Lab 1: Regression Analysis on Boston Dataset

Sofia Hein Machado

2024-06-24

##Q1 Load the data. Please download the Boston Dataset from Blackboard and read it in R Use read.csv function to load the data, row.names = 1 means the first column will be used as row names and header = T means the first row will be used as variable names.

```
BostonData <- read.csv("~/Desktop/UH courses/INDE 4364/Boston.csv", row.names = 1, heade
r = T)
head(BostonData) #check the first few rows and columns of the dataset
```

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

```
ncol(BostonData) #check the number of columns(variables) in the dataset
```

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

```
nrow(BostonData) #check the number of rows (samples) in the dataset
```

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

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

14 variables, they are all quantitative variables.

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

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

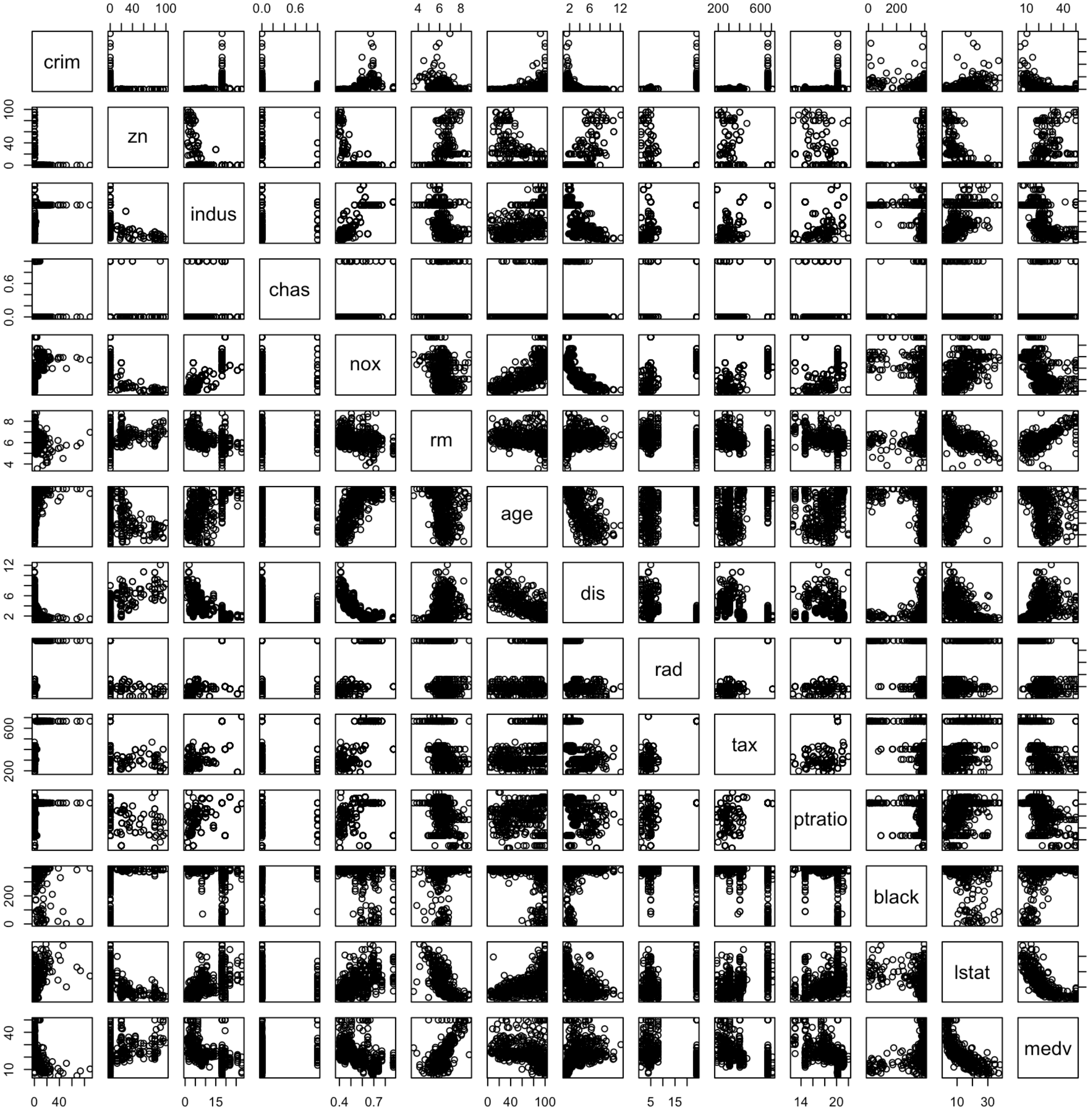
```
summary (BostonData) #statistics of each variable
```

```
##       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       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv      
##  Min.   : 1.73   Min.   : 5.00  
##  1st Qu.: 6.95   1st Qu.:17.02  
##  Median :11.36   Median :21.20  
##  Mean   :12.65   Mean   :22.53  
##  3rd Qu.:16.95   3rd Qu.:25.00  
##  Max.   :37.97   Max.   :50.00
```

Q2 Data visualization.

Please use the scatterplots to visualize this dataset.

```
pairs(BostonData, pch = 21)
```

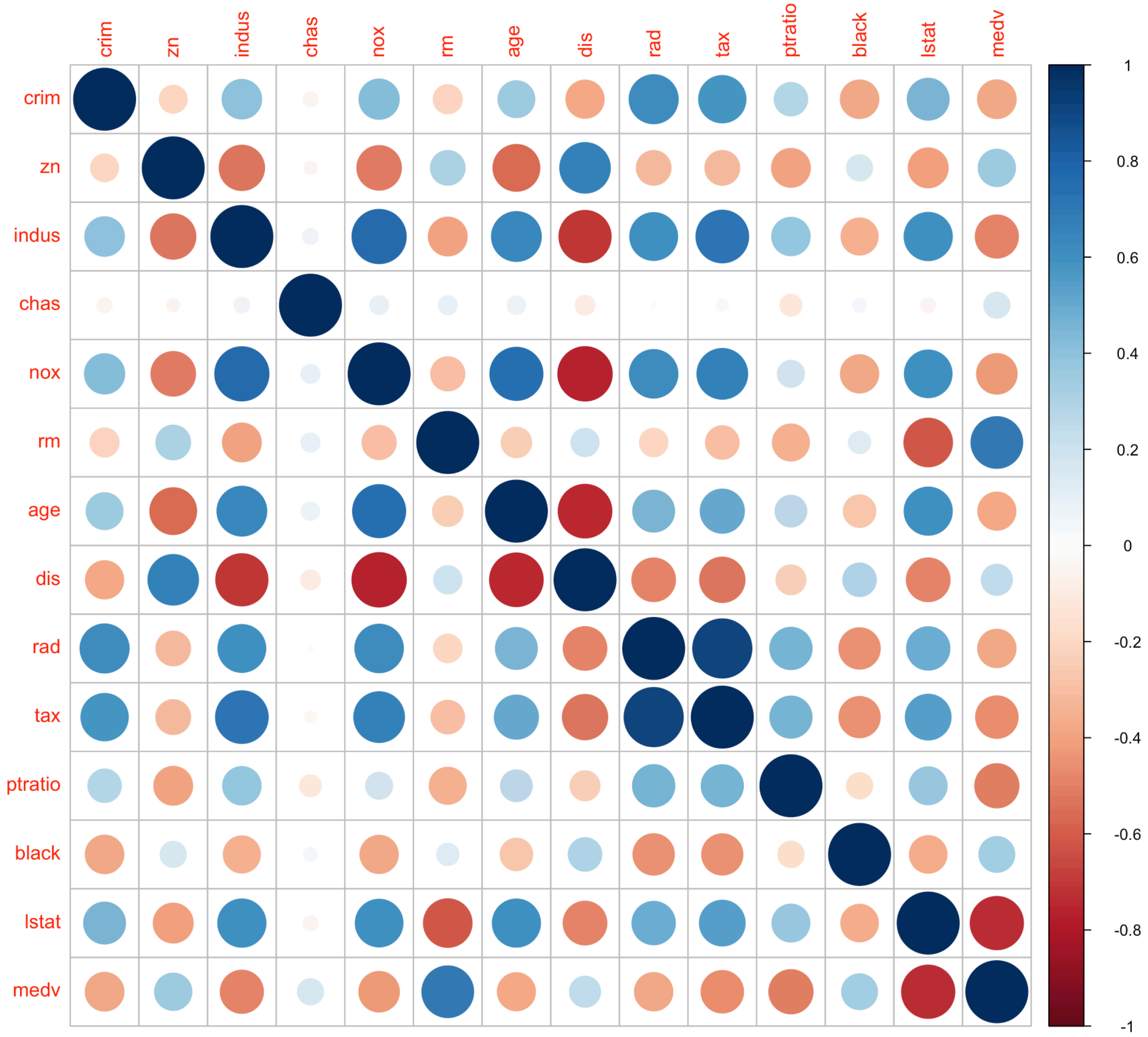


Correlation plot using package “corrplot”

```
library(corrplot) #load the package to current environment
```

```
## corrplot 0.92 loaded
```

```
Corr <- cor(BostonData) #calculate the correlation coefficient matrix of variables
corrplot(Corr,method = 'circle') #circle, square, ellipse, pie
```



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
```

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

Yes, as the p-value of t-test on the coefficient is significant.

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

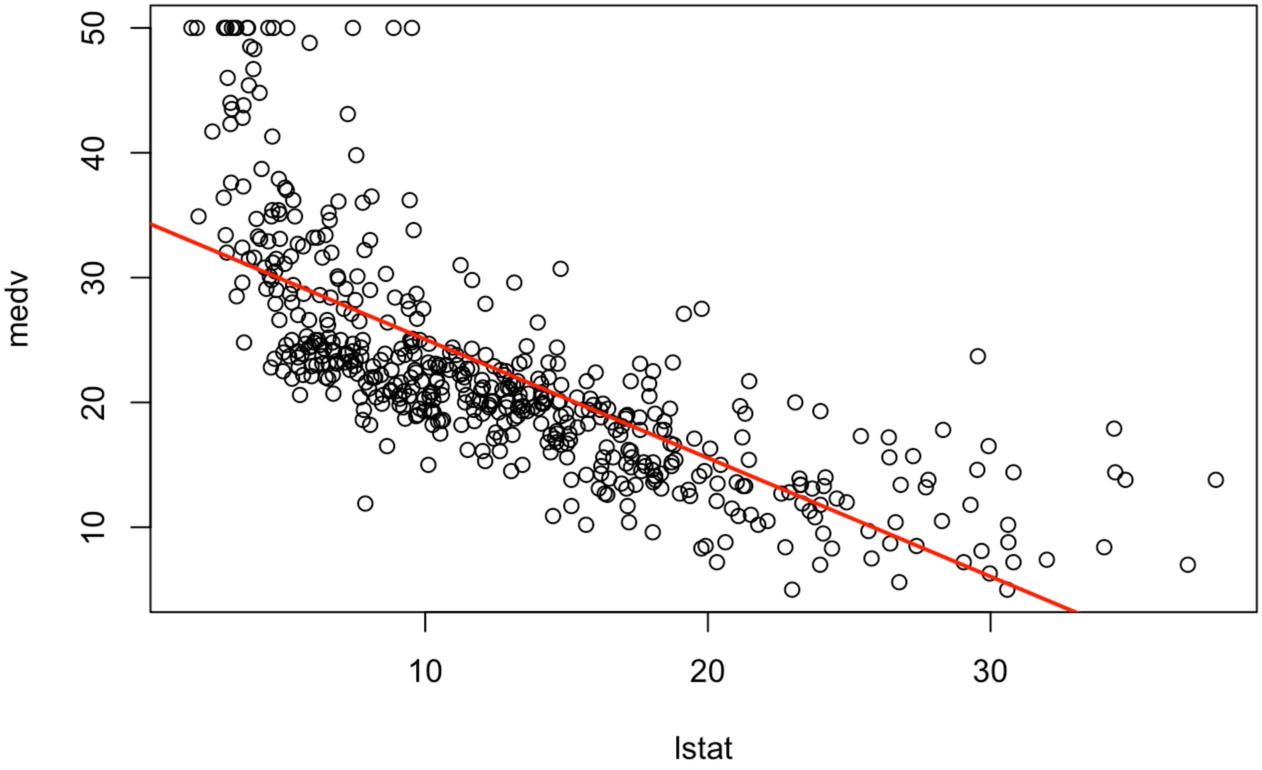
-0.95005

c) How good this model fits the data?

Residual standard error = 6.216 Variance explained by this model is 54.41%

d) Visualize the fitted line

```
plot(BostonData$lstat, BostonData$medv, xlab = 'lstat', ylab = 'medv')
abline(fit.simple,col = 'red',lwd = 2, lty = 1)
```



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?

```
lstat.new <- data.frame(lstat=c(5,10,15))
medv.new <- (predict(fit.simple, lstat.new))
medv.new
```

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

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

```
medv.new.conf <- predict(fit.simple, lstat.new, interval = "confidence")
medv.new.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
```

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

residual - observation - prediction The first neighborhood is the most accurate.

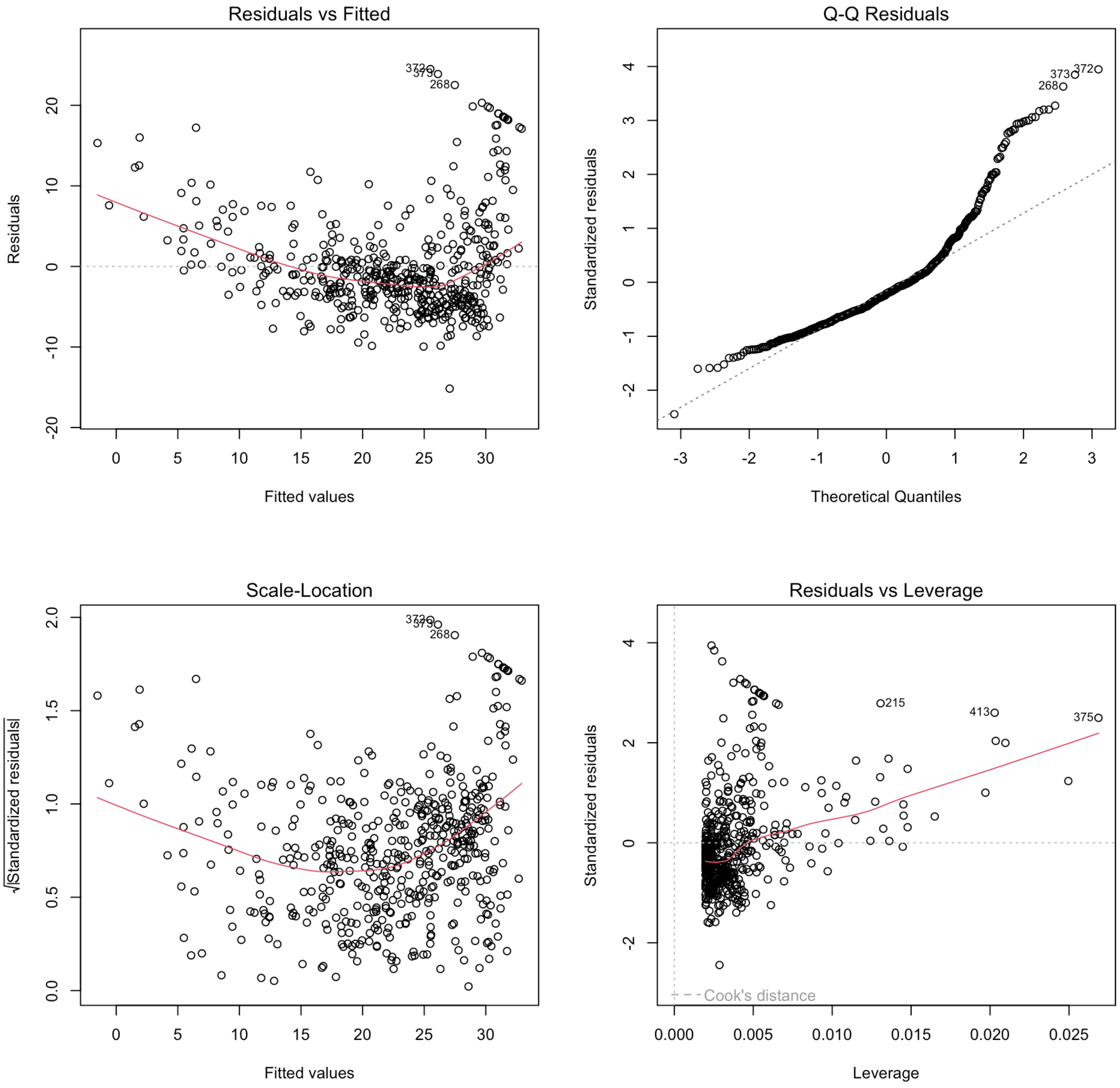
```
c(33,20,50) - medv.new
```

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

Q4 Residual plot.

Please plot the resdiual 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?

Yes

b) Is there correlation between error terms?

No

c) Is there heteroscedasticity between error terms?

Yes

d) Are there outliers?

Yes

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

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

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

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

P-value of F-statistics is less than 0.05, which indicates that there is a relationship between median house value and the other variables.

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

All variables are significant, besides indus and age, which have a p-value higher than 0.05.

####c) How good this model fits the data? 74% of the variance in the data could be explained by the multiple linear regression model we fit (R-squared = 0.7406).

d) Select the best subset of variables using forward selection, backward selection and mixed selection with AIC criteria.

The best set of variable selected by forward, backward and mixed selection is the same, which equal to the following features: crim + zn + chas + nox + rm + dis + rad + tax + ptratio + black + lstat.

```
fit.null <- (lm(medv~1, data=BostonData))
summary(fit.null)
```

```
##
## Call:
## lm(formula = medv ~ 1, data = BostonData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -17.533  -5.508  -1.333   2.467  27.467
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  22.5328     0.4089   55.11   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.197 on 505 degrees of freedom
```

```
select.forward <- step(fit.null, scope = list(lower=fit.null,upper = fit.multiple), dire
ction = 'forward')
```

```
## Start:  AIC=2246.51
## medv ~ 1
## 
##           Df Sum of Sq   RSS    AIC
## + lstat    1   23243.9 19472 1851.0
## + rm       1   20654.4 22062 1914.2
## + ptratio  1   11014.3 31702 2097.6
## + indus    1    9995.2 32721 2113.6
## + tax      1    9377.3 33339 2123.1
## + nox      1    7800.1 34916 2146.5
## + crim     1    6440.8 36276 2165.8
## + rad      1    6221.1 36495 2168.9
## + age      1    6069.8 36647 2171.0
## + zn       1    5549.7 37167 2178.1
## + black    1    4749.9 37966 2188.9
## + dis      1    2668.2 40048 2215.9
## + chas     1    1312.1 41404 2232.7
## <none>                 42716 2246.5
## 
## Step:  AIC=1851.01
## medv ~ lstat
## 
##           Df Sum of Sq   RSS    AIC
## + rm       1    4033.1 15439 1735.6
## + ptratio  1    2670.1 16802 1778.4
## + chas     1     786.3 18686 1832.2
## + dis      1     772.4 18700 1832.5
## + age      1     304.3 19168 1845.0
## + tax      1     274.4 19198 1845.8
## + black    1     198.3 19274 1847.8
## + zn       1     160.3 19312 1848.8
## + crim     1     146.9 19325 1849.2
## + indus    1      98.7 19374 1850.4
## <none>                 19472 1851.0
## + rad      1      25.1 19447 1852.4
## + nox      1       4.8 19468 1852.9
## 
## Step:  AIC=1735.58
## medv ~ lstat + rm
## 
##           Df Sum of Sq   RSS    AIC
## + ptratio  1   1711.32 13728 1678.1
## + chas     1    548.53 14891 1719.3
## + black    1    512.31 14927 1720.5
## + tax      1    425.16 15014 1723.5
## + dis      1    351.15 15088 1725.9
## + crim     1    311.42 15128 1727.3
## + rad      1    180.45 15259 1731.6
## + indus    1     61.09 15378 1735.6
## <none>                 15439 1735.6
## + zn       1     56.56 15383 1735.7
## + age      1     20.18 15419 1736.9
## + nox      1     14.90 15424 1737.1
##
## Step:  AIC=1678.13
## medv ~ lstat + rm + ptratio
##
##          Df Sum of Sq   RSS    AIC
## + dis    1    499.08 13229 1661.4
## + black  1    389.68 13338 1665.6
## + chas   1    377.96 13350 1666.0
## + crim   1    122.52 13606 1675.6
## + age    1     66.24 13662 1677.7
## <none>               13728 1678.1
## + tax    1     44.36 13684 1678.5
## + nox    1     24.81 13703 1679.2
## + zn     1     14.96 13713 1679.6
## + rad    1      6.07 13722 1679.9
## + indus  1      0.83 13727 1680.1
##
## Step:  AIC=1661.39
## medv ~ lstat + rm + ptratio + dis
##
##          Df Sum of Sq   RSS    AIC
## + nox    1    759.56 12469 1633.5
## + black  1    502.64 12726 1643.8
## + chas   1    267.43 12962 1653.1
## + indus  1    242.65 12986 1654.0
## + tax    1    240.34 12989 1654.1
## + crim   1    233.54 12995 1654.4
## + zn     1    144.81 13084 1657.8
## + age    1     61.36 13168 1661.0
## <none>               13229 1661.4
## + rad    1     22.40 13206 1662.5
##
## Step:  AIC=1633.47
## medv ~ lstat + rm + ptratio + dis + nox
##
##          Df Sum of Sq   RSS    AIC
## + chas   1    328.27 12141 1622.0
## + black  1    311.83 12158 1622.7
## + zn     1    151.71 12318 1629.3
## + crim   1    141.43 12328 1629.7
## + rad    1     53.48 12416 1633.3
## <none>               12469 1633.5
## + indus  1     17.10 12452 1634.8
## + tax    1     10.50 12459 1635.0
## + age    1      0.25 12469 1635.5
##
## Step:  AIC=1621.97
## medv ~ lstat + rm + ptratio + dis + nox + chas
##
##          Df Sum of Sq   RSS    AIC
## + black  1   272.837 11868 1612.5
## + zn     1   164.406 11977 1617.1
## + crim   1   116.330 12025 1619.1
## + rad    1    58.556 12082 1621.5
## <none>               12141 1622.0
## + indus  1    26.274 12115 1622.9
## + tax    1     4.187 12137 1623.8
## + age    1     2.331 12139 1623.9
##
## Step:  AIC=1612.47
## medv ~ lstat + rm + ptratio + dis + nox + chas + black
##
##         Df Sum of Sq   RSS    AIC
## + zn     1   189.936 11678 1606.3
## + rad    1   144.320 11724 1608.3
## + crim   1    55.633 11813 1612.1
## <none>               11868 1612.5
## + indus  1    15.584 11853 1613.8
## + age    1     9.446 11859 1614.1
## + tax    1     2.703 11866 1614.4
##
## Step:  AIC=1606.31
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn
##
##         Df Sum of Sq   RSS    AIC
## + crim   1    94.712 11584 1604.2
## + rad    1    93.614 11585 1604.2
## <none>               11678 1606.3
## + indus  1    16.048 11662 1607.6
## + tax    1     3.952 11674 1608.1
## + age    1     1.491 11677 1608.2
##
## Step:  AIC=1604.19
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn +
##     crim
##
##         Df Sum of Sq   RSS    AIC
## + rad    1   228.604 11355 1596.1
## <none>               11584 1604.2
## + indus  1    15.773 11568 1605.5
## + age    1     2.470 11581 1606.1
## + tax    1     1.305 11582 1606.1
##
## Step:  AIC=1596.1
## medv ~ lstat + rm + ptratio + dis + nox + chas + black + zn +
##     crim + rad
##
##         Df Sum of Sq   RSS    AIC
## + tax    1   273.619 11081 1585.8
## <none>               11355 1596.1
## + indus  1    33.894 11321 1596.6
## + age    1     0.096 11355 1598.1
##
```

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

```
summary(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
```

```
select.backward <- step(fit.multiple, scope = list(lower=fit.null,upper = fit.multiple),
direction = 'backward')
```

```
## Start:  AIC=1589.64
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
##     tax + ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - age      1      0.06 11079 1587.7
## - indus    1      2.52 11081 1587.8
## <none>                 11079 1589.6
## - chas     1    218.97 11298 1597.5
## - tax      1    242.26 11321 1598.6
## - crim     1    243.22 11322 1598.6
## - zn       1    257.49 11336 1599.3
## - black    1    270.63 11349 1599.8
## - rad      1    479.15 11558 1609.1
## - nox      1    487.16 11566 1609.4
## - ptratio  1   1194.23 12273 1639.4
## - dis      1   1232.41 12311 1641.0
## - rm       1   1871.32 12950 1666.6
## - lstat    1   2410.84 13490 1687.3
##
## Step:  AIC=1587.65
## medv ~ crim + zn + indus + chas + nox + rm + dis + rad + tax +
##     ptratio + black + lstat
##
##           Df Sum of Sq   RSS    AIC
## - indus    1      2.52 11081 1585.8
## <none>                 11079 1587.7
## - chas     1    219.91 11299 1595.6
## - tax      1    242.24 11321 1596.6
## - crim     1    243.20 11322 1596.6
## - zn       1    260.32 11339 1597.4
## - black    1    272.26 11351 1597.9
## - rad      1    481.09 11560 1607.2
## - nox      1    520.87 11600 1608.9
## - ptratio  1   1200.23 12279 1637.7
## - dis      1   1352.26 12431 1643.9
## - rm       1   1959.55 13038 1668.0
## - lstat    1   2718.88 13798 1696.7
##
## Step:  AIC=1585.76
## medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##
##           Df Sum of Sq   RSS    AIC
## <none>                 11081 1585.8
## - chas     1    227.21 11309 1594.0
## - crim     1    245.37 11327 1594.8
## - zn       1    257.82 11339 1595.4
## - black    1    270.82 11352 1596.0
## - tax      1    273.62 11355 1596.1
## - rad      1    500.92 11582 1606.1
## - nox      1    541.91 11623 1607.9
## - ptratio  1   1206.45 12288 1636.0
## - dis      1   1448.94 12530 1645.9
## - rm       1   1963.66 13045 1666.3
## - lstat    1   2723.48 13805 1695.0
```

```
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
```

```
select.mixed <- step(fit.multiple, scope = list(lower=fit.null,upper = fit.multiple), di
rection = 'both')
```

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

```
## 
## 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
```

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

Yes. The variable selected are: crim + zn + chas + nox + rm + dis + rad + tax + ptratio + black + lstat.

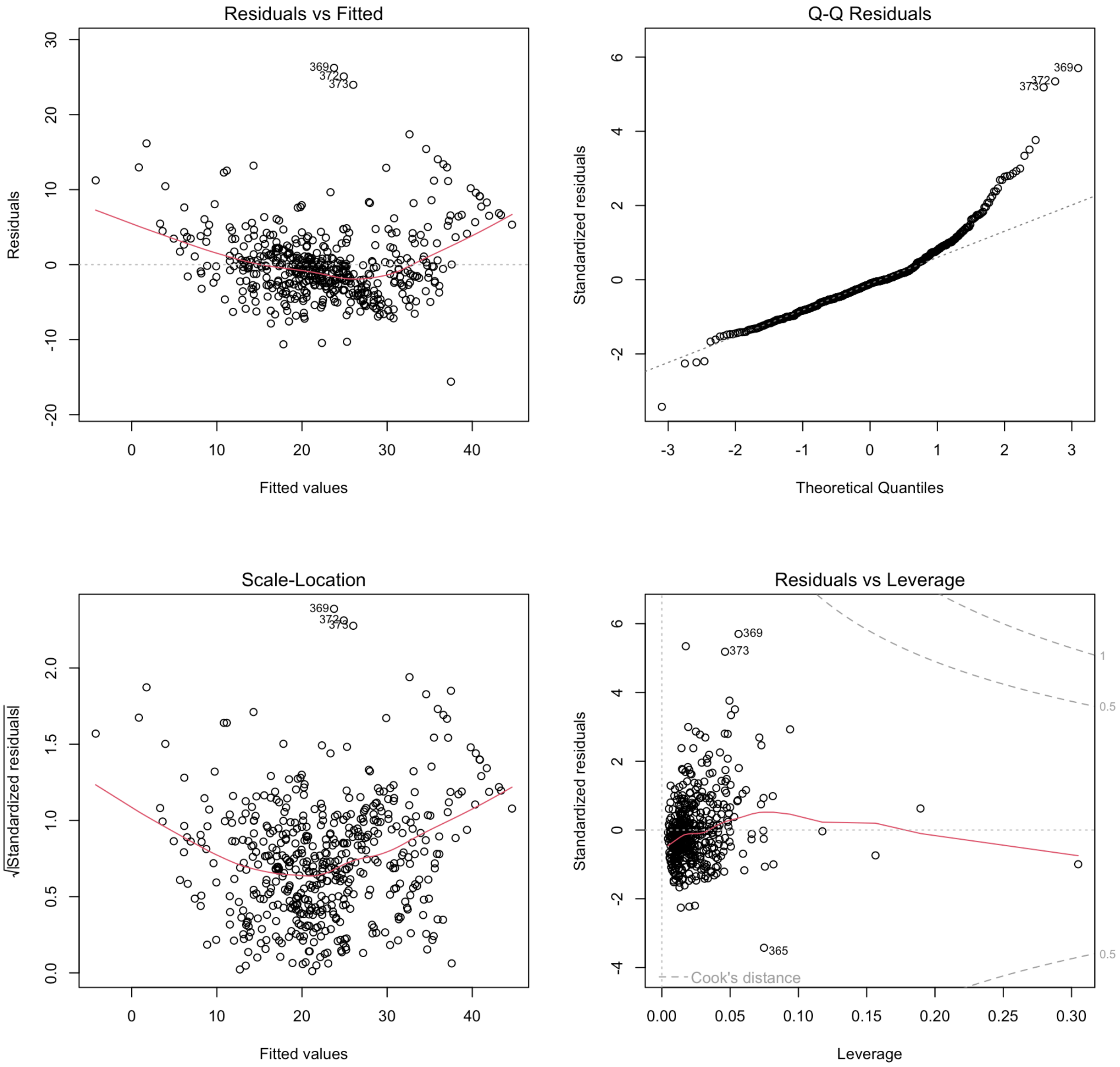
f) Does variable selection improve the R-square? How about the adjusted R-square?

Before variable selection: R-squared = 0.7406; adjusted R-squared = 0.7338 After variable selection: R-squared = 0.7406; adjusted R-squared: 0.7348 The variable selection slightly improved the adjusted R-squared, but the improvement was very small.

Q6 Residual plot.

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

```
par(mfrow =c(2,2))
plot(select.mixed)
```



a) Is there a nonlinear relationship?

Yes

b) Is there correlation between error terms?

No.

c) Is there heteroscedasticity between error terms?

Yes.

d) Are there outliers?

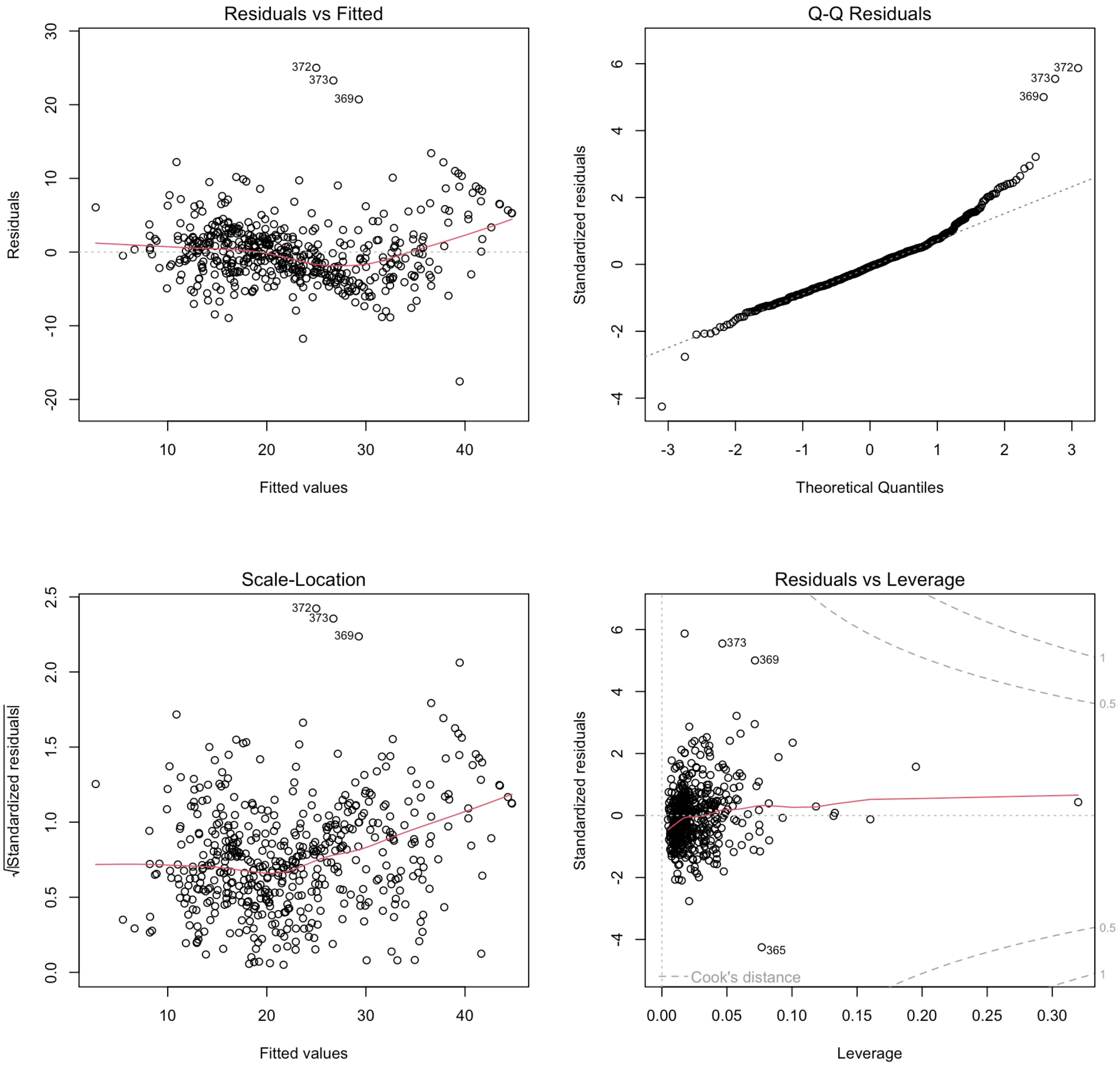
Yes.

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

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

```
##
## Call:
## lm(formula = medv ~ . - indus - age + I(lstat^2), data = BostonData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.5603  -2.6709  -0.3071   1.9469  25.0085
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  42.124187   4.632559   9.093  < 2e-16 ***
## crim         -0.149072   0.030006  -4.968 9.34e-07 ***
## zn            0.021281   0.012500   1.703 0.089291 .
## chas          2.589821   0.775307   3.340 0.000900 ***
## nox         -13.534522   3.229619  -4.191 3.30e-05 ***
## rm            3.233174   0.372802   8.673  < 2e-16 ***
## dis          -1.357892   0.169051  -8.032 7.09e-15 ***
## rad           0.271744   0.057599   4.718 3.11e-06 ***
## tax          -0.009546   0.003068  -3.111 0.001970 **
## ptratio      -0.790820   0.118089  -6.697 5.82e-11 ***
## black         0.008174   0.002429   3.365 0.000824 ***
## lstat        -1.669567   0.119012 -14.029  < 2e-16 ***
## I(lstat^2)    0.033055   0.003198  10.337  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.298 on 493 degrees of freedom
## Multiple R-squared:  0.7868, Adjusted R-squared:  0.7816
## F-statistic: 151.6 on 12 and 493 DF,  p-value: < 2.2e-16
```

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



a) Is the model improved?

Yes, the adjusted R-squared increased (0.7816).

b) Is the nonlinear effect significant?

Yes, the p-value of the t-test is smaller than 0.05.

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

Yes, the nonlinear relationship is solved compared to the previous plot.

Q8 Include the interaction term lstat X black

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

```
##
## Call:
## lm(formula = medv ~ . - indus - age + lstat:black, data = BostonData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.2694  -2.6710  -0.4913   1.8278  25.5984
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.022e+01  5.584e+00   5.412 9.75e-08 ***
## crim        -1.082e-01  3.260e-02  -3.319 0.000970 ***
## zn           4.389e-02  1.347e-02   3.258 0.001199 **
## chas         2.787e+00  8.500e-01   3.280 0.001113 **
## nox         -1.655e+01  3.531e+00  -4.687 3.58e-06 ***
## rm           3.609e+00  4.111e-01   8.780  < 2e-16 ***
## dis         -1.497e+00  1.847e-01  -8.107 4.15e-15 ***
## rad          2.910e-01  6.314e-02   4.608 5.18e-06 ***
## tax         -1.106e-02  3.365e-03  -3.288 0.001082 **
## ptratio     -9.561e-01  1.284e-01  -7.446 4.33e-13 ***
## black        2.860e-02  8.033e-03   3.560 0.000406 ***
## lstat       -1.924e-01  1.379e-01  -1.395 0.163668
## black:lstat -9.796e-04  3.846e-04  -2.547 0.011159 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.71 on 493 degrees of freedom
## Multiple R-squared:  0.744,  Adjusted R-squared:  0.7377
## F-statistic: 119.4 on 12 and 493 DF,  p-value: < 2.2e-16
```

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

```
##
## Call:
## lm(formula = medv ~ . - indus - age + lstat:black + I(lstat^2),
##     data = BostonData)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -17.7694  -2.6763  -0.2364   1.8670  25.0338
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.481e+01  5.299e+00   8.456 3.18e-16 ***
## crim        -1.508e-01  3.005e-02  -5.017 7.33e-07 ***
## zn           2.110e-02  1.250e-02   1.688  0.09207 .
## chas         2.557e+00  7.759e-01   3.296  0.00105 **
## nox         -1.371e+01  3.234e+00  -4.241 2.66e-05 ***
## rm           3.288e+00  3.764e-01   8.734  < 2e-16 ***
## dis         -1.351e+00  1.692e-01  -7.984 1.01e-14 ***
## rad          2.741e-01  5.764e-02   4.756 2.60e-06 ***
## tax         -9.744e-03  3.074e-03  -3.170  0.00162 **
## ptratio     -7.808e-01  1.185e-01  -6.591 1.13e-10 ***
## black        3.918e-04  7.853e-03   0.050  0.96022
## lstat       -1.847e+00  2.078e-01  -8.890  < 2e-16 ***
## I(lstat^2)   3.436e-02  3.433e-03  10.009  < 2e-16 ***
## black:lstat  3.926e-04  3.767e-04   1.042  0.29786
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.298 on 492 degrees of freedom
## Multiple R-squared:  0.7873, Adjusted R-squared:  0.7816
## F-statistic: 140.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

a) Is the model improved?

The model slightly improved the adjusted R-squared.

b) Is the interaction effect significant?

Yes, because p-value is less than 0.05.

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

a. Is the model improved? The R-squared did not change, so the model did not improve.
b. Is the interaction effect significant? No

Q9 Apply K-nearest neighbor regression model on

lstat to predict medv 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.

```
randid <- sample(c(1:nrow(BostonData)))
Boston.train <- BostonData[randid[c(1:400)],]
Boston.test <- BostonData [randid[c(401:506)],]
head(Boston.train)
```

```
##        crim zn indus chas   nox    rm   age    dis rad tax ptratio  black lstat
## 102 0.11432  0  8.56    0 0.520 6.781  71.3 2.8561   5 384    20.9 395.58  7.67
## 260 0.65665 20  3.97    0 0.647 6.842 100.0 2.0107   5 264    13.0 391.93  6.90
## 82  0.04462 25  4.86    0 0.426 6.619  70.4 5.4007   4 281    19.0 395.63  7.22
## 311 2.63548  0  9.90    0 0.544 4.973  37.8 2.5194   4 304    18.4 350.45 12.64
## 256 0.03548 80  3.64    0 0.392 5.876  19.1 9.2203   1 315    16.4 395.18  9.25
## 219 0.11069  0 13.89    1 0.550 5.951  93.8 2.8893   5 276    16.4 396.90 17.92
##     medv
## 102 26.5
## 260 30.1
## 82  23.9
## 311 16.1
## 256 20.9
## 219 21.5
```

```
head(Boston.test)
```

```
##        crim zn indus chas   nox    rm   age    dis rad tax ptratio  black lstat
## 238 0.51183  0  6.20    0 0.507 7.358  71.6 4.1480   8 307    17.4 390.07  4.73
## 434 5.58107  0 18.10    0 0.713 6.436  87.9 2.3158  24 666    20.2 100.19 16.22
## 331 0.04544  0  3.24    0 0.460 6.144  32.2 5.8736   4 430    16.9 368.57  9.09
## 124 0.15038  0 25.65    0 0.581 5.856  97.0 1.9444   2 188    19.1 370.31 25.41
## 488 4.83567  0 18.10    0 0.583 5.905  53.2 3.1523  24 666    20.2 388.22 11.45
## 372 9.23230  0 18.10    0 0.631 6.216 100.0 1.1691  24 666    20.2 366.15  9.53
##     medv
## 238 31.5
## 434 14.3
## 331 19.8
## 124 17.3
## 488 20.6
## 372 50.0
```

```
lstat.train <- Boston.train['lstat']
medv.train <- Boston.train ['medv']
lstat.test <- Boston.test ['lstat']
medv.test <- Boston.test ['medv']
```

Step-2: Use training data to predict test data and calculate the prediction error under K= 1,5,10,50,100,250

```
library(FNN)
#k=1
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_250 = knn.reg(train = lstat.train, test = lstat.test, y = medv.train, k = 250)
pred_001
```

```
## Prediction:
##   [1] 29.0 18.8 20.6  9.7 22.7 24.8  7.2 17.0 22.2 16.6 16.5 23.0 30.5 15.2 32.4
##  [16] 21.9 29.8 27.5 23.7 13.8 23.1 20.0 21.7 27.9 15.4 16.6 13.5 50.0 31.6 21.2
##  [31] 13.0 35.4 31.5 29.8 14.1 19.6 11.9 37.0 21.2 20.9 18.2 18.8 22.4 16.7 22.4
##  [46] 18.2 26.2 13.4 16.6 36.0 22.5 22.6  7.0 26.4 15.4 18.7 11.8 16.2 27.5 21.2
##  [61] 26.6 19.4 15.6 24.8 15.6 18.8 22.0 18.3 23.6 19.5 32.4 24.3 21.5 13.3 20.6
##  [76] 23.4 16.5 28.5 18.8 13.3 23.4 19.4 23.2 23.1 20.6  9.5 26.6 22.2 16.7 23.3
##  [91] 25.0 13.6 50.0 23.1 50.0 29.8 21.9 14.2 28.6 50.0  5.0 20.7  8.4 20.8 44.0
## [106] 24.4
```

```
pred_005
```

```
## Prediction:
##   [1] 31.26 18.62 22.04 11.74 19.82 23.60  9.12 16.50 23.34 22.10 22.52 25.20
##  [13] 32.84 18.64 37.66 28.84 20.82 24.86 30.46 16.50 18.84 19.64 19.24 21.62
##  [25] 14.26 15.76 11.58 43.08 26.44 19.08 15.04 33.84 40.30 20.82 15.48 20.48
##  [37] 24.12 35.86 19.76 21.60 22.68 18.62 22.92 16.50 22.92 19.84 25.68 13.06
##  [49] 22.10 27.26 15.20 20.22 12.30 22.56 14.26 18.68 10.62 16.94 24.00 25.36
##  [61] 29.28 14.92 16.80 23.60 18.80 19.30 22.68 19.88 25.60 16.50 33.36 21.94
##  [73] 16.68 14.94 19.16 33.12 13.58 42.02 18.62 14.16 32.40 14.92 22.40 17.50
##  [85] 19.16 10.12 30.28 23.34 16.50 28.18 24.86 14.16 45.52 27.98 39.72 29.86
##  [97] 21.56 16.48 25.20 32.36 11.70 23.00 10.12 20.62 41.92 20.36
```

```
pred_010
```

```
## Prediction:
##   [1] 32.93 18.37 25.20 11.57 22.17 24.82 11.09 19.39 26.71 20.30 24.79 26.05
##  [13] 32.93 16.87 39.35 32.25 21.06 24.27 32.70 19.58 17.10 20.74 19.65 20.94
##  [25] 12.92 17.44 13.35 44.30 25.87 19.61 15.77 33.23 39.49 20.70 15.74 19.47
##  [37] 24.95 30.41 20.66 22.97 22.89 18.37 23.89 19.39 23.89 21.35 27.43 12.40
##  [49] 20.43 24.96 16.23 19.67 10.71 21.55 12.92 19.59 11.00 16.22 24.27 24.82
##  [61] 29.77 16.77 17.70 24.82 17.22 18.47 22.89 20.47 27.46 17.20 39.68 25.20
##  [73] 15.97 12.92 19.82 29.13 12.03 39.44 18.37 12.59 28.81 16.77 21.08 17.76
##  [85] 19.82 11.00 29.77 25.07 19.39 29.24 24.57 12.59 42.15 26.81 39.49 33.24
##  [97] 20.47 17.66 26.27 33.27 11.40 27.15 11.08 21.35 42.15 19.72
```

```
pred_050
```

```
## Prediction:
##   [1] 32.268 17.724 23.898 11.930 21.062 23.064 11.902 19.050 28.302 19.346
##  [11] 24.296 27.126 32.612 16.822 37.430 29.448 21.126 23.150 29.930 19.050
##  [21] 16.822 20.870 20.254 20.750 13.460 15.898 14.776 37.430 26.884 20.242
##  [31] 15.588 33.006 36.840 21.126 15.530 20.438 26.230 30.460 22.298 23.180
##  [41] 21.264 17.724 22.816 19.068 22.816 22.484 26.774 11.882 19.346 25.842
##  [51] 16.598 20.468 11.902 19.888 13.602 20.220 12.574 16.966 23.150 23.064
##  [61] 32.236 18.668 17.746 23.064 16.822 17.642 21.264 19.730 28.874 17.268
##  [71] 37.430 23.530 16.656 13.754 20.046 26.518 11.902 37.430 17.724 13.754
##  [81] 26.578 18.632 22.346 16.958 20.046 12.176 32.236 28.302 19.068 26.950
##  [91] 23.064 13.924 37.430 27.168 36.840 33.006 19.730 17.936 26.956 33.132
## [101] 12.666 27.632 12.936 22.484 37.430 19.490
```

```
pred_100
```

```
## Prediction:
##   [1] 31.898 18.106 24.126 13.886 21.402 23.752 13.740 18.718 29.967 19.031
##  [11] 24.401 28.190 31.898 16.997 32.490 30.467 21.437 23.885 30.771 18.718
##  [21] 17.107 21.160 20.236 20.800 14.766 16.327 15.197 32.490 27.912 20.382
##  [31] 15.778 31.898 32.490 21.437 15.750 20.396 25.638 31.415 22.116 24.083
##  [41] 21.431 18.106 23.596 19.019 23.596 22.532 27.598 13.857 19.031 25.963
##  [51] 16.880 20.396 13.740 19.867 14.875 20.191 13.980 17.392 23.885 23.721
##  [61] 31.720 18.499 18.106 23.752 17.183 18.121 21.431 19.635 30.163 18.030
##  [71] 32.490 24.098 16.880 14.875 20.181 25.891 13.740 32.490 18.106 14.856
##  [81] 25.891 18.373 22.759 17.107 20.181 13.980 31.720 29.967 19.019 26.071
##  [91] 23.885 14.856 32.490 26.813 32.490 32.015 19.692 18.343 28.190 32.240
## [101] 14.340 26.869 14.567 22.459 32.490 18.997
```

```
pred_250
```

```
## Prediction:
##   [1] 26.7496 19.3132 25.2064 17.8020 22.4076 24.5104 17.8020 20.2984 26.7496
##  [10] 20.4252 25.8320 26.7496 26.7496 18.7652 26.7496 26.7496 22.3512 24.5824
##  [19] 26.7496 20.2228 18.9164 22.1940 21.4312 21.9656 18.0792 18.4880 18.1292
##  [28] 26.7496 26.7496 21.4668 18.3960 26.7496 26.7496 22.3728 18.3960 21.4668
##  [37] 26.6324 26.7496 23.5136 24.9668 22.5660 19.3132 24.2624 20.3424 24.2624
##  [46] 23.8636 26.7496 17.8020 20.4252 26.6324 18.7124 21.5004 17.8020 20.8728
##  [55] 18.0792 21.3952 17.8020 18.9532 24.5824 24.5104 26.7496 19.9660 19.2636
##  [64] 24.5824 18.9164 19.3132 22.5132 20.6940 26.7496 19.2636 26.7496 25.0968
##  [73] 18.7452 18.1292 21.1368 26.7496 17.8020 26.7496 19.3132 18.1292 26.7496
##  [82] 19.9660 23.9164 18.9164 21.1368 17.8020 26.7496 26.7496 20.3424 26.7496
##  [91] 24.5824 18.1292 26.7496 26.7496 26.7496 26.7496 20.7604 19.7116 26.7496
## [100] 26.7496 17.8020 26.7496 17.8596 23.8636 26.7496 20.4252
```

Step-3: Find the best K

Calculate the mse under each K value, sse = mean(residual^2) K = 10 is optimal.

```
MSE_001 = sum((pred_001$pred-medv.test)^2)/106
MSE_005 = sum((pred_005$pred-medv.test)^2)/106
MSE_010 = sum((pred_010$pred-medv.test)^2)/106
MSE_050 = sum((pred_050$pred-medv.test)^2)/106
MSE_100 = sum((pred_100$pred-medv.test)^2)/106
MSE_250 = sum((pred_250$pred-medv.test)^2)/106
c (MSE_001,MSE_005,MSE_010,MSE_050,MSE_100,MSE_250)
```

```
## [1] 49.40925 28.79002 26.87421 27.16634 32.44680 50.25492
```