What is the best model to predict the median value of the houses in the Boston area?

Analyzing Boston Dataset

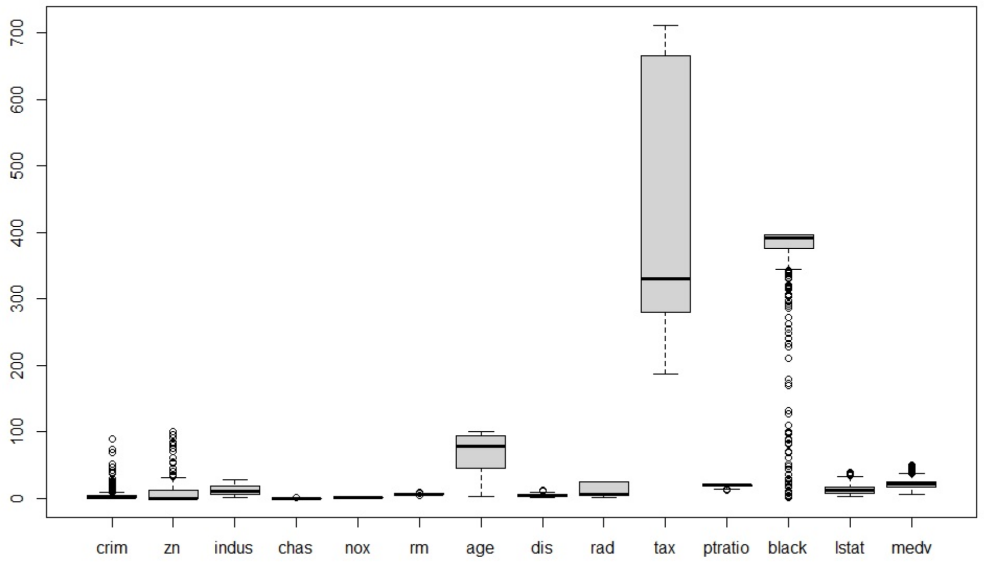
```
> summary(Boston)
      crim                zn             indus            chas        
 Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000  
 1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000  
 Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000  
 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            lstat      
 Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32   Min.   : 1.73  
 1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38   1st Qu.: 6.95  
 Median : 5.000   Median :330.0   Median :19.05   Median :391.44   Median :11.36  
 Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67   Mean   :12.65  
 3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23   3rd Qu.:16.95  
 Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90   Max.   :37.97  
      medv      
 Min.   : 5.00  
 1st Qu.:17.02  
 Median :21.20  
 Mean   :22.53  
 3rd Qu.:25.00  
 Max.   :50.00  
```

```r
#Importing Libraries

library(MASS)
library(dplyr)
library(GGally)
library(glmnet)
library(randomForest)

#Analyzing Boston Data

data(Boston)
summary(Boston)
boxplot(Boston)
boxplot(Boston$medv)
```



There are total 14 variables in the dataset and 506 observations. We see that there are no missing values in any of our variables.

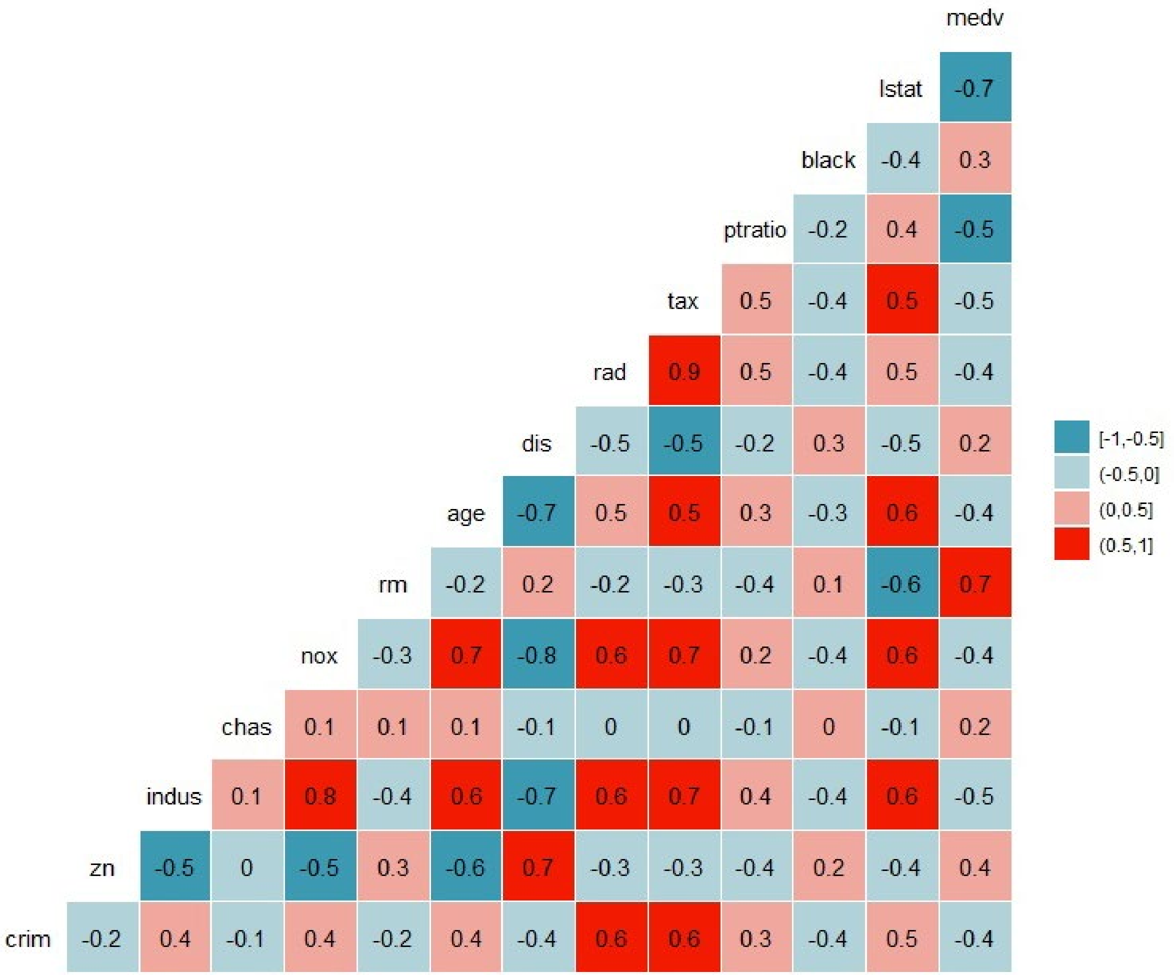
Correlation

```
#Correlation
cor(Boston)
ggcorr(Boston, nbreaks=4,label = TRUE)

#Check Conditions to see if Linear Regression can be

#Independence of observations i.e. no autocorrelatior
cor(Boston, Boston$medv)
```

```
> cor(Boston, Boston$medv)
              [,1]
crim    -0.3883046
zn       0.3604453
indus   -0.4837252
chas     0.1752602
nox     -0.4273208
rm       0.6953599
age     -0.3769546
dis      0.2499287
rad     -0.3816262
tax     -0.4685359
ptratio -0.5077867
black    0.3334608
lstat   -0.7376627
medv     1.0000000
>
```

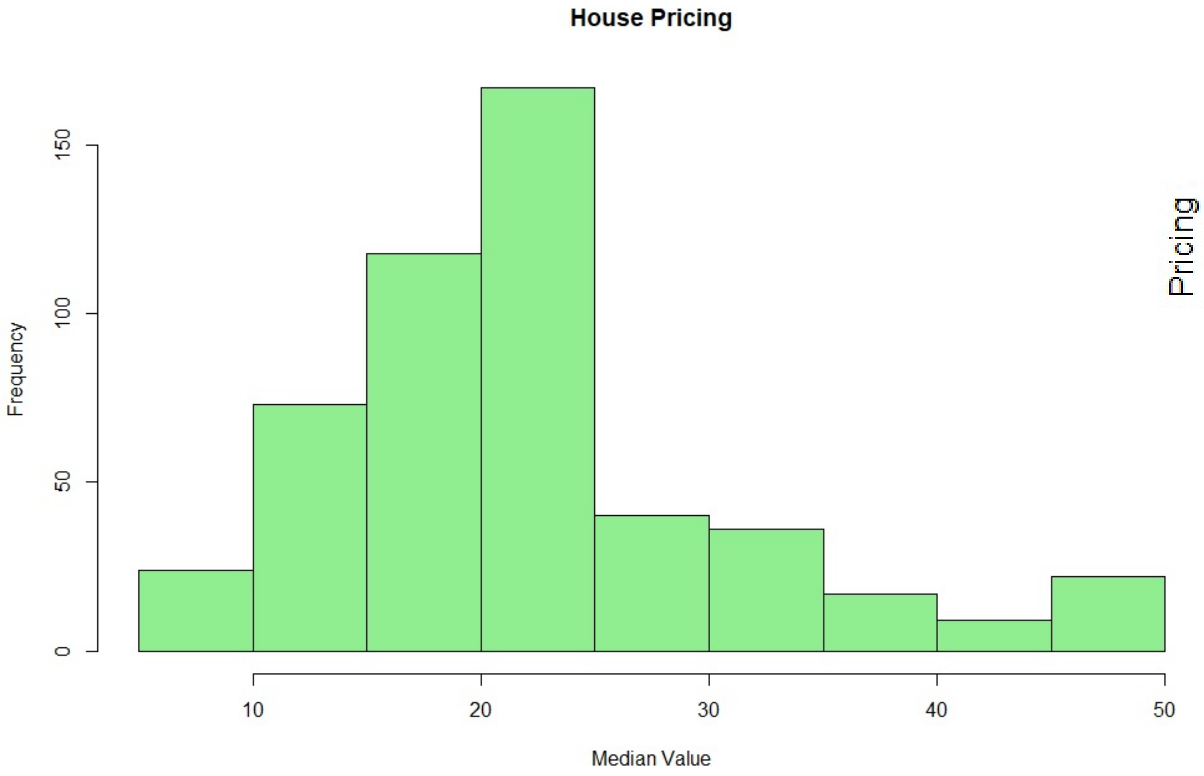


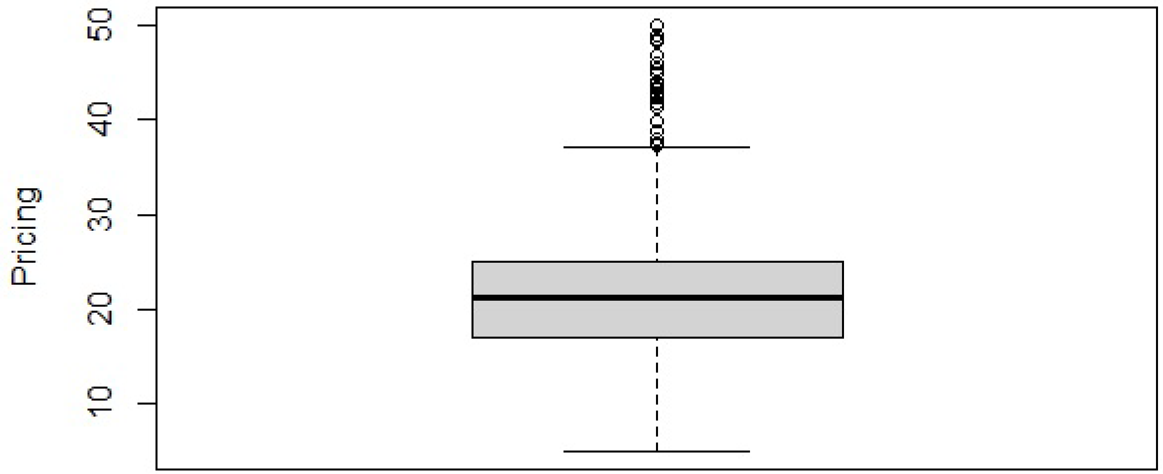
From the graph, we find a strong positive correlation in the number of rooms(rm) and median price (medv) of the house and a negative correlation between percentage of lower status of population (lstat) and median house price(medv). Also, the least correlation to medv is the proximity to Charles River (chas).

Normality

```
#Normality
hist(Boston$medv, col = "Light Green", xlab="Median Value",
     main="House Pricing")
```

```
boxplot(Boston$medv)
```





After visualizing the distribution of 'medv' from the graph we can see that the median value of housing price is skewed to the right, with several outliers to the right. A boxplot is also plotted to show an additional perspective.

Training and Testing

For finding out the best model which can help us in predicting the median value of houses in the Boston dataset, we need to perform linear regression analysis on the dataset. For proceeding with this, we form the Training and Testing data.

We partition the data on an 8/2 ratio as training/test datasets.

```
## Different Regression Models

# Training and Testing

# Partitioning the data on a 8/2 ratio as training/test data sets
set.seed(123456)
sample_data <- sample(nrow(Boston),nrow(Boston)*0.80)
training_set <- Boston[sample_data,]
test_set <- Boston[-sample_data,]
```

Variables Selection

Variable selection is the process of selecting the variables that should be present in the final model. This can be done in different ways – Forward selection, Backward elimination and Step-wise selection.

Forward Selection -

```
#Forward Selection
nullmodel <- lm(medv~1, data = training_set)
fullmodel <- lm(medv~., data = training_set)
forward <- step(nullmodel, scope=list(lower=nullmodel, upper=fullmodel),
                direction='forward')
summary(forward)
```

```
> forward <- step(nullmodel, scope=list(lower=nullmodel, upper=fullmodel),
+                 direction='forward')
Start:  AIC=1780.54
medv ~ 1

          Df Sum of Sq   RSS    AIC
+ lstat    1   18377.9 14605 1453.4
+ rm       1   15059.7 17923 1536.1
+ indus    1    8210.5 24772 1666.9
+ tax      1    7622.8 25360 1676.4
+ ptratio  1    7418.0 25565 1679.6
+ nox      1    7269.9 25713 1682.0
+ age      1    5383.0 27600 1710.6
+ crim     1    5165.2 27818 1713.7
+ rad      1    5027.9 27955 1715.7
+ zn       1    4087.6 28895 1729.1
+ black    1    3901.5 29081 1731.7
+ dis      1    2621.5 30361 1749.1
+ chas     1    1128.5 31854 1768.5
<none>                 32983 1780.5
```

```
> summary(forward)

Call:
lm(formula = medv ~ lstat + rm + ptratio + black + dis + nox +
    chas + zn + crim + rad + tax, data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max
-14.5060  -2.7810  -0.6083   1.6670  26.6449

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  34.853108   5.837433   5.971 5.30e-09 ***
lstat        -0.503293   0.052375  -9.609  < 2e-16 ***
rm            3.745384   0.478422   7.829 4.65e-14 ***
ptratio      -0.869966   0.150051  -5.798 1.38e-08 ***
black         0.009799   0.002920   3.355 0.000870 ***
dis          -1.375104   0.204613  -6.721 6.40e-11 ***
nox         -17.333271   4.146161  -4.181 3.59e-05 ***
chas          2.139151   1.022403   2.092 0.037056 *
zn            0.041702   0.014997   2.781 0.005687 **
crim         -0.115046   0.034534  -3.331 0.000946 ***
rad           0.299765   0.071606   4.186 3.50e-05 ***
tax          -0.012672   0.003781  -3.351 0.000882 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.792 on 392 degrees of freedom
Multiple R-squared:  0.727,     Adjusted R-squared:  0.7194
F-statistic: 94.91 on 11 and 392 DF,  p-value: < 2.2e-16
```

The forward selection method suggests that we drop the variables indus and age.
The adjusted R-square value here is 0.7194

Backward Elimination -

```
#Backward Elimination
backward <- step(fullmodel,direction='backward')

summary(backward)
```

```
Step:  AIC=1277.99
medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
    black + lstat

          Df Sum of Sq     RSS    AIC
<none>                  9003.3 1278.0
- chas     1    100.54  9103.9 1280.5
- zn       1    177.58  9180.9 1283.9
- crim     1    254.90  9258.2 1287.3
- tax      1    257.98  9261.3 1287.4
- black    1    258.60  9261.9 1287.4
- nox      1    401.41  9404.7 1293.6
- rad      1    402.52  9405.8 1293.7
- ptratio  1    772.04  9775.4 1309.2
- dis      1   1037.34 10040.7 1320.0
- rm       1   1407.62 10410.9 1334.7
- lstat    1   2120.88 11124.2 1361.5
> 
```

```
> summary(backward)

Call:
lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad + 
    tax + ptratio + black + lstat, data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.5060  -2.7810  -0.6083   1.6670  26.6449 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  34.853108   5.837433   5.971 5.30e-09 ***
crim         -0.115046   0.034534  -3.331 0.000946 ***
zn            0.041702   0.014997   2.781 0.005687 ** 
chas          2.139151   1.022403   2.092 0.037056 *  
nox         -17.333271   4.146161  -4.181 3.59e-05 ***
rm            3.745384   0.478422   7.829 4.65e-14 ***
dis          -1.375104   0.204613  -6.721 6.40e-11 ***
rad           0.299765   0.071606   4.186 3.50e-05 ***
tax          -0.012672   0.003781  -3.351 0.000882 ***
ptratio      -0.869966   0.150051  -5.798 1.38e-08 ***
black         0.009799   0.002920   3.355 0.000870 ***
lstat        -0.503293   0.052375  -9.609  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.792 on 392 degrees of freedom
Multiple R-squared:  0.727,	Adjusted R-squared:  0.7194 
F-statistic: 94.91 on 11 and 392 DF,  p-value: < 2.2e-16
```

This particular method also shows that we drop ‘indus’ and ‘age’. The adjusted R-square value is found to be 0.7194.

Step – wise Selection -

```
#Step wise Selection
stepwise <- step(nullmodel, scope=list(lower=nullmodel, upper=fullmodel),
                 direction='both')
summary(stepwise)
```

```
> stepwise <- step(nullmodel, scope=list(lower=nullmodel, upper=fullmodel]
+                  direction='both')
Start:  AIC=1780.54
medv ~ 1

          Df Sum of Sq   RSS    AIC
+ lstat    1   18377.9 14605 1453.4
+ rm       1   15059.7 17923 1536.1
+ indus    1    8210.5 24772 1666.9
+ tax      1    7622.8 25360 1676.4
+ ptratio  1    7418.0 25565 1679.6
+ nox      1    7269.9 25713 1682.0
+ age      1    5383.0 27600 1710.6
+ crim     1    5165.2 27818 1713.7
+ rad      1    5027.9 27955 1715.7
+ zn       1    4087.6 28895 1729.1
+ black    1    3901.5 29081 1731.7
+ dis      1    2621.5 30361 1749.1
+ chas     1    1128.5 31854 1768.5
<none>                 32983 1780.5
```

This particular method also shows that we drop ‘indus’ and ‘age’. The adjusted R-square value is found to be 0.7194.

```
> summary(stepwise)

Call:
lm(formula = medv ~ lstat + rm + ptratio + black + dis + nox +
    chas + zn + crim + rad + tax, data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max 
-14.5060  -2.7810  -0.6083   1.6670  26.6449 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  34.853108   5.837433   5.971 5.30e-09 ***
lstat        -0.503293   0.052375  -9.609  < 2e-16 ***
rm            3.745384   0.478422   7.829 4.65e-14 ***
ptratio      -0.869966   0.150051  -5.798 1.38e-08 ***
black         0.009799   0.002920   3.355 0.000870 ***
dis          -1.375104   0.204613  -6.721 6.40e-11 ***
nox         -17.333271   4.146161  -4.181 3.59e-05 ***
chas          2.139151   1.022403   2.092 0.037056 *  
zn            0.041702   0.014997   2.781 0.005687 ** 
crim         -0.115046   0.034534  -3.331 0.000946 ***
rad           0.299765   0.071606   4.186 3.50e-05 ***
tax          -0.012672   0.003781  -3.351 0.000882 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.792 on 392 degrees of freedom
Multiple R-squared:  0.727,	Adjusted R-squared:  0.7194 
F-statistic: 94.91 on 11 and 392 DF,  p-value: < 2.2e-16
```

Model - 1

```
## Model 1
model_1 <- lm(medv~log(lstat)+rm,data = training_set)
pred_1 <- predict(model_1, newdata = test_set)
summary(model_1)
plot(model_1)
step(model_1)
```

Here we do a linear regression model with 'medv' and lstat.

From the corrplot, it was evident that the lstat had the highest negative correlation with medv. Therefore, we take the logarithmic value of lstat.

For testing the accuracy of the model, we also calculate the Adj. R-squared values and AIC values.

The Adj. R-squared value comes out to be 0.7102

```
> step(model_1)
Start:  AIC=1282.19
medv ~ log(lstat) + rm

             Df Sum of Sq   RSS    AIC
<none>                     9512 1282.2
- rm          1     916.2 10428 1317.3
- log(lstat)  1    8411.2 17923 1536.1

Call:
lm(formula = medv ~ log(lstat) + rm, data = training_set)

Coefficients:
(Intercept)   log(lstat)           rm
     27.551      -10.124        2.999
```

```
> summary(model_1)

Call:
lm(formula = medv ~ log(lstat) + rm, data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max
-15.1136  -3.2431  -0.5674   2.3861  26.6339

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  27.5512     4.0031   6.882 2.28e-11 ***
log(lstat)  -10.1240     0.5376 -18.831  < 2e-16 ***
rm            2.9985     0.4825   6.215 1.29e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.87 on 401 degrees of freedom
Multiple R-squared:  0.7116,    Adjusted R-squared:  0.7102
F-statistic: 494.7 on 2 and 401 DF,  p-value: < 2.2e-16
```

Model -2

```
## Model 2
model_2 <- lm(medv~rm,data = training_set)
pred_2 <- predict(model_2, newdata = test_set)
summary(model_2)
plot(model_2)
step(model_2)
```

Here we do a linear regression model with ‘medv’ and rm.

From the corrplot, it was evident that the rm had the highest positive correlation with medv.

For testing the accuracy of the model, we also calculate the Adj. R-squared values and AIC values.

The Adj. R-squared value comes out to be 0.4552

```
> step(model_2)
Start:  AIC=1536.14
medv ~ rm

       Df Sum of Sq   RSS    AIC
<none>              17923 1536.1
- rm    1     15060 32983 1780.5

Call:
lm(formula = medv ~ rm, data = training_set)

Coefficients:
(Intercept)           rm
    -34.457        9.058
```

```
> summary(model_2)

Call:
lm(formula = medv ~ rm, data = training_set)

Residuals:
    Min      1Q  Median      3Q     Max
-23.174  -2.318   0.117   3.143  39.438

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -34.4568     3.1207  -11.04   <2e-16 ***
rm            9.0581     0.4929   18.38   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.677 on 402 degrees of freedom
Multiple R-squared:  0.4566,    Adjusted R-squared:  0.4552
F-statistic: 337.8 on 1 and 402 DF,  p-value: < 2.2e-16
```

Model - 3

```
## Model 3
model_3 <- lm(medv~lstat,data = training_set)
pred_3 <- predict(model_3, newdata = test_set)
summary(model_3)
plot(model_3)
step(model_3)
```

Here we do a linear regression model with ‘medv’ and lstat. From the corrplot, it was evident that the lstat had the highest negative correlation with medv.

For testing the accuracy of the model, we also calculate the Adj. R-squared values and AIC values.

The Adj. R-squared value comes out to be 0.5561

```
> step(model_3)
Start:  AIC=1453.43
medv ~ lstat

        Df Sum of Sq   RSS    AIC
<none>               14605 1453.4
- lstat  1     18378 32983 1780.5

Call:
lm(formula = medv ~ lstat, data = training_set)

Coefficients:
(Intercept)        lstat
    34.2176      -0.9292
```

```
> summary(model_3)

Call:
lm(formula = medv ~ lstat, data = training_set)

Residuals:
   Min     1Q Median     3Q    Max
-9.825 -3.833 -1.320  2.240 24.638

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 34.21756    0.59834   57.19   <2e-16 ***
lstat       -0.92920    0.04131  -22.49   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6.028 on 402 degrees of freedom
Multiple R-squared:  0.5572,    Adjusted R-squared:  0.5561
F-statistic: 505.8 on 1 and 402 DF,  p-value: < 2.2e-16
```

Model - 4

```
## Model 4
model_4 <- lm(medv~log(lstat)+rm+log(crim),data = training_set)
pred_4 <- predict(model_4, newdata = test_set)
summary(model_4)
plot(model_4)
step(model_4)
```

Here we do a linear regression model with ‘medv’ and log(lstat) + rm + log(crim).
For testing the accuracy of the model, we also calculate the Adj. R-squared values and AIC values.
The Adj. R-squared value comes out to be 0.7096

```
> step(model_4)
Start:  AIC=1283.93
medv ~ log(lstat) + rm + log(crim)

             Df Sum of Sq     RSS    AIC
- log(crim)   1       6.3  9512.0 1282.2
<none>                     9505.7 1283.9
- rm          1     919.4 10425.1 1319.2
- log(lstat)  1    5692.8 15198.5 1471.5

Step:  AIC=1282.19
medv ~ log(lstat) + rm

             Df Sum of Sq   RSS    AIC
<none>                     9512 1282.2
- rm          1     916.2 10428 1317.3
- log(lstat)  1    8411.2 17923 1536.1

Call:
lm(formula = medv ~ log(lstat) + rm, data = training_set)

Coefficients:
(Intercept)   log(lstat)           rm
     27.551      -10.124        2.999
```

```
> summary(model_4)

Call:
lm(formula = medv ~ log(lstat) + rm + log(crim), data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max
-14.9272  -3.1697  -0.6181   2.4290  26.8740

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 26.84576    4.23502   6.339 6.25e-10 ***
log(lstat)  -9.94347    0.64245 -15.478  < 2e-16 ***
rm           3.03358    0.48771   6.220 1.25e-09 ***
log(crim)   -0.07175    0.13951  -0.514    0.607
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.875 on 400 degrees of freedom
Multiple R-squared:  0.7118,	Adjusted R-squared:  0.7096
F-statistic: 329.3 on 3 and 400 DF,  p-value: < 2.2e-16
```

Model - 5

```
## Model 5
model_5 <- lm(medv~poly(lstat, 2),data = training_set)
pred_5 <- predict(model_5, newdata = test_set)
summary(model_5)
plot(model_5)
step(model_5)
```

Here we do a linear regression model with ‘medv’ and poly(lstat , 2).

For testing the accuracy of the model, we also calculate the Adj. R-squared values and AIC values.
The Adj. R-squared value comes out to be 0.658

```
> step(model_5)
Start:  AIC=1349.05
medv ~ poly(lstat, 2)

                 Df Sum of Sq   RSS    AIC
<none>                        11224 1349.0
- poly(lstat, 2)  2     21759 32983 1780.5

Call:
lm(formula = medv ~ poly(lstat, 2), data = training_set)

Coefficients:
    (Intercept)  poly(lstat, 2)1  poly(lstat, 2)2
          22.57          -135.57            58.15
```

```
> summary(model_5)

Call:
lm(formula = medv ~ poly(lstat, 2), data = training_set)

Residuals:
    Min      1Q  Median      3Q     Max
-9.9473 -3.8053 -0.4867  2.4104 25.5822

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)      22.5725     0.2632   85.76   <2e-16 ***
poly(lstat, 2)1 -135.5651     5.2905  -25.62   <2e-16 ***
poly(lstat, 2)2   58.1485     5.2905   10.99   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 5.29 on 401 degrees of freedom
Multiple R-squared:  0.6597,	Adjusted R-squared:  0.658
F-statistic: 388.7 on 2 and 401 DF,  p-value: < 2.2e-16
```

Model - 6

```
## Model 6
model_6 <- lm( medv ~ .,data = training_set )
pred_6 <- predict(model_6, newdata = test_set)
summary(model_6)
plot(model_6)
step(model_6)
```

Here we do a linear regression model with ‘medv’ as the dependent variable and all the remaining variables as independent.
For doing this, train the model with the training dataset. After that, we use the trained model to predict the outcome for the testing dataset.

For testing the accuracy of the model, we also calculate the Adj. R-squared values and AIC values.
The Adj. R-squared value comes out to be 0.7183 and AIC value is 1281.45

```
> summary(model_6)

Call:
lm(formula = medv ~ ., data = training_set)

Residuals:
    Min      1Q  Median      3Q     Max 
-14.561  -2.806  -0.611   1.711  26.650 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  35.099186   5.881317   5.968 5.40e-09 ***
crim         -0.114078   0.034626  -3.295 0.001076 ** 
zn            0.042405   0.015170   2.795 0.005441 ** 
indus         0.047954   0.067457   0.711 0.477577    
chas          2.086039   1.029308   2.027 0.043379 *  
nox         -18.089859   4.506975  -4.014 7.17e-05 ***
rm            3.783068   0.491031   7.704 1.10e-13 ***
age          -0.001709   0.014863  -0.115 0.908516    
dis          -1.350839   0.220346  -6.131 2.15e-09 ***
rad           0.311823   0.074038   4.212 3.15e-05 ***
tax          -0.013779   0.004099  -3.361 0.000852 ***
ptratio      -0.883546   0.151862  -5.818 1.24e-08 ***
black         0.009888   0.002935   3.369 0.000830 ***
lstat        -0.505375   0.056580  -8.932  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.802 on 390 degrees of freedom
Multiple R-squared:  0.7274,	Adjusted R-squared:  0.7183 
F-statistic: 80.05 on 13 and 390 DF,  p-value: < 2.2e-16
```

```
> step(model_6)
Start:  AIC=1281.45
medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad + 
    tax + ptratio + black + lstat

          Df Sum of Sq     RSS    AIC
- age      1      0.30  8991.6 1279.5
- indus    1     11.65  9003.0 1280.0
<none>                  8991.3 1281.5
- chas     1     94.69  9086.0 1283.7
- zn       1    180.14  9171.5 1287.5
- crim     1    250.23  9241.6 1290.5
- tax      1    260.50  9251.8 1291.0
- black    1    261.67  9253.0 1291.0
- nox      1    371.41  9362.7 1295.8
- rad      1    408.94  9400.3 1297.4
- ptratio  1    780.40  9771.7 1313.1
- dis      1    866.47  9857.8 1316.6
- rm       1   1368.45 10359.8 1336.7
- lstat    1   1839.30 10830.6 1354.6

 Step:  AIC=1279.46
 medv ~ crim + zn + indus + chas + nox + rm + dis + rad + tax + 
     ptratio + black + lstat

          Df Sum of Sq     RSS    AIC
 - indus   1     11.68  9003.3 1278.0
 <none>                 8991.6 1279.5
 - chas    1     94.39  9086.0 1281.7
 - zn      1    184.00  9175.6 1285.7
 - crim    1    250.54  9242.2 1288.6
 - tax     1    260.82  9252.4 1289.0
 - black   1    261.76  9253.4 1289.1
 - nox     1    406.65  9398.3 1295.3
 - rad     1    411.99  9403.6 1295.6
 - ptratio 1    783.62  9775.3 1311.2
 - dis     1    944.28  9935.9 1317.8
 - rm      1   1419.02 10410.6 1336.7
 - lstat   1   2128.56 11120.2 1363.3
```

```
Step:  AIC=1277.99
medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio + 
    black + lstat

          Df Sum of Sq     RSS    AIC
<none>                  9003.3 1278.0
- chas     1    100.54  9103.9 1280.5
- zn       1    177.58  9180.9 1283.9
- crim     1    254.90  9258.2 1287.3
- tax      1    257.98  9261.3 1287.4
- black    1    258.60  9261.9 1287.4
- nox      1    401.41  9404.7 1293.6
- rad      1    402.52  9405.8 1293.7
- ptratio  1    772.04  9775.4 1309.2
- dis      1   1037.34 10040.7 1320.0
- rm       1   1407.62 10410.9 1334.7
- lstat    1   2120.88 11124.2 1361.5

Call:
lm(formula = medv ~ crim + zn + chas + nox + rm + dis + rad + 
    tax + ptratio + black + lstat, data = training_set)

Coefficients:
(Intercept)         crim           zn         chas          nox           rm  
  34.853108    -0.115046     0.041702     2.139151   -17.333271     3.745384  
        dis          rad          tax      ptratio        black        lstat  
  -1.375104     0.299765    -0.012672    -0.869966     0.009799    -0.503293  
```

Model - 7

```
## Model 7
#Using the selected variables
model_7 <- lm( medv ~ crim + zn + chas + nox + rm + dis + ptratio +
                 rad + black + lstat + tax ,data = training_set )
pred_7 <- predict(model_7, newdata = test_set)
summary(model_7)
plot(model_7)
step(model_7)
```

```
> summary(model_7)

Call:
lm(formula = medv ~ crim + zn + chas + nox + rm + dis + ptratio +
    rad + black + lstat + tax, data = training_set)

Residuals:
     Min       1Q   Median       3Q      Max
-14.5060  -2.7810  -0.6083   1.6670  26.6449

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 34.853108   5.837433   5.971 5.30e-09 ***
crim        -0.115046   0.034534  -3.331 0.000946 ***
zn           0.041702   0.014997   2.781 0.005687 **
chas         2.139151   1.022403   2.092 0.037056 *
nox        -17.333271   4.146161  -4.181 3.59e-05 ***
rm           3.745384   0.478422   7.829 4.65e-14 ***
dis         -1.375104   0.204613  -6.721 6.40e-11 ***
ptratio     -0.869966   0.150051  -5.798 1.38e-08 ***
rad          0.299765   0.071606   4.186 3.50e-05 ***
black        0.009799   0.002920   3.355 0.000870 ***
lstat       -0.503293   0.052375  -9.609  < 2e-16 ***
tax         -0.012672   0.003781  -3.351 0.000882 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.792 on 392 degrees of freedom
Multiple R-squared:  0.727,	Adjusted R-squared:  0.7194
F-statistic: 94.91 on 11 and 392 DF,  p-value: < 2.2e-16
```

```
> step(model_7)
Start:  AIC=1277.99
medv ~ crim + zn + chas + nox + rm + dis + ptratio + rad + black + 
    lstat + tax

          Df Sum of Sq     RSS    AIC
<none>                  9003.3 1278.0
- chas     1    100.54  9103.9 1280.5
- zn       1    177.58  9180.9 1283.9
- crim     1    254.90  9258.2 1287.3
- tax      1    257.98  9261.3 1287.4
- black    1    258.60  9261.9 1287.4
- nox      1    401.41  9404.7 1293.6
- rad      1    402.52  9405.8 1293.7
- ptratio  1    772.04  9775.4 1309.2
- dis      1   1037.34 10040.7 1320.0
- rm       1   1407.62 10410.9 1334.7
- lstat    1   2120.88 11124.2 1361.5

Call:
lm(formula = medv ~ crim + zn + chas + nox + rm + dis + ptratio + 
    rad + black + lstat + tax, data = training_set)

Coefficients:
(Intercept)         crim           zn         chas          nox           rm  
  34.853108    -0.115046     0.041702     2.139151   -17.333271     3.745384  
        dis      ptratio          rad        black        lstat          tax  
  -1.375104    -0.869966     0.299765     0.009799    -0.503293    -0.012672  
```

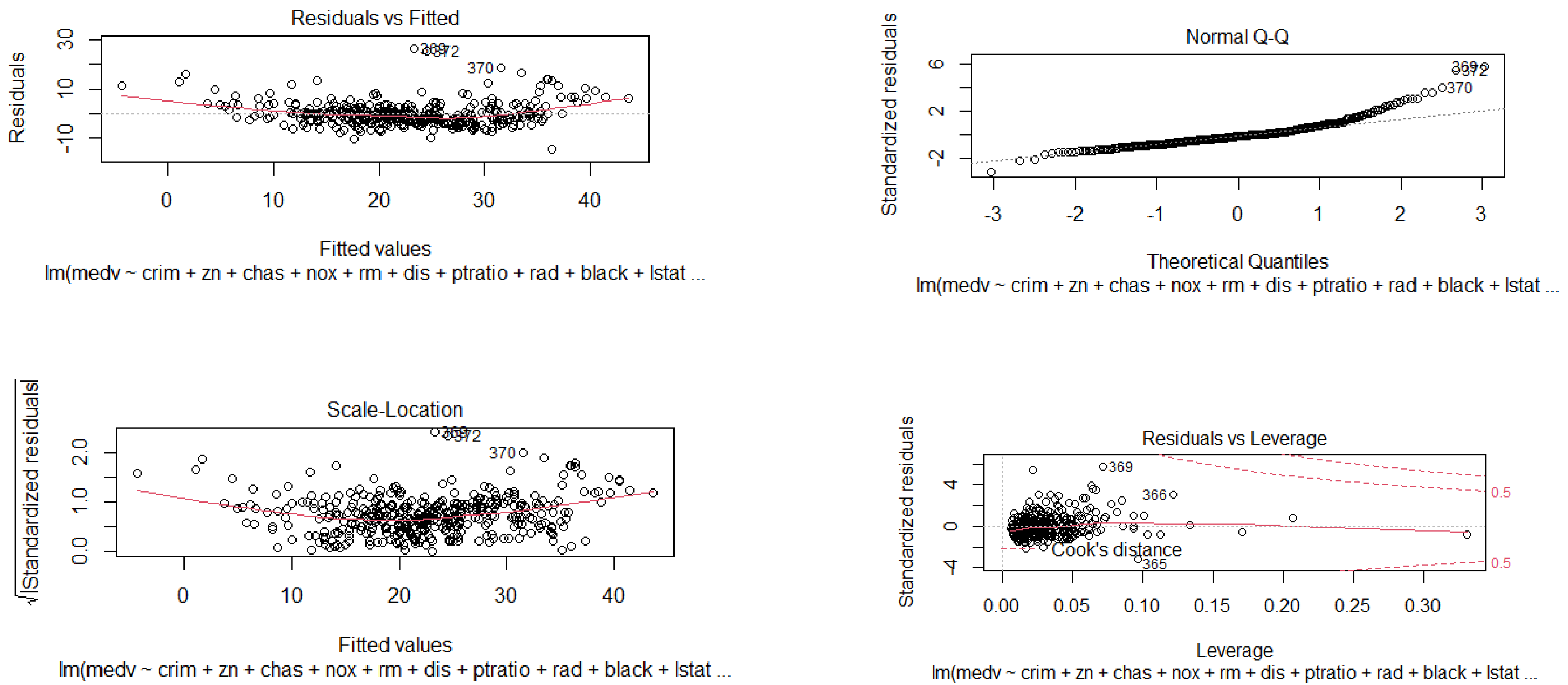
For this model, we are using the variables which we chose by the variable selection method.

From the summary of Model_7 we can see that the 'age' and 'indus' variables have a significant value of 1, which indicates that they are not statistically significant.
So, we can drop these variables from the model and from a model with the remaining variables.
Here we can see that R-squared value increased slightly. The Adj. R-squared value comes out to be 0.7194.
And the AIC value is 1277.99

So, we can say that this is the best model.

Residual Analysis for Final Model

```
> plot(model_7)
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
Hit <Return> to see next plot:
>
```



Conclusion

```
> summary(model_1)$adj.r.squared
[1] 0.7101703
> summary(model_2)$adj.r.squared
[1] 0.4552401
> summary(model_3)$adj.r.squared
[1] 0.5560937
> summary(model_4)$adj.r.squared
[1] 0.7096378
> summary(model_5)$adj.r.squared
[1] 0.6580132
> summary(model_6)$adj.r.squared
[1] 0.7183072
> summary(model_7)$adj.r.squared
[1] 0.7193707
>
```

Model 1 AIC	1282.19
Model 2 AIC	1536.14
Model 3 AIC	1453.43
Model 4 AIC	1283.93
Model 5 AIC	1349.05
Model 6 AIC	1281.45
Model 7 AIC	1277.91

While we tried various types of linear regression models to predict the median value of houses in the Boston dataset, we found out that the Model 7 with simple linear relationship formed using the medv as dependent variable and predictors were the variables selected through the Variable selection method of forward selection yielded the best model to predict the outcome with least AIC value of 1277.91 and greatest Adjusted R-squared value of 0.7193 which is closest value to 1.

Thus, we can conclude that the best model to predict the median value of houses in Boston suburb is the model formed using the variable selection method.