HW5

Efficient Portfolio Theory

3/5/2021

pacman::p\_load(pacman, tseries, tidyverse, knitr)  
knitr::opts\_chunk$set(message = FALSE)

#Question 5.1

mu1 = c(0.04, 0.03, 0.05) # mean vector  
tibble(mu1)

## # A tibble: 3 x 1  
## mu1  
## <dbl>  
## 1 0.04  
## 2 0.03  
## 3 0.05

Sigma1 = matrix(c(.05,.05,.025,.05,.10,.08,.025,.08,.075),3,3)  
Sigma1

## [,1] [,2] [,3]  
## [1,] 0.050 0.05 0.025  
## [2,] 0.050 0.10 0.080  
## [3,] 0.025 0.08 0.075

w1 = c(1/3, 1/3, 1/3)  
tibble(w1)

## # A tibble: 3 x 1  
## w1  
## <dbl>  
## 1 0.333  
## 2 0.333  
## 3 0.333

w2 = c(0.4, 0.4, 0.2)  
tibble(w2)

## # A tibble: 3 x 1  
## w2  
## <dbl>  
## 1 0.4  
## 2 0.4  
## 3 0.2

#a  
wt\_mean1 = w1\*mu1  
tibble(wt\_mean1)

## # A tibble: 3 x 1  
## wt\_mean1  
## <dbl>  
## 1 0.0133  
## 2 0.0100  
## 3 0.0167

sum(wt\_mean1)

## [1] 0.04

variance1 = t(w1) %\*% Sigma1 %\*% w1 # matrix multiplication  
variance1

## [,1]  
## [1,] 0.05944444

std\_dev1 = sqrt(variance1)   
std\_dev1

## [,1]  
## [1,] 0.2438123

tibble(variance1 = variance1,  
 std\_dev1 = std\_dev1) %>%  
 round(4)

## # A tibble: 1 x 2  
## variance1[,1] std\_dev1[,1]  
## <dbl> <dbl>  
## 1 0.0594 0.244

#b  
  
wt\_mean2 = w2\*mu1  
tibble(wt\_mean2)

## # A tibble: 3 x 1  
## wt\_mean2  
## <dbl>  
## 1 0.016  
## 2 0.012  
## 3 0.01

sum(wt\_mean2)

## [1] 0.038

variance2 = t(w2) %\*% Sigma1 %\*% w2 # matrix multiplication  
variance2

## [,1]  
## [1,] 0.0598

std\_dev2 = sqrt(variance2)   
std\_dev2

## [,1]  
## [1,] 0.2445404

tibble(variance2 = variance2,  
 std\_dev2 = std\_dev2) %>%  
 round(4)

## # A tibble: 1 x 2  
## variance2[,1] std\_dev2[,1]  
## <dbl> <dbl>  
## 1 0.0598 0.244

#c  
  
covariance = w1%\*%Sigma1%\*%w2  
covariance

## [,1]  
## [1,] 0.05933333

correlation = covariance/(std\_dev1\*std\_dev2)  
correlation

## [,1]  
## [1,] 0.9951591

round(correlation)

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

#d  
pf1 = sum(wt\_mean1)/variance1  
pf1

## [,1]  
## [1,] 0.6728972

pf2 = sum(wt\_mean2)/variance2  
pf2

## [,1]  
## [1,] 0.6354515

# Based on these results portfolio 1 is preferable to the portfolio 2 because when we divide weighted mean from variance then the value of portfolio 1 is greater therefore it is better than portfolio 2

#Question 5.2

mu2 = c (0.02, 0.10, 0.05, 0.06) #mean vector  
tibble(mu2)

## # A tibble: 4 x 1  
## mu2  
## <dbl>  
## 1 0.02  
## 2 0.1   
## 3 0.05  
## 4 0.06

Sigma2 = matrix(c(.02,.01,.01,0,.01,.05,.02,0,.01,.02,.03,0,0,0,0,0.04),4,4)  
Sigma2

## [,1] [,2] [,3] [,4]  
## [1,] 0.02 0.01 0.01 0.00  
## [2,] 0.01 0.05 0.02 0.00  
## [3,] 0.01 0.02 0.03 0.00  
## [4,] 0.00 0.00 0.00 0.04

A = cbind(c(1,1,1,1),mu2)  
t(A)

## [,1] [,2] [,3] [,4]  
## 1.00 1.0 1.00 1.00  
## mu2 0.02 0.1 0.05 0.06

#a  
library(quadprog)   
  
mrf11 = solve.QP(Dmat = 2\*Sigma2, # 2 x covariance matrix  
 dvec = mu2, # avg return  
 Amat = A, # constraint coefficients  
 bvec = c(1, 0.05), # values of constraints  
 meq = 2) # treat 2 constraints as qualities  
  
wts1 = mrf11$solution  
wts1

## [1] 0.3542393 0.1498258 0.1823461 0.3135889

mean3 = sum(mrf11$solution \* mu2)  
mean3

## [1] 0.05

std\_dev3 = (mrf11$solution %\*% Sigma2 %\*% mrf11$solution) ^ 0.5  
std\_dev3

## [,1]  
## [1,] 0.1095869

tibble(mean3, std\_dev3) %>%  
 round(4)

## # A tibble: 1 x 2  
## mean3 std\_dev3[,1]  
## <dbl> <dbl>  
## 1 0.05 0.110

#b  
mrf12 = solve.QP(Dmat = 2\*Sigma2, # 2 x covariance matrix  
 dvec = mu2, # avg return  
 Amat = A, # constraint coefficients  
 bvec = c(1, 0.06), # values of constraints  
 meq = 2) # treat 2 constraints as qualities  
  
wts2 = mrf12$solution  
wts2

## [1] 0.2229965 0.2648084 0.1672474 0.3449477

mean4 = sum(mrf12$solution \* mu2)  
mean4

## [1] 0.06

std\_dev4 = (mrf12$solution %\*% Sigma2 %\*% mrf12$solution) ^ 0.5  
std\_dev4

## [,1]  
## [1,] 0.1174645

tibble(mean4, std\_dev4) %>%  
 round(4)

## # A tibble: 1 x 2  
## mean4 std\_dev4[,1]  
## <dbl> <dbl>  
## 1 0.06 0.118

mrf13 = solve.QP(Dmat = 2\*Sigma2, # 2 x covariance matrix  
 dvec = mu2, # avg return  
 Amat = A, # constraint coefficients  
 bvec = c(1, 0.07), # values of constraints  
 meq = 2) # treat 2 constraints as qualities  
  
wts3 = mrf13$solution  
wts3

## [1] 0.09175377 0.37979094 0.15214866 0.37630662

mean5 = sum(mrf13$solution \* mu2)  
mean5

## [1] 0.07

std\_dev5 = (mrf13$solution %\*% Sigma2 %\*% mrf13$solution) ^ 0.5  
std\_dev5

## [,1]  
## [1,] 0.1304864

tibble(mean5, std\_dev5) %>%  
 round(4)

## # A tibble: 1 x 2  
## mean5 std\_dev5[,1]  
## <dbl> <dbl>  
## 1 0.07 0.130

#c  
#Based on these results it is not possible to conclude with certainty whether these portfolios are on the efficient frontier or not because we don't know the required return of these portfolios. In order to conclude its certinity we need to know required returns of portfolio and if required return is higher than expected return then portfolio is above the efficient frontier. If required return is lower than expected return then it is below efficiemt frontier. If required and expected returns are same then portfios are on the efficient frontier.

#Question 5.5

mu3 = c (0.25, 0.20, 0.30, 0.275, 0.15) # mean vector  
  
Sigma3 = matrix(c(1,.4,.6,.5,.3,.4,0.7,.5,.4,.25,0.6,0.5,1.3,0.6,0.35,0.5,0.4,0.6,1,0.3,0.3,0.25,0.35,0.3,0.5),5,5)  
Sigma3 # covariance matrix

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1.0 0.40 0.60 0.5 0.30  
## [2,] 0.4 0.70 0.50 0.4 0.25  
## [3,] 0.6 0.50 1.30 0.6 0.35  
## [4,] 0.5 0.40 0.60 1.0 0.30  
## [5,] 0.3 0.25 0.35 0.3 0.50

# find unnormalized weights  
w01 = solve(Sigma3, c(1,1,1,1,1))  
w01

## [1] 0.2275633 0.7338277 -0.1099463 0.2275633 1.4369726

# normalize the weights  
w\_mv1 = w01/sum(w01)  
w\_mv1

## [1] 0.09044715 0.29166667 -0.04369919 0.09044715 0.57113821

#Question 5.11

mu = c(0.04, 0.03, 0.05) #mean vector  
  
Sig = matrix(c(.05,.05,.025,.05,.10,.08,.025,.08,.075),3,3)  
Sig #covarience matrix

## [,1] [,2] [,3]  
## [1,] 0.050 0.05 0.025  
## [2,] 0.050 0.10 0.080  
## [3,] 0.025 0.08 0.075

# find unnormalized weights  
w0 = solve(Sig, c(1,1,1))  
  
# normalize the weights  
w\_mv = w0/sum(w0)  
  
m = sum(w\_mv \* mu)  
m

## [1] 0.0762069

vbar = solve(Sig, mu - m\*c(1,1,1))  
vbar %>% round(5)

## [1] 0.34483 -1.79310 1.44828

lambda1 = 1  
wt\_lam = w\_mv + vbar/lambda1  
wt\_lam %>% round(5)

## [1] 1.51724 -3.68966 3.17241

mu4 = sum((w\_mv + vbar/lambda1)\*mu)  
tibble(mean\_risk\_averse\_1 = mu4) %>%   
 round(5) %>% tibble()

## # A tibble: 1 x 1  
## mean\_risk\_averse\_1  
## <dbl>  
## 1 0.109

var1 = (w\_mv+vbar/lambda1) %\*% Sig %\*% (w\_mv+vbar/lambda1)  
sd1 = var1^0.5  
tibble(std\_dev\_return\_1 = sd1) %>%  
 round(5) %>% tibble()

## # A tibble: 1 x 1  
## std\_dev\_return\_1[,1]  
## <dbl>  
## 1 0.198

lambda2 = 5  
wt\_lam = w\_mv + vbar/lambda2  
  
mu5 = sum((w\_mv + vbar/lambda2)\*mu)  
tibble(mean\_risk\_averse\_2 = mu5) %>%   
 round(5) %>% tibble()

## # A tibble: 1 x 1  
## mean\_risk\_averse\_2  
## <dbl>  
## 1 0.0827

var2 = (w\_mv+vbar/lambda2) %\*% Sig %\*% (w\_mv+vbar/lambda2)  
sd2 = var2^0.5  
tibble(std\_dev\_return\_2 = sd2) %>%  
 round(5) %>% tibble()

## # A tibble: 1 x 1  
## std\_dev\_return\_2[,1]  
## <dbl>  
## 1 0.0905

#Question 5.18

mu3 = c (0.25, 0.20, 0.30, 0.275, 0.15)  
  
Sigma3 = matrix(c(1,.4,.6,.5,.3,.4,0.7,.5,.4,.25,0.6,0.5,1.3,0.6,0.35,0.5,0.4,0.6,1,0.3,0.3,0.25,0.35,0.3,0.5),5,5)  
Sigma3

## [,1] [,2] [,3] [,4] [,5]  
## [1,] 1.0 0.40 0.60 0.5 0.30  
## [2,] 0.4 0.70 0.50 0.4 0.25  
## [3,] 0.6 0.50 1.30 0.6 0.35  
## [4,] 0.5 0.40 0.60 1.0 0.30  
## [5,] 0.3 0.25 0.35 0.3 0.50

w\_T = solve(Sigma3, mu3-0.01)/sum(solve(Sigma3, mu3-0.01))  
w\_T #weight vector of tangency portfolio

## [1] 0.1855630 0.1687238 0.1967891 0.3035881 0.1453360

tibble(weight\_vector\_of\_tangency\_portfolio = w\_T)

## # A tibble: 5 x 1  
## weight\_vector\_of\_tangency\_portfolio  
## <dbl>  
## 1 0.186  
## 2 0.169  
## 3 0.197  
## 4 0.304  
## 5 0.145

sum(w\_T\*mu3)/(w\_T%\*%Sigma3%\*%w\_T)^.5 #This tangency portfolio has the Sharpe ratio

## [,1]  
## [1,] 0.3286023