PH.140.644_HW1

Kunyu An

Chapter 3

Q5 We want to show that $\hat{y}_i = \sum_{i'=1}^{n} a_{i'} y_{i'}$. Given that $\hat{\beta} = (\sum_{i=1}^{n} x_i y_i)/(\sum_{i'=1}^{n} x_{i'}^2)$, we have

$$\hat{y}_i = x_i \hat{\beta} = x_i \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i'=1}^{n} x_{i'}^2} = \frac{x_i}{\sum_{i'=1}^{n} x_{i'}^2} \sum_{i=1}^{n} x_i y_i$$

Since $\frac{x_i}{\sum_{i'=1}^{n} x_{i'}^2}$ is a constant for any i', we can represent it as a constant $C_{i'}$. Then we get,

$$\hat{y}_i = C_{i'} \sum_{i=1}^{n} x_i y_i = \sum_{i=1}^{n} C_{i'} x_i y_i$$

Let $a_{i'} = C_{i'} x_i$, we get,

$$\hat{y}_i = \sum_{i=1}^{n} C_{i'} x_i y_i = \sum_{i'=1}^{n} a_{i'} y_{i'}$$

Q6 For simple linear regression, $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$. According to (3.4), $\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$.Thus, we have,

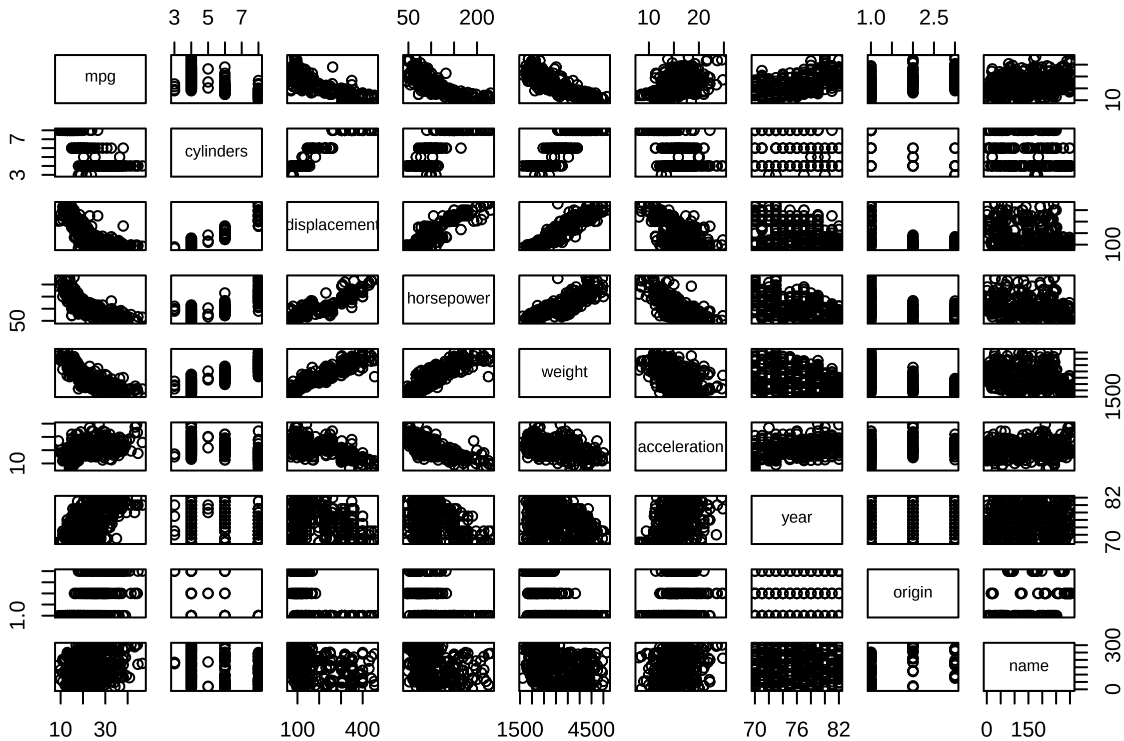
$$\hat{y} = \bar{y} - \hat{\beta}_1 \bar{x} + \hat{\beta}_1 x = \bar{y} - \hat{\beta}_1 (x - \bar{x})$$

Let $x = \bar{x}$, we get $\hat{y} = \bar{y}$. Thus, we can say that the least squares line always passes through the point $(\bar{x}, \bar{y})$.

Q9

a. Produce a scatterplot matrix which includes all of the variables in the data set.

```
pairs(Auto)
```



b. Compute the matrix of correlations between the variables using the function cor(). You will need to exclude the name variable, which is qualitative.

```
names(Auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
## [6] "acceleration" "year"         "origin"       "name"
```

```
new_data = Auto[1:8]
cor(new_data)
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
```

```
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

c. Use the lm() function to perform a multiple linear regression with mpg as the response and all other variables except name as the predictors. Use the summary() function to print the results. Comment on the output. For instance:

```
fit_mpg <- lm(mpg~., data=new_data)
summary(fit_mpg)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = new_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

i. Is there a relationship between the predictors and the response?

Yes. Because F-static is large and p-value is small, we can say there is there a relationship between the predictors and the response.

ii. Which predictors appear to have a statistically significant relationship to the response?

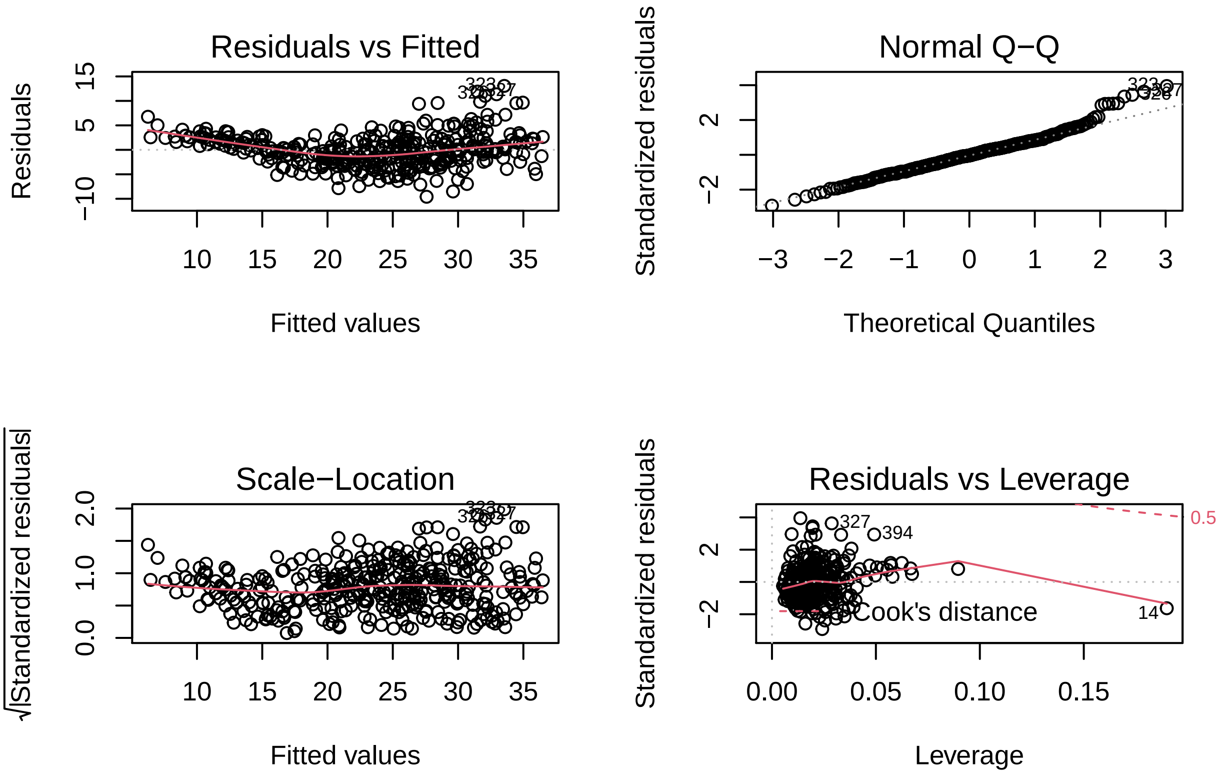
Year, weight, origin and displacement appear to have a statistically significant relationship to the response.

iii. What does the coefficient for the year variable suggest?

The positive coefficient for year indicates year and mpg have the same trend. In other words, we can get larger value of mpg with larger value of year.

d. Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

```r
par(mfrow=c(2,2))
plot(fit_mpg)
```



Point 320, point 323 and point 327 have unusually large residual. Point 14 has unusually high leverage.

e. Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

```r
lm.fit2 = lm(mpg~cylinders*displacement+displacement*weight, data=Auto)
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders * displacement + displacement *
##     weight, data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.2934  -2.5184  -0.3476   1.8399  17.7723
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             5.262e+01  2.237e+00  23.519  < 2e-16 ***
## cylinders               7.606e-01  7.669e-01   0.992    0.322
```

```
## displacement           -7.351e-02  1.669e-02  -4.403 1.38e-05 ***
## weight                 -9.888e-03  1.329e-03  -7.438 6.69e-13 ***
## cylinders:displacement -2.986e-03  3.426e-03  -0.872    0.384
## displacement:weight     2.128e-05  5.002e-06   4.254 2.64e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.103 on 386 degrees of freedom
## Multiple R-squared:  0.7272, Adjusted R-squared:  0.7237
## F-statistic: 205.8 on 5 and 386 DF,  p-value: < 2.2e-16
```

According to the correlation matrix and the p-values, we can see that the interaction between displacement and weight is statistically significant, while the interaction between cylinders and displacement is not.

f. Try a few different transformations of the variables, such as log(X), √X, X2. Comment on your findings.

```
lm.fit3 = lm(mpg~log(weight)+sqrt(horsepower), data=Auto)
summary(lm.fit3)
```

```
##
## Call:
## lm(formula = mpg ~ log(weight) + sqrt(horsepower), data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1029  -2.5380  -0.4015   2.1391  15.6049
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      167.7882     9.6088  17.462  < 2e-16 ***
## log(weight)      -16.5530     1.4473 -11.437  < 2e-16 ***
## sqrt(horsepower)  -1.2514     0.2277  -5.496 7.05e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.041 on 389 degrees of freedom
## Multiple R-squared:  0.7334, Adjusted R-squared:  0.732
## F-statistic:   535 on 2 and 389 DF,  p-value: < 2.2e-16
```

The variables log(weight), sqrt(horsepower) have statistical significance and have good performance in regression.

Q15 This problem involves the Boston data set. 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.

We first launch our dataset and take a first look of our dataset.

```r
library(MASS)
attach(Boston)
head(Boston)
```

```
##      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
```

```r
fit_zn = lm(crim~zn, data = Boston)
summary(fit_zn)
```

```
##
## Call:
## lm(formula = crim ~ zn, data = Boston)
##
## 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
```

For the predictor "zn", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "zn" and "crim".

```r
fit_indus = lm(crim~indus, data = Boston)
summary(fit_indus)
```

```
##
## Call:
## lm(formula = crim ~ indus, data = Boston)
##
```

```
## 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
```

For the predictor "indus", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "indus" and "crim".

```
fit_chas = lm(crim~chas, data = Boston)
summary(fit_chas)
```

```
##
## Call:
## lm(formula = crim ~ chas, data = Boston)
##
## 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 ***
## chas         -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
```

For the predictor "chas", the p value 0.209 is greater than 0.05, that means we *do not have* enough evidence to conclude there's statistically significant association between "chas" and "crim".

```
fit_nox = lm(crim~nox, data = Boston)
summary(fit_nox)
```

```
##
## Call:
## lm(formula = crim ~ nox, data = Boston)
##
## 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
```

For the predictor "nox", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "nox" and "crim".

```r
fit_rm = lm(crim~rm, data = Boston)
summary(fit_rm)
```

```
##
## Call:
## lm(formula = crim ~ rm, data = Boston)
##
## 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
```

For the predictor "rm", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "rm" and "crim".

```r
fit_age = lm(crim~age, data = Boston)
summary(fit_age)
```

```
##
## Call:
## lm(formula = crim ~ age, data = Boston)
##
## 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
```

For the predictor "age", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "age" and "crim".

```
fit_dis = lm(crim~dis, data = Boston)
summary(fit_dis)
```

```
##
## Call:
## lm(formula = crim ~ dis, data = Boston)
##
## 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
```

For the predictor "dis", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "dis" and "crim".

```
fit_rad = lm(crim~rad, data = Boston)
summary(fit_rad)
```

```
##
## Call:
## lm(formula = crim ~ rad, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.164  -1.381  -0.141   0.660  76.433
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.718 on 504 degrees of freedom
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16
```

For the predictor "rad", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "rad" and "crim".

```
fit_tax = lm(crim~tax, data = Boston)
summary(fit_tax)
```

```
##
## Call:
## lm(formula = crim ~ tax, data = Boston)
##
## 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
```

For the predictor "tax", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "tax" and "crim".

```
fit_ptratio = lm(crim~ptratio, data = Boston)
summary(fit_ptratio)
```

```
##
## Call:
## lm(formula = crim ~ ptratio, data = Boston)
##
## 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
```

For the predictor "ptratio", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "ptratio" and "crim".

```
fit_black = lm(crim~black, data = Boston)
summary(fit_black)
```

```
##
## Call:
## lm(formula = crim ~ black, data = Boston)
##
## 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
```

For the predictor "black", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "black" and "crim".

```
fit_lstat = lm(crim~lstat, data = Boston)
summary(fit_lstat)
```

```
##
## Call:
## lm(formula = crim ~ lstat, data = Boston)
##
## 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
```

For the predictor "lstat", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "lstat" and "crim".

```
fit_medv = lm(crim~medv, data = Boston)
summary(fit_medv)
```

```
##
## Call:
## lm(formula = crim ~ medv, data = Boston)
##
## 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
```

For the predictor "medv", the p value is less than 0.05, that means we have strong evidence that there's statistically significant association between "medv" and "crim".

To sum up, all the variables, except the "chas", have statistically significant association with the respond "crim".

b. Fit a multiple regression model to predict the response using all of the predictors. Describe your results.

```
fit_multi = lm(crim ~ ., data = Boston)
summary(fit_multi)
```

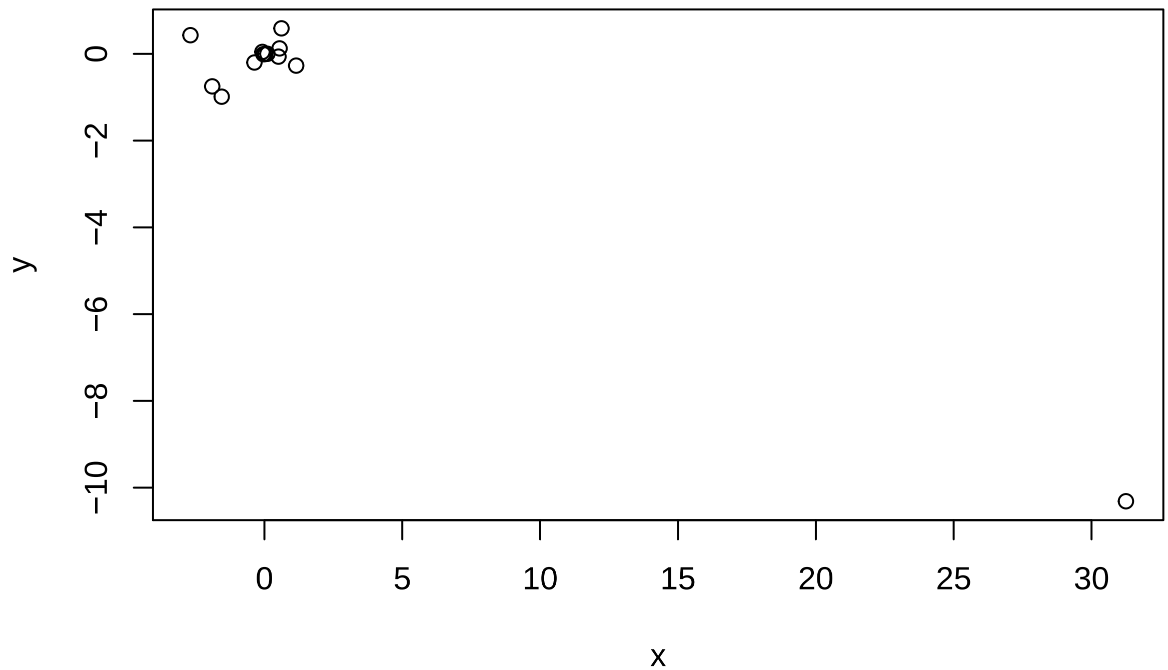
```
##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (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
```

According to the p values shown above, only "zn", "dis", "rad", "black" and "medv" have strong evidence to reject the null hypothesis since their p values are less than 0.05.

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 regres- sion model is shown on the x-axis, and its coefficient estimate in the multiple linear regression model is shown on the y-axis.

```r
x = c(coefficients(fit_zn)[2],
      coefficients(fit_indus)[2],
      coefficients(fit_chas)[2],
      coefficients(fit_nox)[2],
      coefficients(fit_rm)[2],
      coefficients(fit_age)[2],
      coefficients(fit_dis)[2],
      coefficients(fit_rad)[2],
      coefficients(fit_tax)[2],
      coefficients(fit_ptratio)[2],
      coefficients(fit_black)[2],
      coefficients(fit_lstat)[2],
      coefficients(fit_medv)[2])
y = coefficients(fit_multi)[2:14]
plot(x, y)
```



From the figure above, we can see that there are differences between simple and multiple regression coefficients. The simple linear regression only being affected by a single variable but the multiple regression is affected by the average effect of multiple variables

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

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

```
##
## Call:
## lm(formula = crim ~ poly(zn, 3), data = Boston)
##
## 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
```

For the predictor "zn", the p value of quadratic coefficient is less than 0.05 but p values the cubic coefficient is greater than 0.05. That means there's a nonlinear association between "zn" and "crim" but the evidence is not strong.

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

```
##
## Call:
## lm(formula = crim ~ poly(indus, 3), data = Boston)
##
## 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
```

For the predictor "indus", the p value of both quadratic and cubic coefficient is less than 0.05. That means there's strong evidence to show a nonlinear association between "indus" and "crim".

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

```
##
## Call:
## lm(formula = crim ~ poly(nox, 3), data = Boston)
##
## 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
```

For the predictor "nox", the p value of quadratic and cubic coefficient is less than 0.05. That means there's strong evidence to show a nonlinear association between "nox" and "crim".

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

```
##
## Call:
## lm(formula = crim ~ poly(rm, 3), data = Boston)
##
## 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
```

For the predictor "rm", the p value of quadratic coefficient is less than 0.05 but p values the cubic coefficient is greater than 0.05. That means there's a nonlinear association between "rm" and "crim" but the evidence is not strong.

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

```
##
## Call:
## lm(formula = crim ~ poly(age, 3), data = Boston)
##
## 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
```

For the predictor "age", the p value of both quadratic and cubic coefficient is less than 0.05. That means there's strong evidence to show a nonlinear association between "age" and "crim".

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

```
##
## Call:
## lm(formula = crim ~ poly(dis, 3), data = Boston)
##
## 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
```

For the predictor "dis", the p value of both quadratic and cubic coefficient is less than 0.05. That means there's strong evidence to show a nonlinear association between "dis" and "crim".

```r
fit_rad = lm(crim~poly(rad,3), data = Boston)
summary(fit_rad)
```

```
##
## Call:
## lm(formula = crim ~ poly(rad, 3), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
```

```
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.2971  12.164  < 2e-16 ***
## poly(rad, 3)1   120.9074     6.6824  18.093  < 2e-16 ***
## poly(rad, 3)2    17.4923     6.6824   2.618  0.00912 **
## poly(rad, 3)3     4.6985     6.6824   0.703  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
```

For the predictor "rad", the p value of quadratic coefficient is less than 0.05 but p values the cubic coefficient is greater than 0.05. That means there's a nonlinear association between "rad" and "crim" but the evidence is not strong.

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

```
##
## Call:
## lm(formula = crim ~ poly(tax, 3), data = Boston)
##
## 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
```

For the predictor "tax", the p value of quadratic coefficient is less than 0.05 but p values the cubic coefficient is greater than 0.05. That means there's a nonlinear association between "tax" and "crim" but the evidence is not strong.

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

```
##
## Call:
## lm(formula = crim ~ poly(ptratio, 3), data = Boston)
```

```
##
## 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
```

For the predictor "ptratio", the p value of both quadratic and cubic coefficient is less than 0.05. That means there's strong evidence to show a nonlinear association between "ptratio" and "crim".

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

```
##
## Call:
## lm(formula = crim ~ poly(black, 3), data = Boston)
##
## 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
```

For the predictor "black", the p value of both quadratic and cubic coefficient is greater than 0.05. That means there's no enough evidence to conclude a nonlinear association between "black" and "crim".

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

```
##
```

```
## Call:
## lm(formula = crim ~ poly(lstat, 3), data = Boston)
##
## 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
```

For the predictor "lastat", the p value of quadratic coefficient is less than 0.05 but p values the cubic coefficient is greater than 0.05. That means there's a nonlinear association between "lstat" and "crim" but the evidence is not strong.

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

```
##
## Call:
## lm(formula = crim ~ poly(medv, 3), data = Boston)
##
## 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
```

For the predictor "medv", the p value of both quadratic and cubic coefficient is less than 0.05. That means there's strong evidence to show a nonlinear association between "medv" and "crim".

To sum up, for variables "zn", "rm", "rad", "tax" and "lstat", their p-values suggest that the cubic coefficient is not statistically significant.For "indus", "nox", "age", "dis", "ptratio" and

"medv", their p-values suggest there's strong evidence to show a nonlinear association; for the variable "black", the p-values suggest that both quandratic and cubic coefficients are not statistically significant and thus there is no nonlinear association.

Chapter 4

Q1 We can start with the logistic function representation,

$$p(X) = \frac{e^{\beta_0 + \beta_1^X}}{1+e^{\beta_0+\beta_1^X}}$$

Multiplying both side by $1 + e^{\beta_0} + \beta_1^X$, we get

$$e^{\beta_0+\beta_1^X} = p(X)(1 + e^{\beta_0+\beta_1^X}) = p(X) + p(X)e^{\beta_0+\beta_1^X}$$

By subtract $p(X)e^{\beta_0+\beta_1^X}$ on both side, we get

$$e^{\beta_0+\beta_1^X} - p(X)e^{\beta_0+\beta_1^X} = e^{\beta_0+\beta_1^X}(1 - p(X)) = p(X)$$

which can be simplified as,

$$\frac{p(X)}{1-p(X)} = e^{\beta_0+\beta_1^X} \text{ (the logit representation)}$$

Q8 For KNN test, if K = 1, the training error should always be 0 and since the average error is 18%, the actually test error should be 36%, which is greatly than the 30% test error of logistic regression. Thus, we should use logistic regression for this dataset rather than KNN with K =1.

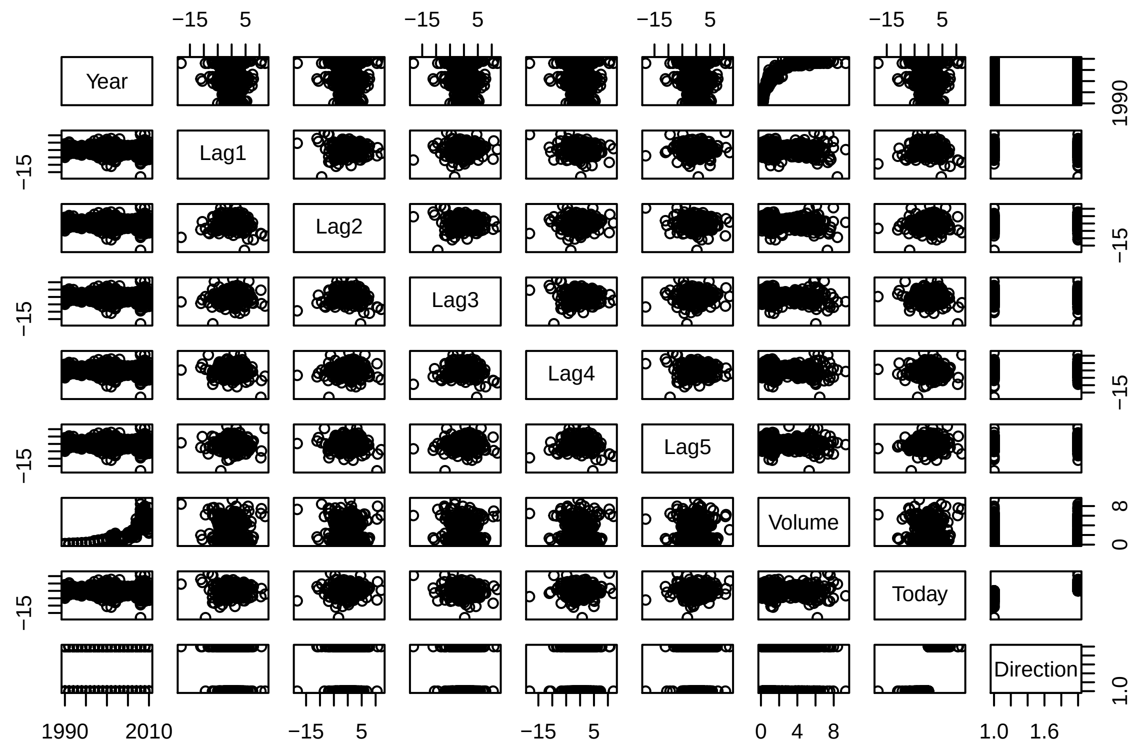
Q10

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

```
names(Weekly)
```

```
## [1] "Year"      "Lag1"      "Lag2"      "Lag3"      "Lag4"      "Lag5"
## [7] "Volume"    "Today"     "Direction"
```

```
pairs(Weekly)
```




```r
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
```

According to the correlations matrix, only Year and Volume have strong correlation. There are only weak relationships between the Lag variables.

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.fit = glm(Direction ~ .-Year-Today, data=Weekly, family=binomial)
summary(glm.fit)
```

```
##
## Call:
## glm(formula = Direction ~ . - Year - Today, 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 is the statistically significant predictor, with p-value of 0.0296.

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.fit.probs = predict(glm.fit, type = "response")
glm.fit.pred = rep("Down", length(Weekly$Direction))
glm.fit.pred[glm.fit.probs > 0.5] = "Up"
table(glm.fit.pred, Weekly$Direction)
```

```
##
## glm.fit.pred Down  Up
##         Down   54  48
##         Up    430 557
```

```
mean(glm.fit.pred == Weekly$Direction)
```

```
## [1] 0.5610652
```

We only have an accuracy of 56.1%, we predict UP more often than Down. The model doesn't predict the negative class well.

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 = Weekly$Year <= 2008
test = !train
glm.fit = glm(Direction ~ Lag2, data=Weekly, family=binomial, subset=train)
glm.fit.prob = predict(glm.fit, Weekly[test, ], type="response")
glm.fit.pred = rep("Down", length(Weekly$Direction[test]))
glm.fit.pred[glm.fit.prob > 0.5] = "Up"
table(glm.fit.pred, Weekly$Direction[test])
```

```
##
## glm.fit.pred Down Up
##         Down    9  5
##         Up     34 56
```

```
mean(glm.fit.pred == Weekly$Direction[test])
```

```
## [1] 0.625
```

e. Repeat (d) using LDA.

```
lda.fit = lda(Direction ~ Lag2, data=Weekly, subset=train)
lda.pred = predict(lda.fit, Weekly[test,])$class
table(lda.pred, Weekly$Direction[test])
```

```
##
## lda.pred Down Up
##     Down    9  5
##     Up     34 56
```

```
mean(lda.pred == Weekly$Direction[test])
```

```
## [1] 0.625
```

f. Repeat (d) using QDA.

```
qda.fit = qda(Direction ~ Lag2, data=Weekly, subset=train)
qda.pred = predict(qda.fit, Weekly[test,])$class
table(qda.pred, Weekly$Direction[test])
```

```
##
## qda.pred Down Up
##     Down    0  0
##     Up     43 61
```

```
mean(qda.pred == Weekly$Direction[test])
```

```
## [1] 0.5865385
```

g. Repeat (d) using KNN with K = 1.

```
set.seed(1)
knn.pred = knn(data.frame(Weekly$Lag2[train]), data.frame(Weekly$Lag2[test]), Weekly$Direction[train], 
table(knn.pred, Weekly$Direction[test])
```

```
##
## knn.pred Down Up
##     Down   21 30
##     Up     22 31
```

```
mean(knn.pred == Weekly$Direction[test])
```

```
## [1] 0.5
```

h. Which of these methods appears to provide the best results on this data?

"Logistic regression" and "LDA" have the test accuracy of 0.625 that provide the best results on this data.

i. Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for K in the KNN classifier.

```
train = Weekly$Year <= 2008
test = !train
glm.fit = glm(Direction ~ Lag2:Lag1, data=Weekly, family=binomial, subset=train)
glm.fit.prob = predict(glm.fit, Weekly[test, ], type="response")
glm.fit.pred = rep("Down", length(Weekly$Direction[test]))
glm.fit.pred[glm.fit.prob > 0.5] = "Up"
table(glm.fit.pred, Weekly$Direction[test])
```

```
##
## glm.fit.pred Down Up
##         Down    1  1
##         Up     42 60
```

```r
mean(glm.fit.pred == Weekly$Direction[test])
```

```
## [1] 0.5865385
```

```r
lda.fit = lda(Direction ~ Lag2:Lag1, data=Weekly, subset=train)
lda.pred = predict(lda.fit, Weekly[test,])$class
table(lda.pred, Weekly$Direction[test])
```

```
##
## lda.pred Down Up
##     Down    0  1
##     Up     43 60
```

```r
mean(lda.pred == Weekly$Direction[test])
```

```
## [1] 0.5769231
```

```r
set.seed(1)
knn.pred = knn(data.frame(Weekly$Lag2[train]), data.frame(Weekly$Lag2[test]), Weekly$Direction[train], 
table(knn.pred, Weekly$Direction[test])
```

```
##
## knn.pred Down Up
##     Down   16 20
##     Up     27 41
```

```r
mean(knn.pred == Weekly$Direction[test])
```

```
## [1] 0.5480769
```

```r
set.seed(1)
knn.pred = knn(data.frame(Weekly$Lag2[train]), data.frame(Weekly$Lag2[test]), Weekly$Direction[train], 
table(knn.pred, Weekly$Direction[test])
```

```
##
## knn.pred Down Up
##     Down   17 20
##     Up     26 41
```

```r
mean(knn.pred == Weekly$Direction[test])
```

```
## [1] 0.5576923
```

```r
set.seed(1)
knn.pred = knn(data.frame(Weekly$Lag2[train]), data.frame(Weekly$Lag2[test]), Weekly$Direction[train], 
table(knn.pred, Weekly$Direction[test])
```

```
##
## knn.pred Down Up
##     Down   21 21
##     Up     22 40
```

```
mean(knn.pred == Weekly$Direction[test])
```

```
## [1] 0.5865385
```

The original LDA and logistic regression have better performance in terms of accuracy rate. Although we increase the value of k in KNN, the results don't change obviously.

Q11

a. Create a binary variable, `mpg01`, that contains a 1 if `mpg` contains a value above its median, and a 0 if `mpg` contains a value below its median. You can compute the median using the `median()` function. Note you may find it helpful to use the `data.frame()` function to create a single data set containing both `mpg01` and the other `Auto` variables.

```
library(ISLR)
attach(Auto)
mpg01 = rep(0, length(mpg))
mpg01[mpg > median(mpg)] = 1
Auto = data.frame(Auto, mpg01)
head(Auto)
```

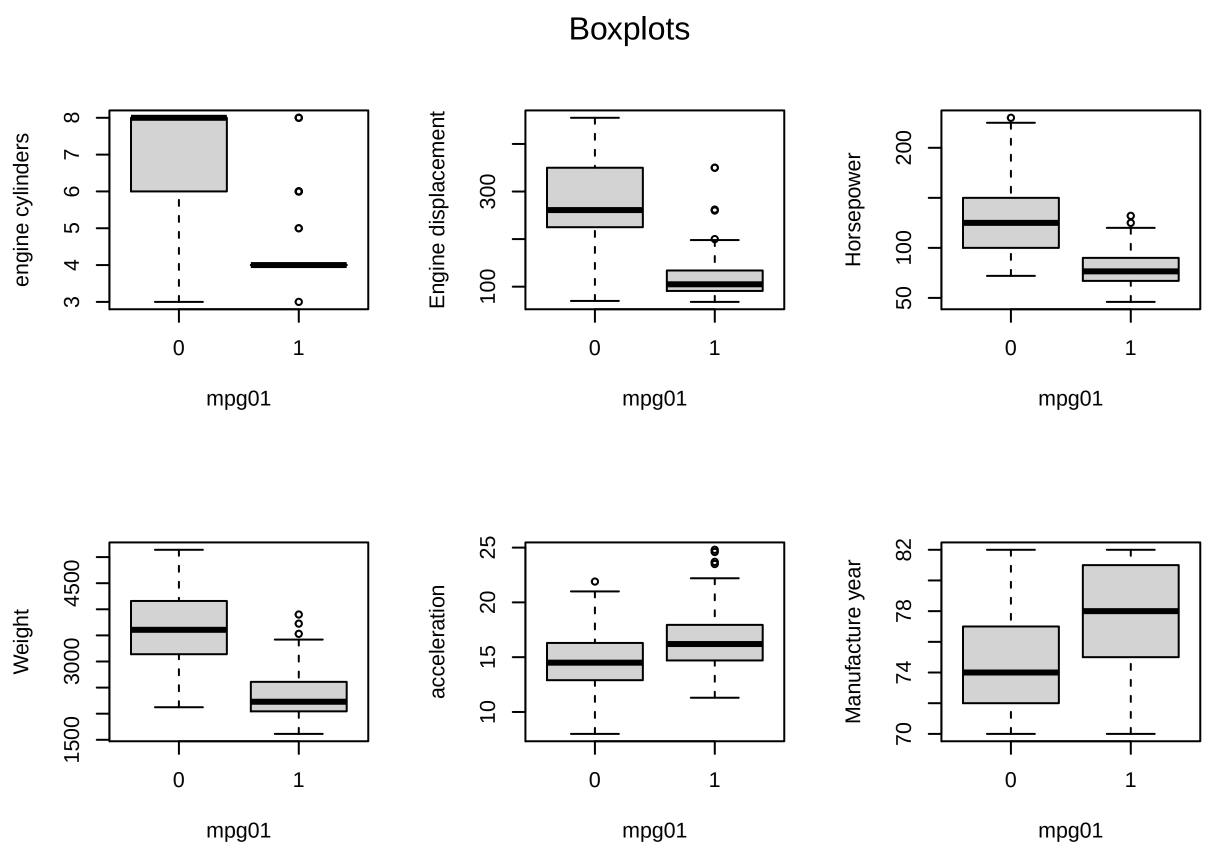
```
##   mpg cylinders displacement horsepower weight acceleration year origin
## 1  18         8          307        130   3504         12.0   70      1
## 2  15         8          350        165   3693         11.5   70      1
## 3  18         8          318        150   3436         11.0   70      1
## 4  16         8          304        150   3433         12.0   70      1
## 5  17         8          302        140   3449         10.5   70      1
## 6  15         8          429        198   4341         10.0   70      1
##                        name mpg01
## 1 chevrolet chevelle malibu     0
## 2         buick skylark 320     0
## 3        plymouth satellite     0
## 4             amc rebel sst     0
## 5               ford torino     0
## 6          ford galaxie 500     0
```

b.Explore the data graphically in order to investigate the association between `mpg01` and the other features. Which of the other features seem most likely to be useful in predicting `mpg01`? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.

We first can plot the boxplots for each variable.

```
par(mfrow = c(2, 3))
plot(factor(mpg01), cylinders, xlab = "mpg01", ylab = "engine cylinders")
plot(factor(mpg01), displacement, xlab = "mpg01",ylab = "Engine displacement")
plot(factor(mpg01), horsepower,xlab = "mpg01",ylab = "Horsepower")
plot(factor(mpg01), weight, xlab = "mpg01",ylab = "Weight")
```

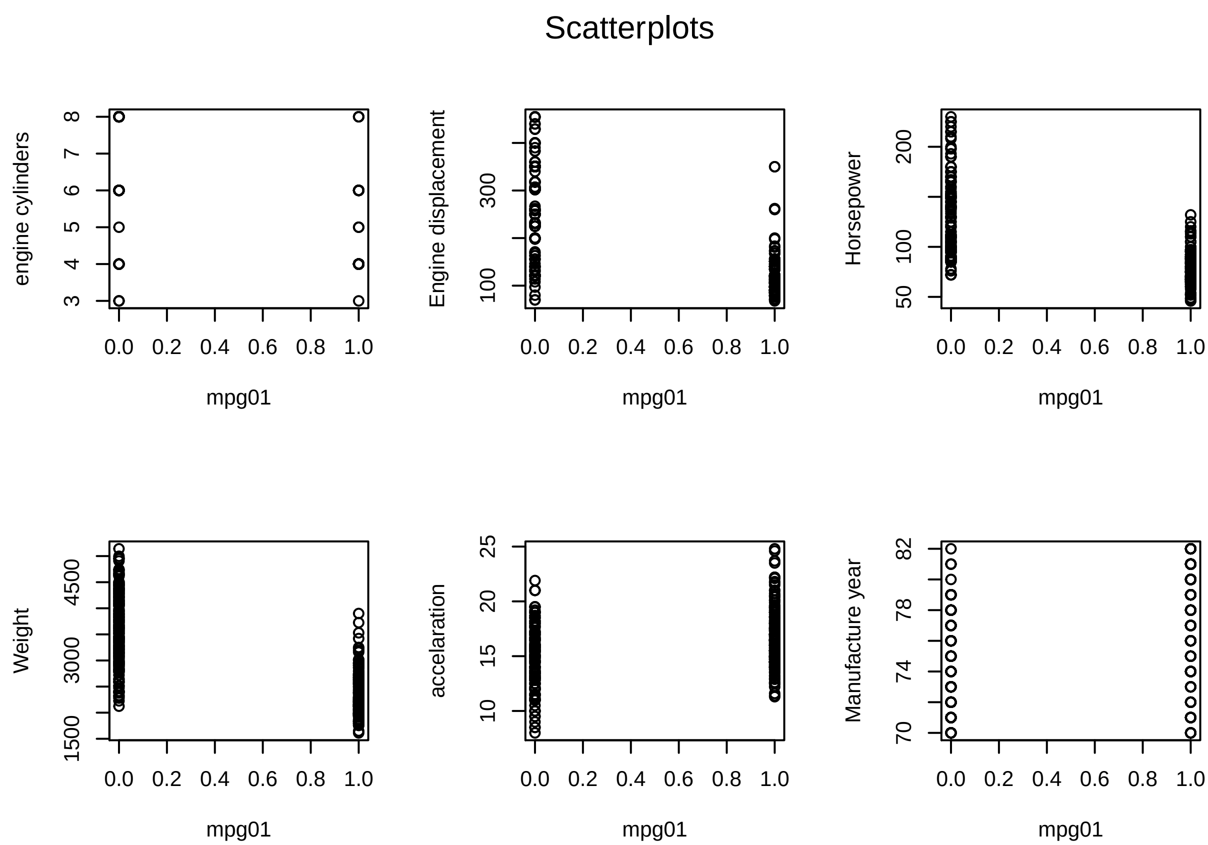
```
plot(factor(mpg01), acceleration, xlab = "mpg01",ylab = "acceleration")
plot(factor(mpg01), year, xlab = "mpg01", ylab = "Manufacture year")
mtext("Boxplots", outer = TRUE, line = -1.5)
```



According the boxplots above, we can see that the predictor `cylinders`,`displacement`,`horsepower` and `weight`are strongly associate with `mpg01`.

Then we can plot the scartterplox for more information.

```
par(mfrow = c(2, 3))
plot(mpg01,cylinders, xlabel = "mpg01",ylab = "engine cylinders")
plot(mpg01,displacement,xlabel = "mpg01",ylab = "Engine displacement")
plot(mpg01,horsepower,xlabel = "mpg01",ylab = "Horsepower")
plot(mpg01,weight, xlabel = "mpg01",ylab = "Weight")
plot(mpg01,acceleration, xlabel = "mpg01",ylab = "accelaration")
plot(mpg01,year,xlabel = "mpg01",ylab = "Manufacture year")
mtext("Scatterplots", outer = TRUE, line = -1.5)
```



From the scatterplot above, the predictor`displacement`, `horsepower` and `weight` seem to be good variables to predict `mpg01`.

To sum up, `cylinder`,`displacement`, `horsepower` and `weight` are the strongest variable to predict `mpg0`.

c.Split the data into a training set and a test set.

Split the Auto data set into 75% trainning sample and 25% testing sample with no replacement.

```
set.seed(1)
index = sample.int(n = nrow(Auto), size = floor(.75*nrow(Auto)), replace = F)
train = Auto[index, ]
test  = Auto[-index, ]
```

d.Perform LDA

```
library(MASS)
lda.fit = lda(mpg01~cylinders+weight+displacement+horsepower,data = Auto,subset = index)
lda_pred = predict(lda.fit, test)
mean(lda_pred$class != Auto[-index, "mpg01"])
```

```
## [1] 0.1326531
```

The test error of LDA is 13.27%

e.Perform QDA

```
library(MASS)
qda.fit = qda(mpg01~cylinders+weight+displacement+horsepower,data = Auto,subset = index)
qda_pred = predict(qda.fit, test)
mean(qda_pred$class != Auto[-index, "mpg01"])
```

```
## [1] 0.122449
```

The test error of QDA is 12.24%

f.Perform logistic regression

```
glm.fit = glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, family = binomial, s
glm.probs = predict(glm.fit, test, type = "response")
glm.pred = rep(0, length(glm.probs))
glm.pred[glm.probs > 0.5] = 1
mean(glm.pred != Auto[-index, "mpg01"])
```

```
## [1] 0.1020408
```

The test error for logistic regression is 10.20%

g. Perform KNN with several values of K.

First try k = 1.

```
library(class)
standardized.X=scale(Auto[, -c(8, 9, 10)])
col = c("cylinders", "displacement", "horsepower", "weight")
train.X=standardized.X[index , col]
test.X=standardized.X[-index , col]
train.Y = Auto[index, "mpg01"]
set.seed(1)
knn.pred = knn(train.X, test.X, train.Y, k = 1)
mean(knn.pred != Auto[-index, "mpg01"])
```

```
## [1] 0.1020408
```

The test error for knn when k = 1 is 10.20%

Then we can try k = 3.

```
knn.pred = knn(train.X, test.X, train.Y, k = 3)
mean(knn.pred != Auto[-index, "mpg01"])
```

```
## [1] 0.1326531
```

The test error for knn when k = 3 is 13.27%

Then try k = 5.

```r
knn.pred = knn(train.X, test.X, train.Y, k = 5)
mean(knn.pred != Auto[-index, "mpg01"])
```

```
## [1] 0.1428571
```

The test error for knn when k = 5 is 14.29%

Try k = 10

```r
knn.pred = knn(train.X, test.X, train.Y, k = 10)
mean(knn.pred != Auto[-index, "mpg01"])
```

```
## [1] 0.1326531
```

The test error for knn when k = 10 is 12.24%

K = 1 works better for KNN.