ORIE 5630 Project

Sicheng Wang

2022-11-29

```
load.Rdata(filename = "/Users/jameswang/Documents/School Work//ORIE 5630/HW3/returns.Au
g.19.2022.RData", "returns")
R = as.data.frame(returns[,c("AAPL","AEP","BAX","ED","F","GE","GOOG","MCD")])
R_m = as.data.frame(returns[, "SP500"])
head(R)
```

```
##                    AAPL          AEP          BAX           ED           F
## 2010-01-04  0.015445158  0.004302279 -0.006668444 -0.001100949 0.027615033
## 2010-01-05  0.001727432 -0.011514450 -0.006885909 -0.013086510 0.064052158
## 2010-01-06 -0.016034116  0.010082068  0.003448901  0.010659521 0.036726124
## 2010-01-07 -0.001850368  0.008562046  0.016052128 -0.003540132 0.025185609
## 2010-01-08  0.006626365  0.011864271  0.002368945  0.003098416 0.002569555
## 2010-01-11 -0.008860762  0.010337005 -0.005762754  0.012517892 0.035297868
##                      GE         GOOG           MCD
## 2010-01-04  0.020929569  0.010860519  0.0054302457
## 2010-01-05  0.005164444 -0.004413373 -0.0076749079
## 2010-01-06 -0.005164444 -0.025531952 -0.0137375761
## 2010-01-07  0.050483868 -0.023554762  0.0072962332
## 2010-01-08  0.021309847  0.013243037 -0.0009698085
## 2010-01-11  0.009592278 -0.001512718  0.0077318707
```

```
tail(R)
```

```
##                     AAPL           AEP          BAX           ED            F
## 2022-08-12  0.0211993076  0.0251745371  0.008137519  0.006376787  0.021869038
## 2022-08-15  0.0063135312  0.0074877859  0.025474941  0.011835676  0.008615438
## 2022-08-16 -0.0009242852 -0.0009568368 -0.002906509  0.005212492  0.006717583
## 2022-08-17  0.0087462666 -0.0012452418 -0.007466339  0.005583315 -0.024646903
## 2022-08-18 -0.0022942884  0.0003833349 -0.005390881  0.003473757  0.007458021
## 2022-08-19 -0.0152170612  0.0054464707  0.003270665 -0.003772061 -0.016859594
##                      GE         GOOG           MCD
## 2022-08-12  0.012969998  0.023344170  0.0111227091
## 2022-08-15 -0.001502467  0.001873458  0.0123575699
## 2022-08-16  0.015664194 -0.003015569  0.0031971403
## 2022-08-17 -0.014286870 -0.018037815  0.0019883257
## 2022-08-18 -0.008923568  0.004477999 -0.0008999626
## 2022-08-19 -0.018989903 -0.022931777 -0.0001499775
```

```
dim(R)
```

```
## [1] 3180    8
```

```
# Used 1 month US T-bill rate as the risk free interest rate
tbill <- read_excel("/Users/jameswang/Documents/School Work/ORIE 5630/Final Project/Trea
suryYield.xlsx", col_names = TRUE)
tbill = tbill[tbill$Date >= "2006-01-10" & tbill$Date <= "2019-06-28", ]
tbill[is.na(tbill)] = 0
```

CAPM Method

```
# load Fama-French three-factor model
FF = read.csv("/Users/jameswang/Documents/School Work/ORIE 5630/Final Project/FF-5 facto
rs.csv", header = TRUE)
FF = FF[FF$date >= "20100104" & FF$date <= "20220821", ]
FF[is.na(FF)] = 0
dim(FF)
```

```
## [1] 3180    7
```

```
R["Mkt.RF"] <- FF$Mkt.RF
R["SMB"] <- FF$SMB
R["HML"] <- FF$HML
R["RMW"] <- FF$RMW
R["CMA"] <- FF$CMA
R["RF"] <- FF$RF
R["r"] <- 100*as.matrix(R[1:8]) - R$RF
# Split the data set into training set and test set
set.seed(1)
sample <- sample(c(TRUE, FALSE), nrow(R), replace=TRUE, prob=c(0.7,0.3))
R_train <- R[sample, ]
R_test <- R[!sample, ]
```

```
# Applying Training Data
# fitting a five factor fama-french model
fitLM = lm(r~Mkt.RF + SMB + HML + RMW + CMA, data = R_train)
summaryLM = summary(fitLM)
summaryLM[[7]]
```

```
##
## Call:
## lm(formula = GOOG ~ Mkt.RF + SMB + HML + RMW + CMA, data = R_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5596 -0.5042 -0.0263  0.4718 11.9483
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01651    0.02183   0.756  0.44967
## Mkt.RF       1.03827    0.02212  46.928  < 2e-16 ***
## SMB         -0.19701    0.03935  -5.006 5.99e-07 ***
## HML         -0.17468    0.03746  -4.663 3.30e-06 ***
## RMW          0.16961    0.05458   3.108  0.00191 **
## CMA         -0.68701    0.07409  -9.273  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.027 on 2221 degrees of freedom
## Multiple R-squared:  0.5642, Adjusted R-squared:  0.5632
## F-statistic:   575 on 5 and 2221 DF,  p-value: < 2.2e-16
```

```
#fitting three factor model as comparison
fitLM2 = lm(r~Mkt.RF + SMB + HML, data = R_train)
summaryLM2 = summary(fitLM2)
summaryLM2[[7]]
```

```
##
## Call:
## lm(formula = GOOG ~ Mkt.RF + SMB + HML, data = R_train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5050 -0.5164 -0.0320  0.4994 12.2401
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01013    0.02225   0.455    0.649
## Mkt.RF       1.07046    0.02160  49.551  < 2e-16 ***
## SMB         -0.20466    0.03817  -5.362 9.08e-08 ***
## HML         -0.35272    0.02905 -12.142  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.048 on 2223 degrees of freedom
## Multiple R-squared:  0.5454, Adjusted R-squared:  0.5448
## F-statistic: 889.1 on 3 and 2223 DF,  p-value: < 2.2e-16
```

```
# coefficient estimation of the three factor model
res = residuals(fitLM)
betas = fitLM$coefficients[2:6,]
signif(betas, 2)
```

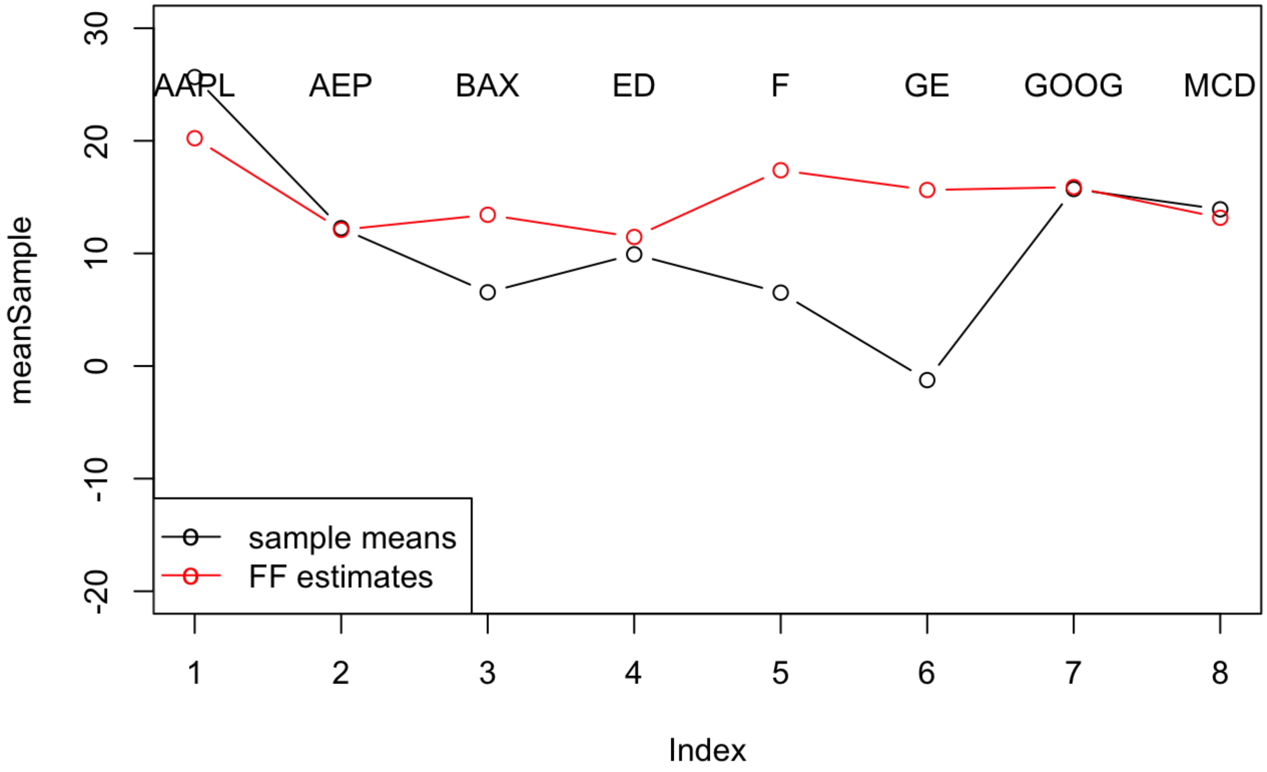
```
##           AAPL     AEP   BAX     ED      F     GE  GOOG    MCD
## Mkt.RF  1.200  0.6200  0.76  0.550 1.2000  1.100  1.00  0.730
## SMB    -0.210 -0.3700 -0.27 -0.410 0.4500  0.120 -0.20 -0.140
## HML    -0.510  0.0031 -0.12 -0.019 0.6100  0.710 -0.17  0.016
## RMW     0.550  0.3900  0.13  0.400 0.1400 -0.065  0.17  0.430
## CMA     0.031  0.5500  0.43  0.720 0.0056  0.260 -0.69  0.120
```

```
# Prediction on testing set
pred <- predict(fitLM2, newdata = R_test)
# Calculating MSPE(mean squared prediction error)
c(MSPE_train = mean((R_train$r - predict.lm(fitLM2, R_train)) ^ 2),
  MSPE_test = mean((R_test$r - predict.lm(fitLM2, R_test)) ^ 2))
```

```
## MSPE_train  MSPE_test
##   1.998861   1.945220
```

```
# Finding the sample mean and the mean estimated by FF
n = nrow(R)
meanFF = 253*(R$RF[n] + t(betas) %*% (colMeans(R[9:13])))
meanSample = 253*colMeans(R$r,na.rm = TRUE)
```

```
plot(meanSample, type="b", ylim=c(-20, 30))
lines(1:8, meanFF, type="b", col="red")
for (j in 1:8)
{
  text(j, 25, strtrim(names(R[j]),4))
}
legend("bottomleft", c("sample means", "FF estimates"), pch="o", lty=1, col=c("black",
"red"))
```



```
# Covariance Matrix estimated by Famma French
N = 3180
sigF5 = as.matrix(var(cbind(R_test$Mkt.RF,
                            R_test$SMB,
                            R_test$HML,
                            R_test$RMW,
                            R_test$CMA)))
sigeps5 = crossprod(res)/(N-6)
sigeps5 = diag(diag(sigeps5))
cov_FF = t(betas) %*% sigF5 %*% (betas) + sigeps5
cov_FF
```

```
##           AAPL       AEP       BAX        ED         F        GE      GOOG
## AAPL 3.1167971 0.8916731 1.2232059 0.7490709 1.8087718 1.5862357 1.9117797
## AEP  0.8916731 1.2293974 0.6228380 0.4911947 0.9526087 0.9153808 0.7878356
## BAX  1.2232059 0.6228380 1.7128064 0.5435493 1.2569368 1.1637571 1.1317392
## ED   0.7490709 0.4911947 0.5435493 1.1908642 0.8025233 0.7903540 0.6384393
## F    1.8087718 0.9526087 1.2569368 0.8025233 4.2017729 2.3352057 1.7637896
## GE   1.5862357 0.9153808 1.1637571 0.7903540 2.3352057 3.8037257 1.5415609
## GOOG 1.9117797 0.7878356 1.1317392 0.6384393 1.7637896 1.5415609 2.5928558
## MCD  1.1581629 0.5958817 0.7515384 0.5178257 1.2359961 1.1298573 1.0736509
##            MCD
## AAPL 1.1581629
## AEP  0.5958817
## BAX  0.7515384
## ED   0.5178257
## F    1.2359961
## GE   1.1298573
## GOOG 1.0736509
## MCD  1.3494977
```

```
# Sample Covariance Matrix
cov_sample <- cov(R$r)
cov_sample
```

```
##           AAPL       AEP       BAX        ED         F        GE      GOOG
## AAPL 3.1992918 0.5609036 0.8721414 0.4399083 1.4270972 1.2455059 1.6509382
## AEP  0.5609036 1.4458856 0.5988618 1.1299393 0.6152814 0.6825282 0.5132121
## BAX  0.8721414 0.5988618 1.9711668 0.5762858 0.8992229 0.9451165 0.8948237
## ED   0.4399083 1.1299393 0.5762858 1.4152456 0.5227168 0.5826824 0.3969391
## F    1.4270972 0.6152814 0.8992229 0.5227168 4.4626714 2.2334353 1.3518373
## GE   1.2455059 0.6825282 0.9451165 0.5826824 2.2334353 4.0483832 1.2018547
## GOOG 1.6509382 0.5132121 0.8948237 0.3969391 1.3518373 1.2018547 2.7845829
## MCD  0.8499815 0.5334708 0.5985095 0.4980781 1.0312999 0.9623700 0.8146396
##            MCD
## AAPL 0.8499815
## AEP  0.5334708
## BAX  0.5985095
## ED   0.4980781
## F    1.0312999
## GE   0.9623700
## GOOG 0.8146396
## MCD  1.4221366
```

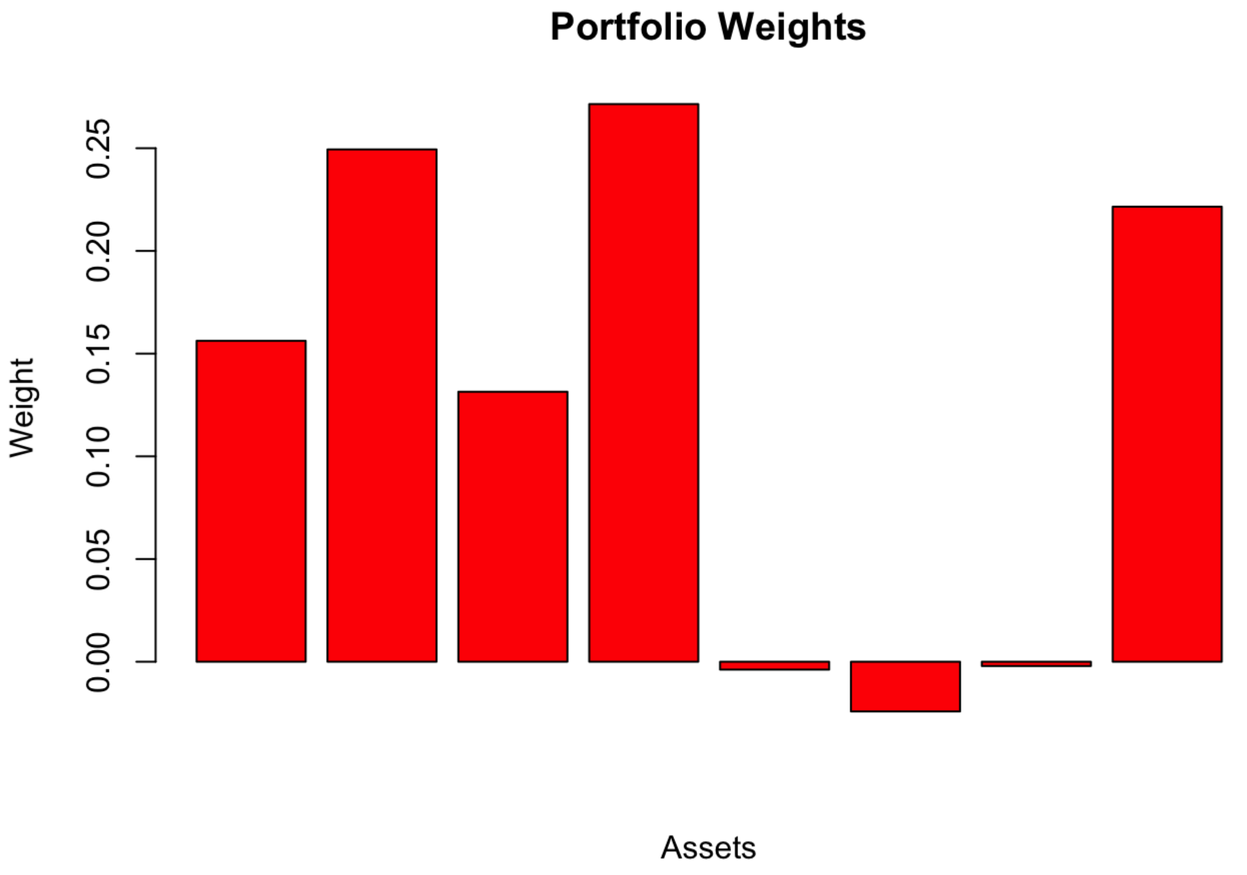
```
# compute tangency portfolio using Famma French estimation
mu_f = mean(R$RF)
tan.port <- tangency.portfolio(meanFF, cov_FF, mu_f, shorts= TRUE)
tan.port
```

```
## Call:
## tangency.portfolio(er = meanFF, cov.mat = cov_FF, risk.free = mu_f,
##     shorts = TRUE)
##
## Portfolio expected return:     13.49253
## Portfolio standard deviation:  0.924237
## Portfolio weights:
## [1]  0.1562  0.2494  0.1314  0.2715 -0.0038 -0.0242 -0.0021  0.2215
```

```
summary(tan.port, risk.free=mu_f)
```

```
## Call:
## tangency.portfolio(er = meanFF, cov.mat = cov_FF, risk.free = mu_f,
##     shorts = TRUE)
##
## Portfolio expected return:     13.49253
## Portfolio standard deviation:  0.924237
## Portfolio Sharpe Ratio:        14.59656
## Portfolio weights:
## [1]  0.1562  0.2494  0.1314  0.2715 -0.0038 -0.0242 -0.0021  0.2215
```

```
plot(tan.port, col="red")
```



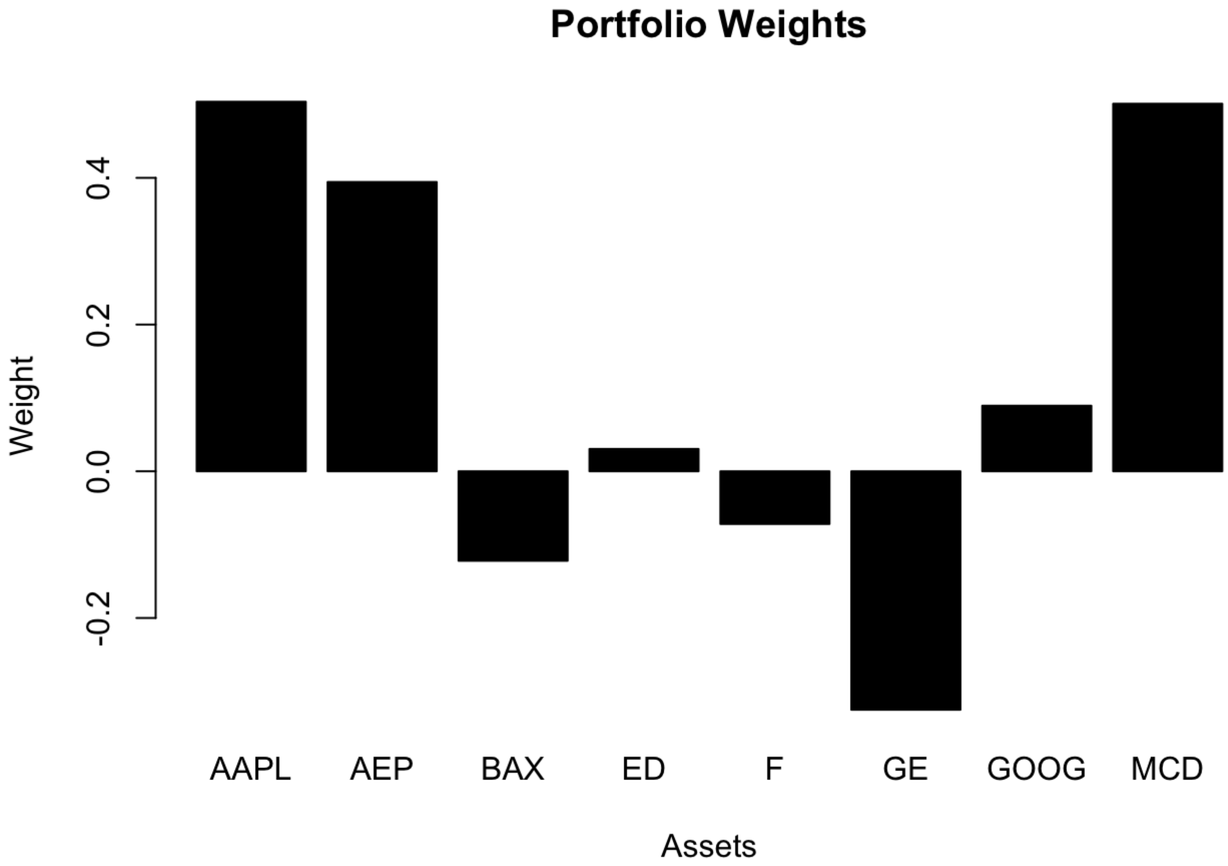
```
# compute tangency portfolio using sample data
tan.port <- tangency.portfolio(meanSample, cov_sample, mu_f, shorts = TRUE)
tan.port
```

```
## Call:
## tangency.portfolio(er = meanSample, cov.mat = cov_sample, risk.free = mu_f,
##     shorts = TRUE)
##
## Portfolio expected return:     25.57207
## Portfolio standard deviation:  1.341451
## Portfolio weights:
##    AAPL     AEP     BAX      ED       F      GE    GOOG     MCD
##  0.5038  0.3943 -0.1220  0.0304 -0.0719 -0.3251  0.0894  0.5011
```

```
summary(tan.port, risk.free=mu_f)
```

```
## Call:
## tangency.portfolio(er = meanSample, cov.mat = cov_sample, risk.free = mu_f,
##     shorts = TRUE)
##
## Portfolio expected return:     25.57207
## Portfolio standard deviation:  1.341451
## Portfolio Sharpe Ratio:        19.06162
## Portfolio weights:
##    AAPL     AEP     BAX      ED       F      GE    GOOG     MCD
##  0.5038  0.3943 -0.1220  0.0304 -0.0719 -0.3251  0.0894  0.5011
```

```
plot(tan.port, col="black")
```



```
# min var portfolio using Famma French estimation
min.port = globalMin.portfolio(meanFF, cov_FF, shorts = FALSE)
attributes(min.port)
```

```
## $names
## [1] "call"    "er"      "sd"      "weights"
##
## $class
## [1] "portfolio"
```

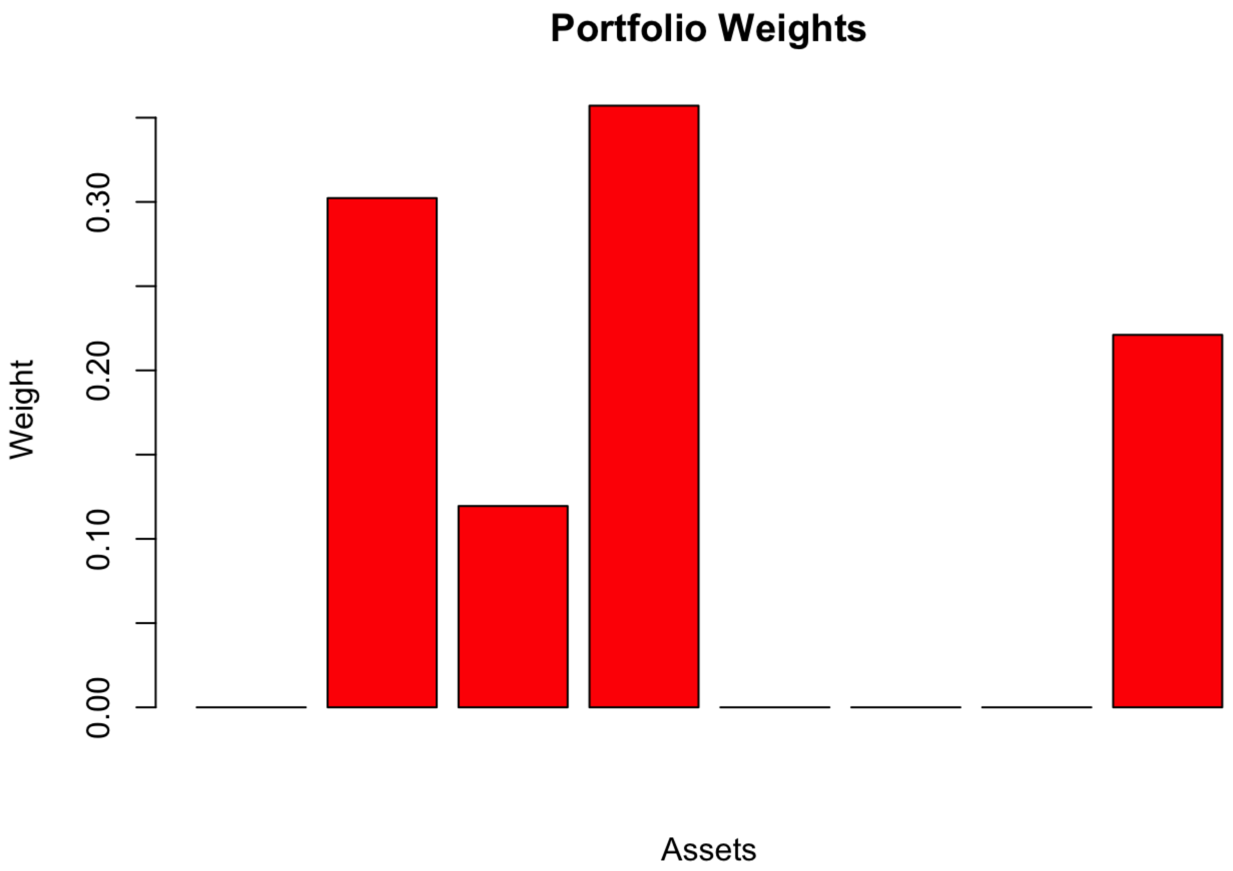
```
print(min.port)
```

```
## Call:
## globalMin.portfolio(er = meanFF, cov.mat = cov_FF, shorts = FALSE)
##
## Portfolio expected return:     12.26455
## Portfolio standard deviation:  0.8678809
## Portfolio weights:
## [1] 0.0000 0.3023 0.1195 0.3571 0.0000 0.0000 0.0000 0.2211
```

```
summary(min.port, risk.free=mu_f)
```

```
## Call:
## globalMin.portfolio(er = meanFF, cov.mat = cov_FF, shorts = FALSE)
##
## Portfolio expected return:     12.26455
## Portfolio standard deviation:  0.8678809
## Portfolio Sharpe Ratio:        14.12947
## Portfolio weights:
## [1] 0.0000 0.3023 0.1195 0.3571 0.0000 0.0000 0.0000 0.2211
```

```
plot(min.port, col="red")
```



```
# min var portfolio using sample mean and cov
min.port2 = globalMin.portfolio(meanSample, cov_sample, shorts = FALSE)
attributes(min.port2)
```

```
## $names
## [1] "call"    "er"      "sd"      "weights"
##
## $class
## [1] "portfolio"
```

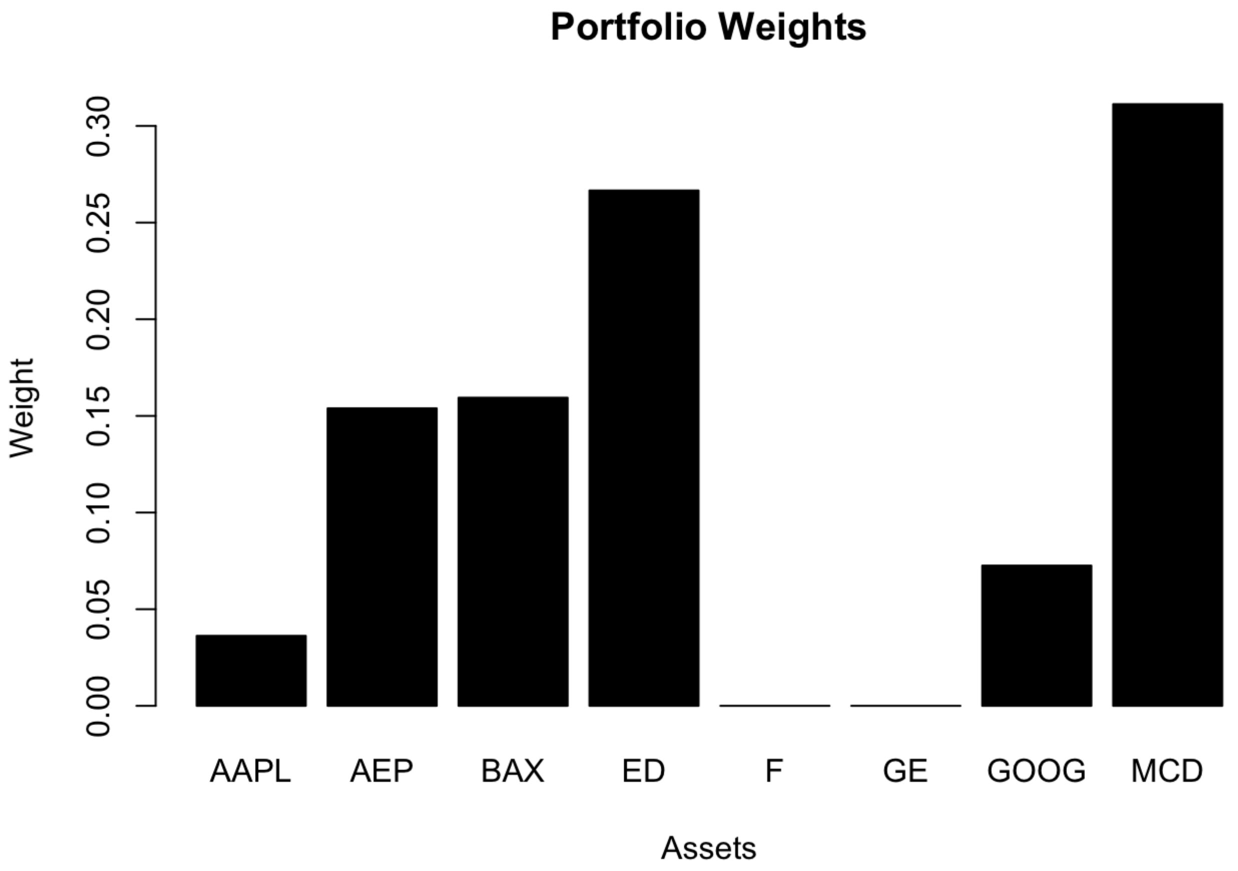
```
print(min.port2)
```

```
## Call:
## globalMin.portfolio(er = meanSample, cov.mat = cov_sample, shorts = FALSE)
##
## Portfolio expected return:     11.97324
## Portfolio standard deviation:  0.9180861
## Portfolio weights:
##   AAPL    AEP    BAX     ED      F     GE   GOOG    MCD
## 0.0362 0.1539 0.1594 0.2666 0.0000 0.0000 0.0726 0.3113
```

```
summary(min.port2, risk.free=mu_f)
```

```
## Call:
## globalMin.portfolio(er = meanSample, cov.mat = cov_sample, shorts = FALSE)
##
## Portfolio expected return:     11.97324
## Portfolio standard deviation:  0.9180861
## Portfolio Sharpe Ratio:        13.0395
## Portfolio weights:
##   AAPL    AEP    BAX     ED      F     GE   GOOG    MCD
## 0.0362 0.1539 0.1594 0.2666 0.0000 0.0000 0.0726 0.3113
```

```
plot(min.port2, col="black")
```


Portfolio Weights
Weight
AAPL
AEP
BAX
ED
F
GE
GOOG
MCD
Assets