Exercise 7 – Boosted Tree (Boston Housing Data)

```
##############   Boosting   ########################

library(gbm)
    # We use the gbmpackage, and within it the gbm() function,
    # to fit boosted regression trees to the Boston data set.

# Perform boosting on the training data set, treating this as a regression problem.
set.seed (1)
boost.boston=gbm(medv~.,data=Boston[train,],
                distribution="gaussian",n.trees=5000, interaction.depth=4)
    # We run gbm() with the option distribution="gaussian" since this
    #     is a regression problem.
    # If it were a binary classification problem,
    #     we would use distribution="bernoulli".
    # "interaction.depth" refers to the maximum depth of variable interactions.
    #     1 implies an additive model,
    #     2 implies a model with up to 2-way interactions, etc.

summary(boost.boston)
    # We see that lstat and rm are by far the most important variables (again).
    # If you dont see lstst in your graph, make the plot window higher :-)

# Producing partial dependence plots for these two variables (lstat and rm).
par(mfrow=c(1,2))
plot(boost.boston ,i="rm")
plot(boost.boston ,i="lstat")
    # These plots illustrate the marginal effect of the selected variables
    # on the response after integrating out the other variables.
        # I.e., for every value of the selected variable, we calculate the predicted response
        # for every combination of values of the other variables. We then average ("integrate out")
        # over all these predicted responses. We do that for each value of teh selected variable,
        # which gives the graph.
    # As we might expect, median house prices are increasing with rm and decreasing
    # with lstat.

# Now use the boosted model to predict medv on the test set. Report the MSE.
yhat.boost=predict(boost.boston,newdata=Boston[-train,], n.trees=5000)
mean((yhat.boost -boston.test)^2)
    # The test MSE obtained is 11.8; similar to the test MSE for random
    # forests and superior to that for bagging.

# What happens if we vary the shrinkage parameter from its
# default of 0.001 to 0.02? Report the test MSE.
boost.boston=gbm(medv~.,data=Boston[train,],distribution="gaussian",n.trees=5000, interaction.depth=4,shrinkage = 0.2, verbose = F)
yhat.boost=predict(boost.boston,newdata=Boston[-train,], n.trees=5000)
mean((yhat.boost -boston.test)^2)
    # In this case, using $\lambda = 0.2$ leads to a slightly
    # lower test MSE than $\lambda = 0.001.$
```