Relationship between variables and miles per gallon

## Executive Summary

### Task

You work for Motor Trend, a magazine about the automobile industry. Looking at a data set of a collection of cars, they are interested in exploring the relationship between a set of variables and miles per gallon (MPG) (outcome). They are particularly interested in the following two questions:

* “Is an automatic or manual transmission better for MPG”
* “Quantify the MPG difference between automatic and manual transmission”

### Analysis

#### Data description and summary

The first step of our analysis is to load the mtcard dataset and check its data structure.

data("mtcars")  
str(mtcars)

## 'data.frame': 32 obs. of 11 variables:  
## $ 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 ...

For a clear overview following a data description extracted from `?mtcars':

|  |  |
| --- | --- |
| Var | Descripton |
| mpg | Miles/(US) gallon |
| disp | Displacement (cu.in.) |
| cyl | Number of cylinders |
| hp | Gross horsepower |
| drat | Rear axle ratio |
| wt | Weight (lb/1000) |
| qsec | 1/4 mile time |
| vs | V/S |
| am | Transmission (0 = automatic, 1 = manual) |
| gear | Number of forward gears |

Now we will focus on study the relation between the type of transmission and the number of miles per gallon, am (0 = automatic and 1 = manual) and mpg variables respectively.

#### Exploration Data analysis

We will plot some relations between the transmission am and miles per gallon mpg in order to identify it help us to identify any patterns. We will work with ggplot2 and GGally library so we need to load both.

library(ggplot2)  
library(GGally)

Our first figure will show the values of MPG related to manual and automatic cars but first we will add factor and label it as automatic and manual transmission.

mtcars$trans <- factor(mtcars$am, labels = c("automatic", "manual"))

plot1 <- ggplot(mtcars, aes(x=mpg, fill = factor(mtcars$trans))) + theme\_bw() +  
 geom\_bar( binwidth = 1, col = 'black', position="dodge") + labs( fill = 'Transmission') +   
 labs( title = "Fig 1. Miles per gallon", x = "MPG")

From the Fig 1. (attached to the appendix) we can identify that manual cars have a higher values on miles per gallon.

Following the Fig 2. a boxplot for miles per gallon and transmission.

plot2 <- ggplot(mtcars, aes(x=factor(mtcars$trans), y=mpg, fill = factor(mtcars$trans))) +  
 geom\_boxplot(adjust = 1) + geom\_jitter(size = 3) + labs( fill = 'Transmission') +   
 labs( title = 'Fig 2. MPG for transmission', x = 'Transmission', y = 'MPG')

As you can see in both figures attached in the appendix, seems that there is some pattern between MPG and Transmission, anyway this patterns can not be explained just with this evidences. We have to check if it is related to other variables. In order to check this relations we can proceed by plot correlation heatmap which show us in which measure the variables are correlated.

heatmap(cor(mtcars[1:11]), main= "mtcars dataset heatmap correlation")

From the previous plot Fig 3. a heatmap showing correlation (also attached to the appendix), we notice that there is very little correlation between transmission and miles per gallon, variables 'am' and 'mpg' respectively.

#### Regression models

In this section we will study a simple model and a multivariable model.

##### Simple model

Following we will give an estimation of the effect of transmission on miles per gallon. In order to do that we will build a linear model and compute the confidence interval.

simple\_model <-lm(mpg ~ trans, data=mtcars)  
model\_coef <- summary(simple\_model)$coefficients  
model\_coef

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 17.147368 1.124603 15.247492 1.133983e-15  
## transmanual 7.244939 1.764422 4.106127 2.850207e-04

And the interval:

interval\_simple <- model\_coef[2,1] + c(-1,1) \* qt(0.975, df = simple\_model$df) \* model\_coef[2,2]  
interval\_simple

## [1] 3.64151 10.84837

From the above results we obtained a p-value: 2.850207410^{-4} and an interval of [3.6415096, 10.848369].

Considering the model and the results, compared both transmisions we can conclude that a manual transmission increases on average 7.2449393 miles per gallon over the automatic transmission and the 95% confidence interval is [3.6415096, 10.848369]

For complet the analysis in a simple model we can plot the residuals and check for heteroskedasticity that suggest the presence of more variables.

plot3 <- ggplot(mtcars, aes(x = trans, y = resid(simple\_model), fill = trans)) +  
 theme\_bw() + geom\_boxplot(adjust = 1) + geom\_jitter(size = 3) +  
 labs( title = 'Fig 4. Residuals - Simple Model', x = 'Transmission', y = 'Residuals')

From the result shown in the appendix we can observe a large variability for manual transmission and a possible heteroskedasticity case, presence of large variance in the model. Will be necessary to consider a multivariable model.

We can ensure the presence of heteroskedasticity by checking studentized Breusch-Pagan test. For this purpose we will need to load lmtest library.

library(lmtest)

## Loading required package: zoo  
##   
## Attaching package: 'zoo'  
##   
## The following objects are masked from 'package:base':  
##   
## as.Date, as.Date.numeric

bptest(simple\_model)

##   
## studentized Breusch-Pagan test  
##   
## data: simple\_model  
## BP = 5.0771, df = 1, p-value = 0.02424

The presence of low p-value bptest(simple\_model)$p.value give us strong evidences of possible hidden variables so we will proceed with a multivariable model.

##### Multivariable model

Following the multivariable study:

multi\_model <- lm(mpg ~ ., data = mtcars)  
mmodel\_coef <- summary(multi\_model)$coefficients  
mmodel\_coef

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 12.30337416 18.71788443 0.6573058 0.51812440  
## cyl -0.11144048 1.04502336 -0.1066392 0.91608738  
## disp 0.01333524 0.01785750 0.7467585 0.46348865  
## hp -0.02148212 0.02176858 -0.9868407 0.33495531  
## drat 0.78711097 1.63537307 0.4813036 0.63527790  
## wt -3.71530393 1.89441430 -1.9611887 0.06325215  
## qsec 0.82104075 0.73084480 1.1234133 0.27394127  
## vs 0.31776281 2.10450861 0.1509915 0.88142347  
## am 2.52022689 2.05665055 1.2254035 0.23398971  
## gear 0.65541302 1.49325996 0.4389142 0.66520643  
## carb -0.19941925 0.82875250 -0.2406258 0.81217871

One approach in this multiple model is to perform a backwards elimination, start with all the predictors in the model and remove the variable with the higher P value. We will perform this process to removes the less significant variable till all values are smaller than an specific value in this case 0.05

Following the iterative process:

dat <- mtcars[, c("mpg", "wt", "qsec", "am")]  
fit <- lm(mpg ~ . - 1, data = dat)  
summary(fit)$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## wt -3.185455 0.4827586 -6.598442 3.128844e-07  
## qsec 1.599823 0.1021276 15.664944 1.091522e-15  
## am 4.299519 1.0241147 4.198279 2.329423e-04

dat <- mtcars[,1:11]  
dat <- dat[, names(dat) != "cyl"]  
summary(lm(mpg ~ ., data = dat))$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 10.96007405 13.53030251 0.8100391 0.42659327  
## disp 0.01282839 0.01682215 0.7625891 0.45380797  
## hp -0.02190885 0.02091131 -1.0477031 0.30615002  
## drat 0.83519652 1.53625251 0.5436584 0.59214373  
## wt -3.69250814 1.83953550 -2.0073046 0.05715727  
## qsec 0.84244138 0.68678068 1.2266527 0.23291993  
## vs 0.38974986 1.94800204 0.2000767 0.84325850  
## am 2.57742789 1.94034563 1.3283344 0.19768373  
## gear 0.71155439 1.36561933 0.5210489 0.60753821  
## carb -0.21958316 0.78855537 -0.2784626 0.78325783

dat <- dat[, names(dat) != "vs"]  
summary(lm(mpg ~ ., data = dat))$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 9.76827789 11.89230469 0.8213949 0.41985460  
## disp 0.01214441 0.01612373 0.7532010 0.45897019  
## hp -0.02095020 0.01992567 -1.0514175 0.30398892  
## drat 0.87509822 1.49112525 0.5868710 0.56300717  
## wt -3.71151106 1.79833544 -2.0638592 0.05049085  
## qsec 0.91082822 0.58311935 1.5619928 0.13194532  
## am 2.52390094 1.88128007 1.3415870 0.19282690  
## gear 0.75984464 1.31577205 0.5774896 0.56921947  
## carb -0.24796312 0.75933250 -0.3265541 0.74695821

dat <- dat[, names(dat) != "carb"]  
summary(lm(mpg ~ ., data = dat))$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 9.19762837 11.54220381 0.7968693 0.433339841  
## disp 0.01551976 0.01214235 1.2781513 0.213420001  
## hp -0.02470716 0.01596302 -1.5477746 0.134763097  
## drat 0.81022794 1.45006779 0.5587518 0.581507634  
## wt -4.13065054 1.23592980 -3.3421401 0.002717119  
## qsec 1.00978651 0.48883274 2.0657097 0.049814778  
## am 2.58979984 1.83528342 1.4111171 0.171042438  
## gear 0.60644020 1.20596266 0.5028681 0.619640616

dat <- dat[, names(dat) != "gear"]  
summary(lm(mpg ~ ., data = dat))$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 10.71061639 10.97539399 0.9758753 0.338475309  
## disp 0.01310313 0.01098299 1.1930387 0.244054196  
## hp -0.02179818 0.01465399 -1.4875257 0.149381426  
## drat 1.02065283 1.36747598 0.7463772 0.462401185  
## wt -4.04454214 1.20558182 -3.3548467 0.002536163  
## qsec 0.99072948 0.48002393 2.0639168 0.049550895  
## am 2.98468801 1.63382423 1.8268110 0.079692318

dat <- dat[, names(dat) != "drat"]  
summary(lm(mpg ~ ., data = dat))$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 14.36190396 9.74079485 1.474408 0.152378367  
## disp 0.01123765 0.01060333 1.059823 0.298972150  
## hp -0.02117055 0.01450469 -1.459565 0.156387279  
## wt -4.08433206 1.19409972 -3.420428 0.002075008  
## qsec 1.00689683 0.47543287 2.117853 0.043907652  
## am 3.47045340 1.48578009 2.335779 0.027487809

dat <- dat[, names(dat) != "disp"]  
summary(lm(mpg ~ ., data = dat))$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 17.44019110 9.3188688 1.871492 0.072149342  
## hp -0.01764654 0.0141506 -1.247052 0.223087932  
## wt -3.23809682 0.8898986 -3.638726 0.001141407  
## qsec 0.81060254 0.4388703 1.847021 0.075731202  
## am 2.92550394 1.3971471 2.093913 0.045790788

dat <- dat[, names(dat) != "hp"]  
summary(lm(mpg ~ ., data = dat))$coefficients

## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 9.617781 6.9595930 1.381946 1.779152e-01  
## wt -3.916504 0.7112016 -5.506882 6.952711e-06  
## qsec 1.225886 0.2886696 4.246676 2.161737e-04  
## am 2.935837 1.4109045 2.080819 4.671551e-02

complex\_model <- lm(mpg ~ . - 1, data = dat)  
model\_coef2 <- summary(complex\_model)$coefficients  
model\_coef2

## Estimate Std. Error t value Pr(>|t|)  
## wt -3.185455 0.4827586 -6.598442 3.128844e-07  
## qsec 1.599823 0.1021276 15.664944 1.091522e-15  
## am 4.299519 1.0241147 4.198279 2.329423e-04

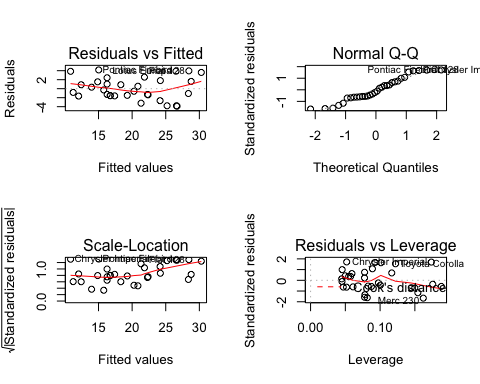
As is shown above we removed the unrelated variables one by one, the final model just contains wt,qsec and am. We can conclude that the predicting model for the miles per gallon of a vehicle is y = -3.185455wt + 1.599823qsec + 4.299519am

interval\_complex <- model\_coef2[2,1] + c(-1,1) \* qt(0.975, df = complex\_model$df) \* model\_coef2[2,2]  
interval\_complex

## [1] 1.390948 1.808697

The adjusted R-squared for the model is 0.9857902, which is satisfying. And the adjusted 95% confidence interval is 1.3909482, 1.8086969.

par(mfrow=c(2, 2))  
plot(complex\_model)



#### Conclusions

In the following study we are trying to answer two questions:

* “Is an automatic or manual transmission better for MPG”
* “Quantify the MPG difference between automatic and manual transmission”

The study has shown that there is no correlation between the variables am and mpg that justify that we cannot answer this questions without consider other relevant variables as wt and qsec, weigh and 1/4 mile time for explain the mile per gallon consumption.

The better aproximation to both answer is to evaluate the function y = -3.185455wt + 1.599823qsec + 4.299519am

## Appendix

