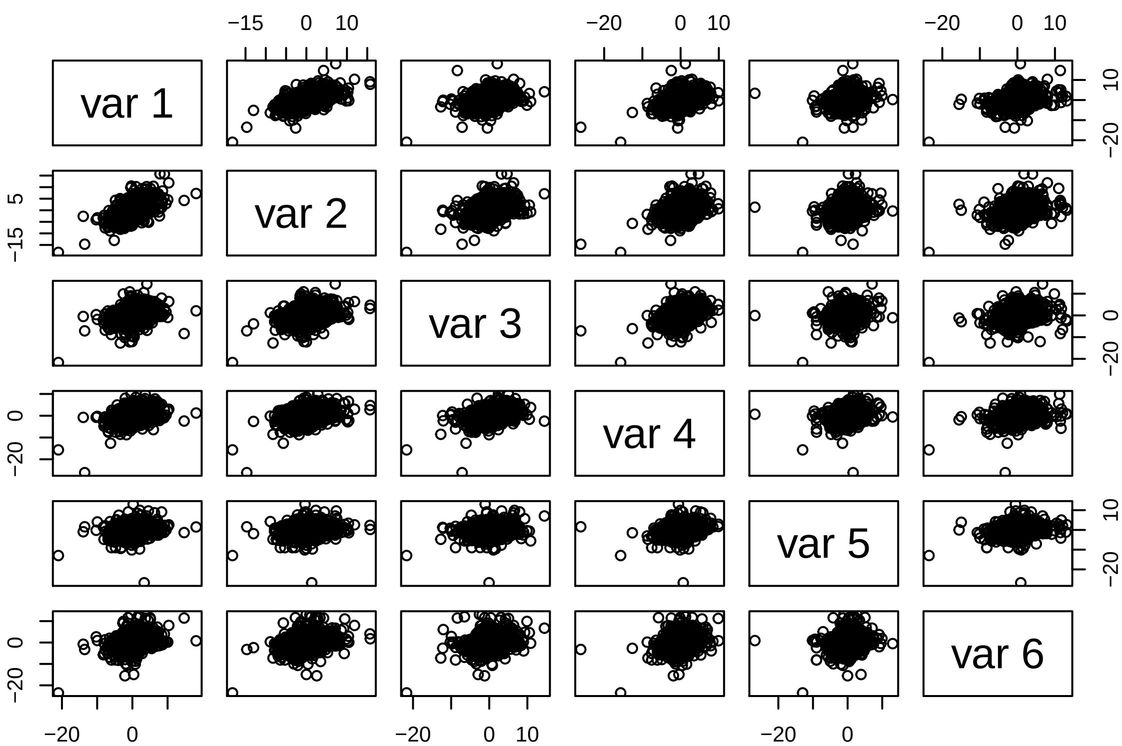
STAT509-001-HW6-Xinye Xu

Q1

(a) Problem 1 Write an R program to find the efficient frontier, the tangency portfolio, and the minimum variance portfolio, and plot on "reward-risk space" the location of each of the six stocks, the efficient frontier, the tangency portfolio, and the line of efficient portfolios. Use the constraints that $0.1 \leq wj \leq 0.5$. for each stock. The first constraint limits short sales but does not rule them out completely. The second constraint prohibits more than 50 % of the investment in any single stock. Assume that the annual risk-free rate is 3 % and convert this to a daily rate by dividing by 365, since interest is earned on trading as well as nontrading days.

```
dat = read.csv("./Data/Stock_Bond.csv", header = T)
prices = cbind(dat$GM_AC, dat$F_AC, dat$CAT_AC, dat$UTX_AC, dat$MRK_AC, dat$IBM_AC)
n = dim(prices)[1]
returns =  100 * (prices[2:n, ] / prices[1:(n-1), ] - 1)
pairs(returns)
```




```
mean_vect = colMeans(returns)
cov_mat = cov(returns)
sd_vect = sqrt(diag(cov_mat))
```

efficient frontier, the tangency portfolio, and the minimum variance portfolio $0.1 \leq wj \leq 0.5$; efficient frontier is the curve above the "+". the tangency portfolio: -0.09126 -0.00294 0.33541 0.38378 0.31950 0.05551 the minimum variance portfolio: 0.0829 0.0577 0.1287 0.2353 0.2960 0.1994

```
library(Ecdat)
```

```
## Loading required package: Ecfun
##
```

```
## Attaching package: 'Ecfun'

## The following object is masked from 'package:base':
##
##     sign

##
## Attaching package: 'Ecdat'

## The following object is masked from 'package:datasets':
##
##     Orange
```

```
library(quadprog)
datalength=400
muP = seq(.04,.08,length=datalength) # set of 300 possible target values
# for the expect portfolio return
sdP = muP # set up storage for std devs of portfolio returns
weights = matrix(0,nrow=datalength,ncol=6) # storage for portfolio weights
M = length(mean_vect)
A.Equality <- matrix(rep(1,M), ncol=1)
Amat <- cbind(A.Equality, mean_vect, diag(M), -diag(M))
B1 = -0.5 # -wi >= -B1
B2 = -0.1 # wi >= B1

# find the optimal portfolios for each target expected return
for (i in 1:length(muP)) {
bvec <- c(1, muP[i], rep(B2, M), rep(B1, M)) # w >= -0.1; -2 >= -0.5
result = solve.QP(Dmat=2*cov_mat,dvec=rep(0,M),Amat=Amat,bvec=bvec,meq=2)
sdP[i] = sqrt(result$value)
weights[i,] = result$solution
}
# first meq: # of equality constraints; dvec -- vector appearing in the quadratic function to be minimi

# the efficient frontier and inefficient portfolios
plot(sdP,muP,type="l",xlim=c(0,2.4),ylim=c(0,.10),lty=3, lwd = 2) # plot

# below the min var portfolio)
mufree = 3/365 # input value of risk-free interest rate
points(0,mufree,cex=4,pch="*") # show risk-free asset
sharpe =(muP-mufree)/sdP # compute Sharpes ratios
ind = (sharpe == max(sharpe)) # Find maximum Sharpe ratio
options(digits=3)
w_tang <- weights[ind,] # print the weights of the tangency portfolio
w_tang
```

```
## [1] -0.09126 -0.00294  0.33541  0.38378  0.31950  0.05551
```

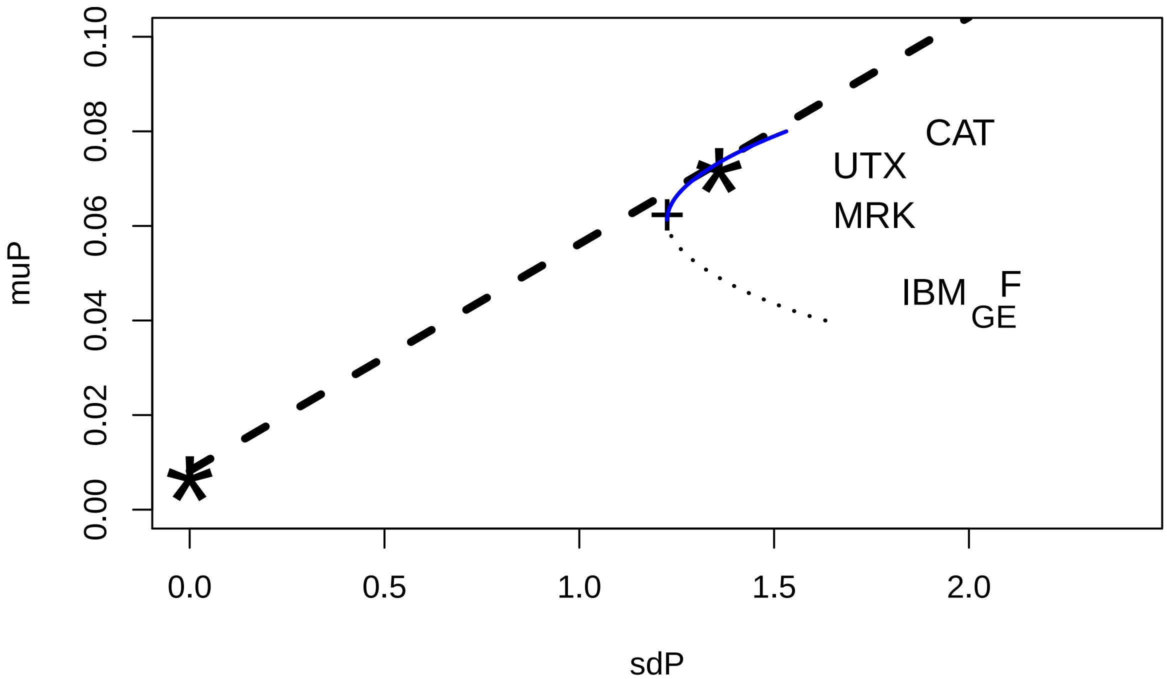
```
lines(c(0,2),mufree+c(0,2)*(muP[ind]-mufree)/sdP[ind],lwd=4,lty=2)

# show line of optimal portfolios
points(sdP[ind],muP[ind],cex=4,pch="*") # show tangency portfolio
ind2 = (sdP == min(sdP)) # find the minimum variance portfolio
points(sdP[ind2],muP[ind2],cex=2,pch="+") # show min var portfolio
weights[ind2,] # print the weights of minimum variance portfolio
```

```
## [1] 0.0829 0.0577 0.1287 0.2353 0.2960 0.1994
```

```
ind3 = (muP > muP[ind2])
lines(sdP[ind3],muP[ind3],type="l",xlim=c(0,.25), ylim=c(0,.3),lwd=2, col='blue') # plot the efficient 

text(sd_vect[1], mean_vect[1], 'GE', cex=1)
text(sd_vect[2], mean_vect[2], 'F', cex=1.15)
text(sd_vect[3], mean_vect[3], 'CAT', cex=1.15)
text(sd_vect[4], mean_vect[4], 'UTX', cex=1.15)
text(sd_vect[5], mean_vect[5], 'MRK', cex=1.15)
text(sd_vect[6], mean_vect[6], 'IBM', cex=1.15)
```



(b) Problem 2 If an investor wants an efficient portfolio with an expected daily return of 0.07%, how should the investor allocate his or her capital to the six stocks and to the risk-free asset? Assume that the investor wishes to use the tangency portfolio computed with the constraints −0.1 wj 0.5, not the unconstrained tangency portfolio.

Percentag for mufree is mufree_pre, exp_tang represents for expected return of tangent portfolio. And then $ mufree_pre * mufree + (1-mufree_pre)* exp_tang = 0.07$ Solve the equation, we have mufree_pre = 5.19%, percentage of tangent portfolio is 94.81%. Then it means: -0.08652 -0.00279 0.31800 0.36386 0.30291 0.05263 for 6 assets separately.

```
mufree = 3/365 # input value of risk-free interest rate
exp_tang = t(w_tang) %*% mean_vect
mufree_pre <-  (0.07 - exp_tang) / (mufree - exp_tang)
print(c(mufree_pre, 1 - mufree_pre))
```

```
## [1] 0.0519 0.9481
```

```
as.vector(1 - mufree_pre) * w_tang
```

```
## [1] -0.08652 -0.00279  0.31800  0.36386  0.30291  0.05263
```

Q2

(a) For problem 1, state the equations that need to be satisfied for each of the assets in order to satisfy the Security Market Line relative to the Tangent portfolio. We knwo tangeent portfolio has μ_M = exp_tang, $\mu_f = mufree$, and tangent portfolio has $\sigma_M = sdP[ind]$, $\beta = Cov(\mu_R, \mu_M)/\sigma_M^2 = Cov(\mu_R, \mu_M)/1.35856^2$. Therefore, the SML of portfolio: $\mu_R - \mu_f = \beta * (\mu_{tan} - \mu_f)$ which can be expressed as $\mu_R - 0.00822 = \beta * (0.07338 - 0.00822)$

```
sd_tang = sdP[ind]
c(exp_tang, sd_tang, mufree = 3/365)
```

```
##                  mufree
## 0.07338 1.35856 0.00822
```

(b) Verify that each of the assets does actually satisfy the Security Market Line relative to the Tangent porfolio. It equals to verify whether the betas are the same under two methods. $\beta = Cov(\mu_R, \mu_{tan})/\sigma_{tan}^2$ and $\mu_R = (1-w) * \mu_f + w * \mu_{tan}$, $\beta = w = \frac{\mu_R - \mu_{tan}}{\mu_{tan} - \mu_f}$ From the result below, they are almost the same, so each of the assets does satisfy the SML.

```
returns_tang <- returns %*% weights[ind,]
# method 1
beta_1 <- cov(returns_tang,returns) / (sd_tang^2)
# method 2
beta_2 <- rep(0, 6)
for (i in 1:6){
  beta_2[i] <- (mean_vect[i] - mufree) / (exp_tang - mufree)
}
beta_1
```

```
##       [,1]  [,2] [,3]  [,4]  [,5]  [,6]
## [1,] 0.503 0.609  1.1 0.992 0.832 0.583
```

```
beta_2
```

```
## [1] 0.501 0.608 1.100 0.992 0.831 0.582
```

Q3.

Problem 6 on page 513 in Ruppert/Matteson. Suppose that the riskless rate of return is 4% and the expected market return is 12 %. The standard deviation of the market return is 11 %. Suppose as well that the covariance of the return on Stock A with the market return is$165\%^2 = 165/10000$

(a) What is the beta of Stock A? $\beta = Cov(\mu_R, \mu_M)/\sigma_M^2 = 165/11^2 = 1.36$.

(b) What is the expected return on Stock A? Based on the SML, $\mu_R = \beta * (\mu_{tan} - \mu_f) + \mu_f = 1.36 * (12 - 4) + 4 = 14.9$. So it's 14.9%.

(c) If the variance of the return on Stock A is $250\%^2$, what percentage of this variance is due to market risk? the $R^2 = corr^2 = \frac{(Cov(\mu_R, \mu_M))^2}{Var_R * Var_M} = 165^2/(250 * 11^2) = 90\%$ It suggests 90% of the variance can be explained by the market risk.

```
165 / 11^2
```

```
## [1] 1.36
```

```
1.36*(12-4) + 4
```

```
## [1] 14.9
```

```r
165^2 /(250*11^2)
```

```
## [1] 0.9
```

Problem 11 on page 514 in Ruppert/Matteson. Suppose there are three risky assets with the following betas and 2 when j regressed on the market portfolio. Assume 1, 2, and 3 are uncorrelated. Suppose also that the variance of RM − f is 0.02.

(a) What is the beta of an equally weighted portfolio of these three assets? $(\beta_1 + \beta_2 + \beta_3)/3 = (0.7 + 0.8 + 0.6)/3 = 0.7$

(b) What is the variance of the excess return on the equally weighted portfolio? $Var_{new} = \beta_{new}^2 * \sigma_{R-f}^2 + (1/3)^2(\sigma_{\epsilon_1}^2 + \sigma_{\epsilon_2}^2 + \sigma_{\epsilon_3}^2) = 0.015$

(c) What proportion of the total risk of asset 1 is due to market risk? $R^2 = corr^2 = \beta_1^2 * \sigma_{R-f}^2 / Var_1 = 49.5\%$

```r
beta_n = (0.7+0.8+0.6)/3
beta_n
```

```
## [1] 0.7
```

```r
var_new = beta_n^2 * 0.02 + (1/3)^2*(0.01+0.025+0.012)
var_new
```

```
## [1] 0.015
```

```r
var_1 = 0.7^2 * 0.02 + 0.01
var_1
```

```
## [1] 0.0198
```

```r
0.7^2 * 0.02 / var_1
```

```
## [1] 0.495
```