Assignemnt 4 - Group 7

Yakkali Pavan Kalyan - 2249432 Pooja sree prasanna - 2311609
Teja Akula - 1862814 Prajith - 2313014

2023-11-14

Q1 Load the data. Please download the Boston Dataset from canvas and read it in R

```
BostonData <- read.csv('/Users/leo/Downloads/Boston-2.csv',header = TRUE)
```

```
head(BostonData,n=5)
```

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

```
print(" no of columns in data ", ncol(BostonData))
```

```
## [1] " no of columns in data "
```

```
print(ncol(BostonData))
```

```
## [1] 15
```

```
print(" No of ROWSs in data ")
```

```
## [1] " No of ROWSs in data "
```

```
print(nrow(BostonData))
```

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

How many variables in the dataset? What are they? Are they quantitative or qualitative variables?

How many variables in the dataset

```
num <- ncol(BostonData)
print(num)
```

```
## [1] 15
```

What are they?

```
# names of all the variables
names(BostonData)
```

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

Are they quantitative or qualitative variables?

```
str(BostonData)
```

```
## 'data.frame':    506 obs. of  15 variables:
##  $ X      : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ crim   : num  0.00632 0.02731 0.02729 0.03237 0.06905 ...
##  $ zn     : num  18 0 0 0 0 0 12.5 12.5 12.5 12.5 ...
##  $ indus  : num  2.31 7.07 7.07 2.18 2.18 2.18 7.87 7.87 7.87 7.87 ...
##  $ chas   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ nox    : num  0.538 0.469 0.469 0.458 0.458 0.458 0.524 0.524 0.524 0.524 ...
##  $ rm     : num  6.58 6.42 7.18 7 7.15 ...
##  $ age    : num  65.2 78.9 61.1 45.8 54.2 58.7 66.6 96.1 100 85.9 ...
##  $ dis    : num  4.09 4.97 4.97 6.06 6.06 ...
##  $ rad    : int  1 2 2 3 3 3 5 5 5 5 ...
##  $ tax    : int  296 242 242 222 222 222 311 311 311 311 ...
##  $ ptratio: num  15.3 17.8 17.8 18.7 18.7 18.7 15.2 15.2 15.2 15.2 ...
##  $ black  : num  397 397 393 395 397 ...
##  $ lstat  : num  4.98 9.14 4.03 2.94 5.33 ...
##  $ medv   : num  24 21.6 34.7 33.4 36.2 28.7 22.9 27.1 16.5 18.9 ...
```

- In the Dataset total 15 variables are present. In those total 13 are quantitative variables and 2 are Qualitative variables. Quantitative variables are: crim, zn, indus, nox, rm, age, dis, ptratio, tax, ptratio, black, lstat, and medv Qualitative variables are: chas and rad. chas : values between 0 and 1 - indicating category rad : values between 1,2,3,4,5.-indicating a category

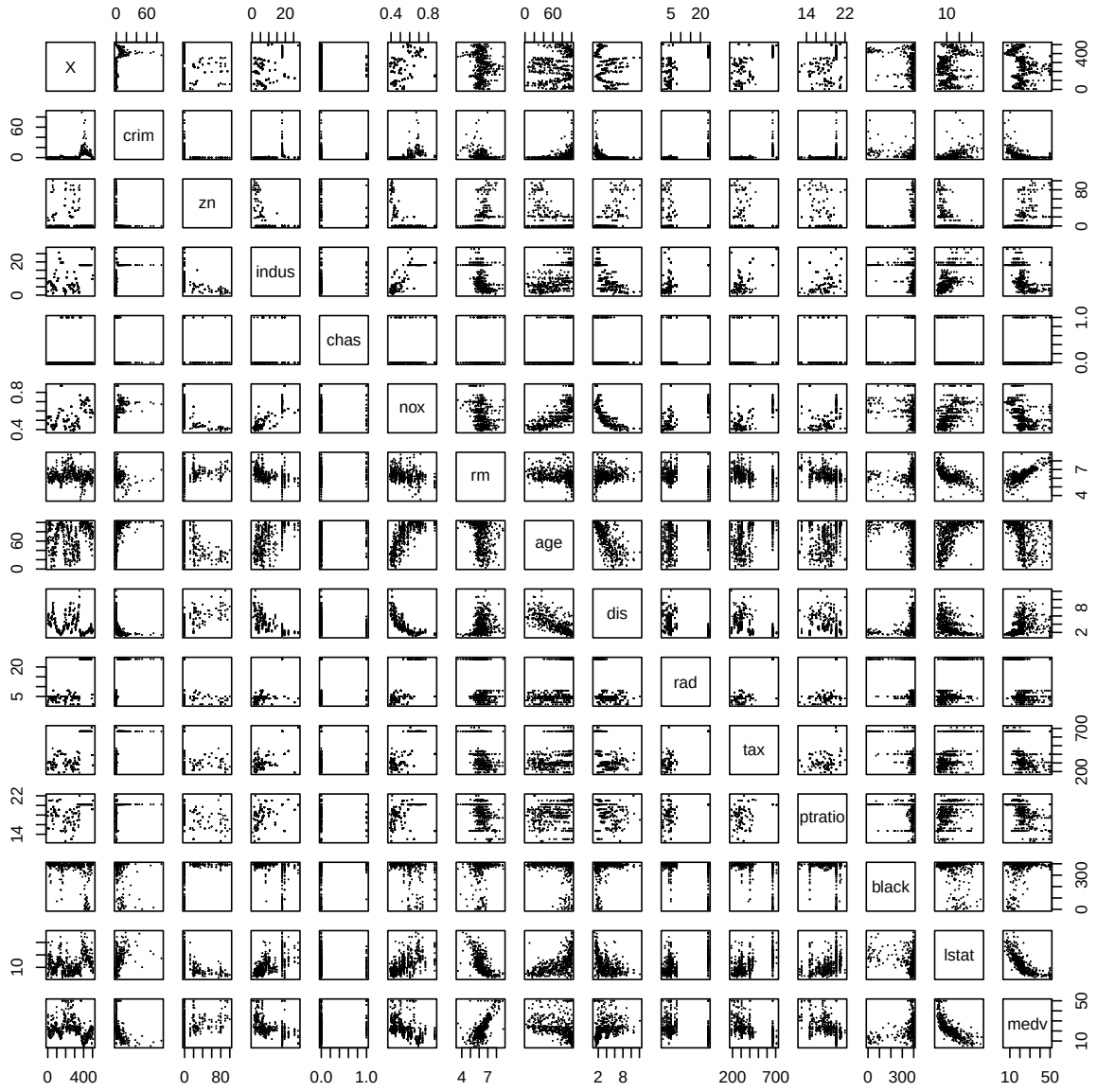
statistics of each variable

```
summary(BostonData)
```

```
##        X                crim                zn             indus
##  Min.   :  1.0   Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46
##  1st Qu.:127.2   1st Qu.: 0.08205   1st Qu.:  0.00   1st Qu.: 5.19
##  Median :253.5   Median : 0.25651   Median :  0.00   Median : 9.69
##  Mean   :253.5   Mean   : 3.61352   Mean   : 11.36   Mean   :11.14
##  3rd Qu.:379.8   3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10
##  Max.   :506.0   Max.   :88.97620   Max.   :100.00   Max.   :27.74
##       chas              nox               rm             age
##  Min.   :0.00000   Min.   :0.3850   Min.   :3.561   Min.   :  2.90
##  1st Qu.:0.00000   1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02
##  Median :0.00000   Median :0.5380   Median :6.208   Median : 77.50
##  Mean   :0.06917   Mean   :0.5547   Mean   :6.285   Mean   : 68.57
##  3rd Qu.:0.00000   3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08
##  Max.   :1.00000   Max.   :0.8710   Max.   :8.780   Max.   :100.00
##       dis               rad              tax           ptratio
##  Min.   : 1.130   Min.   : 1.000   Min.   :187.0   Min.   :12.60
##  1st Qu.: 2.100   1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40
##  Median : 3.207   Median : 5.000   Median :330.0   Median :19.05
##  Mean   : 3.795   Mean   : 9.549   Mean   :408.2   Mean   :18.46
##  3rd Qu.: 5.188   3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20
##  Max.   :12.127   Max.   :24.000   Max.   :711.0   Max.   :22.00
##      black            lstat            medv
##  Min.   :  0.32   Min.   : 1.73   Min.   : 5.00
##  1st Qu.:375.38   1st Qu.: 6.95   1st Qu.:17.02
##  Median :391.44   Median :11.36   Median :21.20
##  Mean   :356.67   Mean   :12.65   Mean   :22.53
##  3rd Qu.:396.23   3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :396.90   Max.   :37.97   Max.   :50.00
```

Q2 Data visualiz

```
pairs(BostonData,cex=0.05)
```



Correlation plot using package "corrplot"

```
# Set the CRAN mirror
options(repos = 'https://cloud.r-project.org')

# Install corrplot package
install.packages('corrplot')
```

```
##
## The downloaded binary packages are in
##  /var/folders/km/6fl99qs97wq655h4rwdy0hjh0000gn/T//RtmpWMWIlQ/downloaded_packages
```

```r
install.packages("corrplot")
```

```
##
## The downloaded binary packages are in
##  /var/folders/km/6fl99qs97wq655h4rwdy0hjh0000gn/T//RtmpWMWIlQ/downloaded_packages
```

```r
library(corrplot)
```

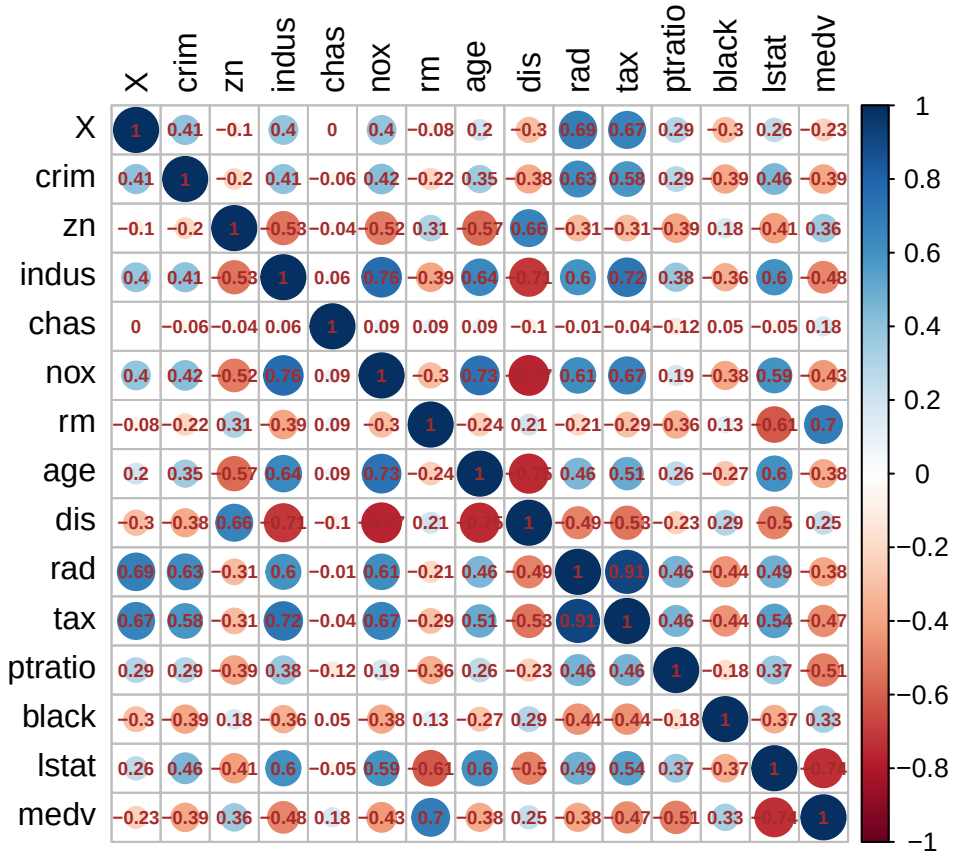
```
## corrplot 0.92 loaded
```

```r
Corr <- cor(BostonData)  # Calculate the correlation coefficient matrix of variables
Corr
```

```
##                   X        crim          zn       indus         chas
## X         1.000000000  0.40740717 -0.10339336  0.39943885 -0.003759115
## crim      0.407407172  1.00000000 -0.20046922  0.40658341 -0.055891582
## zn       -0.103393357 -0.20046922  1.00000000 -0.53382819 -0.042696719
## indus     0.399438850  0.40658341 -0.53382819  1.00000000  0.062938027
## chas     -0.003759115 -0.05589158 -0.04269672  0.06293803  1.000000000
## nox       0.398736174  0.42097171 -0.51660371  0.76365145  0.091202807
## rm       -0.079971150 -0.21924670  0.31199059 -0.39167585  0.091251225
## age       0.203783510  0.35273425 -0.56953734  0.64477851  0.086517774
## dis      -0.302210959 -0.37967009  0.66440822 -0.70802699 -0.099175780
## rad       0.686001976  0.62550515 -0.31194783  0.59512927 -0.007368241
## tax       0.666625924  0.58276431 -0.31456332  0.72076018 -0.035586518
## ptratio   0.291074227  0.28994558 -0.39167855  0.38324756 -0.121515174
## black    -0.295041232 -0.38506394  0.17552032 -0.35697654  0.048788485
## lstat     0.258464770  0.45562148 -0.41299457  0.60379972 -0.053929298
## medv     -0.226603643 -0.38830461  0.36044534 -0.48372516  0.175260177
##                  nox          rm         age         dis          rad
## X         0.39873617 -0.07997115  0.20378351 -0.30221096  0.686001976
## crim      0.42097171 -0.21924670  0.35273425 -0.37967009  0.625505145
## zn       -0.51660371  0.31199059 -0.56953734  0.66440822 -0.311947826
## indus     0.76365145 -0.39167585  0.64477851 -0.70802699  0.595129275
## chas      0.09120281  0.09125123  0.08651777 -0.09917578 -0.007368241
## nox       1.00000000 -0.30218819  0.73147010 -0.76923011  0.611440563
## rm       -0.30218819  1.00000000 -0.24026493  0.20524621 -0.209846668
## age       0.73147010 -0.24026493  1.00000000 -0.74788054  0.456022452
## dis      -0.76923011  0.20524621 -0.74788054  1.00000000 -0.494587930
## rad       0.61144056 -0.20984667  0.45602245 -0.49458793  1.000000000
## tax       0.66802320 -0.29204783  0.50645559 -0.53443158  0.910228189
## ptratio   0.18893268 -0.35550149  0.26151501 -0.23247054  0.464741179
## black    -0.38005064  0.12806864 -0.27353398  0.29151167 -0.444412816
## lstat     0.59087892 -0.61380827  0.60233853 -0.49699583  0.488676335
## medv     -0.42732077  0.69535995 -0.37695457  0.24992873 -0.381626231
##                 tax    ptratio       black      lstat       medv
## X         0.66662592  0.2910742 -0.29504123  0.2584648 -0.2266036
## crim      0.58276431  0.2899456 -0.38506394  0.4556215 -0.3883046
## zn       -0.31456332 -0.3916785  0.17552032 -0.4129946  0.3604453
## indus     0.72076018  0.3832476 -0.35697654  0.6037997 -0.4837252
## chas     -0.03558652 -0.1215152  0.04878848 -0.0539293  0.1752602
## nox       0.66802320  0.1889327 -0.38005064  0.5908789 -0.4273208
```

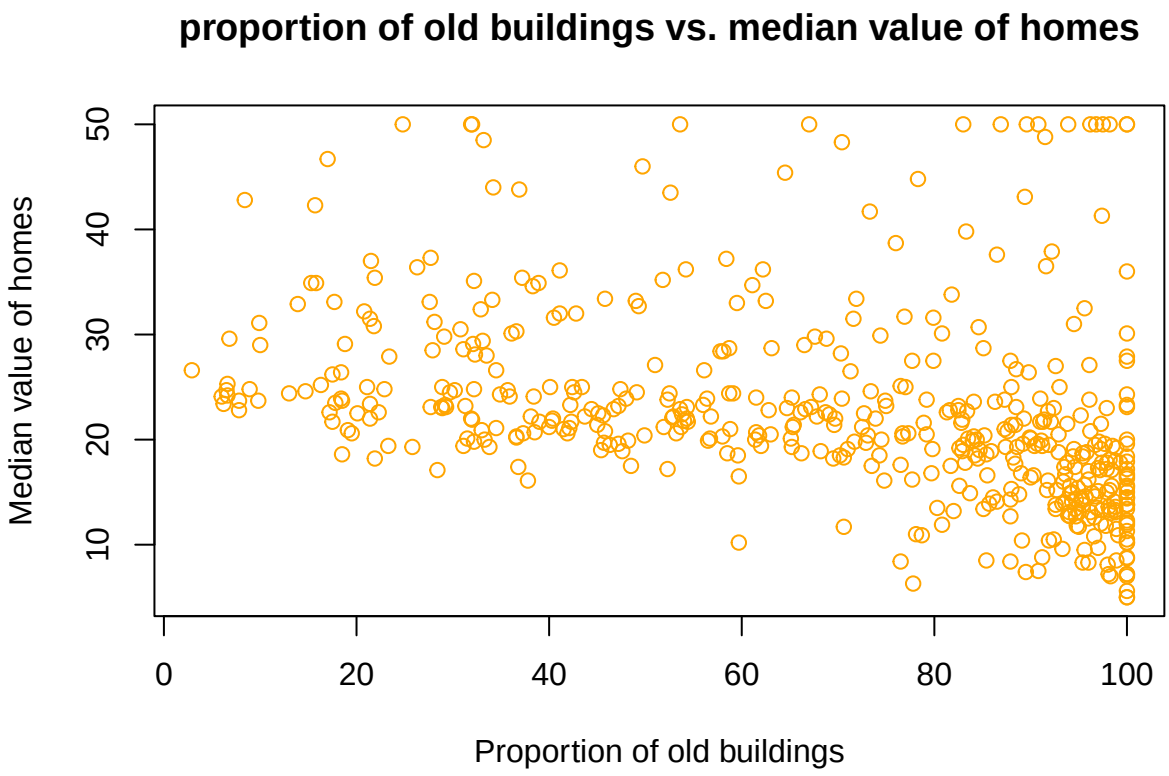
```
## rm       -0.29204783 -0.3555015  0.12806864 -0.6138083  0.6953599
## age       0.50645559  0.2615150 -0.27353398  0.6023385 -0.3769546
## dis      -0.53443158 -0.2324705  0.29151167 -0.4969958  0.2499287
## rad       0.91022819  0.4647412 -0.44441282  0.4886763 -0.3816262
## tax       1.00000000  0.4608530 -0.44180801  0.5439934 -0.4685359
## ptratio   0.46085304  1.0000000 -0.17738330  0.3740443 -0.5077867
## black    -0.44180801 -0.1773833  1.00000000 -0.3660869  0.3334608
## lstat     0.54399341  0.3740443 -0.36608690  1.0000000 -0.7376627
## medv     -0.46853593 -0.5077867  0.33346082 -0.7376627  1.0000000
```

```
corrplot(Corr,
         tl.col = "black", tl.srt = 90,
         addCoef.col = "brown", number.cex = 0.6
         )
```

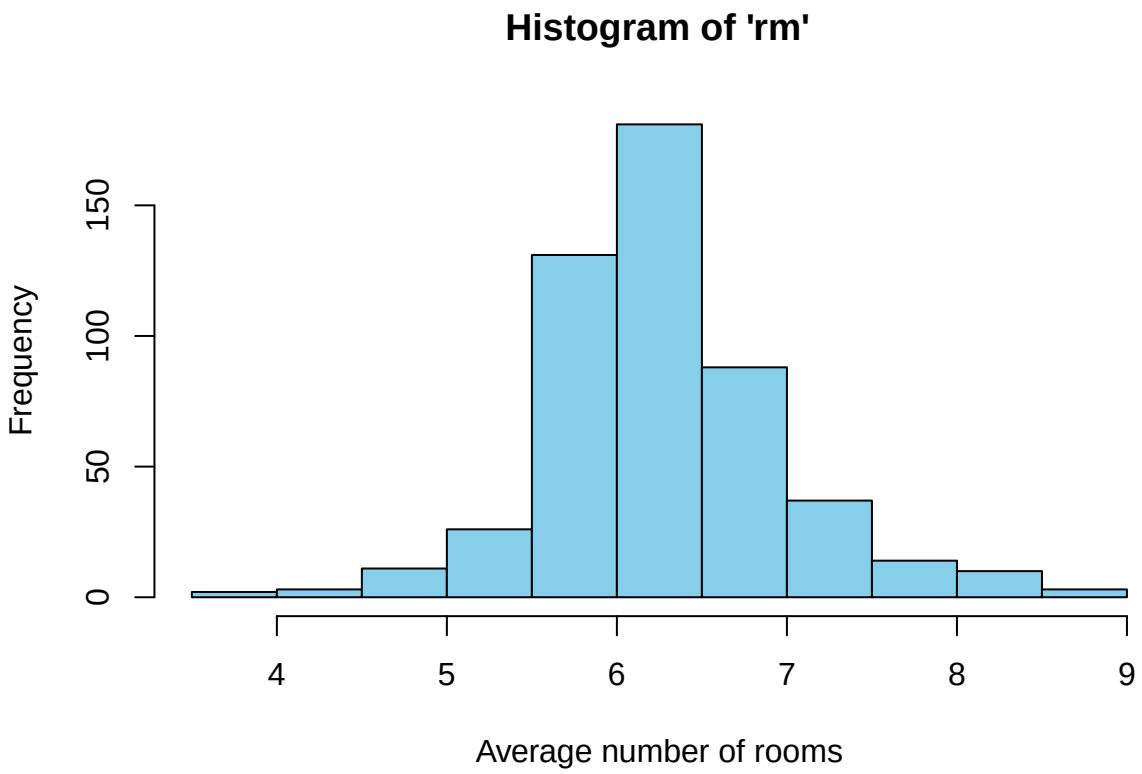


Scatterplot between 'age' and 'medv' (proportion of old buildings vs. median value of homes)

```
plot(BostonData$age, BostonData$medv, col = "orange",
     xlab = "Proportion of old buildings", ylab = "Median value of homes",
     main = "proportion of old buildings vs. median value of homes")
```

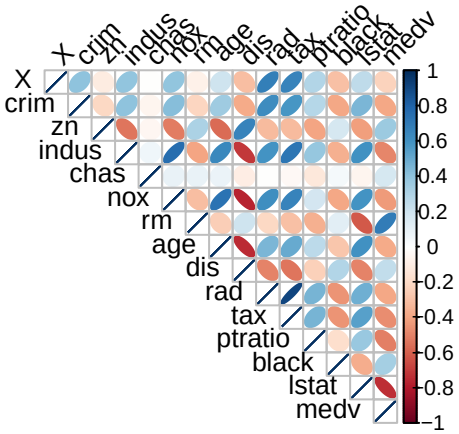


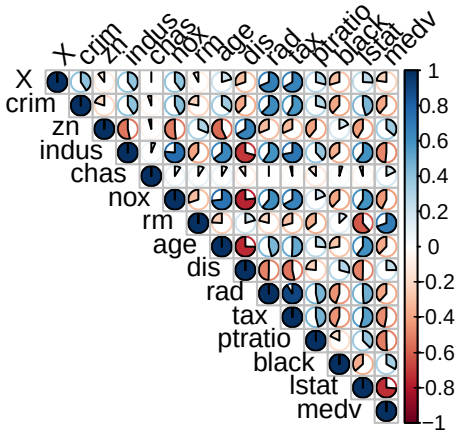

```
hist(BostonData$rm, col = "skyblue", xlab = "Average number of rooms",
     main = "Histogram of 'rm'")
```



#Corr plot

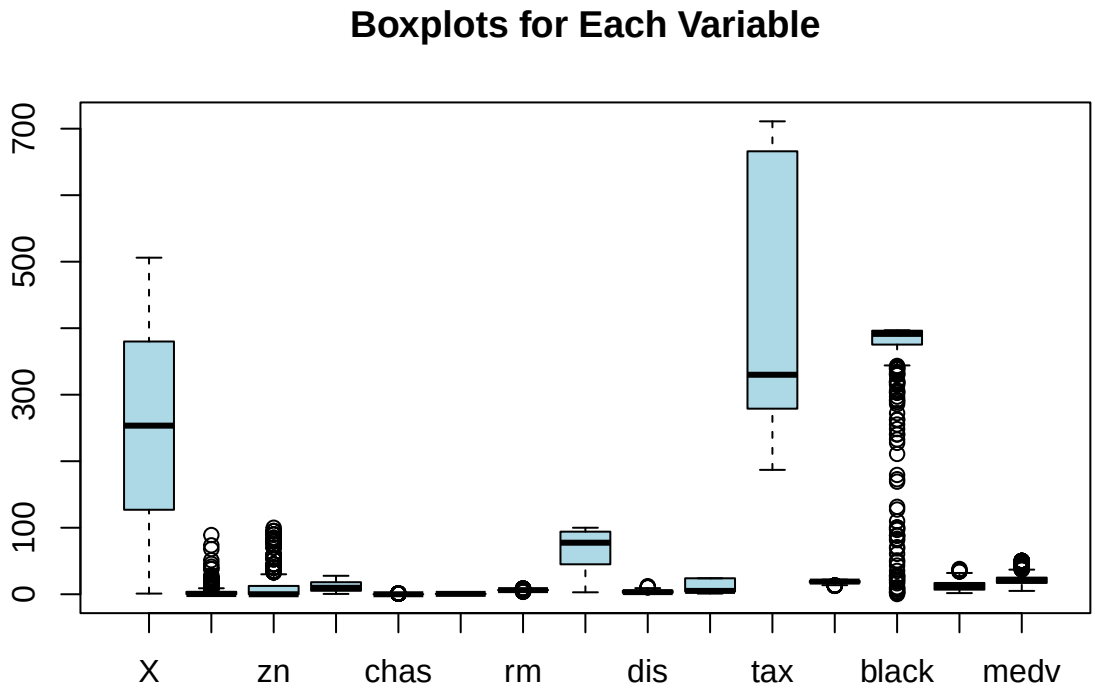
```
par(mfrow=c(2,2))
corrplot(Corr, method = "ellipse", type = "upper", tl.col = "black", tl.srt = 45)

corrplot(Corr, method = "pie", type = "upper", tl.col = "black", tl.srt = 45)
```





Box plot

```
boxplot(BostonData, main="Boxplots for Each Variable", col="lightblue", names=names(BostonData))
```



Q3 Simple linear regression. Please fit a simple linear regression model between medv (median house value) and lstat (percent of households with low socioeconomic status).

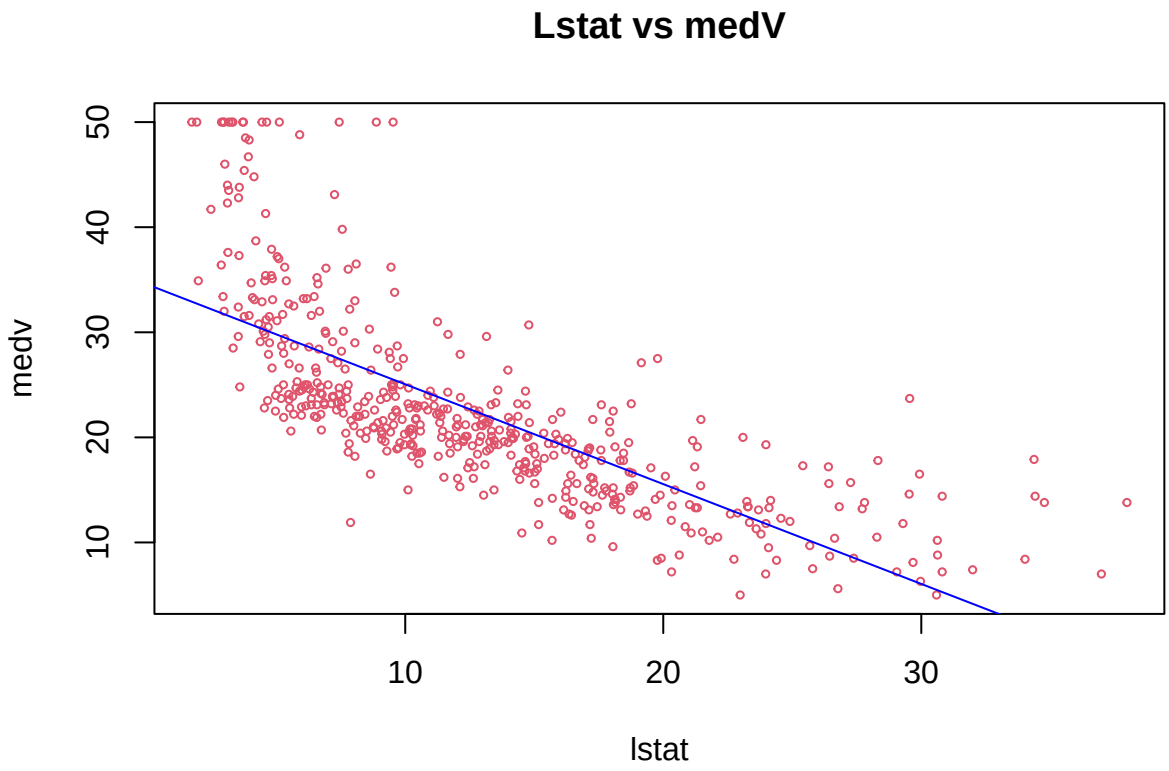
```
fit.simple <- lm(medv ~ lstat, data = BostonData)
summary(fit.simple)
```

```
##
## Call:
## lm(formula = medv ~ lstat, data = BostonData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
```

```
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

Here, we plot the relationshipship between the lstat and medv variable with abline.

```
plot(medv~ lstat, data = BostonData,cex=0.5,col=2,xlab="lstat",ylab="medv",main="Lstat vs medV")
abline(fit.simple, col = "blue")
```



a) Is there a relationship between median house value and percent of households with low socioeconomic status?

- By observing the statistics, The coeffiecient of lstat is -0.95005, P-value associated to lstat is $< 2.2e\text{-}16$, R squared value is 0.5441
- Yes. Negative relationship is observed between medv and Istat.
- If Istat values increases, The medv value decreases.
- Infer based on p-value of t-test on the coefficient.
- It indicates that as the percentage of households with low socioeconomic status increases, The median house value likely decrease.
- null hypothesis is no relationship between the variables.
- but the p-value(almost equals to zero) is very very less than 0.05, so here we reject the null hypothesis.
- Finally, there is some strong relationship between these variables.

b) How large is the effect of percent of households with low socioeconomic status on median house value?

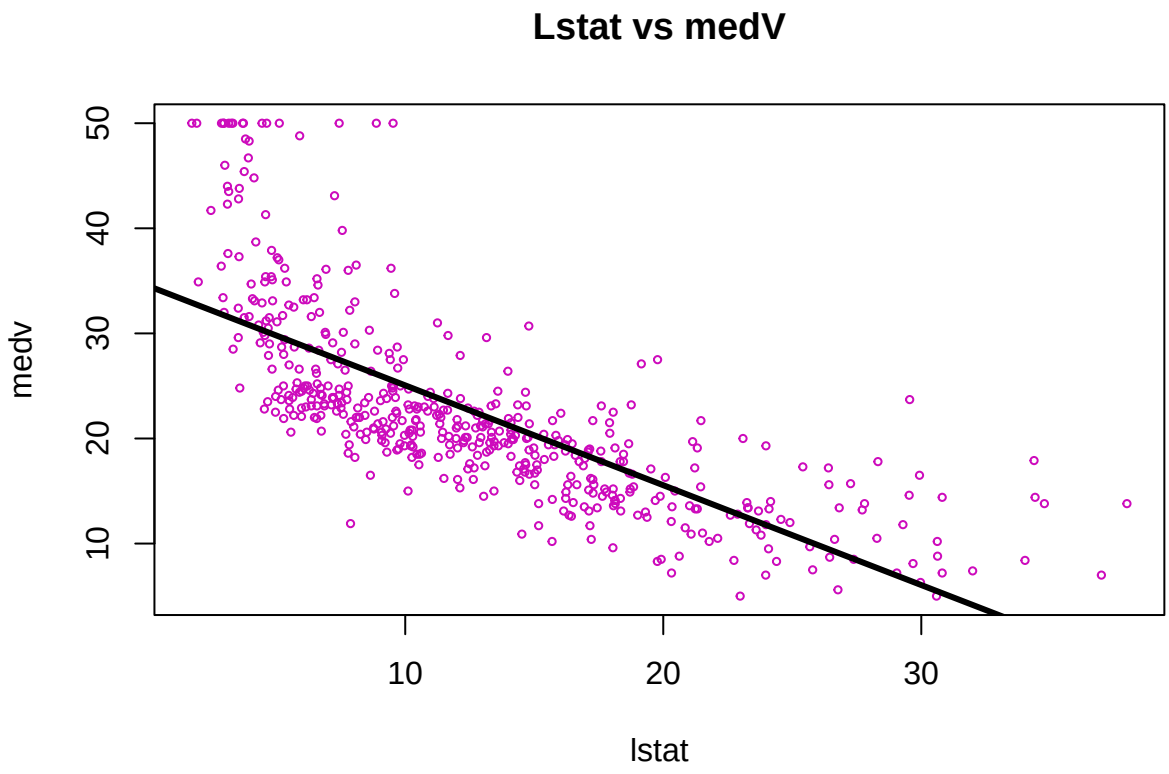
- Its respective coefficeient If lstat(percent of households with low socioeconomic status) value increases by one unit then the medv (median house) value is estimated to decrease by 0.95005 units.
- for each one unit rise in lstat results in decrease in 0.95005 units in medv.

c) How good this model fits the data?

- model goodness fit is depend on the values like R squared error, RSE, P-value etc.
- from the summary write the Residual standard error and variance what do they mean Residual standard error: 6.21 It means the predicted value is around x-6.21 to x+6.21. The predicted value range is 5 to 50.
- And also from R-squared it is 54.2% variability depends on input. *To conclude that, The model is moderately performing to predict the medv from the boston data.

d) Visualize the fitted line.

```
plot(medv~ lstat, data = BostonData,cex=0.5,col=6,xlab="lstat",ylab="medv",main="Lstat vs medV")
abline(fit.simple,lwd=3)
```



e) If the percent of households with low socioeconomic status for three new neighborhoods are 5, 10 and 15, what will be the predictions of their median house value?

```
med_pred=predict.lm(fit.simple,data.frame(lstat=c(5,10,15)))
print(med_pred)
```

```
##        1        2        3
## 29.80359 25.05335 20.30310
```

Based on the linear regression model for the new neighborhoods 5, 10, 15, The median house value predictions are 29.80,25.05 and 20.30.

f) What are the 95% confidence intervals of your predictions?

```
med_pred_conf=predict.lm(fit.simple,data.frame(lstat=c(5,10,15)),interval="confidence",level=0.95)
print(med_pred_conf)
```

```
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

for new neighborhood 5 Predicted medv value: 29.80359 95% Confidence Interval Lower Bound: 29.00741 Upper Bound: 30.59978 **for new neighborhood 10** Predicted medv value: 25.05335 95% Confidence Interval Lower Bound: 24.47413 Upper Bound: 25.63256 **for new neighborhood 15** Predicted medv value: 20.30310 95% Confidence Interval Lower Bound: 19.73159 Upper Bound: 20.87461

g) If the true median house values for three new neighborhoods are 33, 20, 50 respectively, what are residuals, what are the prediction errors? Which prediction is more accurate?

```
med_ori=c(33,20,50)
print(med_ori-med_pred)
```

```
##         1         2         3
##  3.196406 -5.053347 29.696899
```

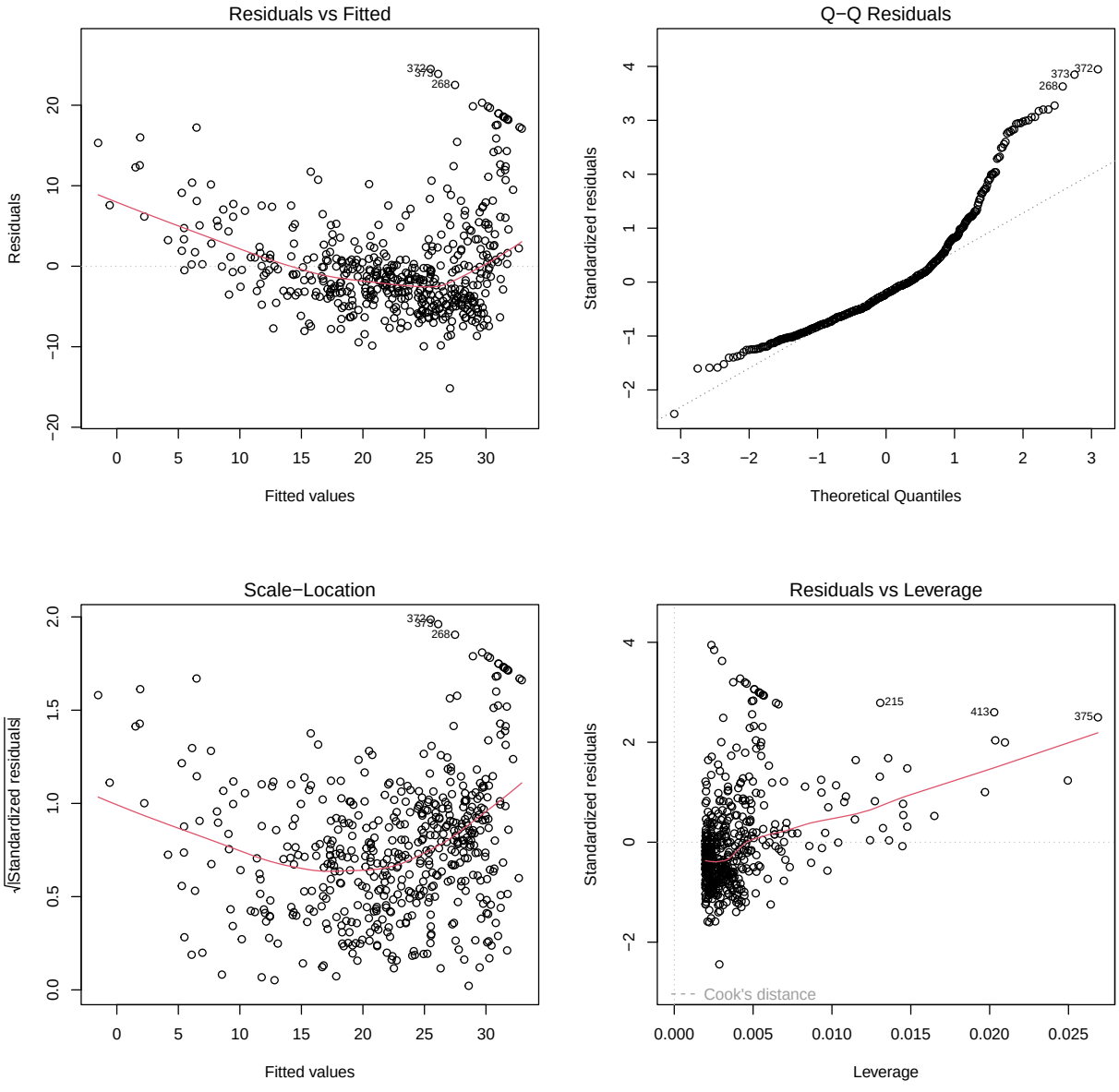
```
pred_err <- abs(med_ori-med_pred)
pred_err
```

```
##         1         2         3
##  3.196406  5.053347 29.696899
```

Estimated median house values for neighbourhoods are 29.80359,25.05335,20.30310 True median house values for three new neighborhoods are 33, 20, 50 **For lstat = 5** Residual = True value - Predicted value = 33 - 29.80359 = 3.19641 prediction error= |Residual|=|3.19641|=**3.20 For lstat = 10** Residual = True value - Predicted value = 20 - 25.05335 = -5.05335 prediction error= |Residual|=|-5.05335|=5.05 **For lstat = 15** Residual = True value - Predicted value = 50 - 20.30310 = 29.69690 prediction error= |Residual|=|29.69690|=29.69690 * The prediction with the smallest prediction error is considered as more accurate. * so the smallest prediction error is belong to the **"lstat =5"**

Q4 Residual plot. Please plot the residual plots of simple linear regression model fitted in Problem 3 and answer the following question

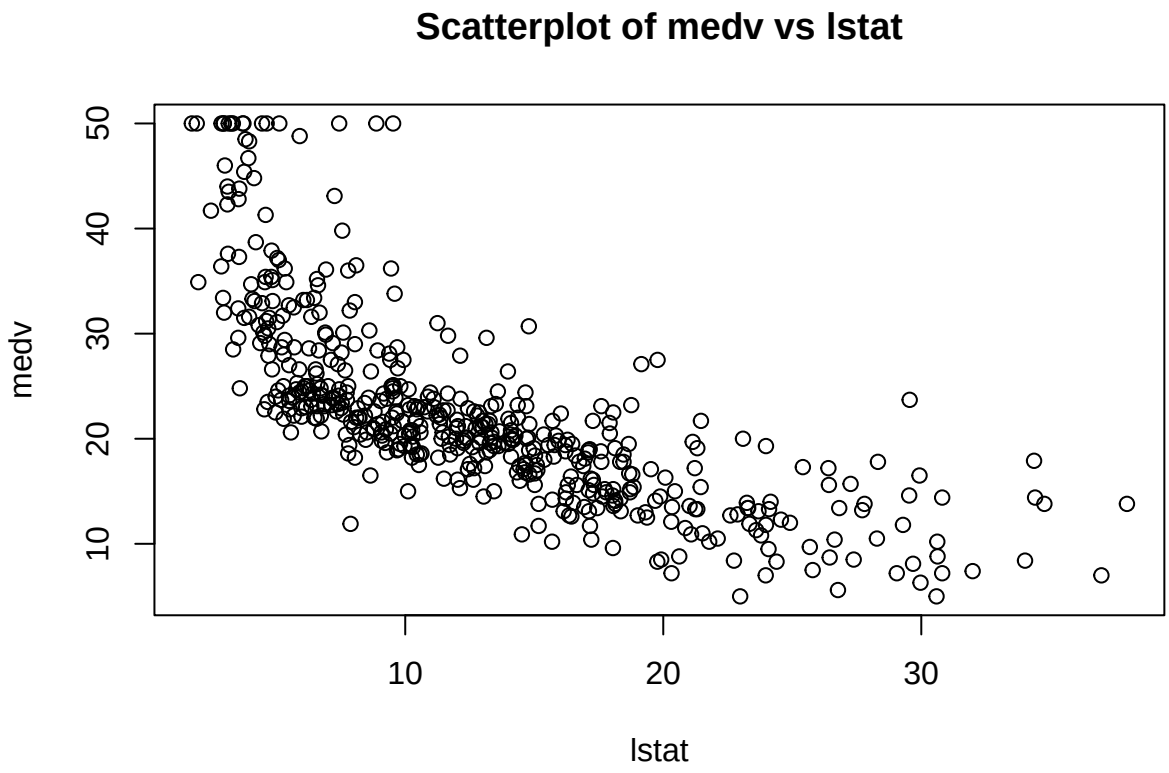
```
par(mfrow =c(2,2))
plot(fit.simple)
```



a) Is there a nonlinear relationship between medv and lstat?

```
plot(BostonData$lstat, BostonData$medv,
     xlab = "lstat",
```

```
     ylab = "medv",
     main = "Scatterplot of medv vs lstat")
```



Yes, we can see a curve which means there is non linear relationship between medv and lstat.

b) Is there correlation between error terms?

```
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

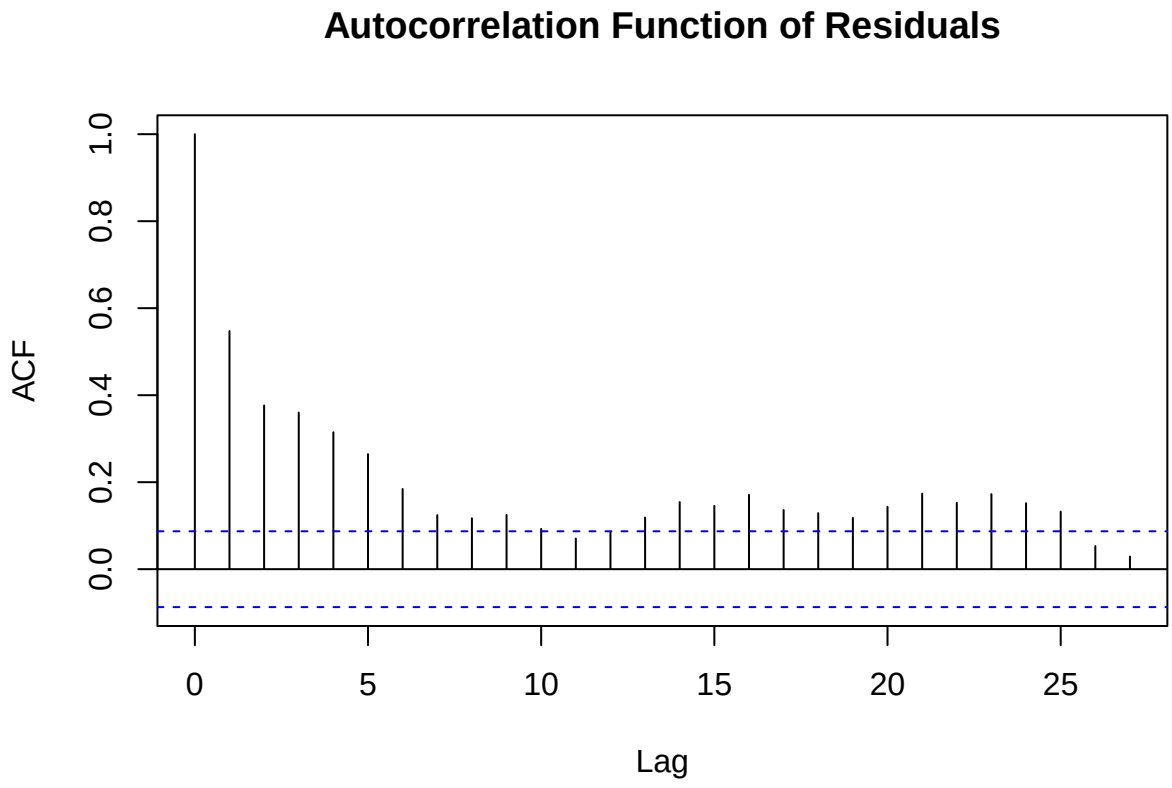
```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
dwtest(fit.simple)
```

```
##
##  Durbin-Watson test
##
## data:  fit.simple
## DW = 0.8915, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

- DW (Durbin-Watson) statistic: DW = 0.9
- p-value: p-value < 2e-16
- This DW test Values below 2 suggest positive autocorrelation, while values above suggest negative auto correlation. here DW is 0.9 suggesting a presence of positive autocorrelation.
- Null hypothesis is there is no corelation among the residuals while alternate hypothesis is residuals are autocorrelated
- p-value is less than the significance level we can reject the null hypothesis and conclude that the residuals in this regression model are autocorrelated.

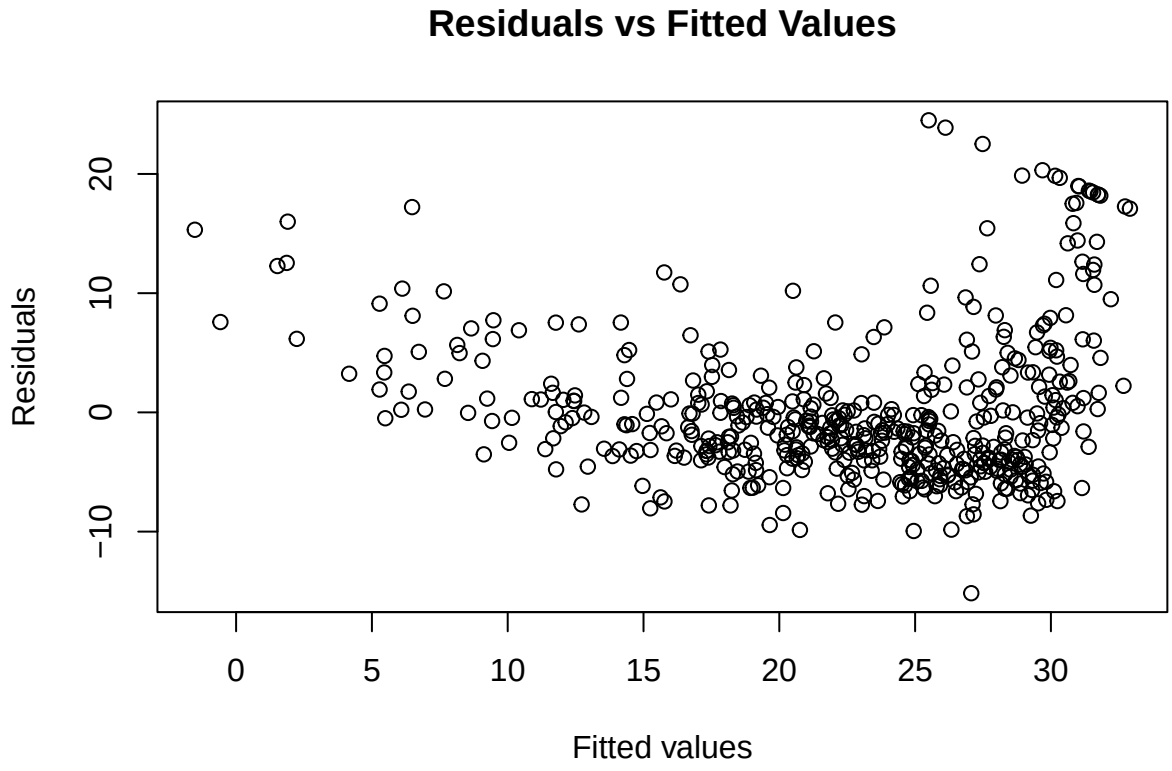
```
residuals <- residuals(fit.simple)
acf(residuals, main = "Autocorrelation Function of Residuals")
```



c) Is there heteroscedasticity between error terms?

```
fitted_values <- fitted(fit.simple)
```

```r
# Create a plot of residuals vs fitted values
plot(fitted_values, residuals,
     xlab = "Fitted values",
     ylab = "Residuals",
     main = "Residuals vs Fitted Values")
```




```r
library(lmtest)
bptest(fit.simple)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  fit.simple
## BP = 15.497, df = 1, p-value = 8.262e-05
```

- In this test, the p-value is too low, so there is evidence that model has heteroscedasticity between errors.
- The BP test given a p-value of 8e-05, which is below the 0.01 threshold.
- So, we reject Ho the variance of the errors is constant - homoscedasticity and come to the conclusion that the error terms are heteroscedastic.

d) Are there outliers?

```r
lower_bound <- median(BostonData$medv) - 3 * mad(BostonData$medv, constant = 1)
print(lower_bound)
```

```
## [1] 9.2
```

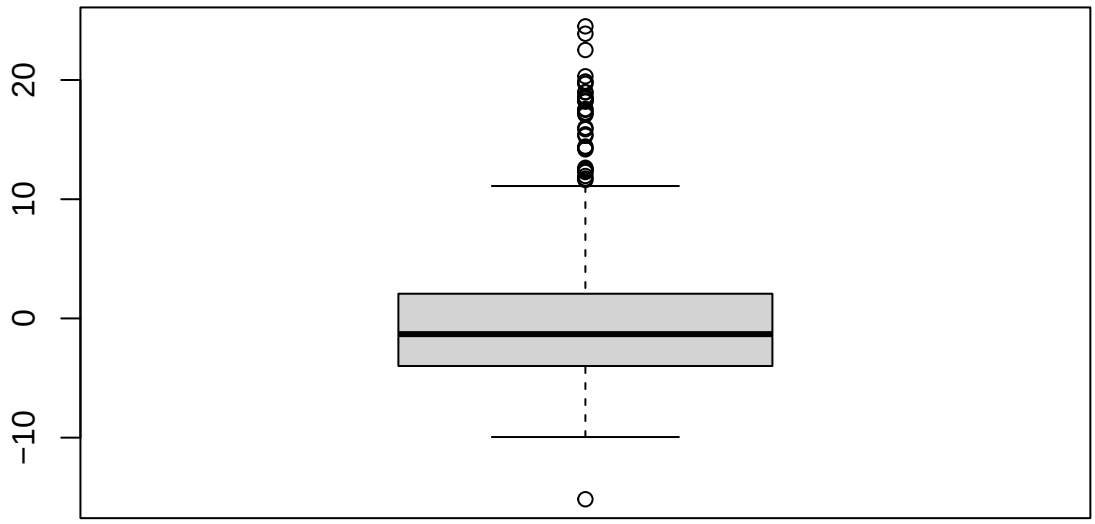
```r
upper_bound <- median(BostonData$medv) + 3 * mad(BostonData$medv, constant = 1)
print(upper_bound)
```

```
## [1] 33.2
```

```r
cd = cooks.distance(fit.simple)
res=resid(fit.simple)
pot.out <- which(cd > 4 / length(cd))
print(res[pot.out])
```

```
##          9         49         99        142        148        149        162
##  10.381136   9.117180  12.637835  12.537357   8.101117  10.151557  17.089745
##        163        164        167        187        196        203        204
##  17.270254  18.600323  18.961342  19.673879  18.267806  10.700813  17.565847
##        205        215        225        226        229        234        254
##  18.182301  17.220118  14.179363  19.844888  15.870353  17.498854  11.609334
##        257        258        262        263        268        269        281
##  12.400813  20.310412  15.443517  19.860951  22.514526  11.948315  14.418345
##        283        284        369        370        371        372        373
##  14.305808  18.448315  18.543320  18.989843  18.258305  24.500129  23.882597
##        374        375        413        415        439        506
##  12.279375  15.319533  15.999355   7.578984   6.166838 -15.167452
```

```r
print(boxplot(res))
```




```
## $stats
##           [,1]
## [1,] -9.948842
## [2,] -3.990469
## [3,] -1.318186
## [4,]  2.071434
## [5,] 11.106886
##
## $n
## [1] 506
##
## $conf
##           [,1]
## [1,] -1.743972
## [2,] -0.892401
##
## $out
##        99       142       162       163       164       167       181       187
##  12.63784  12.53736  17.08974  17.27025  18.60032  18.96134  12.42853  19.67388
##       196       204       205       215       225       226       229       234
##  18.26781  17.56585  18.18230  17.22012  14.17936  19.84489  15.87035  17.49885
##       254       257       258       262       263       268       269       281
##  11.60933  12.40081  20.31041  15.44352  19.86095  22.51453  11.94832  14.41834
##       283       284       369       370       371       372       373       374
##  14.30581  18.44832  18.54332  18.98984  18.25831  24.50013  23.88260  12.27938
##       375       410       413       506
```

```
##  15.31953  11.73814  15.99936 -15.16745
##
## $group
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##
## $names
## [1] "1"
```

So, upper limit is 33.2. the actual values of medv is 33,20,50. Out of three one value is outlier which is 50. and the residual for that one is 29.2

Q5 Multiple linear regression. Please fit a multiple linear regression model between medv and the other variables.

```r
fit.multiple <- lm(medv ~ ., data = BostonData)
summary(fit.multiple)
```

```
##
## Call:
## lm(formula = medv ~ ., data = BostonData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.8948  -2.7585  -0.4663   1.7963  26.0911
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  36.461352   5.100994   7.148 3.21e-12 ***
## X            -0.002526   0.002080  -1.215 0.225046
## crim         -0.108762   0.032855  -3.310 0.001000 **
## zn            0.048031   0.013785   3.484 0.000538 ***
## indus         0.019932   0.061468   0.324 0.745871
## chas          2.705245   0.861298   3.141 0.001786 **
## nox         -17.541602   3.822390  -4.589 5.66e-06 ***
## rm            3.839225   0.418422   9.175  < 2e-16 ***
## age          -0.001938   0.013380  -0.145 0.884866
## dis          -1.493304   0.199892  -7.471 3.68e-13 ***
## rad           0.324925   0.068111   4.771 2.43e-06 ***
## tax          -0.011598   0.003807  -3.046 0.002443 **
## ptratio      -0.947985   0.130822  -7.246 1.67e-12 ***
## black         0.009357   0.002685   3.485 0.000536 ***
## lstat        -0.526184   0.050704 -10.377  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.743 on 491 degrees of freedom
## Multiple R-squared:  0.7414, Adjusted R-squared:  0.734
## F-statistic: 100.6 on 14 and 491 DF,  p-value: < 2.2e-16
```

a) Is there a relationship between median house value and the other variables?

- From the summary results after applying the multiple regression, we can see the different outputs. The P-value can help portray which predictor is significant.
- yes,There is a relationship between medv and other variables.
- A low P-value means the variable and response variable are fairly related. Low here is typically less than 0.05. All the columns comply under this condition.
- for example rm has a coefficient of 3.83923.The median house value (medv) tends to increase by approximately 3.84 units, assuming other variables are constant.it is a positive coefficient.lstat has a coefficient of -0.52618.For each one-unit increase in lstat, the medv tends to decrease by approximately 0.53 units.it is negetive coefficient.

b) Which variables are significant and how large are the effect?

- Typically, if the P-value falls under the alpha significance level we can deem it as statistically significant. If not specified, the assumed alpha is 0.05.
- If we want to use this in a 2-sided tail test, we want to use 0.025 as the significance level per side. This leads to t-test statistic of +/- 1.96.
- But for this case we will examine the one-tailed test, it will show at a higher statistical power. As we can see, when comparing the p-value to the alpha sig value,
- the following are statistically significant: crim, zn, chas, nox, rm, dis, rad, tax, ptratio, black, lstat.rm, lstat, and dis are the most statistically significant in this case.

t value Pr(>|t|) (Intercept) 7.148 3.21e-12 ... 1 -1.215 0.225046 crim -3.310 0.001000 zn 3.484 0.000538 indus 0.324 0.745871 chas 3.141 0.001786 nox -4.589 5.66e-06 rm 9.175 < 2e-16 age -0.145 0.884866 dis -7.471 3.68e-13 rad 4.771 2.43e-06 tax -3.046 0.002443 ptratio -7.246 1.67e-12 black 3.485 0.000536 lstat -10.377 < 2e-16

c) How good this model fits the data?

- Multiple R-squared: 0.7414, Adjusted R-squared: 0.734
- Residual standard error: 4.743 on 491 degrees of freedom ,
- Residuals are the difference between the regression line and the data instances.
- These residuals are used in the model when it adjusting the line.
- A higher value R-squared value can indicate small differences between the fitted and data instances.
- Here, the adjusted R-squared 0.734 out of 1, which is pretty high so this can be one of the pieces of evidence that we can use to say that the model fit the data well.

d) Select the best subset of variables using forward selection, backward selection and mixed selection with AIC criteria. (write what they are and what is AIC criteria)

Forward selection, backward selection, and mixed are methods of feature selection. * This is used when there are features in a data set, but we don't see the necessity in including all of them. * Some of them are redundant or may not have a significant impact on the independent parameter. * So, forward selection starts at zero features and keeps adding them one by one, iterations This is done until there is no impact on the independent variable despite the presence of said additional feature. * Backward selection is having the full stack of features and, again in an iterative process, removing them one by one until the performance of the model declines. * Mixed selection, of course, is the combination of them both. We use a p-value to check this, P-value being the probability value. * Forward selection is usually less computationally expensive when compared to the backwards selection method. But, P-value can be used in both. * AIC is Akaike Information Criterion.It compares how well a model fit whilst considering the complexity of the model itself. The complexity aspect is found in the form of a penalty term. In this case, a lower AIC is more ideal.

```r
fit.null <- lm(medv~1,data = BostonData)
select.forward <- step(fit.null, scope=list(lower=fit.null, upper=fit.multiple),
                       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
## + X        1    2193.4 40523 2221.8
## + 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
## + X        1      59.1 19413 1851.5
## + 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
```

```
## + X        1    205.01 15234 1730.8
## + 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
## + X      1     20.62 13707 1679.4
## + 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
## + X      1     76.13 13153 1660.5
## + 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
## + X      1      1.09 12468 1635.4
```

```
## + 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
## + X      1     0.540 12140 1624.0
##
## 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
## + X      1     2.601 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
## + X      1     2.199 11676 1608.2
## + 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
## + X      1     0.317 11583 1606.2
##
## 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
## + X      1    70.508 11284 1595.0
## <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
## + X      1    32.937 11048 1586.2
## + indus  1     2.518 11079 1587.7
## + age    1     0.063 11081 1587.8
```

```
summary(select.forward)
```

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

```r
fit.null <- lm(medv ~ 1, data = BostonData)
select.forward <- step(fit.null, scope = list(lower = fit.null, upper = fit.multiple), direction = "forw
```

```
## 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
## + X        1    2193.4 40523 2221.8
## + 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
## + X        1      59.1 19413 1851.5
## + 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
## + X        1    205.01 15234 1730.8
```

```
## + 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
## + X      1     20.62 13707 1679.4
## + 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
## + X      1     76.13 13153 1660.5
## + 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
## + X      1      1.09 12468 1635.4
## + 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
## + X      1     0.540 12140 1624.0
##
## 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
## + X      1     2.601 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
## + X      1     2.199 11676 1608.2
## + 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
## + X      1     0.317 11583 1606.2
##
## 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
## + X      1    70.508 11284 1595.0
## <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
## + X      1    32.937 11048 1586.2
## + indus  1     2.518 11079 1587.7
## + age    1     0.063 11081 1587.8
```

```
summary(select.forward)
```

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

```
# Perform backward selection
select.backward <- step(fit.multiple, direction = "backward")
```

```
## Start:  AIC=1590.12
## medv ~ X + crim + zn + indus + chas + nox + rm + age + dis +
##     rad + tax + ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - age      1      0.47 11046 1588.2
## - indus    1      2.37 11048 1588.2
## - X        1     33.20 11079 1589.6
## <none>                 11046 1590.1
## - tax      1    208.73 11254 1597.6
## - chas     1    221.93 11268 1598.2
## - crim     1    246.53 11292 1599.3
## - zn       1    273.12 11319 1600.5
## - black    1    273.20 11319 1600.5
## - nox      1    473.78 11519 1609.4
## - rad      1    511.97 11558 1611.0
## - ptratio  1   1181.26 12227 1639.5
## - dis      1   1255.48 12301 1642.6
## - rm       1   1893.94 12940 1668.2
## - lstat    1   2422.66 13468 1688.5
##
## Step:  AIC=1588.15
## medv ~ X + crim + zn + indus + chas + nox + rm + dis + rad +
##     tax + ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - indus    1      2.37 11048 1586.2
## - X        1     32.79 11079 1587.7
## <none>                 11046 1588.2
## - tax      1    210.46 11256 1595.7
## - chas     1    221.47 11268 1596.2
## - crim     1    246.53 11293 1597.3
## - black    1    272.88 11319 1598.5
## - zn       1    278.41 11324 1598.7
## - rad      1    513.75 11560 1609.2
## - nox      1    519.51 11566 1609.4
## - ptratio  1   1193.16 12239 1638.0
## - dis      1   1362.40 12408 1645.0
## - rm       1   1970.67 13017 1669.2
## - lstat    1   2749.73 13796 1698.6
##
## Step:  AIC=1586.25
## medv ~ X + crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - X        1     32.94 11081 1585.8
## <none>                 11048 1586.2
## - chas     1    228.66 11277 1594.6
## - tax      1    236.05 11284 1595.0
## - crim     1    248.68 11297 1595.5
## - black    1    271.50 11320 1596.5
## - zn       1    276.10 11324 1596.7
## - rad      1    533.86 11582 1608.1
```

```
## - nox      1    541.12 11590 1608.5
## - ptratio  1   1199.77 12248 1636.4
## - dis      1   1458.98 12507 1647.0
## - rm       1   1975.19 13024 1667.5
## - lstat    1   2754.60 13803 1696.9
##
## 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
```

```
# Summary of backward selection
summary(select.backward)
```

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

The backwards AIC, which starts with all the features, reached the lower AIC level sooner with less iterations when compared to the forward selection. The forward selection starts with 0 features, and it started with a high AIC level, taking more iterations to reach the low value.

```
# Perform mixed selection
fit.null <- lm(medv ~ 1, data = BostonData)
fit.multiple <- lm(medv ~ ., data = BostonData)
select.mixed <- step(fit.null, scope = list(lower = fit.null, upper = fit.multiple), direction = "both"
```

```
## 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
## + X        1    2193.4 40523 2221.8
## + 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
## + X        1      59.1 19413 1851.5
## + rad      1      25.1 19447 1852.4
## + nox      1       4.8 19468 1852.9
## - lstat    1   23243.9 42716 2246.5
##
## Step:  AIC=1735.58
```

```
## medv ~ lstat + rm
##
##           Df Sum of Sq   RSS    AIC
## + ptratio  1    1711.3 13728 1678.1
## + chas     1     548.5 14891 1719.3
## + black    1     512.3 14927 1720.5
## + tax      1     425.2 15014 1723.5
## + dis      1     351.2 15088 1725.9
## + crim     1     311.4 15128 1727.3
## + X        1     205.0 15234 1730.8
## + rad      1     180.5 15259 1731.6
## + indus    1      61.1 15378 1735.6
## <none>                 15439 1735.6
## + zn       1      56.6 15383 1735.7
## + age      1      20.2 15419 1736.9
## + nox      1      14.9 15424 1737.1
## - rm       1    4033.1 19472 1851.0
## - lstat    1    6622.6 22062 1914.2
##
## Step:  AIC=1678.13
## medv ~ lstat + rm + ptratio
##
##           Df Sum of Sq   RSS    AIC
## + dis      1     499.1 13229 1661.4
## + black    1     389.7 13338 1665.6
## + chas     1     378.0 13350 1666.0
## + crim     1     122.5 13606 1675.6
## + age      1      66.2 13662 1677.7
## <none>                 13728 1678.1
## + tax      1      44.4 13684 1678.5
## + nox      1      24.8 13703 1679.2
## + X        1      20.6 13707 1679.4
## + zn       1      15.0 13713 1679.6
## + rad      1       6.1 13722 1679.9
## + indus    1       0.8 13727 1680.1
## - ptratio  1    1711.3 15439 1735.6
## - rm       1    3074.3 16802 1778.4
## - lstat    1    5013.6 18742 1833.7
##
## Step:  AIC=1661.39
## medv ~ lstat + rm + ptratio + dis
##
##           Df Sum of Sq   RSS    AIC
## + nox      1     759.6 12469 1633.5
## + black    1     502.6 12726 1643.8
## + chas     1     267.4 12962 1653.1
## + indus    1     242.6 12986 1654.0
## + tax      1     240.3 12989 1654.1
## + crim     1     233.5 12995 1654.4
## + zn       1     144.8 13084 1657.8
## + X        1      76.1 13153 1660.5
## + age      1      61.4 13168 1661.0
## <none>                 13229 1661.4
## + rad      1      22.4 13206 1662.5
```

```
## - dis      1     499.1 13728 1678.1
## - ptratio  1    1859.3 15088 1725.9
## - rm       1    2622.6 15852 1750.9
## - lstat    1    5349.2 18578 1831.2
##
## Step:  AIC=1633.47
## medv ~ lstat + rm + ptratio + dis + nox
##
##           Df Sum of Sq   RSS    AIC
## + chas     1     328.3 12141 1622.0
## + black    1     311.8 12158 1622.7
## + zn       1     151.7 12318 1629.3
## + crim     1     141.4 12328 1629.7
## + rad      1      53.5 12416 1633.3
## <none>                 12469 1633.5
## + indus    1      17.1 12452 1634.8
## + tax      1      10.5 12459 1635.0
## + X        1       1.1 12468 1635.4
## + age      1       0.2 12469 1635.5
## - nox      1     759.6 13229 1661.4
## - dis      1    1233.8 13703 1679.2
## - ptratio  1    2116.5 14586 1710.8
## - rm       1    2546.2 15016 1725.5
## - lstat    1    3664.3 16134 1761.8
##
## Step:  AIC=1621.97
## medv ~ lstat + rm + ptratio + dis + nox + chas
##
##           Df Sum of Sq   RSS    AIC
## + black    1     272.8 11868 1612.5
## + zn       1     164.4 11977 1617.1
## + crim     1     116.3 12025 1619.1
## + rad      1      58.6 12082 1621.5
## <none>                 12141 1622.0
## + indus    1      26.3 12115 1622.9
## + tax      1       4.2 12137 1623.8
## + age      1       2.3 12139 1623.9
## + X        1       0.5 12140 1624.0
## - chas     1     328.3 12469 1633.5
## - nox      1     820.4 12962 1653.1
## - dis      1    1146.8 13288 1665.6
## - ptratio  1    1924.9 14066 1694.4
## - rm       1    2480.7 14622 1714.0
## - lstat    1    3509.3 15650 1748.5
##
## Step:  AIC=1612.47
## medv ~ lstat + rm + ptratio + dis + nox + chas + black
##
##           Df Sum of Sq   RSS    AIC
## + zn       1    189.94 11678 1606.3
## + rad      1    144.32 11724 1608.3
## + crim     1     55.63 11813 1612.1
## <none>                 11868 1612.5
## + indus    1     15.58 11853 1613.8
```

```
## + age      1      9.45 11859 1614.1
## + tax      1      2.70 11866 1614.4
## + X        1      2.60 11866 1614.4
## - black    1    272.84 12141 1622.0
## - chas     1    289.27 12158 1622.7
## - nox      1    626.85 12495 1636.5
## - dis      1   1103.33 12972 1655.5
## - ptratio  1   1804.30 13672 1682.1
## - rm       1   2658.21 14526 1712.7
## - lstat    1   2991.55 14860 1724.2
##
## Step:  AIC=1606.31
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn
##
##           Df Sum of Sq   RSS    AIC
## + crim     1     94.71 11584 1604.2
## + rad      1     93.61 11585 1604.2
## <none>                 11678 1606.3
## + indus    1     16.05 11662 1607.6
## + tax      1      3.95 11674 1608.1
## + X        1      2.20 11676 1608.2
## + age      1      1.49 11677 1608.2
## - zn       1    189.94 11868 1612.5
## - black    1    298.37 11977 1617.1
## - chas     1    300.42 11979 1617.2
## - nox      1    627.62 12306 1630.8
## - dis      1   1276.45 12955 1656.8
## - ptratio  1   1364.63 13043 1660.2
## - rm       1   2384.55 14063 1698.3
## - lstat    1   3052.50 14731 1721.8
##
## Step:  AIC=1604.19
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn +
##     crim
##
##           Df Sum of Sq   RSS    AIC
## + rad      1    228.60 11355 1596.1
## <none>                 11584 1604.2
## + indus    1     15.77 11568 1605.5
## + age      1      2.47 11581 1606.1
## + tax      1      1.31 11582 1606.1
## + X        1      0.32 11583 1606.2
## - crim     1     94.71 11678 1606.3
## - black    1    222.18 11806 1611.8
## - zn       1    229.02 11813 1612.1
## - chas     1    284.34 11868 1614.5
## - nox      1    578.44 12162 1626.8
## - ptratio  1   1192.90 12776 1651.8
## - dis      1   1345.70 12929 1657.8
## - rm       1   2419.57 14003 1698.2
## - lstat    1   2753.42 14337 1710.1
##
## Step:  AIC=1596.1
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn +
```

```
##     crim + rad
##
##           Df Sum of Sq   RSS    AIC
## + tax      1    273.62 11081 1585.8
## + X        1     70.51 11284 1595.0
## <none>                 11355 1596.1
## + indus    1     33.89 11321 1596.6
## + age      1      0.10 11355 1598.1
## - zn       1    171.14 11526 1601.7
## - rad      1    228.60 11584 1604.2
## - crim     1    229.70 11585 1604.2
## - chas     1    272.67 11628 1606.1
## - black    1    295.78 11651 1607.1
## - nox      1    785.16 12140 1627.9
## - dis      1   1341.37 12696 1650.6
## - ptratio  1   1419.77 12775 1653.7
## - rm       1   2182.57 13538 1683.1
## - lstat    1   2785.28 14140 1705.1
##
## 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
## + X        1     32.94 11048 1586.2
## + indus    1      2.52 11079 1587.7
## + age      1      0.06 11081 1587.8
## - 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
```

```
# Summary of mixed selection
summary(select.mixed)
```

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

e) Do different selection algorithms find the same subset? Which variables are selected?

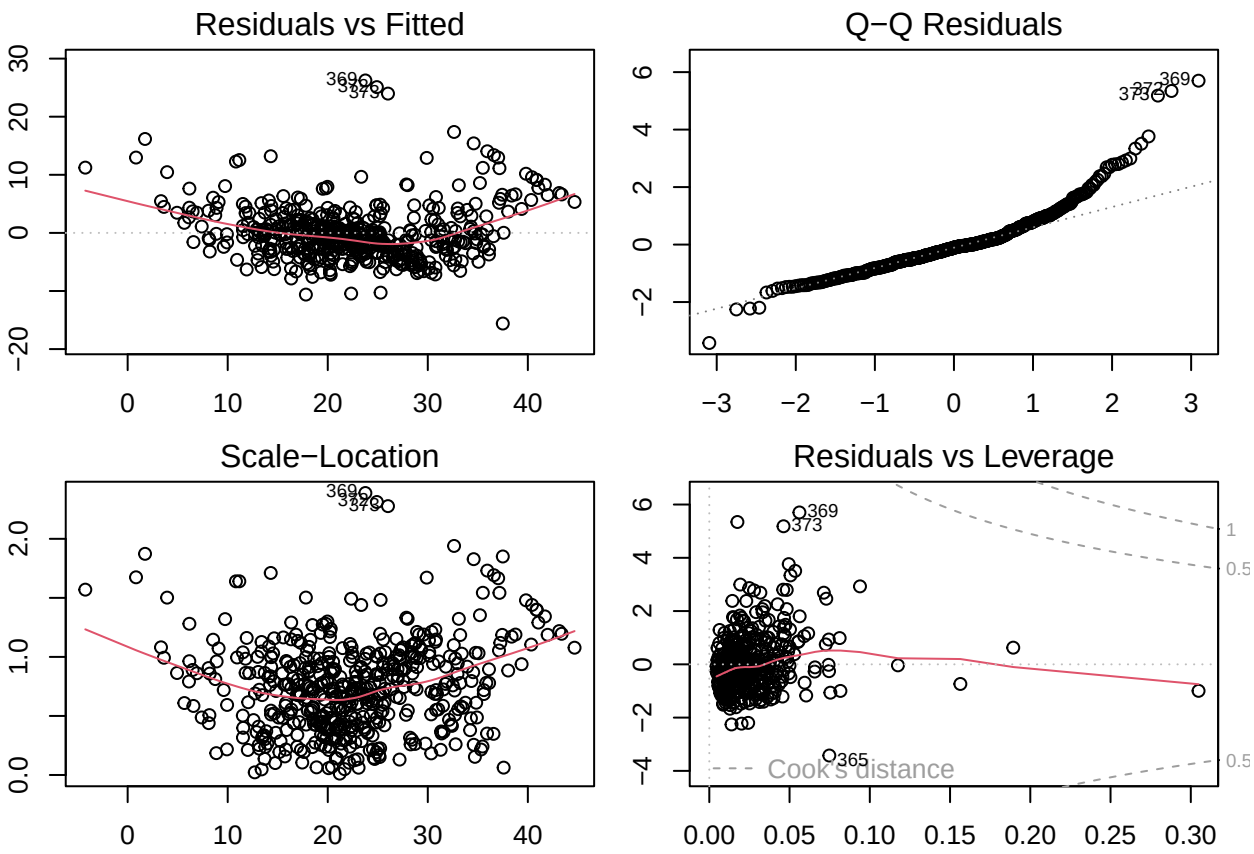
No, different algorithms do not always find the same subset. There is no guarantee. The choice of variables is also dependent on the algorithm itself.

f) Does variable selection improve the R-square? How about the adjusted R-square?

Variable selection in general impacts the R-square value, it can change up or down based on the selection method. For forward and backward selection, the R-squares increases. The adjusted R-squared may show an increase in forward method, but it can decrease when the added variables no longer contributes to the model. The backward selection usually increases the adjusted R-value.

Q6 Residual plot. Please plot the residual plots of multiple linear regression model fitted in Problem 5 and answer the following question.

```r
par(mfrow = c(2, 2), mar = c(2, 2, 2, 1))
# Create individual residual plots
plot(select.mixed)
```



Q7 Use non-linear transformation to include lstat^2.

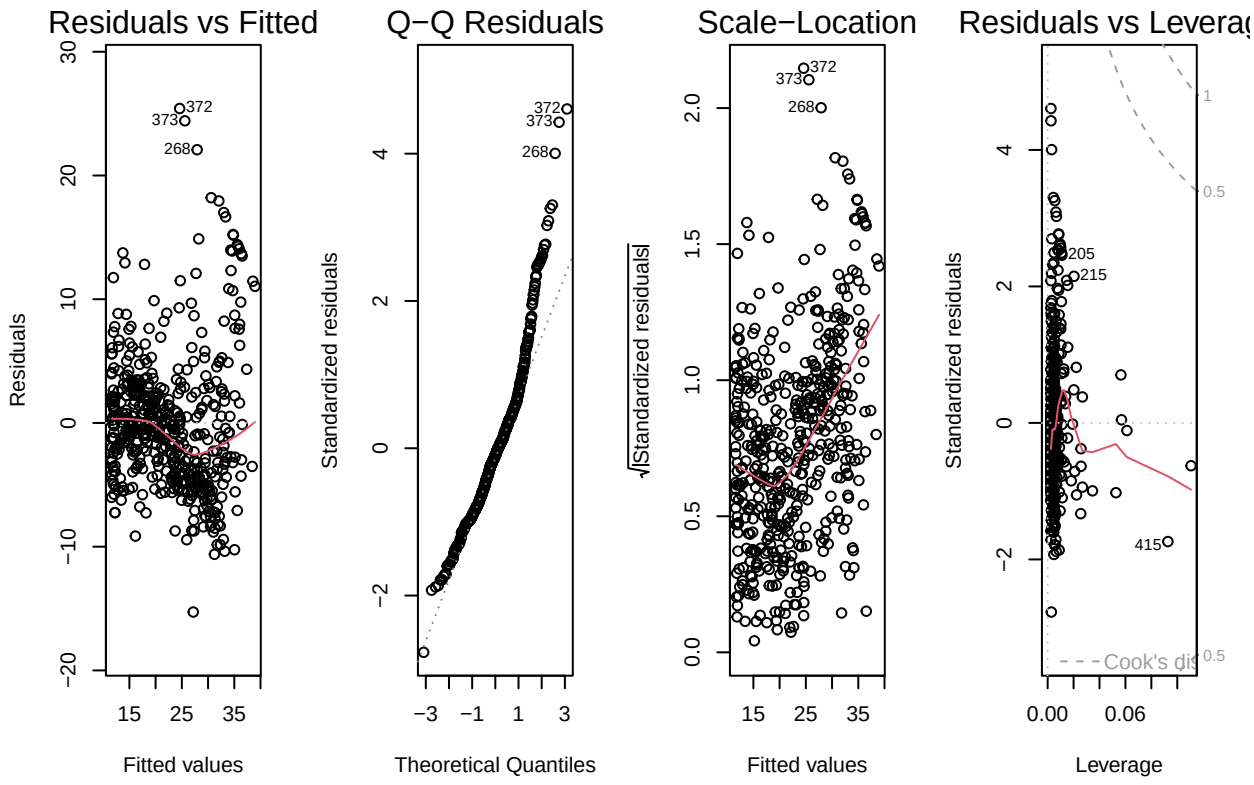
Startlingly, the idea behind non-linear transformation is to transform the non-linearity to linear.

```
fit.nonlinear <- lm(medv~ lstat + I(lstat^2), data = BostonData)
summary(fit.nonlinear)
```

```
##
## Call:
## lm(formula = medv ~ lstat + I(lstat^2), data = BostonData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2834  -3.8313  -0.5295   2.3095  25.4148
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 42.862007   0.872084   49.15   <2e-16 ***
## lstat       -2.332821   0.123803  -18.84   <2e-16 ***
## I(lstat^2)   0.043547   0.003745   11.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.524 on 503 degrees of freedom
## Multiple R-squared:  0.6407, Adjusted R-squared:  0.6393
```

```
## F-statistic: 448.5 on 2 and 503 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(1,4))
plot(fit.nonlinear)
```



a) Is the model improved?

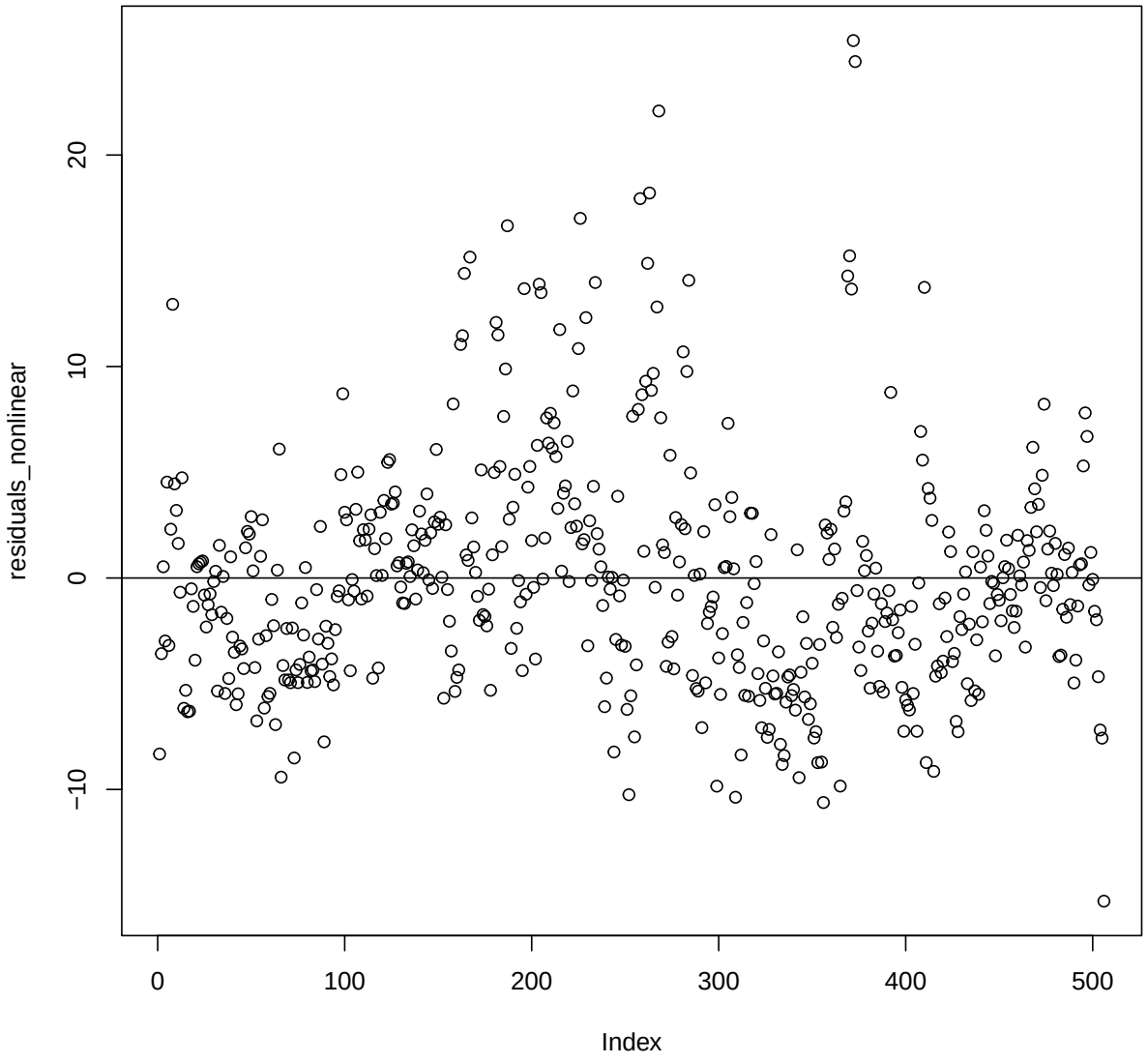
- The residual standard error is now 5.524.
- The multiple R-square value is 0.6407 and the adjusted R-squared is 0.6393.
- When comparing both the plots to each other we can clearly tell that the non-linear transformation had a positive impact on the model and it did in fact better it.
- The data points fit closer and better to the regression line particularly in direct comparison with the prior.

b) Is the nonlinear effect significant?

Looking at the p-value we can see that it is 2.23e-16, which is considerably less than 0.05, so it is in fact statistically significant. It is roughly the same, however, so we can say for this particular case it did not affect it too much, at least in terms of the p-value statistical difference. Regardless, the model better fits the data.

c) Use the residual plot to see if the nonlinear relationship is solved.

```
#Residual plot to see if non linear relationship is solved
residuals_nonlinear <- residuals(fit.nonlinear)
plot(residuals_nonlinear)
abline(h=0)
```



Typically, when the pattern is more random in the new model versus the original, then it means the non-linear transformation has solved some of the relationship. In this case it has.

Q8 Include the interaction term lstat X black.(so what are these processes where we add non linearity to a lineary model.why are they important)

```
fit.interact <- lm(medv ~ . + lstat:black, data = BostonData)
summary(fit.interact)
```

```
##
## Call:
## lm(formula = medv ~ . + lstat:black, data = BostonData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.5087  -2.6967  -0.4767   1.8172  25.2041
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.011e+01  5.590e+00   5.386 1.12e-07 ***
## X           -2.861e-03  2.070e-03  -1.382 0.167660
## crim        -1.084e-01  3.265e-02  -3.320 0.000967 ***
## zn           4.709e-02  1.370e-02   3.437 0.000638 ***
## indus        3.405e-02  6.130e-02   0.555 0.578892
## chas         2.745e+00  8.560e-01   3.207 0.001431 **
## nox         -1.723e+01  3.800e+00  -4.535 7.24e-06 ***
## rm           3.613e+00  4.242e-01   8.519  < 2e-16 ***
## age          2.623e-03  1.340e-02   0.196 0.844916
## dis         -1.469e+00  1.988e-01  -7.390 6.38e-13 ***
## rad          3.243e-01  6.768e-02   4.792 2.20e-06 ***
## tax         -1.116e-02  3.787e-03  -2.946 0.003372 **
## ptratio     -9.654e-01  1.302e-01  -7.417 5.31e-13 ***
## black        3.006e-02  8.135e-03   3.695 0.000245 ***
## lstat       -1.790e-01  1.384e-01  -1.294 0.196237
## black:lstat -1.052e-03  3.906e-04  -2.694 0.007303 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.713 on 490 degrees of freedom
## Multiple R-squared:  0.7452, Adjusted R-squared:  0.7374
## F-statistic: 95.54 on 15 and 490 DF,  p-value: < 2.2e-16
```

```
AIC_mul_new <- AIC(fit.interact)
AIC_mul_new
```

```
## [1] 3022.651
```

```
summary(fit.interact)$r.squared
```

```
## [1] 0.7451937
```

a) Is the model improved?

- The fit.interact model has an adjusted R-squared value of 0.737.

- It is slightly less than the previous model adjusted R-squared value.
- AIC value increases.
- So adding this non linear term it may not be justified to include the transformation. it is not explicitly improved in explaining the variance in 'medv'.

b) Is the interaction effect significant?

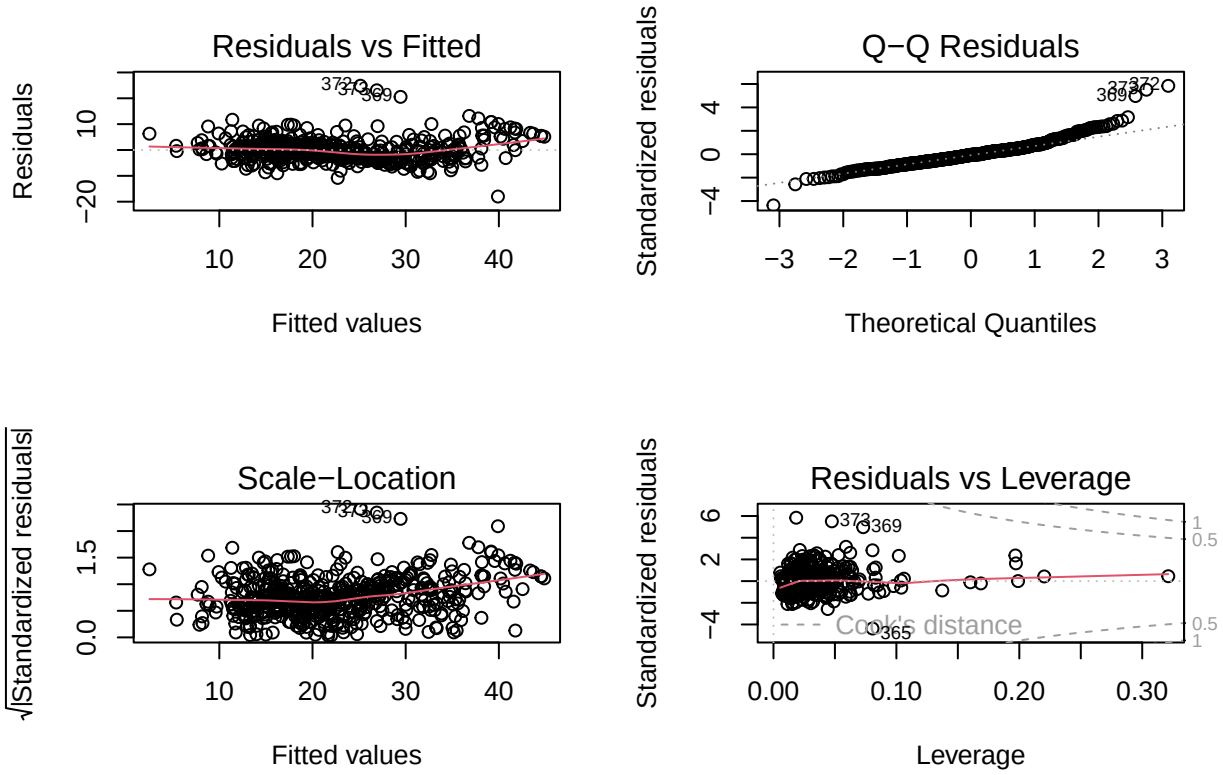
- Yes, The interaction effect is significant
- The interaction effect 'black:lstat' has a coefficient of approximately -0.001052 with a p-value of 0.007303, indicating significance at the 1% level (denoted by **).
- Based on the p-value criterion, the interaction effect appears to be statistically significant.

c) Include both nonlinear term in Q7 and interaction term, answer a) and b).

```
fit.interact.nonlinear <- lm(medv ~ . + I(lstat^2) - indus - age + lstat:black, data = BostonData)
summary(fit.interact.nonlinear)
```

```
##
## Call:
## lm(formula = medv ~ . + I(lstat^2) - indus - age + lstat:black,
##     data = BostonData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18.0007  -2.6072  -0.2572   1.8976  24.8574
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.461e+01  5.297e+00   8.422 4.11e-16 ***
## X           -2.459e-03  1.865e-03  -1.319 0.187923
## crim        -1.513e-01  3.003e-02  -5.040 6.54e-07 ***
## zn           2.300e-02  1.257e-02   1.829 0.067961 .
## chas         2.569e+00  7.753e-01   3.314 0.000989 ***
## nox         -1.368e+01  3.232e+00  -4.233 2.75e-05 ***
## rm           3.294e+00  3.762e-01   8.756  < 2e-16 ***
## dis         -1.357e+00  1.691e-01  -8.024 7.61e-15 ***
## rad          2.934e-01  5.942e-02   4.937 1.09e-06 ***
## tax         -9.041e-03  3.117e-03  -2.901 0.003891 **
## ptratio     -7.793e-01  1.184e-01  -6.583 1.19e-10 ***
## black        1.239e-03  7.873e-03   0.157 0.874970
## lstat       -1.834e+00  2.079e-01  -8.822  < 2e-16 ***
## I(lstat^2)   3.424e-02  3.431e-03   9.980  < 2e-16 ***
## black:lstat  3.504e-04  3.778e-04   0.927 0.354175
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.294 on 491 degrees of freedom
## Multiple R-squared:  0.788,  Adjusted R-squared:  0.782
## F-statistic: 130.4 on 14 and 491 DF,  p-value: < 2.2e-16
```

```
par(mfrow = c(2, 2))
plot(fit.interact.nonlinear)
```



a) Is the model improved?

- compared to the previous models the adjusted R squared value has significantly increased.
- now the Adjusted R squared value is 0.782.
- this indicates the improvement in variability in medv when compared to previous models

b) Is the interaction effect significant?

- The coefficient for the 'lstat:black' interaction term is 3.50e-04 with a p-value of 0.354175.
- Here ,The p-value is > the commonly used significance level of 0.05, indicating that the 'lstat:black' interaction term is not statistically significant at a five percent significance level.
- The interaction effect is not appear to be statistically significant in predicting 'medv'.
- The model's adjusted R-squared has improved, indicating a better fit compared to the previous models.
- The interaction term 'lstat:black' seems not to be statistically significant in this model.

Q9 Apply K-nearest neighbor regression model on this dataset and find the optimal K.

Step-1: Randomly separate the dataset into training and test data

we used sample function to randomly reorder the samples and use first 400 samples as training and the remaining samples as test.

```
# by using sample method, generating the random permutation of numbers from first to the last number of
randid <- sample(c(1:nrow(BostonData)))
#dividing to training data by using randid function first 400 rows are assigned to the training data
Boston.train <- BostonData[randid[c(1:400)],]
# dividing in to the test data, from 401 to 506 rows data is assigned to the test set.
Boston.test <- BostonData[randid[c(401:506)],]
# from the training data, Extracting the lstat variable data and assigning it to the lstat.train variab
lstat.train <- Boston.train['lstat']
# Extracting the medv variable data from the train set and assigning it to the medv.train variable
medv.train <- Boston.train['medv']
#In the same way from the test data set extracting the lstat and medv column data separately and assign
lstat.test <-Boston.test['lstat']
medv.test <- Boston.test['medv']
```

Step-2: Use training data to predict medv values at test data and select the best K

Predict the medv under K=1,5,10,50,100,250

```
install.packages("FNN")
```

```
##
## The downloaded binary packages are in
##  /var/folders/km/6fl99qs97wq655h4rwdy0hjh0000gn/T//RtmpWMWIlQ/downloaded_packages
```

```
library(FNN)
pred_001 = knn.reg(train = lstat.train, test = lstat.test, y = medv.train, k = 1)
pred_005 = knn.reg(train = lstat.train, test = lstat.test, y = medv.train, k = 5)
pred_010 = knn.reg(train = lstat.train, test = lstat.test, y = medv.train, k = 10)
pred_050 = knn.reg(train = lstat.train, test = lstat.test, y = medv.train, k = 50)
pred_100 = knn.reg(train = lstat.train, test = lstat.test, y = medv.train, k = 100)
pred_200 = knn.reg(train = lstat.train, test = lstat.test, y = medv.train, k = 200)
pred <- c(pred_001,pred_005,pred_010,pred_050,pred_100,pred_200)
pred
```

```
## $call
## knn.reg(train = lstat.train, test = lstat.test, y = medv.train,
##     k = 1)
##
## $k
## [1] 1
##
## $n
## [1] 106
##
## $pred
##   [1] 20.5 18.5 28.0 14.1 18.2 50.0 21.9 11.7 50.0  7.2 14.9 37.6 27.5 19.5 23.7
##  [16] 26.6 25.3 24.4 43.5 21.7  5.6 24.4 22.2 43.8 29.9 13.8 23.2 33.4 50.0 17.4
##  [31] 18.2 15.6 21.4 50.0 10.2 27.9 36.2 17.2 20.5 19.7 28.0 14.9 15.7 16.8 50.0
##  [46] 43.8 43.8  8.4 28.4 24.6 29.9 28.4 21.7 45.4 23.7 13.6 24.6 18.4 50.0 20.9
##  [61] 14.2 26.6 18.2 20.4 13.1 22.5 21.6 14.9 24.4 21.2 16.8 24.6 16.8  7.5 13.4
##  [76] 13.0 13.8 43.8 24.5 35.2 12.8 50.0 36.4 21.2 19.9 10.2 23.1 35.2 19.1 36.2
```

```
##  [91] 14.9 28.2 21.6  9.6 21.7 18.7 16.5 16.1 22.9 20.0 34.7 14.9 37.6 18.9 20.8
## [106] 50.0
##
## $residuals
## NULL
##
## $PRESS
## NULL
##
## $R2Pred
## NULL
##
## $call
## knn.reg(train = lstat.train, test = lstat.test, y = medv.train,
##     k = 5)
##
## $k
## [1] 5
##
## $n
## [1] 106
##
## $pred
##   [1] 20.46 20.76 30.90 15.48 20.64 43.96 28.06 15.00 28.60 11.88 15.24 42.10
##  [13] 25.64 16.50 13.28 31.06 27.06 25.26 41.12 15.56 11.50 30.52 19.02 42.80
##  [25] 25.58 12.68 16.28 25.70 43.16 18.06 20.86 16.60 19.48 41.12 13.34 21.62
##  [37] 30.90 19.36 26.36 19.84 30.90 17.44 11.20 20.94 32.96 39.42 39.42 13.70
##  [49] 23.22 26.72 26.90 23.00 15.22 48.12 33.92 14.04 26.72 20.72 43.16 23.74
##  [61] 16.48 31.06 20.64 15.78 12.92 15.78 23.36 17.44 24.68 20.72 20.94 26.72
##  [73] 16.98 13.26 14.76 15.04 16.68 39.42 19.84 29.34 13.70 33.20 43.16 19.76
##  [85] 20.08 16.50 25.58 29.34 16.62 25.70 15.24 28.66 20.38 15.78 20.76 18.98
##  [97] 22.84 19.10 19.48 20.94 39.92 15.24 42.10 20.72 20.34 28.60
##
## $residuals
## NULL
##
## $PRESS
## NULL
##
## $R2Pred
## NULL
##
## $call
## knn.reg(train = lstat.train, test = lstat.test, y = medv.train,
##     k = 10)
##
## $k
## [1] 10
##
## $n
## [1] 106
##
## $pred
##   [1] 20.70 19.97 31.19 15.74 20.32 44.69 28.31 16.00 28.00 13.23 16.69 41.92
```

```
##  [13] 27.72 15.87 11.54 28.37 27.59 27.60 41.92 15.53 12.23 26.67 19.80 41.93
##  [25] 25.05 12.25 17.22 28.03 41.92 18.44 20.32 16.91 19.86 40.76 14.24 21.03
##  [37] 28.47 19.70 27.03 20.15 31.19 16.29 12.28 18.99 31.64 43.12 43.12 13.79
##  [49] 22.46 27.35 27.85 21.39 15.53 42.38 31.93 13.81 27.35 19.39 41.92 22.73
##  [61] 17.30 28.37 20.32 16.21 12.55 16.74 24.95 16.29 26.94 19.39 18.99 27.35
##  [73] 18.99 11.49 13.21 16.43 17.65 43.12 20.22 28.31 13.21 31.75 44.69 20.77
##  [85] 20.89 17.09 25.95 28.31 16.68 28.55 16.69 29.25 21.70 16.74 19.97 20.10
##  [97] 22.87 19.65 19.63 18.99 38.63 16.69 41.92 19.81 20.32 28.00
##
## $residuals
## NULL
##
## $PRESS
## NULL
##
## $R2Pred
## NULL
##
## $call
## knn.reg(train = lstat.train, test = lstat.test, y = medv.train,
##     k = 50)
##
## $k
## [1] 50
##
## $n
## [1] 106
##
## $pred
##   [1] 22.892 22.990 30.054 15.818 23.150 37.196 27.400 16.460 23.016 15.618
##  [11] 17.162 37.196 27.386 16.644 12.378 32.600 28.204 25.798 37.196 16.682
##  [21] 12.636 25.798 20.580 36.830 27.250 12.586 15.884 26.952 37.196 19.170
##  [31] 23.086 16.460 20.314 37.196 14.134 21.012 29.978 20.592 23.188 19.996
##  [41] 30.136 15.884 12.636 19.328 30.882 36.830 36.830 13.976 24.170 27.108
##  [51] 27.172 24.050 16.740 36.390 30.210 15.122 27.108 20.580 37.196 24.106
##  [61] 17.340 32.600 23.140 18.270 13.512 16.538 25.812 15.884 25.798 20.580
##  [71] 19.328 27.108 19.328 12.792 13.512 15.822 18.458 36.830 19.944 27.400
##  [81] 13.694 30.650 37.196 22.734 19.610 18.028 27.652 27.400 16.288 23.106
##  [91] 17.200 26.576 23.286 16.538 22.990 20.314 24.400 20.492 20.546 19.162
## [101] 36.390 17.200 37.196 20.064 23.086 23.058
##
## $residuals
## NULL
##
## $PRESS
## NULL
##
## $R2Pred
## NULL
##
## $call
## knn.reg(train = lstat.train, test = lstat.test, y = medv.train,
##     k = 100)
##
```

```
## $k
## [1] 100
##
## $n
## [1] 106
##
## $pred
##   [1] 22.677 22.235 30.945 15.631 22.555 32.221 27.611 17.260 23.773 15.450
##  [11] 17.901 32.221 26.643 17.724 13.976 31.878 30.200 26.036 32.221 17.151
##  [21] 14.108 26.107 20.377 32.221 27.037 13.976 16.328 28.164 32.221 18.904
##  [31] 22.555 17.570 20.184 32.221 14.983 20.869 31.069 20.447 23.806 20.110
##  [41] 31.069 16.389 13.989 18.725 31.878 32.221 32.221 14.857 24.326 28.269
##  [51] 26.696 24.357 17.260 32.221 31.158 15.273 28.269 20.387 32.221 24.328
##  [61] 18.070 31.878 22.555 18.131 14.485 16.579 25.888 16.389 26.016 20.387
##  [71] 18.725 28.269 18.722 14.194 14.662 15.973 18.416 32.221 20.009 27.611
##  [81] 14.789 31.878 32.221 22.163 19.381 18.101 27.091 27.611 16.579 23.742
##  [91] 17.985 26.023 24.229 16.579 22.233 20.193 24.418 20.535 20.566 18.725
## [101] 32.221 17.901 32.221 20.189 22.667 23.736
##
## $residuals
## NULL
##
## $PRESS
## NULL
##
## $R2Pred
## NULL
##
## $call
## knn.reg(train = lstat.train, test = lstat.test, y = medv.train,
##     k = 200)
##
## $k
## [1] 200
##
## $n
## [1] 106
##
## $pred
##   [1] 23.4880 23.1390 28.1155 17.5745 23.3145 28.1155 28.1155 18.1775 24.1175
##  [10] 17.4170 18.7970 28.1155 27.8100 18.5930 16.6125 28.1155 28.1155 26.9215
##  [19] 28.1155 18.0415 16.6125 26.9550 20.7565 28.1155 27.8100 16.6125 17.7170
##  [28] 28.1155 28.1155 19.7995 23.3405 18.5060 20.4385 28.1155 16.8585 21.7500
##  [37] 28.1155 20.7935 24.2295 20.0845 28.1155 17.7170 16.6125 19.6865 28.1155
##  [46] 28.1155 28.1155 16.8585 24.8415 28.1155 27.8100 24.7925 18.0415 28.1155
##  [55] 28.1155 17.1695 28.1155 20.7420 28.1155 25.1755 19.0650 28.1155 23.3145
##  [64] 19.2920 16.7115 17.9380 26.6105 17.7170 26.9215 20.6645 19.6865 28.1155
##  [73] 19.6865 16.6125 16.7115 17.6005 19.4345 28.1155 20.0595 28.1155 16.7525
##  [82] 28.1155 28.1155 22.9650 19.9635 19.1390 27.8100 28.1155 17.9380 24.2295
##  [91] 18.9830 27.2175 24.3035 17.9380 23.1390 20.4385 25.3695 20.9945 21.2015
## [100] 19.6045 28.1155 18.7970 28.1155 20.2635 23.3405 24.1175
##
## $residuals
## NULL
```

```
##
## $PRESS
## NULL
##
## $R2Pred
## NULL
```

```
MSE_001 = sum((pred_001$pred-medv.test)^2)/106
cat("for k value",1,"MSE is",MSE_001,"\n")
```

```
## for k value 1 MSE is 61.30651
```

```
MSE_005 = sum((pred_005$pred-medv.test)^2)/106
cat("for k value",5,"MSE is",MSE_005,"\n")
```

```
## for k value 5 MSE is 28.01965
```

```
MSE_010 = sum((pred_010$pred-medv.test)^2)/106
cat("for k value",10,"MSE is",MSE_010,"\n")
```

```
## for k value 10 MSE is 29.30078
```

```
MSE_050 = sum((pred_050$pred-medv.test)^2)/106
cat("for k value",50,"MSE is",MSE_050,"\n")
```

```
## for k value 50 MSE is 25.43017
```

```
MSE_100 = sum((pred_100$pred-medv.test)^2)/106
cat("for k value",100,"MSE is",MSE_100,"\n")
```

```
## for k value 100 MSE is 32.82738
```

```
MSE_200 = sum((pred_200$pred-medv.test)^2)/106
cat("for k value",200,"MSE is",MSE_200,"\n")
```

```
## for k value 200 MSE is 46.2241
```

Out of all, K value is very less for 50. By observing above MSE values we can say that k =50 is the optimal k value.