Code

HW 12 regression in r

Problem 1

Alumno: Daniel Nuño, daniel.nuno@iteso.mx (mailto:daniel.nuno@iteso.mx)

Alumno: David Cisneros

Alumno: Juan Maro Ochoa

Alumno: Rodrigo Huerta

4/18/2022

Chapter 3, exercise 8

This question involves the use of simple linear regression on the `Auto` data set.

a. Use the `lm()` function to perform a simple linear regression with mpg as the response and `horsepower` as the predictor. Use the `summary()` function to print the results. Comment on the output. For example:

Hide

```
summary(Auto)
```

```
      mpg          cylinders      displacement     horsepower        weight      acc
eleration         year            origin
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.
: 8.00   Min.   :70.00   Min.   :1.000
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225   1st
Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804   Medi
an :15.50   Median :76.00   Median :1.000
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978   Mean
:15.54   Mean   :75.98   Mean   :1.577
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615   3rd
Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.
:24.80   Max.   :82.00   Max.   :3.000

                 name         mpg01
 amc matador       :  5   Min.   :0.0
 ford pinto        :  5   1st Qu.:0.0
 toyota corolla    :  5   Median :0.5
 amc gremlin       :  4   Mean   :0.5
 amc hornet        :  4   3rd Qu.:1.0
 chevrolet chevette:  4   Max.   :1.0
 (Other)           :365
```

Hide

```
lm.fit = lm(mpg ~ horsepower)
summary(lm.fit)
```

```
Call:
lm(formula = mpg ~ horsepower)

Residuals:
     Min       1Q   Median       3Q      Max 
-13.5710  -3.2592  -0.3435   2.7630  16.9240 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 39.935861   0.717499   55.66   <2e-16 ***
horsepower  -0.157845   0.006446  -24.49   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.906 on 390 degrees of freedom
Multiple R-squared:  0.6059,	Adjusted R-squared:  0.6049 
F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

I. is there a relationship between the predictor and the response?

By testing the null hypothesis of all regression coefficients equal to zero, it shows a relationship between horsepower and mpg. Since the F-statistic is far larger than 1 and the p-value of the F-statistic is close to zero we can reject the null hypothesis and state there is a statistically significant relationship between horsepower and mpg.

II.How strong is the relationship between the predictor and the response?

The RSE of the lm.fit was 4.906 which indicates a percentage error of 20.9248%. The R2 of the lm.fit was about 0.6059, meaning 60.5948% of the variance in mpg is explained by horsepower.

III. Is the relationship between the predictor and the response positive or negative?

The relationship between mpg and horsepower is negative. The more horsepower an automobile has the linear regression indicates the less mpg fuel efficiency the automobile will have.

IV. What is the predicted mpg associated with a horsepower of 98? What are the associated 95 % confidence and prediction intervals?

Hide

```
predict(lm.fit, data.frame(horsepower=c(98)), interval="confidence")
```

```
       fit      lwr      upr
1 24.46708 23.97308 24.96108
```

Hide

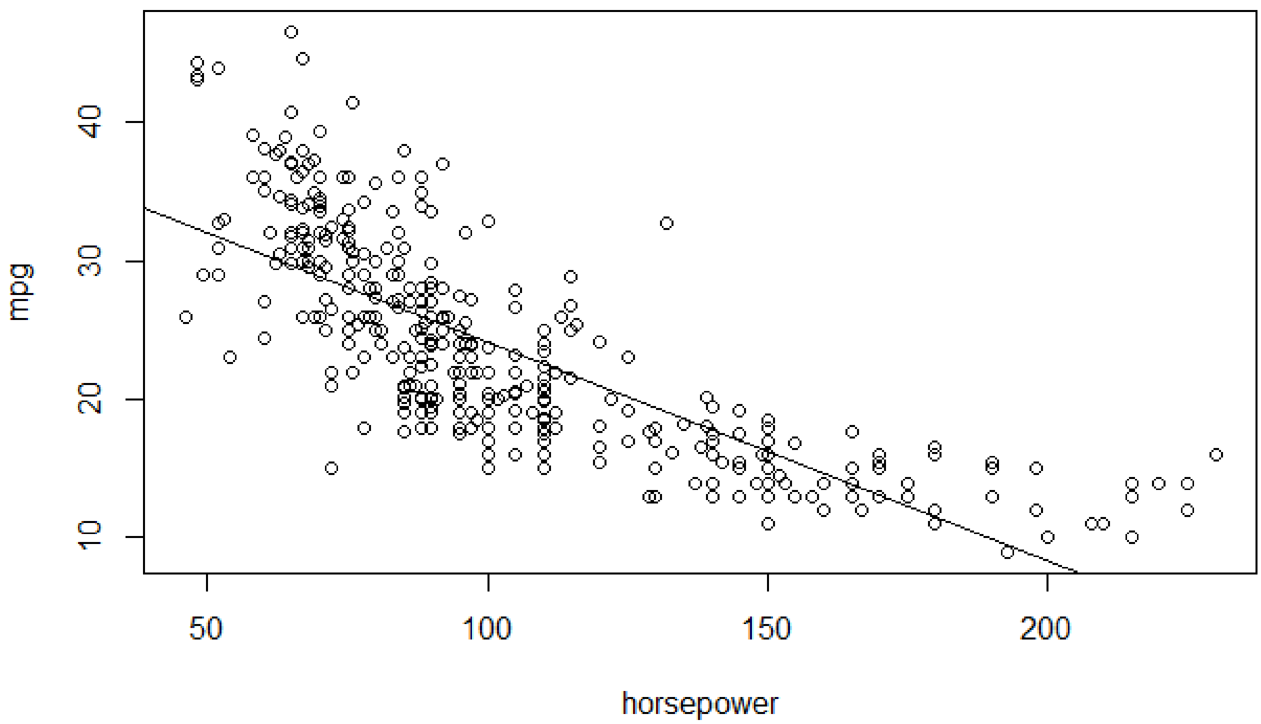
```
predict(lm.fit, data.frame(horsepower=c(98)), interval="prediction")
```

```
       fit      lwr      upr
1 24.46708 14.8094 34.12476
```

b. Plot the response and the predictor. Use the abline() function to display the least squares regression line.

Hide

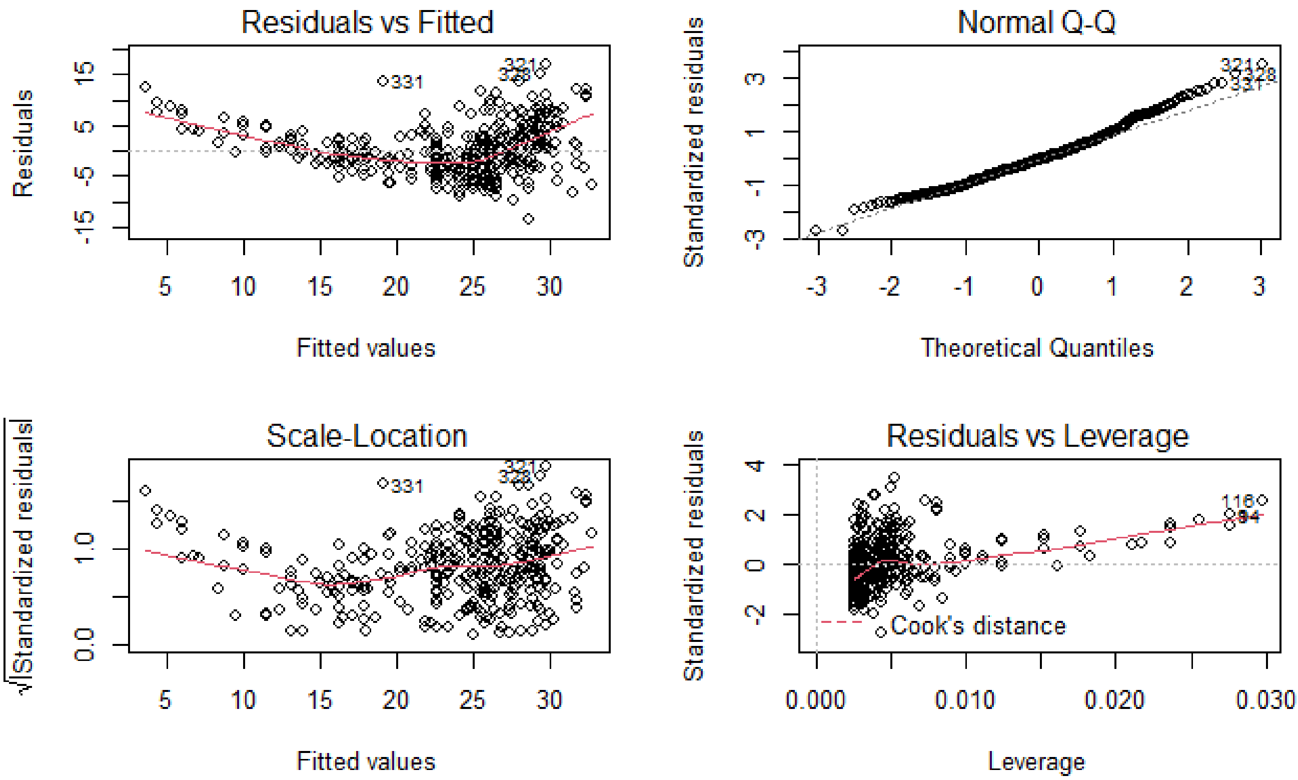
```
plot(horsepower, mpg)
abline(lm.fit)
```



c. Use the plot() function to produce diagnostic plots of the least squares regression fit. Comment on any problems you see with the fit.

Hide

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



Base on the above plots, the is evidence of non-linearity.

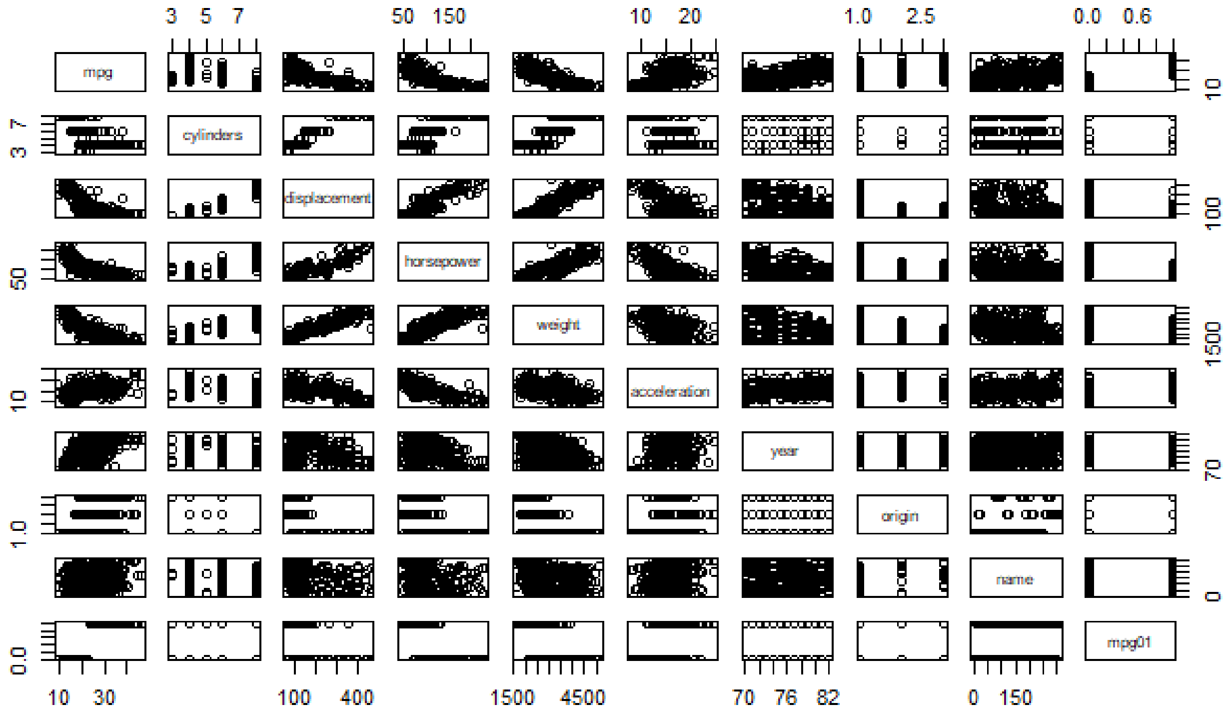
Chapter 3, exercise 9

This question involves the use of multiple linear regression on the `Auto` data set.

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

Hide

```
pairs(Auto)
```



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

Hide

```
cor(subset(Auto, select=-name))
```

```
                    mpg  cylinders displacement horsepower     weight acceleration
year     origin      mpg01
mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442    0.4233285
0.5805410  0.5652088  0.8369392
cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273   -0.5046834 -
0.3456474 -0.5689316 -0.7591939
displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944   -0.5438005 -
0.3698552 -0.6145351 -0.7534766
horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377   -0.6891955 -
0.4163615 -0.4551715 -0.6670526
weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000   -0.4168392 -
0.3091199 -0.5850054 -0.7577566
acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392    1.0000000
0.2903161  0.2127458  0.3468215
year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199    0.2903161
1.0000000  0.1815277  0.4299042
origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054    0.2127458
0.1815277  1.0000000  0.5136984
mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215
0.4299042  0.5136984  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:

Hide

```
lm.fit1 = lm(mpg~.-name, data=Auto)
summary(lm.fit1)
```

```
Call:
lm(formula = mpg ~ . - name, data = Auto)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.8658 -1.7465 -0.0228  1.3575 13.0212 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -1.358e+01  4.002e+00  -3.394  0.00076 ***
cylinders     1.820e-01  2.836e-01   0.642  0.52140    
displacement  1.796e-02  6.458e-03   2.780  0.00570 ** 
horsepower   -2.912e-02  1.189e-02  -2.449  0.01478 *  
weight       -4.833e-03  5.774e-04  -8.371 1.09e-15 ***
acceleration  6.741e-02  8.492e-02   0.794  0.42780    
year          5.823e-01  4.609e-02  12.635  < 2e-16 ***
origin        1.159e+00  2.400e-01   4.827 2.00e-06 ***
mpg01         5.711e+00  4.874e-01  11.718  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.859 on 383 degrees of freedom
Multiple R-squared:  0.8686,    Adjusted R-squared:  0.8658 
F-statistic: 316.4 on 8 and 383 DF,  p-value: < 2.2e-16
```

I. Is there a relationship between the predictors and the response? There is a relationship between the predictors and the response by testing the null hypothesis of whether all the regression coefficients are zero. F-statistic (the one that evaluates the complete model) p-value is very small, indicating evidence against the null hypothesis.

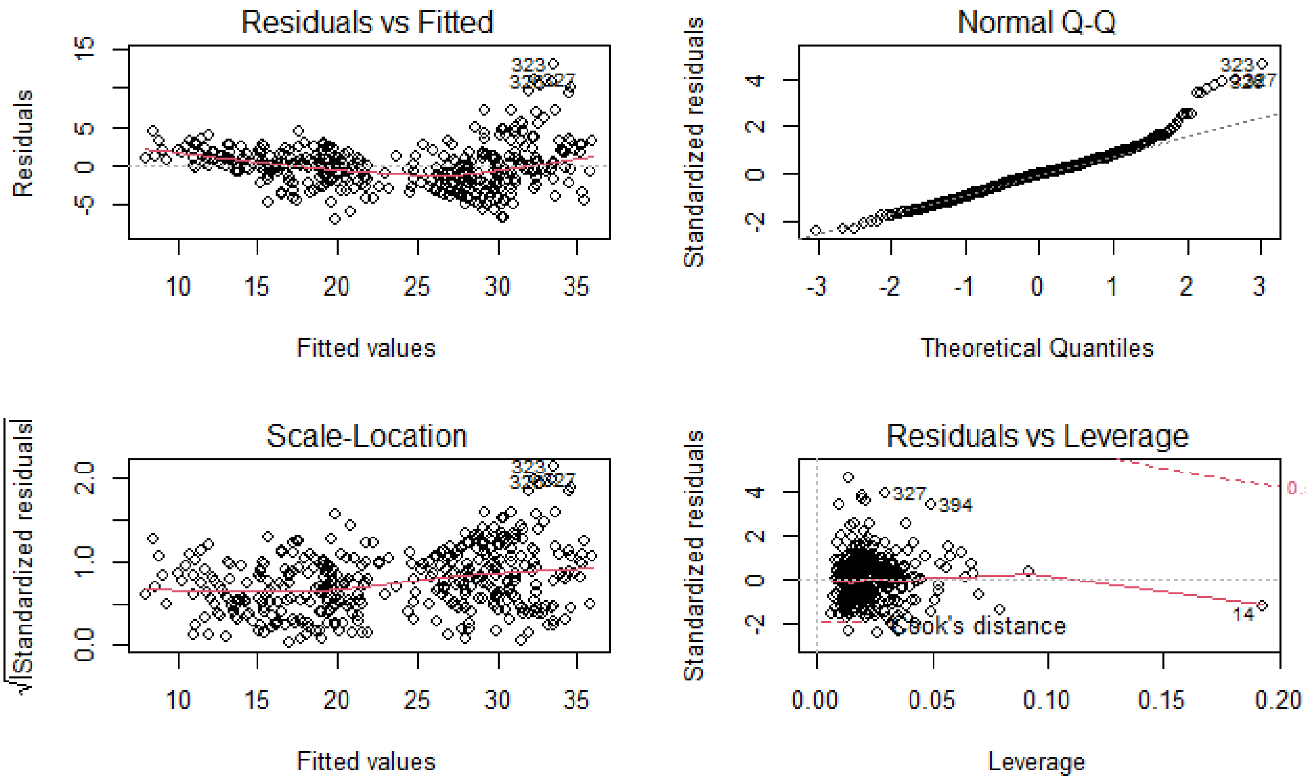
II. Judging by each t-statistic (the one that test each predictor), we see that displacement, weight, year and origin have a statistically significant relationship, while cylinders, horsepower, and acceleration do not.

III. Year coefficient suggest that for every one year, mps increases by the coefficient. Cars become more efficient every year by almost 1 mpg/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?

Hide

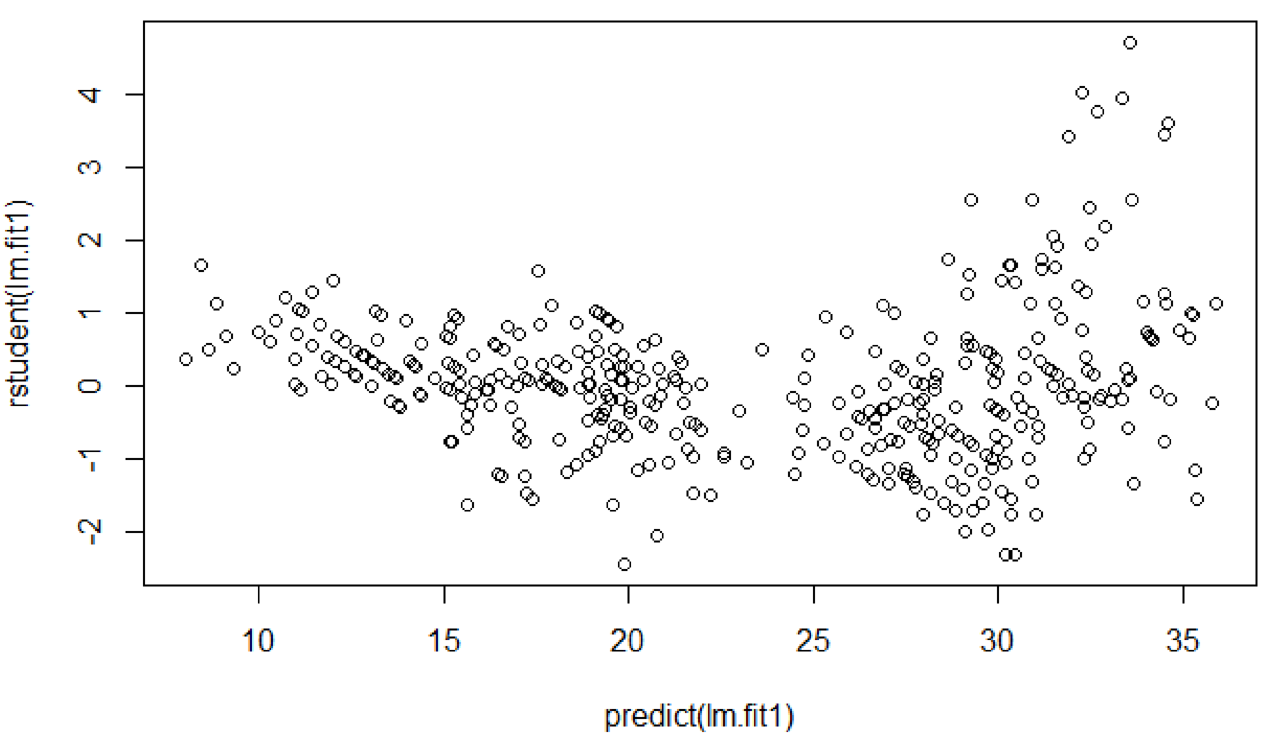
```
par(mfrow=c(2,2))
plot(lm.fit1)
```



The fit does not appear to be accurate because there is a discernible curve pattern to the residuals plots. From the leverage plot, point 14 appears to have high leverage, although not a high magnitude residual.

Hide

```
plot(predict(lm.fit1), rstudent(lm.fit1))
```



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

Hide

```
lm.fit2 = lm(mpg~cylinders*displacement+displacement*weight)
summary(lm.fit2)
```

```

Call:
lm(formula = mpg ~ cylinders * displacement + displacement * 
    weight)

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
```

Cylinders is the most correlated to displacement, followed by displacement to weight. From 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^2 $. Comment on your findings.

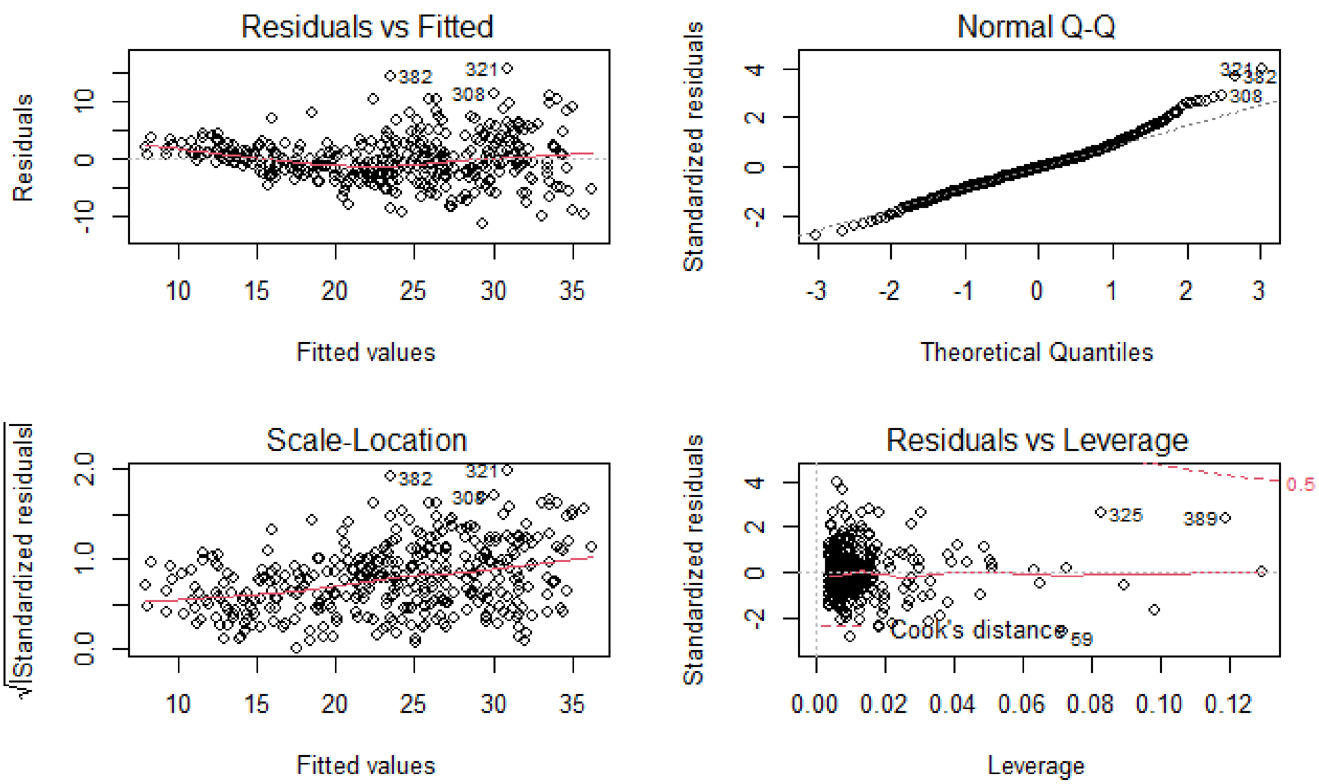
Hide

```
lm.fit3 = lm(mpg~log(weight)+sqrt(horsepower)+acceleration+I(acceleration^2))
summary(lm.fit3)
```

```
Call:
lm(formula = mpg ~ log(weight) + sqrt(horsepower) + acceleration +
    I(acceleration^2))

Residuals:
     Min       1Q   Median       3Q      Max
-11.2932  -2.5082  -0.2237   2.0237  15.7650

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)       178.30303   10.80451  16.503  < 2e-16 ***
log(weight)       -14.74259    1.73994  -8.473 5.06e-16 ***
sqrt(horsepower)   -1.85192    0.36005  -5.144 4.29e-07 ***
acceleration       -2.19890    0.63903  -3.441 0.000643 ***
I(acceleration^2)   0.06139    0.01857   3.305 0.001037 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.99 on 387 degrees of freedom
Multiple R-squared:  0.7414,	Adjusted R-squared:  0.7387
F-statistic: 277.3 on 4 and 387 DF,  p-value: < 2.2e-16
```

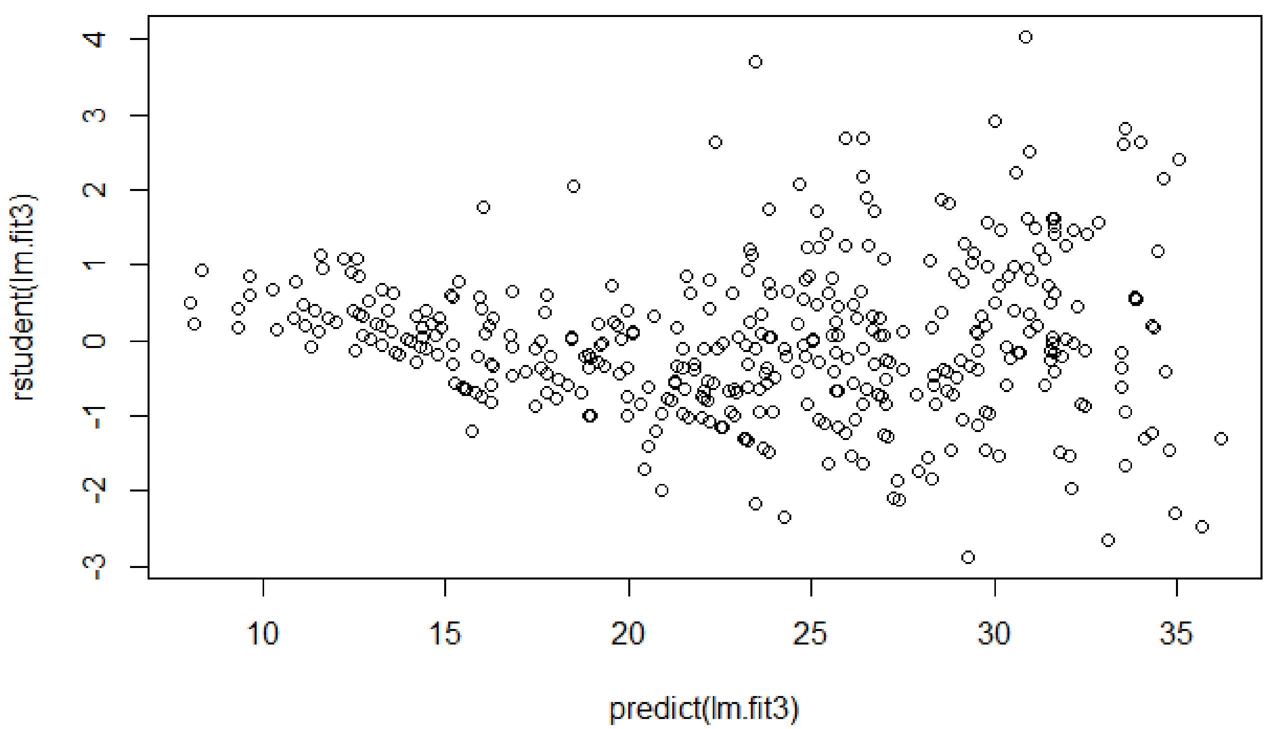
Hide

```
par(mfrow=c(2,2))
plot(lm.fit3)
```



Hide

```
plot(predict(lm.fit3), rstudent(lm.fit3))
```



From the p-values, the log(weight), sqrt(horsepower), and acceleration^2 all have statistical significance of some sort. The residuals plot has less of a discernible pattern than the plot of all linear regression terms. The studentized residuals displays potential outliers (>3). The leverage plot indicates more than three points with high leverage.

However, 2 problems are observed from the above plots: 1) the residuals vs fitted plot indicates heteroskedasticity (unconstant variance over mean) in the model. 2) The Q-Q plot indicates somewhat unnormality of the residuals.

Chapter 3, exercise 10

This question should be answered using the Carseats data set.

a. Fit a multiple regression model to predict Sales using Price, Urban, and US.

Hide

```
summary(Carseats)
```

```
     Sales          CompPrice       Income        Advertising       Population      
Price        ShelveLoc         Age       
 Min.   : 0.000   Min.   : 77   Min.   : 21.00   Min.   : 0.000   Min.   : 10.0   Mi
n.   : 24.0   Bad   : 96   Min.   :25.00  
 1st Qu.: 5.390   1st Qu.:115   1st Qu.: 42.75   1st Qu.: 0.000   1st Qu.:139.0   1s
t Qu.:100.0   Good  : 85   1st Qu.:39.75  
 Median : 7.490   Median :125   Median : 69.00   Median : 5.000   Median :272.0   Me
dian :117.0   Medium:219   Median :54.50  
 Mean   : 7.496   Mean   :125   Mean   : 68.66   Mean   : 6.635   Mean   :264.8   Me
an   :115.8                Mean   :53.32  
 3rd Qu.: 9.320   3rd Qu.:135   3rd Qu.: 91.00   3rd Qu.:12.000   3rd Qu.:398.5   3r
d Qu.:131.0                3rd Qu.:66.00  
 Max.   :16.270   Max.   :175   Max.   :120.00   Max.   :29.000   Max.   :509.0   Ma
x.   :191.0                Max.   :80.00  
   Education     Urban       US     
 Min.   :10.0   No :118   No :142  
 1st Qu.:12.0   Yes:282   Yes:258  
 Median :14.0                      
 Mean   :13.9                      
 3rd Qu.:16.0                      
 Max.   :18.0                      
```

Hide

```
attach(Carseats)
```

```
The following objects are masked from Carseats (pos = 3):

    Advertising, Age, CompPrice, Education, Income, Population, Price, Sales, Shelve
Loc, Urban, US
```

Hide

```
lm.fit = lm(Sales~Price+Urban+US)
summary(lm.fit)
```

```

Call:
lm(formula = Sales ~ Price + Urban + US)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.9206 -1.6220 -0.0564  1.5786  7.0581 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
Price       -0.054459   0.005242 -10.389  < 2e-16 ***
UrbanYes    -0.021916   0.271650  -0.081    0.936    
USYes        1.200573   0.259042   4.635 4.86e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.472 on 396 degrees of freedom
Multiple R-squared:  0.2393,	Adjusted R-squared:  0.2335 
F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

b. Provide an interpretation of each coefficient in the model. Becareful—some of the variables in the model are qualitative!

R fit automatically changed Urban and US to 1 as a dummy variable.

There is a relationship between price and sales given the low p-value of the t-statistic. The relationship is negative.

Urban coefficient is negative however, the p-value is above the recommend alpha.

uSYes The linear regression suggests there is a relationship between whether the store is in the US or not and the amount of sales. The coefficient states a positive relationship between USYes and Sales: if the store is in the US, the sales will increase by approximately 1201 units.

c. Write out the model in equation form, being careful to handle the qualitative variables properly.

Sales = 13.04 -0.05 Price -0.02 UrbanYes + 1.20 USYes

d. For which of the predictors can you reject the null hypothesis H0 : βj = 0?

Price and USYes, based on the p-values, F-statistic, and p-value of the F-statistic.

e. On the basis of your response to the previous question, fit a smaller model that only uses the predictors for which there is evidence of association with the outcome.

Hide

```
lm.fit2 = lm(Sales ~ Price + US)
summary(lm.fit2)
```

```
Call:
lm(formula = Sales ~ Price + US)

Residuals:
    Min      1Q  Median      3Q     Max 
-6.9269 -1.6286 -0.0574  1.5766  7.0515 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 13.03079    0.63098  20.652  < 2e-16 ***
Price       -0.05448    0.00523 -10.416  < 2e-16 ***
USYes        1.19964    0.25846   4.641 4.71e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.469 on 397 degrees of freedom
Multiple R-squared:  0.2393,	Adjusted R-squared:  0.2354 
F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16
```

f. How well do the models in (a) and (e) fit the data?

R^2 of the liner regression suggest that the model from (e) is slightly better.

g. Using the model from (e), obtain 95 % confidence intervals for the coefficient(s).

Hide

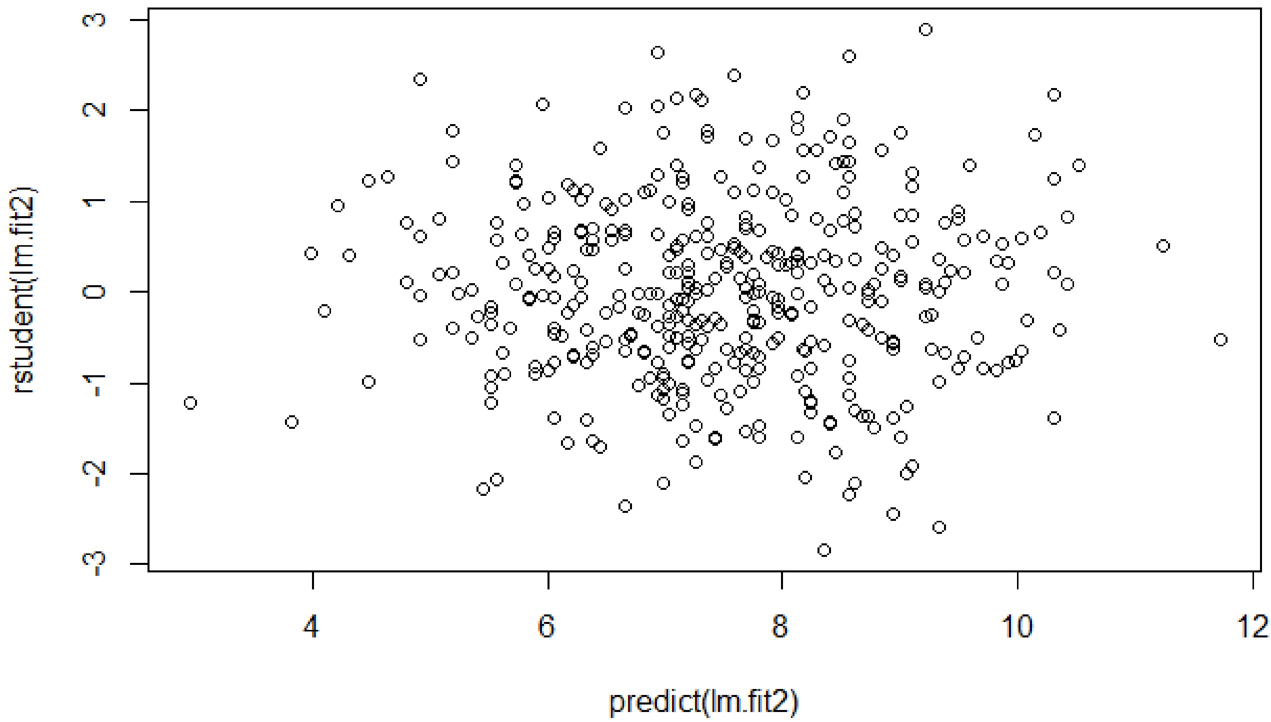
```
confint(lm.fit2)
```

```
                  2.5 %      97.5 %
(Intercept) 11.79032020 14.27126531
Price       -0.06475984 -0.04419543
USYes        0.69151957  1.70776632
```

h. Is there evidence of outliers or high leverage observations in the model from (e)?

Hide

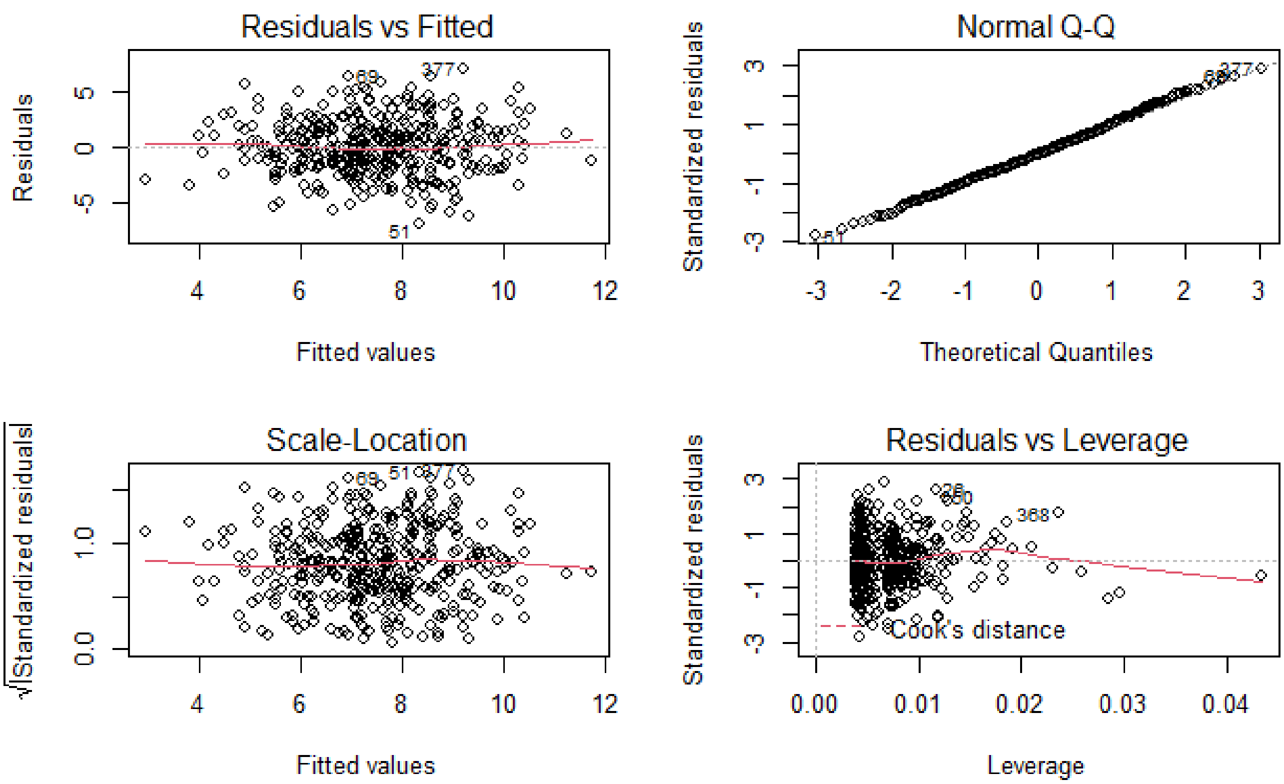
```
plot(predict(lm.fit2), rstudent(lm.fit2))
```



Student residuals look to be bounded by -3 to 3. Not potential outliers.

Hide

```
par(mfrow=c(2,2))
plot(lm.fit2)
```



The are few observations that greatly exceed $ (p + 1) / n $ on the leverage-statistic plot that suggest the corresponding points have high leverage.

Chapter 4, exercise 10

This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter's lab, except that it contains 1,089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

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

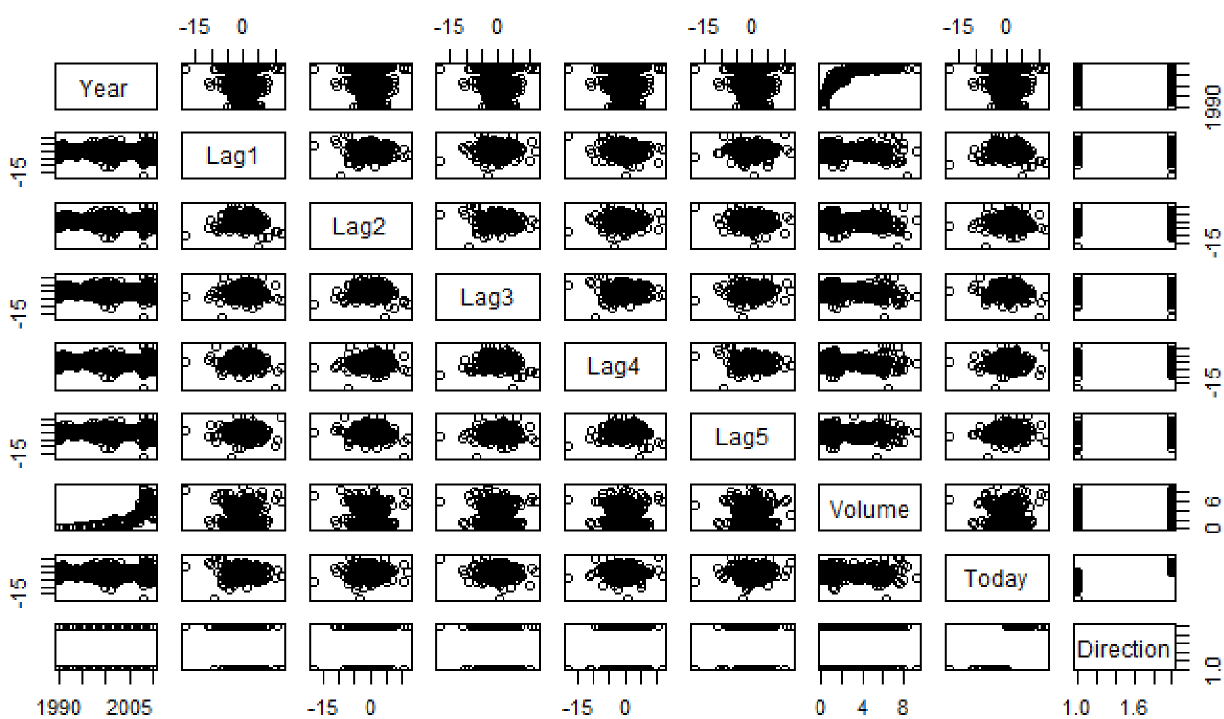
Hide

```
summary(Weekly)
```

```
      Year           Lag1               Lag2               Lag3               Lag4
Lag5              Volume
 Min.   :1990   Min.   :-18.1950   Min.   :-18.1950   Min.   :-18.1950   Min.   :-1
8.1950   Min.   :-18.1950   Min.   :0.08747
 1st Qu.:1995   1st Qu.: -1.1540   1st Qu.: -1.1540   1st Qu.: -1.1580   1st Qu.: -
1.1580   1st Qu.: -1.1660   1st Qu.:0.33202
 Median :2000   Median :  0.2410   Median :  0.2410   Median :  0.2410   Median :
0.2380   Median :  0.2340   Median :1.00268
 Mean   :2000   Mean   :  0.1506   Mean   :  0.1511   Mean   :  0.1472   Mean   :
0.1458   Mean   :  0.1399   Mean   :1.57462
 3rd Qu.:2005   3rd Qu.:  1.4050   3rd Qu.:  1.4090   3rd Qu.:  1.4090   3rd Qu.:
1.4090   3rd Qu.:  1.4050   3rd Qu.:2.05373
 Max.   :2010   Max.   : 12.0260   Max.   : 12.0260   Max.   : 12.0260   Max.   : 1
2.0260   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
```

Hide

```
pairs(Weekly)
```



Hide

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

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

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?

Hide

```
attach(Weekly)
```

```
The following objects are masked from Weekly (pos = 11):

    Direction, Lag1, Lag2, Lag3, Lag4, Lag5, Today, Volume, Year
```

Hide

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

```
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
```

Lag number 2 seems to have some statistical significance with Pr(>z) = 3%

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.

Hide

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

```
        Direction
glm.pred Down  Up
    Down   54  48
    Up    430 557
```

Percentage of correct predictions: (54+557)/(54+557+48+430) = 56.1%. Weeks when the market goes up, the logistic regression is right most of the time, 557/(557+48) = 92.1%. Weeks when the market goes down, the logistic regression is wrong most of the time 54/(430+54) = 11.2%.

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).

Hide

```
train = (Year < 2009)
Weekly.0910 = Weekly[!train, ]
glm.fit = glm(Direction ~ Lag2, data = Weekly, family = binomial, subset = train)
glm.probs = predict(glm.fit, Weekly.0910, type = "response")
glm.pred = rep("Down", length(glm.probs))
glm.pred[glm.probs > 0.5] = "Up"
Direction.0910 = Direction[!train]
table(glm.pred, Direction.0910)
```

```
        Direction.0910
glm.pred Down Up
    Down    9  5
    Up     34 56
```

Hide

```
mean(glm.pred == Direction.0910)
```

```
[1] 0.625
```

The correct predictions percentage 0.625 = (9+56)/(9+5+34+56). Pretty good actually.

e. Repeat (d) using LDA.

Hide

```
library(MASS)
lda.fit = lda(Direction ~ Lag2, data = Weekly, subset = train)
lda.pred = predict(lda.fit, Weekly.0910)
table(lda.pred$class, Direction.0910)
```

```
      Direction.0910
       Down Up
  Down    9  5
  Up     34 56
```

Hide

```
mean(lda.pred$class == Direction.0910)
```

```
[1] 0.625
```

Using LDA returns the same results percentage.

f. Repeat (d) using QDA.

Hide

```
qda.fit = qda(Direction ~ Lag2, data = Weekly, subset = train)
qda.class = predict(qda.fit, Weekly.0910)$class
table(qda.class, Direction.0910)
```

```
         Direction.0910
qda.class Down Up
     Down    0  0
     Up     43 61
```

Hide

```
mean(qda.class == Direction.0910)
```

```
[1] 0.5865385
```

58% of accuracy even though the market only went up.

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

Hide

```
library(class)
train.X = as.matrix(Lag2[train])
test.X = as.matrix(Lag2[!train])
train.Direction = Direction[train]
set.seed(1)
knn.pred = knn(train.X, test.X, train.Direction, k = 1)
table(knn.pred, Direction.0910)
```

```
        Direction.0910
knn.pred Down Up
    Down   21 30
    Up     22 31
```

Hide

```
mean(knn.pred == Direction.0910)
```

```
[1] 0.5
```

Using KNN gives half of times correct results.

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

Logistic regression and LDA methods provide similar test error rates.

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.

First consider using lag two interaction with lag one using logistic regression:

Hide

```
# Logistic regression with Lag2:Lag1
glm.fit = glm(Direction ~ Lag2:Lag1, data = Weekly, family = binomial, subset = trai
         n)
glm.probs = predict(glm.fit, Weekly.0910, type = "response")
glm.pred = rep("Down", length(glm.probs))
glm.pred[glm.probs > 0.5] = "Up"
Direction.0910 = Direction[!train]
table(glm.pred, Direction.0910)
```

```
        Direction.0910
glm.pred Down Up
    Down    1  1
    Up     42 60
```

Hide

```
mean(glm.pred == Direction.0910)
```

```
[1] 0.5865385
```

Now apply the same to LDA method:

Hide

```
# LDA with Lag2 interaction with Lag1
lda.fit = lda(Direction ~ Lag2:Lag1, data = Weekly, subset = train)
lda.pred = predict(lda.fit, Weekly.0910)
mean(lda.pred$class == Direction.0910)
```

```
[1] 0.5769231
```

Chapter 4, exercise 11

In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set.

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.

Hide

```
summary(Auto)
```

```
      mpg          cylinders      displacement     horsepower        weight      acc
eleration        year           origin     
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.
: 8.00   Min.   :70.00   Min.   :1.000  
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st Qu.:2225   1st 
Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000  
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median :2804   Medi
an :15.50   Median :76.00   Median :1.000  
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   :2978   Mean
:15.54   Mean   :75.98   Mean   :1.577  
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd Qu.:3615   3rd 
Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000  
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.
:24.80   Max.   :82.00   Max.   :3.000  
                                                                                     
                 name         mpg01    
 amc matador       :  5   Min.   :0.0  
 ford pinto        :  5   1st Qu.:0.0  
 toyota corolla    :  5   Median :0.5  
 amc gremlin       :  4   Mean   :0.5  
 amc hornet        :  4   3rd Qu.:1.0  
 chevrolet chevette:  4   Max.   :1.0  
 (Other)           :365                
```

Hide

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

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.

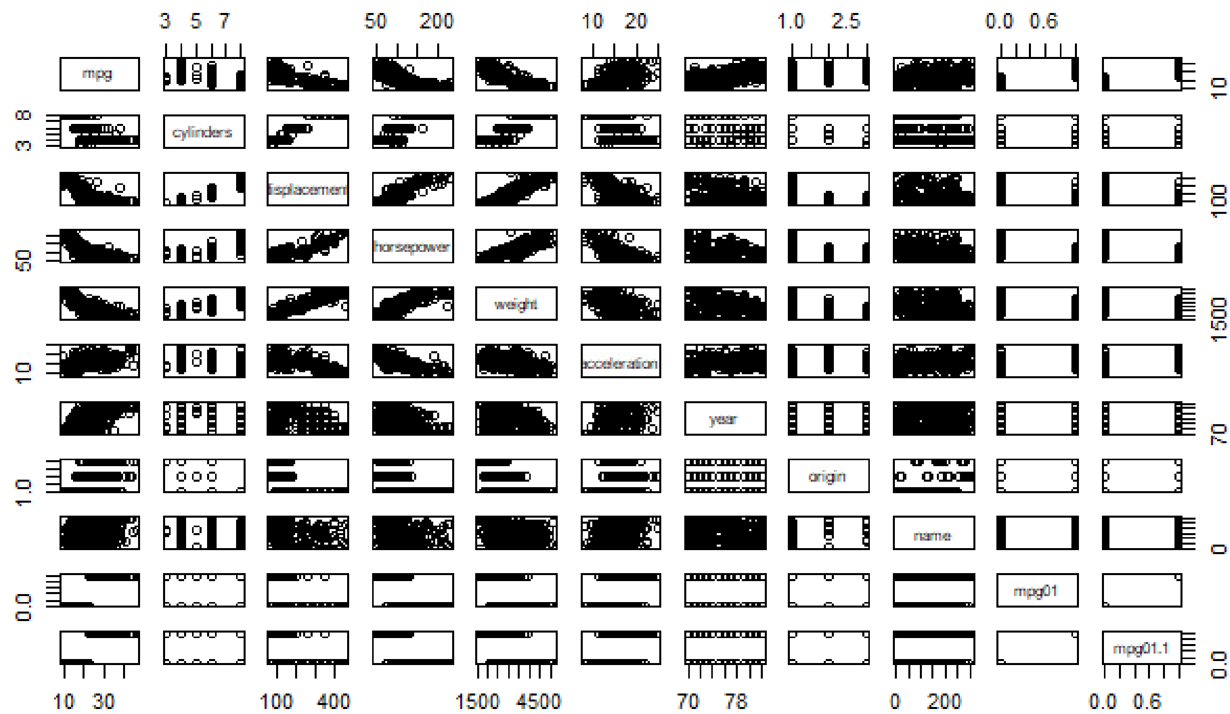
Hide

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

```
                    mpg  cylinders displacement horsepower     weight acceleration
year     origin      mpg01    mpg01.1
mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442    0.4233285
0.5805410  0.5652088  0.8369392  0.8369392
cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273   -0.5046834 -
0.3456474 -0.5689316 -0.7591939 -0.7591939
displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944   -0.5438005 -
0.3698552 -0.6145351 -0.7534766 -0.7534766
horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377   -0.6891955 -
0.4163615 -0.4551715 -0.6670526 -0.6670526
weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000   -0.4168392 -
0.3091199 -0.5850054 -0.7577566 -0.7577566
acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392    1.0000000
0.2903161  0.2127458  0.3468215  0.3468215
year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199    0.2903161
1.0000000  0.1815277  0.4299042  0.4299042
origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054    0.2127458
0.1815277  1.0000000  0.5136984  0.5136984
mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215
0.4299042  0.5136984  1.0000000  1.0000000
mpg01.1       0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566    0.3468215
0.4299042  0.5136984  1.0000000  1.0000000
```

Hide

```
pairs(Auto)
```



Anti-correlated with cylinders, weight, displacement, horsepower.

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

Hide

```
train = sample(c(rep(0, 0.7 * nrow(Auto)), rep(1, 0.3 * nrow(Auto))))
test = !train
Auto.train = Auto[train, ]
Auto.test = Auto[test, ]
mpg01.test = mpg01[test]
```

d. Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

Hide

```
# LDA
library(MASS)
lda.fit = lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto)
lda.pred = predict(lda.fit, Auto.test)
mean(lda.pred$class != mpg01.test)
```

```
[1] 0.1163636
```

9.09% test error rate.

e. Perform QDA on the training data in order to predict `mpg01` using the variables that seemed most associated with `mpg01` in (b). What is the test error of the model obtained?

Hide

```
# QDA
qda.fit = qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto)
qda.pred = predict(qda.fit, Auto.test)
mean(qda.pred$class != mpg01.test)
```

```
[1] 0.1345455
```

9.4% test error rate.

f. Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?

Hide

```
# Logistic regression
glm.fit = glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, f
         amily = binomial)
glm.probs = predict(glm.fit, Auto.test, type = "response")
glm.pred = rep(0, length(glm.probs))
glm.pred[glm.probs > 0.5] = 1
mean(glm.pred != mpg01.test)
```

```
[1] 0.12
```

0.09% test error rate.

g. Perform KNN on the training data, with several values of K, in order to predict mpg01. Use only the variables that seemed most associated with mpg01 in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?

Hide

```
library(class)
train.X = cbind(cylinders, weight, displacement, horsepower)[train, ]
test.X = cbind(cylinders, weight, displacement, horsepower)[test, ]
train.mpg01 = mpg01[train]
set.seed(1)
# KNN(k=1)
knn.pred = knn(train.X, test.X, train.mpg01, k = 1)
mean(knn.pred != mpg01.test)
```

```
[1] 0.48
```

Hide

```
# KNN(k=10)
knn.pred = knn(train.X, test.X, train.mpg01, k = 10)
mean(knn.pred != mpg01.test)
```

```
[1] 0.48
```

Hide

```
# KNN(k=100)
knn.pred = knn(train.X, test.X, train.mpg01, k = 100)
mean(knn.pred != mpg01.test)
```

```
[1] 0.48
```

All KNN test resulted in the same value.