## ##############################################################################
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:xts':
##
##     first, last
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyr)
library(ggplot2)
library(quadprog)
library(ggrepel)
library(PerformanceAnalytics)
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

Part 1: Portfolio Return Calculation

```
#1. Selecting 8 publicly traded stocks.
tickers <- c("AAPL", "MSFT", "AMZN", "NVDA", "JPM", "GOOGL", "TSLA", "META")
start_date <- as.Date("2021-01-01")
end_date   <- as.Date("2025-09-30")
initial_investment <- 1000

PriceList <- lapply(tickers, function(s) Ad(getSymbols(s, src="yahoo",
                                                       from=start_date, to=end_date,
                                                       auto.assign=FALSE)))
Prices <- do.call(merge, PriceList)
colnames(Prices) <- tickers
MonthlyPrices <- to.monthly(Prices, indexAt="lastof", OHLC=FALSE)
```

Equal-Weighted and Value-weighted Portfolio Selection:

```
#initial investment = 0
initial_inv = 1000
#3. (a) Equal-weighted portfolio:
n <- length(tickers)
w_equal <- rep(1/n, n)
shares_eq <- (initial_inv * w_equal) / as.numeric(first(MonthlyPrices))
port_val_eq <- as.numeric(MonthlyPrices %*% shares_eq)

#3. (b) Value-weighted portfolio:
set.seed(42)

# simulating the market caps
mkt_caps <- round(runif(n, 500, 5000))
w_val <- mkt_caps / sum(mkt_caps)

#4. Calculating the portfolio value at the end of each month for both portfolios using matrix
# multiplication or weighted sums, without rebalancing

shares_val <- (initial_inv * w_val) / as.numeric(first(MonthlyPrices))
port_val <- as.numeric(MonthlyPrices %*% shares_val)


portfolio_df <- data.frame(
  Date = index(MonthlyPrices),
  Equal_Weighted = port_val_eq,
  Value_Weighted = port_val
)

head(portfolio_df)
```

```
##         Date Equal_Weighted Value_Weighted
## 1 2021-01-31       1000.000       1000.000
## 2 2021-02-28       1005.453       1001.630
## 3 2021-03-31       1029.054       1012.981
## 4 2021-04-30       1120.938       1098.189
## 5 2021-05-31       1114.884       1093.928
## 6 2021-06-30       1197.866       1186.757
```

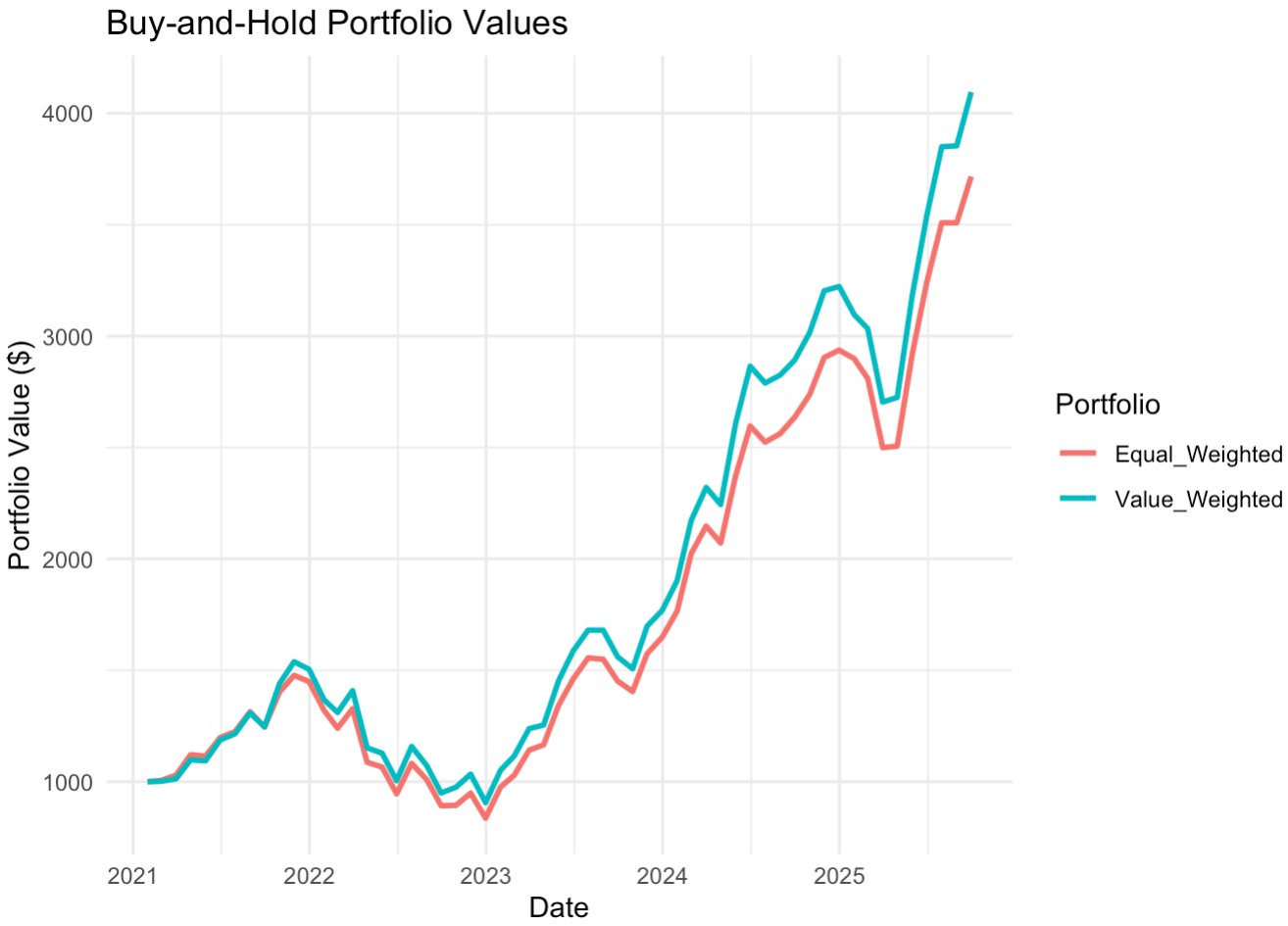
The table shows the month-end portfolio values for both strategies. The equal-weighted portfolio initially grows slightly faster than the value-weighted portfolio because all stocks contribute equally, giving more weight to smaller-cap stocks with higher returns. The value-weighted portfolio grows more steadily, reflecting the performance of larger-cap stocks which dominate its weighting.

Plot for the Cumulative Returns

```
#5. Plotting the cumulative returns of both portfolios over time
plot_df <- portfolio_df %>%
  pivot_longer(cols = c(Equal_Weighted, Value_Weighted),
               names_to = "Portfolio", values_to = "Value")

ggplot(plot_df, aes(x=Date, y=Value, color=Portfolio)) +
  geom_line(size=1) +
  labs(title="Buy-and-Hold Portfolio Values",
       x="Date",
       y="Portfolio Value ($)") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



The graph shows the cumulative values of the Equal-Weighted and Value-Weighted portfolios over time. Early in 2021, the Equal-Weighted portfolio grows a bit faster because it has more exposure to smaller stocks. During the 2022–2023 downturn, it drops more sharply and underperforms the Value-Weighted portfolio. From mid-2023 onward, the Value-Weighted portfolio pulls ahead and stays higher, driven by strong gains in larger stocks. Overall, equal-weighting gives early boosts but is more volatile, while value-weighting ends with a higher and steadier portfolio value.

```
final_equal <- last(portfolio_df$Equal_Weighted)
final_value <- last(portfolio_df$Value_Weighted)
years <- as.numeric(difftime(last(portfolio_df$Date), first(portfolio_df$Date), units="weeks")) / 52.25

cagr_equal <- (final_equal/initial_investment)^(1/years) - 1
cagr_value <- (final_value/initial_investment)^(1/years) - 1
cagr_equal
```

```
## [1] 0.3256954
```

```
cagr_value
```

```
## [1] 0.353656
```

The CAGR values indicate the annualized growth of the portfolios over the sample period. The value-weighted portfolio grew slightly faster than the equal-weighted portfolio, which aligns with the earlier observation from the cumulative returns graph. This is because the value-weighted portfolio assigns higher weights to larger-cap stocks, which performed better during this period, while the equal-weighted portfolio treats all stocks equally, including smaller underperforming stocks.

If monthly rebalancing were applied:

1. Equal-weighted portfolio might outperform slightly in volatile periods since weights are reset to equal each month.
2. Value-weighted portfolio would involve lower transaction costs if market caps don’t change drastically but might reduce potential gains from consistently high-performing stocks.

PART - 2: Portfolio Optimization

```
#Selecting 2 stocks
stocks_selec <- c("TSLA", "JNJ")
start2 <- as.Date("2022-01-01")
end2 <- as.Date("2025-09-30")

PricesList_2 <- lapply(stocks_selec, function(s) Ad(getSymbols(s, src="yahoo",
                                                               from=start2, to=end2,
                                                               auto.assign=FALSE)))
Prices2 <- do.call(merge, PricesList_2)
colnames(Prices2) <- stocks_selec
MonthlyPrices_2 <- to.monthly(Prices2, indexAt="lastof", OHLC=FALSE)
rets2 <- na.omit(Return.calculate(MonthlyPrices_2, method="discrete"))[-1, ]

# Annualized mean & covariance
mu <- colMeans(rets2) * 12
Sigma <- cov(rets2) * 12

w_grid <- seq(0, 1, by = 0.001)
results <- data.frame(
  w1 = w_grid,
  w2 = 1 - w_grid,
  Port_Mean = NA_real_,
  Port_Sigma = NA_real_
)

for (i in seq_along(w_grid)) {
  w <- c(w_grid[i], 1 - w_grid[i])
  results$Port_Mean[i]  <- sum(w * mu)
  results$Port_Sigma[i] <- sqrt(t(w) %*% Sigma %*% w)
}

# minimum-variance portfolio
mvp_index <- which.min(results$Port_Sigma)
mvp <- results[mvp_index, ]

rf_annual <- 0.01
```

```
rf_monthly <- (1 + rf_annual)^(1/12) - 1
results$Sharpe <- (results$Port_Mean - rf_annual*12) / results$Port_Sigma

index_opt <- which.max(results$Sharpe)
opt_port <- results[index_opt, ]

points_df <- results[c(mvp_index, index_opt), ]

points_df
```

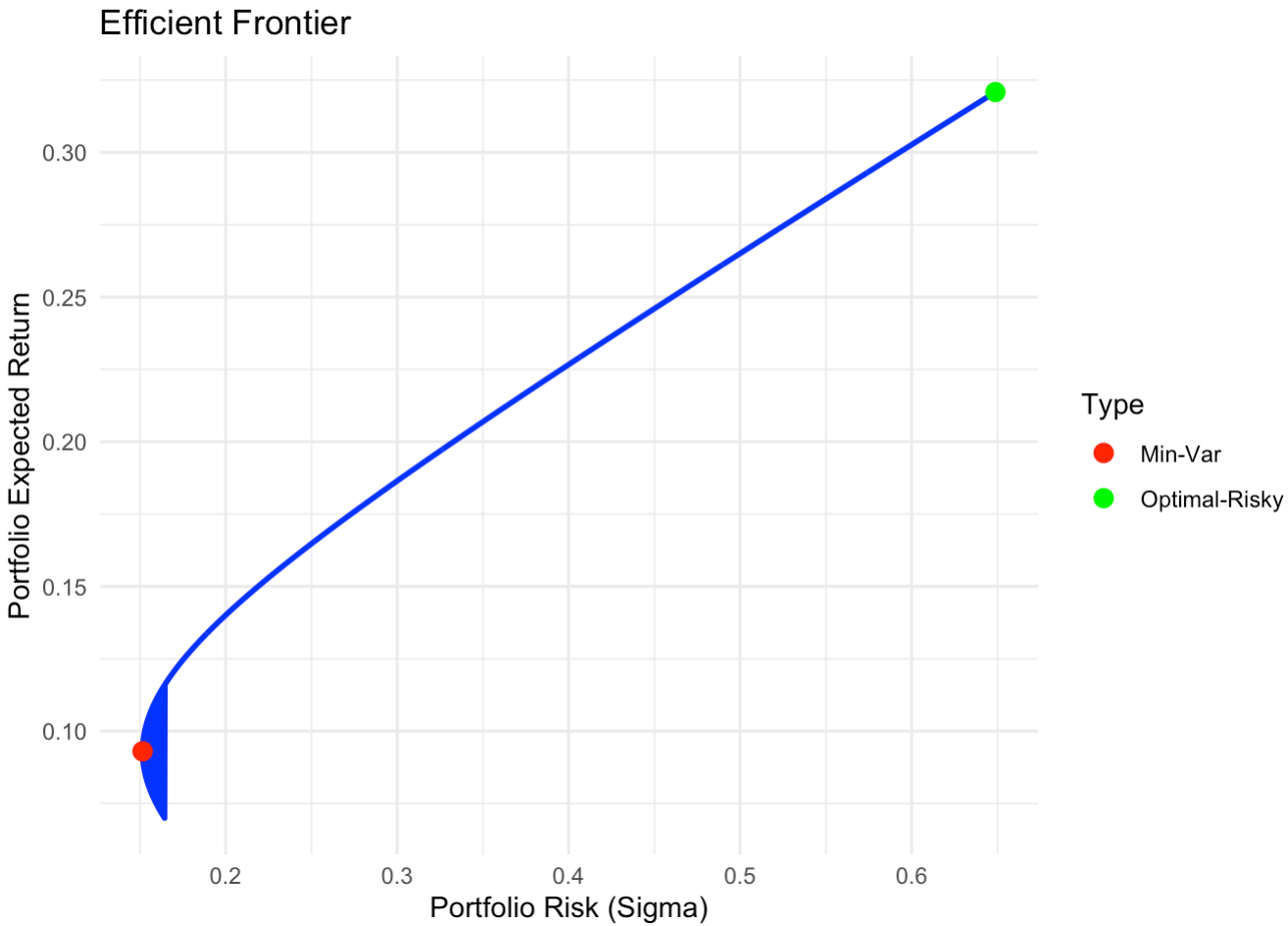
```
##         w1    w2  Port_Mean Port_Sigma     Sharpe
## 93   0.092 0.908 0.09300312  0.1515251 -0.1781677
## 1001 1.000 0.000 0.32087500  0.6488747  0.3095744
```

The minimum-variance portfolio consists of 9.2% in TSLA and 90.8% in JNJ, with an expected annual return of 9.3% and portfolio risk of 15.2%. The optimal risky portfolio consists of 100% in TSLA, giving an expected return of 32.1% and risk of 64.9%. The minimum-variance portfolio prioritizes low risk, while the optimal risky portfolio maximizes risk-adjusted return (Sharpe ratio).

Efficient Frontier Plot

```
points_df$Type <- c("Min-Var", "Optimal-Risky")

ggplot(results, aes(x=Port_Sigma, y=Port_Mean)) +
  geom_line(color="blue", size=1) +
  geom_point(data = points_df, aes(x=Port_Sigma, y=Port_Mean, color=Type), size=3) +
  scale_color_manual(values = c("red", "green")) +
  labs(title="Efficient Frontier",
       x="Portfolio Risk (Sigma)", y="Portfolio Expected Return") +
  theme_minimal()
```



The graph shows the relationship between risk (Portfolio Risk σ) and return (Portfolio Expected Return) for all possible ways of combining the two stocks. The blue curve shows all achievable portfolios, and the top part of the curve is the efficient frontier, giving the highest return for a given level of risk. Portfolios below this curve are less efficient because they give lower returns for the same risk. The Minimum-Variance Portfolio (red point) has the lowest risk (σ ≈ 0.15) and a moderate return (≈ 9%), making it suitable for investors who want stability. The Optimal Risky Portfolio (green point) has higher risk (σ ≈ 0.65) but the best risk-adjusted return (≈ 32%, highest Sharpe Ratio) and is the ideal choice for investors willing to take more risk to earn higher returns. Overall, the graph shows the trade-off between taking risk and getting higher returns.