Math189 HW8

Group Members:

Yaqi Chen (PID: A15742547; Section: A03)

Yuetong Lyu (PID: A13779993; Section: A04)

Siyi He (PID: A13400569; Section: A03)

Zhenyuan Xu (PID: A92067995; Section: A02)

Jiawei Chao (PID: A13818001; Section: A02)

Problem 1

```
library(ISLR)
library(MASS)
library(tree)
library(randomForest)

Boston <- read.csv('Boston.csv')
train <- sample(1:nrow(Boston), (1/2)*nrow(Boston)+1)
train <- sort(train)
Boston_train <- Boston[train,]
Boston_test <- Boston[-train,]
```

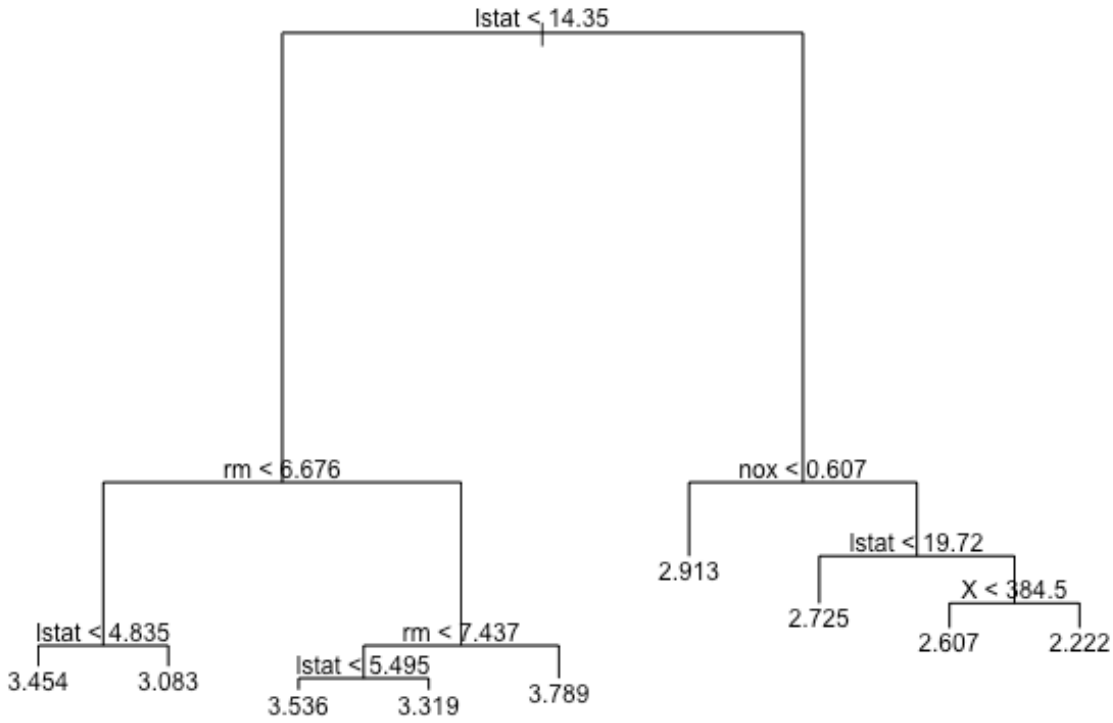
Problem 2

```
# Train a regression tree
tree.boston = tree(log(medv)~.,Boston_train)
summary(tree.boston)

# Plot the generated tree
png(file = "tree_boston.png", width=640, height=480)
plot(tree.boston)
text(tree.boston ,pretty=0)
dev.off()
```

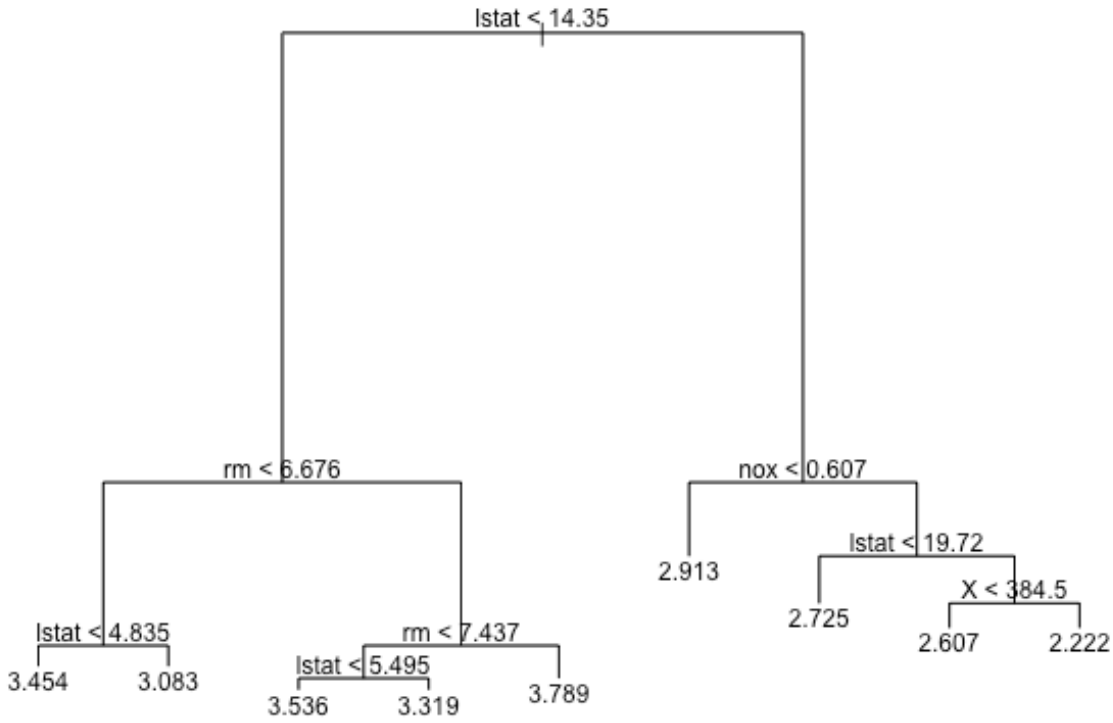
```
# Use the cv.tree() function to see whether pruning the tree will improve
performance.
cv.boston =cv.tree(tree.boston, K=6)
png(file = "cv_boston.png", width=640, height=480)
plot(cv.boston$size, cv.boston$dev, type="b", xlab="Size", ylab="CV MSE",
col="red")
dev.off()
cv.size = cv.boston$size[which.min(cv.boston$dev)]

# Use the cv selected tree size to prune the tree
prune.boston = prune.tree(tree.boston, best=cv.size)

png(file = "prune_boston.png", width=640, height=480)
plot(prune.boston)
text(prune.boston, pretty=0)
dev.off()

# Calculate prediction error on the test set
yhat = predict(tree.boston, newdata=Boston_test)
boston.test = log(Boston_test[,"medv"])

png(file = "tree_test_boston.png", width=640, height=480)
plot(yhat, boston.test)
abline (0,1)
dev.off()

# MSE on testing set
MSE.tree = mean((yhat-boston.test)^2)

yhat.subtree = predict(prune.boston, newdata=Boston_test)
boston.test = log(Boston_test[,"medv"])
MSE.subtree = mean((yhat.subtree-boston.test)^2)
```

Output:

Tree: tree_boston

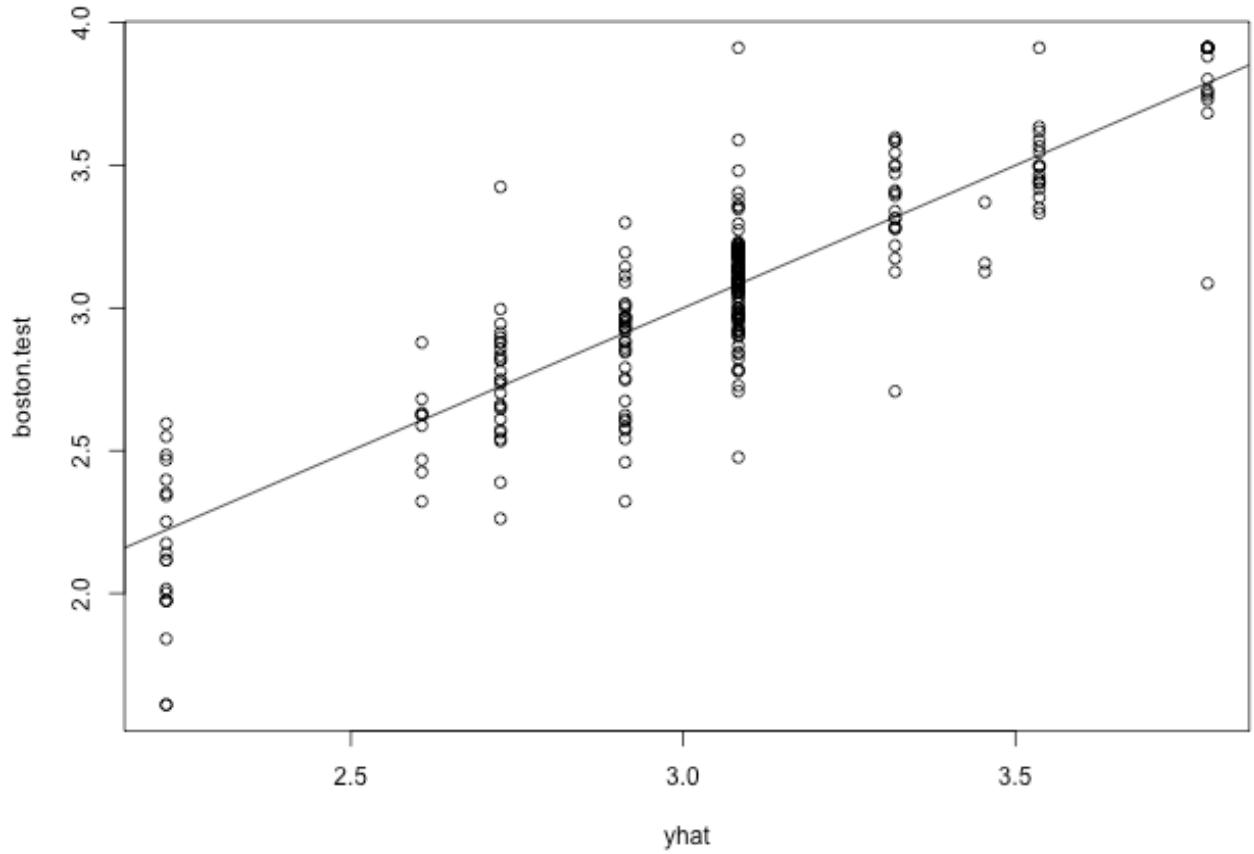


Subtree: prune_boston

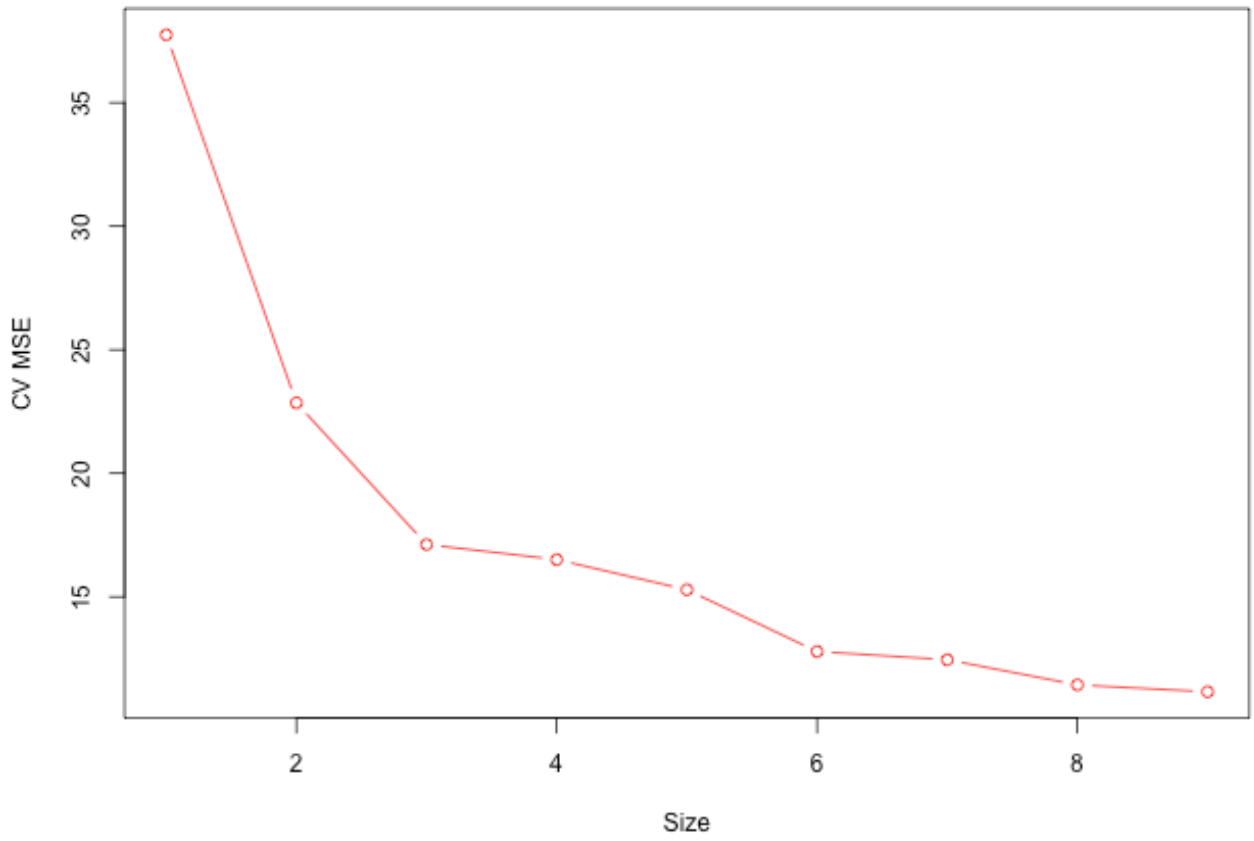



```
> MSE.tree
[1] 0.0409215
> MSE.subtree
[1] 0.0409215
```

tree_test_boston



cv_boston



Problem 3

```
# Bagging (Random forests with m=p) and 100 trees
bag.boston = randomForest(log(medv)~., data=Boston, subset=train,
                          mtry=(ncol(Boston)-1), importance=TRUE, ntree=100)

# Prediction on test set
yhat.bag = predict(bag.boston, newdata=Boston_test)
boston.test = log(Boston_test[,"medv"])

# Plot prediction performance
png(file = "bag_boston.png", width=640, height=480)
plot(yhat.bag, boston.test)
abline (0,1)
dev.off()

# MSE on test set
MSE.bag=mean((yhat.bag-boston.test)^2)

# Random forests with m=sqrt(p) and 100 trees
```

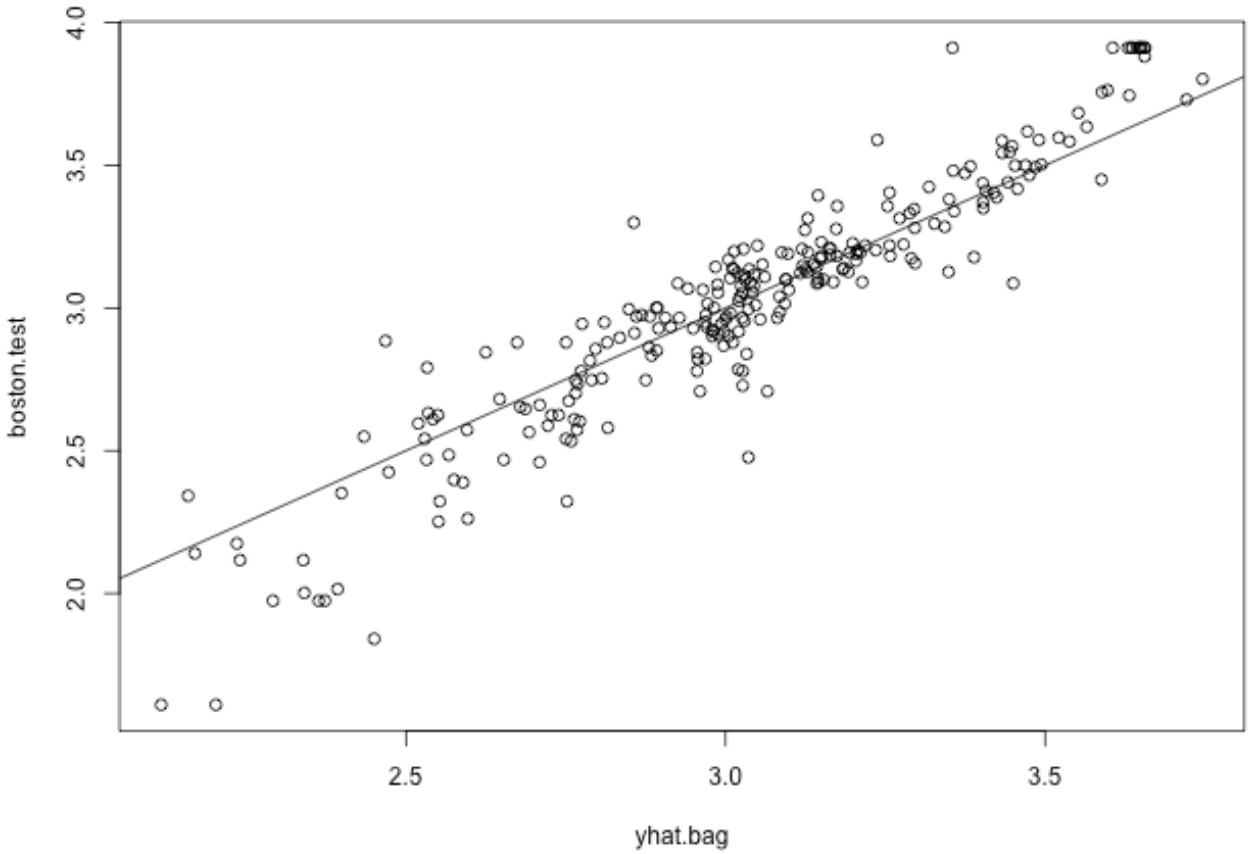
```
RF.boston =randomForest(log(medv)~.,data=Boston_train,
                        mtry=4, importance =TRUE, ntree=100)
RF.boston

# Prediction on test set
yhat.RF = predict(RF.boston, newdata=Boston_test)

# Plot prediction performance
png(file = "RandomForest_boston.png", width=640, height=480)
plot(yhat.RF, boston.test)
abline(0,1)
dev.off()

# MSE on test set
MSE.RF=mean((yhat.RF-boston.test)^2)
```

Output:

```
> MSE.bag
[1] 0.02696835
> MSE.RF
[1] 0.02656208
```

The mean squared error becomes smaller when we use bagging and random forest.

bag_boston


boston.test
yhat.bag
2.0
2.5
3.0
3.5
4.0
2.5
3.0
3.5