

Stats C183 HW 1

Charles Liu (304804942)

4/11/2020

Loading Necessary Packages/Data:

```
library(readr)
a <- read.csv("C:/Users/cliuk/Documents/UCLA Works/UCLA Spring 2020/Stats C183/Project/stockData.csv",
```

Create Returns and Matrices:

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

# 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
```

Exercise 1:

1a)

```
# Set up column of Ones, A - E formulas, & both Lagranges:
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)
Lagrange_1 <- ((C * E) - A)/D
Lagrange_2 <- (B - (A * E))/D
```

```
# Values for our formulas:
```

```
Exc_1a <- matrix(c(A, B, C, D), byrow = TRUE)
```

```
rownames(Exc_1a) <- c("A:", "B:", "C:", "D:")
```

```
Exc_1a
```

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

```
## A:    20.627969
```

```
## B:      1.155529
```

```
## C: 2841.013797
```

```
## D: 2857.361175
```

```
Lagrange_1
```

```
## [1] -4.978613512 -4.879185627 -4.779757741 -4.680329856 -4.580901970
```

```
## [6] -4.481474085 -4.382046199 -4.282618314 -4.183190428 -4.083762543
```

```
## [11] -3.984334657 -3.884906772 -3.785478886 -3.686051001 -3.586623115
```

```
## [16] -3.487195230 -3.387767344 -3.288339459 -3.188911573 -3.089483688
```

```
## [21] -2.990055802 -2.890627917 -2.791200031 -2.691772146 -2.592344260
```

```
## [26] -2.492916375 -2.393488489 -2.294060604 -2.194632718 -2.095204833
```

```
## [31] -1.995776947 -1.896349062 -1.796921176 -1.697493291 -1.598065405
```

```
## [36] -1.498637520 -1.399209634 -1.299781749 -1.200353863 -1.100925978
```

```
## [41] -1.001498092 -0.902070207 -0.802642321 -0.703214436 -0.603786550
```

```
## [46] -0.504358665 -0.404930779 -0.305502894 -0.206075008 -0.106647123
```

```
## [51] -0.007219237  0.092208648  0.191636534  0.291064419  0.390492305
```

```
## [56]  0.489920190  0.589348076  0.688775961  0.788203847  0.887631732
```

```
## [61]  0.987059618  1.086487503  1.185915389  1.285343274  1.384771159
```

```
## [66]  1.484199045  1.583626930  1.683054816  1.782482701  1.881910587
```

```
## [71]  1.981338472  2.080766358  2.180194243  2.279622129  2.379050014
```

```
## [76]  2.478477900  2.577905785  2.677333671  2.776761556  2.876189442
```

```
## [81]  2.975617327  3.075045213  3.174473098  3.273900984  3.373328869
```

```
## [86]  3.472756755  3.572184640  3.671612526  3.771040411  3.870468297
```

```
## [91]  3.969896182  4.069324068  4.168751953  4.268179839  4.367607724
```

```
## [96]  4.467035610  4.566463495  4.665891381  4.765319266  4.864747152
```

```
## [101]  4.964175037
```

```
Lagrange_2
```

```
## [1]  0.0365005914  0.0357786677  0.0350567439  0.0343348202  0.0336128965
```

```
## [6]  0.0328909727  0.0321690490  0.0314471252  0.0307252015  0.0300032777
```

```
## [11]  0.0292813540  0.0285594303  0.0278375065  0.0271155828  0.0263936590
```

```
## [16]  0.0256717353  0.0249498116  0.0242278878  0.0235059641  0.0227840403
```

```
## [21]  0.0220621166  0.0213401928  0.0206182691  0.0198963454  0.0191744216
```

```
## [26]  0.0184524979  0.0177305741  0.0170086504  0.0162867266  0.0155648029
```

```
## [31]  0.0148428792  0.0141209554  0.0133990317  0.0126771079  0.0119551842
```

```
## [36]  0.0112332604  0.0105113367  0.0097894130  0.0090674892  0.0083455655
```

```
## [41]  0.0076236417  0.0069017180  0.0061797942  0.0054578705  0.0047359468
```

```
## [46]  0.0040140230  0.0032920993  0.0025701755  0.0018482518  0.0011263280
```

```
## [51]  0.0004044043 -0.0003175194 -0.0010394432 -0.0017613669 -0.0024832907
```

```
## [56] -0.0032052144 -0.0039271382 -0.0046490619 -0.0053709856 -0.0060929094
```

```
## [61] -0.0068148331 -0.0075367569 -0.0082586806 -0.0089806044 -0.0097025281
```

```
## [66] -0.0104244518 -0.0111463756 -0.0118682993 -0.0125902231 -0.0133121468
```

```
## [71] -0.0140340705 -0.0147559943 -0.0154779180 -0.0161998418 -0.0169217655
```

```
## [76] -0.0176436893 -0.0183656130 -0.0190875367 -0.0198094605 -0.0205313842
```

```
## [81] -0.0212533080 -0.0219752317 -0.0226971555 -0.0234190792 -0.0241410029
```

```
## [86] -0.0248629267 -0.0255848504 -0.0263067742 -0.0270286979 -0.0277506217
```

```
## [91] -0.0284725454 -0.0291944691 -0.0299163929 -0.0306383166 -0.0313602404
## [96] -0.0320821641 -0.0328040879 -0.0335260116 -0.0342479353 -0.0349698591
## [101] -0.0356917828
```

1b)

```
E <- seq(-5,5,.1) # Choosing arbitrary E values.
```

```
# Efficient Frontier Formulas:
```

```
minvar <- 1/C
```

```
minE <- A/C
```

```
# Efficient Frontier Values:
```

```
minvar
```

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

```
## [1,] 0.000351987
```

```
minE
```

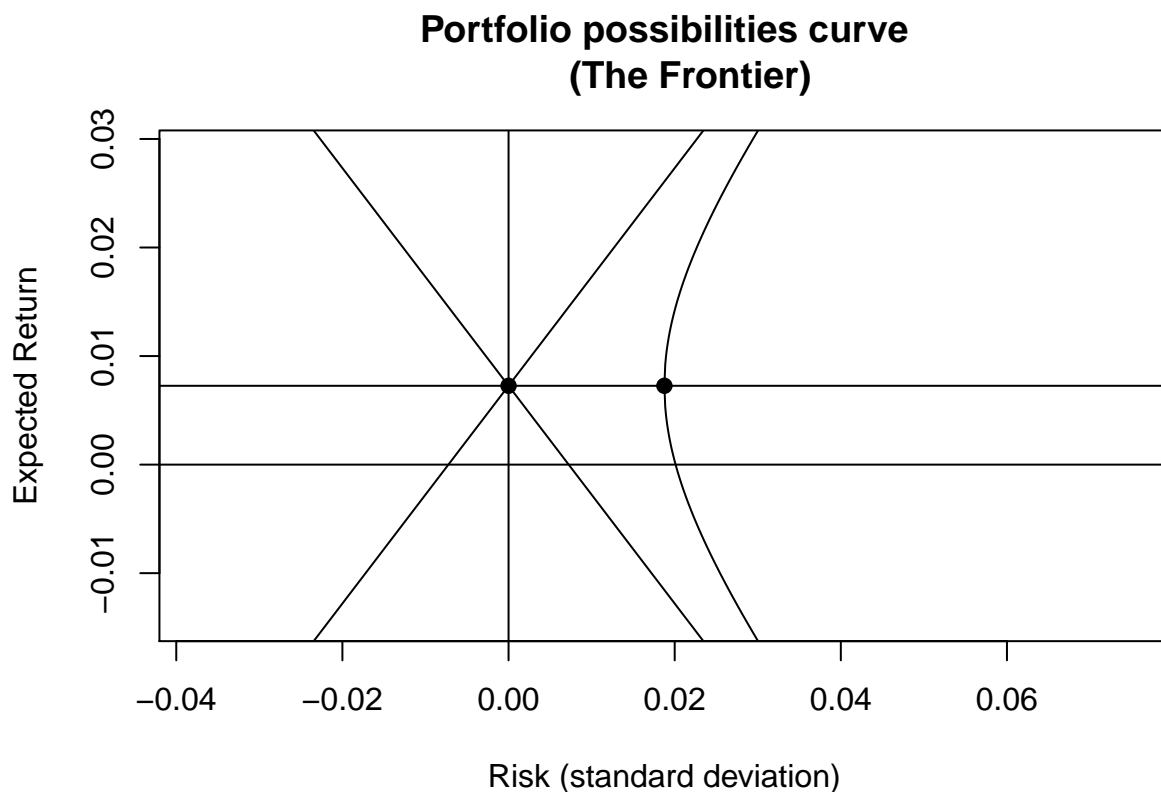
```
##           [,1]
```

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

1c)

```
plot(0, A/C, main = "Portfolio possibilities curve
(The Frontier)", 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) # Plot center of the hyperbola
abline(v = 0) # Plot transverse and conjugate axes. Also this is the y-axis.
abline(h = A/C)
abline(h = 0) # Plot the x-axis
points(sqrt(1/C), A/C, pch=19) # Plot the minimum risk portfolio
V <- seq(-1, 1, 0.001) # Find the asymptotes
A1 <- A/C + V * sqrt(D/C)
A2 <- A/C - V * sqrt(D/C)
points(V, A1, type = "l")
points(V, A2, type = "l")

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")
```



Exercise 3:

```
R_bar <- matrix(c(0.005174, 0.010617, 0.016947))
var_cov <- diag(c(0.010025, 0.002123, 0.005775), 3, 3)
ones <- rep(1, 3)

R_bar_min <- (t(ones) %*% solve(var_cov) %*% R_bar)/(t(ones) %*% solve(var_cov) %*% ones)
sigma_min <- (1)/((t(ones) %*% solve(var_cov) %*% ones)^1/2)

X_numer <- solve(var_cov) %*% ones
X_denom <- t(ones) %*% solve(var_cov) %*% ones
# X_denom = 743.9424
X_vec <- X_numer/743.9424
rownames(X_vec) <- c("C", "XOM", "AAPL")

# Composition of the Minimum Risk Portfolio:
X_vec

##           [,1]
## C      0.1340838
## XOM    0.6331560
## AAPL   0.2327602

R_bar_min

##           [,1]
```

```
## [1,] 0.01136055
```

```
sigma_min
```

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

```
## [1,] 0.00268838
```