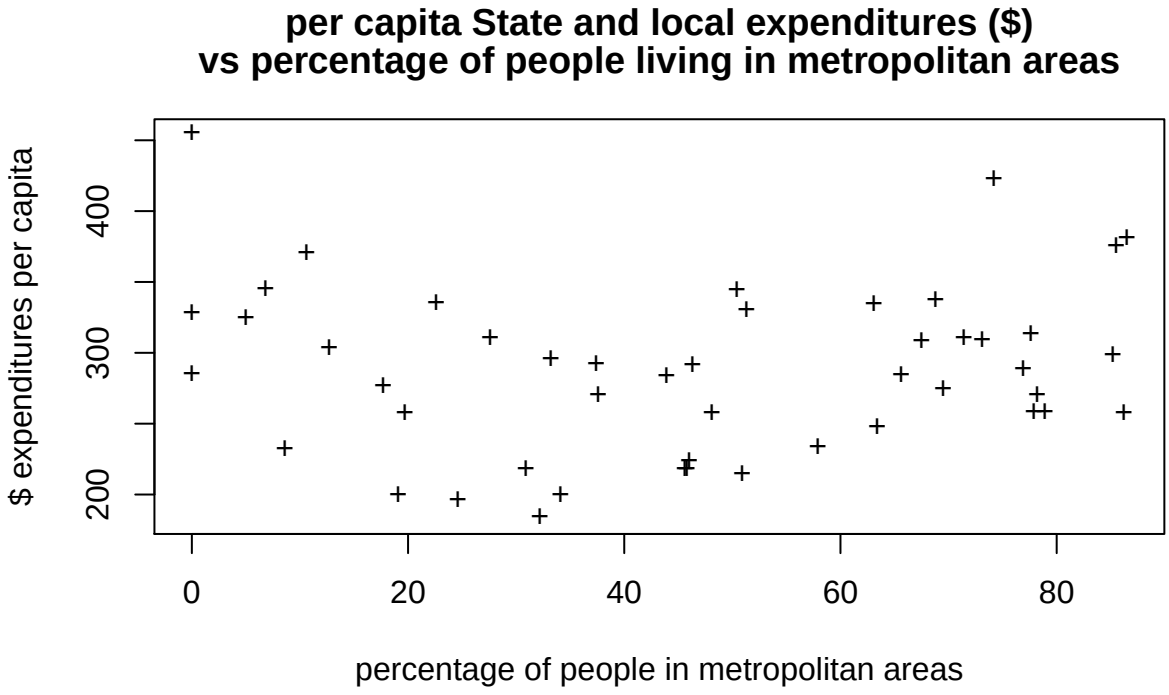
Computer Lab 4

Thomas Zhang

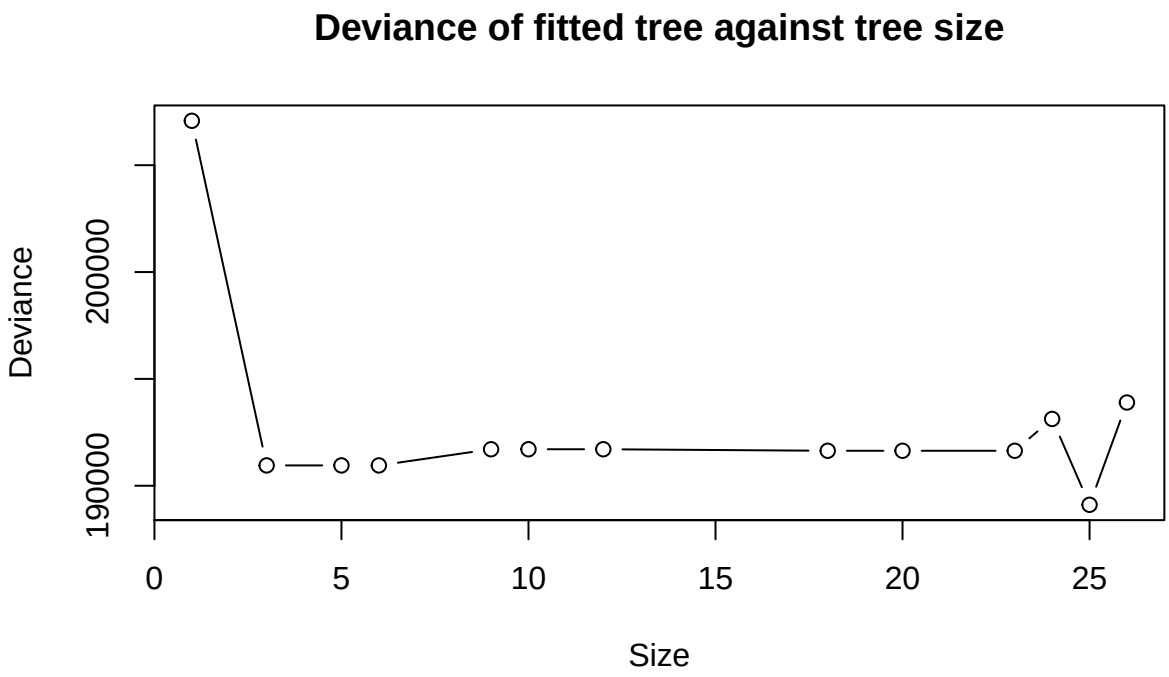
2015-11-16

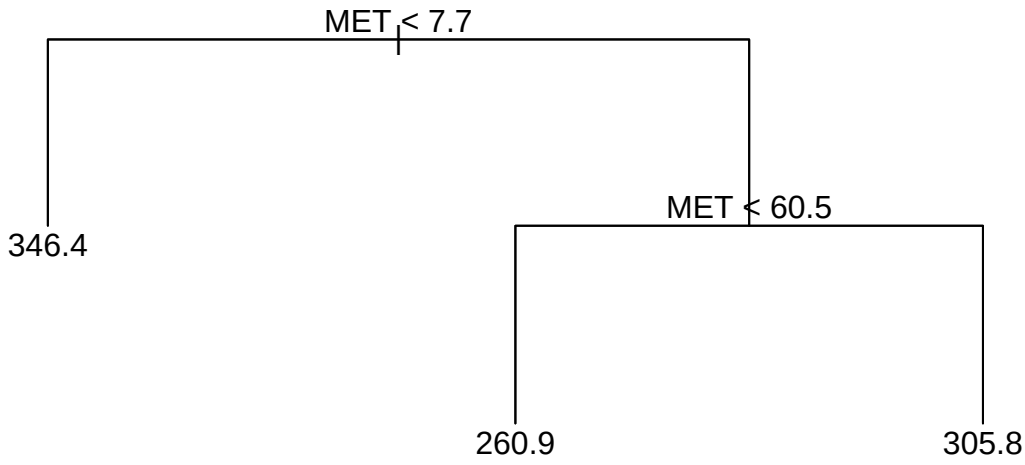
Assignment 1

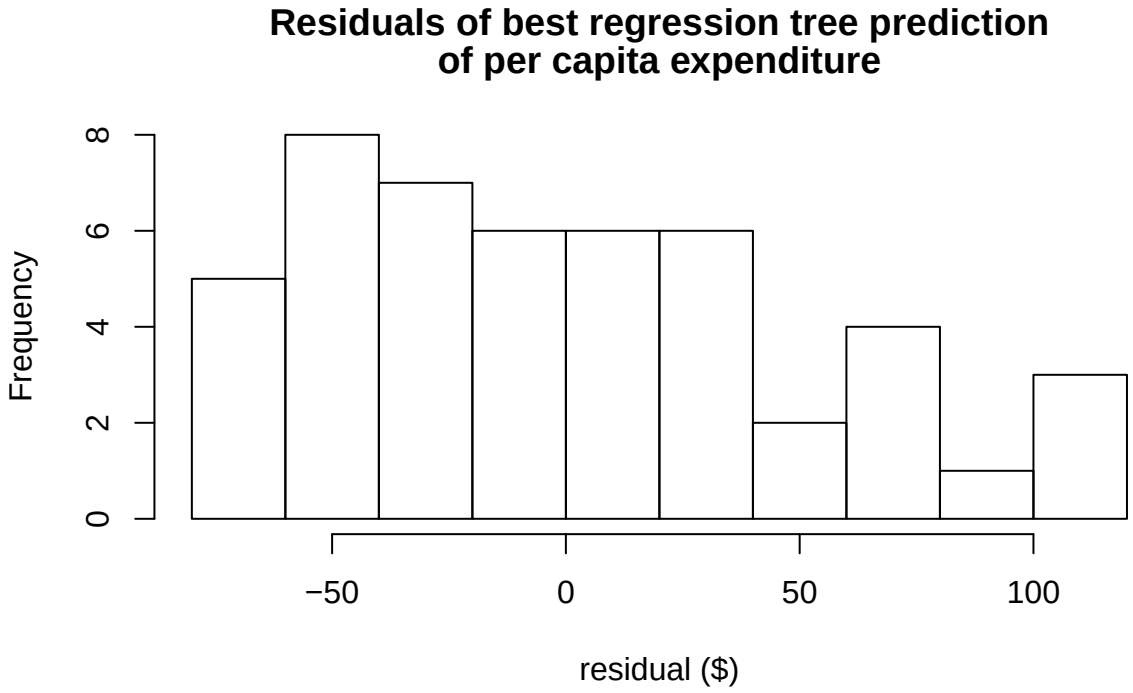
We plot EX vs. MET in a scatterplot.

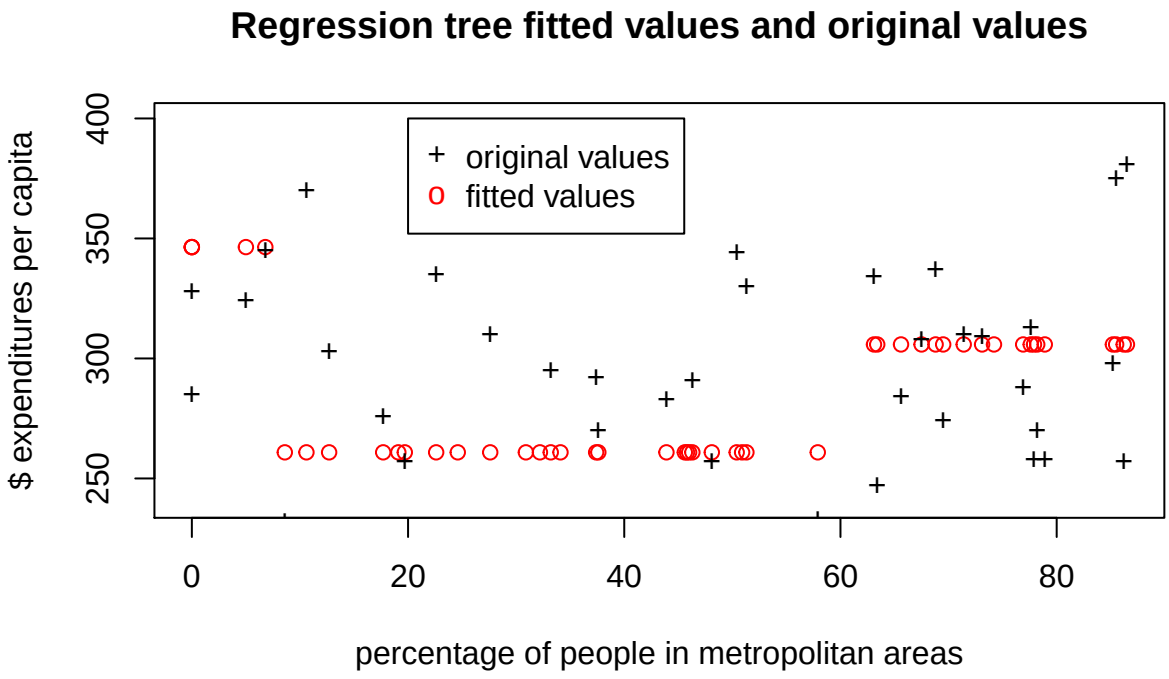


The data looks random, but one can with some goodwill discern a U-shape in the data. We try to fit regression trees with minimum number of observations in a leaf set to two and then greedily prune it and pick the pruned regression tree with the lowest cross-validation score in terms of deviance. Then we plot a histogram of the residuals of the best regression tree and its fitted response values.


Deviance of fitted tree against tree size
Deviance
190000
200000
0
5
10
15
20
25
Size

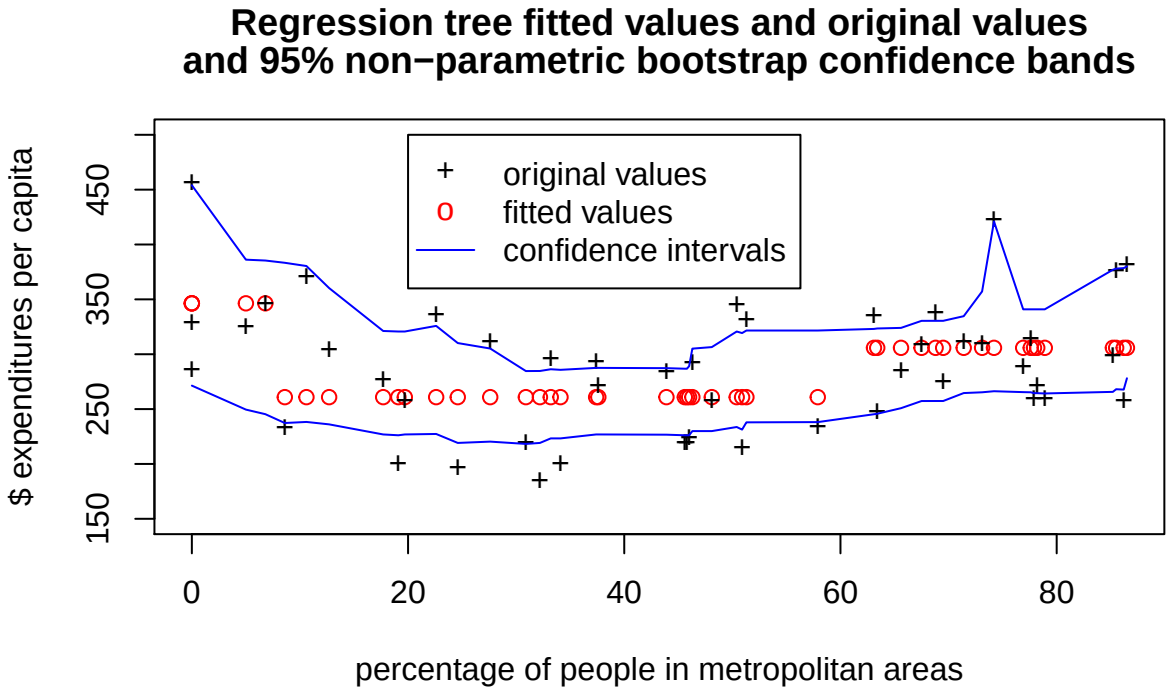

MET < 7.7
MET < 60.5
346.4
260.9
305.8





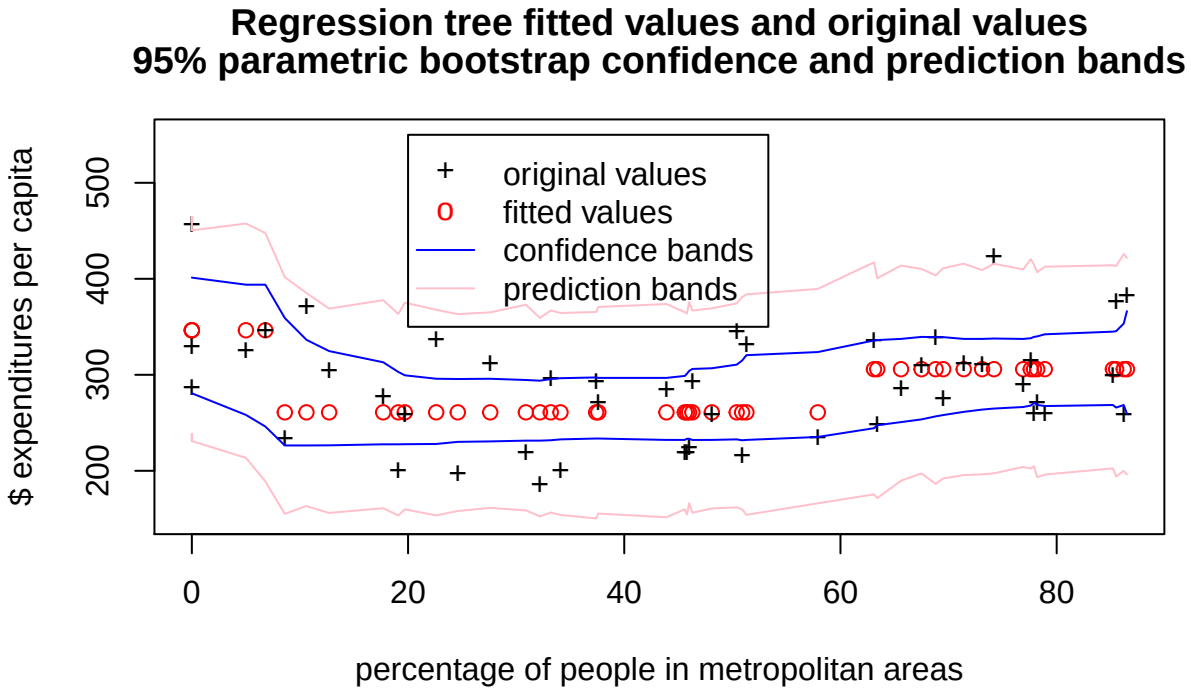
From the deviance plot it looks like the optimal number of tree leaves is three. The regression tree produced by pruning seems good and easy to understand. The residuals look more uniformly distributed than normally distributed, but that may be an effect of the location of the data points. The fitted values are a crude estimation but probably as good as we can get with tree size three.

We proceed to plot the non-parametric bootstrap 95% confidence bands for this regression tree fit.



We see that the bumps in the confidence bands happen where the data has outliers. My guess is that during the bootstrap (re)sampling of data points it catches these outliers and use them to generate the confidence interval. Therefore the confidence band becomes bumpy. We see that the confidence band does exhibit the hypothesized U-shape of the data but that many data points lie outside the confidence bands. The value of this confidence band is thus limited.

We try to improve this situation by finding parametric bootstrap 95% confidence bands and 95% prediction bands. The parametrization happens in the tree leaves, where we assume that the data is normally distributed with mean the leaf label and variance the sample variance of the residuals of the regression tree fitted data.



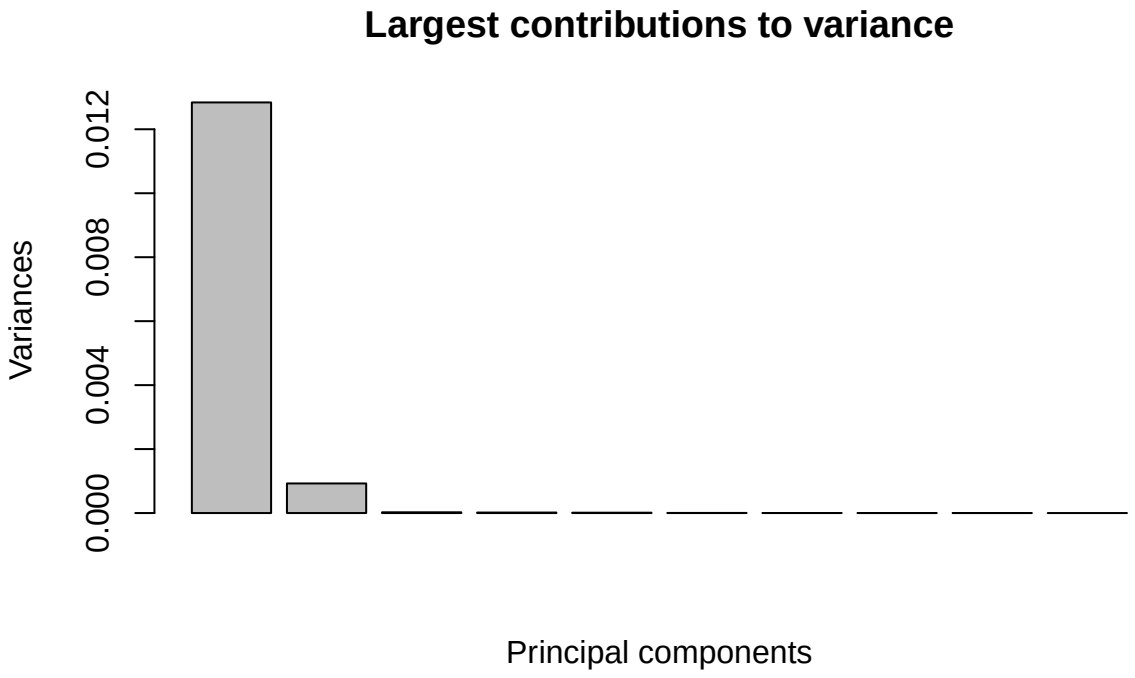
It is seen that the confidence bands now are smoother and more narrow, and they follow the fitted data a little better. Only two of 48 observations lie outside the prediction bands, which is expected. It seems that the assumed normal distribution of data in tree leaves have improved the accuracy of the bootstrap and made parametric bootstrap the superior choice compared to non-parametric bootstrap.

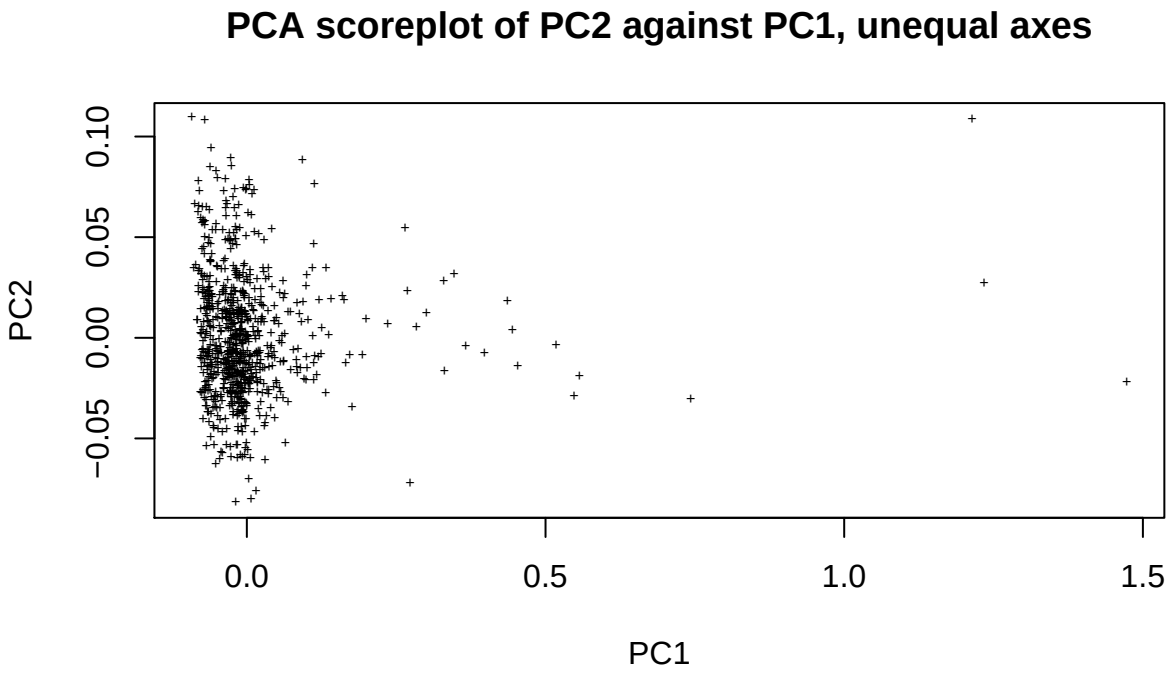
Assignment 2

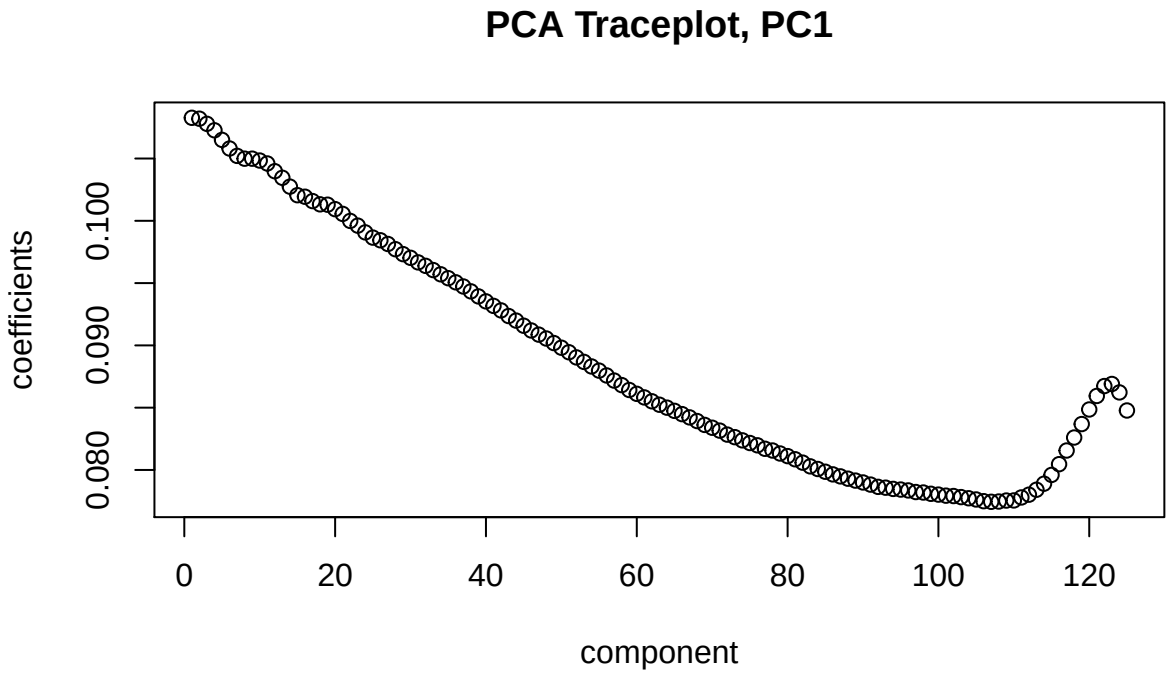
This assignment invites us to explore how the Viscosity variable in file `NIRspectra.xls` depends on the other variables which represent near infrared spectrum intensities. We start with an unsupervised PCA. Since PCA is unsupervised, it doesn't matter that many of the observations in`NIRspectra.xls` have missing Viscosity values.

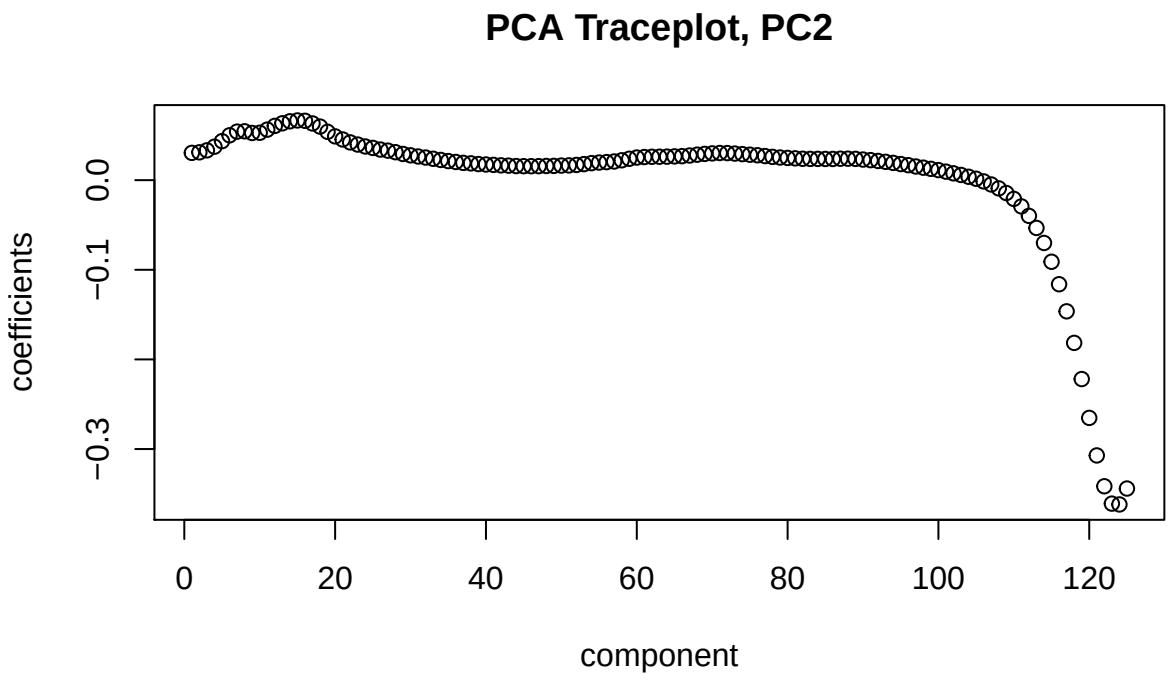
```
## [1] "Percentage of variance explained by first PC:  92.9"
```

```
## [1] "Percentage of variance explained by second PC:  6.7"
```


Largest contributions to variance
Variances
0.000
0.004
0.008
0.012
Principal components


PCA scoreplot of PC2 against PC1, unequal axes
PC2
−0.05
0.00
0.05
0.10
0.0
0.5
1.0
1.5
PC1





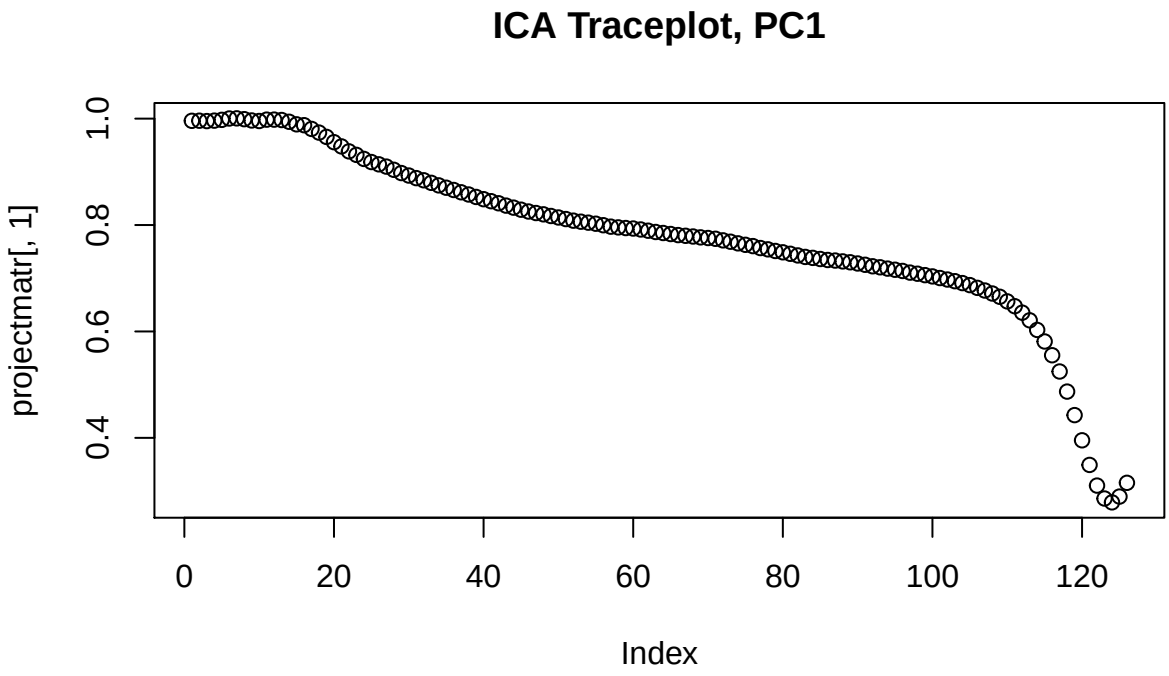
It appears only two Principal Components (PCs) can explain more than 99% of the variance of spectrum data. From the scoreplot we see that in the direction of PC1 the data is assymetrically distributed toward positive PC1 values with a few extreme outliers. Most of the data points are located in an oval cluster around the origin, though. The traceplots tell us the coefficients of how much a unit vector in each original component contributes to the PCs. It is seen that for PC1 every component contributes in equal magnitude while for

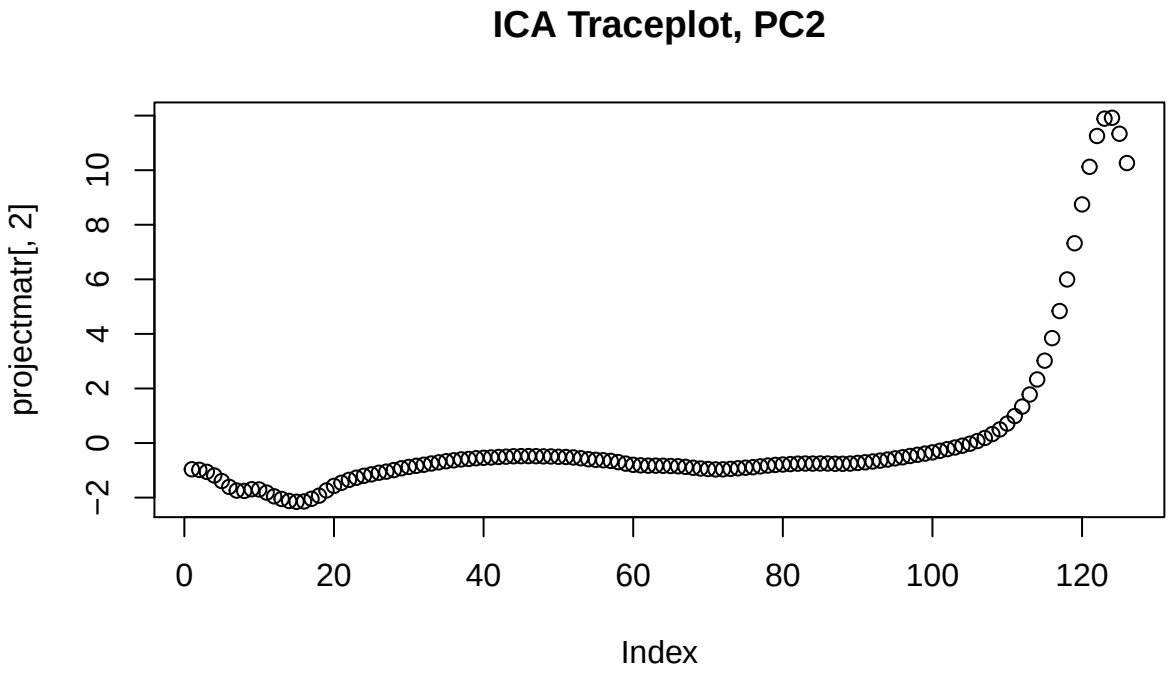
PC2 only a minority of components make most of the contribution.

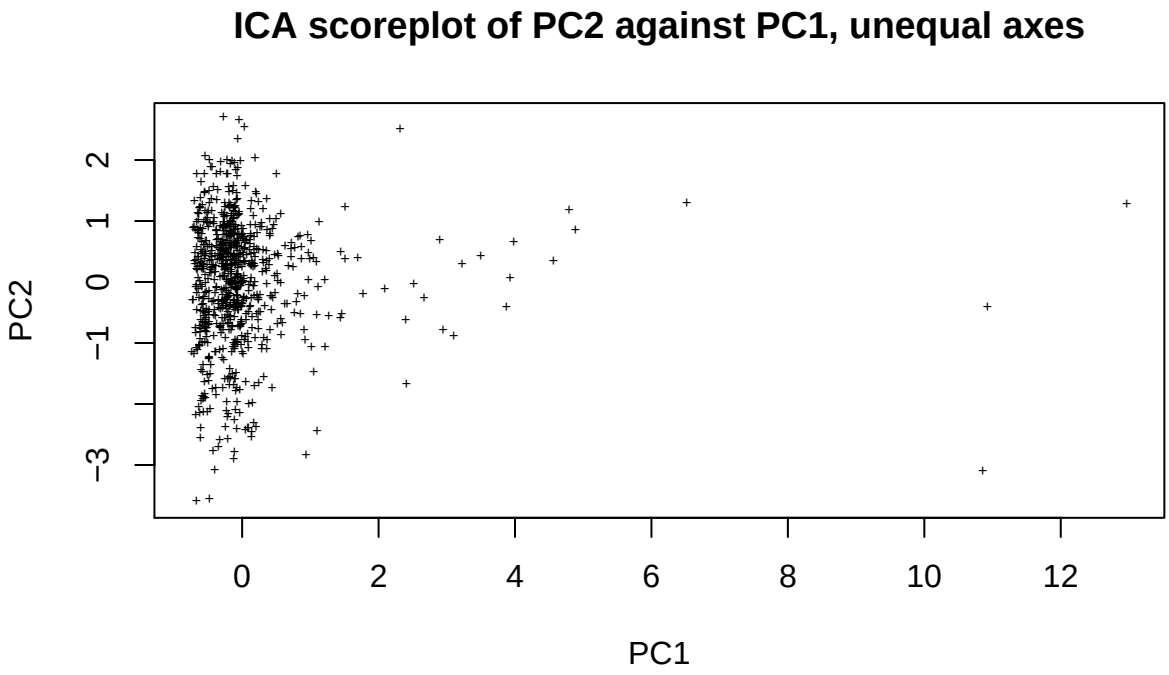
Let us compare these results with ICA results. the function `fastICA` in package of same name returns, among other things, matrices S, the (most) independent components,K, a matrix which projects original data onto the PCs from PCA and W, the unmixing matrix in such a form that they satisfy

$$XKW = S$$

where X is the original data. Unmixing matrix W is orthogonal and represents rotating the axes so that they coincide with the most independent directions. We find traceplots of columns of matrix $W' = KW$ and the scoreplot of the independent components.



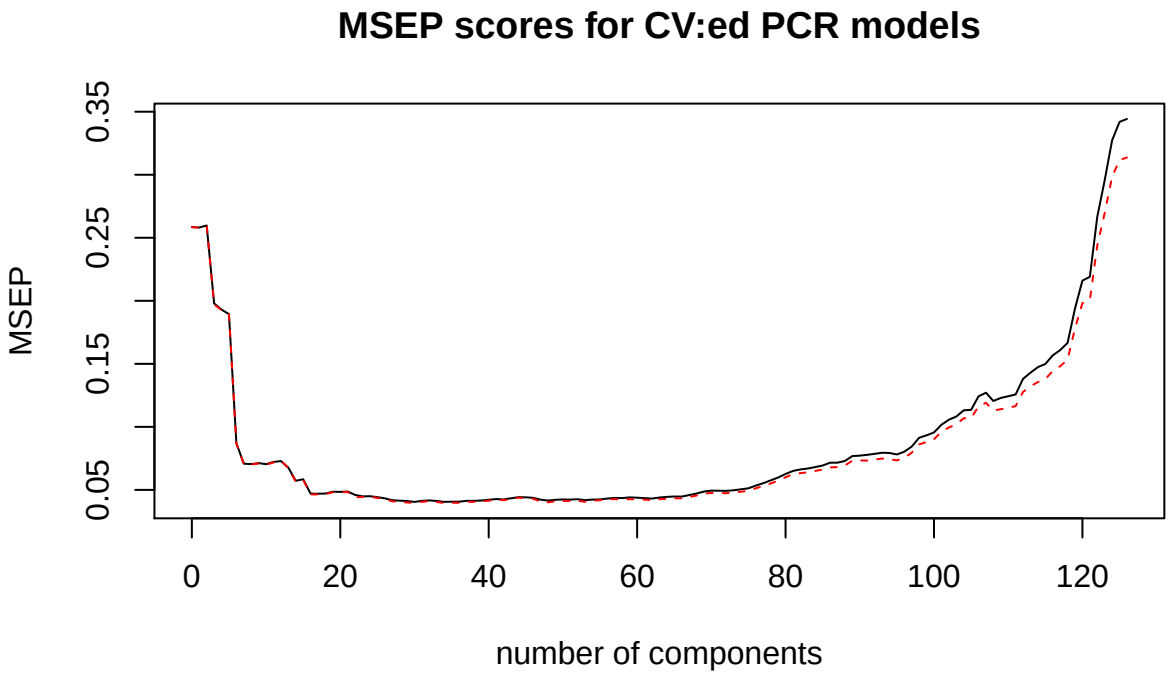




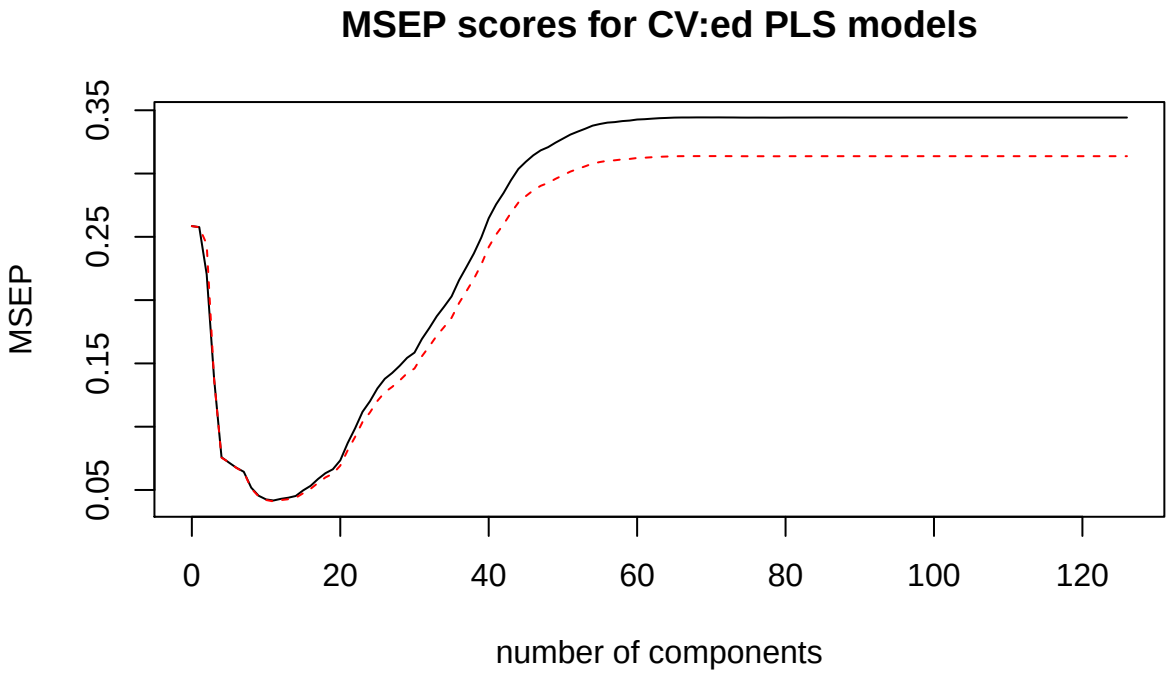
These ICA trace plots tell us a similar story regarding which components contribute to PC1 and PC2. The ICA scoreplot also displays the same features as the scoreplot for PCA.

Let us now divide data into equally large train and test subsets and,using train data, fit PCR and PLS models and pick the best models using crossvalidation with MSEP as cv-score. We then evaluate the best model by finding MSE between model predictions and test data. Only observations with non-missing Viscosity values

are used to calculate MSE.




```
## [1] "MSE for test data for PCR model with 20 components: 0.0607"
```



```
## [1] "MSE for test data for PLS model with 10 components: 0.0649"
```

We see that the best model of PLS has half (10) as many components as the best PCR model has (20). The best models were selected with consideration of number of components involved, as well as based on CV-score. The performance on test data are about equal for the two models.

Appendix

R code

```
library(tree)
library(boot)
data <-read.csv2("C:/Users/Dator/Documents/R_HW/ML-Lab-2/data/State.csv")
data <- data[order(data$MET),]
plot(data$MET,data$EX, main=c("per capita State and local expenditures ($)",
                              "vs percentage of people living in metropolitan areas"),
ylab="$ expenditures per capita",
xlab="percentage of people in metropolitan areas",pch="+")

regtree <- tree(EX ~ MET,data, control = tree.control(48,minsize=2))
set.seed(12345)
cvresult <- cv.tree(regtree)
# best result at size = 3
plot(cvresult$size,cvresult$dev, type="b", main="Deviance of fitted tree against tree size",
     xlab="Size",ylab="Deviance")

bestregtree <- prune.tree(regtree,best=3)
plot(bestregtree)
text(bestregtree,pretty=1)

pred <- predict(bestregtree,data)

resid <- data$EX - pred
#residuals look uniform

hist(resid,main=c("Residuals of best regression tree prediction", "of per capita expenditure"),
xlab="residual ($)")
plot(data$MET,pred,col="red",ylim=c(240,400),main="Regression tree fitted values and original values",
     ylab="$ expenditures per capita",
     xlab="percentage of people in metropolitan areas")
points(data$MET,data$EX,pch="+")
legend(x=20,y=400,c("original values","fitted values"),
       pch=c("+","o"),
       col=c("black","red"))
f <- function(datainp,ind){
  data1 <- datainp[ind,]
  res <- tree(EX ~ MET,data1, control = tree.control(dim(data1)[1],minsize=2))
  bestrestree <- prune.tree(res,best=3)
  predictions <- predict(bestrestree, newdata=data)
  return(predictions)
}
```

```
set.seed(12345)
bootobj1 <- boot(data,f, R=1000)
confintvs <- envelope(bootobj1)
#plot(bootobj1)

plot(data$MET,pred,col="red",ylim=c(150,500),
     main=c("Regression tree fitted values and original values",
            "and 95% non-parametric bootstrap confidence bands"),
     ylab="$ expenditures per capita",
     xlab="percentage of people in metropolitan areas")
points(data$MET,data$EX,pch="+")
points(data$MET,confintvs$point[2,], type="l", col="blue")
points(data$MET,confintvs$point[1,], type="l", col="blue")

legend(x=20,y=500,c("original values","fitted values","confidence intervals"),
       pch=c("+","o",NA),lwd=1,lty=c(NA,NA,1),
       col=c("black","red","blue"))
rng <- function(data2,mle){
  data1 = data.frame(MET = data2$MET, EX = data2$EX)
  n = length(data2$EX)
  data1$EX = rnorm(n,predict(mle, newdata=data1),
                   sd(data$EX-predict(mle, newdata=data1)))
  return(data1)
}

f1 = function(data1){
  res <- tree(EX ~ MET,data1, control = tree.control(dim(data1)[1],minsize=2))
  bestrestree <- prune.tree(res,best=3)
  expenditures <- predict(bestrestree, newdata=data)
  return(expenditures)
}

f2 = function(data1){
  res <- tree(EX ~ MET,data1, control = tree.control(dim(data1)[1],minsize=2))
  bestrestree <- prune.tree(res,best=3)
  expenditures <- rnorm(dim(data)[1],predict(bestrestree, newdata=data),
                        sd(resid))
  return(expenditures)
}

set.seed(12345)
bootobj2 <- boot(data, statistic= f1,R=1000,
                 mle=bestregtree,ran.gen=rng,sim="parametric")
set.seed(12345)
bootobj3 <- boot(data,statistic= f2,R=1000,
                 mle=bestregtree,ran.gen=rng,sim="parametric")
#plot(bootobj2)
confintvs2 <- envelope(bootobj2)
confintvs3 <- envelope(bootobj3)
plot(data$MET,pred,col="red",ylim=c(150,550),
     main=c("Regression tree fitted values and original values",
            "95% parametric bootstrap confidence and prediction bands"),
     ylab="$ expenditures per capita",
```

```
     xlab="percentage of people in metropolitan areas")
points(data$MET,data$EX,pch="+")
points(data$MET,confintvs2$point[2,], type="l", col="blue")
points(data$MET,confintvs2$point[1,], type="l", col="blue")
points(data$MET,confintvs3$point[2,], type="l", col="pink")
points(data$MET,confintvs3$point[1,], type="l", col="pink")
legend(x=20,y=550,c("original values","fitted values",
                    "confidence bands","prediction bands"),
       pch=c("+","o",NA,NA),lwd=1,lty=c(NA,NA,1,1),
       col=c("black","red","blue","pink"))

library(XLConnect)
library(fastICA)
library(pls)
#FROM THIS FILE LOCATION, EXCEL FILES SHOULD BE FOUND IN A SUBFOLDER IN THIS FILE LOCATION CALLED DATA
wb = loadWorkbook("C:/Users/Dator/Documents/R_HW/ML-Lab-2/data/NIRSpectra.xls")
data2 = readWorksheet(wb, sheet = "NIRSpectra", header = TRUE)
data2 <- data2[,-1]
data2 <- data2[,-length(data2)]
#data2 <-data2[complete.cases(data2),]
#head(data2)
set.seed(12345)
res=prcomp(data2)
#resul <- princomp(data2,scale=TRUE)
lambda=res$sdev^2
percentage <- lambda/sum(lambda)*100
#eigenvalues
# lambda
#proportion of variation
paste("Percentage of variance explained by first PC: ",signif(percentage[1],3))
paste("Percentage of variance explained by second PC: ",signif(percentage[2],2))
# two pcs are needed to explain 99% of
screeplot(res,main="Largest contributions to variance",xlab="Principal components")
plot(res$x[,1],res$x[,2],pch="+",
     main="PCA scoreplot of PC2 against PC1, unequal axes",
     xlab="PC1",ylab="PC2",cex=0.5)
# biplot(res)
# U=res$rotation
# head(U)
#U=loadings(resul)
plot(res$rotation[-nrow(res$rotation),1],
     main="PCA Traceplot, PC1",
     xlab="component",ylab="coefficients")
plot(res$rotation[-nrow(res$rotation),2],
     main="PCA Traceplot, PC2",
     xlab="component",ylab="coefficients")
# plot(U[,1], main="PCA Traceplot, PC1")
# plot(U[,2],main="PCA Traceplot, PC2")
set.seed(12345)
res2 <- fastICA(data2,2)
#res2$W is the unmixing matrix "X"W = S where S contain independent components
#res2$K is a pre-whitening matrix which projects data onto first 2 principal components XKW = S
projectmatr <- res2$K %*% res2$W
```

```r
#plot(res2$K[,1], main="ICA K-matrix Traceplot, PC1")
#plot(res2$K[,2], main="ICA K-matrix Traceplot, PC2")
#^^These are same as for PCA
plot(projectmatr[,1], main="ICA Traceplot, PC1")
plot(projectmatr[,2],main="ICA Traceplot, PC2")
plot(res2$S,pch="+",
     main="ICA scoreplot of PC2 against PC1, unequal axes",
     xlab="PC1",ylab="PC2",cex=0.5)

data2 <- readWorksheet(wb, sheet = "NIRSpectra", header = TRUE)
data2 <- data2[,-1]
#data2 <-data2[complete.cases(data2),]
set.seed(12345)
ind <- sample(1:784,392)
train <- data2[ind,]
test <- data2[-ind,]
n <- dim(test[complete.cases(test),])[1]

set.seed(12345)
pcr.fit=pcr(Viscosity~., data=train, validation="CV")
validationplot(pcr.fit,val.type="MSEP",main="MSEP scores for CV:ed PCR models")
pcrmodel=pcr(Viscosity~., 20,data=train, validation="none")
pcr.pred <- predict(pcrmodel, newdata=test[complete.cases(test),], ncomp = 20)
pcr.mse <- 1/n * sum((test[complete.cases(test),]$Viscosity - pcr.pred)^2, na.rm=TRUE)
paste("MSE for test data for PCR model with 20 components:", signif(pcr.mse,3))

set.seed(12345)
plsr.fit <- plsr(Viscosity~., data=train, validation="CV")
validationplot(plsr.fit,val.type="MSEP",main="MSEP scores for CV:ed PLS models")
plsmodel=plsr(Viscosity~., 10,data=train, validation="none")
pls.pred <- predict(plsmodel, newdata=test[complete.cases(test),], ncomp = 10)
pls.mse <- 1/n * sum((test[complete.cases(test),]$Viscosity - pls.pred)^2, na.rm=TRUE)
paste("MSE for test data for PLS model with 10 components:", signif(pls.mse,3))
## NA
```