Statistical Learning (5454) - Assignment 3

Matthias Hochholzer, Lukas Pirnbacher, Anne Valder

Due: 2024-05-20

Exercise 1

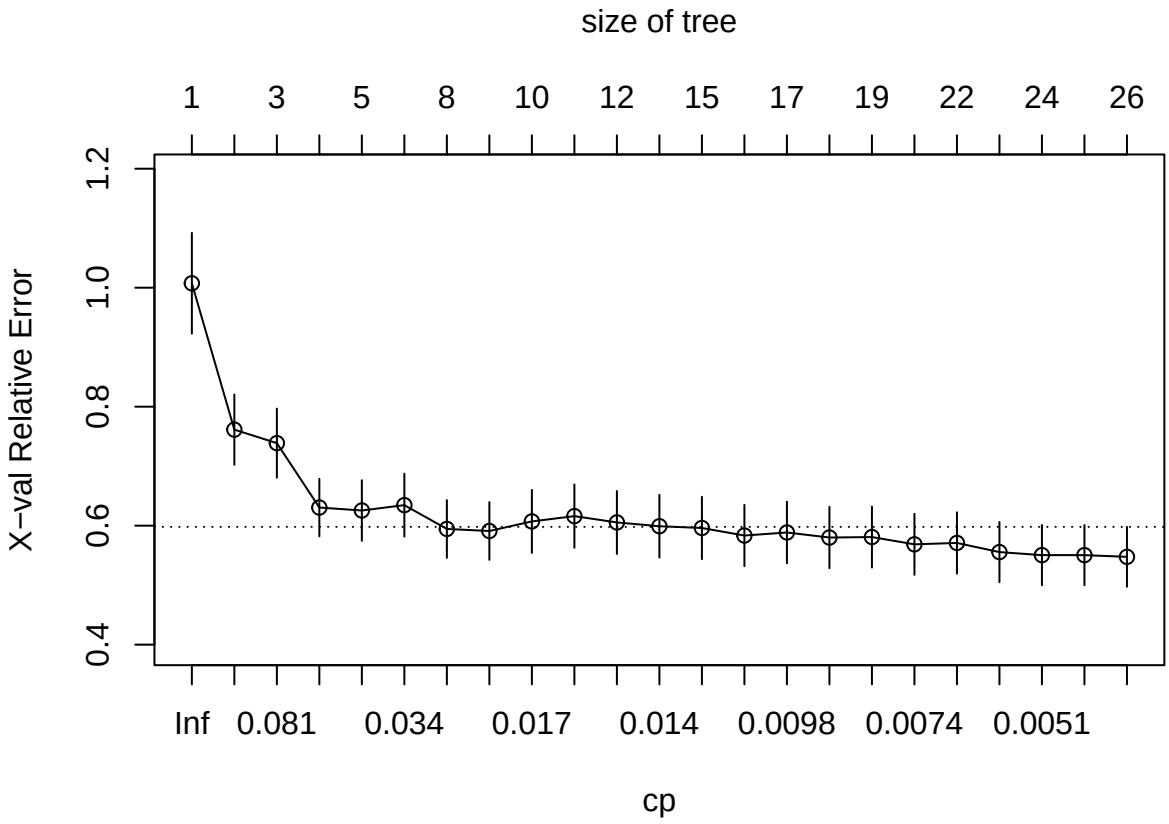
We load the data set `Carseats` from package **ISLR2**. It is a simulated data set containing sales of child car seats at 400 different stores. We then randomly select 280 observations as training data and use the remaining ones as test data (70:30 split).

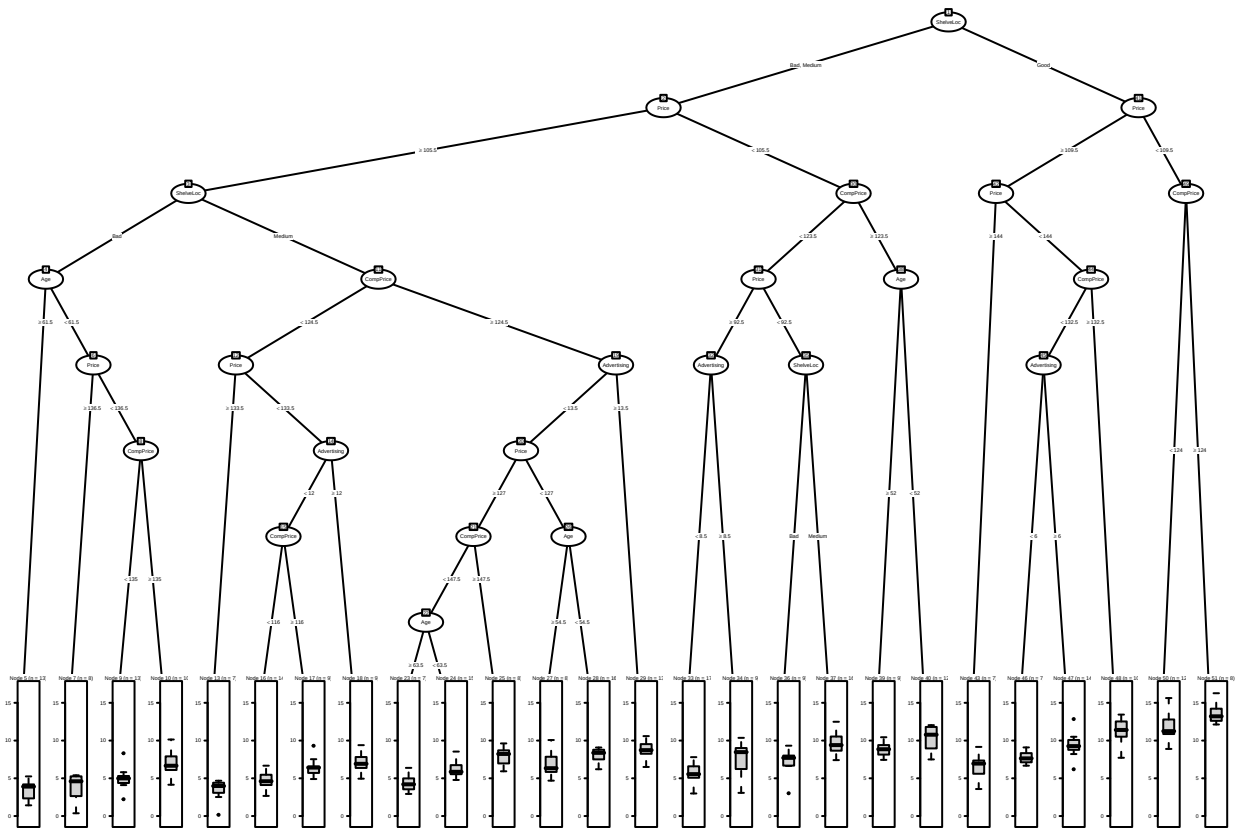
Next, we fit a regression tree to the training set. We set the complexity parameter to $cp = 10^{-4}$ in order to initially grow a rather large tree, which we intend to prune later on.

```
##
## Regression tree:
## rpart(formula = Sales ~ ., data = train, method = "anova", control = list(cp = 1e-04))
##
## Variables actually used in tree construction:
## [1] Advertising Age         CompPrice   Price
## [5] ShelveLoc
##
## Root node error: 2227/280 = 8
##
## n= 280
##
##        CP nsplit rel error xerror  xstd
## 1  0.2546      0      1.00   1.01 0.085
## 2  0.0922      1      0.75   0.76 0.059
## 3  0.0709      2      0.65   0.74 0.058
## 4  0.0432      3      0.58   0.63 0.048
## 5  0.0360      4      0.54   0.63 0.051
## 6  0.0323      5      0.50   0.63 0.053
## 7  0.0243      7      0.44   0.59 0.049
## 8  0.0175      8      0.41   0.59 0.048
## 9  0.0159      9      0.40   0.61 0.053
## 10 0.0156     10      0.38   0.62 0.053
## 11 0.0141     11      0.37   0.61 0.053
## 12 0.0135     12      0.35   0.60 0.053
## 13 0.0127     14      0.32   0.60 0.052
## 14 0.0105     15      0.31   0.58 0.052
## 15 0.0092     16      0.30   0.59 0.052
## 16 0.0089     17      0.29   0.58 0.052
## 17 0.0078     18      0.28   0.58 0.051
## 18 0.0069     20      0.27   0.57 0.051
## 19 0.0068     21      0.26   0.57 0.052
## 20 0.0053     22      0.25   0.56 0.051
## 21 0.0049     23      0.25   0.55 0.051
```

```
## 22 0.0037     24      0.24   0.55 0.051
## 23 0.0001     25      0.24   0.55 0.050
```

A plot of the complexity parameter table and the plot of the tree can be seen below.





The fitted tree has 26 terminal nodes (25 splits). Overall, 5 of the 10 available predictors are used in the construction of the tree (shelf location, price, price of competitor, average age of the local population, advertising budget). Since the tree is rather long we do further interpretations with the pruned tree.

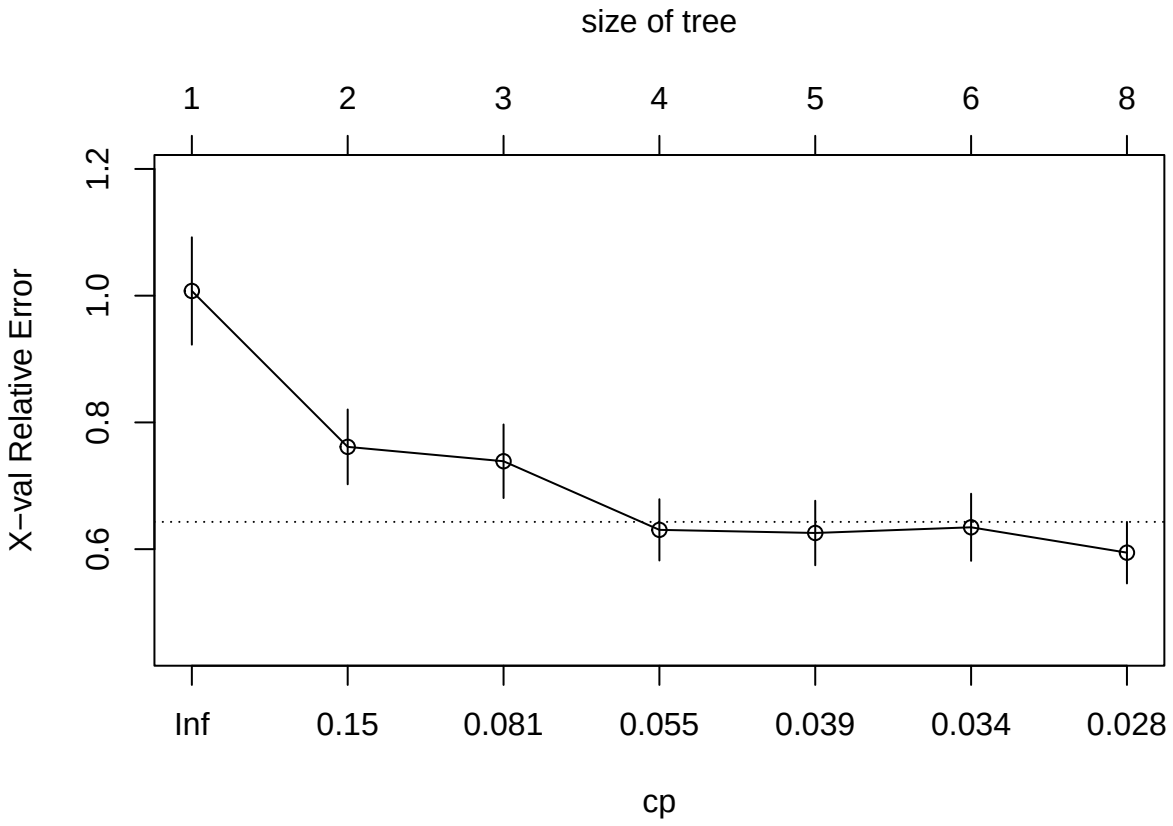
Given the loose stopping criterion we chose, we are in principle facing the risk of overiftting. In order to solve this problem, we prune the tree later on. However, as the cross-validated errors (`xerror`) are not increasing for smaller values of `cp`, we believe that overfitting may not be a problem. For now, we calculate the test MSE for the full tree, which is given by

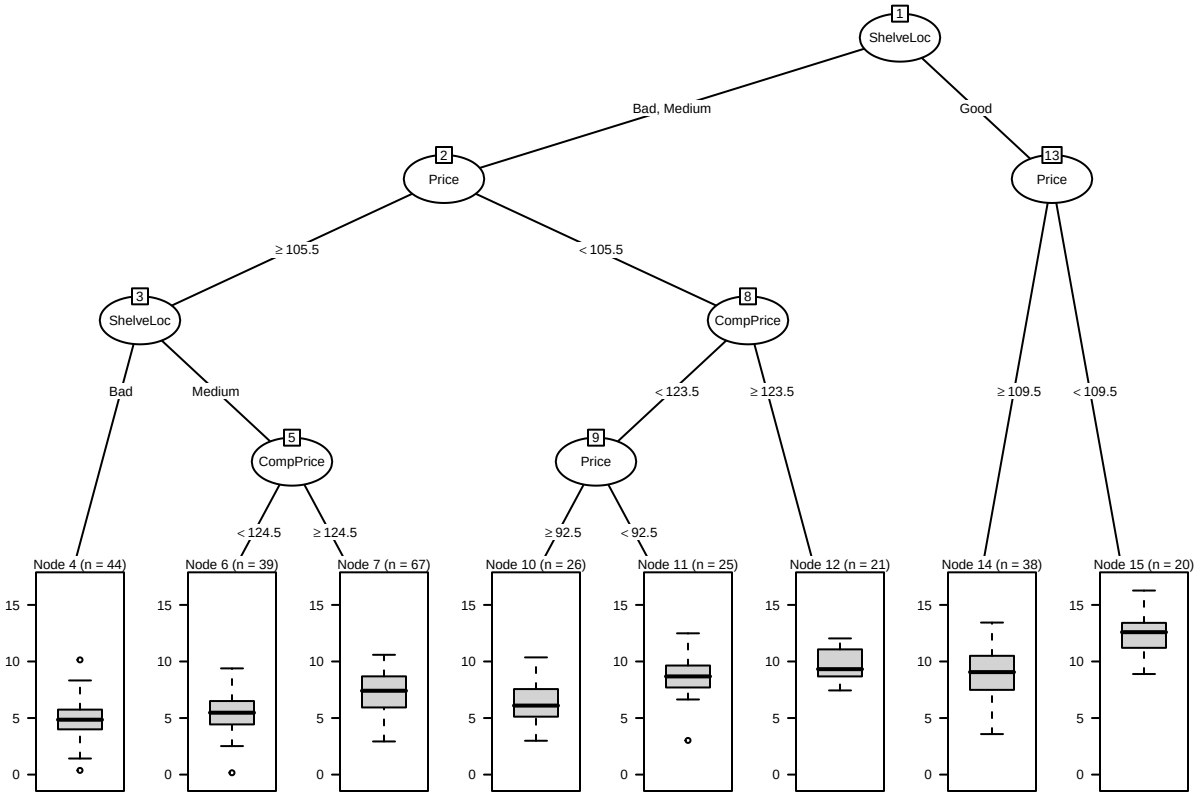
```
## [1] 3.519
```

We now use 10-fold cross-validation (which is the default in `rpart()`) in order to determine the optimal level of tree complexity. We do so by applying the 1-SE rule, i.e. we choose the largest complexity parameter such that the corresponding cross-validated error is smaller than the minimum error plus one standard deviation.

```
##
## Regression tree:
## rpart(formula = Sales ~ ., data = train, method = "anova", control = list(cp = 1e-04))
##
## Variables actually used in tree construction:
## [1] CompPrice Price     ShelveLoc
##
## Root node error: 2227/280 = 8
##
## n= 280
##
##       CP nsplit rel error xerror  xstd
## 1 0.255      0      1.00   1.01 0.085
```

```
## 2 0.092      1      0.75   0.76 0.059
## 3 0.071      2      0.65   0.74 0.058
## 4 0.043      3      0.58   0.63 0.048
## 5 0.036      4      0.54   0.63 0.051
## 6 0.032      5      0.50   0.63 0.053
## 7 0.024      7      0.44   0.59 0.049
```





The pruned tree has 8 terminal nodes (7 splits). In contrast to the full tree, the advertising budget as well as the average age of the local population are no longer used to construct the tree. The first split is made according to the quality of the shelving location for the car seats. It splits into *Bad & Medium* and *Good*. The second split is based on the car seat price for a *Bad & Medium* shelving location. The third split distinguishes between *Bad* and *Medium* shelving location for products with higher prices. For the medium shelving location there is another split based on the price of the competitor. For low-price products with *Bad & Medium* location there are two more splits based on the competitor's and the own price. Finally, for products with *Good* shelving location, a split is made based upon the price. Looking at the boxplots at each terminal node, it seems that even the last split was still important.

Overall, the pruned tree displays quite intuitive patterns: Sales increase with the quality of the shelving location, increase for higher prices of the competitors and decrease as the price of the car seats increases. It is interesting, though, that the price of competitors is a relevant explanatory factor only for car seats in bad or medium shelving locations.

The test MSE with pruning is

```
## [1] 4.353
```

and therefore higher than for the initial tree. Hence, pruning the tree does not improve the test MSE, which indicates that overfitting was actually not too much of a problem. However, while pruning has not improved the predictive performance of the tree, it has significantly simplified the interpretation.

Exercise 2

We draw 100 observations from four independent variables $X_1, \ldots, X_4$, where

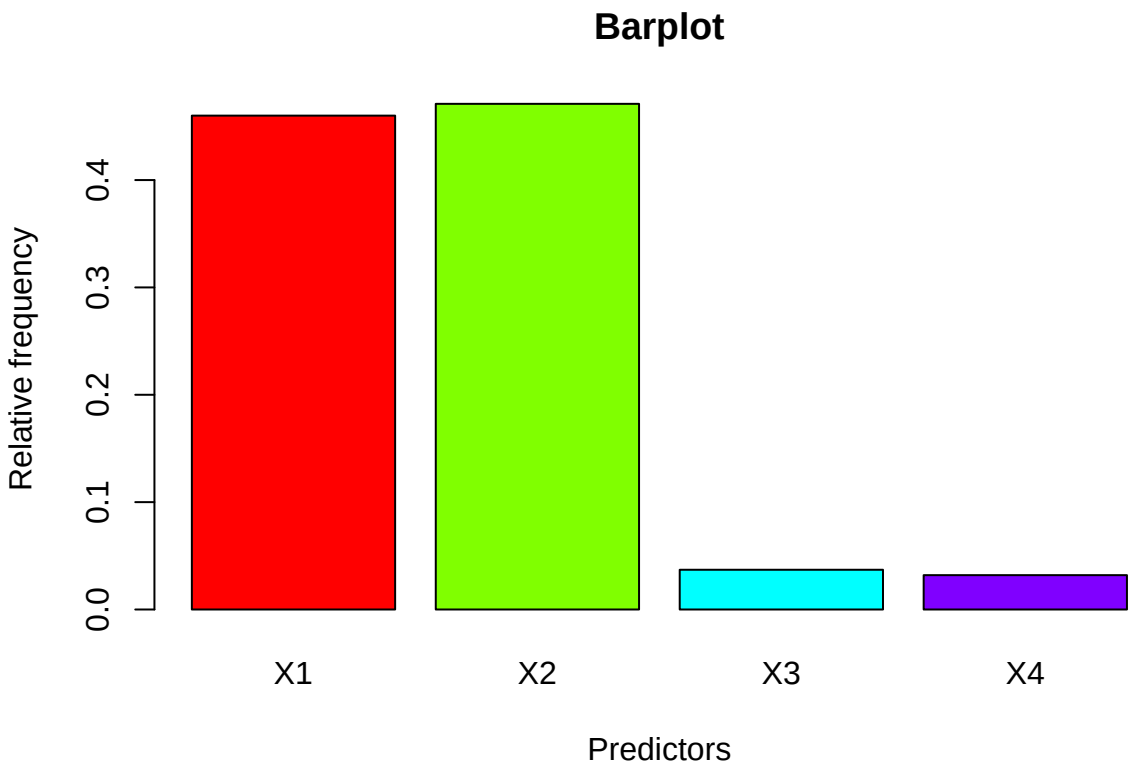
- X_1 follows a standard uniform distribution,
- X_2 follows a standard normal distribution,
- X_3 follows a Bernoulli distribution with success probability $\pi = 0.5$,
- X_4 follows a Bernoulli distribution with success probability $\pi = 0.1$.

We now repeat the following 1000 times:

- Draw a dependent variable y from a standard normal distribution, which is independent of the four independent variables.
- Fit a tree stump, i.e. a tree which contains only one split.
- Determine which variable was used for splitting.

Below is the table of relative frequencies, displaying how often each of the variables was selected for splitting. We also provide a barplot to visualize the findings.

```
##
##    X1    X2    X3    X4
## 0.460 0.471 0.037 0.032
```



The probability of including a particular independent variable is not the same. X_2 has the highest probability, closely followed by X_1. The inclusion probabilities for X_3 and X_4 are much lower. Hence, while all predictors are independent of y, the two continuous variables (X_1, X_2) are much more likely to be selected as splitting variable. This should not come as a surprise, as a continuous split variable allows for more flexibility in

partitioning the X-space. By choosing different split points, one can define a wide variety of different pairs of half-planes. For a binary split variable, in contrast, there is only a single possible partition, with all observations where the split variable is equal to 1 being part of one region and the remaining observations being part of the other region. Thus, the probability of finding a partition of X-space such that the sum of squared residuals is (by chance) small, is much higher for continuous split variables.

Exercise 3

We assume the following data generating process

$$Y = X + \epsilon,$$

with $X \sim N(0, 1)$ and $\epsilon \sim N(0, 1)$ independent. In addition, 20 covariates $Z_1, \ldots, Z_{20}$ are given with

$$Z_i \sim \sqrt{0.9}X + \epsilon_{Z_i},$$

where $\epsilon_{Z_i} \sim N(0, 0.1)$.

We draw training data with 30 observations and test data with 10,000 observations from the data generating process, including the additional covariates $\mathbf{Z}$.

We then create 100 bootstrap samples of size 30 from the training data by drawing with replacement.

To each bootstrap sample we fit:

1. a regression tree,
2. the null model with predicted value equal to the observed empirical mean of Y,
3. a linear model including linear effects for X and all Z variables and
4. a linear model potentially including linear effects for X and all Z variables, but using model selection with the AIC to select a suitable model starting from the null model.

We determine the predicted values on the test data for the bagged model estimator by calculating the average predictions over the 100 trees fitted to the bootstrap samples, the 100 null models, the 100 linear models including all linear effects and the 100 linear models based on model selection.

Finally, we determine the mean squared error (MSE) of the four bagged model estimators on the test sample of size 10,000.

```
##   Model   MSE
## 1  Tree 1.211
## 2  Null 2.036
## 3    LM 4.436
## 4   AIC 1.765
```

The lowest MSE is achieved by the regression tree, followed by the model selected by AIC and the null model. The highest MSE is obtained in case of the linear model including X and all Z variables, which ultimately suffers from high multicollinearity.

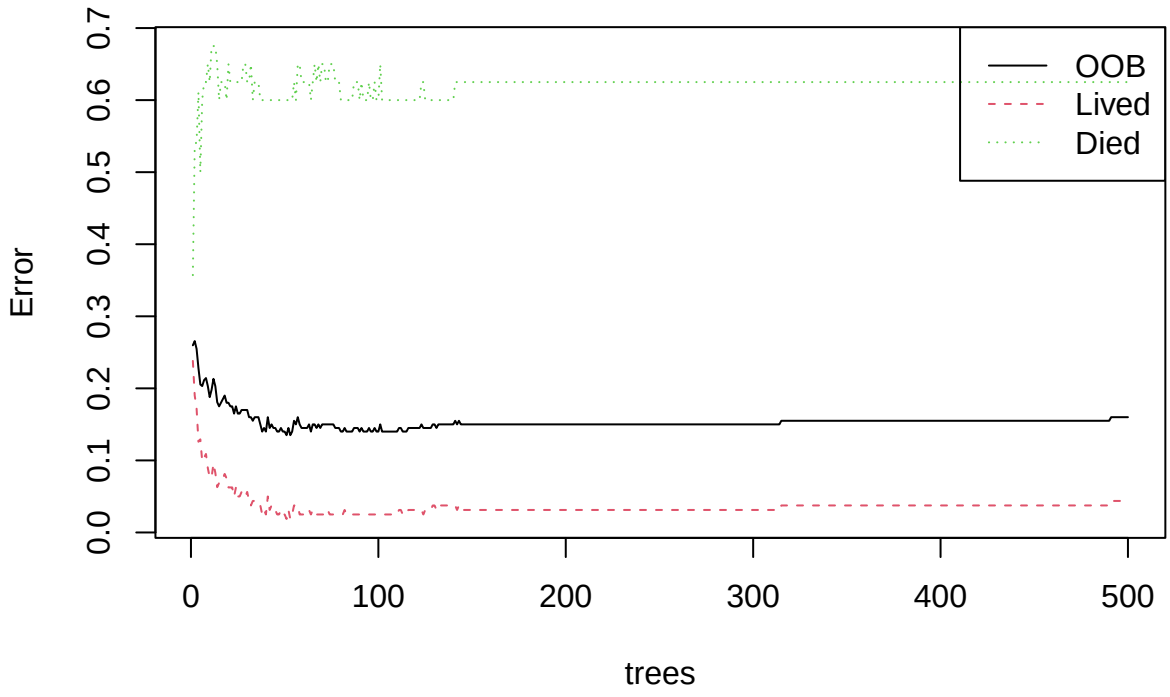
Exercise 4

We load the dataset `icu` in package **aplore3**, which contains information on patients who were admitted to an adult intensive care unit (ICU). We drop the variable `id`, which is just an ID number of the patients, and use the variable `sta` as dependent variable in order to develop a predictive model of the patients' survival.

We first fit a random forest using the default settings of `randomForest()` for the number of bootstrap iterations ($ntree = 500$) and the number of candidate variables at each split ($m = \sqrt{p}$, where p is the number of predictors).

```
##
## Call:
##  randomForest(formula = sta ~ ., data = icu, importance = TRUE)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 4
##
##         OOB estimate of  error rate: 16%
## Confusion matrix:
##       Lived Died class.error
## Lived   153    7     0.04375
## Died     25   15     0.62500
```

We now want to evaluate whether the default value of 500 bootstrap iterations is sufficiently large. In order to do so, we look at the evolution of the OOB error rate for increasing number of trees.

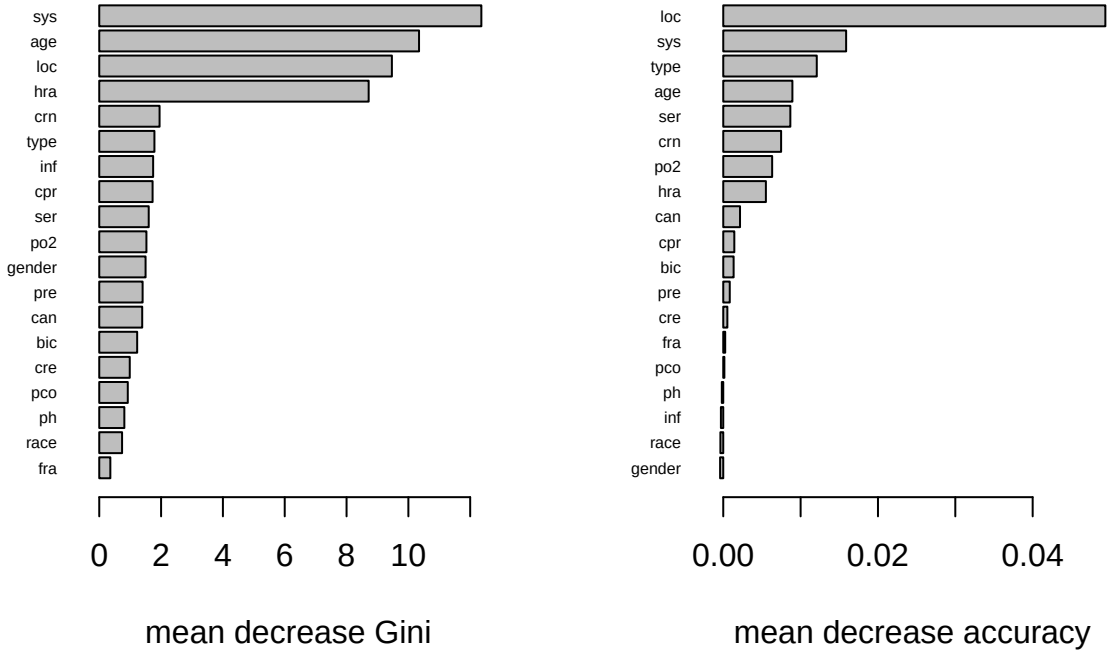


The plot shows that the error rates stabilize rather quickly and that 150 to 200 bootstrap iterations seem to be enough. However, since computational costs are limited in this exercise we continue our analysis with $ntree = 500$.

Next, we want to tune the hyperparameter m and assess its influence on the OOB error. Since the dataset is rather small, we consider all possible values $m = 1, \dots 19$.

```
##  m = 1  m = 2  m = 3  m = 4  m = 5  m = 6  m = 7  m = 8
##  0.200  0.165  0.170  0.155  0.150  0.165  0.170  0.160
##  m = 9 m = 10 m = 11 m = 12 m = 13 m = 14 m = 15 m = 16
##  0.165  0.170  0.170  0.165  0.170  0.170  0.165  0.160
## m = 17 m = 18 m = 19
##  0.165  0.165  0.180
```

We find that the OOB error rate is quite insensitive to the choice of m and that even a choice of $m = 1$ does not drastically increase the OOB error. The OOB error rate is minimized for $m = 5$ and we choose this value for the final task of this exercise, where we want to inspect the variable importance measures.



By far the most important variable according to mean decrease accuracy (MDA) is the level of consciousness at ICU admission (`loc`), followed by systolic blood pressure, type of admission (elective or emergency), age and the type of service (medical or surgery). Based on mean decrease Gini (MDG) as importance measure, four explanatory variables stand out: Systolic blood pressure is the most important predictor, followed by age, the level of consciousness and the heart rate at ICU admission.

Looking at the `class` of the five most important variables according to each measure, we find that the MDG favors numeric predictors:

```
## [1] "Top 5 - MDA:"
```

```
##       loc       sys      type       age       ser
##  "factor" "integer"  "factor" "integer"  "factor"
```

```
## [1] "Top 5 - MDG:"
```

```
##       sys       age       loc       hra       crn
## "integer" "integer"  "factor" "integer"  "factor"
```

While only 2 out of 5 most important predictors according to MDA are numeric, 3 out of the 4 variables with significantly higher MDG are numeric.

Exercise 5

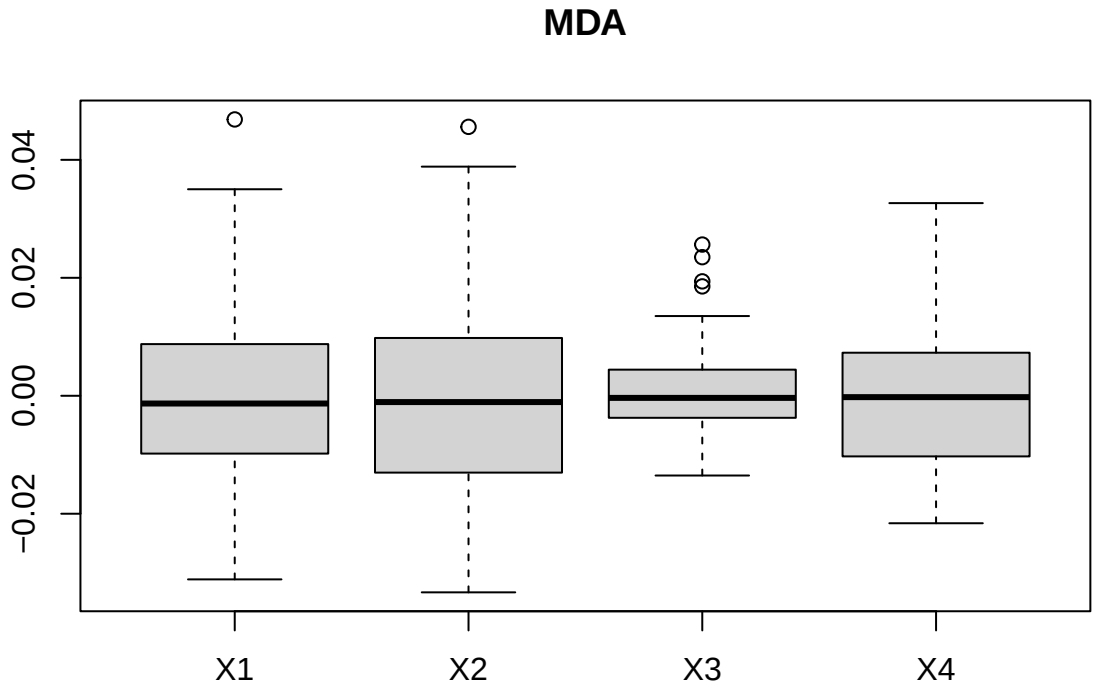
We consider four predictor variables, which have the following distributions:

$$X_1 \sim N(0,1), \qquad X_2 \sim U(0,1), \tag{1}$$
$$X_3 \sim M(1,(0.5,0.5)), \qquad X_4 \sim M(1,(0.2,0.2,0.2,0.2,0.2)). \tag{2}$$

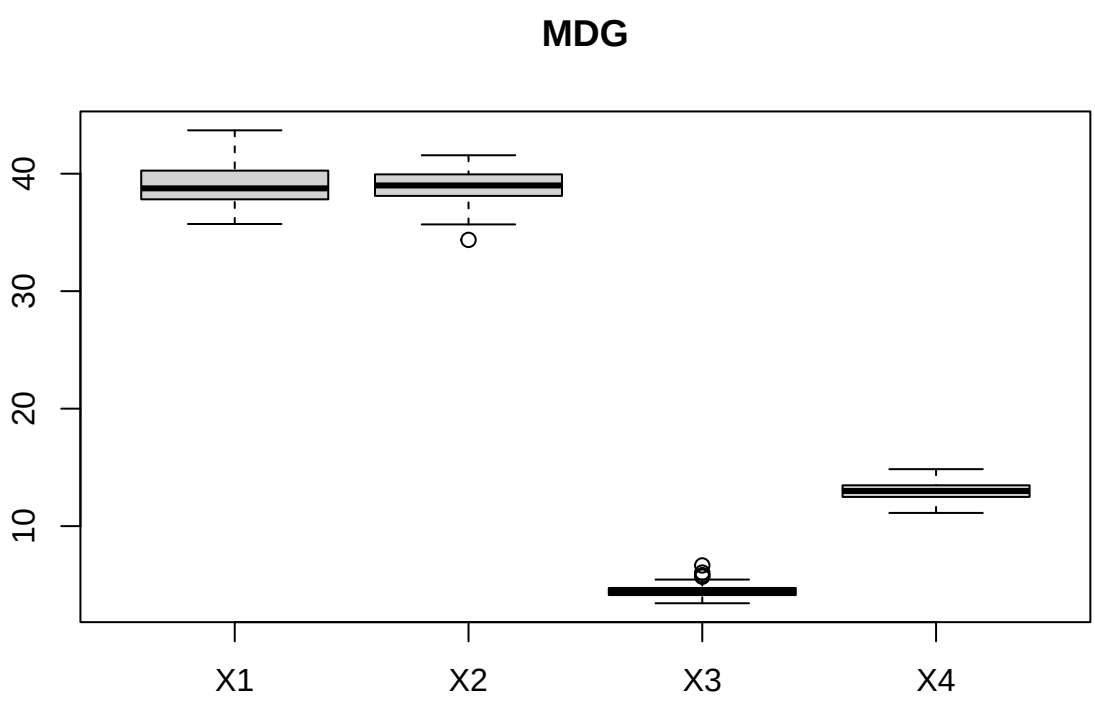
This means that we have two continuous variables, which follow either a standard normal or a standard uniform distribution, and two categorical variables with balanced categories with either 2 or 5 categories. The dependent variable y is assumed to be a binary categorical variable with equal-sized classes.

We generate 100 datasets of sample size $N = 200$ and then fit a random forest to each dataset and determine the mean decrease Gini and mean decrease accuracy values for each of the predictor variables.

Let us first have a look at the distribution of mean decrease accuracy (MDA) across the different samples for each predictor:



On average, each predictor performs equally well (or rather bad) in terms of MDA. However, in case of the binary predictor (X_3), the distribution of MDA across samples is not as dispersed. Looking at mean decrease Gini (MDG), we come to a very different conclusion:



While the performance of the two continuous predictors is very similar, MDG suggests that the two categorical variables are of much smaller importance, in particular the binary variable X_3. We conclude that MDG is sensitive to the scale of a variable and favors numeric variables compared to categorical ones, especially if the number of levels of a categorical variable is small.