Project 2

Yaman Yucel

2023-04-17

Compute returns using the data for project 1

```
a_all <- read.csv("stockData.csv", sep=",", header=TRUE)
# Use 5 year data to train
a <- a_all[1:60,]
#Convert adjusted close prices into returns:
r <- (a[-1,4:ncol(a)]-a[-nrow(a),4:ncol(a)])/a[-nrow(a),4:ncol(a)]
```

Compute means, covariance, variances

```
# Using only 30 stocks(GSPC not included)
n_stocks = 30
#Compute mean vector:
means <- colMeans(r)
#Compute variance covariance matrix
covmat <- cov(r)
#Compute correlation matrix:
cormat <- cor(r)
#Compute the vector of variances:
variances <- diag(covmat)
#Compute the vector of standard deviations:
stdev <- diag(covmat)^.5
#Compute inverse of variance covariance matrix
inv_covmat <- solve(covmat)
#ones vector
ones = rep(1,n_stocks)
```

Part a: Compute A,B,C,D

```
#Compute A,B,C,D
A = as.numeric(t(means) %*% inv_covmat %*% ones)
B = as.numeric(t(means) %*% inv_covmat %*% means)
C = as.numeric(t(ones)  %*% inv_covmat %*% ones)
D = B*C - A^2
print(paste("A:",A))
```

```
## [1] "A: 24.1377672361173"
```

```
print(paste("B :",B))
```

```
## [1] "B : 1.03939838797641"
```

```
print(paste("C :",C))
```

```
## [1] "C : 2464.3207703212"
```

```
print(paste("D :",D))
```

```
## [1] "D : 1978.77922898366"
```

Part b: Compute lambda1 and lambda2

```
E = 0.025

l1 = (function(x) (C*x - A) / D)
l2 = (function(x) (B - A*x) / D)

lambda_1 = l1(E)
lambda_2 = l2(E)
print(paste("Lambda 1 :",lambda_1))
```

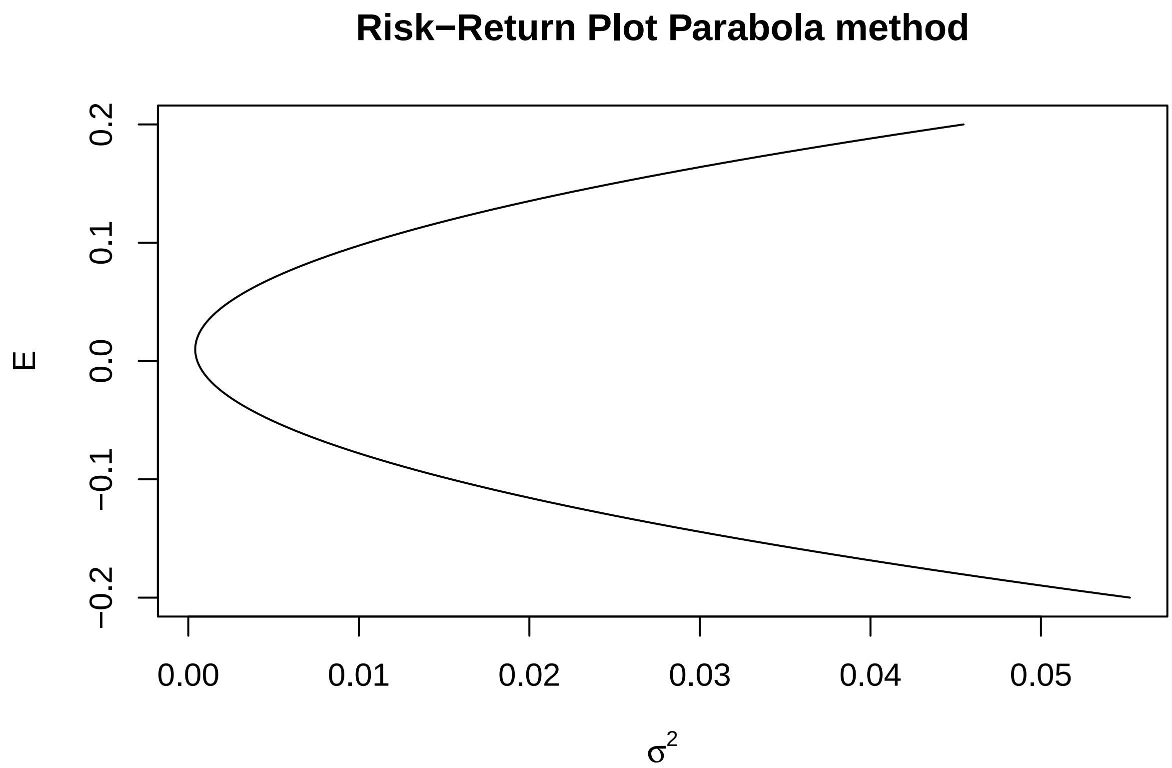
```
## [1] "Lambda 1 : 0.0189360447457082"
```

```
print(paste("Lambda 2 :", lambda_2))
```

```
## [1] "Lambda 2 : 0.000220314727731093"
```

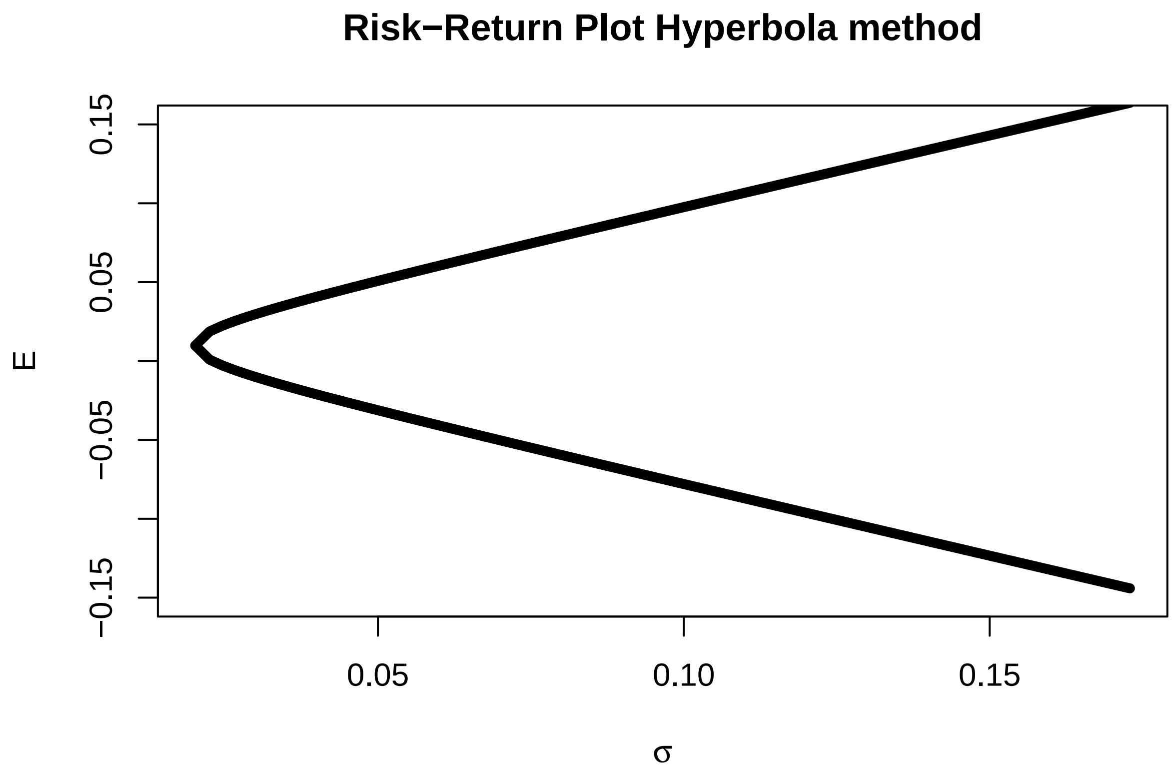
Part c and d: Find composition with expected return E and plot frontier using parabola method

```
#Composition of the efficient portfolio given the return E
investor_weight = inv_covmat %*% (l1(E) * means + l2(E) * ones)

#Span values of E:
E <- seq(-0.2,0.2,.001)

#Compute variance of efficient frontier portfolios (parabola)
var_ef_p <- (C * E^2 - 2*A*E + B) / D

#Parabola: part d
plot(var_ef_p, E, type="l", ylab = 'E', xlab = expression(sigma^2), main="Risk-Return Plot Parabola met
```



Part e: Plot using hyperbola method

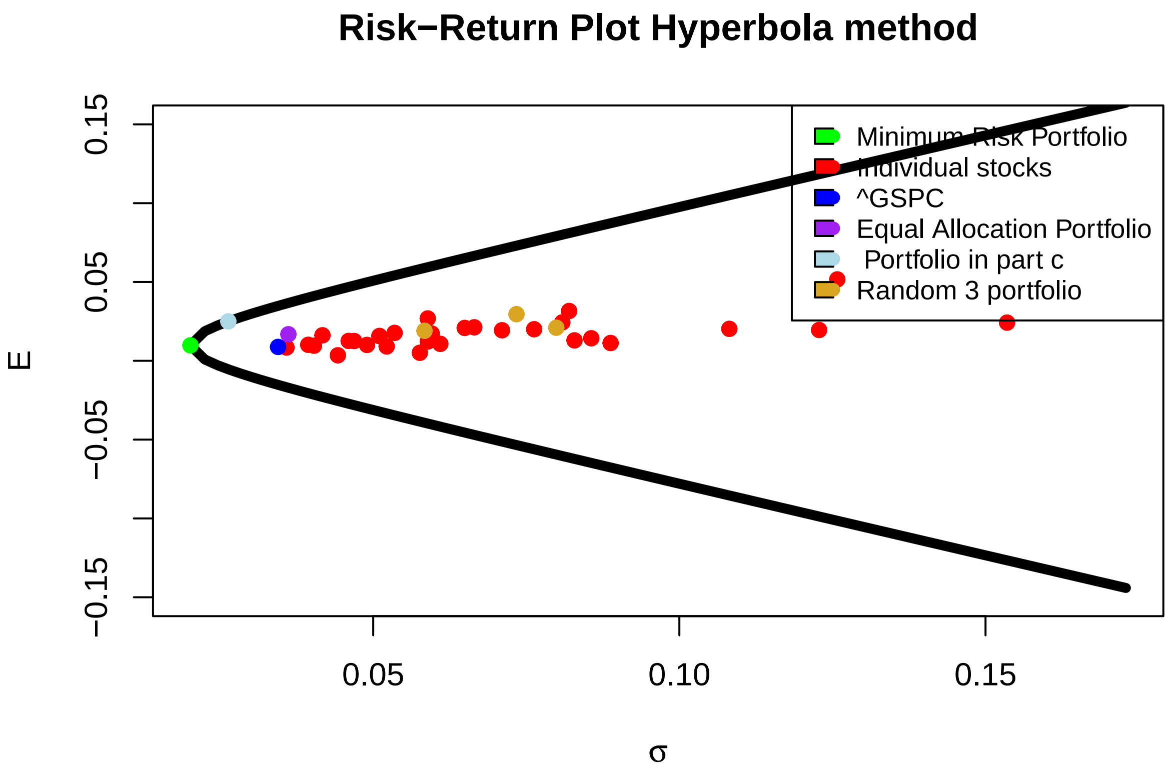
```
#Span values of vars
sigmas <- sqrt(seq(1/C,0.03,.0001))
upper_part <- (A + sqrt(D*(C*sigmas^2 - 1)))*(1/C)
lower_part <- (A - sqrt(D*(C*sigmas^2 - 1)))*(1/C)

#Hyperbola: part e
plot(sigmas, upper_part, lwd=5,type = "l",ylab = 'E', xlab = expression(sigma),ylim = c(-0.15,0.15),mai
lines(sigmas,lower_part, lwd=5,type = "l")
```



Part f and g: Add 30 stocks, GSPC, equal allocation portfolio, min risk portfolio, portfolio in c, 3 random portfolios

```
#f Add n_stocks stocks, GSPC, equal allocation portfolio, min risk portfolio, portfolio in c
plot(sigmas, upper_part, lwd=5,type = "l",ylab = 'E', xlab = expression(sigma),ylim = c(-0.15,0.15),mai
lines(sigmas,lower_part, lwd=5,type = "l")
# Mark stocks
for (i in 1:n_stocks) {
  points(sqrt(covmat[i,i]), means[i], pch = 19, lwd=1, col = "red")
}

#Mark GSPC
r_gpsc = (a[-1,3]-a[-nrow(a),3])/a[-nrow(a),3]
gspc_mean = mean(r_gpsc)
gspc_var = var(r_gpsc)
points(sqrt(gspc_var),gspc_mean, pch = 19, lwd=1, col = "blue")

#Mark Equal allocation portfolio
equal_weight_vector <- ones/n_stocks
equal_varp <- t(equal_weight_vector) %*% covmat %*% equal_weight_vector
equal_Rp <- t(equal_weight_vector) %*% means
points(sqrt(equal_varp), equal_Rp, pch = 19, lwd=1, col= "purple")

#Mark minimum risk portfolio:
min_risk_weight_vector <- inv_covmat  %*% ones /as.numeric(t(ones)  %*% inv_covmat  %*% ones)
min_risk_varp <- t(min_risk_weight_vector) %*% covmat %*% min_risk_weight_vector
min_risk_Rp <- t(min_risk_weight_vector) %*% means
```

```r
points(sqrt(min_risk_varp), min_risk_Rp, pch=19,lwd=1,col="green")

#mark portfolio in c
investor_varp <- t(investor_weight) %*% covmat %*% investor_weight
investor_Rp <- t(investor_weight) %*% means
points(sqrt(investor_varp), investor_Rp, pch = 19, lwd=1, col= "lightblue")

#Part g add 3 random portfolios

#create vector with 10 random numbers between 1 and 20
set.seed(1)
r_portf_x1 <- runif(n=n_stocks, min=-0.75, max=1)
set.seed(2)
r_portf_x2 <- runif(n=n_stocks, min=-0.75, max=1)
set.seed(3)
r_portf_x3 <- runif(n=n_stocks, min=-0.75, max=1)

r_portf_x1 <- r_portf_x1 / sum(r_portf_x1)
r_portf_x2 <- r_portf_x2 / sum(r_portf_x2)
r_portf_x3 <- r_portf_x3 / sum(r_portf_x3)

r_varp1 <- t(r_portf_x1) %*% covmat %*% r_portf_x1
r_Rp1 <- t(r_portf_x1) %*% means
points(sqrt(r_varp1), r_Rp1, pch = 19, lwd=1, col= "goldenrod")

r_varp2 <- t(r_portf_x2) %*% covmat %*% r_portf_x2
r_Rp2 <- t(r_portf_x2) %*% means
points(sqrt(r_varp2), r_Rp2, pch = 19, lwd=1, col= "goldenrod")

r_varp3 <- t(r_portf_x3) %*% covmat %*% r_portf_x3
r_Rp3 <- t(r_portf_x3) %*% means
points(sqrt(r_varp3), r_Rp3, pch = 19, lwd=1, col= "goldenrod")

legend("topright",
       legend=c("Minimum Risk Portfolio", "Individual stocks", "^GSPC","Equal Allocation Portfolio", " I
       col=c("green", "red","blue","purple","lightblue","goldenrod"),
       pch = 19,
       fill =c("green", "red","blue","purple","lightblue","goldenrod"),
       cex=0.8)
```


Risk−Return Plot Hyperbola method
Minimum Risk Portfolio
Individual stocks
^GSPC
Equal Allocation Portfolio
Portfolio in part c
Random 3 portfolio
0.15
0.05
−0.05
−0.15
E
0.05
0.10
0.15
σ