Boston Housing Data: Analysis

Manjusha Kancharla

September 16, 2015

This data-set is available in the library **MASS** in R. We can load it using:

```
#library(MASS)
#data(Boston)
```

OR load the data through a file on your computer.

Let us take a cursory look at what variables the data-set *Boston* contains.

```
colnames(Boston)
```

```
##  [1] "CRIM"    "ZN"      "INDUS"   "CHAS"    "NOX"     "RM"      "AGE"
##  [8] "DIS"     "RAD"     "TAX"     "PTRATIO" "B"       "LSTAT"   "MEDV"
```

The description of these 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	Nitric 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 distances to five Boston employment centers
RAD	Index of accessibility to radial highways
TAX	Full-value property-tax rate per $10,000
PTRATIO	Pupil-teacher ratio by town
B	1000(Bk - 0.63)^2 where Bk is the proportion of African-Americans by town
LSTAT	% Lower status of the population
MEDV	Median value of owner-occupied homes in $1000's
CAT.MEDV	Binary variable that indicates based on the MEDV variable. If MEDV > 30, CAT.MEDV = 1

Size of the data:

```
nrow(Boston) # To check how many observations we have

## [1] 506
```

Let us check if we have any missing data points:

```
sum(is.na(Boston))

## [1] 0

# If this value is '0' we dont have to worry about missing data.
```

The goal is to predict new house prices based on the information available in the data. Your respondent variable will be **MEDV** and the remaining variables can be seen as potential regressors.

Am going to divide the data into two parts, one for performing the analysis (training set) and the other for validation (testing set) in the ratio 80:20. We have to randomly select the rows to be used for validation.

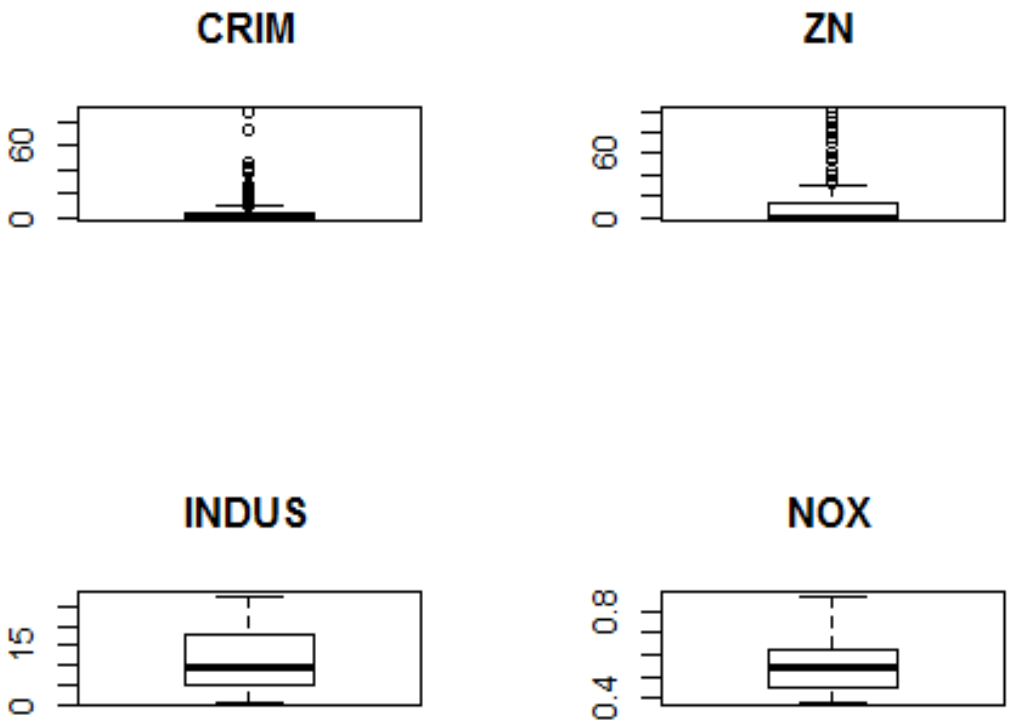
```
Boston_train<-read.csv("Boston_train.csv",header=T)
Boston_test<-read.csv("Boston_test.csv",header=T)

### Select rows randomly
#row.number<- sample(1:nrow(Boston), size=0.2*nrow(Boston))

# Split the data
#Boston_test<-  Boston[row.number,]
dim(Boston_test) ## Size of the testing set

## [1] 101  15

#Boston_train<- Boston[-row.number,]
dim(Boston_train)  ## Size of the training set

## [1] 405  15
```

We will now work only on the training data set: **Boston_train**.

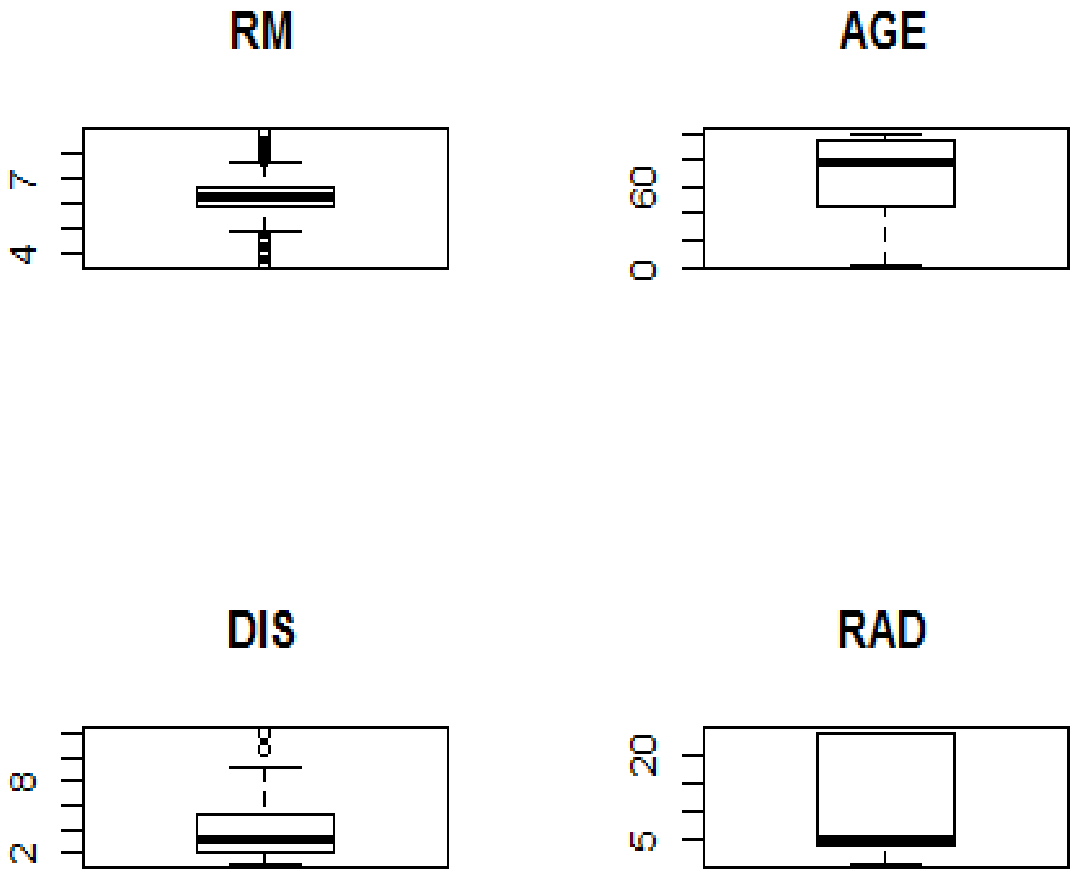
Before going forward, we must keep in mind that there is no one way to perform an analysis. The following is **not** intended to serve as a template for analyses.

1. Let us first look at a summary of our variables. Say, through Box-Plots:

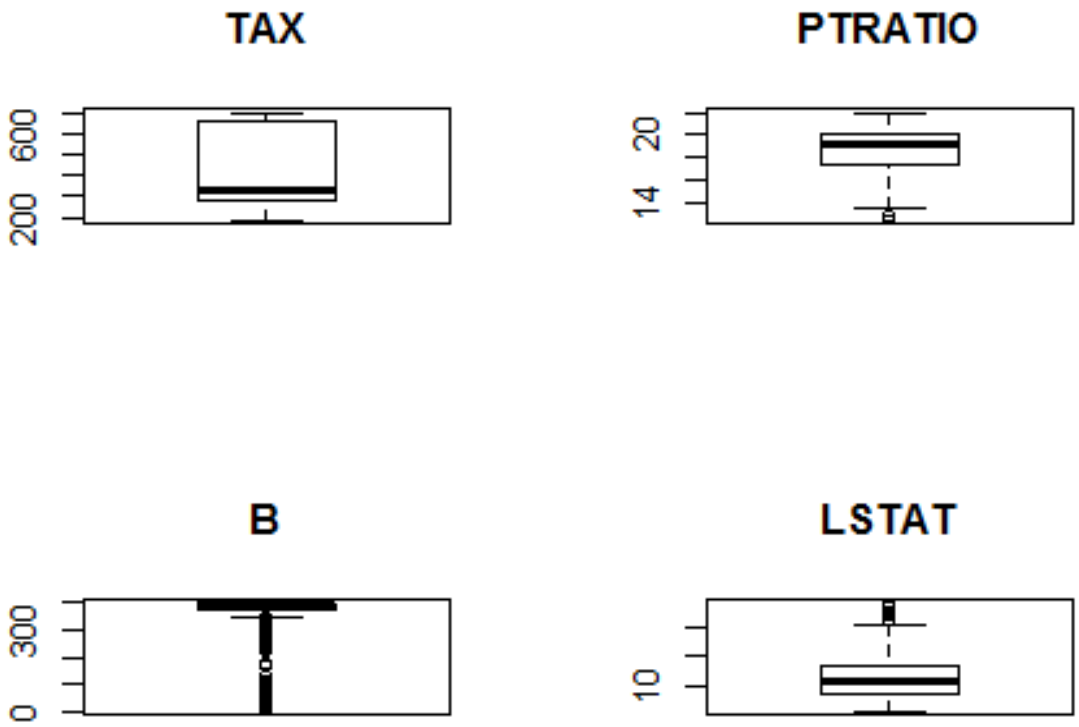
```
par(mfrow=c(2, 2))
boxplot(Boston_train$CRIM, main="CRIM")
boxplot(Boston_train$ZN, main="ZN")
boxplot(Boston_train$INDUS, main="INDUS")
boxplot(Boston_train$NOX, main="NOX")
```



```
par(mfrow=c(2, 2))
boxplot(Boston_train$RM, main="RM")
boxplot(Boston_train$AGE, main="AGE")
boxplot(Boston_train$DIS, main="DIS")
boxplot(Boston_train$RAD, main="RAD")
```



```
par(mfrow=c(2, 2))
boxplot(Boston_train$TAX, main="TAX")
boxplot(Boston_train$PTRATIO, main="PTRATIO")
boxplot(Boston_train$B, main="B")
boxplot(Boston_train$LSTAT, main="LSTAT")
```



Several things we observe here:

a. Variables means are not comparable

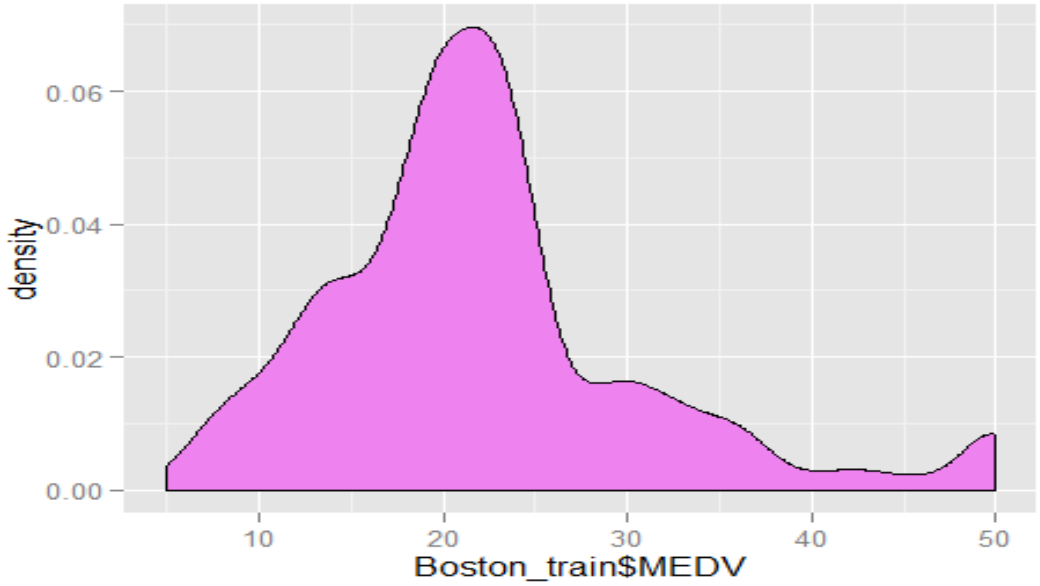
b. Numerous outliers in a few variables, e.g. in __B, CRIM__.

c. Variables don't look symmetric, e.g __RAD , TAX__.

Though Linear Regression does not make any assumptions on the distribution of Regressors, it is always useful to have an idea about them. They help us when looking for to find (influential) outliers. We will come back to this later once our initial model is fit.

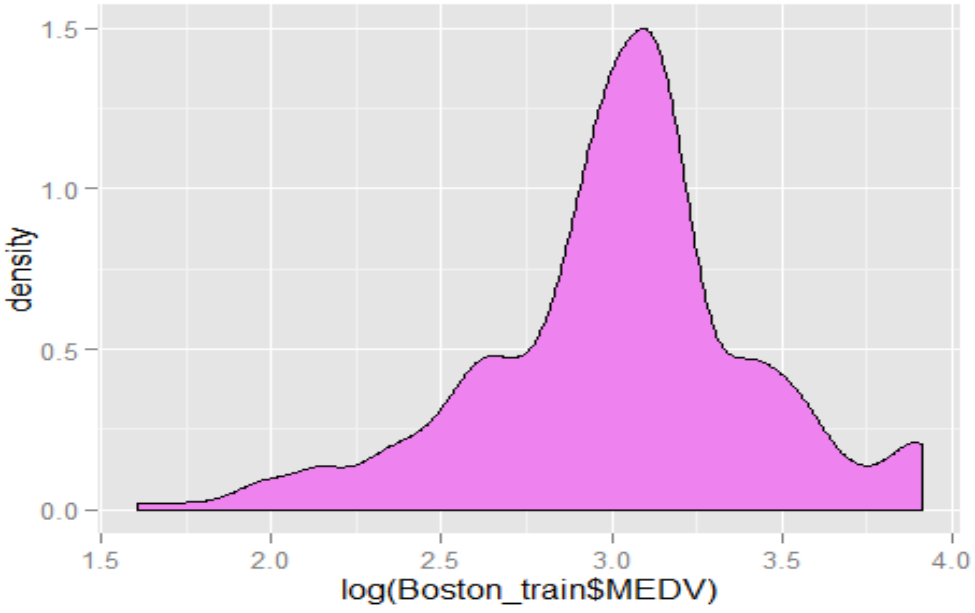
2. Distribution of MEDV

```
library(ggplot2)
dat <- data.frame(x = Boston_train$MEDV)
ggplot(dat, aes(x=Boston_train$MEDV)) + geom_density(fill="violet")
```



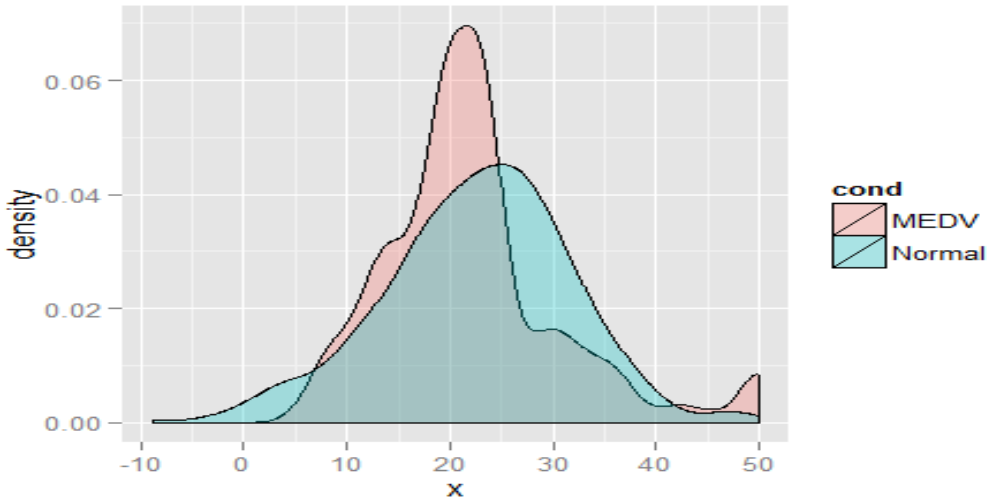
We see that **MEDV** is skewed to the right. We can try a *log* transformation.

```
library(ggplot2)
dat <- data.frame(x = log(Boston_train$MEDV))
ggplot(dat, aes(x=log(Boston_train$MEDV))) + geom_density(fill="violet")
```

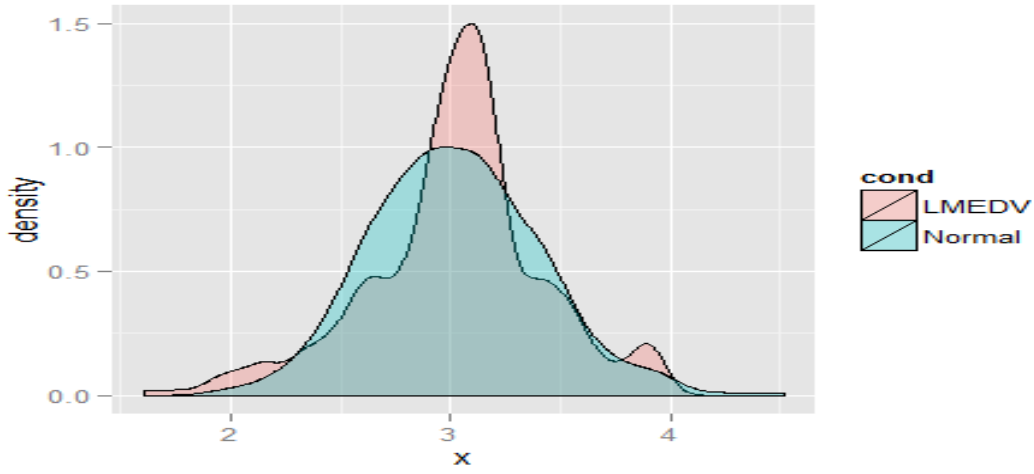


Let us compare the distribution of **MEDV** and **log(MEDV)** with Normal distributions with similar means and standard deviations.

```
norm<-rnorm(405, mean=mean(Boston_train$MEDV), sd=sd(Boston_train$MEDV))
dat <- data.frame(cond = factor(rep(c("MEDV","Normal"), each=405)),
                  x = c(Boston_train$MEDV,norm))
ggplot(dat, aes(x, fill=cond)) + geom_density(alpha=.3)
```




```
lnorm<-rnorm(405, mean=mean(log(Boston_train$MEDV)), sd=sd(log(Boston_train$MEDV)))
dat <- data.frame(cond = factor(rep(c("LMEDV","Normal"), each=405)),
                  x = c(log(Boston_train$MEDV),lnorm))
ggplot(dat, aes(x, fill=cond)) + geom_density(alpha=.3)
```



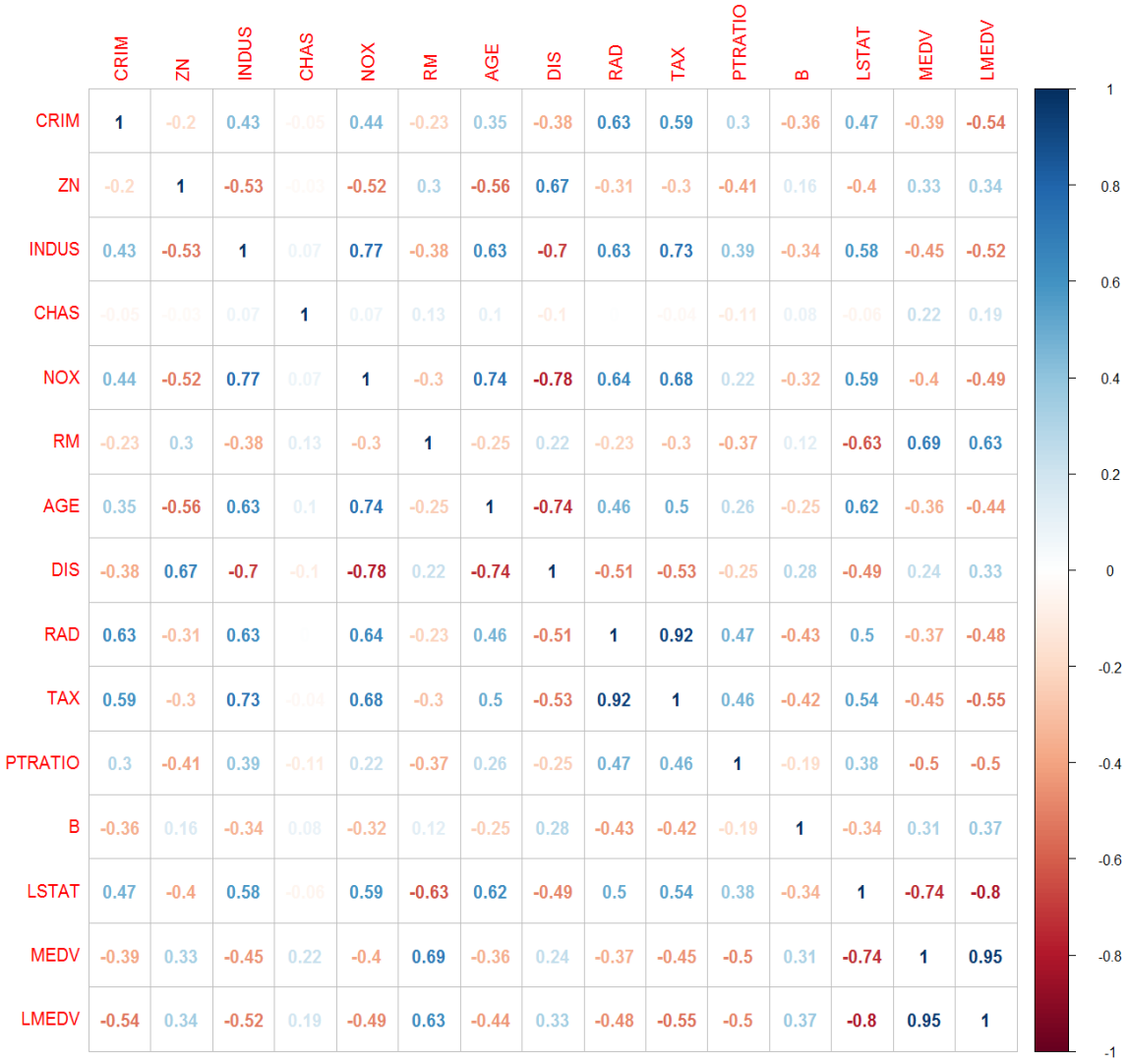
Note: A checking for Normality of the Dependent Variable is not needed as the assumption LM makes is that of Normality of errors. Our goal in transforming MEDV is not to make it symmetric but to achieve linear relations. In our case a log-transformation is needed to achieve this.

3. Correlation matrix and Scatterplot matrix

Considering the large number of variables in our data, we will not insert a scatter plot matrix here. I strongly suggest you plot the Scatter plot matrix and look at it.

Here, we will look at the **Correlation matrix**.

```
mcor<-round(cor(Boston_train),2)
library(corrplot)
png(height=1200, width=1500, pointsize=20, file="Correlation Matrix.jpg")
corrplot(mcor, method="number")
```



	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV	LMEDV
CRIM	1	-0.2	0.43	-0.05	0.44	-0.23	0.35	-0.38	0.63	0.59	0.3	-0.36	0.47	-0.39	-0.54
ZN	-0.2	1	-0.53	-0.03	-0.52	0.3	-0.56	0.67	-0.31	-0.3	-0.41	0.16	-0.4	0.33	0.34
INDUS	0.43	-0.53	1	0.07	0.77	-0.38	0.63	-0.7	0.63	0.73	0.39	-0.34	0.58	-0.45	-0.52
CHAS	-0.05	-0.03	0.07	1	0.07	0.13	0.1	-0.1	0	-0.04	-0.11	0.08	-0.06	0.22	0.19
NOX	0.44	-0.52	0.77	0.07	1	-0.3	0.74	-0.78	0.64	0.68	0.22	-0.32	0.59	-0.4	-0.49
RM	-0.23	0.3	-0.38	0.13	-0.3	1	-0.25	0.22	-0.23	-0.3	-0.37	0.12	-0.63	0.69	0.63
AGE	0.35	-0.56	0.63	0.1	0.74	-0.25	1	-0.74	0.46	0.5	0.26	-0.25	0.62	-0.36	-0.44
DIS	-0.38	0.67	-0.7	-0.1	-0.78	0.22	-0.74	1	-0.51	-0.53	-0.25	0.28	-0.49	0.24	0.33
RAD	0.63	-0.31	0.63	0	0.64	-0.23	0.46	-0.51	1	0.92	0.47	-0.43	0.5	-0.37	-0.48
TAX	0.59	-0.3	0.73	-0.04	0.68	-0.3	0.5	-0.53	0.92	1	0.46	-0.42	0.54	-0.45	-0.55
PTRATIO	0.3	-0.41	0.39	-0.11	0.22	-0.37	0.26	-0.25	0.47	0.46	1	-0.19	0.38	-0.5	-0.5
B	-0.36	0.16	-0.34	0.08	-0.32	0.12	-0.25	0.28	-0.43	-0.42	-0.19	1	-0.34	0.31	0.37
LSTAT	0.47	-0.4	0.58	-0.06	0.59	-0.63	0.62	-0.49	0.5	0.54	0.38	-0.34	1	-0.74	-0.8
MEDV	-0.39	0.33	-0.45	0.22	-0.4	0.69	-0.36	0.24	-0.37	-0.45	-0.5	0.31	-0.74	1	0.95
LMEDV	-0.54	0.34	-0.52	0.19	-0.49	0.63	-0.44	0.33	-0.48	-0.55	-0.5	0.37	-0.8	0.95	1

Combining this with a Scatter plot matrix may give us hints as to which of these numbers are potentially spurious , which of the independent variables are expected to be significant in explaining **MEDV**, if there is possible *pairwise collinearity* etc..

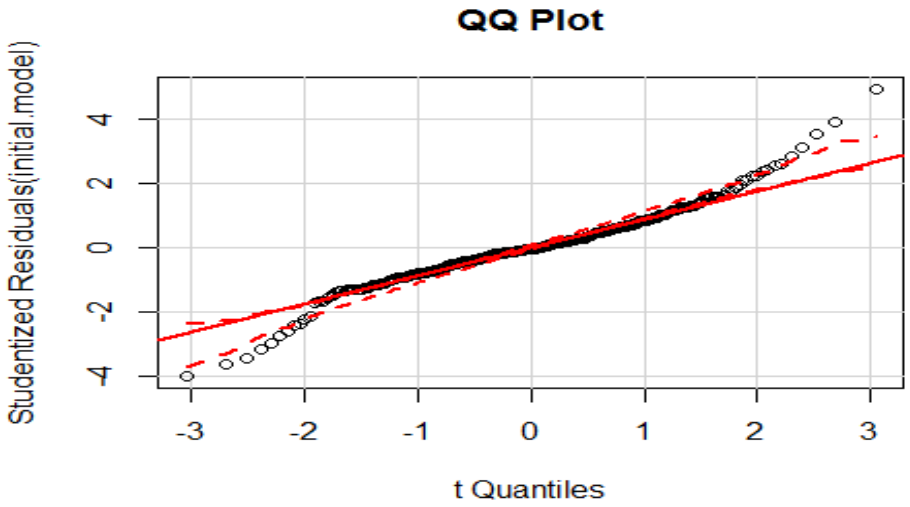
4. Initial Linear Model and Basic Diagnostics

Now that we have all the basic information on our data, let us start with a basic model.

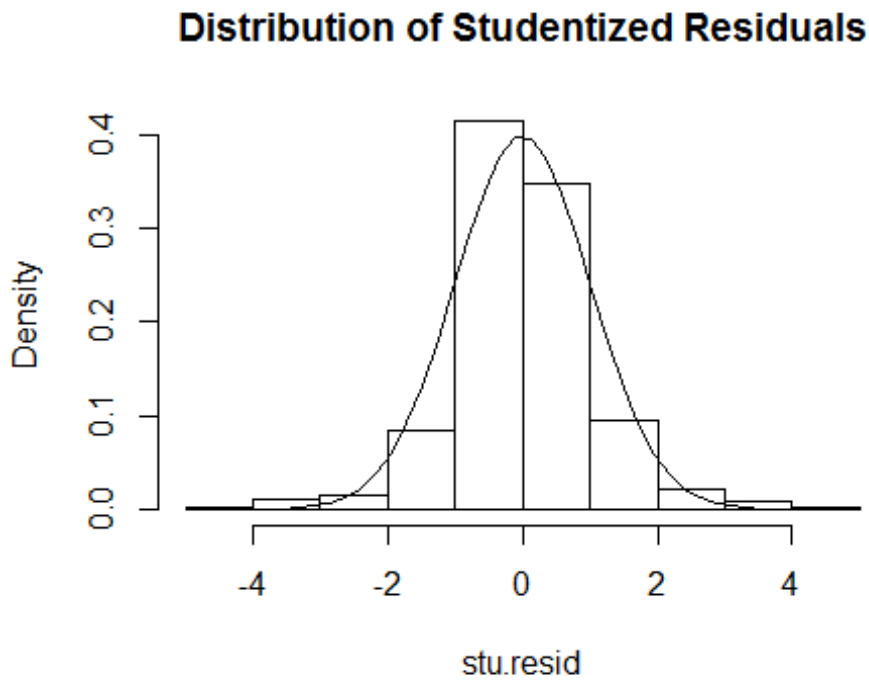
$$\log(\text{MEDV}) = \beta_0 + \beta_1 * \text{CRIM} + \beta_2 * \text{ZN} + \beta_3 * \text{INDUS} + \beta_4 * \text{factor(CHAS)} + \beta_5 * \text{NOX} + \beta_6 * \text{RM} + \beta_7 * \text{AGE} + \beta_8 * \text{DIS} + \beta_9 * \text{RAD} + \beta_{10} * \text{TAX} + \beta_{11} * \text{PTRATIO} + \beta_{12} * \text{B} + \beta_{13} * \text{LSTAT} + \epsilon$$

The summary of this model:

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    4.0888831  0.2241471  18.242  < 2e-16 ***
## CRIM          -0.0092685  0.0015116  -6.132 2.13e-09 ***
## ZN             0.0014746  0.0006210   2.375 0.018047 *
## INDUS          0.0022510  0.0026790   0.840 0.401293
## factor(CHAS)1  0.1051435  0.0400359   2.626 0.008973 **
## NOX           -0.7452449  0.1717924  -4.338 1.83e-05 ***
## RM             0.0857975  0.0188035   4.563 6.77e-06 ***
## AGE            0.0003038  0.0005974   0.509 0.611336
## DIS           -0.0505835  0.0090642  -5.581 4.49e-08 ***
## RAD            0.0133004  0.0029440   4.518 8.29e-06 ***
## TAX           -0.0006213  0.0001677  -3.705 0.000242 ***
## PTRATIO       -0.0365373  0.0058699  -6.225 1.25e-09 ***
## B              0.0004099  0.0001178   3.480 0.000558 ***
## LSTAT         -0.0296774  0.0023224 -12.779  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Residual standard error: 0.1871 on 391 degrees of freedom
## Multiple R-squared:  0.7929, Adjusted R-squared:  0.7861
## F-statistic: 115.2 on 13 and 391 DF,  p-value: < 2.2e-16
```



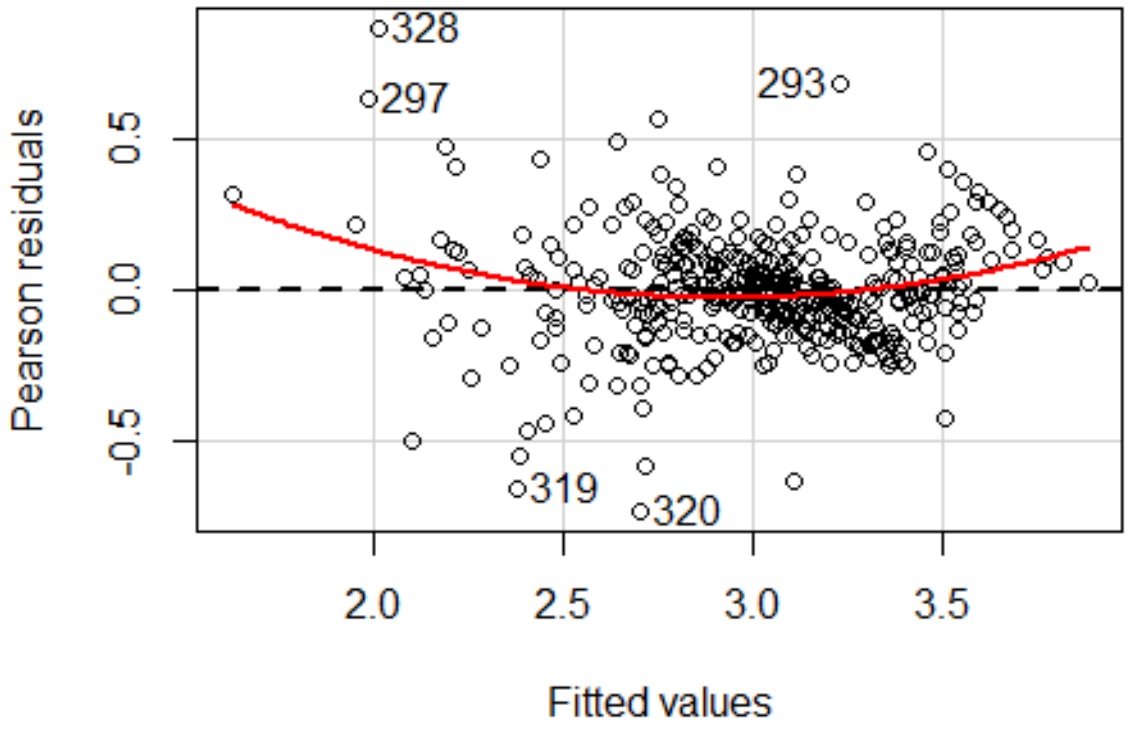
```
# distribution of studentized residuals
library(MASS)

##
## Attaching package: 'MASS'
##
## The following object is masked _by_ '.GlobalEnv':
##
##     Boston

stu.resid <- studres(initial.model)
hist(stu.resid, freq=FALSE,
  main="Distribution of Studentized Residuals")
xfit<-seq(min(stu.resid),max(stu.resid),length=40)
yfit<-dnorm(xfit)
lines(xfit, yfit)
```



Looking at the above results, without looking at anything we have done before this, it looks like a pretty good fit. There appears to be no serious problem.

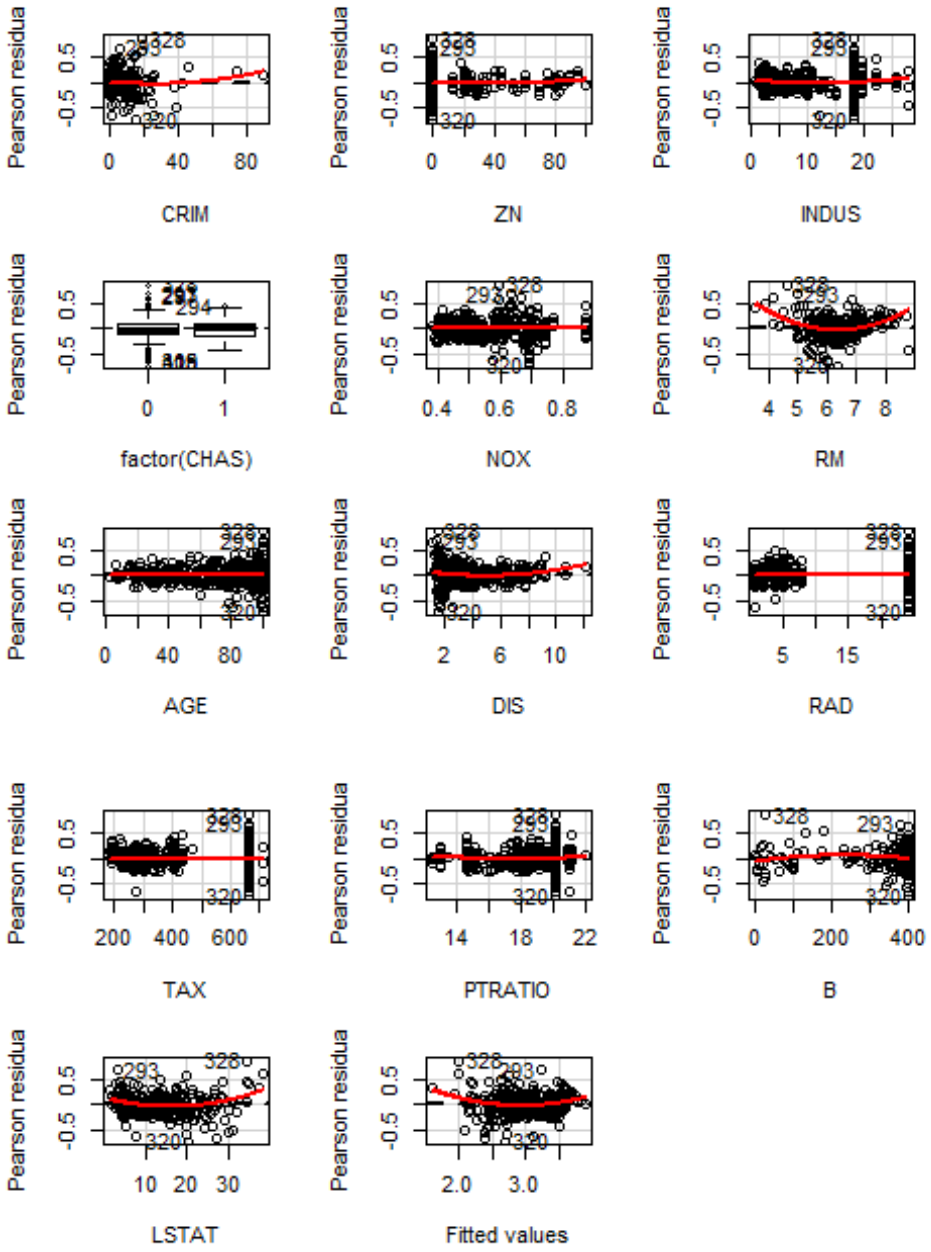
Take a look at the fitted values vs. Residuals plot below. There seems to be a trend. Clearly, something is wrong with our initial model.

```
library(car)
residualPlot(initial.model, id.n=5)
```



Maybe we can learn more if we look at the plots of each regressors vs. residuals.

```
library(car)
residualPlots(initial.model,id.n=3)
```



```
##            Test stat Pr(>|t|)
## CRIM           2.714    0.007
## ZN             1.341    0.181
## INDUS          2.196    0.029
## factor(CHAS)      NA       NA
## NOX            0.030    0.976
## RM             7.966    0.000
## AGE            0.593    0.553
## DIS            3.393    0.001
## RAD           -0.582    0.561
## TAX           -0.258    0.796
## PTRATIO        1.250    0.212
## B             -2.039    0.042
## LSTAT          5.683    0.000
## Tukey test     5.625    0.000
```

Looking at the above residual plots and earlier Box-Plots, we have:

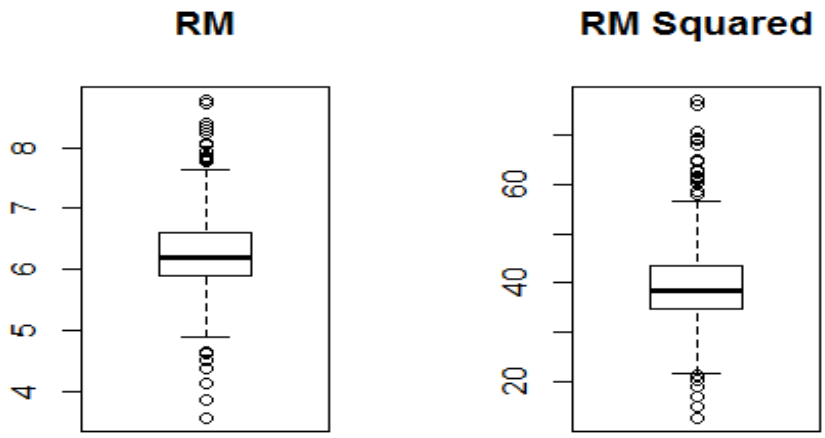
1. **RM**: Quadratic trasformation
2. **DIS**: Outlier influence.[Observations: 293,320, 328]
3. **RAD**: Logarithimic transformation (Since skewed to right)
4. **B**: Reflected logarithmic transformation (since skewed to the left)
5. **LSTAT**: Outlier influence[Observations: 293,320, 328]
6. **AGE**: Reflected logarithmic transformation (since skewed to the left)
7. **CRIM**: Logarithimic transformation (Since skewed to right)

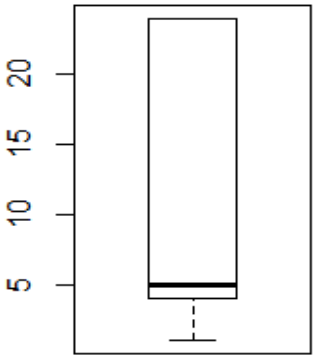
In our initial model summary, check how each of the above variables performs.

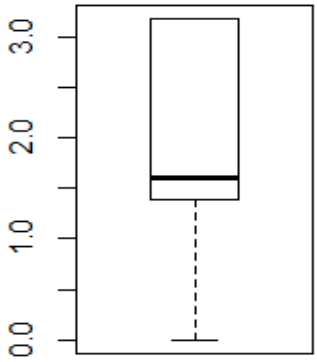
5. Transformations

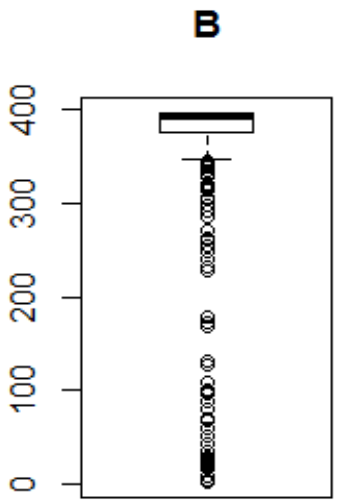
```
## R Code

RM_Sq<-(Boston_train$RM)^2
LRAD<-log(Boston_train$RAD)
LB<-(log(max(Boston_train$B)+1-Boston_train$B))
LAGE<-(log(max(Boston_train$AGE)+1-Boston_train$AGE))
LCRIM<-log(Boston_train$CRIM)
LDIS<-log(Boston_train$DIS)
```

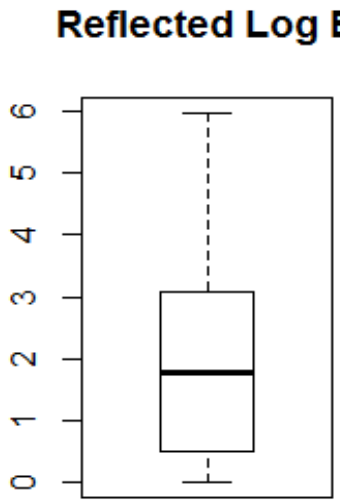
Boxplots of the transformed variables:



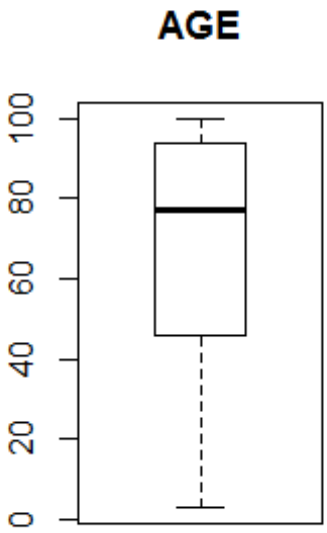


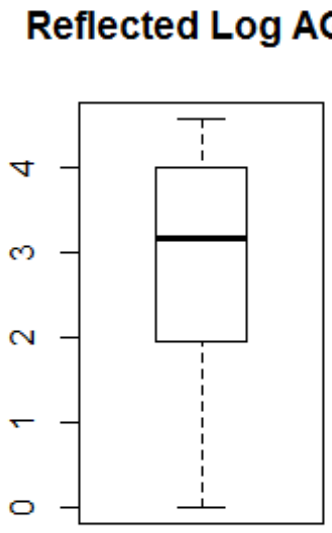


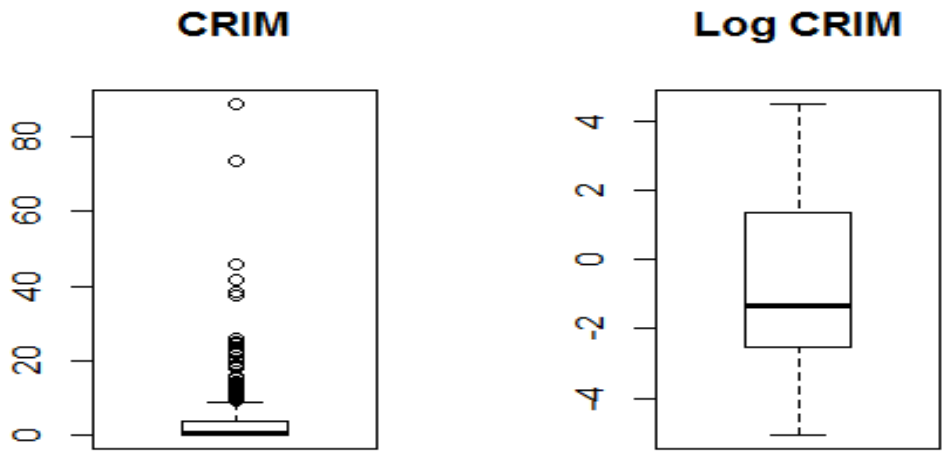




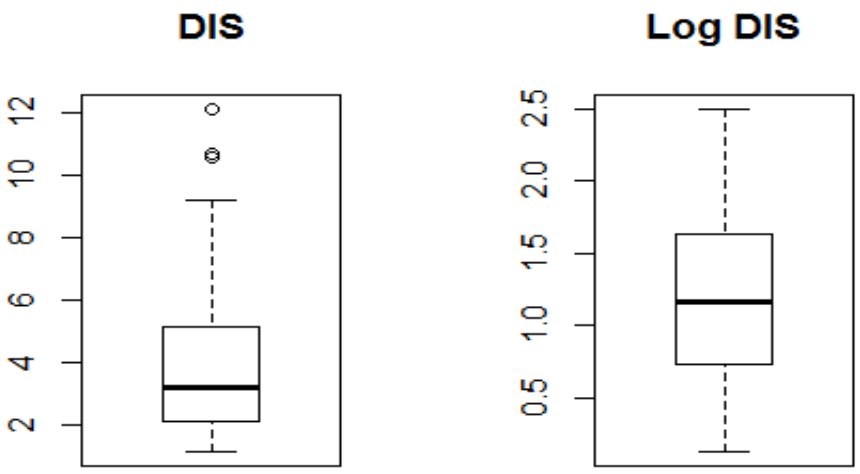

```
par(mfrow=c(1, 2))
boxplot(Boston_train$AGE, main="AGE")
boxplot(LAGE, main="Reflected Log AGE")
```







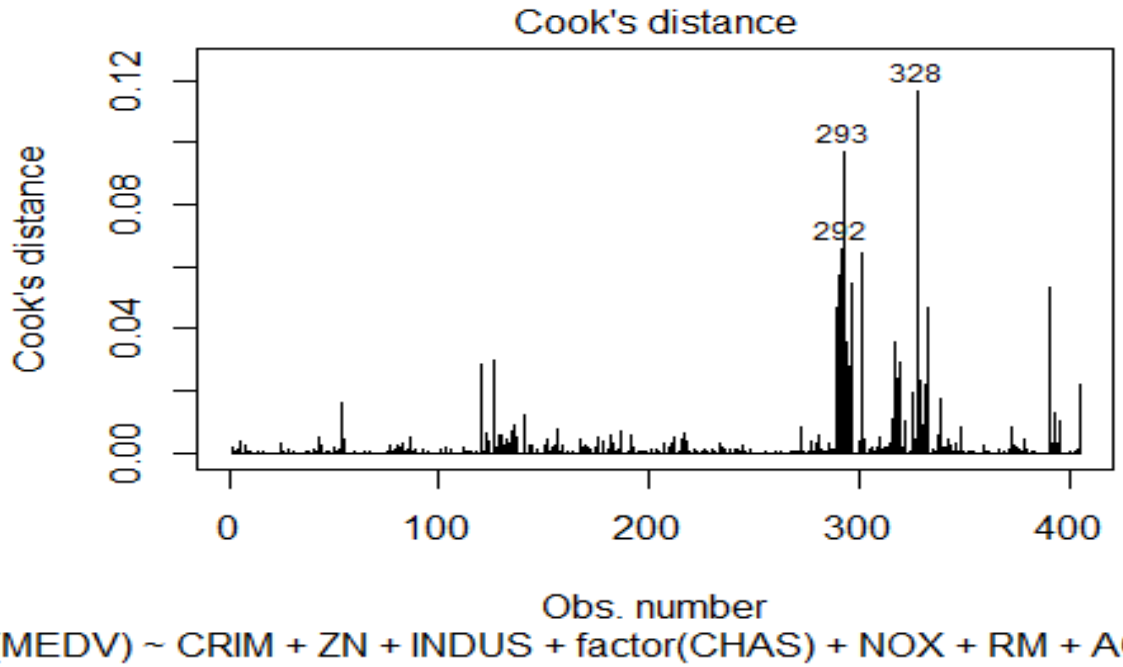

```
par(mfrow=c(1, 2))
boxplot(Boston_train$DIS, main="DIS")
boxplot(LDIS, main="Log DIS")
```



6. Checking for Influential Observations/ Deletion Diagnostics

We have seen in earlier that a few observations (293,320, 328) are outliers. Let us check if they are influential.

```
# Delition Diagnostics
#influence.measures(initial.model)
# Cook's Distance plot
# identify D values > 4/(n-k-1)
cutoff <- 4/((nrow(Boston_train)-length(initial.model$coefficients)-2))
plot(initial.model, which=4, cook.levels=cutoff)
```



We can see that the observation 320 is not influential, but 293 and 328 are. This can be crosschecked with other influential observations diagnostics.

What to do with the influential observations [289,293 and 328]?

We can check how our model behaves with and without them. If there is a difference or reversal in behavior we cannot ignore them. You can check that in our case, their removal doesn't affect the model.

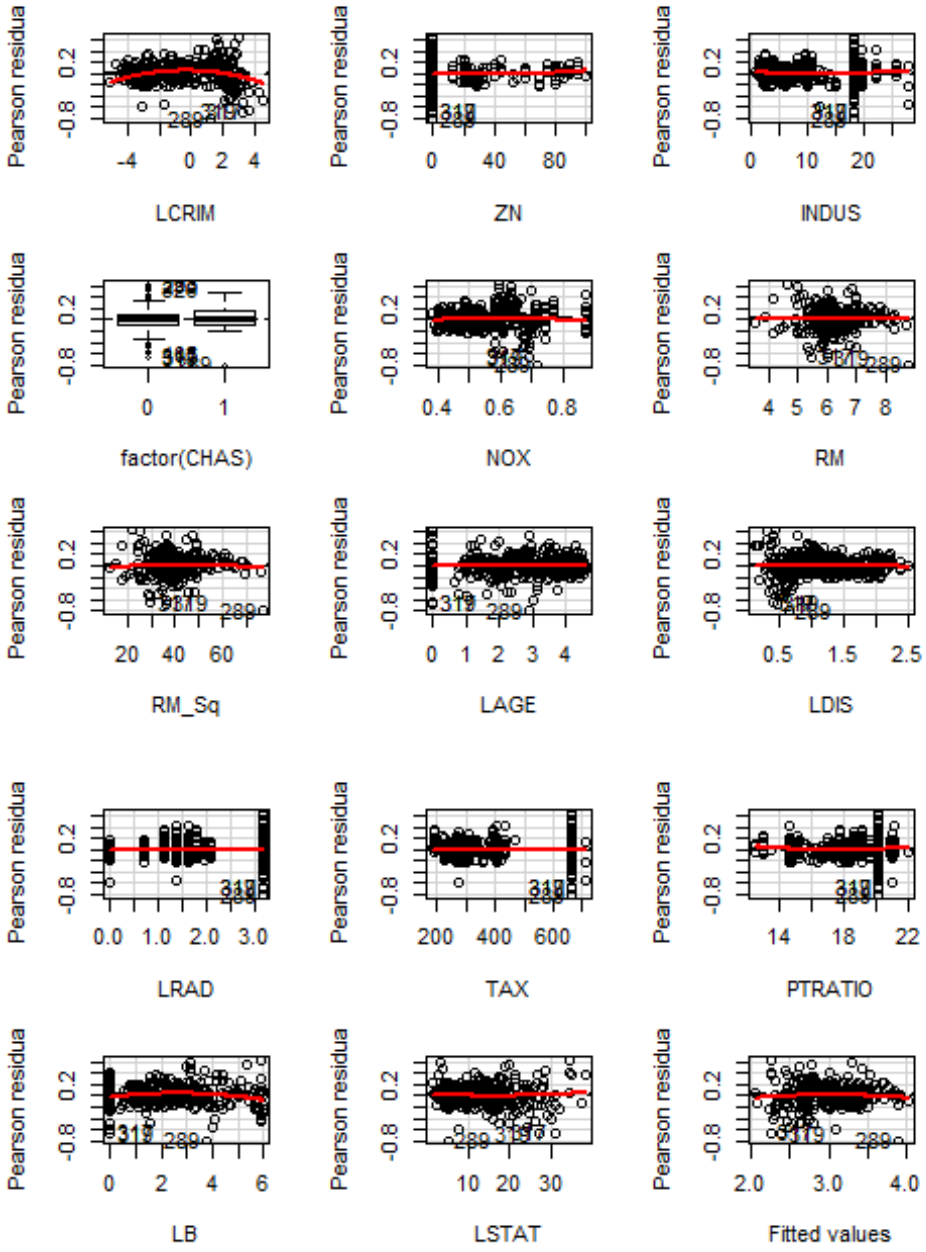
7. Next iteration of our model

This model will include the transformed variables and the data used will not have the 320th observation.

Summary of the new model:

```
# Modified data
Boston_train0<-cbind(Boston_train,RM_Sq,LRAD,LB,LAGE,LCRIM,LDIS)
model_1<-
lm(log(MEDV)~LCRIM+ZN+INDUS+factor(CHAS)+NOX+RM+RM_Sq+LAGE+LDIS+LRAD+TAX+PTRATIO+LB+
LSTAT, data=Boston_train1)

## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.7983312  0.4261829  15.952  < 2e-16 ***
## LCRIM        -0.0268675  0.0124232  -2.163 0.031174 *
## ZN            0.0001379  0.0005916   0.233 0.815851
## INDUS         0.0046818  0.0026444   1.770 0.077433 .
## factor(CHAS)1  0.1180356  0.0389663   3.029 0.002616 **
## NOX          -0.6406518  0.1763129  -3.634 0.000317 ***
## RM           -0.8193874  0.1317146  -6.221 1.28e-09 ***
## RM_Sq         0.0720628  0.0103837   6.940 1.65e-11 ***
## LAGE         -0.0064994  0.0121118  -0.537 0.591841
## LDIS         -0.1523251  0.0403880  -3.772 0.000187 ***
## LRAD          0.0853041  0.0241345   3.535 0.000458 ***
## TAX          -0.0005315  0.0001372  -3.874 0.000126 ***
## PTRATIO      -0.0287443  0.0057125  -5.032 7.44e-07 ***
## LB           -0.0068256  0.0058785  -1.161 0.246308
## LSTAT        -0.0335144  0.0022177 -15.112  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.183 on 389 degrees of freedom
## Multiple R-squared:  0.7997, Adjusted R-squared:  0.7924
## F-statistic: 110.9 on 14 and 389 DF,  p-value: < 2.2e-16
```

The Residuals vs. Regressors plots



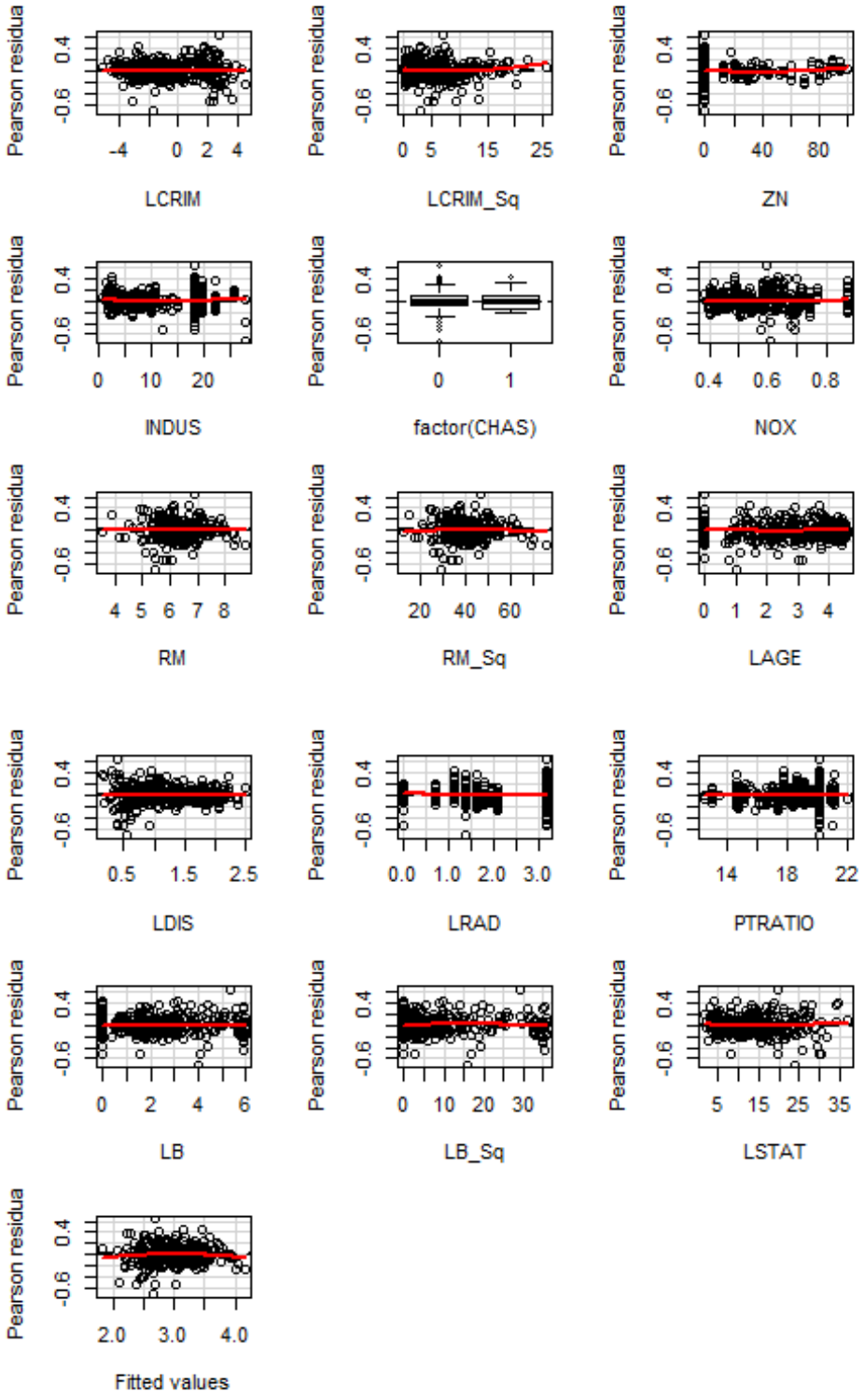
```
##            Test stat Pr(>|t|)
## LCRIM          -5.782    0.000
## ZN              1.230    0.219
## INDUS           1.123    0.262
## factor(CHAS)       NA       NA
## NOX            -0.583    0.560
## RM              0.523    0.601
## RM_Sq          -2.863    0.004
## LAGE           -0.107    0.915
## LDIS           -0.727    0.468
## LRAD            0.156    0.876
## TAX            -0.126    0.899
## PTRATIO         1.649    0.100
## LB             -4.246    0.000
## LSTAT           1.025    0.306
## Tukey test      -1.974    0.048
```

We can spot that **LCRIM and LB** need quadratic components as well. Next model [Model_3] will include these quadratic terms.

```
LCRIM_Sq<-LCRIM^2
LB_Sq<-LB^2
Boston_train3<-cbind(Boston_train0,LCRIM_Sq,LB_Sq)
```

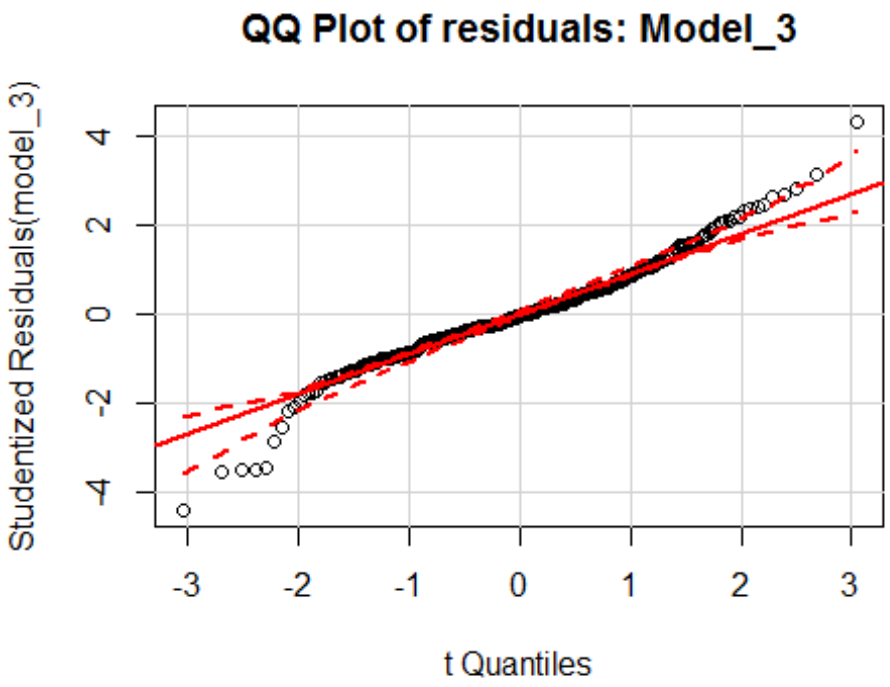
```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.3493421  0.3924133  16.180  < 2e-16 ***
## LCRIM         -0.0449008  0.0107662  -4.171 3.76e-05 ***
## LCRIM_Sq      -0.0153798  0.0022740  -6.763 5.06e-11 ***
## ZN             0.0010212  0.0005318   1.920 0.055561 .
## INDUS          0.0007066  0.0021467   0.329 0.742218
## factor(CHAS)1  0.1405395  0.0340597   4.126 4.53e-05 ***
## NOX           -0.4367429  0.1541489  -2.833 0.004851 **
## RM            -0.8492680  0.1227450  -6.919 1.92e-11 ***
## RM_Sq          0.0782907  0.0097281   8.048 1.07e-14 ***
## LAGE           0.0165986  0.0107200   1.548 0.122358
## LDIS          -0.1542581  0.0353123  -4.368 1.61e-05 ***
## LRAD           0.0431834  0.0181793   2.375 0.018021 *
## PTRATIO       -0.0239697  0.0048685  -4.923 1.27e-06 ***
## LB             0.0381835  0.0143201   2.666 0.007991 **
## LB_Sq         -0.0102086  0.0027114  -3.765 0.000193 ***
## LSTAT         -0.0282013  0.0020439 -13.798  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Residual standard error: 0.1569 on 383 degrees of freedom
## Multiple R-squared:  0.8488, Adjusted R-squared:  0.8429
## F-statistic: 143.4 on 15 and 383 DF,  p-value: < 2.2e-16
```

The Residuals vs. Regressors plots

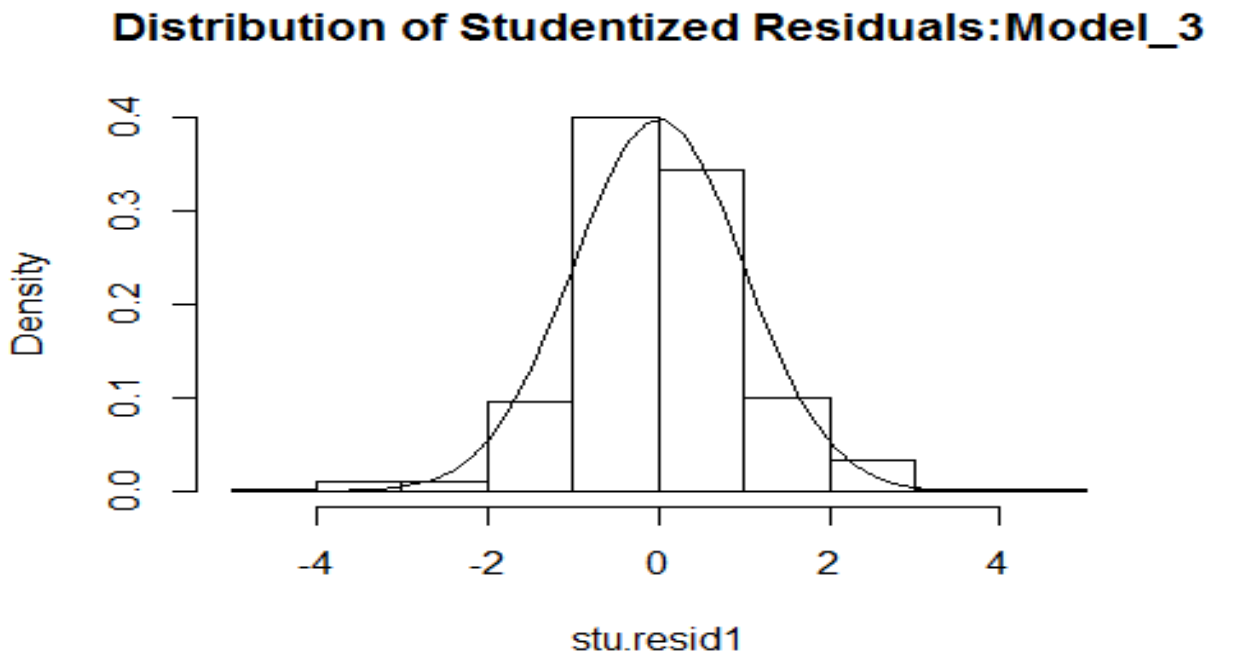


```
##            Test stat Pr(>|t|)
## LCRIM          -0.678    0.498
## LCRIM_Sq        2.121    0.035
## ZN              1.575    0.116
## INDUS           1.030    0.304
## factor(CHAS)       NA       NA
## NOX             0.308    0.758
## RM              1.904    0.058
## RM_Sq          -2.390    0.017
## LAGE            1.094    0.275
## LDIS           -0.257    0.798
## LRAD            1.818    0.070
## PTRATIO        -0.562    0.574
## LB             -0.183    0.855
## LB_Sq          -1.568    0.118
## LSTAT           0.495    0.621
## Tukey test     -2.681    0.007
```

```
# Normality of Residuals
# qq plot for studentized resid
qqPlot(model_3, main="QQ Plot of residuals: Model_3")
```



```
# distribution of studentized residuals
stu.resid1 <- studres(model_3)
hist(stu.resid1, freq=FALSE,
  main="Distribution of Studentized Residuals:Model_3")
xfit1<-seq(min(stu.resid1),max(stu.resid1),length=40)
yfit1<-dnorm(xfit1)
lines(xfit1, yfit1)
```



There appears to be no problem here.

8. Checking for Collinearity

We have our correlations with us; if not anything else we know that there might be pairwise collinearity. Let us use the Variance Decomposition Proportions to check this.

```
# Evaluate Collinearity
vif(model_3)
```

```
##      LCRIM    LCRIM_Sq         ZN      INDUS factor(CHAS)
##   9.603058    1.958434    2.719912    4.231214    1.072325
##        NOX         RM       RM_Sq       LAGE       LDIS
##   5.200486  108.634813  111.332468    3.221758    5.768494
##       LRAD         TAX     PTRATIO         LB       LB_Sq
##   5.489093    6.986848    1.892195   10.314008   11.362218
##      LSTAT
##   3.275982
```

```
library(perturb)
colldiag(Boston_train4[,-c(8,10,13,15)], center = TRUE)
```

```
## Condition
## Index    Variance Decomposition Proportions
##          X    CRIM  ZN    INDUS CHAS  NOX   RM    DIS   TAX   PTRATIO
## 1    1.000 0.002 0.001 0.002 0.002 0.000 0.002 0.000 0.000 0.001 0.002
## 2    2.028 0.020 0.007 0.012 0.000 0.001 0.000 0.001 0.000 0.003 0.005
## 3    2.109 0.006 0.006 0.006 0.000 0.030 0.002 0.001 0.002 0.002 0.022
## 4    2.424 0.028 0.005 0.000 0.000 0.000 0.000 0.000 0.000 0.002 0.009
## 5    2.843 0.009 0.013 0.024 0.000 0.184 0.003 0.000 0.000 0.001 0.099
## 6    3.014 0.037 0.006 0.000 0.000 0.618 0.001 0.000 0.001 0.002 0.003
## 7    3.529 0.085 0.043 0.052 0.001 0.126 0.013 0.000 0.000 0.004 0.236
## 8    4.088 0.258 0.035 0.004 0.016 0.004 0.000 0.000 0.005 0.001 0.027
## 9    4.193 0.014 0.065 0.125 0.029 0.005 0.000 0.000 0.003 0.006 0.196
## 10   5.050 0.291 0.000 0.011 0.223 0.000 0.003 0.000 0.000 0.029 0.001
## 11   6.054 0.159 0.134 0.004 0.106 0.001 0.001 0.000 0.017 0.010 0.000
## 12   6.189 0.000 0.011 0.018 0.243 0.001 0.002 0.002 0.004 0.000 0.063
## 13   6.677 0.005 0.066 0.480 0.001 0.002 0.184 0.000 0.021 0.013 0.047
## 14   7.189 0.015 0.011 0.125 0.049 0.002 0.613 0.000 0.001 0.041 0.268
## 15   9.692 0.059 0.000 0.009 0.276 0.014 0.001 0.000 0.000 0.688 0.002
## 16  12.999 0.000 0.129 0.001 0.010 0.005 0.046 0.000 0.001 0.024 0.005
## 17  14.758 0.002 0.453 0.003 0.007 0.004 0.054 0.000 0.000 0.168 0.003
## 18  22.714 0.011 0.012 0.122 0.013 0.003 0.051 0.002 0.923 0.000 0.001
## 19  43.717 0.000 0.003 0.002 0.022 0.001 0.024 0.993 0.020 0.006 0.012
##    LSTAT RM_Sq LRAD  LB    LAGE  LCRIM LDIS  LCRIM_Sq LB_Sq
## 1  0.003 0.000 0.001 0.000 0.002 0.001 0.000 0.000    0.000
## 2  0.002 0.001 0.004 0.005 0.001 0.000 0.000 0.010    0.004
## 3  0.005 0.001 0.001 0.000 0.005 0.000 0.001 0.017    0.000
## 4  0.002 0.000 0.004 0.024 0.000 0.000 0.000 0.003    0.018
## 5  0.013 0.000 0.006 0.001 0.016 0.000 0.001 0.038    0.000
```

```
## 6  0.003 0.000 0.011 0.000 0.014 0.000 0.001 0.011    0.000
## 7  0.000 0.000 0.003 0.001 0.000 0.000 0.000 0.022    0.002
## 8  0.003 0.000 0.055 0.002 0.019 0.013 0.003 0.048    0.003
## 9  0.032 0.000 0.002 0.000 0.121 0.001 0.000 0.000    0.000
## 10 0.223 0.000 0.005 0.000 0.070 0.000 0.000 0.019    0.000
## 11 0.188 0.000 0.145 0.000 0.122 0.003 0.005 0.045    0.000
## 12 0.504 0.001 0.022 0.003 0.224 0.000 0.003 0.012    0.001
## 13 0.000 0.000 0.009 0.000 0.046 0.003 0.012 0.174    0.000
## 14 0.000 0.000 0.020 0.002 0.202 0.005 0.000 0.012    0.001
## 15 0.002 0.000 0.625 0.000 0.022 0.020 0.000 0.023    0.000
## 16 0.016 0.000 0.014 0.724 0.007 0.151 0.004 0.056    0.686
## 17 0.000 0.000 0.070 0.206 0.024 0.787 0.000 0.502    0.255
## 18 0.001 0.004 0.003 0.022 0.105 0.000 0.911 0.003    0.018
## 19 0.003 0.992 0.000 0.009 0.000 0.015 0.058 0.003    0.009
```

We see no problematic relationships.

The high VIFs and Conditional Indices are due to the high correlations between transformed variables. [RM and RM^2 for example].

We can now move on to finding the best subset of variables.

9. Best Subset selection

We will use the AIC criterion for obtaining the best subset.

```
step <- stepAIC(model_3, direction="both")
```

```
## Start:  AIC=-1465.7
## log(MEDV) ~ LCRIM + LCRIM_Sq + ZN + INDUS + factor(CHAS) + NOX +
##     RM + RM_Sq + LAGE + LDIS + LRAD + TAX + PTRATIO + LB + LB_Sq +
##     LSTAT
##
##              Df Sum of Sq     RSS     AIC
## - INDUS       1    0.0322  9.3349 -1466.3
## <none>                     9.3027 -1465.7
## - LAGE        1    0.0672  9.3699 -1464.8
## - TAX         1    0.1210  9.4237 -1462.5
## - ZN          1    0.1385  9.4412 -1461.8
## - NOX         1    0.1740  9.4767 -1460.3
## - LB          1    0.1921  9.4948 -1459.5
## - LRAD        1    0.2446  9.5473 -1457.3
## - LCRIM       1    0.2827  9.5854 -1455.8
## - LB_Sq       1    0.3655  9.6682 -1452.3
## - factor(CHAS)  1    0.3757  9.6784 -1451.9
## - PTRATIO     1    0.4771  9.7798 -1447.8
## - LDIS        1    0.4774  9.7801 -1447.7
## - LCRIM_Sq    1    0.7980 10.1007 -1434.9
## - RM          1    1.1361 10.4388 -1421.7
## - RM_Sq       1    1.5412 10.8439 -1406.5
## - LSTAT       1    4.7375 14.0402 -1303.5
##
## Step:  AIC=-1466.32
## log(MEDV) ~ LCRIM + LCRIM_Sq + ZN + factor(CHAS) + NOX + RM +
##     RM_Sq + LAGE + LDIS + LRAD + TAX + PTRATIO + LB + LB_Sq +
##     LSTAT
##
##              Df Sum of Sq     RSS     AIC
## <none>                     9.3349 -1466.3
## + INDUS       1    0.0322  9.3027 -1465.7
## - LAGE        1    0.0642  9.3991 -1465.6
## - TAX         1    0.0915  9.4264 -1464.4
## - ZN          1    0.1361  9.4710 -1462.5
## - NOX         1    0.1562  9.4911 -1461.7
## - LB          1    0.2024  9.5373 -1459.8
## - LRAD        1    0.2183  9.5532 -1459.1
## - LCRIM       1    0.2833  9.6183 -1456.4
## - LB_Sq       1    0.3714  9.7063 -1452.8
## - factor(CHAS)  1    0.3923  9.7272 -1451.9
```

```
## - PTRATIO      1    0.4534  9.7883 -1449.4
## - LDIS         1    0.6043  9.9392 -1443.3
## - LCRIM_Sq     1    0.8893 10.2242 -1432.0
## - RM           1    1.1057 10.4406 -1423.7
## - RM_Sq        1    1.5090 10.8439 -1408.5
## - LSTAT        1    4.7291 14.0640 -1304.8
```

```
step$anova # display results
```

```
## Stepwise Model Path
## Analysis of Deviance Table
##
## Initial Model:
## log(MEDV) ~ LCRIM + LCRIM_Sq + ZN + INDUS + factor(CHAS) + NOX +
##     RM + RM_Sq + LAGE + LDIS + LRAD + TAX + PTRATIO + LB + LB_Sq +
##     LSTAT
##
## Final Model:
## log(MEDV) ~ LCRIM + LCRIM_Sq + ZN + factor(CHAS) + NOX + RM +
##     RM_Sq + LAGE + LDIS + LRAD + TAX + PTRATIO + LB + LB_Sq +
##     LSTAT
##
##
##      Step Df   Deviance Resid. Df Resid. Dev       AIC
## 1                          382   9.302707 -1465.704
## 2 - INDUS  1 0.03220803       383   9.334915 -1466.325
```

We see that <u>INDUS</u> was the only variable dropped.

Hence, our best model is:

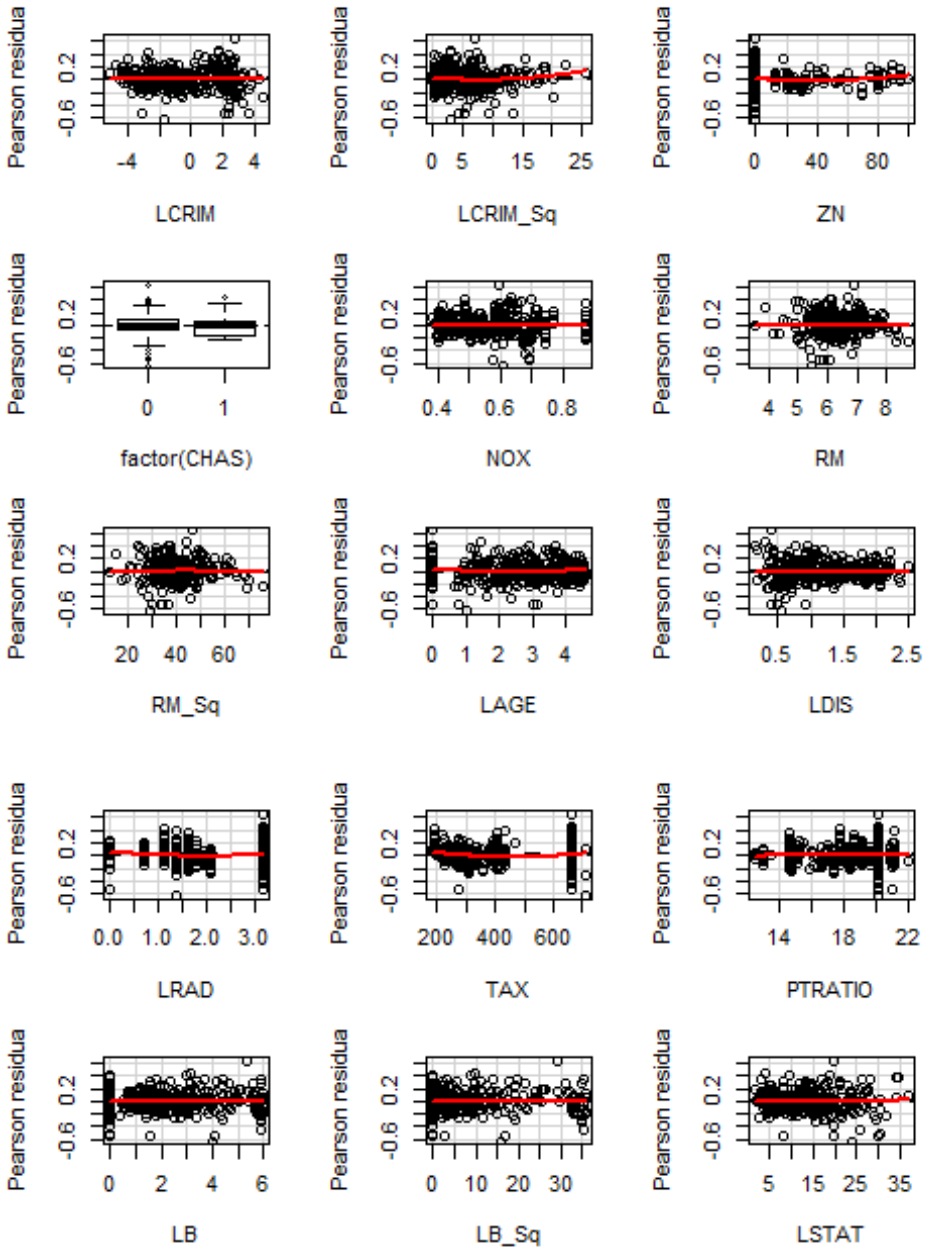
```
best_model<-
lm(log(MEDV)~LCRIM+LCRIM_Sq+ZN+factor(CHAS)+NOX+RM+RM_Sq+LAGE+LDIS+LRAD+TAX+PTRATIO+
LB+LB_Sq+LSTAT, data=Boston_train4)
summary(best_model)
```

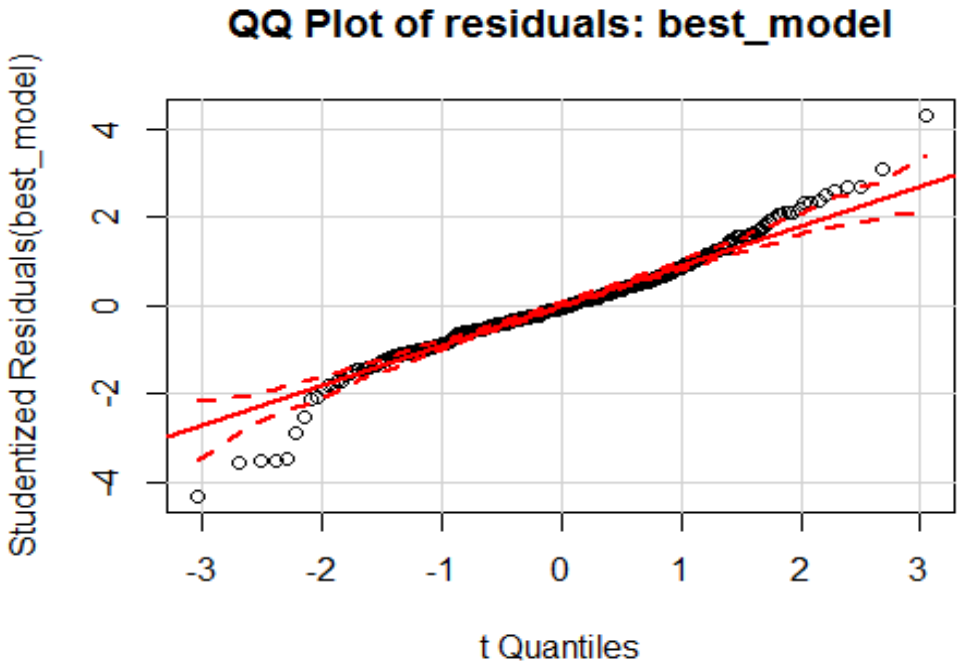
```
##
## Call:
## lm(formula = log(MEDV) ~ LCRIM + LCRIM_Sq + ZN + factor(CHAS) +
##     NOX + RM + RM_Sq + LAGE + LDIS + LRAD + TAX + PTRATIO + LB +
##     LB_Sq + LSTAT, data = Boston_train4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62581 -0.08175 -0.00830  0.07916  0.64084
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.2536263  0.3895059  16.055  < 2e-16 ***
## LCRIM         -0.0380336  0.0111550  -3.410 0.000720 ***
## LCRIM_Sq      -0.0141824  0.0023479  -6.040 3.64e-09 ***
## ZN             0.0012836  0.0005433   2.363 0.018638 *
## factor(CHAS)1  0.1362594  0.0339636   4.012 7.24e-05 ***
## NOX           -0.3853472  0.1522232  -2.531 0.011758 *
## RM            -0.8170749  0.1213091  -6.735 6.00e-11 ***
## RM_Sq          0.0754609  0.0095904   7.868 3.71e-14 ***
## LAGE           0.0173291  0.0106757   1.623 0.105362
## LDIS          -0.1673781  0.0336152  -4.979 9.67e-07 ***
## LRAD           0.0605184  0.0202216   2.993 0.002944 **
## TAX           -0.0002223  0.0001147  -1.938 0.053413 .
## PTRATIO       -0.0211239  0.0048979  -4.313 2.05e-05 ***
## LB             0.0410730  0.0142521   2.882 0.004176 **
## LB_Sq         -0.0105399  0.0027000  -3.904 0.000112 ***
## LSTAT         -0.0283561  0.0020357 -13.929  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1561 on 383 degrees of freedom
## Multiple R-squared:  0.8502, Adjusted R-squared:  0.8444
## F-statistic:   145 on 15 and 383 DF,  p-value: < 2.2e-16
```

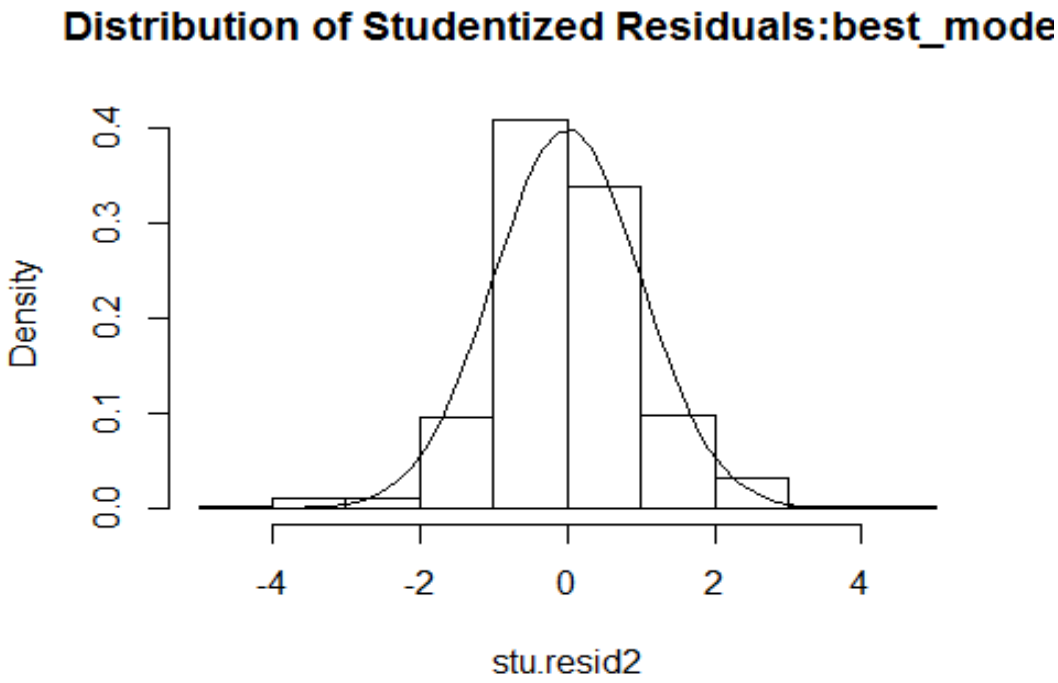
Notice that **LAGE** is part of our best model even though it doesn't have a significant-test. Why do you think this is?

It is because collectively these variables are important in explaining MEDV.

You can check that the predicted intervals are (slightly) tighter when the model includes AGE.


Pearson residua
-0.6 0.2
-4 0 2 4
LCRIM
Pearson residua
-0.6 0.2
0 5 15 25
LCRIM_Sq
Pearson residua
-0.6 0.2
0 40 80
ZN
Pearson residua
-0.6 0.2
0 1
factor(CHAS)
Pearson residua
-0.6 0.2
0.4 0.6 0.8
NOX
Pearson residua
-0.6 0.2
4 5 6 7 8
RM
Pearson residua
-0.6 0.2
20 40 60
RM_Sq
Pearson residua
-0.6 0.2
0 1 2 3 4
LAGE
Pearson residua
-0.6 0.2
0.5 1.5 2.5
LDIS
Pearson residua
-0.6 0.2
0.0 1.0 2.0 3.0
LRAD
Pearson residua
-0.6 0.2
200 400 600
TAX
Pearson residua
-0.6 0.2
14 18 22
PTRATIO
Pearson residua
-0.6 0.2
0 2 4 6
LB
Pearson residua
-0.6 0.2
0 10 20 30
LB_Sq
Pearson residua
-0.6 0.2
5 15 25 35
LSTAT


QQ Plot of residuals: best_model
Studentized Residuals(best_model)
t Quantiles
-4
-2
0
2
4
-3
-2
-1
0
1
2
3


Distribution of Studentized Residuals:best_mode
Density
stu.resid2
0.0
0.1
0.2
0.3
0.4
-4
-2
0
2
4

10. Validation

Remember at the onset we had kept aside 20% of our data to test our model on. We will now calculate our Predicted R^2 . This will tell us how well this model explains new data.

```
RM_Sq1<-(Boston_test$RM)^2
LRAD1<-log(Boston_test$RAD)
LB1<-(log(max(Boston_test$B)+1-Boston_test$B))
LCRIM1<-log(Boston_test$CRIM)
LDIS1<-log(Boston_test$DIS)
LCRIM_Sq1<-LCRIM1^2
LB_Sq1<-LB1^2
LAGE1<-(log(max(Boston_test$AGE)+1-Boston_test$AGE))
testData<-cbind(Boston_test,RM_Sq1,LRAD1,LB1,LCRIM1,LDIS1,LCRIM_Sq1,LB_Sq1,LAGE1)


model<-
lm(log(MEDV)~LCRIM1+LCRIM_Sq1+ZN+factor(CHAS)+NOX+RM+RM_Sq1+LAGE1+LDIS1+LRAD1+PTRATI
O+LB1+LB_Sq1+LSTAT)

y_hat<-predict.lm(model,newdata=testData, se.fit=TRUE)$fit
y_hat<-as.vector(y_hat)
dev<-log(testData$MEDV)-(y_hat)
num<-sum(dev^2)
dev1<-log(testData$MEDV)-mean(log(testData$MEDV))
den<-sum(dev1^2)
Predicted.Rsq<-1-(num/den)
Predicted.Rsq
```

```
## [1] 0.7887088
```

The predicted R^2= 79%. This can be considered as a very good fit.

Another useful Statistic to test the predictive power of our model is the PRESS statistic. Here is what Wikipedia says:

PRESS statistic

From Wikipedia, the free encyclopedia

In statistics, the **predicted residual sum of squares (PRESS) statistic** is a form of cross-validation used in regression analysis to provide a summary measure of the fit of a model to a sample of observations that were not themselves used to estimate the model. It is calculated as the sums of squares of the prediction residuals for those observations.[1][2][3]

A *fitted model* having been produced, each observation in turn is removed and the model is refitted using the remaining observations. The out-of-sample predicted value is calculated for the omitted observation in each case, and the PRESS statistic is calculated as the sum of the squares of all the resulting prediction errors:[4]

$$\mathrm{PRESS} = \sum_{i=1}^{n} (y_i - \hat{y}_{i,-i})^2$$

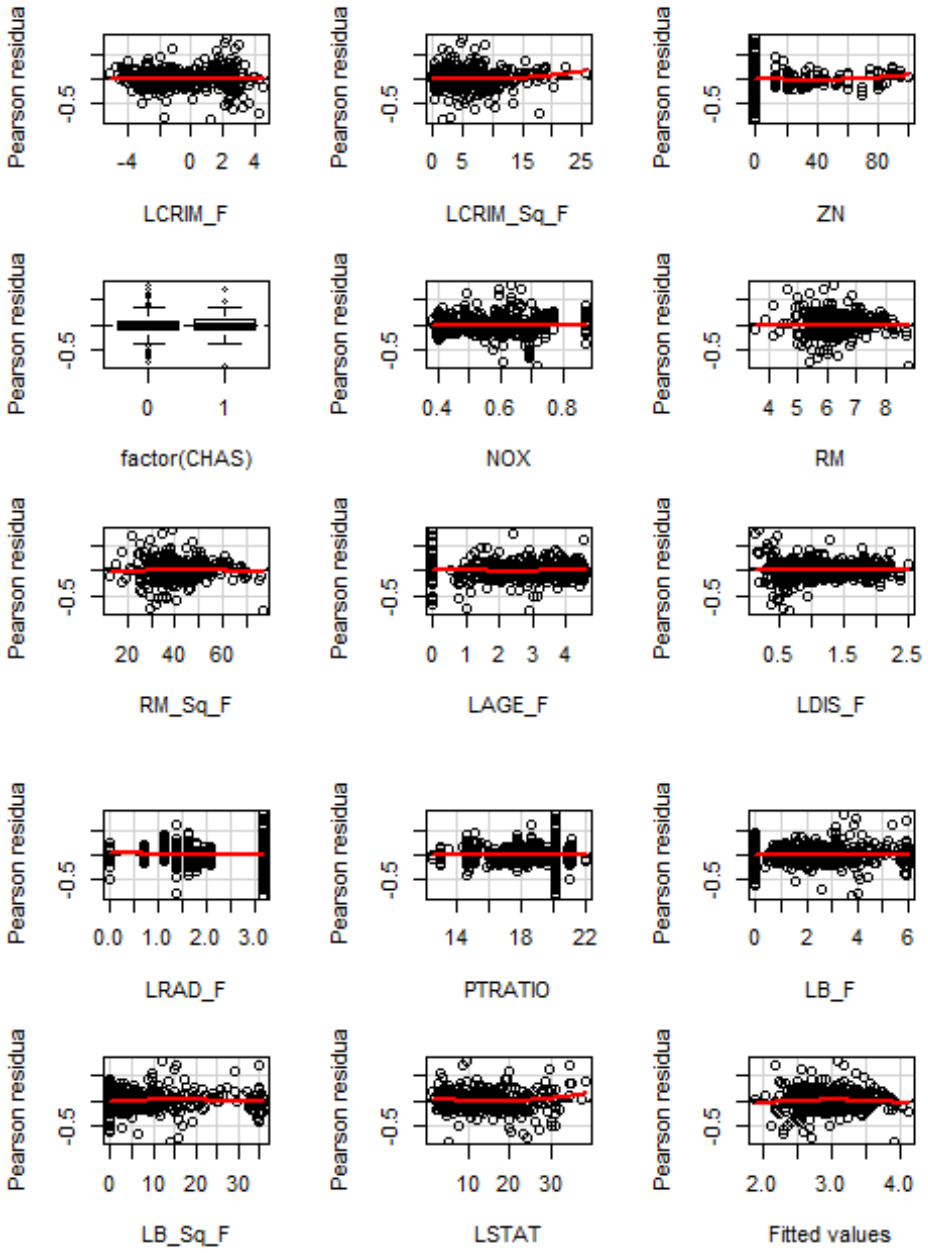
Given this procedure, the PRESS statistic can be calculated for a number of candidate model structures for the same dataset, with the lowest values of PRESS indicating the best structures. Models that are over-parameterised (over-fitted) would tend to give small residuals for observations included in the model-fitting but large residuals for observations that are excluded.

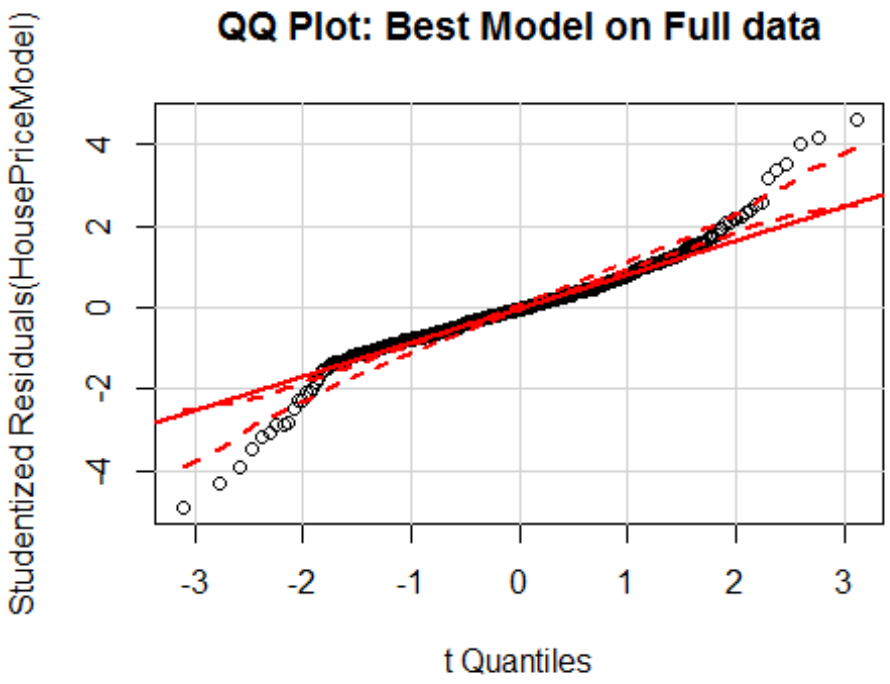
```
library(qpcR)

PRESS(best_model)$P.square

## [1] -2816.996
```

The PRESS statistics is -2816.996. By definition of the PRESS, we seem to have a very good model!!

11. Running the model on our original data. [Using the entire data(n=506)]

```
## Call:
## lm(formula = log(MEDV) ~ LCRIM_F + LCRIM_Sq_F + ZN + factor(CHAS) +
##     NOX + RM + RM_Sq_F + LAGE_F + LDIS_F + LRAD_F + PTRATIO +
##     LB_F + LB_Sq_F + LSTAT, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.81220 -0.09955 -0.00763  0.08504  0.80925
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    6.7303971  0.3948523  17.045  < 2e-16 ***
## LCRIM_F       -0.0499808  0.0109472  -4.566 6.30e-06 ***
## LCRIM_Sq_F    -0.0167637  0.0023358  -7.177 2.65e-12 ***
## ZN             0.0011053  0.0005371   2.058  0.04014 *
## factor(CHAS)1  0.0996734  0.0329752   3.023  0.00264 **
## NOX           -0.8189531  0.1508856  -5.428 8.99e-08 ***
## RM            -0.7854551  0.1186720  -6.619 9.51e-11 ***
## RM_Sq_F        0.0689114  0.0091752   7.511 2.80e-13 ***
## LAGE_F        -0.0003017  0.0107112  -0.028  0.97754
## LDIS_F        -0.1893771  0.0332612  -5.694 2.14e-08 ***
## LRAD_F         0.0622903  0.0188287   3.308  0.00101 **
## PTRATIO       -0.0303782  0.0048046  -6.323 5.78e-10 ***
## LB_F           0.0414762  0.0146478   2.832  0.00482 **
## LB_Sq_F       -0.0091681  0.0027993  -3.275  0.00113 **
## LSTAT         -0.0299831  0.0019569 -15.322  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1827 on 491 degrees of freedom
## Multiple R-squared:  0.8058, Adjusted R-squared:  0.8003
## F-statistic: 145.5 on 14 and 491 DF,  p-value: < 2.2e-16
```


Pearson residua
-0.5
-4 0 2 4
LCRIM_F
0 5 15 25
LCRIM_Sq_F
0 40 80
ZN
0 1
factor(CHAS)
0.4 0.6 0.8
NOX
4 5 6 7 8
RM
20 40 60
RM_Sq_F
0 1 2 3 4
LAGE_F
0.5 1.5 2.5
LDIS_F
0.0 1.0 2.0 3.0
LRAD_F
14 18 22
PTRATIO
0 2 4 6
LB_F
0 10 20 30
LB_Sq_F
10 20 30
LSTAT
2.0 3.0 4.0
Fitted values



Compare this output to the previous ones.

12. Interpretation

Now that we have our final model, what do we do with it? How do we interpret our estimated coefficiennts? Let us learn through a few examples.

Let us look at our regression output and then try to interpret the numbers.

```
HousePriceModel$coefficients
```

```
##   (Intercept)      LCRIM_F    LCRIM_Sq_F          ZN factor(CHAS)1
##  6.7303970537 -0.0499807907 -0.0167637444  0.0011052581  0.0996733951
##          NOX          RM       RM_Sq_F       LAGE_F       LDIS_F
## -0.8189530774 -0.7854551187  0.0689113690 -0.0003016607 -0.1893771352
##       LRAD_F      PTRATIO         LB_F      LB_Sq_F        LSTAT
##  0.0622903005 -0.0303782068  0.0414762333 -0.0091681060 -0.0299830947
```

I have a house in Boston, that has 6 rooms. I plan on (with permits) expanding my house by adding another room. By how much should I expect the price of my house to increase?

We have $\mathrm{RM} + (\mathrm{RM})^2$ is our model.

$$\Rightarrow \log(\mathrm{MEDV}) = -0.78 * \mathrm{RM} + 0.069 * \mathrm{RM}^2 + \in$$

Let us differentiate this w.r.t MEDV:

$$\left(\frac{1}{\mathrm{MEDV}}\right) = -0.78 * \left(\frac{\delta \mathrm{RM}}{\delta \mathrm{MEDV}}\right) + 2 * 0.069 * \mathrm{RM} * \left(\frac{\delta \mathrm{RM}}{\delta \mathrm{MEDV}}\right)$$

From this we see that:

- If I add another room, the price of my house will decrease **by -0.78*MEDV** (Where MEDV here is the current price of my house).
- The coefficient of RM^2 tells us how fast/slow the house price decreases as the number of rooms increase.
- This means that as the number of rooms increase the house price first diminishes and then rises.
- Visualize it like a **U** Shaped curve.

Try to answer this:
Recently there was an Air Quality Index study performed (say) in Boston. It was found that Nitric Oxides concentration was 1ppm more than last year's measurements. How will this affect the house prices in the area?

Someone comes to you and tells you that they want to estimate the price of a home they wish to buy in the Boston area.They are under a tight budget and cannot afford houses that cost more than $25,000. They have with them the following information:

CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	MEDV
0.027	0	7.07	0	0.469	6.42	78.9	4.96	2	242	17.8	396.90	9.14	21.6

What will you tell them?

We can put in this in formation in our model to get the Log House Price, along with their 95% Confidence Interval. This is

	fit	lwr	upr
1	3.03	2.97	3.089

You can now tell them that the expected median price of the house is approximately $20,000.

```
exp(3.030626)#Since we used LOG(MEDV)

## [1] 20.71019
```

13. Conclusion:

The analysis is not over. We were able to fit the best model with the constraint of a particular dataset. Our R^2 was around 80%, which means there is a lot left to be explained. Going forward one can try to get data on other variables that might have an impact on the House Prices., for example **Mortgage availability.** On the other hand, we should not include variables only to achieve a high R^2, i.e. we must not *over fit* our model.