Regression Models Course Project of Motor Trend

Relationship between MPG and transmissions

JackHo | 4 June 2017

Executive summary

The target of this analysis lies on finding the relationship between a set of variables and miles per gallon (MPG) and the company is particularly interested in **"Is an automatic or manual transmissions better for MPG"** and **"Quantify the MPG difference between automatic and manual transmissionss"**, I will firstly refine the data with only keeping the `am` and `mpg` variables, then try to add more predictors into the model to improve it. In the end, I got a model: $mpg = 12.0638 \times am - 3.5641 \times wt + 0.9835 \times qesc - 3.5799 \times am \times wt + 12.4791$ which r.square value is equal to 90.4% data and is overall significant (p.value = 7.335181e-13).

1. Data Preparation

Load essential packages.

```
library(MASS);library(car);library(dplyr);library(ggplot2);library(knitr);library(broom);
mt_cars <- mt_cars %>% dplyr:: select(mpg,am)
```

2. Data Sleuthing

The variable `am` is a binary variable, which represents the status of the transmissions (0 = automatic, 1 = manual) of every cars. While the `mpg` variable means the miles/galon (US). With `tapply(X = mt_cars$mpg,INDEX = mt_cars$am,length),` there are 13 cars have the manual transmissions and 19 cars have the automatic one.

It can be concluded that automatic cars are more petrol-efficient than manual cars. With `tapply(X = mt_cars$mpg,INDEX = mt_cars$am,summary),` *for those cars have the automatic transmissions (`am` = 0), the min(mpg) is 10.40/galon (US), the max(mpg) is 24.40/galon (US) and the mean(mpg) is 17.15/galon (US). While, the values of mpg of cars with manual transmissions (`am` = 1) are larger than those of mpg of automatic cars. (Such comparison can also be perceived from the Appendix 1)*.

3. Build the Linear Model

Firstly, let's build a one-variable linear model, where mpg is a the outcome and the am is independent variable.

```
mt_cars$am <- as.factor(mt_cars$am)
model <- lm(data = mt_cars, mpg~ am)
```

From the `model`, we could basically get the formul: $mpg = 7.244939 \times am + 17.147368$, which means the manual cars cost more *7.244949*/galon (US) in average of mpg than automatic cars do. The `p.value` is *0.0002850207* shows that the model is credible. However, from the value of the `r.squared` and its adjusted value (*0.3597989* and *0.3384589*), the model can only explain about 35% of dependent data. Thus, I decide to add more variables in order to improve the model.

4. Improve the model and Diagnose It

Next, after adding more predictors into the model, the results show that the overall model is significant (3.793152e-07) and the r-squared value is high (87%), but many independent variables are non-significant and I guess this might be because of the multicollinearity problem.

```
mt_cars <- datasets::mtcars
mt_cars$am <- as.factor(mt_cars$am)
model <- lm(data = mt_cars,mpg~.)
```

Then, to figure out whether I add in some real-unnecessary variables, I decide to apply the `stepAIC()` function to help me filter the useful variables.

```
model1 <- stepAIC(object = model, direction = "both")
```

Now, based on the new model, I get a new formula: $mpg = -3.916504 \times wt + 1.225886 \times qsec + 2.935837 \times am + 9.617781$, which means with the `wt` and `qsec` remained, manual cars cost more *2.935837*/galon (US) in average of mpg than automatic cars do. Meanwhile, the new model shows the coefficients of variables are significant, explains about 85% original data and it is a significant model (`p.value` = *1.210446e^-11*).

Although the statics show that the new model is quite good, there are some essential diagnostics should be processed in order to make sure the model is really creidble.

- Outliers: (Refer to the Appendix 2)

```
outlierTest(model1)
##                     rstudent unadjusted p-value Bonferonni p
## Chrysler Imperial 2.323119           0.027949      0.89437

hatvalues(model1)[order(hatvalues(model1),decreasing = T)] %>% head(4)

##  Merc 230   Lincoln Continental   Chrysler Imperial Cadillac Fleetwood
##  0.2970422      0.2642151             0.2296338       0.2270069
```

With the `outlierTest()`, the car "Chrysler Imperial" has been identified as an outlier, and with the `hatvalues()` "Chrysler Imperial" is also in the influence list. Thus, looks the record "Chrysler Imperial" should be excluded.

```
new_model <- lm(data= mt_cars[-which(row.names(mt_cars) == "Chrysler Imperial"),], mpg~.)
%>% stepAIC()
```

In model without "Chrysler Imperial", the model is more significant and can explain more data. However, the variable am is not significant anymore. Let's see whether it should be excluded.

```
anova(update(new_model,.~.-am),new_model)
## Model 1: mpg ~ wt + qsec
## Model 2: mpg ~ wt + qsec + am
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     28 154.94
## 2     27 141.09  1    13.858 2.6521  0.115

summary(update(new_model,.~.-am))$r.squared - summary(new_model)$r.squared

## [1] -0.01264364
```

Although the `anova()` tell us that even the `am` is excluded, the model will not be affected a lot, to keep a higher r.squared value, I decide not to delete it.

To improve the model, I try to interact the am, wt and qsec.

```
model_int <- update(new_model,.~.+ am:wt + am:qsec) %>% stepAIC()

anova(new_model,model_int)
## Model 1: mpg ~ wt + qsec + am
```

```
## Model 2: mpg ~ wt + qsec + am + wt:am
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     27 141.09
## 2     26 105.04  1    36.046 8.9224 0.006073 **
```

```
model_int %>% summary() %>% glance()
```

```
##   r.squared adj.r.squared    sigma statistic      p.value df
## 1 0.9041659     0.8894222 2.009966  61.32557 7.335181e-13  5
```

Finally, the model could explain *90.4%* data and is overall significant (p.value = *7.335181e-13*).

- Multicollinearity:

Owning to the help of `stepAIC()`, I exclude several of vairables at the beginning, now let's check whether the `model_int` got that issue:

```
sqrt(vif(model_int))>2
```

```
##    wt  qsec    am wt:am
## FALSE FALSE  TRUE  TRUE
```

```
tmp <- mtcars %>% dplyr::select(wt,am,qsec) %>% mutate(wt_am = wt*am)
kappa(cor(tmp))
```

```
## [1] 110.0319
```

`vif()` shows the model has certain multicollinearity issue, but calculating the `kappa()` shows that the model does not have to many issues on severe multicollinearity--- the kappa value is `110`, which represents very slight multicollinearity (kappa_value < 100: no multicollinearity; 100 < kappa_value < 1000: a fair multicollinearity, but can be kept; kappa_value > 1000: must be handled).

Thus, the final model does not have strong multicollinearity issue and all the current predictors will be kept.

- Heteroskedasticity: (Refer to the Appendix 2)

The dots in the graph "Check the Heteroskedasticity" have no particular pattern, which means this model does not have non-linear relationships and no heteroskedasticity.

- Normality: (Refer to the Appendix 2)

The plot "Check the normality of the new model" shows the residuals are basically normally distributed because most of dots are are within the dashed red intervals, indicating that the residuals follow a normal distribution.

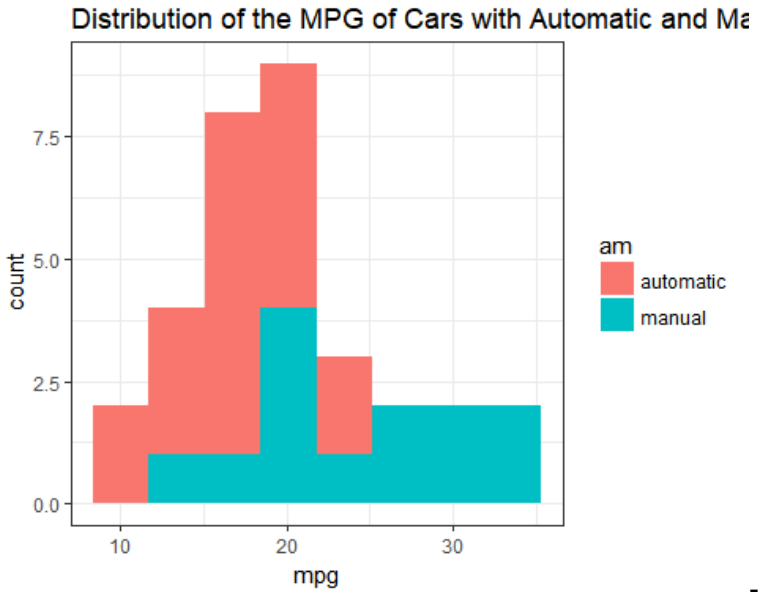
5. Summary

The `mpg` can be explained by `am`, `wt` and `qesc`, and their relationship can be denoted as: $mpg = 12.0638 \times am - 3.5641 \times wt + 0.9835 \times qesc - 3.5799 \times am \times wt + 12.4791$, which means when the `wt` and `qesc` remained, automatic cars will save more *8.4839* (12.0638-3.5799) /galon (US) petrol than manual cars.

Appendix

- Appendix 1

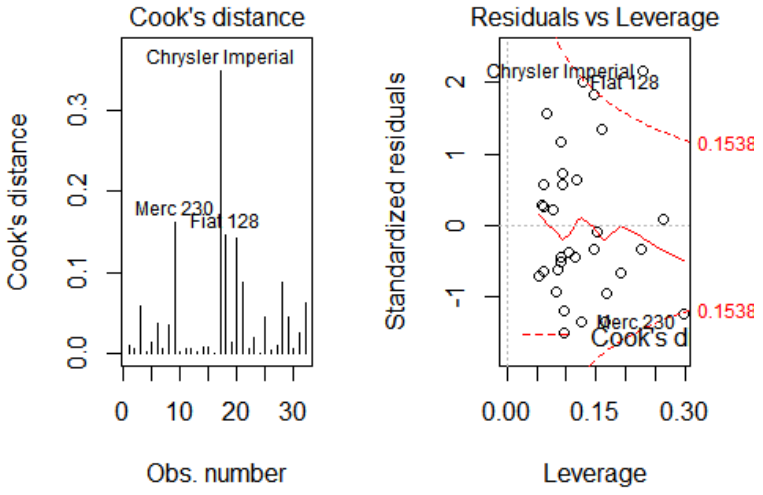
```
ggplot(data = mt_cars %>% mutate(am =factor(mt_cars$am,levels = c("0","1"),labels =
c("automatic","manual"))), aes(x=mpg,fill=am)) + geom_histogram(bins = 8) + ggtitle(label
= "Distribution of the MPG of Cars with Automatic and Manual transmissions") + theme_bw()
```



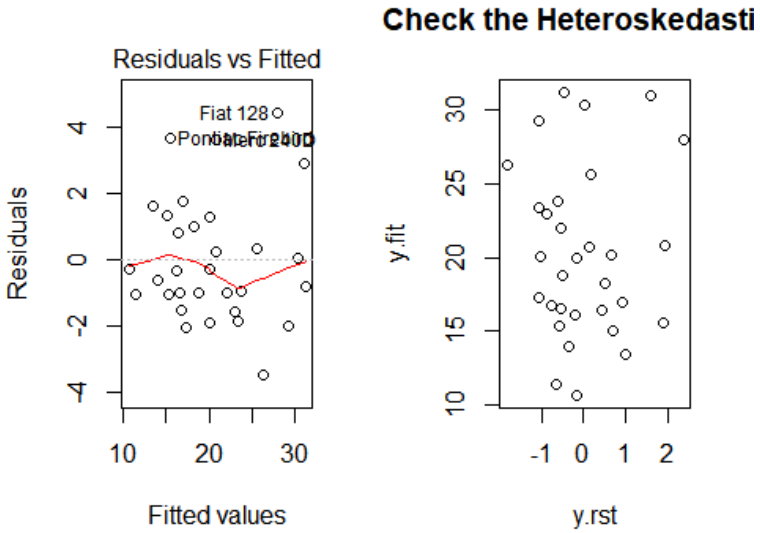
-

- Appendix 2

```
par(mfrow=c(1,2))
cutoff <- 4/((nrow(mtcars)-length(model1$coefficients)-2))
plot(model1, which=c(4,5), cook.levels=cutoff)
```




```
plot(model_int,which = 1)
y.rst<-rstandard(model_int)
y.fit<-predict(model_int)
plot(y.fit~y.rst,main="Check the Heteroskedasticity")
```




```
par(mfrow=c(1,1))
qqPlot(model1, main = "Check the normality of the new model")
```