Boston Housing

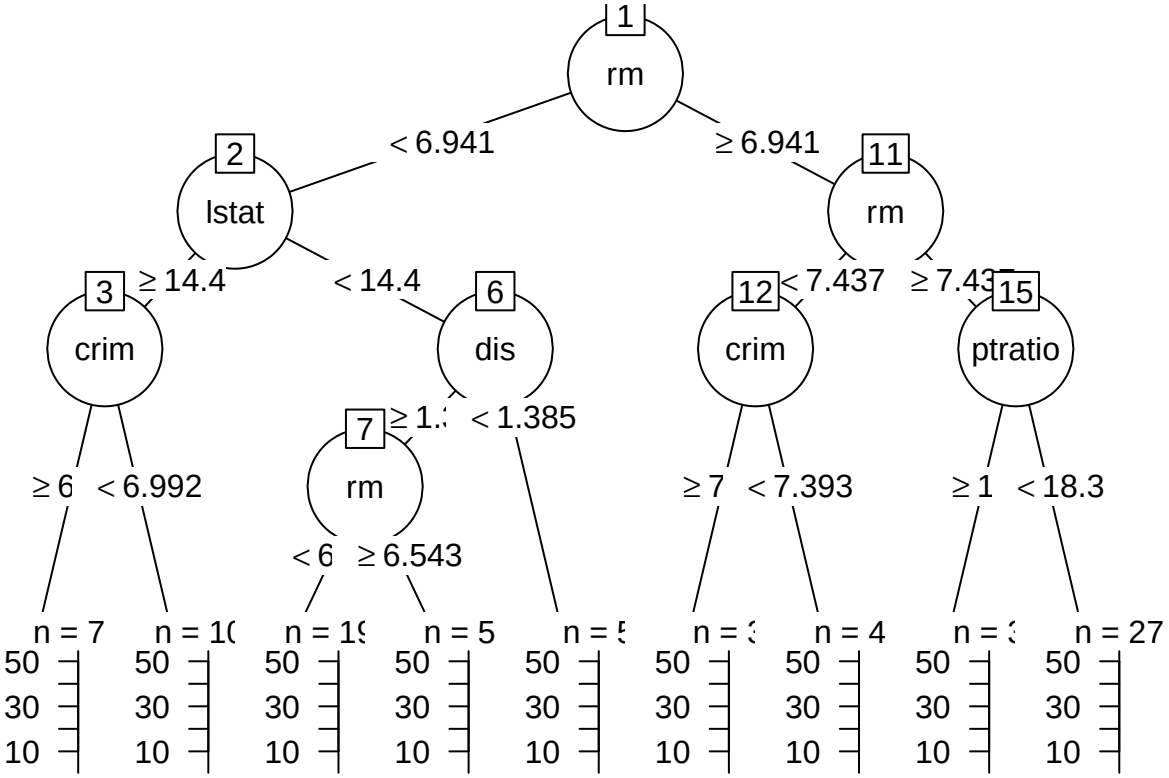
Thomas Pattara

09/25/2018

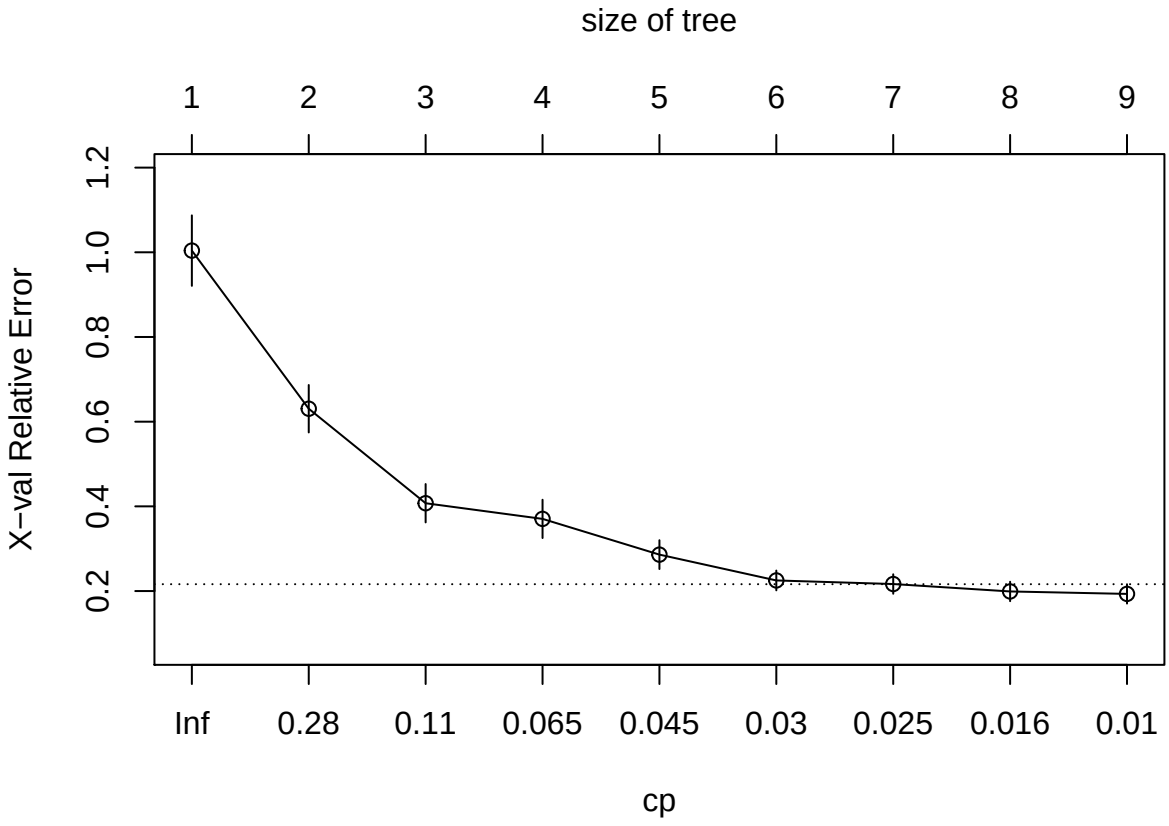
```
library(lattice)
library(randomForest)
library(partykit)
library(mboost)
library(TH.data)
library(ipred)
library(rpart)
library(mlbench)
library(ggplot2)
library(dplyr)
library(tidyr)
library(boot)
library(fastAdaboost)
```

Question 1

The **BostonHousing** dataset reported by Harrison and Rubinfeld (1978) is available as data.frame package **mlbench** (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. Use this dataset to do the following: **Construct a regression tree using rpart().**



CP Plot of the Tree



Part a

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

The lowest x error occurs at a tree size of 9 can be seen from the plot. This can also be seen with the CP table below in the next step.

CP Table of the Tree

```
##           CP nsplit rel error    xerror       xstd
## 1 0.45274420      0 1.0000000 1.0039957 0.08324106
## 2 0.17117244      1 0.5472558 0.6305946 0.05607611
## 3 0.07165784      2 0.3760834 0.4074182 0.04529558
## 4 0.05900152      3 0.3044255 0.3704079 0.04521515
## 5 0.03375589      4 0.2454240 0.2860290 0.03408051
## 6 0.02661300      5 0.2116681 0.2250617 0.02338934
## 7 0.02357238      6 0.1850551 0.2165841 0.02301406
## 8 0.01085935      7 0.1614827 0.1990835 0.02300119
## 9 0.01000000      8 0.1506234 0.1933019 0.02287841
```

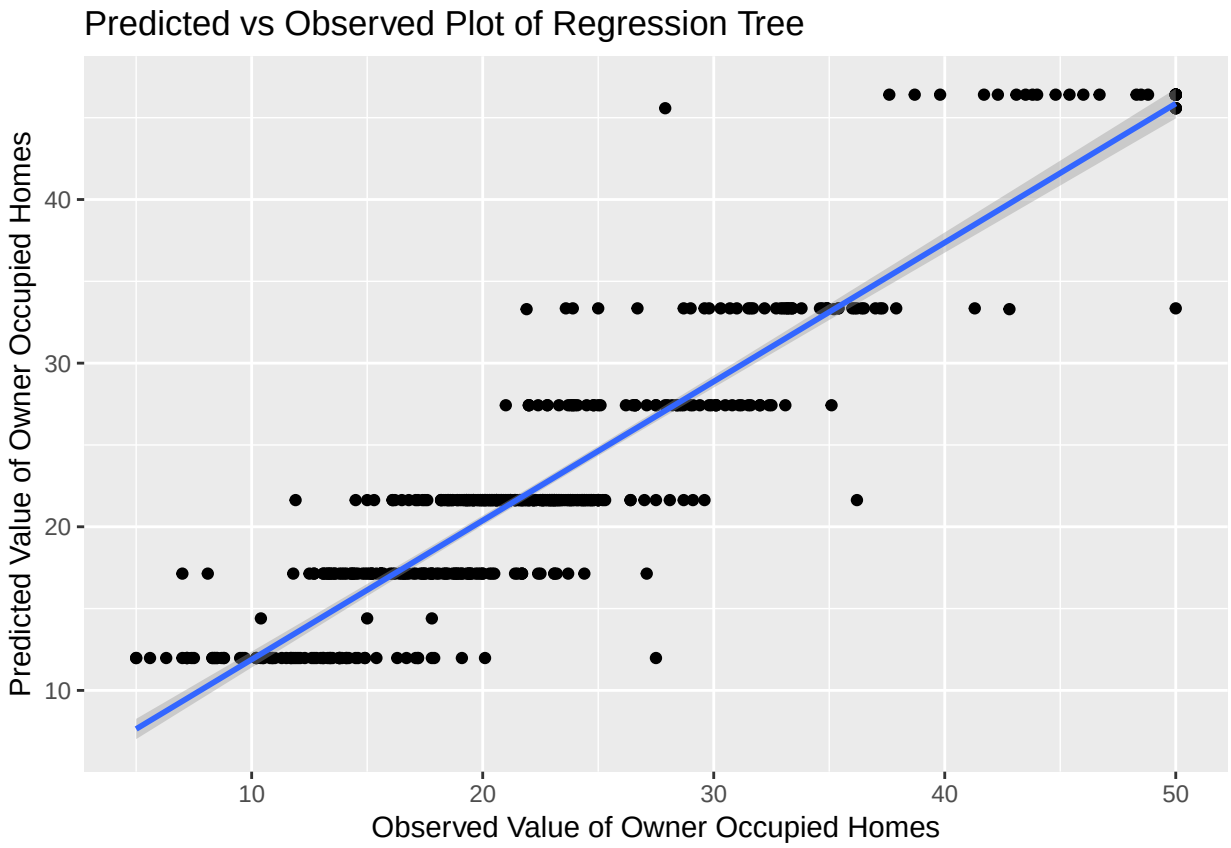
The lowest x error occurs at a tree size of 9 and an x error of 0.1933019.

MSE

```
## [1] 12.716
```

There is no reason to prune the tree because the current tree has the lowest x error available.

Fitted Vs Observed Values Plot



From the plot there is noticeably some prediction error from the fitted values.

Part b

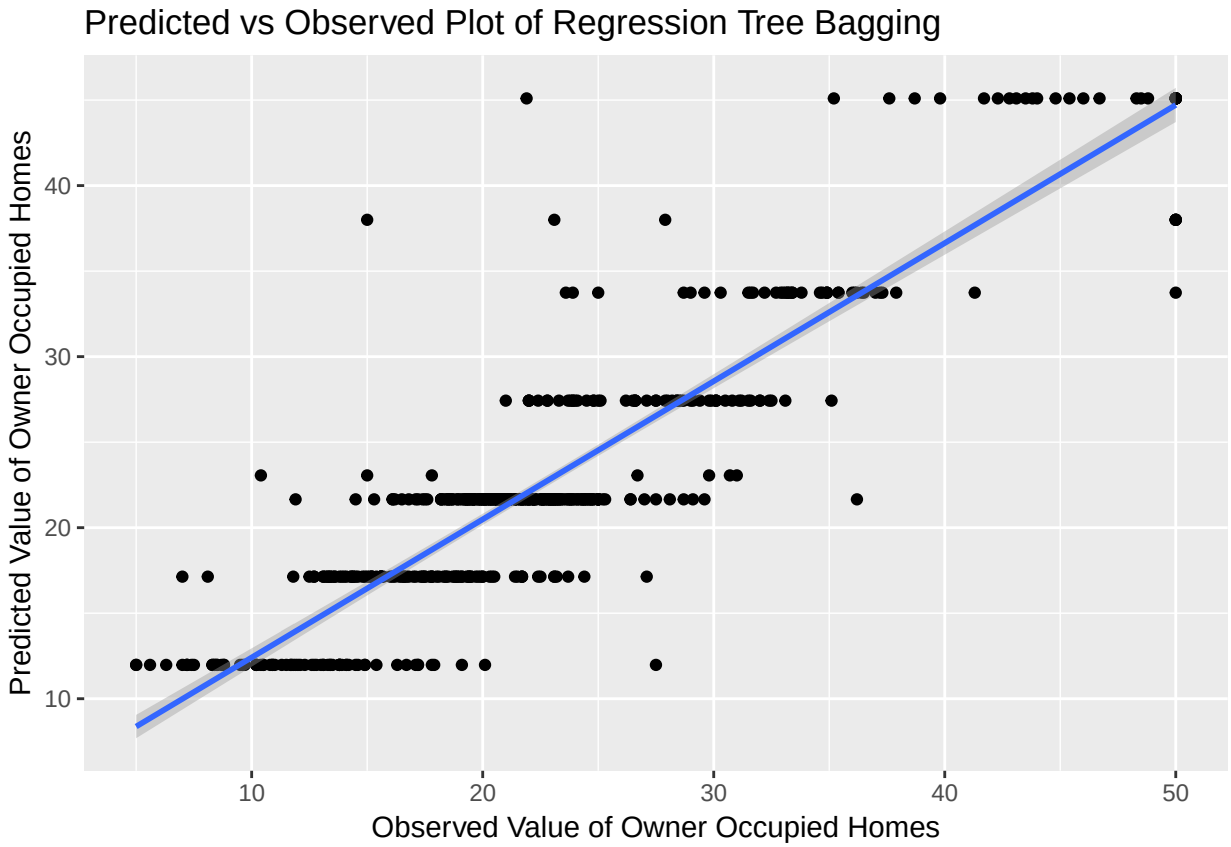
Perform bagging with 50 trees. Report the prediction error (MSE). Provide the predicted vs observed plot.

MSE Bagging

```
## [1] 16.245
```

The MSE from this bagging is higher than that of the previous tree.

Predicted Vs Observed Plot of Bagging With 50 Trees



From the predicted vs observed plot of bagging we can say that the predicted values for the medv variable are much further out than that of the trees.

Part c

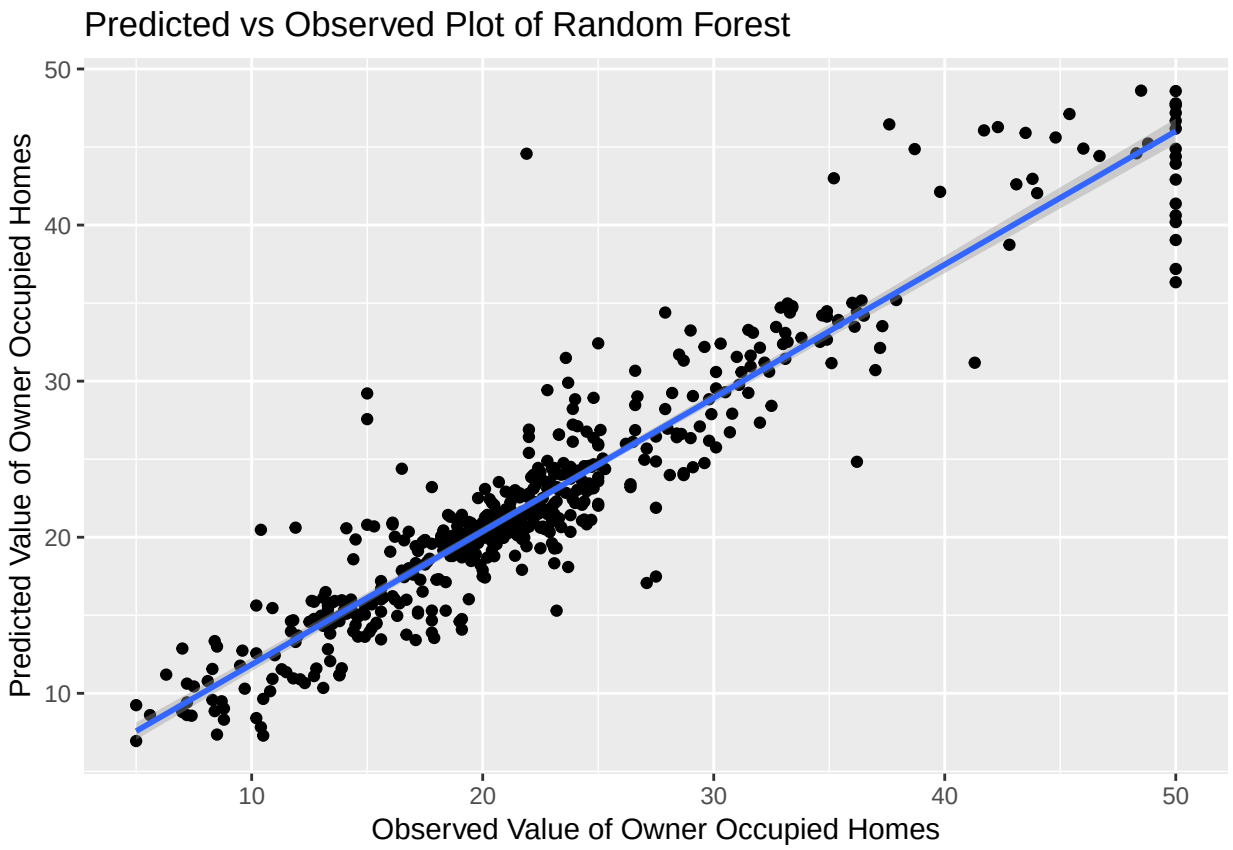
Use randomForest() function in R to perform bagging. Report the prediction error (MSE). Was it the same as (b)? If they are different what do you think caused it? Provide a plot of the predicted vs. observed values.

Build the RandomForest and Print the MSE

```
## [1] 10.355
```

The randomForest uses a random sample to build the trees whereas bagging uses all of the feautures to build the trees. There could be a problem with multicollinearity, over fitting, or some other problem with the covariates in the data. If the randomForest returns a lower error it is because the bagging example is using covariates that actually increase the error.

Plot the Predicted Vs Observed Plot of the Random Forest



We can see that there is a much lower error rate from the fitted vs observed plot.

Part d

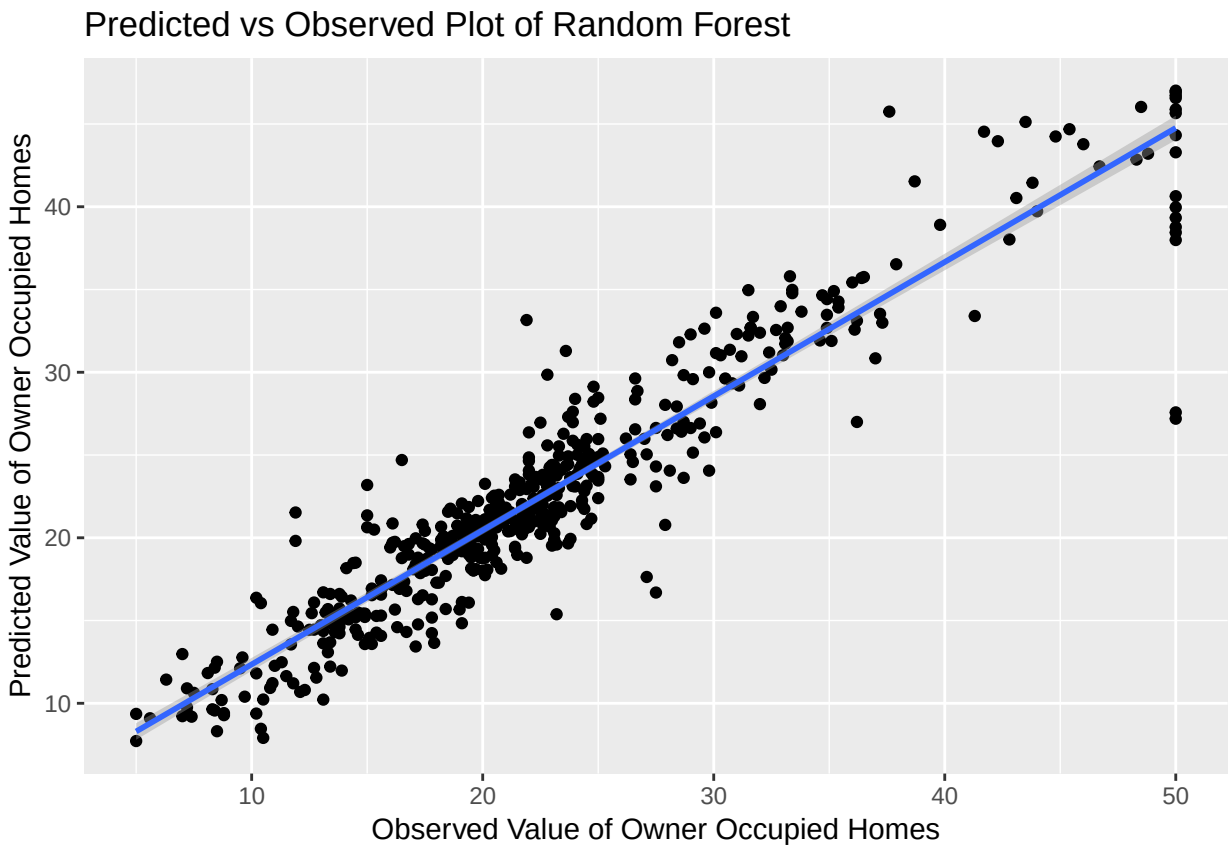
Use randomForest() function in R to perform random forest. Report the prediction error (MSE). Provide a plot of the predicted vs. observed values.

Build the RandomForest and Calculate the MSE

```
## [1] 10.117
```

The random subset method provides some validation to our above hypothesis that some of the variables in the data set are not conducive to predicting the median value of owner-occupied homes. Without defining mtry we get a lower MSE.

Fitted Vs Observed Plot of the RandomForest Plot



Part e

Provide a table containing each method and associated MSE. Which method is more accurate?

```
##    Trees Bagging RandomForest_mtry13 RandomForest
## 1 12.716  16.245              10.355       10.117
```

The most accurate model is the randomForest without the tuning parameter set. This is because the randomForest randomly selects covariates from subsets of the data without having to select from all 13 covariates in the data set. This tells me that there could be covariates that either correlate with one another or are not conducive to predicting the median value of owner-occupied homes which means this model is probably over-fitted.

Question 2

Consider the glacoma data (data = "GlaucomaM", package = "TH.data").

Part a

Build a logistic regression model. Note that most of the predictor variables are highly correlated. Hence, a logistic regression model using the whole set of variables will not work here as it is sensitive to correlation.

```
The solution is to select variables that seem to be important for predicting the response and using

Do not print out the summaries of every single model built using variable selection. That will end u
```

Stepwise Logistic Regression For Variable Selection

```
## Class ~ phct + phci + vbrn + vari + tms + mr + rnf + emd
```

The step function to step through the model and find the combination of variables with the lowest AIC. Before performing stepwise regerssion with will remove all variables with a correlation absolute value greater than 0.8. The formula function will capture the formula for this model and use it throughout the problem.

Part b

Build a logistic regression model with K-fold cross validation (k = 10). Report the error rate.

logistic regression model with 10 fold cross validation

```
## [1] 0.1
```

The error rate using the 10 fold method is 0.1.

Part c

Find a function (package in R) that can conduct the "adaboost" ensemble modeling. Use it to predict glaucoma and report error rate. Be sure to mention the package you used.

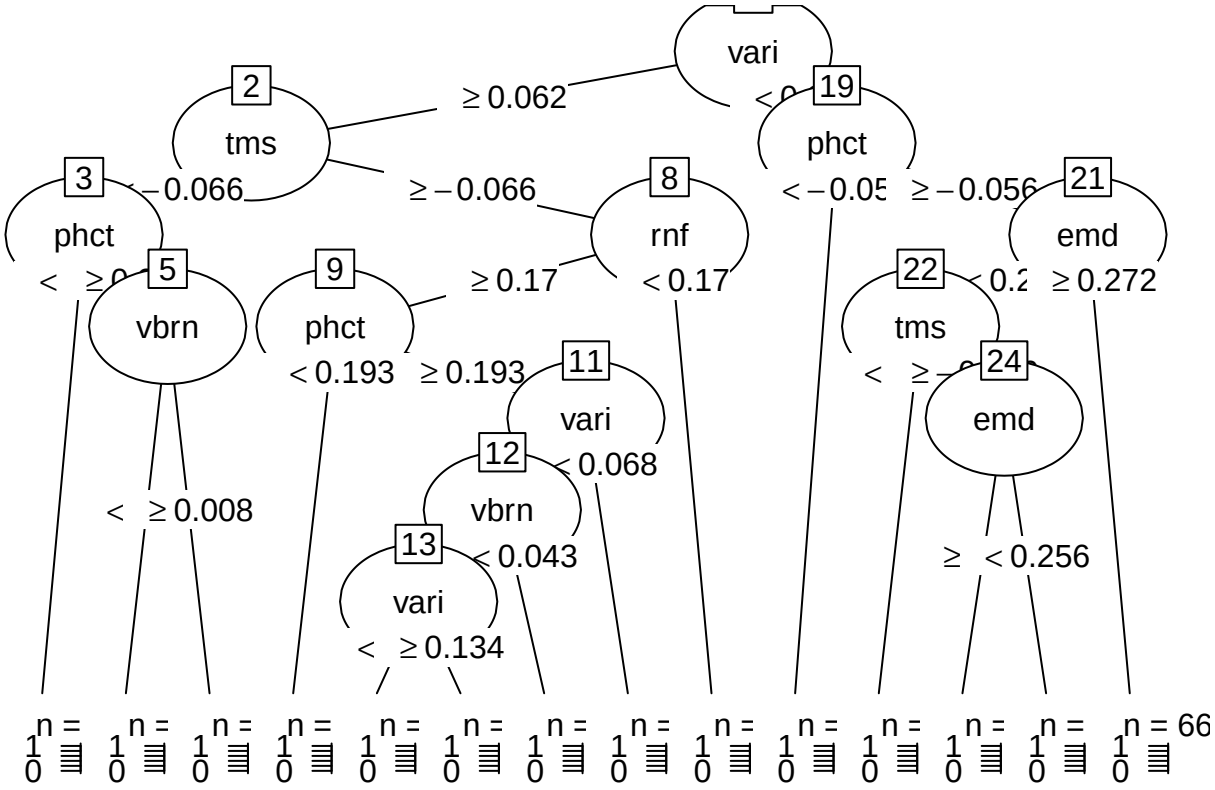
```
## [1] 0
```

The function used for this problem is called adaboost from the package fastAdaboost to create the adaboost. This algorithm predicts glaucoma perfectly with a zero error rate.

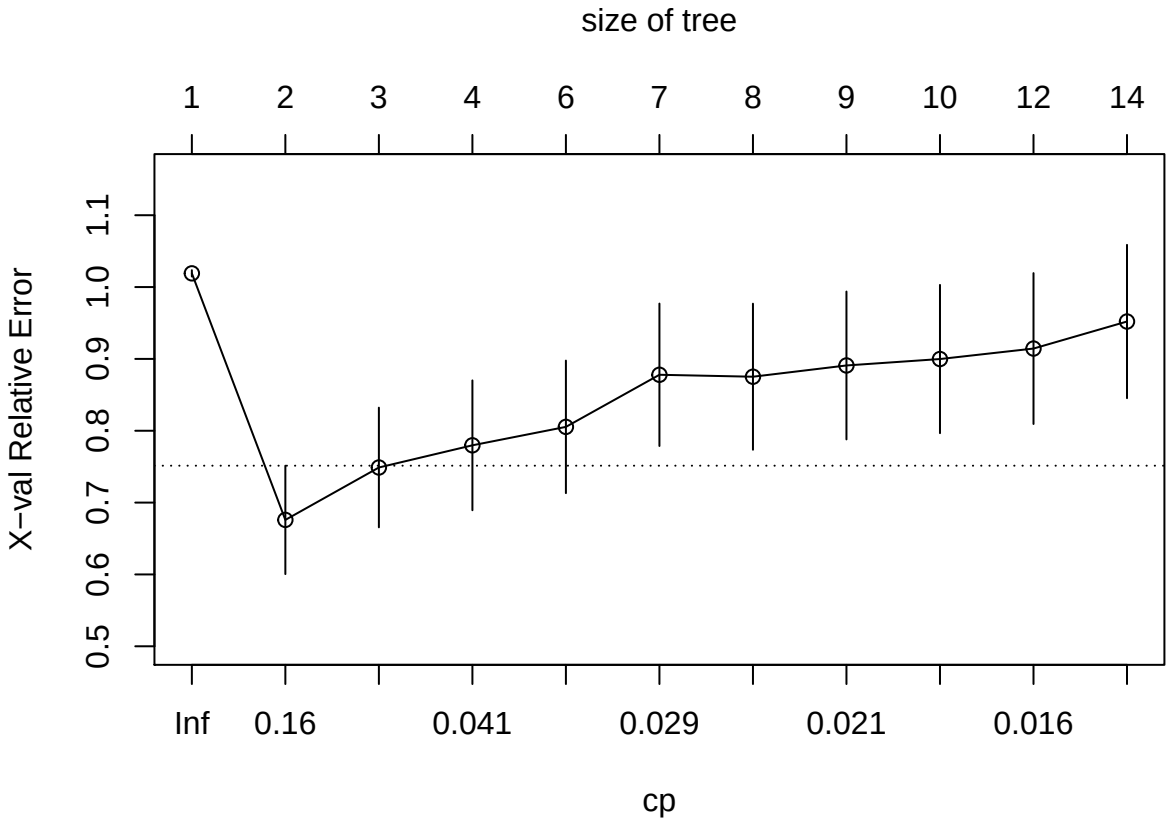
Part d

Report the error rates based on single tree, bagging and random forest. (A table would be great for this).

Tree From the GlaucomaM Dataset



Plot the CP of the Tree



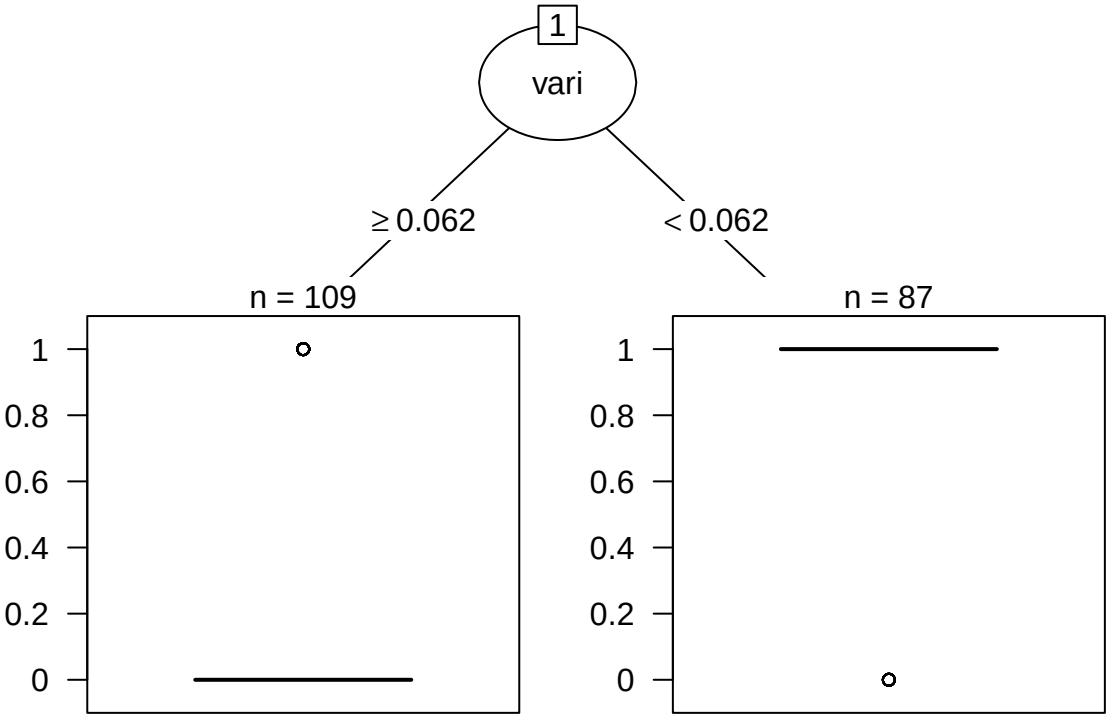
We can say from the increased error rate that pruning needs to be done.

Print the CP Table of the Glauc Tree

```
##            CP nsplit rel error    xerror        xstd
## 1  0.41853844      0 1.0000000 1.0190767 0.004429992
## 2  0.06359004      1 0.5814616 0.6759317 0.075428733
## 3  0.04630239      2 0.5178715 0.7488561 0.083287691
## 4  0.03670627      3 0.4715691 0.7797163 0.090366973
## 5  0.03537562      5 0.3981566 0.8054005 0.092242351
## 6  0.02305684      6 0.3627810 0.8779052 0.099119021
## 7  0.02093145      7 0.3397241 0.8752049 0.101656330
## 8  0.02083691      8 0.3187927 0.8908552 0.102923205
## 9  0.01834305      9 0.2979558 0.8998133 0.103185397
## 10 0.01438561     11 0.2612697 0.9143890 0.105006340
## 11 0.01000000     13 0.2324985 0.9520380 0.106703287
```

Examining the x error it appears the optimum number of nodes is 2.

Tree with Pruning



The tree produced from pruning only has two nodes. Splitting a tree like this will not be very predictive for Glaucoma.

10 Fold Cross Validation on the Tree

```
## [1] 0.03333
```

RandomForest With 10 Fold Cross Validation

```
## [1] 0.055
```

Bagging With 50 & 10 Fold Cross Validation

```
## [1] 0.0879
```

Bagging with 10 fold cross validation produces an error rate very close to that of the glm.

Table of Errors from All Models

```
##                          V1
## GLM Error           0.10000
## AdaBoost Error      0.00000
## Tree Error          0.03333
## randomForest Error 0.05500
## Bagging Error       0.08790
```

The adaboost algorithm is the clear choice here with an error rate of zero. The randomForest still has substantial prediction power and may actually be beneficial on a different data set of the same test.

Part e

Write a conclusion comparing the above results (use a table to report models and corresponding error rates). Which one is the best model?

The AdaBoost algorithm had a zero error rate and predicted the Class variable perfectly. Since the same 10 fold cross validation method was not used in the adaboost algorithm the randomForest may still be usable for predicting glaucoma.

Part f

From the above analysis, which variables seem to be important in predicting Glaucoma?

```
##
## Call:
## glm(formula = Class ~ phct + phci + vbrn + vari + tms + mr +
##     rnf + emd, family = "binomial", data = GlaucomaM)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.17940  -0.53719   0.01616   0.48980   3.08580
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)    5.485      2.781   1.972   0.0486 *
## phct           5.571      3.306   1.685   0.0919 .
## phci           6.859      2.946   2.328   0.0199 *
## vbrn          -5.171      2.366  -2.185   0.0289 *
## vari         -15.743      9.093  -1.731   0.0834 .
## tms            6.946      2.426   2.863   0.0042 **
## mr            -4.778      3.063  -1.560   0.1188
## rnf          -12.198      5.788  -2.107   0.0351 *
## emd            7.778      3.254   2.390   0.0168 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 271.71  on 195  degrees of freedom
## Residual deviance: 134.90  on 187  degrees of freedom
## AIC: 152.9
##
## Number of Fisher Scoring iterations: 6
```

The tms variable has the highest significance followed by: emd, rnf, vbrnn, phci, phct, vari, and the mr variables. This means the moment superior has the highest significance when predicting glaucoma from the tests.