8.3.2

Group 14

Lab 8.3.2: Fitting Regression Trees

Fit the Tree

We fit a regression tree to the Boston Housing Data, which is available through the MASS package in R studio and include 14 variables and 506 observations. First, we split data half and half into training and test sets , then fit the regression tree model on the training data only.

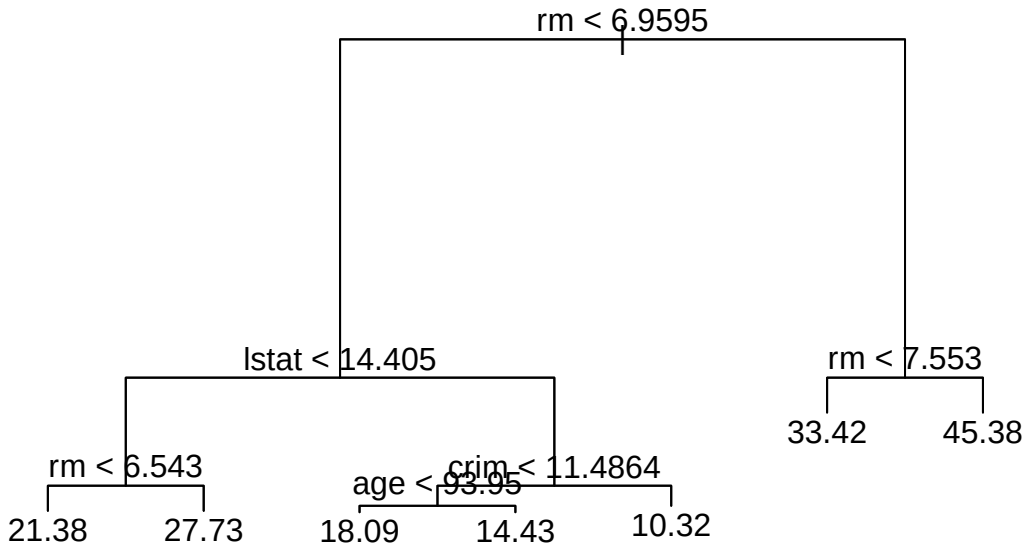
```
library(tree)
library(MASS)
set.seed(1)
train = sample(1:nrow(Boston), nrow(Boston)/2)
tree.boston = tree(medv~., Boston, subset = train)
summary(tree.boston)
```

```
##
## 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
```

Plot Tree

Notice that although the tree grown to full depth has 7 leaves, only four of the variables (rm, lstat, crim and age) have been used to construct this tree. In the context of a regression tree, the deviance is simply the sum of squared errors for the tree. Now, we plot the tree.

```
plot(tree.boston)
text(tree.boston,pretty=0)
```



The rm variable measures the average number of rooms per dwelling, and the lstat variable measures the percentage of individuals with lower socioeconomic status. The tree shows that higher values of rm correspond to higher house values and lower values of rm correspond to cheap houses.

Preform Cross-Validation

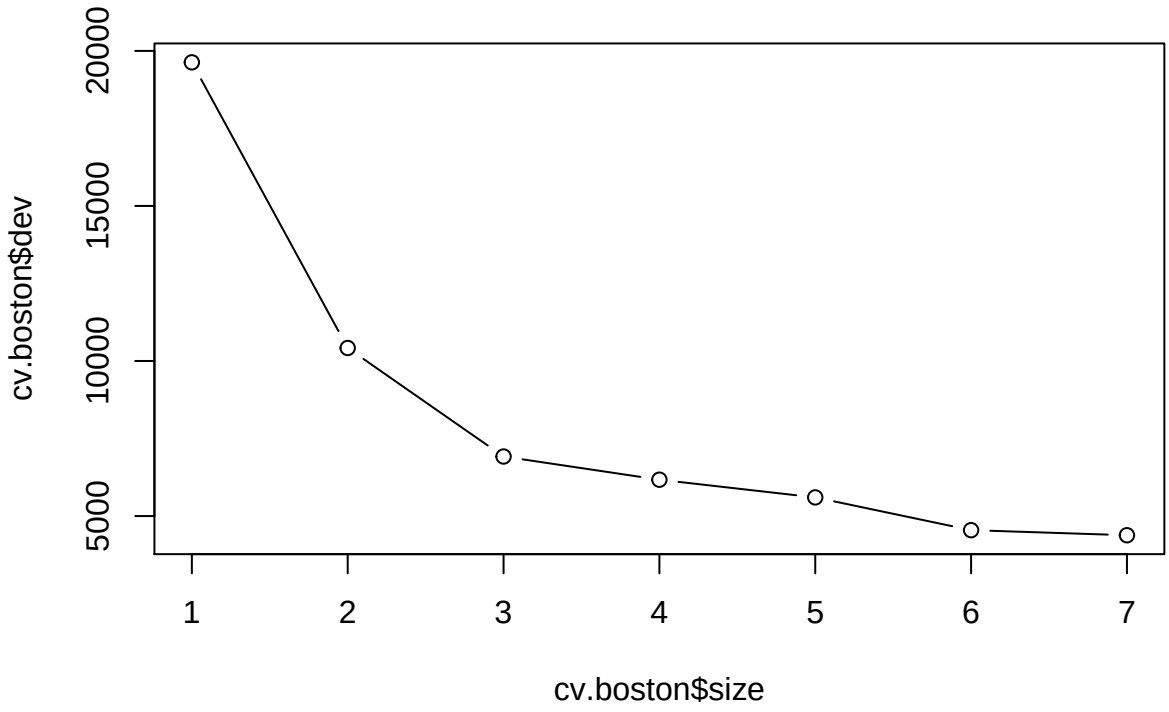
Now we use 10-fold cross validation by using cv.tree() function in order to determine the optimal level of tree complexity. This will help us decide whether pruning the tree will improve performance.

```
cv.boston=cv.tree(tree.boston)
cv.boston
```

```
## $size
## [1] 7 6 5 4 3 2 1
##
## $dev
## [1]  4380.849  4544.815  5601.055  6171.917  6919.608 10419.472 19630.870
##
## $k
## [1]       -Inf   203.9641   637.2707   796.1207  1106.4931  3424.7810 10724.5951
##
## $method
## [1] "deviance"
##
## attr(,"class")
## [1] "prune"         "tree.sequence"
```

Plot the Cross-Validation

```
plot(cv.boston$size,cv.boston$dev,type='b')
```



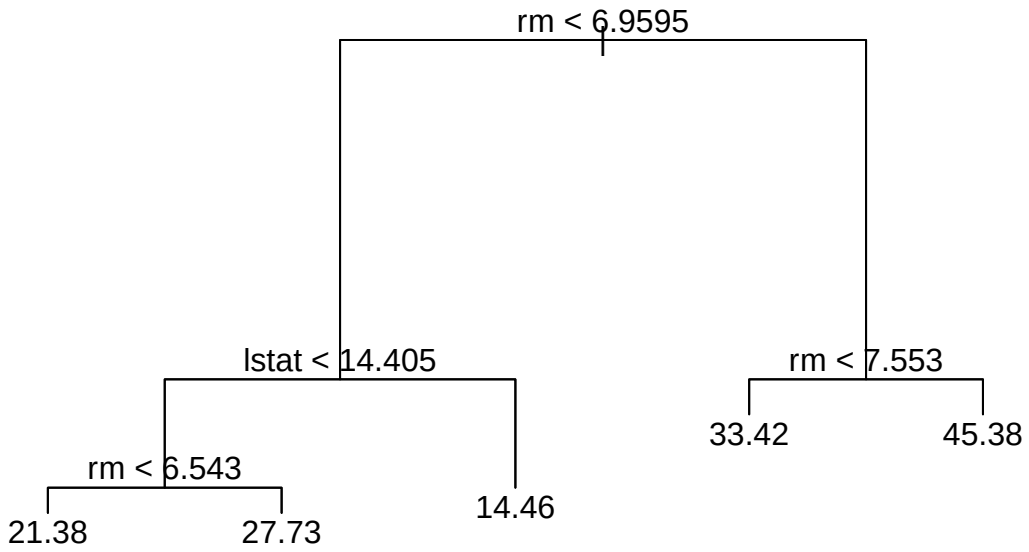
Through the plot, we can find that the most complex tree is selected by cross-validation. However if we wanted to prune the tree, we would use the prune.tree() function as the following.

Make model by using Best Pruned Tree

```
prune.boston=prune.tree(tree.boston,best=5)
summary(prune.boston)
```

```
##
## Regression tree:
## snip.tree(tree = tree.boston, nodes = 5L)
## Variables actually used in tree construction:
## [1] "rm"    "lstat"
## Number of terminal nodes:  5
## Residual mean deviance:  13.69 = 3396 / 248
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -10.1800  -1.9770  -0.1775   0.0000   2.4230  16.5800
```

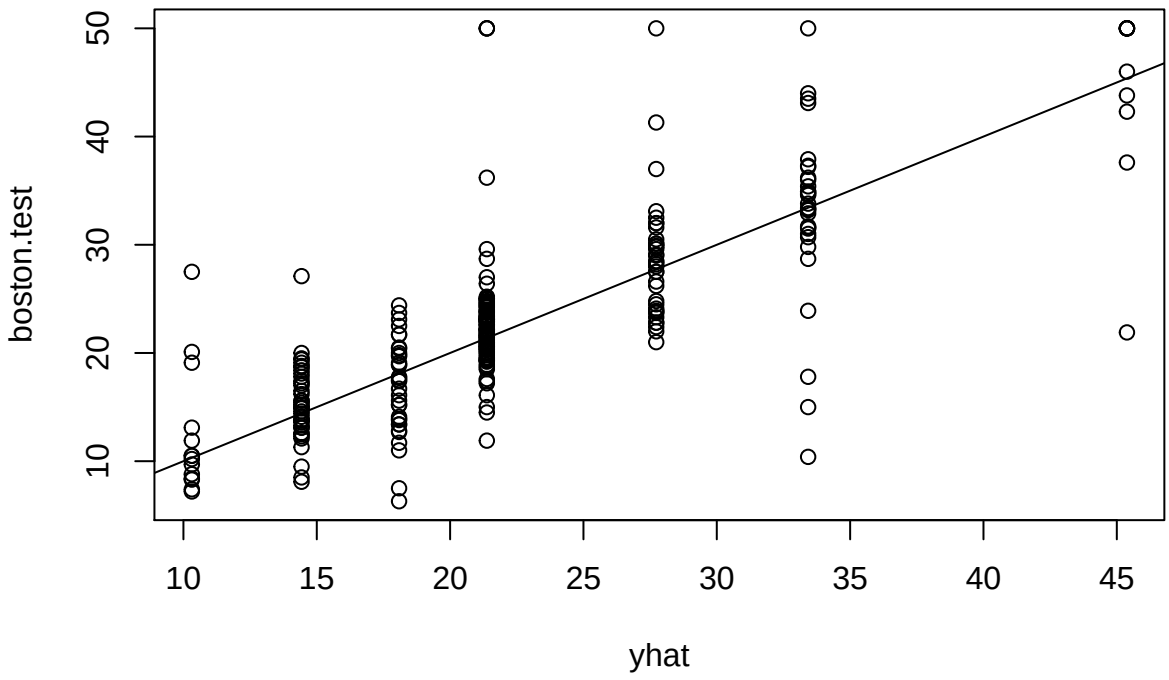
```
plot(prune.boston)
text(prune.boston,pretty=0)
```



Find the Test Error Rate

We use the unpruned tree to make predictions on the test set to keep the cross-validation results.

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




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

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

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

```
## [1] 5.940276
```

We got the result that the test set MSE for the regression tree is 35.29, and the square root is around 5.940 which means that this model gives predictions that are within around $5,940 of the true median home value.