Lab 2 Report

Zuxiang Li

12/6/2019

Assignment 2. Analysis of credit scoring

The data file creditscoring.xls contains data retrieved from a database in a private enterprise. Each row contains information about one customer. The variable good/bad indicates how the customers have managed their loans. The other features are potential predictors. Your task is to derive a prediction model that can be used to predict whether or not a new customer is likely to pay back the loan.

1. Import the data to R and divide into training/validation/test as 50/25/25: use data partitioning code specified in Lecture 1e.

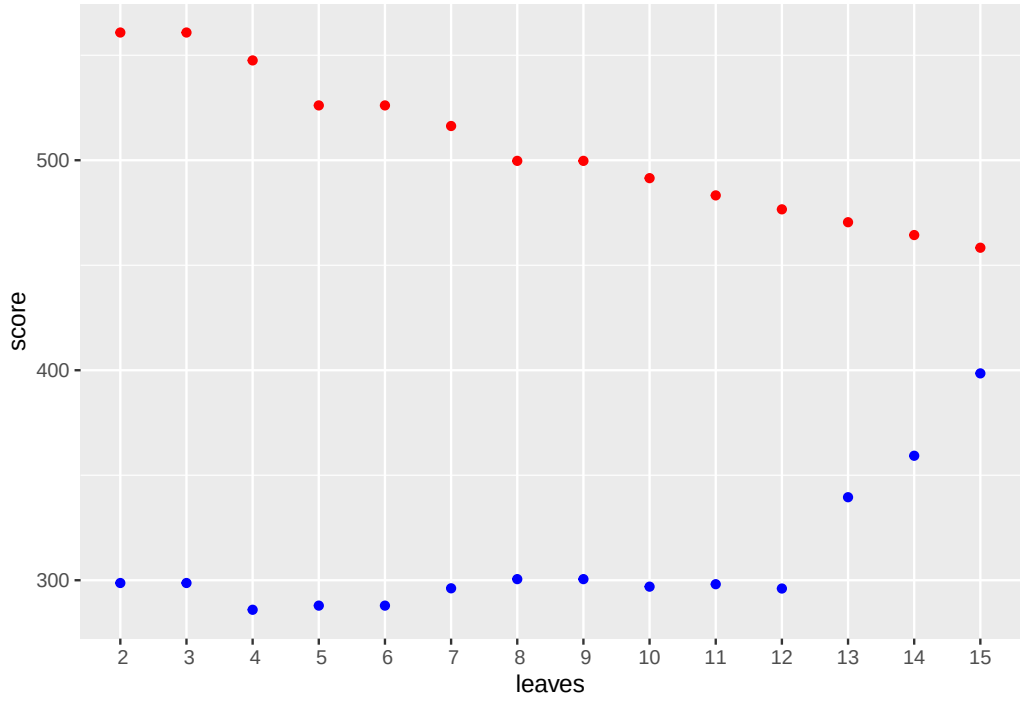
2. Fit a decision tree to the training data by using the following measures of impurity a. Deviance b. Gini index and report the misclassification rates for the training and test data. Choose the measure providing the better results for the following steps.

```
## using  deviance as measure
##
##  Confusion matrix and misclassfication rate for train data      actual
## pred   bad good
##   bad   61   20
##   good  86  333
## [1] 21.2
##
##  Confusion matrix and misclassfication rate for test data
##       actual
## pred   bad good
##   bad   28   19
##   good  48  155
## [1] 26.8
```

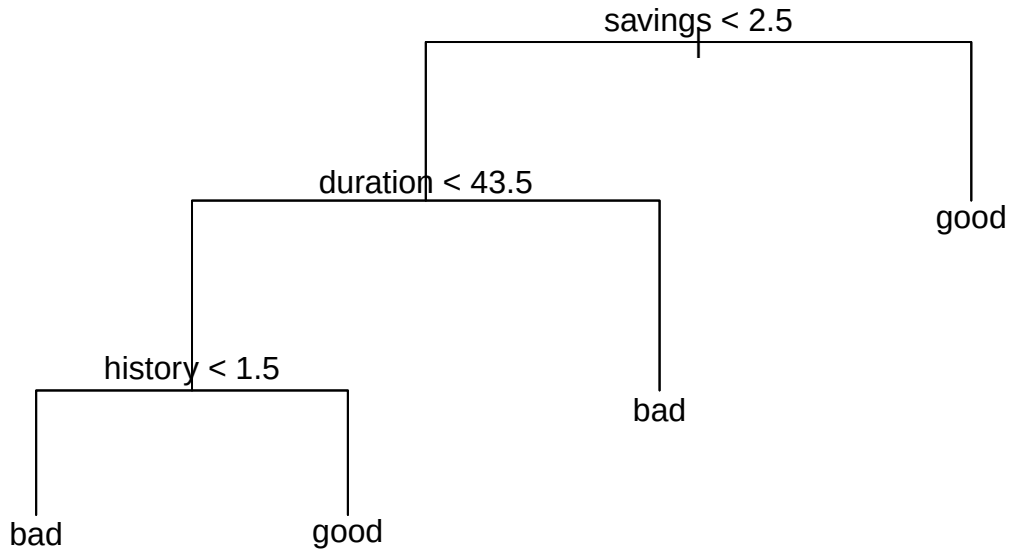
```
## using  gini as measure
##
##  Confusion matrix and misclassfication rate for train data      actual
## pred   bad good
##   bad   66   38
##   good  81  315
## [1] 23.8
##
##  Confusion matrix and misclassfication rate for test data
##       actual
## pred   bad good
##   bad   18   35
##   good  58  139
## [1] 37.2
```

From the result we can observe that while using deviance as measure of impurity, the misclassfication rate for both train and test data are lower than using gini. So we will choose deviance as model parameter for following steps.

3. Use training and validation sets to choose the optimal tree depth.Present the graphs of the dependence of deviances for the training and the validation data on the number of leaves. Report the optimal tree, report it's depth and the variables used by the tree. Interpret the information provided by the tree structure. Estimate the misclassification rate for the test data.




```
##
## Classification tree:
## snip.tree(tree = tree_model, nodes = c(5L, 3L, 9L))
## Variables actually used in tree construction:
## [1] "savings"  "duration" "history"
## Number of terminal nodes:  4
## Residual mean deviance:  1.117 = 547.5 / 490
## Misclassification error rate: 0.251 = 124 / 494
```




```
##       actual
## pred   bad good
##   bad   18    6
##   good  58  168
## [1] 25.6
```

The red dots in plot represent train score, blue dots mean for test score. As we can see when leaves=4 the modal got best performance. The tree used Savings, Duration and History as variables. When saving>2.5 the result is good else goes to check duration value, if value>43.5 then the result is bad, else we check history, if history>1.5 the result is good,else it's bad. The misclassfication rate for test data is 25.6%

4. Use training data to perform classification using Naive Bayes and report the confusion matrices and misclassification rates for the training and for the test data. Compare the results with those from step 3.

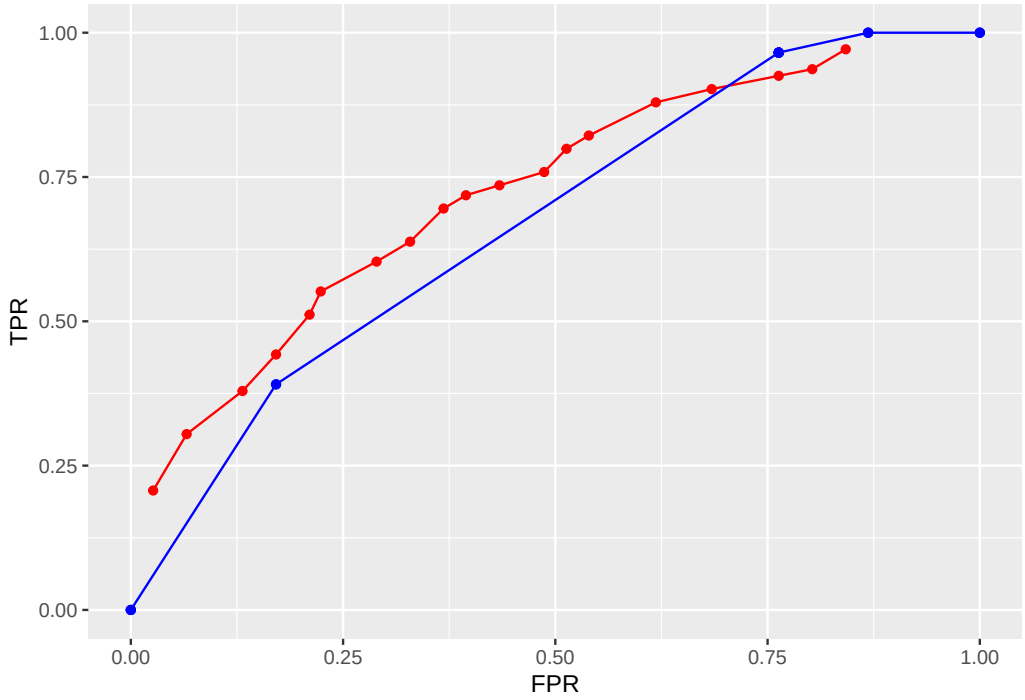
```
##
##  Confusion matrix and misclassfication rate for train data

##       actual
## pred   bad good
##   bad   95   98
##   good  52  255
## [1] 30

##
##  Confusion matrix and misclassfication rate for test data

##       actual
## pred   bad good
##   bad   46   49
##   good  30  125
## [1] 31.6
```

The misclassfication rate for train data is 30%, 31.6% for test data. Both of them are higher than result in step 3.

5. Use the optimal tree and the Naive Bayes model to classify the test data by using the following principle:

$$\widehat{Y} = 1 \quad if \quad p(Y =' good'|X) > \pi, otherwise \quad \widehat{Y} = 0$$

where {pi} =0.05,0.1,0.15,...0.9,0.95. Compute the TPR and FPR values for the two models and plot the corresponding ROC curves. Conclusion?



The ROC curves for two models are as shown above. Red line for naive bayes model and blue line for tree model. A better classfier should have a greater area under curve. So in this case the naive bayes is a better classfier.

6. Repeat Naive Bayes classification as it was in step 4 but use the following loss matrix

```
##
##  Confusion matrix and misclassfication rate for train data

##     actual
## pred bad good
##    0  27   17
##    1 120  336
## [1] 27.4

##
##  Confusion matrix and misclassfication rate for test data

##     actual
## pred bad good
##    0  14   10
##    1  62  164
## [1] 28.8
```

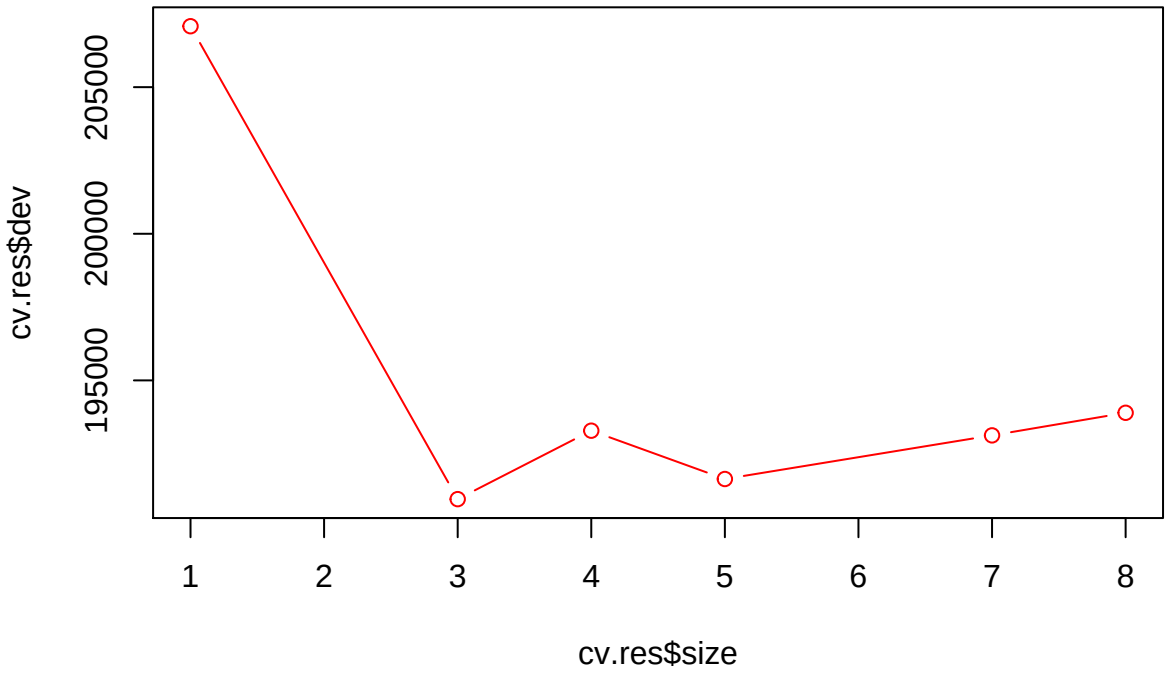
Compare the result with step 4 we can see that the misclassfication with loss matrix is lower.

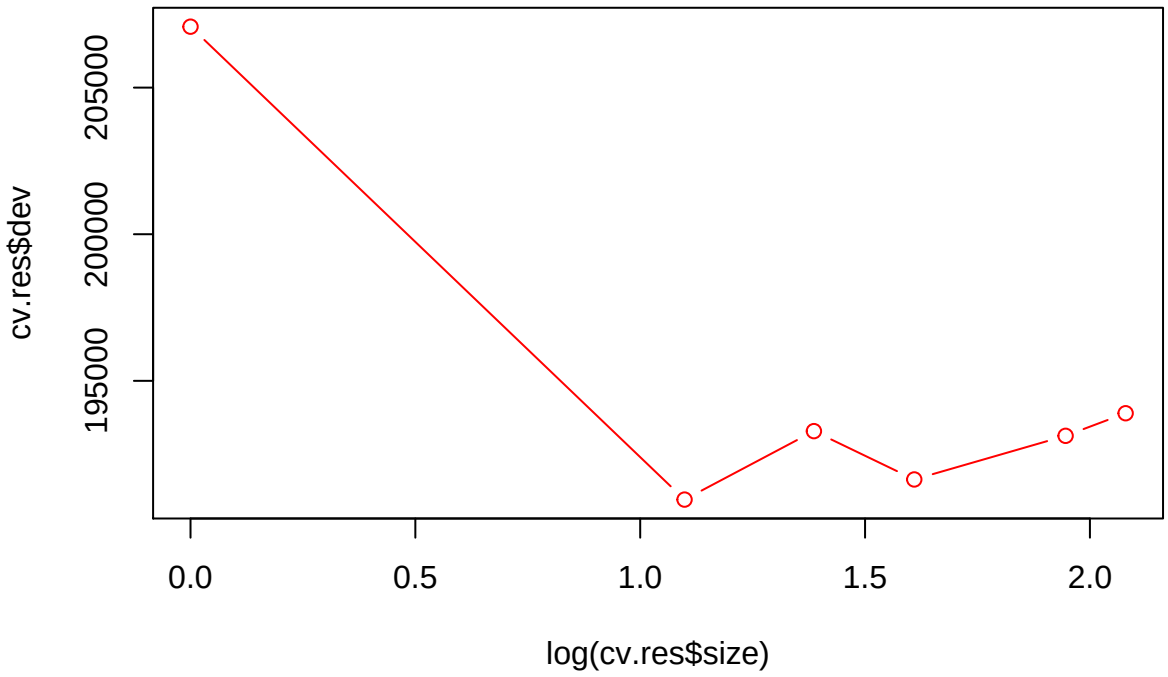
Assignment 3. Uncertainty estimation

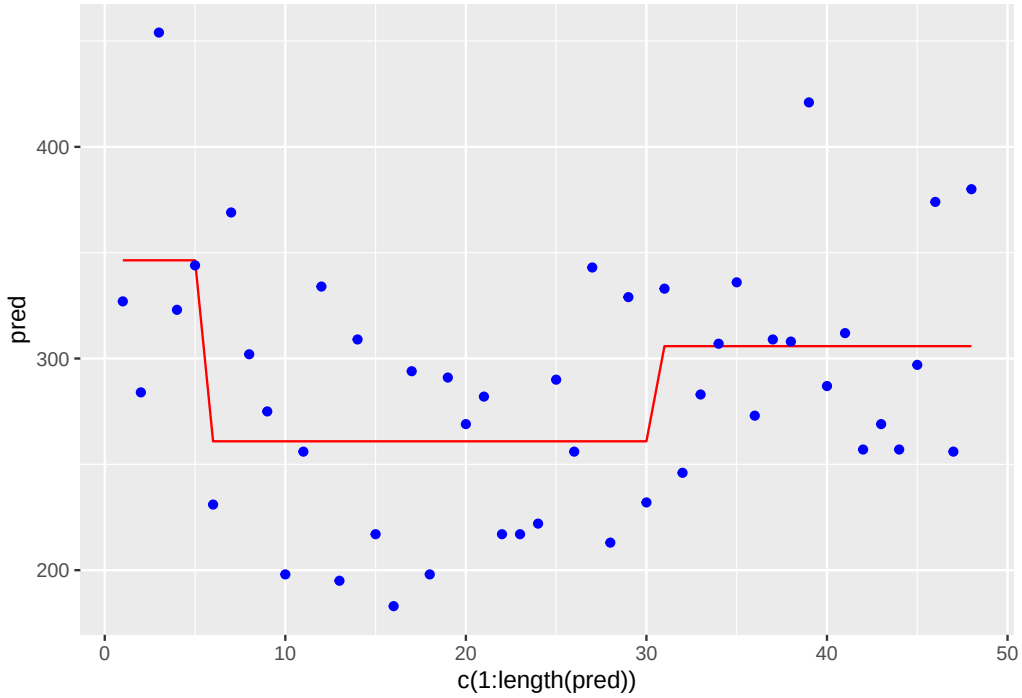
The data file State.csv contains per capita state and local public expenditures and associated state demographic and economic characteristics, 1960, and there are variables * MET: Percentage of population living in standard metropolitan areas * EX: Per capita state and local public expenditures ($)

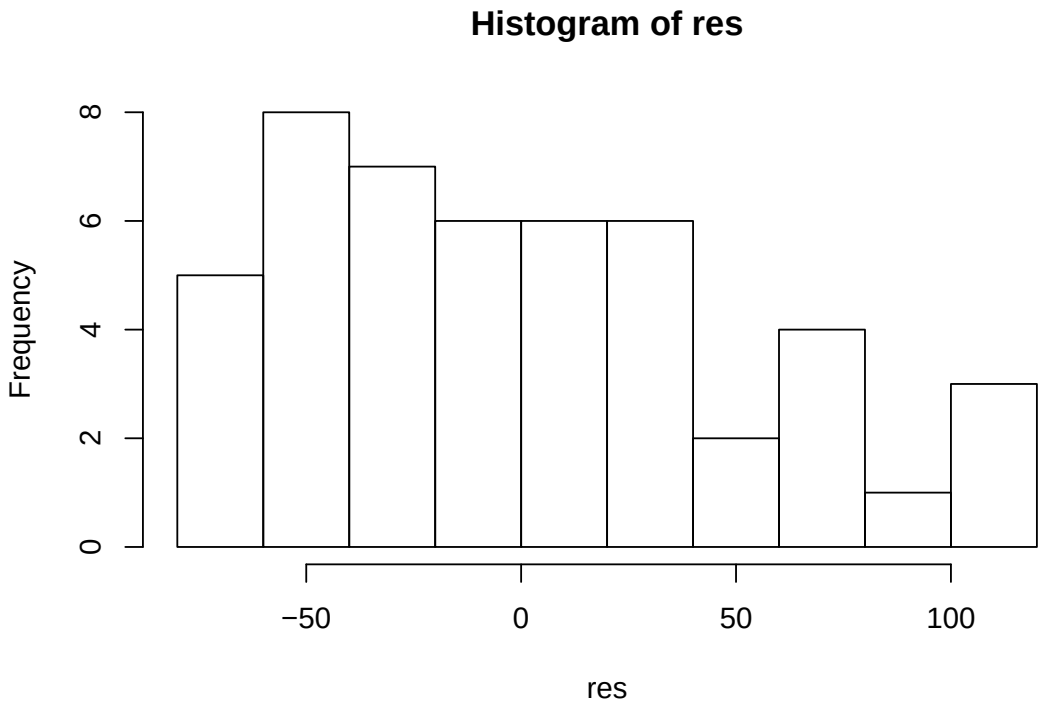
1. Reorder your data with respect to the increase of MET and plot EX versus MET. Discuss what kind of model can be appropriate here. Use the reordered data in steps 2-5.

2. Use package tree and fit a regression tree model with target EX and feature MET in which the number of the leaves is selected by cross-validation, use the entire data set and set minimum number of observations in a leaf equal to 8 (setting minsize in tree.control). Report the selected tree. Plot the original and the fitted data and histogram of residuals. Comment on the distribution of the residuals and the quality of the fit.




cv.res$dev
205000
200000
195000
0.0
0.5
1.0
1.5
2.0
log(cv.res$size)


pred
400
300
200
0
10
20
30
40
50
c(1:length(pred))



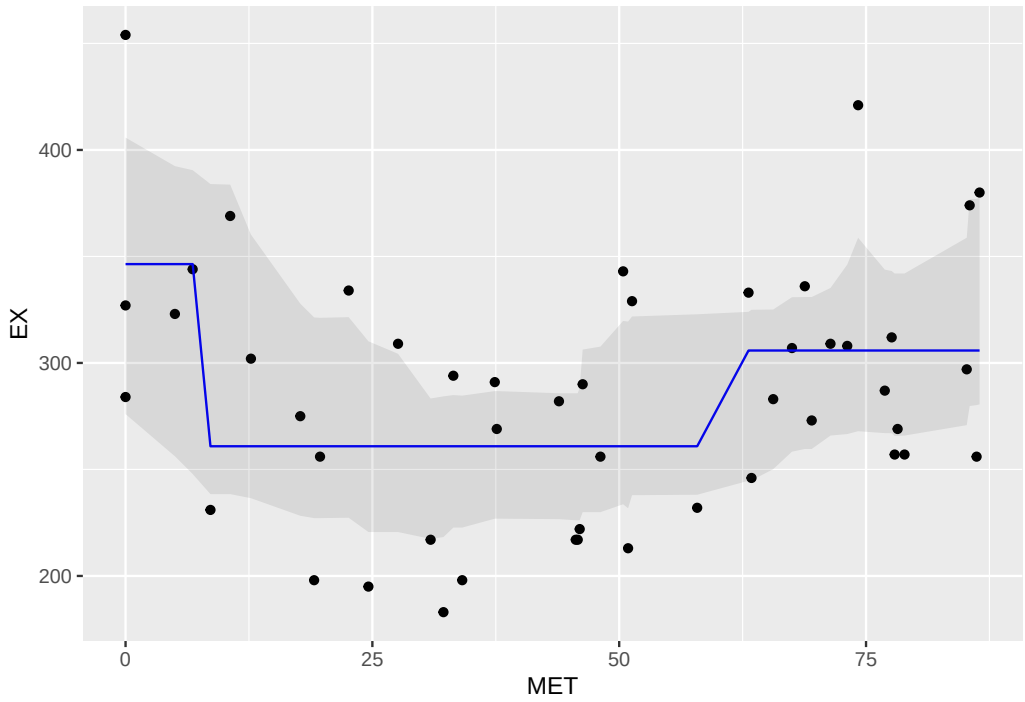

```
## [1] 1.184223e-14
```

```
## [1] 50.82183
```

The selected tree should be 3 in this case, the residuals distributed as normal distribution with mean 0 and standard deviance 50.8. The histogram of residuals suggest it is slightly skewed to the right, which means it's a good fit initially.

3. Compute and plot the 95% confidence bands for the regression tree model from step 2 (fit a regression tree with the same settings and the same number of leaves as in step 2 to the resampled data) by using a non-parametric bootstrap. Comment whether the band is smooth or bumpy and try to explain why. Consider the width of the confidence band and comment whether results of the regression model in step 2 seem to be reliable.

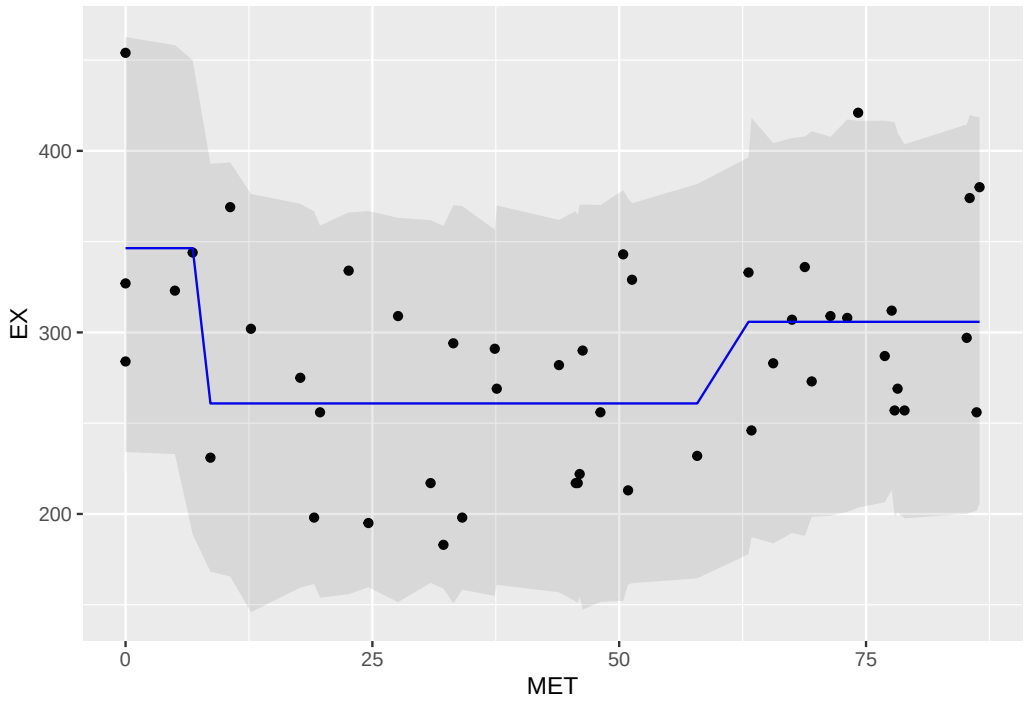
```
## Warning in prune.tree(tree_mod, best = 3): best is bigger than tree size
```



From the plot, we can see that the confidence band is quite bumpy and almost half of the dots are outside of the confidence band, which indicate that it's not a reliable model. The band is bumpy because nonparametric bootstrap does not work well for discrete estimators.

4. Compute and plot the 95% confidence and prediction bands the regression tree model from step 2 (fit a regression tree with the same settings and the same number of leaves as in step 2 to the resampled data) by using a parametric bootstrap, assume are labels in the tree leaves and is the residual variance. Consider the width of the confidence band and comment whether results of the regression model in step 2 seem to be reliable. Does it look like only 5% of data are outside the prediction band? Should it be?

```
## Warning in envelope(boot_res1, level = 0.95): unable to achieve requested
## overall error rate
```



From the plot we can see that there are two points are outside the confidence band, it's more reliable than previous models.In addition, the sample with small size using nonparametric bootstrap can underestimate the amount of variation, but using parametric bootstrap which take values from known distribution covering a wider range.

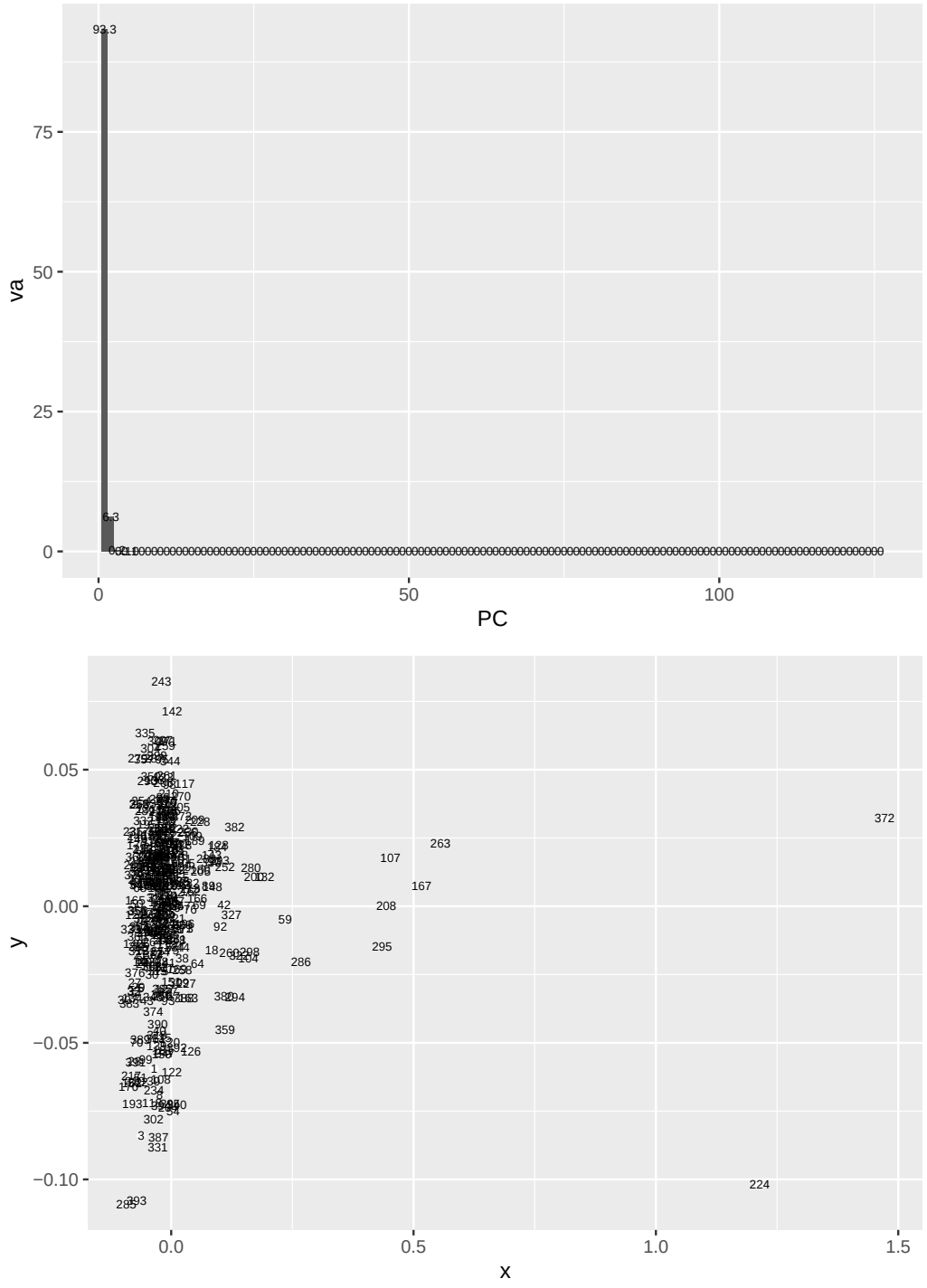
5. Consider the histogram of residuals from step 2 and suggest what kind of bootstrap is actually more appropriate here.

Parametric bootstrap is more suitble here since the histrogram of residuals looks like normal distribution.

Assignment 4. Principal components

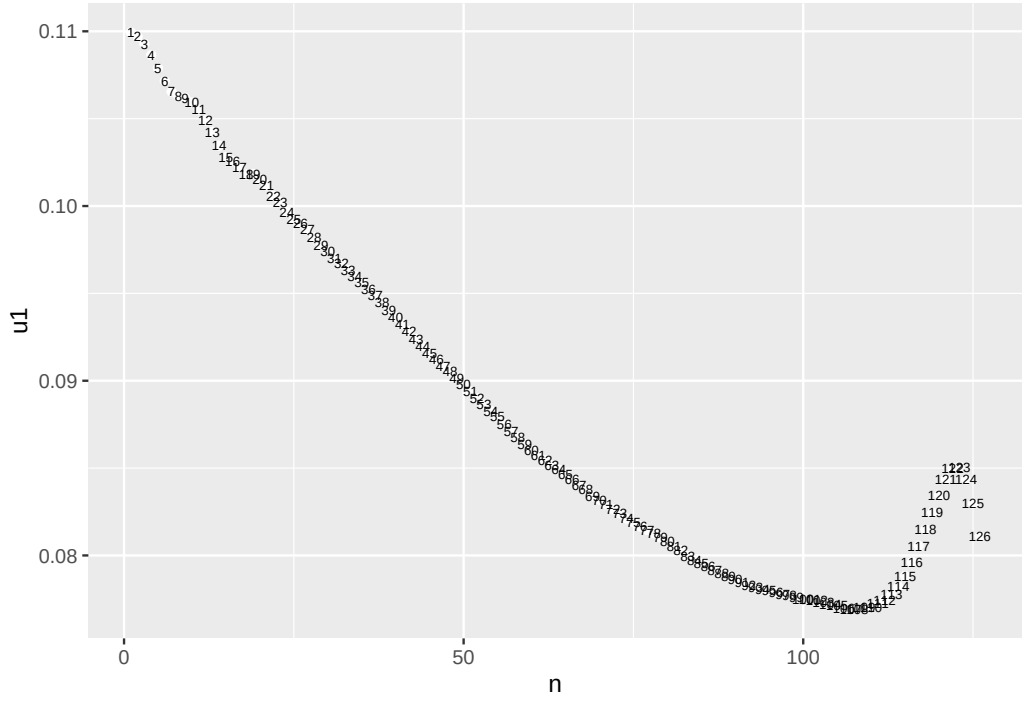
The data file NIRspectra.csv contains near-infrared spectra and viscosity levels for a collection of diesel fuels. Your task is to investigate how the measured spectra can be used to predict the viscosity.

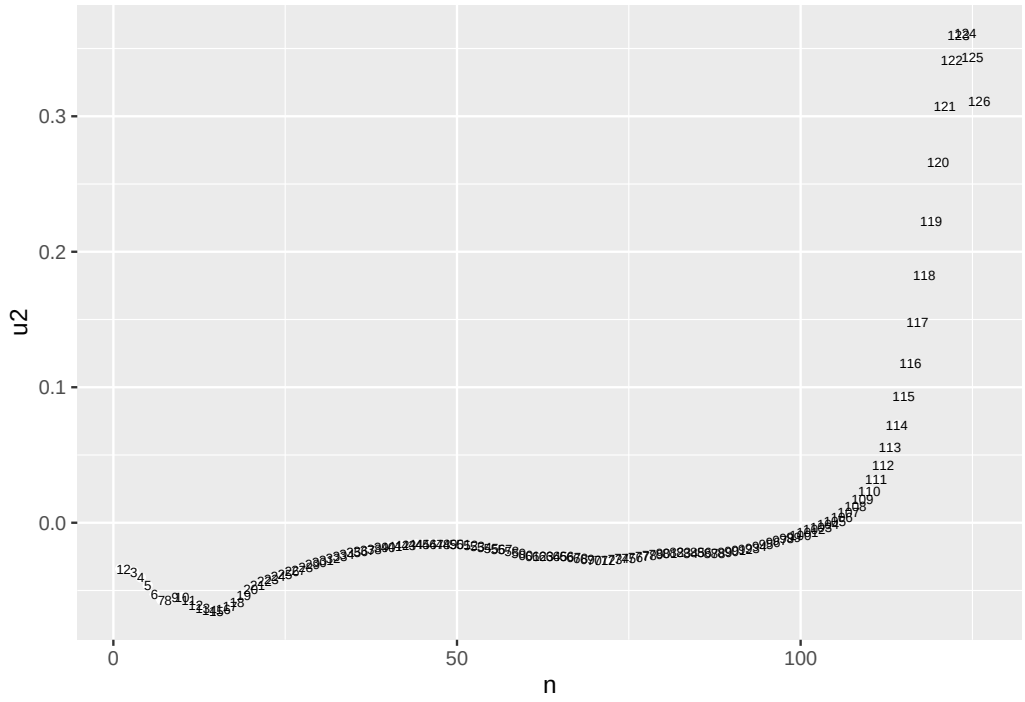
1. Conduct a standard PCA by using the feature space and provide a plot explaining how much variation is explained by each feature. Does the plot show how many PC should be extracted? Select the minimal number of components explaining at least 99% of the total variance. Provide also a plot of the scores in the coordinates (PC1, PC2). Are there unusual diesel fuels according to this plot?



The plot shows that PC1 took 93.3% of variation and PC2 owns 6.3%. With these two components explaining 99.6% of total variance. In this case, the minimal number of components should be 2. Observation 224 and 372 are unusual diesel fuels.

2. Make trace plots of the loadings of the components selected in step 1. Is there any principle component that is explained by mainly a few original features?

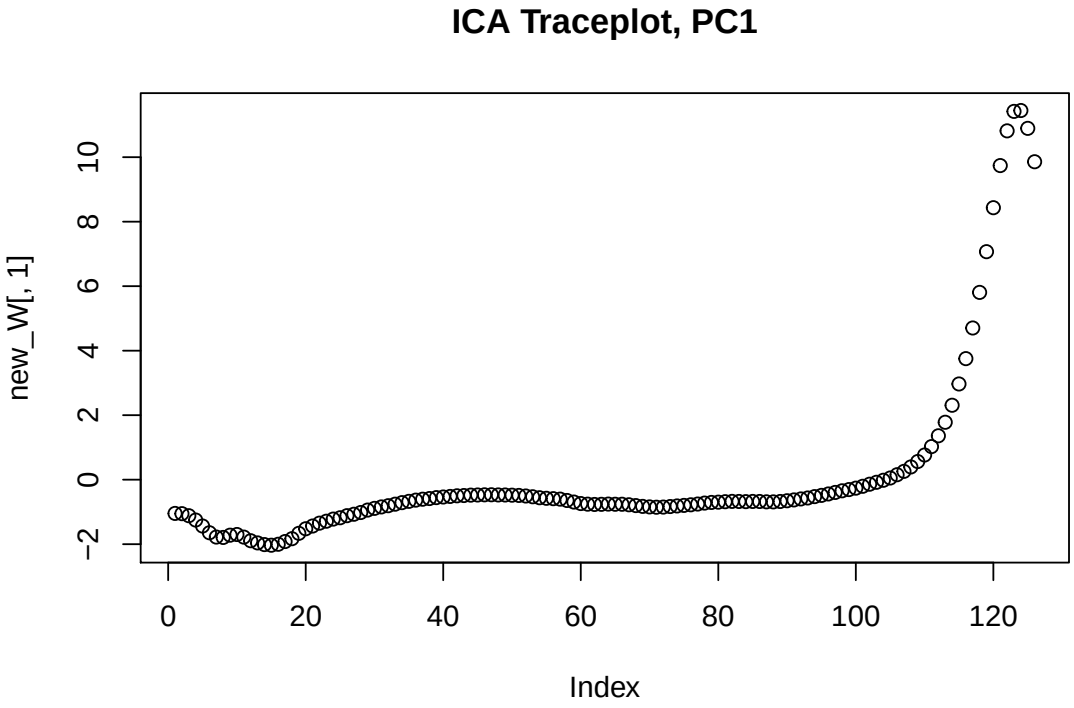


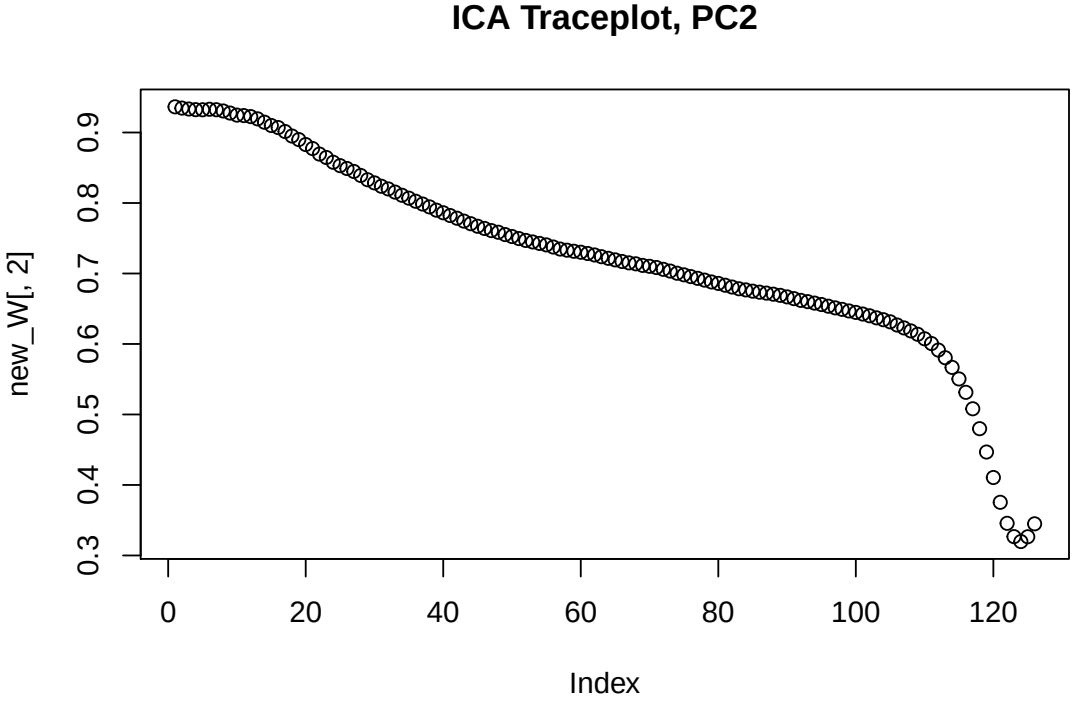


For PC2, it can be explained by feature from 121 to 126.

3.Perform Independent Component Analysis with the number of components selected in step 1 (set seed 12345). Check the documentation for the fastICA method in R and do the following:

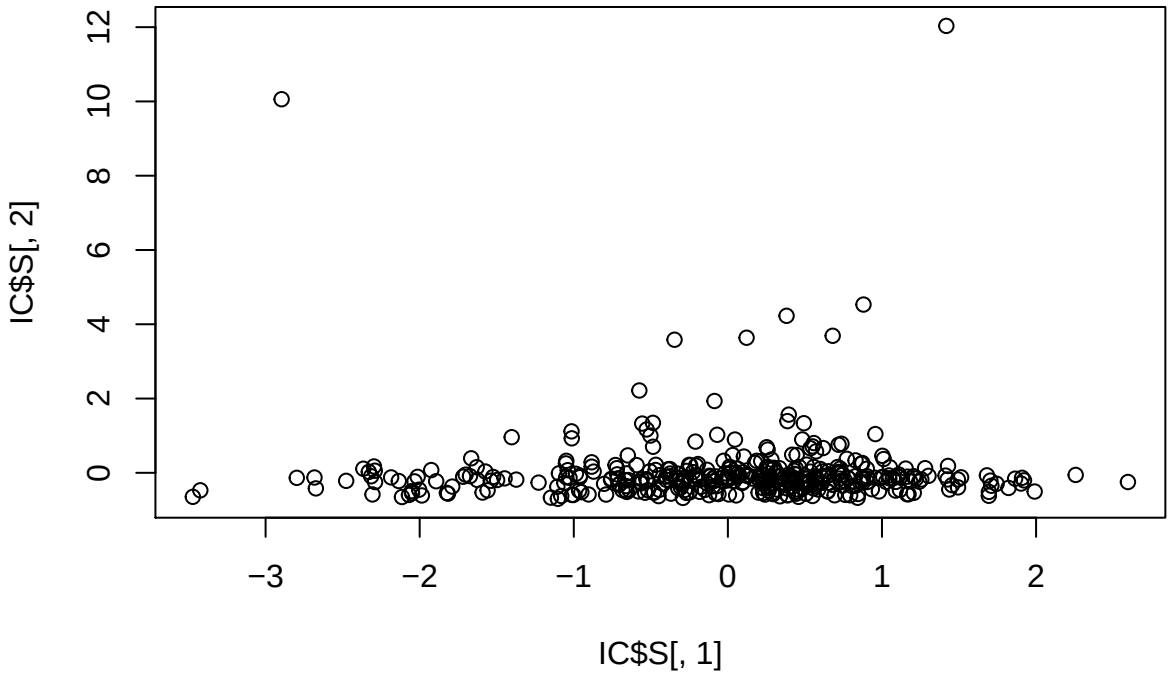
a. Compute $W^T = KW$ and present the columns of W^T in form of the trace plots. Compare with the trace plots in step 2 and make conclusions. What kind of measure is represented by the matrix W^T?





The trace plots are the same with plots in previous step but reversed which means we are getting the same result but using different approaches. W is unmixing matrix and K is pre-whiteninng matrix.

b. Make a plot of the scores of the first two latent features and compare it with the score plot from step 1.



It's the same plot from step 1 but in a different coordinates.

Appendix

Assignment 1

1.

```
RNGversion("3.5.1")
data<-read_xls("data/creditscoring.xls")
data$good_bad<-as.factor(data$good_bad)
n=dim(data)[1]
set.seed(12345)
id=sample(1:n, floor(n*0.5))
train=data[id,]
id1=setdiff(1:n, id)
set.seed(12345)
id2=sample(id1, floor(n*0.25))
valid=data[id2,]
id3=setdiff(id1,id2)
test=data[id3,]
```

2.

```
get_misclassfication_rate<-function(pred,actual){
  cft<-table(pred,actual)
  print(cft)
  rate<-(cft[1,2]+cft[2,1])/sum(cft)*100
  print(rate)
}

create_tree_model<-function(measure){
  tree_model=tree(good_bad~.,data=train,split = c(measure))

  pred_train=predict(tree_model,train,type="class")
  pred_test=predict(tree_model,test,type="class")

  cat("using ",measure,"as measure\n")
  cat("\n Confusion matrix and misclassfication rate for train data")
  get_misclassfication_rate(pred_train,train$good_bad)
  cat("\n Confusion matrix and misclassfication rate for test data\n")
  get_misclassfication_rate(pred_test,test$good_bad)
}

create_tree_model("deviance")
create_tree_model("gini")
```

3.

```
tree_model=tree(good_bad~.,data=train,split = c("deviance"))

trainScore=rep(0,15)
testScore=rep(0,15)
```

```
for(i in 2:15){
  prunedTree=prune.tree(tree_model,best=i)
  pred=predict(prunedTree, newdata=valid, type="tree")
  trainScore[i]=deviance(prunedTree)
  testScore[i]=deviance(pred)
}
ggplot()+geom_point(aes(x=c(2:15),y=trainScore[2:15]),col="red")+
  geom_point(aes(x=c(2:15),y=testScore[2:15]),col="blue")+
  scale_x_discrete(limits=c(2:15))+xlab("leaves")+ylab("score")

prunedTree=prune.tree(tree_model,best=4)
summary(prunedTree)
plot(prunedTree)
text(prunedTree,pretty = 0)

pred_prunedTree=predict(prunedTree,test,type = "class")
get_misclassfication_rate(pred_prunedTree,test$good_bad)
```

4.

```
library(e1071)
naive_bayes_model=naiveBayes(good_bad~.,train)

pred_bayes_train=predict(naive_bayes_model,train)
cat("\n Confusion matrix and misclassfication rate for train data")
get_misclassfication_rate(pred_bayes_train,train$good_bad)

pred_bayes_test=predict(naive_bayes_model,test)
cat("\n Confusion matrix and misclassfication rate for test data")
get_misclassfication_rate(pred_bayes_test,test$good_bad)
```

5.

```
prunedTree=prune.tree(tree_model,best=4)
pred_prunedTree=predict(prunedTree,test)
pred_bayes_train=predict(naive_bayes_model,train,type="raw")
pred_bayes_test=predict(naive_bayes_model,test,type="raw")

TPR_tree<-c()
FPR_tree<-c()

for(pi in seq(0.05,0.95,0.05)){
  res<-ifelse(pred_prunedTree[,2]>pi,1,0)
  actual<-ifelse(test$good_bad=="good",1,0)

  factor(res,levels = c(0,1))
  cft<-table(factor(res,levels = c(0,1)),factor(actual))
  TP <- cft[2, 2]
  TN <- cft[1, 1]
  FP <- cft[2, 1]
  FN <- cft[1, 2]
```

```
  current_TPR<-TP/(TP+FN)
  current_FPR<-FP/(FP+TN)
  TPR_tree<-c(TPR_tree,current_TPR)
  FPR_tree<-c(FPR_tree,current_FPR)
}
TPR_nb<-c()
FPR_nb<-c()
for(pi in seq(0.05,0.95,0.05)){
  res<-ifelse(pred_bayes_test[,2]>pi,1,0)
  actual<-ifelse(test$good_bad=="good",1,0)

  factor(res,levels = c(0,1))
  cft<-table(factor(res,levels = c(0,1)),factor(actual))
  TP <- cft[2, 2]
  TN <- cft[1, 1]
  FP <- cft[2, 1]
  FN <- cft[1, 2]
  current_TPR<-TP/(TP+FN)
  current_FPR<-FP/(FP+TN)
  TPR_nb<-c(TPR_nb,current_TPR)
  FPR_nb<-c(FPR_nb,current_FPR)
}
ggplot()+geom_line(aes(x=FPR_nb,y=TPR_nb),col="red")+geom_point(aes(x=FPR_nb,y=TPR_nb),col="red")+geom_
```

6.

```
loss_matrix<-matrix(c(0,10,1,0),ncol=2)
res_train<-c()
for(i in c(1:dim(pred_bayes_train)[1])){
  if((pred_bayes_train[i,1]/pred_bayes_train[i,2])>(loss_matrix[2,1]/loss_matrix[1,2])){
    res_train<-c(res_train,0)
  }else{
    res_train<-c(res_train,1)
  }
}
cat("\n Confusion matrix and misclassfication rate for train data\n")
get_misclassfication_rate(res_train,train$good_bad)

res_test<-c()
for(i in c(1:dim(pred_bayes_test)[1])){
  if((pred_bayes_test[i,1]/pred_bayes_test[i,2])>(loss_matrix[2,1]/loss_matrix[1,2])){
    res_test<-c(res_test,0)
  }else{
    res_test<-c(res_test,1)
  }
}
cat("\n Confusion matrix and misclassfication rate for test data\n")
get_misclassfication_rate(res_test,test$good_bad)
```

Assignment 3

1.

```r
RNGversion("3.5.1")
data<-read.csv("data/State.csv",header = TRUE,sep=";",dec = ",")
set.seed(12345)
data<-data[order(data$MET),]
```

2.

```r
tree_mod<-tree(EX~MET,data=data,control = tree.control(minsize = 8,nobs = nrow(data)))
cv.res=cv.tree(tree_mod)
plot(cv.res$size, cv.res$dev, type="b", col="red")
plot(log(cv.res$size), cv.res$dev, type="b", col="red")

prunedTree=prune.tree(tree_mod,best=3)
pred=predict(prunedTree,data=data)

ggplot()+geom_line(aes(x=c(1:length(pred)),y=pred),col="red")+geom_point(aes(x=c(1:length(data$EX)),y=da
res=data$EX-pred
hist(res)

mean(res)
sd(res)
```

3.

```r
f=function(data,ind){
  data1=data[ind,]
  tree_mod<-tree(EX~MET,data=data1,control = tree.control(minsize = 8,nobs = nrow(data)))
  prunedTree=prune.tree(tree_mod,best=3)
  pred=predict(prunedTree,data)
  return(pred)
}

boot_res=boot(data,f,R=1000)
CE<-envelope(boot_res,level = 0.95)


tree_mod<-tree(EX~MET,data=data,control = tree.control(minsize = 8,nobs = nrow(data)))
prunedTree=prune.tree(tree_mod,best=3)
pred=predict(prunedTree,data)

CI_band=data.frame(upper=CE$point[1,],lower=CE$point[2,],EX=data$EX,MET=data$MET,pred=pred)
ggplot(data=CI_band)+geom_point(aes(x=MET,y=EX))+geom_line(aes(x=MET,y=pred),col="blue")+geom_ribbon(aes
```

4.

```r
regression_tree=tree(EX~MET,data=data,control = tree.control(minsize = 8,nobs = nrow(data)))
mle=prune.tree(regression_tree,best=3)

rng<-function(data,mle){
  data1=data.frame(EX=data$EX,MET=data$MET)
  data1$EX=rnorm(length(data1$EX),predict(mle,data1),sd=sd(residuals(mle)))
```

```
  return(data1)
}
f1<-function(data1){
  new_tree<-tree(EX~MET,data=data1,control = tree.control(minsize = 8,nobs = nrow(data)))
  new_prunedTree=prune.tree(new_tree,best=3)
  new_pred<-predict(new_prunedTree,data)
  new_pred<-rnorm(length(new_pred),mean=new_pred,sd=sd(residuals(mle)))
  return(new_pred)
}

boot_res1=boot(data,statistic = f1,R=1000,mle = prunedTree,ran.gen = rng,sim="parametric")

CE<-envelope(boot_res1,level = 0.95)

tree_mod<-tree(EX~MET,data=data,control = tree.control(minsize = 8,nobs = nrow(data)))
prunedTree=prune.tree(tree_mod,best=3)
pred=predict(prunedTree,data)

CI_band=data.frame(upper=CE$point[1,],lower=CE$point[2,],EX=data$EX,MET=data$MET,pred=pred)
ggplot(data=CI_band)+geom_point(aes(x=MET,y=EX))+geom_line(aes(x=MET,y=pred),col="blue")+geom_ribbon(aes
```

Assignment 4

1.

```
#eigenvalues
lambda=res$sdev^2
lambda=lambda/sum(lambda)*100


#df<-data.frame(PC=1:5,va=lambda[1:5])
df<-data.frame(PC=c(1:length(lambda)),va=lambda)
ggplot(data=df,aes(x=PC,y=va))+geom_bar(stat = "identity")+geom_text(mapping=aes(label=round(va,1)),size

df1<-data.frame(x=res$x[,1],y=res$x[,2],n=1:length(res$x[,1]))
#ggplot(data=data,aes(x=res$x[,1],y=res$x[,2]))+geom_point()
#plot(x=res$x[,1],y=res$x[,2])
ggplot(data=df1,aes(x=x,y=y))+geom_point(col="white")+geom_text(label=(df1$n),size=2,col="black")
```

2.

```
u=res$rotation
df<-data.frame(u1=u[,1],u2=u[,2],n=c(1:length(u[,1])))
ggplot(df,aes(x=n,y=u1))+geom_point(col="white")+geom_text(label=df$n,size=2)
ggplot(df,aes(x=n,y=u2))+geom_point(col="white")+geom_text(label=df$n,size=2)
```

3.

```
IC<-fastICA(data,n.comp = 2)
new_W<-IC$K%*%IC$W
```

```r
plot(new_W[,1],main="ICA Traceplot, PC1")
plot(new_W[,2],main="ICA Traceplot, PC2")
plot(IC$S[,1],IC$S[,2])
```