STATS3850 - Assignment #1

Question 1

2.3 Q9. This exercise involves the Auto data set studied in the lab. Make sure that the missing values have been removed from the data.

```
library(ISLR)
dim(Auto)
```

```
## [1] 392   9
```

```
summary(Auto)
```

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

(a) Which of the predictors are quantitative, and which are qualitative? **quantitative**: mpg, cylinders, displacement, horsepower, weight, # acceleration, year **qualitative**: name, origin

(b) What is the range of each quantitative predictor? You can answer this using the range() function.

```
sapply(Auto[, 1:7], range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year
## [1,]  9.0         3           68         46   1613          8.0   70
## [2,] 46.6         8          455        230   5140         24.8   82
```

(c) What is the mean and standard deviation of each quantitative predictor?

```
print("Means")
```

```
## [1] "Means"
```

```r
sapply(Auto[, 1:7], mean)
```

```
##          mpg    cylinders displacement   horsepower       weight 
##    23.445918     5.471939   194.411990   104.469388  2977.584184 
## acceleration         year 
##    15.541327    75.979592
```

```r
print("Standard Deviations")
```

```
## [1] "Standard Deviations"
```

```r
sapply(Auto[, 1:7], sd)
```

```
##          mpg    cylinders displacement   horsepower       weight 
##     7.805007     1.705783   104.644004    38.491160   849.402560 
## acceleration         year 
##     2.758864     3.683737
```

(d) Now remove the 10th through 85th observations. What is the range, mean, and standard deviation of each predictor in the subset of the data that remains?

```r
newAuto = Auto[-(10:85),]

# stats
sapply(newAuto[, 1:7], range)
```

```
##       mpg cylinders displacement horsepower weight acceleration year
## [1,] 11.0         3           68         46   1649          8.5   70
## [2,] 46.6         8          455        230   4997         24.8   82
```

```r
sapply(newAuto[, 1:7], mean)
```

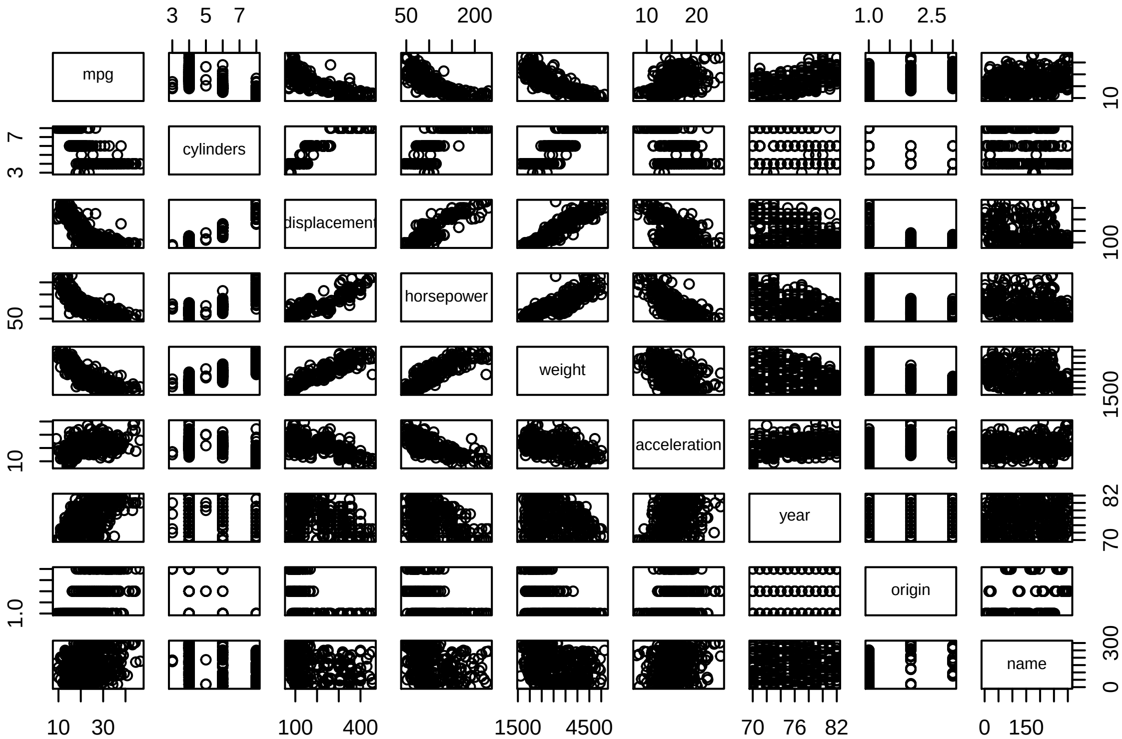
```
##          mpg    cylinders displacement   horsepower       weight 
##    24.404430     5.373418   187.240506   100.721519  2935.971519 
## acceleration         year 
##    15.726899    77.145570
```

```r
sapply(newAuto[, 1:7], sd)
```

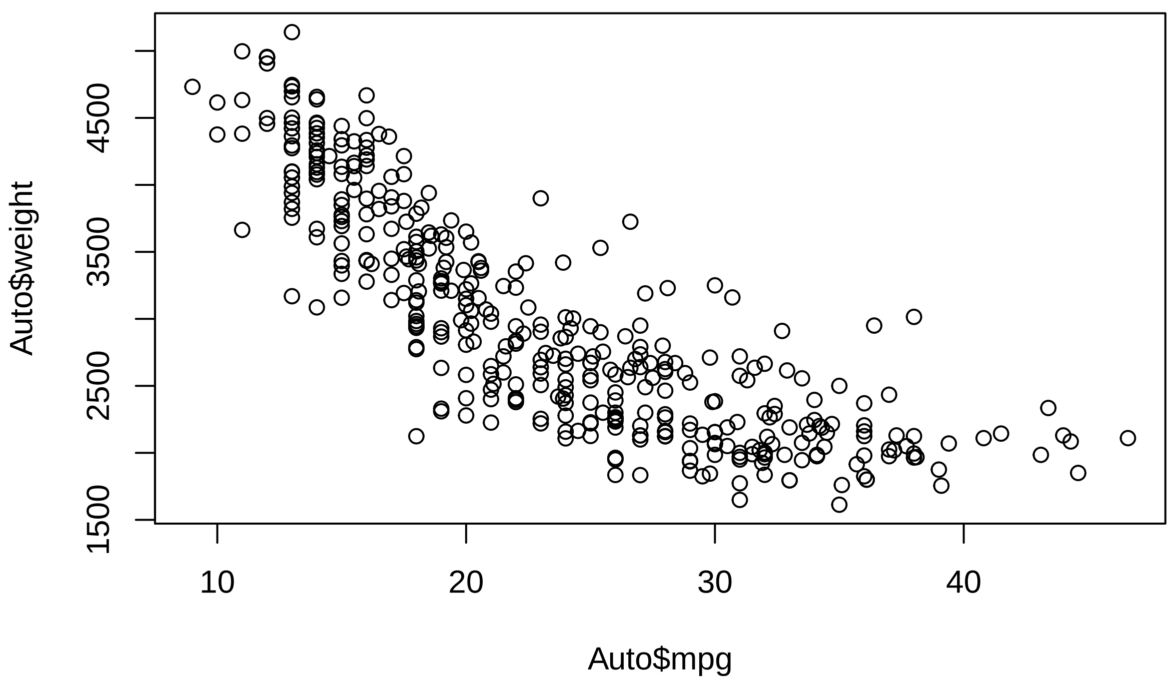
```
##          mpg    cylinders displacement   horsepower       weight 
##     7.867283     1.654179    99.678367    35.708853   811.300208 
## acceleration         year 
##     2.693721     3.106217
```

(e) Using the full data set, investigate the predictors graphically, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.

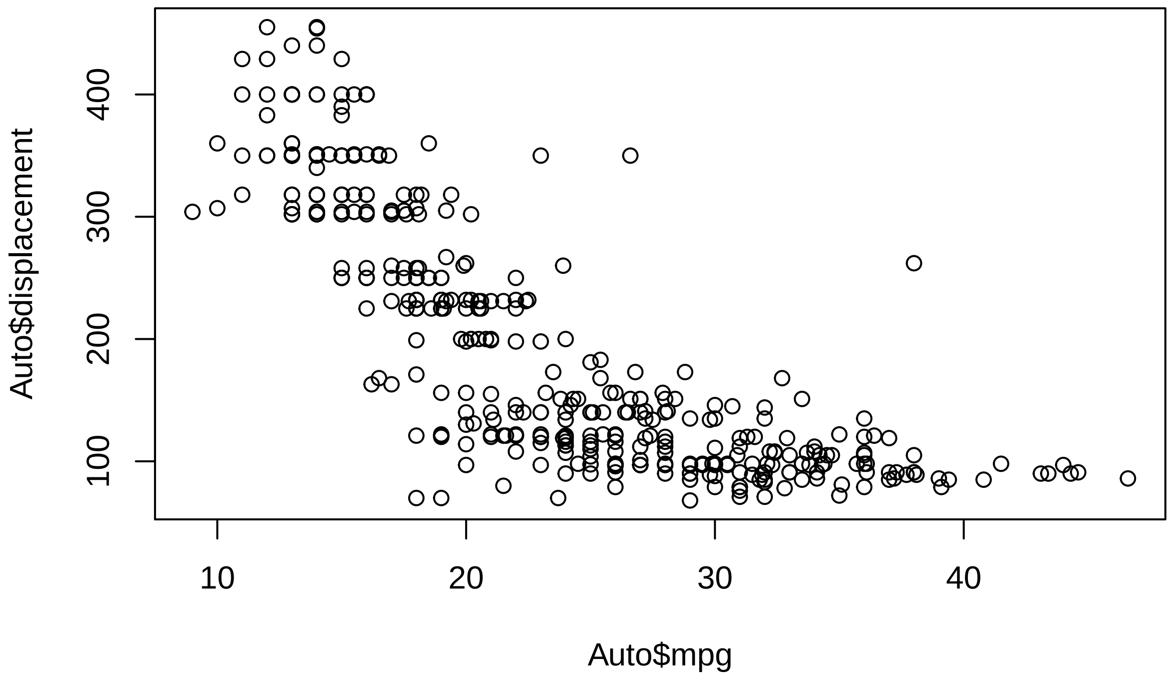
```r
pairs(Auto)
```



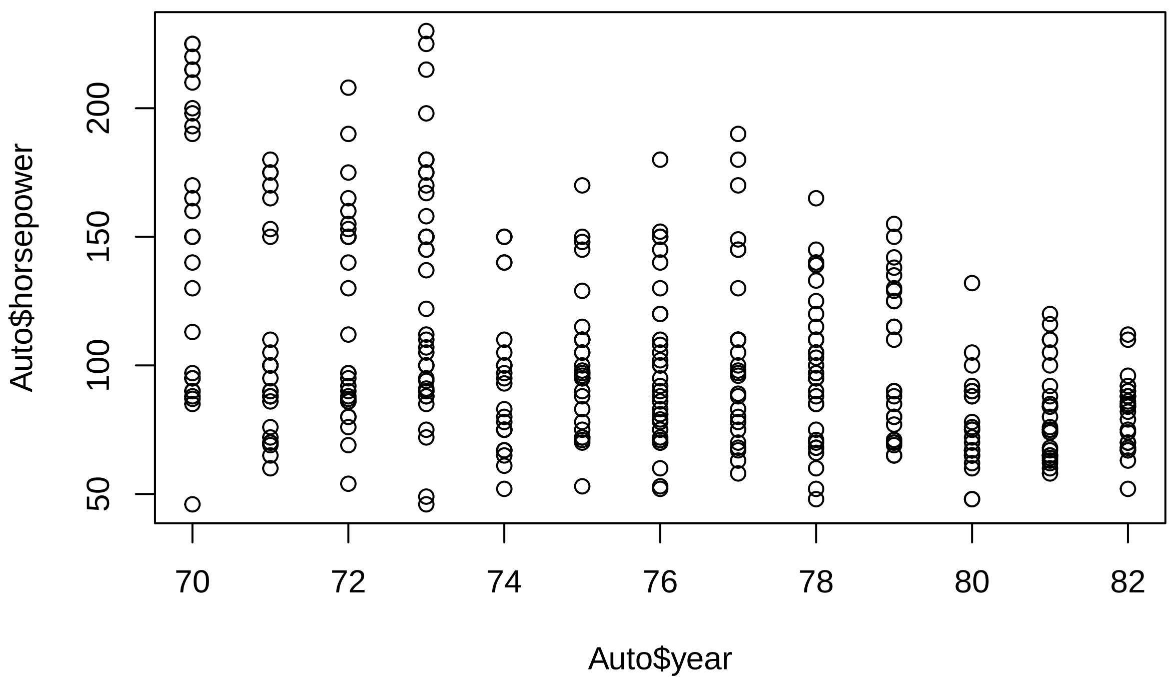

```r
plot(Auto$mpg, Auto$weight)
```




```r
# The heavier the card the lower the mpg.
plot(Auto$mpg, Auto$displacement)
```



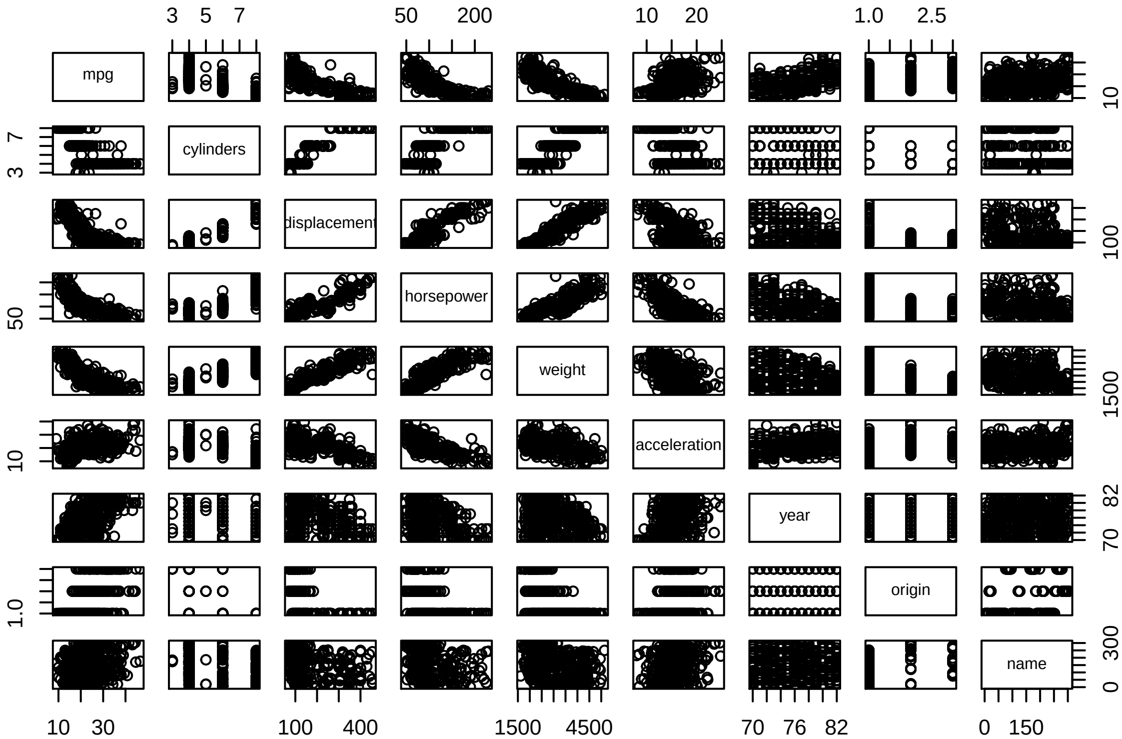

```r
# Higher displacement, less Mpg.
plot(Auto$year, Auto$horsepower)
```




```r
# It apears the newer cars have less horsepower than the older ones.
```

(f) Suppose that we wish to predict gas mileage (mpg) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer.

```r
pairs(Auto)
```




```
# It looks like weight, horsepower and displacement are clear indicators of MPG.
```

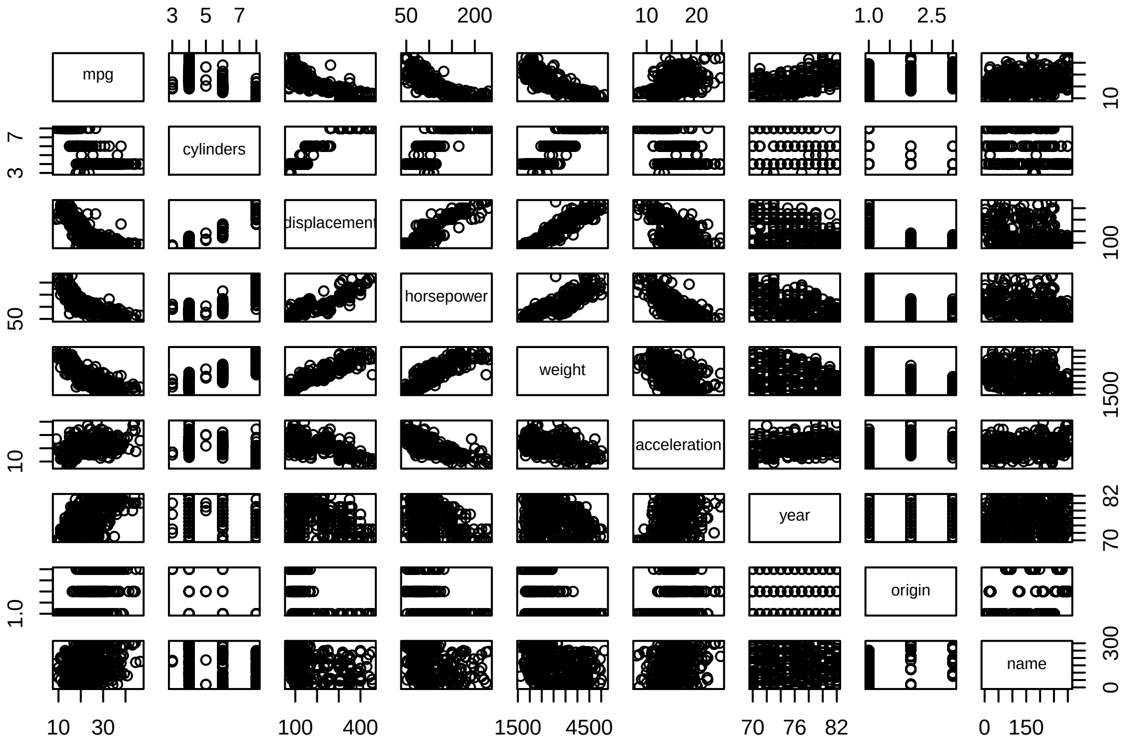
3.6 Q9. This question involves the use of multiple linear regression on the Auto data set.

```
library(ISLR)
summary(Auto)
```

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

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

```r
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.

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

```
##                     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:

i. Is there a relationship between the predictors and the re- sponse? Yes, there is clearly a relationship between these variables and the response. This is evident by the p-values being significant and the a handful of co-efficients not being ~ 0.

ii. Which predictors appear to have a statistically significant relationship to the response? Displacement, weight, year, and origin. Judged by p-values of each predictors t-value.

iii. What does the coefficient for the year variable suggest? It's `year's` co-efficient of 0.7508 seems to suggest that for ever year brings a increase 0.75 mpg increase.

```r
fit_1 = lm(mpg~.-name, data=Auto)
summary(fit_1)
```

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

Answer: i. Is there a relationship between the predictors and the response? * Yes, there is a relatioship between the predictors and the response by testing the null hypothesis of whether all the regression coefficients are zero. The F -statistic is far from 1 (with a small p-value), indicating evidence against the null hypothesis.

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

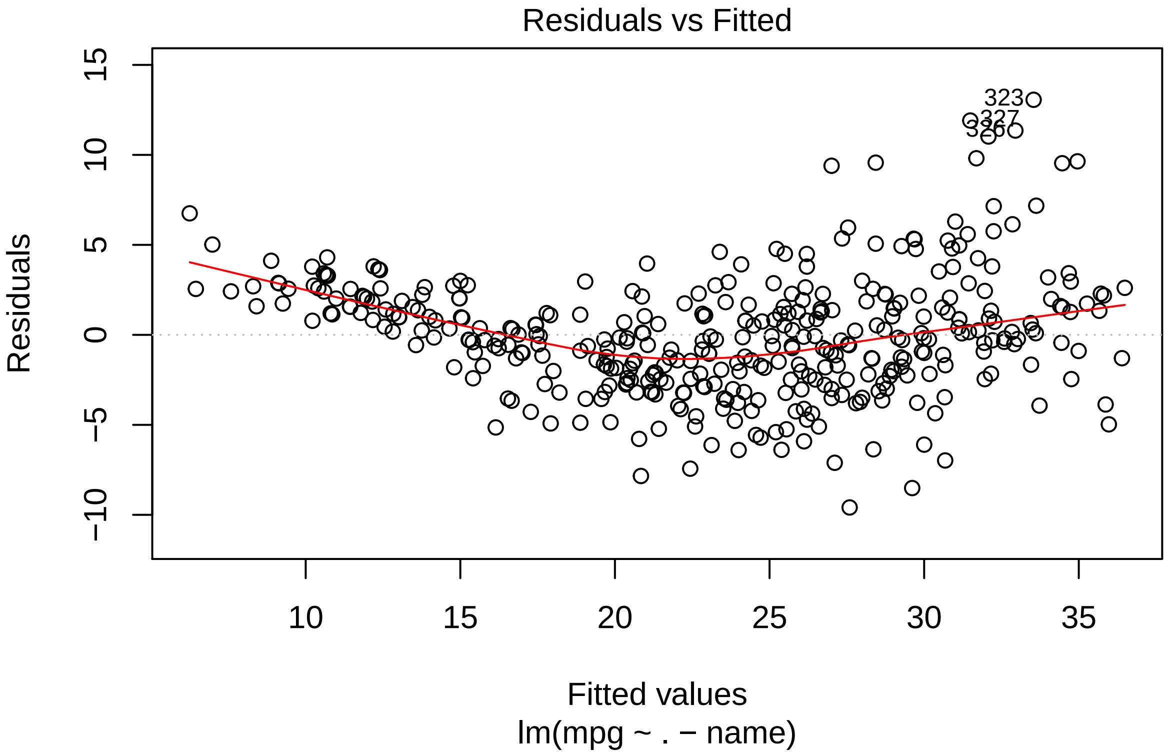
- Looking at the p-values associated with each predictor's t-statistic, we see that displacement, weight, year, and origin have a statistically significant relationship, while cylinders, horsepower, and acceleration do not.

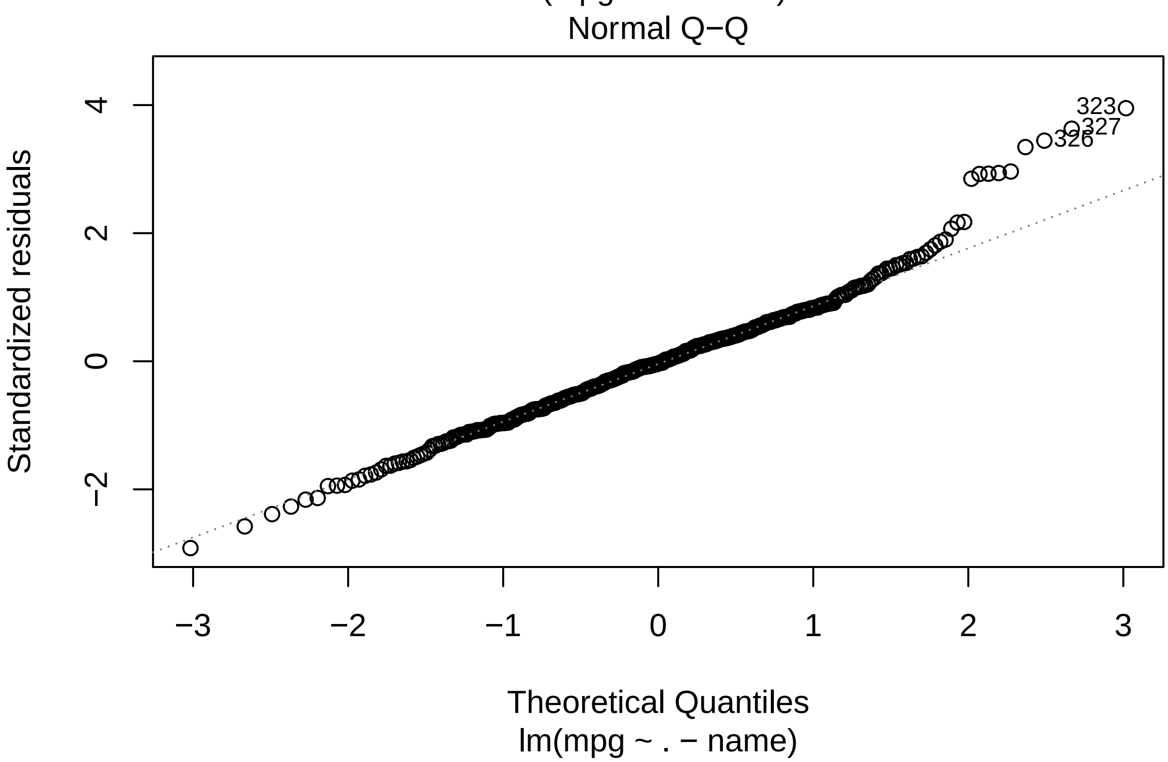
iii. What does the coefficient for the year variable suggest?

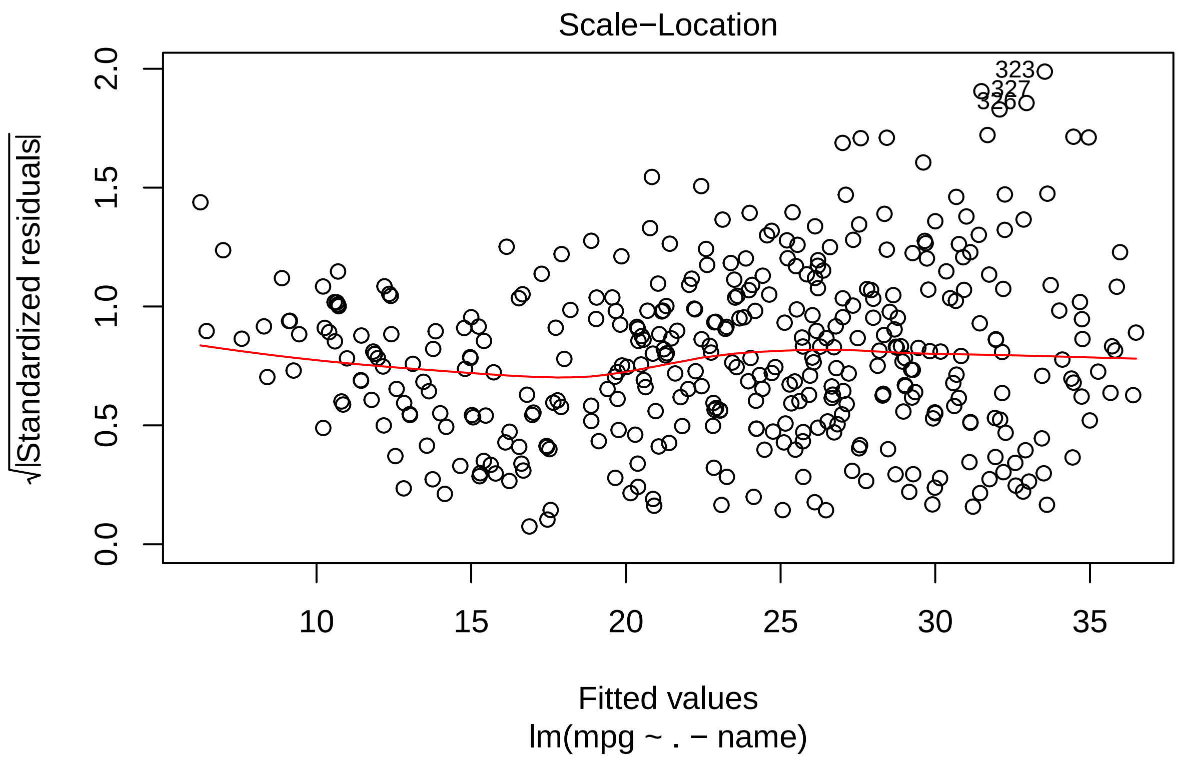
- The regression coefficient for year, 0.7508, suggests that for every one year, mpg increases by the coefficient. In other words, cars become more fuel efficient every year by almost 1 mpg / year.

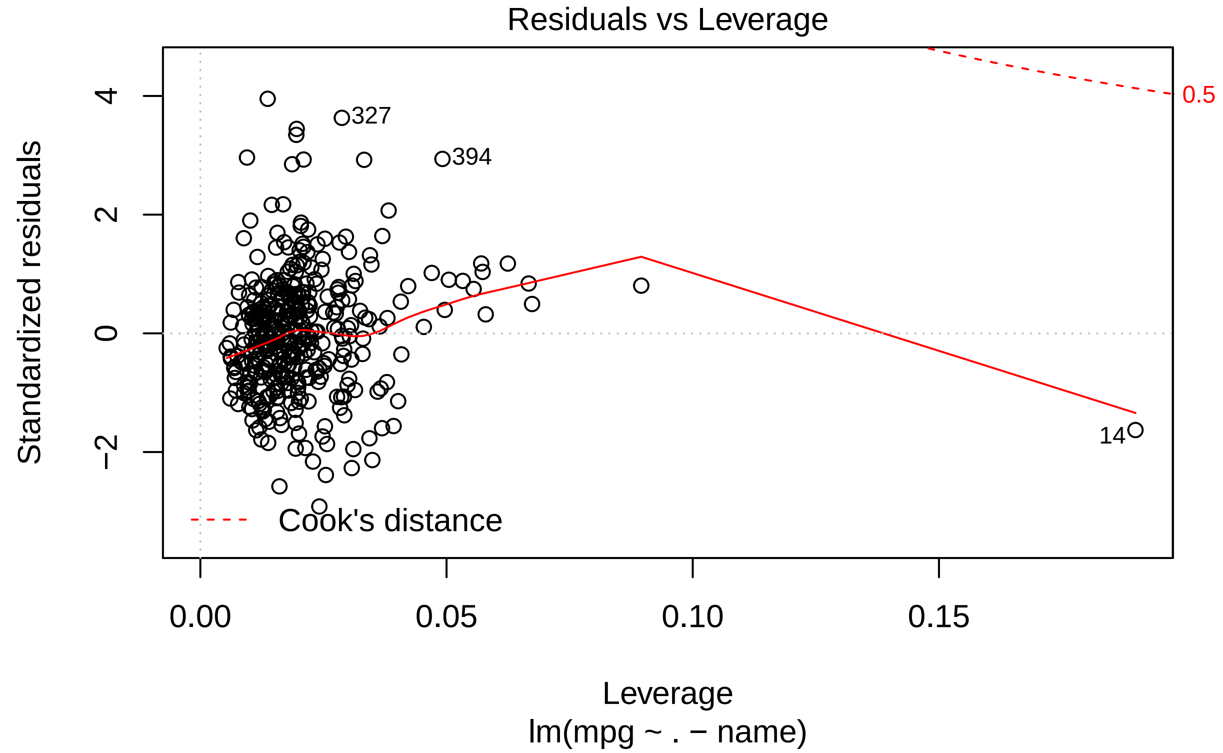
(d) Use the plot() function to produce diagnostic plots of the linear regression fit. Comment on any problems you see with the fit. Do the residual plots suggest any unusually large outliers? Does the leverage plot identify any observations with unusually high leverage?

```r
plot(fit_1)
```


Residuals vs Fitted
Residuals
−10
−5
0
5
10
15
10
15
20
25
30
35
323
327
326
Fitted values
lm(mpg ~ . − name)


Normal Q−Q
Standardized residuals
−2
0
2
4
−3
−2
−1
0
1
2
3
323
327
326
Theoretical Quantiles
lm(mpg ~ . − name)






```
#plot(predict(fit_1), rstudent(fit_1))

# The residual plot shows that the model may not be a very good fit. Since it follows a curve and as op

# Point 14 seems to have unusually high leverage
```

(e) Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant? I used the co-relation matix to find the most co-related variables. They tend to be the variables with interaction efects.

```
fit_2 = lm(mpg~cylinders*displacement+displacement*weight, data=Auto)
summary(fit_2)
```

```
##
## 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
```

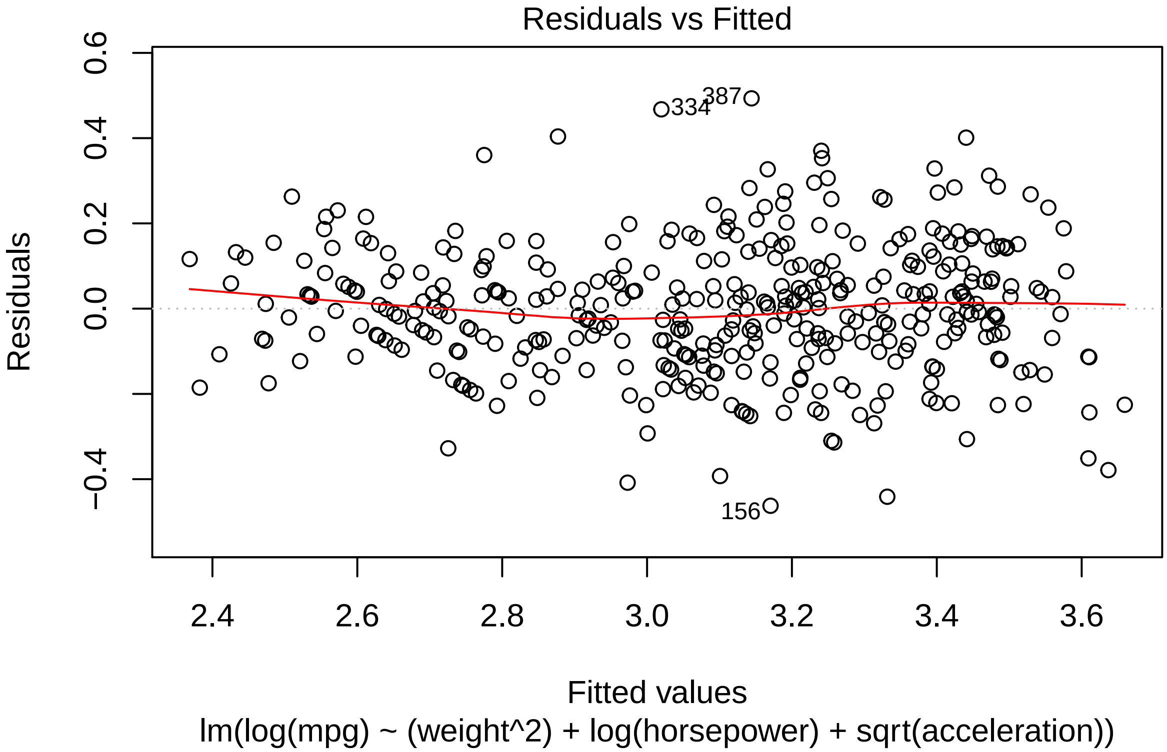
(f) Try a few different transformations of the variables, such as log(X), root(X), X2. Comment on your findings.

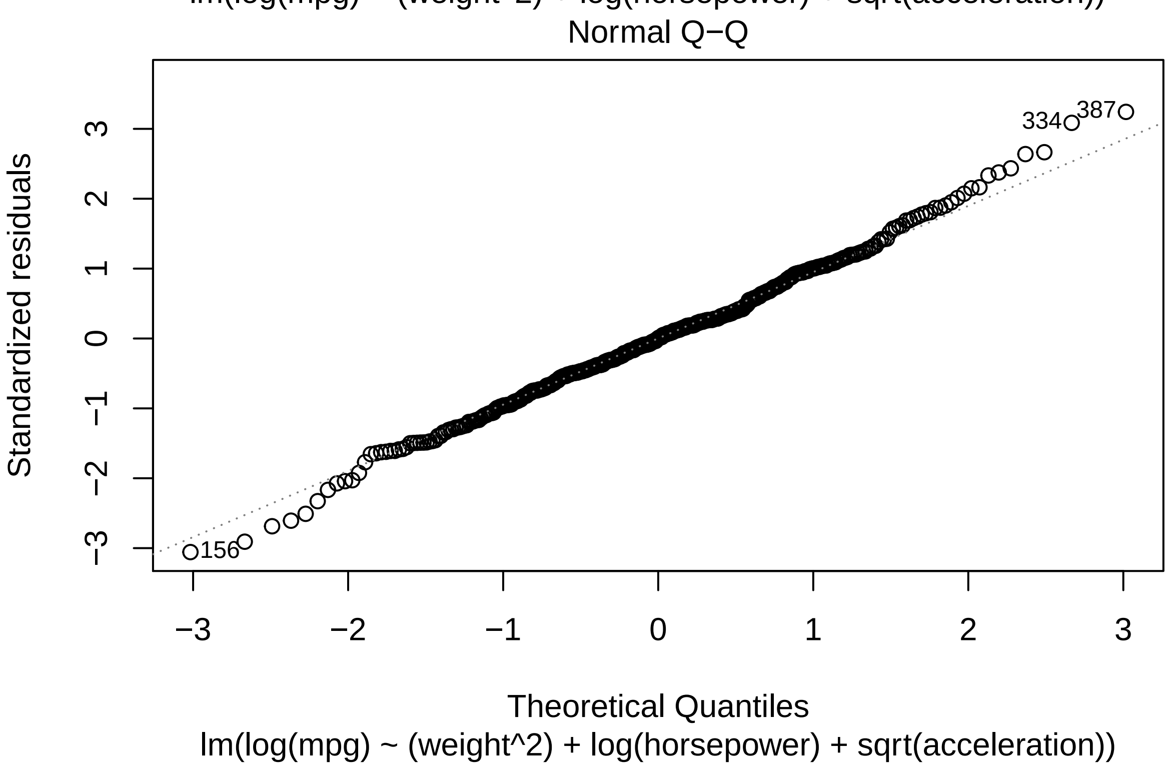
```
fit_3 = lm(log(mpg)~(weight^2)+log(horsepower)+sqrt(acceleration), data=Auto)
summary(fit_3)
```

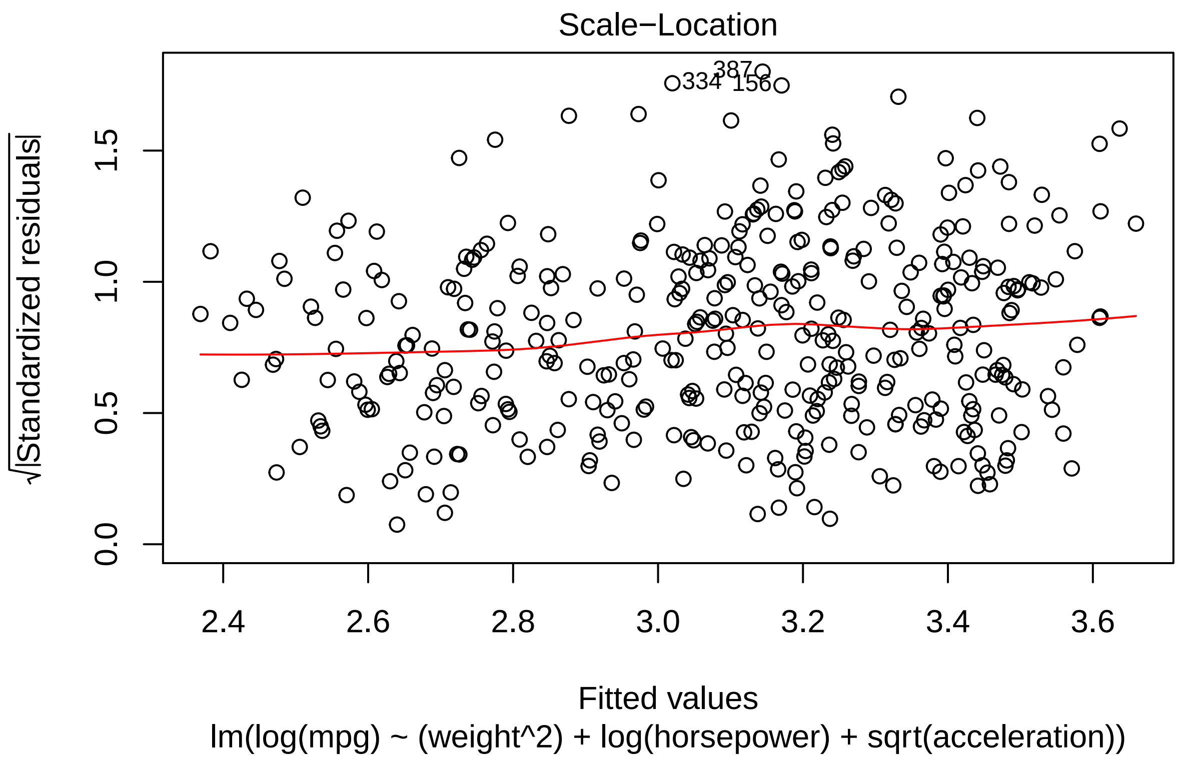
```
##
## Call:
## lm(formula = log(mpg) ~ (weight^2) + log(horsepower) + sqrt(acceleration),
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46240 -0.09677  0.00029  0.09750  0.49334
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         6.422e+00  3.997e-01  16.069  < 2e-16 ***
## weight             -1.889e-04  2.282e-05  -8.281 2.00e-15 ***
## log(horsepower)    -5.100e-01  7.225e-02  -7.058 7.84e-12 ***
## sqrt(acceleration) -1.073e-01  3.772e-02  -2.845  0.00467 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1525 on 388 degrees of freedom
```

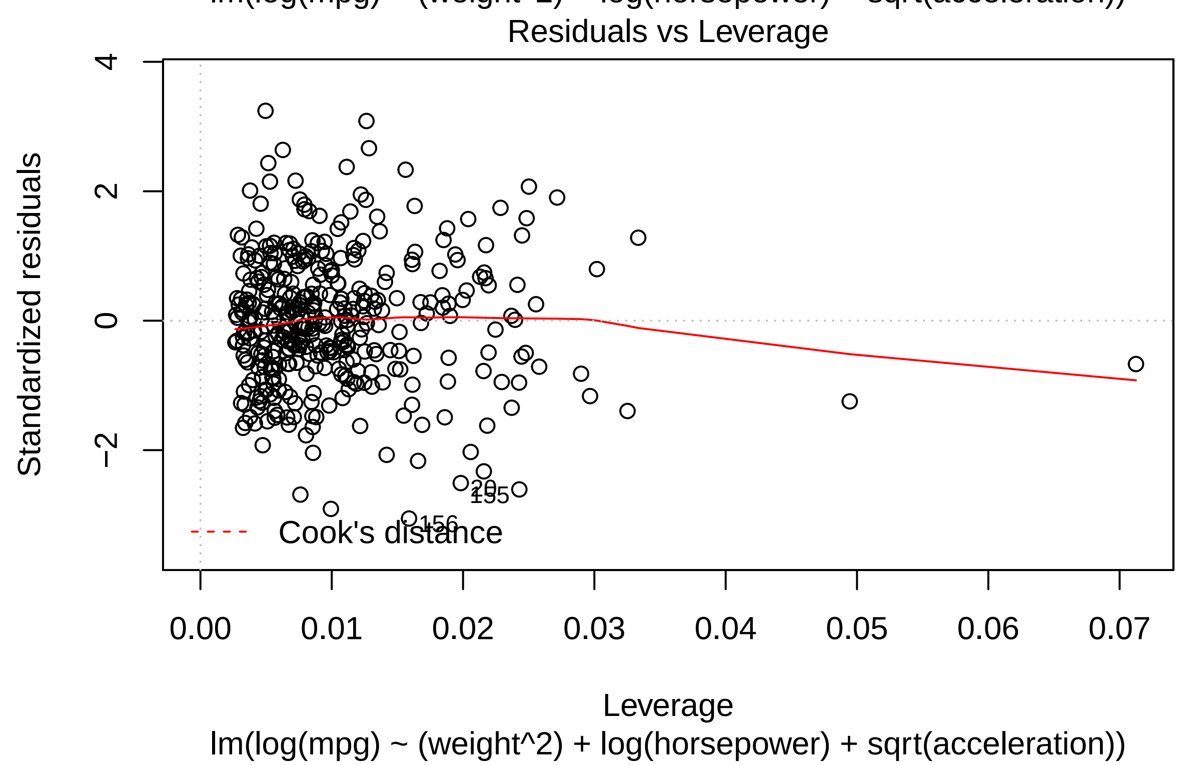
```
## Multiple R-squared:  0.8004, Adjusted R-squared:  0.7988
## F-statistic: 518.6 on 3 and 388 DF,  p-value: < 2.2e-16
```

```
plot(fit_3)
```










```
# This set of transformations actually seems to help the model achieve a nearly straight residual curve
```

Question 2

In this exercise you will create some simulated data and will fit simple linear regression models to it. Make sure to use `set.seed(1)` prior to starting part (a) to ensure consistent results.

```
set.seed(1)
```

(a) Using the `rnorm()` function, create a vector, `x`, containing 100 observations drawn from a N(0,1) distribution. This represents a feature, X.

```
x <- rnorm(100)
```

(b) Using the `rnorm()` function, create a vector, eps, containing 100 observations drawn from a N(0,0.25) distribution i.e. a normal distribution with mean zero and variance 0.25.

```
eps <- rnorm(100, 0, 0.25)
```

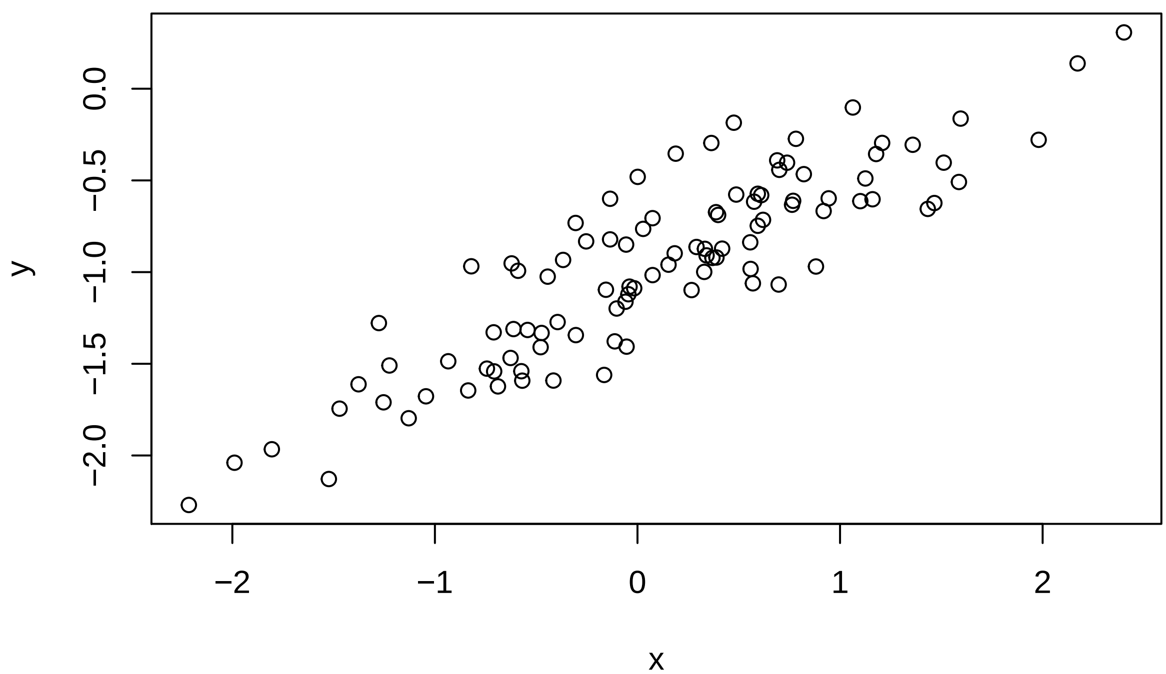
(c) Using `x` and `eps`, generate a vector y according to the model What is the length of the vector y? What are the values of B0 and B1 in this linear model?

Clearly, B0 = -1 and B1 = 0.5

```
y <- (0.5 * x) + eps - 1
```

(d) Create a scatterplot displaying the relationship between x and y. Comment on what you observe.

```
plot(x,y)
```



(e) Fit a least squares linear model to predict y using x. Comment on the model obtained. How do B^0 and B^1 compare to B0 and B1? B^0 = -1.00942 & B^1 =0.49973, The estimates arequite close to the actual coefficients of the population model.

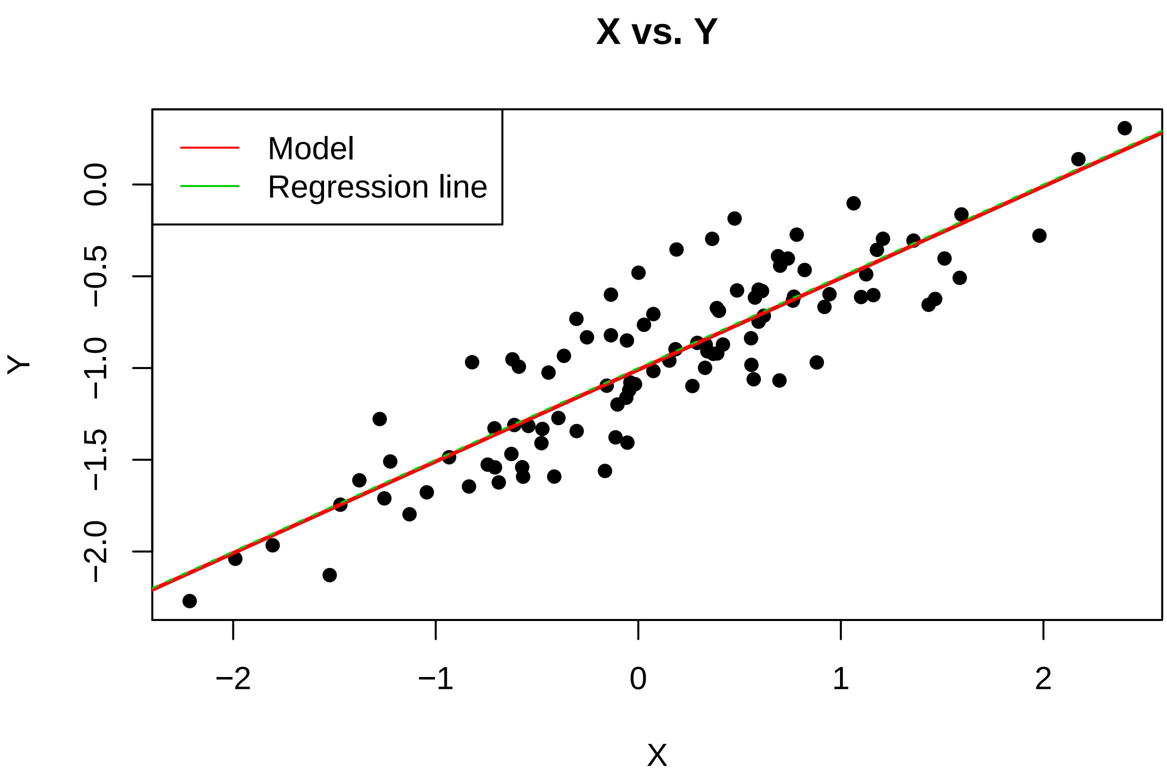
```
fit_3 <- lm(y~x)
summary(fit_3)
```

```
##
## Call:
## lm(formula = y ~ x)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.46921 -0.15344 -0.03487  0.13485  0.58654
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.00942    0.02425  -41.63   <2e-16 ***
## x            0.49973    0.02693   18.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2407 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

(f) Display the least squares line on the scatterplot obtained in (d). Draw the population regression line on the plot, in a different color. Use the legend() command to create an appropriate legend.

```
plot(x,y,pch=16, xlab="X",ylab="Y",main="X vs. Y")
abline(coefficients(fit_3), lwd=2, col=2,lty=1)
abline(a=-1,b=0.5,col=3,lty=2)
legend( x= "topleft",
        legend=c("Model","Regression line"),
        col=c(2, 3), lty=1)
```



(g) Now fit a polynomial regression model that predicts y using x and x2. Is there evidence that the quadratic term improves the model fit? Explain your answer. EDIT: There is evidence that model fit has increased over the training data given the slight increase in R2 and RSE. Although, the p-value of the t-statistic suggests that there isn't a relationship between y and x2.

```r
model_squared_x = lm(y~x+I(x^2))
summary(model_squared_x)
```

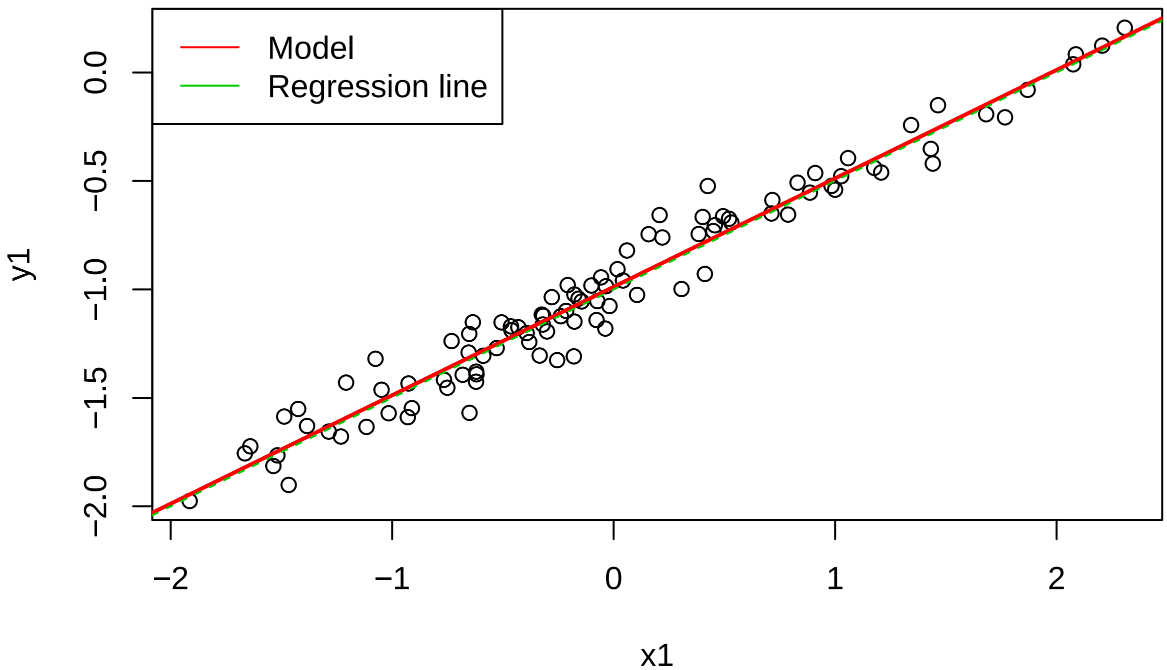
```
##
## Call:
## lm(formula = y ~ x + I(x^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4913 -0.1563 -0.0322  0.1451  0.5675
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.98582    0.02941 -33.516   <2e-16 ***
## x            0.50429    0.02700  18.680   <2e-16 ***
## I(x^2)      -0.02973    0.02119  -1.403    0.164
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2395 on 97 degrees of freedom
## Multiple R-squared:  0.7828, Adjusted R-squared:  0.7784
## F-statistic: 174.8 on 2 and 97 DF,  p-value: < 2.2e-16
```

(h) Repeat (a)–(f) after modifying the data generation process in such a way that there is less noise in the data. The model (3.39) should remain the same. You can do this by decreasing the vari- ance of the normal distribution used to generate the error term in (b). Describe your results.

```r
set.seed(1)
eps_2 = rnorm(100, 0, 0.11)
x1 = rnorm(100)
y1 = -1 + 0.5*x1 + eps_2
plot(x1, y1)
fit_2h = lm(y1~x1)
summary(fit_2h)
```

```
##
## Call:
## lm(formula = y1 ~ x1)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.255657 -0.066397  0.000589  0.064135  0.252248
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.988026   0.009938  -99.42   <2e-16 ***
## x1           0.499897   0.010419   47.98   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0993 on 98 degrees of freedom
## Multiple R-squared:  0.9592, Adjusted R-squared:  0.9587
## F-statistic:  2302 on 1 and 98 DF,  p-value: < 2.2e-16
```

```r
abline(coefficients(fit_2h), lwd=2, col=2,lty=1)
abline(a=-1,b=0.5,col=3,lty=2)
legend( x= "topleft",
        legend=c("Model","Regression line"),
        col=c(2, 3), lty=1)
```



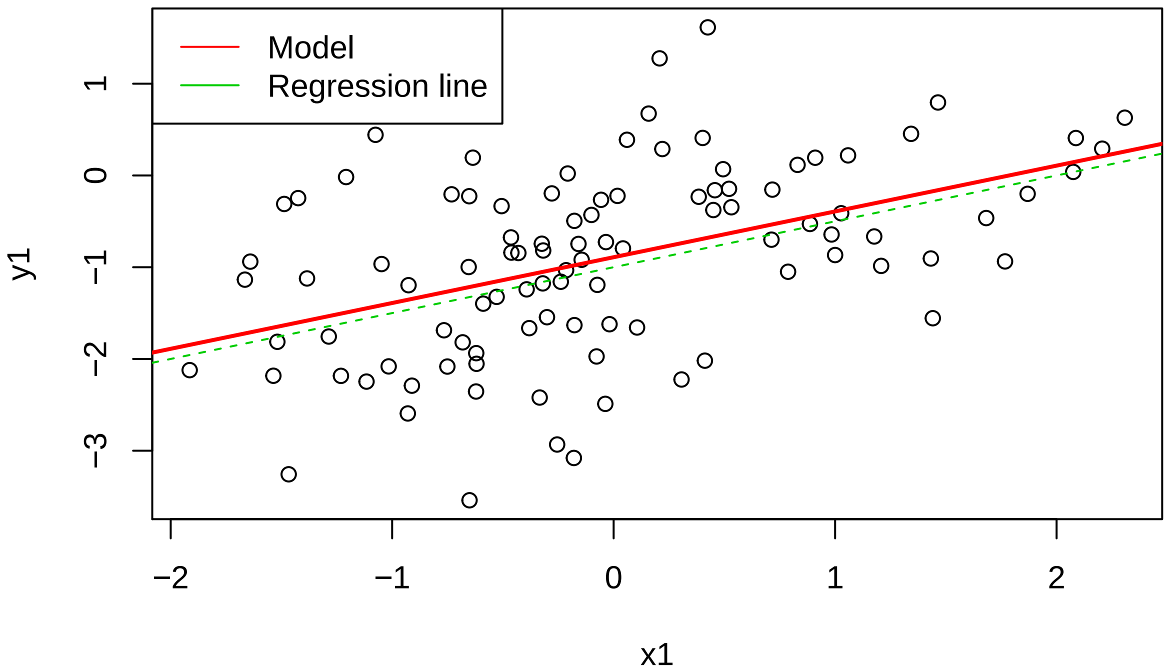
(i) Repeat (a)–(f) after modifying the data generation process in such a way that there is more noise in the data. The model (3.39) should remain the same. You can do this by increasing the variance of the normal distribution used to generate the error term in (b). Describe your results.

```r
set.seed(1)
eps_3 = rnorm(100, 0, 1)
x1 = rnorm(100)
y1 = -1 + 0.5*x1 + eps_3
plot(x1, y1)
fit_2i = lm(y1~x1)
summary(fit_2i)
```

```
##
## Call:
## lm(formula = y1 ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.32416 -0.60361  0.00536  0.58305  2.29316
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.89115    0.09035  -9.864 2.39e-16 ***
## x1           0.49907    0.09472   5.269 8.16e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.9028 on 98 degrees of freedom
## Multiple R-squared:  0.2207, Adjusted R-squared:  0.2128
## F-statistic: 27.76 on 1 and 98 DF,  p-value: 8.158e-07
```

```r
abline(coefficients(fit_2i), lwd=2, col=2,lty=1)
abline(a=-1,b=0.5,col=3,lty=2)
legend( x= "topleft",
        legend=c("Model","Regression line"),
        col=c(2, 3), lty=1)
```



(j) What are the confidence intervals for B0 and B1 based on the original data set, the noisier data set, and the less noisy data set? Comment on your results.

```r
confint(fit_3)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0575402 -0.9613061
## x            0.4462897  0.5531801
```

```r
confint(fit_2h)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0077485 -0.9683041
## x1           0.4792205  0.5205743
```

```r
confint(fit_2i)
```

```
##                  2.5 %     97.5 %
## (Intercept) -1.0704405 -0.7118552
## x1           0.3110958  0.6870395
```

Question 3

a)

```r
n=10;
x<-rnorm(n);
y<-4.20+6.9*x+rnorm(n,0,2.5)
fit_1<-lm(y~x)
print("Beta 0 & 1")
```

```
## [1] "Beta 0 & 1"
```

```r
summary(fit_1)$coefficients[1:2,1]
```

```
## (Intercept)           x
##    4.115329    7.074350
```

b)

```r
vcov(fit_1)
```

```
##             (Intercept)          x
## (Intercept)   0.6952192 -0.2024136
## x            -0.2024136  0.3839785
```

c)

```r
beta_0_list <- c(1:1000)
beta_1_list <- c(1:1000)
for (i in 1:1000){
  n=10
  x<-rnorm(n)
  y<-4.20+6.9*x+rnorm(n,0,2.5)
  fit_2<-lm(y~x)
  beta_0_list[i] <- summary(fit_2)$coefficients[1,1]
  beta_1_list[i] <- summary(fit_2)$coefficients[2,1]
}
print('Covariance')
```

```
## [1] "Covariance"
```

```r
cov(beta_0_list, beta_1_list)
```

```
## [1] -0.0003285326
```

Question 4

```r
# 1

q_4_func <- function(n) {
  x<-rnorm(n);
  y<-4.20+6.9*x+rnorm(n,0,2.5)
  fit_2<-lm(y~x)
  t0 <- summary(fit_2)$coefficients[2,3]
  return(t0)
}

t0 <- q_4_func(100)
cat("The value of t0: ", t0)
```

```
## The value of t0:  24.85661
```

```r
# 2
values = list()
for (i in 1:1000) {
  values[i] <- q_4_func(100)

}
count = 0
for (k in values){
  if(k > t0){
    count = count + 1
  }
}
portion <- count/1000
cat("Portion of ti > t0: ", portion)
```

```
## Portion of ti > t0:  0.837
```