Analysis of different MPG between automatic and amnual transmission

Charles

August 21, 2017

Executive Summary

This analysis focuses on the differences of MPG between automatic and manual transmissions. The paper firstly set environment, load data and do a exploratory data analysis to see that there is a difference of MPG. Then try to fit a simple linear model that only use 'am' as regressor. The result shows that there's a obvious difference of MPG between two groups, but the model does not fit very well. Later on the paper selects two more regressors ('wt' and 'sqec') according to the correlation and fit a new multi-variable linear model. The result shows the same that manul transmission has a better MPG than automatic and the model fits pretty good. In the end, the paper did residual analysis.

Exploratory Data Analysis

Firstly set working environment and load data:

```
setwd("C:/Study/Coursera/1 Data-Science/2 RStudio/7 Class 7/Coursera_DataScience_Class7Regression_Final
library(UsingR)
library(ggplot2)
library(dplyr)
library(grid)
library(gridExtra)
data(mtcars)
```

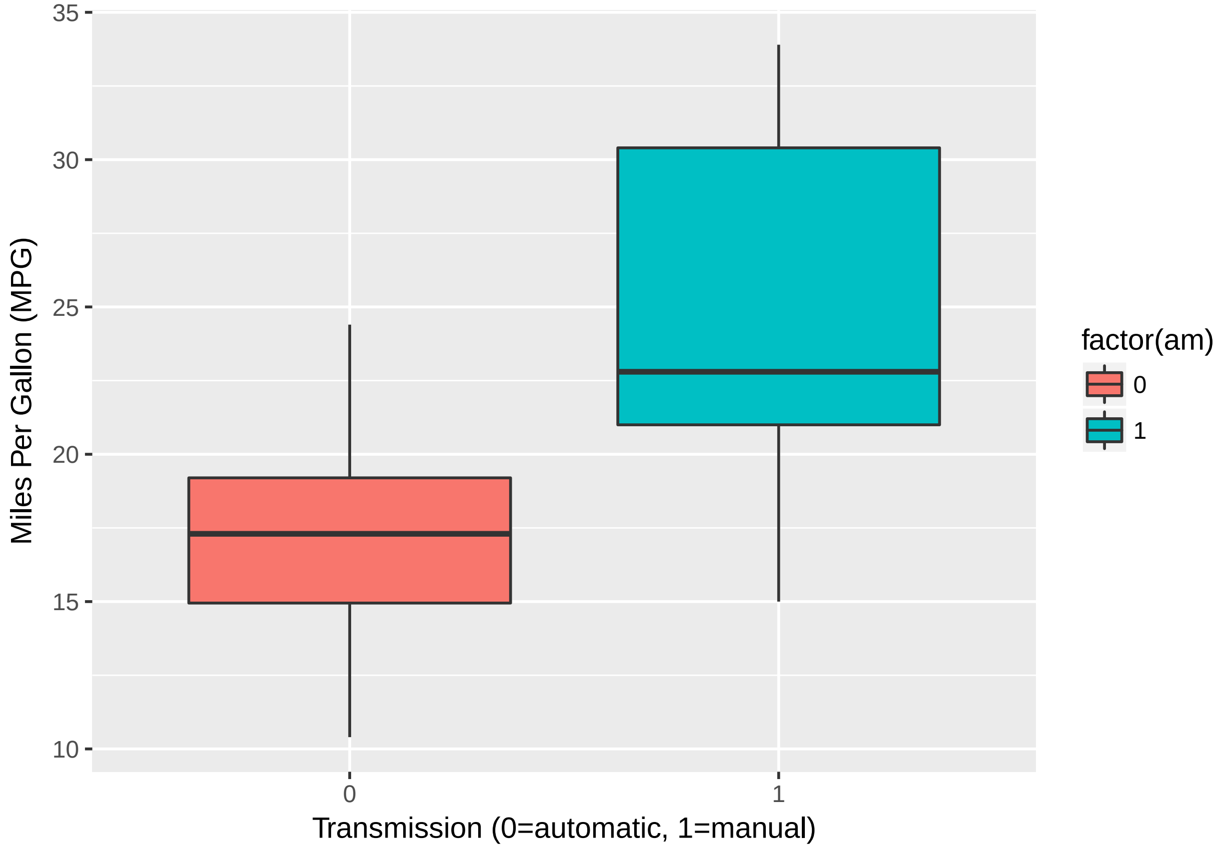
Then have a brief idea about data:

```
head(mtcars)
```

```
##                    mpg cyl disp  hp drat    wt  qsec vs am gear carb
## Mazda RX4         21.0   6  160 110 3.90 2.620 16.46  0  1    4    4
## Mazda RX4 Wag     21.0   6  160 110 3.90 2.875 17.02  0  1    4    4
## Datsun 710        22.8   4  108  93 3.85 2.320 18.61  1  1    4    1
## Hornet 4 Drive    21.4   6  258 110 3.08 3.215 19.44  1  0    3    1
## Hornet Sportabout 18.7   8  360 175 3.15 3.440 17.02  0  0    3    2
## Valiant           18.1   6  225 105 2.76 3.460 20.22  1  0    3    1
```

In order to see the difference mpg between different transmissions, draw a boxplot:

```
g <- ggplot(mtcars, aes(x=factor(am), y=mpg, fill = factor(am))) +
        geom_boxplot() +
        xlab('Transmission (0=automatic, 1=manual)') +
        ylab('Miles Per Gallon (MPG)')
g
```



The x-axis is the transmission. 0 stands for automatic and 1 stands for manual. The y-axis is Miles Per Gallon (MPG). As we can see above, it's quite obvious that there is a difference between different transmissions.

Fit a simple linear model

In order to figure out the relationship between MPG and Transmissions, the most straight forward way is to get a simple linear regression between these two and have a look at the coefficients and R^2.

```
fitAm <- lm(mpg ~ am, data = mtcars)
summary(fitAm)$coef
```

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

```
summary(fitAm)$adj.r.squared
```

```
## [1] 0.3384589
```

The intercept coefficent stands for the MPG of automatic cars (regressor=0). The am coefficient stands for MPG increase for unit increase of manual cars. R^2 is small so this linear model does not fit quite well. Try to get a confidence interval:

```
alpha <- 0.05
n <- length(mtcars)
pe <- coef(summary(fitAm))["am", "Estimate"]
se <- coef(summary(fitAm))["am", "Std. Error"]
```

```
tvalue <- qt(1 - alpha/2, n - 2)
pe + c(-1, 1) * (se * tvalue)
```

```
## [1]  3.25354 11.23634
```

We can see above the confidence interval doesn't include 0. So we can reject the null hypothesis in favor of the alternative one that there is a significant difference of MPG between two groups of transmissions at alpha=0.5.

Fit a complex linear model

We could select a multi-variable linear model. Firstly fit a linear model for all variables:

```
fitAll <- lm(mpg ~ ., data = mtcars)
summary(fitAll)$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
```

Then calculate the correlation:

```
corCars <- cor(mtcars)
data.frame(Cor.mpg = corCars[,which(names(mtcars)=='mpg')], Cor.wt = corCars[,which(names(mtcars)=='wt'
```

```
##         Cor.mpg     Cor.wt
## mpg   1.0000000 -0.8676594
## cyl  -0.8521620  0.7824958
## disp -0.8475514  0.8879799
## hp   -0.7761684  0.6587479
## drat  0.6811719 -0.7124406
## wt   -0.8676594  1.0000000
## qsec  0.4186840 -0.1747159
## vs    0.6640389 -0.5549157
## am    0.5998324 -0.6924953
## gear  0.4802848 -0.5832870
## carb -0.5509251  0.4276059
```

The selection of regressors follow the rules:

- Select the variable most correlated to MPG. In this case is 'wt'.
- Select the variable that is least correlated to 'wt'. In this case is 'qsec'.
- Add the selected variables together.

So we get our regressors: 'wt', 'qsec', 'am'. Then fit the multi-variable linear regression model with selected variables:

```r
fitNew <- lm(mpg ~ am + qsec + wt, data = mtcars)
summary(fitNew)$coef
```

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

```r
summary(fitNew)$adj.r.squared
```

```
## [1] 0.8335561
```

As we can see above, the linear model fits quite well. Use nested model to test:

```r
fit1 <- lm(mpg ~ wt, data = mtcars)
fit2 <- update(fit1, mpg ~ wt + qsec)
fit3 <- update(fit1, mpg ~ wt + qsec + am)
anova(fit1,fit2,fit3)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ wt
## Model 2: mpg ~ wt + qsec
## Model 3: mpg ~ wt + qsec + am
##   Res.Df    RSS Df Sum of Sq       F    Pr(>F)
## 1     30 278.32
## 2     29 195.46  1    82.858 13.7048 0.0009286 ***
## 3     28 169.29  1    26.178  4.3298 0.0467155 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

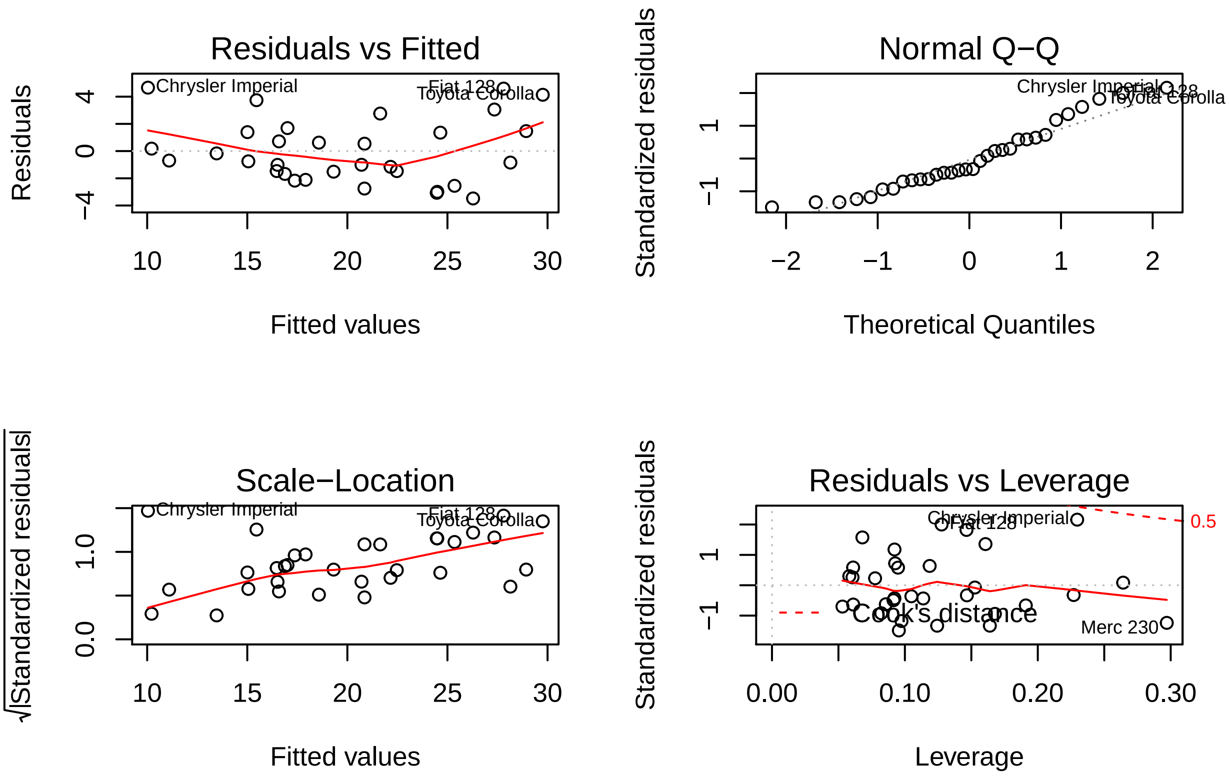
As we can see above, all the variables are significant important. Calculate confidence interval again:

```r
pe <- coef(summary(fitNew))["am", "Estimate"]
se <- coef(summary(fitNew))["am", "Std. Error"]
tvalue <- qt(1 - alpha/2, n - 2)
pe + c(-1, 1) * (se * tvalue)
```

```
## [1] -0.2558506  6.1275249
```

So the CI still don't contain 0, which means our previous conclusion holds. Then we could do residual analysis:

```r
par(mfrow = c(2,2))
plot(fitNew)
```


Residuals vs Fitted
Residuals
Fitted values
Chrysler Imperial
Fiat 128
Toyota Corolla
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Chrysler Imperial
Fiat 128
Toyota Corolla
Scale−Location
√|Standardized residuals|
Fitted values
Chrysler Imperial
Fiat 128
Toyota Corolla
Residuals vs Leverage
Standardized residuals
Leverage
Chrysler Imperial
Fiat 128
Merc 230
Cook's distance
0.5