Boston Housing Price Prediction - Regression

By Vishwa Pardeshi

In this notebook, we will predict the price of houses in Boston region using the ever famous Boston Housing Price Dataset.

Learning Outcome: By following the notebook you will be able to

1. Perform context inspired EDA to understand relationship between predictor variables and medv (the median house value)
2. Implement & infer Simple & Multiple Linear Regression
3. Perform feature selection using forward and backward stepwise selection
4. Evaluate statistical assumptions of linear regression models

Setup

In this problem set you will need, at minimum, the following R packages.

```
# Load standard libraries
library(tidyverse)
library(MASS) # Modern applied statistics functions
library(corrplot)
```

Exploratory Data Analysis

Housing Values in Suburbs of Boston

In this problem we will use the Boston dataset that is available in the `MASS` package. This dataset contains information about median house value for 506 neighborhoods in Boston, MA. Load this data and use it to answer the following questions.

```
#loading housing price data in variable Boston
data("Boston")
```

1. Describe the data and variables that are part of the `Boston` dataset. Tidy data as necessary.

```
#checking dimensions, columns and snippet of data
dim(Boston)
```

```
## [1] 506  14
```

```
colnames(Boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "black"   "lstat"   "medv"
```

```r
summary(Boston)
```

```
##       crim                zn              indus            chas        
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917  
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000  
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000  
##       nox               rm              age              dis        
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130  
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100  
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207  
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795  
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188  
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127  
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      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
```

```r
head(Boston)
```

```
##      crim zn indus chas   nox    rm  age    dis rad tax ptratio  black
## 1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3 396.90
## 2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8 396.90
## 3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8 392.83
## 4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7 394.63
## 5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7 396.90
## 6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7 394.12
##   lstat medv
## 1  4.98 24.0
## 2  9.14 21.6
## 3  4.03 34.7
## 4  2.94 33.4
## 5  5.33 36.2
## 6  5.21 28.7
```

```r
#checking for missing and duplicate values in Boston respectively
sum(is.na(Boston))
```

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

```r
sum(duplicated(Boston))
```

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

There are 506 rows of 14 variables which are as follows:

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)ˆ2 where Bk is the proportion of blacks by town.

lstat - lower status of the population (percent).

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

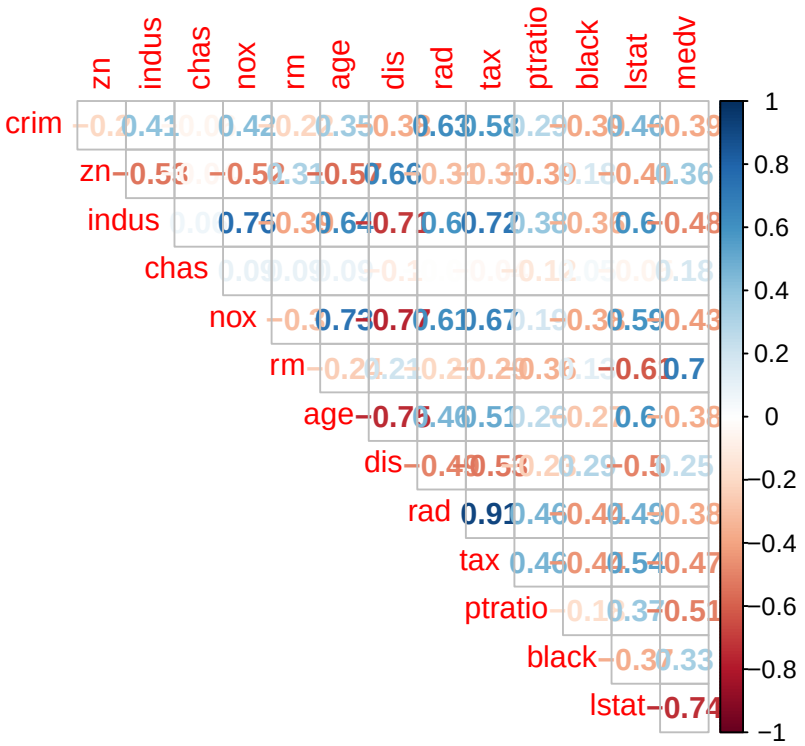
Since,there are no missing and duplicated value no data cleaning is required. The dataset is already tidy.

2. Consider this data in context, what is the response variable of interest?

   In this data context, the response variable of interest is the price of the house, medv.

```r
corr_matrix<-cor(Boston)
corrplot(corr_matrix,method = "number", type="upper", diag = FALSE, main = "Figure 1: Corrplot for
```

Figure 1: Corrplot for Boston Data

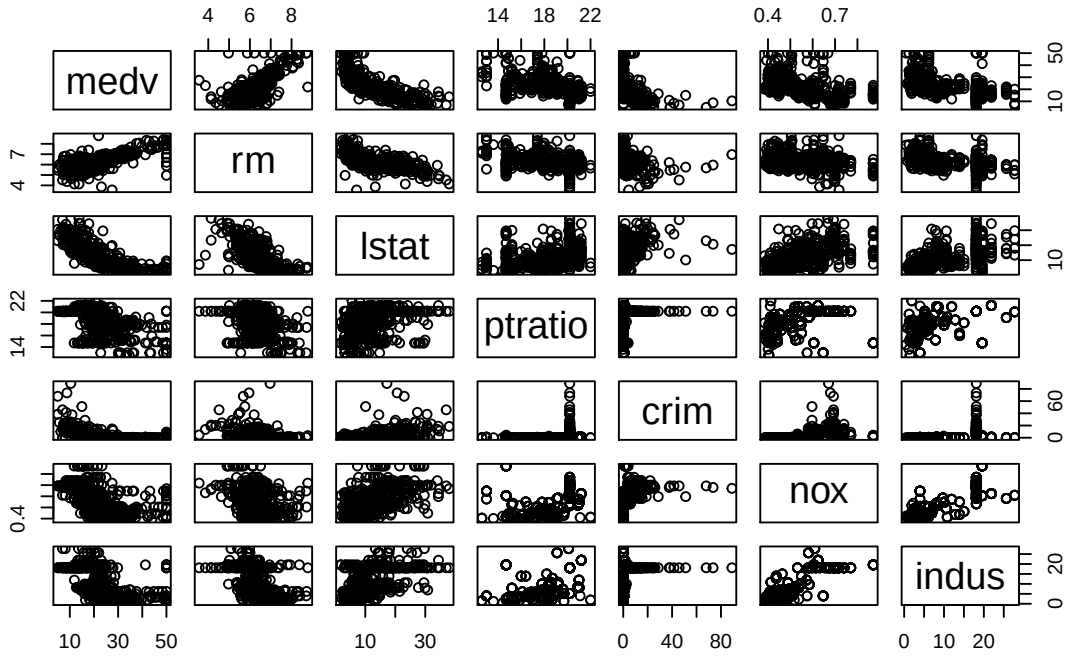


From Figure 1, we notice that there is a strong correlation between the response variable and the predictors - lstat and rm.

Additionally, it can be noticed from figure 1 that medv decreases with increase in crim (medium), indus (medium), nox(medium), age(low), rad(low), tax(medium), ptratio(high), lstat(high) and increase with chas(low),zn(low), rm (high), dis(low), black(low). The weakest correlation is with chas.

```
pairs(~ medv + rm + lstat + ptratio  + crim + nox  + indus , data = Boston, main = "Figure 2: Scatt
```

Figure 2: Scatterplot for Boston Data



Thus, we deduce from Figure 2 that there is a strong linear relationship between medv and rm. On the other hand, there is a non-linear relationship between medv and lstat.

Simple Linear Regression

3. For each predictor, fit a simple linear regression model to predict the response. In which of the models is there a statistically significant association between the predictor and the response? Create some plots to back up your assertions.

```
#vector for predictor, response and variables
variables <- c(colnames(Boston))
response <- "medv"
predictor <- variables[!(variables %in% response)]
coefficientLinear <- c()
cat("\nLinear Regression for response variable medv.")
```

```
##
## Linear Regression for response variable medv.
```

```
for(var in predictor){
  cat("\n\nFor predictor variable", var, ":\nThe estimated coefficient and p-value: \n")
  model <- lm(medv ~ Boston[,var], data = Boston)
  summaryModel <- summary(model)
  printCoefmat(coef(summary(model)))
  coefficientLinear[length(coefficientLinear) + 1] <- as.numeric(summaryModel$coefficients[2,1])
  if(summaryModel$coefficients[2,4] < 0.05){
```

```
    cat("At a signifance level of 0.05, the predictor variable", var, "is statistically significant
  }
  else{
    cat("At a signifance level of 0.05, the predictor variable", var, "is not statistically signifi
  }
}
```

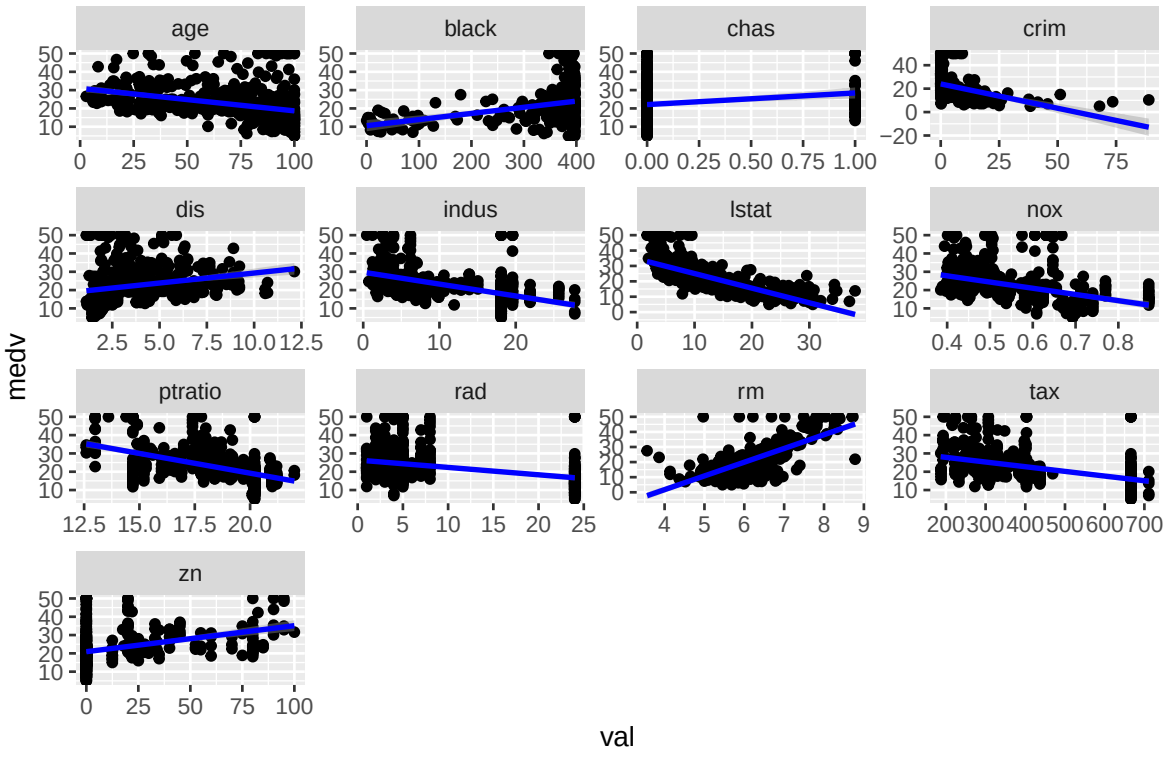
```
##
##
## For predictor variable crim :
## The estimated coefficient and p-value:
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   24.03311    0.40914 58.7403 < 2.2e-16 ***
## Boston[, var] -0.41519    0.04389 -9.4597 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable crim is statistically significant.
##
## For predictor variable zn :
## The estimated coefficient and p-value:
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   20.917579   0.424741 49.2479 < 2.2e-16 ***
## Boston[, var]  0.142140   0.016385  8.6751 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable zn is statistically significant.
##
## For predictor variable indus :
## The estimated coefficient and p-value:
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   29.754897   0.683445  43.537 < 2.2e-16 ***
## Boston[, var] -0.648490   0.052264 -12.408 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable indus is statistically significant.
##
## For predictor variable chas :
## The estimated coefficient and p-value:
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   22.09384    0.41763 52.9023 < 2.2e-16 ***
## Boston[, var]  6.34616    1.58795  3.9964 7.391e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable chas is statistically significant.
##
## For predictor variable nox :
## The estimated coefficient and p-value:
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)    41.3459     1.8112  22.828 < 2.2e-16 ***
## Boston[, var] -33.9161     3.1963 -10.611 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable nox is statistically significant.
##
```

```
## For predictor variable rm :
## The estimated coefficient and p-value:
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   -34.67062    2.64980 -13.084 < 2.2e-16 ***
## Boston[, var]   9.10211    0.41903  21.722 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable rm is statistically significant.
##
## For predictor variable age :
## The estimated coefficient and p-value:
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   30.97868    0.99911 31.0064 < 2.2e-16 ***
## Boston[, var] -0.12316    0.01348 -9.1366 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable age is statistically significant.
##
## For predictor variable dis :
## The estimated coefficient and p-value:
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   18.39009    0.81739 22.4986 < 2.2e-16 ***
## Boston[, var]  1.09161    0.18838  5.7948 1.207e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable dis is statistically significant.
##
## For predictor variable rad :
## The estimated coefficient and p-value:
##                Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   26.382128   0.561757  46.964 < 2.2e-16 ***
## Boston[, var] -0.403095   0.043489  -9.269 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable rad is statistically significant.
##
## For predictor variable tax :
## The estimated coefficient and p-value:
##                 Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   32.9706545  0.9482956  34.768 < 2.2e-16 ***
## Boston[, var] -0.0255681  0.0021474 -11.906 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable tax is statistically significant.
##
## For predictor variable ptratio :
## The estimated coefficient and p-value:
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   62.34463    3.02917  20.581 < 2.2e-16 ***
## Boston[, var] -2.15718    0.16302 -13.233 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable ptratio is statistically significant.
##
```

```
## For predictor variable black :
## The estimated coefficient and p-value:
##                  Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   10.5510341  1.5574626  6.7745 3.492e-11 ***
## Boston[, var]  0.0335931  0.0042305  7.9407 1.318e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable black is statistically significant.
##
## For predictor variable lstat :
## The estimated coefficient and p-value:
##                 Estimate Std. Error t value  Pr(>|t|)
## (Intercept)   34.553841   0.562627  61.415 < 2.2e-16 ***
## Boston[, var] -0.950049   0.038733 -24.528 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## At a signifance level of 0.05, the predictor variable lstat is statistically significant.
```

```r
#plots for each predictors
Boston %>%
  gather(key, val, -medv) %>%
  ggplot(aes(x = val, y = medv)) +
  geom_point() +
  stat_smooth(method = "lm", se = TRUE, col = "blue") +
  facet_wrap(~key, scales = "free") +
  theme_gray() +
  ggtitle("Figure 3: Scatter plot of predictor variables vs Median Value (medv)")
```

Figure 3: Scatter plot of predictor variables vs Median Value (medv)



Even though it is observed that all the models created above have statistically significant coefficient estimates at 0.05 alpha level, Figure 3 paints a different picture in terms of the nature of relationship.

Though the regression line fitted for each predictor response pair has a line that has statistically significant coefficients, this doesn't necessarily imply a linear relationship between the predictor and the variable

Multiple Linear Regression

4. Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis $H_0 : \beta_j = 0$?

```
#multiple regression
multiModel <- lm(formula = medv ~ ., data = Boston)
summary(multiModel)
```

```
##
## Call:
## lm(formula = medv ~ ., data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.595  -2.730  -0.518   1.777  26.199
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
## crim        -1.080e-01  3.286e-02  -3.287 0.001087 **
## zn           4.642e-02  1.373e-02   3.382 0.000778 ***
## indus        2.056e-02  6.150e-02   0.334 0.738288
## chas         2.687e+00  8.616e-01   3.118 0.001925 **
## nox         -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
## rm           3.810e+00  4.179e-01   9.116  < 2e-16 ***
## age          6.922e-04  1.321e-02   0.052 0.958229
## dis         -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
## rad          3.060e-01  6.635e-02   4.613 5.07e-06 ***
## tax         -1.233e-02  3.760e-03  -3.280 0.001112 **
## ptratio     -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
## black        9.312e-03  2.686e-03   3.467 0.000573 ***
## lstat       -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.745 on 492 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7338
## F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

```
coefficientMultiple <- summary(multiModel)$coefficients[-1,1]
coefficient <- cbind(coefficientLinear, coefficientMultiple)
```

On observing the coefficients and their respective p-value, we notice indus and age's p value which makes it statistically insignificant at an alpha level of 0.05.

At a statistial significance level of 0.05, all predictor variables except indus and age have statistically significant coefficient estimates thus the null hypothesis can be rejected for these predictor variables.

Thus, multiple regression line will contain all predictor variables except indus and age.

5. How do your results from (3) compare to your results from (4)? Use visualization to support your response.

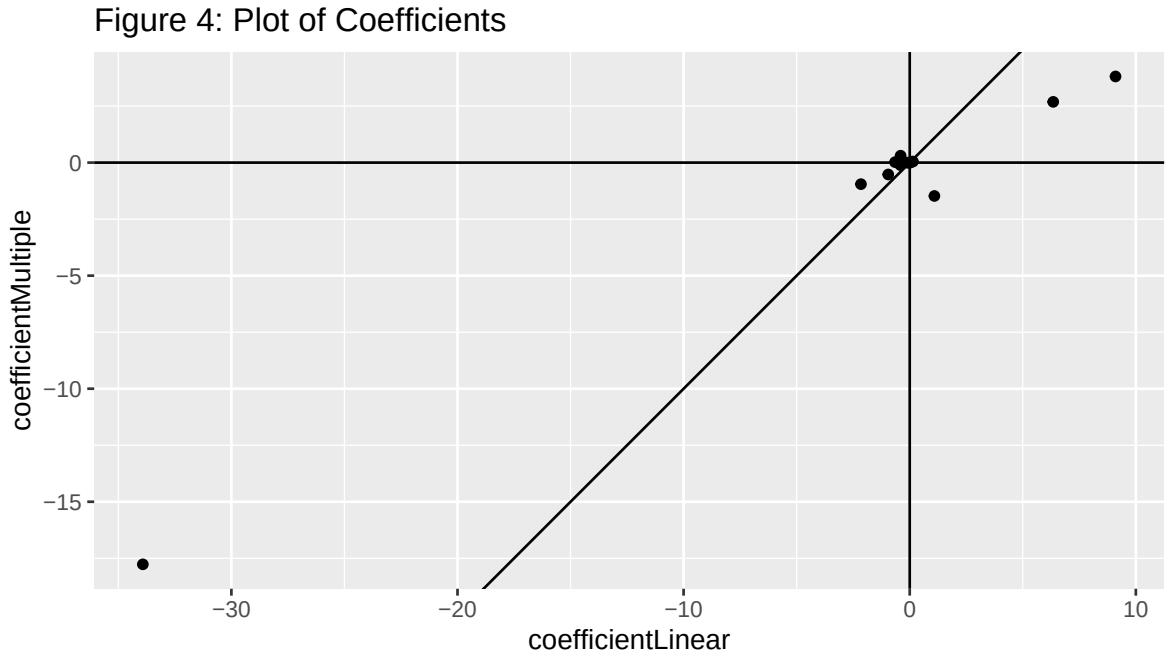
   The results from (3) and (4) are compared using the plot. If each of the predictor variable is independent of the othe predictor variable then its influence or extent of association with the response variable should be almost similar in both - single and multiple regression model. If the coefficients are similar for both multiple and single regression, they will lie on the line passing through origin with a slope of 1.

   However, as is observed from previous results i.e. of correlation matrix, we know that a few predictor variables have a strong correlation and hence as a result this will be reflected in the Figure 4.

```
#plot for coefficients

ggplot(as.data.frame(coefficient), aes(x=coefficientLinear, y=coefficientMultiple)) +
  geom_point() +
  coord_fixed() +
  geom_vline(xintercept = 0) + geom_hline(yintercept = 0) +
  geom_abline(a = 0, b = 1)  +
  ggtitle("Figure 4: Plot of Coefficients")
```

```
## Warning: Ignoring unknown parameters: a, b
```



```
#plot(x=coefficientLinear, y=coefficientMultiple)
#identify(x=coefficientLinear, y=coefficientMultiple, labels = row.names(coefficient))
```

Figure 4 demonstrates how most coefficients in Multiple and single model are pretty close to one another, however a few are a little far off possibly due to other hidden relationships which we have not explored.

Non-linear transformation

6. Is there evidence of a non-linear association between any of the predictors and the response? To answer this question, for each predictor X fit a model of the form:

$$Y = \beta_0 + \beta_1 X + \beta_2 X^2 + \beta_3 X^3 + \epsilon$$

Observation of figure 2 showed us that there are indeed some non-linear associations between predictors and responses.

```
#evaluate the non-linear model for each predictor variable
for(var in predictor){
  cat("\n\nFor predictor variable", var, ":\nThe estimated coefficient and p-value: \n")
  model <- lm(medv ~ poly(Boston[,var],3,  raw = TRUE), data = Boston)
  summaryModel <- coef(summary(model))
  printCoefmat(summaryModel)

}
```

```
##
##
## For predictor variable crim :
## The estimated coefficient and p-value:
##                                      Estimate  Std. Error t value
## (Intercept)                         2.5190e+01  4.3548e-01 57.8456
## poly(Boston[, var], 3, raw = TRUE)1 -1.1364e+00  1.4444e-01 -7.8676
## poly(Boston[, var], 3, raw = TRUE)2  2.3785e-02  6.8079e-03  3.4937
## poly(Boston[, var], 3, raw = TRUE)3 -1.4887e-04  6.6408e-05 -2.2418
##                                      Pr(>|t|)
## (Intercept)                         < 2.2e-16 ***
## poly(Boston[, var], 3, raw = TRUE)1 2.235e-14 ***
## poly(Boston[, var], 3, raw = TRUE)2 0.0005184 ***
## poly(Boston[, var], 3, raw = TRUE)3 0.0254110 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable zn :
## The estimated coefficient and p-value:
##                                      Estimate  Std. Error t value
## (Intercept)                         2.0449e+01  4.3595e-01 46.9054
## poly(Boston[, var], 3, raw = TRUE)1  6.4337e-01  1.1056e-01  5.8191
## poly(Boston[, var], 3, raw = TRUE)2 -1.6765e-02  3.8872e-03 -4.3128
## poly(Boston[, var], 3, raw = TRUE)3  1.2570e-04  3.1601e-05  3.9777
##                                      Pr(>|t|)
```

```
## (Intercept)                           < 2.2e-16 ***
## poly(Boston[, var], 3, raw = TRUE)1 1.056e-08 ***
## poly(Boston[, var], 3, raw = TRUE)2 1.942e-05 ***
## poly(Boston[, var], 3, raw = TRUE)3 7.981e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable indus :
## The estimated coefficient and p-value:
##                                       Estimate Std. Error t value
## (Intercept)                          37.0801601  1.6633262 22.2928
## poly(Boston[, var], 3, raw = TRUE)1 -2.8069941  0.5093489 -5.5109
## poly(Boston[, var], 3, raw = TRUE)2  0.1404617  0.0415541  3.3802
## poly(Boston[, var], 3, raw = TRUE)3 -0.0023989  0.0010110 -2.3729
##                                      Pr(>|t|)
## (Intercept)                          < 2.2e-16 ***
## poly(Boston[, var], 3, raw = TRUE)1 5.714e-08 ***
## poly(Boston[, var], 3, raw = TRUE)2 0.0007807 ***
## poly(Boston[, var], 3, raw = TRUE)3 0.0180257 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable chas :
## The estimated coefficient and p-value:
##                                      Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                          22.09384    0.41763 52.9023 < 2.2e-16
## poly(Boston[, var], 3, raw = TRUE)1  6.34616    1.58795  3.9964 7.391e-05
##
## (Intercept)                          ***
## poly(Boston[, var], 3, raw = TRUE)1 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable nox :
## The estimated coefficient and p-value:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           -22.486     38.520 -0.5838  0.55965
## poly(Boston[, var], 3, raw = TRUE)1  315.096    195.100  1.6150  0.10693
## poly(Boston[, var], 3, raw = TRUE)2 -615.827    320.476 -1.9216  0.05522 .
## poly(Boston[, var], 3, raw = TRUE)3  350.186    170.923  2.0488  0.04100 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable rm :
## The estimated coefficient and p-value:
##                                       Estimate Std. Error t value
## (Intercept)                          241.31081   47.32746  5.0987
## poly(Boston[, var], 3, raw = TRUE)1 -109.39061   22.96895 -4.7625
## poly(Boston[, var], 3, raw = TRUE)2   16.49102    3.67505  4.4873
## poly(Boston[, var], 3, raw = TRUE)3   -0.74039    0.19348 -3.8268
```

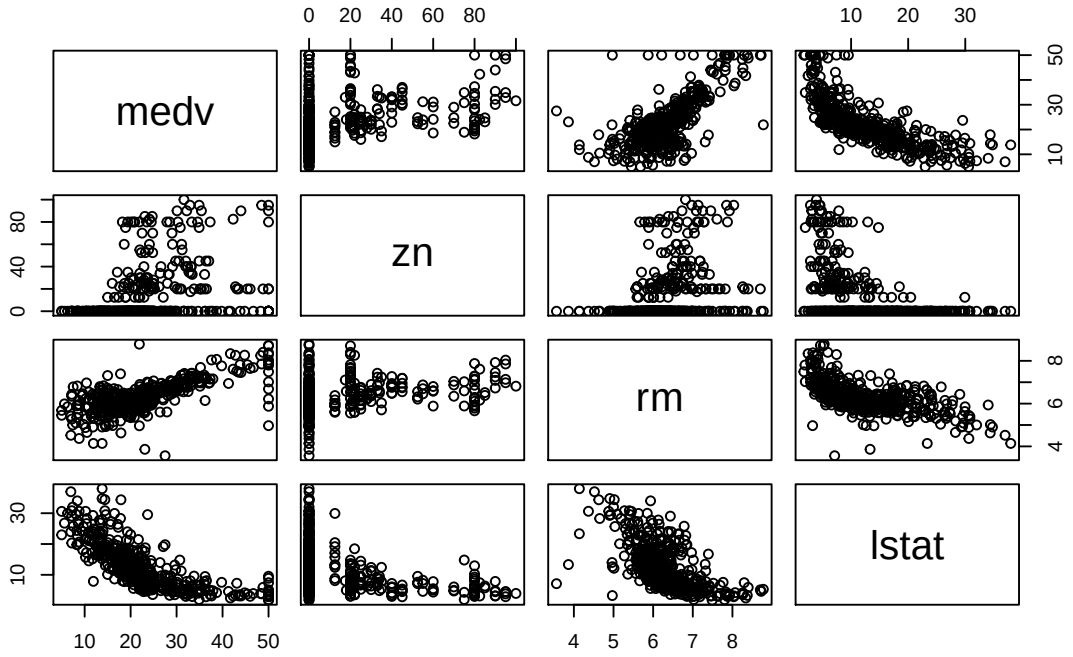
```
##                                      Pr(>|t|)
## (Intercept)                         4.853e-07 ***
## poly(Boston[, var], 3, raw = TRUE)1 2.505e-06 ***
## poly(Boston[, var], 3, raw = TRUE)2 8.952e-06 ***
## poly(Boston[, var], 3, raw = TRUE)3 0.0001462 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable age :
## The estimated coefficient and p-value:
##                                       Estimate  Std. Error t value
## (Intercept)                          2.8931e+01  2.9924e+00  9.6683
## poly(Boston[, var], 3, raw = TRUE)1 -1.2242e-01  2.0140e-01 -0.6078
## poly(Boston[, var], 3, raw = TRUE)2  2.3546e-03  3.9302e-03  0.5991
## poly(Boston[, var], 3, raw = TRUE)3 -2.3179e-05  2.2794e-05 -1.0169
##                                      Pr(>|t|)
## (Intercept)                            <2e-16 ***
## poly(Boston[, var], 3, raw = TRUE)1   0.5436
## poly(Boston[, var], 3, raw = TRUE)2   0.5494
## poly(Boston[, var], 3, raw = TRUE)3   0.3097
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable dis :
## The estimated coefficient and p-value:
##                                      Estimate Std. Error t value  Pr(>|t|)
## (Intercept)                          7.037894   2.911336  2.4174  0.015987
## poly(Boston[, var], 3, raw = TRUE)1  8.592844   2.066334  4.1585 3.768e-05
## poly(Boston[, var], 3, raw = TRUE)2 -1.249528   0.412345 -3.0303  0.002569
## poly(Boston[, var], 3, raw = TRUE)3  0.056019   0.024283  2.3070  0.021463
##
## (Intercept)                         *
## poly(Boston[, var], 3, raw = TRUE)1 ***
## poly(Boston[, var], 3, raw = TRUE)2 **
## poly(Boston[, var], 3, raw = TRUE)3 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable rad :
## The estimated coefficient and p-value:
##                                       Estimate Std. Error t value
## (Intercept)                         30.2513027  2.5678599 11.7807
## poly(Boston[, var], 3, raw = TRUE)1 -3.7994539  1.3071558 -2.9067
## poly(Boston[, var], 3, raw = TRUE)2  0.6163466  0.1860574  3.3127
## poly(Boston[, var], 3, raw = TRUE)3 -0.0200864  0.0057166 -3.5137
##                                      Pr(>|t|)
## (Intercept)                         < 2.2e-16 ***
## poly(Boston[, var], 3, raw = TRUE)1 0.0038146 **
## poly(Boston[, var], 3, raw = TRUE)2 0.0009908 ***
## poly(Boston[, var], 3, raw = TRUE)3 0.0004819 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable tax :
## The estimated coefficient and p-value:
##                                     Estimate  Std. Error t value
## (Intercept)                         5.2216e+01  1.3966e+01  3.7387
## poly(Boston[, var], 3, raw = TRUE)1 -1.6347e-01  1.1329e-01 -1.4430
## poly(Boston[, var], 3, raw = TRUE)2  3.0293e-04  2.8718e-04  1.0548
## poly(Boston[, var], 3, raw = TRUE)3 -2.0787e-07  2.2363e-07 -0.9295
##                                     Pr(>|t|)
## (Intercept)                         0.0002062 ***
## poly(Boston[, var], 3, raw = TRUE)1 0.1496461
## poly(Boston[, var], 3, raw = TRUE)2 0.2920043
## poly(Boston[, var], 3, raw = TRUE)3 0.3530609
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable ptratio :
## The estimated coefficient and p-value:
##                                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         312.286417 152.486930  2.0480  0.04108
## poly(Boston[, var], 3, raw = TRUE)1 -48.691136  26.884407 -1.8111  0.07072
## poly(Boston[, var], 3, raw = TRUE)2   2.839951   1.564131  1.8157  0.07002
## poly(Boston[, var], 3, raw = TRUE)3  -0.056865   0.030049 -1.8924  0.05901
##
## (Intercept)                         *
## poly(Boston[, var], 3, raw = TRUE)1 .
## poly(Boston[, var], 3, raw = TRUE)2 .
## poly(Boston[, var], 3, raw = TRUE)3 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable black :
## The estimated coefficient and p-value:
##                                     Estimate  Std. Error t value
## (Intercept)                         1.2598e+01  2.5166e+00  5.0061
## poly(Boston[, var], 3, raw = TRUE)1 -1.7033e-02  6.1500e-02 -0.2770
## poly(Boston[, var], 3, raw = TRUE)2  2.0361e-04  3.2582e-04  0.6249
## poly(Boston[, var], 3, raw = TRUE)3 -2.2243e-07  4.7650e-07 -0.4668
##                                     Pr(>|t|)
## (Intercept)                         7.701e-07 ***
## poly(Boston[, var], 3, raw = TRUE)1    0.7819
## poly(Boston[, var], 3, raw = TRUE)2    0.5323
## poly(Boston[, var], 3, raw = TRUE)3    0.6409
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## For predictor variable lstat :
## The estimated coefficient and p-value:
##                                      Estimate Std. Error  t value
```

```
## (Intercept)                            48.6496253  1.4347240  33.9087
## poly(Boston[, var], 3, raw = TRUE)1 -3.8655928  0.3287861 -11.7572
## poly(Boston[, var], 3, raw = TRUE)2  0.1487385  0.0212987   6.9834
## poly(Boston[, var], 3, raw = TRUE)3 -0.0020039  0.0003997  -5.0134
##                                         Pr(>|t|)
## (Intercept)                            < 2.2e-16 ***
## poly(Boston[, var], 3, raw = TRUE)1 < 2.2e-16 ***
## poly(Boston[, var], 3, raw = TRUE)2 9.178e-12 ***
## poly(Boston[, var], 3, raw = TRUE)3 7.428e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
pairs(~ medv + zn + rm + lstat, data = Boston, main = "Figure 5: Exploring non-linear relationships
```

Figure 5: Exploring non−linear relationships



Thus, the most significant association is noticed for zn, rm and lstat.

As can be observed from the above figure 5, the relationship indeed is non linear for lstat.

Feature Selection - Stepwise Forward & Backward Selection

7. Consider performing a stepwise model selection procedure to determine the bets fit model. Discuss your results. How is this model different from the model in (4)?

   Here, I will perform both forward and backward stepwise model selection procedure to observe the AIC of both. Consequently, I will compare this with the AIC value of the multiple regression model (4).

```r
#Variable selection using stepwise regression
nullmodel <- lm(medv ~ 1, data = Boston)
fullmodel <- lm(medv ~ ., data = Boston)

#stepwise selection

#backward
modelBack <- step(fullmodel, direction = "backward")
```

```
## Start:  AIC=1589.64
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
##     tax + ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - age      1      0.06 11079 1587.7
## - indus    1      2.52 11081 1587.8
## <none>                 11079 1589.6
## - chas     1    218.97 11298 1597.5
## - tax      1    242.26 11321 1598.6
## - crim     1    243.22 11322 1598.6
## - zn       1    257.49 11336 1599.3
## - black    1    270.63 11349 1599.8
## - rad      1    479.15 11558 1609.1
## - nox      1    487.16 11566 1609.4
## - ptratio  1   1194.23 12273 1639.4
## - dis      1   1232.41 12311 1641.0
## - rm       1   1871.32 12950 1666.6
## - lstat    1   2410.84 13490 1687.3
##
## Step:  AIC=1587.65
## medv ~ crim + zn + indus + chas + nox + rm + dis + rad + tax +
##     ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - indus    1      2.52 11081 1585.8
## <none>                 11079 1587.7
## - chas     1    219.91 11299 1595.6
## - tax      1    242.24 11321 1596.6
## - crim     1    243.20 11322 1596.6
## - zn       1    260.32 11339 1597.4
## - black    1    272.26 11351 1597.9
## - rad      1    481.09 11560 1607.2
## - nox      1    520.87 11600 1608.9
## - ptratio  1   1200.23 12279 1637.7
## - dis      1   1352.26 12431 1643.9
## - rm       1   1959.55 13038 1668.0
## - lstat    1   2718.88 13798 1696.7
##
## Step:  AIC=1585.76
## medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##
##           Df Sum of Sq   RSS    AIC
## <none>                 11081 1585.8
```

```
## - chas     1    227.21 11309 1594.0
## - crim     1    245.37 11327 1594.8
## - zn       1    257.82 11339 1595.4
## - black    1    270.82 11352 1596.0
## - tax      1    273.62 11355 1596.1
## - rad      1    500.92 11582 1606.1
## - nox      1    541.91 11623 1607.9
## - ptratio  1   1206.45 12288 1636.0
## - dis      1   1448.94 12530 1645.9
## - rm       1   1963.66 13045 1666.3
## - lstat    1   2723.48 13805 1695.0
```

```
#forward
modelFront<- step(nullmodel, scope = list(upper=fullmodel,lower=nullmodel), direction = "forward")
```

```
## Start:  AIC=2246.51
## medv ~ 1
##
##           Df Sum of Sq   RSS    AIC
## + lstat    1   23243.9 19472 1851.0
## + rm       1   20654.4 22062 1914.2
## + ptratio  1   11014.3 31702 2097.6
## + indus    1    9995.2 32721 2113.6
## + tax      1    9377.3 33339 2123.1
## + nox      1    7800.1 34916 2146.5
## + crim     1    6440.8 36276 2165.8
## + rad      1    6221.1 36495 2168.9
## + age      1    6069.8 36647 2171.0
## + zn       1    5549.7 37167 2178.1
## + black    1    4749.9 37966 2188.9
## + dis      1    2668.2 40048 2215.9
## + chas     1    1312.1 41404 2232.7
## <none>                 42716 2246.5
##
## Step:  AIC=1851.01
## medv ~ lstat
##
##           Df Sum of Sq   RSS    AIC
## + rm       1    4033.1 15439 1735.6
## + ptratio  1    2670.1 16802 1778.4
## + chas     1     786.3 18686 1832.2
## + dis      1     772.4 18700 1832.5
## + age      1     304.3 19168 1845.0
## + tax      1     274.4 19198 1845.8
## + black    1     198.3 19274 1847.8
## + zn       1     160.3 19312 1848.8
## + crim     1     146.9 19325 1849.2
## + indus    1      98.7 19374 1850.4
## <none>                 19472 1851.0
## + rad      1      25.1 19447 1852.4
## + nox      1       4.8 19468 1852.9
##
## Step:  AIC=1735.58
## medv ~ lstat + rm
```

```
##
##           Df Sum of Sq   RSS    AIC
## + ptratio  1   1711.32 13728 1678.1
## + chas     1    548.53 14891 1719.3
## + black    1    512.31 14927 1720.5
## + tax      1    425.16 15014 1723.5
## + dis      1    351.15 15088 1725.9
## + crim     1    311.42 15128 1727.3
## + rad      1    180.45 15259 1731.6
## + indus    1     61.09 15378 1735.6
## <none>                 15439 1735.6
## + zn       1     56.56 15383 1735.7
## + age      1     20.18 15419 1736.9
## + nox      1     14.90 15424 1737.1
##
## Step:  AIC=1678.13
## medv ~ lstat + rm + ptratio
##
##         Df Sum of Sq   RSS    AIC
## + dis    1    499.08 13229 1661.4
## + black  1    389.68 13338 1665.6
## + chas   1    377.96 13350 1666.0
## + crim   1    122.52 13606 1675.6
## + age    1     66.24 13662 1677.7
## <none>               13728 1678.1
## + tax    1     44.36 13684 1678.5
## + nox    1     24.81 13703 1679.2
## + zn     1     14.96 13713 1679.6
## + rad    1      6.07 13722 1679.9
## + indus  1      0.83 13727 1680.1
##
## Step:  AIC=1661.39
## medv ~ lstat + rm + ptratio + dis
##
##         Df Sum of Sq   RSS    AIC
## + nox    1    759.56 12469 1633.5
## + black  1    502.64 12726 1643.8
## + chas   1    267.43 12962 1653.1
## + indus  1    242.65 12986 1654.0
## + tax    1    240.34 12989 1654.1
## + crim   1    233.54 12995 1654.4
## + zn     1    144.81 13084 1657.8
## + age    1     61.36 13168 1661.0
## <none>               13229 1661.4
## + rad    1     22.40 13206 1662.5
##
## Step:  AIC=1633.47
## medv ~ lstat + rm + ptratio + dis + nox
##
##         Df Sum of Sq   RSS    AIC
## + chas   1    328.27 12141 1622.0
## + black  1    311.83 12158 1622.7
## + zn     1    151.71 12318 1629.3
## + crim   1    141.43 12328 1629.7
```

```
## + rad    1     53.48 12416 1633.3
## <none>               12469 1633.5
## + indus  1     17.10 12452 1634.8
## + tax    1     10.50 12459 1635.0
## + age    1      0.25 12469 1635.5
##
## Step:  AIC=1621.97
## medv ~ lstat + rm + ptratio + dis + nox + chas
##
##         Df Sum of Sq   RSS    AIC
## + black  1   272.837 11868 1612.5
## + zn     1   164.406 11977 1617.1
## + crim   1   116.330 12025 1619.1
## + rad    1    58.556 12082 1621.5
## <none>               12141 1622.0
## + indus  1    26.274 12115 1622.9
## + tax    1     4.187 12137 1623.8
## + age    1     2.331 12139 1623.9
##
## Step:  AIC=1612.47
## medv ~ lstat + rm + ptratio + dis + nox + chas + black
##
##         Df Sum of Sq   RSS    AIC
## + zn     1   189.936 11678 1606.3
## + rad    1   144.320 11724 1608.3
## + crim   1    55.633 11813 1612.1
## <none>               11868 1612.5
## + indus  1    15.584 11853 1613.8
## + age    1     9.446 11859 1614.1
## + tax    1     2.703 11866 1614.4
##
## Step:  AIC=1606.31
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn
##
##         Df Sum of Sq   RSS    AIC
## + crim   1    94.712 11584 1604.2
## + rad    1    93.614 11585 1604.2
## <none>               11678 1606.3
## + indus  1    16.048 11662 1607.6
## + tax    1     3.952 11674 1608.1
## + age    1     1.491 11677 1608.2
##
## Step:  AIC=1604.19
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn +
##     crim
##
##         Df Sum of Sq   RSS    AIC
## + rad    1   228.604 11355 1596.1
## <none>               11584 1604.2
## + indus  1    15.773 11568 1605.5
## + age    1     2.470 11581 1606.1
## + tax    1     1.305 11582 1606.1
##
## Step:  AIC=1596.1
```

```
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn +
##     crim + rad
##
##         Df Sum of Sq   RSS    AIC
## + tax    1   273.619 11081 1585.8
## <none>               11355 1596.1
## + indus  1    33.894 11321 1596.6
## + age    1     0.096 11355 1598.1
##
## Step:  AIC=1585.76
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn +
##     crim + rad + tax
##
##         Df Sum of Sq   RSS    AIC
## <none>               11081 1585.8
## + indus  1   2.51754 11079 1587.7
## + age    1   0.06271 11081 1587.8
```

```
summary(modelFront)
```

```
##
## Call:
## lm(formula = medv ~ lstat + rm + ptratio + dis + nox + chas +
##     black + zn + crim + rad + tax, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5984  -2.7386  -0.5046   1.7273  26.2373
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
## lstat        -0.522553   0.047424 -11.019  < 2e-16 ***
## rm            3.801579   0.406316   9.356  < 2e-16 ***
## ptratio      -0.946525   0.129066  -7.334 9.24e-13 ***
## dis          -1.492711   0.185731  -8.037 6.84e-15 ***
## nox         -17.376023   3.535243  -4.915 1.21e-06 ***
## chas          2.718716   0.854240   3.183 0.001551 **
## black         0.009291   0.002674   3.475 0.000557 ***
## zn            0.045845   0.013523   3.390 0.000754 ***
## crim         -0.108413   0.032779  -3.307 0.001010 **
## rad           0.299608   0.063402   4.726 3.00e-06 ***
## tax          -0.011778   0.003372  -3.493 0.000521 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.736 on 494 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7348
## F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
```

```
summary(modelBack)
```

```
##
```

```
## Call:
## lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad +
##     tax + ptratio + black + lstat, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5984  -2.7386  -0.5046   1.7273  26.2373
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.341145   5.067492   7.171 2.73e-12 ***
## crim         -0.108413   0.032779  -3.307 0.001010 **
## zn            0.045845   0.013523   3.390 0.000754 ***
## chas          2.718716   0.854240   3.183 0.001551 **
## nox         -17.376023   3.535243  -4.915 1.21e-06 ***
## rm            3.801579   0.406316   9.356  < 2e-16 ***
## dis          -1.492711   0.185731  -8.037 6.84e-15 ***
## rad           0.299608   0.063402   4.726 3.00e-06 ***
## tax          -0.011778   0.003372  -3.493 0.000521 ***
## ptratio      -0.946525   0.129066  -7.334 9.24e-13 ***
## black         0.009291   0.002674   3.475 0.000557 ***
## lstat        -0.522553   0.047424 -11.019  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.736 on 494 degrees of freedom
## Multiple R-squared:  0.7406, Adjusted R-squared:  0.7348
## F-statistic: 128.2 on 11 and 494 DF,  p-value: < 2.2e-16
```

```
cat("The AIC for multiple regression model:", extractAIC(multiModel)[2],"\nThe AIC for forward step
```

```
## The AIC for multiple regression model: 1589.643
## The AIC for forward stepwise selection model: 1585.761
## The AIC for backward stepwise selection model: 1585.761
```

From stepwise selection, we see that all variables except indus and age are significant and retained in the model.

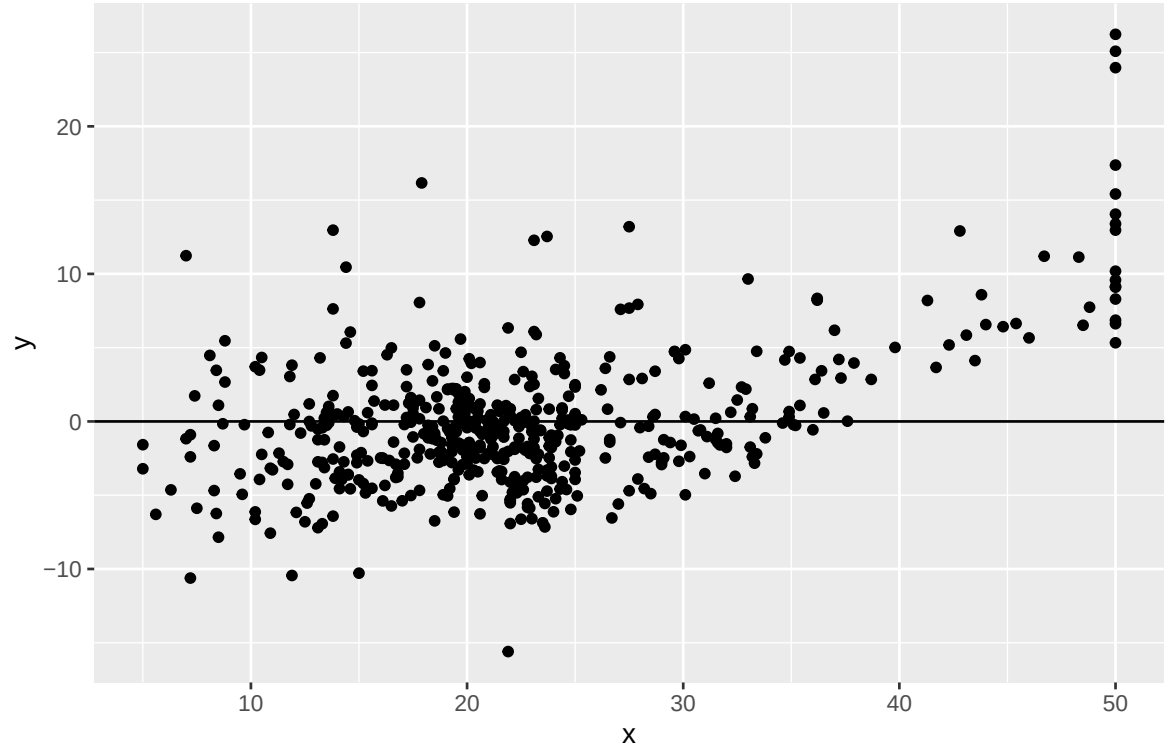
The AIC for all the models is almost comparable. AIC of the stepwise selection model is better than that of multiple regressionmodel as it get rids of predictor variables which have weak/no association with the response variable thus improving the overall fit of the model.

Evaluating Statistical Assumption

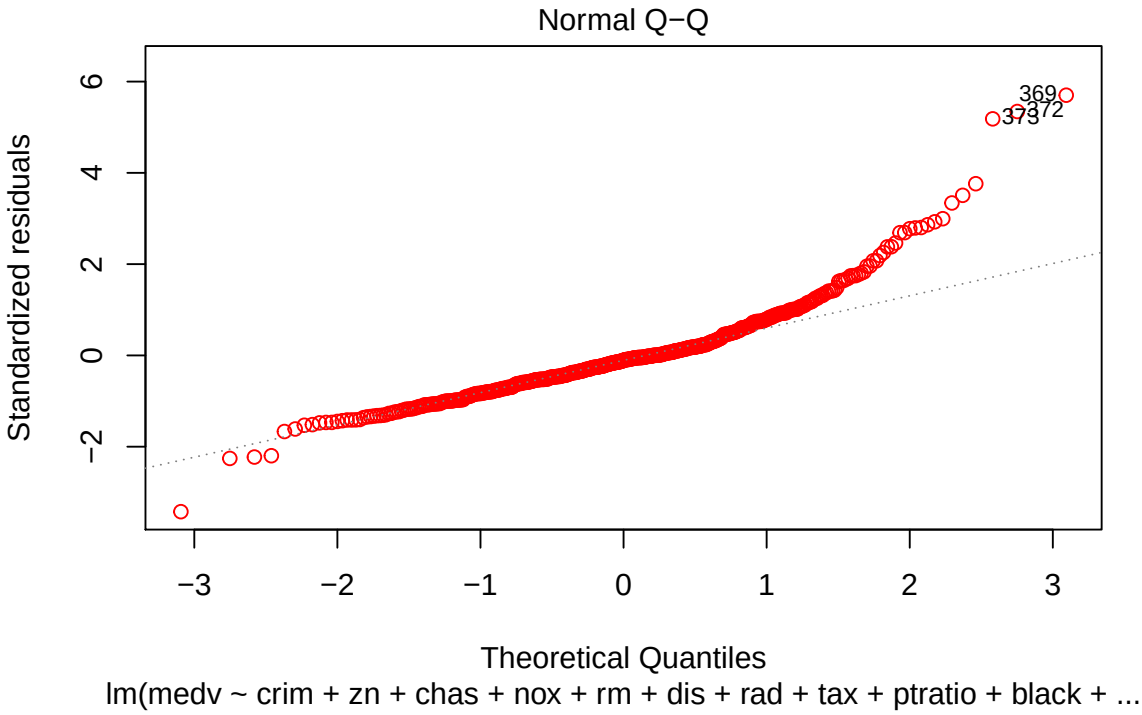
8. Evaluate the statistical assumptions in your regression analysis from (7) by performing a basic analysis of model residuals and any unusual observations.

```
residual <- modelBack$residuals
ggplot(data = data.frame(x = Boston$medv,
                         y = residual),
       aes(x = x, y = y)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  ggtitle("Figure 6: Residual Plot for ModelBack")
```

Figure 6: Residual Plot for ModelBack




```
plot(modelBack, which=2, col=c("red"))
```



The statistical assumption for a regression analysis is that

(a) Error is independent for each observation and identically distributed with a common variance.

(b) Linearity

(c) Normality

Thus, analysis of the residual plot for the modelBack generated in (7) helps us understand if our assumptions are reasonable in this data context.

Figure 6 helps establish that there is no symmetric pattern or trend in the residual plot which tells us that the linearity assumption holds true. Though, it is a bit crowded at around $20000 the overall distribution of residual is fairly spread around the horizontal line without a pattern. This establishes that our assumption 1 holds true to a significant exten. Though there is an unusual trend towards higher value of medv which could easily be explained as there are outliers. In addition to this, there is an unusual crowding as mentioned before at $20000. We might want to look into in further.

Figure 7, the Normal QQ plot is used to indictate that the residuals are fairly normally distributed as is expected of them (assumption 3). There is a slight deviation from the expected towards the end but that is probably not significant, making this a reasonable alignment.

One of the concern about the models is that the most strongly correlated predictor variable, lstat has a non-linear relationship with response variable, medv. Yet we continue to use a linear regression model to fit their relationship which I believe is not the "best" fit model available and we should explore other methods.