MPGTransmissionStudy.Rmd

Jim Callahan

October 7-20, 2015

Executive Summary

This project was intended to answer the following two questions:

1. "Is an automatic or manual transmission better for MPG?"

2. "Quantify the MPG difference between automatic and manual transmissions?"

using statistical regression analysis in **R** on the **"Motor Trend"**, **"mtcars"** data set included with the **R** system.

Data Vintage

The source of the **"mtcars"** data set (as described in the documentation **help(mtcars)**) is Henderson and Velleman (1981), **Building multiple regression models interactively.** Biometrics, 37, 391–411. http://www.mortality.org/INdb/2008/02/12/8/document.pdf

The **help(mtcars)** documentation states:

> "The data was extracted from the **1974 Motor Trend** US magazine,
> and comprises fuel consumption and 10 aspects of automobile design and performance
> for 32 automobiles (**1973–74 models**)."

So, it should be noted the **"mtcars"** data set is vintage **mid-1970s** and is therefore unlikely to be representative of the contemporary state of the automotive art.

Exploratory Data Analysis

According to the **help(mtcars)** documentation, **"mtcars"** is

> "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 (lb/1000)
> [, 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"

The documentation was confirmed using the **str()** (structure) function in **R** (ommited because the confimatory listing is redundant).

Preliminary Analysis

On the surface the minimum requirements of this project are trivially simple:
1. Convert the zero-one transmission indicator variable, **"am"** to an **R "factor"**.
2. Run a regression with mpg = f(am) or in **R** notation **lm(mpg ~ am))**
I have supressed the intercept ("0 +"), so the coefficients can be read off directly without having to calculate the manual transmision as a base plus an offset.

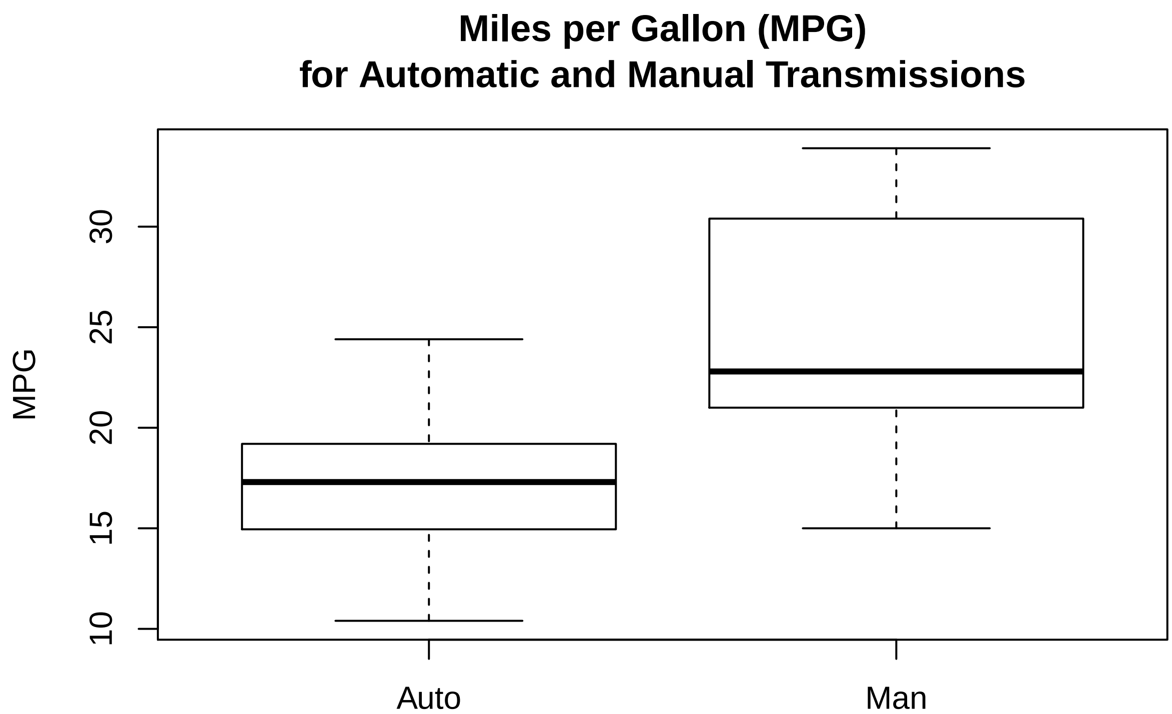
```
# MPG Model zero "000" -- our "quick and dirty" literal regression
mtcars$am <- factor(mtcars$am,levels=c(0,1), labels=c("Auto","Man"))
MPGmod000 <- lm(mpg ~ 0 + as.factor(am), data=mtcars)
MPGmod000
```

```
##
## Call:
## lm(formula = mpg ~ 0 + as.factor(am), data = mtcars)
##
## Coefficients:
## as.factor(am)Auto   as.factor(am)Man
##             17.15              24.39
```

So, the "quick and dirty" interpretation our base model zero, would be that the
average 1975 vintage car with automatic transmission gets 17+ miles per gallon while the average 1975 vintage car with a manual transmisson gets an additional 7+ miles per gallon for a total of 24+ miles per gallon.

We can picture this with a box plot (also known as a "box and whiskers" plot):
https://en.wikipedia.org/wiki/Box_plot

```
plot(as.factor(mtcars$am), mtcars$mpg,
     main = "Miles per Gallon (MPG)\nfor Automatic and Manual Transmissions",
          ylab = "MPG")
abline(mtcars$mpg ~ as.factor(mtcars$am))
```



Clearly, as indicated by the dark horizontal line, the mean mpg of the manual transmission cars is higher than the mean mpg of the automatic transmission cars. But, the "whiskers" of the "box and whiskers" plot (the interquartile range) shows that the two ranges overlap; in other words, some cars with manual transmissions have mpgs as low or lower than some cars with automatic transmissions. If manual transmission cars always had higher mpg, there would be no overlap of the interquartile ranges.

Of course to accept this analysis at face value, one would have to invoke the economist's assumption of "***ceteris paribus***" (all other things being equal).

Of course we know all other things are **NOT EQUAL**. There are **confounding variables**. For instance, the cars vary in weight, number of cylinders in their engines and the size of their engines measured in cubic inch displacement.

One, **low tech** way of seeing what is going on is simply to **sort the data set by mpg** and look at the data.

```
mtcars[order(-mtcars$mpg), ]
```

```
##                      mpg cyl  disp  hp drat    wt  qsec vs   am gear carb
## Toyota Corolla      33.9   4  71.1  65 4.22 1.835 19.90  1  Man    4    1
## Fiat 128            32.4   4  78.7  66 4.08 2.200 19.47  1  Man    4    1
## Honda Civic         30.4   4  75.7  52 4.93 1.615 18.52  1  Man    4    2
## Lotus Europa        30.4   4  95.1 113 3.77 1.513 16.90  1  Man    5    2
## Fiat X1-9           27.3   4  79.0  66 4.08 1.935 18.90  1  Man    4    1
## Porsche 914-2       26.0   4 120.3  91 4.43 2.140 16.70  0  Man    5    2
## Merc 240D           24.4   4 146.7  62 3.69 3.190 20.00  1 Auto    4    2
## Datsun 710          22.8   4 108.0  93 3.85 2.320 18.61  1  Man    4    1
## Merc 230            22.8   4 140.8  95 3.92 3.150 22.90  1 Auto    4    2
```

```
## Toyota Corona       21.5   4 120.1  97 3.70 2.465 20.01  1 Auto    3    1
## Hornet 4 Drive      21.4   6 258.0 110 3.08 3.215 19.44  1 Auto    3    1
## Volvo 142E          21.4   4 121.0 109 4.11 2.780 18.60  1  Man    4    2
## Mazda RX4           21.0   6 160.0 110 3.90 2.620 16.46  0  Man    4    4
## Mazda RX4 Wag       21.0   6 160.0 110 3.90 2.875 17.02  0  Man    4    4
## Ferrari Dino        19.7   6 145.0 175 3.62 2.770 15.50  0  Man    5    6
## Merc 280            19.2   6 167.6 123 3.92 3.440 18.30  1 Auto    4    4
## Pontiac Firebird    19.2   8 400.0 175 3.08 3.845 17.05  0 Auto    3    2
## Hornet Sportabout   18.7   8 360.0 175 3.15 3.440 17.02  0 Auto    3    2
## Valiant             18.1   6 225.0 105 2.76 3.460 20.22  1 Auto    3    1
## Merc 280C           17.8   6 167.6 123 3.92 3.440 18.90  1 Auto    4    4
## Merc 450SL          17.3   8 275.8 180 3.07 3.730 17.60  0 Auto    3    3
## Merc 450SE          16.4   8 275.8 180 3.07 4.070 17.40  0 Auto    3    3
## Ford Pantera L      15.8   8 351.0 264 4.22 3.170 14.50  0  Man    5    4
## Dodge Challenger    15.5   8 318.0 150 2.76 3.520 16.87  0 Auto    3    2
## Merc 450SLC         15.2   8 275.8 180 3.07 3.780 18.00  0 Auto    3    3
## AMC Javelin         15.2   8 304.0 150 3.15 3.435 17.30  0 Auto    3    2
## Maserati Bora       15.0   8 301.0 335 3.54 3.570 14.60  0  Man    5    8
## Chrysler Imperial   14.7   8 440.0 230 3.23 5.345 17.42  0 Auto    3    4
## Duster 360          14.3   8 360.0 245 3.21 3.570 15.84  0 Auto    3    4
## Camaro Z28          13.3   8 350.0 245 3.73 3.840 15.41  0 Auto    3    4
## Cadillac Fleetwood  10.4   8 472.0 205 2.93 5.250 17.98  0 Auto    3    4
## Lincoln Continental 10.4   8 460.0 215 3.00 5.424 17.82  0 Auto    3    4
```

The **top 5 high mileage cars** tend to have **smaller engines (as measured by cylinders (cyl) displacemnet (disp) and horsepower (hp))** and **weigh less than 2,200 pounds.** The **high milage cars** also tend to be **slower** (as measured by their quarter mile times (**qsec**)), have **manual transmissions (am = 1 or "Man")** with more gears (**gear**) and fewer carburetors (**carb**).

```
head(mtcars[order(-mtcars$mpg), ], 5)
```

```
##                 mpg cyl disp  hp drat    wt  qsec vs  am gear carb
## Toyota Corolla 33.9   4 71.1  65 4.22 1.835 19.90  1 Man    4    1
## Fiat 128       32.4   4 78.7  66 4.08 2.200 19.47  1 Man    4    1
## Honda Civic    30.4   4 75.7  52 4.93 1.615 18.52  1 Man    4    2
## Lotus Europa   30.4   4 95.1 113 3.77 1.513 16.90  1 Man    5    2
## Fiat X1-9      27.3   4 79.0  66 4.08 1.935 18.90  1 Man    4    1
```

While the **bottom 5 low mileage cars** tend to have **bigger engines (as measured by cylinders (cyl) displacemnet (disp) and horsepower (hp))** and **weigh more than 3,500 pounds.** The **low milage cars** also tend to be **faster** (as measured by their quarter mile times (**qsec**)), have **automatic transmissions (am = 0 or "Auto")** with fewer gears (**gear**) and more carburetors (**carb**).

```
tail(mtcars[order(-mtcars$mpg), ], 5)
```

```
##                      mpg cyl disp  hp drat    wt  qsec vs   am gear carb
## Chrysler Imperial   14.7   8  440 230 3.23 5.345 17.42  0 Auto    3    4
## Duster 360          14.3   8  360 245 3.21 3.570 15.84  0 Auto    3    4
## Camaro Z28          13.3   8  350 245 3.73 3.840 15.41  0 Auto    3    4
## Cadillac Fleetwood  10.4   8  472 205 2.93 5.250 17.98  0 Auto    3    4
## Lincoln Continental 10.4   8  460 215 3.00 5.424 17.82  0 Auto    3    4
```

So, what variables in addition to the automatic versus manual transmission variable (**"am"**) should be tested as possible explainations for the difference in mpg between different models of cars?

The traditional manual approach would be to look at a **correlation matrix** (which measures linear associations between variables):

```
# Correlation matrix
# From, "R Graphics Cookbook" by Winston Chang, Chapter 13, page 267
data(mtcars)     # reload raw data -- so all variables are numeric and not factor
mcor = cor(mtcars)
round(mcor, digits = 2)
```

```
##        mpg   cyl  disp    hp  drat    wt  qsec    vs    am  gear  carb
## mpg   1.00 -0.85 -0.85 -0.78  0.68 -0.87  0.42  0.66  0.60  0.48 -0.55
## cyl  -0.85  1.00  0.90  0.83 -0.70  0.78 -0.59 -0.81 -0.52 -0.49  0.53
## disp -0.85  0.90  1.00  0.79 -0.71  0.89 -0.43 -0.71 -0.59 -0.56  0.39
## hp   -0.78  0.83  0.79  1.00 -0.45  0.66 -0.71 -0.72 -0.24 -0.13  0.75
## drat  0.68 -0.70 -0.71 -0.45  1.00 -0.71  0.09  0.44  0.71  0.70 -0.09
## wt   -0.87  0.78  0.89  0.66 -0.71  1.00 -0.17 -0.55 -0.69 -0.58  0.43
## qsec  0.42 -0.59 -0.43 -0.71  0.09 -0.17  1.00  0.74 -0.23 -0.21 -0.66
## vs    0.66 -0.81 -0.71 -0.72  0.44 -0.55  0.74  1.00  0.17  0.21 -0.57
## am    0.60 -0.52 -0.59 -0.24  0.71 -0.69 -0.23  0.17  1.00  0.79  0.06
## gear  0.48 -0.49 -0.56 -0.13  0.70 -0.58 -0.21  0.21  0.79  1.00  0.27
## carb -0.55  0.53  0.39  0.75 -0.09  0.43 -0.66 -0.57  0.06  0.27  1.00
```

Zero indicates no linear association. As values approach positive one (+1.00) or negative one (-1.00), that indicates a stronger linear assocation. As shown on the diagonal of the correlation matrix, all variables have a positive one (+1.00) linear association with themselves. If we look down the **mpg** column (or equivalently across the **mpg** row) we see that variables **"wt"** (weight), **"cyl"** (number of engine cylinders) and **"disp"** engine displacement measured in cubic inches have the strongest (largest absolute value) association with **"mpg"**. Weight (**"wt"**) has a -0.87 correlation with **"mpg"**; while cylinders (**"cyl"**) and displacement (**"disp"**) have a -0.85 correlation and thus would be good candidates to try with the **"am"** variable (automatic/manual transmission) in the regression.

The negative sign on the correlation coefficients indicates an inverse relationship, for example one would expect as weight goes up mpg goes down (recall weight, **"wt"** has a -0.87 correlation with **"mpg"**).

Winston Chang's **"R Graphics Cookbook"** has a very pretty color coded and sorted correlation matrix (using the mtcars data) on page 270, Figure 13-3.

Regression Analysis

Let's recreate the factors we removed for the correlation matrix:

```
# create factors with value labels
data(mtcars)
mtcars$gear <- factor(mtcars$gear,levels=c(3,4,5),
labels=c("3gears","4gears","5gears"))
mtcars$am <- factor(mtcars$am,levels=c(0,1),
labels=c("Automatic","Manual"))
mtcars$cyl <- factor(mtcars$cyl,levels=c(4,6,8),
labels=c("4 cylinder","6 cylinder","8 cylinder"))
```

Let's revist our original mpg regression that just used the transmission variable, **"am"**, but with a more detailed look at the statistics (and include a y-intercept this time).

```r
# Just Transmission variable, "am" (automatic/manual) with y-intercept
MPGmod000 <- lm(mpg ~ as.factor(am), data=mtcars)
summary(MPGmod000)
```

```
##
## Call:
## lm(formula = mpg ~ as.factor(am), data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3923 -3.0923 -0.2974  3.2439  9.5077
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)           17.147      1.125  15.247 1.13e-15 ***
## as.factor(am)Manual    7.245      1.764   4.106 0.000285 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.902 on 30 degrees of freedom
## Multiple R-squared:  0.3598, Adjusted R-squared:  0.3385
## F-statistic: 16.86 on 1 and 30 DF,  p-value: 0.000285
```

This model has great p-values and t-statistics, why reject it? The problem with this model is with the **"residual"** error. Although the **median residual** is great with less a third of an mpg error (-0.2974) the extremes, the **max and min residual** are almost 10 mpg! The **min residual** is -9.3923 and the max 9.5077 an almost 10 mpg error on the data we used to train the model (we would expect even worse errors with a new set of testing data). In other words this model would do awful with Toyotas and Cadilliacs only do well with very average cars.

Since, weight (**"wt"**) had the highest correlation with **mpg** (in the correlation matrix) why don't we try weight in addition to the transmission variable, **am**?

```r
# Weight is significant
MPGmod001 <- lm(mpg ~ as.factor(am)+wt, data=mtcars)
summary(MPGmod001)
```

```
##
## Call:
## lm(formula = mpg ~ as.factor(am) + wt, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.5295 -2.3619 -0.1317  1.4025  6.8782
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         37.32155    3.05464  12.218 5.84e-13 ***
## as.factor(am)Manual -0.02362    1.54565  -0.015    0.988
## wt                  -5.35281    0.78824  -6.791 1.87e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 3.098 on 29 degrees of freedom
## Multiple R-squared:  0.7528, Adjusted R-squared:  0.7358
## F-statistic: 44.17 on 2 and 29 DF,  p-value: 1.579e-09
```

The residual errors have improved across the board min, max and even median are all smaller, indicating improved prediction accuracy. **But, something strange has happened!** The weight variable, **"wt"** has great p-values and t-statistics, but the transmission variable **"am"** does not. Worse yet the transmission variable **"am"** has changed signs from positive to negative and has shrunk to almost zero with an "Estimate" of -0.02362.

A reversal of a sign when another variable is included is not uncommon in statistical research, "three statistical paradoxes that pervade epidemiological research: Simpson's paradox, Lord's paradox, and suppression. . . . Although the three statistical paradoxes occur in different types of variables, they share the same characteristic: the association between two variables can be reversed, diminished, or enhanced when another variable is statistically controlled for." "**Simpson's Paradox, Lord's Paradox, and Suppression Effects are the same phenomenon – the reversal paradox**" by Yu-Kang Tu,corresponding author David Gunnell and Mark S Gilthorpe1 http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2254615/

Recall, the purpose of this project was intended to answer the following two questions:

1. "Is an automatic or manual transmission better for MPG?"

2. "Quantify the MPG difference between automatic and manual transmissions?"

In terms of these two questions, including the weight variable **"wt"** is a disaster! In response to question #1 we could say the transmission variable was "statistically insignificant once weight was accounted for", but if pressed we could truthfully say the coefficient was "near zero", but if really pressed we would have to admit the negative sign means that the automatic transmission was better by a tiny hair of a difference!

Another approach would be to simply automate the process of variable selection with a **"stepwise regression"**. **R** has the **step()** function.

```r
# Stepwise regression
# based on example at bottom of R help(step) page
# step example used swiss data, but the example is an exact analogy.
# First we do a regression with all the variables.
modAll <- lm(mpg ~ ., data = mtcars)
# Then we feed the results of the all the variables regression to the step() function
modStep <- step(modAll)
```

```
## Start:  AIC=70.87
## mpg ~ cyl + disp + hp + drat + wt + qsec + vs + am + gear + carb
##
##        Df Sum of Sq    RSS    AIC
## - gear  2    5.1061 135.16 68.103
## - drat  1    0.9408 130.99 69.101
## - disp  1    3.4354 133.49 69.705
## - carb  1    3.9503 134.00 69.828
## - vs    1    6.5693 136.62 70.447
## - qsec  1    7.1353 137.19 70.579
## - cyl   2   16.4500 146.50 70.682
## <none>              130.05 70.870
## - am    1   14.6316 144.68 72.282
## - hp    1   22.1573 152.21 73.905
```

```
## - wt    1   23.6065 153.66 74.208
##
## Step:  AIC=68.1
## mpg ~ cyl + disp + hp + drat + wt + qsec + vs + am + carb
##
##        Df Sum of Sq    RSS    AIC
## - drat  1     0.025 135.18 66.108
## - carb  1     3.866 139.02 67.005
## - vs    1     4.035 139.19 67.044
## - disp  1     4.732 139.89 67.204
## - qsec  1     4.941 140.10 67.251
## - cyl   2    14.238 149.40 67.308
## <none>              135.16 68.103
## - am    1    15.929 151.09 69.668
## - hp    1    18.284 153.44 70.163
## - wt    1    31.992 167.15 72.901
##
## Step:  AIC=66.11
## mpg ~ cyl + disp + hp + wt + qsec + vs + am + carb
##
##        Df Sum of Sq    RSS    AIC
## - vs    1     4.250 139.43 65.099
## - carb  1     4.808 139.99 65.227
## - disp  1     4.895 140.08 65.247
## - qsec  1     4.918 140.10 65.252
## - cyl   2    17.095 152.28 65.919
## <none>              135.18 66.108
## - am    1    16.829 152.01 67.863
## - hp    1    19.891 155.07 68.501
## - wt    1    33.543 168.73 71.201
##
## Step:  AIC=65.1
## mpg ~ cyl + disp + hp + wt + qsec + am + carb
##
##        Df Sum of Sq    RSS    AIC
## - carb  1     2.898 142.33 63.757
## - disp  1     4.214 143.65 64.052
## - cyl   2    13.993 153.43 64.160
## <none>              139.43 65.099
## - qsec  1    10.717 150.15 65.469
## - am    1    14.361 153.79 66.236
## - hp    1    15.649 155.08 66.503
## - wt    1    36.334 175.77 70.510
##
## Step:  AIC=63.76
## mpg ~ cyl + disp + hp + wt + qsec + am
##
##        Df Sum of Sq    RSS    AIC
## - disp  1     1.651 143.98 62.126
## - cyl   2    11.107 153.44 62.162
## - qsec  1     8.078 150.41 63.524
## <none>              142.33 63.757
## - hp    1    15.403 157.73 65.046
## - am    1    17.424 159.75 65.453
```

```
## - wt    1    40.707 183.04 69.807
##
## Step:  AIC=62.13
## mpg ~ cyl + hp + wt + qsec + am
##
##        Df Sum of Sq    RSS    AIC
## - cyl   2    16.085 160.07 61.515
## - qsec  1     7.044 151.03 61.655
## <none>              143.98 62.126
## - hp    1    15.443 159.42 63.387
## - am    1    16.566 160.55 63.611
## - wt    1    52.932 196.91 70.145
##
## Step:  AIC=61.52
## mpg ~ hp + wt + qsec + am
##
##        Df Sum of Sq    RSS    AIC
## - hp    1     9.219 169.29 61.307
## <none>              160.07 61.515
## - qsec  1    20.225 180.29 63.323
## - am    1    25.993 186.06 64.331
## - wt    1    78.494 238.56 72.284
##
## Step:  AIC=61.31
## mpg ~ wt + qsec + am
##
##        Df Sum of Sq    RSS    AIC
## <none>              169.29 61.307
## - am    1    26.178 195.46 63.908
## - qsec  1   109.034 278.32 75.217
## - wt    1   183.347 352.63 82.790
```

```r
summary(modStep)
```

```
##
## 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 ***
## amManual      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
modStep$anova
```

```
##      Step Df    Deviance Resid. Df Resid. Dev      AIC
## 1         NA          NA        19   130.0513 70.87017
## 2 - gear  2  5.10605544        21   135.1573 68.10251
## 3 - drat  1  0.02529014        22   135.1826 66.10850
## 4   - vs  1  4.25043766        23   139.4330 65.09916
## 5 - carb  1  2.89754287        24   142.3306 63.75733
## 6 - disp  1  1.65114072        25   143.9817 62.12642
## 7  - cyl  2 16.08472969        27   160.0665 61.51530
## 8   - hp  1  9.21946935        28   169.2859 61.30730
```

The final formula produced by the **step()** function **"mpg ~ wt + qsec + am"** solves our problem. With the addition of the **"one quarter mile time"** variable, **"qsec"** both the transmission variable, **"am"** and the weight variable **"wt"** can be retained.

So, the sign of **"am"** coefficient has reversed again. By itself (with a y-intercept) **"am"** coefficient had a value of: **7.245** .

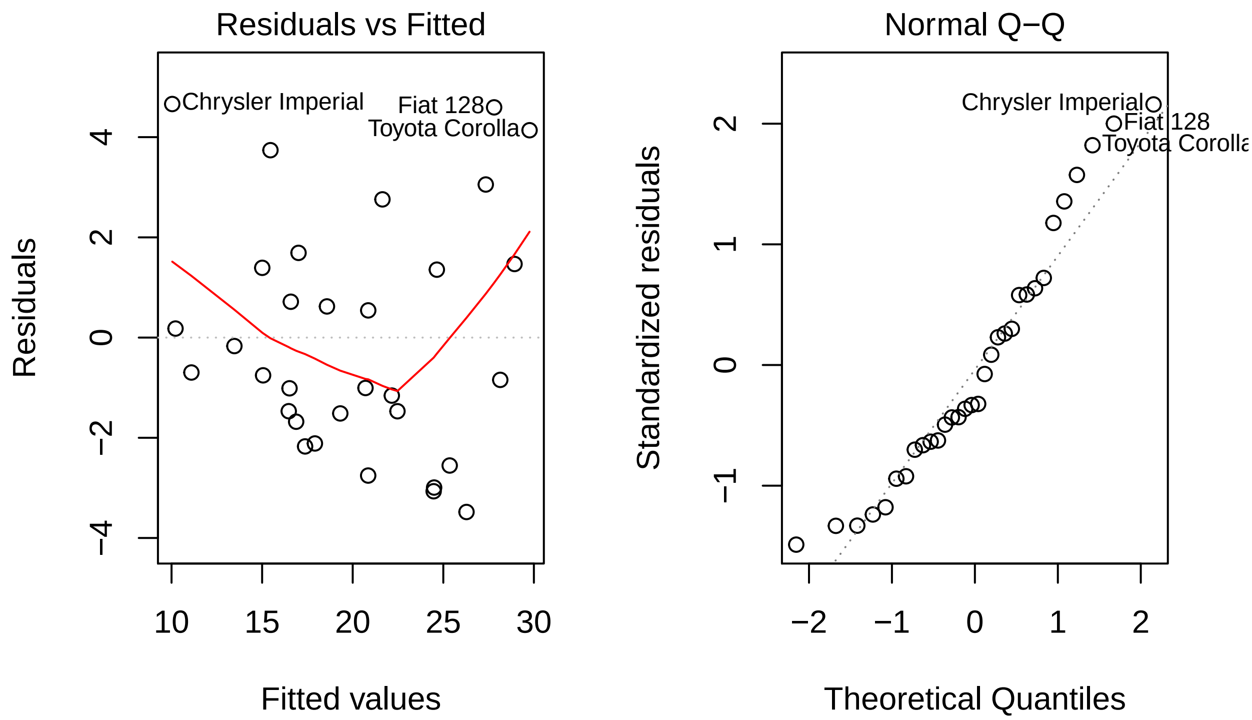
When combined with **"wt"**, the coefficient of **"am"** reversed sign and went towards zero: **-0.02362** .

Now, when both **"wt"** and **"qsec"** are included the coefficient of the transmission variable **"am"** (automatic/manual) changes back to positive and has a plausible value of: **2.9358** . This is an answer we can use, controlling for weight and how fast the car can do a quarter mile, **a standard transmision adds almost 3 mpg.**

The **"qsec"** variable is the amount of time it takes to go a quarter mile (from rest?). The qsec variable is not a speed, it is a stopwatch time like track and field (how many seconds for the hundred yard dash?). Thus, a larger **"qsec"** value means a slower speed (it took longer). For example, a 20 second quarter mile time is very slow (6 cylinder, Valiant, automatic = 20.22 seconds) and a 14 second quarter mile time is very fast (Maserati = 14.6 seconds). So, a positive **"qsec"** coefficient means that cars with larger **"qsec"** times (slower cars) get higher **mpg** and cars with smaller **"qsec"** times (faster cars) get lower **mpg**.

Let's take a look at the residual plots for the final model the **step()** function came up with: **"mpg ~ wt + qsec + am"**. First we have to re-estimate the model and then we can look at two plots side by side.

```r
par(mfrow = c(1,2))
MPGmod003 <- lm(mpg ~ wt + qsec + am, data=mtcars)
plot(MPGmod003, which = 1)
plot(MPGmod003, which = 2)
```



The labeled outliers are **Fiat 128**, **Toyota Corolla** and **Chrysler Imperial**.

Looking at the sorted list of data we printed earlier, the **Fiat 128** and the **Toyota Corolla** are high mileage cars that have higher mpg than cars with comparable weights. At the other extreme the **Chrysler Imperial** is a heavyweight car like the Cadillac Fleetwood and the Lincoln Continental (over 5,200 pounds), but has almost 50% better gas mileage (14.7 vs. 10.4 mpg).

More importantly, the theorectical quantiles versus the standardized residuals are close to the diagonal indicating that residuals are very near normal. Nearly normal residuals means that not much much information remains after subtracting our predictions from the actual data. At the corners of the Q-Q plot divergence from the 45 degree line is slightly larger perhaps suggesting that there should be more curvature or that the outlier points are distorting the fit.

To get a better picture of what is going a Google search for “ggplot2 facet examples” found a **QuickR** blog post, “**Graphics with ggplot2**” by Robert I. Kabacoff, PhD. http://www.statmethods.net/advgraphs/ggplot2.html

Because of the output of the **step()** function, I wound up not using the engine cylinders variable, **“cyl”** and so I no longer needed the faceted plot, but I was able to use or modify the other aspects of the plot:

```
library(ggplot2)
mtcars$am <- factor(mtcars$am, labels=c("Automatic","Manual"))
#
qplot(x = wt, y = mpg, data=mtcars, geom=c("point", "smooth"),
      method="lm", formula=y~x, color=am, size=qsec,
      main="Regression of MPG on Weight
controlling for Quarter Mile Time amd Transmission Type
modified from QuickR blog post by Robert I. Kabacoff",
```

```
    xlab="Weight (1,000 lbs)",
    ylab="Miles per Gallon - MPG")
```

In the graph, automatic transmissions are red and manual are blue. The size of the point is the larger the **"qsec"** variable (note the larger the **"qsec"** variable the LONGER TIME/SLOWER the time took it took the car to cover one quarter mile)

Conclusion

The effect (both magnitude and direction) of manual versus automatic transmission is very sensitive to what other variables are included in the regression.

In response to our two questions:

1. "Is an automatic or manual transmission better for MPG?"
   According to this analysis, **a manual transmission is better for MPG**.

2. "Quantify the MPG difference between automatic and manual transmissions?" *The results **ranged from 0 to 7+ mpg**, the **best answer** seems to be **about 3 mpg**.

Specifically, by itself (with a y-intercept) **"am"** coefficient had a value of: **7.245** mpg.

When combined with **"wt"**, the coefficient of **"am"** reversed sign and went towards zero: **-0.02362** mpg .

Now, when both **"wt"** and **"qsec"** are included the coefficient of the transmission variable **"am"** (automatic/manual) changes back to positive and has a plausible value of: **2.9358** mpg. So, controlling for weight and how fast the car can do a quarter mile, **a standard transmision adds almost 3 mpg (final answer).**

Bibliography

"**Regression Models for Data Science in R: A companion book for the Coursera 'Regression Models' class** by Brian Caffo (LeanPub), This version was published on 2015-08-05. The book is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License4, which requires author attribution for derivative works, non-commercial use of derivative works and that changes are shared in the same way as the original work.

R Core Team (2015). "**R: A language and environment for statistical computing**". R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/.

"mtcars" data set (as described in the documentation ** R help(mtcars)**) is from
Henderson and Velleman (1981), "**Building multiple regression models interactively.**" Biometrics, 37, 391–411. http://www.mortality.org/INdb/2008/02/12/8/document.pdf

"**ggplot2**"" R package by H. Wickham. "**ggplot2: elegant graphics for data analysis**". Springer New York, 2009. https://cran.r-project.org/package=ggplot2 http://ggplot2.org/book/

"R Graphics Cookbook" by Winston Chang (O'Reilly). Copyright 2013 Winston Chang, ISBN 978-1-449-31695-2. http://oreil.ly/R_Graphics_Cookbook http://www.cookbook-r.com/Graphs/

"**A Regression Paradox for Linear Models: Sufficient Conditions and Relation to Simpson's Paradox**"
by Aiyou CHEN, Thomas BENGTSSON, and Tin Kam HO.
The American Statistician, August 2009, Vol. 63, No. 3
Copyright 2009 American Statistical Association
http://ect.bell-labs.com/who/aychen/regressionparadox.pdf

"**Lord's Paradox Revisited { (Oh Lord! Kumbaya!)**"
by Judea Pearl
TECHNICAL REPORT R-436 October 2014
http://ftp.cs.ucla.edu/pub/stat_ser/r436.pdf

"**Simpson's Paradox, Lord's Paradox, and Suppression Effects are the same phenomenon – the reversal paradox**"
by Yu-Kang Tu,corresponding author David Gunnell and Mark S Gilthorpe1
Emerg Themes Epidemiol. 2008; 5: 2
PMCID: PMC2254615
Copyright 2008 Tu et al; licensee BioMed Central Ltd.
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2254615/

"Doing Data Science"
by Cathy O'Neil and Rachel Schutt (O'Reilly).
Copyright 2014 Cathy O'Neil and Rachel Schutt, ISBN 978-1-449-35865-5
http://oreil.ly/doing_data_science
http://mathbabe.org/

"Data analysis with Open Source Tools"
by Phillip K. Janert (O'Reilly).
Copyright 2011 Phillip K. Janert, ISBN 978-0-596-80235-6.
http://shop.oreilly.com/product/9780596802363.do
http://www.beyondcode.org/

Wikipedia
https://en.wikipedia.org/wiki/Berkson%27s_paradox
https://en.wikipedia.org/wiki/Box_plot
https://en.wikipedia.org/wiki/Collider_(epidemiology)
https://en.wikipedia.org/wiki/Confounding
https://en.wikipedia.org/wiki/Mediation_(statistics)
https://en.wikipedia.org/wiki/Simpson%27s_paradox