Portfolio Optimization Project

```r
# Set CRAN mirror
options(repos = c(CRAN = "https://cran.rstudio.com/"))
```

```r
#Install packages
install.packages(c("quantmod", "PerformanceAnalytics", "quadprog"))
```

```
## package 'quantmod' successfully unpacked and MD5 sums checked
## package 'PerformanceAnalytics' successfully unpacked and MD5 sums checked
## package 'quadprog' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\chanc\AppData\Local\Temp\Rtmpw7UEVC\downloaded_packages
```

```r
library(quantmod)
library(PerformanceAnalytics)
library(quadprog)
```

```r
#List of stocks to download
stocks <- c("AAPL", "MSFT", "GOOG")

#Download stock data from Yahoo Finance from 2020 to 2023
getSymbols(stocks, src = "yahoo", from = "2020-01-01", to = "2023-01-01")
```

```
## [1] "AAPL" "MSFT" "GOOG"
```

```r
#Merge adj close prices into one df
prices <- do.call(merge, lapply(stocks, function(x) Ad(get(x))))

#Remove rows with NA
prices <- na.omit(prices)
```

```r
#Calculate daily returns using ROC
returns <- na.omit(ROC(prices, type = "discrete"))
```

```r
#Calculate the mean returns
mean_returns <- colMeans(returns)

#Calculate covariance matric of returns
cov_matrix <- cov(returns)
```

```r
# Num assets in protfolio
num_assets <- length(mean_returns)

#Objective function: Minimize variance (risk)
Dmat <- 2 * cov_matrix
dvec <- rep(0, num_assets)

#Constraints: Weights sum to 1 and non-negative weights
Amat <- cbind(rep(1, num_assets), diag(num_assets))
```

```r
bvec <- c(1, rep(0, num_assets))
```

```r
#Perform Optimization

#Solve the quadratic programming problem
opt <- solve.QP(Dmat, dvec, Amat, bvec, meq = 1)

#Extract optimal weights
optimal_weights <- opt$solution
```

```r
# Evaluate Optimal Portfolio

#Calculate expected return of optimal portfolio by summing the products of the optimal weights and the
expected_return <- sum(optimal_weights * mean_returns)
#Calcualte protfolio risk using the optimal weights and the covariance matrix
expected_risk <- sqrt(t(optimal_weights) %*% cov_matrix %*% optimal_weights)
```

```r
#Display Results

cat("Optimal Weights: ", optimal_weights, "\n")
```

```
## Optimal Weights:  0.2396665 0.2882674 0.4720661
```

```r
cat("Expected Return: ", expected_return, "\n")
```

```
## Expected Return:  0.00075201
```

```r
cat("Expected Risk: ", expected_risk, "\n")
```

```
## Expected Risk:  0.02054048
```