0. Project plan

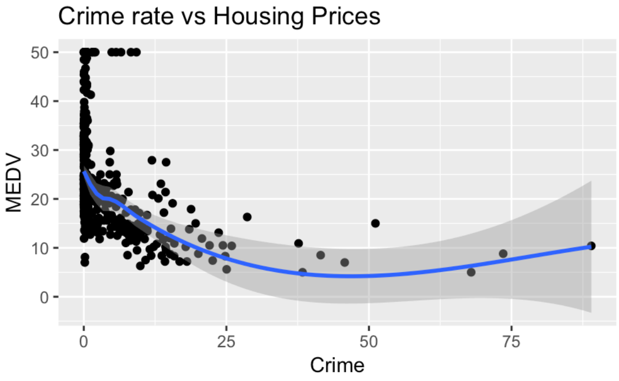
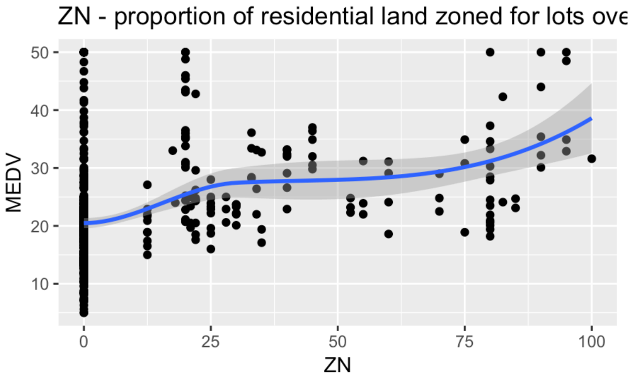
Project Plan: Building a model to predict median housing prices with factors which affect housing prices in the Boston Housing data. In order to build a model, I will divide data into two groups- first 70% for training data, the other 30% testing data

1. Selecting factors for a model

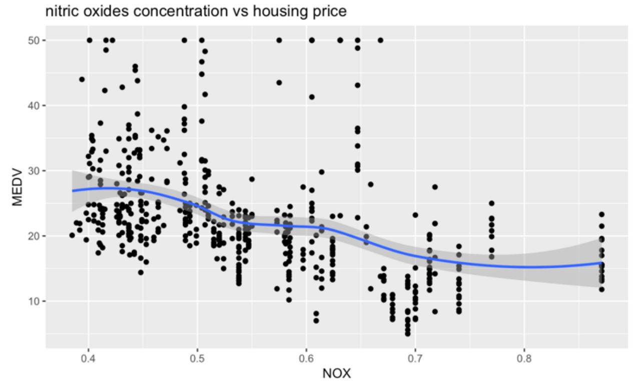
The first step of my project is to determine the meaningful dataset which would affect the housing price in the data set. There are 14 different variables such as Tax rates and the proportion of African American people. Among the 14 variables, I choose above 7 variables to predict the housing prices.

The crime rate, which means that if the rate is high, it would be dangerous, is an unarguable factor for mortgage prices. The second factor is ZN, which is proportion of residential land zones for lots over 25,000 square feet. This is also one of the important factors for mortgage prices because larger territory should have a higher value. Like ZN, fourth factor is average number of rooms. The fifth factor is the Low Status Population ratio. I think this is another important variable for predicting housing prices because if the town has a lot of low-class people, people would think it is a dangerous town and would not want to live there. Therefore, the house price would be lower than the same conditions other than the status. The sixth and seventh variables are Student-Teacher ratio and existence of the river. As I am Korean, for parents, their children's education is the most important factor for deciding where to live. Moreover, people like to see the river at their homes, so they pay more for the houses to be able to see the river. Therefore, I choose these seven variables to build my model.

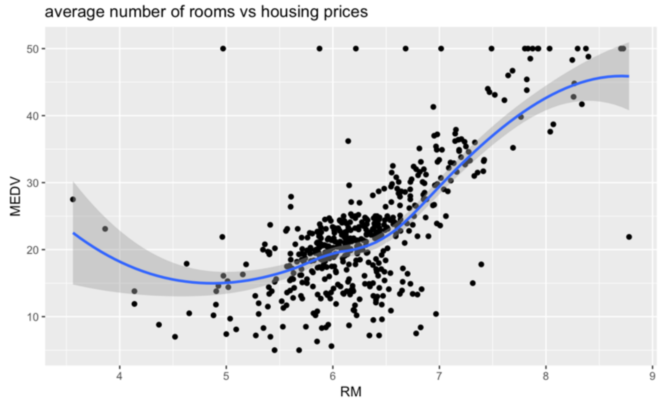
As you can see below, charts depending on each factor, the housing price value differs. However, you also can find that the MEDV=50 appears multiple times irrespectively with each variable because it is censored. So I deleted MEDV=50 from the Boston housing data set.

Crime Rate and Housing Price	Proportion of big house and Housing Price
	

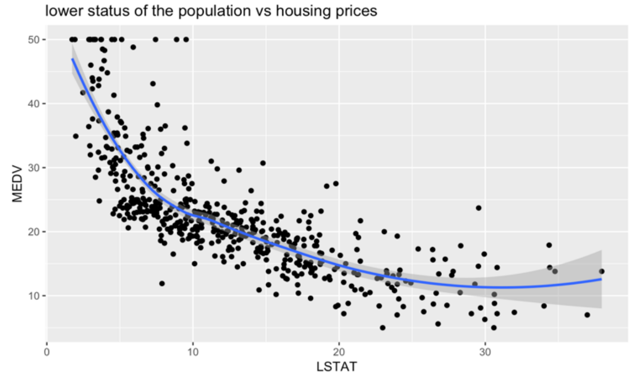
Nitric Oxides concentration and Housing price



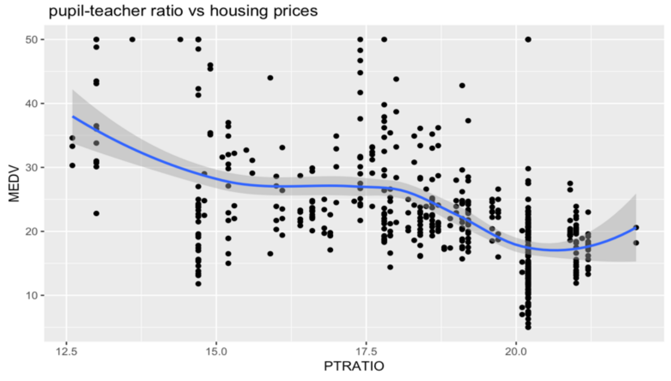
Average number of room and Housing Price



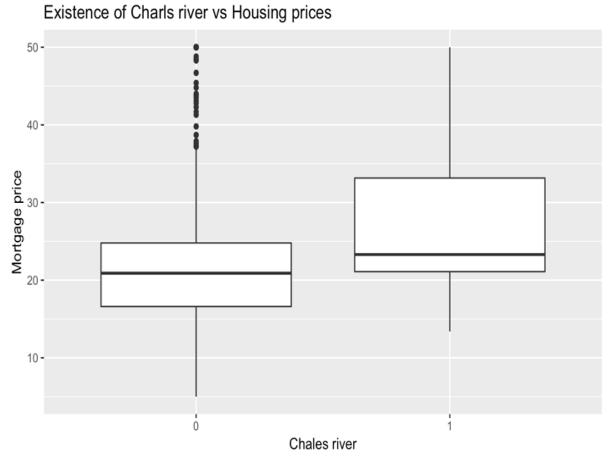
Low Status Population and Housing Price



Student-Teacher Ration and Housing Price

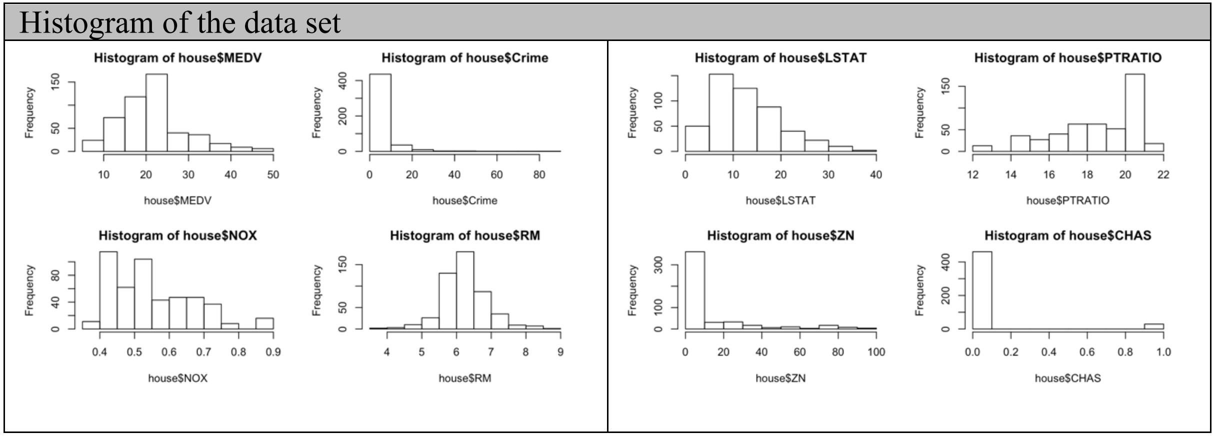


Charls river and Housing Price(1-river, 0-no)

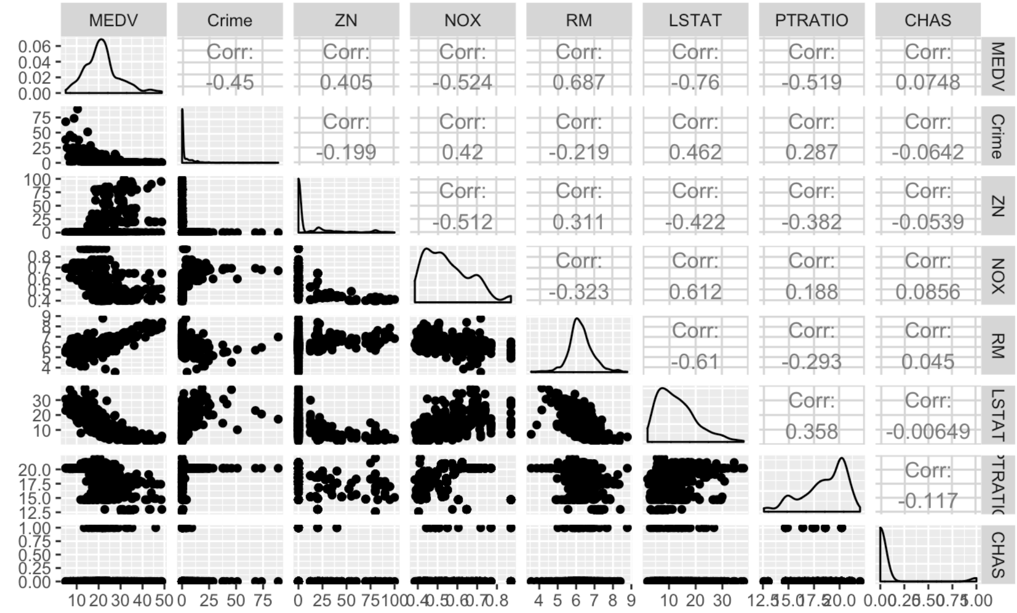


<After removing MEDV=50>

Histogram of the data set



Correlation Plot



2. Building a model

Dividing data set into two groups

```
house<-house %>% filter(MEDV !=50)
house<-house %>% select(MEDV, Crime, NOX,RM, LSTAT, PTRATIO, ZN, CHAS)
set.seed(123)
in.train<-createDataPartition(y=house$MEDV, p=0.7, list=F);
training<-house[in.train,]
testing<-house[-in.train,]
fit.lm<-lm(MEDV~., data=training)
```

With the code above, MEDV=50 is eliminated and the data set has only the variables for our model. As I said on the project plan, I divided the data into two groups (70% training, 30% testing). With this data set, I built a model to predict the MEDV.

Summary of the model

```
Call:
lm(formula = MEDV ~ ., data = training)

Residuals:
     Min       1Q   Median       3Q      Max 
-11.5115  -2.5079  -0.5826   1.9512  12.5536 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 17.22265    4.14191   4.158 4.07e-05 ***
Crime       -0.08193    0.03218  -2.546  0.01135 *  
NOX         -8.66747    2.65815  -3.261  0.00122 ** 
RM           5.25542    0.45106  11.651  < 2e-16 ***
LSTAT       -0.35524    0.05197  -6.835 3.85e-11 ***
PTRATIO     -1.00736    0.11705  -8.606 2.92e-16 ***
ZN          -0.01534    0.01168  -1.313  0.19003    
CHAS         1.23083    1.01611   1.211  0.22662    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.999 on 337 degrees of freedom
Multiple R-squared:  0.7574,	Adjusted R-squared:  0.7524 
F-statistic: 150.3 on 7 and 337 DF,  p-value: < 2.2e-16
```

3. Model’s predicting power

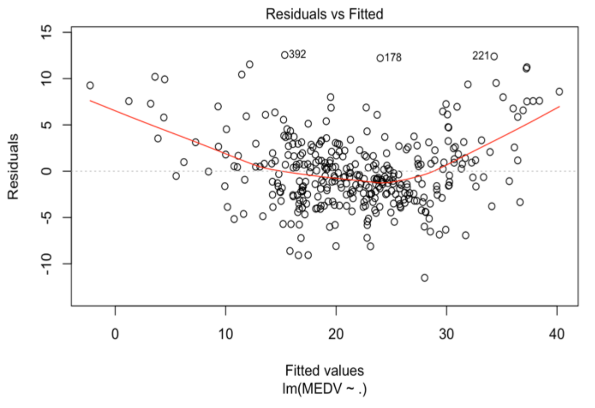
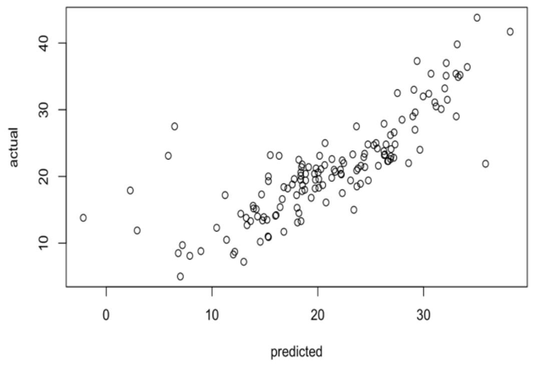
We already built a model with testing data set to predict the housing price with 70% of data set. Now, it is the time to test the model with the testing data set.

Predicting with testing data set

```
pred.lm<-predict(fit.lm, newdata=testing)
rmse.lm<-sqrt(sum((pred.lm-testing$MEDV)^2/length(testing$MEDV)))
c(RMSE = rmse.lm, R2 = summary(fit.lm)$r.squared)
plot(pred.lm,testing$MEDV, xlab="predicted", ylab="actual")

RMSE        R2
4.4808533 0.7574309
```

With the model, we get the root mean squared error 4.48 and coefficient of determination value 0.7574. And below is the plot between predicted value and actual value of house price.

Residual Plot	Actual versus Predicted Price
	

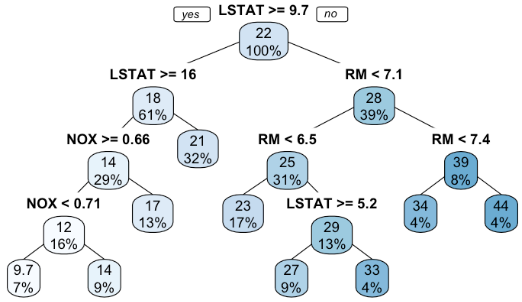
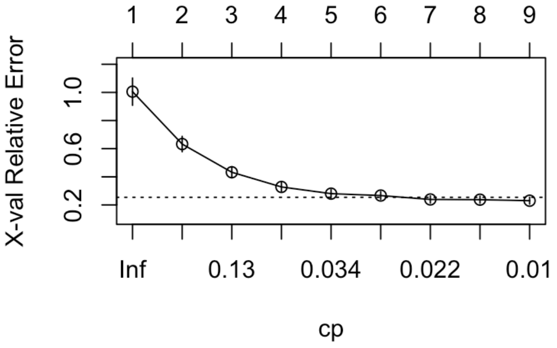
The right side of plot is the plot between actual housing price versus the predicted housing price predicted by our model. It seems to fit pretty well, but there are some points deviated from our model. And the same thing is shown on the first plot. We can see the bottom left side and up-right side have outliers from the right-side plot. And the residual plot also shows that lower housing prices and higher housing prices are slightly deviated from 0. It means that either cheaper or more expensive housing prices, the model is not well fitting.

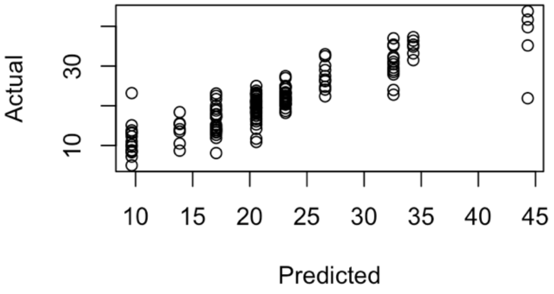
4. Tree model

Code for Tree Model

```
set.seed(123456)
boston.tree<-rpart(MEDV~., data=training, cp=0.01)
rpart.plot(boston.tree, type=1, fallen.leaves=F)
plotcp(boston.tree)
printcp(boston.tree)
cp<- data.frame(boston.tree$cptable)
(best<-cp[which(cp$xerror == min(cp$xerror)),])
prune.tree<-prune(boston.tree, cp=best$cp)
tree_preds<-predict(prune.tree, newdata=testing)
plot(tree_preds,testing$MEDV, xlab="Predicted", ylab="Actual")
```

Our regression model has a pretty good 0.7574 coefficient of determination, but there are some other ways to increase our predicting power. One of the methods is the Tree model. Using the codes learnt from the machine learning class, we can build another model to predict the housing prices.

Tree Model	
 yes LSTAT >= 9.7 no 22 100% LSTAT >= 16 RM < 7.1 18 61% 28 39% NOX >= 0.66 21 32% RM < 6.5 RM < 7.4 14 29% 25 31% 39 8% NOX < 0.71 17 13% 23 17% LSTAT >= 5.2 34 4% 44 4% 12 16% 29 13% 9.7 7% 14 9% 27 9% 33 4%	

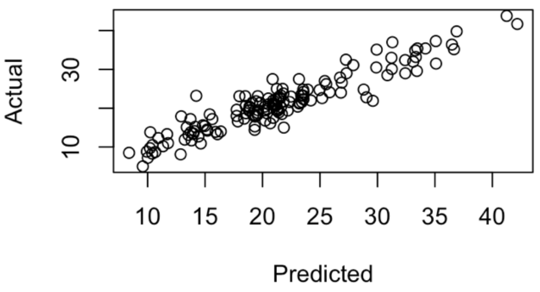
Actual price versus Predicted price with Tree	Summary of the model
 	Call: randomForest(formula = pre, data = training, mtry = 13, importance = T) Type of random forest: regression Number of trees: 500 No. of variables tried at each split: 7 Mean of squared residuals: 10.04452 % Var explained: 84.4 RMSE 3.943266

The Tree model concept is simple. It divides data into each category appeared on the tree model, it increases the predicting power and reduces errors. Therefore, the R^2 is 0.844 and the error reduces as well.

5. Random Forest Model

Random Forest Code

frf<-randomForest(MEDV~., data= training, mtry=2, ntree=200, keep.forest=T, importance=T)
rf_preds<-predict(frf, type="response", newdata=testing)
varImpPlot(frf, type=2)
frf
rmse.lm2<-sqrt(sum((rf_preds-testing$MEDV)^2/length(testing$MEDV)));rmse.lm2
plot(rf_preds, testing$MEDV, xlab="Predicted", ylab="Actual")

Actual price versus Predicted price with RF	Summary
	Call: randomForest(formula = MEDV ~ ., data = training, mtry = 2, ntree = 200, keep.forest = T, importance = T) Type of random forest: regression Number of trees: 200 No. of variables tried at each split: 2 Mean of squared residuals: 9.131613 % Var explained: 85.82 RMSE 2.590781

The last predicting model is the Random Forest model. By using the Random Forest model, we can slightly further reduce the error and increase the R^2.

6. Conclusion and Limitation

The projects purpose is predicting the housing prices with variables we chose. We built three different models: Linear Regression model, Tree model, and Random Forest model. Each model explained pretty much the variation of housing prices with low errors. Therefore, we found that the variables and housing prices are related each other. If we had more details provided by the data set, such as whether the area is used for commercial or industrial regions, we would have better fitting models.

However, the project has some limitations. First limitation is our models do not predict better than the multivariate linear model using all variables in data set. Even though our models have much lower errors and it is true that having more explanatory variables increases R-Square value, I feel this is one of limitations because our models seem just fancier but not better predicting models. Second limitation is choosing the variables I used for building a model is subjective. I decided the variables would be related to the housing prices but I did not show whether it is really related or whether there is another variable better than those variables. Third limitation is there is no abline on the plot between predicted value and actual value. I thought it is a simple line with slope=1 so I put a=0, b=1 on R, but it did not work. Fourth limitation is there is no PCA analysis. The project would become better if there was a PCA analysis so covering all material learned from the class.