Test_GSOC_2024_Portfolio

Simone Mugnai

2024-02-29

Easy Part

```
library(PortfolioAnalytics)
```

```
## Warning: package 'PortfolioAnalytics' was built under R version 4.2.3
```

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

```
## Warning: package 'zoo' was built under R version 4.2.2
```

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

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

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

```
## Warning: package 'xts' was built under R version 4.2.3
```

```
## Loading required package: foreach
```

```
## Warning: package 'foreach' was built under R version 4.2.2
```

```
## Loading required package: PerformanceAnalytics
```

```
## Warning: package 'PerformanceAnalytics' was built under R version 4.2.3
```

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

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

```r
library(PerformanceAnalytics)
library(ROI)
```

```
## Warning: package 'ROI' was built under R version 4.2.3
```

```
## Registered S3 method overwritten by 'ROI':
##   method           from
##   print.constraint PortfolioAnalytics
```

```
## ROI: R Optimization Infrastructure
```

```
## Registered solver plugins: nlminb, glpk, quadprog.
```

```
## Default solver: auto.
```

```
##
## Attaching package: 'ROI'
```

```
## The following objects are masked from 'package:PortfolioAnalytics':
##
##     is.constraint, objective
```

```r
library(ROI.plugin.glpk)
```

```
## Warning: package 'ROI.plugin.glpk' was built under R version 4.2.3
```

```r
library(ROI.plugin.quadprog)
```

```
## Warning: package 'ROI.plugin.quadprog' was built under R version 4.2.3
```

```r
library(DEoptim)
```

```
## Warning: package 'DEoptim' was built under R version 4.2.3
```

```
## Loading required package: parallel
```

```
##
## DEoptim package
## Differential Evolution algorithm in R
## Authors: D. Ardia, K. Mullen, B. Peterson and J. Ulrich
```

Simulating Asset return

```r
set.seed(123)

simulated_returns <- matrix(rnorm(60 * 3, mean = 0.01, sd = 0.05), ncol = 3)
colnames(simulated_returns) <- c("Asset1", "Asset2", "Asset3")

dates <- seq(as.Date("2019-01-01"), by="month", length.out=60)
asset_returns <- xts(simulated_returns, order.by=dates)
cov_matrix <- cov(asset_returns)
```

Portfolio Specification and Constraints

```
portfolio <- portfolio.spec(assets = c("Asset1", "Asset2", "Asset3"))
portfolio <- add.constraint(portfolio, type = "box", min = 0.1, max = 0.9)
```

Random Portfolio Generation

```
random_portfolios <- rp_simplex(portfolio, permutations = 100, fev = 0:5)
```

```
## Warning: executing %dopar% sequentially: no parallel backend registered
```

Minimum Variance Portfolio (MVP)

```
mvp <- portfolio  # Start with the original portfolio specification
mvp <- add.objective(mvp, type="risk", name="StdDev")
mvp_result <- optimize.portfolio(asset_returns, mvp, optimize_method="ROI", trace=TRUE)
print(mvp_result)
```

```
## ***********************************
## PortfolioAnalytics Optimization
## ***********************************
##
## Call:
## optimize.portfolio(R = asset_returns, portfolio = mvp, optimize_method = "ROI",
##     trace = TRUE)
##
## Optimal Weights:
## Asset1 Asset2 Asset3
## 0.2992 0.4236 0.2772
##
## Objective Measure:
##  StdDev
## 0.02914
```

Markowitz Portfolio Optimization

```
markowitz <- portfolio.spec(assets = colnames(asset_returns))
markowitz <- add.constraint(markowitz, type = "box", min = 0.1, max = 0.9)
markowitz <- add.objective(markowitz, type="return", name="mean")
markowitz <- add.objective(markowitz, type="risk", name="StdDev")
markowitz_result <- optimize.portfolio(asset_returns, markowitz, optimize_method="ROI", trace=TRUE)
print(markowitz_result)
```

```
## ***********************************
## PortfolioAnalytics Optimization
```

```
## **********************************
##
## Call:
## optimize.portfolio(R = asset_returns, portfolio = markowitz,
##     optimize_method = "ROI", trace = TRUE)
##
## Optimal Weights:
## Asset1 Asset2 Asset3
##    0.8    0.1    0.1
##
## Objective Measure:
##    mean
## 0.01238
##
##
##  StdDev
## 0.03852
```

Results Visualization

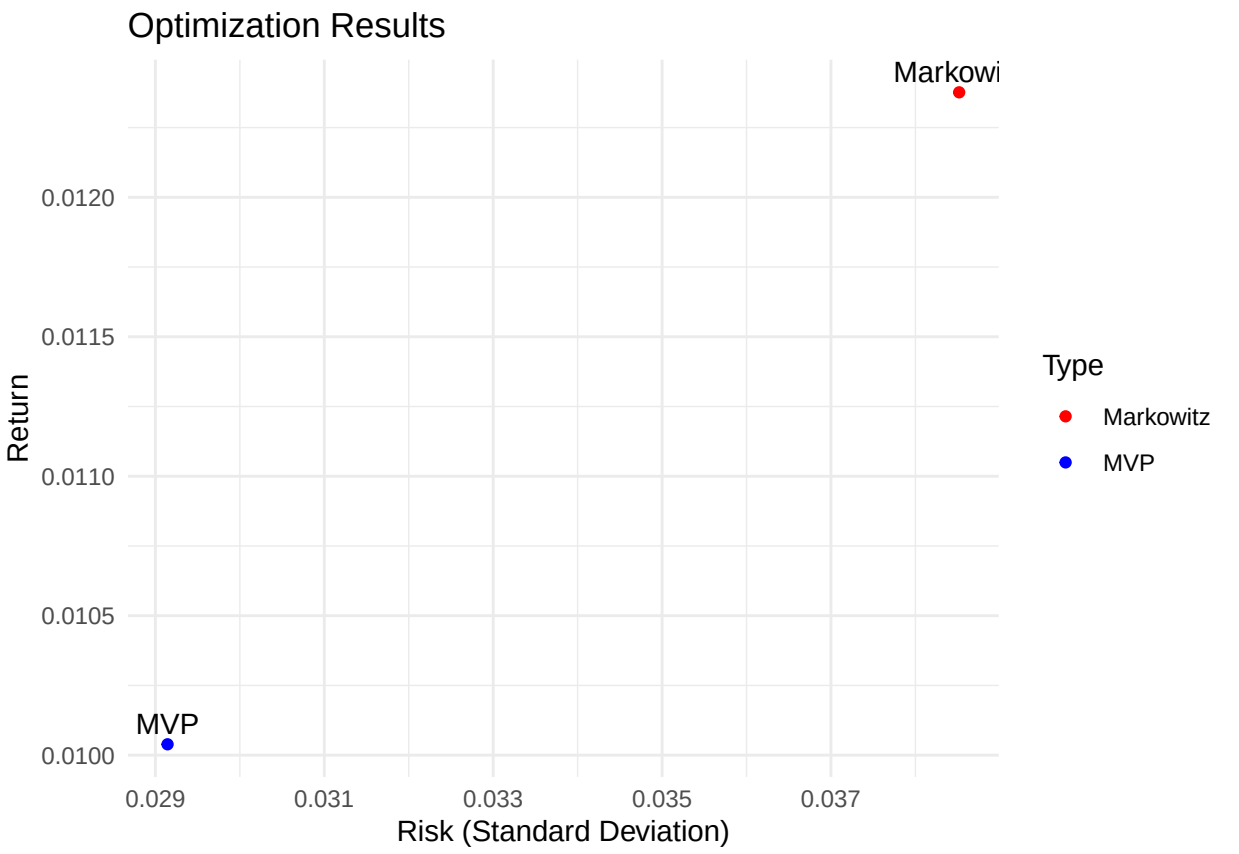
```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.2.3
```

```
mvp_expected_return <- sum(mvp_result$weights * colMeans(asset_returns))
mvp_portfolio_risk <- sqrt(t(mvp_result$weights) %*% cov_matrix %*% mvp_result$weights)
markowitz_expected_return <- sum(markowitz_result$weights * colMeans(asset_returns))
markowitz_portfolio_risk <- sqrt(t(markowitz_result$weights) %*% cov_matrix %*% markowitz_result$weights

data_to_plot <- data.frame(
  Risk = c(mvp_portfolio_risk, markowitz_portfolio_risk),
  Return = c(mvp_expected_return, markowitz_expected_return),
  Type = c("MVP", "Markowitz")
)

ggplot(data_to_plot, aes(x = Risk, y = Return, label = Type)) +
  geom_point(aes(color = Type)) +
  geom_text(vjust = -0.5) +
  scale_color_manual(values = c("MVP" = "blue", "Markowitz" = "red")) +
  theme_minimal() +
  labs(x = "Risk (Standard Deviation)", y = "Return", title = "Optimization Results")
```



#Hard Part

Additional Model Portfolio Objectives for a Retirement Portfolio:

Minimize Drawdown: Retirement portfolios should aim to minimize large losses, as retirees may not have the capacity to recover from significant drawdowns. This can be added as an objective:

```
retirement <- add.objective(portfolio, type="risk", name="maxDrawdown")
```

Maximize Income: Since retirees often rely on their investments for income, maximizing yield can be a critical objective. This involves selecting assets or asset classes known for generating consistent income (e.g., dividend-paying stocks, bonds)

```
dividend_yield <- c(0.02, 0.03, 0.04) # Example values

# Define a custom objective function for dividend yield
calculate_dividend_yield <- function(weights, dividend_yields) {
  total_yield <- sum(weights * dividend_yields)
  return(-total_yield) # Return negative yield for maximization
}


retirement <- add.objective(portfolio, type="custom", name="calculate_dividend_yield", FUN=calculate_di
```

```
#retirement <- add.objective(portfolio, type="return", name="custom", FUN="sum", arguments=list(x=divid
```

Long-term Capital Appreciation: While minimizing risk, the portfolio should still aim for growth over the long term to counteract inflation and support withdrawals.

```
retirement <- add.objective(retirement, type="return", name="mean", multiplier=1)
```

Constraints Tailored to a Retirement Portfolio:

Asset Allocation Constraints: These can ensure diversification and limit exposure to high-volatility assets. For example, setting a constraint to limit the allocation to stocks to a certain percentage and ensure a minimum allocation to bonds.

```
retirement <- add.constraint(retirement, type="box", min=c(0.4, 0.3, 0.2), max=c(0.6, 0.5, 0.4), asset=
```

Liquidity Requirement: Ensuring that a portion of the portfolio is held in highly liquid assets for covering unexpected expenses.

```
retirement <- add.constraint(retirement, type="weight_sum", min=0.2, asset="Cash")
```

Appropriate Risk Metrics:

Maximum Drawdown: Important for retirement portfolios as it measures the largest single drop from peak to bottom in the value of the portfolio, highlighting the worst-case scenario for retirees. Value at Risk (VaR) and Conditional Value at Risk (CVaR): These metrics can help in understanding potential losses in the portfolio over a specified time frame, which is crucial for planning withdrawals and understanding risk exposure. Appropriate Return Metrics: Yield: The income return on an investment, such as interest or dividends received, is particularly relevant for retirees who may rely on their investments for income. Total Return: Reflects both the capital appreciation and the income of the portfolio, showing the overall performance.

Comparing to MVP and Markowitz Portfolios: After specifying your retirement portfolio, optimize it using the same process as for the MVP and Markowitz portfolios. You can then visualize and compare the outcomes using similar risk and return metrics. The comparison could focus on:

Risk-adjusted Return: Compare the Sharpe ratios to evaluate how efficiently each portfolio compensates for its risk.

Income Generation: Compare the expected income (e.g., dividends or interest) generated by each portfolio, which is crucial for a retirement portfolio. Stability: Compare maximum drawdowns and VaR/CVaR metrics across portfolios to assess stability and risk of significant losses.

```
retirement_result <- optimize.portfolio(asset_returns, retirement, optimize_method="DEoptim", trace=TRU
```

```
## Warning in (function (VTR = -Inf, strategy = 2, bs = FALSE, NP = NA, itermax =
## 200, : Resetting NP to the number of rows in initialpop
```

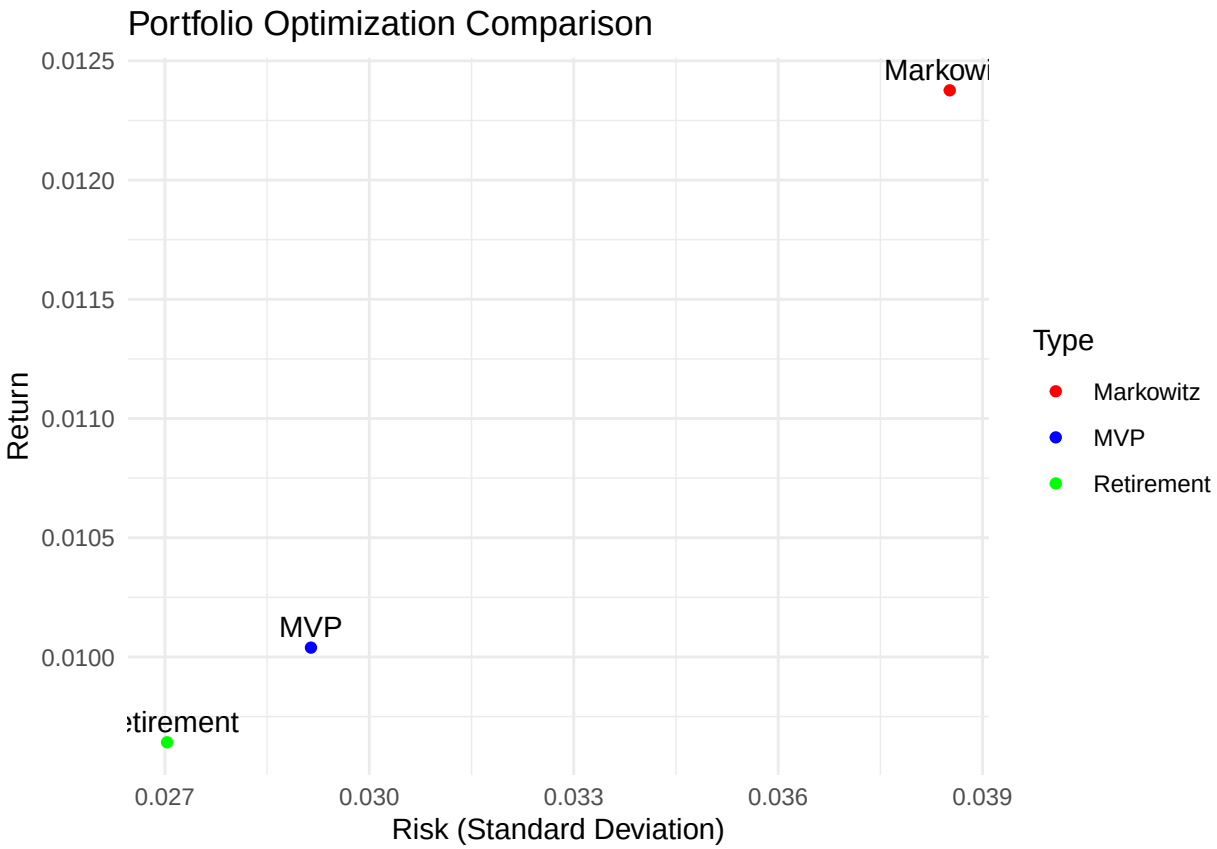
```
## Iteration: 1 bestvalit: 0.009821 bestmemit:    0.400000    0.306000    0.214000
## Iteration: 2 bestvalit: 0.009642 bestmemit:    0.400000    0.300000    0.200000
## Iteration: 3 bestvalit: 0.009642 bestmemit:    0.400000    0.300000    0.200000
## Iteration: 4 bestvalit: 0.009642 bestmemit:    0.400000    0.300000    0.200000
## Iteration: 5 bestvalit: 0.009642 bestmemit:    0.400000    0.300000    0.200000
## Iteration: 6 bestvalit: 0.009642 bestmemit:    0.400000    0.300000    0.200000
## Iteration: 7 bestvalit: 0.009642 bestmemit:    0.400000    0.300000    0.200000
## [1] 0.4 0.3 0.2
```

```r
# Calculate expected return, portfolio risk, etc., for the retirement portfolio
retirement_expected_return <- sum(retirement_result$weights * colMeans(asset_returns))
retirement_portfolio_risk <- sqrt(t(retirement_result$weights) %*% cov_matrix %*% retirement_result$weig
# Example values for demonstration
retirement_max_drawdown <- 0.15
retirement_yield <- sum(retirement_result$weights * dividend_yield)

# Combine all data for plotting
data_to_plot_extended <- rbind(data_to_plot, data.frame(
  Risk = retirement_portfolio_risk,
  Return = retirement_expected_return,
  Type = "Retirement"
))
```

Comparison

```r
ggplot(data_to_plot_extended, aes(x = Risk, y = Return, label = Type)) +
  geom_point(aes(color = Type)) +
  geom_text(vjust = -0.5) +
  scale_color_manual(values = c("MVP" = "blue", "Markowitz" = "red", "Retirement" = "green")) +
  theme_minimal() +
  labs(x = "Risk (Standard Deviation)", y = "Return", title = "Portfolio Optimization Comparison")
```


Portfolio Optimization Comparison
0.0125
0.0120
0.0115
0.0110
0.0105
0.0100
Return
Markowi
MVP
etirement
0.027
0.030
0.033
0.036
0.039
Risk (Standard Deviation)
Type
Markowitz
MVP
Retirement