Regression analysis on the impact of transmission on miles per gallon

Executive Summary:

The exploratory analysis, and the multivariable linear regression model support the conclusion that the cars with manual transmission are likely to be more efficient than automatic by roughly ~**1.81** more miles per gallon (expected 95% confidence interval difference is -1.06 to 4.68 miles per gallon) , under our bestfit model (which explains 84% of the variance). Since our bestfit model contains 0, we can't totally neglect the idea that sometime an automatic transmission car can outperform a manual one. Approach is below.

Exploratory Analysis:

Problem statement is to look at effect of transmission on mpg. Let's have a look at mpg vs transmission using violin plot in **figure-1**. From the violin plot we can clearly see that automatic cars have on an average, 8 mpg lower mileage than the manual cars. Also note that the manual cars have a stretched (big range) of mpg's. To confirm it we will do a 2 group t.test.

```r
t.test(mpg~factor(am),data = mtcars)$p.value
```

```
## [1] 0.001373638
```

Since p-value is less than 5% (0.13% here), it means that the difference of mean of 2 groups is significant, meaning the 2 groups are different and hence their treatment, in this case am(auto/manual) indeed has an affect in mpg. Thus here, we can safely conclude that manual cars are more efficient. By how much, let's quantify with building a suitable model.

Studying correlation:

Let's look at the correlation matrix and identify which predictors are significant. Matrix in **figure-2** shows that mpg is:

- negatively related to:
  - **cyl (strong)**
  - **disp (strong)**
  - **hp (strong)**
  - **wt (strong)**
  - carb (mild)
- positively related to:
  - **drat (strong)**
  - qsec (weak)
  - **vs (strong)**
  - **am (strong)**
  - gear (weak)

Looking above we can rule out **qsec**, **gear** and **carb** for being weakly correlated, and proceed with rest. By carefully looking at the table, we also find that **vs** is strongly related to 4 other predictor: wt, hp, disp, cyl and hence it can be dropped in order to avoid residual inflation.

Fitting regressions:

```r
#Model1: reject
fit_all<-lm(mpg ~  factor(am) + factor(cyl) + disp +  hp + drat + wt + qsec + factor(vs) + factor(gear)
result<-data.frame(sqrt(vif(fit_all)))
colnames(result)<-c("GVIF","Df","GVIF..1..2.Df..")
result[with(result,order(GVIF)),]
```

```
##                    GVIF       Df GVIF..1..2.Df..
## drat           2.609533 1.000000        1.615405
## factor(vs)     2.843970 1.000000        1.686407
## factor(am)     3.151269 1.000000        1.775181
## qsec           3.284842 1.000000        1.812413
## wt             4.881683 1.000000        2.209453
## hp             5.312210 1.000000        2.304823
## factor(gear)   7.131081 1.414214        1.634138
## disp           7.769536 1.000000        2.787389
## factor(cyl)   11.319053 1.414214        1.834225
## factor(carb)  22.432384 2.236068        1.364858
```

Producing 11 possible models with different combinations of predictors.

```r
#Model:
fit1<-lm(mpg ~ factor(am) , data = mtcars)
fit2<-lm(mpg ~ factor(am) + drat , data = mtcars)
fit3<-lm(mpg ~ factor(am) + drat + wt , data = mtcars)
fit4<-lm(mpg ~ factor(am) + drat + wt + hp , data = mtcars)
fit5<-lm(mpg ~ factor(am) + drat + wt + hp + disp , data = mtcars)
fit6<-lm(mpg ~ factor(am) + drat + wt + hp + disp + factor(cyl), data = mtcars)
fit7<-lm(mpg ~ factor(am) + wt , data = mtcars)
fit8<-lm(mpg ~ factor(am) + wt + hp , data = mtcars)
fit9<-lm(mpg ~ factor(am) + wt + hp + disp , data = mtcars)
fit10<-lm(mpg ~ factor(am) + wt + hp + disp + factor(cyl), data = mtcars)
fit11<-lm(mpg ~ factor(am) + wt + disp + factor(cyl), data = mtcars)
```

Checking the nested models.

```r
anova(fit1,fit2,fit3,fit4,fit5,fit6,fit7,fit8,fit9,fit10,fit11)
```

```
## Analysis of Variance Table
##
## Model  1: mpg ~ factor(am)
## Model  2: mpg ~ factor(am) + drat
## Model  3: mpg ~ factor(am) + drat + wt
## Model  4: mpg ~ factor(am) + drat + wt + hp
## Model  5: mpg ~ factor(am) + drat + wt + hp + disp
## Model  6: mpg ~ factor(am) + drat + wt + hp + disp + factor(cyl)
## Model  7: mpg ~ factor(am) + wt
## Model  8: mpg ~ factor(am) + wt + hp
## Model  9: mpg ~ factor(am) + wt + hp + disp
## Model 10: mpg ~ factor(am) + wt + hp + disp + factor(cyl)
## Model 11: mpg ~ factor(am) + wt + disp + factor(cyl)
##    Res.Df    RSS Df Sum of Sq       F    Pr(>F)
## 1      30 720.90
## 2      29 573.64  1   147.256 23.5452 6.046e-05 ***
## 3      28 266.99  1   306.653 49.0316 3.065e-07 ***
```

```
## 4      27 176.96  1    90.023 14.3940 0.0008854 ***
## 5      26 175.67  1     1.297  0.2074 0.6528923
## 6      24 150.10  2    25.567  2.0440 0.1514552
## 7      29 278.32 -5  -128.219  4.1003 0.0078302 **
## 8      28 180.29  1    98.029 15.6741 0.0005843 ***
## 9      27 179.91  1     0.383  0.0613 0.8065370
## 10     25 150.41  2    29.499  2.3583 0.1161262
## 11     26 182.87 -1   -32.461  5.1902 0.0319077 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Storing the R-Squared values in a table, ordering and printing

```r
rss<-c()
for(i in 1:11) {
rss<-rbind(rss,summary(eval(parse(text = paste("fit",i,sep=""))))$r.squared)
}
model<-c(1:11)
check<-data.frame(cbind(model,rss))
colnames(check)<-c("Model","Rsquared")
check[with(check,order(-check$Rsquared)),]
```

```
##    Model  Rsquared
## 6      6 0.8667014
## 10    10 0.8664276
## 5      5 0.8439962
## 4      4 0.8428442
## 9      9 0.8402309
## 8      8 0.8398903
## 11    11 0.8376007
## 3      3 0.7628984
## 7      7 0.7528348
## 2      2 0.4905716
## 1      1 0.3597989
```

From the last table above we can say that model 6 explains by far the most of variation of mpg. So we will lock model 6 for now:

```r
mpg ~ factor(am) + drat + wt + hp + disp + factor(cyl)
```

Looking at summary

```r
#Removing intercept in Model-6
summary(lm(mpg ~ factor(am) + drat + wt + hp + disp + factor(cyl)-1, data = mtcars))
```

```
##
## Call:
## lm(formula = mpg ~ factor(am) + drat + wt + hp + disp + factor(cyl) -
##     1, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.8267 -1.4366 -0.4153  1.1649  5.0671
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## factor(am)0  32.611986   6.274227   5.198 2.52e-05 ***
## factor(am)1  34.293117   6.569708   5.220 2.38e-05 ***
## drat          0.326616   1.471086   0.222   0.8262    
## wt           -2.726729   1.200207  -2.272   0.0323 *  
## hp           -0.033038   0.014476  -2.282   0.0316 *  
## disp          0.004395   0.013090   0.336   0.7400    
## factor(cyl)6 -3.026760   1.576680  -1.920   0.0669 .  
## factor(cyl)8 -2.541967   3.059145  -0.831   0.4142    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 2.501 on 24 degrees of freedom
## Multiple R-squared:  0.9893, Adjusted R-squared:  0.9857 
## F-statistic: 277.7 on 8 and 24 DF,  p-value: < 2.2e-16
```

By looking at summary, we see that drat and disp are not significant so removing them.

```r
#Modifying Model-6 further
fit66<-lm(mpg ~ factor(am)  + wt + hp + factor(cyl), data = mtcars)
summary(fit66)$r.squared
```

```
## [1] 0.8658799
```

Adjusted R-squared looks pretty good: 0.8401. Let's finally look at the residual plots in **figure-3**. Residual plot looks ok.

- Residual vs fitted values is fairly scattered showing independence
- Normal Q-Q shows that residual are normally distributed
- Scale-location haas points which are not converging/diverging hence maintaining homoskedacity
- Very few points holds higher leverage and thus doesnt affect

So the manual transmission cars are better than auto transmission by:

```r
#mean estimate of the difference
coef(fit66)[2]
```

```
## factor(am)1 
##    1.809211
```

```r
#95% confidence interval
1.80921 + c(1,-1) * qt(.975,26) * 1.39630
```

```
## [1]  4.679346 -1.060926
```

Appendix:

figure-1

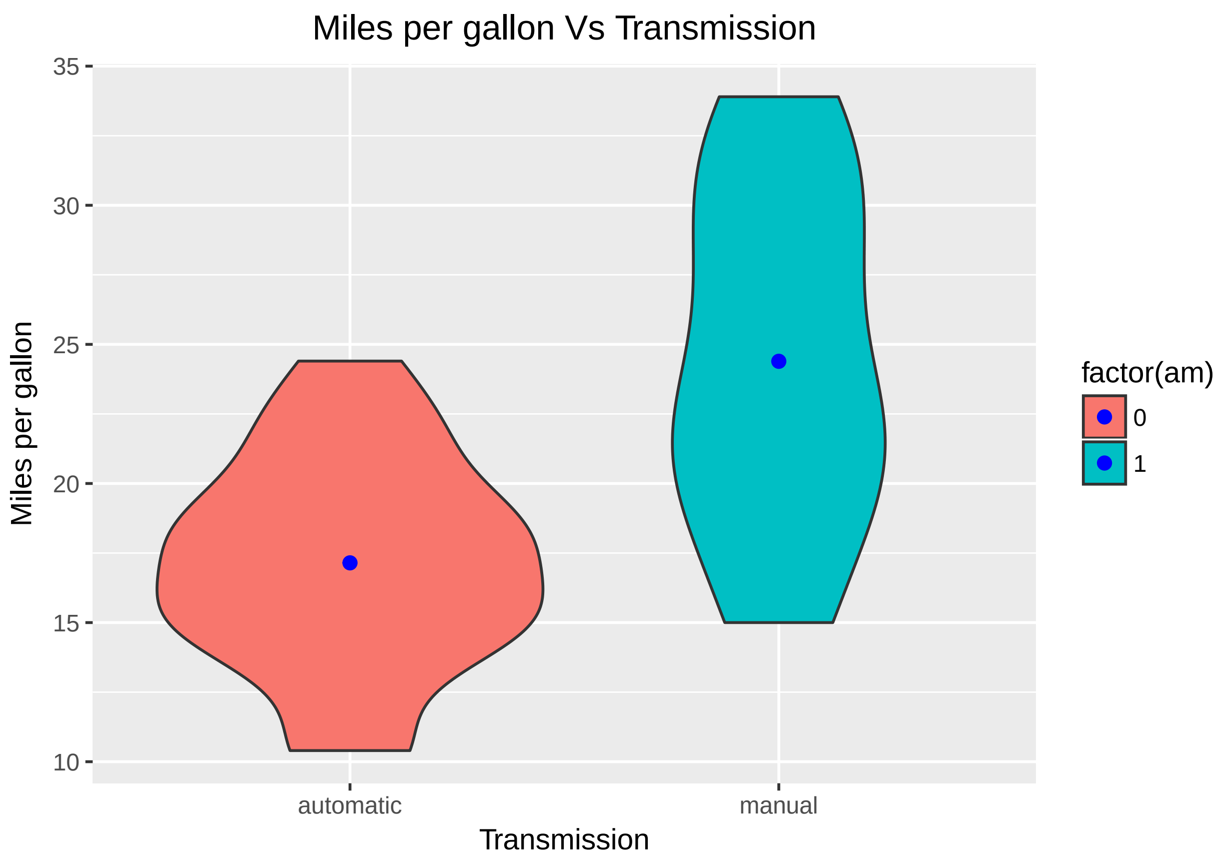


figure-2

```
#Correlation matrix:
corrplot(cor(mtcars),method = "pie")
```

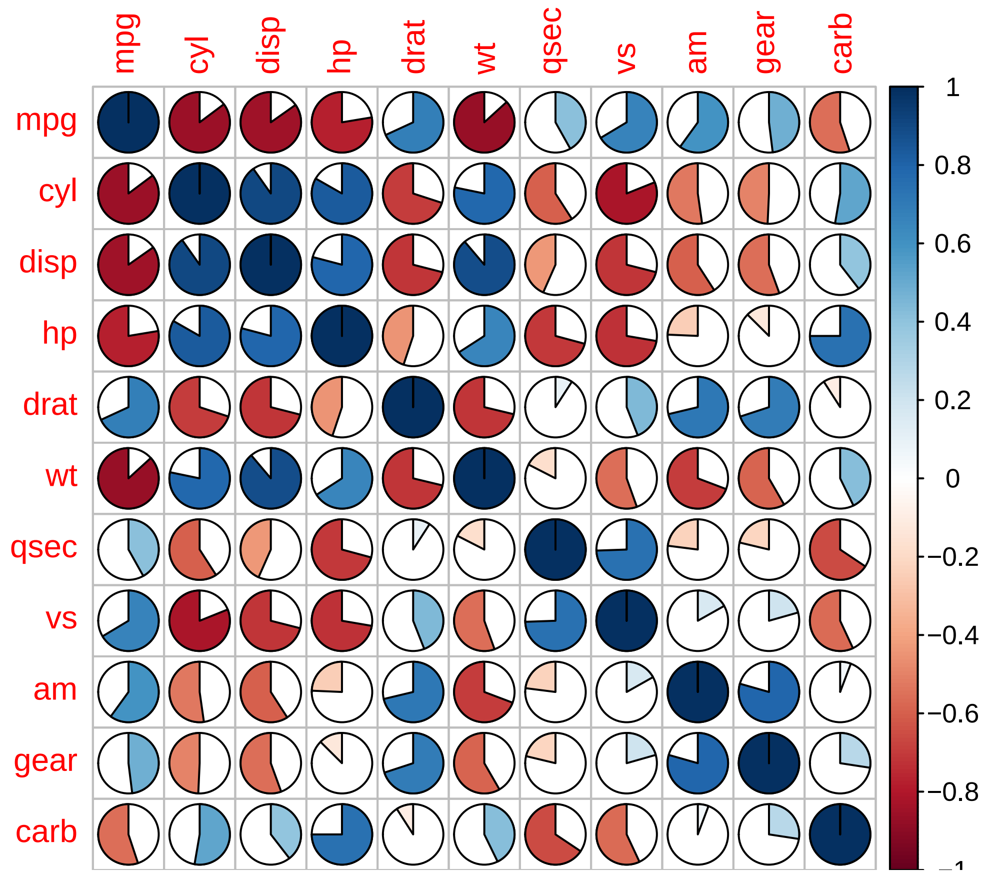
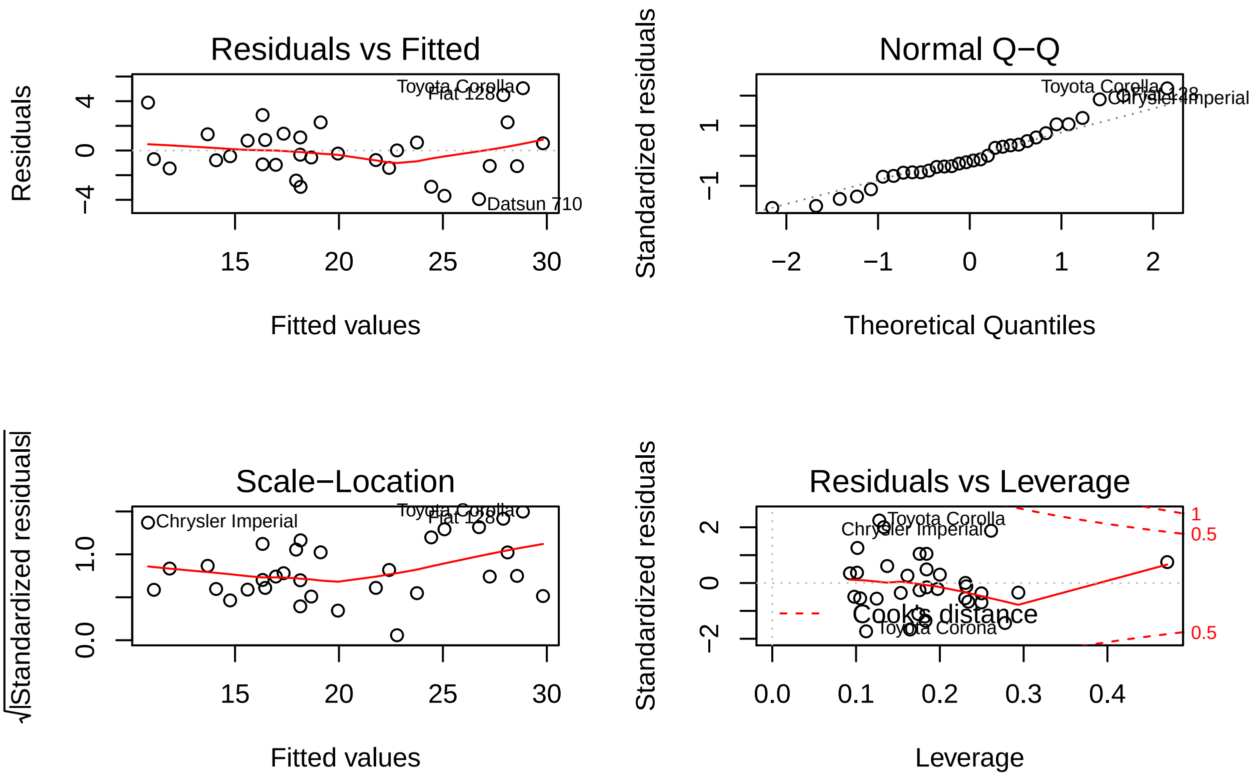

mpg
cyl
disp
hp
drat
wt
qsec
vs
am
gear
carb
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8

figure-3


Residuals vs Fitted
Residuals
Fitted values
Toyota Corolla
Fiat 128
Datsun 710
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Toyota Corolla
Fiat 128
Chrysler Imperial
Scale−Location
√|Standardized residuals|
Fitted values
Chrysler Imperial
Toyota Corolla
Fiat 128
Residuals vs Leverage
Standardized residuals
Leverage
Toyota Corolla
Chrysler Imperial
Toyota Corona
Cook's distance
1
0.5
0.5