Assignment7

Haojin Li (Declan)

2020/7/30

## Contents

8.3.2 Fitting Regression Trees

```r
library(tree)
library(MASS)
set.seed(1)
train=sample(1:nrow(Boston),nrow(Boston)/2)
tree.boston=tree(medv~.,Boston,subset=train)
summary(tree.boston)
```

```
##
## Regression tree:
## tree(formula = medv ~ ., data = Boston, subset = train)
## Variables actually used in tree construction:
## [1] "rm"    "lstat" "crim"  "age"
## Number of terminal nodes:  7
## Residual mean deviance:  10.38 = 2555 / 246
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -10.1800  -1.7770  -0.1775   0.0000   1.9230  16.5800
```
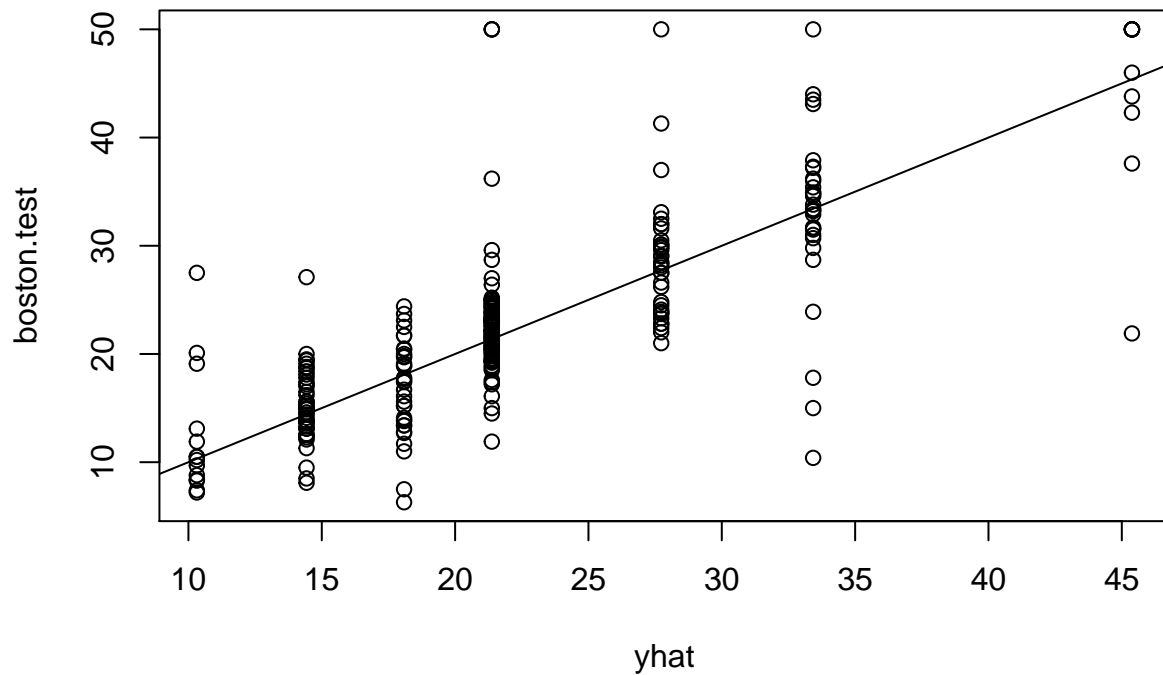
```r
plot(tree.boston)
text(tree.boston,pretty=0)
```
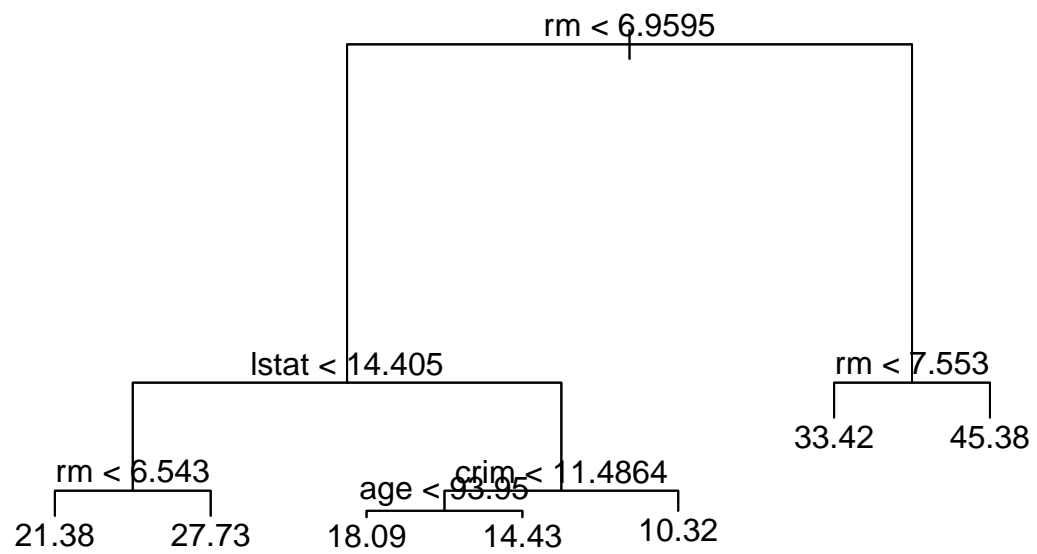
```r
cv.boston=cv.tree(tree.boston)
plot(cv.boston$size,cv.boston$dev,type='b')
```

```
yhat=predict(tree.boston,newdata=Boston[-train,])
boston.test=Boston[-train,"medv"]
plot(yhat,boston.test)
abline(0,1)
```

```r
mean((yhat-boston.test)^2)
```

```
## [1] 35.28688
```

8.3.3 Bagging and Random Forests

```r
library(randomForest)
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```r
set.seed(1)
bag.boston = randomForest(medv~., data = Boston, mtry = 13, importance = TRUE)
bag.boston
```

```
##
## Call:
##  randomForest(formula = medv ~ ., data = Boston, mtry = 13, importance = TRUE)
##                Type of random forest: regression
##                      Number of trees: 500
```

```
## No. of variables tried at each split: 13
##
##          Mean of squared residuals: 10.44785
##                  % Var explained: 87.62
```

```r
yhat.bag = predict(bag.boston, newdata = Boston[-train,])
plot(yhat.bag, boston.test)
abline(0,1)
```
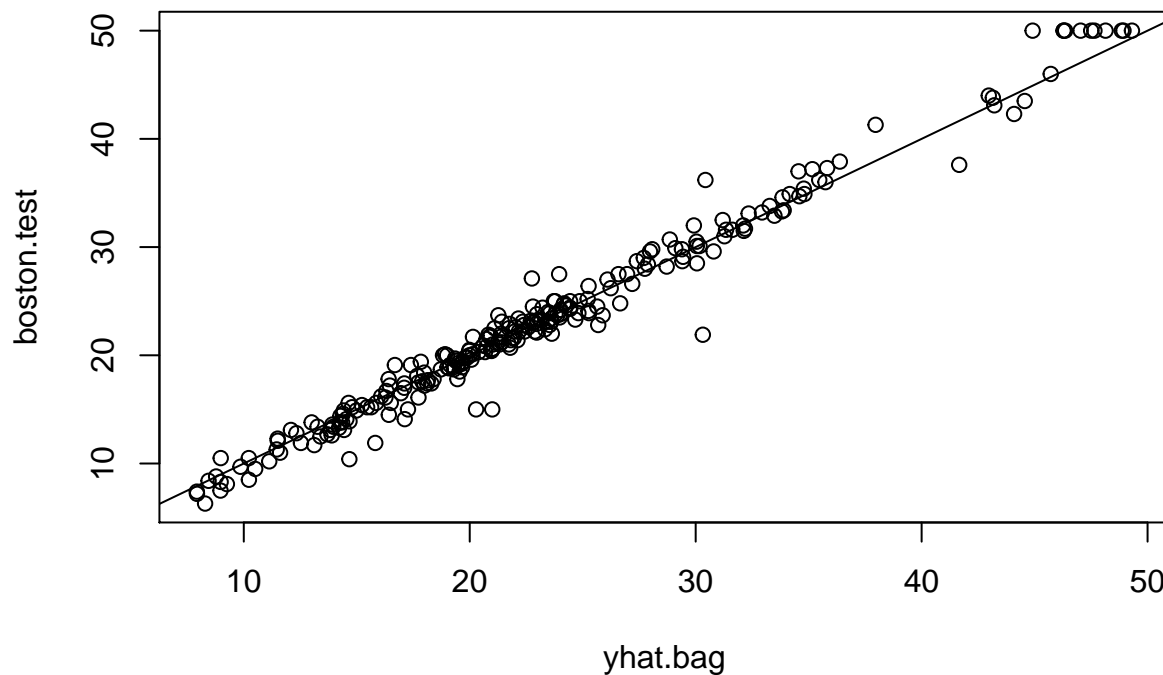


```r
mean((yhat.bag-boston.test)^2)
```

```
## [1] 2.187479
```

```r
bag.boston = randomForest(medv~., data = Boston, mtry = 13, ntree = 25)
yhat.bag = predict(bag.boston, newdata = Boston[-train,])
mean((yhat.bag-boston.test)^2)
```

```
## [1] 2.664612
```

```r
set.seed(1)
rf.boston= randomForest(medv~.,data=Boston , subset=train ,mtry=6,
importance =TRUE)
yhat.rf = predict(rf.boston ,newdata=Boston[-train ,])
mean((yhat.rf-boston.test)^2)
```

```
## [1] 19.62021
```

importance (rf.boston)

```
##            %IncMSE IncNodePurity
## crim    16.697017    1076.08786
## zn       3.625784      88.35342
## indus    4.968621     609.53356
## chas     1.061432      52.21793
## nox     13.518179     709.87339
## rm      32.343305    7857.65451
## age     13.272498     612.21424
## dis      9.032477     714.94674
## rad      2.878434      95.80598
## tax      9.118801     364.92479
## ptratio  8.467062     823.93341
## black    7.579482     275.62272
## lstat   27.129817    6027.63740
```

varImpPlot (rf.boston)

rf.boston