```
#### Decision tree

library(rpart)
library(rpart.plot)
mower.df <- read.csv("RidingMowers.csv")

# use rpart() to run a classification tree.
# define rpart.control() in rpart() to determine the depth of the tree.
class.tree <- rpart(Ownership ~ ., data = mower.df,
                    control = rpart.control(maxdepth = 2), method = "class")
## plot tree
# use prp() to plot the tree. You can control plotting parameters such as color, shape,
# and information displayed (which and where).
prp(class.tree, type = 1, extra = 1, split.font = 1, varlen = -10)

# Bank data tree
bank.df <- read.csv("UniversalBank.csv")
bank.df <- bank.df[ , -c(1, 5)]  # Drop ID and zip code columns.

# partition
set.seed(1)
train.index <- sample(c(1:dim(bank.df)[1]), dim(bank.df)[1]*0.6)
train.df <- bank.df[train.index, ]
valid.df <- bank.df[-train.index, ]

# classification tree
default.ct <- rpart(Personal.Loan ~ ., data = train.df, method = "class")
# plot tree
prp(default.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10)

deeper.ct <- rpart(Personal.Loan ~ ., data = train.df, method = "class", cp = 0,
minsplit = 1)
# count number of leaves
length(deeper.ct$frame$var[deeper.ct$frame$var == "<leaf>"])
# plot tree
prp(deeper.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10,
    box.col=ifelse(deeper.ct$frame$var == "<leaf>", 'gray', 'white'))

train.df$Personal.Loan <- factor(train.df$Personal.Loan)
str(train.df)
library(caret)
# classify records in the validation data.
# set argument type = "class" in predict() to generate predicted class membership.
default.ct.point.pred.train <- predict(default.ct,train.df,type = "class")
# generate confusion matrix for training data
confusionMatrix(default.ct.point.pred.train, train.df$Personal.Loan)
### repeat the code for the validation set, and the deeper tree

# argument xval refers to the number of folds to use in rpart's built-in
# cross-validation procedure
# argument cp sets the smallest value for the complexity parameter.
cv.ct <- rpart(Personal.Loan ~ ., data = train.df, method = "class",
               cp = 0.00001, minsplit = 5, xval = 5)
# use printcp() to print the table.
printcp(cv.ct)
```

```
# prune by lower cp
pruned.ct <- prune(cv.ct,
                   cp = cv.ct$cptable[which.min(cv.ct$cptable[,"xerror"]),"CP"])
length(pruned.ct$frame$var[pruned.ct$frame$var == "<leaf>"])
prp(pruned.ct, type = 1, extra = 1, split.font = 1, varlen = -10)

set.seed(1)
cv.ct <- rpart(Personal.Loan ~ ., data = train.df,
                           method = "class", cp = 0.00001, minsplit = 1, xval = 5)
# minsplit is the minimum number of observations in a node for a split to be attempted.
#xval is number K of folds in a K-fold cross-validation.
printcp(cv.ct)  # Print out the cp table of cross-validation errors.
#The R-squared for a regression tree is 1 minus rel error. xerror
#(or relative cross-validation error where "x" stands for "cross") is a scaled
#version of overall average of the 5 out-of-sample errors across the 5 folds.
pruned.ct <- prune(cv.ct, cp = 0.0154639)
prp(pruned.ct, type = 1, extra = 1, under = TRUE, split.font = 1, varlen = -10,
    box.col=ifelse(pruned.ct$frame$var == "<leaf>", 'gray', 'white'))
```