# Recursive Partitioning

Mohamed Ahmed

Collaboration:

I collaborated with Amin baabol in the completion of the R program for part a of this recursive partitioning assignment.I worked individually on all the problems, except for the base R analogous ggplot of the decision tree and the observed vs. predicted median value ggplot in part a. Other than that, our collaboration was mainly about helping each other conceptually comprehend the various algorithms and models covered in chapter 9.

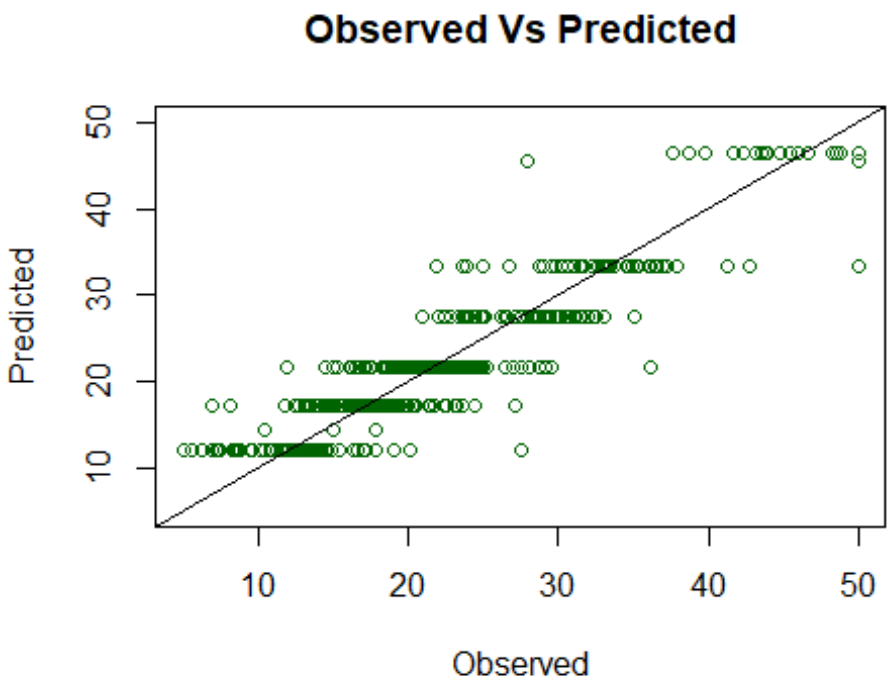
1. (Ex. 9.1 pg 186 in HSAUR, modified for clarity) The **BostonHousing** dataset reported by Harrison and Rubinfeld (1978) is available as a `data.frame` structure in the **mlbench** package (Leisch and Dimitriadou, 2009). The goal here is to predict the median value of owner-occupied homes (`medv` variable,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

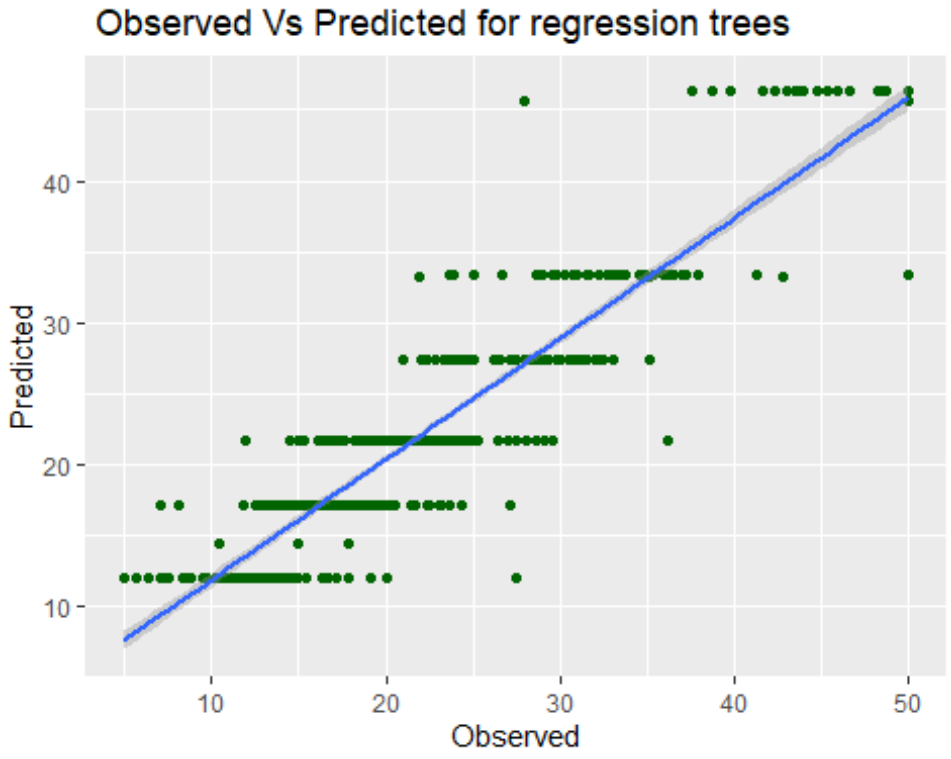
    a) Construct a regression tree using rpart(). Discuss the results, including these key components:
    - How many nodes does your tree have?
    - Did you prune the tree? Did it decrease the number of nodes?
    - What is the prediction error (MSE)?
    - Plot the predicted vs. observed values.
    - Plot the final tree.

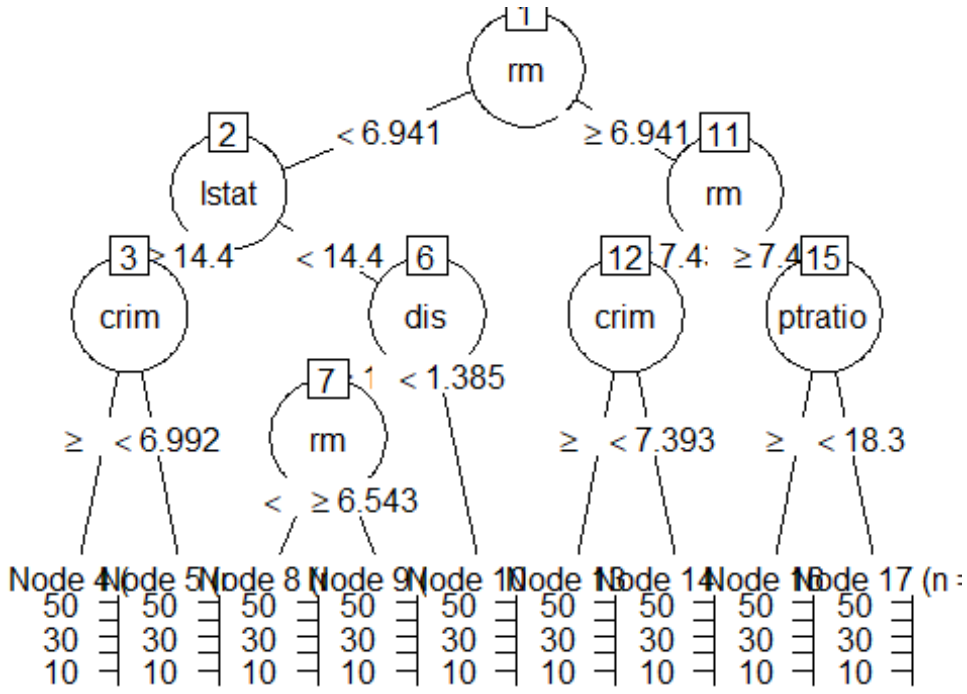
Answer: The tree has nine nodes yes, we pruned the tree at the ninth node, which had the smallest cross validation error(0.23990). No, it did not decrease the number of nodes because we pruned the tree at the last node prediction Error is 12.72

```
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 1.00416 0.083164
2 0.171172      1   0.54726 0.64131 0.060078
3 0.071658      2   0.37608 0.44371 0.050167
```

```
4 0.059002      3   0.30443 0.38218 0.045850
5 0.033756      4   0.24542 0.35226 0.044196
6 0.026613      5   0.21167 0.29357 0.035731
7 0.023572      6   0.18506 0.26635 0.035287
8 0.010859      7   0.16148 0.25854 0.035421
9 0.010000      8   0.15062 0.22473 0.031197

prediction Error 12.71556
```

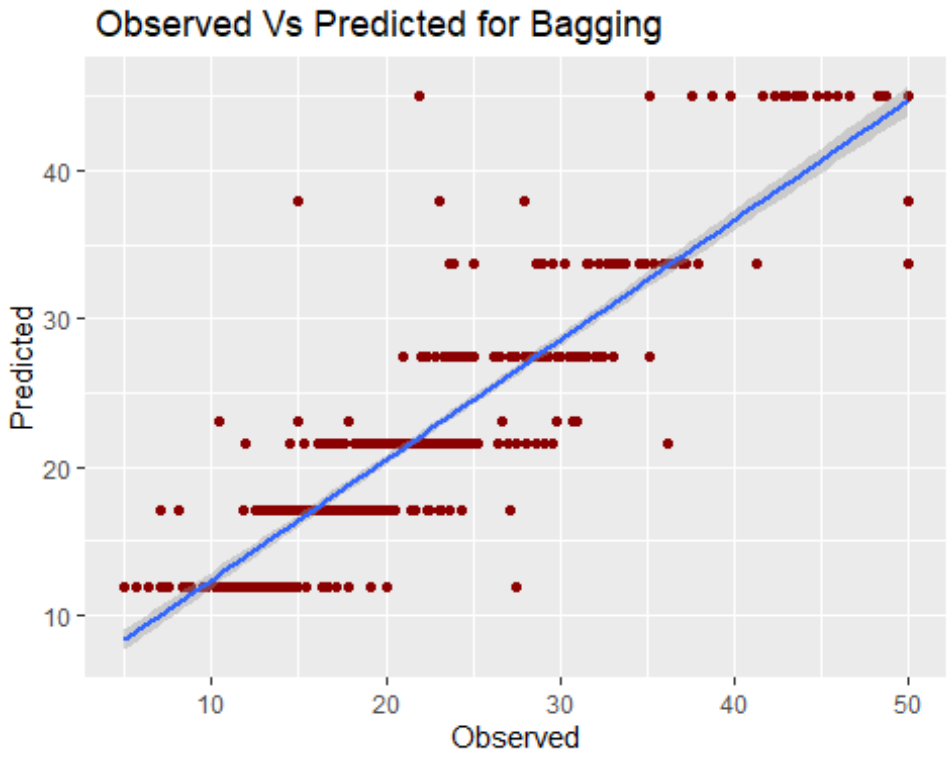

Observed Vs Predicted
Predicted
Observed
10
20
30
40
50

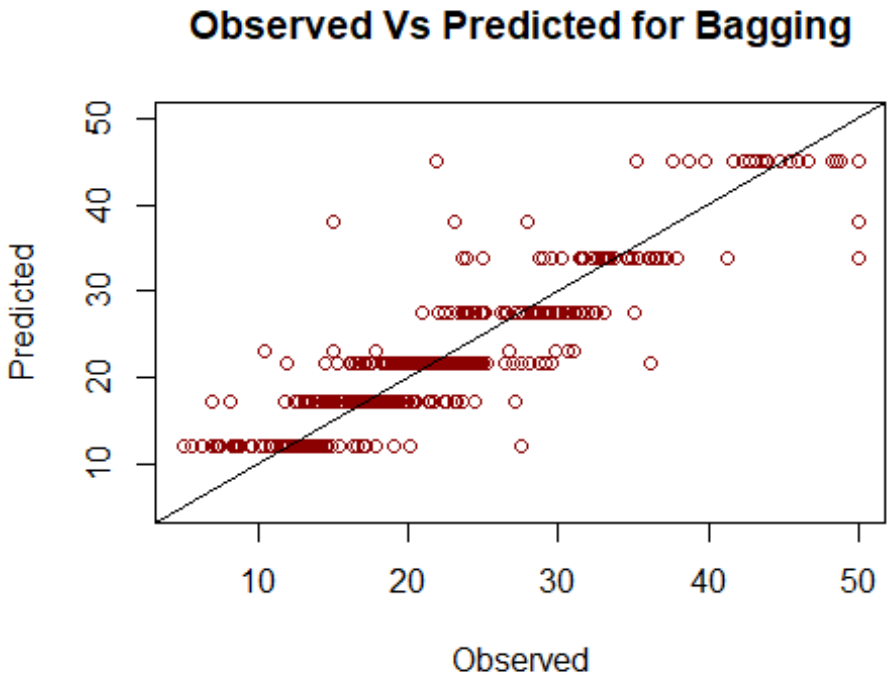

Observed Vs Predicted for regression trees
Predicted
Observed
10
20
30
40
50




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

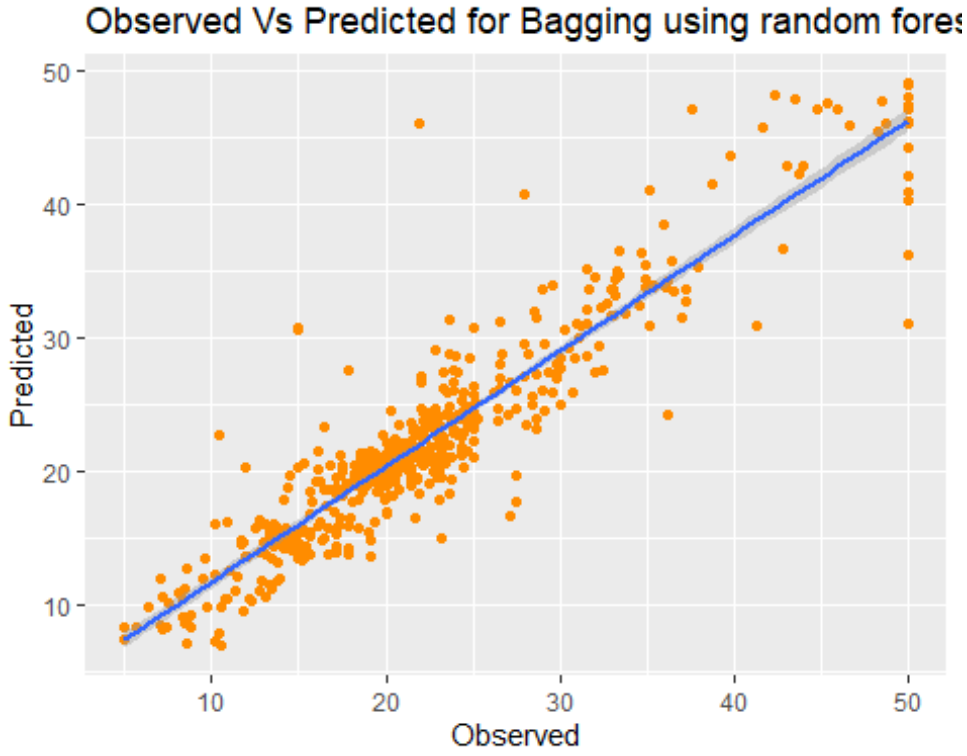
prediction Error 16.24467

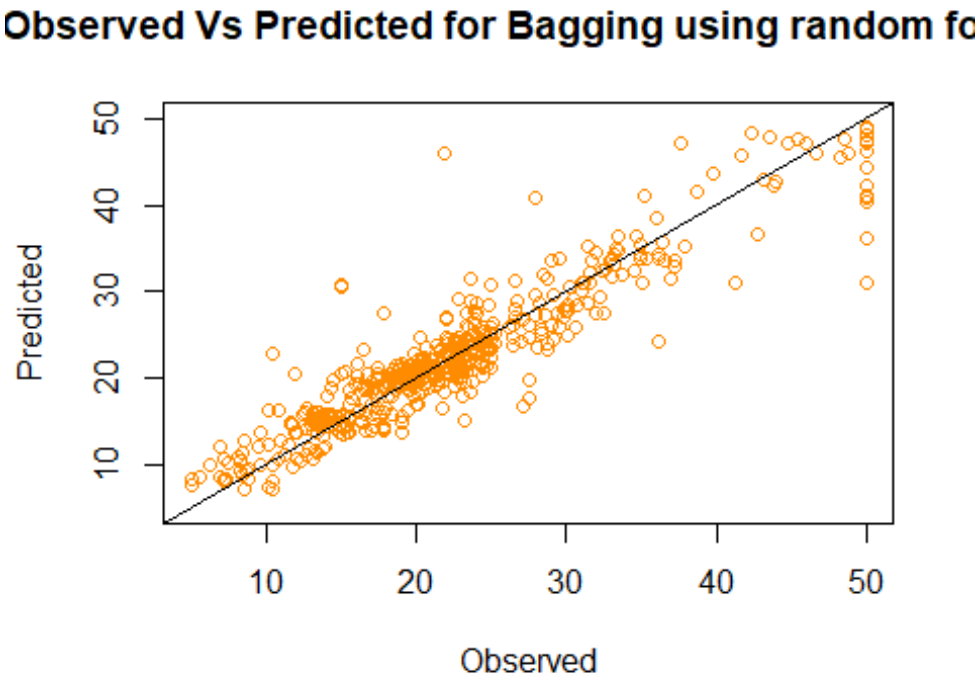



c) Apply bagging using the randomForest() function. Report the prediction error (MSE). Was it the same as (b)? If they are different what do you think caused it? Plot the predicted vs. observed values.

```
prediction Error 11.71395
```

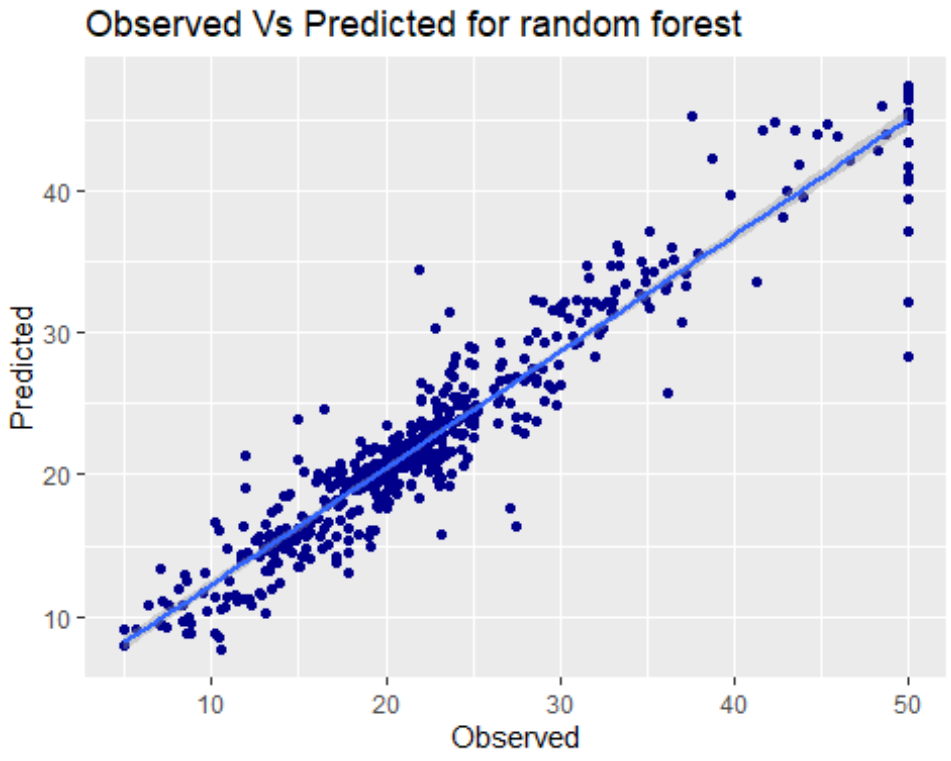


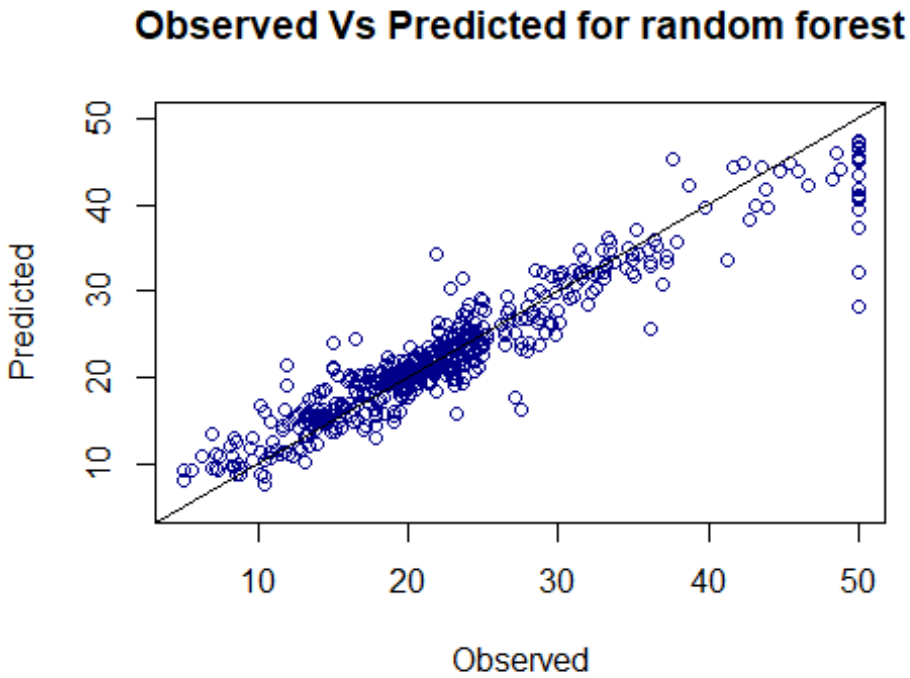




Answer: The MSE (11.71395) for part c is smaller and better than part b MSE (12.71556). The difference in MSE was caused by the way bagging and random forest work. when

choosing a split point, bagging looks through all variables and variable values and to choose the most optimal split point. However, random forests algorithm is limited to random sample of variables of which to search. Although, for this part of the question, we are specifying the number of nodes and the number of variables in order to do bagging while using the random forest algorithm.

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

prediction Error 9.476546
```




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

```
                        Method   MSE
1             Regression tree 12.72
2                     Bagging 16.24
3 Bagging using Random Forest 11.71
4               Random Forest  9.48
```

Answer: The most accurate method is Random Forest with and the least accurate method is Bagging.Random Forest is more accurate than other methods because it is limited to random sample of variables of which to search.

citations: Saunders, C. (n.d.). Chris Sanders. Retrieved October 13, 2020, from https://d2l.sdbor.edu/d2l/le/content/1452614/viewContent/8305217/View Brownlee, J. (2020, August 14). Bagging and Random Forest Ensemble Algorithms for Machine Learning. Retrieved October 14, 2020, from https://machinelearningmastery.com/bagging-and-random-forest-ensemble-algorithms-for-machine-learning/