# Home_Work_2

*Group: Bob [Maryla Wozniak, Shakila Hoque, Elai Shalev, Matthew Perez]*

*10/4/2019*

## Initial setup for following assignment

### 1. Clearing the slate

In order to avoid potential issues while running the code it's best to start with a fresh slate

```
# THE FOLLOWING COMMAND WILL CLEAR ALL VARIABLES IN THE ENVIRONMENT
# SAVE ANY FILES OR DATA YOU WANT TO HOLD ONTO BEFORE RUNNING IT
rm(list = ls())
# THE FOLLOWING COMMAND CLEARS THE PLOT PANE AND MAY SAVE AND CLOSE
# ANY OPEN FILES YOU MAY BE EDITING
# SAVE ANY FILES OR DATA YOU WANT TO HOLD ONTO BEFORE RUNNING IT
# IF NO PLOTS AOR VISUALS ARE RUNNING IT WILL PRINT AN ERROR, IGNORE THAT.
dev.off()
```

```
## null device
##           1
```

```
## NOTE: THE FOLLWOING SECTION WAS ADDED AFTER THE INITIAL
## INITIAL SUBMISSION OF THE PROJECT


## Step by step instructions
## Set the working dorectory to the location where the file was saved

## Setp: 1
## The 'rstudioapi' package allows you to access
## system information using the R language
## the 'getSourceEditorContext' function allows you
## to retreive informaton about the source files locaiton
## this includes the file path without you having to
## know where the fiole is. THe computer doest that work for you.

## rstudioapi::getSourceEditorContext()


## Step: 2
## The line above is of type 'list' meaning there are multiple
## peices of information contained within. In order to get the
## information we need we need to acess the file path
## under the 'path' acesor in the return of the functioncall
## to do that we append '$path' this will print the file
## path including the name of the file.

## rstudioapi::getSourceEditorContext()$path


## Step: 3
```

```
## In order to omit the name of the file from printing
## we can use the 'dirname' function that will ignore
## file names at the end of a file path and print
## only the path leading up to where the file is saved.

## dirname(rstudioapi::getSourceEditorContext()$path)


## Step: 4
## Lastly we use the 'setwd' comand to set the
## directory to the returned value of the entire function
## call in order to set the working directory to the
## locaiton of the file without ever having to look for where
## the file is actually stored.

## setwd(dirname(rstudioapi::getSourceEditorContext()$path))

## NOTE: THE LINES ABOVE ARE COMMENTED OUT IN ORDER TO
## PREVENT COMPILATION ERRORS IN 'RMD' AND OTHER 'TEX' BASED
## ENGINES. THIS IS A COMMAND IN ORDER TO SET YOUR WORKING
## DIRECTORY AND SHOULD BE COMMENTED OUT OR COMPLETELY
## OMITED FROM THE FINAL PRODUCT/PRESENTATION.
```

## 2. Setting the working directory

```
# COPY THE OUTPUT FROM EXECUTING THE FOLLOWING LINE
current_working_directory = getwd()
# AND PASS THE STRING AS A PARAMETER TO THE FOLLOWING COMMAND
setwd(current_working_directory)
```

## 3. Importing the data

```
web = "http://faculty.marshall.usc.edu/gareth-james/ISL/Auto.data"
# THE ' na.string="?" ' PARAMETER WILL MARK THE RECORDS WITH '?'
# THIS WILL HELP WITH CLEANING
Auto=read.table(web,header=T,na.string="?")
```

### 4. Cleaning and formattig the data

The dataset provided by the University of South California is not compppletely clean so we need to omit a few records

```
# CHECKING FOR MISSING DATA

# CHECK SIZE OF DATASET
dim(Auto)

# COUNT THE ENTRIES THAT WERE MARKED USING ' na.string="?" '
colSums(is.na(Auto))

# REDEFINE THE DATASET OMITTING THE MARKED VALUES
Auto=na.omit(Auto)
```

```r
# COMPARE MODIFIED SIZE OF DATASET TO ORIGINAL OUTPUT
dim(Auto)

# OPTIONAL OUTPUT OF TABLE CONFIRMING IF ANY OF THE VALUES WERE MARKED AND NOT OMITED
# NOTE: RUNNING THE FOLLOWING COMAND WILL PRINT A TABLE THE SIZE OF THE DATASET
# IT'S NOT ADVISED TO USE ON LARGE DATASETS WITH MORE THAN 300 RECORDS

# is.na(Auto)

# ATTACH COLOMN HEADERS AS REFERENCE VARIABLES IN THE ENVIRONMENT FOR EASIER ACCESS LATER
attach(Auto)
```

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

Use the steps taken in the **initial setup(link)** before following allong

### 8. (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.

```
lm.fit = lm(Auto$mpg~Auto$horsepower)
lm.fit
```

```
##
## Call:
## lm(formula = Auto$mpg ~ Auto$horsepower)
##
## Coefficients:
##     (Intercept)  Auto$horsepower
##         39.9359          -0.1578
```

```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = Auto$mpg ~ Auto$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 ***
## Auto$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
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

#### 8. i. Is there a relationship between the predictor and the response?

**Comments**

- Yes, there is a connection between horsepower and mpg

#### 8. ii. How strong is the relationship between the predictor and the response?

**Comments**

- The correlation is strong: -0.78

**8. iii. Is the relationship between the predictor and the response positive or negative?**

**Comments**

- The correlation is negative

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

```
attach(Auto)
predict(lm(mpg~horsepower),data.frame(horsepower=98),interval="confidence")
```

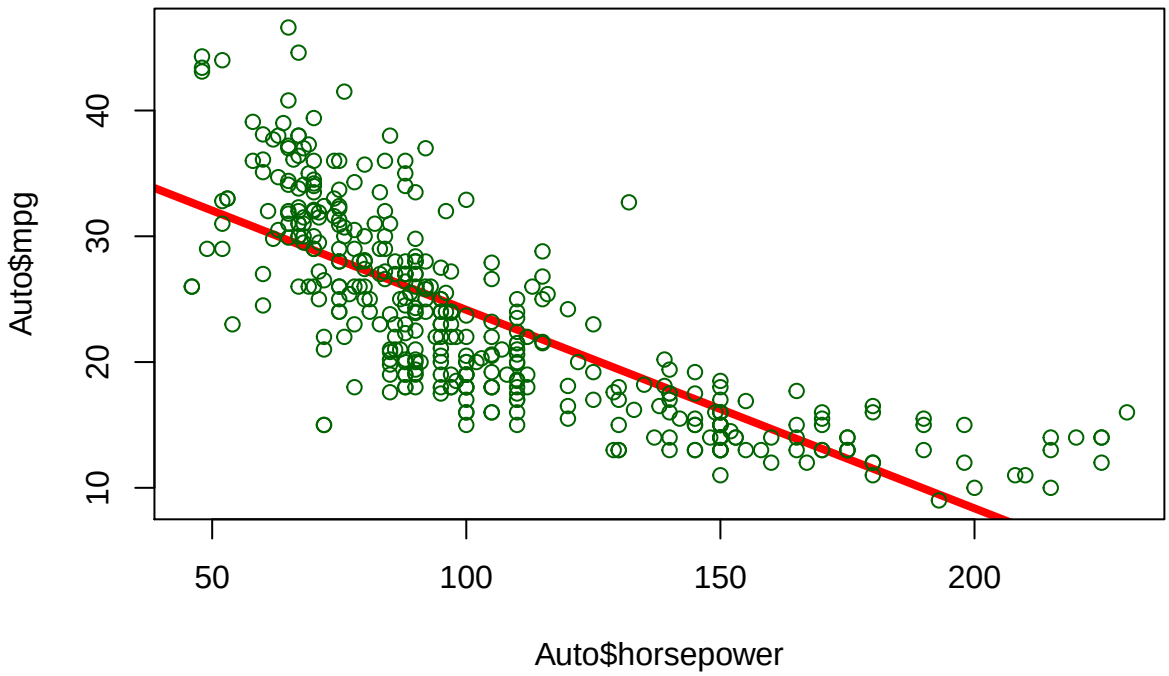
```
##        fit      lwr      upr
## 1 24.46708 23.97308 24.96108
```

```
predict(lm(mpg~horsepower),data.frame(horsepower=98),interval="prediction")
```

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

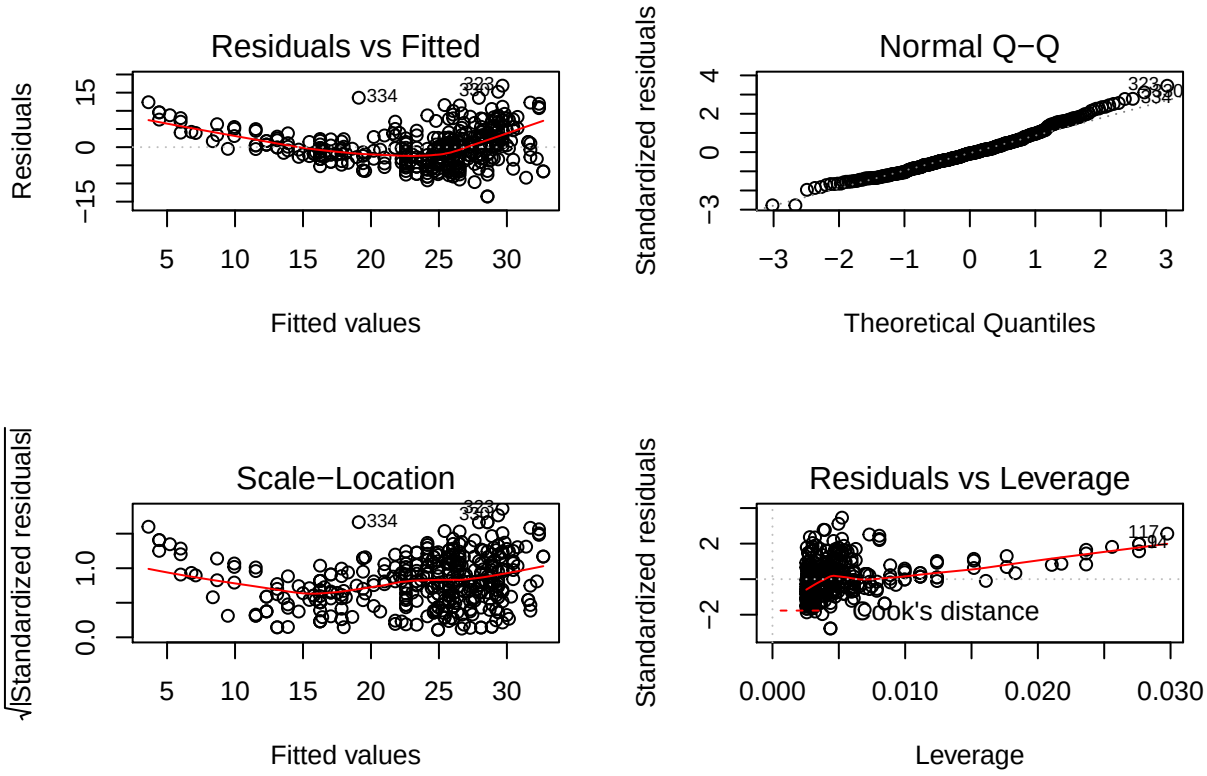
## 8. (b) Plot the response and the predictor. Use the abline() function to display the least squares regression line.

```
plot(Auto$horsepower,Auto$mpg, pch=1, col="darkgreen", abline(lm.fit,col="red",lwd=4))
```




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

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




**Comments:**

- In the residual vs. fitted model, the plots show a U shape pattern, that is violating the linearity. In the second model, The normal Q-Q, the plot shows there is a linear relationship. The 3rd model represents the variance of the error stayed the same and the 4th model showed there were few outliers and leverage but the fitted line violated linearity

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

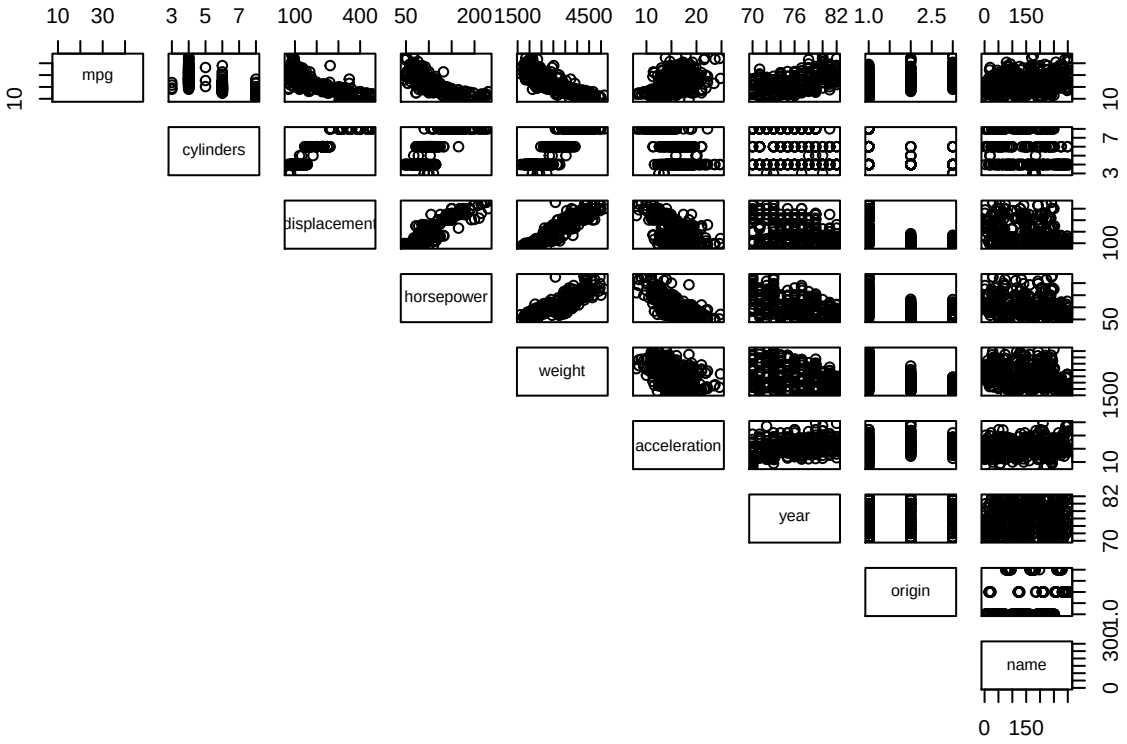
Use the steps taken in the **initial setup(link)** before following allong

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

```
names(Auto)
```

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

```
pairs( Auto, lower.panel = NULL)
```




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

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

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7762599   -0.8044430         NA -0.8317389
## cylinders    -0.7762599  1.0000000    0.9509199         NA  0.8970169
## displacement -0.8044430  0.9509199    1.0000000         NA  0.9331044
## horsepower           NA         NA           NA          1         NA
## weight       -0.8317389  0.8970169    0.9331044         NA  1.0000000
## acceleration  0.4222974 -0.5040606   -0.5441618         NA -0.4195023
## year          0.5814695 -0.3467172   -0.3698041         NA -0.3079004
## origin        0.5636979 -0.5649716   -0.6106643         NA -0.5812652
```

```
##              acceleration       year     origin
## mpg             0.4222974  0.5814695  0.5636979
## cylinders      -0.5040606 -0.3467172 -0.5649716
## displacement   -0.5441618 -0.3698041 -0.6106643
## horsepower             NA         NA         NA
## weight         -0.4195023 -0.3079004 -0.5812652
## acceleration    1.0000000  0.2829009  0.2100836
## year            0.2829009  1.0000000  0.1843141
## origin          0.2100836  0.1843141  1.0000000
```

**9. (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.**

```
fit=lm(Auto$mpg~
         Auto$cylinders+
         Auto$displacement+
         Auto$horsepower+
         Auto$weight+
         Auto$acceleration+
         Auto$year+
         Auto$origin
       )
summary(fit)
```

```
##
## Call:
## lm(formula = Auto$mpg ~ Auto$cylinders + Auto$displacement +
##     Auto$horsepower + Auto$weight + Auto$acceleration + Auto$year +
##     Auto$origin)
##
## 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 ***
## Auto$cylinders     -0.493376   0.323282  -1.526  0.12780
## Auto$displacement   0.019896   0.007515   2.647  0.00844 **
## Auto$horsepower    -0.016951   0.013787  -1.230  0.21963
## Auto$weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## Auto$acceleration   0.080576   0.098845   0.815  0.41548
## Auto$year           0.750773   0.050973  14.729  < 2e-16 ***
## Auto$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
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

```r
coefficients(fit)
```

```
##       (Intercept)    Auto$cylinders Auto$displacement   Auto$horsepower
##     -17.218434622      -0.493376319       0.019895644      -0.016951144
##       Auto$weight Auto$acceleration         Auto$year       Auto$origin
##      -0.006474043       0.080575838       0.750772678       1.426140495
```

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

**Comments**

- Yes, there is a relationship between the predictor and the response, we confirmed it by looking at the H null hypothesis. The p-value corresponds the f stats at 2.037

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

**Comments**

- All the predictors are statistically significant, excluding "cylinder", "horsepower" and "acceleration" because p-value associated each predictor's t statistics

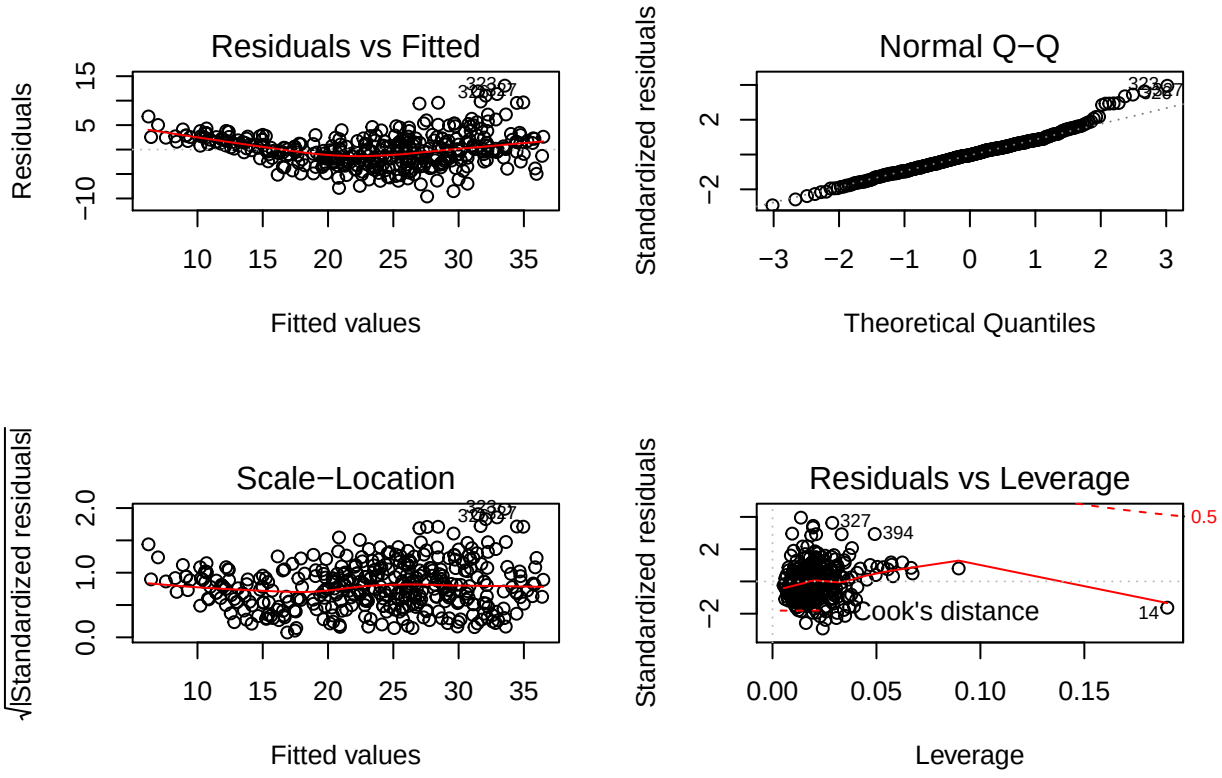
**9. iii. What does the coefficient for the year variable suggest?**

**Comments**

- The coefficient for the year suggests that as we increase the year by 1 the fuel-efficient also increases by 0.750 while the other predictor value stays the same. For example, the car became more fuel-efficient every year.

## 9. (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)
```




**Comments**

- The residual vs. fitted model indicated a curvier line and that violates the linearity. The Q-Q plot does satisfy the linearity because of having a straight fitted line with a positive strong correlation between standard residual and theoretical quantiles. The scale location model shows a lot of unusual outliers in the model but the standard residual vs leverage have shown fewer outliers and have a high leverage point in the model.

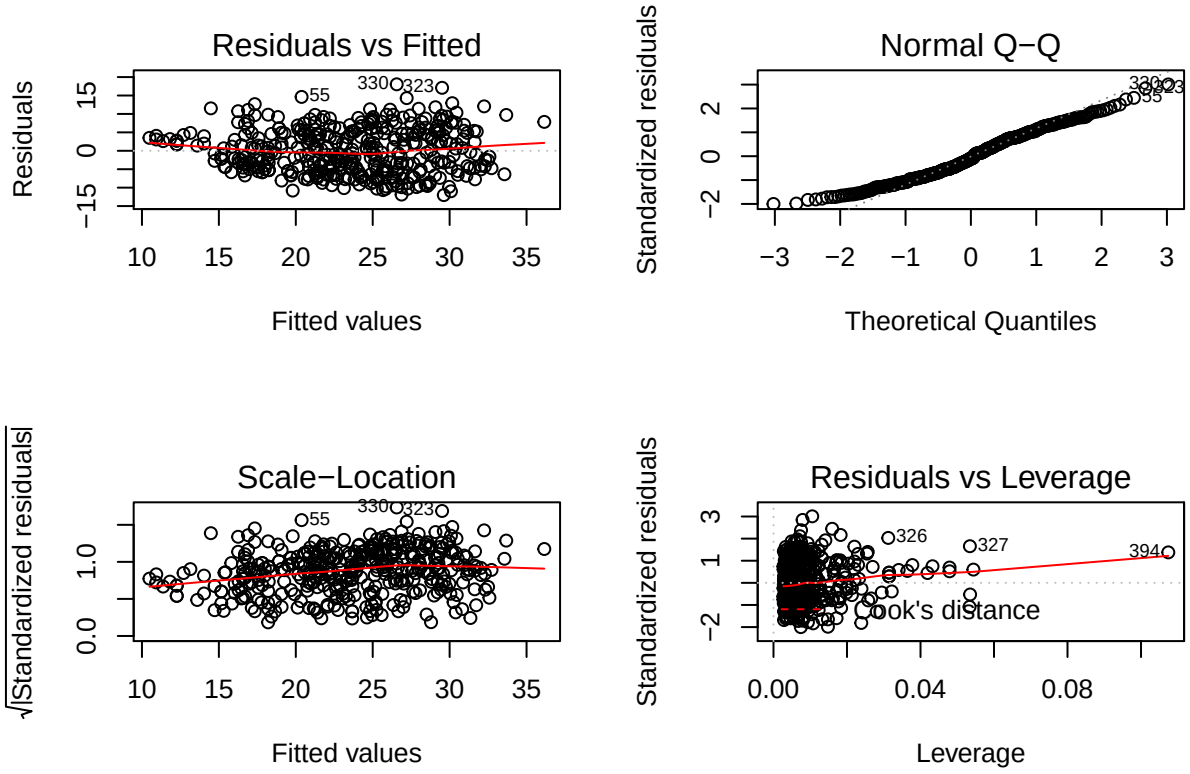
## 9. (e) Use the * and : symbols to fit linear regression models with interaction effects. Do any interactions appear to be statistically significant?

```
fitNew=lm(Auto$mpg~Auto$year*Auto$acceleration)
summary(fitNew)
```

```
##
## Call:
## lm(formula = Auto$mpg ~ Auto$year * Auto$acceleration)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.0206  -4.8624  -0.6655   4.7175  18.0394
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -89.29202   33.94879  -2.630  0.00887 **
## Auto$year                       1.32408    0.45225   2.928  0.00361 **
## Auto$acceleration               2.05917    2.16612   0.951  0.34238
## Auto$year:Auto$acceleration    -0.01675    0.02871  -0.583  0.56003
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.03 on 393 degrees of freedom
## Multiple R-squared:  0.4109, Adjusted R-squared:  0.4064
## F-statistic: 91.36 on 3 and 393 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(fitNew)
```




**Comments**

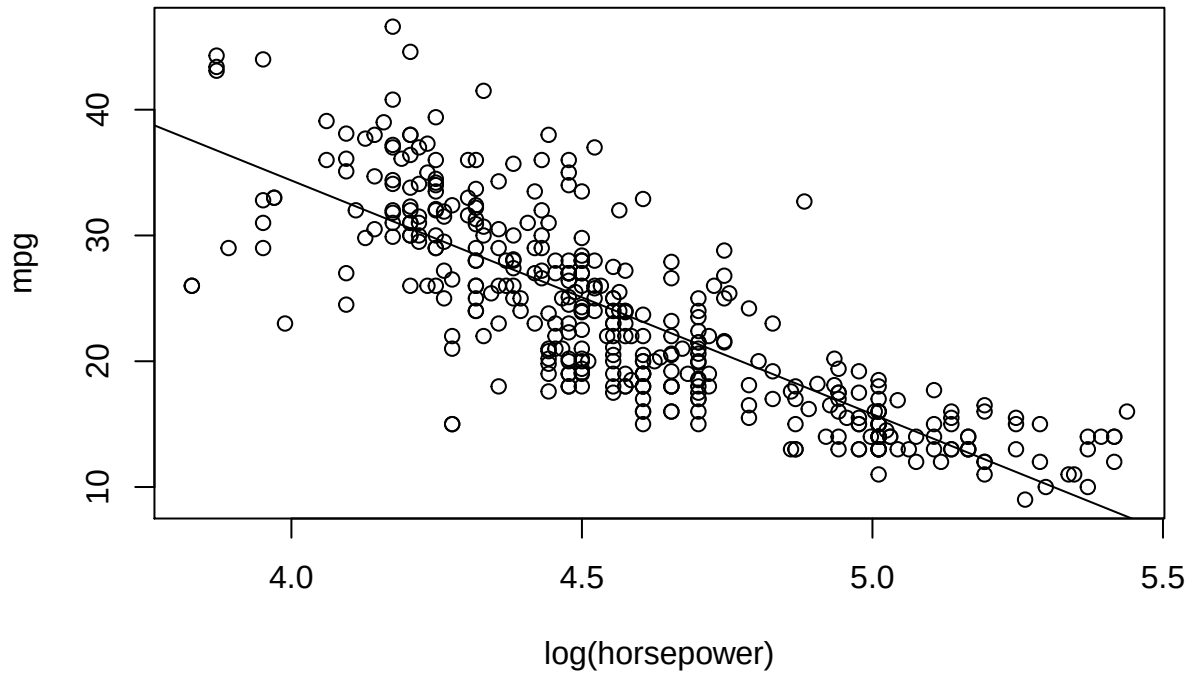
- When we look at the p-value the association between the displacement and the weight seems to be more statistically significant, however, the association between the cylinder and displacement does not look statistically significant.

## 9. (f) Try a few different transformations of the variables, such as log(X), sqrt(X), 2X. Comment on your findings.

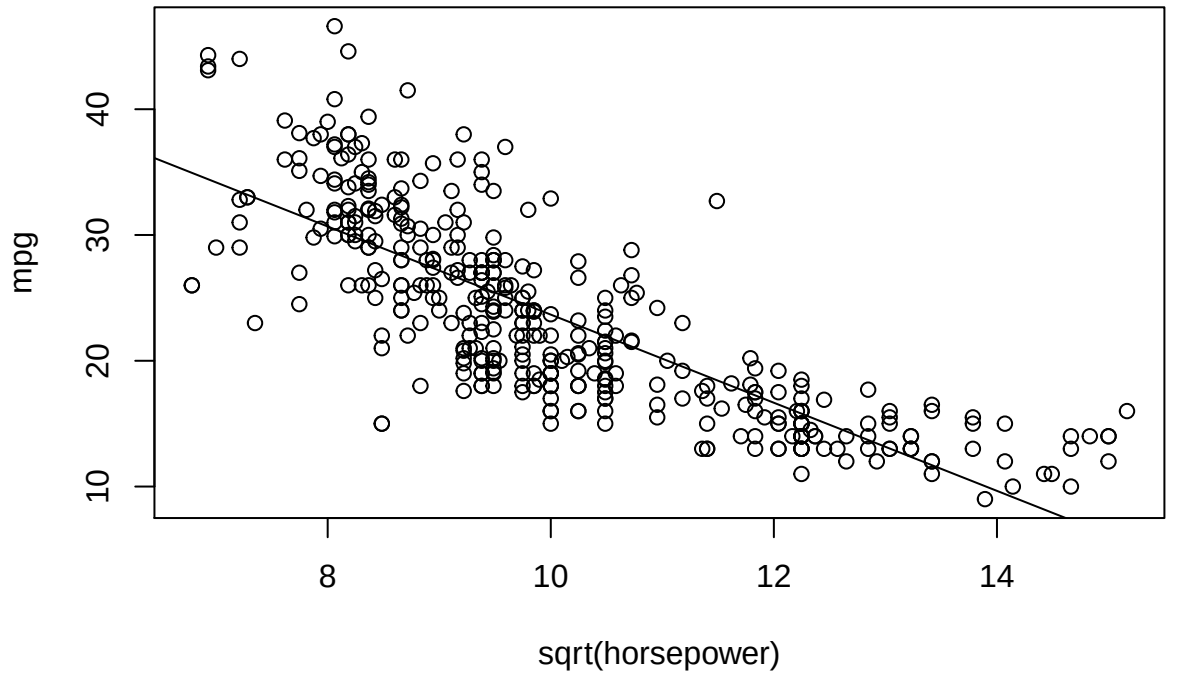
```
attach(Auto)
```

```
## The following objects are masked from Auto (pos = 3):
##
##     acceleration, cylinders, displacement, horsepower, mpg, name,
##     origin, weight, year
```

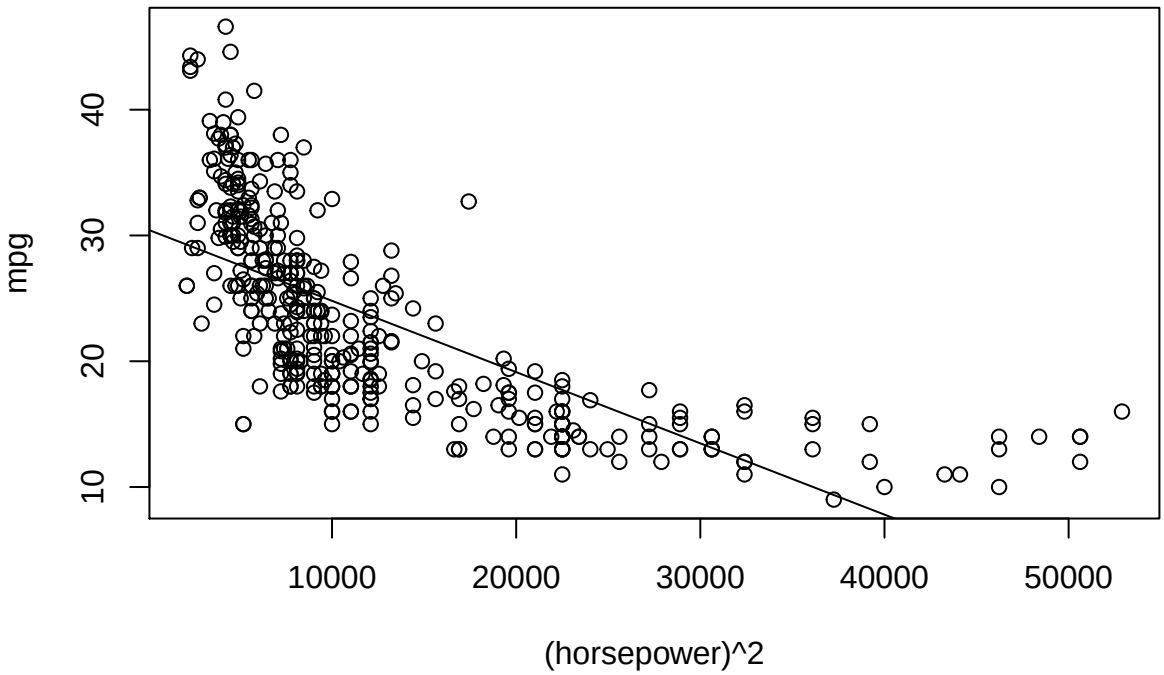
```
plot( log(horsepower), mpg, abline(lm( mpg~log(horsepower))))
```




```
plot( sqrt(horsepower), mpg, abline(lm(mpg~sqrt(horsepower))))
```




```
plot( (horsepower)^2, mpg, abline(lm(mpg~I(horsepower^2))))
```




```
#?abline
```

**Comments**

- When we used “horsepower” as our single predictor and to try different transformation, we noticed only the log transformation gave us the most liner looking plot for the “horsepower”