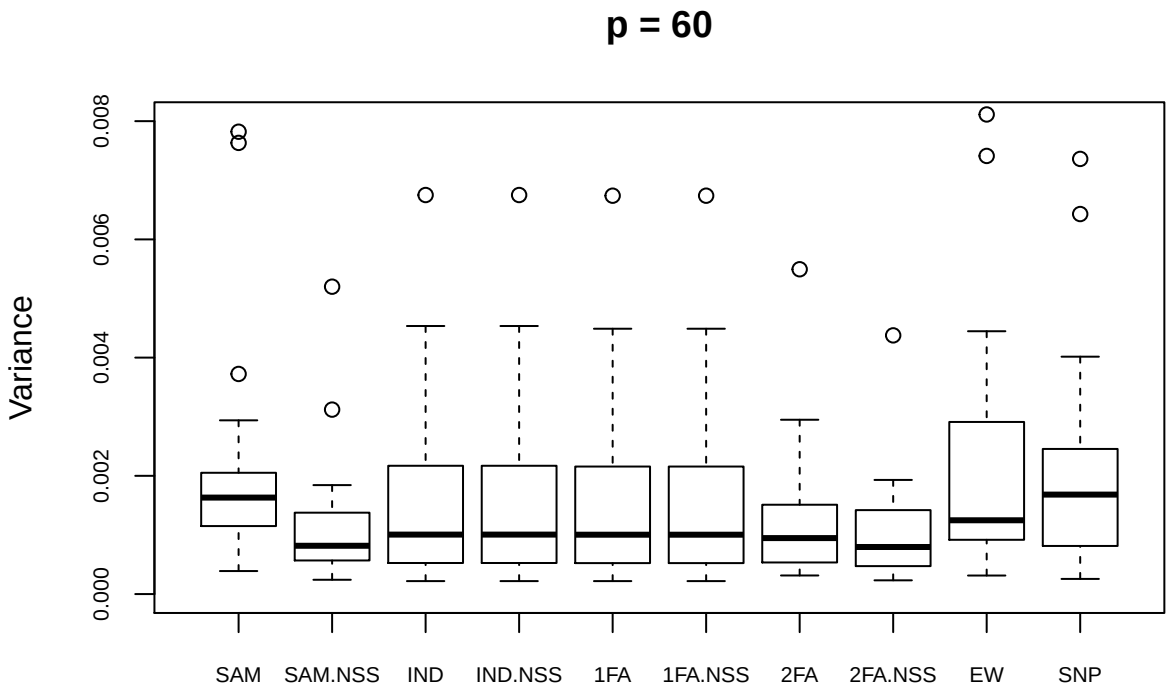
Financial Data Mining
Homework 4
Yen-Hsiu Chang

1.(a) (b)

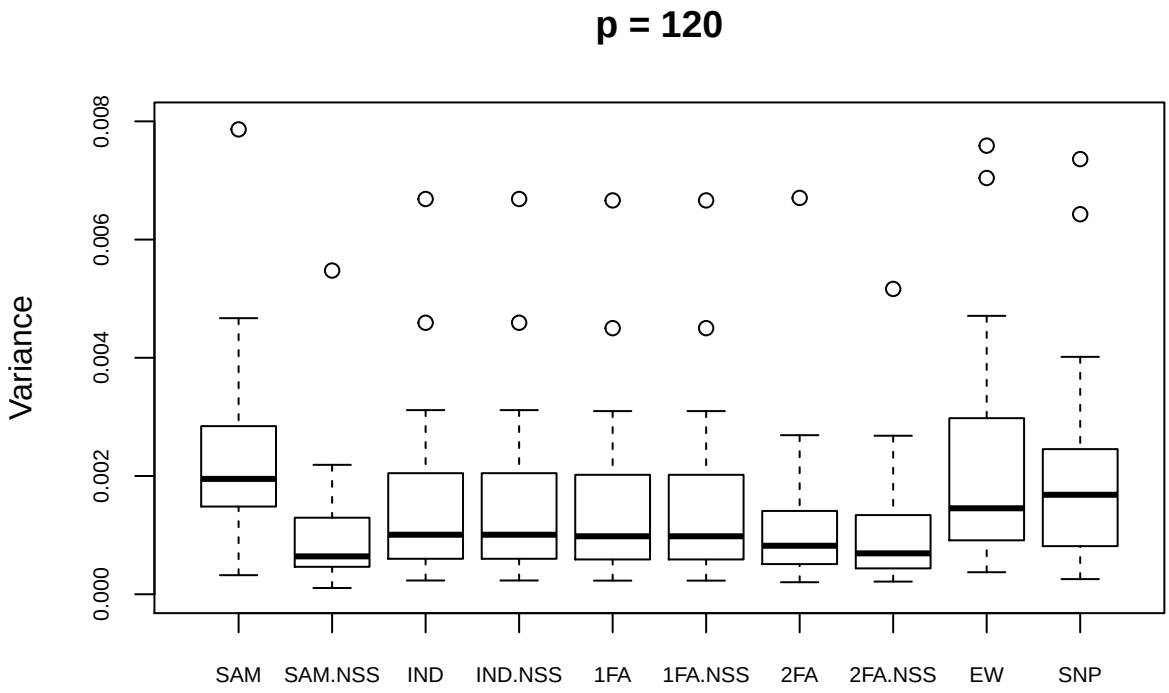
```
rm(list=ls())
library(quadprog)
library(matrixcalc)

sp <- read.table("s_p6306.txt")
r6306 <- read.table("r6306_new.txt")

n <- 2005-1973+1
stock.num <- c(60,120,250,500)
rtn <- array(NA,dim=c(12*33,10,4))
var.array <- array(NA,dim=c(33,10,4))
for(pp in 1:4){
  for(i in 1:n){
    ## (1)
    p <- stock.num[pp]
    ind1<- (5+(i-1)*12):(136+(i-1)*12)
    temp <- r6306[,ind1+1]
    temp <- na.omit(temp)
    ind2 <- sample(dim(temp)[1],p)
    r.temp <- t(temp[ind2,]) # p stocks
    r.train <- r.temp[1:120,]; r.test <- r.temp[121:132,]
    sp.temp <- sp[ind1,2] # S&P 500
    sp.train <- sp.temp[1:120]; sp.test <- sp.temp[121:132]

    ## (2)
    ## SAM
    Sigma.sam <- cov(r.train) ## sample covariance matrix
    if(!is.positive.definite(Sigma.sam))
      Sigma.sam <- 0.99*Sigma.sam + 0.01*diag(diag(Sigma.sam))
    ## IND
    Sigma.ind <- diag(diag(Sigma.sam)) ## independence model

    ## 1FA ##########################################################
    r.mean <- apply(r.train,MARGIN=2,mean)
    r.c <- t(t(r.train)-r.mean)
    sp.c <- sp.train-mean(sp.train)
    sp.s <- sp.c/sd(sp.c)
    beta.hat <- t(r.c) %*% sp.c/(sum(sp.s^2)) ### ???? c>s
    beta.mat <- beta.hat %*% t(beta.hat)
    Psi.hat <- diag(diag(Sigma.sam-beta.mat))
    Sigma.1fa <- beta.mat + Psi.hat ## 1-factor using S&P500 index
    if(!is.positive.definite(Sigma.1fa)){
      Sigma.1fa <- 0.99*Sigma.1fa + 0.01*diag(diag(Sigma.1fa))
    }
    ## 2FA
    Sigma.svd=svd((1-.01)*Sigma.sam+.01*Sigma.ind) ## avoid lipack error
    ## Sigma.svd <- svd(Sigma.sam)
```

```
beta.hat <- Sigma.svd$v[,1:2] %*% diag(sqrt(Sigma.svd$d[1:2]))
beta.mat <- beta.hat %*% t(beta.hat)
Psi.hat <- diag(diag(Sigma.sam-beta.mat))
Sigma.2fa <- beta.mat + Psi.hat ## 2-factor model by PCA
if(!is.positive.definite(Sigma.2fa))
  Sigma.2fa <- 0.99*Sigma.2fa + 0.01*diag(diag(Sigma.2fa))
# Sigma.1fa <- Sigma.2fa
## (3) (4)
A.mat <- cbind(rep(1,p),diag(1,p))
## sample covariance matrix GMVP
w <- solve.QP(Dmat=Sigma.sam,dvec=rep(0,p),Amat=cbind(rep(1,p))
           ,bvec=1,meq=1)$solution
rtn[((i-1)*12+1):(i*12),1,pp] <- r.test%*%w
var.array[i,1,pp] <- var(r.test%*%w)
## sample covariance matrix GMVP NSS
w <- solve.QP(Dmat=Sigma.sam,dvec=rep(0,p),Amat=A.mat
           ,bvec=c(1,rep(0,p)),meq=1)$solution
rtn[((i-1)*12+1):(i*12),2,pp] <- r.test%*%w
var.array[i,2,pp] <- var(r.test%*%w)
## independence model GMVP
w <- solve.QP(Dmat=Sigma.ind,dvec=rep(0,p),Amat=cbind(rep(1,p))
              ,bvec=1,meq=1)$solution
rtn[((i-1)*12+1):(i*12),3,pp] <- r.test%*%w
var.array[i,3,pp] <- var(r.test%*%w)
## independence model GMVP NSS
w <- solve.QP(Dmat=Sigma.ind,dvec=rep(0,p),Amat=A.mat
           ,bvec=c(1,rep(0,p)),meq=1)$solution
rtn[((i-1)*12+1):(i*12),4,pp] <- r.test%*%w
var.array[i,4,pp] <- var(r.test%*%w)
## 1-factor model GMVP
w <- solve.QP(Dmat=Sigma.1fa,dvec=rep(0,p),Amat=cbind(rep(1,p))
              ,bvec=1,meq=1)$solution
rtn[((i-1)*12+1):(i*12),5,pp] <- r.test%*%w
var.array[i,5,pp] <- var(r.test%*%w)
## 1-factor model GMVP NSS
w <- solve.QP(Dmat=Sigma.1fa,dvec=rep(0,p),Amat=A.mat
           ,bvec=c(1,rep(0,p)),meq=1)$solution
rtn[((i-1)*12+1):(i*12),6,pp] <- r.test%*%w
var.array[i,6,pp] <- var(r.test%*%w)
## 2-factor model GMVP
w <- solve.QP(Dmat=Sigma.2fa,dvec=rep(0,p),Amat=cbind(rep(1,p))
              ,bvec=1,meq=1)$solution
rtn[((i-1)*12+1):(i*12),7,pp] <- r.test%*%w
var.array[i,7,pp] <- var(r.test%*%w)
## 2-factor model GMVP NSS
w <- solve.QP(Dmat=Sigma.2fa,dvec=rep(0,p),Amat=A.mat
           ,bvec=c(1,rep(0,p)),meq=1)$solution
rtn[((i-1)*12+1):(i*12),8,pp] <- r.test%*%w
var.array[i,8,pp] <- var(r.test%*%w)
## equally weighted portfolio
rtn[((i-1)*12+1):(i*12),9,pp] <- r.test%*%rep(1,p)/p
var.array[i,9,pp] <- var(r.test%*%rep(1,p)/p)
## S&P500 return
```

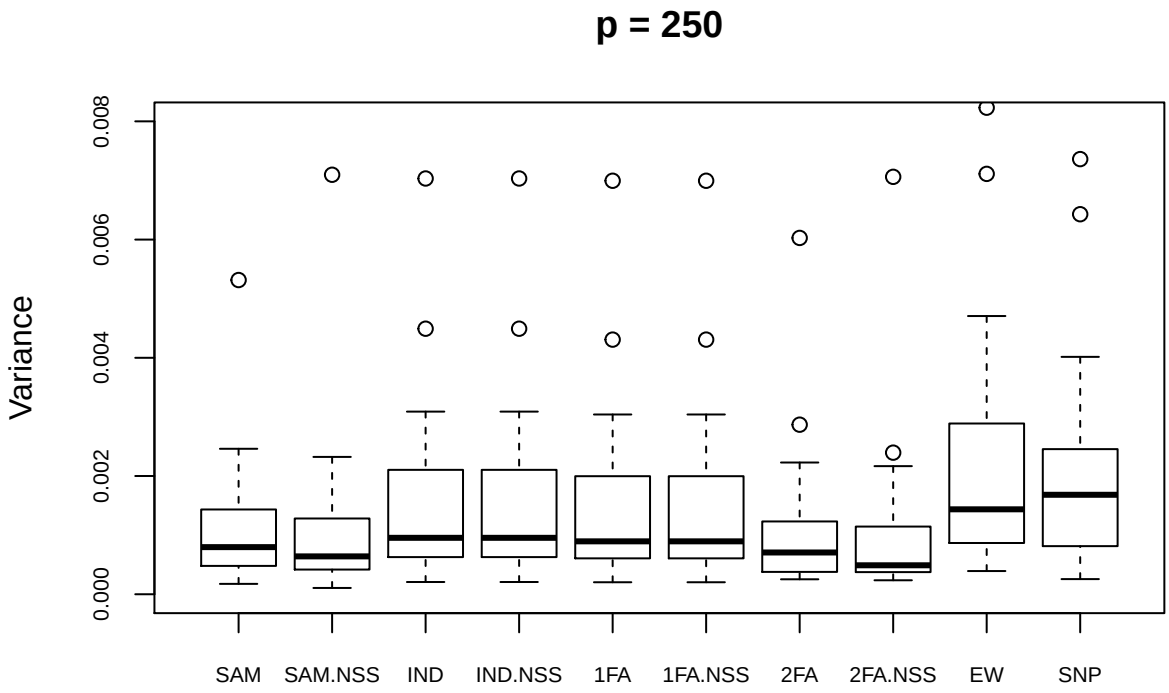
```
    rtn[((i-1)*12+1):(i*12),10,pp] <- sp.test
    var.array[i,10,pp] <- var(sp.test)
  }
}
## Use boxplot to display the portfolio variances
methods <- c("SAM","SAM.NSS","IND","IND.NSS","1FA","1FA.NSS","2FA","2FA.NSS","EW","SNP")
colnames(var.array) <- methods
boxplot(var.array[,,1],ylab="Variance",ylim=c(0,0.008),cex.axis=0.7)
title("p = 60")
```



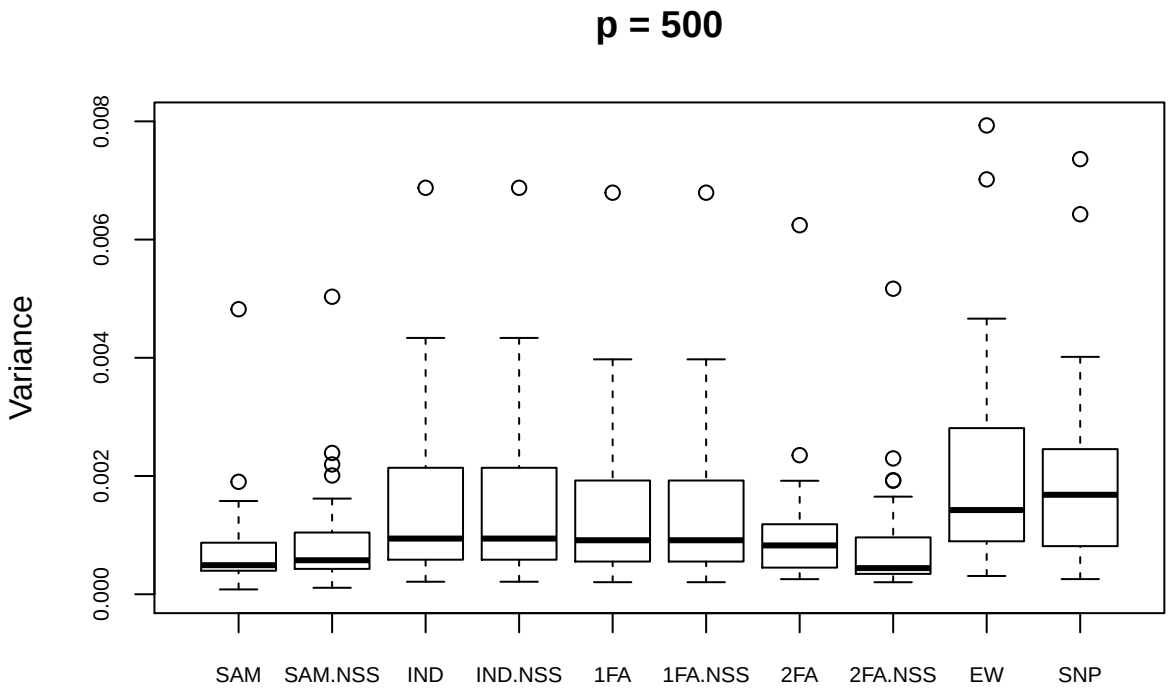

```
boxplot(var.array[,,2],ylab="Variance",ylim=c(0,0.008),cex.axis=0.7)
title("p = 120")
```




```
boxplot(var.array[,,3],ylab="Variance",ylim=c(0,0.008),cex.axis=0.7)
title("p = 250")
```




```
boxplot(var.array[,,4],ylab="Variance",ylim=c(0,0.008),cex.axis=0.7)
title("p = 500")
```




```
## Which portfolio has the smallest variance, on average?
var.avg <- c()
for(k in seq(10)){
  var.avg <- c(var.avg,mean(c(var.array[,k,1],var.array[,k,2],
                              var.array[,k,3],var.array[,k,4])))
}
which.min(var.avg)
```

```
## [1] 8
```

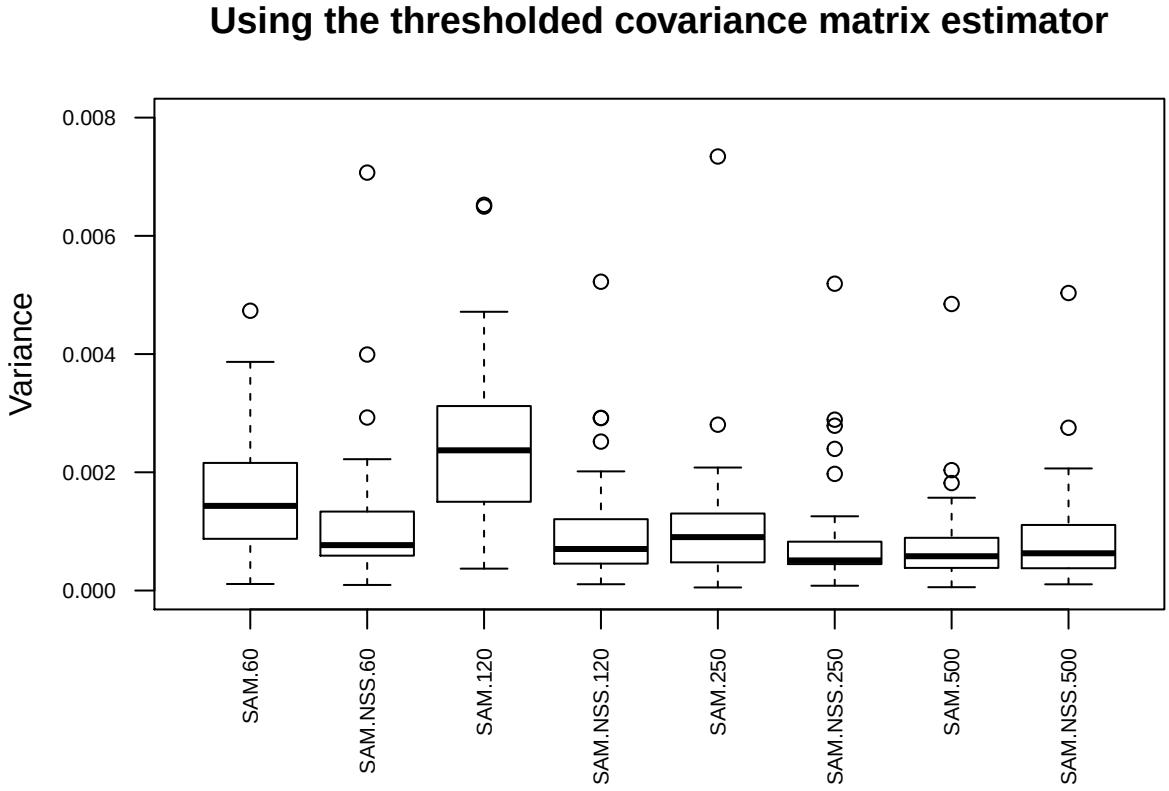
Two-factor model with no-short-selling(NSS) has the smallest variance on average. 2FA and 2FA.NSS perform very well. And, roughly speaking, when the number of stocks increase, SAM will perform better.

(c)

```
n <- 2005-1973+1
stock.num <- c(60,120,250,500)
rtn.arr <- array(NA,dim=c(12*33,2,4))
var.arr <- array(NA,dim=c(33,2,4))
for(pp in 1:4){
  for(i in 1:n){
    ## (1)
    p <- stock.num[pp]
    ind1<- (5+(i-1)*12):(136+(i-1)*12)
```

```
temp <- r6306[,ind1+1]
temp <- na.omit(temp)
ind2 <- sample(dim(temp)[1],p)
r.temp <- t(temp[ind2,]) # p stocks
r.train <- r.temp[1:120,]; r.test <- r.temp[121:132,]
sp.temp <- sp[ind1,2] # S&P 500
sp.train <- sp.temp[1:120]; sp.test <- sp.temp[121:132]

## Choose the threshold level T by five fold cross-validation
n.train <- 120
set.seed(1)
nfolds <- 5
t <- 50
s <- split(sample(n.train),rep(1:nfolds,length=n.train))
if(pp == 1)
  T <- seq(1e-06,1e-04,length=t)
else if(pp ==2)
  T <- seq(1e-07,1e-05,length=t)
else
  T <- seq(1e-08,1e-06,length=t)
var.cv.avg <- c()
for(ii in t){
  var.cv <- c()
  for(j in seq(nfolds)){
    Sigma.temp <- cov(r.train[-s[[j]],])
    Sigma.temp[abs(Sigma.temp) < T[ii]] <- 0
    is.positive.definite(Sigma.temp)
    if(!is.positive.definite(Sigma.temp))
      Sigma.temp <- 0.99*Sigma.temp + 0.01*diag(diag(Sigma.temp))
    w <- solve.QP(Dmat=Sigma.temp,dvec=rep(0,p),Amat=cbind(rep(1,p))
          ,bvec=1,meq=1)$solution
    var.cv <- c(var.cv,var(r.train[s[[j]],]%*%w))
  }
  var.cv.avg <- c(var.cv.avg,mean(var.cv))
}
T <- T[which.min(var.cv.avg)]

## (2)
## SAM
Sigma.sam <- cov(r.train) ## sample covariance matrix
Sigma.sam[abs(Sigma.sam) < T] <- 0
if(!is.positive.definite(Sigma.sam))
  Sigma.sam <- 0.99*Sigma.sam + 0.01*diag(diag(Sigma.sam))

## (3) (4)
A.mat <- cbind(rep(1,p),diag(1,p))
## sample covariance matrix GMVP
w <- solve.QP(Dmat=Sigma.sam,dvec=rep(0,p),Amat=cbind(rep(1,p))
           ,bvec=1,meq=1)$solution
rtn.arr[((i-1)*12+1):(i*12),1,pp] <- r.test%*%w
var.arr[i,1,pp] <- var(r.test%*%w)
## sample covariance matrix GMVP NSS
w <- solve.QP(Dmat=Sigma.sam,dvec=rep(0,p),Amat=A.mat
```

```
                ,bvec=c(1,rep(0,p)),meq=1)$solution
    rtn.arr[((i-1)*12+1):(i*12),2,pp] <- r.test%*%w
    var.arr[i,2,pp] <- var(r.test%*%w)
  }
}
data.plot <- cbind(var.arr[,,1],var.arr[,,2],var.arr[,,3],var.arr[,,4])
colnames(data.plot) <- c("SAM.60","SAM.NSS.60","SAM.120","SAM.NSS.120","SAM.250","SAM.NSS.250","SAM.500
boxplot(data.plot,ylab="Variance",ylim=c(0,0.008),las=2,cex.axis=0.7,main="Using the thresholded covaria
```

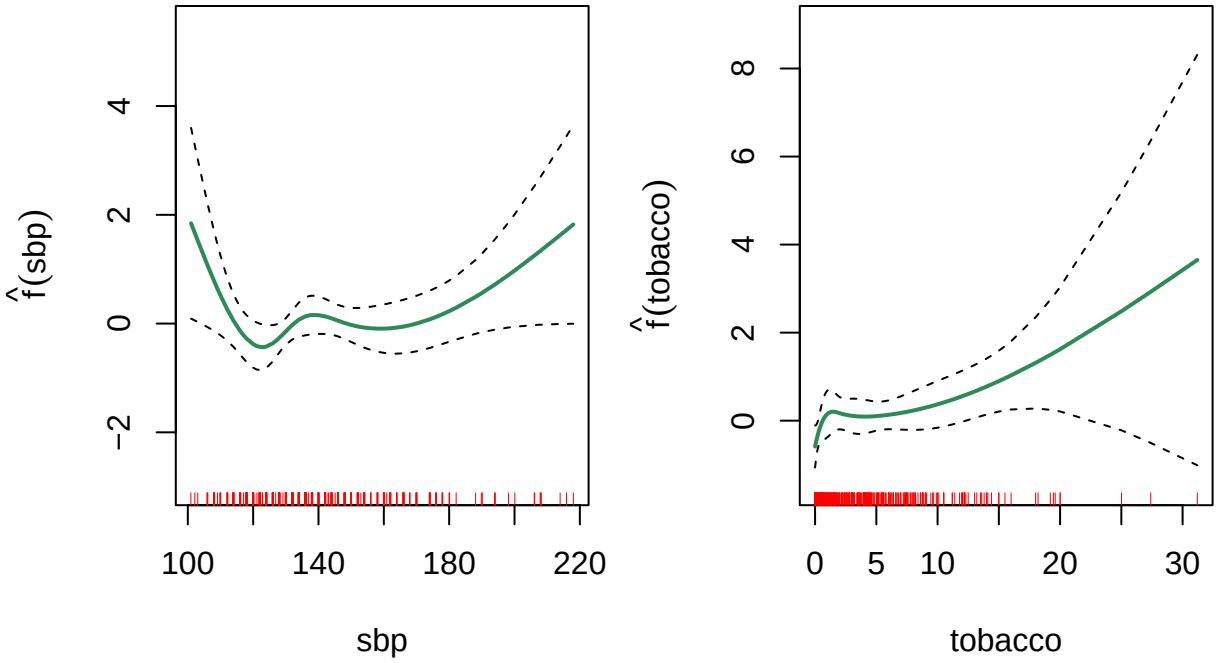


After using the thresholded covariance matrix estimator, the variances of the portfolios will not change too much!

2.

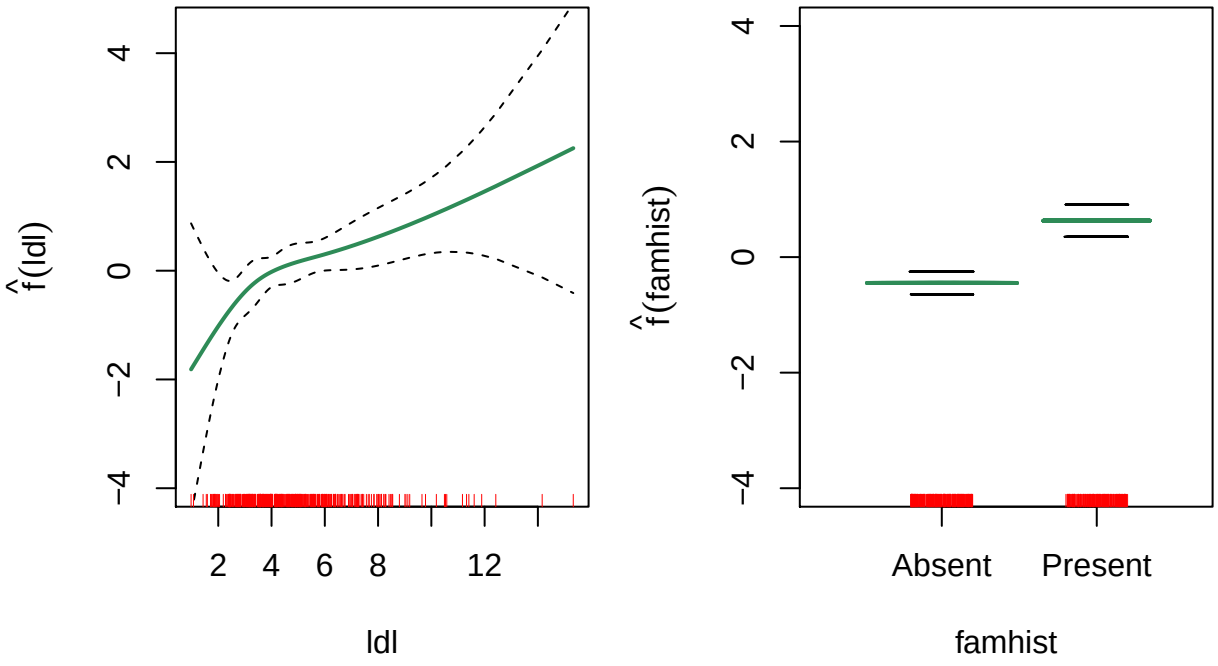
```
rm(list=ls())
library(glmnet)
library(graphics) ## rug plot
library(gam)
library(splines)
SAheart <- read.csv("SAheart.data",row.names=1)
SAheart[,"famhist"] <- scale(as.numeric(SAheart[,"famhist"]))
# SAheart <- data.frame(scale(SAheart))
attach(SAheart)
form <- "chd ~ ns(sbp,4) + ns(tobacco,4) + ns(ldl,4) + famhist + ns(obesity,4)  + ns(age,4)"
x <- data.frame(cbind(ns(sbp,4),ns(tobacco,4),ns(ldl,4)
```

```
                      ,famhist,ns(obesity,4),ns(age,4)))
form <- formula(form)
# mdl <- glm(form,data=SAheart,family=binomial)
# par(mfrow=c(1,2),mar=c(4.5,4.5,1,1),oma=c(0,0,4,0))
# plot.gam(mdl,terms="ns(sbp, 4)",col="seagreen",se=T)
# plot.gam(mdl,terms="ns(tobacco, 4)",col="seagreen",se=T)
# plot.gam(mdl,terms="ns(ldl, 4)",col="seagreen",se=T)
# plot.gam(mdl,terms="famhist",col="seagreen",se=T)
# plot.gam(mdl,terms="ns(obesity, 4)",col="seagreen",se=T)
# plot.gam(mdl,terms="ns(age, 4)",col="seagreen",se=T)


mdl.gam <- gam(form,data=SAheart,family=binomial)
terms.pred <- predict(mdl.gam,type="terms",newdata=x,se=T)
par(mfrow=c(1,2),mar=c(4.5,4.5,1,1),oma=c(0,0,4,0))
## sbp
f.hat <- terms.pred$fit[,1]
se <- terms.pred$se.fit[,1]
data.temp <- data.frame(sbp,f.hat,f.hat+2*se,f.hat-2*se)
data.new <- data.temp[order(sbp),]
plot(data.new[,1],data.new[,2],type="l",xlab="sbp",ylab=expression(hat(f)(sbp)),
     ylim=c(-3,5.5),col="seagreen",lwd=2)
lines(data.new[,1],data.new[,3],lty="dashed")
lines(data.new[,1],data.new[,4],lty="dashed")
rug(jitter(sbp),col="red")

## tobacco
f.hat <- terms.pred$fit[,2]
se <- terms.pred$se.fit[,2]
data.temp <- data.frame(tobacco,f.hat,f.hat+2*se,f.hat-2*se)
data.new <- data.temp[order(tobacco),]
plot(data.new[,1],data.new[,2],type="l",xlab="tobacco",
     ylab=expression(hat(f)(tobacco)),ylim=c(-1.5,9),col="seagreen",lwd=2)
lines(data.new[,1],data.new[,3],lty="dashed")
lines(data.new[,1],data.new[,4],lty="dashed")
rug(jitter(tobacco),col="red")
```



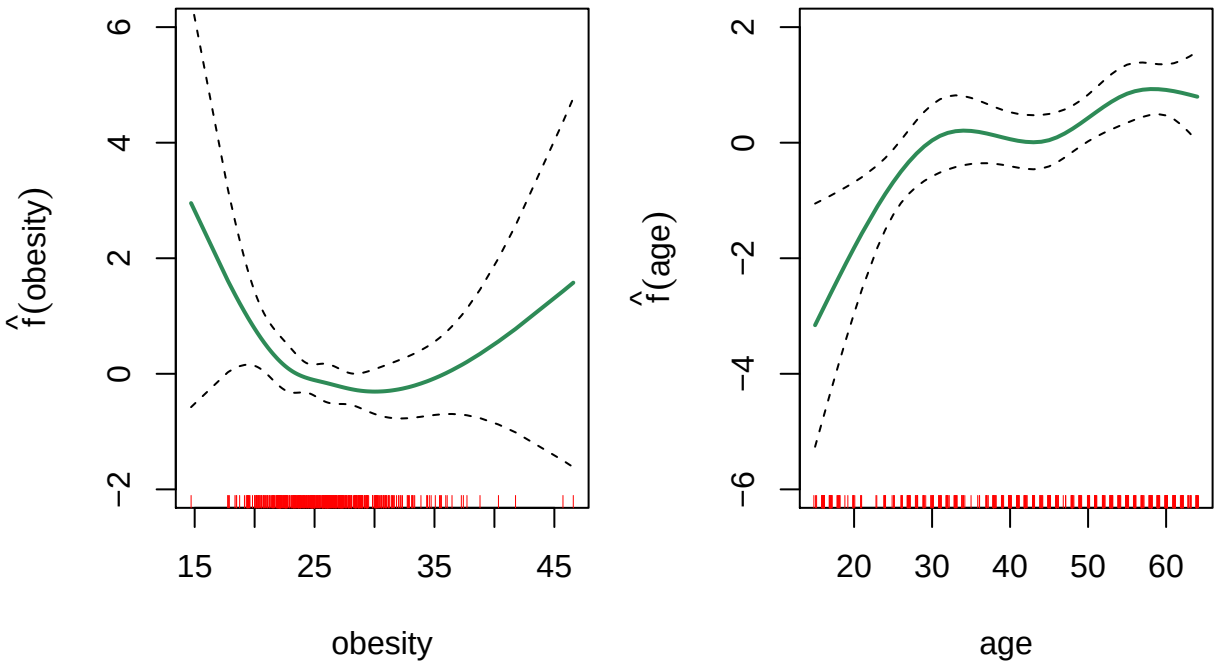

```
## ldl
f.hat <- terms.pred$fit[,3]
se <- terms.pred$se.fit[,3]
data.temp <- data.frame(ldl,f.hat,f.hat+2*se,f.hat-2*se)
data.new <- data.temp[order(ldl),]
plot(data.new[,1],data.new[,2],type="l",xlab="ldl",ylab=expression(hat(f)(ldl)),
     ylim=c(-4,4.5),col="seagreen",lwd=2)
lines(data.new[,1],data.new[,3],lty="dashed")
lines(data.new[,1],data.new[,4],lty="dashed")
rug(jitter(ldl),col="red")

## famhist
f.hat <- terms.pred$fit[,4]
se <- terms.pred$se.fit[,4]
data.temp <- data.frame(famhist,f.hat,f.hat+2*se,f.hat-2*se)
data.new <- data.temp[order(famhist),]
data.uni <- unique(data.temp)
absent.num <- sum(data.new[,1]<0)
plot(jitter(data.new[1:absent.num,1],a=0.7*270/192),data.new[1:absent.num,2],
     type="l",xlab="famhist",ylab=expression(hat(f)(famhist)),xlim=c(-2.5,2.5),
     ylim=c(-4,4),col="seagreen",xaxt="n",lwd=2)
lines(jitter(data.new[(absent.num+1):462,1],a=0.7),data.new[(absent.num+1):462,2],
      col="seagreen",lwd=2)
lines(jitter(data.new[1:absent.num,1],a=0.3*270/192),data.new[1:absent.num,3],lty=2)
lines(jitter(data.new[(absent.num+1):462,1],a=0.3*270/192),
      data.new[(absent.num+1):462,3],lty=2)
```

```
lines(jitter(data.new[1:absent.num,1],a=0.3*270/192),data.new[1:absent.num,4],lty=2)
lines(jitter(data.new[(absent.num+1):462,1],a=0.3*270/192),
      data.new[(absent.num+1):462,4],lty=2)
axis(side=1, at=unique(famhist), labels=c("Present","Absent"))
rug(jitter(famhist),col="red")
```




```
## obesity
f.hat <- terms.pred$fit[,5]
se <- terms.pred$se.fit[,5]
data.temp <- data.frame(obesity,f.hat,f.hat+2*se,f.hat-2*se)
data.new <- data.temp[order(obesity),]
plot(data.new[,1],data.new[,2],type="l",xlab="obesity",
     ylab=expression(hat(f)(obesity)),ylim=c(-2,6),col="seagreen",lwd=2)
lines(data.new[,1],data.new[,3],lty="dashed")
lines(data.new[,1],data.new[,4],lty="dashed")
rug(jitter(obesity),col="red")

## age
f.hat <- terms.pred$fit[,6]
se <- terms.pred$se.fit[,6]
data.temp <- data.frame(age,f.hat,f.hat+2*se,f.hat-2*se)
data.new <- data.temp[order(age),]
plot(data.new[,1],data.new[,2],type="l",xlab="age",
     ylab=expression(hat(f)(age)),ylim=c(-6,2),col="seagreen",lwd=2)
lines(data.new[,1],data.new[,3],lty="dashed")
```

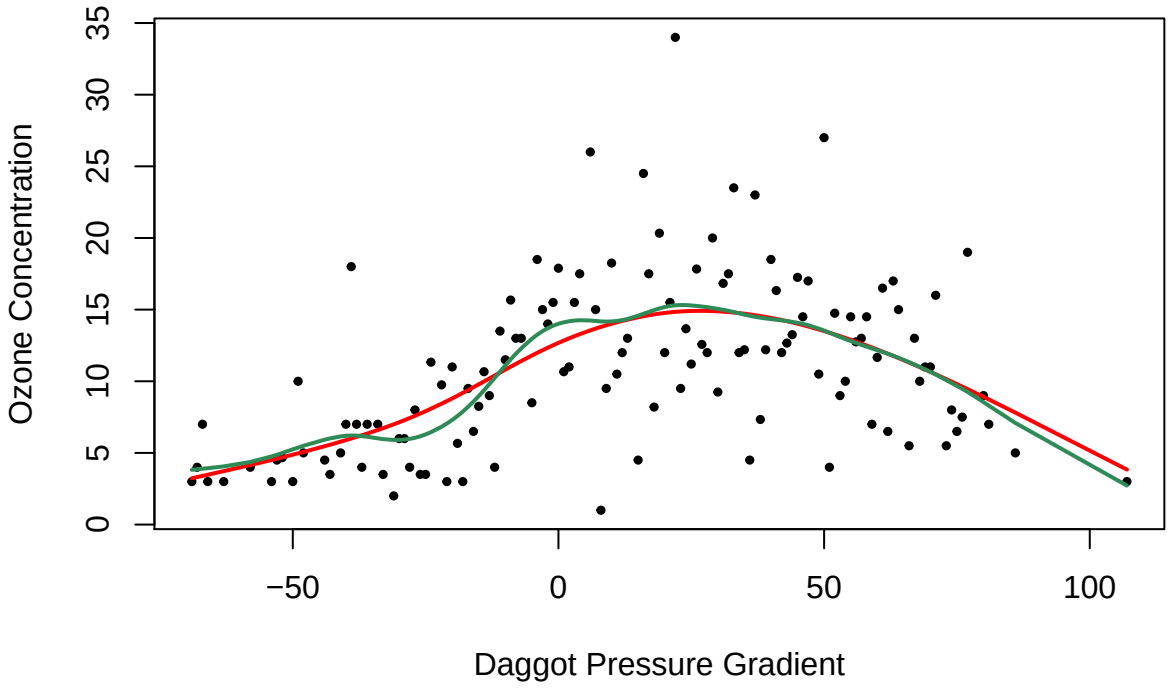
```
lines(data.new[,1],data.new[,4],lty="dashed")
rug(jitter(age),col="red")
```



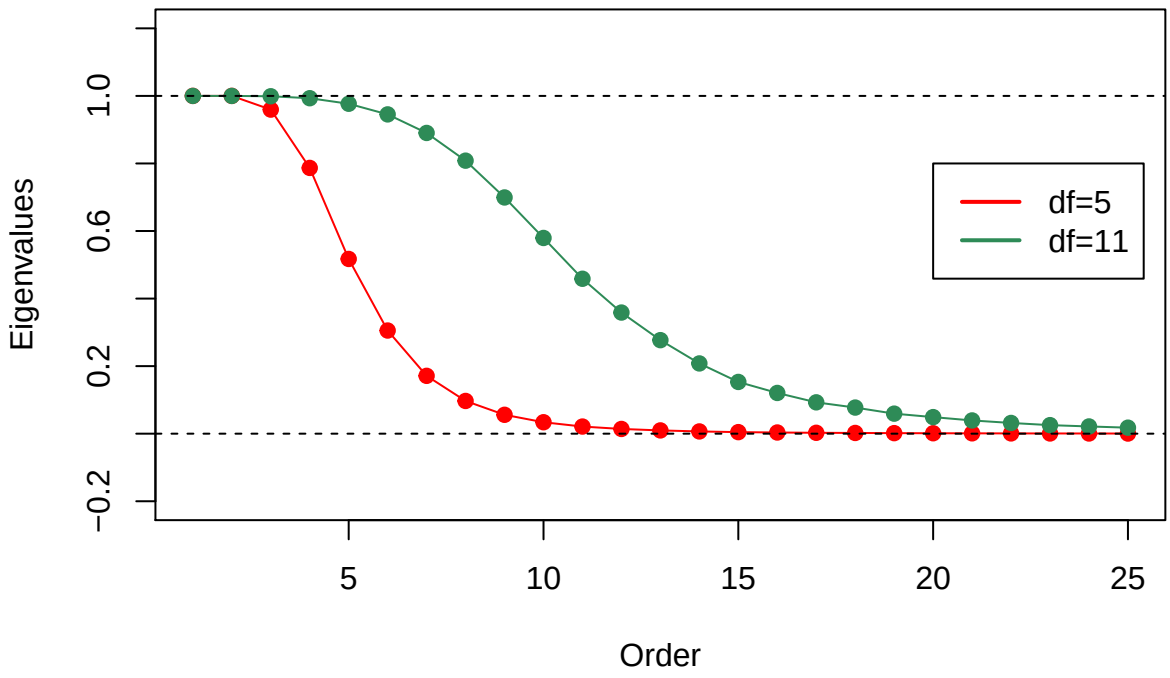

```
detach(SAheart)
################################################################
```

3.

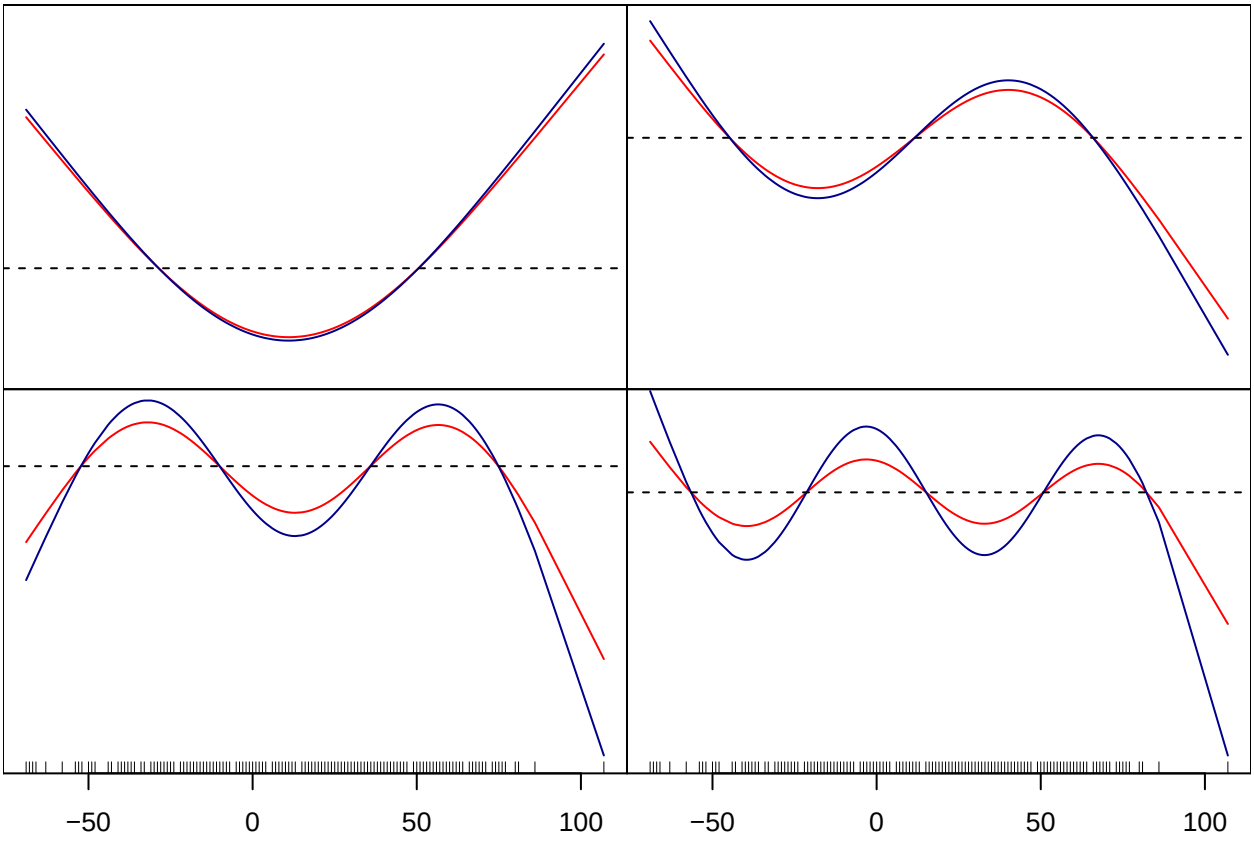
```
rm(list=ls())
library(sp)
library(rgeos)
LAozone <- read.csv("LAozone.data")
attach(LAozone)
df <- cbind(ozone,dpg)
df <- aggregate(df,by=list(dpg=dpg),FUN=mean)[,-3]

plot(df[,1],df[,2],cex=0.5,pch=19,xlab="Daggot Pressure Gradient",ylab="Ozone Concentration")
fit1 <- smooth.spline(df[,1],df[,2],df=5)
lines(fit1,col="red",lwd=2)
fit2 <- smooth.spline(df[,1],df[,2],df=11)
lines(fit2,col="seagreen",lwd=2)
```



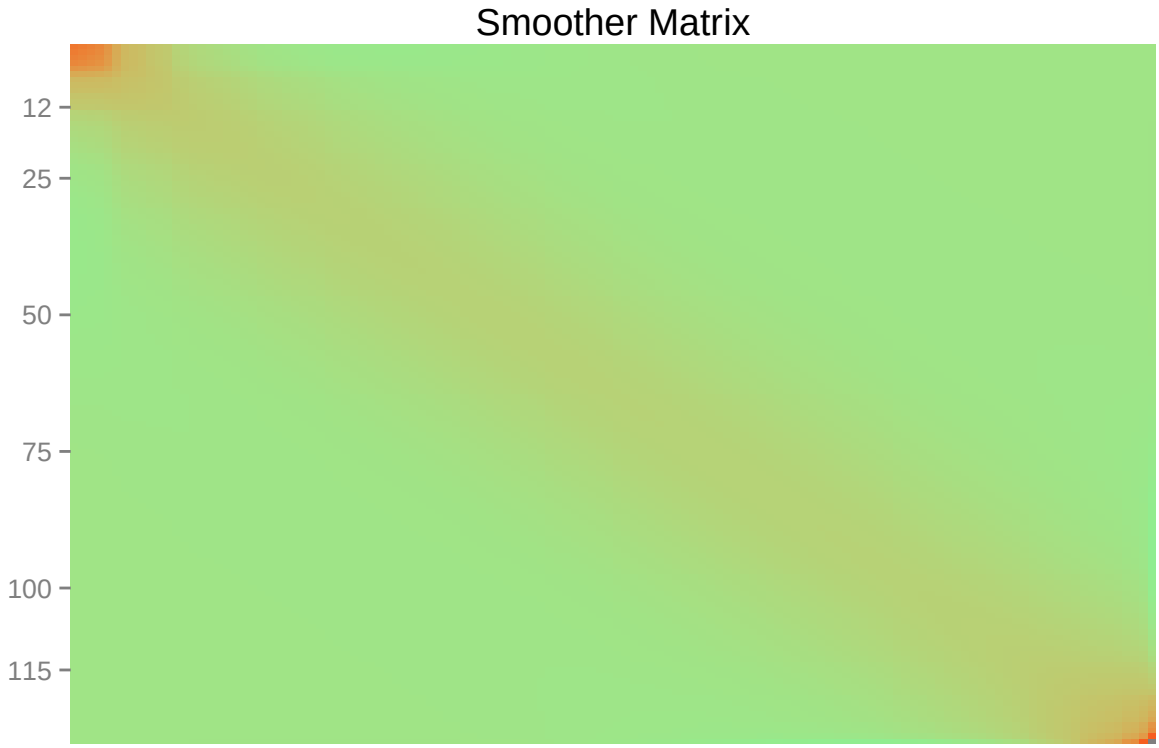

```
## Function to get the smoother matrix
smooth.matrix = function(x, df){
n = length(x);
A = matrix(0, n, n);
for(i in 1:n){
y = rep(0, n); y[i]=1;
yi = smooth.spline(x, y, df=df)$y;
A[,i]= yi;
}
(A+t(A))/2;
}

smooth.matrix.5 <- smooth.matrix(df[,1],5)
smooth.matrix.11 <- smooth.matrix(df[,1],11)
aux1 <- eigen(smooth.matrix.5)
U1 <- aux1$vectors ; rho1 <- aux1$values ## eigen vectors and eigenvalues
aux2 <- eigen(smooth.matrix.11)
U2 <- aux2$vectors ; rho2 <- aux2$values
Order <- 1:25
plot(Order,rho1[1:25],pch=19,col="red",ylab="Eigenvalues",ylim=c(-0.2,1.2))
lines(Order,rho1[1:25],col="red")
points(Order,rho2[1:25],pch=19,col="seagreen")
lines(Order,rho2[1:25],col="seagreen")
abline(h=1,lty=2) ; abline(h=0,lty=2)
legend(x=20,y=0.8,legend=c("df=5","df=11"),col=c("red","seagreen"),lty=1,lwd=2)
```



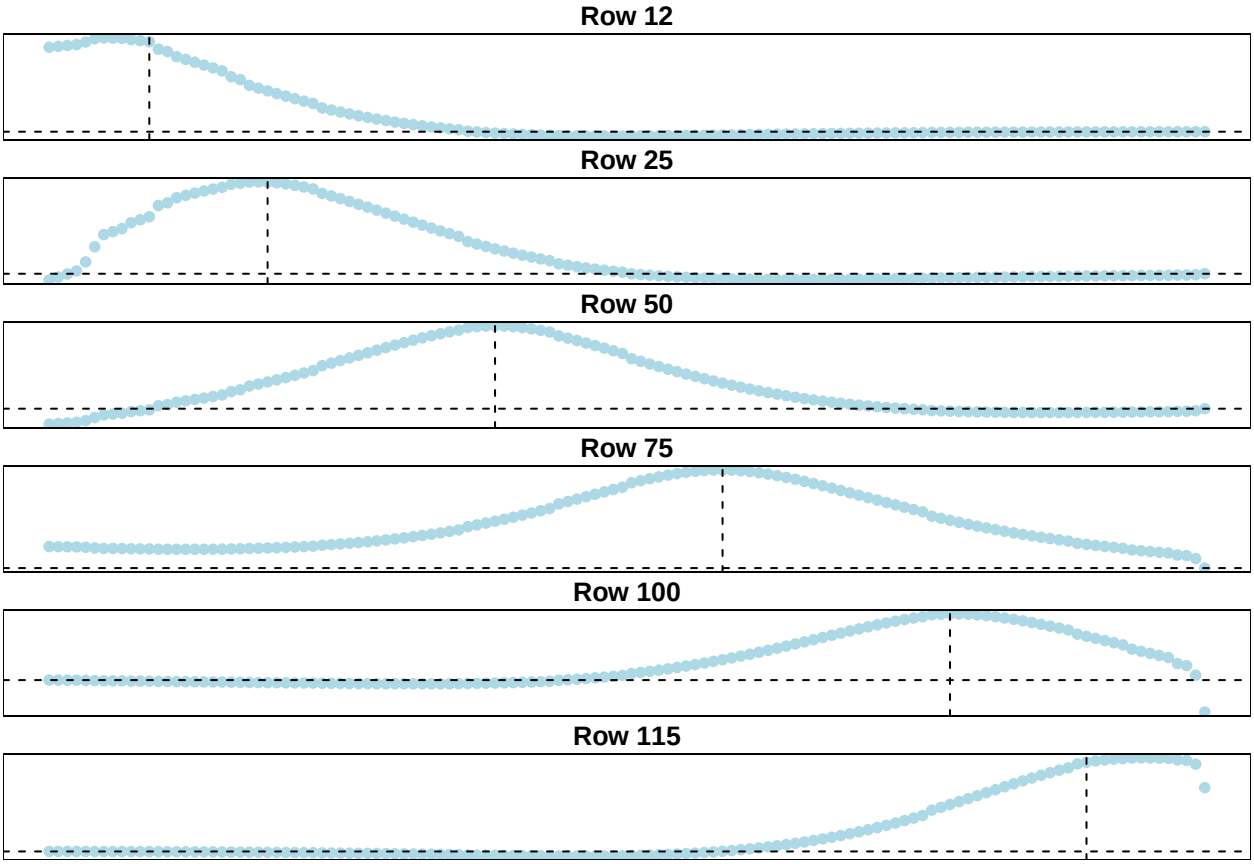

```
par(mfrow=c(2,2),mar=c(0,0,0,0),oma=c(3,0,0,0))
plot(df[,1],U1[,3],typ="l",col="red",ylim=c(-0.15,0.35),xlab="",
     ylab="",xaxt="n",yaxt="n")
lines(df[,1],1.05*U1[,3],col="darkblue")
abline(h=0,lty=2)
## abline(h=0.001240704,lty=2) ## !!!!
plot(df[,1],U1[,4],typ="l",col="red",ylim=c(-0.5,0.25),xlab="",
     ylab="",xaxt="n",yaxt="n")
lines(df[,1],1.2*U1[,4],col="darkblue")
abline(h=0,lty=2)
plot(df[,1],U1[,5],typ="l",col="red",ylim=c(-0.7,0.15),xlab="",
     ylab="",yaxt="n")
lines(df[,1],1.5*U1[,5],col="darkblue")
abline(h=0,lty=2)
rug(df[,1])
plot(df[,1],-U1[,6],typ="l",ylim=c(-0.9,0.3),col="red",xlab="",
     ylab="",yaxt="n")
lines(df[,1],-2*U2[,6],col="darkblue")
abline(h=0,lty=2)
rug(df[,1])
```




```
par(mfrow=c(1,1))
detach(LAozone)

smooth.matrix.plot <- smooth.matrix.5[,128:1]
library(ggplot2)
library(reshape)

qq <- melt(smooth.matrix.plot)
p <-ggplot(qq, aes(X1, X2, fill = value)) + geom_raster() +
    labs(title = "Smoother Matrix",x="",y="") +
    scale_fill_gradientn(colours=c("lightgreen","red")
                         ,values=c(0,0.5),guide=F)+
    scale_x_continuous(breaks=c(),expand = c(0, 0)) +
    scale_y_continuous(breaks=c(14,29,54,79,104,117),labels=c("115","100","75","50","25","12"),expand =
p
```




```
## Equivalent Kernels
par(mfrow=c(6,1),mar=c(0,0,1.5,0),oma=c(0,0,0,0))
x <- 1:128
plot(x,smooth.matrix.5[12,],pch=19,xaxt="n",yaxt="n",
     main="Row 12",col="lightblue")
abline(v=12,lty=2) ; abline(h=smooth.matrix.5[12,128],lty=2)
plot(x,smooth.matrix.5[25,],pch=19,xaxt="n",yaxt="n",
     main="Row 25",col="lightblue")
abline(v=25,lty=2) ; abline(h=smooth.matrix.5[25,128],lty=2)
plot(x,smooth.matrix.5[50,],pch=19,xaxt="n",yaxt="n",
     main="Row 50",col="lightblue")
abline(v=50,lty=2) ; abline(h=smooth.matrix.5[50,128],lty=2)
plot(x,smooth.matrix.5[75,],pch=19,xaxt="n",yaxt="n",
     main="Row 75",col="lightblue")
abline(v=75,lty=2) ; abline(h=smooth.matrix.5[75,128],lty=2)
plot(x,smooth.matrix.5[100,],pch=19,xaxt="n",yaxt="n",
     main="Row 100",col="lightblue")
abline(v=100,lty=2) ; abline(h=smooth.matrix.5[100,1],lty=2)
plot(x,smooth.matrix.5[115,],pch=19,xaxt="n",yaxt="n",
     main="Row 115",col="lightblue")
abline(v=115,lty=2) ; abline(h=smooth.matrix.5[115,1],lty=2)
```


Row 12
Row 25
Row 50
Row 75
Row 100
Row 115