The Impact of Transmission Type on Fuel Economy in the mtcars Dataset

Justin Z

April 29, 2019

Overview

This report is for the peer-graded project in the Regression Models course from Johns Hopkins University within the Data Science Specialization on Coursera. The instructions say to perform exploratory data analysis and use regression models with the `mtcars` data set to answer these questions:

1. Is an automatic or manual transmission better for fuel economy
2. Quantify the impact of transmission type on fuel economy

Executive Summary

The objective was to determine the impact of transmission type on fuel economy in the `mtcars` data set. The data were loaded into R and pre-processed then exploratory data analysis was performed. The model selection phase aimed to identify the best predictive model for fuel economy so that the role of transmission type could be assessed. Model selection progressed in stages, and the best model was found to take the form `mpg ~ hp + wt + hp * wt`. Adding the `am` variable to this model did not significantly improve the model, so it was concluded that transmission type does not have a significant impact on fuel economy in this data set.

Part 1) Loading and Pre-processing the Data

In order to meet the report length requirement some text and figure output will be suppressed, when applicable this is noted at the beginning of the code chunk, and the output will be described as comments. Additionally, 3 functions were written to facilitate model selection, the code for these is not shown either. If interested, unabriged analysis can be viewed in the GitHub repo for this project submission.

The first step in the analysis loads the necessary packages and data.

```
# Messages suppressed for this code chunk
library(datasets); library(tidyverse); library(car)

data("mtcars")  # Load the data
mtcars <- as_tibble(mtcars, rownames = "vehicle")  # Put row names in a column
str(mtcars)  # Check structure, all numeric besides vehicle names, no NA values
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    32 obs. of  12 variables:
##  $ vehicle: chr  "Mazda RX4" "Mazda RX4 Wag" "Datsun 710" "Hornet 4 Drive" ...
##  $ mpg    : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl    : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp   : num  160 160 108 258 360 ...
##  $ hp     : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat   : num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt     : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec   : num  16.5 17 18.6 19.4 17 ...
##  $ vs     : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am     : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear   : num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb   : num  4 4 1 1 2 1 4 2 2 4 ...
```

Except for the vehicle names, all the variables came through as numeric, but it is clear that there is a mix of numeric, categorical, and discrete variables here. The help document can provide additional information on the descriptions, units, and coding of the factor variables. After assessing the varibles it was decided to treat `cyl` and `carb` as numeric and `gear` as categorical, so conversions occur accordingly below.

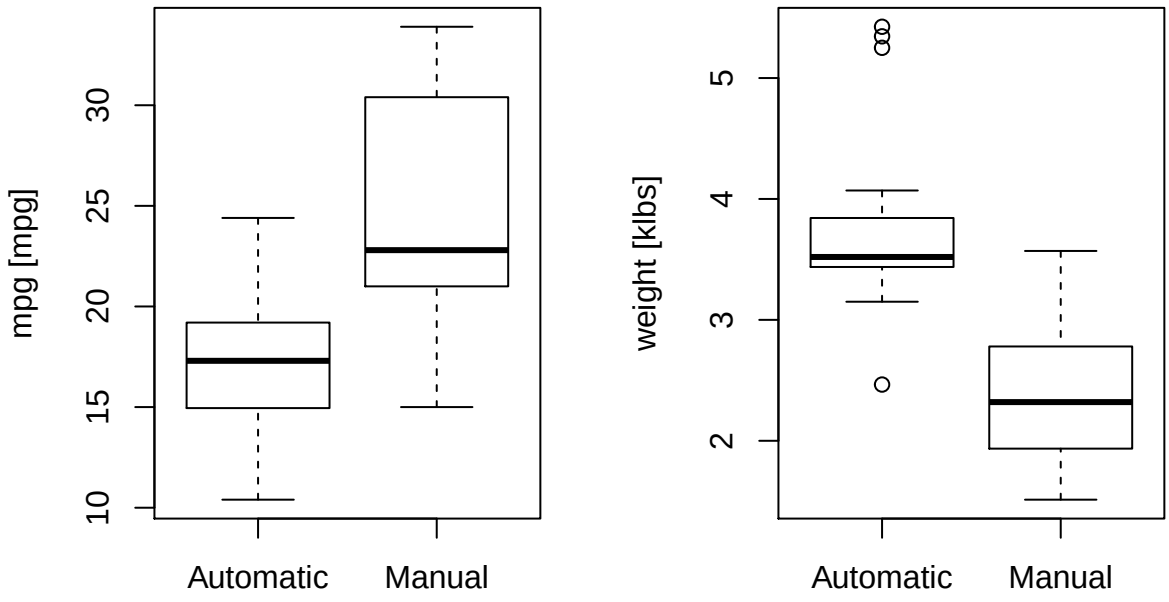
```
# Proceed with converting data types
mtcars <- mtcars %>%
  mutate(vs = as.factor(vs)) %>%
  mutate(am = as.factor(am)) %>%
  mutate(gear = as.factor(gear))

# Convert the zeros and ones of cateogrical variables
levels(mtcars$vs) <- c("V-shaped", "Straight")
levels(mtcars$am) <- c("Automatic", "Manual")
```

Part 2) Exploratory Data Analysis

Next the variation of the variables was checked, but to save space that code is not shown here, if interested the complete EDA can be found in the GitHub repo. A key portion of this project is dealing with the covariation of the variables as shown in the next code chunk.

```
par(mfrow = c(1, 2))  # Setup plot space
boxplot(mtcars$mpg ~ mtcars$am, ylab = "mpg [mpg]")  # mpg vs trans type
boxplot(mtcars$wt ~ mtcars$am, ylab = "weight [klbs]")  # wt vs. trans type
```



In the first plot above one could jump to the conclusion that `mpg` is clearly higher with manual transmission. However in the second plot one can see that manual transmission vehicles tend to be lighter in this dataset, and common sense suggests that `mpg` should be higher for lower weight vehicles. Many of these variables

can be expected to correlate with `mpg` and/or with each other. The regression model must correct for these interactions and isolate the effect of the transmission type on fuel economy. The variance inflation factors for these variables were checked, and the values were quite high, confirming that these variables are all very much related. In the next code chunk the covariations with mpg were checked.

```
# These plots are commented out to conserve space
# Check how mpg varies with the other variables
#plot(mtcars$mpg ~ mtcars$cyl)  # Negative slope, as expected
#plot(mtcars$mpg ~ mtcars$disp)  # Negative slope, to be expected
#plot(mtcars$mpg ~ mtcars$hp)  # Negative slope, to be expected
#plot(mtcars$mpg ~ mtcars$drat)  # Positively sloped, opposite of expected
#plot(mtcars$mpg ~ mtcars$wt)  # Negative slope, very much expected
#plot(mtcars$mpg ~ mtcars$qsec)  # Generally positive, faster cars consume more
#plot(mtcars$mpg ~ mtcars$vs)  # Strong relationship, but that is not expected
#plot(mtcars$mpg ~ mtcars$am)  # Strong relationship, but that is not expected
#plot(mtcars$mpg ~ mtcars$gear)  # Weak relationship, as expected
#plot(mtcars$mpg ~ mtcars$carb)  # Negative relationship, but not expected
```

A few of the interesting notes above were followed up in the analysis, and the unexpected trends disappear when accounting for the significant variables. Additional covariations were explored, but that code is not shown here. Based on the checks above and general knowledge about the variables, 17 interaction effects will be considered in the analysis (complete list below).

Part 3a) Model Selection, Round 1

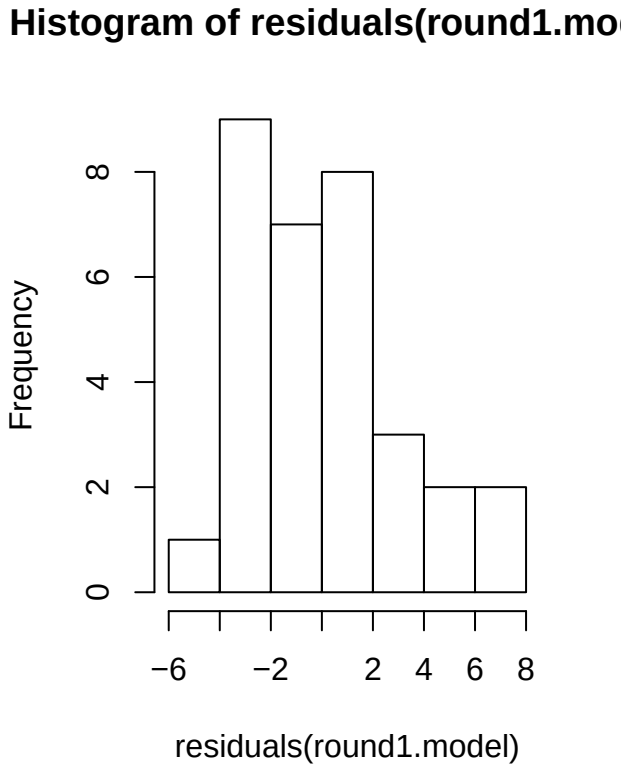
The model was built up sequentially with the idea of forward selection in mind. First the variables were checked to see which had the strongest relationship with `mpg`. To facilitate model selection a function called `FitAndSortModels` was written which takes variables as input, fits models, collects summary statistics, and sorts the models by performance. The code for this function is not shown, but its input and output is below.

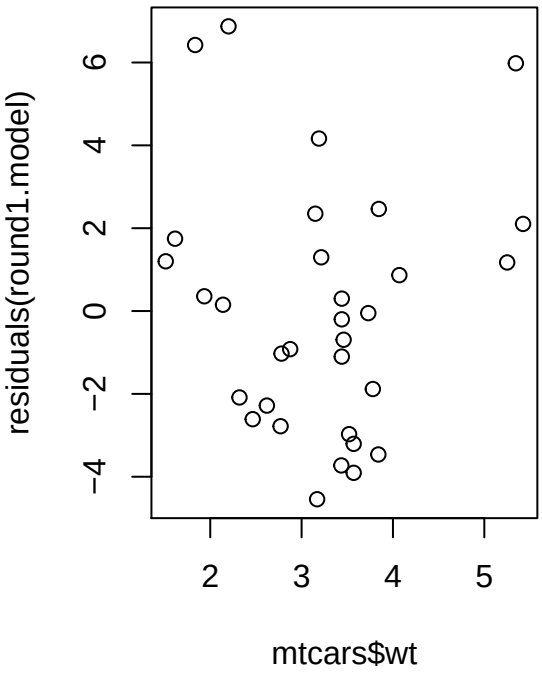
```
# First check which variable has the strongest relationship with mpg by itself
# Collect all the predictor variables into a character vector
round1.vars <- mtcars %>%
  select(-vehicle, -mpg) %>%  # Drop the non-predictor variables
  names()  # Character vector, 10 long
# Fit, summarize, and sort a series of models
round1.table <- FitAndSortModels(data = mtcars, response = "mpg",
                                 predictors.unique = round1.vars)
print(head(round1.table, n = 4))  # mpg ~ wt is the top model
```

```
## # A tibble: 4 x 5
##   model.name model.object R.Squared Ad.R.Squared model.pvalue
##   <chr>      <list>           <dbl>        <dbl>        <dbl>
## 1 mpg ~ wt   <S3: lm>         0.753        0.745     1.29e-10
## 2 mpg ~ cyl  <S3: lm>         0.726        0.717     6.11e-10
## 3 mpg ~ disp <S3: lm>         0.718        0.709     9.38e-10
## 4 mpg ~ hp   <S3: lm>         0.602        0.589     1.79e- 7
```

```
round1.model <- round1.table$model.object[[1]]  # Store the model object

# Check out the residuals of the mpg ~ wt model
par(mfrow = c(1, 2))  # Setup plot space
hist(residuals(round1.model))  # Right skew, ranges from -4.54 to 6.87
plot(residuals(round1.model) ~ mtcars$wt)  # V shape high at ends low in center
```






```
rm(round1.vars, round1.table, round1.model)
```

Due to the shape of the residuals plot above, log transforms of the wt variable were explored. To save space those details are omitted here, but they can be found in the GitHub repo. While the log transforms did improve the model very slightly, ultimately the simpler model was selected.

Part 3b) Model Selection, Round 2

The next round of model selection will consider interaction effects along with the main variables.

```
# Build a character vector of the interaction terms
round2.interactions <- c("cyl * disp", "disp * hp", "disp * wt", "disp * qsec",
                         "hp * wt", "hp * carb", "hp * qsec", "drat * wt",
                         "drat * qsec", "wt * qsec", "wt * vs", "wt * am",
                         "wt * gear", "wt * carb", "qsec * am", "qsec * gear",
                         "qsec * carb")
# Consolidate into a single character vector of variables
round2.vars <- mtcars %>%
  select(-vehicle, -mpg, -wt) %>%  # Reduce to only the repeated vars
  names() %>%  # Extract main variables
  c(round2.interactions)  # Add in interaction effects, chr vector, 26 long
# Fit, summarize, and sort a series of models
round2.table <- FitAndSortModels(data = mtcars, response = "mpg",
                                 predictors.unique = round2.vars,
                                 predictors.repeat = "wt")
print(head(round2.table, n = 4))  # mpg ~ wt + hp + hp * wt is the top model
```

```
## # A tibble: 4 x 5
##   model.name            model.object R.Squared Ad.R.Squared model.pvalue
##   <chr>                 <list>           <dbl>        <dbl>        <dbl>
## 1 mpg ~ wt + hp * wt    <S3: lm>         0.885        0.872     8.43e-14
## 2 mpg ~ wt + disp * hp  <S3: lm>         0.883        0.866     2.77e-13
## 3 mpg ~ wt + qsec * am  <S3: lm>         0.872        0.853     7.96e-13
## 4 mpg ~ wt + cyl * disp <S3: lm>         0.860        0.839     2.34e-12
```

```
round2.model <- round2.table$model.object[[1]]  # Store the model object
rm(round2.interactions, round2.vars, round2.table)
```

The code chunk above shows that `mpg ~ wt + hp + hp * wt` is the top performing model from this round. `anova` was checked for this model, and it suggested that all terms should be kept in the model. The residuals were checked as well, and they looked good, like random noise.

An additional round of model selection was performed where 24 models were fit - each with between one and three additional terms on top of the round 2 model. None of the additions improved the model enough to pass an `anova` test. Now that the model has been identified the original questions can be addressed.

Part 4) Conclusions

The assignment instructions ask:

1. Is an automatic or manual transmission better for fuel economy
2. Quantify the impact of transmission type on fuel economy

This is somewhat of a trick question since the data suggest that the type of transmission does not have a significant impact on fuel economy. This can be shown by adding `am` to the model and checking `anova`.

```
part4.model <- lm(mpg ~ hp + wt + hp * wt + am, data = mtcars)
print(summary(part4.model)$coefficients)  # coef of amManual is 0.13, pvalue is 0.9259
```

```
##                Estimate  Std. Error     t value     Pr(>|t|)
## (Intercept) 49.45224079 5.280730731  9.36465866 5.694894e-10
## hp          -0.11930318 0.026549992 -4.49352965 1.187315e-04
## wt          -8.10055755 1.789325217 -4.52715777 1.084926e-04
## amManual     0.12510693 1.333430965  0.09382333 9.259423e-01
## hp:wt        0.02748826 0.008472529  3.24439879 3.130390e-03
```

```
print(anova(round2.model, part4.model))  # p-value 0.93
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt + hp * wt
## Model 2: mpg ~ hp + wt + hp * wt + am
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     28 129.76
## 2     27 129.72  1  0.042292 0.0088 0.9259
```

The code chunk above shows that once the `wt` and `hp` variables are controlled for that the impact of `am` is both small and insignificant, and with a pvalue of 0.9259 there is little uncertainty about this. If one includes the variable anyways there appears to be a 0.13 mpg improvement when using a manual transimission instead of automatic, but this observation is very likely due to chance.