Sayan Chakraborti

Eric Rupinski

**Housing Values in Suburbs of Boston**

Overview of Project on Kaggle:

<https://www.kaggle.com/c/boston-housing/overview>

Data Source on Kaggle:

<https://www.kaggle.com/c/boston-housing/overview>

Online CSV download of data I conveniently stored:

<https://docs.google.com/spreadsheets/d/e/2PACX-1vRfbdE0TtF6kWyvAEQVcRzhCf5H5A0lw7KDdjyYAaTiVVONcS357weVP-iYotmTnEnYXrJdGU3BMTqJ/pub?gid=1515730767&single=true&output=csv>

**WARNING**

Before running my code I would recommend installing the following packages: regclass, tidyverse, rstanarm, tree, rpart, randomForest

**CODE**

# Sayan Chakraborti

# Eric Rupinski

# Final Project Stat486

# saving my data in an online google doc and importing it into my R session

myurl <- "https://docs.google.com/spreadsheets/d/e/2PACX-1vRfbdE0TtF6kWyvAEQVcRzhCf5H5A0lw7KDdjyYAaTiVVONcS357weVP-iYotmTnEnYXrJdGU3BMTqJ/pub?gid=1515730767&single=true&output=csv"

boston <- read.csv(url(myurl))

#plotting with median home value as (y), and all other variables as (x)

# add in lowess lines as well

plot(boston$crim, boston$medv, main="Crime and Home Prices", xlab = "per capita crime rate by town.", ylab = "home price")

lines(lowess(boston$crim,boston$medv), col = "red")

plot(boston$zn, boston$medv, main= "Residential Land Zoned and Home Prices", xlab = "proportion of residential land zoned for lots over 25,000 sq.ft.", ylab = "home price")

lines(lowess(boston$zn,boston$medv), col = "red")

plot(boston$indus, boston$medv, main="Industry and Home Prices", xlab = "proportion of non-retail business acres per town.", ylab = "home price")

lines(lowess(boston$indus,boston$medv), col = "red")

plot(boston$chas, boston$medv, main="Charles River Dummy and Home Prices", xlab = "Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).", ylab = "home price")

lines(lowess(boston$chas,boston$medv), col = "red")

plot(boston$nox, boston$medv, main="Nitrogen Oxdes and Home Prices", xlab = "nitrogen oxides concentration (parts per 10 million).", ylab = "home price")

lines(lowess(boston$nox,boston$medv), col = "red")

plot(boston$rm, boston$medv, main="Rooms/dwelling and Home Prices", xlab = "average number of rooms per dwelling.", ylab = "home price")

lines(lowess(boston$rm,boston$medv), col = "red")

plot(boston$age, boston$medv, main="Older Houses and Home Prices", xlab = "proportion of owner-occupied units built prior to 1940.", ylab = "home price")

lines(lowess(boston$age,boston$medv), col = "red")

plot(boston$dis, boston$medv, main="Employment centres and Home Prices", xlab = "weighted mean of distances to five Boston employment centres.", ylab = "home price")

lines(lowess(boston$dis,boston$medv), col = "red")

plot(boston$rad, boston$medv, main="Highways and Home Prices", xlab = "index of accessibility to radial highways.", ylab = "home price")

lines(lowess(boston$rad,boston$medv), col = "red")

plot(boston$tax, boston$medv, main="Taxes and Home Prices", xlab = "full-value property-tax rate per $10,000.", ylab = "home price")

lines(lowess(boston$tax,boston$medv), col = "red")

plot(boston$ptratio, boston$medv, main="PTRatio and Home Prices", xlab = "pupil-teacher ratio by town.", ylab = "home price")

lines(lowess(boston$ptratio,boston$medv), col = "red")

plot(boston$black, boston$medv, main="Black and Home Prices", xlab = "proportion of blacks by town.", ylab = "home price")

lines(lowess(boston$black,boston$medv), col = "red")

plot(boston$lstat, boston$medv, main="Lower Status and Home Prices", xlab = "lower status of the population (percent).", ylab = "home price")

lines(lowess(boston$lstat,boston$medv), col = "red")

# here I bootstrap a train and test data set

v1<-sort(sample(10000,6000))

boston.train<-boston[v1,]

boston.test<-boston[-v1,]

library(regclass)

# first I try simple multiple linear regression on the data set

medv.simple\_linear.train <- glm(medv~crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+black+lstat, data=boston.train)

# look at summary of model and check for multicollinearity

summary(medv.simple\_linear.train)

VIF(medv.simple\_linear.train)

# assess the accuracy of this specific model

medv.simple\_linear.predict<-predict(medv.simple\_linear.train, boston.test)

plot(medv.simple\_linear.predict, boston.test$medv)

cor(medv.simple\_linear.predict, boston.test$medv)

# now I try a multiple polynomial regression on the data set

medv.poly\_2.train <- glm(medv~I(crim^2)+I(zn^2)+I(indus^2)

+I(chas^2)+I(nox^2)+I(rm^2)+I(age^2)+I(dis^2)+I(rad^2)

+I(tax^2)+I(ptratio^2)+I(black^2)+I(lstat^2), data=boston.train)

# look at summary of model and check for multicollinearity

summary(medv.poly\_2.train)

VIF(medv.poly\_2.train)

# assess the accuracy of this specific model

medv.poly\_2.predict<-predict(medv.poly\_2.train, boston.test)

plot(medv.poly\_2.predict, boston.test$medv)

cor(medv.poly\_2.predict, boston.test$medv)

# Bayesian multiple first order linear regression

library(tidyverse)

library(rstanarm)

medv.bayes\_linear<- stan\_glm(

medv~crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+black+lstat,

data = boston.train

)

# Gives us the trace plots, a 95% prediction interval and a posterior predictive check

plot(medv.bayes\_linear, plotfun = "trace")

plot(medv.bayes\_linear, plotfun = "dens")

posterior\_interval(medv.bayes\_linear)

pp\_check(medv.bayes\_linear)

summary(medv.bayes\_linear)

# Hierarchical Bayesian Linear Model

medv.HIERARCHICALbayes\_linear <- stan\_glmer(

medv~crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+black+lstat

+ (1 | lstat),

control = list(adapt\_delta = 0.99),

data = boston.train

)

posterior\_interval(medv.HIERARCHICALbayes\_linear)

pp\_check(medv.HIERARCHICALbayes\_linear)

summary(medv.HIERARCHICALbayes\_linear)

library(tree)

library(rpart)

library(randomForest)

# useful plots to understand my decision tree steps and see if

# random forest classifier converges or not

medv.tree.train <- tree(medv~crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+black+lstat, data=boston.train)

medv.rpart.train <- rpart(medv~crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+black+lstat, data=boston.train)

medv.forest.train <- randomForest(medv~crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+black+lstat, data=boston.train, na.action = na.roughfix)

par(mfrow=c(1,1))

plot(medv.tree.train)

text(medv.tree.train)

plot(medv.rpart.train)

text(medv.rpart.train)

plot(medv.forest.train)

medv.tree.predict<-predict(medv.tree.train, boston.test)

medv.rpart.predict<-predict(medv.rpart.train,boston.test)

medv.forest.predict<-predict(medv.forest.train,boston.test)

# plots for the accuracy of my tree, rpart and random forest models

plot(medv.tree.predict, boston.test$medv)

plot(medv.rpart.predict, boston.test$medv)

plot(medv.forest.predict, boston.test$medv)

# get the correlations of my tree, rpart and random forest models

cor(medv.tree.predict, boston.test$medv)

cor(medv.rpart.predict, boston.test$medv)

cor(medv.forest.predict, boston.test$medv)

**ANALYSIS REPORT AND CODE EXPLANATION**

My Project will be predicting the median home value in Boston using a variety of different variables. For my final project I will be building a Home Value Estimation Tool that can predict the price of a real estate asset using several publicly accessible variables. Most Home Value Estimation Tools are often very unreliable have an error of about 30% on average. This means that if you own a home worth 100k most estimation tools would estimate a home value between 70k-130k. This is very problematic because real estate is the largest investment Americans make and not knowing the true value of your home is incredibly financially irresponsible. Also knowing an accurate predicted value of your home gives you an advantage when it comes to buying or selling your home.

My dependent variable is medv...

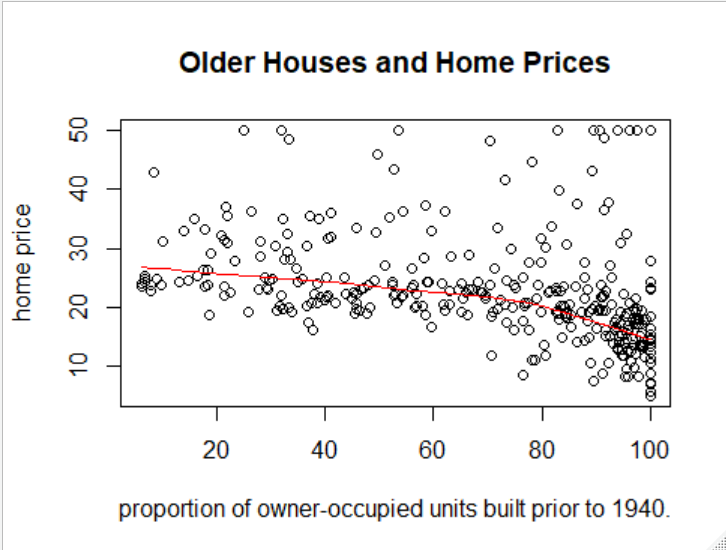
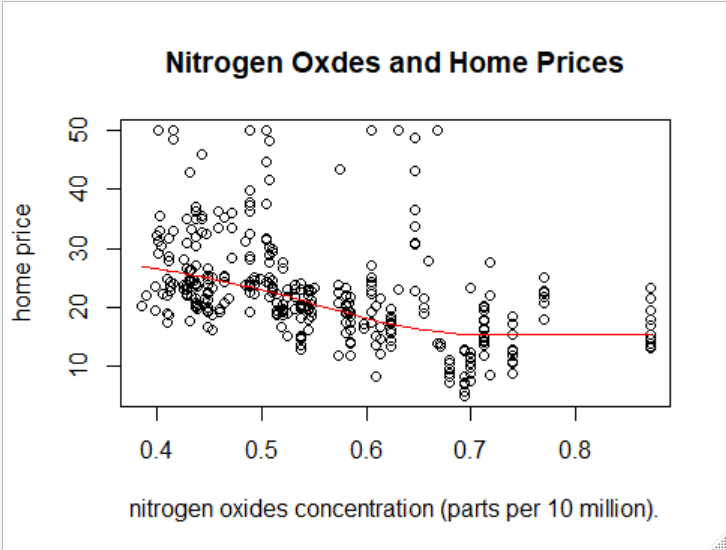
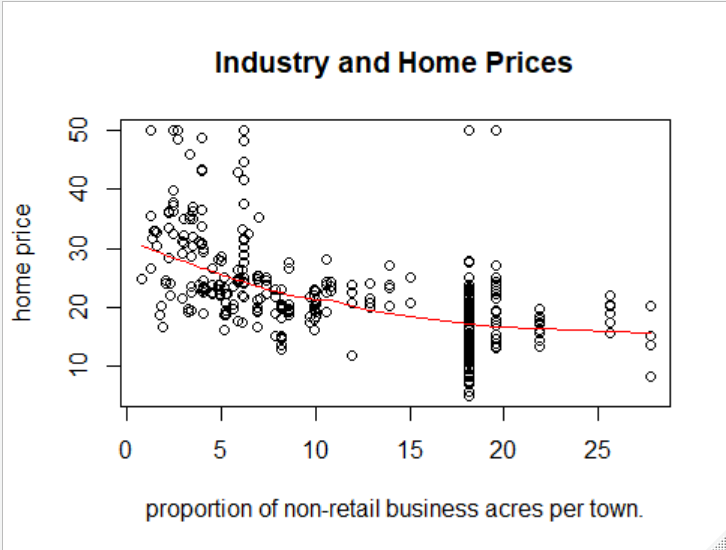
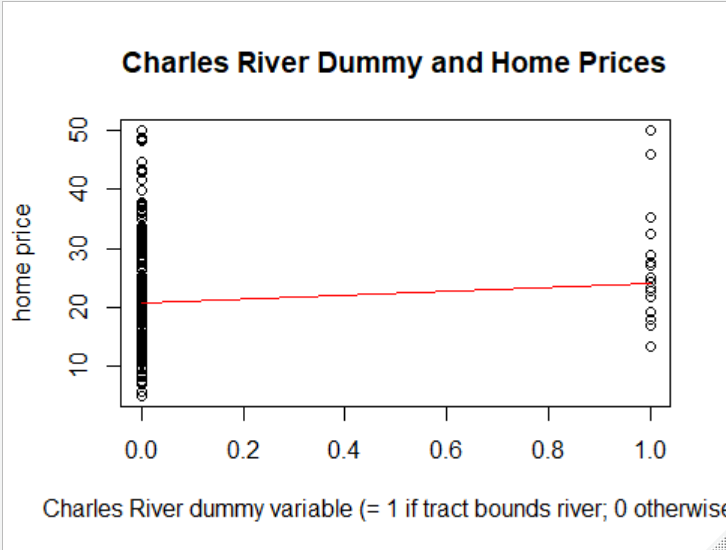
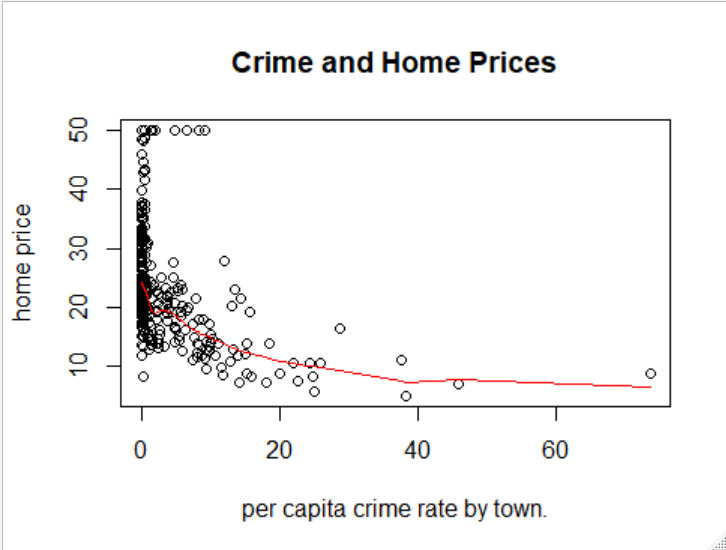
* **Medv**- median value of owner-occupied homes in $1000s.

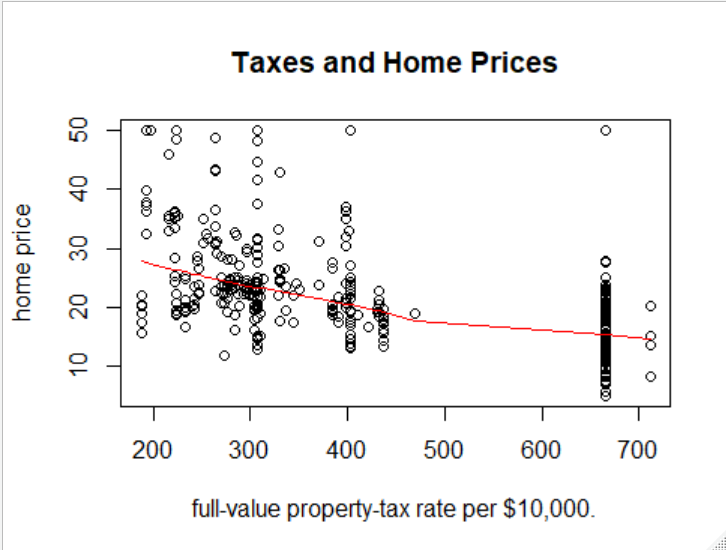
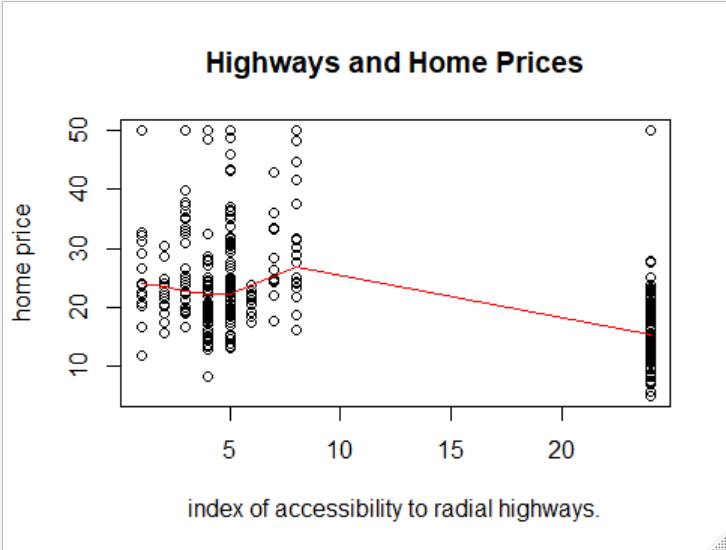
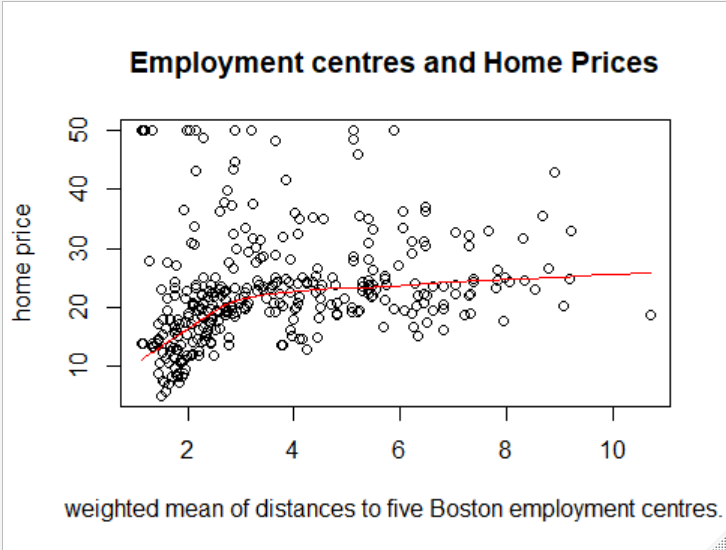
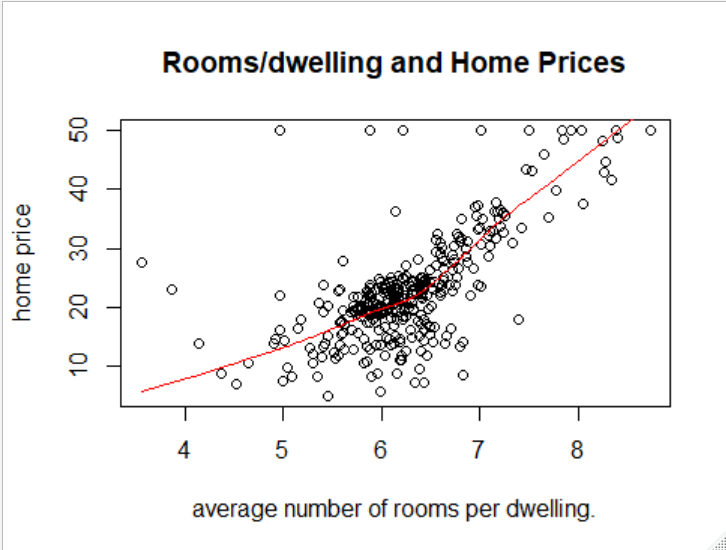
The following are my independent variables...

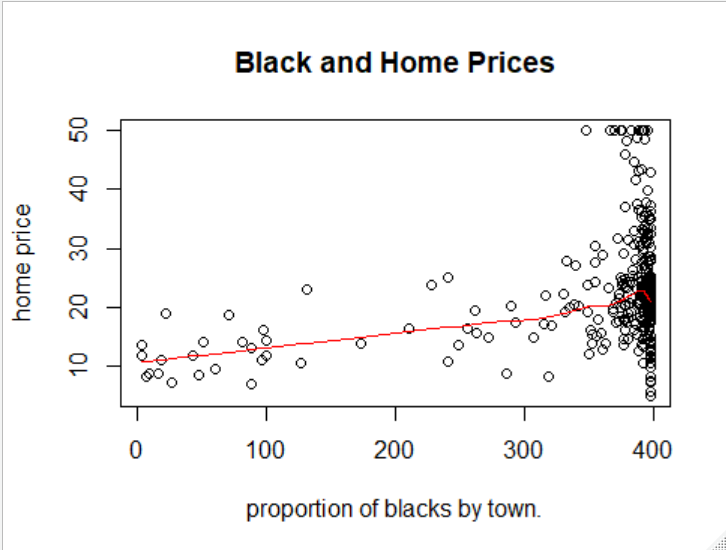
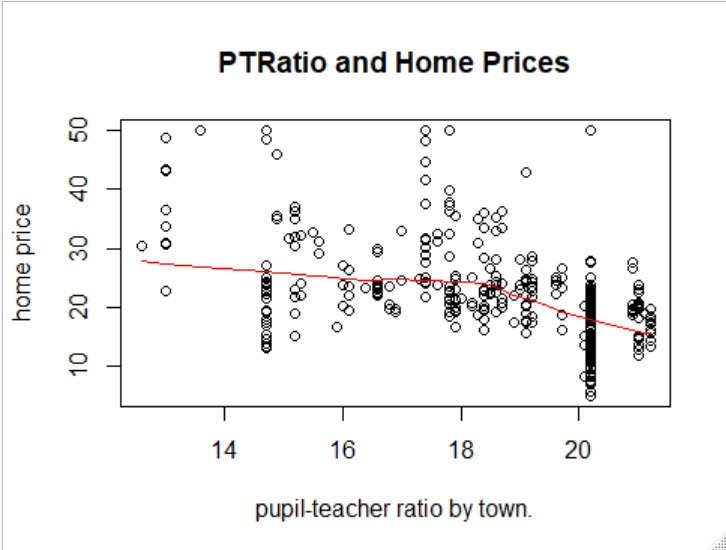
* **Crim**- per capita crime rate by town.
* **Zn**- proportion of residential land zoned for lots over 25,000 sq.ft.
* **Indus**- proportion of non-retail business acres per town.
* **Chas**- Charles River dummy variable (= 1 if tract bounds river; 0 otherwise).
* **Nox**- nitrogen oxides concentration (parts per 10 million).
* **Rm**- average number of rooms per dwelling.
* **Age**- proportion of owner-occupied units built prior to 1940.
* **Dis**- weighted mean of distances to five Boston employment centres.
* **Rad-** index of accessibility to radial highways.
* **Tax**-full- value property-tax rate per $10,000.
* **Ptratio**- pupil-teacher ratio by town.
* **Black**- 1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town.
* **Lstat**- lower status of the population (percent).

**ANALYSIS OF CODE:**

To begin my analysis I first start by importing my data into the proper places. I chose to store the CSV file of my data on the internet because it was easier to import than if I stored the file locally on my computer. Next I plotted every single combination of dependent and independent variables. On top of each plot I plotting a lowess line in each plot to help visualize what is going on in the data set. After doing this I use the pairs function to get an overall idea of each plot. Here are the following plots with lowess lines in them…





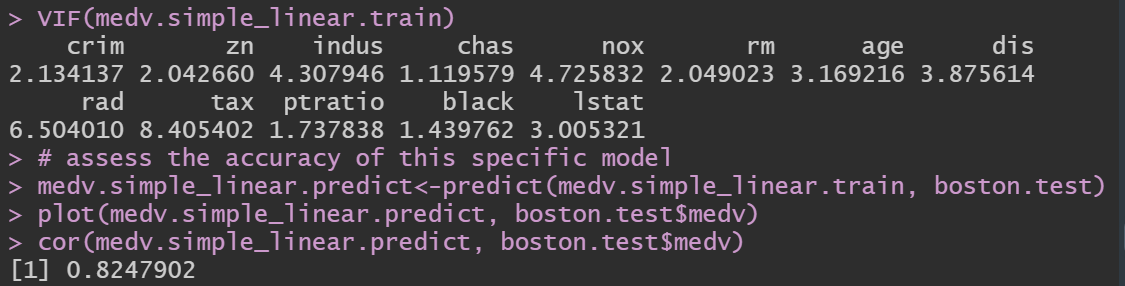




**Multiple Linear Model and Multiple Polynomial Model**

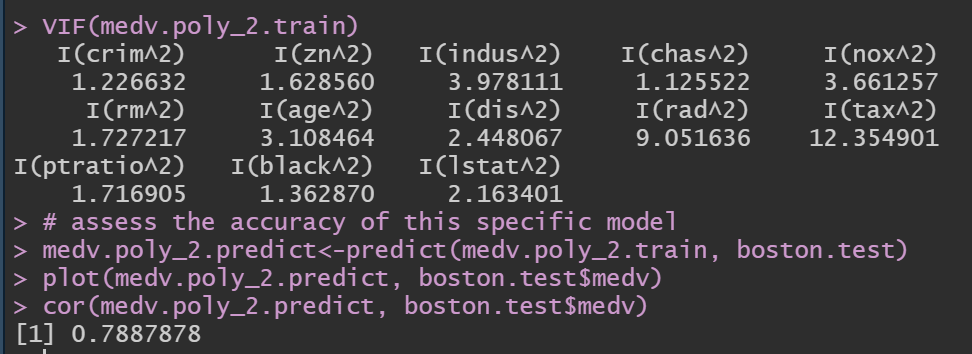
For all of my models I bootstrap a train and test data set to help train my models and to test the accuracy of each model.

My first model that I build is a multiple linear model that uses the following formula (medv~crim+zn+indus+chas+nox+rm+age+dis+rad+tax+ptratio+black+lstat). I give this model my train data set and do a number of tests. By looking at the Pr(>|t|) column I can tell what variables will be statistically significant or not. And by looking at the Pr(>|t|) I can see that the following variables are statistically significant in my multiple linear model: rm, dis, rad, ptratio, lstat). My residual deviance for my multiple linear model is 3947.1 on 199 degrees of freedom. Now multicollinearity is a problem when we build models because it makes it tough for us to analyze what variables are useful and what variables are not. By looking at my Variable Inflation Factor I will be able to see what variables may cause me some problems. For all of my models in my project I will have a VIF tolerance of 10. This means that if my VIF for any predictor variable is greater than 10 I may have to tweak the model to fix that. After looking at the VIF for my multiple linear model I can see that no VIF values are greater than 10.



Next I investigate the correlation of this model by testing it to the outcome variable of my test data set. The outcome variable for my test data set is: medv. Medv represents the median home value of a property. To conclude my analysis for this model I will have to assess the accuracy of the model itself. For my project I will assess the accuracy of each model by looking at the correlation between my model and my test data. For my multiple linear model I have a correlation of .**82479.** This means that my model is doing fairly well compared to my test data. But I will continue my analysis till I have a higher correlation

Next I move onto my multiple polynomial model. For this model I will be using a quadratic polynomial model and my model has the following formula (medv~I(crim^2)+I(zn^2)+I(indus^2) +I(chas^2)+I(nox^2)+I(rm^2)+I(age^2)+I(dis^2)+I(rad^2)+I(tax^2)+I(ptratio^2)+I(black^2)+I(lstat^2). I give this model my train data set and do a number of tests. By looking at the Pr(>|t|) column I can tell what variables will be statistically significant or not. And by looking at the Pr(>|t|) I can see that the following variables are statistically significant in my multiple polynomial model: rm, dis, ptratio, lstat. My residual deviance for this model is 4613 on 199 degrees of freedom. I test for multicollinearity by using the VIF() function. If any predictor variable has a VIF greater than 10 I will consider that an issue. By looking at the VIF function I see that the tax is a variable that has a VIF of 12.354901.



This indicates to me that tax correlates with another variable in my model and I will have to fix this issue if I want to pursue this model for my final model for this project. My correlation for my multiple polynomial model to my test data is **0.7887.** I am unsatisfied with a correlation that low and will pursue more analysis till I reach a model with a higher correlation.

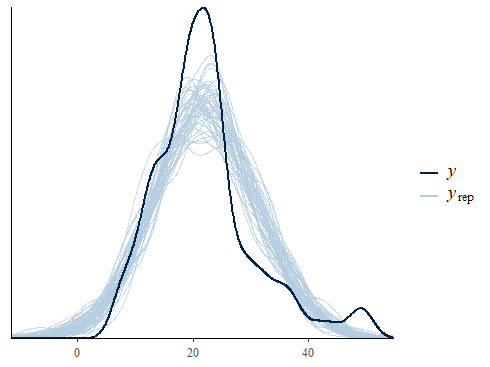
**Bayesian Analysis: Bayesian Regression and Hierarchical Bayesian Regression**

Most of statistics follows a frequentist approach but when it comes to model building. Bayesian Analysis has some advantage because it allows us to make an almost “scientific method” type of approach to statistics. Bayesian Statistics is very popular in the medical research community because of this major advantage. I will be using Markov Chain Monte Carlo (MCMC) methods for my Bayesian Analysis.

By looking at my trace plots I can see If my MCMC Chains are running smoothly or not. By looking at the trace plots for each variable I see that my MCMC is running very well. The MCMC warms up very nicely.



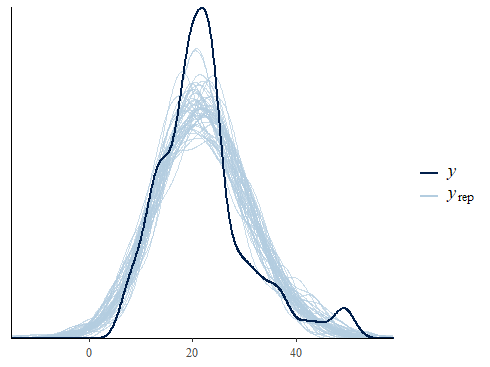
Next I look at my posterior predictive check and see some glaring issues. The posterior predictive check is important to assess the quality of our model because it will allow me to analyze if my simulated data; using my MCMC methods fits, my observed data well or not. My simulated data does not match my observed data very well. The difference between my simulated and observed data is quite drastic and indicates to me that this Bayesian Model will not be useful for my project.



Next I move onto Hierarchical Bayesian Models. A Hierarchical Bayesian Model may fit my observed data better than my simple bayesian model. The advantages of Hierarchical Models is that it allows me to analyze the particular importance of a certain variable. Given my earlier regression models I can see that lstat will be an important variable for my Hierarchical Model.

To fit a hierarchical model, we can add an intercept that varies by “Lower Status” by adding `(1 | lstat)` to the model. I change my adapt\_delta to .99 because for my MCMC I want to take smaller steps. The advantage of taking “smaller steps” will be that it may match my observed data better.

My trace plots for my Hierarchical Model is not running very smoothly. They tend to have several “divergent transitions”. A "divergent transition" indicates that the trajectory of the curve involved in determining the next proposal has diverged from what was expected (that's bad). Also my My simulated data does not match my observed data very well. The difference between my simulated and observed data is quite drastic and indicates to me that this Bayesian Model will not be useful for my project.

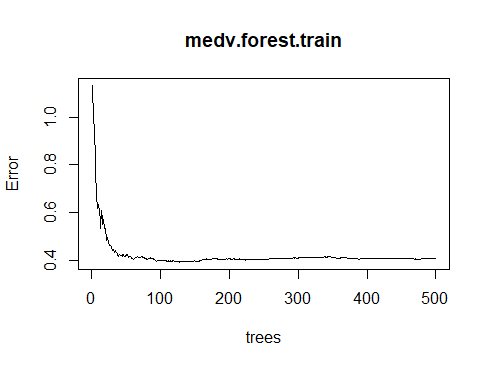
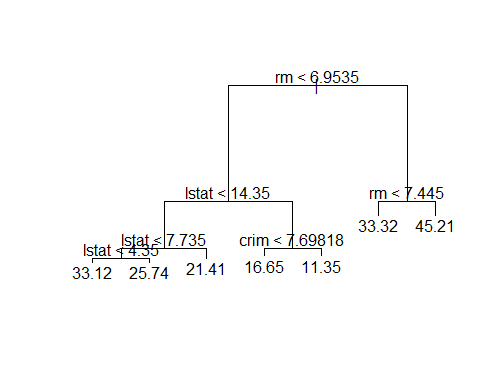


I can see that regression methods are not very good for this particular data set. In order to further increase my correlation I move onto classification methods.

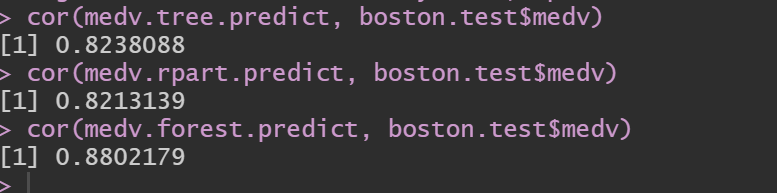
**Classification Methods: tree, rpart, randomForest**

For all of my models I bootstrap a train and test data set to help train my models and to test the accuracy of each model.

First I train my train, rpart and randomForest models using my bootstrapped data set. For my decision trees and my rpart I want to analyze the trees in more detail so I plot out the tree. Also I include a plot to see if my random forest classifier converges or not. My random forest classifier converges very well.



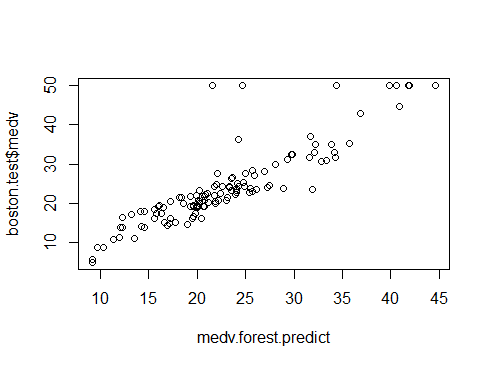
Next I investigate the correlation for all of my classification models by testing it to the outcome variable of my test data set. The outcome variable for my test data set is: medv. Medv represents the median home value of a property.



My correlation for my tree model is **.82381.** My correlation for my rpart model is **.82131.** My correlation for my model is **.88021.** My correlation for my tree and rpart models are not particularly very high so I will not pursue them for the final model for my data set.

My correlation for my random forest is actually very good. A correlation of **.88021** in my opinion is a satisfactory level of correlation for my data set. The best model for my data set would be a random forest classification model.

In the real world, real estate data seems to be very messy and many regression methods cannot overcome this problem with real estate data. A highly tuned random forest model would be the perfect model for this kind of data because it can overcome this problem with ease. I have included the plot of the quality of my random forest classifier compared to my tested outcome variable.



I am satisfied with the correlation of my basic random forest classifier. I conclude my analysis for my model at this point. If I were to pursue this further I would tune the parameters in my random forest classifier till my model fitted my tested outcome variable better. Some parameters in the random forest classifier to tune further would include: **mtry, and ntree.**