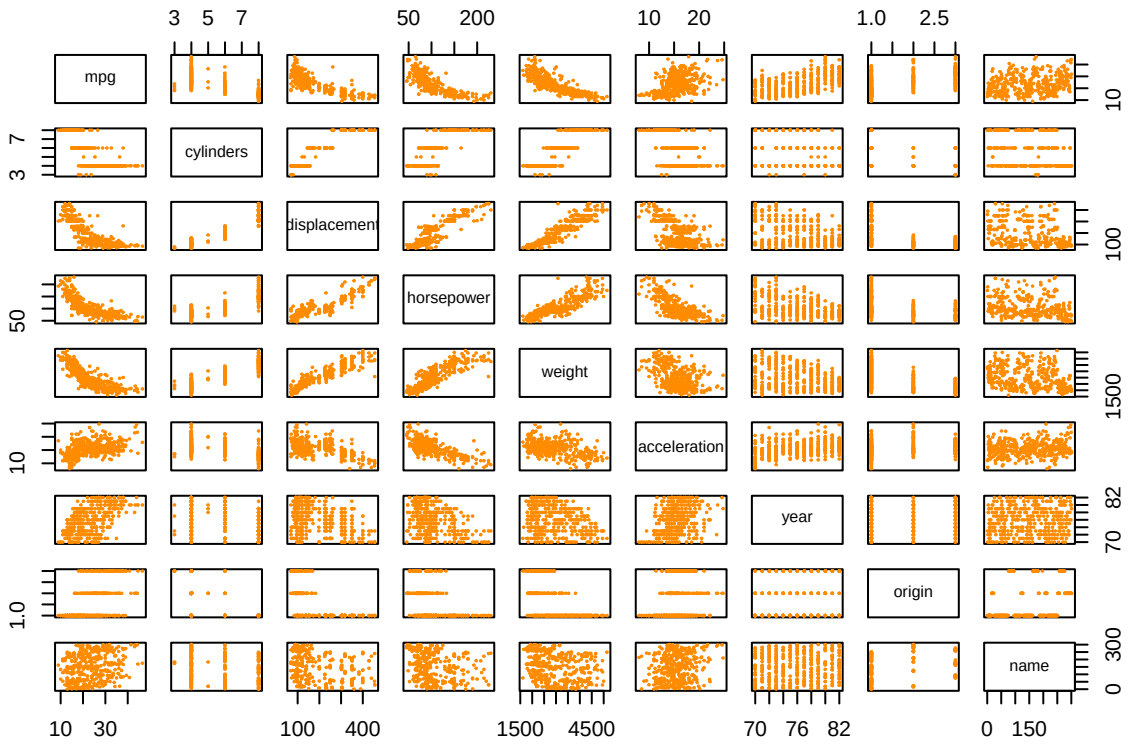
Auto MPG Linear Regression

2024-09-27

1 - Produce a scatterplot matrix that includes all of the variables in the dataset

```
plot(auto_df, pch=20, cex=0.25, col='darkorange')
```



2 - Compute the matrix of correlations using cor()

```
#exclude the name column and omit NAs
auto_df_corr <- subset(auto_df, select = -name)
auto_df_corr <- na.omit(auto_df_corr)
cor(auto_df_corr)
```

```
##                     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
```

3 - Use the lm() function to perform multiple linear regression with mpg as the response and all other variables except name as the predictors.

```r
#create lm model
mpg_lm <- lm(mpg ~ cylinders + displacement + horsepower + weight + acceleration + year +
           origin, data=auto_df)

#print summary
summary(mpg_lm)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year + origin, data = auto_df)
##
## 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
```

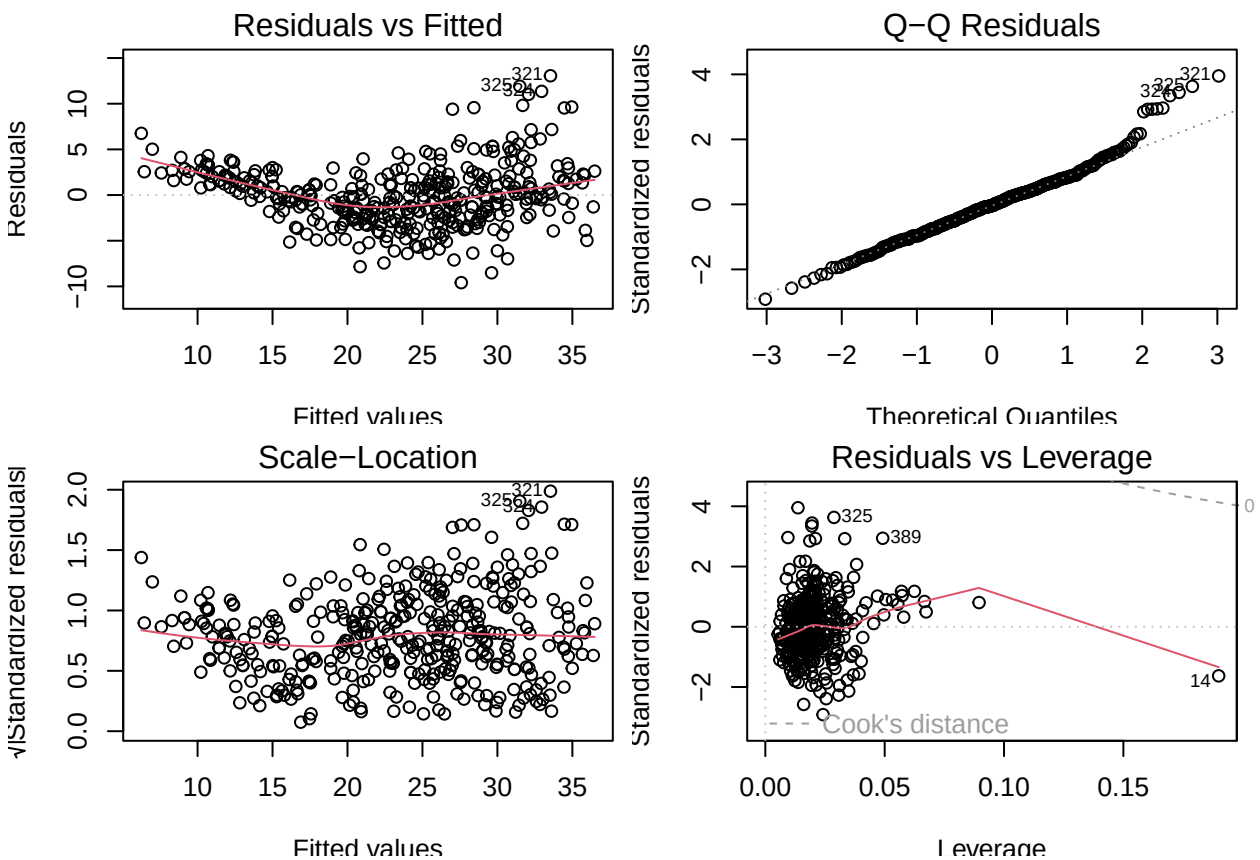
Model Summary

Overall, there are only a few variables with notable relationships to the response variable. Cylinders, for example, has a negative relationship to the mpg response variable, but the p value indicates that it is not statistically significant. Displacement's relationship is positive and statistically significant, but the coefficient is low. Weight, year, and origin are the other statistically significant variables with low p-values. Year and Origin specifically have higher coefficients, which means these have a stronger influence the response variable.

The coefficient for the year variable suggests that, as time goes on and increases, so does mpg. This could be due to enhancements in car technology, fuel efficiency, and a focus on getting more out of a gallon of fuel.

4 - Produce diagnostic plots of the linear regression fit.

```r
#format and plot the diagnostics
par(mfrow=c(2,2))
par(mai = c(0.6, 0.6, 0.3, 0.1))
par(cex = 0.8)
plot(mpg_lm)
```



Diagnostic Summary

For the residuals vs fitted plot, there is a slight 'U' curve with the dip happening around the middle of the plot. Many of the residuals are not close to 0. This could indicate that the linear model does not capture all of the non-linear relationships in the data.

The Q-Q residuals chart suggests that, for most of the residuals, there is a standard distribution. The outliers towards the top of the diagonal line could mean there are some large outliers not fully captured in the model.

The Scale-Location plot is very scattered and the line on the chart is curved. There is a larger spread of residuals, and variance is not constant across the model.

Lastly, the Residuals vs. Leverage plot shows a large cluster of values towards the left of the chart, indicating most points have low leverage and residuals relatively near 0. The one outlier point on the right of the chart shows there is one point that has strong influence on the model results.

5 - Use * and : to fit linear regression models with interaction effects.

```
#create the model
mpg_lm_interaction <- lm(mpg ~ cylinders * displacement * horsepower * weight * acceleration *
                           year * origin, data = auto_df)
```

```r
#output is large, so collect coefficients and filter for statistical significance
mpg_interaction_summary <- summary(mpg_lm_interaction)
coefficients <- mpg_interaction_summary$coefficients
significant_terms <- coefficients[coefficients[, 4] < 0.05, ]
print(significant_terms)
```

```
##      Estimate Std. Error t value Pr(>|t|)
```

There are no significant interactions from this model. The variables and the response likely don't depend on each other at the same time.

Cylinders

```r
fit_cylinder_only <- lm(mpg ~ cylinders:displacement + cylinders:horsepower +
                            cylinders:weight + cylinders:acceleration +
                            cylinders:year + cylinders:origin, data = auto_df)
summary(fit_cylinder_only)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders:displacement + cylinders:horsepower +
##     cylinders:weight + cylinders:acceleration + cylinders:year +
##     cylinders:origin, data = auto_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.1070  -3.0311  -0.3255   2.3193  15.7533
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)              26.6418609  1.6770276  15.886  < 2e-16 ***
## cylinders:displacement    0.0013531  0.0013683   0.989  0.32336
## cylinders:horsepower     -0.0080651  0.0026364  -3.059  0.00238 **
## cylinders:weight         -0.0007679  0.0001308  -5.869 9.48e-09 ***
## cylinders:acceleration -0.0623903  0.0222878  -2.799  0.00538 **
## cylinders:year            0.0385670  0.0072550   5.316 1.80e-07 ***
## cylinders:origin          0.3702274  0.0807930   4.582 6.22e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.296 on 385 degrees of freedom
## Multiple R-squared:  0.7017, Adjusted R-squared:  0.6971
## F-statistic:   151 on 6 and 385 DF,  p-value: < 2.2e-16
```

Taking the interactions between cylinders and other variables, we can see that there are some statistically significant interactions here. Cylinders and origin are the variables that have the highest coefficient, meaning the origin of a vehicle has a larger positive effect on vehicles with more cylinders.

Horsepower

```
fit_horsepower_only <- lm(mpg ~ horsepower:displacement +  horsepower:weight +
                            horsepower:acceleration + horsepower:year +
                            horsepower:origin, data = auto_df)
summary(fit_horsepower_only)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower:displacement + horsepower:weight +
##     horsepower:acceleration + horsepower:year + horsepower:origin,
##     data = auto_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.1452  -2.9521  -0.4776   2.6225  16.2613
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                3.907e+01  1.766e+00  22.121  < 2e-16 ***
## horsepower:displacement    6.275e-06  5.549e-05   0.113  0.91002
## horsepower:weight         -1.786e-05  6.992e-06  -2.555  0.01101 *
## horsepower:acceleration -8.314e-03  1.366e-03  -6.088 2.76e-09 ***
## horsepower:year            1.512e-04  3.574e-04   0.423  0.67251
## horsepower:origin          1.317e-02  3.983e-03   3.307  0.00103 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.453 on 386 degrees of freedom
## Multiple R-squared:  0.6786, Adjusted R-squared:  0.6745
## F-statistic:   163 on 5 and 386 DF,  p-value: < 2.2e-16
```

One interaction that is statistically significant here is horsepower:acceleration. This may suggest that, given the relationship between acceleration and mpg is negative, higher horsepower may lead to less fuel efficiency for higher acceleration cars.

6 - Trying out transformations of variables

Using log(x), sqrt(x), and x^2, I will transform some of the variables in the model:

```
#Transform some of the variables to build the model

# Log
log_lm <- lm(mpg ~ log(cylinders) + log(displacement) + log(horsepower) + log(weight)
             + acceleration + year + origin, data = auto_df)

# sqrt
sqrt_lm <- lm(mpg ~ sqrt(cylinders) + sqrt(displacement) + sqrt(horsepower) + sqrt(weight)
              + acceleration + year + origin, data = auto_df)
```

```r
# x^2
square_lm <- lm(mpg ~ I(cylinders^2) + I(displacement^2) + I(horsepower^2) + I(weight^2)
                + acceleration + year + origin, data = auto_df)
```

```r
print(summary(log_lm))
```

```
##
## Call:
## lm(formula = mpg ~ log(cylinders) + log(displacement) + log(horsepower) +
##     log(weight) + acceleration + year + origin, data = auto_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4552 -1.8762 -0.0127  1.5501 12.7385
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       107.93281    9.93669  10.862  < 2e-16 ***
## log(cylinders)      1.58175    1.66035   0.953 0.341363
## log(displacement)  -0.92671    1.49978  -0.618 0.537011
## log(horsepower)    -5.65034    1.56503  -3.610 0.000346 ***
## log(weight)       -13.81131    2.20028  -6.277 9.33e-10 ***
## acceleration       -0.19980    0.10276  -1.944 0.052593 .
## year                0.72730    0.04709  15.446  < 2e-16 ***
## origin              0.85362    0.27998   3.049 0.002456 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.078 on 384 degrees of freedom
## Multiple R-squared:  0.8473, Adjusted R-squared:  0.8445
## F-statistic: 304.3 on 7 and 384 DF,  p-value: < 2.2e-16
```

Log Summary

Using the log transformation gives some interesting findings. Looking at horsepower and weight in this view shows that, as these two increase, mpg drastically decreases. These are also statistically significant.

```r
print(summary(sqrt_lm))
```

```
##
## Call:
## lm(formula = mpg ~ sqrt(cylinders) + sqrt(displacement) + sqrt(horsepower) +
##     sqrt(weight) + acceleration + year + origin, data = auto_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.4549 -1.9931 -0.1517  1.7287 13.0146
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)        5.07249    5.19241   0.977   0.3292
```

```
## sqrt(cylinders)    -0.16961    1.53597  -0.110   0.9121
## sqrt(displacement)  0.21619    0.22621   0.956   0.3398
## sqrt(horsepower)   -0.65320    0.30364  -2.151   0.0321 *
## sqrt(weight)       -0.64051    0.07766  -8.247 2.61e-15 ***
## acceleration       -0.04534    0.10250  -0.442   0.6585
## year                0.73590    0.04928  14.933  < 2e-16 ***
## origin              1.15820    0.28083   4.124 4.56e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.21 on 384 degrees of freedom
## Multiple R-squared:  0.8339, Adjusted R-squared:  0.8308
## F-statistic: 275.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

Sqrt Summary

Similar to the log transformation, we can see that weight has a strong impact on mpg. This transformation produced less statistically significant results.

```
print(summary(square_lm))
```

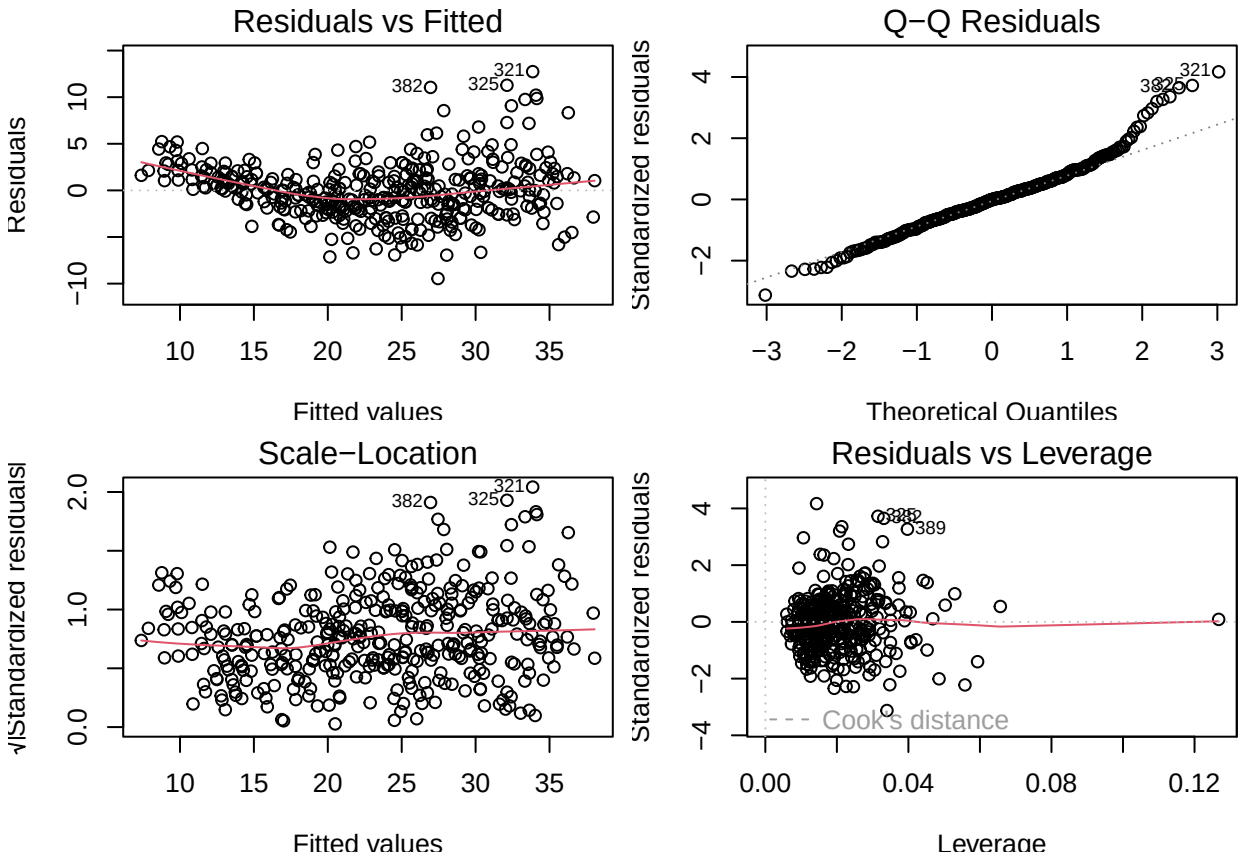
```
##
## Call:
## lm(formula = mpg ~ I(cylinders^2) + I(displacement^2) + I(horsepower^2) +
##     I(weight^2) + acceleration + year + origin, data = auto_df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.7163 -2.3649 -0.0442  1.8824 13.0095
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -3.049e+01  4.535e+00  -6.723 6.45e-11 ***
## I(cylinders^2)    -8.819e-02  2.523e-02  -3.495 0.000529 ***
## I(displacement^2)  6.070e-05  1.389e-05   4.370 1.60e-05 ***
## I(horsepower^2)   -4.585e-05  5.085e-05  -0.902 0.367851
## I(weight^2)       -9.396e-07  9.154e-08 -10.265  < 2e-16 ***
## acceleration       1.735e-01  9.314e-02   1.863 0.063181 .
## year               7.633e-01  5.375e-02  14.201  < 2e-16 ***
## origin             1.750e+00  2.772e-01   6.313 7.58e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.541 on 384 degrees of freedom
## Multiple R-squared:  0.7979, Adjusted R-squared:  0.7942
## F-statistic: 216.5 on 7 and 384 DF,  p-value: < 2.2e-16
```

Square Summary

Almost all of the variables become significant when squaring them, which is an interesting effect. The coefficients are also much smaller here. Interestingly, cylinder has a more clear negative relationship with mpg in this model than some of the others, and is statistically significant.

Given this information, I will create the log transformation diagnostic plots.

```
#format and plot the diagnostics
par(mfrow=c(2,2))
par(mai = c(0.6, 0.6, 0.3, 0.1))
par(cex = 0.8)
plot(log_lm)
```



This does improve the results a little bit, especially looking at the residuals vs. leverage plot.