Motor Trend Multiple Regression

JeremiShane

1/31/2018

Executive Summary

The following analysis explores the relationship between a set of variables and miles per gallon (MPG) in the Motor Trend Cars (mtcars) dataset. The focus of this analysis is to answer the following:

```
1. Is an automatic or manual transmission better for MPG?
2. Quantify the MPG difference between automatic and manual transmissions
```

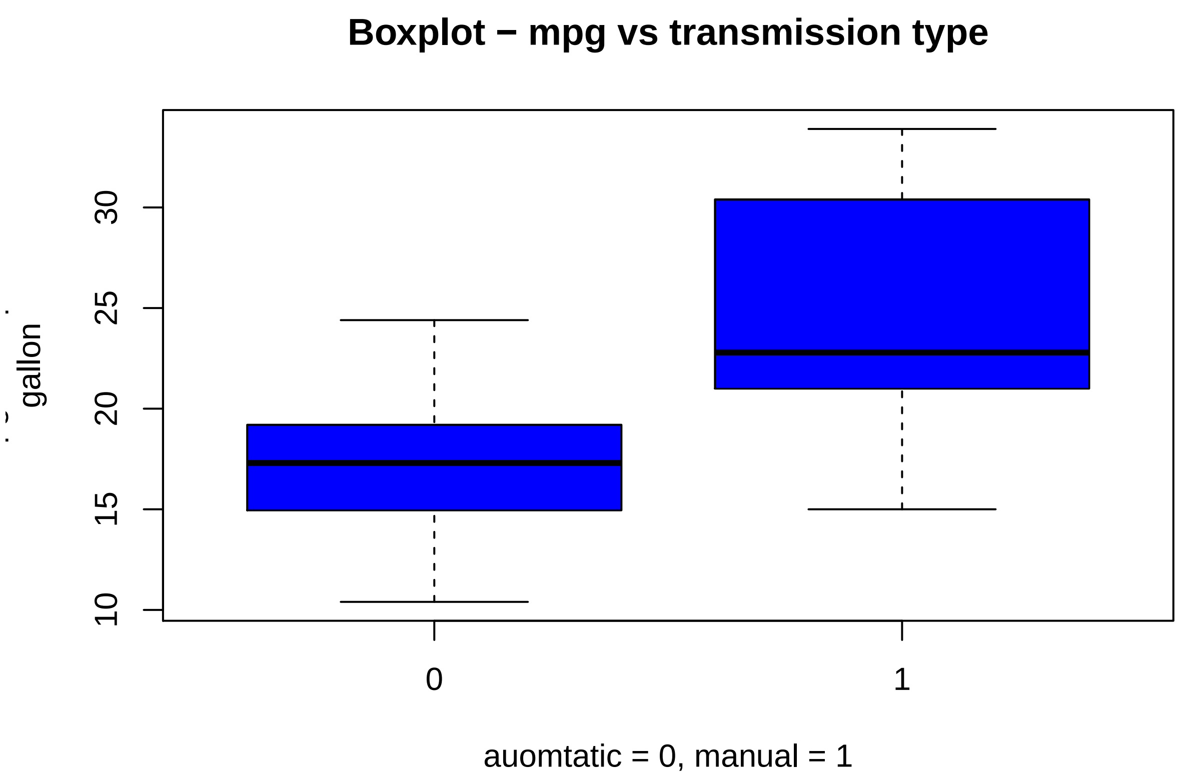
From this analysis we do see a significant difference in mpg between automatic and manual transmissions.

Modeling indicates there could be ~ 5.27 mpg improvement with manual transmission, but in the same model mpg will be reduced by the same for an approximate increase in horsepower by ~90 hp.

Based on the data and analysis we can't draw strong conclusions. We need more data, and possibly more variables to come to more confident conclusions.

This boxplot demonstrates the difference in automatic(0) vs. manual(1)

```
boxplot(mtcars$mpg ~ mtcars$am, data = mtcars, outpch = 19, ylab="mpg:miles per
gallon",xlab="auomtatic = 0, manual = 1",main="Boxplot - mpg vs transmission type", col="blue")
```



The following t-test further proves the mpg difference in transmission types

```
## hypothesis t-test
t.test(mtcars$mpg~mtcars$am,conf.level=0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  mtcars$mpg by mtcars$am
## t = -3.7671, df = 18.332, p-value = 0.001374
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -11.280194  -3.209684
## sample estimates:
## mean in group 0 mean in group 1
##        17.14737        24.39231
```

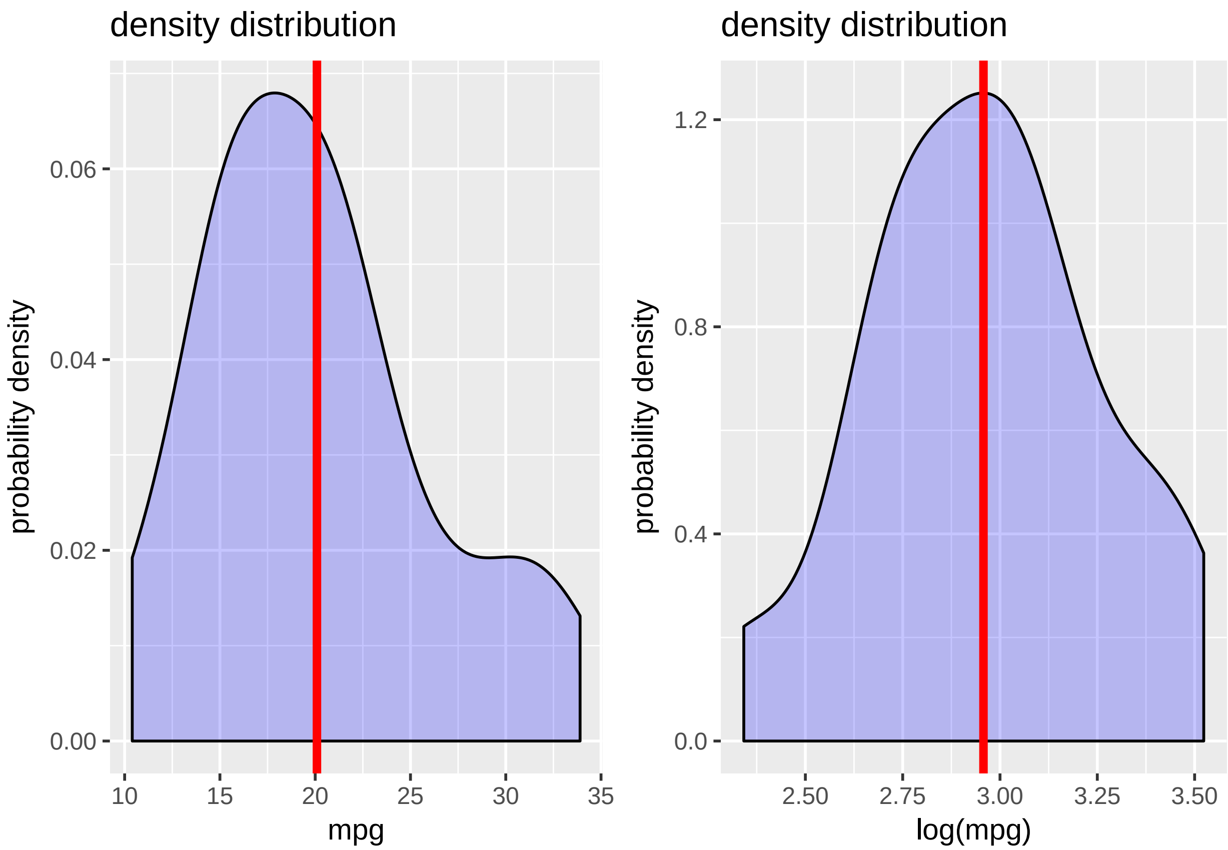
Here we reject the null hypothesis that the means are the same. Let's further explore and work towards regression analysis to better understand the other factors that impact mpg, and also to help quantify the impact of transmission type.

Regression Analysis

Distribution of mpg

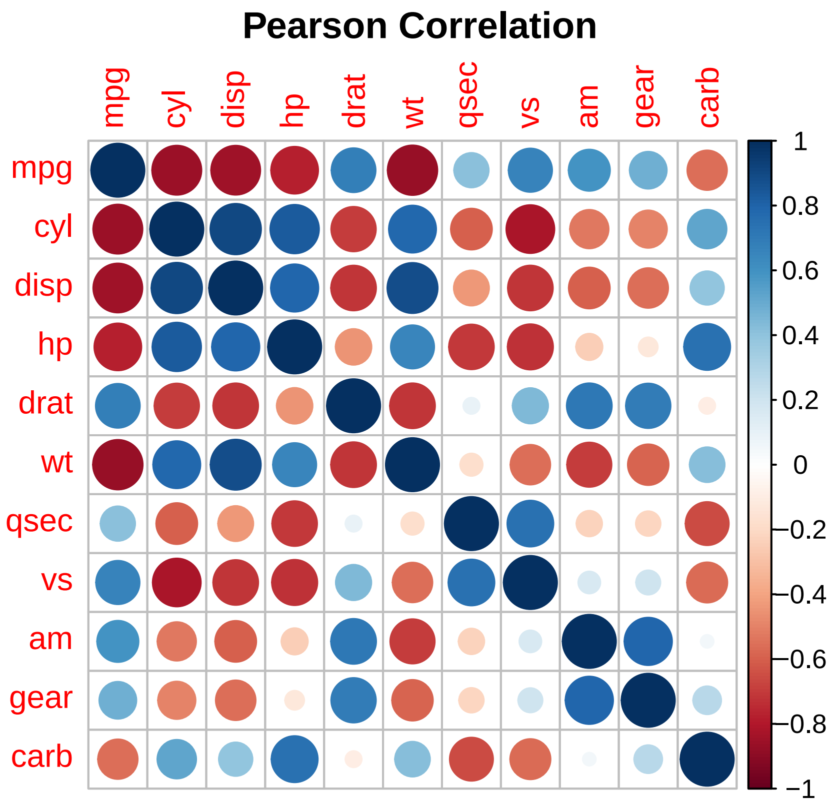
mpg and especially log(mpg) appear to be normally distributed. In addition to mtcars dataset, another dataframe df contains the log transforms of all the variables.

```
## density distributions of mpg and log(mpg)
plot1 <- ggplot(data = mtcars, aes(x=mtcars$mpg)) +
        geom_density(fill="blue", alpha=.23) +
        labs(x="mpg", y="probability density", title="density distribution") +
        geom_vline(aes(xintercept=mean(mtcars$mpg)), col="red", lwd=1.5)
plot2 <- ggplot(data = mtcars, aes(x=log(mtcars$mpg))) +
        geom_density(fill="blue", alpha=.23) +
        labs(x="log(mpg)", y="probability density", title="density distribution") +
        geom_vline(aes(xintercept=mean(log(mtcars$mpg))), col="red", lwd=1.5)
grid.arrange(plot1, plot2, ncol=2)
```



Correlation between Variables

```
## pearson correlations for numeric variables
cor <- cor(mtcars, use="pairwise", method="pearson")
## graphical display of correlations
corrplot(cor, mar=c(0,0,3,0))
title(main="Pearson Correlation")
```




```
## more on correlation in the Appendix
```

These correlations indicate a possible need to reduce the variable selection for regression analysis.

Variable Selection

Two methods were used to influence variable selection (See Appendix for details), keep in mind that our focus is on the impact of transmission(am) to mpg so we will aim to keep am as a regressor.

```
1.  Principal Components Analysis on the log transforms
2.  Stepwise automated regression and selection of regressors
```

Note - the residuals vs. leverage plots all indicate the possibility of outliers that could impact the model. For this anlysis we won't remove the identified outliers at this time.

Models

When looking at the impact of transmission type alone we see evidence again that manual transmission has a positive influence on mpg. However, with the model only considering "am" as a regressor we have a very low R Squared indicating there is a high (~70%) likelihood of obtaining values outside of the observed.

We get a much improved R Squared value when we add "hp" as a regressor with "am", and we start to see the influence other variables have on mpg, and a more complete understanding of mpg.

```
## model only with "am"
fit <- lm(mpg ~ am, df)
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ am, data = df)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4749 -0.1213  0.0073  0.1693  0.3779
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.81669    0.05653  49.830  < 2e-16 ***
## am           0.34664    0.08869   3.909 0.000491 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2464 on 30 degrees of freedom
## Multiple R-squared:  0.3374, Adjusted R-squared:  0.3153
## F-statistic: 15.28 on 1 and 30 DF,  p-value: 0.0004905
```

```
## this is the model I choose to trust most from this analysis, but still have limited confidence in the
fit <- glm(mpg ~ am + hp, mtcars, family = "gaussian")
summary(fit)
```

```
##
## Call:
## glm(formula = mpg ~ am + hp, family = "gaussian", data = mtcars)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -4.3843  -2.2642   0.1366   1.6968   5.8657
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 26.584914   1.425094  18.655  < 2e-16 ***
## am           5.277085   1.079541   4.888 3.46e-05 ***
## hp          -0.058888   0.007857  -7.495 2.92e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 8.463424)
##
##     Null deviance: 1126.05  on 31  degrees of freedom
## Residual deviance:  245.44  on 29  degrees of freedom
## AIC: 164.01
##
## Number of Fisher Scoring iterations: 2
```

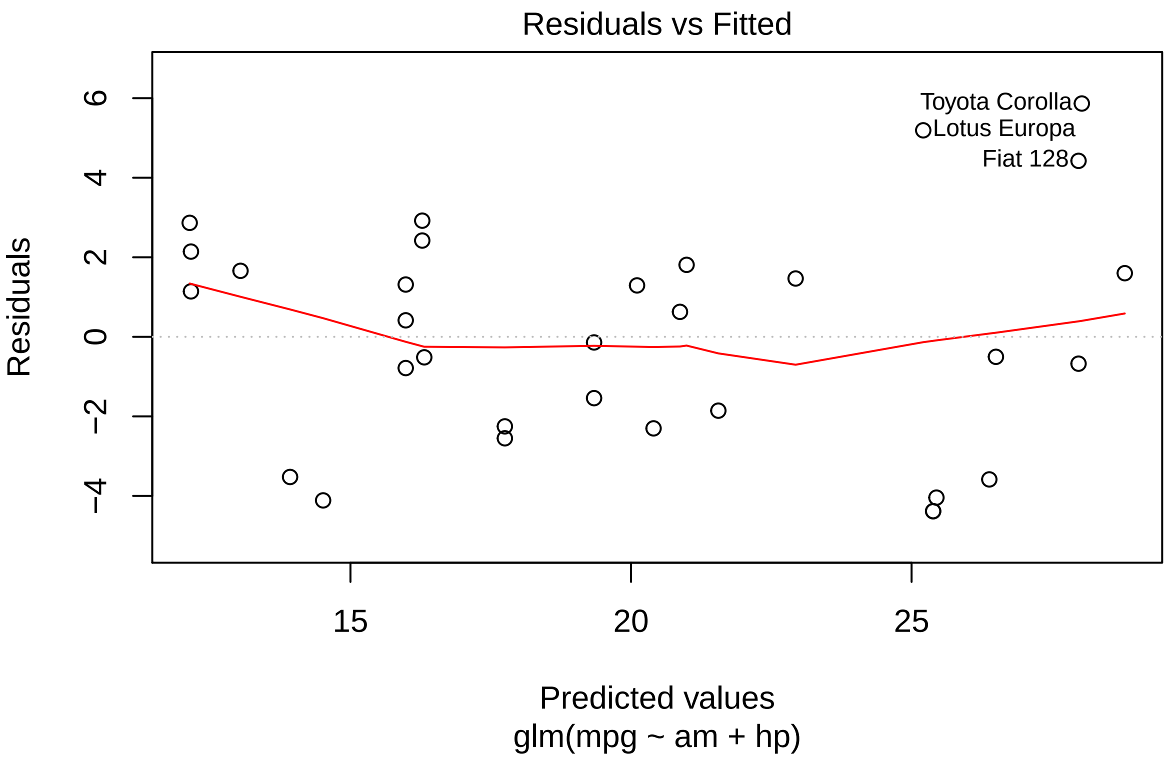
```
shapiro.test(fit$residuals) ## the null hypotheisis is normality of the residuals
```

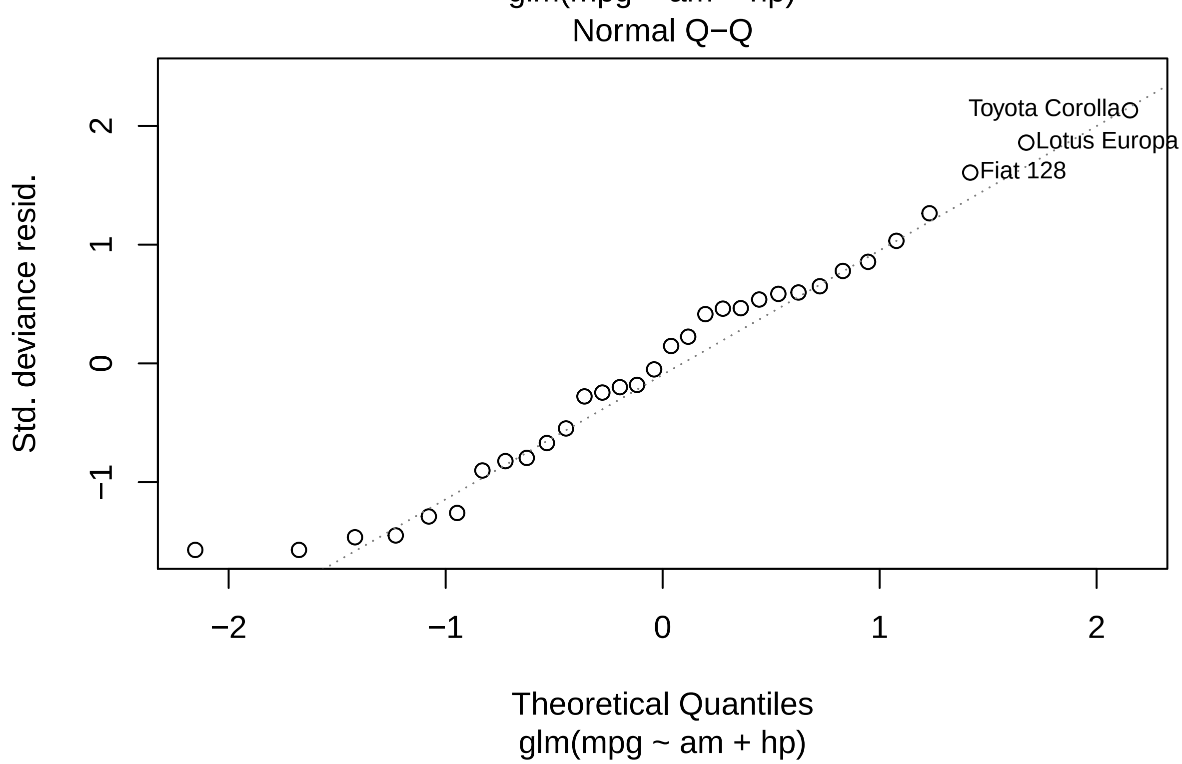
```
##
##  Shapiro-Wilk normality test
##
## data:  fit$residuals
## W = 0.96485, p-value = 0.3706
```

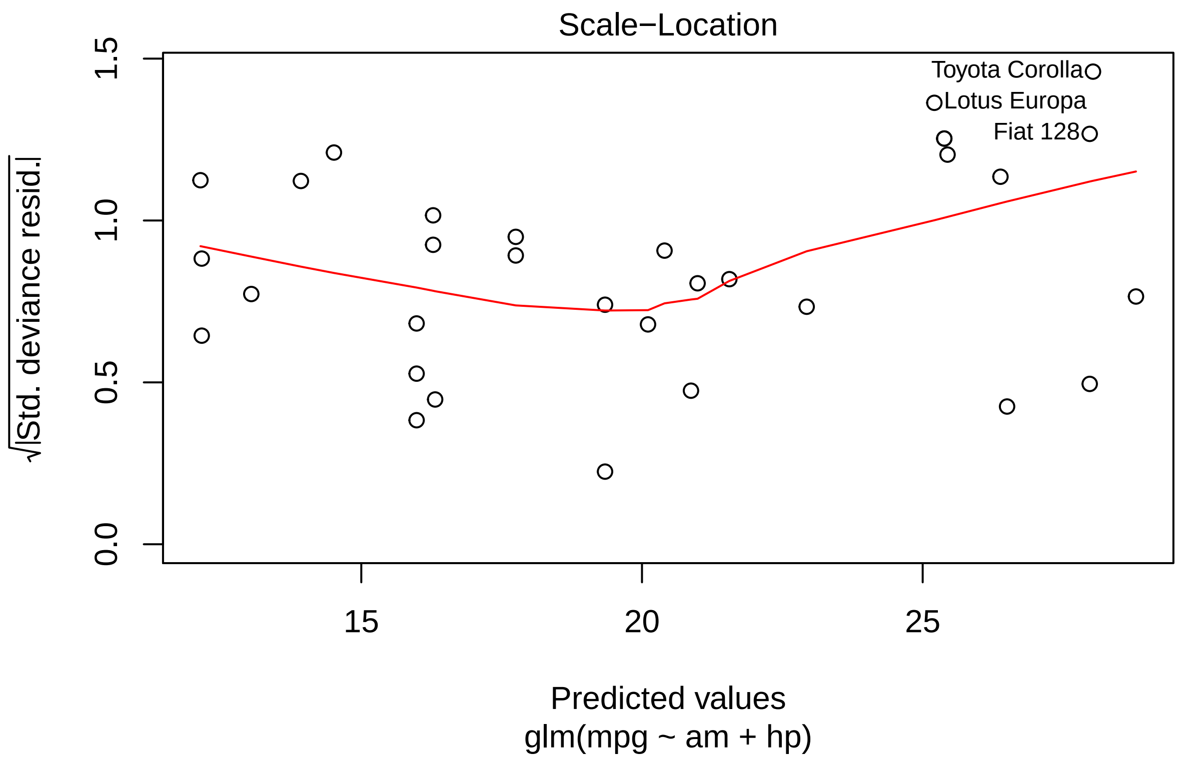
```r
VIF(fit) ## less than 10
```

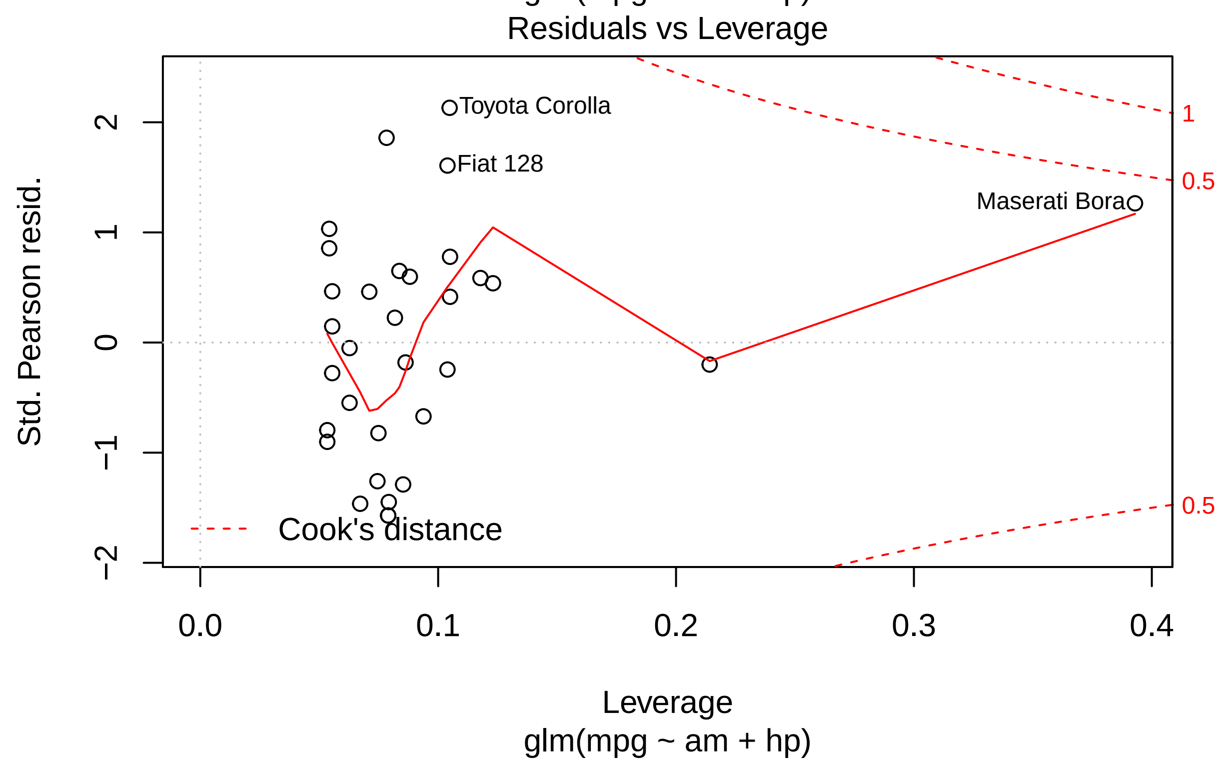
```
## numeric(0)
```

```r
plot(fit)
```










```r
model <- glm(mpg~ factor(am):wt + factor(am):qsec,data=mtcars, family = "gaussian")
summary(model)
```

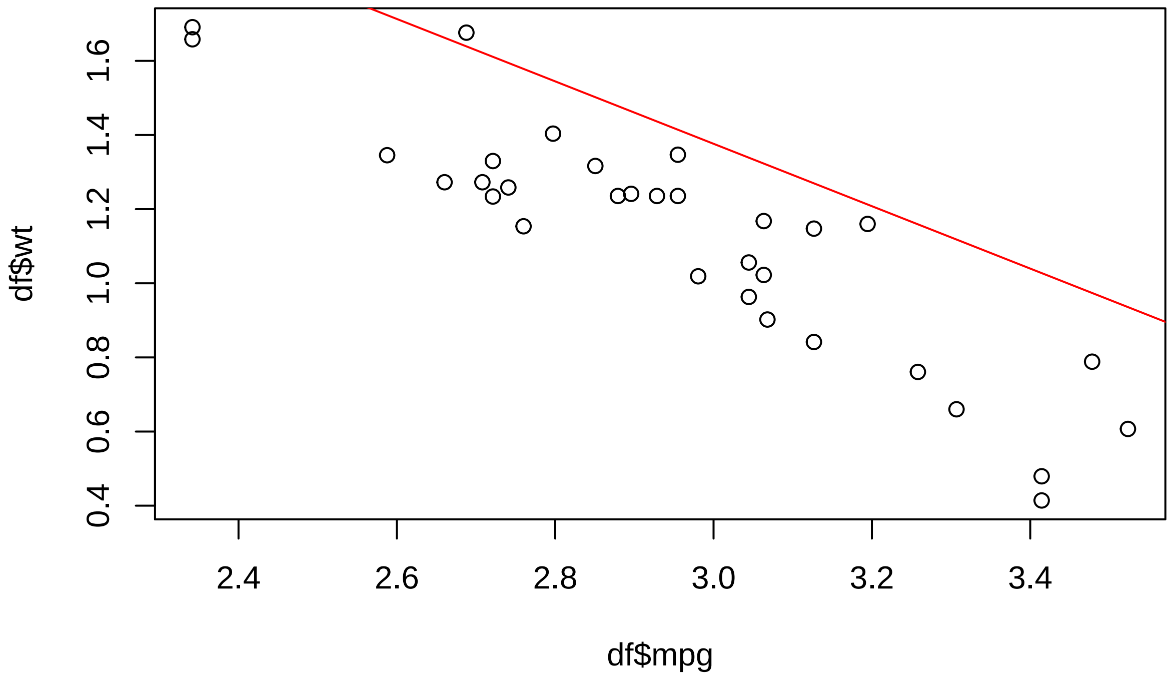
```
##
## Call:
## glm(formula = mpg ~ factor(am):wt + factor(am):qsec, family = "gaussian",
```

```
##     data = mtcars)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.9361  -1.4017  -0.1551   1.2695   3.8862
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       13.9692     5.7756   2.419  0.02259 *
## factor(am)0:wt    -3.1759     0.6362  -4.992 3.11e-05 ***
## factor(am)1:wt    -6.0992     0.9685  -6.297 9.70e-07 ***
## factor(am)0:qsec   0.8338     0.2602   3.205  0.00346 **
## factor(am)1:qsec   1.4464     0.2692   5.373 1.12e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 4.396256)
##
##     Null deviance: 1126.0  on 31  degrees of freedom
## Residual deviance:  118.7  on 27  degrees of freedom
## AIC: 144.76
##
## Number of Fisher Scoring iterations: 2
```

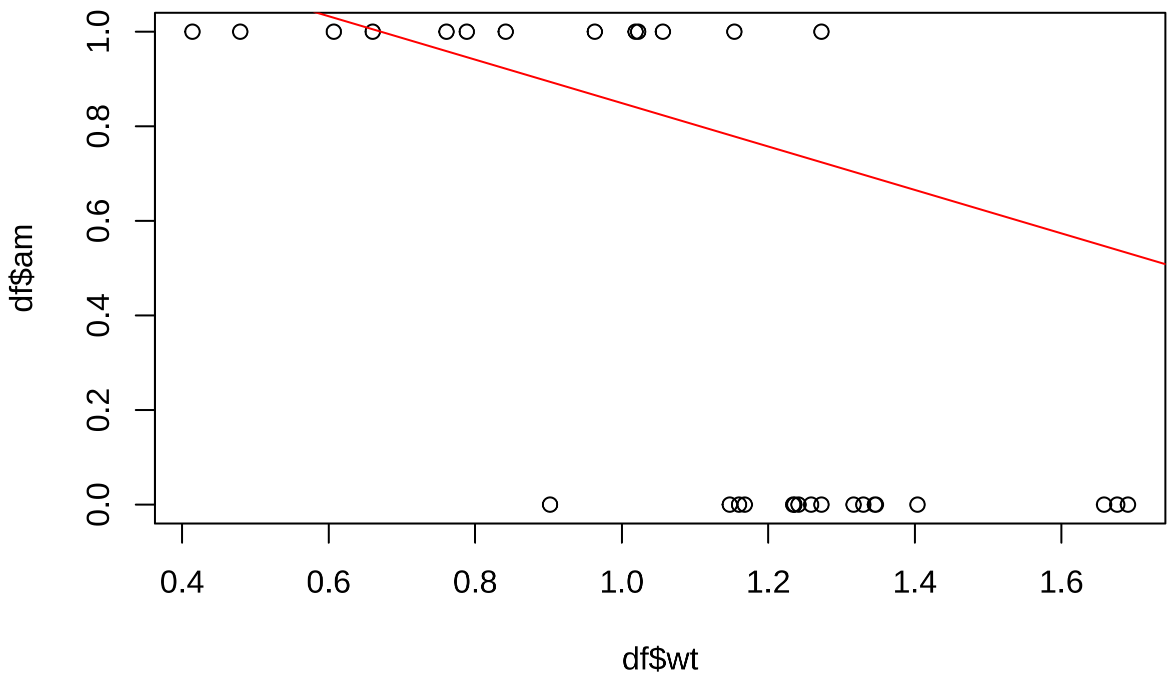
This last model is the strongest produced with strong p values for the regressors, very significant difference between residual deviance and null deviance indicating that this model with the regressors selected is significant. The lm() with same regressors in the appendix produces an Adjusted R Squared of 0.879.

However, this model could be strong based on a bias in our data which could be the result of a small/limited dataset. I draw this conclusion based on the clear visual of the mpg distributions for am=0 vs. am=1. The charts below also help to support this conclusion. Primarily, I believe we would need a more equal distribution of cars with manual transmissions in higher wt ranges and automatic transmission cars in lower wt ranges in order for the last model to be trusted.

```r
## indicates that as wt increases mpg decreases
plot(df$mpg, df$wt)
abline(lm(mpg ~ wt, data=df), col="red")
```




```
## indicates that the manual transmission cars in our dataset are of lower wt values, and that higher wt
plot(df$wt, df$am)
abline(lm(wt ~ am, data=df), col="red")
```



Appendix

Data Overview

A data frame with 32 observations on 11 variables.

```
[, 1]   mpg Miles/(US) gallon
[, 2]   cyl Number of cylinders
[, 3]   disp    Displacement (cu.in.)
[, 4]   hp  Gross horsepower
[, 5]   drat    Rear axle ratio
[, 6]   wt  Weight (1000 lbs)
[, 7]   qsec    1/4 mile time
[, 8]   vs  V/S
[, 9]   am  Transmission (0 = automatic, 1 = manual)
[,10]   gear    Number of forward gears
[,11]   carb    Number of carburetors
```

```
## str(mtcars)
summary(mtcars)
```

```
##       mpg             cyl             disp             hp
##  Min.   :10.40   Min.   :4.000   Min.   : 71.1   Min.   : 52.0
##  1st Qu.:15.43   1st Qu.:4.000   1st Qu.:120.8   1st Qu.: 96.5
##  Median :19.20   Median :6.000   Median :196.3   Median :123.0
##  Mean   :20.09   Mean   :6.188   Mean   :230.7   Mean   :146.7
##  3rd Qu.:22.80   3rd Qu.:8.000   3rd Qu.:326.0   3rd Qu.:180.0
##  Max.   :33.90   Max.   :8.000   Max.   :472.0   Max.   :335.0
##       drat             wt             qsec             vs
##  Min.   :2.760   Min.   :1.513   Min.   :14.50   Min.   :0.0000
##  1st Qu.:3.080   1st Qu.:2.581   1st Qu.:16.89   1st Qu.:0.0000
##  Median :3.695   Median :3.325   Median :17.71   Median :0.0000
##  Mean   :3.597   Mean   :3.217   Mean   :17.85   Mean   :0.4375
##  3rd Qu.:3.920   3rd Qu.:3.610   3rd Qu.:18.90   3rd Qu.:1.0000
##  Max.   :4.930   Max.   :5.424   Max.   :22.90   Max.   :1.0000
##        am              gear             carb
##  Min.   :0.0000   Min.   :3.000   Min.   :1.000
##  1st Qu.:0.0000   1st Qu.:3.000   1st Qu.:2.000
##  Median :0.0000   Median :4.000   Median :2.000
##  Mean   :0.4062   Mean   :3.688   Mean   :2.812
##  3rd Qu.:1.0000   3rd Qu.:4.000   3rd Qu.:4.000
##  Max.   :1.0000   Max.   :5.000   Max.   :8.000
```

```
## mtcars$am <- as.factor(mtcars$am)
## levels(mtcars$am) <-c("Automatic", "Manual")
```

Multiple Regression

```
## regression with all variables not a good model
fit <- glm(mpg ~ ., mtcars, family="gaussian")
summary(fit)
```

```
##
## Call:
## glm(formula = mpg ~ ., family = "gaussian", data = mtcars)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4506  -1.6044  -0.1196   1.2193   4.6271
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.30337   18.71788   0.657   0.5181
## cyl         -0.11144    1.04502  -0.107   0.9161
## disp         0.01334    0.01786   0.747   0.4635
## hp          -0.02148    0.02177  -0.987   0.3350
## drat         0.78711    1.63537   0.481   0.6353
## wt          -3.71530    1.89441  -1.961   0.0633 .
## qsec         0.82104    0.73084   1.123   0.2739
## vs           0.31776    2.10451   0.151   0.8814
## am           2.52023    2.05665   1.225   0.2340
## gear         0.65541    1.49326   0.439   0.6652
## carb        -0.19942    0.82875  -0.241   0.8122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 7.023544)
##
##     Null deviance: 1126.05  on 31  degrees of freedom
## Residual deviance:  147.49  on 21  degrees of freedom
## AIC: 163.71
##
## Number of Fisher Scoring iterations: 2
```

```r
## stepwise automated model selection
## first with our mtcars dataset
stepmodel = step(lm(data = mtcars, mpg ~ .),trace=0,steps=10000)
summary(stepmodel)
```

```
##
## Call:
## lm(formula = mpg ~ wt + qsec + am, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4811 -1.5555 -0.7257  1.4110  4.6610
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.6178     6.9596   1.382 0.177915
## wt           -3.9165     0.7112  -5.507 6.95e-06 ***
## qsec          1.2259     0.2887   4.247 0.000216 ***
## am            2.9358     1.4109   2.081 0.046716 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.459 on 28 degrees of freedom
## Multiple R-squared:  0.8497, Adjusted R-squared:  0.8336
## F-statistic: 52.75 on 3 and 28 DF,  p-value: 1.21e-11
```

```r
## now with our df dataset containing the log transform.
stepmodel = step(lm(data = df, mpg ~ .),trace=0,steps=10000)
summary(stepmodel)
```

```
##
## Call:
```

```
## lm(formula = mpg ~ hp + wt, data = df)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.201432 -0.079563  0.002145  0.078784  0.196150
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.83469    0.22440  21.545  < 2e-16 ***
## hp          -0.25532    0.05840  -4.372 0.000145 ***
## wt          -0.56228    0.08742  -6.432  4.9e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1054 on 29 degrees of freedom
## Multiple R-squared:  0.8829, Adjusted R-squared:  0.8748
## F-statistic: 109.3 on 2 and 29 DF,  p-value: 3.138e-14
```

```
## does not select our variable of interest [am]

## based on the principal components analysis
## the most trusted model
fit <- lm(mpg ~ am + hp, df)
summary(fit)
```

```
##
## Call:
## lm(formula = mpg ~ am + hp, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.33379 -0.08653  0.00964  0.07157  0.26994
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5.11957    0.27274  18.771  < 2e-16 ***
## am           0.19543    0.05146   3.798  0.00069 ***
## hp          -0.45913    0.05403  -8.498 2.31e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1341 on 29 degrees of freedom
## Multiple R-squared:  0.8102, Adjusted R-squared:  0.7971
## F-statistic: 61.88 on 2 and 29 DF,  p-value: 3.442e-11
```

```
shapiro.test(fit$residuals) ## the null hypotheisis is normality of the residuals
```

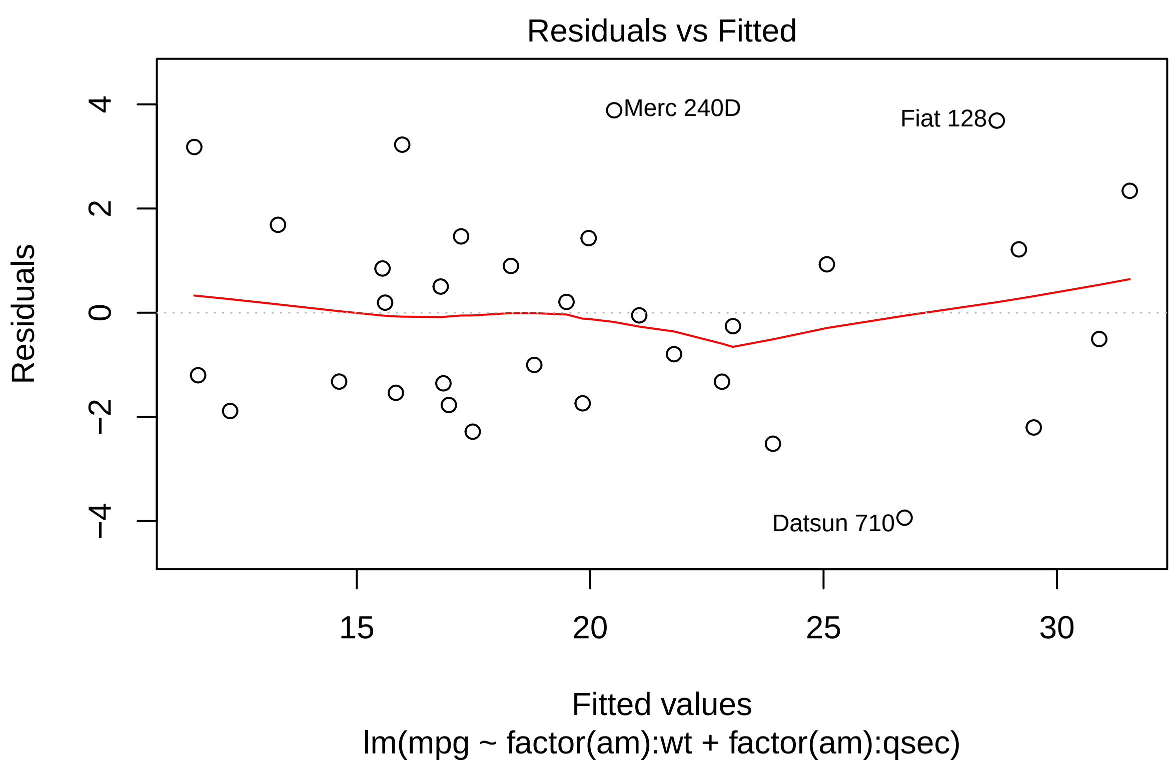
```
##
##  Shapiro-Wilk normality test
##
## data:  fit$residuals
## W = 0.95346, p-value = 0.1805
```

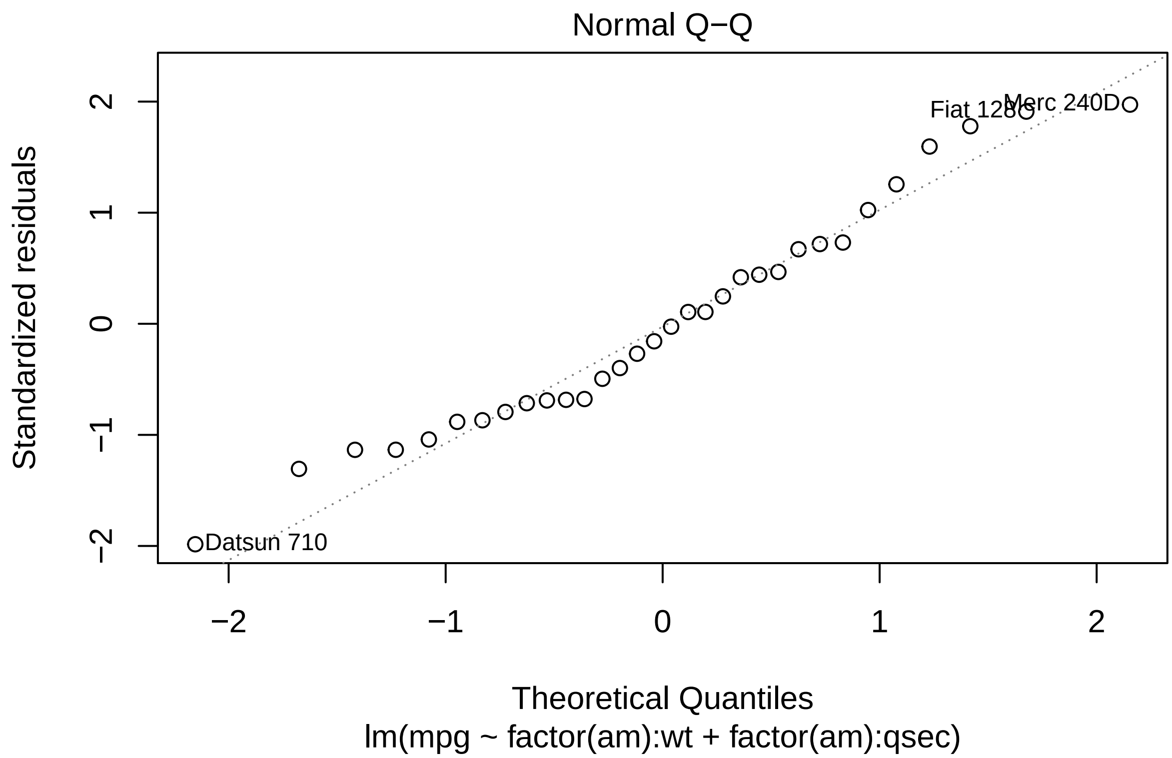
```
## VIF(fit) ## less than 10
## plot(fit)
## influence.measures(fit)
```

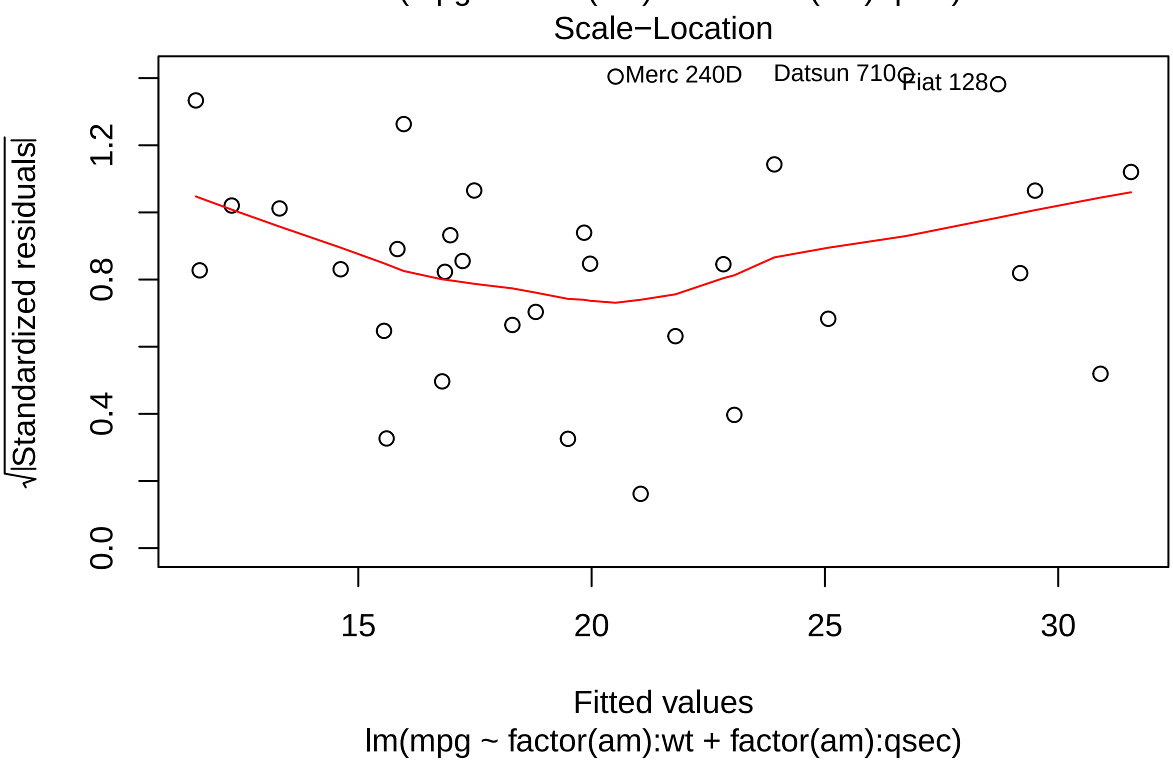
```r
## modified from stepwise model
model <- lm(mpg~ factor(am):wt + factor(am):qsec,data=mtcars)
summary(model)
```

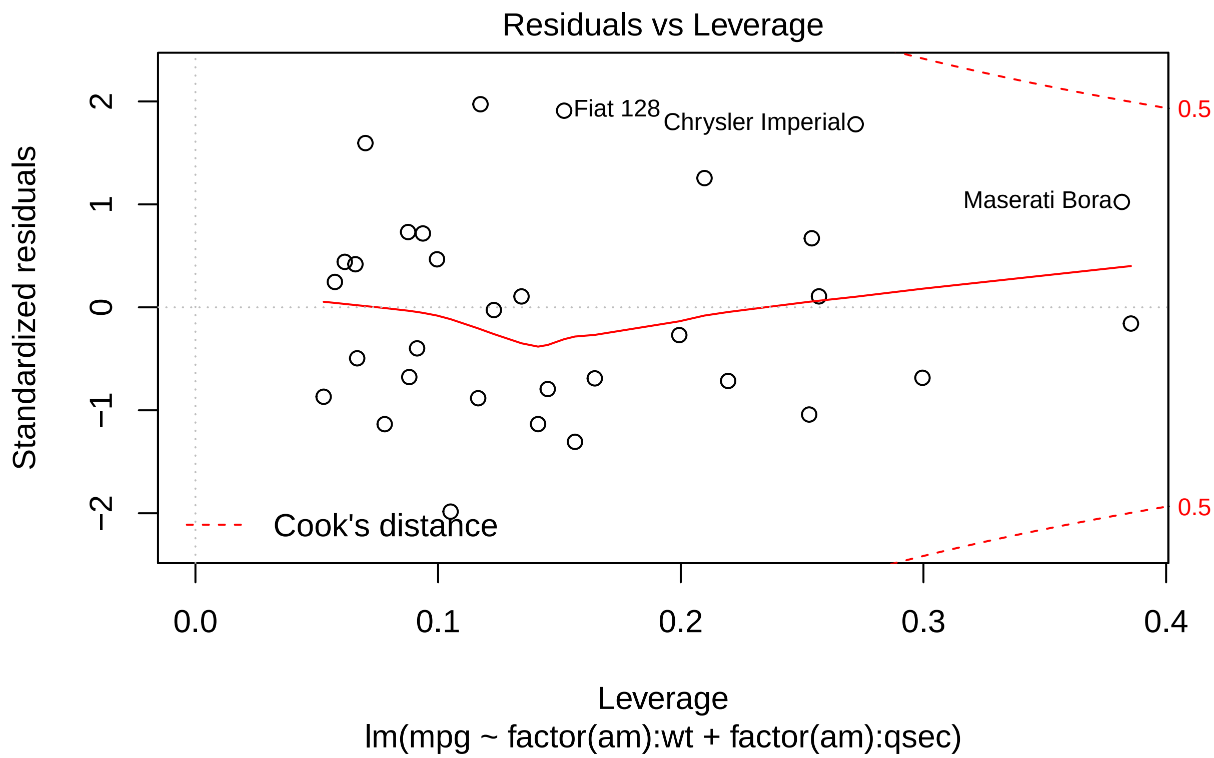
```
##
## Call:
## lm(formula = mpg ~ factor(am):wt + factor(am):qsec, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9361 -1.4017 -0.1551  1.2695  3.8862
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       13.9692     5.7756   2.419  0.02259 *
## factor(am)0:wt    -3.1759     0.6362  -4.992 3.11e-05 ***
## factor(am)1:wt    -6.0992     0.9685  -6.297 9.70e-07 ***
## factor(am)0:qsec   0.8338     0.2602   3.205  0.00346 **
## factor(am)1:qsec   1.4464     0.2692   5.373 1.12e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.097 on 27 degrees of freedom
## Multiple R-squared:  0.8946, Adjusted R-squared:  0.879
## F-statistic: 57.28 on 4 and 27 DF,  p-value: 8.424e-13
```

```r
plot(model)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(mpg ~ factor(am):wt + factor(am):qsec)
Datsun 710
Fiat 128
Merc 240D
−2
−1
0
1
2


Scale−Location
√|Standardized residuals|
Fitted values
lm(mpg ~ factor(am):wt + factor(am):qsec)
Merc 240D
Datsun 710
Fiat 128
0.0
0.4
0.8
1.2
15
20
25
30




```r
shapiro.test(model$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model$residuals
## W = 0.96823, p-value = 0.4521
```

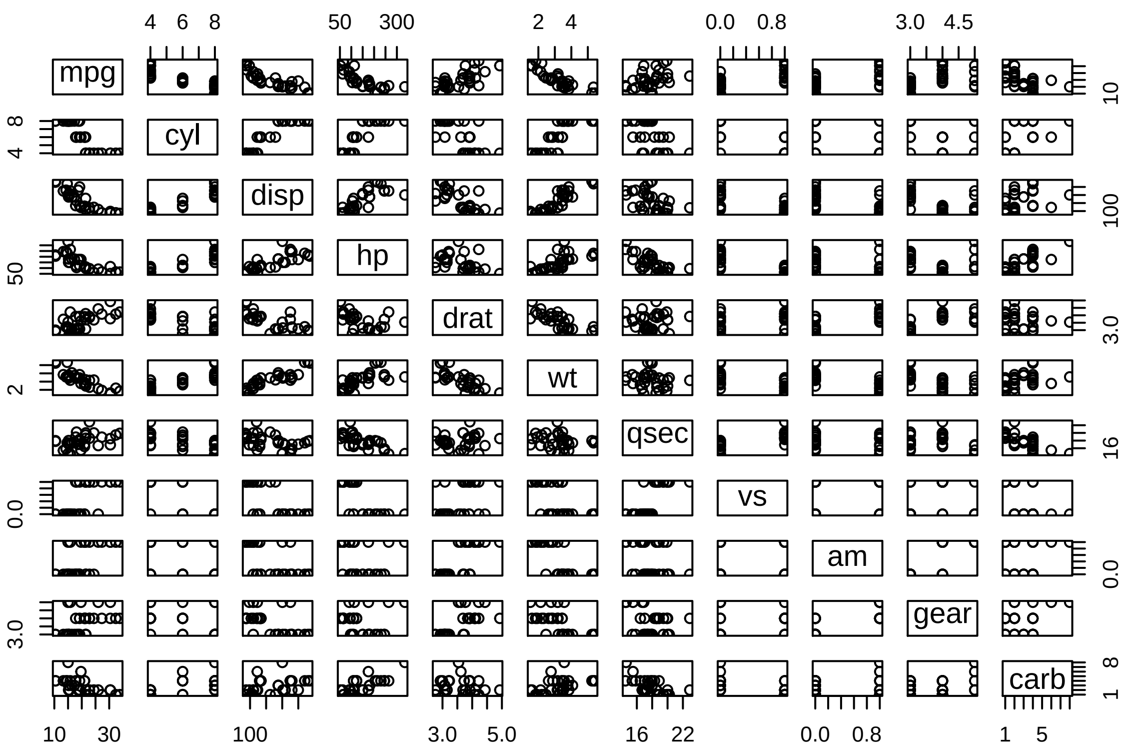
```r
influence.measures(model)
```

```
## Influence measures of
##   lm(formula = mpg ~ factor(am):wt + factor(am):qsec, data = mtcars) :
##
##                       dfb.1_ dfb.fctr.m.0.w dfb.fctr.m.1.w dfb.fctr.m.0.q
## Mazda RX4           -0.02212        0.01326       -0.03127        0.02016
## Mazda RX4 Wag        0.00237       -0.00142       -0.00587       -0.00216
## Datsun 710           0.25009       -0.14990       -0.03504       -0.22788
## Hornet 4 Drive      -0.02097       -0.09655        0.01320        0.09202
## Hornet Sportabout    0.15529       -0.09790       -0.09776       -0.12267
## Valiant              0.15041        0.03087       -0.09468       -0.22148
## Duster 360          -0.26795        0.10458        0.16868        0.25493
## Merc 240D           -0.19346       -0.26982        0.12179        0.42543
## Merc 230             0.08037        0.01021       -0.05059       -0.10811
## Merc 280             0.02691       -0.04179       -0.01694       -0.00126
## Merc 280C            0.00401        0.03847       -0.00253       -0.03619
## Merc 450SE           0.02008        0.03108       -0.01264       -0.03037
## Merc 450SL           0.02154       -0.00617       -0.01356       -0.01741
## Merc 450SLC         -0.02840       -0.00152        0.01788        0.01480
## Cadillac Fleetwood   0.28133       -0.53968       -0.17710       -0.09879
## Lincoln Continental  0.19975       -0.40124       -0.12575       -0.05990
## Chrysler Imperial   -0.40727        1.00203        0.25638        0.04115
```

```
## Fiat 128            -0.39107        0.23440       -0.01830        0.35633
## Honda Civic         -0.03060        0.01834        0.09275        0.02788
## Toyota Corolla      -0.17595        0.10546       -0.23036        0.16032
## Toyota Corona       -0.03519        0.27334        0.02216       -0.11420
## Dodge Challenger    -0.14533        0.07647        0.09148        0.12266
## AMC Javelin         -0.20661        0.14574        0.13006        0.15118
## Camaro Z28          -0.23462        0.03951        0.14770        0.24959
## Pontiac Firebird     0.21762       -0.00897       -0.13699       -0.22190
## Fiat X1-9            0.04244       -0.02544        0.18650       -0.03867
## Porsche 914-2        0.07999       -0.04794       -0.06640       -0.07288
## Lotus Europa         0.24894       -0.14921       -0.32238       -0.22683
## Ford Pantera L       0.01515       -0.00908        0.03141       -0.01380
## Ferrari Dino         0.01378       -0.00826        0.01233       -0.01255
## Maserati Bora       -0.00442        0.00265        0.58260        0.00403
## Volvo 142E           0.35314       -0.21167       -0.32862       -0.32177
##                     dfb.fctr.m.1.q    dffit cov.r   cook.d    hat inf
## Mazda RX4                 0.028951 -0.12438 1.290 3.19e-03 0.0913
## Mazda RX4 Wag            -0.000958 -0.00961 1.377 1.92e-05 0.1230
## Datsun 710               -0.370609 -0.72229 0.614 9.25e-02 0.1051
## Hornet 4 Drive            0.019240  0.22868 1.210 1.07e-02 0.0937
## Hornet Sportabout        -0.142475  0.22469 1.198 1.03e-02 0.0876
## Valiant                  -0.137994 -0.31924 1.181 2.06e-02 0.1165
## Duster 360                0.245832 -0.32474 1.256 2.14e-02 0.1452
## Merc 240D                 0.177495  0.76351 0.628 1.04e-01 0.1175
## Merc 230                 -0.073735 -0.12247 1.956 3.11e-03 0.3855   *
## Merc 280                 -0.024685  0.11143 1.241 2.56e-03 0.0615
## Merc 280C                -0.003682 -0.13040 1.236 3.50e-03 0.0667
## Merc 450SE               -0.018424  0.10956 1.251 2.48e-03 0.0659
## Merc 450SL               -0.019765  0.05985 1.267 7.42e-04 0.0575
## Merc 450SLC               0.026058 -0.20418 1.107 8.42e-03 0.0528
## Cadillac Fleetwood       -0.258107 -0.60690 1.317 7.34e-02 0.2529
## Lincoln Continental      -0.183262 -0.44322 1.580 4.01e-02 0.2996   *
## Chrysler Imperial         0.373653  1.13548 0.890 2.36e-01 0.2721   *
## Fiat 128                  0.571366  0.85308 0.689 1.31e-01 0.1519
## Honda Civic              -0.018480 -0.13223 1.488 3.62e-03 0.1994
## Toyota Corolla            0.383366  0.65412 1.131 8.37e-02 0.2098
## Toyota Corona             0.032289 -0.37574 1.406 2.88e-02 0.2195
## Dodge Challenger          0.133329 -0.20839 1.216 8.87e-03 0.0881
## AMC Javelin               0.189552 -0.33185 1.026 2.18e-02 0.0780
## Camaro Z28                0.215257 -0.30322 1.323 1.88e-02 0.1646
## Pontiac Firebird         -0.199654  0.45155 0.792 3.84e-02 0.0701
## Fiat X1-9                -0.188468 -0.46243 1.102 4.23e-02 0.1412
## Porsche 914-2            -0.049650  0.15290 1.288 4.82e-03 0.0996
## Lotus Europa             -0.122206  0.38742 1.488 3.07e-02 0.2540
## Ford Pantera L           -0.030352  0.06157 1.622 7.87e-04 0.2569   *
## Ferrari Dino             -0.019446  0.04098 1.392 3.49e-04 0.1344
## Maserati Bora            -0.240617  0.80522 1.603 1.29e-01 0.3818   *
## Volvo 142E               -0.325067 -0.57036 1.033 6.33e-02 0.1564
```

Correlations And Variable Selection

```
## pairs
pairs(mtcars)
```




```r
## print correlations
print(cor)
```

```
##             mpg        cyl       disp         hp        drat         wt
## mpg   1.0000000 -0.8521620 -0.8475514 -0.7761684  0.68117191 -0.8676594
## cyl  -0.8521620  1.0000000  0.9020329  0.8324475 -0.69993811  0.7824958
## disp -0.8475514  0.9020329  1.0000000  0.7909486 -0.71021393  0.8879799
## hp   -0.7761684  0.8324475  0.7909486  1.0000000 -0.44875912  0.6587479
## drat  0.6811719 -0.6999381 -0.7102139 -0.4487591  1.00000000 -0.7124406
## wt   -0.8676594  0.7824958  0.8879799  0.6587479 -0.71244065  1.0000000
## qsec  0.4186840 -0.5912421 -0.4336979 -0.7082234  0.09120476 -0.1747159
## vs    0.6640389 -0.8108118 -0.7104159 -0.7230967  0.44027846 -0.5549157
## am    0.5998324 -0.5226070 -0.5912270 -0.2432043  0.71271113 -0.6924953
## gear  0.4802848 -0.4926866 -0.5555692 -0.1257043  0.69961013 -0.5832870
## carb -0.5509251  0.5269883  0.3949769  0.7498125 -0.09078980  0.4276059
##             qsec         vs          am       gear        carb
## mpg   0.41868403  0.6640389  0.59983243  0.4802848 -0.55092507
## cyl  -0.59124207 -0.8108118 -0.52260705 -0.4926866  0.52698829
## disp -0.43369788 -0.7104159 -0.59122704 -0.5555692  0.39497686
## hp   -0.70822339 -0.7230967 -0.24320426 -0.1257043  0.74981247
## drat  0.09120476  0.4402785  0.71271113  0.6996101 -0.09078980
## wt   -0.17471588 -0.5549157 -0.69249526 -0.5832870  0.42760594
## qsec  1.00000000  0.7445354 -0.22986086 -0.2126822 -0.65624923
## vs    0.74453544  1.0000000  0.16834512  0.2060233 -0.56960714
## am   -0.22986086  0.1683451  1.00000000  0.7940588  0.05753435
## gear -0.21268223  0.2060233  0.79405876  1.0000000  0.27407284
## carb -0.65624923 -0.5696071  0.05753435  0.2740728  1.00000000
```

Principal Components Analysis of Log Transform Independent Variables

Analysis of the principal components shows log(hp) may be a good regressor to include along with am for regression analysis of mpg. This is based on orthoginal vectors that can be clearly seen below.

```
dfi <- df[,2:10] ## data frame of log independent variables
pc <- prcomp(dfi, scale=TRUE, center=TRUE, tol=0)
dfi2 <- df[, c(4,9)]
pc2 <- prcomp(dfi2, scale=TRUE, center=TRUE, tol=0)
par(mfrow = c(1,2))
biplot(pc, main="PCA All")
biplot(pc2, main="PCA [am] and [hp]")
```