Portfolio Analysis

Get Price Data

```
tickers = c(
  "DELL","NVDA", "MU", # Tech
  "GE","UBER", # Industrials
  "AXP","JPM", # Financials
  "AMZN","HLT", # Consumer Discretionary
  "TRGP", # Energy
  "LLY","BSX", # Health Care
  "NRG", # Utilities
  "BND","GSG" # Fixed-Income and Commodity ETFs
)
getSymbols(tickers, from = "2023-01-01", to = "2024-05-31")
```

```
 [1] "DELL" "NVDA" "MU"   "GE"   "UBER" "AXP"  "JPM"  "AMZN" "HLT"  "TRGP"
[11] "LLY"  "BSX"  "NRG"  "BND"  "GSG"
```

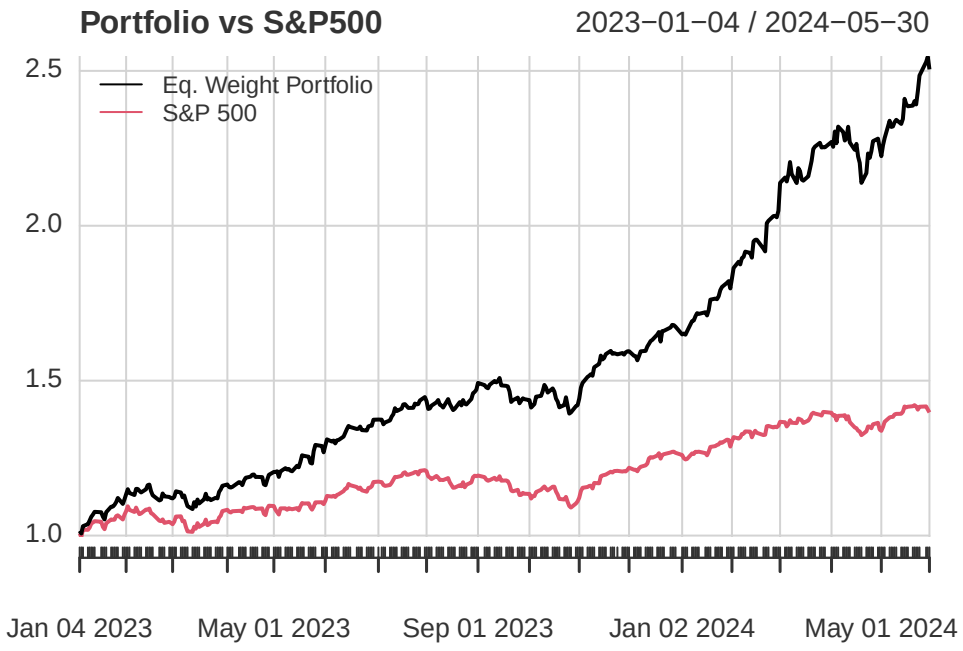
```
prices <- do.call(merge, lapply(tickers, function(t) Ad(get(t))))
returns <- na.omit(Return.calculate(prices))

tickers <- c("SPY") # S&P500 as benchmark
getSymbols(tickers, from = "2023-01-01", to = "2024-05-31")
```

```
[1] "SPY"
```

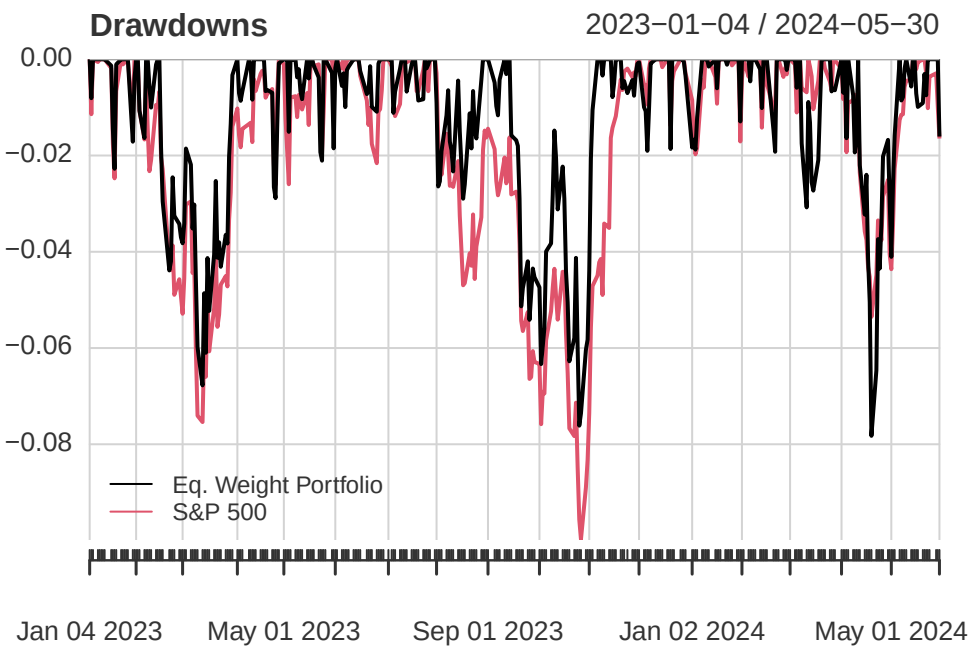
```
spy_prices <- do.call(merge, lapply(tickers, function(t) Ad(get(t))))
spy_returns <- na.omit(Return.calculate(spy_prices))
```

Equal Weight Portfolio

```
n_assets = ncol(returns)
weights = rep(1 / n_assets, n_assets)

portfolio_returns = Return.portfolio(returns, weights = weights)
common_dates = index(portfolio_returns)[index(portfolio_returns) %in% index(spy_returns)]
portfolio_returns_aligned = portfolio_returns[common_dates]
spy_returns_aligned = spy_returns[common_dates]

combined_returns= merge.xts(portfolio_returns_aligned, spy_returns_aligned)
colnames(combined_returns) = c("Eq. Weight Portfolio", "S&P 500")

chart.CumReturns(combined_returns,
                 wealth.index = T,
                 main = "Portfolio vs S&P500",
                 legend.loc = "topleft")
```




```
chart.Drawdown(combined_returns,
               legend.loc = "bottomleft",
               main = "Drawdowns")
```




```
SharpeRatio.annualized(combined_returns, Rf = 0)
```

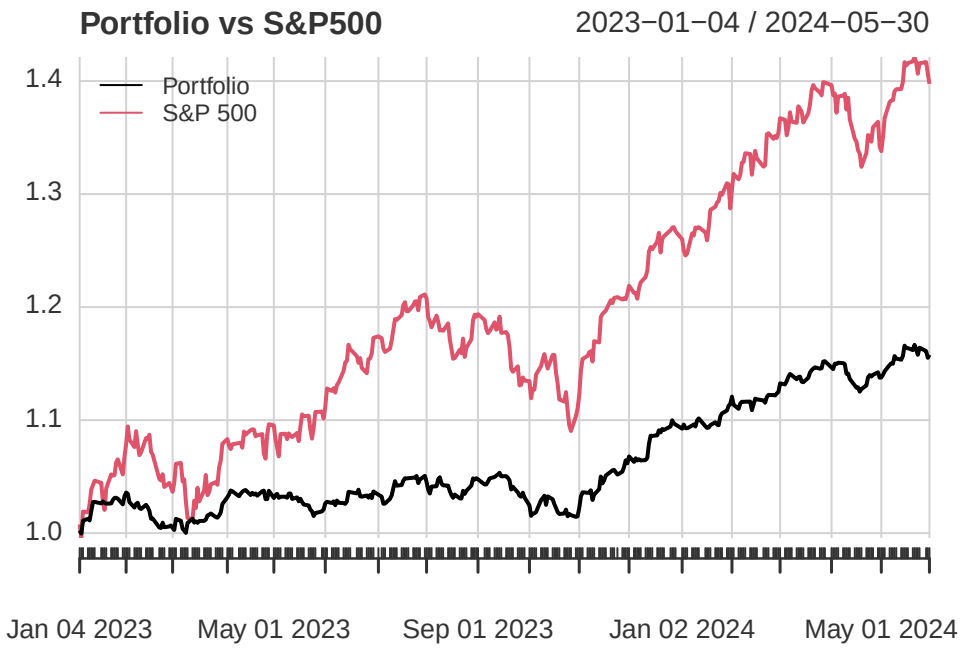
```
                                 Eq. Weight Portfolio  S&P 500
Annualized Sharpe Ratio (Rf=0%)             5.290217 2.146388
```

An equal weighted portfolio across sectors offers nearly twice the market return for similar draw down leading to a 2x difference in Sharpe Ratio (risk-adjusted return)

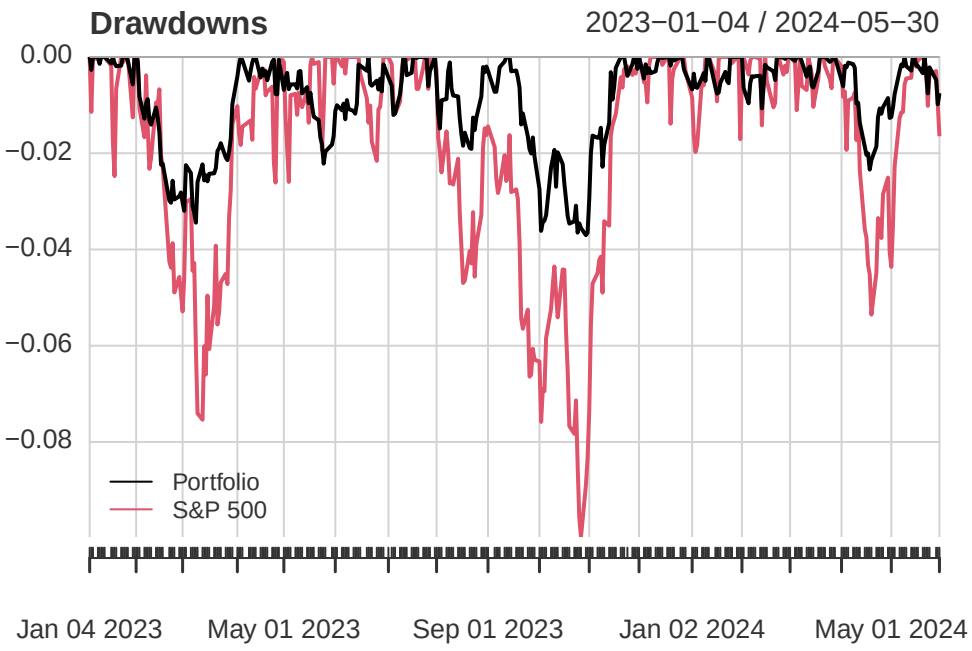
Limited Drawdown Portfolio

```
cov_matrix <- cov(returns)
n_assets <- ncol(returns)

# Dmat and dvec for quadprog (minimize (1/2) x' D x - d' x)
Dmat <- 2 * cov_matrix
dvec <- rep(0, n_assets)

# Constraints: sum of weights = 1, weights >= 0 (no short selling)
Amat <- cbind(rep(1, n_assets), diag(n_assets))
bvec <- c(1, rep(0, n_assets))
```

```r
# Solve QP
sol <- solve.QP(Dmat, dvec, Amat, bvec, meq = 1)

# Optimal weights
opt_weights <- sol$solution
names(opt_weights) <- colnames(returns)
portfolio_returns = Return.portfolio(returns, weights = opt_weights)

common_dates = index(portfolio_returns)[index(portfolio_returns) %in% index(spy_returns)]

portfolio_returns_aligned = portfolio_returns[common_dates]
spy_returns_aligned = spy_returns[common_dates]

combined_returns= merge.xts(portfolio_returns_aligned, spy_returns_aligned)
colnames(combined_returns) = c("Portfolio", "S&P 500")

chart.CumReturns(combined_returns,
                 wealth.index = T,
                 main = "Portfolio vs S&P500",
                 legend.loc = "topleft")
```



```
chart.Drawdown(combined_returns,
               legend.loc = "bottomleft",
               main = "Drawdowns")
```




```
SharpeRatio.annualized(combined_returns, Rf = 0)
```

```
                                 Portfolio  S&P 500
Annualized Sharpe Ratio (Rf=0%)   1.88946 2.146388
```

While the optimized portfolio has a lower annualized return, it also shows limited draw down, offering a less volatile growth rate compared to the broader market