Stats C183 Project 2

Charles Liu (304804942)

4/13/2020

Load Necessary Packages:

```r
library(readr)
```

Project 1:

1A)

```r
# Read your csv file:
a <- read.csv("C:/Users/cliuk/Documents/UCLA Works/UCLA Spring 2020/Stats C183/Project/stockData.csv", s
```

1B)

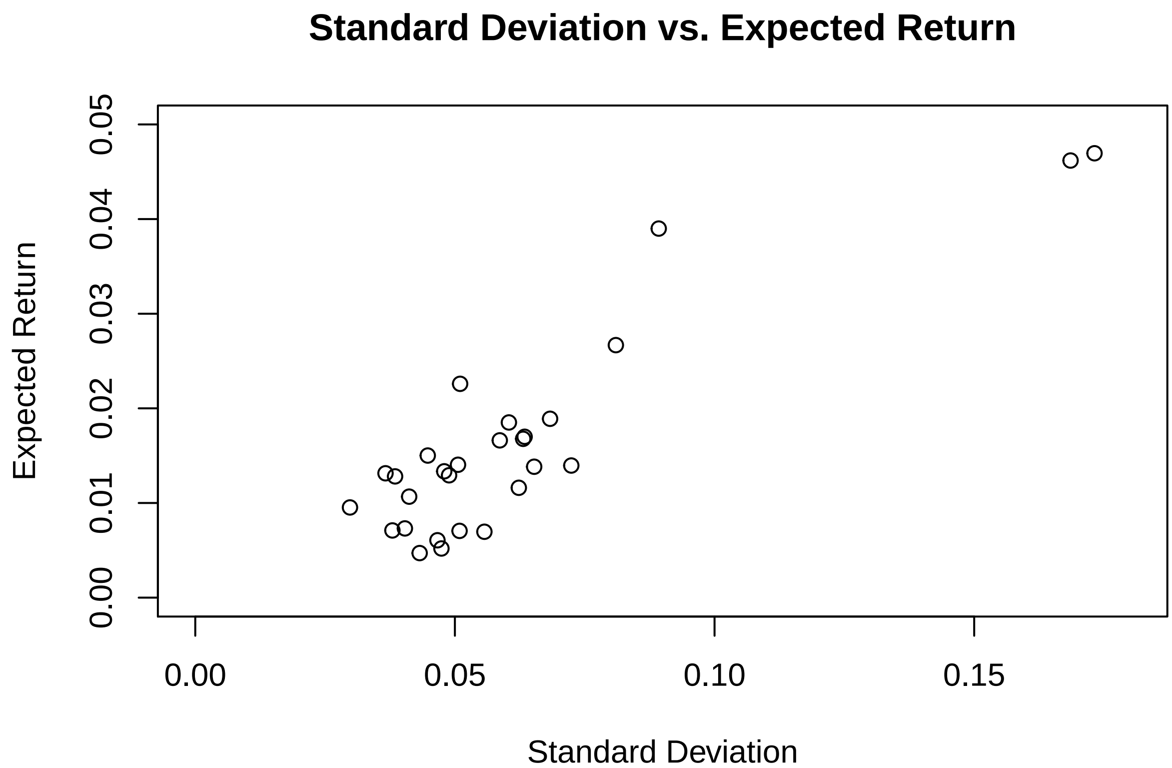
```r
# Convert adjusted close prices into returns:
r <- (a[-1,3:ncol(a)]-a[-nrow(a),3:ncol(a)])/a[-nrow(a),3:ncol(a)]
```

1C)

```r
# Compute mean vector:
means_31 <- colMeans(r) # With ^GSPC

# Compute variance covariance matrix:
covmat_31 <- cov(r) # With ^GSPC

# Compute correlation matrix:
cormat_31 <- cor(r) # With ^GSPC

# Compute the vector of variances:
variances_31 <- diag(covmat_31)

# Compute the vector of standard deviations:
stdev_31 <- diag(covmat_31)^.5
```

1D)

```r
plot(stdev_31, means_31, xlim = c(0, 0.18), ylim = c(0, 0.05),
     main = "Standard Deviation vs. Expected Return",
     xlab = "Standard Deviation", ylab = "Expected Return")
```



1E)

```r
# Compute mean vector:
means <- colMeans(r[-1]) # Without ^GSPC

# Compute variance covariance matrix:
covmat <- cov(r[-1]) # Without ^GSPC

# Compute correlation matrix:
cormat <- cor(r[-1]) # Without ^GSPC

# Compute the vector of variances:
variances <- diag(covmat)

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

# Equal Allocation Formulas:
x <- rep(1/30, 30)
R_equal <- t(x) %*% means
sigma_equal <- t(x) %*% covmat %*% x

# Equal Allocation Numbers:
R_equal
```

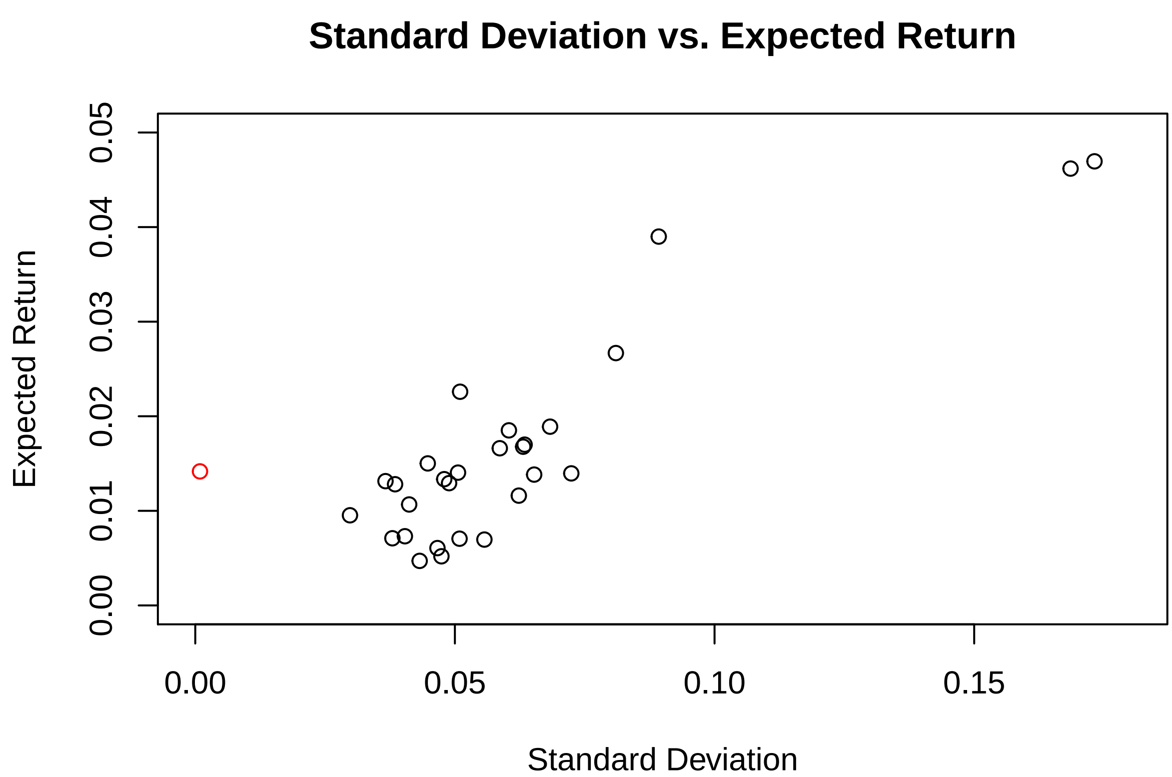
```
##              [,1]
```

```
## [1,] 0.01416942
```

```r
sigma_equal
```

```
##                 [,1]
## [1,] 0.0009098234
```

```r
# Plot Equal Allocation point to part C:
par(mfrow = c(1,1))
plot(stdev_31, means_31, xlim = c(0, 0.18), ylim = c(0, 0.05),
     main = "Standard Deviation vs. Expected Return",
     xlab = "Standard Deviation", ylab = "Expected Return")
points(sigma_equal, R_equal, col = "red")
```



1F)

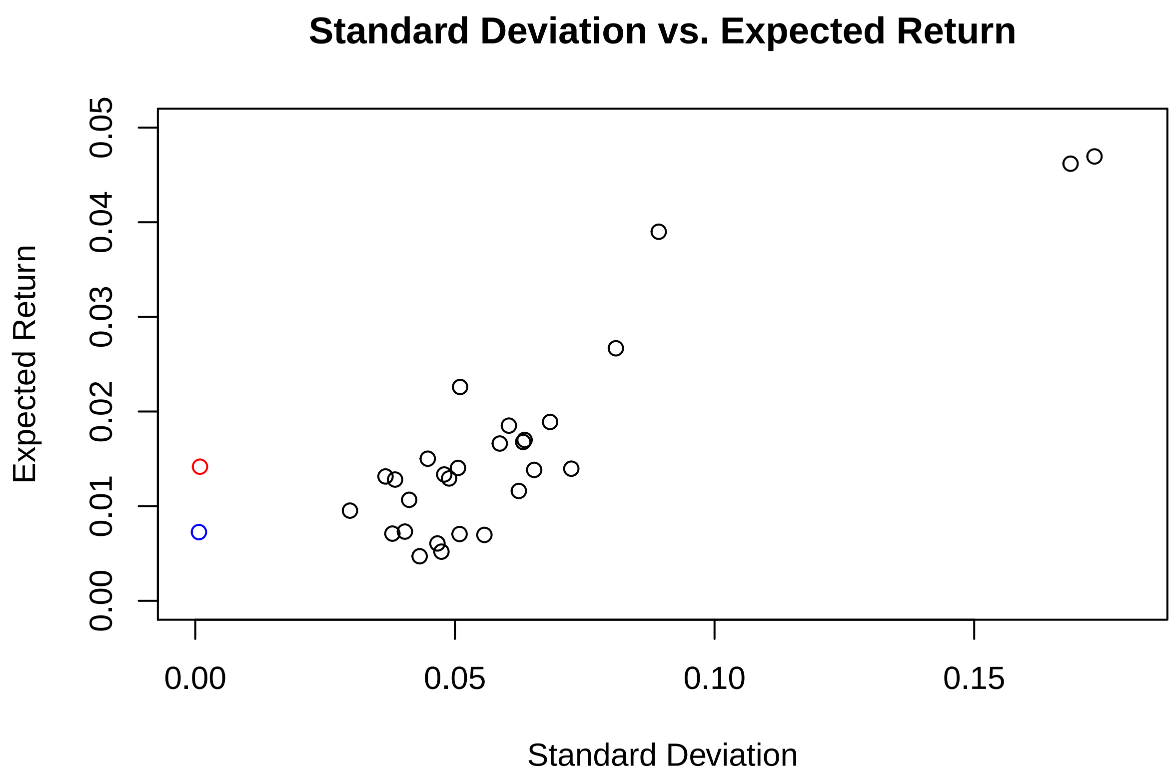
```r
# Min Risk Formulas:
ones <- rep(1, 30)
R_min <- (t(ones) %*% solve(covmat) %*% means)/(t(ones) %*% solve(covmat) %*% ones)
sigma_min <- (1)/((t(ones) %*% solve(covmat) %*% ones)^1/2)

# Min Risk Numbers
R_min
```

```
##                [,1]
## [1,] 0.007260777
```

```
sigma_min
```

```
##              [,1]
## [1,] 0.0007039741
```

```
# Plot Minimum Risk point to part C:
par(mfrow = c(1,1))
plot(stdev_31, means_31, xlim = c(0, 0.18), ylim = c(0, 0.05),
     main = "Standard Deviation vs. Expected Return",
     xlab = "Standard Deviation", ylab = "Expected Return")
points(sigma_equal, R_equal, col = "red")
points(sigma_min, R_min, col = "blue")
```

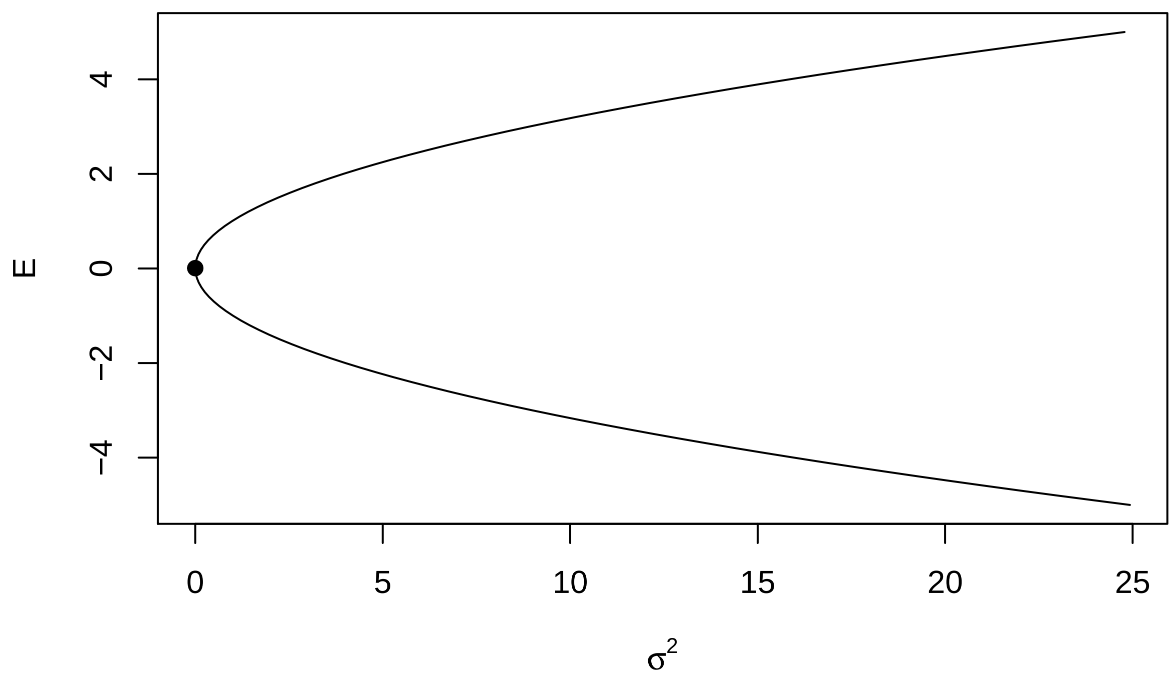


Project 2:

```
# Set up A - E formulas & column of Ones:
ones <- rep(1, 30)
A <- t(ones) %*% solve(covmat) %*% means
B <- t(means) %*% solve(covmat) %*% means
C <- t(ones) %*% solve(covmat) %*% ones
D <- B * C - A^2
E <- seq(-5,5,.1)
```

2A)

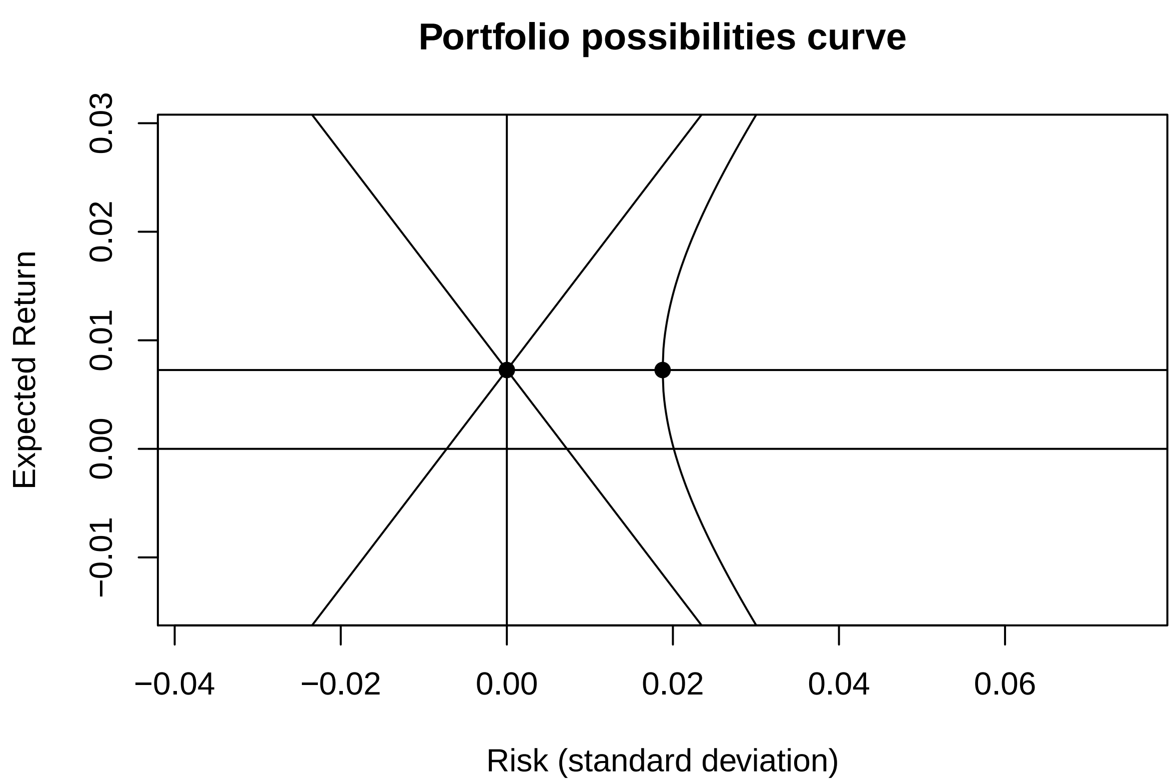
```r
#Compute sigma2 as a function of A,B,C,D, and E:
sigma2 <- (C*E^2 - 2*A*E +B) /D

# 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)
```



2B-i)

```r
### Hyperbola:
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)
```

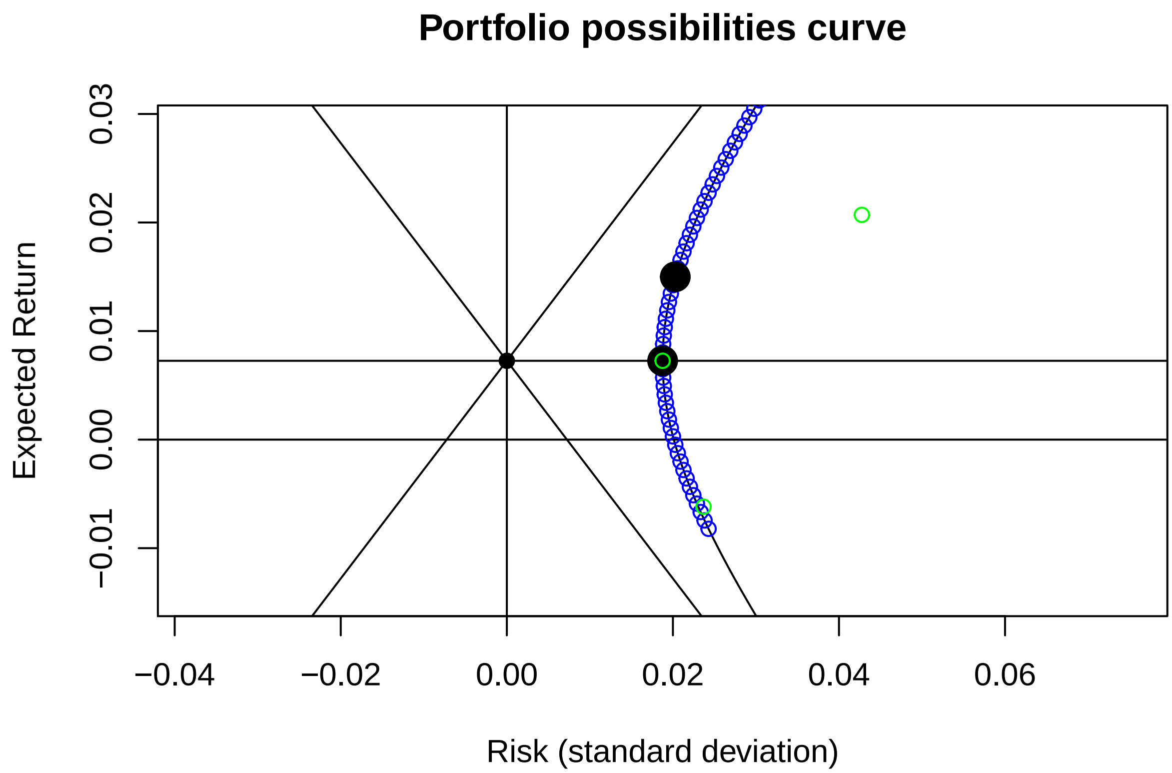
```r
# Plot the x-axis:
abline(h = 0)

# 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)
A2 <- A/C - V * sqrt(D/C)
points(V, A1, type = "l")
points(V, A2, type = "l")

# 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")
```



2B-ii)

```r
# Set up the plot:
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))
points(0, A/C, pch = 19)
abline(v = 0)
abline(h = A/C)
abline(h = 0)
points(sqrt(1/C), A/C, pch=19)
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")
points(V, A2, type = "l")
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")

# ---------------------------------------------------------------------

# Fnding two portfolios on the Efficient Frontier:
x1 <- ( solve(covmat) %*% ones ) / as.numeric( t(ones) %*% solve(covmat) %*% ones )

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

# Variance:
v1 <- t(x1) %*% covmat %*% 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 <- 0.015
lambda1 <- (C*E-A)/D
lambda2 <- (B-A*E)/D
x2=as.numeric(lambda1)*solve(covmat) %*% means +
  as.numeric(lambda2)* solve(covmat) %*% ones

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

# Variance:
```

```r
v2 <- t(x2) %*% covmat %*% x2

# We also need the covariance between portfolio 1 and portfolio 2:
cov_ab <- t(x1) %*% covmat %*% 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
points(sd_ab, r_ab, col="blue")

# These are the two portfolios:
points(v1^.5, m1, pch=19, cex=2)
points(v2^.5, m2, pch=19, cex=2)

# Note:  The two portfolios must be on the frontier!  For example, suppose we chose the equal allocatio
xe <- rep(1/3, 30)

# Compute its mean:
m3 <- t(xe) %*% means

# And its variance:
v3 <- t(xe) %*% covmat %*% xe

# Now, suppose we combine portfolio 1 and the equal allocation portfolio:
r_abc <- a*m1 + b*m3

# To find the variance of the combination we also need the covariance between the two portfolios:
cov_abc <- t(x1) %*% covmat %*% xe
var_abc <- a^2*v1 + b^2*v3 + 2*a*b*cov_abc
sd_abc <- var_abc^.5
points(sd_abc, r_abc, col="green")
```

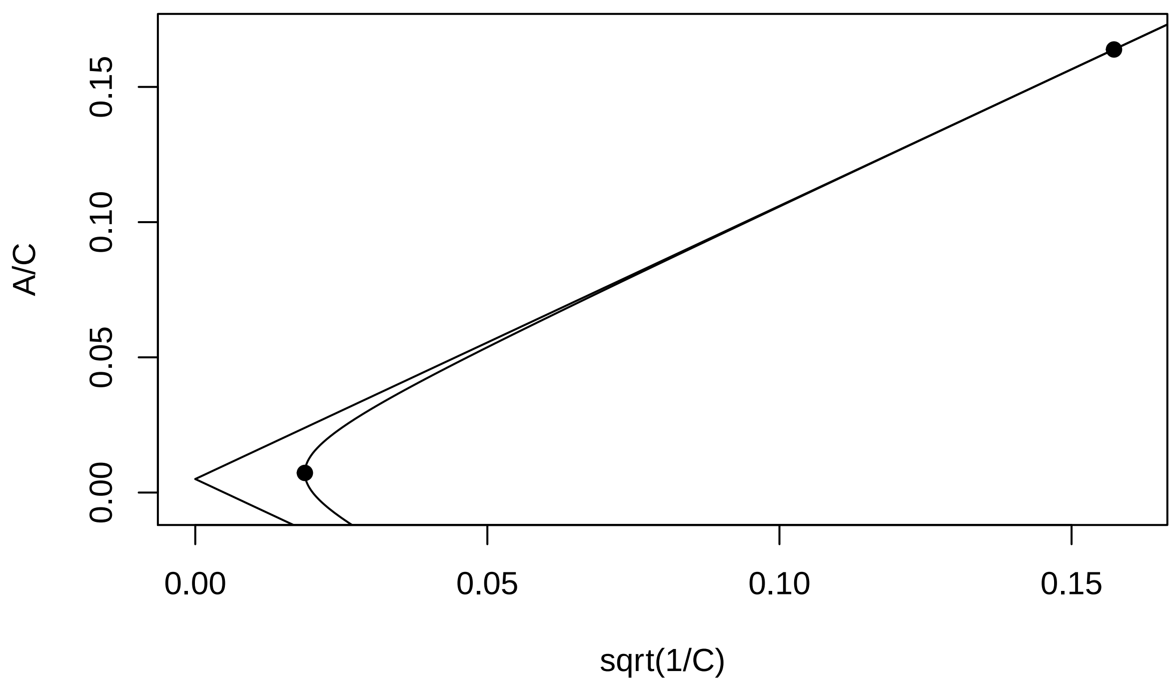



```
# We see that combinations of portfolio 1 and the equal allocation portfolio are not on the frontier.
```

2C)

```
# Set up the plot:
plot(sqrt(1/C), A/C,  xlim=c(0,0.16), ylim=c(-.005,0.17),pch=19)
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")

# 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)
```

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

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

#====================================
#====================================
# Point of tangency:
R <- means-Rf
z <- solve(covmat) %*% R
xx <- z/sum(z)
rr <- t(xx) %*% means
varr <- t(xx) %*% covmat %*% xx
sdev <- varr^.5
points(sdev, rr, pch=19)
```




```r
# Point of tangency values for composition, mean, and variance:
xx
```

```
##             [,1]
## BHP   -1.38429303
## GOLD  -0.28815823
## VALE  -0.03852787
## GOOG  -0.53838586
## T      0.53214551
## NFLX   0.07668800
## AMZN   0.89202985
```

```
## MCD   -0.19436359
## TSLA   0.11683207
## WMT   -0.80579089
## KO    -3.78756211
## COST  -1.85679162
## XOM   -5.23400404
## CVX    1.76387439
## TRP    1.40306402
## BRK.B  0.03666775
## V      2.59256541
## JPM    0.17436872
## JNJ    3.56180140
## AMGN  -1.39160356
## CVS   -0.38644722
## UNP    2.72878782
## BA     0.12116450
## GE     2.59945983
## DLR    0.92057358
## BXP   -2.75630634
## O      1.69028085
## MSFT  -0.83620777
## AAPL   0.75198063
## NVDA   0.53615781
```

```r
rr
```

```
##           [,1]
## [1,] 0.1638496
```

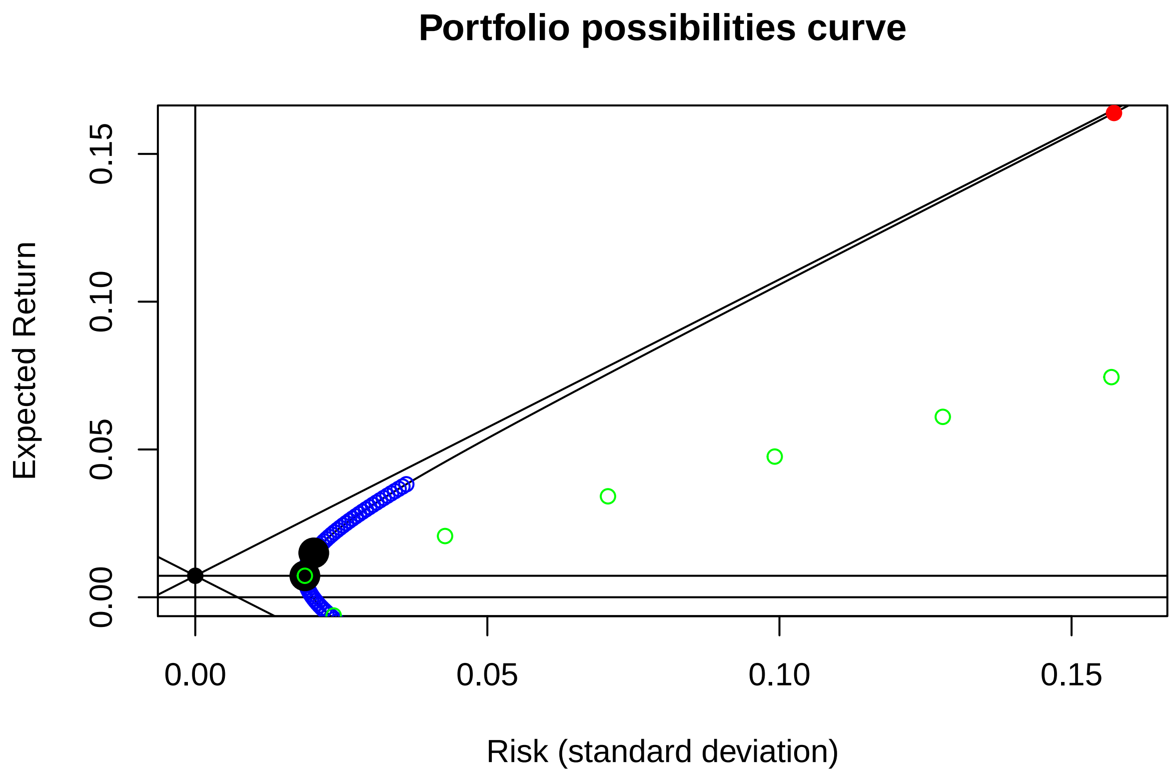
```r
varr
```

```
##            [,1]
## [1,] 0.02473175
```

2D)

```r
# Set up the plot:
plot(0, A/C, main = "Portfolio possibilities curve", xlab = "Risk (standard deviation)",
     ylab = "Expected Return", type = "n",
     xlim = c(0, 0.16),
     ylim = c(0, 0.16))
points(0, A/C, pch = 19)
abline(v = 0)
abline(h = A/C)
abline(h = 0)
points(sqrt(1/C), A/C, pch=19)
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")
points(V, A2, type = "l")
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)
```

```r
y2 <- (A - sqrt(D*(C*sdeff^2 - 1)))*(1/C)
options(warn = 0)
points(sdeff, y1, type = "l")
points(sdeff, y2, type = "l")
#================================================================
x1 <- ( solve(covmat) %*% ones ) / as.numeric( t(ones) %*% solve(covmat) %*% ones )
m1 <- t(x1) %*% means
v1 <- t(x1) %*% covmat %*% x1
E <- 0.015
lambda1 <- (C*E-A)/D
lambda2 <- (B-A*E)/D
x2=as.numeric(lambda1)*solve(covmat) %*% means +
  as.numeric(lambda2)* solve(covmat) %*% ones
m2 <- t(x2) %*% means
v2 <- t(x2) %*% covmat %*% x2
cov_ab <- t(x1) %*% covmat %*% x2
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
points(sd_ab, r_ab, col="blue")
points(v1^.5, m1, pch=19, cex=2)
points(v2^.5, m2, pch=19, cex=2)
xe <- rep(1/3, 30)
m3 <- t(xe) %*% means
v3 <- t(xe) %*% covmat %*% xe
r_abc <- a*m1 + b*m3
cov_abc <- t(x1) %*% covmat %*% xe
var_abc <- a^2*v1 + b^2*v3 + 2*a*b*cov_abc
sd_abc <- var_abc^.5
points(sd_abc, r_abc, col="green")
#================================================================
# Point of tangency:
R <- means-Rf
z <- solve(covmat) %*% R
xx <- z/sum(z)
rr <- t(xx) %*% means
varr <- t(xx) %*% covmat %*% xx
sdev <- varr^.5
points(sdev, rr, pch=19, col = "red")
```


Portfolio possibilities curve
Expected Return
0.00
0.05
0.10
0.15
0.00
0.05
0.10
0.15
Risk (standard deviation)