Compulsory exercise 2: Group 5

TMA4268 Statistical Learning V2022

Aleksander Johnsen Solberg and Gjermund Oscar Lyckander

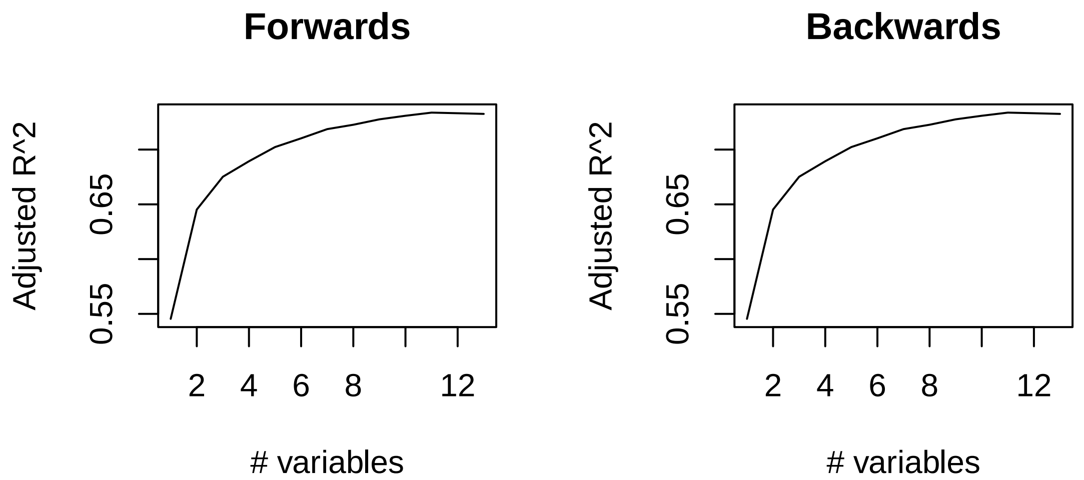
04 April, 2022

Problem 1

```
set.seed(1)
boston <- scale(Boston, center = T, scale =  T)

train.ind = sample(1:nrow(boston), 0.8 * nrow(boston))
boston.train = data.frame(boston[train.ind, ])
boston.test = data.frame(boston[-train.ind, ])
```

a)

```
set.seed(1)
forward_stepwise = regsubsets(medv ~ ., data = boston.train, nvmax = 13, method = 'forward')
backward_stepwise = regsubsets(medv ~ ., data = boston.train, nvmax = 13, method = 'backward')

forward_stepwise_summary = summary(forward_stepwise)
backward_stepwise_summary = summary(backward_stepwise)
#forward_stepwise_summary
#backward_stepwise_summary

par(mfrow=c(1,2))
plot(forward_stepwise_summary$adjr2, xlab = '# variables', ylab = 'Adjusted R^2', type='l', main='Forwa
plot(forward_stepwise_summary$adjr2, xlab = '# variables', ylab = 'Adjusted R^2', type='l', main='Backwa
```



b)

```r
forward_stepwise_summary$outmat
```

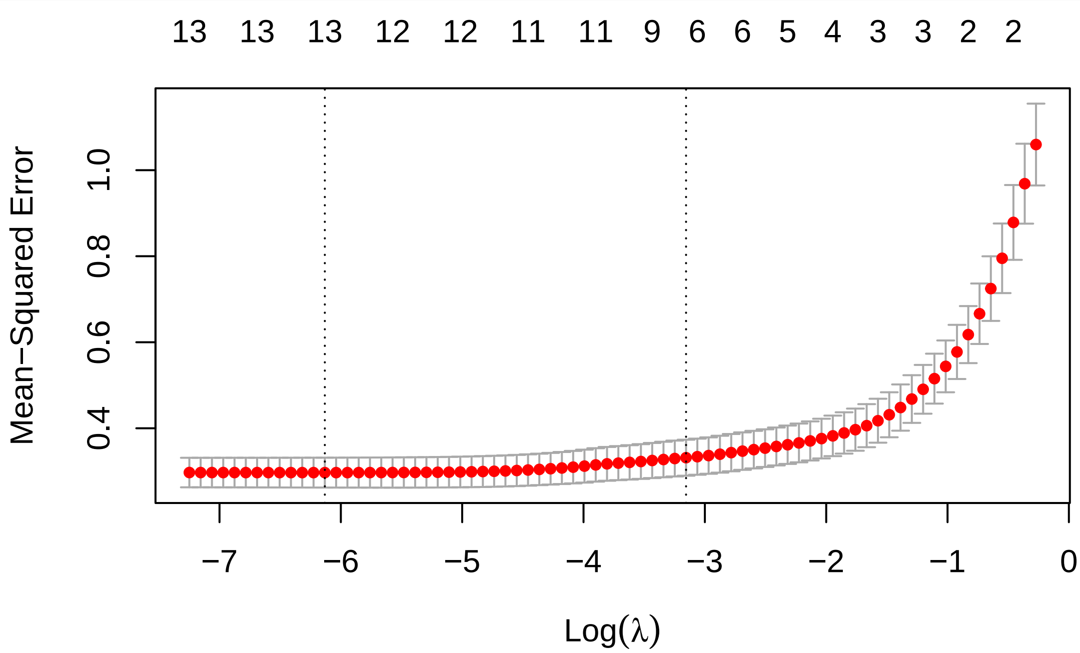
```
##           crim zn  indus chas nox rm  age dis rad tax ptratio black lstat
## 1  ( 1 )  " "  " " " "   " "  " " " " " " " " " " " " " "     " "   "*"  
## 2  ( 1 )  " "  " " " "   " "  " " "*" " " " " " " " " " "     " "   "*"  
## 3  ( 1 )  " "  " " " "   " "  " " "*" " " " " " " " " "*"     " "   "*"  
## 4  ( 1 )  " "  " " " "   " "  " " "*" " " "*" " " " " "*"     " "   "*"  
## 5  ( 1 )  " "  " " " "   " "  " " "*" " " "*" " " " " "*"     "*"   "*"  
## 6  ( 1 )  " "  " " " "   " "  "*" "*" " " "*" " " " " "*"     "*"   "*"  
## 7  ( 1 )  " "  " " " "   "*"  "*" "*" " " "*" " " " " "*"     "*"   "*"  
## 8  ( 1 )  " "  " " " "   "*"  "*" "*" " " "*" "*" " " "*"     "*"   "*"  
## 9  ( 1 )  " "  " " " "   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"  
## 10  ( 1 ) " "  "*" " "   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"  
## 11  ( 1 ) "*"  "*" " "   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"  
## 12  ( 1 ) "*"  "*" "*"   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"  
## 13  ( 1 ) "*"  "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"  
```

We choose the predictors ‘rm, ’dis’, ‘ptratio’ and ‘lstat’.

c)

i)

```r
set.seed(1)
y = boston.train$medv
x = data.matrix(boston.train[, -14])

cv_lasso = cv.glmnet(x, y, alpha=1, nfolds=5)
plot(cv_lasso)
```



ii)

```
lasso_best_lambda = cv_lasso$lambda.min
lasso_best_lambda
```

```
## [1] 0.002172032
```

iii)

```
coef(glmnet(x, y, alpha=1, lambda=lasso_best_lambda))
```

```
## 14 x 1 sparse Matrix of class "dgCMatrix"
##                        s0
## (Intercept)  0.023622904
## crim        -0.081992849
## zn           0.094717791
## indus        0.002619428
## chas         0.087341100
## nox         -0.175365927
## rm           0.312648954
## age         -0.011212120
## dis         -0.317143728
## rad          0.270168177
## tax         -0.207314714
## ptratio     -0.204052488
## black        0.102877803
## lstat       -0.428298373
```

d)

TRUE, FALSE, FALSE, TRUE

Problem 2

a)

b)

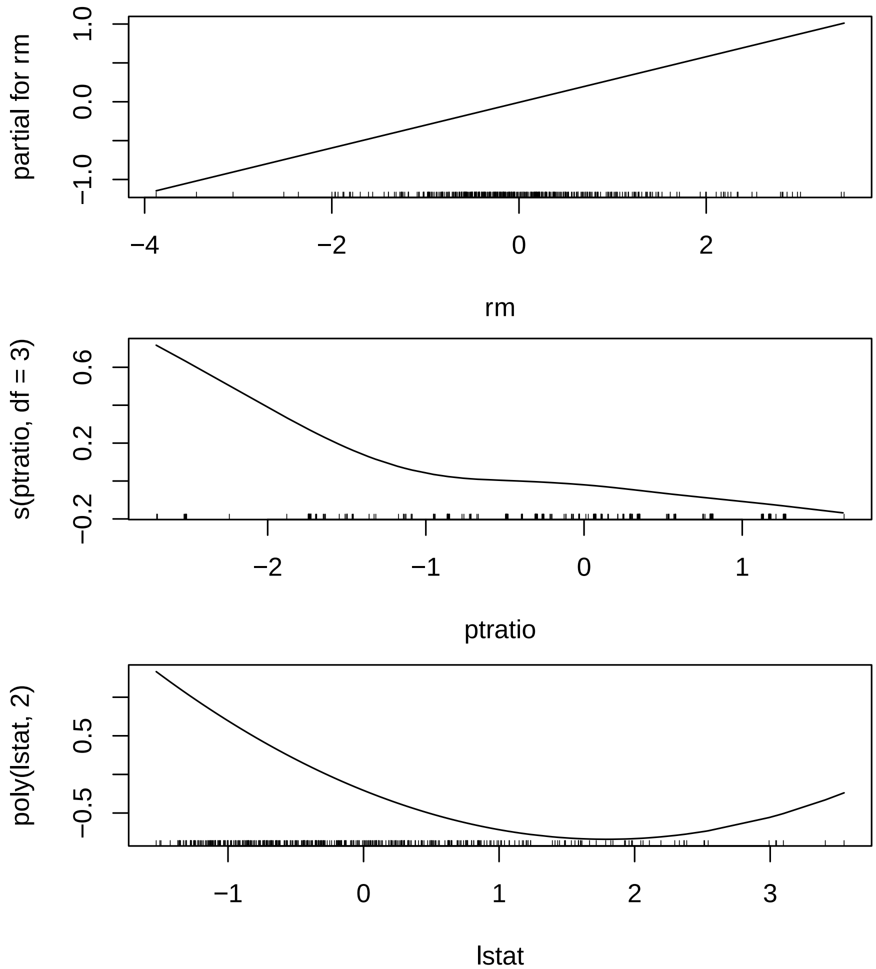
Problem 3

a)

TRUE, FALSE, FALSE, TRUE

b)

```
additive_model = gam(medv ~ rm + s(ptratio, df=3) + poly(lstat, 2), data=boston.train)
plot(additive_model)
```



Problem 4

a)

FALSE, TRUE, TRUE, TRUE

b)

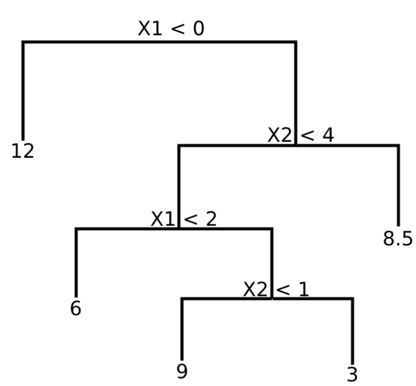


Figure 1: Tree

c)

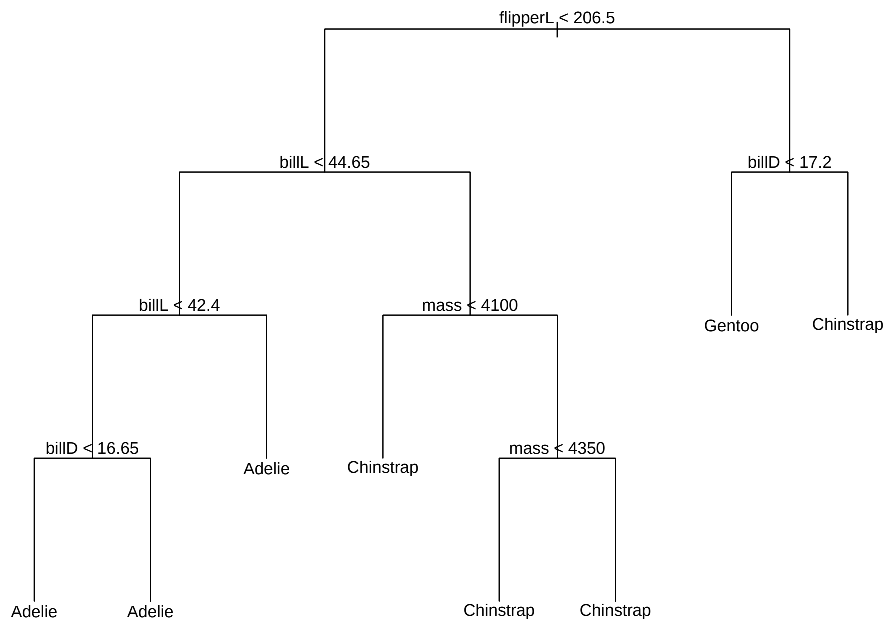
```
library(tidyverse)
library(palmerpenguins) # Contains the data set "penguins".
data(penguins)

names(penguins) <- c("species", "island", "billL", "billD", "flipperL", "mass", "sex", "year")

Penguins_reduced <- penguins %>% dplyr::mutate(mass = as.numeric(mass), flipperL = as.numeric(flipperL)

# We do not want "year" in the data (this will not help for future predictions)
Penguins_reduced <- Penguins_reduced[,-c(8)]

set.seed(4268)
# 70% of the sample size for training set
training_set_size <- floor(0.7 * nrow(Penguins_reduced))
train_ind <- sample(seq_len(nrow(Penguins_reduced)), size = training_set_size)
train <- Penguins_reduced[train_ind, ]
test <- Penguins_reduced[-train_ind, ]
```

i)

```
penguin.tree = tree(formula=species ~ ., data=train, split='gini' )
summary(penguin.tree)
```

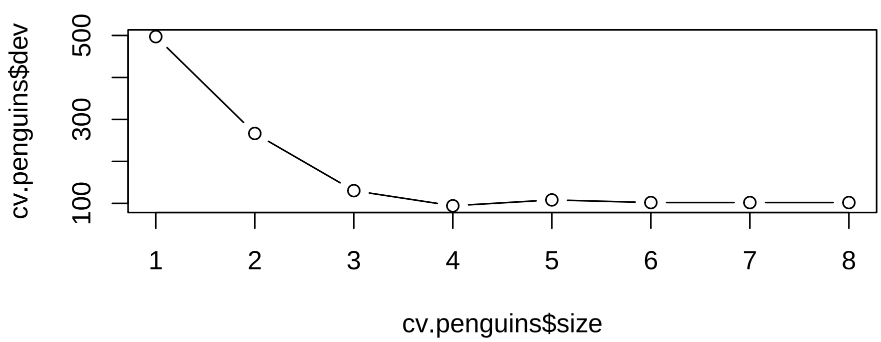
```
##
## Classification tree:
## tree(formula = species ~ ., data = train, split = "gini")
## Variables actually used in tree construction:
## [1] "flipperL" "billL"    "billD"    "mass"
## Number of terminal nodes:  8
## Residual mean deviance:  0.1869 = 42.06 / 225
## Misclassification error rate: 0.04292 = 10 / 233
```

```
plot(penguin.tree, type='uniform')
text(penguin.tree, pretty=0)
```



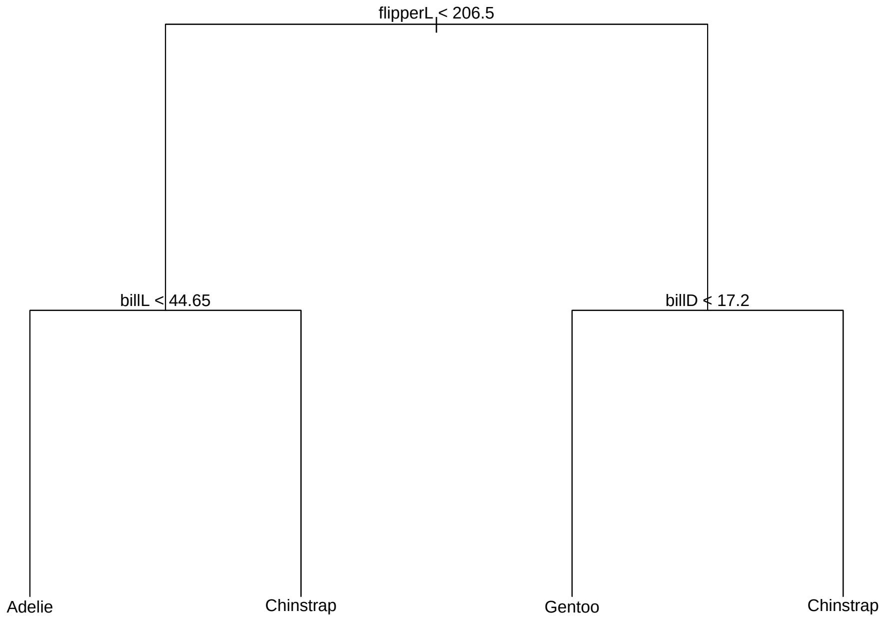
ii)

```
set.seed(123)
cv.penguins = cv.tree(penguin.tree, K=10)
#cv.penguins$dev
plot(cv.penguins$dev ~ cv.penguins$size, type='b')
```



iii)

```
prune.penguins = prune.tree(penguin.tree, best=4)
plot(prune.penguins, type='uniform')
text(prune.penguins, pretty=0)
```




```
tree.predict = predict(prune.penguins, test, type='class')
misclass = table(tree.predict, test$species)
misclass
```

```
##
## tree.predict Adelie Chinstrap Gentoo
##    Adelie        42         5      1
##    Chinstrap      0        15      0
##    Gentoo         0         0     37
```

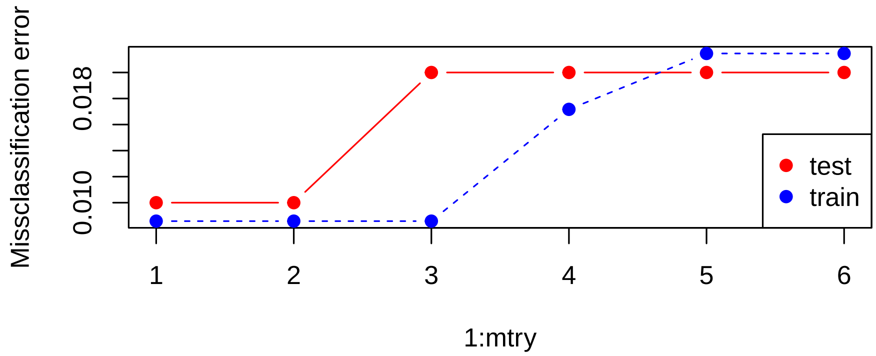
```
1-sum(diag(misclass))/sum(misclass)
```

```
## [1] 0.06
```

d)

Using random forest. Trying different choices for variable mtry, and plotting the misclassification errors.

```
set.seed(1001)

train.err = double(6)
test.err = double(6)

for(mtry in 1:6) {
  rf.penguins = randomForest(species ~ ., data=train, mtry=mtry, ntree=500)
  train.err[mtry] = rf.penguins$err.rate[500]

  rf.predict = predict(rf.penguins, newdata=test, type='class')
  misclass = table(rf.predict, test$species)
  misclass
  test.err[mtry] = 1-sum(diag(misclass))/sum(misclass)
}

matplot(1:mtry, cbind(test.err, train.err), pch=19, type='b', ylab='Missclassification error', col=c('r
legend('bottomright', legend=c('test', 'train'), pch=19, col=c('red','blue'))
```



We find that a good choice for mtry is 2, which also approximately corresponds to the square root of the number of covariates.

```
rf.penguins = randomForest(species ~ ., data=train, mtry=2, ntree=500)

rf.predict = predict(rf.penguins, newdata=test, type='class')
misclass = table(rf.predict, test$species)
misclass
```

```
##
## rf.predict  Adelie Chinstrap Gentoo
##   Adelie        42         2      0
##   Chinstrap      0        18      0
##   Gentoo         0         0     38
```

```
1-sum(diag(misclass))/sum(misclass)
```

```
## [1] 0.02
```

```
importance(rf.penguins)
```

```
##          MeanDecreaseGini
## island         17.3964364
## billL          52.2639236
## billD          25.0545916
## flipperL       37.1186075
## mass           14.5237520
## sex             0.9137075
```

We see that the two most influential variables are 'billL' and 'flipperL'.

Problem 5