TDDE01: Lab 4

A report by

Adam Nyberg
adany869

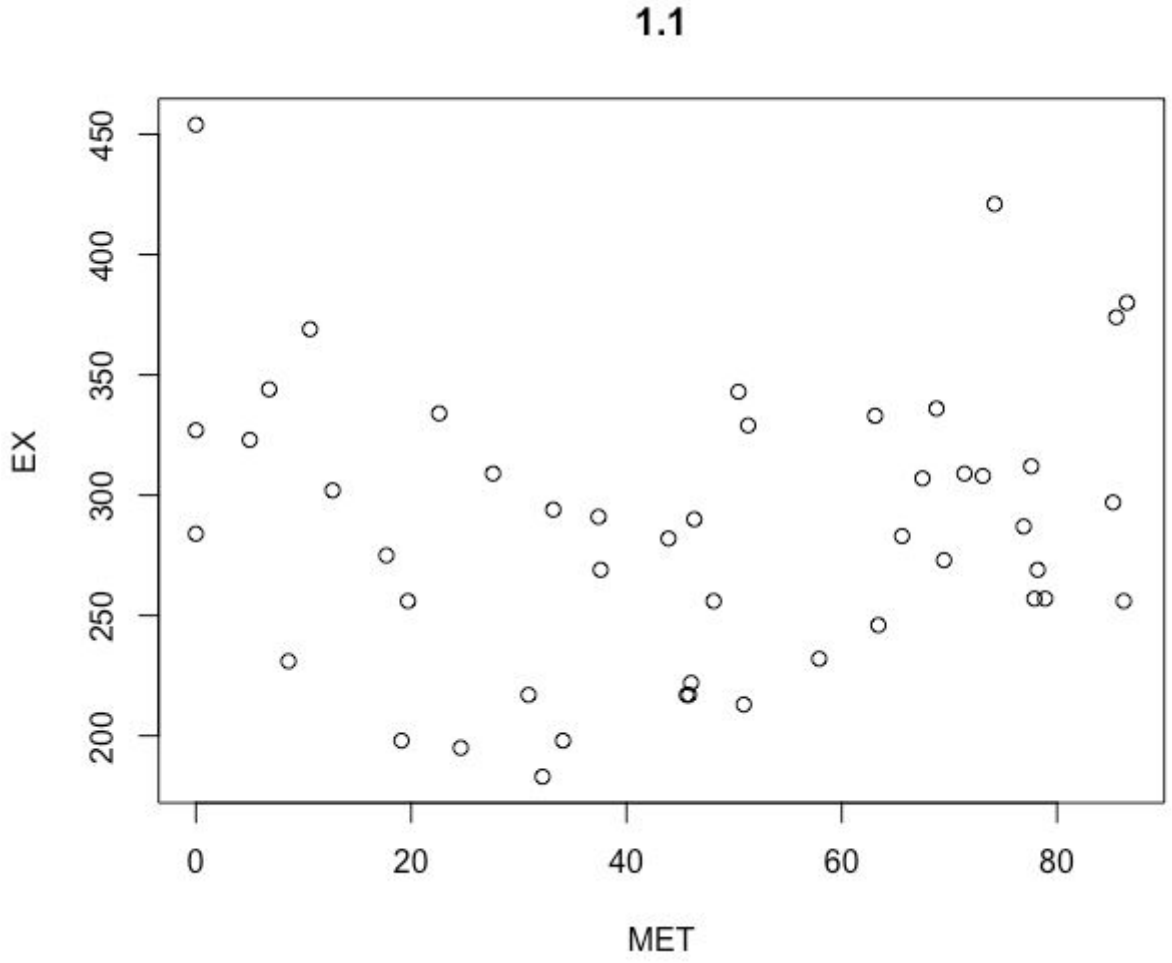
Assignment 1

This assignment was about working with per capita data state and local public expenditures and associated state demographic and economic characteristics with the following variables.

- MET: Percentage of population living in standard metropolitan areas
- EX: Per capita state and local public expenditures ($)

1.1

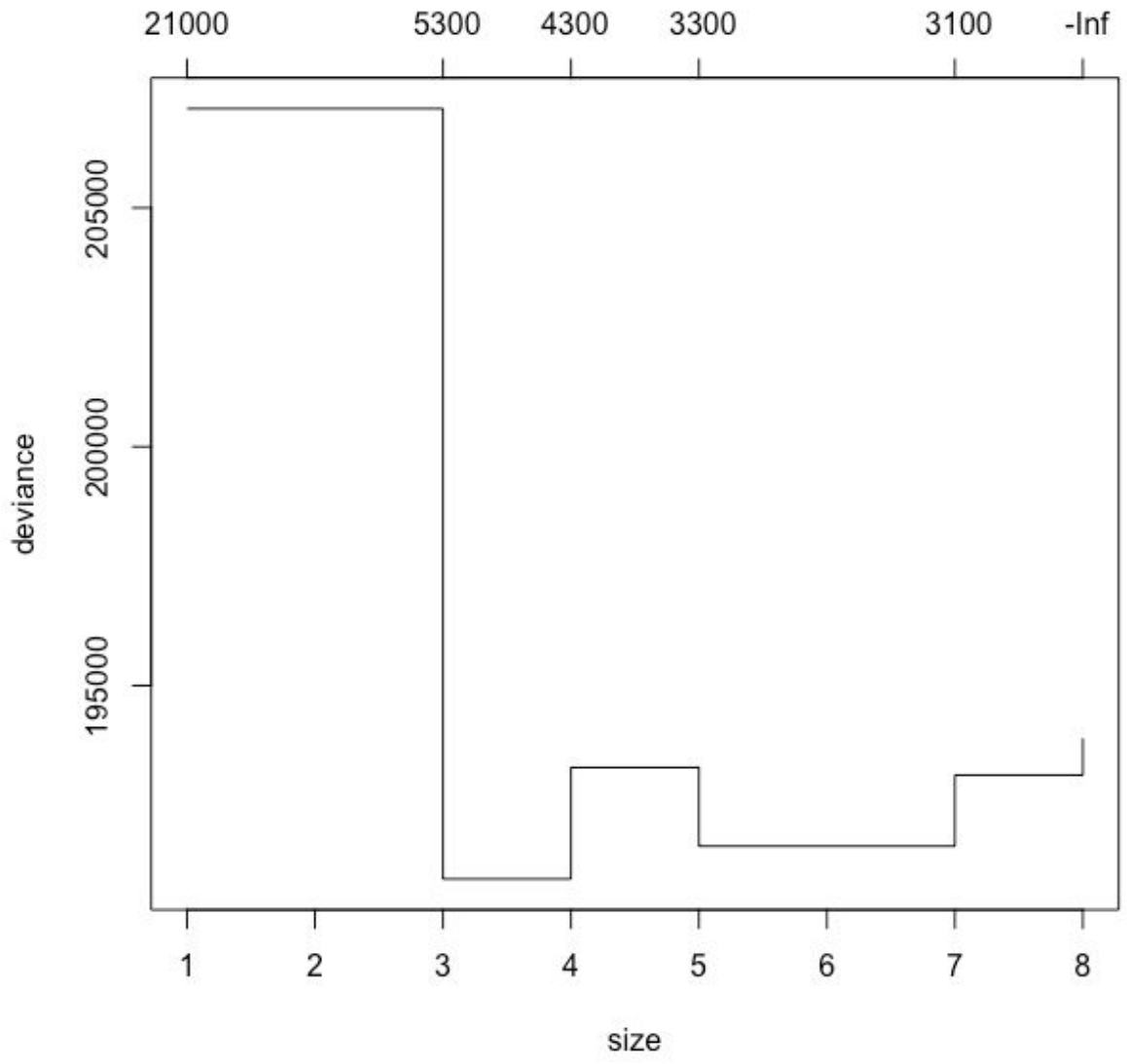
In this task I reordered the data with respect to the increase of MET and plotted EX versus MET shown below.

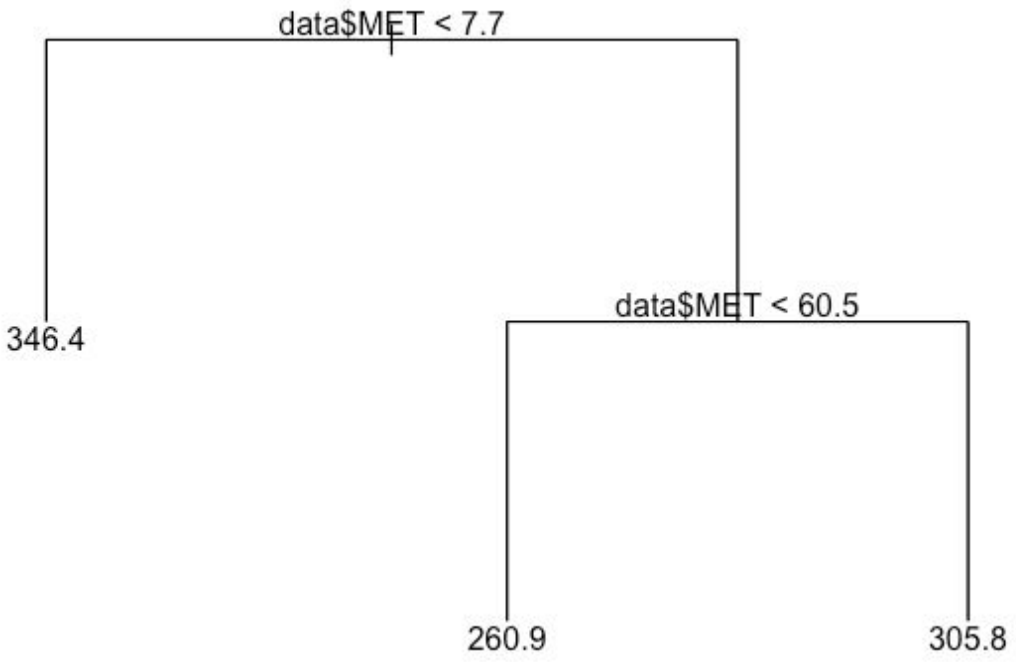


It is hard to find any structure in this data so a linear does not seem to fit, for that reason I think that a tree model would be good.

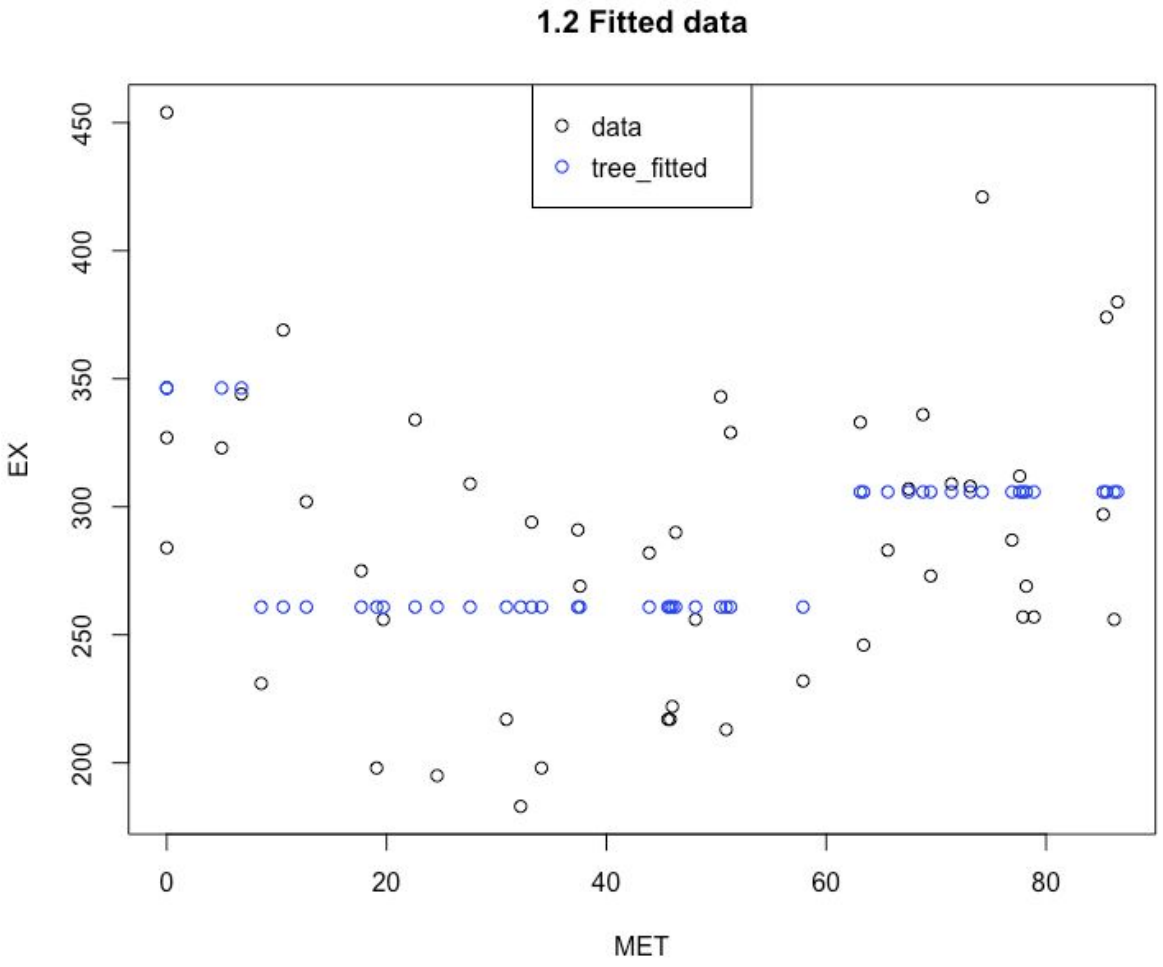
1.2

The fitted tree model with target EX and feature MET and where the number of leaves was selected by cross-validation is shown below.

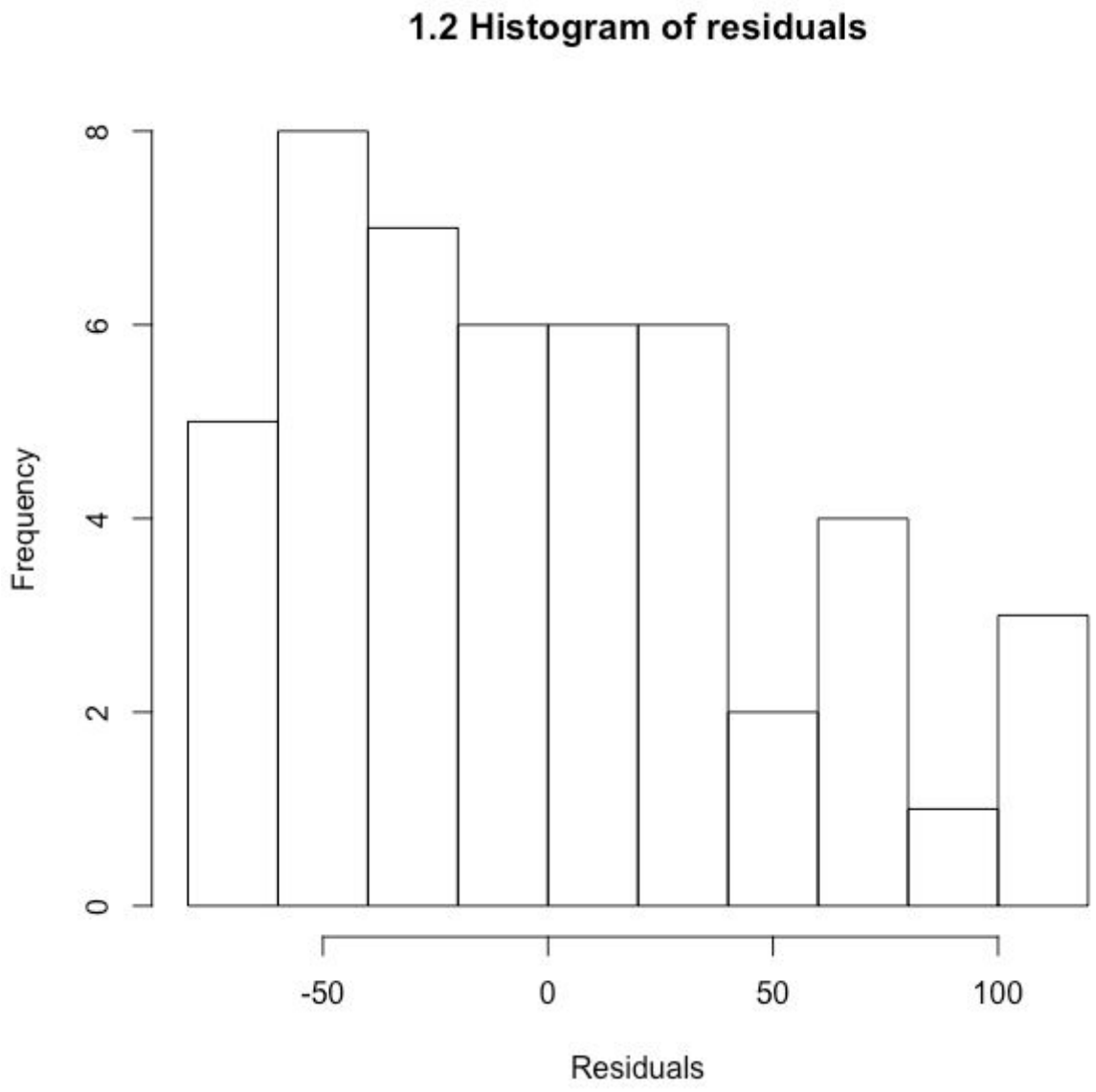

21000
5300
4300
3300
3100
-Inf
205000
200000
195000
deviance
1
2
3
4
5
6
7
8
size



Below is the fitted data from that tree model. We can see that the quality of fit is in line with what you'd expect from a tree model with three leaves.

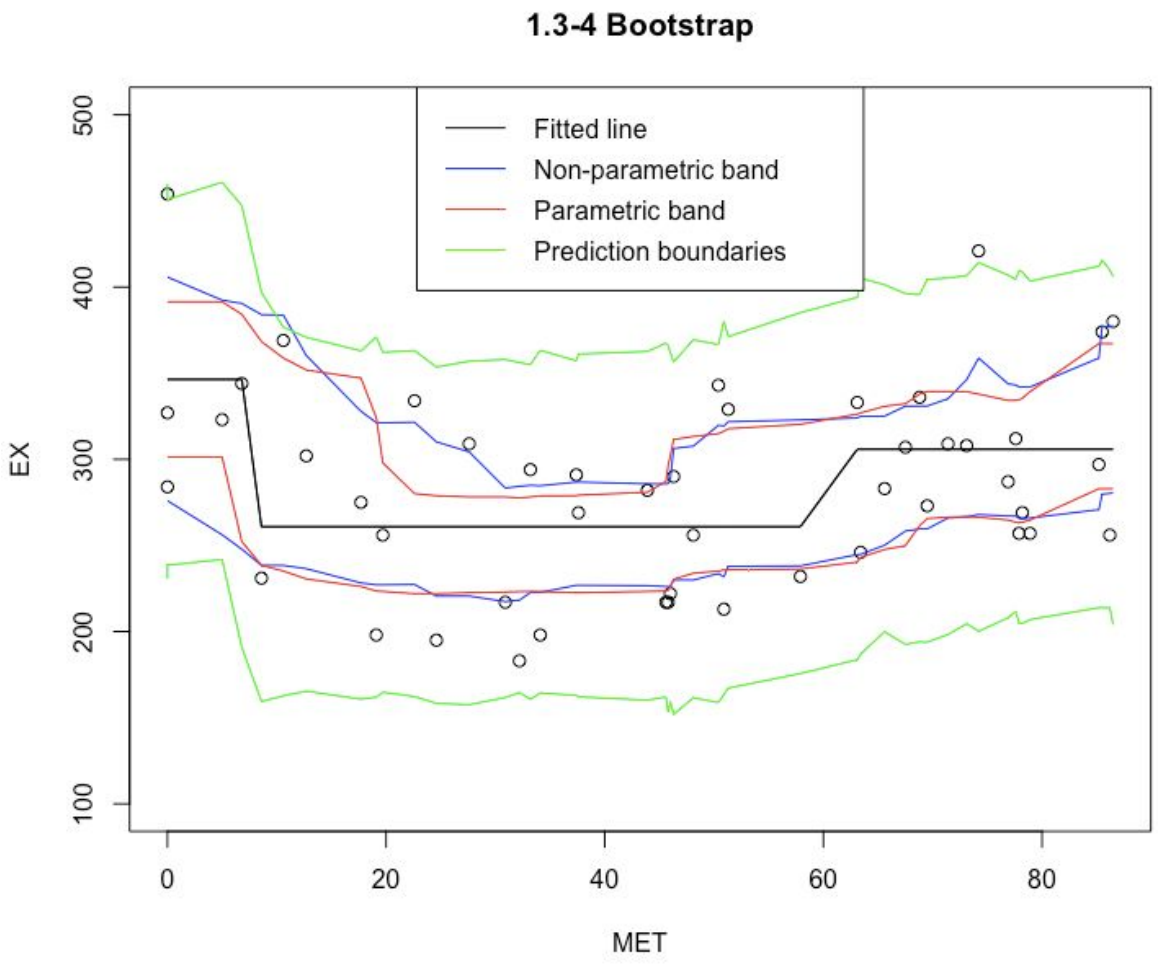


Below is the histogram of residuals. The histogram shows us that it's more likely to underestimate with a small error but when it's overestimates the error is usally larger. We can here see that the distrubution looks a little right skewed.



1.3

In this task we computed the 95% confidence bands for the regression tree by using a non-parametric bootstrap and the plot is shown below. The bands look pretty bumpy and thats because the bootstrap uses a subset of the data and tries to capture 95% of the computed distrubutions. The results from 1.2 is reliable because the fitted model is almost centered within the confidence bands.



1.4

In this task we used a parametric bootstrap instead of a non-parametric in 1.3. Parametric bootstrap is smoother vecause it uses the model instead of the original data to generate the distrubutions.

As we can see in the plot in 1.3 the fitted line is reliable with parametric bootstrap as well. Less than 5% of the data are outside of the prediction bands. That's because the prediction bands will cover the data and future observations.

1.5

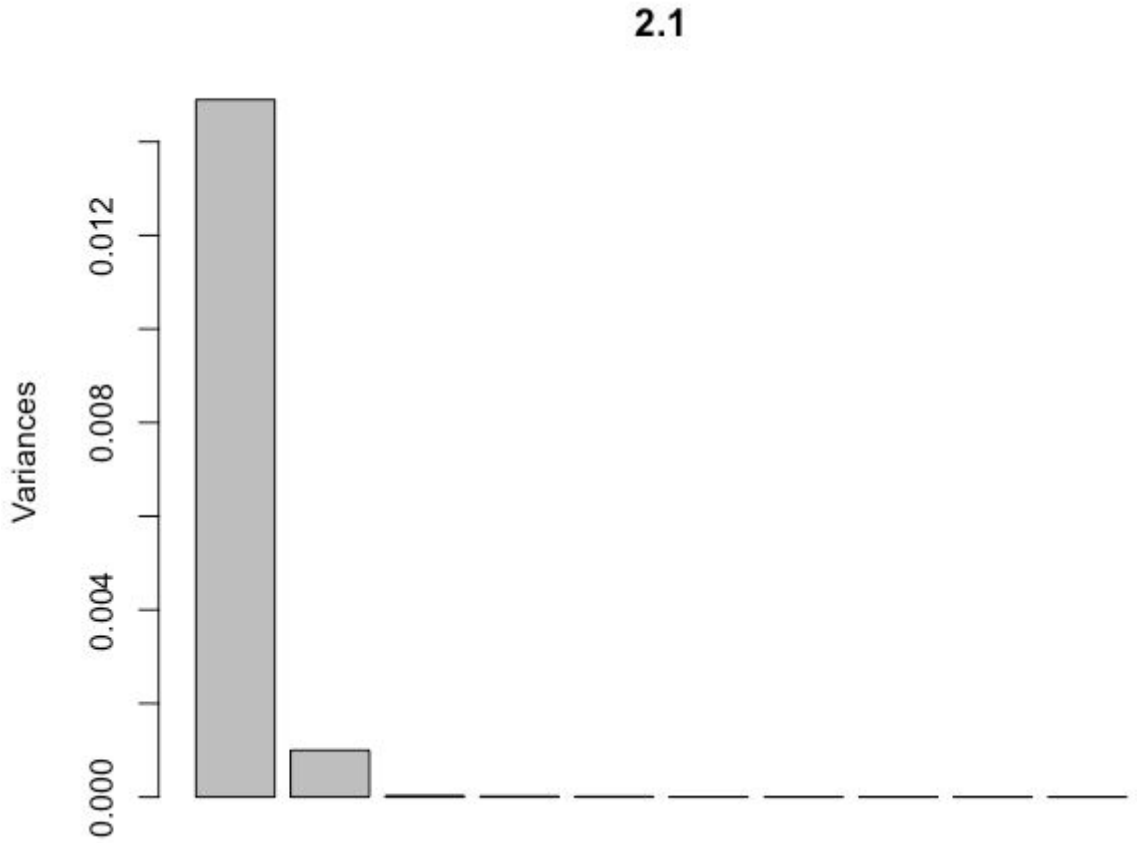
When looking at the histogram we can see that the residuals does not look like a normal distrurbution and therefore a parametric bootstrap would be perferred.

Assignment 2

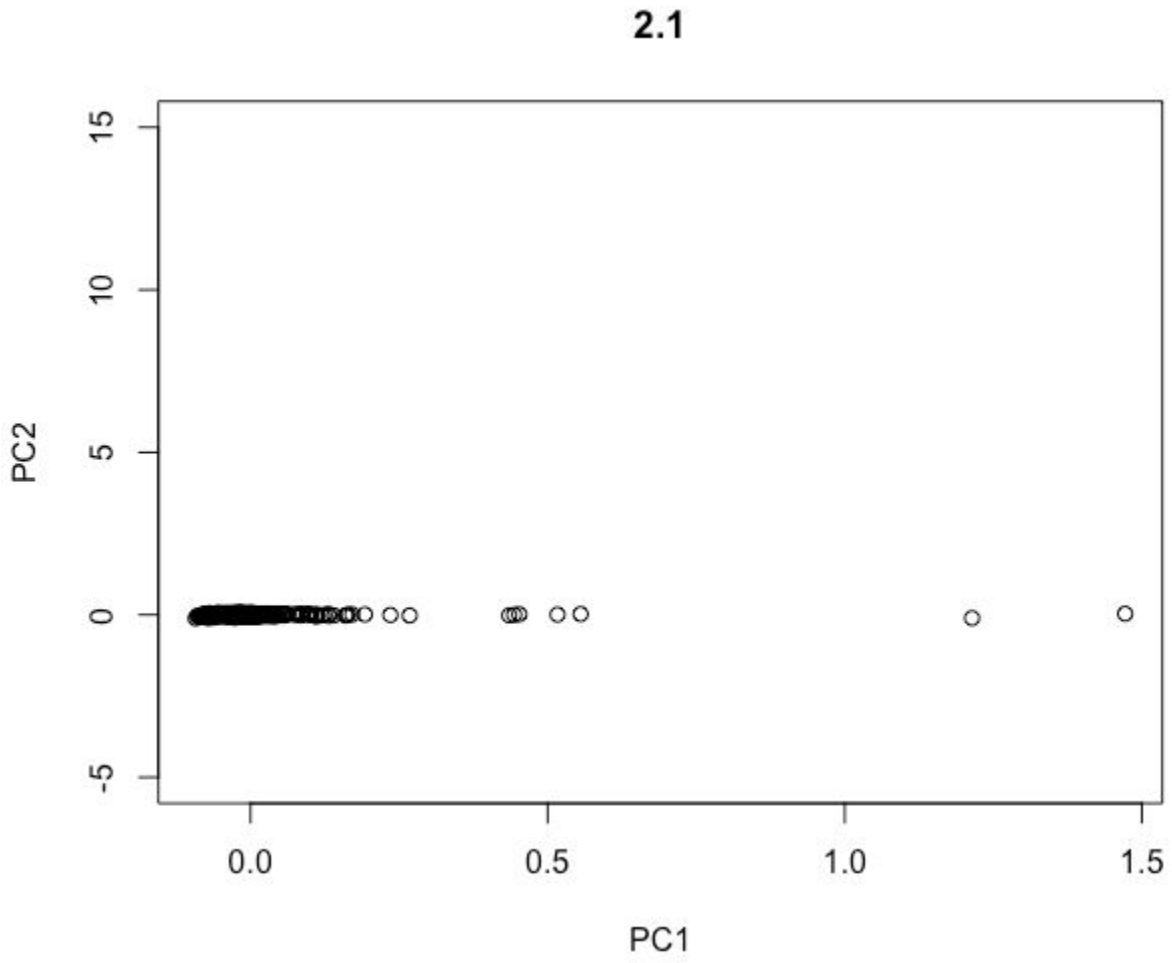
The data file NIRspectra.xls contains near-infrared spectra and viscosity levels for a collection of diesel fuels. My task is to investigate how the measured spectra can be used to predict the viscosity.

2.1

In this task we conducted a standard PCA and made a plot showing the variance captured by each PC, shown below.



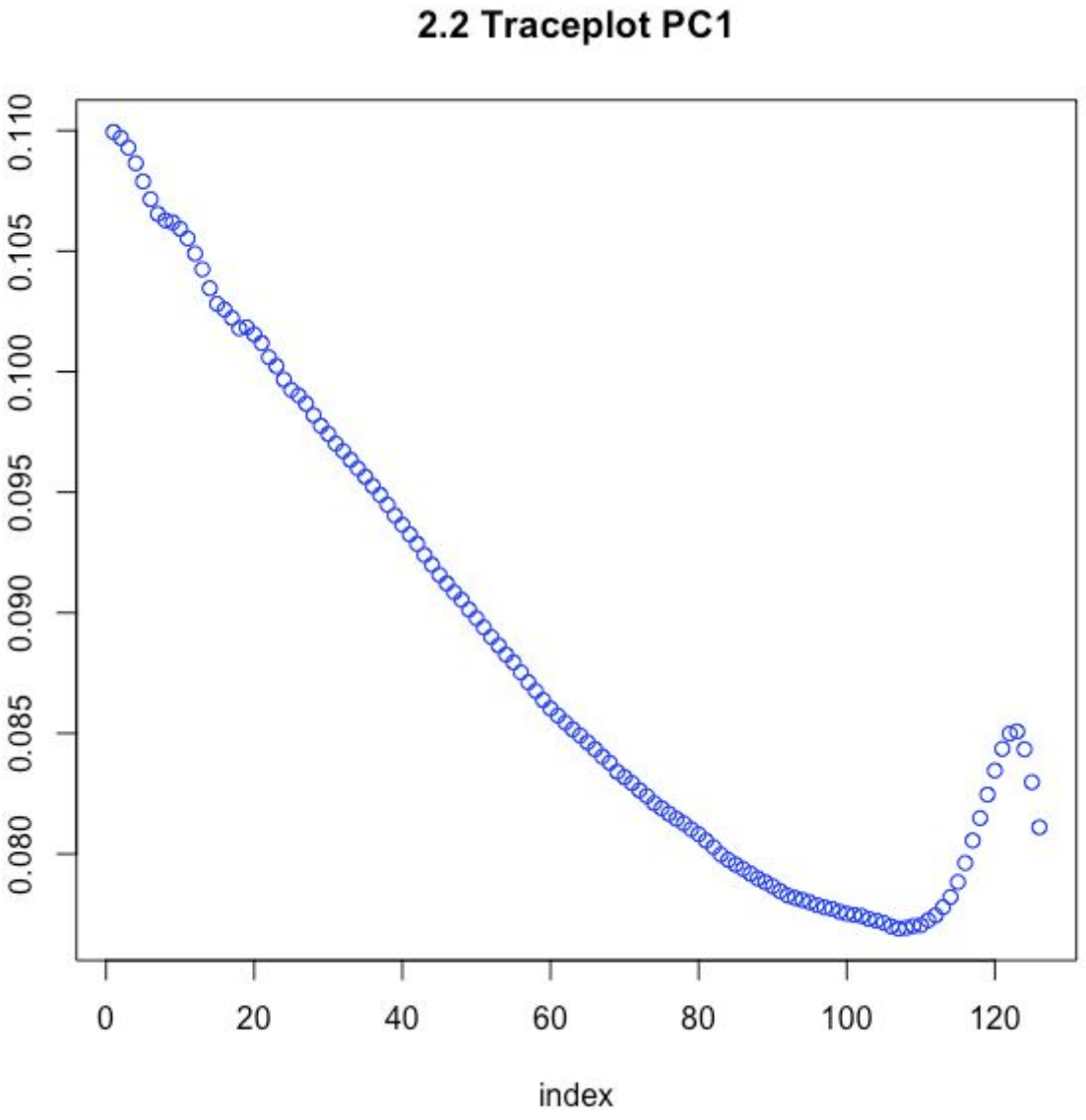
The plot clearly tells us that we only need to use two PCs to capture 99% of all variance. Below is the plot of the scores in the coordinates (PC1, PC2).

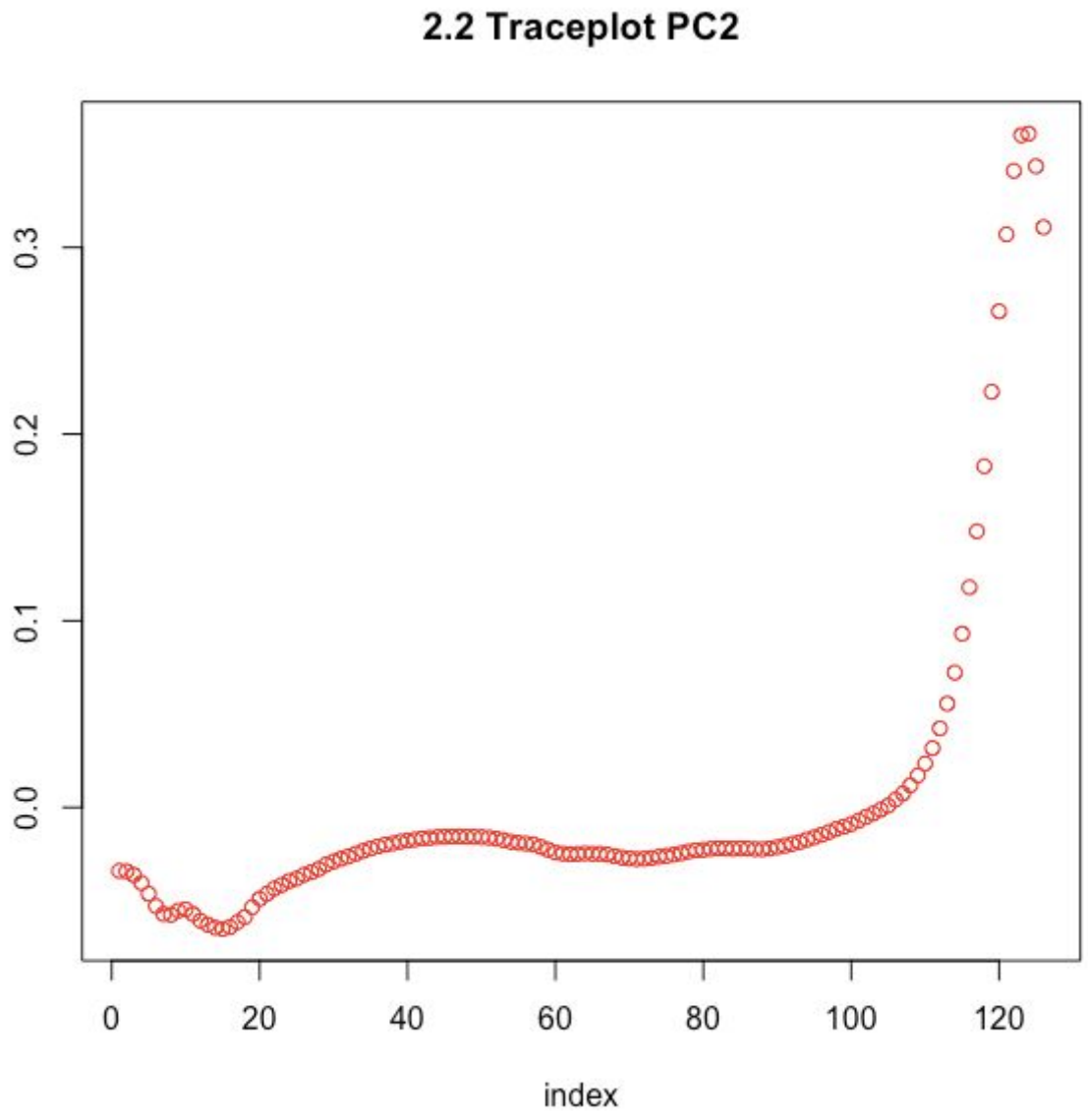


As we can see there are some unusual diesel fuels to the right.

2.2

In this task we plotted the trace plot of PC1 and PC2. PC1 is described by most features but PC2 have a few features around index 120 that explains most of it.


2.2 Traceplot PC1
0.080
0.085
0.090
0.095
0.100
0.105
0.110
0
20
40
60
80
100
120
index



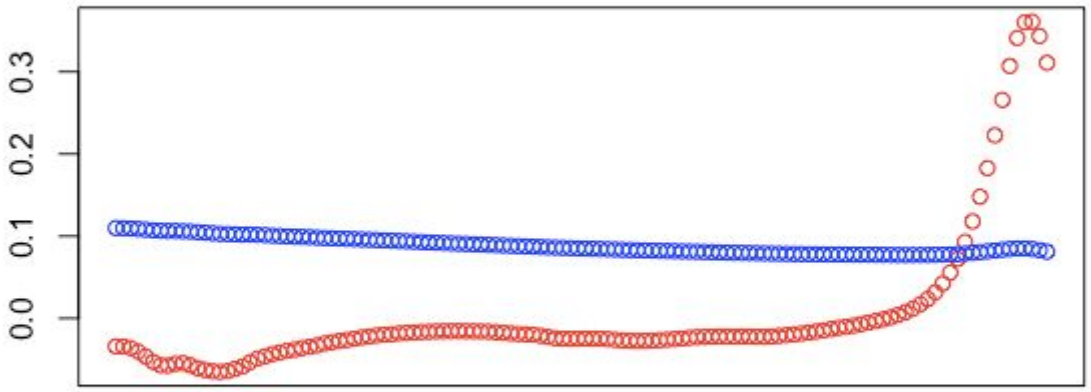
2.3

Using fastICA, an Independent Component analysis was performed.

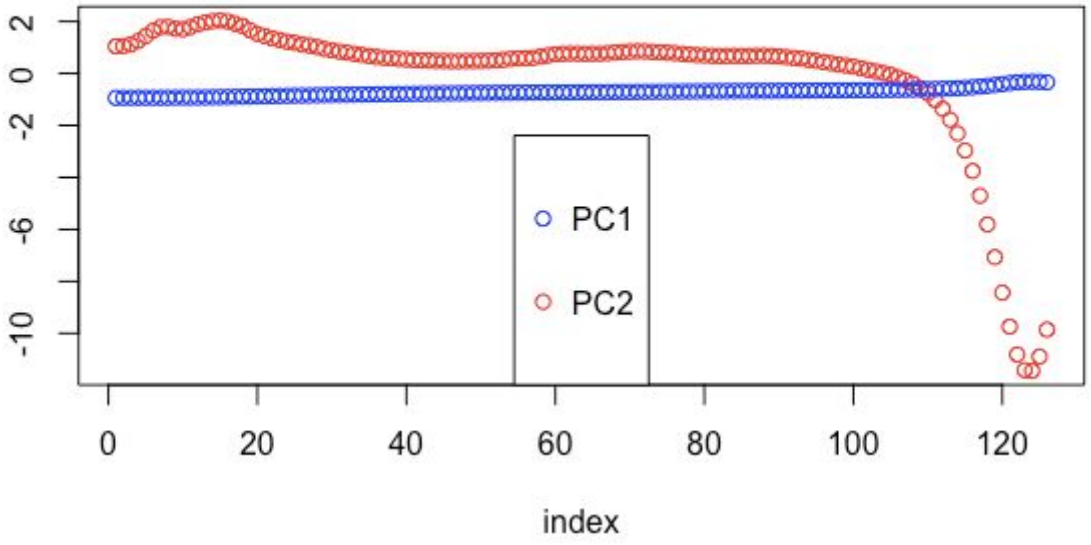
a)

As we can see in the plot below ICA looks like it is laterally reversed from PCA in 2.2. Matrix W' represent an un-mixing matrix projected onto the principal components, where un-mixing mean to separating the data into independent components.

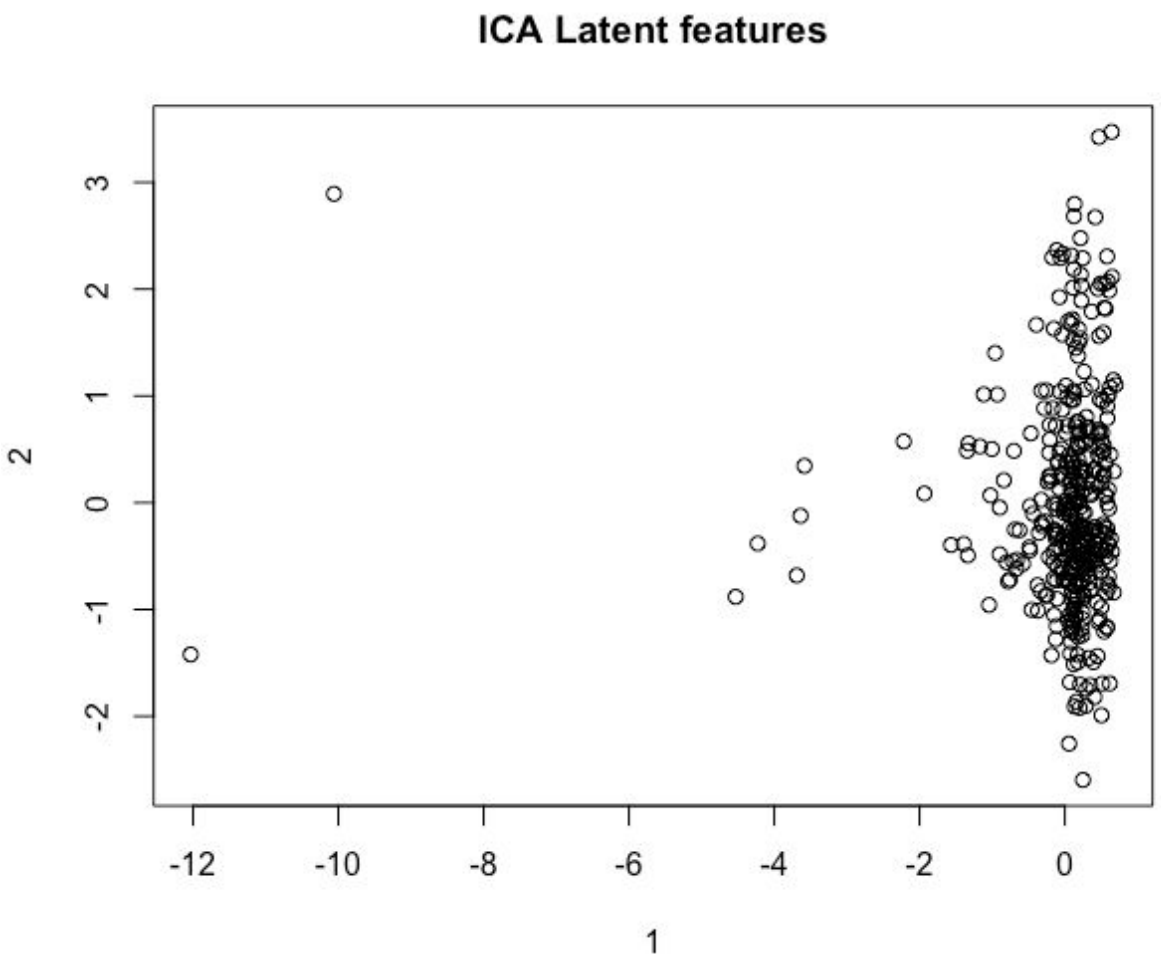
2.2 Traceplot PCA PC1 & PC2



2.3 Traceplot ICA PC1 & PC2



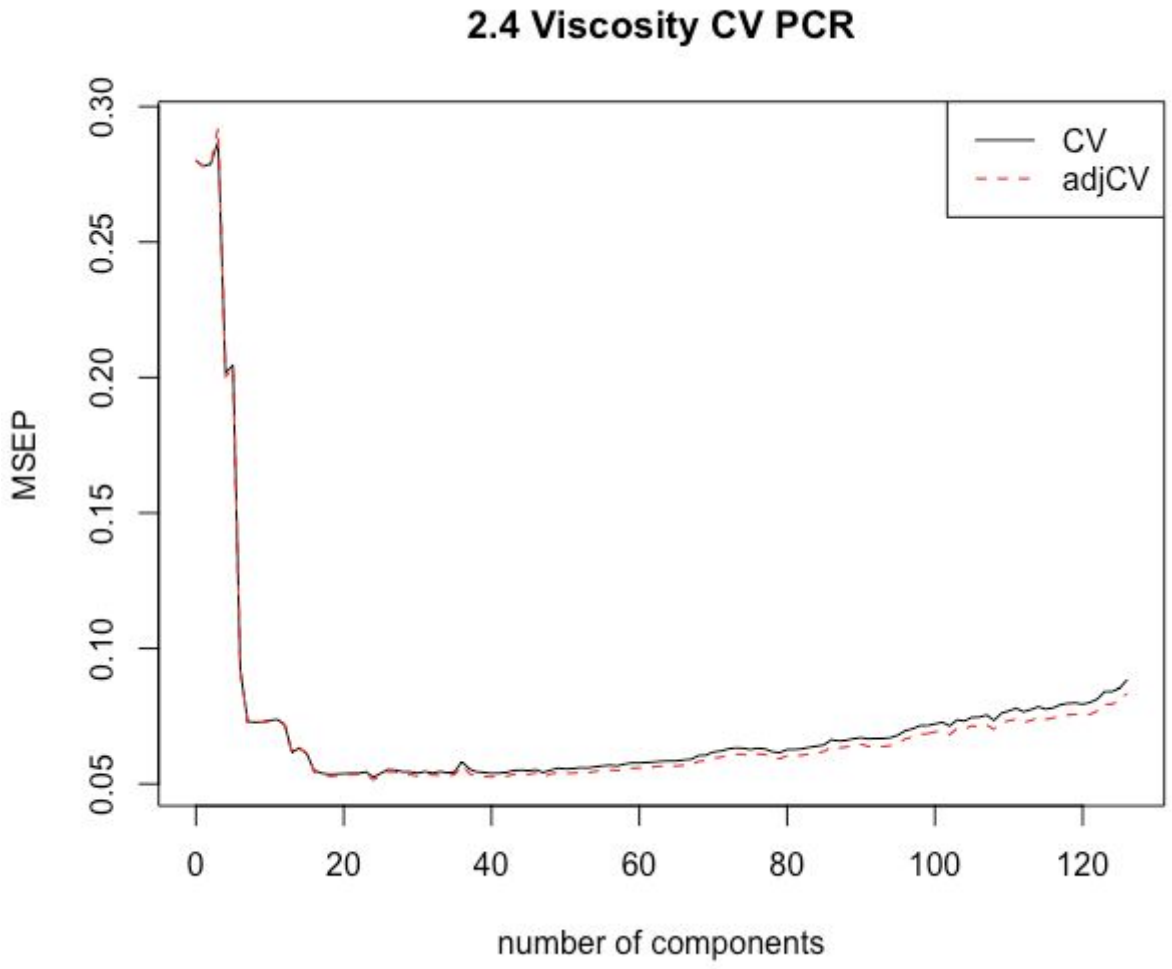
b)



When plottning the two first latent features and comparing it with the plot from 2.1 we see that they look the same except reversed.

2.4

In this task we used PCR to decide how many components to select.



As we can see it’s reasonable to select 20 components as that minimizes the MSEP but without increasing complexity of the model.

```
setwd("~/code/skola/tdde01/adam")

library(readxl)
library(tree)
library(boot)
set.seed(12345)

unsorted_data = read_excel("lab4/State.xls")

# ASSIGNMENT 1.1
data = unsorted_data[with(unsorted_data, order(unsorted_data$MET)), ] # HAXXXOR
n = nrow(data)

plot(data$MET, data$EX, ylab="EX", xlab="MET", main="1.1")#, ylim = range(150,500), xlim
= range(0,350))


# ASSIGNMENT 1.2

tree_m = tree(data$EX~data$MET, data=data, control=tree.control(n, minsize=8))

tree_m_cv <- cv.tree(object=tree_m)
k = tree_m_cv$size[which(tree_m_cv$dev==min(tree_m_cv$dev))]
tree_m_pruned <- prune.tree(tree_m, best=k)

plot(tree_m_pruned, main="1.2 Tree model")
text(tree_m_pruned, main="1.2 Tree model")

tree_pred_ex = predict(tree_m_pruned, data)

plot(data$MET, data$EX, ylab="EX", xlab="MET", main="1.2 Fitted data")
points(data$MET, tree_pred_ex, ylab="EX", xlab="MET", col="blue")
legend("top", pch=c(1,1), col=c("black", "blue"), legend = c("data", "tree_fitted"))

tree_m_residuals = data$EX-tree_pred_ex

hist(tree_m_residuals, main="1.2 Histogram of residuals", xlab="Residuals")

# ASSIGNMENT 1.3

# computing bootstrap samples
bootstrap=function(in_data, ind){
  data1=in_data[ind,]# extract bootstrap sample
  model=tree(EX~MET, data=data1, control=tree.control(n, minsize=8))
  model_pruned <- prune.tree(model, best=k)
```

```
  priceP=predict(model_pruned,newdata=data)
  return(priceP)
}

boot_res=boot(data, bootstrap, R=1000) #make bootstrap

env=envelope(boot_res, level=0.95) #compute non-parametric confidence bands

plot(data$MET, data$EX, pch=21, main="1.3-4 Bootstrap", xlab="MET", ylab="EX",
ylim=range(100,500))
points(data$MET, tree_pred_ex, type="l") #plot fitted line
#plot cofidence bands
points(data$MET, env$point[2,], type="l", col="blue")
points(data$MET, env$point[1,], type="l", col="blue")


# ASSIGNMENT 1.4

# Parametric confidence bands
rng=function(in_data, model) {
  data1=data.frame(EX=in_data$EX, MET=in_data$MET)
  n=length(in_data$EX)
  pred = predict(model, newdata=data1)
  data1$EX=rnorm(n, pred, sd(data$EX-pred))
  return(data1)
}

bootstrap_2=function(data1){
  model=tree(EX~MET, data=data1, control=tree.control(n, minsize=8))
  model_pruned <- prune.tree(model, best=k)
  priceP=predict(model_pruned, newdata=data)
  return(priceP)
}

boot_parametric_res=boot(data, statistic=bootstrap_2, R=1000, mle=tree_m, ran.gen=rng,
sim="parametric")

env_parametric=envelope(boot_parametric_res, level=0.95)

points(data$MET, tree_pred_ex, type="l") #plot fitted line
points(data$MET, env_parametric$point[2,], type="l", col="red")
points(data$MET, env_parametric$point[1,], type="l", col="red")


# Prediction boundary
```

```
bootstrap_3 = function(data1) {
  model = tree(EX~MET, data=data1, control = tree.control(nobs = nrow(data1), minsize =
8))
  model_pruned = prune.tree(model, best=3)
  pred = predict(model_pruned, data1)
  n = length(data1$EX)
  ndata = rnorm(n, pred, sd(resid(model_pruned)))
  return(ndata)
}

boot_prediction = boot(data, bootstrap_3, R=1000, mle=tree_m_pruned, ran.gen=rng,
sim="parametric")
boot_prediction_env = envelope(boot_prediction, level=0.95)

points(data$MET, boot_prediction_env$point[2,], type="l", col="green")
points(data$MET, boot_prediction_env$point[1,], type="l", col="green")


legend("top", lty=c(1,1), col=c("black", "blue", "red", "green"), legend = c("Fitted line",
"Non-parametric band", "Parametric band", "Prediction boundaries"))
```

```
setwd("~/code/skola/tdde01/adam")

library(readxl)
library(fastICA)
library(pls)

set.seed(12345)

data = read_excel("lab4/NIRSpectra.xls")

# ASSIGNMENT 2.1
data1 = data

data1$Viscosity = c()
res = prcomp(data1)

lambda = res$sdev^2
#eigenvalues lambda
#proportion of variation
sprintf("%2.3f", lambda/sum(lambda)*100)

screeplot(res, main="2.1")
plot(res$x[,1], res$x[,2], ylim=c(-5,15), main="2.1", xlab="PC1", ylab="PC2")

# ASSIGNMENT 2.2
U = res$rotation
plot(U[,2], main="2.2 Traceplot PCA", xlab="index", ylab="", col="red")
points(U[,1], col="blue")
legend("top", pch=c(1,1), col=c("red", "blue"), legend = c("PC2", "PC1"), xpd = TRUE)

# ASSIGNMENT 2.3
a <- fastICA(data1, 2, alg.typ="parallel", fun="logcosh", alpha=1, method="R",
row.norm=FALSE, maxit=200, tol=0.0001, verbose=TRUE)

Wp = a$K%*%a$W
plot(Wp[,1], main="2.3 Traceplot ICA", xlab="index", ylab="PC1", col="blue")
plot(Wp[,2], main="2.3 Traceplot ICA", xlab="index", ylab="PC2", col="red")


layout(matrix(c(1, 2), nrow = 2, ncol = 1))
plot(U[,2], main="2.2 Traceplot PCA PC1 & PC2", xaxt='n', xlab="", ylab="", col="red")
points(U[,1], col="blue")

plot(Wp[,2], main="2.3 Traceplot ICA PC1 & PC2", xlab="index", ylab="", col="red")
points(Wp[,1], col="blue")
legend("bottom", pch=c(1,1), col=c("blue", "red"), legend = c("PC1", "PC2"), xpd = TRUE)
```

```
layout(matrix(c(1), 1))

plot(a$S[,1], a$S[,2], main="ICA Latent features", xlab="1", ylab="2")


# ASSIGNMENT 2.4

PCR = pcr(Viscosity~., data=data, validation="CV")
validationplot(PCR, val.type="MSEP", legendpos = "topright", main="2.4 Viscosity CV PCR")
```