project2

Juan Piao

4/8/2020

a. Use your data to plot the frontier in the mean-variance space.

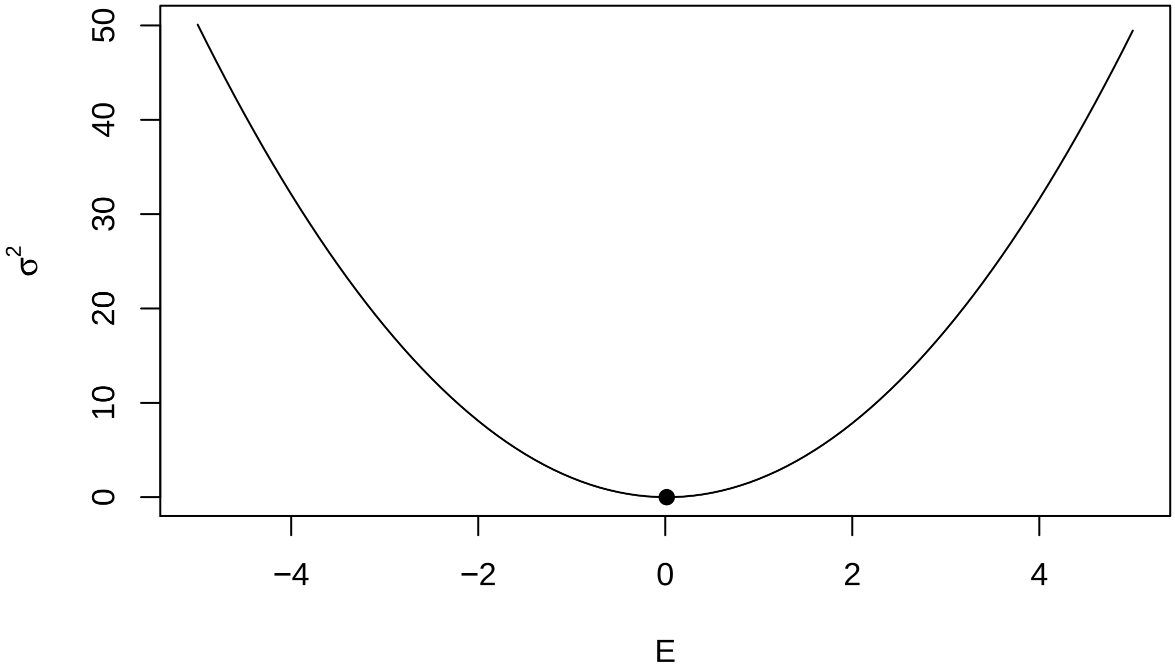
```
#Read your csv file:
a <- read.csv("stockData.csv", sep=",", header=TRUE)
#Convert adjusted close prices into returns:
rr <- (a[-1,4:ncol(a)]-a[-nrow(a),4:ncol(a)])/a[-nrow(a),4:ncol(a)]

#Compute the means:
means <- colMeans(rr)

#Find the covariance matrix:
cov.matrix <- cov(rr)

#Compute the vector of standard deviations:
stdev <- diag(cov.matrix)^.5

ones <- rep(1, 30)

#Compute A:
A <- t(ones) %*% solve(cov.matrix) %*% means


#Compute B:
B <- t(means) %*% solve(cov.matrix) %*% means


#Compute C:
C <- t(ones) %*% solve(cov.matrix) %*% ones


#Compute D:
D <- B*C - A^2
#Give values for E:
E <- seq(-5,5,.1)

#Compute sigma2 as a function of A,B,C,D, and E:
sigma2 <- (C*E^2 - 2*A*E +B) /D
```

```
## Warning in C * E^2: Recycling array of length 1 in array-vector arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

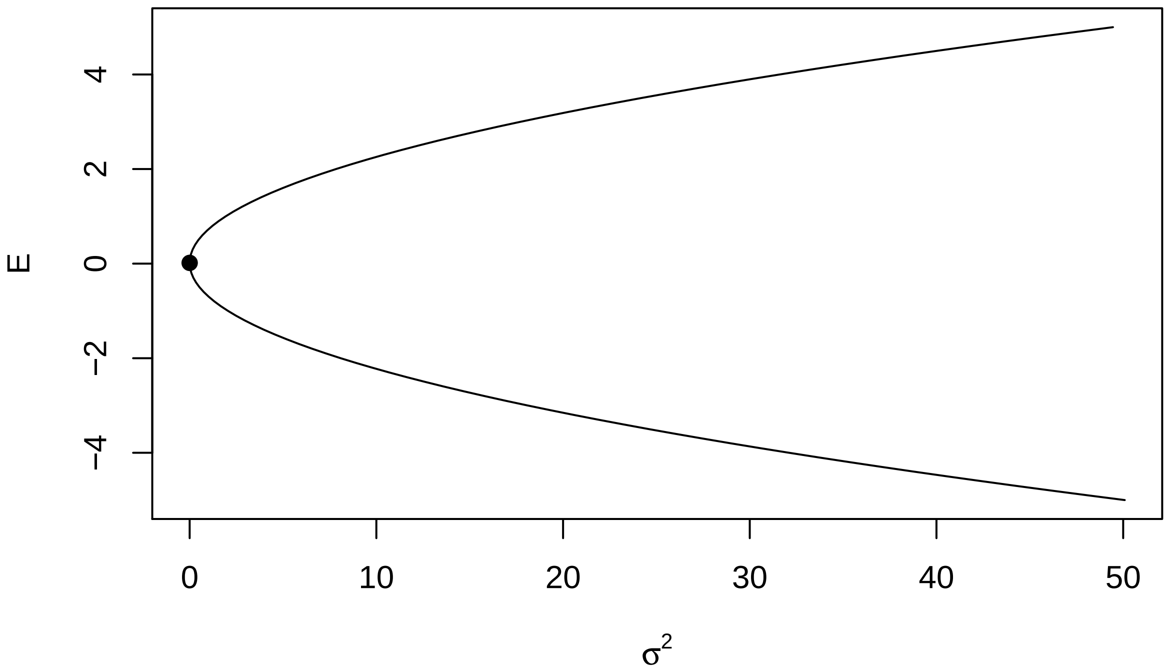
```
## Warning in 2 * A * E: Recycling array of length 1 in array-vector arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
## Warning in C * E^2 - 2 * A * E + B: Recycling array of length 1 in vector-array arithmetic is depreca
##   Use c() or as.vector() instead.

## Warning in (C * E^2 - 2 * A * E + B)/D: Recycling array of length 1 in vector-array arithmetic is dep
##   Use c() or as.vector() instead.
```

```
#Plot sigma2 against E:
plot(E, sigma2, type="l", ylab=expression(sigma^2))

#Add the minimum risk portfolio:
points(A/C, 1/C, pch=19)
```




```
#Or plot E against sigma2:
plot(sigma2, E,type="l", xlab=expression(sigma^2))

#Add the minimum risk portfolio:
points(1/C, A/C, pch=19)
```



b. Use your data to plot the frontier in the mean-standard deviation space:

1. Using the hyperbola method.

See the plot below. The frontier obtained using hyperbola method is in color black.

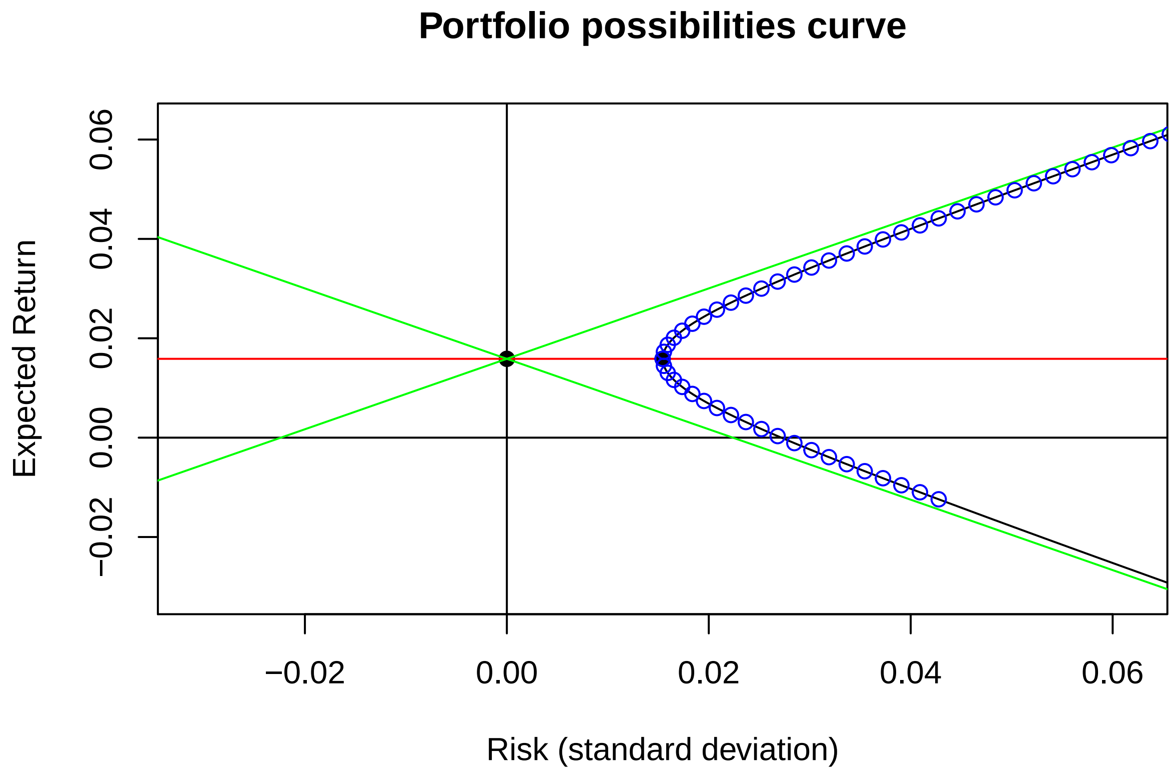
2. By finding two portfolios on the efficient frontier. Note: You will need to choose two values of the risk free asset.

See the plot below. The frontier obtained using by finding two portfolios on frontier is in color blue.

```
#Composition:
x1 <- ( solve(cov.matrix) %*% ones ) / as.numeric( t(ones) %*% solve(cov.matrix) %*% ones )

#Mean:
m1 <- t(x1) %*% means

#Variance:
v1 <- t(x1) %*% cov.matrix %*% x1

#Portfolio 2:  (It doesn't have to be efficient, as long as it is on the frontier).
#Need to choose a value of E.  Let's say, E=0.015.
#To find x2 we use our class notes (see week 2 - lecture 1 notes):
#x2=lambda1*Sigma^-1*means + lambda2*Sigma^-1*ones
#lambda1 = (CE-A)/D and lambda2=(B-AE)/D.

E <- 0.03
lambda1 <- (C*E-A)/D
lambda2 <- (B-A*E)/D

x2=as.numeric(lambda1)*solve(cov.matrix) %*% means +
as.numeric(lambda2)* solve(cov.matrix) %*% ones
```

```r
#Mean:
m2 <- t(x2) %*% means

#Variance:
v2 <- t(x2) %*% cov.matrix %*% x2

#We also need the covariance between portfolio 1 and portfolio 2:
cov_ab <- t(x1) %*% cov.matrix %*% x2

#Now we have two portfolios on the frontier.  We can combine them to trace out the entire frontier:
#Let a be the proportion of investor's wealth invested in portfolio 1.
#Let b be the proportion of investor's wealth invested in portfolio 2.

a <- seq(-3,3,.1)
b <- 1-a

r_ab <- a*m1 + b*m2

var_ab <- a^2*v1 + b^2*v2 + 2*a*b*cov_ab
sd_ab <- var_ab^.5


plot(0, A/C, main = "Portfolio possibilities curve", xlab = "Risk (standard deviation)",
  ylab = "Expected Return", type = "n",
  xlim = c(-2*sqrt(1/C), 4*sqrt(1/C)),
  ylim = c(-2*A/C, 4*A/C))

#Plot center of the hyperbola:
    points(0, A/C, pch = 19)

#Plot transverse and conjugate axes:
    abline(v = 0) #Also this is the y-axis.
    abline(h = A/C, col="red")

#Plot the x-axis:
    abline(h = 0)

#Plot the minimum risk portfolio:
    points(sqrt(1/C), A/C, pch=19)

#Find the asymptotes:
    V <- seq(-1, 1, 0.001)
    A1 <- A/C + V * sqrt(D/C)
    A2 <- A/C - V * sqrt(D/C)
    points(V, A1, type = "l", col="green")
    points(V, A2, type = "l", col="green")

#Efficient frontier:
    minvar <- 1/C
    minE <- A/C
    sdeff <- seq((minvar)^0.5, 1, by = 0.0001)
    options(warn = -1)
    y1 <- (A + sqrt(D*(C*sdeff^2 - 1)))*(1/C)
```

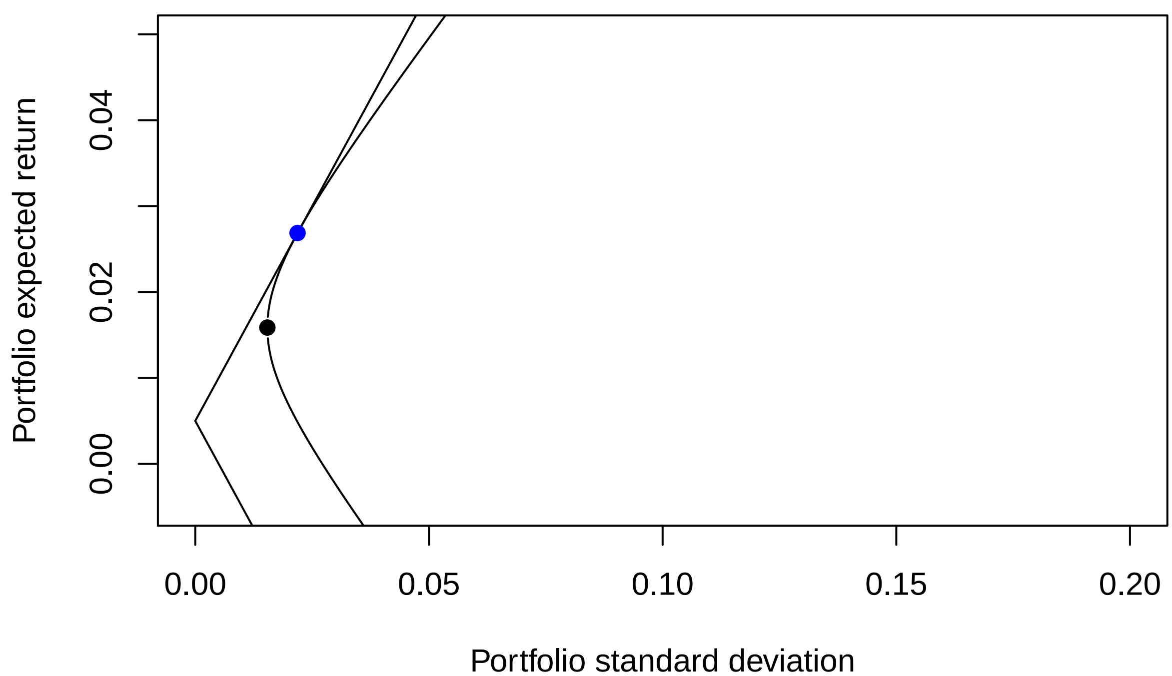
```
    y2 <- (A - sqrt(D*(C*sdeff^2 - 1)))*(1/C)
    options(warn = 0)

    points(sdeff, y1, type = "l")
    points(sdeff, y2, type = "l")


points(sd_ab, r_ab, col="blue")
```



c. Choose a value of Rf , draw the tangent line to the efficient frontier, find the composition of the point of tangency, and the mean and variance of the point of tangency.

```
#Plot the minimum risk portfolio:
    plot(sqrt(1/C), A/C,  xlim=c(0,.2), ylim=c(-.005,.05),pch=19, xlab="Portfolio standard deviation", y


#Efficient frontier:
    minvar <- 1/C
    minE <- A/C
    sdeff <- seq((minvar)^0.5, 1, by = 0.0001)
    options(warn = -1)
    y1 <- (A + sqrt(D*(C*sdeff^2 - 1)))*(1/C)
    y2 <- (A - sqrt(D*(C*sdeff^2 - 1)))*(1/C)
    options(warn = 0)

    points(sdeff, y1, type = "l")
    points(sdeff, y2, type = "l")
```

```r
#Choose risk-free return:
Rf <- 0.005

#Range of expected return:
sigma <- seq(0,.5, .001)

Rp1 <- Rf + sigma*sqrt(C*Rf^2-2*Rf*A+B)

Rp2 <- Rf - sigma*sqrt(C*Rf^2-2*Rf*A+B)

points(sigma, Rp1, type="l")

points(sigma, Rp2, type="l")

#===================================
#===================================
#Point of tangency:
R <- means - Rf
z <- solve(cov.matrix) %*% R

#the composition of the point of tangency
xx <- z/sum(z)
# the mean of the point of tangency
rr <- t(xx) %*% means
# the variance of the point of tangency
varr <- t(xx) %*% cov.matrix %*% xx
sdev <- varr^.5

points(sdev, rr, pch=19, col="blue")
```



```r
xx
```

```
##                [,1]
## AAPL   0.1598911524
## CSCO  -0.1104241473
## MSFT   0.0933535631
## SNE   -0.0794699011
## ADBE   0.1783072471
## MU     0.0108672795
## CWT   -0.0875140290
## SJW    0.2237366317
## NRG   -0.1215312252
## KEP    0.0710850896
## NEE    0.3596013404
## SO    -0.3084326072
## BRK.B  0.5511108217
## V      0.4404663962
## JPM   -0.1479924086
## C      0.0417453996
## MA    -0.2705893021
## LFC    0.0069470660
## JNJ   -0.0007269433
## CVS   -0.2353652933
## ISRG   0.0408267955
## NVO    0.1892911981
## ABT   -0.2287385620
## CI     0.2900275677
## AMZN   0.0792995596
## TM    -0.0258340844
## HD    -0.0471392552
## NKE   -0.0147785581
## TSLA   0.0339442060
## LOW   -0.0919649973
```

```r
rr  # the mean of the point of tangency
```

```
##            [,1]
## [1,] 0.02687105
```

```r
sdev  # the variance of the point of tangency
```

```
##            [,1]
## [1,] 0.02189197
```

The composition of the point of tangency is as listed above.

The mean of the point of tangency is 0.02687105.

The variance of the point of tangency is 0.02189197.

d. Go back to the plot you constructed in project 1 and add the efficient frontier, and the tangency point from (b).

```r
 plot(stdev, means, col="lightblue", pch=19, cex=2, xlim=c(0, 0.3))
text(stdev, means, labels=names(means))
#points(equal_portfolio_sd, equal_portfolio_mean, col="green", pch=19, cex=2)
```

```r
#minimum risk portfolio
points(minvar^0.5, minE, col="red", pch=19, cex=2)

#point of tangency
points(sdev, rr, pch=19)

#Efficient frontier
    points(sdeff, y1, type = "l")
    points(sdeff, y2, type = "l")
```