Boston Housing Project

Shrey Agarwal

18/12/2021

Question 1 - Exploring the data

In this Question we are going to Explore the given boston Housing dataset using exploratory data analysis and perform a statistical summary to find any patterns between the variables.

```
# importing the dataset and providing a summary
require(devtools)
```

```
## Loading required package: devtools
```

```
## Loading required package: usethis
```

```
library(ggplot2)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
install_github("ropensci/plotly")
```

```
## Downloading GitHub repo ropensci/plotly@HEAD
```

```
## rlang        (0.4.11 -> 1.0.3  ) [CRAN]
## Rcpp         (1.0.7  -> 1.0.8.3) [CRAN]
## glue         (1.4.2  -> 1.6.2  ) [CRAN]
## vctrs        (0.3.8  -> 0.4.1  ) [CRAN]
## fansi        (0.5.0  -> 1.0.3  ) [CRAN]
## crayon       (1.4.2  -> 1.5.1  ) [CRAN]
## magrittr     (2.0.1  -> 2.0.3  ) [CRAN]
## pillar       (1.6.4  -> 1.7.0  ) [CRAN]
```

```
## tidyselect   (1.1.1  -> 1.1.2  ) [CRAN]
## tibble       (3.1.6  -> 3.1.7  ) [CRAN]
## generics     (0.1.1  -> 0.1.2  ) [CRAN]
## colorspace   (2.0-2  -> 2.0-3  ) [CRAN]
## RColorBrewer (1.1-2  -> 1.1-3  ) [CRAN]
## jsonlite     (1.7.2  -> 1.8.0  ) [CRAN]
## dplyr        (1.0.7  -> 1.0.9  ) [CRAN]
## yaml         (2.2.1  -> 2.3.5  ) [CRAN]
## withr        (2.4.3  -> 2.5.0  ) [CRAN]
## scales       (1.1.1  -> 1.2.0  ) [CRAN]
## httr         (1.4.2  -> 1.4.3  ) [CRAN]
## ggplot2      (3.3.5  -> 3.3.6  ) [CRAN]

## Installing 20 packages: rlang, Rcpp, glue, vctrs, fansi, crayon, magrittr, pillar, tidyselect, tibble

## Warning: packages 'dplyr', 'ggplot2' are in use and will not be installed

## Installing packages into 'C:/Users/SHREY AGARWAL/OneDrive/Documents/R/win-library/4.0'
## (as 'lib' is unspecified)

##
##   There are binary versions available but the source versions are later:
##        binary source needs_compilation
## rlang   1.0.2  1.0.3              TRUE
## tibble  3.1.6  3.1.7              TRUE
## httr    1.4.2  1.4.3             FALSE
##
## package 'Rcpp' successfully unpacked and MD5 sums checked
## package 'glue' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'glue'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem
## copying C:\Users\SHREY AGARWAL\OneDrive\Documents\R\win-
## library\4.0\00LOCK\glue\libs\x64\glue.dll to C:\Users\SHREY
## AGARWAL\OneDrive\Documents\R\win-library\4.0\glue\libs\x64\glue.dll: Permission
## denied

## Warning: restored 'glue'

## package 'vctrs' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'vctrs'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem
## copying C:\Users\SHREY AGARWAL\OneDrive\Documents\R\win-
## library\4.0\00LOCK\vctrs\libs\x64\vctrs.dll to C:\Users\SHREY
## AGARWAL\OneDrive\Documents\R\win-library\4.0\vctrs\libs\x64\vctrs.dll:
## Permission denied

## Warning: restored 'vctrs'
```

```
## package 'fansi' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'fansi'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem
## copying C:\Users\SHREY AGARWAL\OneDrive\Documents\R\win-
## library\4.0\00LOCK\fansi\libs\x64\fansi.dll to C:\Users\SHREY
## AGARWAL\OneDrive\Documents\R\win-library\4.0\fansi\libs\x64\fansi.dll:
## Permission denied

## Warning: restored 'fansi'

## package 'crayon' successfully unpacked and MD5 sums checked
## package 'magrittr' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'magrittr'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem
## copying C:\Users\SHREY AGARWAL\OneDrive\Documents\R\win-
## library\4.0\00LOCK\magrittr\libs\x64\magrittr.dll to C:\Users\SHREY
## AGARWAL\OneDrive\Documents\R\win-library\4.0\magrittr\libs\x64\magrittr.dll:
## Permission denied

## Warning: restored 'magrittr'

## package 'pillar' successfully unpacked and MD5 sums checked
## package 'tidyselect' successfully unpacked and MD5 sums checked
## package 'generics' successfully unpacked and MD5 sums checked
## package 'colorspace' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'colorspace'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem
## copying C:\Users\SHREY AGARWAL\OneDrive\Documents\R\win-
## library\4.0\00LOCK\colorspace\libs\x64\colorspace.dll to C:\Users\SHREY
## AGARWAL\OneDrive\Documents\R\win-library\4.0\colorspace\libs\x64\colorspace.dll:
## Permission denied

## Warning: restored 'colorspace'

## package 'RColorBrewer' successfully unpacked and MD5 sums checked
## package 'jsonlite' successfully unpacked and MD5 sums checked
## package 'yaml' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'yaml'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem
## copying C:\Users\SHREY AGARWAL\OneDrive\Documents\R\win-
## library\4.0\00LOCK\yaml\libs\x64\yaml.dll to C:\Users\SHREY
## AGARWAL\OneDrive\Documents\R\win-library\4.0\yaml\libs\x64\yaml.dll: Permission
## denied
```

```
## Warning: restored 'yaml'

## package 'withr' successfully unpacked and MD5 sums checked
## package 'scales' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\SHREY AGARWAL\AppData\Local\Temp\RtmpwzyKfk\downloaded_packages

## installing the source packages 'rlang', 'tibble', 'httr'

## Warning in i.p(...): installation of package 'rlang' had non-zero exit status

## Warning in i.p(...): installation of package 'tibble' had non-zero exit status

##          checking for file 'C:\Users\SHREY AGARWAL\AppData\Local\Temp\RtmpwzyKfk\remotes3334fd81dbb\p
##       -  preparing 'plotly': (5.9s)
##    checking DESCRIPTION meta-information ...     checking DESCRIPTION meta-information ...   v  check
##       -  checking for LF line-endings in source and make files and shell scripts
##       -  checking for empty or unneeded directories
##       -  building 'plotly_4.10.0.9001.tar.gz'
##
##

## Installing package into 'C:/Users/SHREY AGARWAL/OneDrive/Documents/R/win-library/4.0'
## (as 'lib' is unspecified)

## Warning in i.p(...): installation of package 'C:/Users/SHREYA~1/AppData/Local/
## Temp/RtmpwzyKfk/file3334792d2c38/plotly_4.10.0.9001.tar.gz' had non-zero exit
## status
```

```
housingdata<- read.csv("housing.csv")
Housing<- as.data.frame(housingdata)
summary(Housing)
```

```
##       CRIM               CRO1            ZN              INDUS
##  Min.   : 0.00632   Min.   :0.0   Min.   :  0.00   Min.   : 0.46
##  1st Qu.: 0.08205   1st Qu.:0.0   1st Qu.:  0.00   1st Qu.: 5.19
##  Median : 0.25651   Median :0.5   Median :  0.00   Median : 9.69
##  Mean   : 3.61352   Mean   :0.5   Mean   : 11.36   Mean   :11.14
##  3rd Qu.: 3.67708   3rd Qu.:1.0   3rd Qu.: 12.50   3rd Qu.:18.10
##  Max.   :88.97620   Max.   :1.0   Max.   :100.00   Max.   :27.74
##       CHAS              NOX               RM             AGE
##  Min.   :0.00000   Min.   :0.3850   Min.   :3.561   Min.   :  2.90
##  1st Qu.:0.00000   1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02
##  Median :0.00000   Median :0.5380   Median :6.208   Median : 77.50
##  Mean   :0.06917   Mean   :0.5547   Mean   :6.285   Mean   : 68.57
##  3rd Qu.:0.00000   3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08
##  Max.   :1.00000   Max.   :0.8710   Max.   :8.780   Max.   :100.00
##       DIS              RAD              TAX           PTRATIO
##  Min.   : 1.130   Min.   : 1.000   Min.   :187.0   Min.   :12.60
##  1st Qu.: 2.100   1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40
```

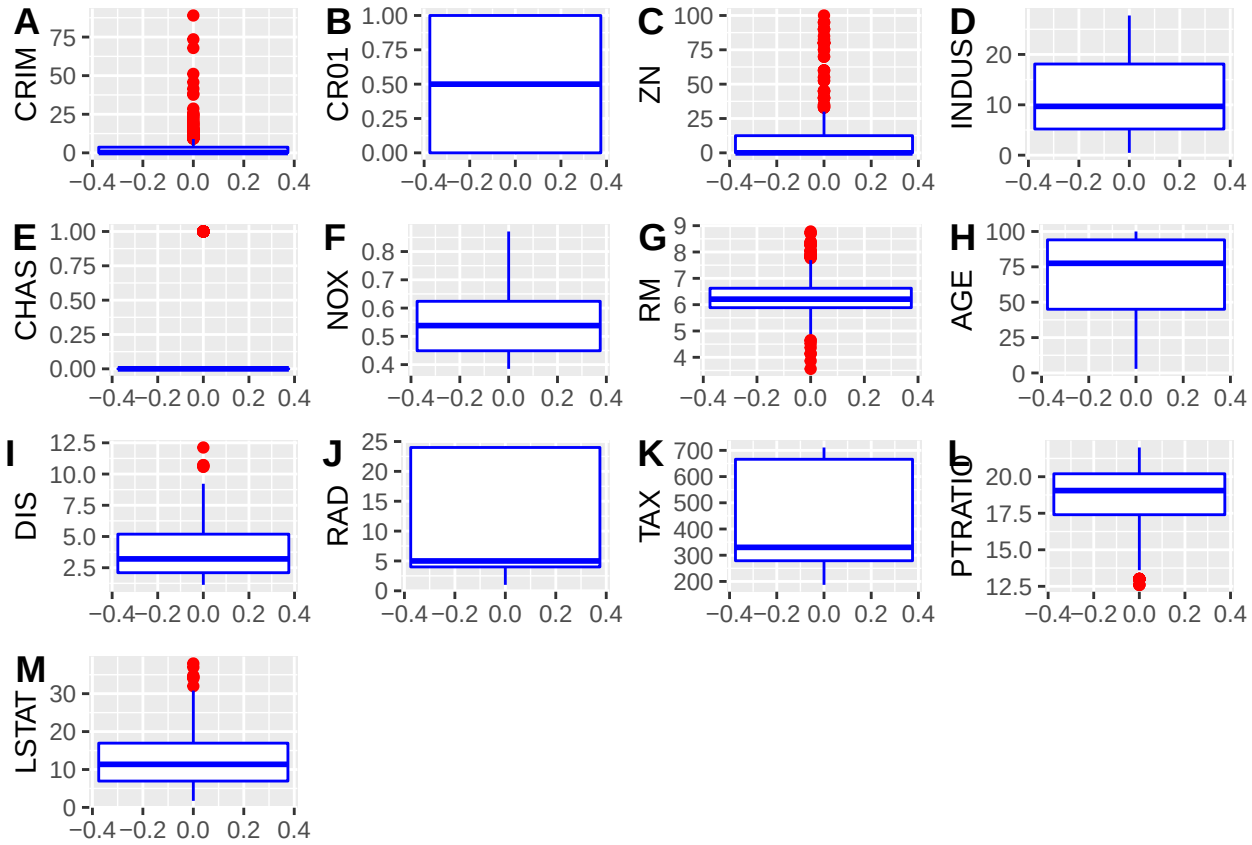
```
##  Median : 3.207   Median : 5.000   Median :330.0   Median :19.05
##  Mean   : 3.795   Mean   : 9.549   Mean   :408.2   Mean   :18.46
##  3rd Qu.: 5.188   3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20
##  Max.   :12.127   Max.   :24.000   Max.   :711.0   Max.   :22.00
##      LSTAT            MEDV
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

```
str(Housing)
```

```
## 'data.frame':    506 obs. of  14 variables:
##  $ CRIM   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ CR01   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ ZN     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ INDUS  : num  2.31 7.07 7.07 2.18 2.18 ...
##  $ CHAS   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ NOX    : num  0.538 0.469 0.469 0.458 0.458 ...
##  $ RM     : num  6.57 6.42 7.18 7 7.15 ...
##  $ AGE    : num  65.2 78.9 61.1 45.8 54.2 ...
##  $ DIS    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ RAD    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ TAX    : int  296 242 242 222 222 222 311 311 311 311 ...
##  $ PTRATIO: num  15.3 17.8 17.8 18.7 18.7 ...
##  $ LSTAT  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ MEDV   : num  24 21.6 34.7 33.4 36.2 ...
```

We observe that most of the data in ZN, CHAS is 0. That's why they may not be good predicting variables for other data. We also see that all the data is numerical which is beneficial for classification. Now we will see their boxplots to find how many outliers are there in the data.

```
# boxplot
library(cowplot)
A= ggplot(data = Housing, mapping = aes(y =CRIM))+ geom_boxplot(color="blue",outlier.color="red")
B= ggplot(data = Housing, mapping = aes(y =CR01))+ geom_boxplot(color="blue",outlier.color="red")
C=ggplot(data = Housing, mapping = aes(y =ZN))+ geom_boxplot(color="blue",outlier.color="red")
D=ggplot(data = Housing, mapping = aes(y =INDUS))+ geom_boxplot(color="blue",outlier.color="red")
E=ggplot(data = Housing, mapping = aes(y =CHAS))+ geom_boxplot(color="blue",outlier.color="red")
J=ggplot(data = Housing, mapping = aes(y =NOX))+ geom_boxplot(color="blue",outlier.color="red")
G=ggplot(data = Housing, mapping = aes(y =RM))+ geom_boxplot(color="blue",outlier.color="red")
H=ggplot(data = Housing, mapping = aes(y =AGE))+ geom_boxplot(color="blue",outlier.color="red")
I=ggplot(data = Housing, mapping = aes(y =DIS))+ geom_boxplot(color="blue",outlier.color="red")
K=ggplot(data = Housing, mapping = aes(y =RAD))+ geom_boxplot(color="blue",outlier.color="red")
L=ggplot(data = Housing, mapping = aes(y =TAX))+ geom_boxplot(color="blue",outlier.color="red")
M=ggplot(data = Housing, mapping = aes(y =PTRATIO))+ geom_boxplot(color="blue",outlier.color="red")
N=ggplot(data = Housing, mapping = aes(y =LSTAT))+ geom_boxplot(color="blue",outlier.color="red")
O=ggplot(data = Housing, mapping = aes(y =MEDV))+ geom_boxplot(color="blue",outlier.color="red")
plot_grid(A,B,C,D,E,J,G,H,I,K,L,M,N, labels = "AUTO")
```

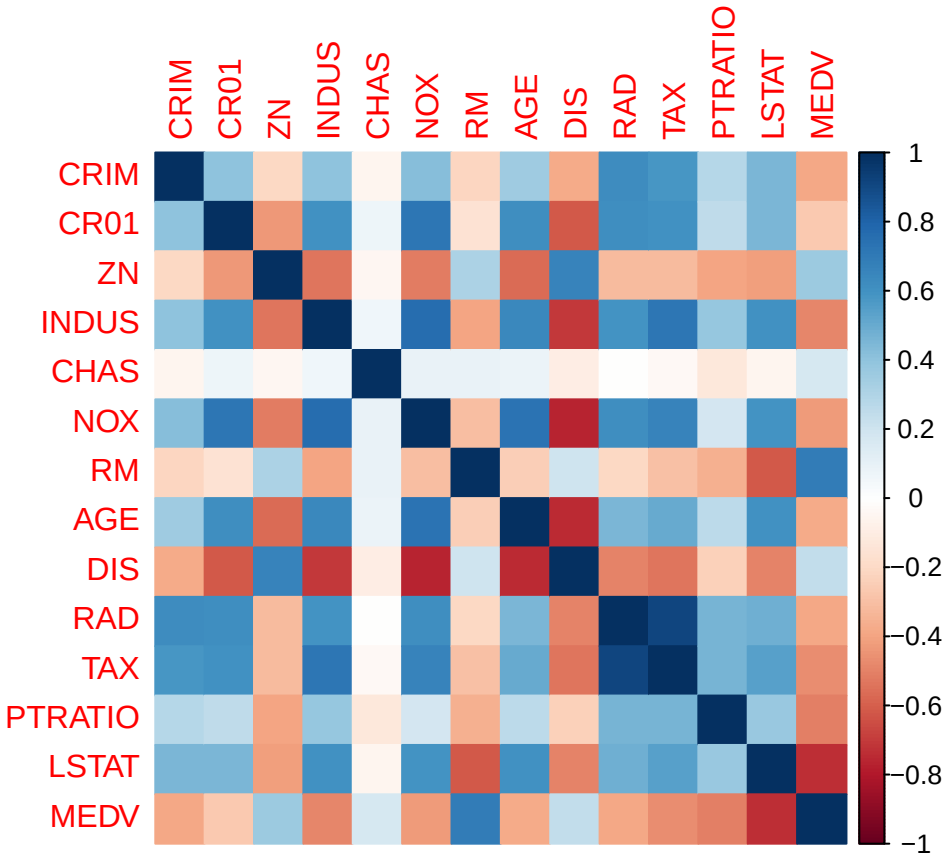


We will now look look for correlation in our dataset using corrplot. This will help us in predicting our data for regression and we might be able to find some insights too.

```
# Corrplot
library(corrplot)
```

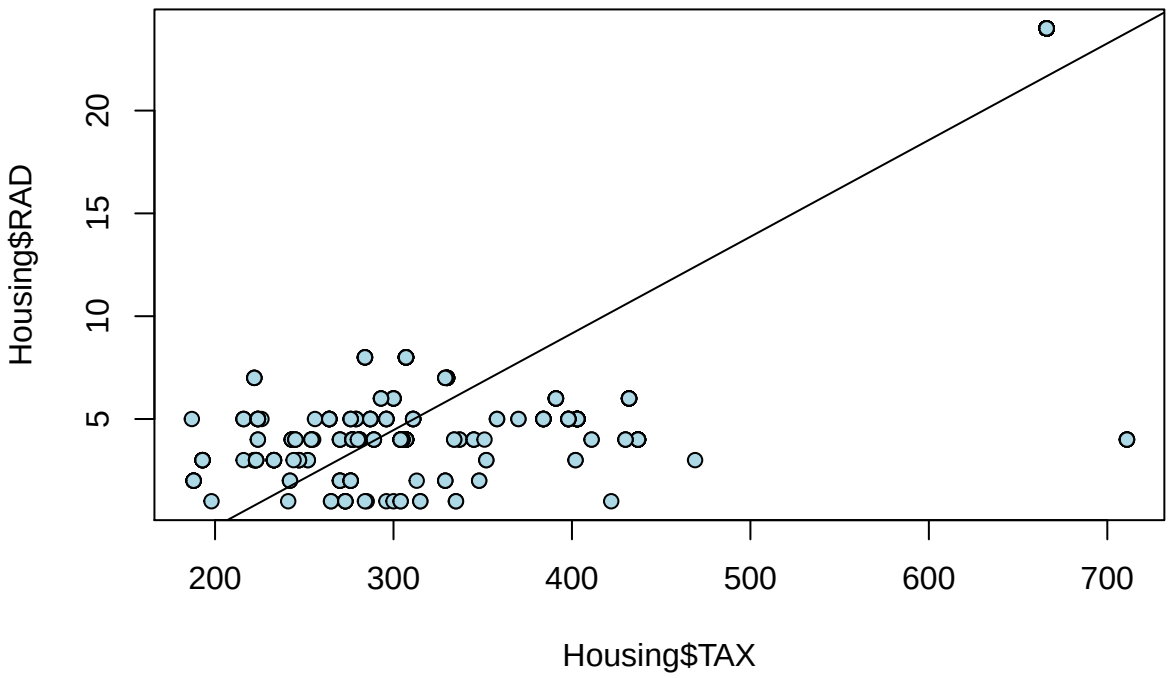
```
## corrplot 0.92 loaded
```

```
Correlation=cor(Housing)
corrplot(Correlation, method ="color")
```



We observe that there is 0.91 correlation between RAD and TAX. Let's plot it to see how it is and try to see if this is related to crime rate as well

```
# plotting the RAD vs TAx
x<- plot(Housing$RAD ~ Housing$TAX, pch = 21, bg = "lightblue", col = "black")
y<- lm(Housing$RAD~Housing$TAX)
abline(y)
```



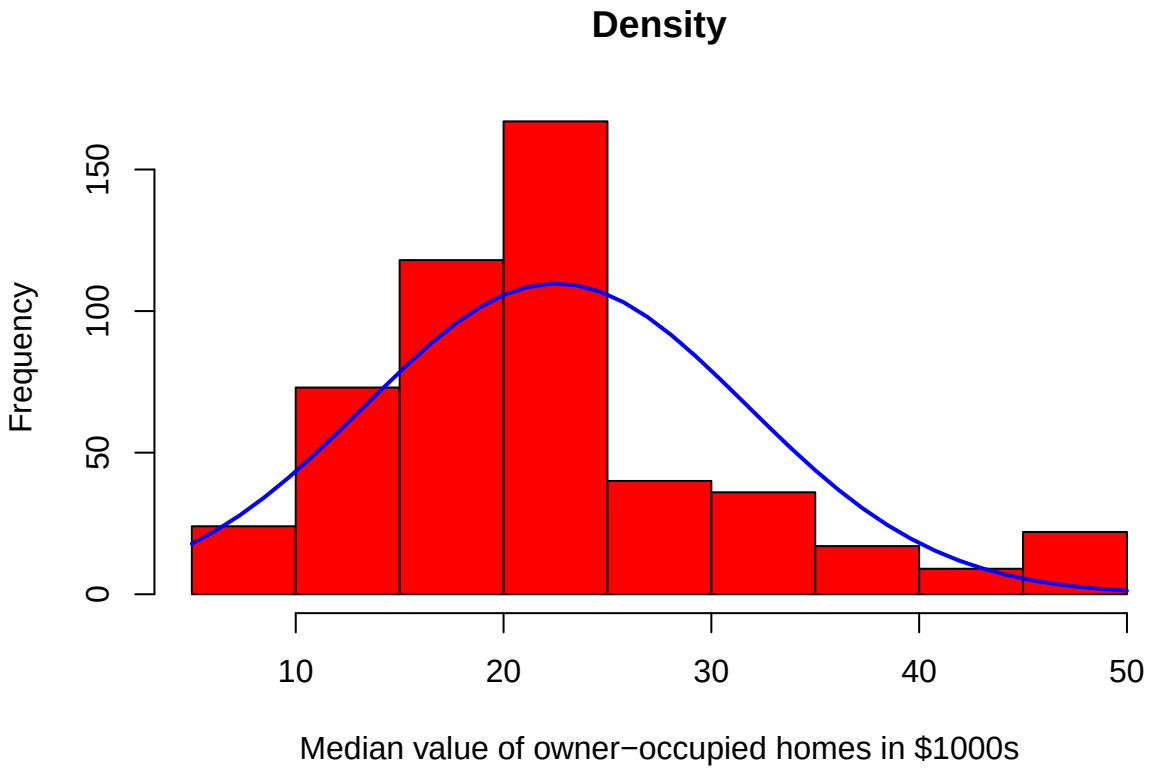

```
#require(rCharts)
#n1 = nPlot(RAD ~ TAX, group = 'CR01', type = 'multiBarChart', data = Housing)
#n1
```

We can't deduce much from the graph but we did see that the in some cases places with high Index of accessibility to radial highways has higher crime rate.

Question 2 - Develop a Regression Model for MEDV

First we will take a look at MEDV

```
# Density plot for MEDV variable
x<- Housing$MEDV
h<-hist(Housing$MEDV, breaks=10, col="red", xlab="Median value of owner-occupied homes in $1000s",main=
xfit<-seq(min(x),max(x),length=40)
yfit<-dnorm(xfit,mean=mean(x),sd=sd(x))
yfit <- yfit*diff(h$mids[1:2])*length(x)
lines(xfit, yfit, col="blue", lwd=2)
```



We see that the data is normally distributed, therefore we won't have any problem in predicting the variable. We will take a look at the the corrplot again to find significant correlation with MEDV variable. Since a model with high number of predictors is not significant, we will try to eliminate those from the start

From the corplot,we observe many variables that are highly negatively correlated with MEDV which include LSTAT, PTRATIO, INDUS and TAX and is positively correlated with RM. We will use these variables for the regression models.

We will create both linear and non linear models and check which has better goodness to fit and performance. we will use PRESS (sum of squared cross-validated residuals)and RMSE(Root Mean Square Error) to compare the models. ### Model 1

```
# running a linear regression model on MEDV variable
n = nrow(Housing)
cv_res1 = vector(length=n)
for(i in 1:n){
    fiti = lm(MEDV ~ LSTAT+PTRATIO+TAX+RM+INDUS  , data=Housing[-i,])
    predi = predict(fiti, newdata=Housing[i,])
    cv_res1[i] = Housing$MEDV[i] - predi
}
# Finding PRESS, RMSE and R2 value for the model
PRESS1 = sum(cv_res1^2)
RMSE1 = sqrt(PRESS1/n)
summary(fiti)
```

```
##
## Call:
```

```
## lm(formula = MEDV ~ LSTAT + PTRATIO + TAX + RM + INDUS, data = Housing[-i,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.2512  -3.0630  -0.9127   1.7526  30.2534
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 17.431334   3.963133   4.398 1.33e-05 ***
## LSTAT       -0.563667   0.048753 -11.562  < 2e-16 ***
## PTRATIO     -0.857617   0.125334  -6.843 2.29e-11 ***
## TAX         -0.003814   0.002126  -1.794   0.0734 .
## RM           4.606001   0.429448  10.725  < 2e-16 ***
## INDUS        0.062151   0.052601   1.182   0.2379
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.209 on 499 degrees of freedom
## Multiple R-squared:  0.6821, Adjusted R-squared:  0.679
## F-statistic: 214.2 on 5 and 499 DF,  p-value: < 2.2e-16
```

For the next model we will just keep the negative correlations as predictors ### Model 2

```
# running a linear regression model on MEDV variable
cv_res2 = vector(length=n)
for(i in 1:n){
    fiti2 = lm(MEDV ~ LSTAT+PTRATIO+TAX+INDUS  , data=Housing[-i,])
    predi2 = predict(fiti2, newdata=Housing[i,])
    cv_res2[i] = Housing$MEDV[i] - predi2
}
# PRESS is sum of squared cross-validated residuals
# Finding PRESS, RMSE and R2 value for the model
PRESS2 = sum(cv_res2^2)
RMSE2 = sqrt(PRESS2/n)
```

For the next model we will remove TAX and INDUS since their P value is quite high

Model 3

```
# running a linear regression model on MEDV variable
cv_res3 = vector(length=n)
for(i in 1:n){
    fiti3 = lm(MEDV ~ LSTAT+PTRATIO , data=Housing[-i,])
    predi3 = predict(fiti3, newdata=Housing[i,])
    cv_res3[i] = Housing$MEDV[i] - predi3
}
# Finding PRESS, RMSE and R2 value for the model
PRESS3 = sum(cv_res3^2)
RMSE3 = sqrt(PRESS3/n)
summary(fiti3)
```

```
##
## Call:
## lm(formula = MEDV ~ LSTAT + PTRATIO, data = Housing[-i, ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.2373  -3.6480  -0.8712   1.8511  26.8363
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 53.79506    2.23872  24.029   <2e-16 ***
## LSTAT       -0.82437    0.03877 -21.263   <2e-16 ***
## PTRATIO     -1.12748    0.12800  -8.808   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.762 on 502 degrees of freedom
## Multiple R-squared:  0.6088, Adjusted R-squared:  0.6073
## F-statistic: 390.6 on 2 and 502 DF,  p-value: < 2.2e-16
```

We will now try to iterate the 1st model, just without the INDUS and TAX. ### Model 4

```
# running a linear regression model on MEDV variable
cv_res4 = vector(length=n)
for(i in 1:n){
    fiti4 = lm(MEDV ~ RM+LSTAT+PTRATIO , data=Housing[-i,])
    predi4 = predict(fiti4, newdata=Housing[i,])
    cv_res4[i] = Housing$MEDV[i] - predi4
}
# Finding PRESS, RMSE and R2 value for the model
PRESS4 = sum(cv_res4^2)
RMSE4 = sqrt(PRESS4/n)
summary(fiti4)
```

```
##
## Call:
## lm(formula = MEDV ~ RM + LSTAT + PTRATIO, data = Housing[-i,
##     ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.5023  -3.1411  -0.8484   1.7782  29.5311
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  18.5589     3.9031   4.755 2.60e-06 ***
## RM            4.4893     0.4250  10.563  < 2e-16 ***
## LSTAT        -0.5768     0.0422 -13.667  < 2e-16 ***
## PTRATIO      -0.9169     0.1176  -7.798 3.66e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.216 on 501 degrees of freedom
```

```
## Multiple R-squared:  0.6801, Adjusted R-squared:  0.6781
## F-statistic:   355 on 3 and 501 DF,  p-value: < 2.2e-16
```

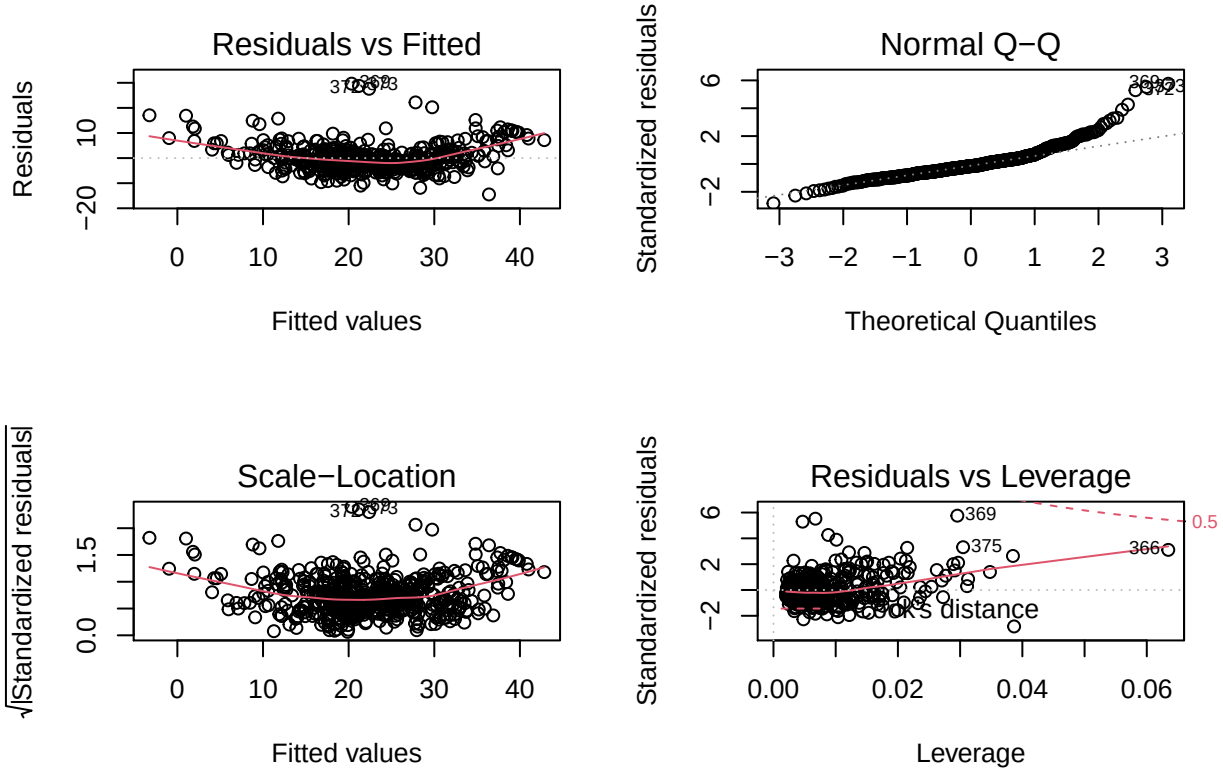
```
dv<- data.frame(PRESS=c(PRESS1,PRESS2,PRESS3,PRESS4),RMSE=c(RMSE1,RMSE2,RMSE3,RMSE4))
dv
```

```
##      PRESS     RMSE
## 1 14231.36 5.303321
## 2 17240.83 5.837190
## 3 17064.01 5.807181
## 4 14117.50 5.282065
```

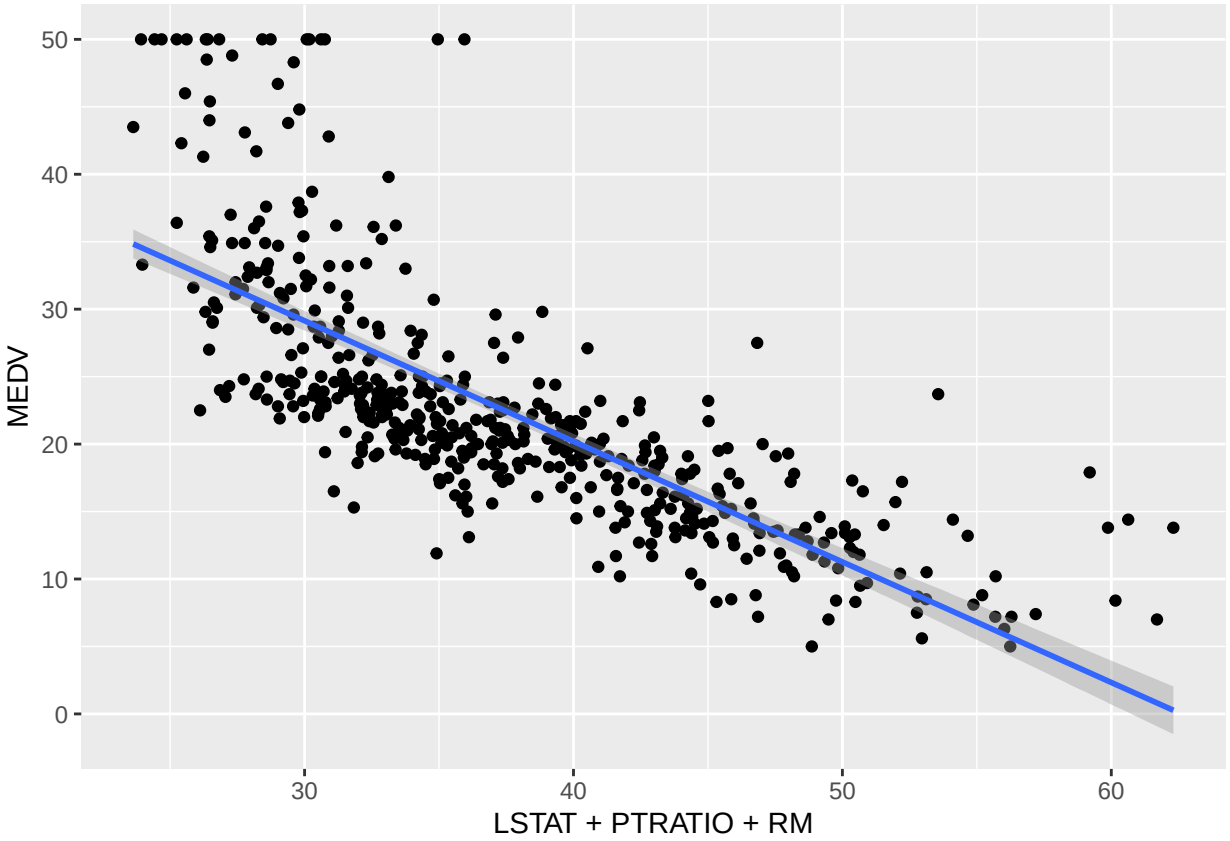
```
par(mfrow = c(2, 2))
fit2 <- lm(MEDV ~ LSTAT+PTRATIO+RM , data = Housing)
summary(fit2)
```

```
##
## Call:
## lm(formula = MEDV ~ LSTAT + PTRATIO + RM, data = Housing)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14.4871  -3.1047  -0.7976   1.8129  29.6559
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 18.56711    3.91320   4.745 2.73e-06 ***
## LSTAT       -0.57181    0.04223 -13.540  < 2e-16 ***
## PTRATIO     -0.93072    0.11765  -7.911 1.64e-14 ***
## RM           4.51542    0.42587  10.603  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.229 on 502 degrees of freedom
## Multiple R-squared:  0.6786, Adjusted R-squared:  0.6767
## F-statistic: 353.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
# Plotting the linear equation on MEDV
plot(fit2)
```




```
ggplot(Housing, aes(LSTAT+PTRATIO+RM, MEDV) ) +
    geom_point() +
    stat_smooth(method = lm, formula = y ~ x)
```



For a good regression model, we need Goodness to fit and high performance for which we tried 4 linear models and found the 4th model to be the best fit since it has the least value for PRESS and RMSE, which is ideal for the model. It also has the highest Multiple R-squared: 0.6088, Adjusted R-squared: 0.6073 which we need for the model.

From the plotted graphs above, we can see that a polynomial regression model could be better for finding a better model, since in Residual vs Fitted plot - Some values are above the median line Normal Q-Q - the end values divert from the linear nine. The good thing is the values is scale- location is quite diversified in the plot. So, now we will look at non-linear polynomial models. In the first model we will take previous best model for polynomial. ### Model 5

```
# running a linear regression model on MEDV variable
cv_res5 = vector(length=n)
for(i in 1:n){
    fiti5 = lm(MEDV ~ poly(LSTAT+PTRATIO+RM, 5, raw = TRUE), data=Housing[-i,])
    predi5 = predict(fiti5, newdata=Housing[i,])
    cv_res5[i] = Housing$MEDV[i] - predi5
}
# Finding PRESS, RMSE and R2 value for the model
PRESS5 = sum(cv_res5^2)
RMSE5 = sqrt(PRESS5/n)
```

The model improved in terms of R2 value but not in terms of PRESS, therefore we will try without the positively correlated value and check. ### Model 6

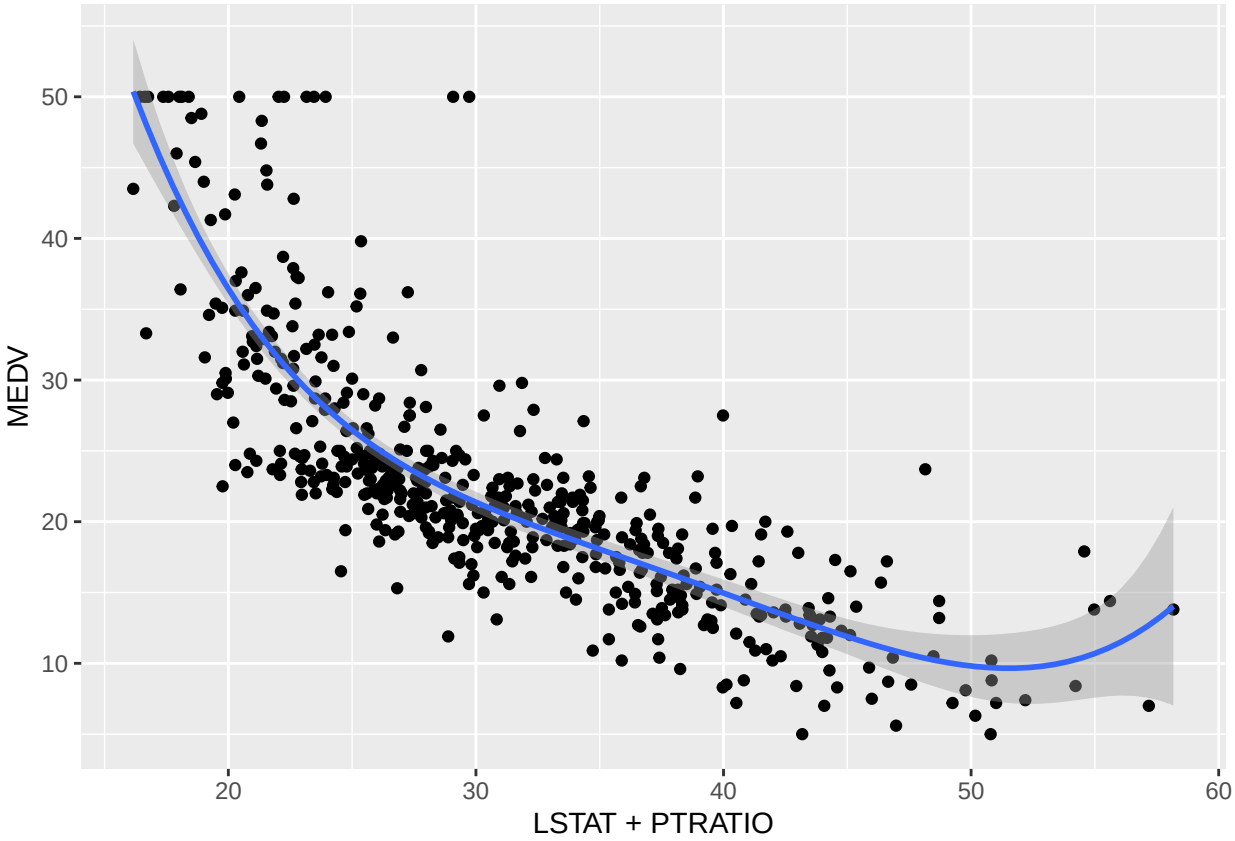
```
# running a non-linear regression model on MEDV variable
cv_res6 = vector(length=n)
for(i in 1:n){
    fiti6 = lm(MEDV ~ poly(LSTAT+PTRATIO, 5, raw = TRUE), data=Housing[-i,])
    predi6 = predict(fiti6, newdata=Housing[i,])
    cv_res6[i] = Housing$MEDV[i] - predi6
}
# Finding PRESS, RMSE and R2 value for the model
PRESS6 = sum(cv_res6^2)
RMSE6 = sqrt(PRESS6/n)
```

The model improved quite Significantly in terms of everything PRESS6 = 13184, RMSE = 5.1 and R2 = 0.7. Now we will try with just LSTAT to see if it improves. ### Model 7

```
# running a non-linear regression model on MEDV variable
cv_res7 = vector(length=n)
for(i in 1:n){
    fiti7 = lm(MEDV ~ poly(LSTAT, 5, raw = TRUE), data=Housing[-i,])
    predi7 = predict(fiti7, newdata=Housing[i,])
    cv_res7[i] = Housing$MEDV[i] - predi7
}
# Finding PRESS, RMSE and R2 value for the model
PRESS7 = sum(cv_res7^2)
RMSE7 = sqrt(PRESS7/n)
```

The model shifted to the negative side, so we will stop our analysis for finding a new model and will consider the 6th model to be the best fit. Let's check it in a plot

```
# plotting the non-linear equation on MEDV
ggplot(Housing, aes(LSTAT+PTRATIO, MEDV) ) +
  geom_point() +
  stat_smooth(method = lm, formula = y ~ poly(x, 5, raw = TRUE))
```



Classification Model for predicting per capita crime rate

First We will split the dataset into two parts- Test Dataset and Train Dataset and then we will use logistic regression for Classification of crime rate. Here we are given a dummy variable CR01 which divides CRIM to 0 or 1 depending on the frequency of per capita crime rate(=1 if above median; 0 otherwise) which we will use. Then we will analyze the variable. For the model we will use use NOX, AGE, DIS and RAD as predictor variables since they have the highest correlation with CR01. From then we will use leave one out cross validation to further improve our model.

```
set.seed(599)
# splitting the data
testindex=sample(1:n,n/3)
HouseTest= Housing[testindex,]
HouseTrain = Housing[-testindex,]
# running a logarithmic(glm model) regression on CR01
train_glm=glm(CR01~DIS+RAD+TAX+NOX+MEDV,family=binomial,data=HouseTrain)
testprob1=predict(train_glm,HouseTest,type="response")
testpred1=rep("Low",length=length(testprob1))
testpred1[testprob1>0.5]="High"
table(HouseTest$CR01, testpred1)
```

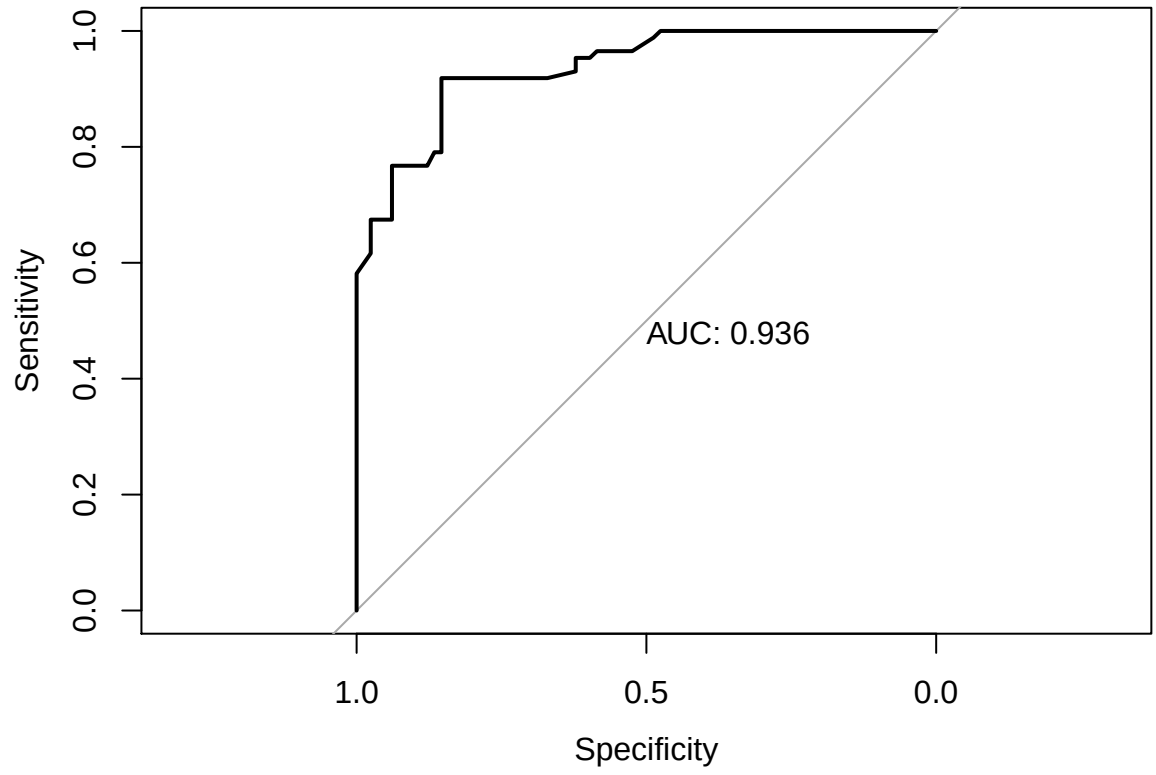
```
##    testpred1
##     High Low
##   0    5  77
##   1   66  20
```

```
train_glm2=glm(CR01~RAD+TAX+NOX,family=binomial,data=HouseTrain)
testprob2=predict(train_glm2,HouseTest,type="response")
testpred2=rep("Low",length=length(testprob2))
testpred2[testprob2>0.5]="High"
# confusion matrix
con<-table(HouseTest$CR01, testpred2)
# ROC plot
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.

##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
test_roc<- roc(HouseTest$CR01~ testprob2,plot=TRUE,print.auc = TRUE)
```

```
## Setting levels: control = 0, case = 1

## Setting direction: controls < cases
```



```
Truepositiverate <- con[2,2]/(con[2,2]+con[2,1])
Falsepositiverate <- con[1,2]/(con[1,1]+con[1,2])
```

We tried different models to achieve precision and came up with a logarithmic regression model with 'RAD', 'TAX', 'NOX'. The confusion matrix shows that the model is not very good i.e. has low sensitivity and specificity.With respect to that the ROC curve also shows that the model can have few improvement with area under the curve to be 0.966 and more the area under the curve (AUC), better is the predicted model.