Assignment 3

Savitaa Venkateswaran, Ruchita Robert Rozario, Slavvy Coelho

2019-10-04

Instructions

Fill in your computations and answers to the assignment questions in this RMarkdown document. When you are finished, click the “Knit” button on RStudio to render an HTML document. You can then use your browser or tool of choice to convert the HTML document to a PDF file.

This assignment is to be handed in through canvas on Monday Oct 7 at 11:00pm. (Note that this due date is different from the due date given on the canvas Admin page.) This is a group assignment. You must join a group on canvas even if you want to work alone. Please upload one PDF file with your solutions per group.

Question 1 (Chapter 3, #15)

15. This problem involves the Boston data set, which we saw in the lab for this chapter. We will now try to predict per capita crime rate using the other variables in this data set. In other words, per capita crime rate is the response, and the other variables are the predictors.

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

Predictor- age vs crim

```
library(MASS)
attach(Boston)
fit_age = lm(crim ~ age)
summary(fit_age)
```

```
##
## Call:
## lm(formula = crim ~ age)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.789 -4.257 -1.230  1.527 82.849
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.057 on 504 degrees of freedom
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

Predictor- black vs crim

```
fit_black = lm(crim ~ black)
summary(fit_black)
```

```
##
## Call:
## lm(formula = crim ~ black)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.756  -2.299  -2.095  -1.296  86.822
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -0.036280   0.003873  -9.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.946 on 504 degrees of freedom
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16
```

Predictor- chas vs crim

```
chas = as.factor(chas)
fit_chas = lm(crim ~ chas)
summary(fit_chas)
```

```
##
## Call:
## lm(formula = crim ~ chas)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas1        -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,   Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

Predictor- dis vs crim

```
fit_dis = lm(crim ~ dis)
summary(fit_dis)
```

```
##
## Call:
## lm(formula = crim ~ dis)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16
```

Predictor- indus vs crim

```
fit_indus = lm(crim ~ indus)
summary(fit_indus)
```

```
##
## Call:
## lm(formula = crim ~ indus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16
```

Predictor- lstat vs crim

```
fit_lstat = lm(crim ~ lstat)
summary(fit_lstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```

Predictor- medv vs crim

```
fit_medv = lm(crim ~ medv)
summary(fit_medv)
```

```
##
## Call:
## lm(formula = crim ~ medv)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16
```

Predictor- nox vs crim

```
fit_nox = lm(crim ~ nox)
summary(fit_nox)
```

```
##
## Call:
## lm(formula = crim ~ nox)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

Predictor- ptratio vs crim

```
fit_ptratio = lm(crim ~ ptratio)
summary(fit_ptratio)
```

```
##
## Call:
## lm(formula = crim ~ ptratio)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11
```

Predictor- rad vs crim

```
rad = as.factor(rad)
fit_rad = lm(crim ~ rad)
summary(fit_rad)
```

```
##
## Call:
## lm(formula = crim ~ rad)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.597  -0.076   0.085  76.217
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03603    1.50132   0.024    0.981
## rad2         0.04726    2.03280   0.023    0.981
## rad3         0.06133    1.85480   0.033    0.974
## rad4         0.35787    1.63211   0.219    0.827
## rad5         0.65176    1.62664   0.401    0.689
## rad6         0.11403    1.99695   0.057    0.954
## rad7         0.11437    2.21488   0.052    0.959
## rad8         0.33538    2.03280   0.165    0.869
## rad24       12.72326    1.61105   7.897 1.84e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.714 on 497 degrees of freedom
## Multiple R-squared:  0.4004, Adjusted R-squared:  0.3907
## F-statistic: 41.48 on 8 and 497 DF,  p-value: < 2.2e-16
```

Predictor- rm vs crim

```
fit_rm = lm(crim ~ rm)
summary(fit_rm)
```

```
##
## Call:
## lm(formula = crim ~ rm)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.604 -3.952 -2.654  0.989 87.197
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.401 on 504 degrees of freedom
## Multiple R-squared:  0.04807,    Adjusted R-squared:  0.04618
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07
```

Predictor- tax vs crim

```
fit_tax = lm(crim ~ tax)
summary(fit_tax)
```

```
##
## Call:
## lm(formula = crim ~ tax)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16
```

Predictor- zn vs crim

```
fit_zn = lm(crim ~ zn)
summary(fit_zn)
```

```
##
## Call:
## lm(formula = crim ~ zn)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.429 -4.222 -2.620  1.250 84.523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

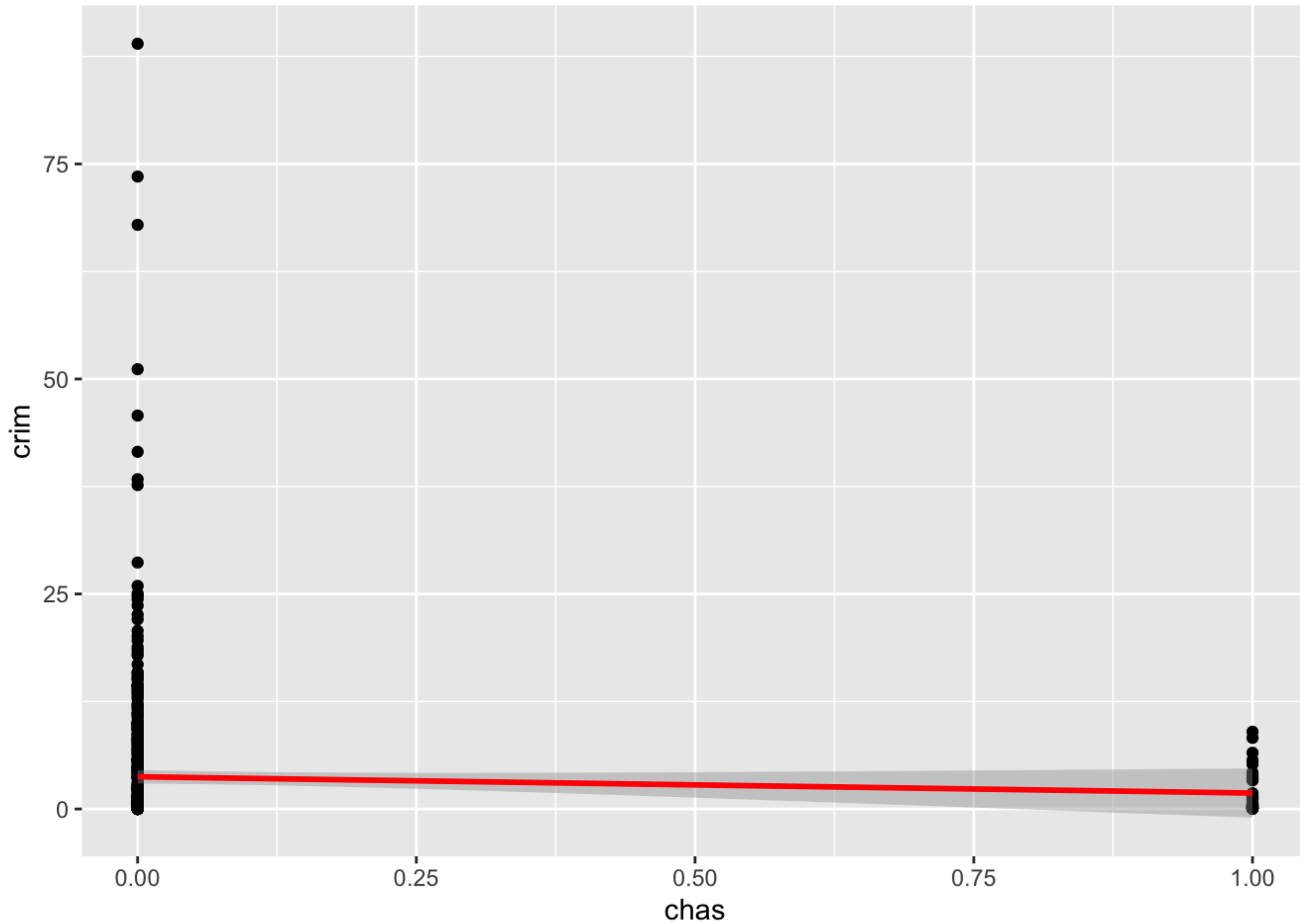
Observation:

Predictors having significant impact can be deduced using the test H0:β1=0. All predictors from the boston dataset have a “p-value” less than 0.05 except “chas”. This concludes that there is a statistically significant relationship between each predictor variables and the response (crim) variable except for the “chas” and “age” being a relatively weakly associated predictor variable (when compared to others).

To back up our observation:

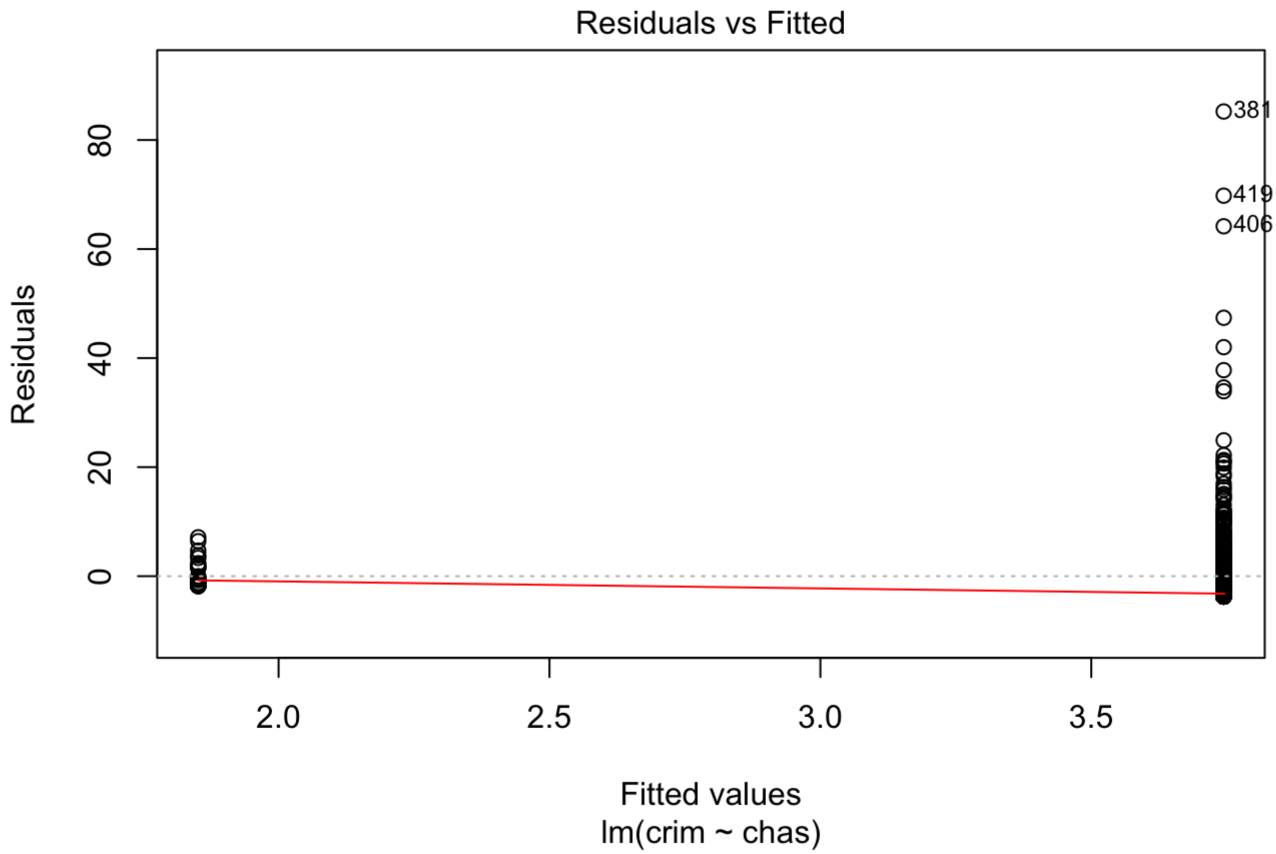
plot linear fit between chas vs crim lm fit plot

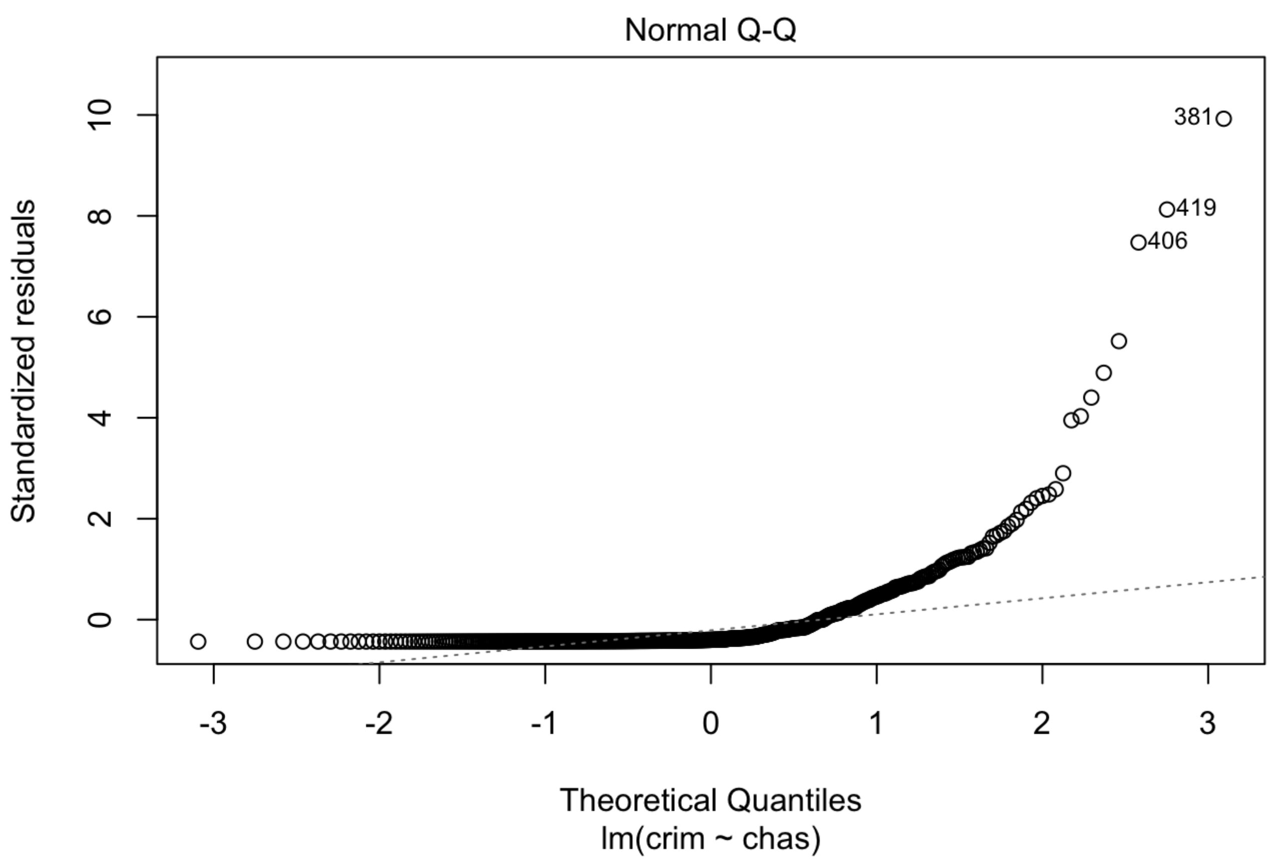
```
library(ggplot2)
ggplot(Boston, aes(x = chas, y = crim)) +
    geom_point() +
    stat_smooth(method = "lm", col = "red")
```

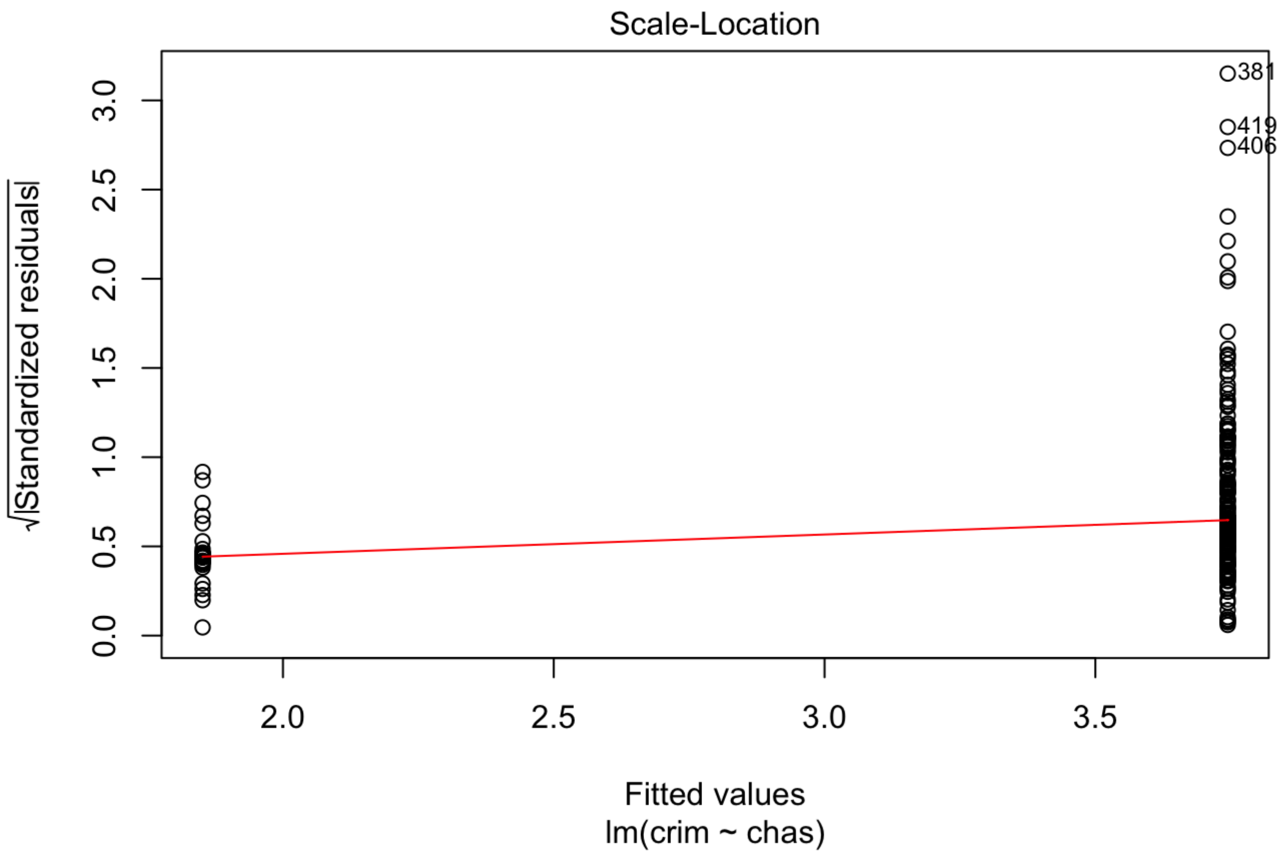


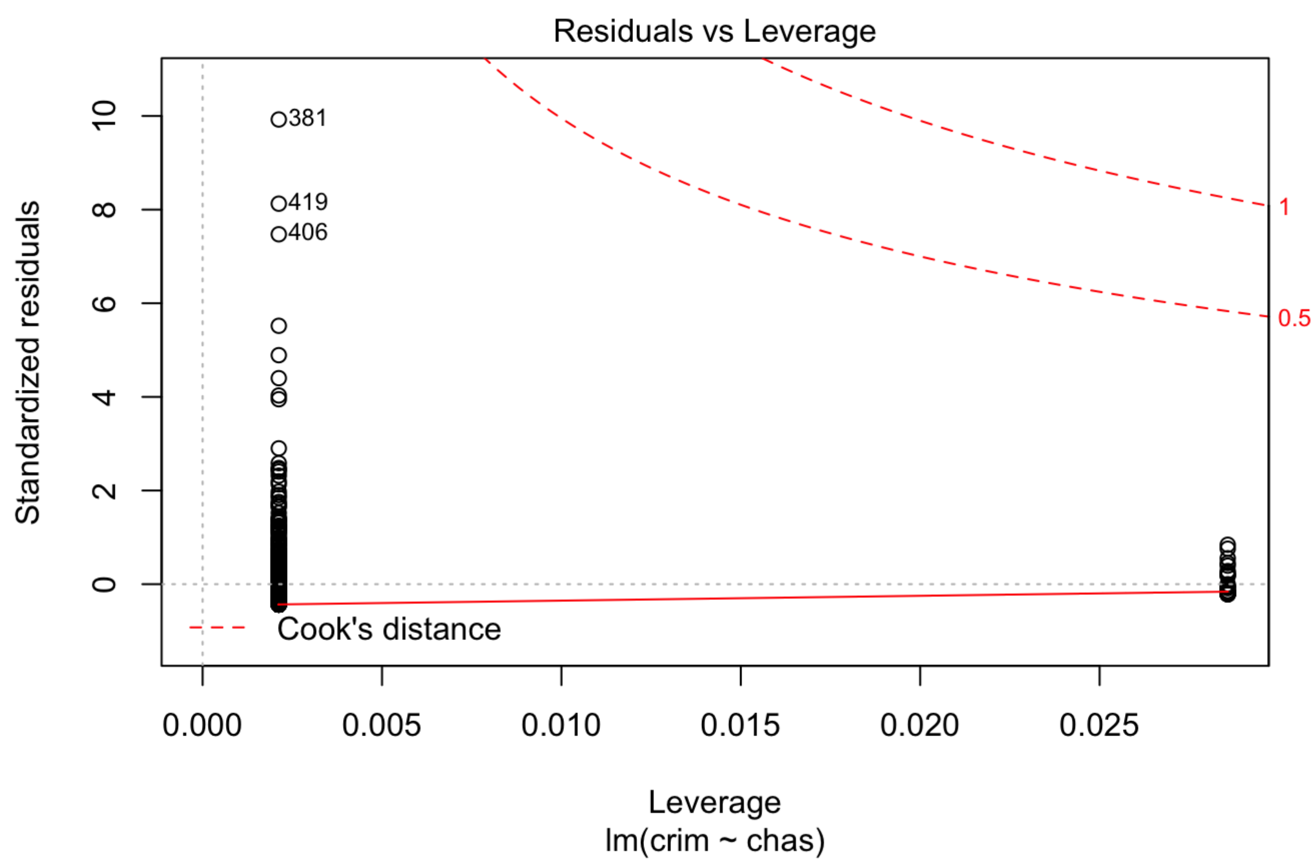
plot() function to visually examine the relationship between chas and crim variables

```
plot(fit_chas)
```


Residuals vs Fitted
Residuals
Fitted values
lm(crim ~ chas)
381
419
406
80
60
40
20
0
2.0
2.5
3.0
3.5

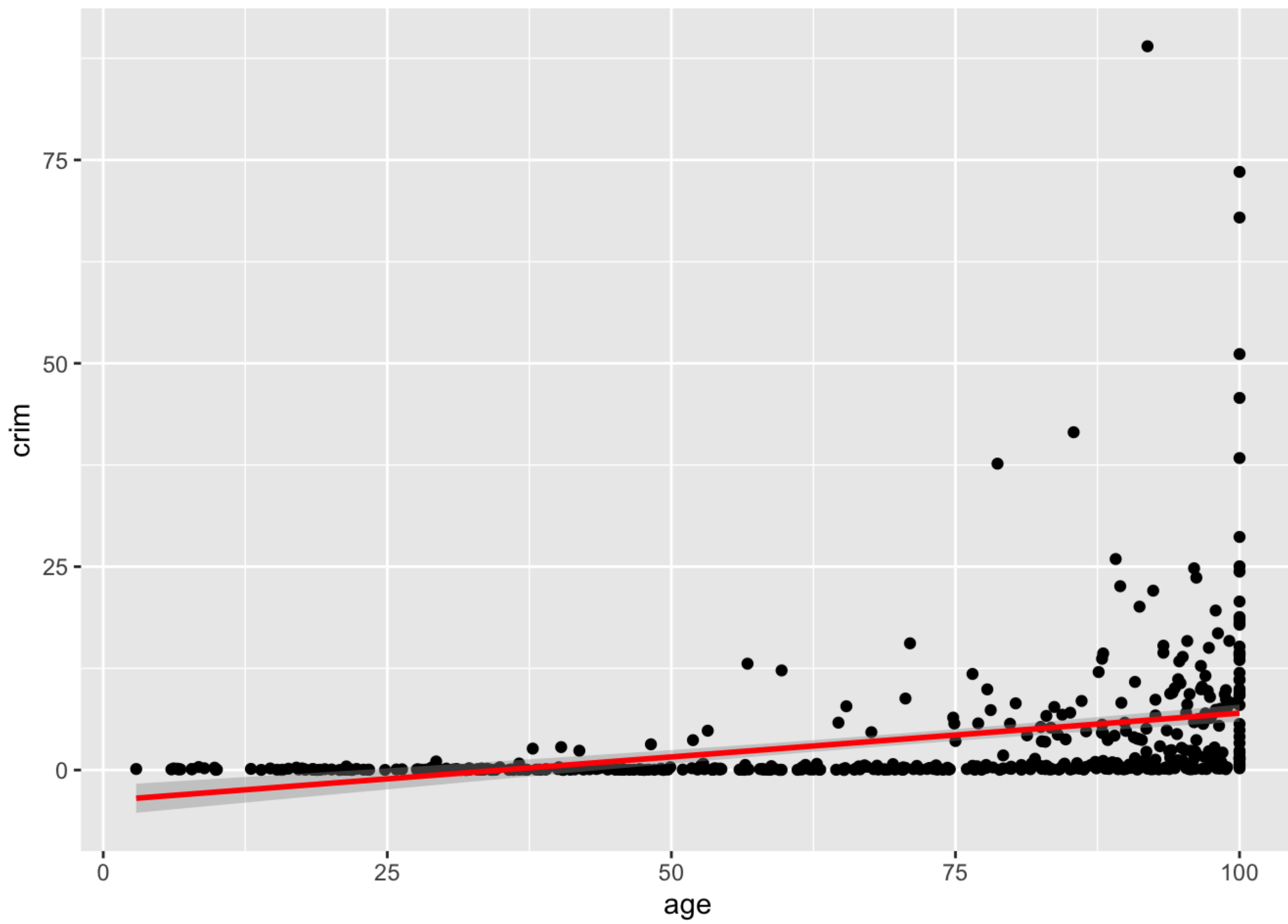

Normal Q-Q
Standardized residuals
Theoretical Quantiles
lm(crim ~ chas)
381
419
406
10
8
6
4
2
0
-3
-2
-1
0
1
2
3





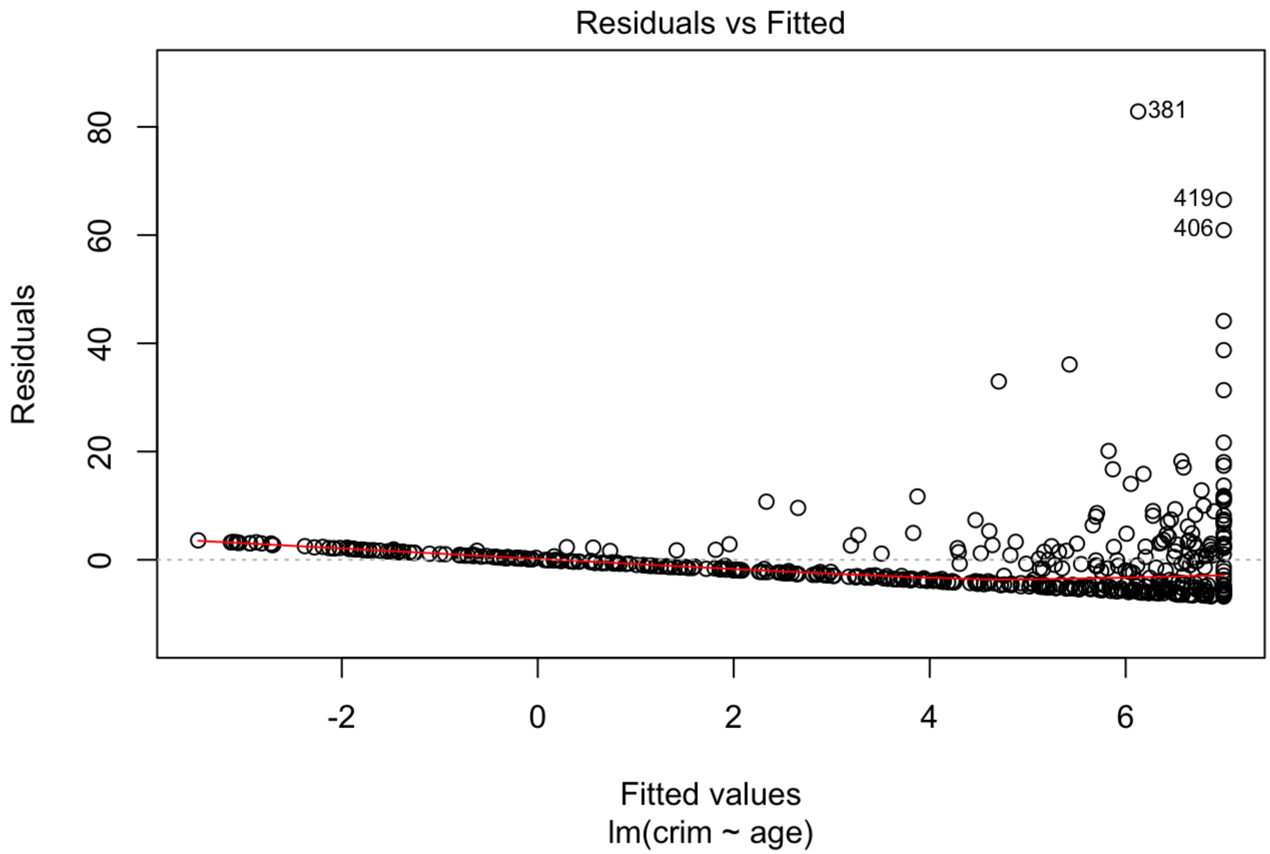
plot linear fit between age vs crim lm fit plot

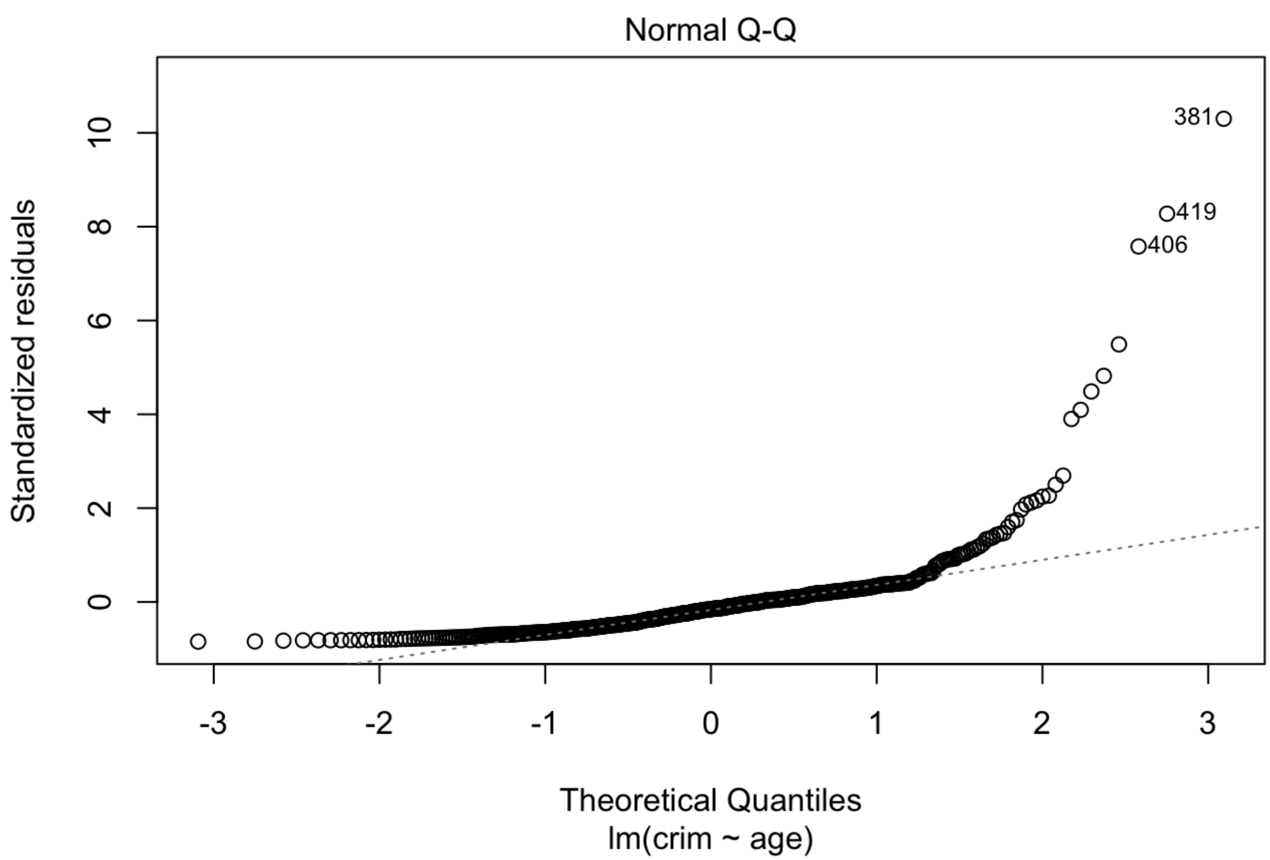
```
ggplot(Boston, aes(x = age, y = crim)) +
    geom_point() +
    stat_smooth(method = "lm", col = "red")
```

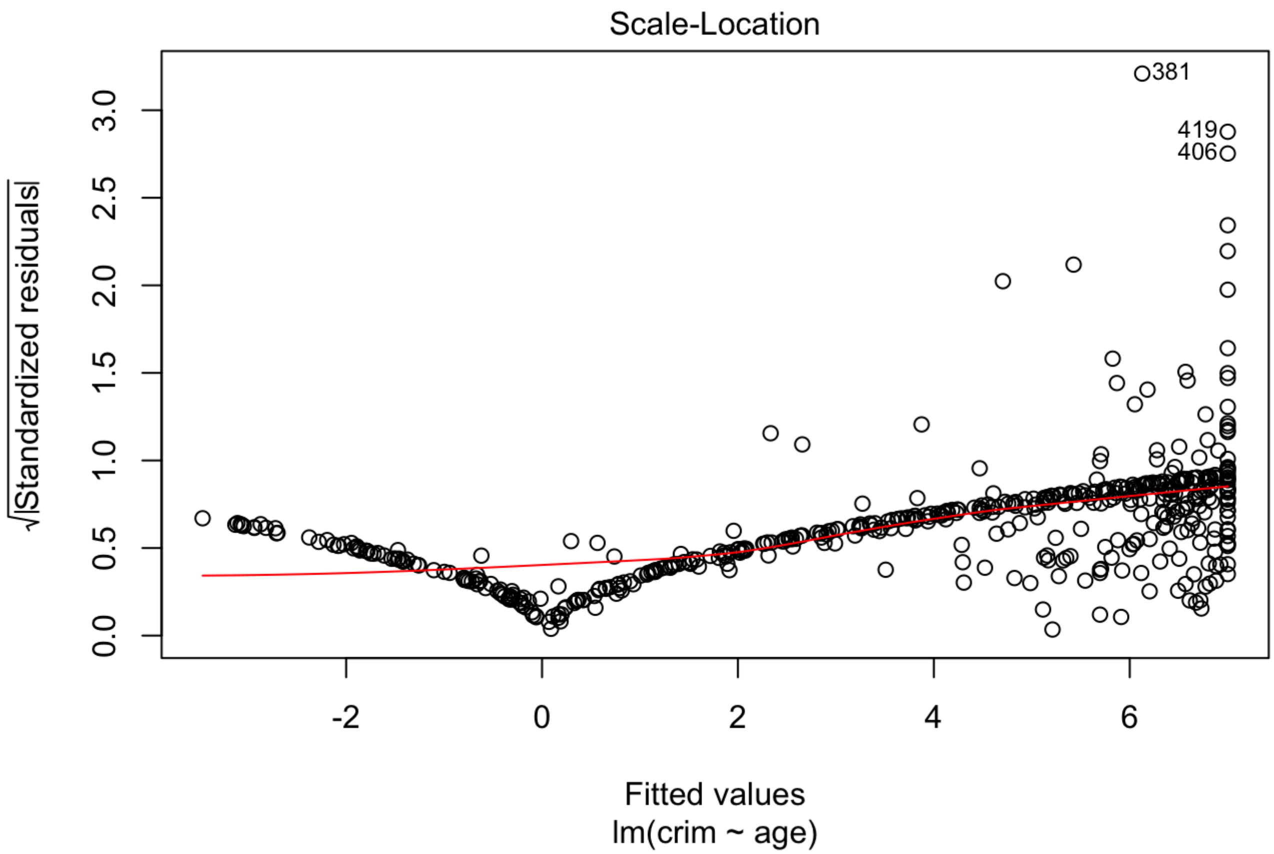


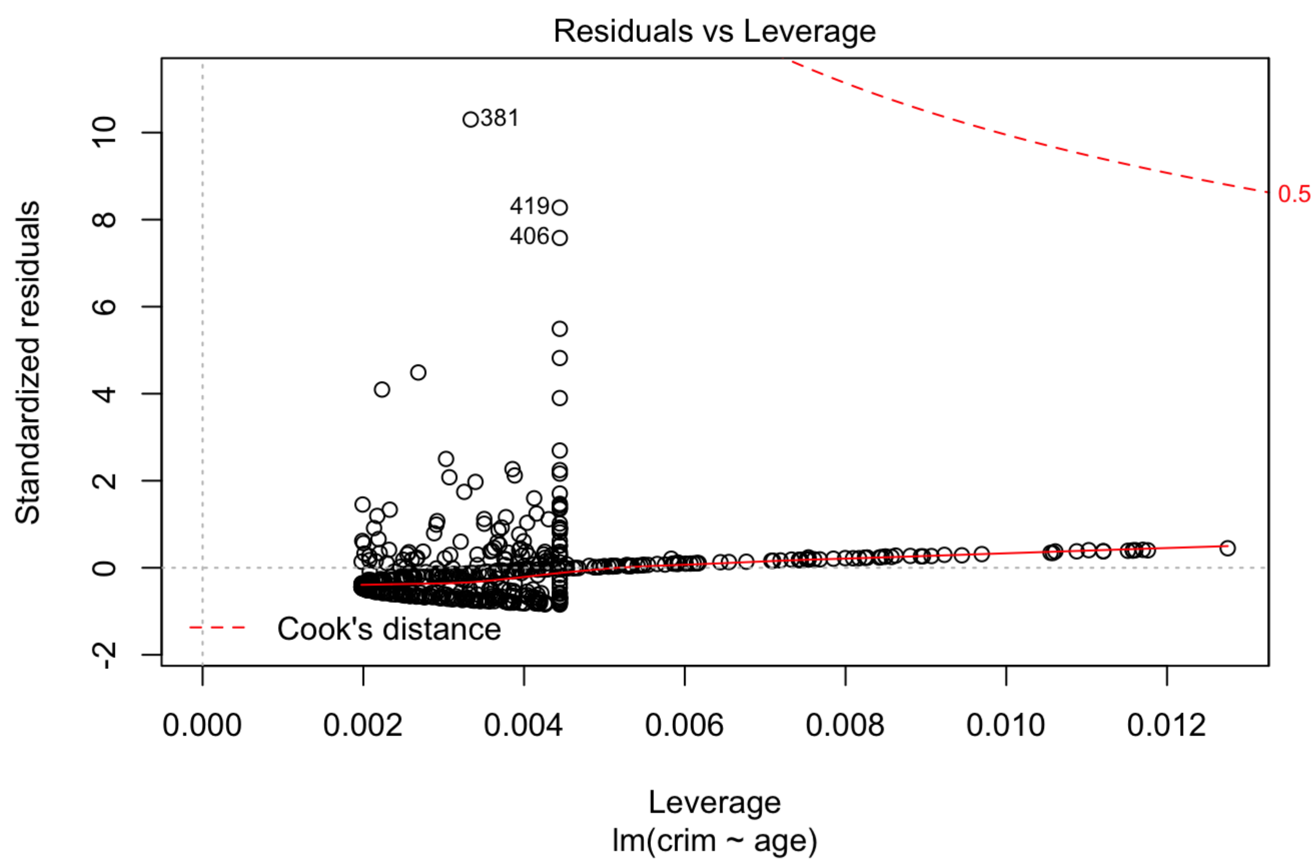
plot() function to visually examine the relationship between age and crim variables

```
plot(fit_age)
```


Residuals vs Fitted
Residuals
80
60
40
20
0
381
419
406
-2
0
2
4
6
Fitted values
lm(crim ~ age)


Normal Q-Q
Standardized residuals
10
8
6
4
2
0
381
419
406
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(crim ~ age)


Scale-Location
381
419
406
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
2.5
3.0
-2
0
2
4
6
Fitted values
lm(crim ~ age)


Residuals vs Leverage
381
419
406
0.5
Standardized residuals
-2
0
2
4
6
8
10
Cook's distance
0.000
0.002
0.004
0.006
0.008
0.010
0.012
Leverage
lm(crim ~ age)

b. Fit a multiple regression model to predict the response using all of the predictors. Describe your results. For which predictors can we reject the null hypothesis H0 : βj = 0?

```
fit_all <- lm(crim ~ ., data = Boston)
summary(fit_all)
```

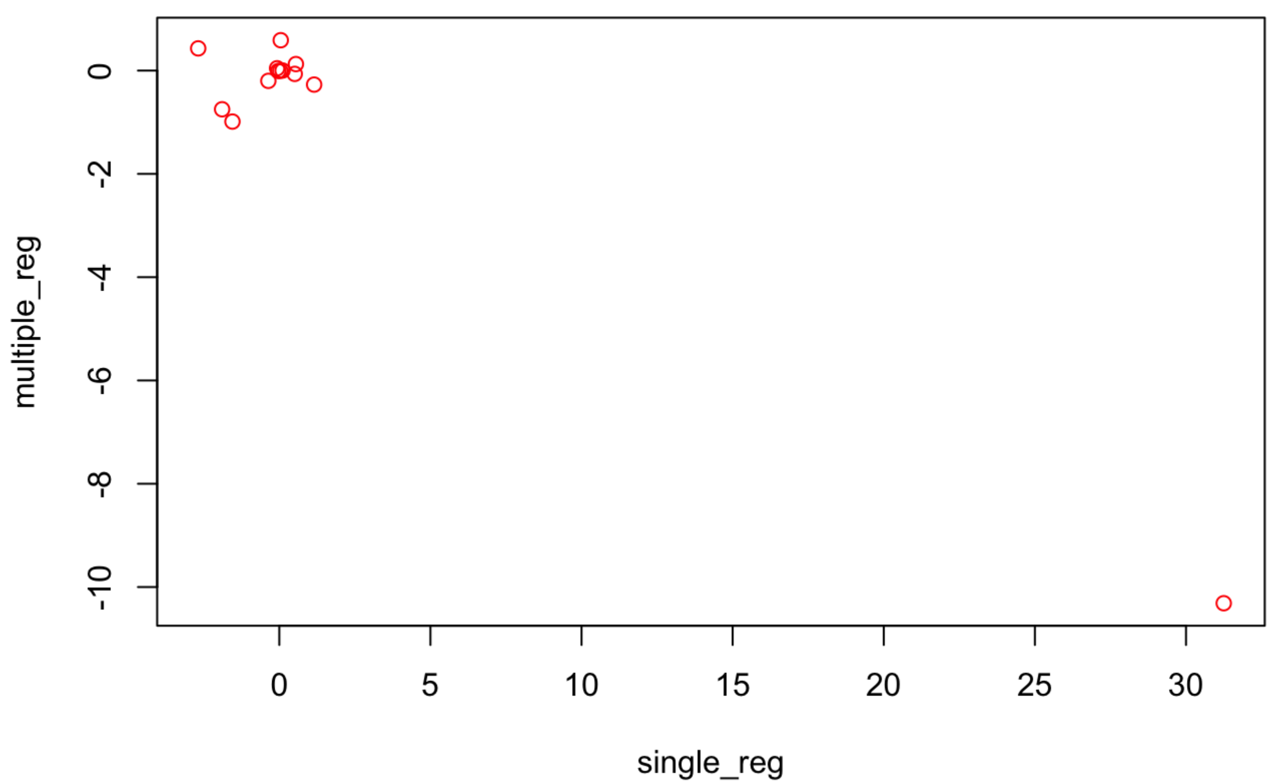
```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16
```

Summary of the results obtained:

For the following predictors: “zn”, “dis”, “rad”, “black” and “medv” of the Boston dataset, Null Hypothesis can be rejected as their p-values are lesser than 0.05 and hence, making them statiscally significant in predicting the repsonse variable (crim) in boston dataset.

c. How do your results from (a) compare to your results from (b)? Create a plot displaying the univariate regression coefficients from (a) on the x-axis, and the multiple regression coefficients from (b) on the y-axis. That is, each predictor is displayed as a single point in the plot. Its coefficient in a simple linear regression model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis.

```
single_reg <- vector("numeric",0)
single_reg <- c(single_reg, fit_zn$coefficient[2])
single_reg <- c(single_reg, fit_indus$coefficient[2])
single_reg <- c(single_reg, fit_chas$coefficient[2])
single_reg <- c(single_reg, fit_nox$coefficient[2])
single_reg <- c(single_reg, fit_rm$coefficient[2])
single_reg <- c(single_reg, fit_age$coefficient[2])
single_reg <- c(single_reg, fit_dis$coefficient[2])
single_reg <- c(single_reg, fit_rad$coefficient[2])
single_reg <- c(single_reg, fit_tax$coefficient[2])
single_reg <- c(single_reg, fit_ptratio$coefficient[2])
single_reg <- c(single_reg, fit_black$coefficient[2])
single_reg <- c(single_reg, fit_lstat$coefficient[2])
single_reg <- c(single_reg, fit_medv$coefficient[2])
multiple_reg <- vector("numeric", 0)
multiple_reg <- c(multiple_reg, fit_all$coefficients)
multiple_reg<- multiple_reg[-1]
plot(single_reg, multiple_reg, col = "red")
```




```
cor(Boston[-c(1, 4)])
```

```
##                 zn      indus        nox         rm        age        dis
## zn       1.0000000 -0.5338282 -0.5166037  0.3119906 -0.5695373  0.6644082
## indus   -0.5338282  1.0000000  0.7636514 -0.3916759  0.6447785 -0.7080270
## nox     -0.5166037  0.7636514  1.0000000 -0.3021882  0.7314701 -0.7692301
## rm       0.3119906 -0.3916759 -0.3021882  1.0000000 -0.2402649  0.2052462
## age     -0.5695373  0.6447785  0.7314701 -0.2402649  1.0000000 -0.7478805
## dis      0.6644082 -0.7080270 -0.7692301  0.2052462 -0.7478805  1.0000000
## rad     -0.3119478  0.5951293  0.6114406 -0.2098467  0.4560225 -0.4945879
## tax     -0.3145633  0.7207602  0.6680232 -0.2920478  0.5064556 -0.5344316
## ptratio -0.3916785  0.3832476  0.1889327 -0.3555015  0.2615150 -0.2324705
## black    0.1755203 -0.3569765 -0.3800506  0.1280686 -0.2735340  0.2915117
## lstat   -0.4129946  0.6037997  0.5908789 -0.6138083  0.6023385 -0.4969958
## medv     0.3604453 -0.4837252 -0.4273208  0.6953599 -0.3769546  0.2499287
##                rad        tax    ptratio      black      lstat       medv
## zn      -0.3119478 -0.3145633 -0.3916785  0.1755203 -0.4129946  0.3604453
## indus    0.5951293  0.7207602  0.3832476 -0.3569765  0.6037997 -0.4837252
## nox      0.6114406  0.6680232  0.1889327 -0.3800506  0.5908789 -0.4273208
## rm      -0.2098467 -0.2920478 -0.3555015  0.1280686 -0.6138083  0.6953599
## age      0.4560225  0.5064556  0.2615150 -0.2735340  0.6023385 -0.3769546
## dis     -0.4945879 -0.5344316 -0.2324705  0.2915117 -0.4969958  0.2499287
## rad      1.0000000  0.9102282  0.4647412 -0.4444128  0.4886763 -0.3816262
## tax      0.9102282  1.0000000  0.4608530 -0.4418080  0.5439934 -0.4685359
## ptratio  0.4647412  0.4608530  1.0000000 -0.1773833  0.3740443 -0.5077867
## black   -0.4444128 -0.4418080 -0.1773833  1.0000000 -0.3660869  0.3334608
## lstat    0.4886763  0.5439934  0.3740443 -0.3660869  1.0000000 -0.7376627
## medv    -0.3816262 -0.4685359 -0.5077867  0.3334608 -0.7376627  1.0000000
```

Inference:

The results obtained from simple linear regression might be offset from the results obtained from multiple regression. This is due to the fact that we consider the rate of change of a single predictor variable affecting the response variable. However, incase of multiple linear regression, inorder to understand the relationship between a predictor and the corresponding response variable we have to keep the other features/predictor variables fixed. This affects the relationship strength. This makes sense for multiple regression to suggest no relationship between the response and some of the predictors while the simple linear regression implies the opposite because the correlation between the predictors show some strong relationships between some of the predictors. This can be clearly observed from the corrleation obtained above, especially with regard to the the ‘age’ variable.

d. Is there evidence of non-linear association between any of the predictors and the response? To answer this question, for each predictor X, fit a model of the form $Y = β0 +β1X +β2X2 +β3X3 +ε$.

Polynomial Regression model for each predictor vs crim

Predictor- age vs crim

```
fit_age = lm(crim ~ poly(age, 3))
summary(fit_age)
```

```
##
## Call:
## lm(formula = crim ~ poly(age, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.762 -2.673 -0.516  0.019 82.842
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.6135     0.3485  10.368  < 2e-16 ***
## poly(age, 3)1  68.1820     7.8397   8.697  < 2e-16 ***
## poly(age, 3)2  37.4845     7.8397   4.781 2.29e-06 ***
## poly(age, 3)3  21.3532     7.8397   2.724  0.00668 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.84 on 502 degrees of freedom
## Multiple R-squared:  0.1742, Adjusted R-squared:  0.1693
## F-statistic: 35.31 on 3 and 502 DF,  p-value: < 2.2e-16
```

Predictor- black vs crim

```
fit_black = lm(crim ~ poly(black, 3))
summary(fit_black)
```

```
##
## Call:
## lm(formula = crim ~ poly(black, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.096  -2.343  -2.128  -1.439  86.790
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.6135     0.3536  10.218   <2e-16 ***
## poly(black, 3)1 -74.4312     7.9546  -9.357   <2e-16 ***
## poly(black, 3)2   5.9264     7.9546   0.745    0.457
## poly(black, 3)3  -4.8346     7.9546  -0.608    0.544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.955 on 502 degrees of freedom
## Multiple R-squared:  0.1498, Adjusted R-squared:  0.1448
## F-statistic: 29.49 on 3 and 502 DF,  p-value: < 2.2e-16
```

Predictor- dis vs crim

```
fit_dis = lm(crim ~ poly(dis, 3))
summary(fit_dis)
```

```
##
## Call:
## lm(formula = crim ~ poly(dis, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3259  11.087  < 2e-16 ***
## poly(dis, 3)1 -73.3886     7.3315 -10.010  < 2e-16 ***
## poly(dis, 3)2  56.3730     7.3315   7.689 7.87e-14 ***
## poly(dis, 3)3 -42.6219     7.3315  -5.814 1.09e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
```

Predictor- indus vs crim

```
fit_indus = lm(crim ~ poly(indus, 3))
summary(fit_indus)
```

```
##
## Call:
## lm(formula = crim ~ poly(indus, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.278 -2.514  0.054  0.764 79.713
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)        3.614      0.330  10.950  < 2e-16 ***
## poly(indus, 3)1   78.591      7.423  10.587  < 2e-16 ***
## poly(indus, 3)2  -24.395      7.423  -3.286  0.00109 **
## poly(indus, 3)3  -54.130      7.423  -7.292  1.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

Predictor- lstat vs crim

```
fit_lstat = lm(crim ~ poly(lstat, 3))
summary(fit_lstat)
```

```
##
## Call:
## lm(formula = crim ~ poly(lstat, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3392  10.654   <2e-16 ***
## poly(lstat, 3)1  88.0697     7.6294  11.543   <2e-16 ***
## poly(lstat, 3)2  15.8882     7.6294   2.082   0.0378 *
## poly(lstat, 3)3 -11.5740     7.6294  -1.517   0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
```

Predictor- medv vs crim

```
fit_medv = lm(crim ~ poly(medv, 3))
summary(fit_medv)
```

```
##
## Call:
## lm(formula = crim ~ poly(medv, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.614      0.292  12.374  < 2e-16 ***
## poly(medv, 3)1  -75.058      6.569 -11.426  < 2e-16 ***
## poly(medv, 3)2   88.086      6.569  13.409  < 2e-16 ***
## poly(medv, 3)3  -48.033      6.569  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Predictor- nox vs crim

```
fit_nox = lm(crim ~ poly(nox, 3))
summary(fit_nox)
```

```
##
## Call:
## lm(formula = crim ~ poly(nox, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.110 -2.068 -0.255  0.739 78.302
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3216  11.237  < 2e-16 ***
## poly(nox, 3)1  81.3720     7.2336  11.249  < 2e-16 ***
## poly(nox, 3)2 -28.8286     7.2336  -3.985 7.74e-05 ***
## poly(nox, 3)3 -60.3619     7.2336  -8.345 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

Predictor- ptratio vs crim

```
fit_ptratio = lm(crim ~ poly(ptratio, 3))
summary(fit_ptratio)
```

```
##
## Call:
## lm(formula = crim ~ poly(ptratio, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3.614      0.361  10.008  < 2e-16 ***
## poly(ptratio, 3)1   56.045      8.122   6.901 1.57e-11 ***
## poly(ptratio, 3)2   24.775      8.122   3.050  0.00241 **
## poly(ptratio, 3)3  -22.280      8.122  -2.743  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
```

Predictor- rm vs crim

```
fit_rm = lm(crim ~ poly(rm, 3))
summary(fit_rm)
```

```
##
## Call:
## lm(formula = crim ~ poly(rm, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.485  -3.468  -2.221  -0.015  87.219
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.6135     0.3703   9.758  < 2e-16 ***
## poly(rm, 3)1 -42.3794     8.3297  -5.088 5.13e-07 ***
## poly(rm, 3)2  26.5768     8.3297   3.191  0.00151 **
## poly(rm, 3)3  -5.5103     8.3297  -0.662  0.50858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,    Adjusted R-squared:  0.06222
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
```

Predictor- tax vs crim

```
fit_tax = lm(crim ~ poly(tax, 3))
summary(fit_tax)
```

```
##
## Call:
## lm(formula = crim ~ poly(tax, 3))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      3.6135     0.3047  11.860  < 2e-16 ***
## poly(tax, 3)1 112.6458     6.8537  16.436  < 2e-16 ***
## poly(tax, 3)2  32.0873     6.8537   4.682 3.67e-06 ***
## poly(tax, 3)3  -7.9968     6.8537  -1.167    0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.854 on 502 degrees of freedom
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16
```

Predictor- zn vs crim

```
fit_zn = lm(crim ~ poly(zn, 3))
summary(fit_zn)
```

```
##
## Call:
## lm(formula = crim ~ poly(zn, 3))
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.821 -4.614 -1.294  0.473 84.130
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3722   9.709  < 2e-16 ***
## poly(zn, 3)1    -38.7498     8.3722  -4.628  4.7e-06 ***
## poly(zn, 3)2     23.9398     8.3722   2.859  0.00442 **
## poly(zn, 3)3    -10.0719     8.3722  -1.203  0.22954
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
```

Inference:

The variables age,dis,indus,nox,medv,ptratio show statical significance using higher degree regressors (using polynomial regression), the p values are lesser than 0.05. This concludes they have some non-linear relationship with the response variable. However, other than the above mentioned features- none of the others depict any significant relationship; p-values are greater than or equal to 0.05. This concludes that in the latter case there are no non-linear trends between the different predictors and then response variable (crim).

Question 2 (Chapter 4, #4)

a. p=1 X is uniformly distributed on [0,1]. Predict: fraction within 10% range of test observations. Solution: Since it is a uniform distribution, For X=0.6, range = [0.55, 0.65] Therefore, for any X, (0.65-0.55)/(1-0) = 0.10
   ie. 10% of the total observation. Since X is evenly distributed, if there is 10% of the total lying in the given range, any range will have the same number of observations as that range.
b. p=2 (two features X1 and X2 are uniformly distributed.) Predict: fraction within 10% range of test observations. Solution: Since it is a uniform distribution, For X=0.6, range = [0.55, 0.65] For X=0.35 range = [0.3,0.4] Since X1 and X2 are independent variables, (10% * 10%) = (10%)2 = 0.01 = 1%
c. p=100 From the above cases, we can generalize that For a uniform distribution, Fractions to be used = (10%)p When p=100, Fraction = (10%)p => almost negligible
d. From a, b and c We observe that as we increase the number of features p, the percentage of observations that can be used to predict with KNN becomes very small.
e. when P side 1 0.1 2 (0.1)1/2=0.316 100 (0.1)1/100=0.977 Here as p increases, we need to include almost entire range of the considered features.

Question 3 (Chapter 4, #10 parts (a)-(h), 9 marks)

```
library(ISLR)
data(Weekly)
head(Weekly)
```

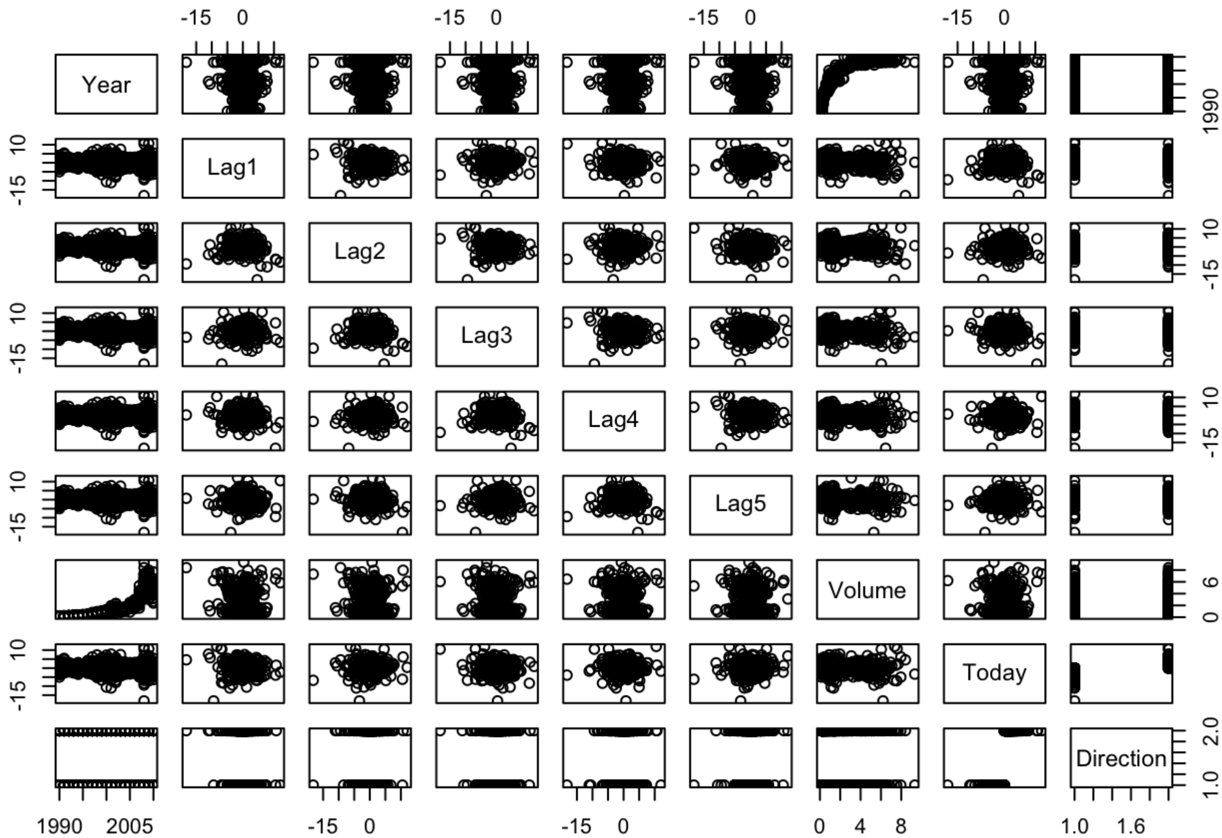
```
##   Year   Lag1   Lag2   Lag3   Lag4   Lag5    Volume  Today Direction
## 1 1990  0.816  1.572 -3.936 -0.229 -3.484 0.1549760 -0.270      Down
## 2 1990 -0.270  0.816  1.572 -3.936 -0.229 0.1485740 -2.576      Down
## 3 1990 -2.576 -0.270  0.816  1.572 -3.936 0.1598375  3.514        Up
## 4 1990  3.514 -2.576 -0.270  0.816  1.572 0.1616300  0.712        Up
## 5 1990  0.712  3.514 -2.576 -0.270  0.816 0.1537280  1.178        Up
## 6 1990  1.178  0.712  3.514 -2.576 -0.270 0.1544440 -1.372      Down
```

a. Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns

```
summary(Weekly)
```

```
##       Year           Lag1               Lag2               Lag3
##  Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950
##  1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580
##  Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410
##  Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472
##  3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090
##  Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260
##       Lag4               Lag5              Volume
##  Min.   :-18.1950   Min.   :-18.1950   Min.   :0.08747
##  1st Qu.: -1.1580   1st Qu.: -1.1660   1st Qu.:0.33202
##  Median :  0.2380   Median :  0.2340   Median :1.00268
##  Mean   :  0.1458   Mean   :  0.1399   Mean   :1.57462
##  3rd Qu.:  1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373
##  Max.   : 12.0260   Max.   : 12.0260   Max.   :9.32821
##      Today          Direction
##  Min.   :-18.1950   Down:484
##  1st Qu.: -1.1540   Up  :605
##  Median :  0.2410
##  Mean   :  0.1499
##  3rd Qu.:  1.4050
##  Max.   : 12.0260
```

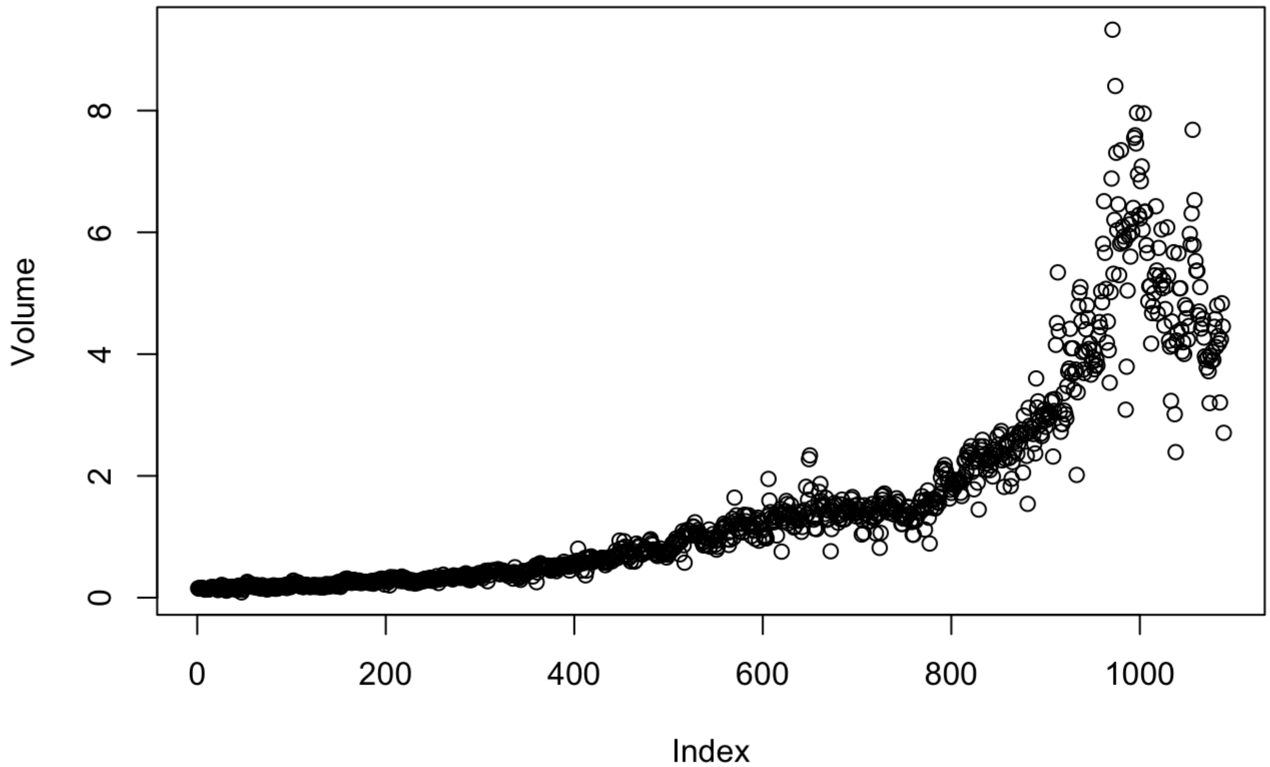
```
pairs(Weekly)
```




```
cor(Weekly[, -9])
```

```
##               Year         Lag1        Lag2        Lag3         Lag4
## Year    1.00000000 -0.032289274 -0.03339001 -0.03000649 -0.031127923
## Lag1   -0.03228927  1.000000000 -0.07485305  0.05863568 -0.071273876
## Lag2   -0.03339001 -0.074853051  1.00000000 -0.07572091  0.058381535
## Lag3   -0.03000649  0.058635682 -0.07572091  1.00000000 -0.075395865
## Lag4   -0.03112792 -0.071273876  0.05838153 -0.07539587  1.000000000
## Lag5   -0.03051910 -0.008183096 -0.07249948  0.06065717 -0.075675027
## Volume  0.84194162 -0.064951313 -0.08551314 -0.06928771 -0.061074617
## Today  -0.03245989 -0.075031842  0.05916672 -0.07124364 -0.007825873
##                Lag5      Volume        Today
## Year   -0.030519101  0.84194162 -0.032459894
## Lag1   -0.008183096 -0.06495131 -0.075031842
## Lag2   -0.072499482 -0.08551314  0.059166717
## Lag3    0.060657175 -0.06928771 -0.071243639
## Lag4   -0.075675027 -0.06107462 -0.007825873
## Lag5    1.000000000 -0.05851741  0.011012698
## Volume -0.058517414  1.00000000 -0.033077783
## Today   0.011012698 -0.03307778  1.000000000
```

```
attach(Weekly)
plot(Volume) #Volume increased with time
```



From the pairs(Weekly) plot it was clear that only Volume showed a proper trend with respect to the year attribute. (The volume increases with year)

b. Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

```
glm = glm(Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data = Weekly,
              family = binomial)
summary(glm)
```

```
##
## Call:
## glm(formula = Direction ~ Lag1 + Lag2 + Lag3 + Lag4 + Lag5 +
##     Volume, family = binomial, data = Weekly)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## Lag1        -0.04127    0.02641  -1.563   0.1181
## Lag2         0.05844    0.02686   2.175   0.0296 *
## Lag3        -0.01606    0.02666  -0.602   0.5469
## Lag4        -0.02779    0.02646  -1.050   0.2937
## Lag5        -0.01447    0.02638  -0.549   0.5833
## Volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

Lag2 has its p-value is less than 0.05 and thus is statistically significant.

c. Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.

```
glm_init = predict(glm, type = "response")
glm_predict = rep("Down", length(glm_init))
glm_predict[glm_init > 0.5] = "Up"
table(glm_predict, Direction)
```

```
##            Direction
## glm_predict Down  Up
##        Down   54  48
##        Up    430 557
```

Percentage of correct predictions on the training data is given by the diagonal elements. (54+557)/1089 =56.1065197%. For weeks when the market goes up, the model is right 92.06% of the time (557/(48+557)). For weeks when the market goes down, the model is right only 11.15% of the time (54/(54+430)).

d. Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).

```
train <- (Year < 2009)
WeeklyTrim <- Weekly[!train, ]
DirectionTrim <- Direction[!train]
fglm2 <- glm(Direction ~ Lag2, data = Weekly, family = binomial, subset = train)
summary(fglm2)
```

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = binomial, data = Weekly,
##     subset = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.536  -1.264   1.021   1.091   1.368
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.20326    0.06428   3.162  0.00157 **
## Lag2         0.05810    0.02870   2.024  0.04298 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1354.5
##
## Number of Fisher Scoring iterations: 4
```

```
mod2 <- predict(fglm2, WeeklyTrim, type = "response")
predict_glm2 <- rep("Down", length(mod2))
predict_glm2[mod2 > 0.5] <- "Up"
table(predict_glm2, DirectionTrim)
```

```
##             DirectionTrim
## predict_glm2 Down Up
##         Down    9  5
##         Up     34 56
```

Percentage of correct predictions on the test data is (9+56)/104= 62.5% For weeks when the market goes up, the model is right 91.80% of the time (56/(56+5)). For weeks when the market goes down, the model is right only 20.93% of the time (9/(9+34)).

e. LDA

```
library(MASS)
lda <- lda(Direction ~ Lag2, data = Weekly, subset = train)
lda
```

```
## Call:
## lda(Direction ~ Lag2, data = Weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
##
## Coefficients of linear discriminants:
##            LD1
## Lag2 0.4414162
```

```
predict_lda <- predict(lda, WeeklyTrim)
table(predict_lda$class, DirectionTrim)
```

```
##       DirectionTrim
##        Down Up
##   Down    9  5
##   Up     34 56
```

Percentage of correct predictions on the test data is 62.5%. For weeks when the market goes up, the model is right 91.80% of the time. For weeks when the market goes down, the model is right only 20.93%of the time.

f. QDA

```
qda <- qda(Direction ~ Lag2, data = Weekly, subset = train)
qda
```

```
## Call:
## qda(Direction ~ Lag2, data = Weekly, subset = train)
##
## Prior probabilities of groups:
##      Down        Up
## 0.4477157 0.5522843
##
## Group means:
##             Lag2
## Down -0.03568254
## Up    0.26036581
```

```
predict_qda <- predict(qda, WeeklyTrim)
table(predict_qda$class, DirectionTrim)
```

```
##       DirectionTrim
##        Down Up
##   Down    0  0
##   Up     43 61
```

Percentage of correct predictions on the test data is 58.65%. For weeks when the market goes up, the model is right 100% of the time. For weeks when the market goes down, the model is right 0% of the time.

g. KNN

```
library(class)
trainX <- as.matrix(Lag2[train])
testX <- as.matrix(Lag2[!train])
trainDir <- Direction[train]
set.seed(1)
predict_knn <- knn(trainX, testX, trainDir, k = 1)
table(predict_knn, DirectionTrim)
```

```
##            DirectionTrim
## predict_knn Down Up
##        Down   21 30
##        Up     22 31
```

Percentage of correct predictions on the test data is 50%. For weeks when the market goes up, the model is right 50.82% of the time. For weeks when the market goes down, the model is right only 48.83% of the time.

h. Comparison

Decreasing order of error rate: 1. Logistic Regression 2. LDA 3. QDA 4. KNN Thus, in our scenario, Logistic regression and LDA performed equally well followed by QDA and KNN.