

```

library(MASS)
library(quadprog)
data = read.csv("Stock_Bond.csv", header = T)
price = cbind(data$GM_AC, data$MRK_AC, data$C_AC)
n = dim(price)[1]
return = log(price[2:n,]/price[1:(n-1),])
mu = colMeans(return)
sigma = cov(return)

muP = seq(.0001,.001,length=200) # target portfolio return
sdP = muP # sd of portfolio return
weight = matrix(0,nrow=200,ncol=3) # storage for portfolio weights
for (i in 1:length(muP)) # find the optimal portfolios
{
  result = solve.QP(Dmat=2*sigma,dvec=rep(0,3),
                    Amat=cbind(rep(1,3),mu),bvec=c(1,muP[i]),meq=2)
  sdP[i] = sqrt(result$value)
  weight[i,] = result$solution
}
par(mfrow = c(1,1))
plot(sdP,muP,type="l",xlim=c(.01,.03),ylim=c(0,.001),lwd=3,col="red")
# efficient frontier

text(sqrt(sigma[1,1]),mu[1],"GM",cex=1.1)
text(sqrt(sigma[2,2]),mu[2],"Merck",cex=1.1)
text(sqrt(sigma[3,3]),mu[3],"Citi",cex=1.1)

set.seed(201801)
for (k in 1:5){
  rboot = mvrnorm(2*n,mu=mu,Sigma=sigma)
  muboot = colMeans(rboot)
  sboot = cov(rboot)
  sdPboot = muP
  wboot = matrix(0,nrow=200,ncol=3) # storage for portfolio weights
  for (i in 1:length(muP)) # find the optimal portfolios
  {
    result = solve.QP(Dmat=2*sboot,dvec=rep(0,3),
                      Amat=cbind(rep(1,3),muboot),bvec=c(1,muP[i]),meq=2)
    sdPboot[i] = sqrt(result$value)
  }
}

```

```

    wboot[i,] = result$solution
  }
  lines(sdPboot,muP,type="l",xlim=c(.01,.03),ylim=c(0,.001),lwd=1) #
efficient frontier
}

#####
#####
muP = seq(min(mu),max(mu),length=200) # target portfolio return
for (i in 1:length(muP)) # find the optimal portfolios
{
  result = solve.QP(Dmat=2*sigma,dvec=rep(0,3),
                    Amat=cbind(rep(1,3),mu,diag(1,3)),
                    bvec=c(1,muP[i],rep(0,3)),meq=2)
  sdP[i] = sqrt(result$value)
  weight[i,] = result$solution
}
par(mfrow = c(1,1))
plot(sdP,muP,type="l",xlim=c(.
01,.03),ylim=c(0,.001),lwd=3,col="blue") # efficient frontier
#lines(sdP,muP,type="l",xlim=c(0,.04),ylim=c(0,.001),lwd=3,col="blue"
) # efficient frontier

text(sqrt(sigma[1,1]),mu[1],"GM",cex=1.1)
text(sqrt(sigma[2,2]),mu[2],"Merck",cex=1.1)
text(sqrt(sigma[3,3]),mu[3],"Citi",cex=1.1)

#####
#####
pdf('CAPM1.pdf')
muP = seq(.0001,.001,length=200) # target portfolio return
sdP = muP # sd of portfolio return
weight = matrix(0,nrow=200,ncol=3) # storage for portfolio weights
for (i in 1:length(muP)) # find the optimal portfolios
{
  result = solve.QP(Dmat=2*sigma,dvec=rep(0,3),
                    Amat=cbind(rep(1,3),mu),bvec=c(1,muP[i]),meq=2)
  sdP[i] = sqrt(result$value)

```

```

    weight[i,] = result$solution
}
par(mfrow = c(1,1))
plot(sdP,muP,type="l",xlim=c(0,.04),ylim=c(0,.001),lty=3,col="red") #
efficient frontier

text(sqrt(sigma[1,1]),mu[1],"GM",cex=1.1)
text(sqrt(sigma[2,2]),mu[2],"Merck",cex=1.1)
text(sqrt(sigma[3,3]),mu[3],"Citi",cex=1.1)

ind1 = (sdP == min(sdP)) # find the minimum variance portfolio
#points(sdP[ind1],muP[ind1],cex=1.1,pch="x") # show min var portfolio
ind2 = (muP > muP[ind1])
lines(sdP[ind2],muP[ind2],type="l",xlim=c(0,.05),
      ylim=c(0,.0015),lwd=3,col="red") # plot the efficient
frontier

rf = .06/253 # riskfree rate
points(0,rf,cex=1.1,pch="x") # show riskfree asset
sharpe =(muP-rf)/sdP # Sharpe's ratios
ind3 = (sharpe == max(sharpe)) # find maximum Sharpe's ratio

lines(c(0,1),rf+c(0,1)*(sharpe[ind3]-.01),lwd=2,lty=3,col="black")
points(sdP[ind3],rf+sdP[ind3]*(sharpe[ind3]-.01),cex=1.1,pch="x")
lines(c(0,1),rf+c(0,1)*(sharpe[ind3]-.005),lwd=2,lty=3,col="black")
points(sdP[ind3]+.002,rf+(sdP[ind3]+.002)*(sharpe[ind3]-.
005),cex=1.1,pch="x")
lines(c(0,1),rf+c(0,1)*(sharpe[ind3]-.015),lwd=2,lty=3,col="black")
points(sdP[ind3]+.005,rf+(sdP[ind3]+.005)*(sharpe[ind3]-.
015),cex=1.1,pch="x")

lines(c(0,1),rf+c(0,1)*sharpe[ind3],lwd=2,lty=1,col="black") # line
of optimal portfolios
points(sdP[ind3],muP[ind3],cex=1.1,pch="x") # show tangency portfolio

dev.off()

```

```

#####
#####

```

```

library(zoo)
mkreturn = log(data$S.P_AC[2:n]/data$S.P_AC[1:(n-1)])
x1rfrate = data$X1.year.Treasury.Constant.Maturity.Rate
x1rfrate[x1rfrate=='#N/A'] = NA
x1rfrate = as.numeric(na.locf(x1rfrate))
rfreturn = x1rfrate[2:n]/100/253

fitGM1 = lm((return[,1]-rfreturn)~-1+(mkreturn-rfreturn)) #GM
fitMRK1 = lm((return[,2]-rfreturn)~-1+(mkreturn-rfreturn)) #Merck
fitC1 = lm((return[,3]-rfreturn)~-1+(mkreturn-rfreturn)) #Citi

summary(fitGM1)
summary(fitMRK1)
summary(fitC1)

fitGM2 = lm((return[,1]-rfreturn)~(mkreturn-rfreturn)) #GM
fitMRK2 = lm((return[,2]-rfreturn)~(mkreturn-rfreturn)) #Merck
fitC2 = lm((return[,3]-rfreturn)~(mkreturn-rfreturn)) #Citi

summary(fitGM2)
summary(fitMRK2)
summary(fitC2)

pdf('LMFIT.pdf')
par(mfrow = c(1,1))
plot(return[,1]-rfreturn,mkreturn-rfreturn,xlim=c(0,.01),ylim=c(-.
  025,.01),type='n')
abline(fitGM1,col='red',lty=1)
abline(fitMRK1,col='blue',lty=1)
abline(fitC1,col='black',lty=1)
abline(fitGM2,col='red',lty=2)
abline(fitMRK2,col='blue',lty=2)
abline(fitC2,col='black',lty=2)

legend("bottomright",legend=c("GM (without intercept)",
  "Merck (without intercept)","Citi (without intercept)",
  "GM (with intercept)","Merck (with intercept)","Citi (with
  intercept)"),

col=c("red", "blue", "black", "red", "blue", "black"),lty=c(1,1,1,2,2,2))

```

dev.off()