yx-portfolio

YingXiang UNI:yx2647

4/5/2021

#just code

```
return=read.csv("return.csv")
return=return[-1,]
m=600
muP=seq(.0001,.006,length=m)
sdP = rep(0, length(muP))
mu=colMeans(return[,3:17])
sigma=cov(return[,3:17])
weight=matrix(0,nrow=m,ncol=15)
#2016-04-04 10 Year Treasury Rate is 1.72%
#daily risk free rate is 1.72%/253
rf=1.72/100/253
```

```
#sharpe ratio
sd=diag(sigma)
sharpe_ratio=(mu-rf)/sd
sharpe_ratio
```

```
##      AAPL       AXP        BA       CAT      CSCO       CVX       DIS       JNJ
## 3.5565622 1.8014333 1.0791718 2.5922358 1.8900934 0.5436540 1.8441497 2.1387845
##        KO       MMM       NKE        PG       RTX       UNH        VZ
## 0.6621439 0.6827477 2.2266743 2.4399558 0.7502105 2.9175210 0.4362877
```

```
#AAPL has the largest Sharpe ratio, which might let it play a significant role in portfolio.
```

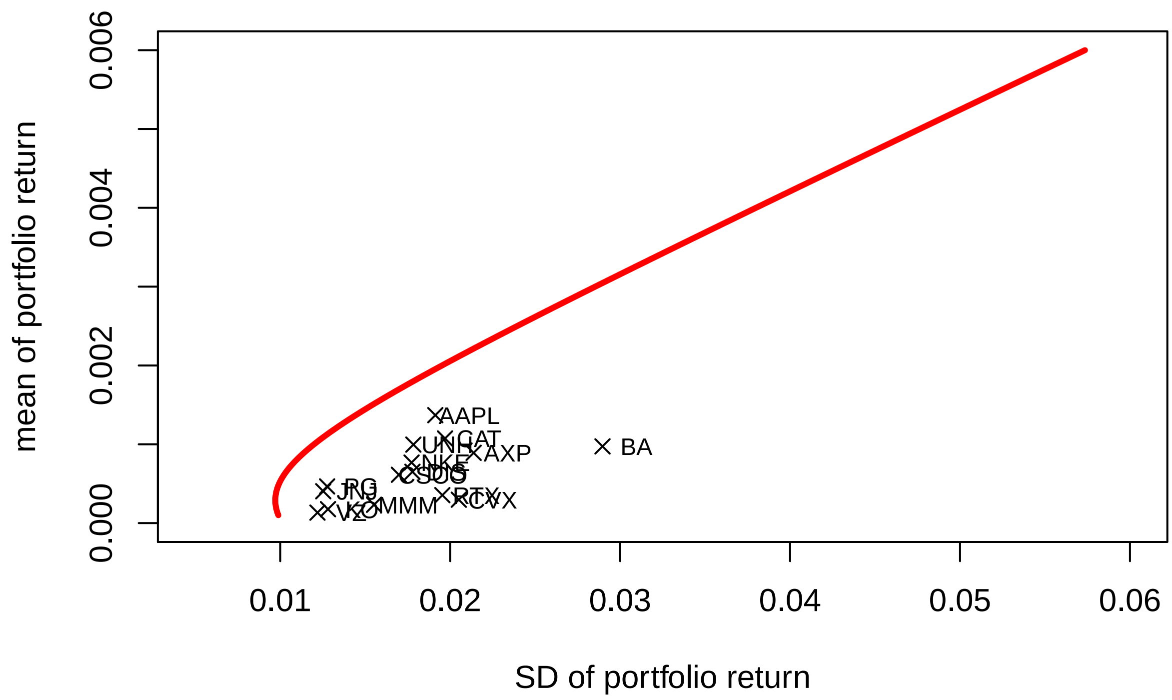
```
#portfolio with short sell
library(quadprog)
for (i in 1:length(muP)){ # find the optimal portfolios
  result = solve.QP(Dmat=2*sigma,dvec=rep(0,15),
  Amat = cbind(rep(1,15),mu),bvec=c(1,muP[i]),meq=2)
  sdP[i] = sqrt(result$value)
  weight[i,] = result$solution
}
```

```
GMP=which.min(sdP)
#MVP-minimum variance portfolio
sd_MVP=sdP[GMP]
mu_MVP=muP[GMP]
weight_MVP=weight[GMP,]
names(weight_MVP)=names(return)[3:17]
weight_MVP
```

```
##         AAPL          AXP           BA          CAT         CSCO          CVX
##  0.053385045 -0.087459726 -0.057309890  0.001993090 -0.037053016  0.001523108
```

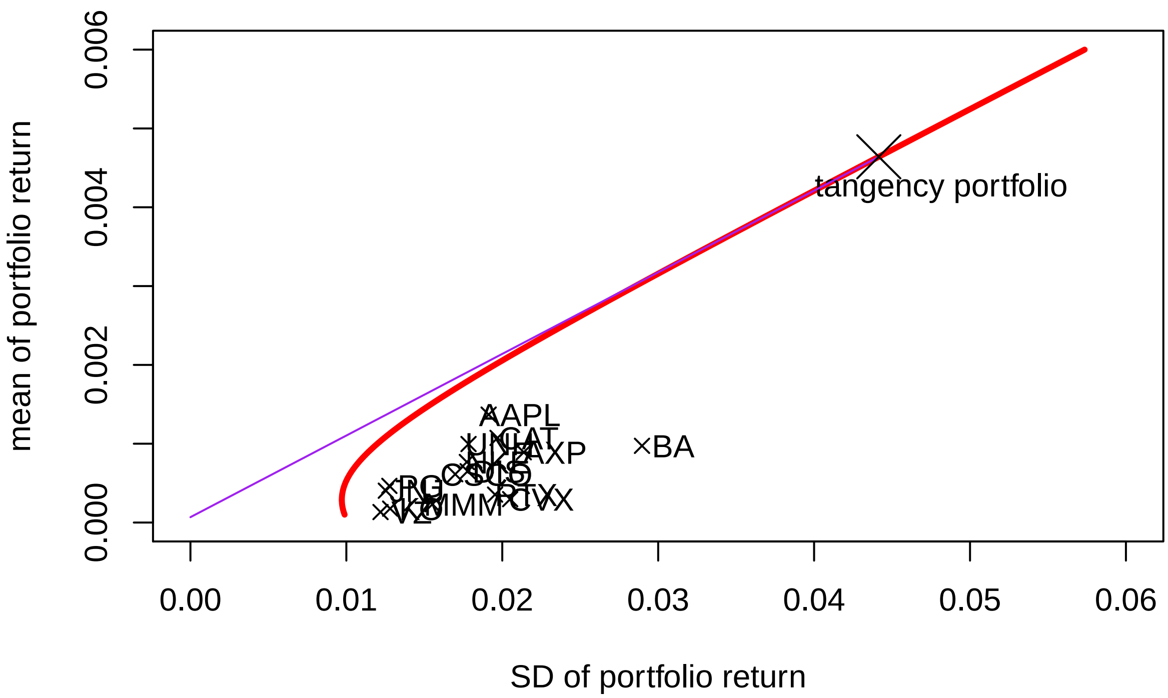
```
##          DIS          JNJ           KO          MMM          NKE           PG
##  0.091110672  0.225669466  0.178675841  0.090923277  0.060236724  0.125730840
##          RTX          UNH           VZ
##  0.030980354  0.007316239  0.314277976
```

```r
#in the case, we allow short selling and MVP daily average return (0.000287) and standard deviation (0.



plot(sdP[GMP:m],muP[GMP:m],type="l",xlim=c(.005,.06),ylim=c(0,.0060),
lwd=3,col="red", xlab = "SD of portfolio return",
ylab = "mean of portfolio return")
points(sdP[1:(GMP-1)],muP[1:(GMP-1)], type="l",
lty = 2,lwd=3, col = "red")
points(sqrt(diag(sigma)), mu, pch = 4)
text(sqrt(diag(sigma))+0.002, mu, names(return)[3:17],cex=0.75)
```




```r
sharpe=(muP[GMP:m]-rf)/sdP[GMP:m]
tangency.portfolio.sd=sdP[GMP:m][which.max(sharpe)]
tangency.portfolio.mu=muP[GMP:m][which.max(sharpe)]
tangency.portfolio.weight=weight[GMP:m,][which.max(sharpe),]
names(tangency.portfolio.weight)=names(return)[3:17]
x=seq(0,tangency.portfolio.sd,length=200)
y=seq(rf,tangency.portfolio.mu,length=200)

#The Efficient Portfolio Frontier is presented below.
#the purple line is the sharp slope of tangency portfolio.
```

```r
#The red line represents the efficient frontier.
plot(sdP[GMP:m],muP[GMP:m],type="l",xlim=c(.00,.06),ylim=c(0,.0060),
lwd=3,col="red", xlab = "SD of portfolio return",
ylab = "mean of portfolio return")
points(sdP[1:(GMP-1)],muP[1:(GMP-1)], type="l",
lty = 2,lwd=3, col = "red")
points(sqrt(diag(sigma)), mu, pch = 4)
text(sqrt(diag(sigma))+0.002, mu, names(return)[3:17])
text(tangency.portfolio.sd+0.004,tangency.portfolio.mu-0.0004,"tangency portfolio")
lines(x,y,col="purple")
points(tangency.portfolio.sd,tangency.portfolio.mu,pch=4,cex=3)
```




```r
#weight of tangency portfolio
tangency.portfolio.weight
```

```
##        AAPL        AXP         BA        CAT       CSCO        CVX        DIS
##  1.2770415  0.7446430  0.2849279  1.5122177 -0.5386046 -0.9703034  0.2539081
##         JNJ         KO        MMM        NKE         PG        RTX        UNH
##  0.2277804 -0.9108869 -1.3700301  0.3553953  0.6645731 -1.0449569  1.0282245
##          VZ
## -0.5139296
```

```r
#return of tangency portfolio
tangency.portfolio.mu
```

```
## [1] 0.004640735
```

```r
#sd of tangency portfolio
tangency.portfolio.sd
```

```
## [1] 0.0441509
```

```r
#sharpe ratio oftangency portfolio
max(sharpe)
```

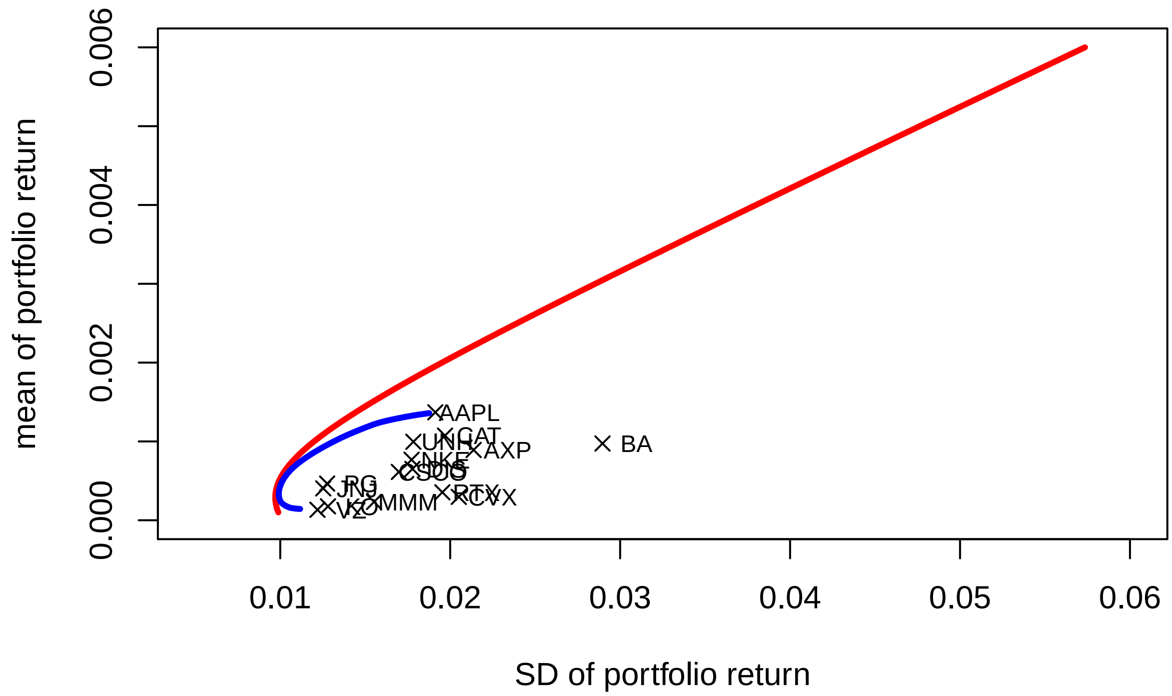
```
## [1] 0.1035709
```

```r
RETURN=c(rf,mu_MVP,tangency.portfolio.mu)
RISK=c(0,sd_MVP,tangency.portfolio.sd)
rf.mvp.tangency=data.frame(rbind(RETURN,RISK))
names(rf.mvp.tangency)=c("daily risk free rate","daily MVP","daily tangency portfolio")
rf.mvp.tangency
```

```
##        daily risk free rate    daily MVP daily tangency portfolio
## RETURN         6.798419e-05 0.0002871452              0.004640735
## RISK           0.000000e+00 0.0097099309              0.044150904
```

```r
#by comparing the weight of MVP and weight of tangency portfolio, we find that both portfolio invest he
```

```r
#without short sell
#muP=seq(.0001,.006,length=m)
muP_noSS = seq(min(mu)+0.00001,max(mu)-0.00001,length=m) # target portfolio return
sdP_noSS = rep(0, length(muP_noSS))
for (i in 1:length(muP_noSS)) { # find the optimal portfolios
result = solve.QP(Dmat=2*sigma,dvec=rep(0,15),Amat=cbind(rep(1,15),mu,diag(1,15)),bvec=c(1,muP_noSS[i],
sdP_noSS[i] = sqrt(result$value)
weight[i,] = result$solution
}

plot(sdP[GMP:m],muP[GMP:m],type="l",xlim=c(.005,.06),ylim=c(0,.0060),
lwd=3,col="red", xlab = "SD of portfolio return",
ylab = "mean of portfolio return")
points(sdP[1:(GMP-1)],muP[1:(GMP-1)], type="l",
lty = 2,lwd=3, col = "red")
points(sqrt(diag(sigma)), mu, pch = 4)
text(sqrt(diag(sigma))+0.002, mu, names(return)[3:17],cex=0.75)

lines(sdP_noSS[GMP:m],muP_noSS[GMP:m],type="l",lwd=3,col="blue")
points(sdP_noSS[1:(GMP-1)],muP_noSS[1:(GMP-1)], type="l",
lty = 2,lwd=3, col = "blue")
```


mean of portfolio return
0.000
0.002
0.004
0.006
SD of portfolio return
0.01
0.02
0.03
0.04
0.05
0.06
AAPL
CAT
UNH
AXP
BA
NKE
CSCO
PG
JNJ
VZ
KO
MMM
RTX
CVX