Introduction to Machine Learning - Lab 5

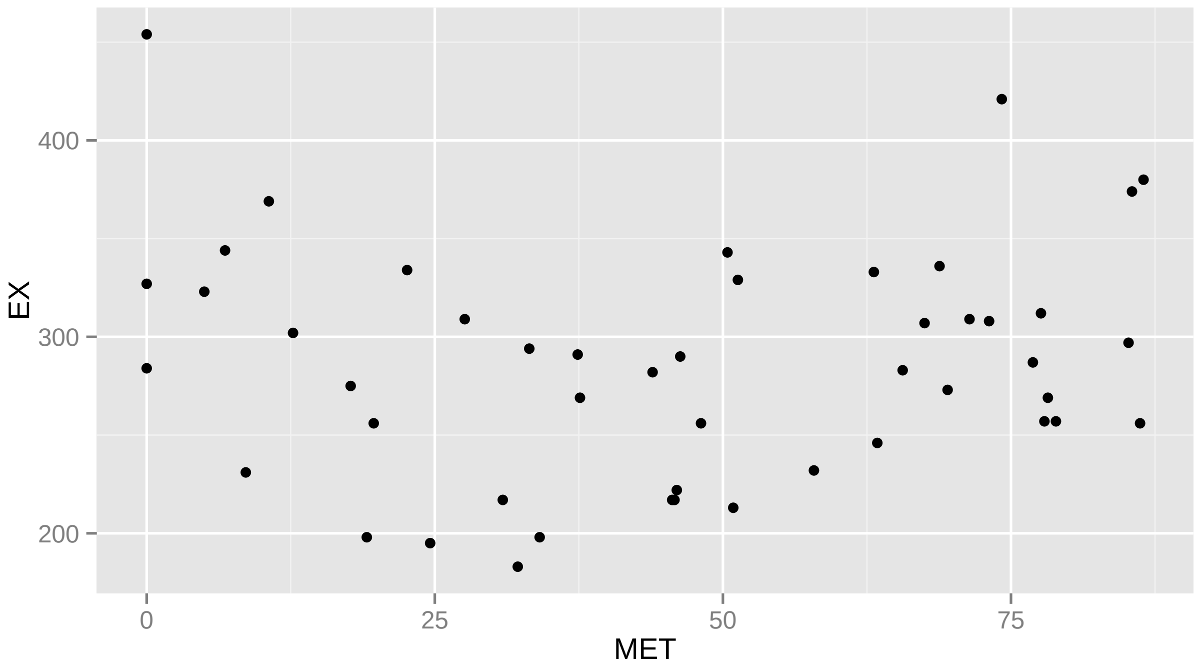
Caroline Svahn, Niclas Lovsjö and Gustav Sternelöv

Monday, November 16, 2015

Assignment 1

1.1

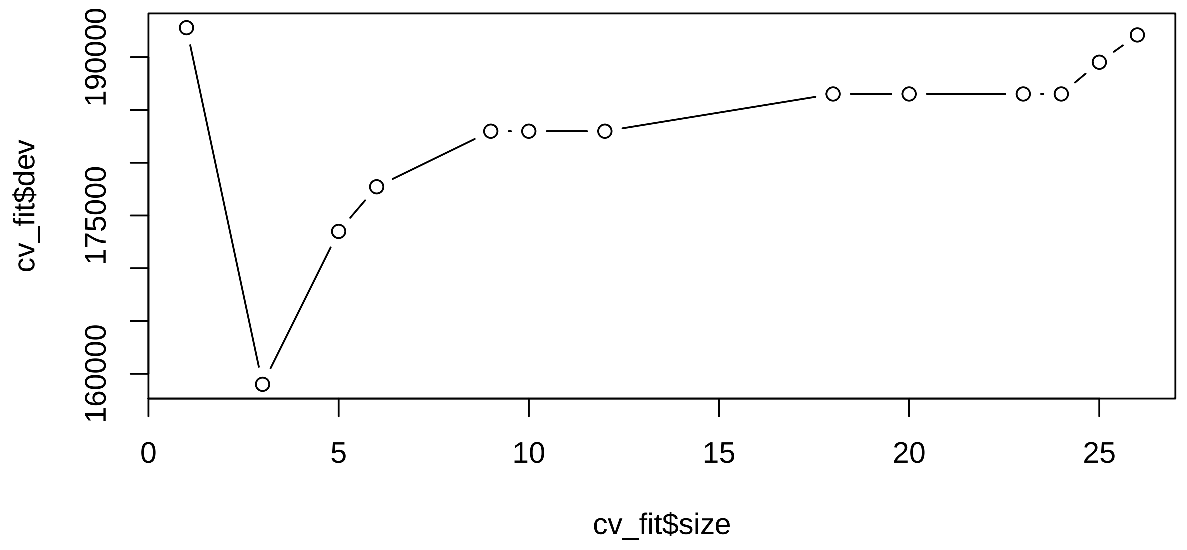
The variables analyzed in the first assignment are MET (Percentage of population living in standard metropolitan areas) and EX (Per capita state and local public expenditures ($)). The latter is the target variable and the former the input variable. A plot of MET versus EX can be seen below.



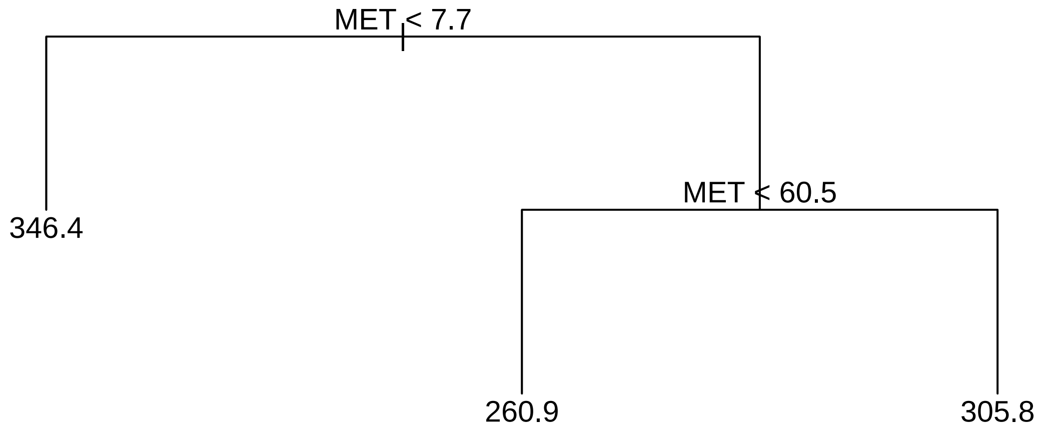
An analyze of the graph gives that for low values and for high values of MET the value of EX is high. An interpretation of this is that models who creates a linear decision boundary probably will result in bad fits. Another type of model, one that can capture a nonlinear pattern is therefore thought to be needed in this case.

1.2

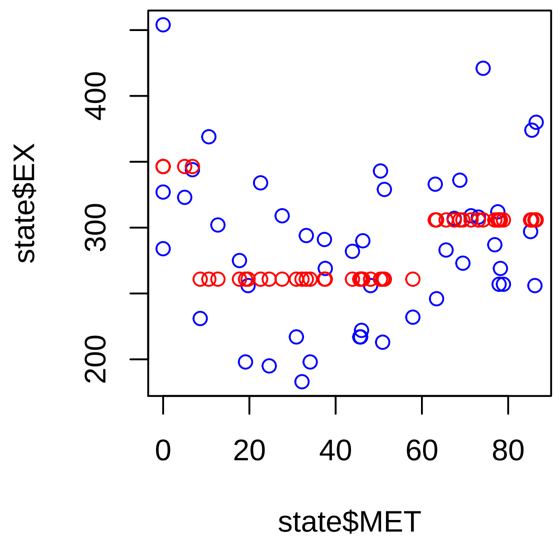
We fit a regression tree, i.e. a decision tree with continuous target variable, using cross-validation to select the number of trees for the entire dataset. We also want to restrict the minimum number of observations for one leaf to 2.

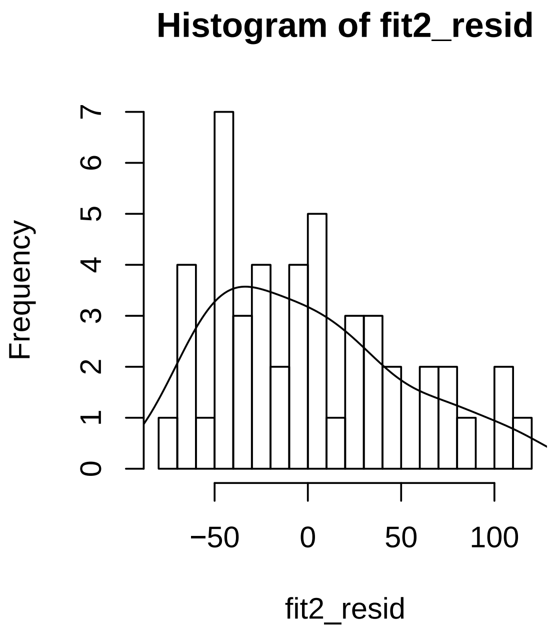


The lowest CV-score is given when three leaves are selected. The original tree is therefore pruned until it only contains three leaves. This pruning of the original tree results in the following regression tree.



As seen in the figure above, the tree only reports three values for expenditures: 346.4, 260.9 and 305.8. The highest value of expenditures is obtained when the percentage of the population living in standard metropolian areas is less than 7.7 %. If the percentage is higher than 7.7 %, the value is evaluated once more. If MET is below 60.5 %, the minimum value of 260.9 is obtained, and if it is higher than 60.5 % the model returns 305.8 The original data is then plotted against the fitted values and the residuals are plotted in a histogram.



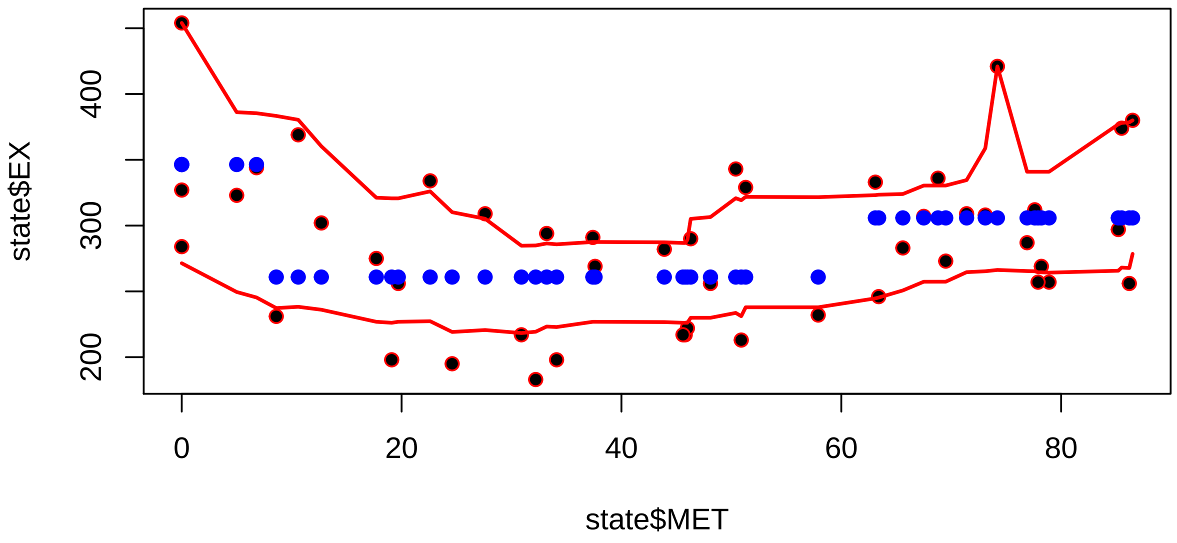


That the model seem to be quite a bad fit is clearly shown by the plot over the fitted values against the original values. Since the variance for the predicted values is low and the bias is high the model is concluded to be underfitted and quite some information seem to be lost when only returning three different values.

The residuals are not apparent normally distributed, which is expected since the fitted values only take three different values, leaving rather large residuals. Although the data set consists of a rather few amount of observations.

1.3

For the selected regression tree model are 95 % confidence bands computed by using a non-parametric bootstrap. The confidence bands are plotted together with both the original and fitted values.

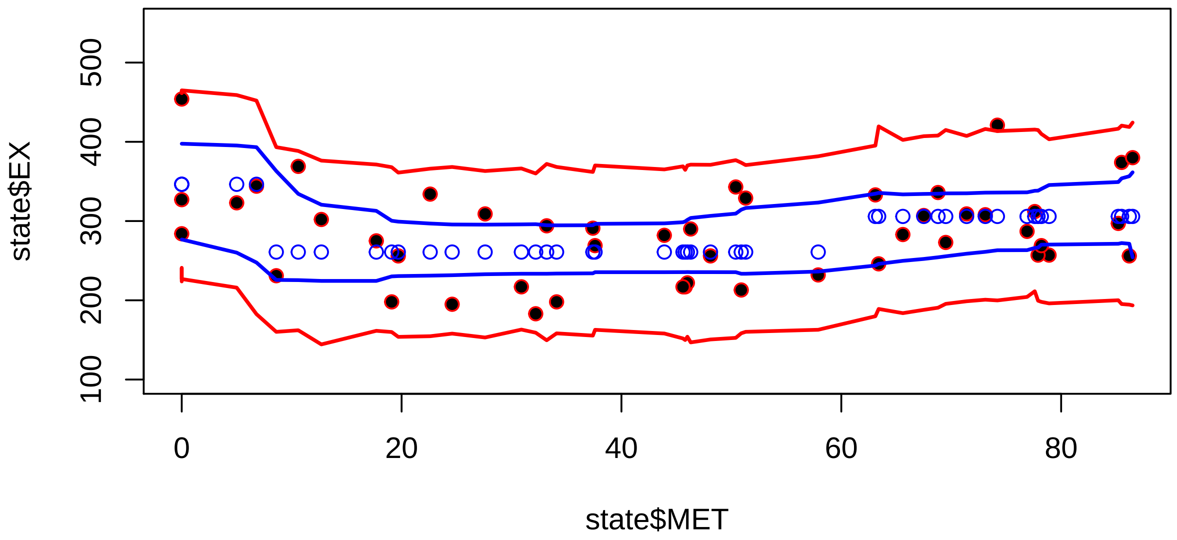


The confidence bands are bumpy, that is because non-parametric bootstrapping do not operate under the assumption of belonging to any particluar distribution. An effect of this is that the bands are applied with more consideration to the actual values - thus, the bands appear "'bumpy".

In general the conclusion is that little knowledge about the expected response is gained by the model. That is true especially for the lowest values on the y-axis.

1.4

When using a parametric bootstrap to compute the 95 % confidence and prediction bands for the regression tree model, the following bands can be plotted:



The fitted model does not seem to be appropriate for this data. This can partly be explained by the fact that there is a lot of uncertainty in the data observed. This is well illustrated by the prediction bands, since they capture both the expected interval for the observations and the potential effects of randomness. Only one point are outside the prediction bands, that is approximately 5 % of data.

1.5

The confidence bands are quite similar. The main difference is the bumpiness of the non-paramateric bands and the more unflexible parametric bands.

The advantage with the parametric bootstrap is thtat it gives confidence intervals that are more general and not too sensitive to extreme values as the confidence interval for the non-parametric bootstrap is.

This advantage gives that the parametric bootstrap may be more appropriate. Although a disadvantage may be that the assumption of normality. As noted in 1.2 there might be uncertainty over making that assumption.

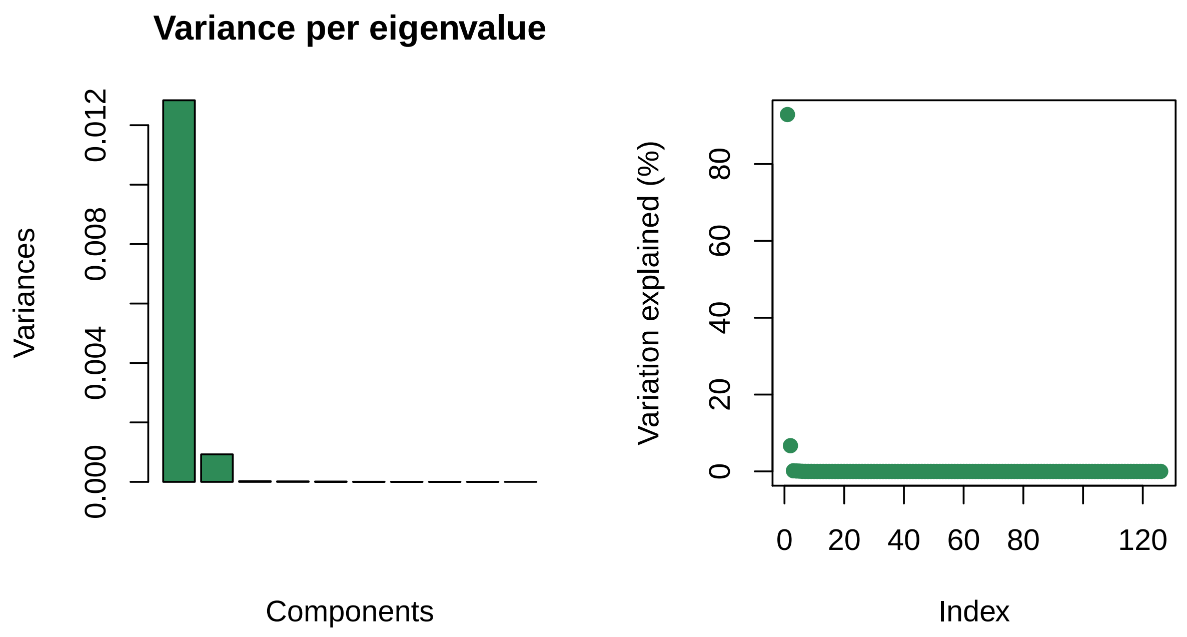
If one of the methods must be suggested, the non-parametric bootstrap is thought to be more appropriate due to the problems with finding an appropriate distribution for the parametric method.

Assignment 2

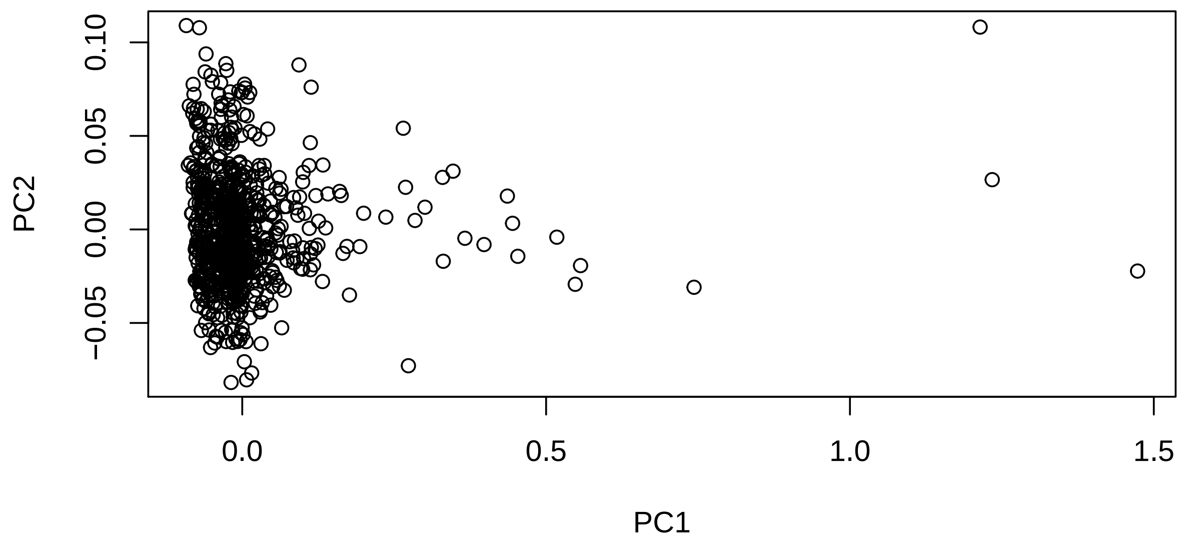
The data analysed in the second assignment consists of near-infrared spectra and viscosity levels for a collection of diesel fuels. The objective with the assignment is to examine how the measured spectra can be used in order to predict the viscosity.

2.1

A standard PCA with all features included is performed to determine how many principal components that are needed to explain at least 99% of the total variance. How much of the variation that is explained by each principal component is presented with the following graph.



From the plot, it is evident that the first component explains by far the most variance, percentwise. The second explains some amount while the rest do not appear to explain any significant variance. Since the plot does not show the percentage of explained variance, it is not sufficient to conclude how many components are needed to obtain 99 % of the variance explained. Further investigation of the components reveals that the first component explains 92.9 % of the variance and the second 6.7 %. Together, the first two components explain 99.6 %, which is then sufficient for reaching 99 % and therefore the first two components will be used. Furthermore, the scores for the two first components:



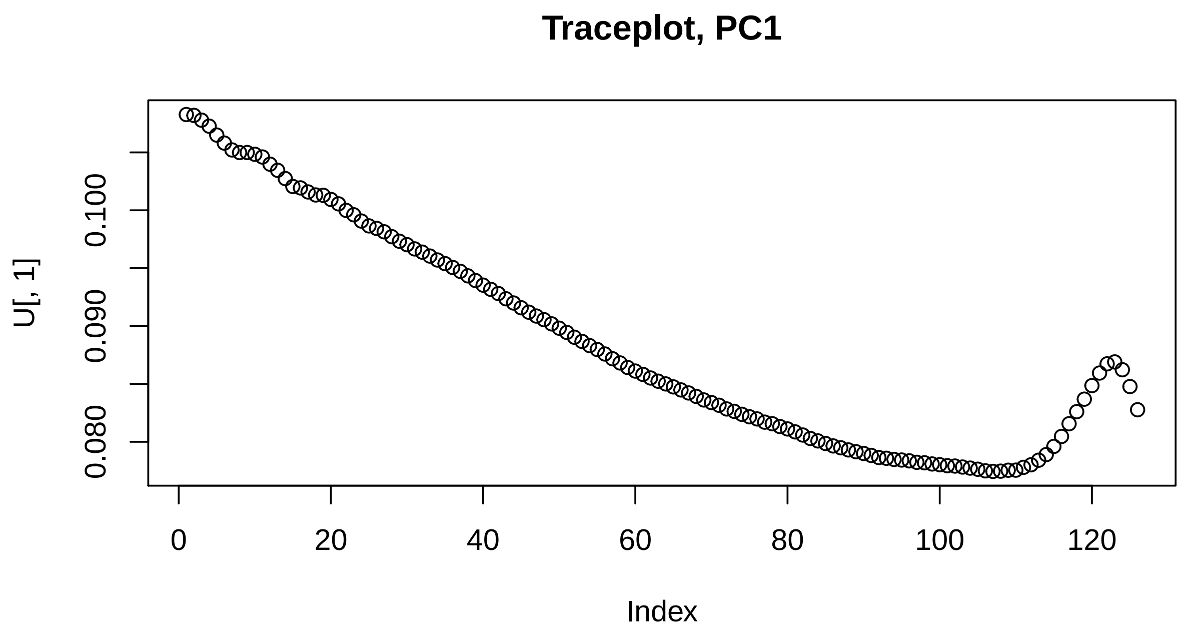
Mainly, the scores of the first component are higher, all above 0, and shows some quite large values. The scores for PC2 is lower, which shows that the second component explains less than the first component.

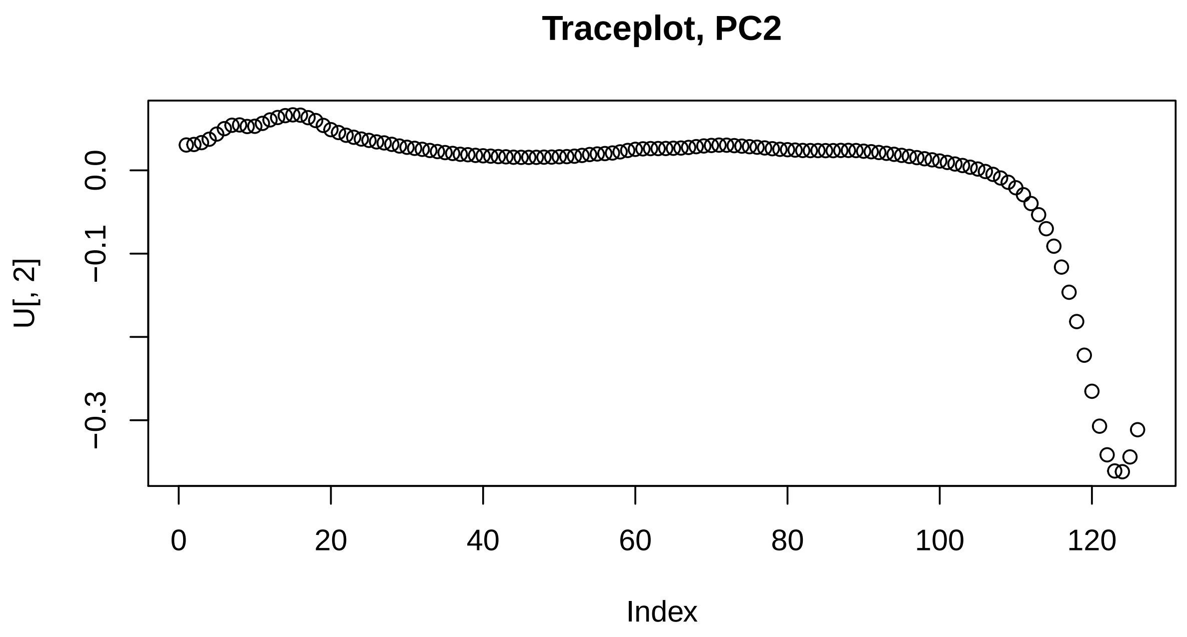
The majority of the scores lies to left in the graph and some outlying points can be noted at the right end of the graph.

The outlying values x-wise can be interpreted to be the features with high scores for PC1. The valeus with low or high values y-wise are features with high scores for PC2.

2.2

The loadings of the components PC1 and PC2 are here visualised by so called trace plots.





We see that the first PC is explained by all factors, but that the first factors have higher degree of explanation. Applying the facts for interpreting the first component also on the second trace plot, more values are closer to 0 than for the first component. The first about a hundred factors keep close to zero, then the plot shows a steep downward trend. Thus, the first features do not impact that much, only the last few.

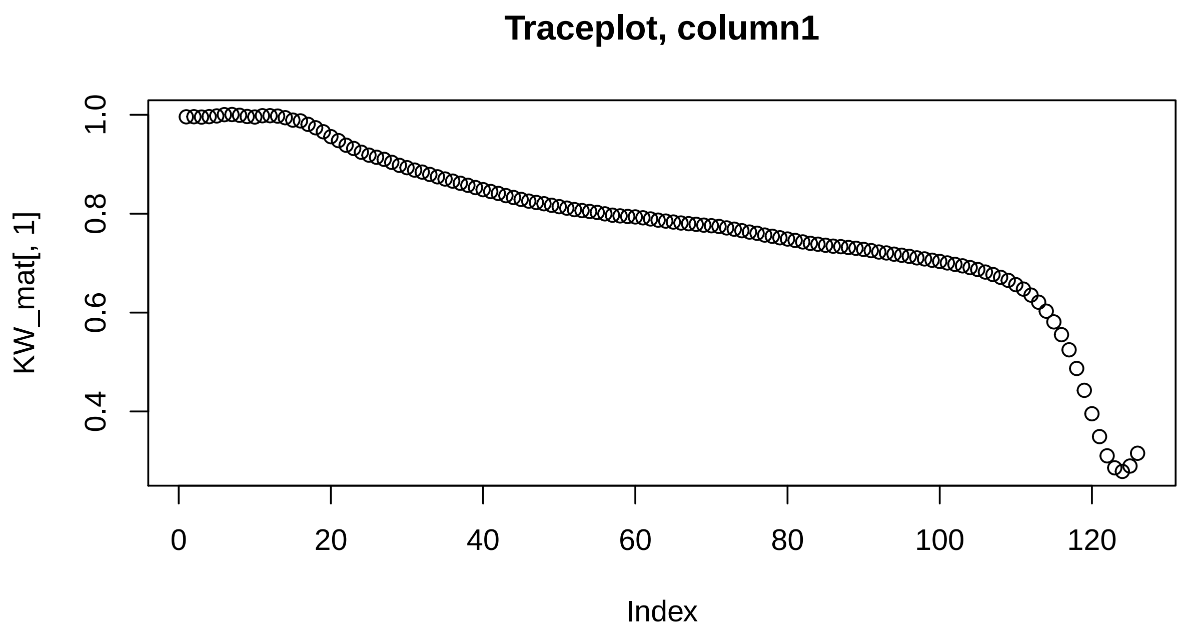
2.3

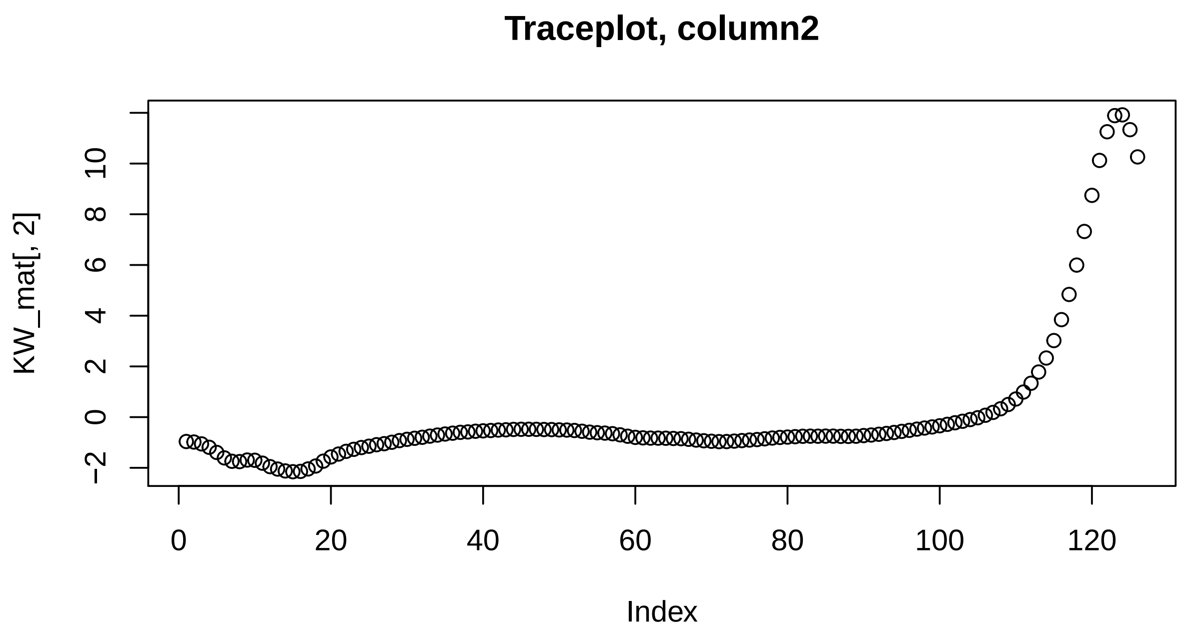
The next model that is conducted is a Independent Component Analysis, ICA, with two components.

a) The matrix W is presented by the table below. W is a weighting matrix that is used to minimize the mutual information between variables.This means that role the of the W matrix is to make the components in the model uncorrelated. An interpretation of the values in the matrix gives that… the first component needs to be transformed quite much since the absolute value of first value in the matrix is high, almost equal to one. The second component is connected to a low value, almost equal to zerom and that means a small transformation is needed for this component.

```
##            [,1]       [,2]
## [1,] -0.9991790 -0.0405144
## [2,] -0.0405144  0.9991790
```

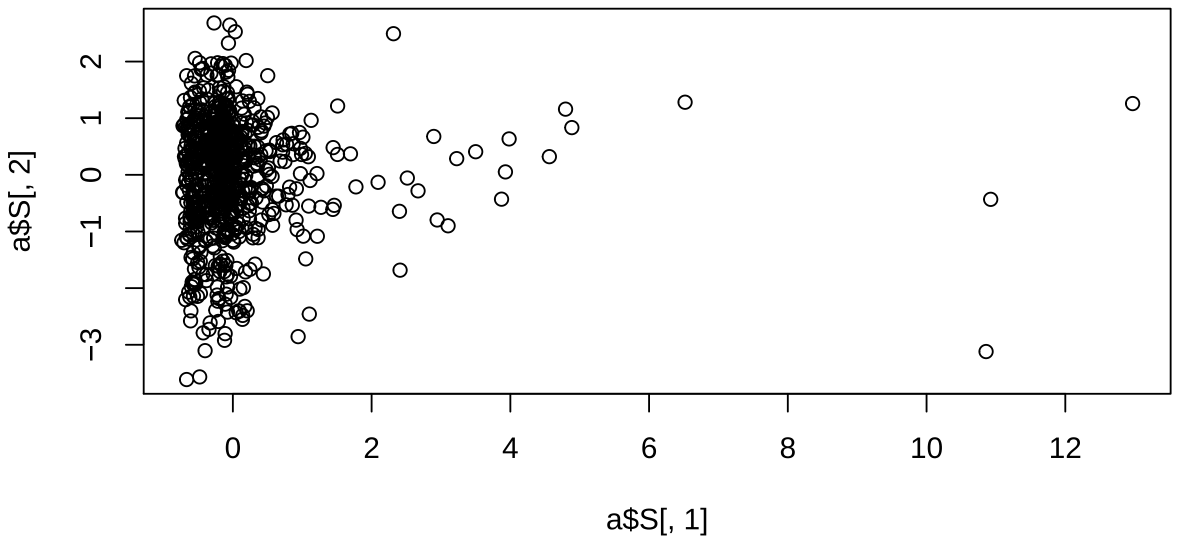
b) The next matrix presented is the W' matrix. The role of this matrix is to… Gives the rotated, uncorrelated, loadings for the components of the ICA model? The two columns of the W' matrix are here presented by trace plots.





When comparing with the trace plots in 2.2 it can be noted that a similar conclusion can be drawn for the respecitive component. For the frist component all factors have non-zero loadings. For the second component the majority of the factors have loadings close to zero and therfore only a few factors explains the component.

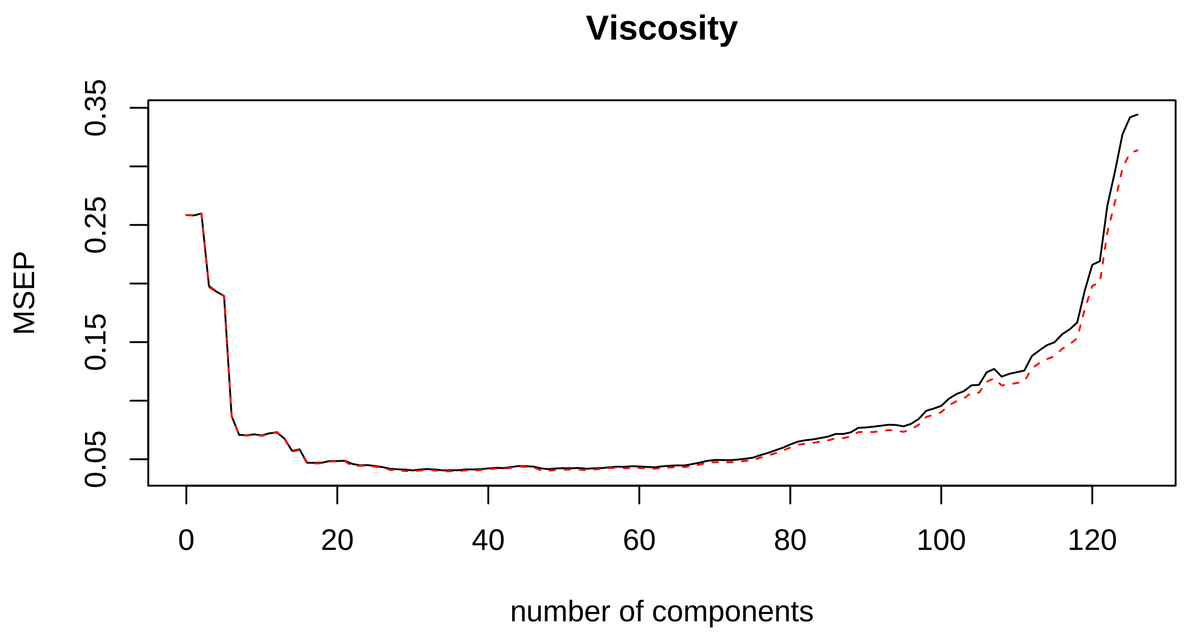
c) A score plot for the components in the ICA model



A comparsion with the score plot in 2.1 gives that the scores for the components in the ICA model is rather similar to those for the first two components of the standard PCA.

2.4 - 2.5

Before fitting a PCR and PLS model the data is divided into a training and a test set. The respective data set contains 50% each of the original data set.
A PCR model is fitted where the number of components in the model is selected by cross-validation. A low mean-square prediction error is wanted and the number of components that corresponds to the lowest error is chosen. To examine this further the dependence between the mean-square prediction error and the number of components is plotted.



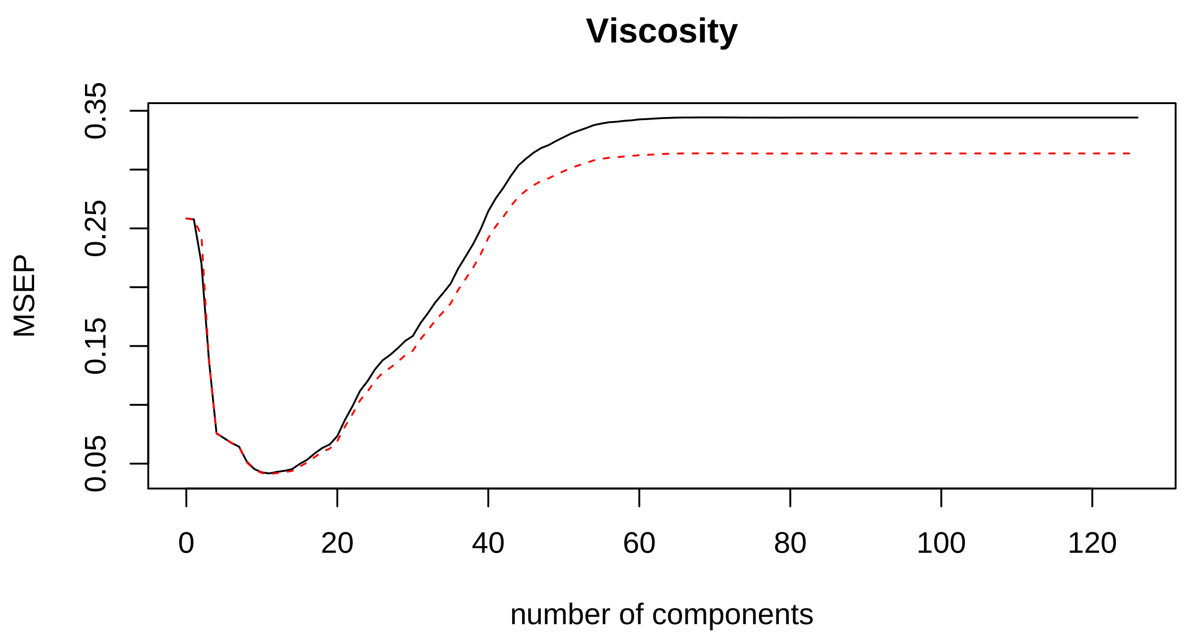
The value for the mean-square prediction error term is approximately the same for a quite longe range of components. For the number of components from about 25 up to around 60 it is a very slight difference between the mean-square prediction errors. Since there is no reason to include more components if they don't improve the results, a reasonable choice could be to select 25 components.

The test set is used to evaluate the selected, optimal, model with 25 components. This is done by calculating the mean-square error for the model when applied on the test set.

```
## [1] 0.06783242
```

2.6

The workflow for 2.5 is repeated in 2.6, with the difference that a PLS model is fitted instead of a PCR.



The lowest mean-square prediction error is given when the number of components is equal to 12. When the model is used on test data the mean-square error is calculated to be equal to 0.0673106. Compared to the mean-square error obtained for the PCR model so is the value slightly lower for the PLS model. The PLS model therefore seems to work equally well as the PCR model, even though it only uses 12 components compared to the 25 components used by the PCR model.

Appendix

R-code

```
state <- read.csv("C:/Users/Gustav/Documents/Machine-Learning/Lab 4/State.csv", sep=";")
library(ggplot2)
ggplot(state, aes(x=MET, y=EX)) + geom_point()
library(tree)
set.seed(12345)
fit2 <- tree(EX ~ MET, data=state, control=tree.control(nobs=48, minsize=2))
cv_fit <- cv.tree(fit2)
plot(cv_fit$size, cv_fit$dev, type="b")
pruneFit2 <- prune.tree(fit2, best=3)
plot(pruneFit2)
text(pruneFit2, pretty=0)
cv_pred <- predict(pruneFit2)
fit2_resid <- state$EX - cv_pred
myhist <- hist(fit2_resid, breaks=14)
par(mfrow=c(1,2))
plot(state$MET, state$EX, col="blue")
```

```r
points(state$MET, cv_pred, col="red")

multiplier <- myhist$counts / myhist$density
mydensity <- density(fit2_resid)
mydensity$y <- mydensity$y * multiplier[1]

plot(myhist)
lines(mydensity)
par(mfrow=c(1,1))
# 1.3
# Non-parametric bootstrap
# 95 % confidence bands
library(boot)
data2=state[order(state$MET),]#reordering data according to MET
# computing bootstrap samples
f=function(data, ind){
  data1=data[ind,]# extract bootstrap sample
  #fit regression tree
  res=tree(EX ~ MET, data=data1, control=tree.control(nobs=48, minsize=2))
  #predict values for all Area values from the original data
  priceP=predict(res,newdata=data2)
  return(priceP)
}
res=boot(data2, f, R=1000) #make bootstrap

# Create lower and upper bound.
e=envelope(res)

fit=prune.tree(fit2, best=3)
priceP=predict(fit)

plot(state$MET, state$EX, pch=21, bg="black", col="red")
points(data2$MET,priceP,type="b", col="blue") #plot fitted line
#plot cofidence bands
points(data2$MET,e$point[2,], type="l", col="red", lwd=2)
points(data2$MET,e$point[1,], type="l", col="red", lwd=2)

# Parametric bootstrap
# 95 % confidence and prediction bands
# Assumes that Y follows the normal distribution
mle=prune.tree(fit2, best=3)
rng=function(data, mle) {
  data1=data.frame(EX=data$EX,
                   MET=data$MET, data=data)
  n=length(data$EX)
  data1$EX=rnorm(n,predict(mle,newdata=data1),sd(mle$y))
  return(data1)
}
f1=function(data1){
  res=tree(EX ~ MET, data=data1, control=tree.control(nobs=48, minsize=2))
  predictedP=predict(res, newdata=data2)
  return(predictedP)
}
```

```r
res=boot(data2, statistic=f1, R=1000, mle=mle, ran.gen=rng , sim="parametric")
e2=envelope(res)
fit=prune.tree(fit2, best=3)
pred2=predict(fit)
plot(state$MET, state$EX, pch=21, bg="black", col="red", ylim=c(100, 550))
points(data2$MET,pred2,type="b", col="blue") #plot fitted line
#plot cofidence bands
points(data2$MET,e2$point[2,], type="l", col="red", lwd=2)
points(data2$MET,e2$point[1,], type="l", col="red", lwd=2)
spectra <- read.csv("C:/Users/Gustav/Documents/Machine-Learning/Lab 4/NIRSpectra.csv", sep=";")
data_a <- spectra
data_a$Viscosity=c()
data_a$ID=c()
res=prcomp(data_a)
lambda=res$sdev^2
#proportion of variation explained by each feature
plot(sprintf("%2.3f",lambda/sum(lambda)*100), ylab="Variation explained (%)")
# Scores in coordinates of PC1 and PC2
plot(res$x[,1], res$x[,2], xlab="PC1", ylab="PC2")
U=res$rotation
plot(U[,1], main="Traceplot, PC1")
plot(U[,2],main="Traceplot, PC2")

library(fastICA)

set.seed(12345)
a <- fastICA(data_a, 2, alg.typ = "parallel", fun = "logcosh", alpha = 1,
             method = "R", row.norm = FALSE, maxit = 200, tol = 0.0001, verbose = TRUE) #ICA

W_mat <- a$W
W_mat
KW_mat <- a$K %*% a$W
# plot the columns as trace plots
plot(KW_mat[,1], main="Traceplot, column1")
plot(KW_mat[,2],main="Traceplot, column2")
# The score plot
plot(a$S[,1], a$S[,2])
dat <- as.matrix(spectra)
n=dim(dat)[1]
set.seed(12345)
id=sample(1:n, floor(n*0.5))
train=dat[id,]
test=dat[-id,]

library(pls)
train <- data.frame(train[, 2:128])
test <- data.frame(test[, 2:128])
set.seed(12345)
pcr.fit=pcr(Viscosity~., data=train, validation="CV")
validationplot(pcr.fit,val.type="MSEP")

pcr.fit1=pcr(Viscosity~., 25,data=train, validation="none")
pcr.pred <- predict(pcr.fit1, newdata=test, ncomp = 25)
pcr.mse <- 1/length(na.omit(test$Viscosity)) * sum((test$Viscosity - pcr.pred)^2, na.rm=TRUE)
pcr.mse
set.seed(12345)
plsr.fit=plsr(Viscosity~., data=train, validation="CV")
validationplot(plsr.fit,val.type="MSEP")
plsr.fit1=plsr(Viscosity~., 12,data=train, validation="none")
plsr.pred <- predict(plsr.fit1, newdata=test, ncomp = 12)
plsr.mse <- 1/length(na.omit(test$Viscosity))* sum((test$Viscosity - plsr.pred)^2, na.rm=TRUE)
##
```