Homework 2

4375 Machine Learning with Dr. Mazidi

Garrett Strealy

9/7/2021

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)

print("Column names of Auto dataframe:")
```

```
## [1] "Column names of Auto dataframe:"
```

```
names(Auto)
```

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

```
print("Summary of Auto dataframe:")
```

```
## [1] "Summary of Auto dataframe:"
```

```
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
```

```
#data(Auto)

# Set seed so that same sample can be reproduced in future
set.seed(1234)

# Selecting 75% of data as sample from total 'n' rows of the data
i = sample(1:nrow(Auto), nrow(Auto)*0.75, replace=FALSE)
train <- Auto[i, ] # 75% train
test <- Auto[-i, ] # 25% test
```

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(mpg ~ horsepower, data=train) # perform linear regression

print("Summary of Model:")
```

```
## [1] "Summary of Model:"
```

```
summary(lm1) # evaluate model
```

```
##
## Call:
## lm(formula = mpg ~ horsepower, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3675  -3.1682  -0.2885   2.8518  17.1357
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 39.648595   0.814676   48.67   <2e-16 ***
## horsepower  -0.156681   0.007276  -21.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.853 on 292 degrees of freedom
## Multiple R-squared:  0.6136, Adjusted R-squared:  0.6123
## F-statistic: 463.7 on 1 and 292 DF,  p-value: < 2.2e-16
```

```
mse <- mean(lm1$residuals^2) # calculate MSE
print(paste("MSE: ", mse)) # print MSE
```

```
## [1] "MSE:  23.3917550461694"
```

```
print(paste("RMSE: ", sqrt(mse))) # calculate and print RMSE
```

```
## [1] "RMSE:  4.83650235667981"
```

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

mpg = 39.648595 - 0.156681*horsepower

- Is there a strong relationship between horsepower and mpg?

Yes, there is a strong relationship between horsepower and mpg.

- Is it a positive or negative correlation?

A strong negative correlation exists between horsepower and mpg.

- Comment on the RSE, R^2, and F-statistic, and how each indicates the strength of the model

1. The RSE of 4.853 tells us that our model is off by 4.853 mpg on average. The RSE measures how off our model was from the data (the lack of fit of the model). The lower the RSE, the better the model. We would like a lower RSE than 4.853.
2. The R^2 of 0.6136 indicates a moderately strong model, although we would like to see it closer to 1. The closer R^2 is to 1 the more variance in the model is explained by the predictors.

3. The F-statistic of 463.7 accompanied with a p-value < 2.2e-16 indicates a strong model. We would like to see a F-stat greater than 1 and a low p-value, which this model satisfies.

- Comment on the RMSE and whether it indicates that a good model was created

The RMSE of 4.8365 is very close to the RSE of 4.853, and it is similarly informative of how off our model was from the data (the lack of fit of the model) in terms of y (mpg).

Step 4: Examine the model graphically

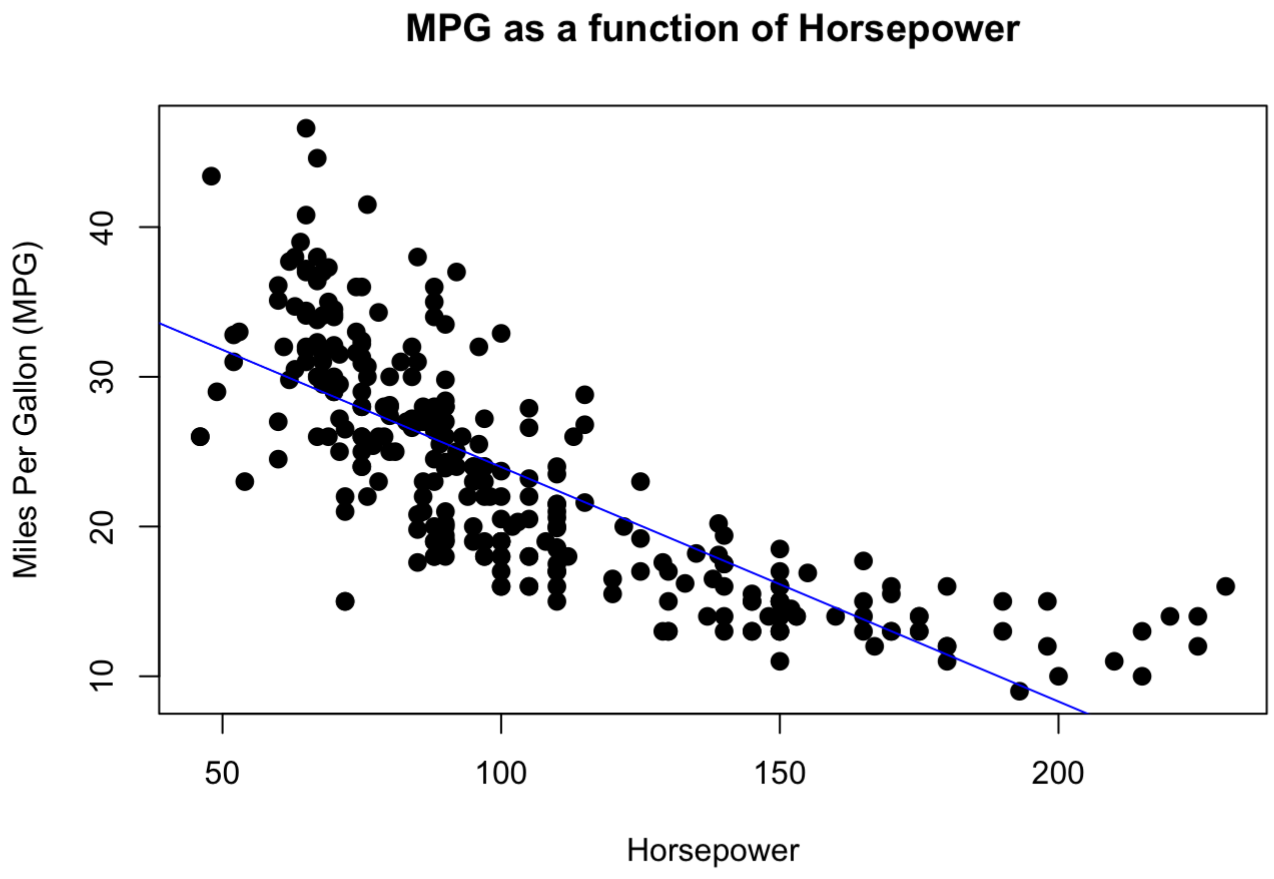
- Plot train$mpg~train$horsepower
- Draw a blue abline()
- Comment on how well the data fits the line
- Predict mpg for horsepower of 98. Hint: See the Quick Reference 5.10.3 on page 96
- Comment on the predicted value given the graph you created

Your commentary here:

The data fits the line moderately well. However, the RSE of 4.85 is evident, as is the high skewness of Auto$horsepower.

The predicted value of 24.3 is consistent with what we could expect from the graph.

```
# your code here
plot(train$mpg ~ train$horsepower, pch = 16, cex = 1.3, col = "black", main = "MPG as a
 function of Horsepower", xlab = "Horsepower", ylab = "Miles Per Gallon (MPG)")
abline(lm1, col="blue")
```



```
predict(lm1, data.frame(horsepower=98), interval="confidence")
```

```
##        fit      lwr      upr
## 1 24.29381 23.72784 24.85978
```

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: There is a strong positive correlation between the predicted values and test data. The MSE of the training data is 23.4, while the MSE of the test data is 25.7. These findings indicate that the model was able to generalize well to the test data.

```
# your code here
p <- predict(lm1, newdata=test)
print(paste("Cor between predicted and test data: ", cor(p, test$mpg)))
```

```
## [1] "Cor between predicted and test data:  0.764210117020695"
```

```
residuals <- p - test$mpg
mseTest <- mean(residuals^2) # calculate MSE
print(paste("Test data MSE: ", mseTest))
```

```
## [1] "Test data MSE:  25.7172652597501"
```

```
print(paste("Train data MSE: ", mse))
```

```
## [1] "Train data MSE:  23.3917550461694"
```

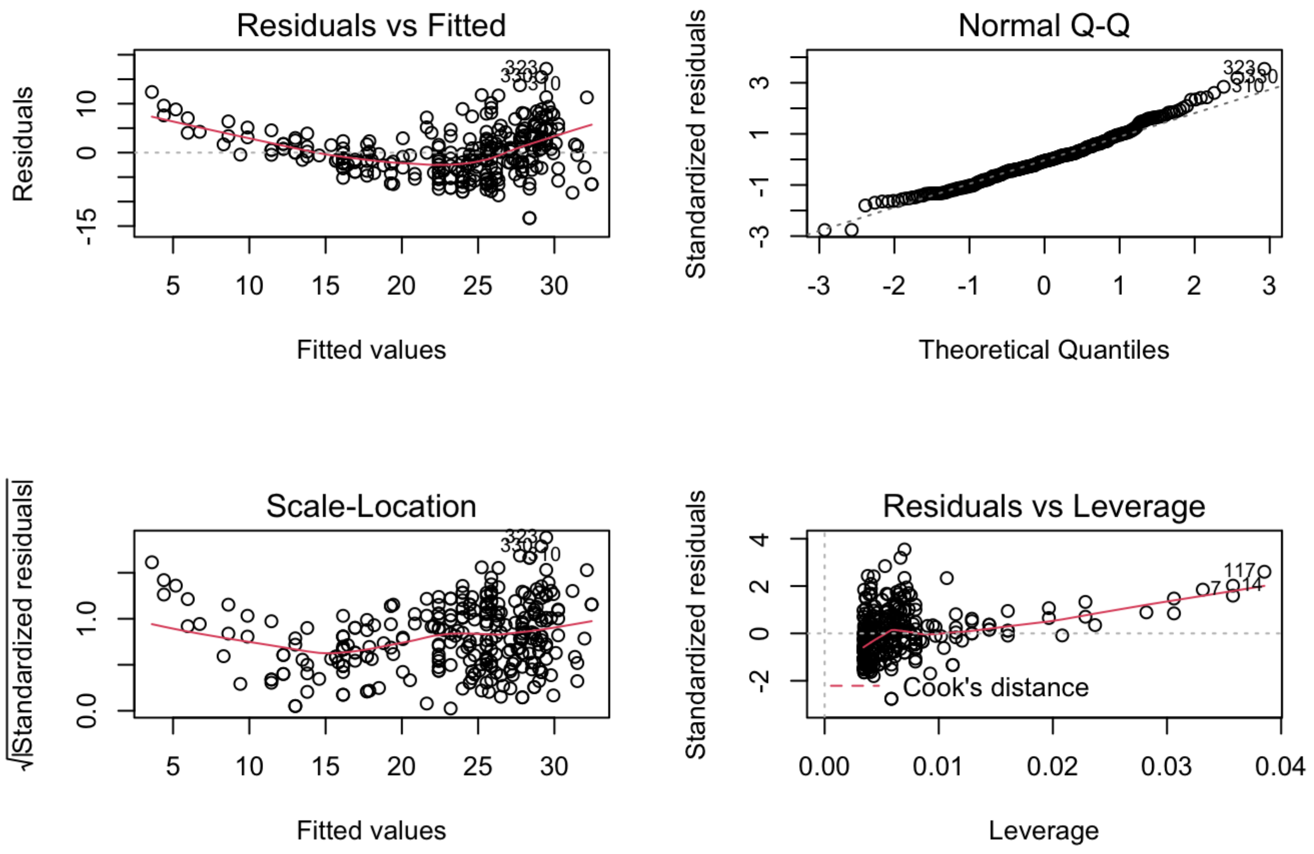
Step 6: Plot the residuals

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

Your commentary here:

The residual plots have a strong curve, indicating issues with the fit. We would like these lines to be more horizontal.

```
# 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: This model has an R^2 of 0.7 compared to model one's 0.6. This model's higher R^2 indicates that it is a stronger model.

```
# your code here
lm2 <- lm(log(mpg) ~ horsepower, train)
print("Summary of Model Two:")
```

```
## [1] "Summary of Model Two:"
```

```
summary(lm2)
```

```
##
## Call:
## lm(formula = log(mpg) ~ horsepower, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62229 -0.12814  0.01443  0.12330  0.61150
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.8631645  0.0319324  120.98   <2e-16 ***
## horsepower  -0.0074003  0.0002852  -25.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1902 on 292 degrees of freedom
## Multiple R-squared:  0.6975, Adjusted R-squared:  0.6965
## F-statistic: 673.3 on 1 and 292 DF,  p-value: < 2.2e-16
```

Step 8: Evaluate the second model graphically

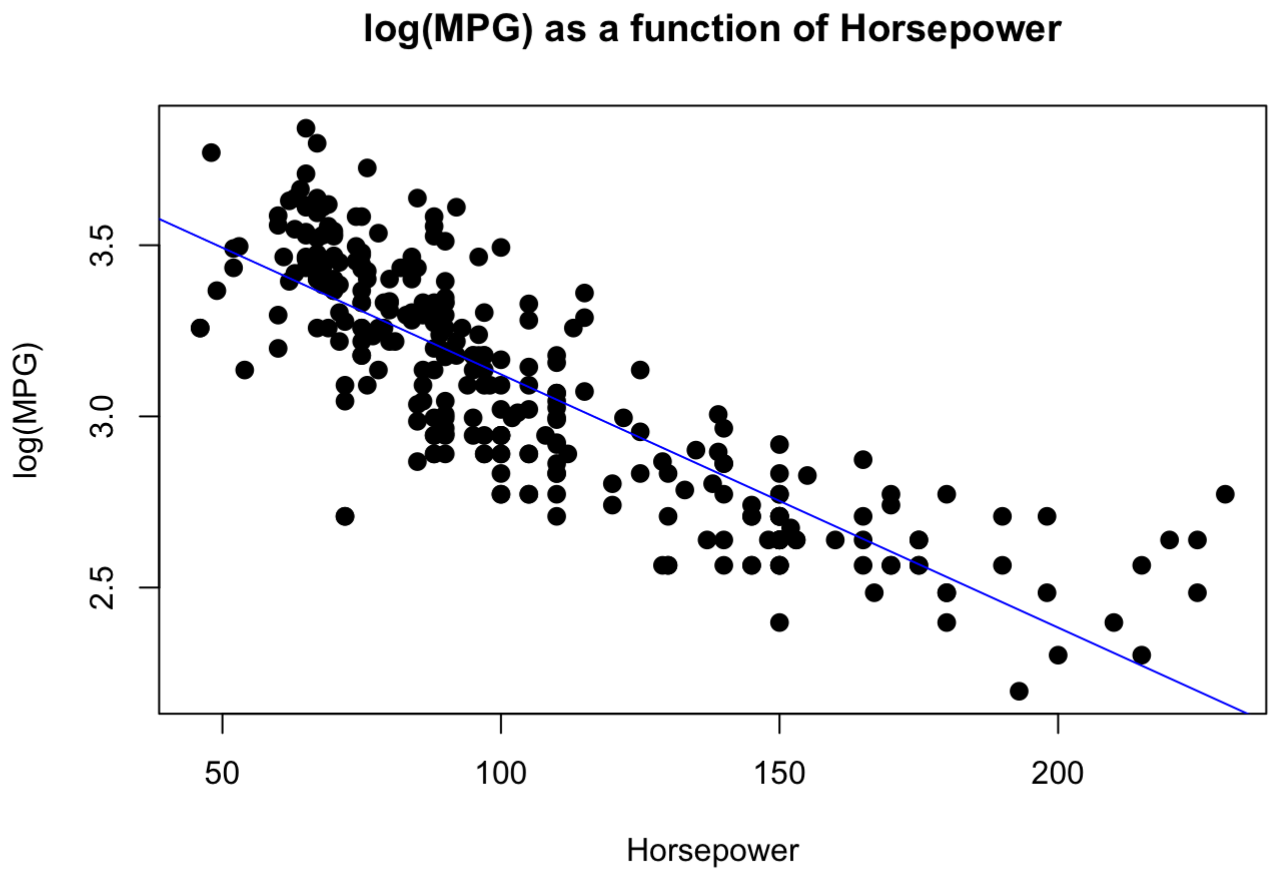
- Plot log(train$mpg)~train$horsepower
- Draw a blue abline()
- Comment on how well the line fits the data compared to model 1 above

Your commentary here:

This data fits the line more closely than Model One's data.

```
# your code here
plot(log(train$mpg) ~ train$horsepower, pch = 16, cex = 1.3, col = "black", main = "log
(MPG) as a function of Horsepower", xlab = "Horsepower", ylab = "log(MPG)")

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 preds 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:

Model Two has a correlation of .82, which is higher than Model One's .76. Model Two has a much smaller MSE than Model One (.03 compared to 25.7).

```
# your code here
pred <- predict(lm2, newdata=test)

print("Summary of predicted values:")
```

```
## [1] "Summary of predicted values:"
```

```
summary(pred)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   2.198   3.012   3.160   3.102   3.269   3.508
```

```
print(paste("Cor between Model Two predicted and test data: ", cor(pred, log(test$mp
g))))
```

```
## [1] "Cor between Model Two predicted and test data:  0.814936032463097"
```

```
print(paste("Cor between Model One predicted and test data: ", cor(p, test$mpg)))
```

```
## [1] "Cor between Model One predicted and test data:  0.764210117020695"
```

```
# calculate MSE
residuals <- pred - log(test$mpg)
mseTwo <- mean(residuals^2)
print(paste("Model Two test data MSE: ", mseTwo))
```

```
## [1] "Model Two test data MSE:  0.0358842425565494"
```

```
print(paste("Model One test data MSE: ", mseTest))
```

```
## [1] "Model One test data MSE:  25.7172652597501"
```

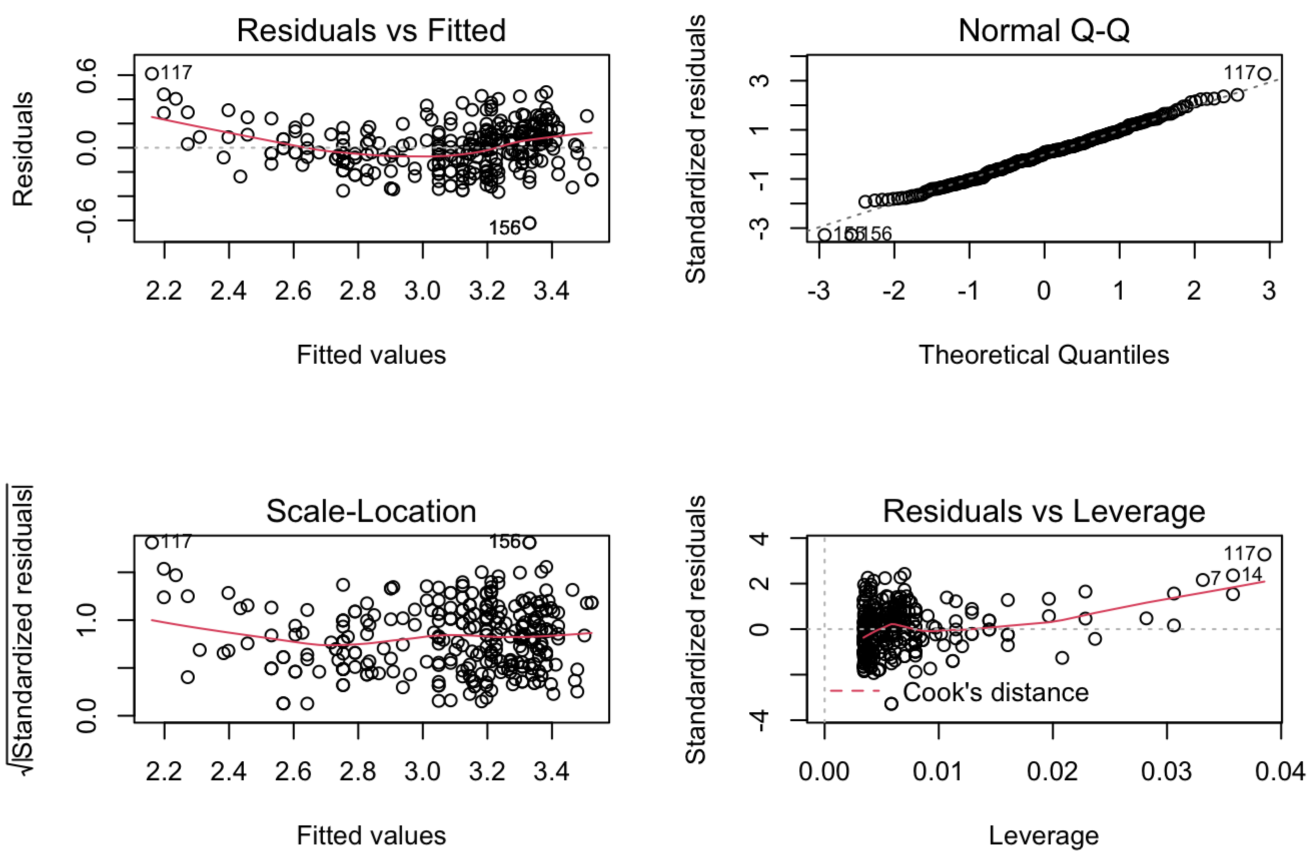
Step 10: Plot the residuals of the second model

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

Your commentary here:

Plot 1: Model Two shows more evidence of linearity than Model One, as the line is more horizontal. Plot 2: Model Two shows more evidence of linearity than Model One, as the standardized residual plots follow the dotted line more closely. Plot 3: The models have similar plots, with both being mostly horizontal. Model One looks slightly more horizontal. This is evidence of linearity. Plot 4: Both models show evidence of possible outliers with unusual x-values. Further investigation is warranted.

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



Problem 2: Multiple Linear Regression

Step 1: Data exploration

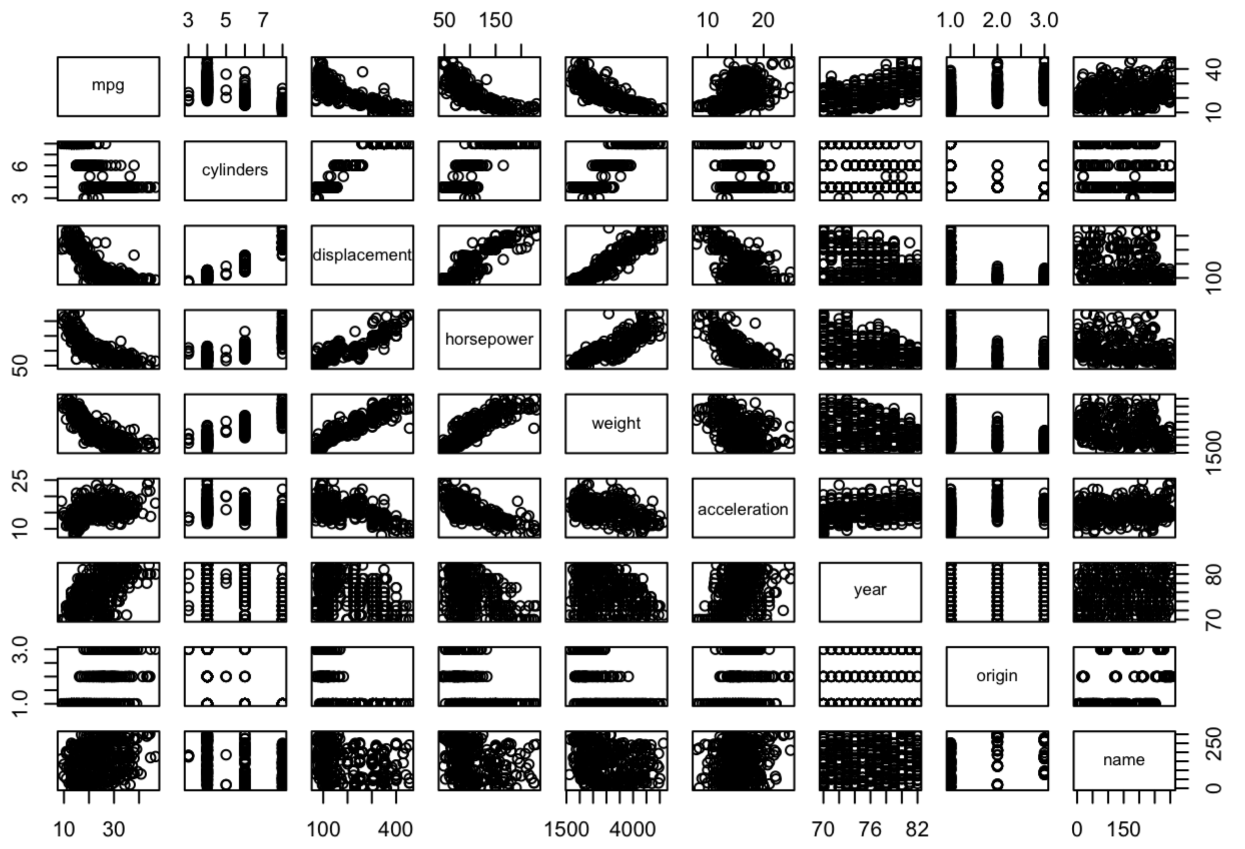
- Produce a scatterplot matrix of correlations which includes all the variables in the data set using the command “pairs(Auto)”
- List any possible correlations that you observe, listing positive and negative correlations separately, with at least 3 in each category.

Your commentary here:

Positive: There appears to be a positive correlation between displacement<->horsepower, displacement<->weight, displacement<->cylinders, horsepower<->weight, cylinders<->weight, and year<->mpg.

Negative: There appears to be a negative correlation between weight<->mpg, horsepower<->acceleration, displacement<->acceleration, and displacement<->mpg.

```
# 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:

Strongest positive: 1. Cylinders and Displacement: .95 2. Weight and Displacement: .933 3. Horsepower and Displacement: .9

Strongest negative: 1. Weight and MPG: -.83 2. Displacement and MPG: -.8 3. Horsepower and MPG: -.8

```
# your code here
Auto$name<-NULL
cor(Auto)
```

```
##                     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:

Year, Weight, and Origin are the predictors that appear to have the most statistically significant relationship to the MPG response.

```
# your code here
print("train$origin changed to a factor:")
```

```
## [1] "train$origin changed to a factor:"
```

```
as.factor(train$origin)
```

```
##   [1] 1 1 1 3 1 1 1 1 1 2 2 3 1 1 1 1 3 1 1 1 1 2 1 1 1 3 1 3 1 1 1 3 2 1 1 2 1
##  [38] 2 1 3 1 3 1 1 1 1 1 1 1 3 1 3 1 2 1 1 2 1 3 1 1 1 1 1 3 3 1 3 2 1 3 1 1 1
##  [75] 1 1 3 1 1 1 1 3 2 3 1 2 1 2 1 1 1 3 2 3 1 1 2 1 1 1 2 1 1 3 1 2 1 1 1 2 2
## [112] 1 1 1 1 2 1 1 3 1 2 3 3 1 3 1 1 3 2 3 1 1 1 3 1 1 1 1 1 3 3 1 2 2 1 1 3 1
## [149] 1 1 1 3 1 2 3 3 2 1 1 1 1 1 1 1 3 1 1 3 1 3 2 1 2 1 1 1 1 3 2 1 2 2 3 3 1
## [186] 1 3 1 3 2 3 3 2 1 1 1 1 2 1 1 2 1 1 1 3 1 1 1 1 2 3 1 3 2 1 3 1 1 3 3 1 1
## [223] 1 1 2 1 1 3 2 3 1 1 2 3 2 1 1 1 2 1 1 1 1 2 2 2 1 1 1 1 1 1 2 2 3 1 1 3 1
## [260] 1 1 1 3 1 3 2 2 1 1 1 3 1 1 1 1 1 1 1 1 1 2 1 1 1 3 1 1 2 3 3 1 3 2
## Levels: 1 2 3
```

```
lm3 <- lm(mpg ~ . - name, data=train)

print("Summary of Model Three:")
```

```
## [1] "Summary of Model Three:"
```

```
summary(lm3)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4220 -1.9922 -0.0728  1.7306 13.3193
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2.003e+01  5.219e+00  -3.837 0.000153 ***
## cylinders    -7.588e-01  3.570e-01  -2.126 0.034390 *
## displacement  2.168e-02  8.156e-03   2.658 0.008311 **
## horsepower   -2.588e-02  1.475e-02  -1.754 0.080469 .
## weight       -5.752e-03  7.157e-04  -8.037 2.44e-14 ***
## acceleration -2.579e-02  1.089e-01  -0.237 0.813025
## year          8.081e-01  5.808e-02  13.912  < 2e-16 ***
## origin        1.370e+00  3.111e-01   4.404 1.50e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.225 on 286 degrees of freedom
## Multiple R-squared:  0.8329, Adjusted R-squared:  0.8288
## F-statistic: 203.7 on 7 and 286 DF,  p-value: < 2.2e-16
```

Step 4: Plot the residuals of the third model

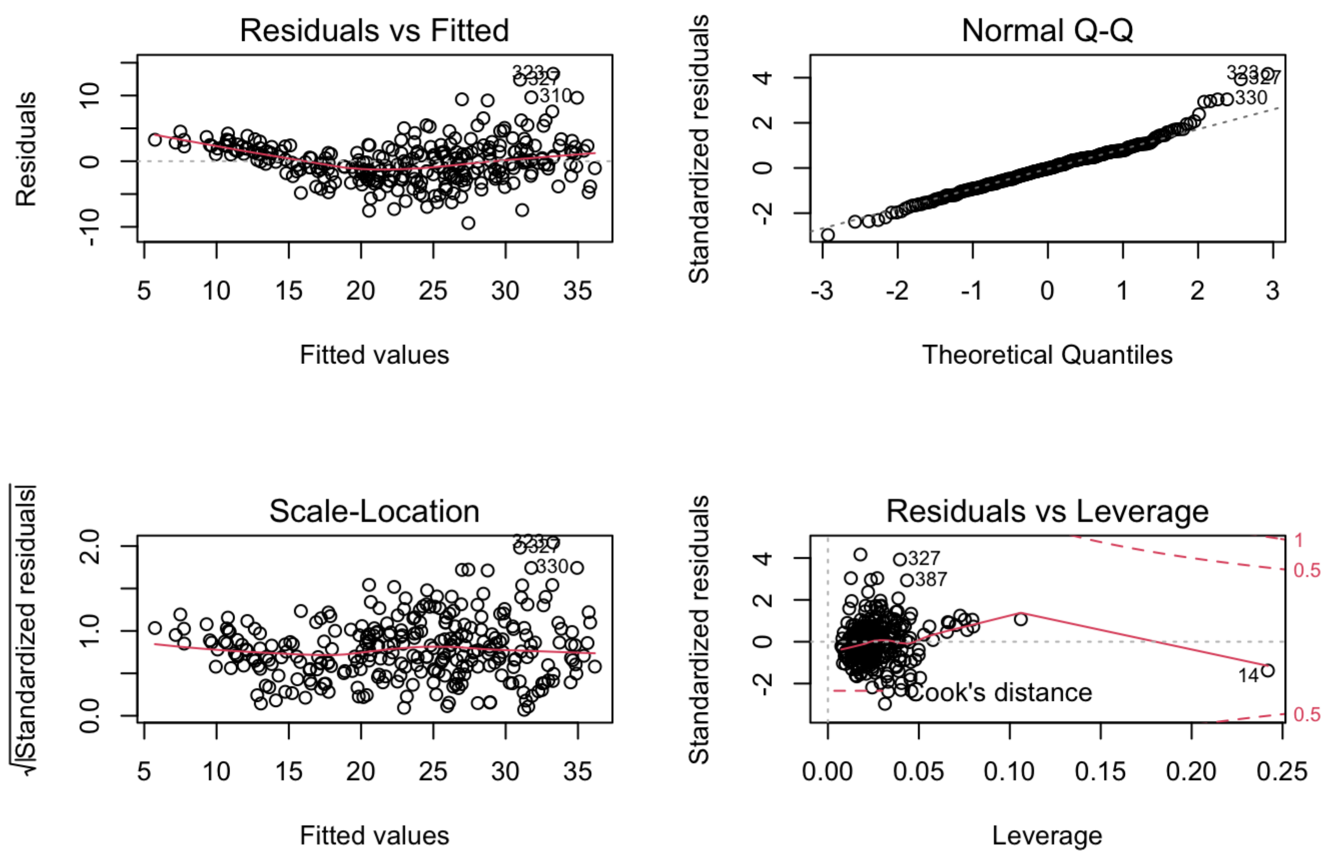
- Use the plot() function to produce diagnostic plots of the linear regression fit
- Comment on any problems you see with the fit
- Are there any leverage points?
- Display a row from the data set that seems to be a leverage point.

Your commentary here:

The residual plot has a strong curve, indicating issues with the fit. Plot 2 indicates possible issues with fit in theoretical quantile 2.

Observations 14 has high leverage.

```
# your code here
par(mfrow=c(2,2))
plot(lm3)
```



```
print(train[14,]) # display Observation 14
```

```
##     mpg cylinders displacement horsepower weight acceleration year origin
## 186  26         4           98         79   2255         17.7   76      1
##            name
## 186 dodge colt
```

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: Model Four has an R^2 of .92, better than the Model Three's R^2 of .83.

Running anova() shows that the RSS is lower for Model Four, and Model Four is given a low p-value. This is confirmation that Model Four outperformed Model Three.

```
# your code here
lm4 <- lm(mpg ~ year * weight * origin * displacement * cylinders * horsepower, data=tra
in)

print("Summary of Linear Model Four:")
```

```
## [1] "Summary of Linear Model Four:"
```

```
summary(lm4)
```

```
##
## Call:
## lm(formula = mpg ~ year * weight * origin * displacement * cylinders *
##     horsepower, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5351 -1.3110 -0.0517  1.1798 10.6512
##
## Coefficients: (2 not defined because of singularities)
##                                                         Estimate Std. Error
## (Intercept)                                            1.095e+04  3.748e+04
## year                                                  -1.355e+02  4.786e+02
## weight                                                -5.522e-01  8.510e+00
## origin                                                -1.186e+04  3.731e+04
## displacement                                           1.738e+02  3.878e+02
## cylinders                                             -2.834e+03  9.320e+03
## horsepower                                            -3.646e+02  1.000e+03
## year:weight                                            1.079e-02  1.089e-01
## year:origin                                            1.471e+02  4.763e+02
## weight:origin                                          1.119e-01  8.398e+00
## year:displacement                                     -2.333e+00  5.076e+00
## weight:displacement                                   -7.577e-02  1.800e-01
## origin:displacement                                   -1.487e+02  3.870e+02
## year:cylinders                                         3.537e+01  1.190e+02
## weight:cylinders                                       7.492e-02  2.115e+00
## origin:cylinders                                       2.905e+03  9.300e+03
## displacement:cylinders                                -4.199e+01  9.680e+01
## year:horsepower                                        4.618e+00  1.272e+01
## weight:horsepower                                      1.064e-01  2.875e-01
## origin:horsepower                                      3.701e+02  9.995e+02
## displacement:horsepower                                2.016e-01  1.841e+00
## cylinders:horsepower                                   9.142e+01  2.495e+02
## year:weight:origin                                    -4.025e-03  1.075e-01
## year:weight:displacement                               9.744e-04  2.289e-03
## year:origin:displacement                               1.986e+00  5.067e+00
## weight:origin:displacement                             7.162e-02  1.798e-01
## year:weight:cylinders                                 -1.844e-03  2.703e-02
## year:origin:cylinders                                 -3.615e+01  1.187e+02
## weight:origin:cylinders                                5.714e-03  2.104e+00
## year:displacement:cylinders                            5.602e-01  1.267e+00
## weight:displacement:cylinders                          1.904e-02  4.503e-02
## origin:displacement:cylinders                          3.839e+01  9.679e+01
## year:weight:horsepower                                -1.393e-03  3.722e-03
## year:origin:horsepower                                -4.678e+00  1.271e+01
## weight:origin:horsepower                              -9.993e-02  2.871e-01
## year:displacement:horsepower                          -1.512e-03  2.260e-02
## weight:displacement:horsepower                         1.667e-05  1.033e-04
## origin:displacement:horsepower                        -4.975e-01  1.827e+00
## year:cylinders:horsepower                             -1.160e+00  3.172e+00
## weight:cylinders:horsepower                           -2.523e-02  7.165e-02
## origin:cylinders:horsepower                           -9.095e+01  2.494e+02
## displacement:cylinders:horsepower                     -7.137e-02  4.515e-01
## year:weight:origin:displacement                         -9.187e-04  2.287e-03
## year:weight:origin:cylinders                             6.265e-04  2.689e-02
## year:weight:displacement:cylinders                       -2.439e-04  5.727e-04
## year:origin:displacement:cylinders                       -5.104e-01  1.267e+00
## weight:origin:displacement:cylinders                     -1.846e-02  4.502e-02
## year:weight:origin:horsepower                             1.296e-03  3.716e-03
## year:weight:displacement:horsepower                      -2.815e-07  1.354e-06
## year:origin:displacement:horsepower                       5.583e-03  2.240e-02
## weight:origin:displacement:horsepower                     3.344e-05  6.584e-05
## year:weight:cylinders:horsepower                          3.302e-04  9.271e-04
## year:origin:cylinders:horsepower                          1.151e+00  3.171e+00
## weight:origin:cylinders:horsepower                        2.403e-02  7.161e-02
## year:displacement:cylinders:horsepower                    7.028e-04  5.529e-03
## weight:displacement:cylinders:horsepower                 -6.058e-06  8.478e-06
## origin:displacement:cylinders:horsepower                  1.090e-01  4.515e-01
## year:weight:origin:displacement:cylinders                 2.361e-04  5.726e-04
## year:weight:origin:displacement:horsepower               -3.906e-07  8.556e-07
## year:weight:origin:cylinders:horsepower                  -3.126e-04  9.266e-04
## year:weight:displacement:cylinders:horsepower             8.119e-08  1.115e-07
## year:origin:displacement:cylinders:horsepower            -1.220e-03  5.529e-03
## weight:origin:displacement:cylinders:horsepower                  NA          NA
## year:weight:origin:displacement:cylinders:horsepower             NA          NA
##                                                          t value Pr(>|t|)
## (Intercept)                                                0.292    0.770
## year                                                      -0.283    0.777
## weight                                                    -0.065    0.948
## origin                                                    -0.318    0.751
## displacement                                               0.448    0.655
## cylinders                                                 -0.304    0.761
## horsepower                                                -0.365    0.716
## year:weight                                                0.099    0.921
## year:origin                                                0.309    0.758
## weight:origin                                              0.013    0.989
## year:displacement                                         -0.460    0.646
## weight:displacement                                       -0.421    0.674
## origin:displacement                                       -0.384    0.701
## year:cylinders                                             0.297    0.766
## weight:cylinders                                           0.035    0.972
## origin:cylinders                                           0.312    0.755
## displacement:cylinders                                    -0.434    0.665
## year:horsepower                                            0.363    0.717
## weight:horsepower                                          0.370    0.712
## origin:horsepower                                          0.370    0.711
## displacement:horsepower                                    0.110    0.913
## cylinders:horsepower                                       0.366    0.714
## year:weight:origin                                        -0.037    0.970
## year:weight:displacement                                   0.426    0.671
## year:origin:displacement                                   0.392    0.695
## weight:origin:displacement                                 0.398    0.691
## year:weight:cylinders                                     -0.068    0.946
## year:origin:cylinders                                     -0.305    0.761
## weight:origin:cylinders                                    0.003    0.998
## year:displacement:cylinders                                0.442    0.659
## weight:displacement:cylinders                              0.423    0.673
## origin:displacement:cylinders                              0.397    0.692
## year:weight:horsepower                                     -0.374    0.709
## year:origin:horsepower                                     -0.368    0.713
## weight:origin:horsepower                                   -0.348    0.728
## year:displacement:horsepower                               -0.067    0.947
## weight:displacement:horsepower                              0.161    0.872
## origin:displacement:horsepower                             -0.272    0.786
## year:cylinders:horsepower                                  -0.366    0.715
## weight:cylinders:horsepower                                -0.352    0.725
## origin:cylinders:horsepower                                -0.365    0.716
## displacement:cylinders:horsepower                          -0.158    0.875
## year:weight:origin:displacement                            -0.402    0.688
## year:weight:origin:cylinders                                0.023    0.981
## year:weight:displacement:cylinders                         -0.426    0.671
## year:origin:displacement:cylinders                         -0.403    0.687
## weight:origin:displacement:cylinders                       -0.410    0.682
## year:weight:origin:horsepower                               0.349    0.728
## year:weight:displacement:horsepower                        -0.208    0.835
## year:origin:displacement:horsepower                         0.249    0.803
## weight:origin:displacement:horsepower                       0.508    0.612
## year:weight:cylinders:horsepower                            0.356    0.722
## year:origin:cylinders:horsepower                            0.363    0.717
## weight:origin:cylinders:horsepower                          0.336    0.737
## year:displacement:cylinders:horsepower                      0.127    0.899
## weight:displacement:cylinders:horsepower                   -0.715    0.476
## origin:displacement:cylinders:horsepower                    0.241    0.810
## year:weight:origin:displacement:cylinders                   0.412    0.680
## year:weight:origin:displacement:horsepower                 -0.456    0.648
## year:weight:origin:cylinders:horsepower                    -0.337    0.736
## year:weight:displacement:cylinders:horsepower               0.728    0.467
## year:origin:displacement:cylinders:horsepower              -0.221    0.825
## weight:origin:displacement:cylinders:horsepower                NA       NA
## year:weight:origin:displacement:cylinders:horsepower           NA       NA
##
## Residual standard error: 2.459 on 232 degrees of freedom
## Multiple R-squared:  0.9212, Adjusted R-squared:  0.9005
## F-statistic: 44.47 on 61 and 232 DF,  p-value: < 2.2e-16
```

```
print("Comparison of Model Three and Model Four:")
```

```
## [1] "Comparison of Model Three and Model Four:"
```

```
anova(lm3, lm4)
```

	Res.Df	RSS	Df	Sum of Sq	F	Pr(>F)
	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	286	2973.694	NA	NA	NA	NA
2	232	1402.288	54	1571.406	4.814435	2.078453e-17

2 rows

```
#pred <- predict(lm4, newdata=test)

#print(paste("Cor between Model Four predicted and test data: ", cor(pred, test$mpg)))

# calculate MSE
#residuals <- pred - test$mpg
#mseFour <- mean(residuals^2)
#print(paste("MSE: ", mseFour))
#print(paste("RMSE: ", sqrt(mseFour)))
```