Homework 2

4375 Machine Learning with Dr. Mazidi

Anthony Martinez | amm180005

9/8/21

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

# load library ISLR
if (!require("ISLR")){
  install.packages("ISLR")
}
```

```
## Loading required package: ISLR
```

```
library(ISLR)

# Use names() and summary() to learn more about the Auto data set
data(Auto)
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
```

```
# Divide the data into 75% train, 25% test, using seed 1234
set.seed(1234)
i <- sample(1:nrow(Auto), .75*nrow(Auto), replace=FALSE)
train <- Auto[i,]
test <- Auto[-i,]
```

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

# use lm to preform simple linear regression on the train data with mpg as respone and horsepowe
r as the predictor
lm1 <- lm(mpg~horsepower, data=train)

#use summary function to evaluate the model
summary(lm1)
```

```
##
## 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
```

```
# calculate the MSE by extracting the residuals from the model like this: mse <- mean(lm1$residu
als^2)
mse <- mean(lm1$residuals^2)
print(paste('mse: ',mse))
```

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

```
#calculate and print RMSE by taking the sqrt of mse
rmse <- sqrt(mse)
print(paste('rmse: ', rmse))
```

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

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

1. Write the equation for the model, y = wx + b, filling in the parameters w, b and variable names x, y

- mpg = -0.1567*horsepower + 39.6486

2. Is there a strong relationship between horsepower and mpg?

- There is a strong negative relationship between horspower and mpg at -.78

3. Is it a positive or negative correlation?

- negative

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

- R^2: The closer the R^2 is to 1, the more variance in the model is explained by the predictors. The R^2 in this case is .61 which is not bad but not the best. This means the variance in the model is somewhat explained by the predictors.

- RSE: The RSE tells us how far off the model was from the data aka the lack of fit of the model. It is measured in terms of y. In this case it is 4.853 which means the average error of the model was about 5.
- F-statistic: While R^2 does not tell us if it is statistically significant but the F-statisitic does. A F-statistic greater than 1 and a low p-value indicates confidence in the model. This model has a F-statistic of 463.7 and a small p value of 2.2e-16 so we have good confidence in the model.

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

- The RMSE is 4.83 This means that the model was off by 4.8 which is not too bad.

Step 4: Examine the model graphically

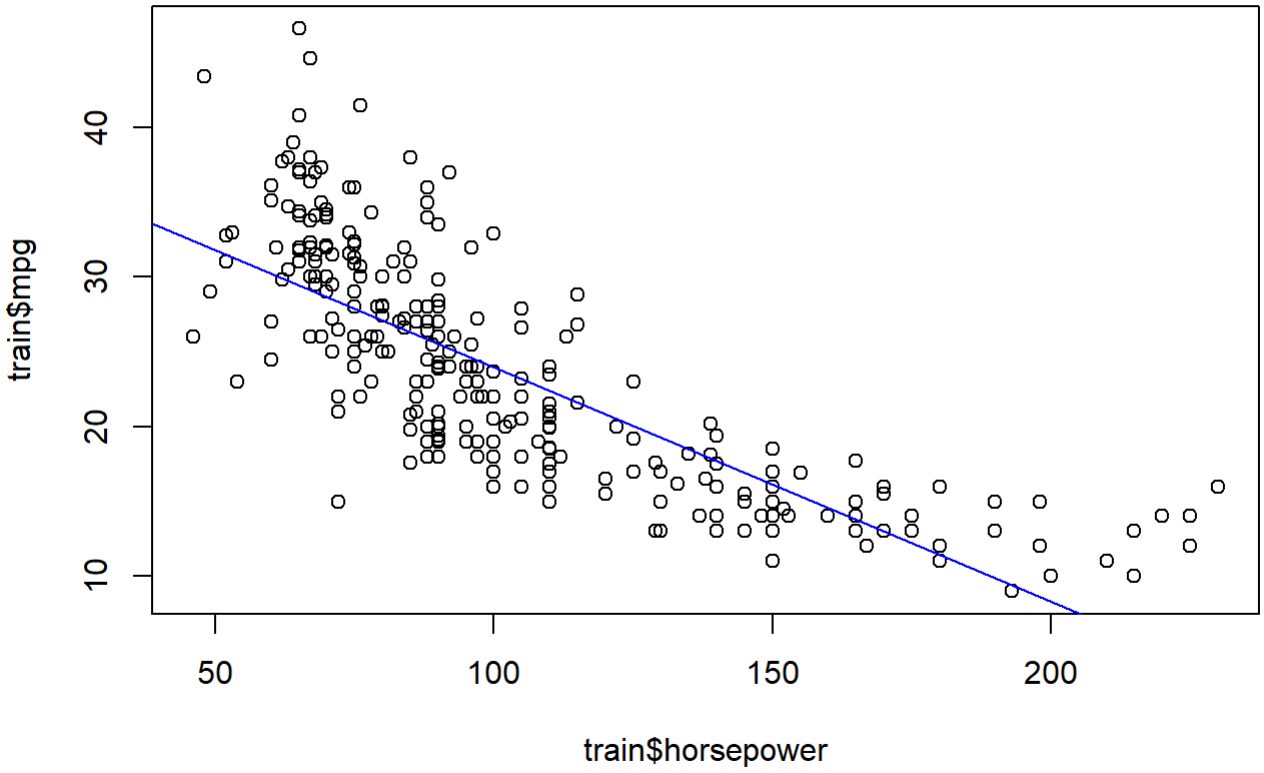
- Plot train$mpg~train$horsepower
- Draw a blue abline()
- Comment on how well the data fits the line
- Predict mpg for horsepower of 98. Hint: See the Quick Reference 5.10.3 on page 96
- Comment on the predicted value given the graph you created

Your commentary here: The blue abline fits the data pretty well. It found a line somewhat in the center of all the points. The pred value is 24.29 which is close to the mse value of 23.39. Which is a good indication of the model's confidence.

```
# your code here

#Plot train\$mpg~train\$horsepower
plot(train$mpg~train$horsepower)

#Draw a blue abline()
abline(lm(train$mpg~train$horsepower), col="blue")
```




```
pred <- predict(lm1, data.frame(horsepower=98))
pred
```

```
##        1
## 24.29381
```

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: The mse of the training data is 23.39 while the mse for the test data was 25.717. This means that the model was able to learn well from the training data. The correlation on the test data was .76 which is high.

```
# your code here
pred1 <- predict(lm1, newdata=test)
correlation <- cor(pred1,test$mpg)
print(paste("correlation: ", correlation))
```

```
## [1] "correlation:  0.764210117020695"
```

```
mse1 <- mean((pred1-test$mpg)^2)
print(paste("mse: ",  mse1))
```

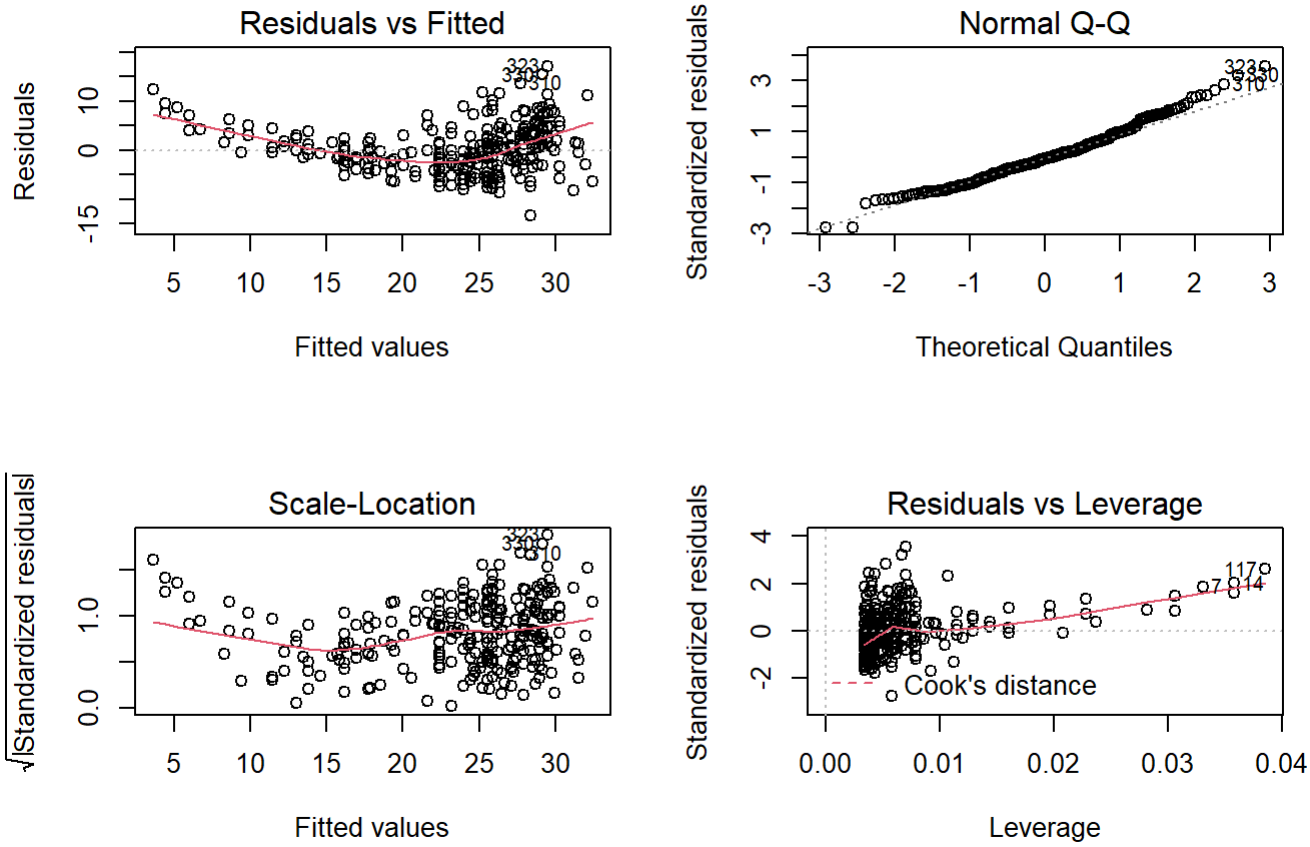
```
## [1] "mse:  25.7172652597501"
```

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: From the residuals vs fitted graph we want to see a horizontal line. As you can see the red line is not very horizontal so this could be an indication of non-linearity From the Normal Q-Q graph we want to see the dotted line follow the graph, which it does. From the scale-location graph we want to see a horizontal line, the red line is fairly 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: The R^2 for the second model is slightly higher than the R^2 of the first model. .69 vs .61

```
# your code here

# Create a second linear model with log(mpg) predicted by horsepower
lm2 <- lm(log(mpg)~horsepower, data=train)
summary(lm2)
```

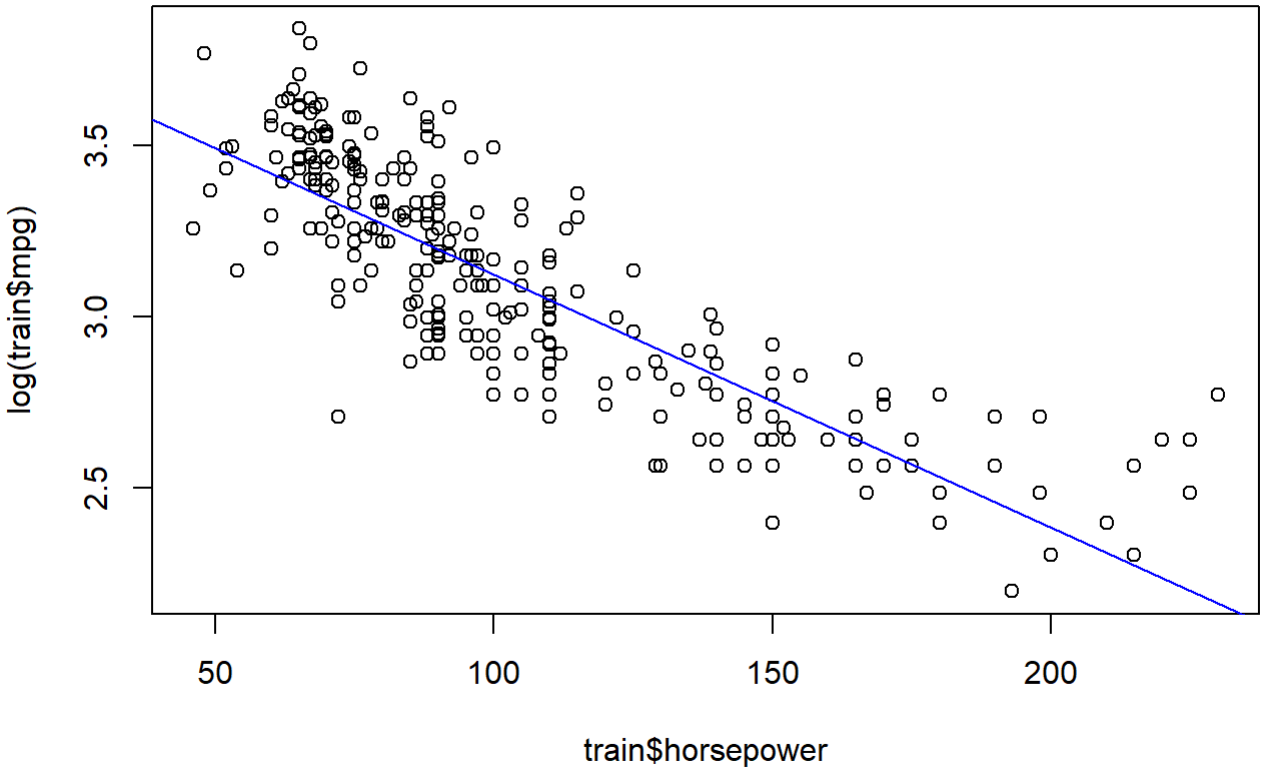
```
##
## Call:
## lm(formula = log(mpg) ~ horsepower, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.62229 -0.12814  0.01443  0.12330  0.61150
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.8631645  0.0319324  120.98   <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: The line fits the data considerly better than the first model. Overall, the data points are closer to the line.

```
# your code here
plot(log(train$mpg)~train$horsepower)
abline(lm(log(train$mpg)~train$horsepower),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 1
- 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: The correlation for this model is higher than model 1. .81 vs .76. The mse

```
# your code here
pred2 <- predict(lm2, newdata=test)
correlation2 <- cor(pred2,log(test$mpg))
print(paste("correlation: ", correlation2))
```

```
## [1] "correlation:  0.814936032463097"
```

```
residuals <- pred2-log(test$mpg)
mse2 <- mean(residuals^2)
print(paste("mse: ",  mse2))
```

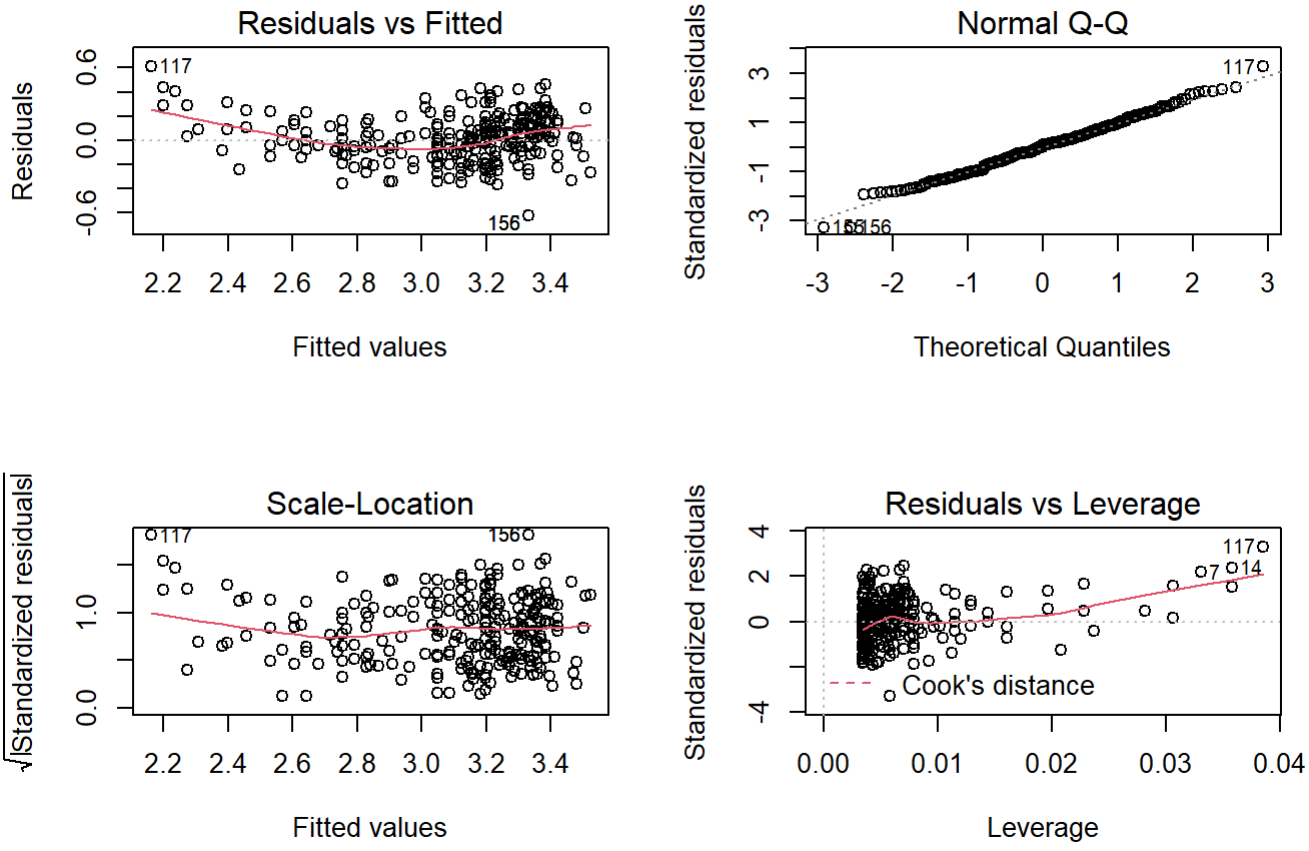
```
## [1] "mse:  0.0358842425565494"
```

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 Residuals vs Fitted graph for model 2 has more of a horizontal line which is what we want. The dotted line in the Normal Q-Q graph goes through more of the points which is what we want. The scare-location graph has more of a horizontal line as well. This means that Module 2 is better.

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



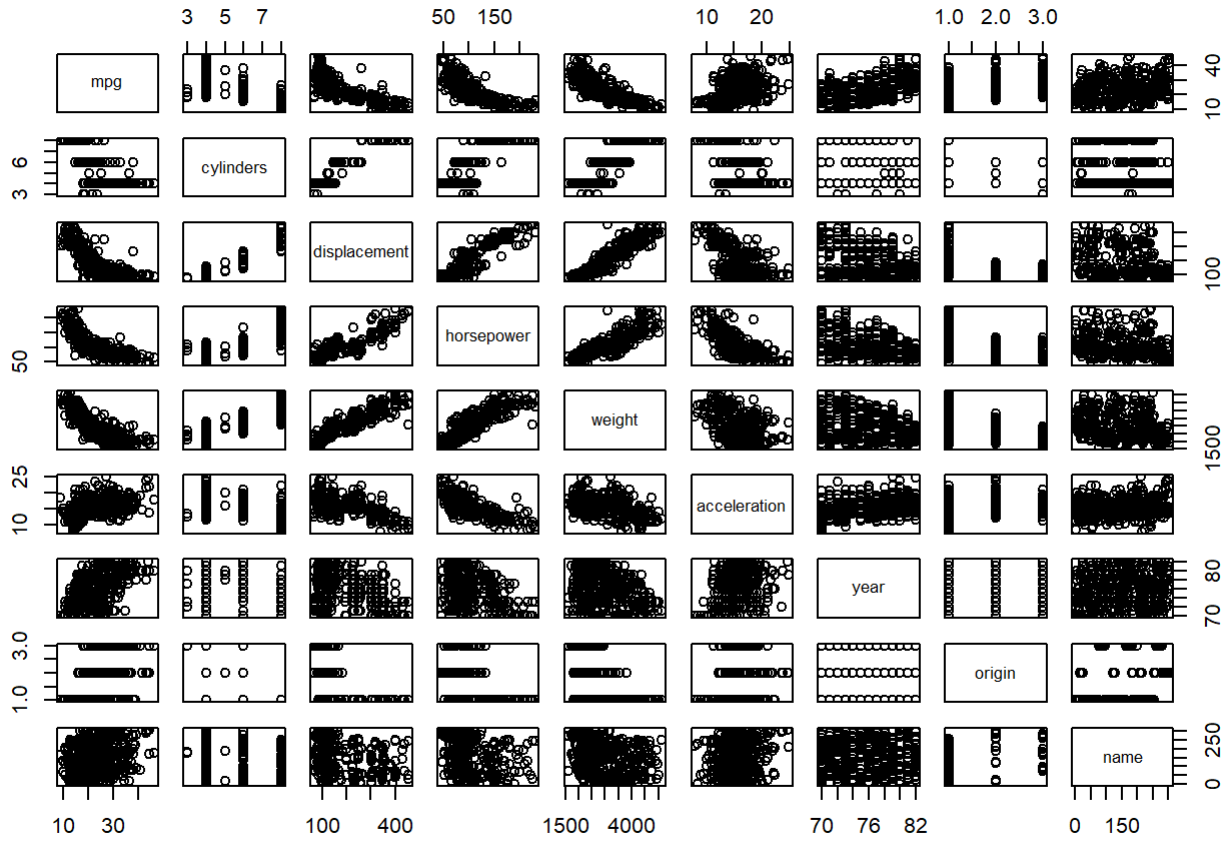
Problem 2: Multiple Linear Regression

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: Negative Correlations: mpg-displacement, mpg-horsepower, mpg-weight Positive Correlations: displacement-horsepower, displacement-weight, horsepower-weight

```
# 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: mgp-mpg: 1.0 , displacement-displacement: 1.0 Strongest Negative: weight-mpg: -0.8422442, displacement-mpg: -0.8051269

```
# your code here

cor(Auto[1:8])
```

```
##                     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: Cylinders, Displacement, Weight, Year, origin2, origin3 all appear to be statistically significant to the response.

```
# your code here
Auto$origin <- as.factor(Auto$origin)

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

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0095 -2.0785 -0.0982  1.9856 13.3608
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.795e+01  4.677e+00  -3.839 0.000145 ***
## cylinders    -4.897e-01  3.212e-01  -1.524 0.128215
## displacement  2.398e-02  7.653e-03   3.133 0.001863 **
## horsepower   -1.818e-02  1.371e-02  -1.326 0.185488
## weight       -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## acceleration  7.910e-02  9.822e-02   0.805 0.421101
## year          7.770e-01  5.178e-02  15.005  < 2e-16 ***
## origin2       2.630e+00  5.664e-01   4.643 4.72e-06 ***
## origin3       2.853e+00  5.527e-01   5.162 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.307 on 383 degrees of freedom
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

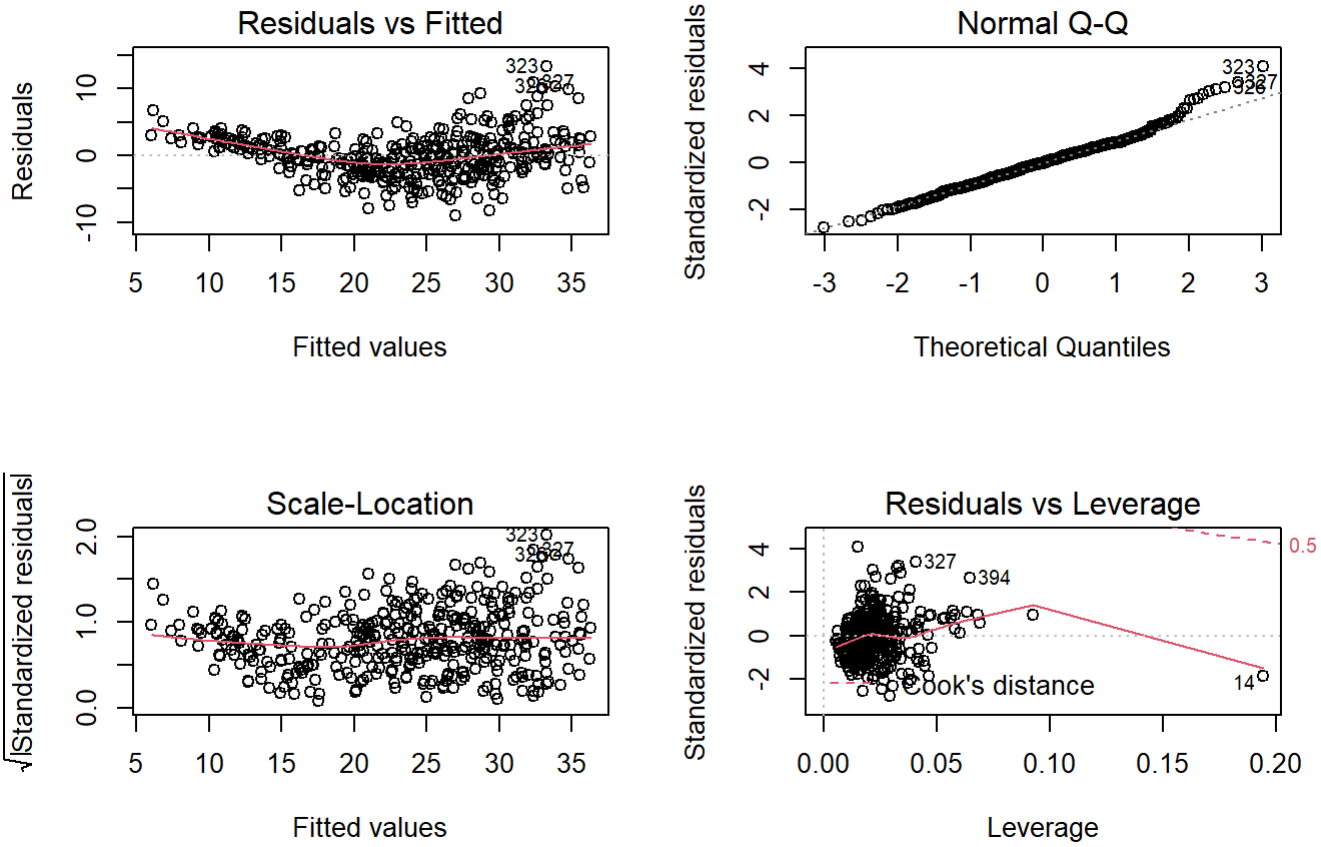
Step 4: Plot the residuals of the third model

- 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: In the Normal Q-Q graph, the dotted line misses all of the points at the theoretical quanitles 2-3 and standardized residuals 2-4. There appears to be a leverage point at Leverage = .20.

```
# your code here

#Use the plot() function to produce diagnostic plots of the linear regression fit
par(mfrow=c(2,2))
plot(lm3)
```




```
# Display a row from the data set that seems to be a leverage point.
Auto[14,]
```

…	cylinders	displacement	horsepower	wei…	acceleration	y…	origin	name
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<fctr>	<fctr>
14 14	8	455	225	3086	10	70	1	buick estate wagon

1 row

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: The R^2 of lm3 is .82 while it is .885 for lm4. A value closer to 1 is more desirable so in this case lm4 is better. lm4 lowered the error, the RSS, and had a low p-value which indicates that lm4 is a better model than lm3.

```
# your code here

# Use the * and + symbols to fit linear regression models with interaction effects, choosing wha
tever variables you think might get better results than your model in step 3 above
lm4 <- lm(mpg~horsepower*weight*year*acceleration, data=Auto)
summary(lm4)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower * weight * year * acceleration,
##     data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.0833 -1.4994  0.0643  1.3287 11.8546
##
## Coefficients:
##                                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)                         6.732e+01  1.855e+02   0.363 0.716888
## horsepower                         -6.957e+00  1.847e+00  -3.766 0.000193 ***
## weight                              8.757e-02  6.921e-02   1.265 0.206532
## year                               -3.770e-01  2.474e+00  -0.152 0.878959
## acceleration                       -1.732e+01  1.109e+01  -1.562 0.119109
## horsepower:weight                   1.211e-03  5.243e-04   2.310 0.021435 *
## horsepower:year                     9.404e-02  2.516e-02   3.737 0.000215 ***
## weight:year                        -1.229e-03  9.258e-04  -1.327 0.185221
## horsepower:acceleration             6.092e-01  1.299e-01   4.690 3.82e-06 ***
## weight:acceleration                -3.690e-03  4.155e-03  -0.888 0.375058
## year:acceleration                   2.493e-01  1.483e-01   1.681 0.093618 .
## horsepower:weight:year             -1.657e-05  7.159e-06  -2.315 0.021143 *
## horsepower:weight:acceleration     -1.130e-04  3.595e-05  -3.143 0.001806 **
## horsepower:year:acceleration       -8.390e-03  1.761e-03  -4.766 2.69e-06 ***
## weight:year:acceleration            4.312e-05  5.550e-05   0.777 0.437693
## horsepower:weight:year:acceleration  1.583e-06  4.915e-07   3.221 0.001389 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.691 on 376 degrees of freedom
## Multiple R-squared:  0.8857, Adjusted R-squared:  0.8811
## F-statistic: 194.2 on 15 and 376 DF,  p-value: < 2.2e-16
```

```
summary(lm3)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.0095 -2.0785 -0.0982  1.9856 13.3608
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.795e+01  4.677e+00  -3.839 0.000145 ***
## cylinders    -4.897e-01  3.212e-01  -1.524 0.128215
## displacement  2.398e-02  7.653e-03   3.133 0.001863 **
## horsepower   -1.818e-02  1.371e-02  -1.326 0.185488
## weight       -6.710e-03  6.551e-04 -10.243  < 2e-16 ***
## acceleration  7.910e-02  9.822e-02   0.805 0.421101
## year          7.770e-01  5.178e-02  15.005  < 2e-16 ***
## origin2       2.630e+00  5.664e-01   4.643 4.72e-06 ***
## origin3       2.853e+00  5.527e-01   5.162 3.93e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.307 on 383 degrees of freedom
## Multiple R-squared:  0.8242, Adjusted R-squared:  0.8205
## F-statistic: 224.5 on 8 and 383 DF,  p-value: < 2.2e-16
```

```
# Run anova() on the two models to see if your second model outperformed the previous one
anova(lm3,lm4)
```

	Res.Df <dbl>	RSS <dbl>	Df <dbl>	Sum of Sq <dbl>	F <dbl>	Pr(>F) <dbl>
1	383	4187.392	*NA*	*NA*	*NA*	*NA*
2	376	2722.493	7	1464.899	28.90219	7.808727e-32

2 rows