```r
# regtree.r
RNGkind(sample.kind = "Rounding")   # to agree with textbook
```

```
## Warning in RNGkind(sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```r
library(MASS)      # Boston dataset
library(tree)      # tree()

dim(Boston)
```

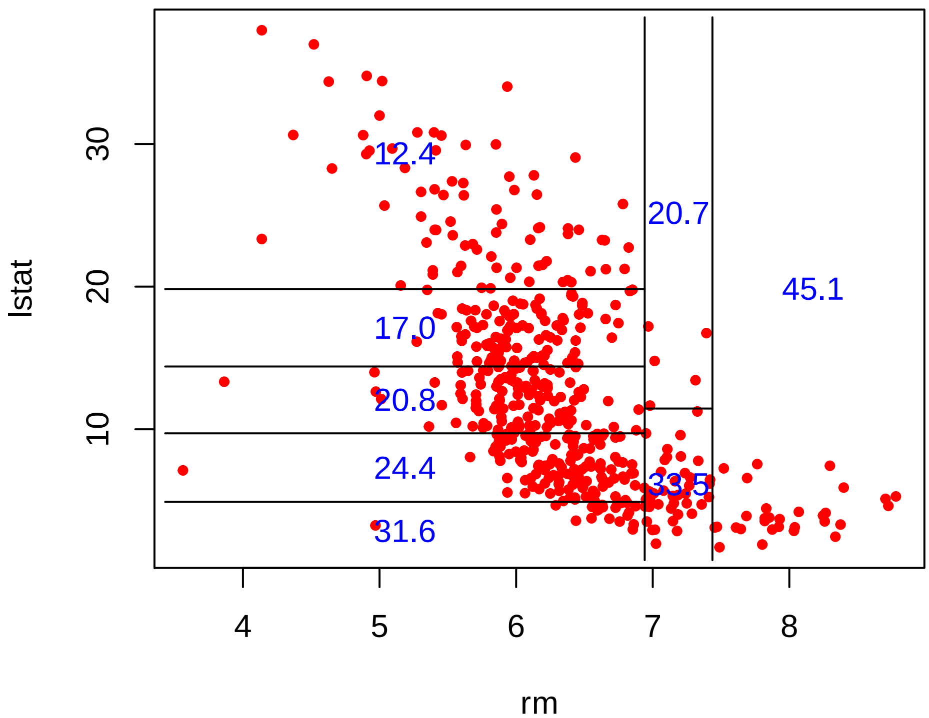
```
## [1] 506  14
```

```r
?Boston
# medv is response, p=13 predictors
round(head(Boston),3)
```

```
##    crim zn indus chas   nox    rm  age   dis rad tax ptratio  black lstat medv
## 1 0.006 18  2.31    0 0.538 6.575 65.2 4.090   1 296    15.3 396.90  4.98 24.0
## 2 0.027  0  7.07    0 0.469 6.421 78.9 4.967   2 242    17.8 396.90  9.14 21.6
## 3 0.027  0  7.07    0 0.469 7.185 61.1 4.967   2 242    17.8 392.83  4.03 34.7
## 4 0.032  0  2.18    0 0.458 6.998 45.8 6.062   3 222    18.7 394.63  2.94 33.4
## 5 0.069  0  2.18    0 0.458 7.147 54.2 6.062   3 222    18.7 396.90  5.33 36.2
## 6 0.030  0  2.18    0 0.458 6.430 58.7 6.062   3 222    18.7 394.12  5.21 28.7
```

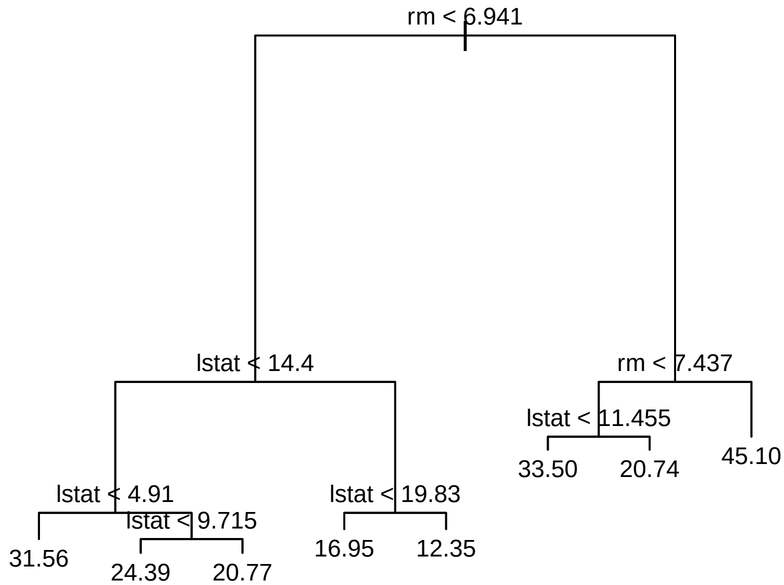
```r
# tree - 2 predictors, full dataset
#===================================================================
tree0=tree(medv~lstat+rm,Boston)

# scatterplot on predictors space
plot(lstat~rm,Boston,pch=19,cex=0.6,col="red")
# regions and predicted averages
partition.tree(tree0,add = T,col="blue")
```




```r
#
#
```

```
# tree plot
plot(tree0)
text(tree0,cex=0.75)
```



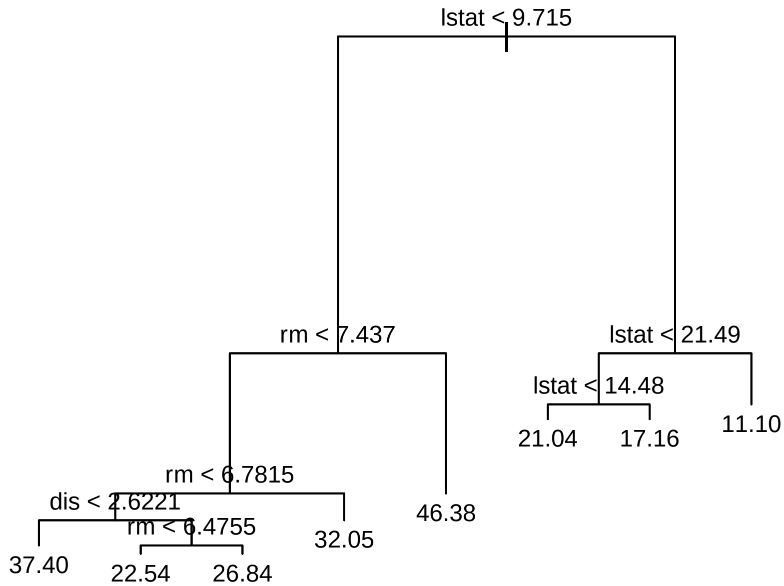

```
# inequality at split is for left arm
# $16950 is house prediction for rm < 6.94, and 14.4 < lstat < 19.83

# all predictors - train set
#=================================================================
set.seed(1)
n = nrow(Boston)
train = sample(1:n,n/2)   # 253 train rows
dtrain = Boston[train,]
dtest = Boston[-train,]

tree1=tree(medv~.,Boston,subset=train)
summary(tree1)
```

```
##
## Regression tree:
## tree(formula = medv ~ ., data = Boston, subset = train)
## Variables actually used in tree construction:
## [1] "lstat" "rm"    "dis"
## Number of terminal nodes:  8
## Residual mean deviance:  12.65 = 3099 / 245
## Distribution of residuals:
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -14.10000  -2.04200  -0.05357   0.00000   1.96000  12.60000
```

```
#
# "lstat" "rm"    "dis"     best classifiers
# RSS is 3099
# tree with 8 terminal nodes
# 253 - 8 = 245 dof
#
#
plot(tree1)
text(tree1,cex=0.75)
```




```r
# partition.tree()  does not apply for 3 classifiers
#
# model components
names(tree1)
```

```
## [1] "frame"   "where"   "terms"   "call"    "y"       "weights"
```

```r
#
#
tree1$frame
```

```
##       var   n        dev     yval splits.cutleft splits.cutright
## 1   lstat 253 20894.6572 22.67312         <9.715          >9.715
## 2      rm 103  7764.5843 30.13204         <7.437          >7.437
## 4      rm  89  3310.1604 27.57640        <6.7815         >6.7815
## 8     dis  61  1994.6223 25.52131        <2.6221         >2.6221
## 16 <leaf>   5   615.7800 37.40000
## 17     rm  56   610.3336 24.46071        <6.4755         >6.4755
## 34 <leaf>  31   136.3555 22.54194
## 35 <leaf>  25   218.3200 26.84000
## 9  <leaf>  28   496.6496 32.05357
## 5  <leaf>  14   177.8436 46.37857
## 3   lstat 150  3464.7147 17.55133         <21.49          >21.49
## 6   lstat 120  1593.6987 19.16333         <14.48          >14.48
## 12 <leaf>  62   398.4892 21.04032
## 13 <leaf>  58   743.2822 17.15690
## 7  <leaf>  30   311.8897 11.10333
```

```r
#
# 22.67312 is mean response (medv) in training set
# dev = deviance (square distance to the mean of that region)
# columns splits.cutleft and .cutright show inequalities for non-leaf rows
# <leaf> rows are terminal nodes
#        sum of deviance of terminal nodes is 3099
#        sum of deviance decreases with large n. splits
# y val  of terminal nodes are means of regions
# leftmost column is order of splitting
# 1st row splits into rows 2 and 3
# 2nd row splits into row 4 and 5
```

```r
# n number of obs before split (unless it is a leaf)

# prunning tree to 5 terminal nodes
#==================================================================

pruned1=prune.tree(tree1,best=5)
pruned1$frame
```

```
##      var   n        dev     yval splits.cutleft splits.cutright
## 1  lstat 253 20894.6572 22.67312         <9.715          >9.715
## 2     rm 103  7764.5843 30.13204         <7.437          >7.437
## 4     rm  89  3310.1604 27.57640        <6.7815         >6.7815
## 8 <leaf>  61  1994.6223 25.52131
## 9 <leaf>  28   496.6496 32.05357
## 5 <leaf>  14   177.8436 46.37857
## 3  lstat 150  3464.7147 17.55133         <21.49          >21.49
## 6 <leaf> 120  1593.6987 19.16333
## 7 <leaf>  30   311.8897 11.10333
```

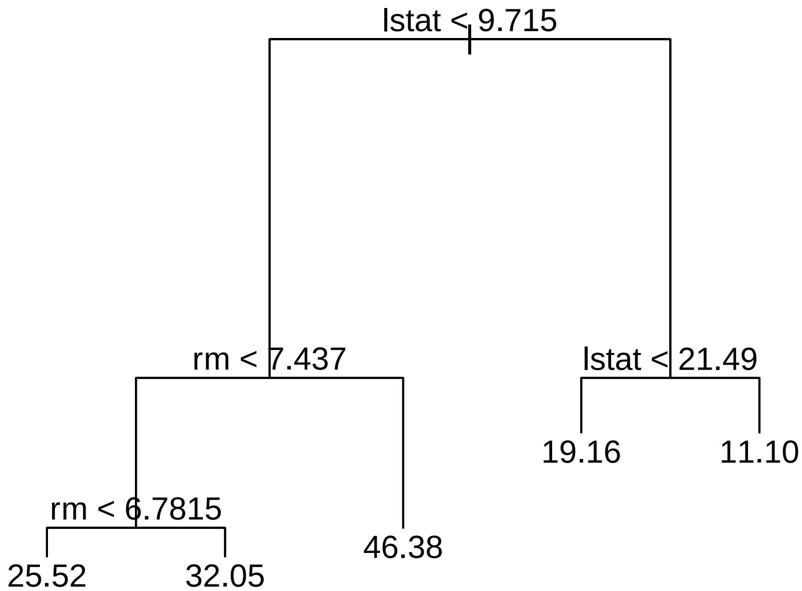
```r
summary(pruned1)
```

```
##
## Regression tree:
## snip.tree(tree = tree1, nodes = c(6L, 8L))
## Variables actually used in tree construction:
## [1] "lstat" "rm"
## Number of terminal nodes:  5
## Residual mean deviance:  18.45 = 4575 / 248
## Distribution of residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -9.56300 -2.86300 -0.06333  0.00000  2.69700 24.48000
```

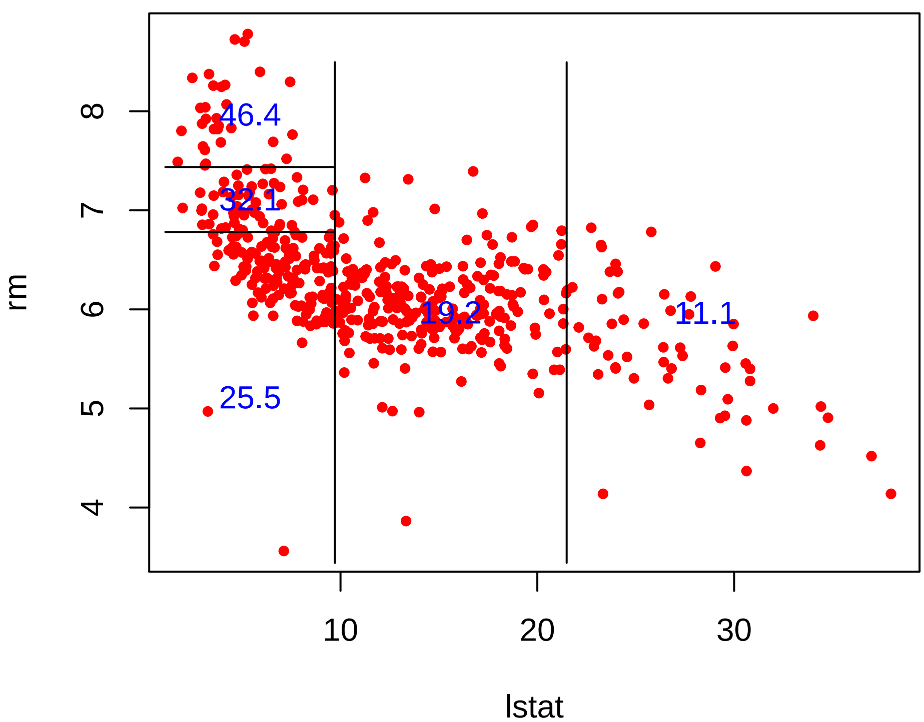
```r
#
# Regression tree:
# snip.tree(tree = tree1, nodes = c(6L, 8L))
# Variables actually used in tree construction:
# [1] "lstat" "rm"
# Number of terminal nodes:  5
# Residual mean deviance:  18.45 = 4575 / 248
# Distribution of residuals:
#     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
# -9.56300 -2.86300 -0.06333  0.00000  2.69700 24.48000

# Residual mean deviance 18.45, larger than 12.65 of non-pruned tree

plot(pruned1)
text(pruned1)
```

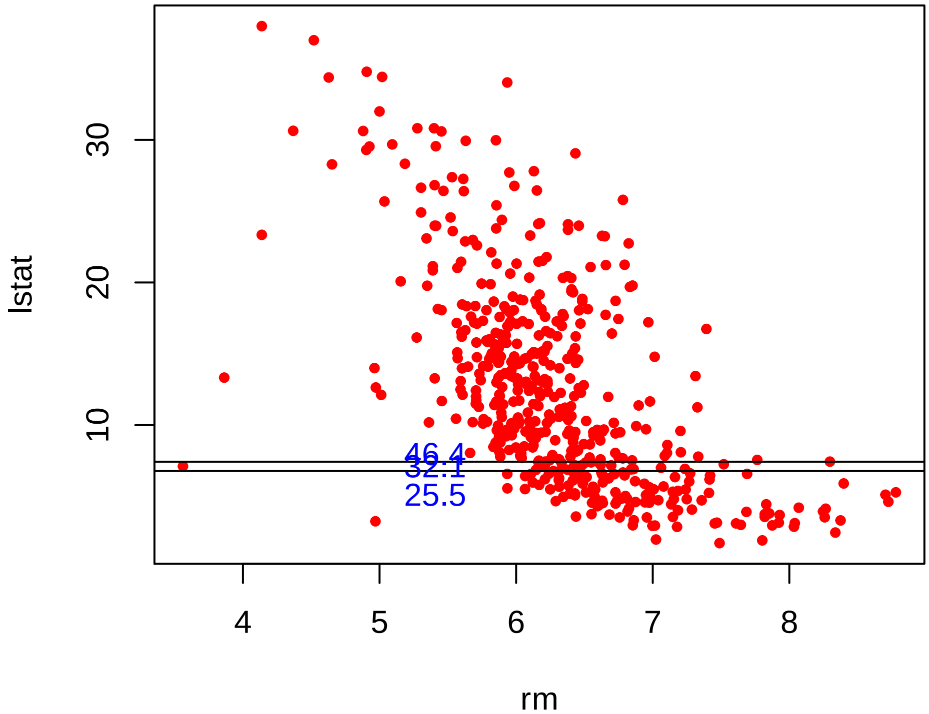



```r
# regions
plot(rm~lstat,Boston,pch=19,cex=0.6,col="red")
partition.tree(pruned1,add = T,col="blue")
```

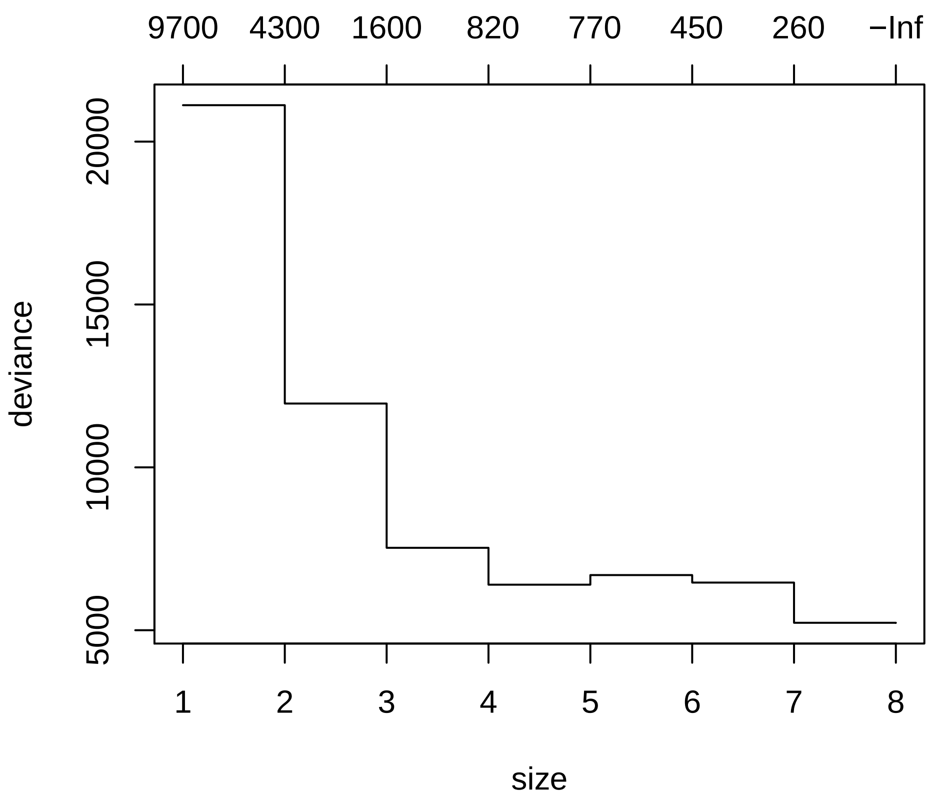



```r
# most important predictor must go on X-axis

# this way is wrong!
plot(lstat~rm,Boston,pch=19,cex=0.6,col="red")
partition.tree(pruned1,add = T,col="blue")
```




```r
# cross validation - best n. terminal nodes
#=================================================================
cv.boston=cv.tree(tree1)
plot(cv.boston)                           # 8 nodes is best tree
```




```r
# test error rate
y.test= Boston[-train,"medv"]            # y values in test set
newval= Boston[-train,]
yhat  = predict(tree1,newval)
```

```r
# n. of predictions = n. of regions
unique(yhat)
```

```
## [1] 26.84000 22.54194 32.05357 17.15690 11.10333 21.04032 37.40000 46.37857
```

```r
# 26.84000 22.54194 32.05357 17.15690 11.10333 21.04032 37.40000 46.37857

# plot means of terminal regions (yhat) vs y
plot(yhat~y.test,pch=19,cex=0.5,ylim=c(10,50))
abline(0,1)
grid()

# test MSE
mspe = mean((yhat-y.test)^2)   # 25.05
sqrt(mspe)                     #[1] 5.004557
```

```
## [1] 5.004557
```

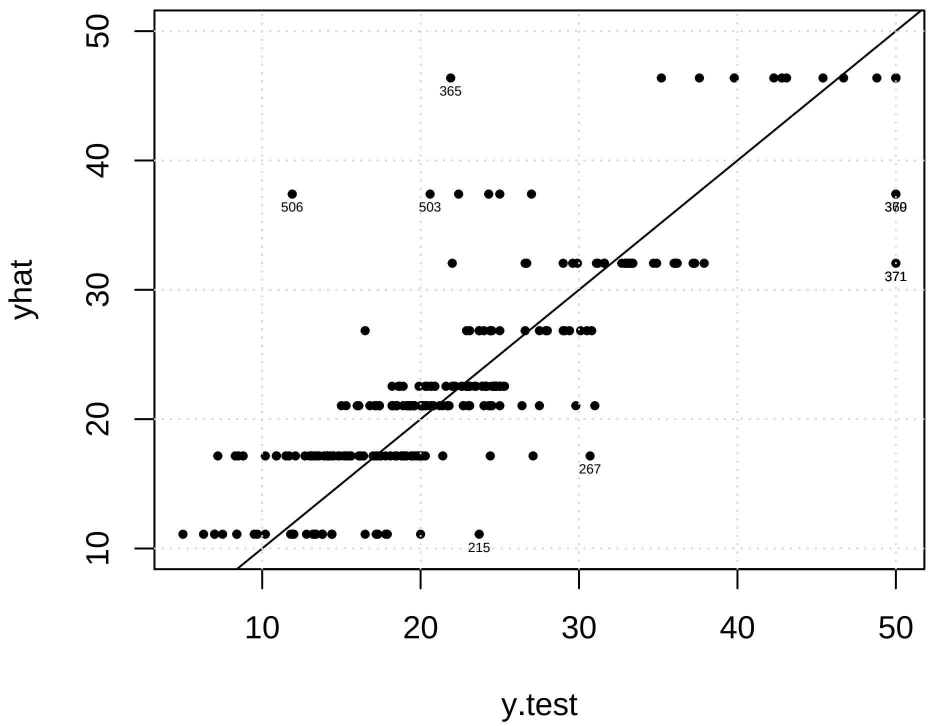
```r
#
# predictions are within $5005 of true median home value
#
# identify houses with large residuals
#
res = y.test - yhat
# yhat is vector
# vector has no rownames
a = rownames(as.matrix(yhat))   # as.matrix required

text(yhat~y.test,labels=ifelse(res>10,a,""),pos=1,offset=0.25,cex=0.4)

# houses with res>5
a[res>5]
```

```
##  [1] "8"   "9"   "124" "149" "180" "183" "185" "209" "210" "215" "223" "264"
## [13] "267" "292" "369" "370" "371" "409" "413" "474"
```

```r
#
text(yhat~y.test,labels=ifelse(res>15|res<(-15),a,""),pos=1,offset=0.25,cex=0.4)
```


yhat
y.test
10
20
30
40
50
365
506
503
371
267
215