Homework 2

4375 Machine Learning with Dr. Mazidi

Tyler Echols

6/6/2022

This homework gives practice in using linear regression in two parts:

- Part 1 Simple Linear Regression (one predictor)
- Part 2 Multiple Linear Regression (many predictors)

You will need to install package ISLR at the console, not in your script.

Problem 1: Simple Linear Regression

Step 1: Initial data exploration

- Load library ISLR (install.packages() at console if needed)
- Use names() and summary() to learn more about the Auto data set
- Divide the data into 75% train, 25% test, using seed 1234

```
# your code here

library(ISLR)
names(Auto)

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

summary(Auto)

##       mpg          cylinders      displacement     horsepower
weight
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.
:1613
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st
Qu.:2225
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Median
:2804
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean
:2978
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd
Qu.:3615
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.
:5140
##
```

```
##   acceleration         year           origin                      name    
##  Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :  5  
##  1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :  5  
##  Median :15.50   Median :76.00   Median :1.000   toyota corolla    :  5  
##  Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :  4  
##  3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :  4  
##  Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:  4  
##                                                  (Other)           :365

Auto <- Auto


set.seed(1234)
sample <- sample.int(n=nrow(Auto), size = floor(.75*nrow(Auto)), replace = F)
train <- Auto[sample,]
test <- Auto[-sample,]
```

Step 2: Create and evaluate a linear model

- Use the lm() function to perform simple linear regression on the train data with mpg as the response and horsepower as the predictor
- Use the summary() function to evaluate the model
- Calculate the MSE by extracting the residuals from the model like this: mse <- mean(lm1$residuals^2)
- Print the MSE
- Calculate and print the RMSE by taking the square root of MSE

```
# your code here


lm1 <- lm(formula = Auto$mpg ~ Auto$horsepower, data = Auto)
summary(lm1)

## 
## Call:
## lm(formula = Auto$mpg ~ Auto$horsepower, data = Auto)
## 
## 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
```

```
## Multiple R-squared:  0.6059, Adjusted R-squared:  0.6049
## F-statistic: 599.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

```
mse <- mean(lm1$residuals^2)
print(mse)
```

```
## [1] 23.94366
```

```
rmse <- sqrt(mse)
print(paste("rmse: ", rmse))
```

```
## [1] "rmse:  4.89322623006571"
```

Step 3 (No code. Write your answers in white space)

- Write the equation for the model, y = wx + b, filling in the parameters w, b and variable names x, y
- Is there a strong relationship between horsepower and mpg?
- Is it a positive or negative correlation?
- Comment on the RSE, R^2, and F-statistic, and how each indicates the strength of the model
- Comment on the RMSE and whether it indicates that a good model was created

y = wx + b

w: being the slop of the graph b: is the bais of people wanting a choice between the 2
x: is the number of horse power y: is the mpg

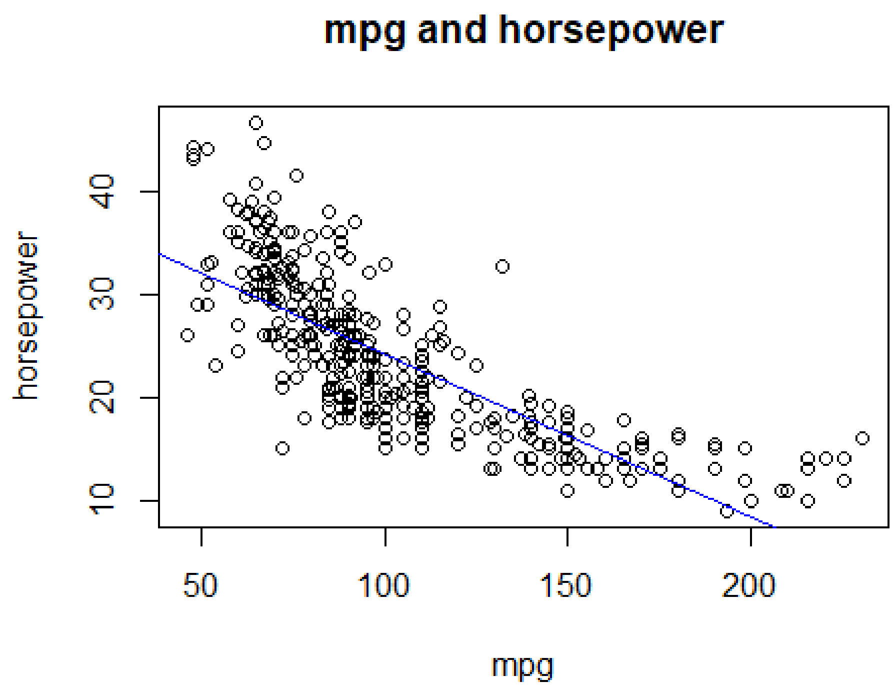
The relation between the mpg and horse power is that more people are wanting to hold more gas and ignore how much the car is able to speed up and burn it. Resulting in a negative downward graph.

Step 4: Examine the model graphically

- Plot train$mpg~train$horsepower
- Draw a blue abline()
- Comment on how well the data fits the line
- Predict mpg for horsepower of 98. Hint: See the Quick Reference 5.10.3 on page 96
- Comment on the predicted value given the graph you created

Your commentary here: there seemes to be an initial high demand for cars that are able to get 100 mpg and 20 horsepower.

```
# your code here
plot(Auto$mpg~Auto$horsepower, main = "mpg and horsepower", xlab = "mpg",
ylab = "horsepower")
abline(lm1, col="Blue")
```




```
pred1 <- predict(lm1,data=train)
```

Step 5: Evaluate on the test data

- Test on the test data using the predict function
- Find the correlation between the predicted values and the mpg values in the test data
- Print the correlation
- Calculate the mse on the test results
- Print the mse
- Compare this to the mse for the training data
- Comment on the correlation and the mse in terms of whether the model was able to generalize well to the test data

Your commentary here: by the 2 numbers given it would seem that it has established a certain ammount of proabilities and an even level of given valuse from the test data

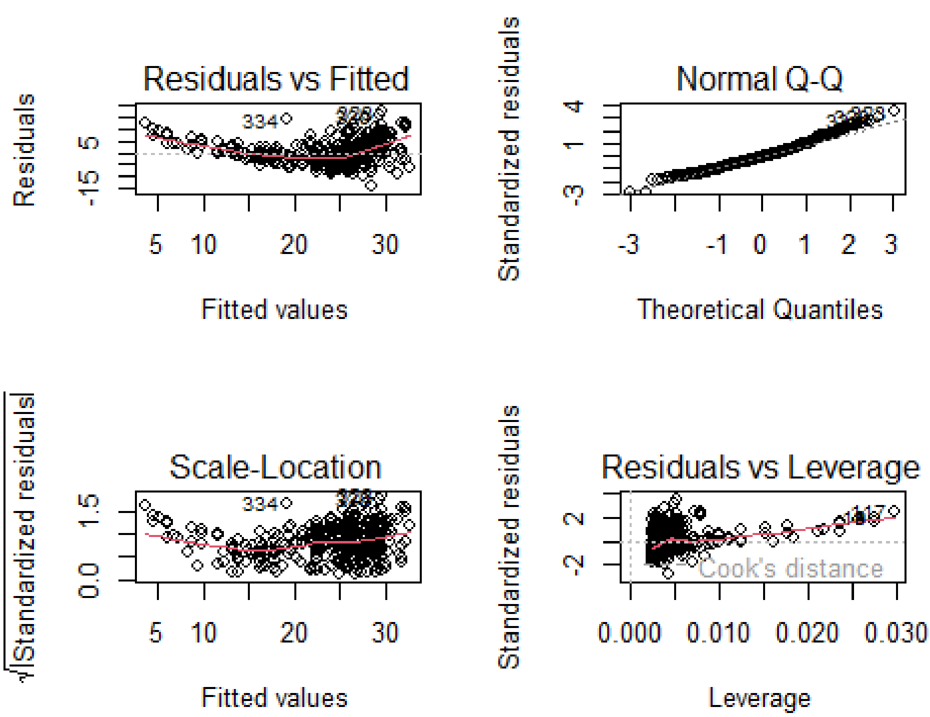
```
# your code here
cor1 <- cor(pred1, Auto$mpg)
print(cor1)

## [1] 0.7784268

mse1 <- mean((pred1-Auto$mpg)^2)
print(mse1)

## [1] 23.94366
```

Step 6: Plot the residuals

- Plot the linear model in a 2x2 arrangement
- Do you see evidence of non-linearity from the residuals?

Your commentary here: There is evidance of non-linearity due to the proof of a few outliers in the data.

```r
# your code here
par(mfrow=c(2,2))
plot(lm1)
```



Step 7: Create a second model

- Create a second linear model with log(mpg) predicted by horsepower
- Run summary() on this second model
- Compare the summary statistic R^2 of the two models

Your commentary here: Both graphs seem to differ on the valuse of both their Median, 3Q, and Max values.

```r
# your code here
lm2 <- lm(log(Auto$mpg) ~ Auto$horsepower, data=Auto)
summary(lm2)

##
## Call:
## lm(formula = log(Auto$mpg) ~ Auto$horsepower, data = Auto)
```

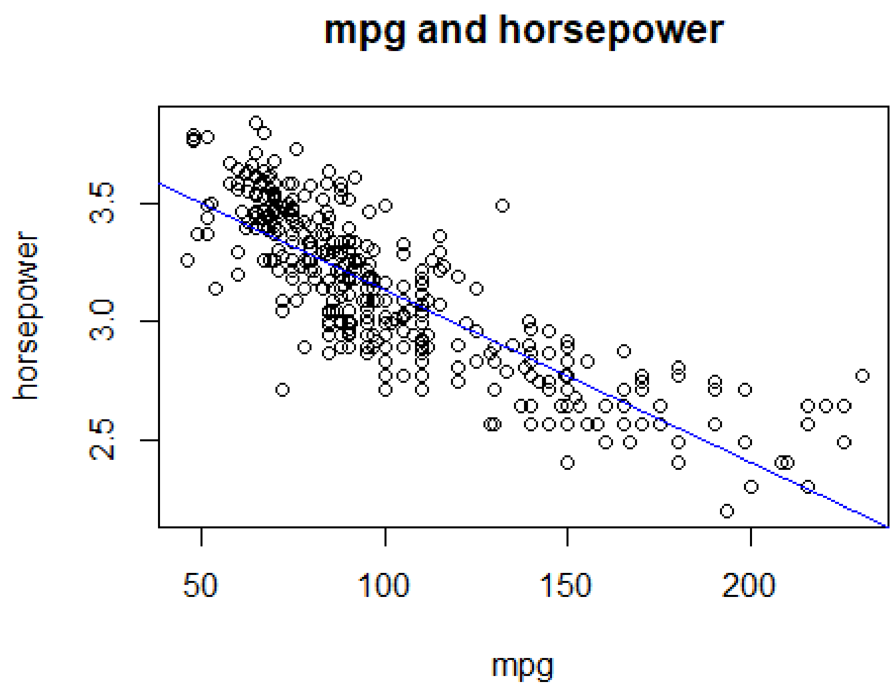
```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62839 -0.12814  0.00914  0.12636  0.59489
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.8644668  0.0277632  139.19   <2e-16 ***
## Auto$horsepower -0.0073338  0.0002494  -29.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1898 on 390 degrees of freedom
## Multiple R-squared:  0.6892, Adjusted R-squared:  0.6884
## F-statistic: 864.7 on 1 and 390 DF,  p-value: < 2.2e-16
```

Step 8: Evaluate the second model graphically

- Plot log(train$mpg)~train$horsepower
- Draw a blue abline()
- Comment on how well the line fits the data compared to model 1 above

Your commentary here:There seems to be less points between the line and more either above or below the line. Also a sigificant number set on the horsepower value.

```
# your code here
plot(log(Auto$mpg) ~ Auto$horsepower, main = "mpg and horsepower", xlab =
"mpg", ylab = "horsepower")
abline(lm2, col="Blue")
```



Step 9: Predict and evaluate on the second model

- Predict on the test data using lm2
- Find the correlation of the predictions and log() of test mpg, remembering to compare pred with log(test$mpg)
- Output this correlation
- Compare this correlation with the correlation you got for model
- Calculate and output the MSE for the test data on lm2, and compare to model 1. Hint: Compute the residuals and mse like this:

```
residuals <- pred - log(test$mpg)
mse <- mean(residuals^2)
```

Your commentary here: There appears to be a correlation around the 0.77 for both graphs.

```
# your code here
pred2 <- predict(lm2, data=train)
cor2 <- cor(pred2,Auto$mpg)
print(cor2)

## [1] 0.7784268

lm2 <- lm(log(Auto$mpg) ~ pred2, data = train )
residuals <- pred2 - log(test$mpg)
mse <- mean(residuals^2)
rmse <- sqrt(mse)
```

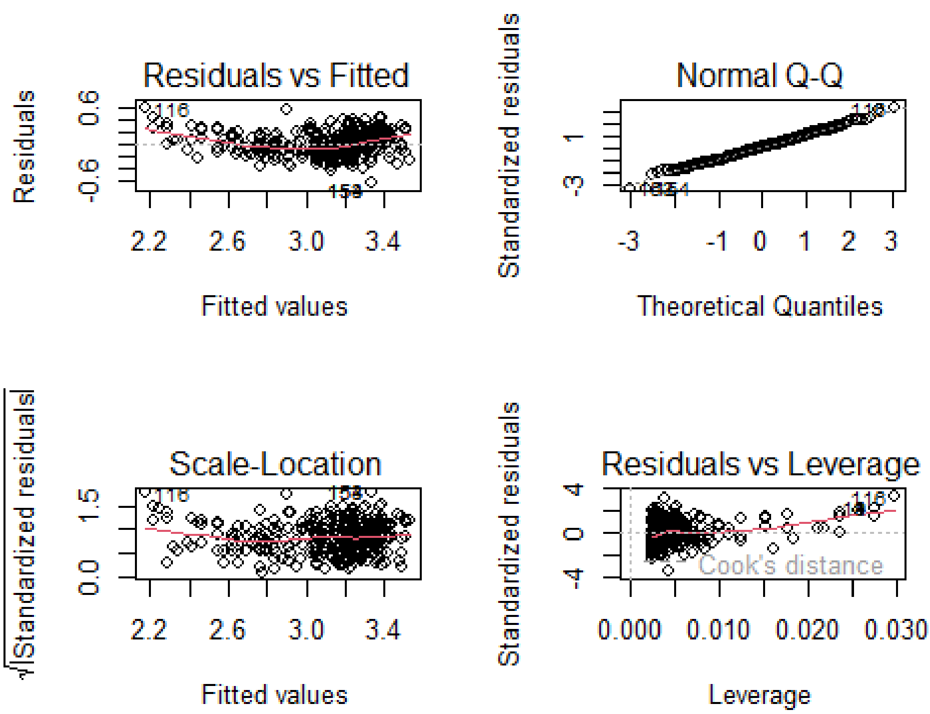
```
print(paste('correlation:', cor2))

## [1] "correlation: 0.778426783897776"

print(paste('mse:', mse))

## [1] "mse: 0.169289045242884"

print(paste('rmse:', rmse))

## [1] "rmse: 0.411447499983758"
```

Step 10: Plot the residuals of the second model

- Plot the second linear model in a 2x2 arrangement
- How does it compare to the first set of graphs?

Your commentary here: The fact that these graphs all end up starting at a lower initial Y value.

```
# your code here
par(mfrow=c(2,2))
plot(lm2)
```



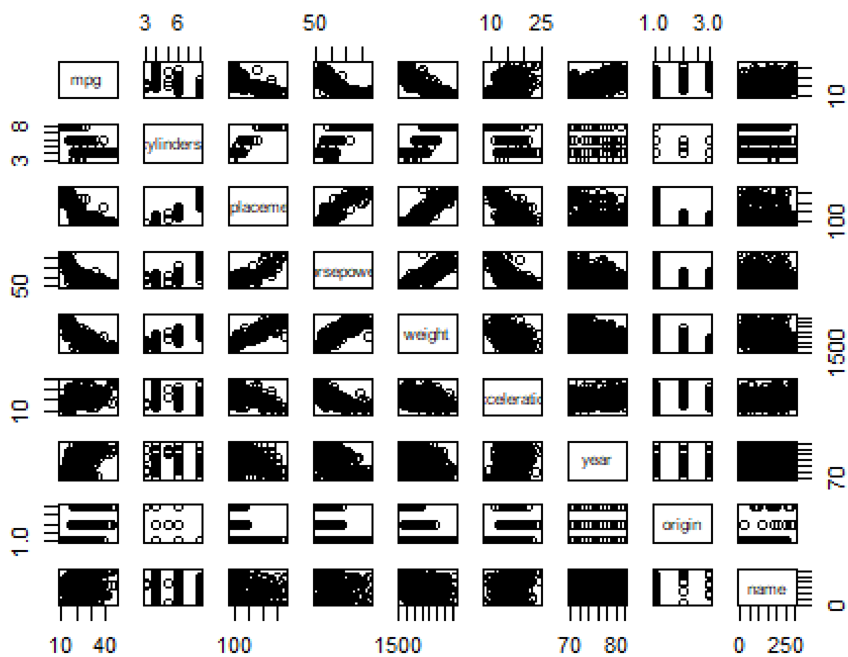
Problem 2: Multiple Linear Regression

Step 1: Data exploration

- Produce a scatterplot matrix of correlations which includes all the variables in the data set using the command “pairs(Auto)”
- List any possible correlations that you observe, listing positive and negative correlations separately, with at least 3 in each category.

Your commentary here: The 3 Positive correlations Visible are between Horsepower, weight, and Displacement. While the apparent negative values are with horsepower, weight, and acceleration.

```
# your code here
pairs(Auto)
```



Step 2: Data visualization

- Display the matrix of correlations between the variables using function cor(), excluding the “name” variable since is it qualitative
- Write the two strongest positive correlations and their values below. Write the two strongest negative correlations and their values as well.

Your commentary here: The 2 strongest Positive correlations are the years and miles per gallon. While the negative correlation are between displacement, and weight.

```r
# your code here
cor(Auto[, names(Auto) !="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
```

Step 3: Build a third linear model

- Convert the origin variable to a factor
- Use the lm() function to perform multiple linear regression with mpg as the response and all other variables except name as predictors
- Use the summary() function to print the results
- Which predictors appear to have a statistically significant relationship to the response?

Your commentary here: There are a good few that start in the negative area or are given more negaitve values.

```r
# your code here
model = lm(mpg ~. -name, data = Auto)
summary(model)
```

```
##
## 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
```

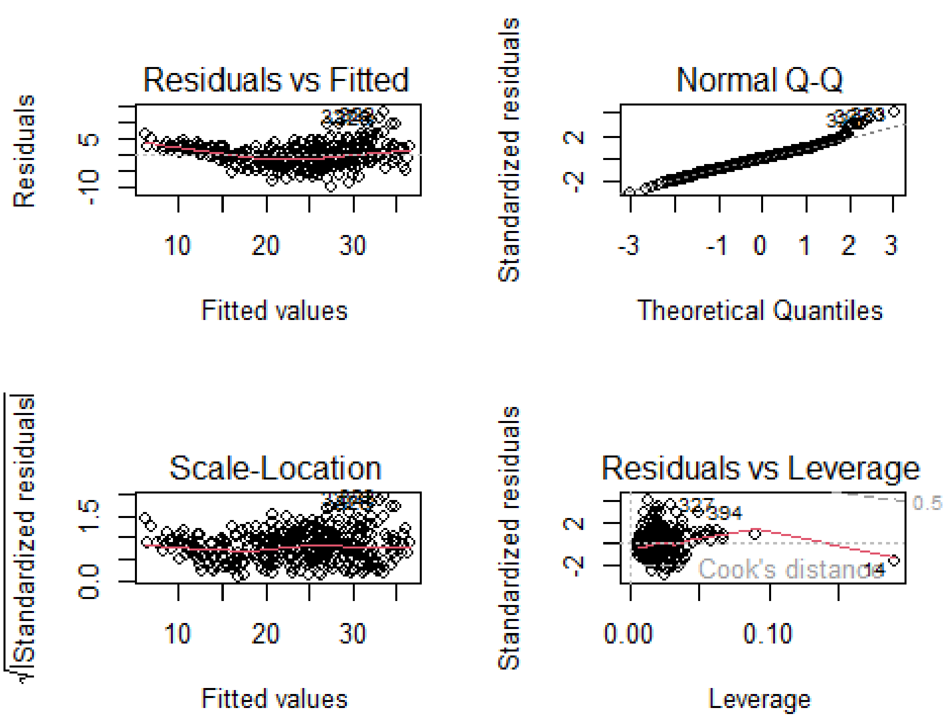
```
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         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
```

Step 4: Plot the residuals of the third model

- Use the plot() function to produce diagnostic plots of the linear regression fit
- Comment on any problems you see with the fit
- Are there any leverage points?
- Display a row from the data set that seems to be a leverage point.

Your commentary here: The leverage point seems to be increasing for a certain amount of time the just immeaditly falls at a rapaid decline.

```r
# your code here
par(mfrow = c(2,2))
plot(model)
```



Step 5: Create and evaluate a fourth model

- Use the * and + symbols to fit linear regression models with interaction effects, choosing whatever variables you think might get better results than your model in step 3 above
- Compare the summaries of the two models, particularly R^2
- Run anova() on the two models to see if your second model outperformed the previous one, and comment below on the results

Your commentary here: There seems to be more of a better result with hte residiuals in the 2nd one rather than the 1st one.

```
# your code here
model1 = lm(mpg ~.-name+displacement:weight, data = Auto)
anova(model, model1)

## Analysis of Variance Table
##
## Model 1: mpg ~ (cylinders + displacement + horsepower + weight +
acceleration +
##     year + origin + name) - name
## Model 2: mpg ~ (cylinders + displacement + horsepower + weight +
acceleration +
##     year + origin + name) - name + displacement:weight
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    384 4252.2
## 2    383 3364.3  1    887.91 101.08 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

summary(model)

##
## 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
```