StatsC283-Project3

Juan Piao

4/22/2020

Access the following data:

These are close monthly prices from January 1986 to December 2003. The first column is the date and P1,P2,P3,P4,P5 represent the close monthly prices for the stocks Exxon-Mobil, General Motors, Hewlett Packard, McDonalds, and Boeing respectively.

```
a <- read.table("http://www.stat.ucla.edu/~nchristo/statistics_c183_c283/statc183c283_5stocks.txt", hea
```

a. Convert the prices into returns for all the 5 stocks.

```
r1 <- (a$P1[-length(a$P1)]-a$P1[-1])/a$P1[-1]

r2 <- (a$P2[-length(a$P2)]-a$P2[-1])/a$P2[-1]

r3 <- (a$P3[-length(a$P3)]-a$P3[-1])/a$P3[-1]

r4 <- (a$P4[-length(a$P4)]-a$P4[-1])/a$P4[-1]

r5 <- (a$P5[-length(a$P5)]-a$P5[-1])/a$P5[-1]

returns <- as.data.frame(cbind(r1,r2,r3,r4,r5))
```

b. Compute the mean return for each stock and the variance-covariance matrix.

```
means<- colMeans(returns)
cov.matrix <- cov(returns)
```

c. Use only Exxon-Mobil and Boeing stocks: For these 2 stocks find the composition, expected return, and standard deviation of the minimum risk portfolio.

```
means_2<-means[c(1, 5)]
cov.matrix_2 <- cov(returns[c(1, 5)])
#Compute A:
A <- rep(1,2) %*% solve(cov.matrix_2) %*% means_2


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


#Compute C:
```

```r
C <- rep(1, 2) %*% solve(cov.matrix_2) %*% rep(1,2)

#Compute D:
#D <- B*C - A^2

# the composition of the minimum risk portfolio
x_2<- solve(cov.matrix_2) %*% rep(1,2) / as.numeric(C)
x_2
```

```
##         [,1]
## r1 0.6393153
## r5 0.3606847
```

```r
# expected return of the minimum risk portfolio
A/C
```

```
##             [,1]
## [1,] 0.003413689
```

```r
# standard deviation of the minimum risk portfolio
sqrt(1/C)
```

```
##            [,1]
## [1,] 0.06478695
```

d. Plot the portfolio possibilities curve and identify the efficient frontier on it.

```r
ones <- rep(1, 5)

#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

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)
```

```r
#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)
```

```
## Warning in V * sqrt(D/C): Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
## Warning in A/C + V * sqrt(D/C): Recycling array of length 1 in array-vector arithmetic is deprecated
##   Use c() or as.vector() instead.
```

```r
    A2 <- A/C - V * sqrt(D/C)
```

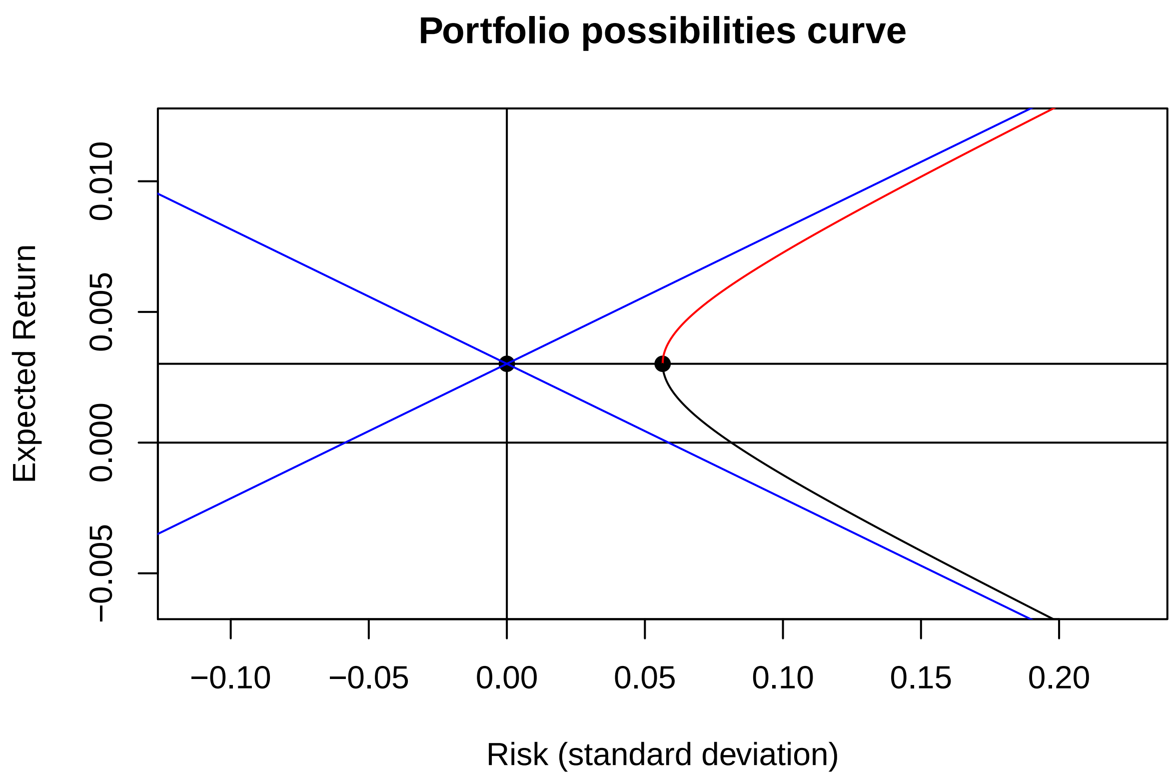
```
## Warning in V * sqrt(D/C): Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
## Warning in A/C - V * sqrt(D/C): Recycling array of length 1 in array-vector arithmetic is deprecated
##   Use c() or as.vector() instead.
```

```r
    points(V, A1, type = "l", col="blue")
    points(V, A2, type = "l", col="blue")

#Efficient frontier:
    minvar <- 1/C
    minE <- A/C
    sdeff <- seq((minvar)^0.5, 1, by = 0.0001)
```

```
## Warning in from + (0L:n) * by: Recycling array of length 1 in array-vector arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```r
    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", col="red")
    points(sdeff, y2, type = "l")
```



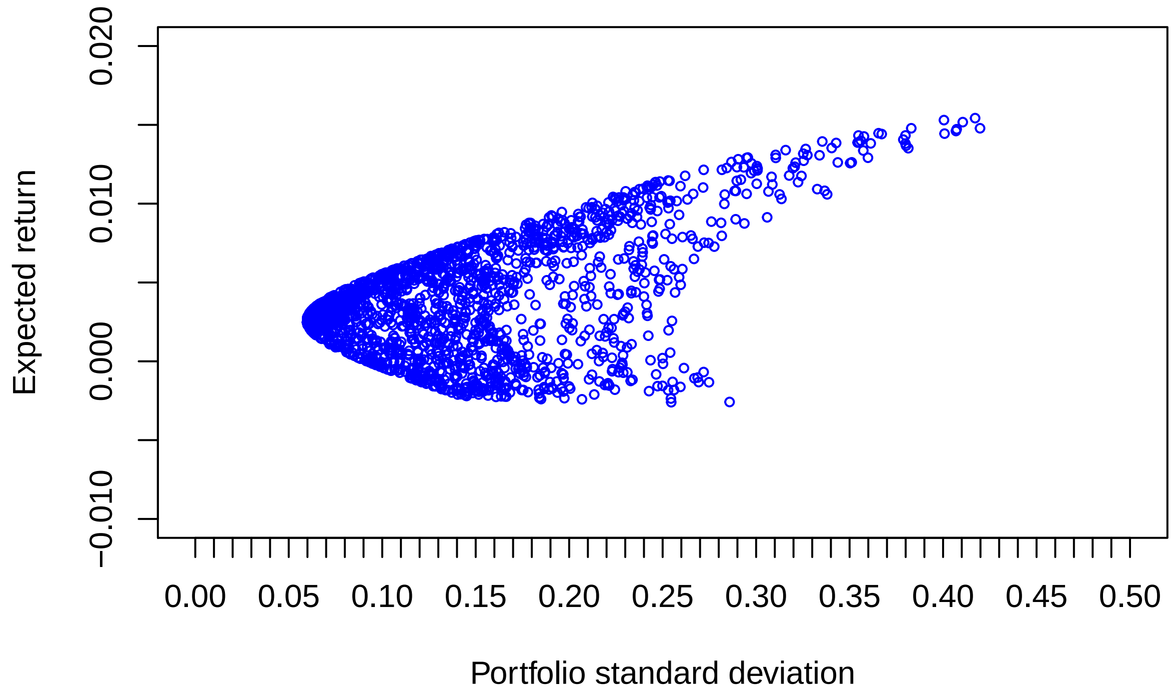
the possibility curve is shown above, the efficient frontier is colored in red.

e. Use only Exxon-Mobil, McDonalds and Boeing stocks and assume short sales are allowed to answer the following question: For these 3 stocks compute the expected return and standard deviation for many combinations of xa, xb, xc with xa + xb + xc = 1 and plot the cloud of points. You can use the following combinations of the three stocks:

```
data <- read.table("http://www.stat.ucla.edu/~nchristo/datac183c283/statc183c283_abc.txt", header=T)
```

```
#Compute the standard deviation of each portfolio:
sigma_p <- ((data$a)^2*diag(cov.matrix)[1]+(data$b)^2*diag(cov.matrix)[4]+(data$c)^2*diag(cov.matrix)[5

#Compute the expected return of each portfolio:
rp_bar <- data$a*means[1]+data$b*means[4]+data$c*means[5]

#Create a matrix with a, b, c, sigma_p, rp_bar:
qq <- cbind(data$a, data$b, data$c, sigma_p, rp_bar)

#Create a matrix with all the points not allowing short sales:
qq2 <- qq[which(qq[,1]>0 & qq[,2]>0 & qq[,3]>0),]

#Create a matrix with all the points allowing short sales:
qq1 <- qq[which(qq[,1]<0 | qq[,2]<0 | qq[,3]<0),]


#Plot the cloud points:
plot(qq[,4], qq[,5], type="n" ,xlim=c(0,0.5), ylim=c(-0.01,0.02), xlab="Portfolio standard deviation", y
```

```
axis(1, at=seq(0, 0.5, 0.01))
axis(2, at=seq(-0.01, 0.02, 0.005))

#points(qq[,4], qq[,5], col="blue", cex=0.6)

points(qq1[,4], qq1[,5], col="blue", cex=0.6)

points(qq2[,4], qq2[,5], col="blue", cex=0.6)
```



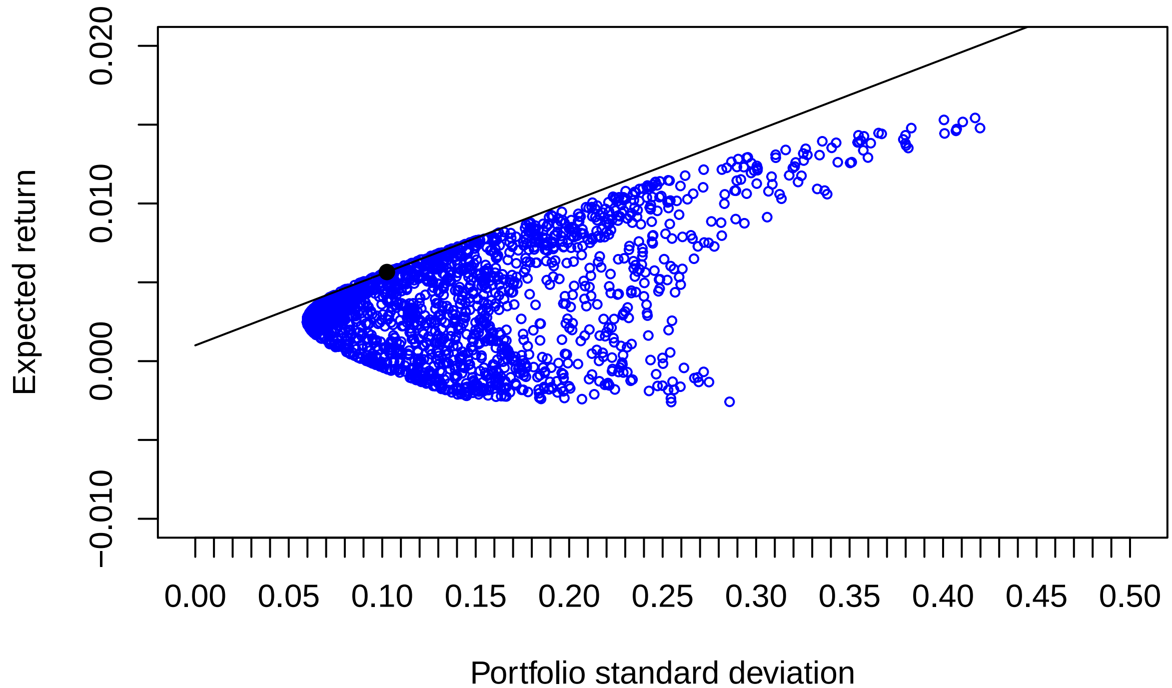
f. Assume Rf = 0.001 and that short sales are allowed. Find the composition, expected return and standard deviation of the portfolio of the point of tangency G and draw the tangent to the efficient frontier of question (e).

```
#Point of tangency:
Rf<-0.001
R <- means[c(1,4,5)]-Rf

cov.matrix_3<- cov(returns[c(1, 4, 5)])
z <- solve(cov.matrix_3) %*% R

#the composition
xx <- z/sum(z)
# expected return
RGbar <- t(xx) %*% means[c(1,4,5)]
#standard deviation
varG <- t(xx) %*% cov.matrix_3 %*% xx
sdevG <- varG ^.5

#Plot the points:
plot(qq[,4], qq[,5], type="n" ,xlim=c(0,0.5), ylim=c(-0.01,0.02), xlab="Portfolio standard deviation", y
```

```r
axis(1, at=seq(0, 0.5, 0.01))
axis(2, at=seq(-0.01, 0.02, 0.005))

#points(qq[,4], qq[,5], col="blue", cex=0.6)

points(qq1[,4], qq1[,5], col="blue", cex=0.6)

points(qq2[,4], qq2[,5], col="blue", cex=0.6)

#######Point of tangency###################
points(sdevG, RGbar, pch=19)


#Compute A:
A <- rep(1,3) %*% solve(cov.matrix_3) %*% means[c(1,4,5)]


#Compute B:
B <- t(means[c(1,4,5)]) %*% solve(cov.matrix_3) %*% means[c(1,4,5)]


#Compute C:
C <- t(rep(1,3)) %*% solve(cov.matrix_3) %*% rep(1,3)

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

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

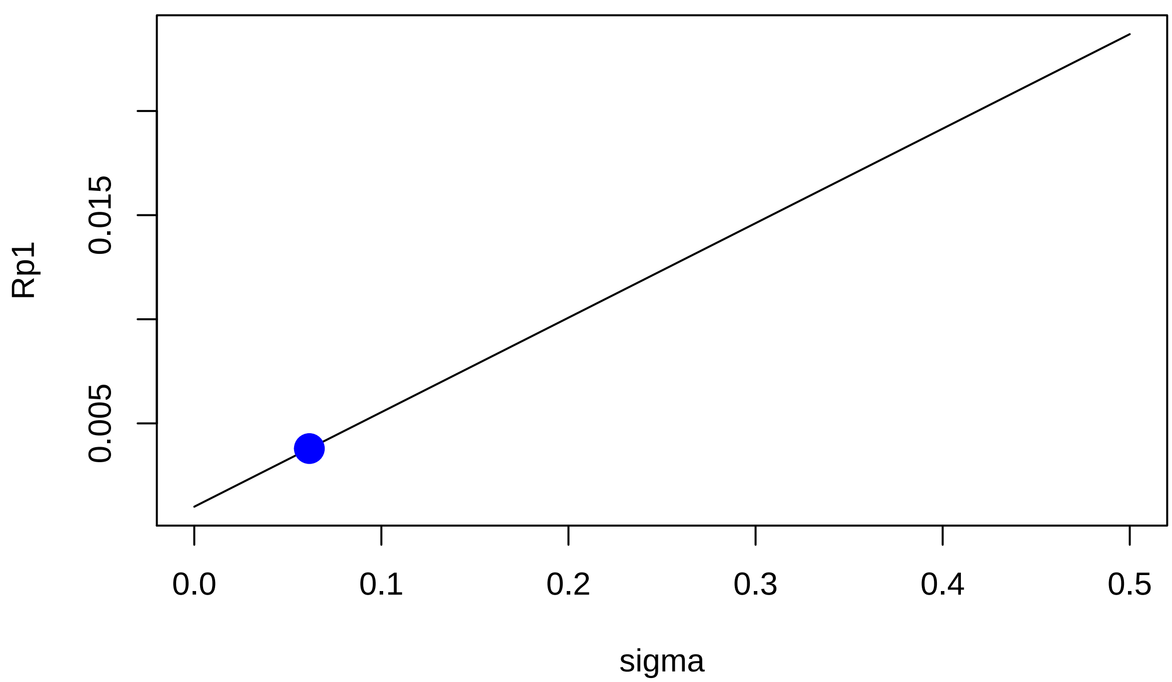
```
## Warning in sigma * sqrt(C * Rf^2 - 2 * Rf * A + B): Recycling array of length 1 in vector-array arit
##   Use c() or as.vector() instead.
```

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

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



g. Find the expected return and standard deviation of the portfolio that consists of 60% G 40% risk free asset. Show this position on the capital allocation line (CAL).

```
sdev_GRf <- 0.6* sdevG
plot(sigma, Rp1, type="l")
points(sdev_GRf, 0.6*RGbar+0.4*Rf, pch=19, col="blue", cex=2)
```



h. Now assume that short sales allowed but risk free asset does not exist.

1. Using Rf1 = 0.001 and Rf2 = 0.002 find the composition of two portfolios A and B (tangent to the efficient frontier - you found the one with Rf1 = 0.001 in question (f)).

portfolio A

```
# portfolios  B
#Point of tangency:
Rf2<-0.002
R2 <- means[c(1,4,5)]-Rf2

z2 <- solve(cov.matrix_3) %*% R2

#the composition
xx2 <- z2/sum(z2)
# expected return
RGbar2 <- t(xx2) %*% means[c(1,4,5)]
#standard deviation
varG2 <- t(xx2) %*% cov.matrix_3 %*% xx2
sdevG2 <- varG2 ^.5
```

2. Compute the covariance between portfolios A and B?

```
cov_ab<-t(xx) %*% cov.matrix_3 %*% xx2
cov_ab
```

```
##             [,1]
## [1,] 0.02264823
```

3. Use your answers to (1) and (2) to trace out the efficient frontier of the stocks Exxon-Mobil, McDonalds, Boeing. Use a different color to show that the frontier is located on top of the cloud of points from question (e). Your graph should look like the one below.

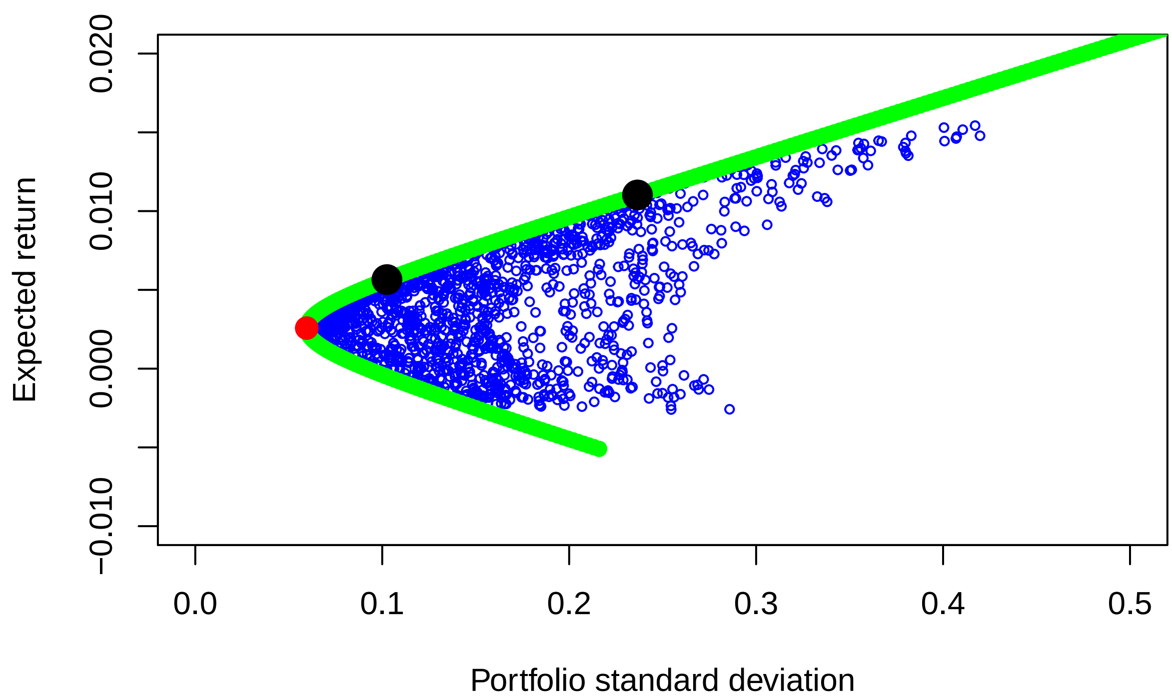
```
#Plot the cloud points:
plot(qq[,4], qq[,5], type="n" ,xlim=c(0,0.5), ylim=c(-0.01,0.02), xlab="Portfolio standard deviation", 

axis(1, at=seq(0, 0.5, 0.1))
axis(2, at=seq(-0.01, 0.02, 0.005))

points(qq[,4], qq[,5], col="blue", cex=0.6)


#Now we have two portfolios on the frontier.  We can combine them to trace out the entire frontier:


a <- seq(-3,3,.02)
b <- 1-a

r_ab <- a*RGbar + b*RGbar2
```

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

## Warning in b * RGbar2: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.
```

```r
var_ab <- a^2*varG + b^2*varG2+ 2*a*b*cov_ab
```

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

## Warning in b^2 * varG2: Recycling array of length 1 in vector-array arithmetic is deprecated.
##   Use c() or as.vector() instead.

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

```r
sd_ab <- var_ab^.5

points(sd_ab, r_ab, col="green", pch=19)

#These are the two portfolios:
points(varG^.5, RGbar, pch=19, cex=2)
points(varG2^.5, RGbar2, pch=19, cex=2)

# minimum risk portfolio
points(sqrt(1/C), A/C, col="red", pch=19, cex=1.5)
```



4. Find the composition of the minimum risk portfolio (how much of each stock) and its expected return, and standard deviation.

```r
# the composition of the minimum risk portfolio
x_3<- solve(cov.matrix_3) %*% rep(1,3) / as.numeric(C)
x_3
```

```
##         [,1]
## r1 0.5269063
## r4 0.2536533
## r5 0.2194404
```

```r
# expected return of the minimum risk portfolio
A/C
```

```
##             [,1]
## [1,] 0.00257322
```

```r
# standard deviation of the minimum risk portfolio
sqrt(1/C)
```

```
##             [,1]
## [1,] 0.05961942
```