Regression Tree with HSAUR

Yamuna Dhungana

Exercises

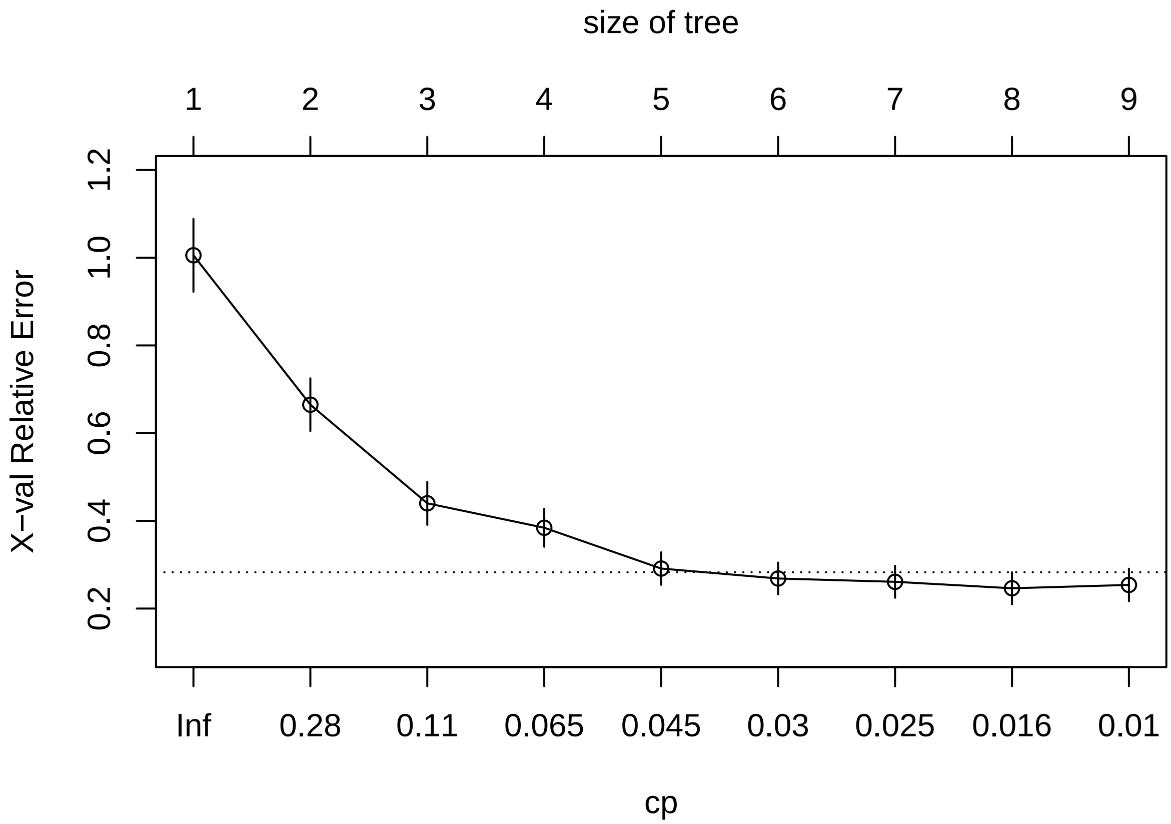
1. (Ex. 9.1 pg 186 in HSAUR, modified for clarity) The **BostonHousing** dataset reported by Harrison and Rubinfeld (1978) is available as a `data.frame` structure in the **mlbench** package (Leisch and Dimitriadou, 2009). The goal here is to predict the median value of owner-occupied homes (`medv` variable, in 1000s USD) based on other predictors in the dataset.

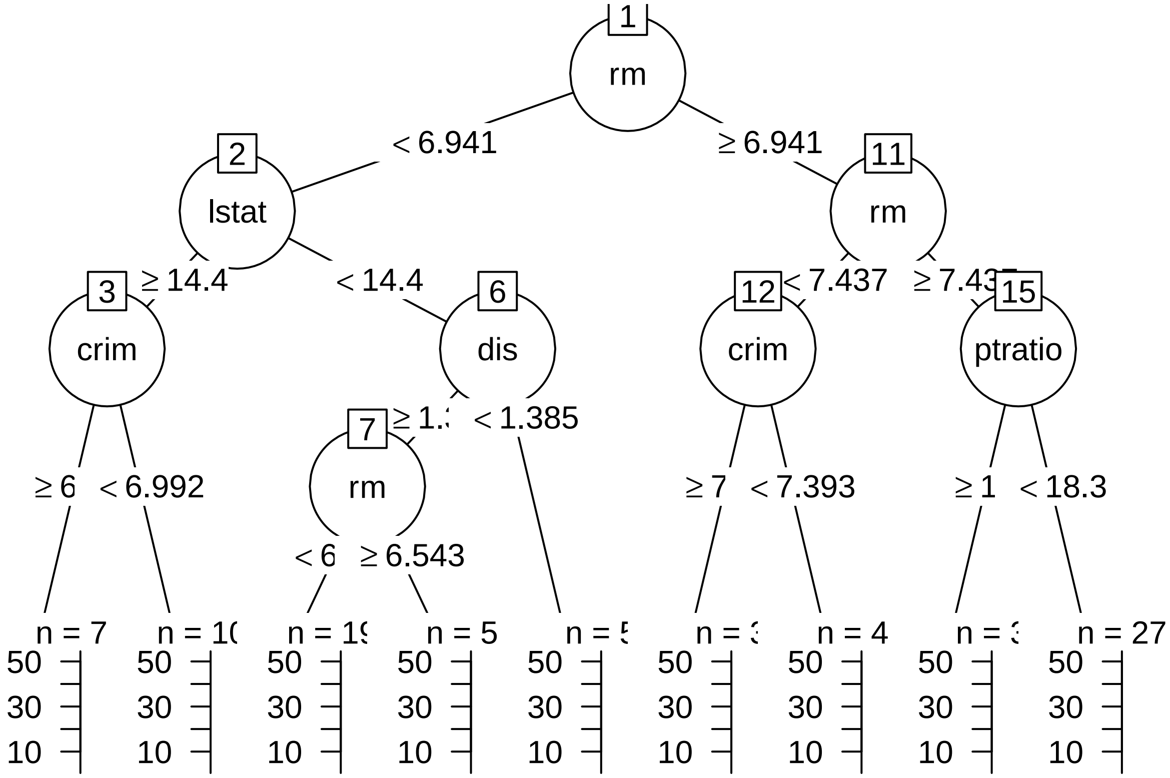
   a) Construct a regression tree using rpart(). Discuss the results, including these key components:

      - How many nodes does your tree have?
      - Did you prune the tree? Did it decrease the number of nodes?
      - What is the prediction error (MSE)?
      - Plot the predicted vs. observed values.
      - Plot the final tree.

First of all, installing all the required libraries for the exercise

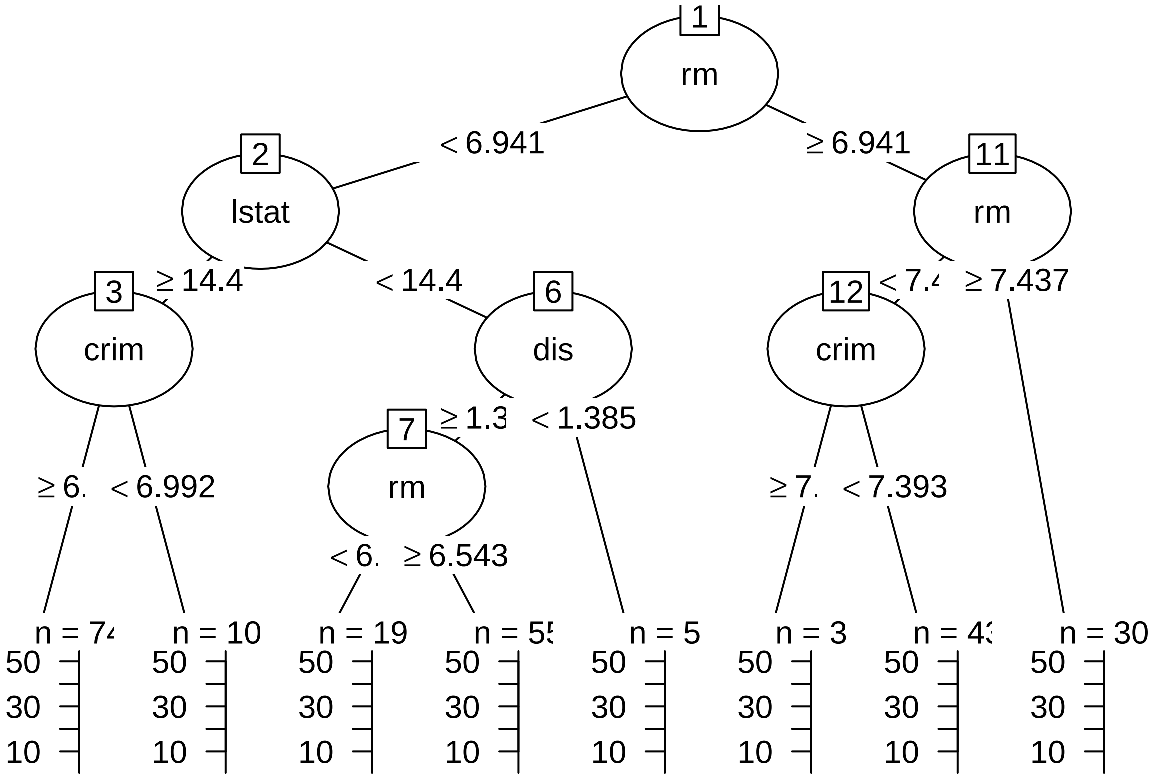
```
##
## Regression tree:
## rpart(formula = medv ~ ., data = BostonHousing, control = rpart.control(minsplit = 10))
##
## Variables actually used in tree construction:
## [1] crim    dis     lstat   ptratio rm
##
## Root node error: 42716/506 = 84.42
##
## n= 506
##
##         CP nsplit rel error  xerror     xstd
## 1 0.452744      0   1.00000 1.00556 0.083187
## 2 0.171172      1   0.54726 0.66485 0.060191
## 3 0.071658      2   0.37608 0.43995 0.049160
## 4 0.059002      3   0.30443 0.38423 0.043400
## 5 0.033756      4   0.24542 0.29121 0.037190
## 6 0.026613      5   0.21167 0.26852 0.036445
## 7 0.023572      6   0.18506 0.26098 0.036438
## 8 0.010859      7   0.16148 0.24626 0.036535
## 9 0.010000      8   0.15062 0.25383 0.037104
```

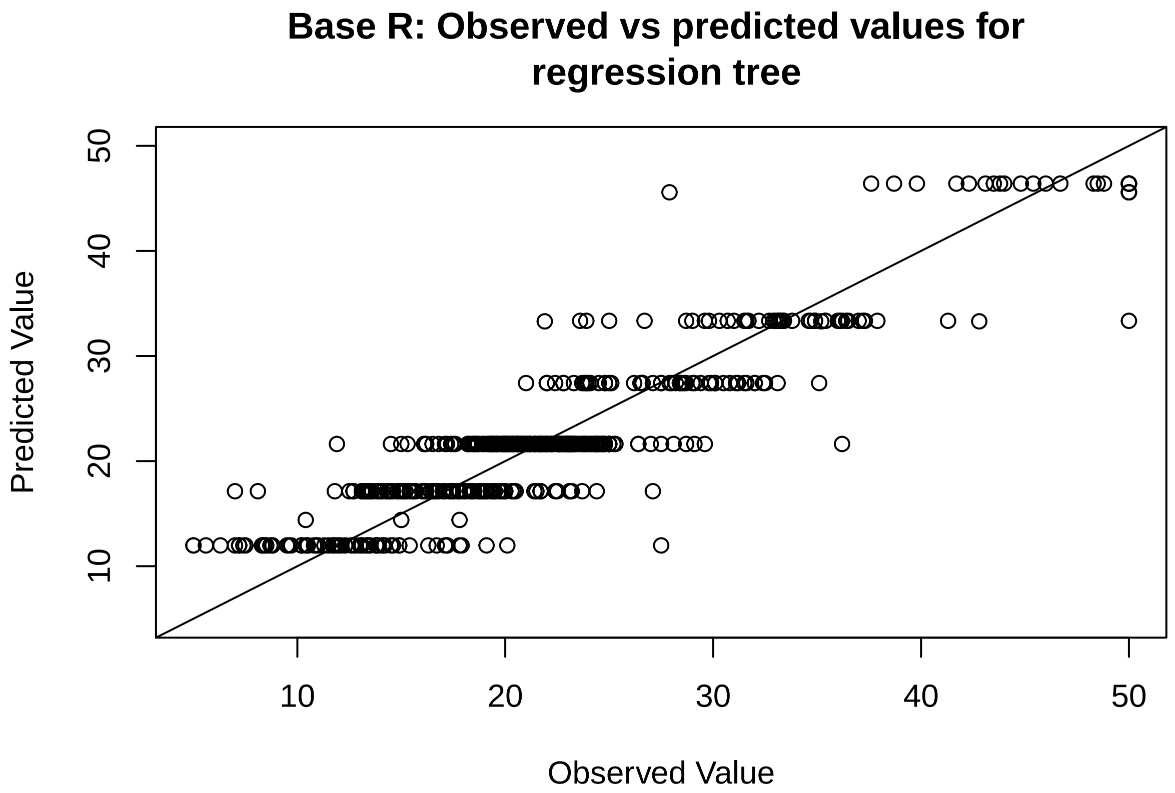

size of tree
1
2
3
4
5
6
7
8
9
X−val Relative Error
0.2
0.4
0.6
0.8
1.0
1.2
Inf
0.28
0.11
0.065
0.045
0.03
0.025
0.016
0.01
cp



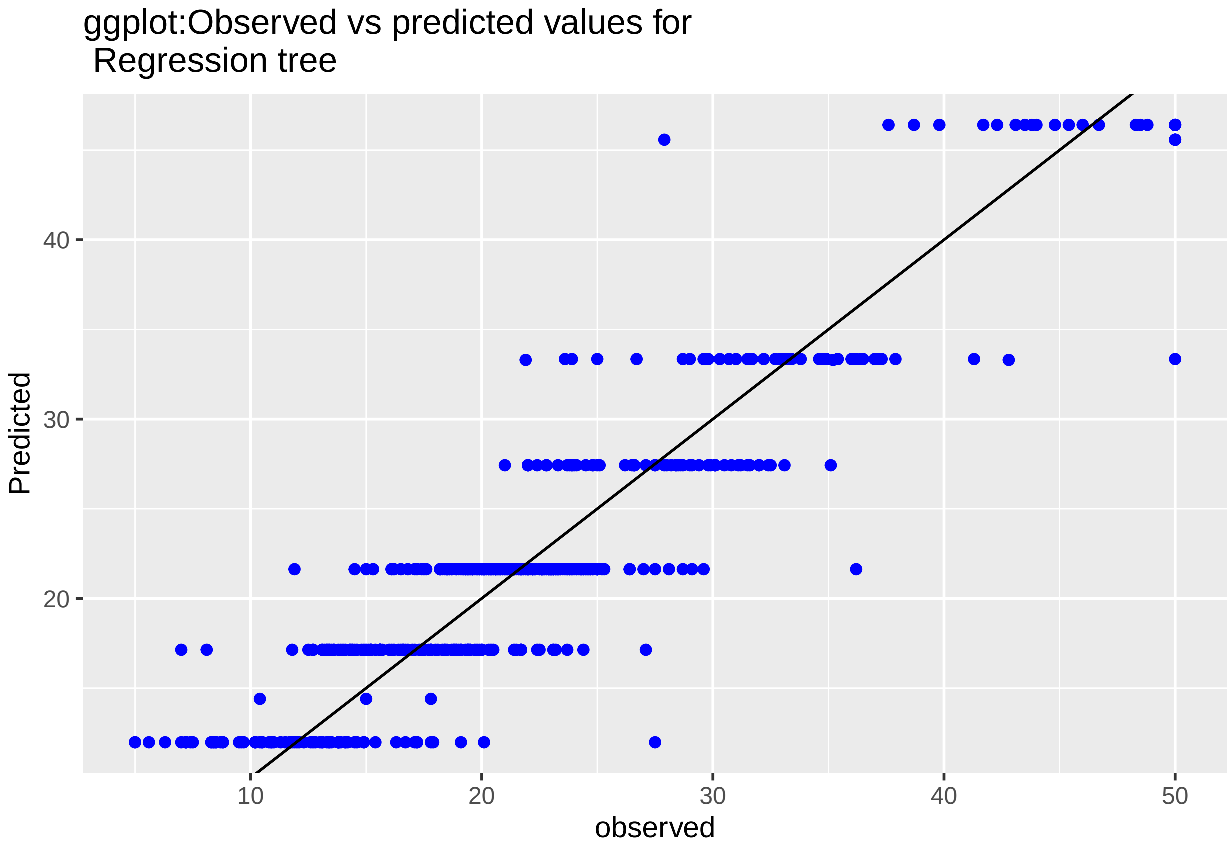
```
## n= 506
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 506 42716.3000 22.53281
##    2) rm< 6.941 430 17317.3200 19.93372
##      4) lstat>=14.4 175  3373.2510 14.95600
##        8) crim>=6.99237 74  1085.9050 11.97838 *
##        9) crim< 6.99237 101  1150.5370 17.13762 *
##      5) lstat< 14.4 255  6632.2170 23.34980
##       10) dis>=1.38485 250  3721.1630 22.90520
##         20) rm< 6.543 195  1636.0670 21.62974 *
##         21) rm>=6.543 55   643.1691 27.42727 *
##       11) dis< 1.38485 5   390.7280 45.58000 *
##    3) rm>=6.941 76  6059.4190 37.23816
##      6) rm< 7.437 46  1899.6120 32.11304
##       12) crim>=7.393425 3    27.9200 14.40000 *
##       13) crim< 7.393425 43   864.7674 33.34884 *
##      7) rm>=7.437 30  1098.8500 45.09667
##       14) ptratio>=18.3 3   223.8200 33.30000 *
##       15) ptratio< 18.3 27   411.1585 46.40741 *
```

```
##           CP nsplit rel error    xerror       xstd
## 1 0.45274420      0 1.0000000 1.0055569 0.08318656
## 2 0.17117244      1 0.5472558 0.6648488 0.06019104
## 3 0.07165784      2 0.3760834 0.4399528 0.04916012
## 4 0.05900152      3 0.3044255 0.3842313 0.04339967
## 5 0.03375589      4 0.2454240 0.2912136 0.03718954
## 6 0.02661300      5 0.2116681 0.2685244 0.03644484
## 7 0.02357238      6 0.1850551 0.2609760 0.03643787
## 8 0.01085935      7 0.1614827 0.2462587 0.03653519
## 9 0.01000000      8 0.1506234 0.2538281 0.03710374
```




Base R: Observed vs predicted values for
regression tree
Predicted Value
Observed Value
10
20
30
40
50
10
20
30
40
50

```
## integer(0)
```

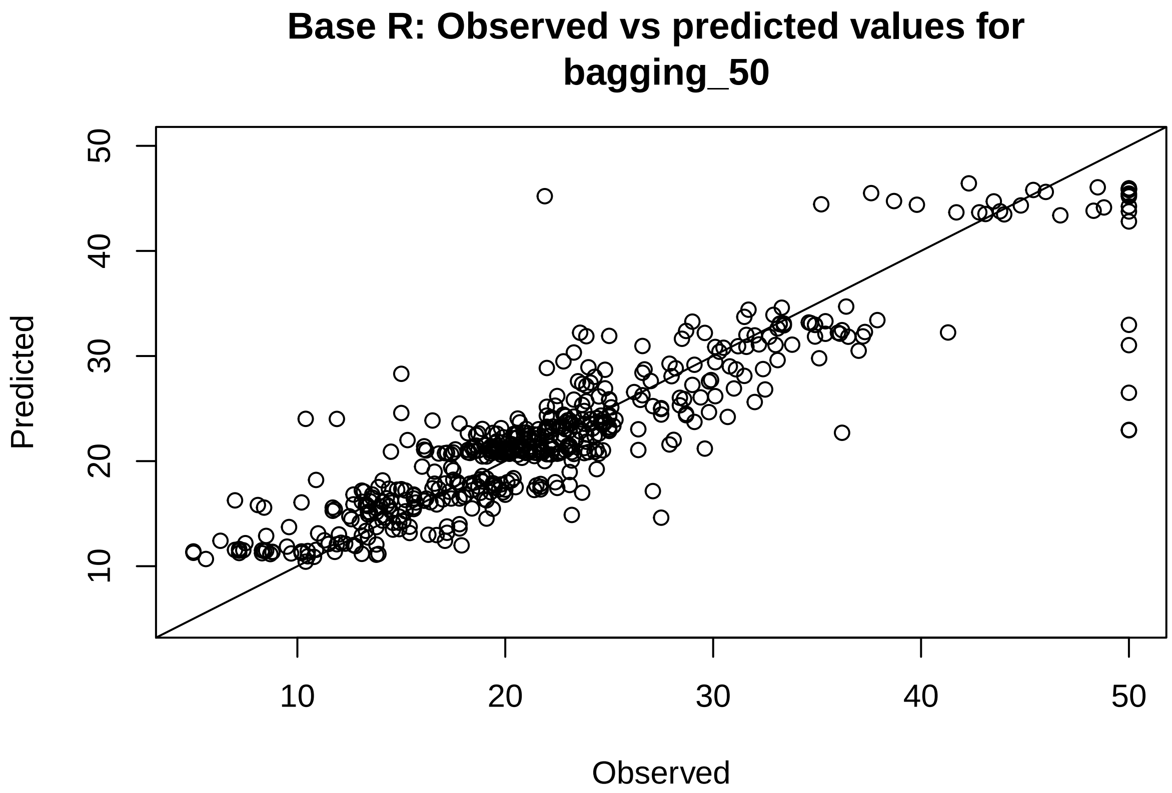


```
## The MSE for Regression Tree is:  12.71556
```

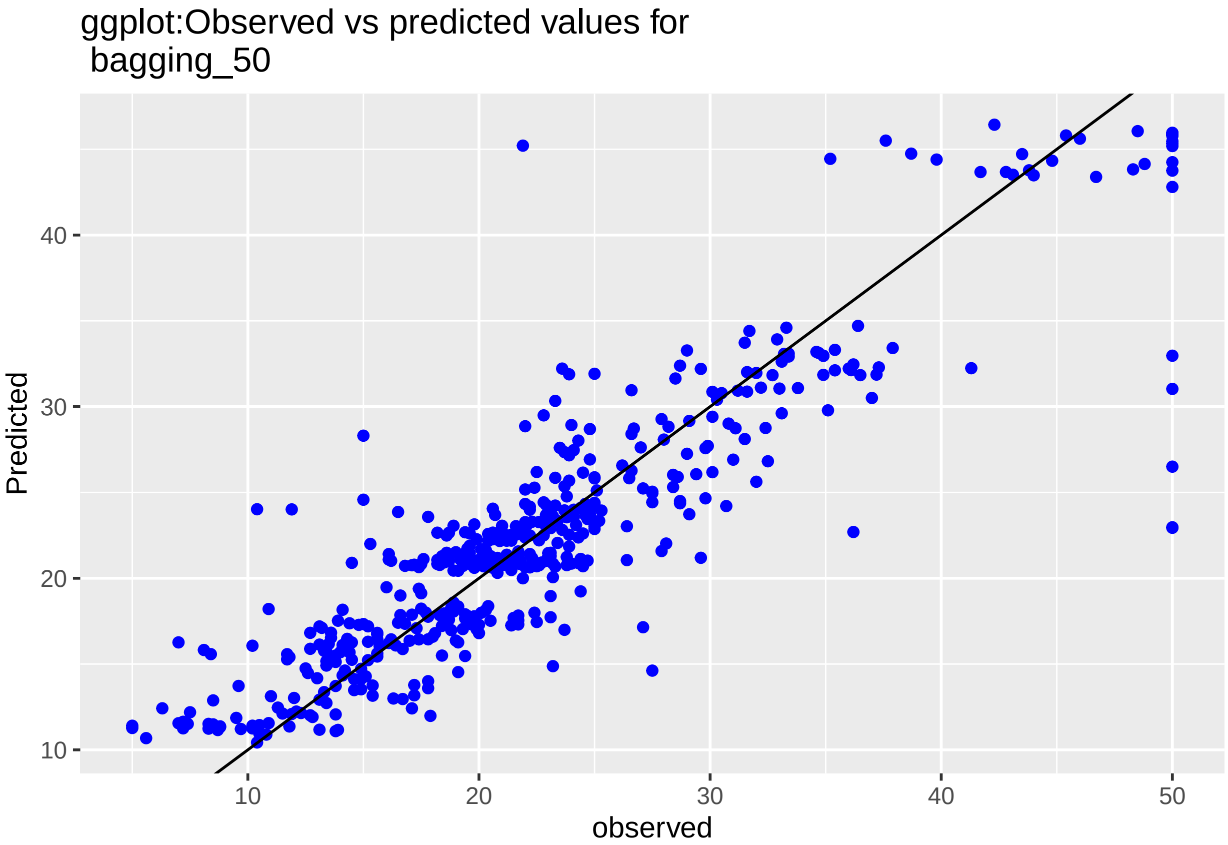
My tree has nine(9) nodes. Yes I pruned the tree but I couldn't see any changes in my tree. It already has the minimum no of tree therefore, the prune was not so effective. The predictive error of the data is 12.71556

b) Apply bagging with 50 trees. Report the prediction error (MSE) and plot the predicted vs observed values.

```
## The MSE Bagging 50 is: 17.21584
```


Base R: Observed vs predicted values for
bagging_50
Predicted
10
20
30
40
50
Observed
10
20
30
40
50

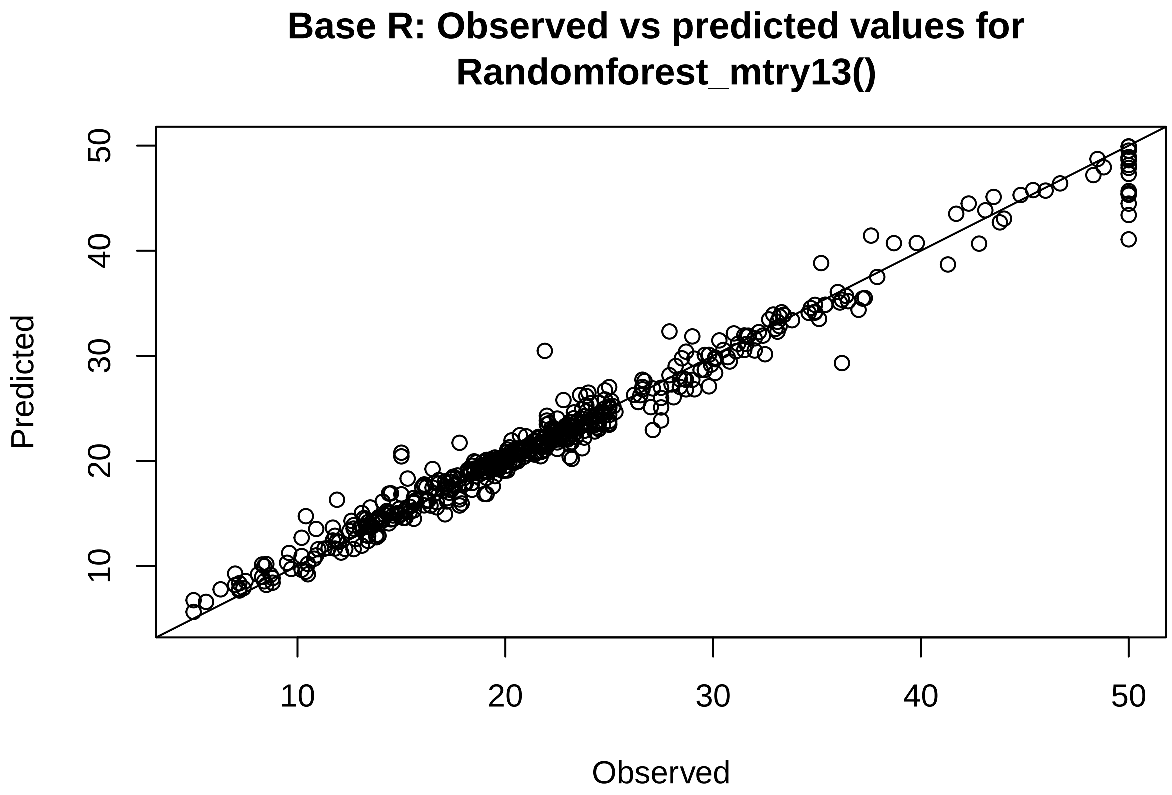
```
## integer(0)
```



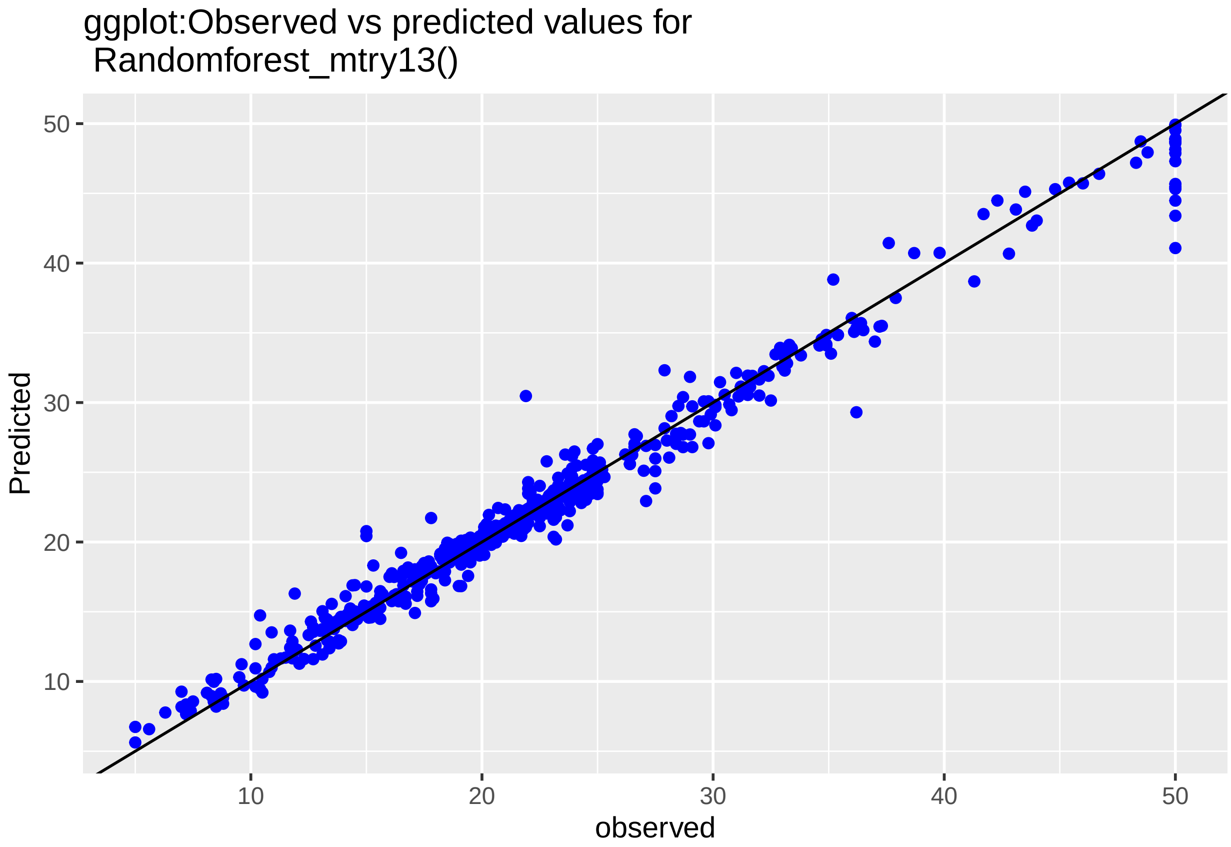
The MSE for the bagging of 50 data is 17.21584. Most of the predicted values are seen in between the 10 to 40 and very few near 50.

c) Apply bagging using the randomForest() function. Report the prediction error (MSE). Was it the same as (b)? If they are different what do you think caused it? Plot the predicted vs. observed values.

```
## MSE for Random Forest mtry13 : 2.044668
```


Base R: Observed vs predicted values for
Randomforest_mtry13()
Predicted
Observed
10
20
30
40
50
10
20
30
40
50

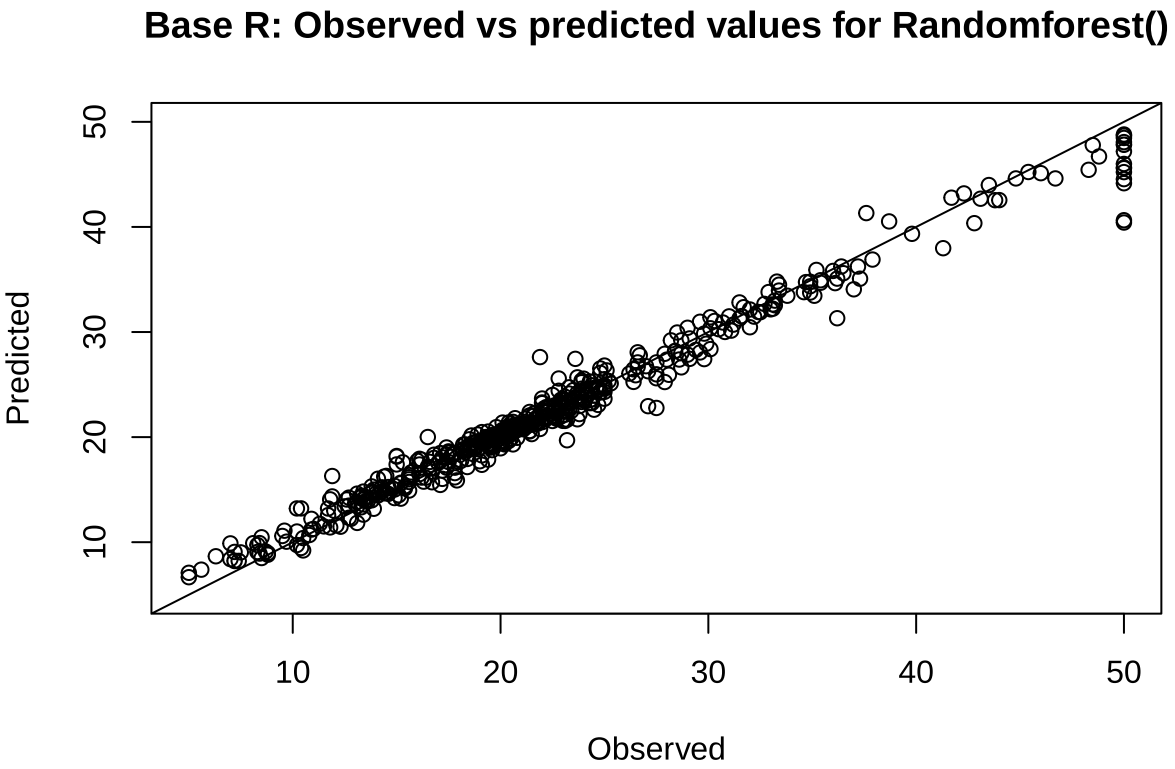
```
## integer(0)
```



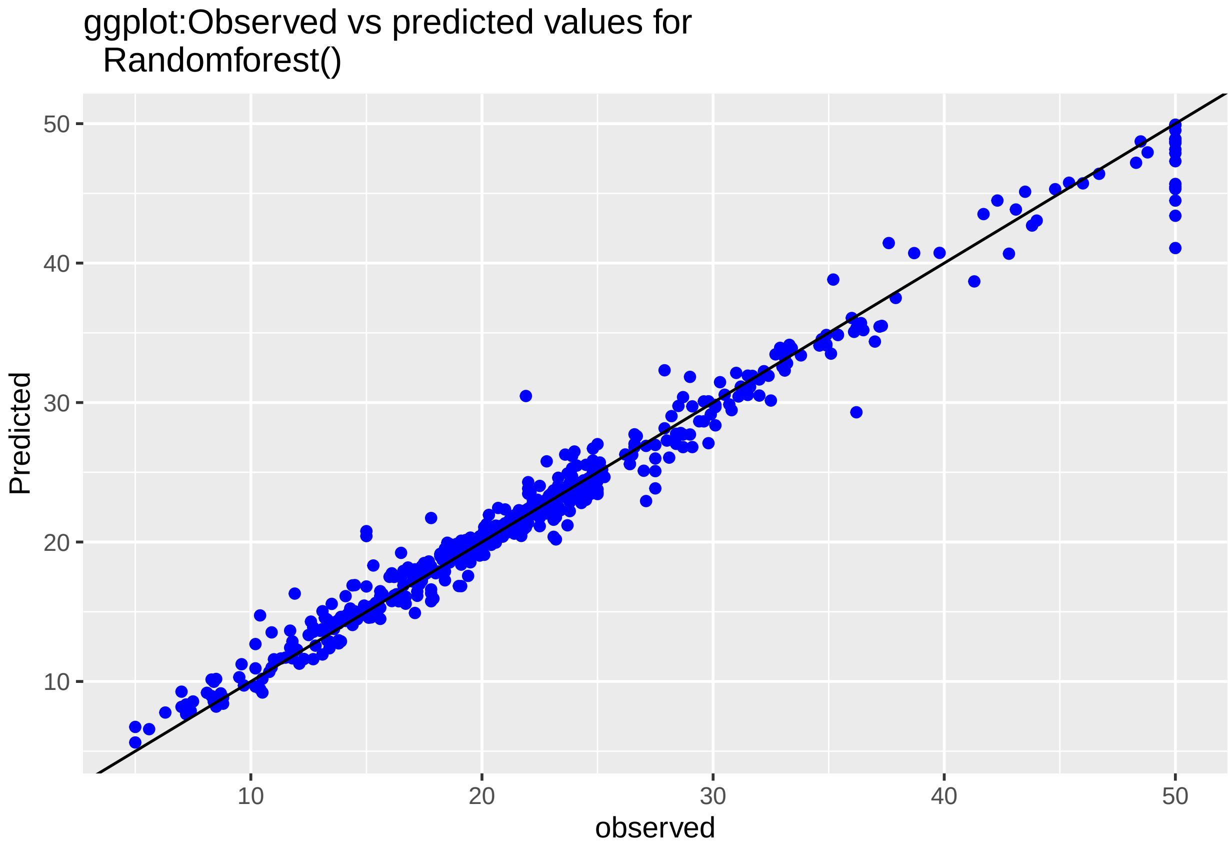
The MSE of the random_forest is 1.788203, I think the graph is some what different from the question-b . In question-b the graph is bit more scattered around the regression line. Whereas, in this question the sample are more closely packed within the lines. This might be because we used mtyr =13 which means using all the variables with 50 trees.

d) Use the randomForest() function to perform random forest. Report the prediction error (MSE). Plot the predicted vs. observed values.

```
## MSE for Random Forest: 1.994478
```


Base R: Observed vs predicted values for Randomforest()
Predicted
Observed
10
20
30
40
50
10
20
30
40
50

```
## integer(0)
```



The MSE of the Random Forest whose default no of trees is 500 is 1.946478. The observed vs predicted data looks similar to the data with the random forest with 50 trees.

e) Include a table of each method and associated MSE. Which method is more accurate?

```
##  The error table for the data
```

```
##   Regression_Tree Bagging_50 Random_Forest_mtry13 Random_Forest
## 1        12.71556   17.21584             2.044668      1.994478
```

Since, more trees is always better because of less error, Here Random forest with the default no of trees of 500 has the lowest MSE. Therefore, Random forest is more acurate.