Exercise 4 - Fitting and pruning a classification tree (Heart data)

```
##############   Classification trees   ##########################

# load and inspect the Heart data set
# These data contain a binary outcome HD for 303 patients who presented with chest pain.
# An outcome value of Yes indicates the presence of heart disease based on an angiographic test,
# while No means no heart disease. There are 13 predic- tors including Age, Sex, Chol
# (a cholesterol measurement), and other heart and lung function measurements.
Heart <- read.csv("./Heart.csv")
attach(Heart)
head(Heart) # AHD = Yes means Heart Disease
Heart <- Heart[,-1] # Remove the row identifier (we dont use it as a predictor)

# the categorical variables need to be of type factor rather than character to work with tree()
str(Heart)
Heart$AHD = as.factor(Heart$AHD)
Heart$ChestPain = as.factor(Heart$ChestPain)
Heart$Thal = as.factor(Heart$Thal)
plot(Heart$AHD)

# sample 70% of the row indices for subsetting as training data
set.seed (2)
trainHeart <- sample(1:nrow(Heart), 0.7*nrow(Heart))
Heart.train <- Heart[trainHeart,]
Heart.test <- Heart[-trainHeart,]

# train classification tree on *training data*
tree.AHDHeart <- tree(AHD ~ .-AHD, data = Heart.train)

plot(tree.AHDHeart)
text(tree.AHDHeart, cex=0.75, pretty=0)
    # cex: set character size to 0.75
    # pretty=0 instructs R to include the category names for any qualitative pre- dictors, rather than simply displaying a letter for each category.
    # Ca is the most important indicator for Heart Disease.
```

Exercise 4 - Fitting and pruning a classification tree (Heart data)

```
summary(tree.AHDHeart)
    # "Deviance":
        # For classification trees this is a scaled version of the entopy,
        # measured as -2 * \left( \sum_{m}\sum{k} n_{mk} * log(p_{mk}) \right).
        # Here, n_{mk} is the number of observations in the mth terminal node that belong to the kth class.
        # A small deviance indicates a tree that provides a good fit to the (training) data.
    # "Residual mean deviance":
        # Deviance devided by (n - |T_{0}|)
    # "Misclassification error rate":
        # training error rate is 8.5%.


tree.AHDHeart
    # node): node number
    # split: split criterion, e.g. Thal: normal, or Ca < 0.5
    # n: number of observations in that branch
    # deviance (the smaller the better)
    # yval: overall prediction for the branch (Yes or No)
    # (ybrob): the fraction of observations in that branch that take on values of (Yes No)
    # * denotes terminal node

# use classification tree to predict *test data*
tree.AHDHeart.pred <- predict(tree.AHDHeart, Heart.test, type="class")

# confusion table to determine classification error on test data
(tree.AHDHeart.pred.ct <- table(tree.AHDHeart.pred, Heart.test$AHD))
(tree.AHDHeart.correct <- (tree.AHDHeart.pred.ct[1,1] + tree.AHDHeart.pred.ct[2,2])/sum(tree.AHDHeart.pred.ct)) # portion of correctly classified observations: 70.3%
(tree.AHD.testError <- 1 - tree.AHDHeart.correct) # test error
```

Exercise 4 - Fitting and pruning a classification tree (Heart data)

```
#####  Cost-Complexity Pruning
    #  Goal: prune trees to avoid high variance and overfitting.
    #  Positive effects:
    #     - smaller *test* errors (due to less overfitting).
    #     - higher interpretability (due to smaller trees).

# use cross-validation to find the optimal parameter \alpha for cost-complexity pruning
set.seed (3)
cv.Heart =cv.tree(tree.AHDHeart, FUN = prune.misclass)
    # Runs a K-fold cross-validation experiment to find the number of
    # misclassifications as a function of the cost-complexity parameter \alpha.
    # "FUN = prune.misclass":
        # The *classification error rate* should guide the
        # cross-validation and pruning process (as opposed to Gini index or entropy).
        # If FUN is not specified, deviance is used as default (which is a version of entropy).
        # --> Remeber: If prediction accuracy is the goal, the error rate is
        # preferrable for pruning .

cv.Heart
    # $k: cost-complexity parameter (corresponds to \alpha)
          # Notice that \alpha is increasing (corresponding to the pruning sequence).
          # \alpha=0.75 gives the lowest cross-validation error.
    # $size: number of terminal nodes of each tree
          # Notice that the size is decreasing (corresponding to the pruning sequence).
    # $dev: *cross-validation* error rate
          # The tree with size 8 (8 terminal nodes) has lowest cross-validation error.

# plot the cross-validation error-rate as a function of both size and \alpha (k):
par(mfrow=c(1,2))
plot(cv.Heart$size, cv.Heart$dev, type="b") # type="b": plot both, points and lines
plot(cv.Heart$k, cv.Heart$dev, type="b")
par(mfrow=c(1,1))
```

Exercise 4 - Fitting and pruning a classification tree (Heart data)

```
# do the actual pruning
prune.AHDHeart <- prune.misclass(tree.AHDHeart, best=7)
    # prune.misclass:
          # is an abbreviation for prune.tree(method = "misclass") for use with cv.tree.
          # Here, prune.tree determines the nested cost-complexity sequence
    # best=7: get the 7-node tree in the cost-complexity sequence

# plot the pruned tree
plot(prune.AHDHeart)
text(prune.AHDHeart,pretty=0)

# use pruned tree to predict *test data*
prune.AHDHeart.pred <- predict(prune.AHDHeart, Heart.test, type="class")

# confusion table to determine classification error on *test data*
(prune.AHDHeart.pred.ct <- table(prune.AHDHeart.pred, Heart.test$AHD))
(prune.AHDHeart.correct <- (prune.AHDHeart.pred.ct[1,1] + prune.AHDHeart.pred.ct[2,2])/sum(prune.AHDHeart.pred.ct)) # portion of correctly classified observations
(prune.AHDHeart.testError <- 1 - prune.AHDHeart.correct) # test error (pruned)

# compare with *test error* of unpruned tree:
tree.AHD.testError # smaller error with pruning!
```