Regression Analysis: MPG in Automatic vs Manual cars

Edwin Seah

19 May 2015

Executive Summary

This analysis checks if miles per gallon (MPG) benefit more from automatic versus manual transmissions, and quantifies any such difference. Although there is a stark difference in expected MPG between automatic and manual cars, in itself it is not a realistic predictor since the variables **wt** (lb/1000) and **qsec** (1/4 mile time) have significant influences on MPG, settling finally on our stepwise derived model of **mpg** ~ **factor(am) + wt + qsec**, after checking it versus a designed model involving groups of regressors that mpg is likely dependent on.

Getting/Transforming Data and some Exploratory Data Analysis

The **mtcars** dataset comprises fuel consumption (MPG) and 10 aspects of automobile design and performance for 32 cars, loaded as `data(mtcars)` and stored in a data frame **m**. We transform **am** as factor variable of 2 levels ("Automatic,"Manual"). Cursorily our box-whisker plot (Fig.1) indicates Manual transmissions have a clear advantage over Automatic transmissions in MPG terms.

Quantifying the relationship via Regression Analysis

As a baseline we simply fit **mpg** (outcome) against Transmission Type **am** (predictor).

```
m$am <- factor(m$am, labels=c("(Automatic)", "(Manual)"))
fitam <- lm(mpg ~ am, data=m) ; summary(fitam)$coef
```

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

This model gives an MPG expected gain of **7.24** going from an Automatic to Manual transmission. However, our adjusted R^2 is **0.34** (DF = NA); low as we had not fit 10 other candidate regressors. Residuals (Fig.2) exhibit homoskedacity (evenly scattered around 0) and nearly normally distributed, but only **33.85%** of MPG variability was explained.

Given limitations in explaining MPG variability with just **am**, a quick parsimonious model can be found using a mechanical backwards stepwise elimination approach (at a somewhat abritrary siginificance level of ($\alpha = 5\%$). For code brevity we use the inbuilt automated AIC method by calling`step()` (*fstep*); it gives us the same resultant model as the manual way (*fman*, see Fig. 3).

```
full <- lm(data=m, mpg ~ .) ; fstep <- summary(step(full, direction="backward", trace=0))
print(rbind(fman$coef, fstep$coef[1:4]))
```

```
##      (Intercept)        wt     qsec am(Manual)
## [1,]    9.617781 -3.916504 1.225886   2.935837
## [2,]    9.617781 -3.916504 1.225886   2.935837
```

We arrive at a model with an adjusted R^2 of **0.83** (DF = NA), with residuals showing some tail-skew in the normal probability plot (Fig. 4).

To check inflation of the estimate's variance by regressor groups we design a model that includes suspected/likely dependent variables of **mpg**, fitting the following models of interest (groups) in order:

Group	Weight	Engine Power	Engine Configuration	Gearing
Regressors	wt	disp, hp	cyl, carb, vs	gear, drat
Model	fit1	fit2	fit3	fit4

Using a nested liklihood ratio test (*fit1* to *fit4*) with our base *fit* helps check their contribution to **mpg** via the ANOVA results:

```
anova(fit, fit1, fit2, fit3, fit4)[1:6]
```

```
##   Res.Df    RSS Df Sum of Sq       F    Pr(>F)
## 1     30 720.90
## 2     29 278.32  1    442.58 62.2716 7.404e-08 ***
## 3     27 179.91  2     98.41  6.9234  0.004653 **
## 4     24 158.76  3     21.14  0.9917  0.415009
## 5     22 156.36  2      2.40  0.1691  0.845481
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
cv <- function(f) {summary(f)$cov.unscaled[2,2]}
c(cv(fit1), cv(fit2), cv(fit3), cv(fit4), cv(fit5))/cv(fit)
```

```
## [1] 1.921413 2.386005 3.597756 4.299664 2.541437
```

We would opt for *fit2* (**p-value = 0.0047**), rejecting for lack of significance) over the others. Our covariances and adjusted R^2 for *fit2* (**2.39**, **0.82**) and *fit5* (**2.54**, **0.83**) are similar, with *fit2* residuals (Fig. 5) showing the Maserati Bora exerting very high leverage. VIF for *fit2* regressors are higher than in our stepwise model *fit5*:

```
library(car) ; sqrt(vif(fit2)) ; sqrt(vif(fit5))
```

```
##       am       wt     disp       hp
## 1.544670 2.442070 2.774015 1.838752
```

```
##       am       wt     qsec
## 1.594189 1.575738 1.168049
```

We note that quarter mile time **qsec** has a very low VIF viz both **hp** and **disp**, which are likely colinear. Intuitively **qsec** may be a good proxy for any/all of the engine power/configuration variables. We conclude in favour of *fit5* (**mpg ~ factor(am) + wt + qsec**); it is simpler (one less regressor) than *fit2* with a marginally better adjusted R^2 of **0.83**, giving an expected **-3.92** MPG per 1000lbs increase in weight and **2.94** gain going to a Manual transmission, and **1.23** MPG gain per 1 second slower 1/4 mile timing **qsec**.

Project Repo

- All files and full code used are available from the Github Project Repository (https://github.com/slothdev/RMproject-Repo)

Appendix

Fig. 1 - MPG by Transmission Type

Both the median and inter-quartile range (or middle 50% of all cars) for Manual transmission type cars are clearly higher than Automatic transmission cars.

```r
boxplot(mpg ~ am,
        data=m,
        xlab="Transmission Type",
        ylab="Miles per Gallon",
        names=c("Automatic", "Manual"),
        main = "Fig.1 - MPG by Transmission Type")
```

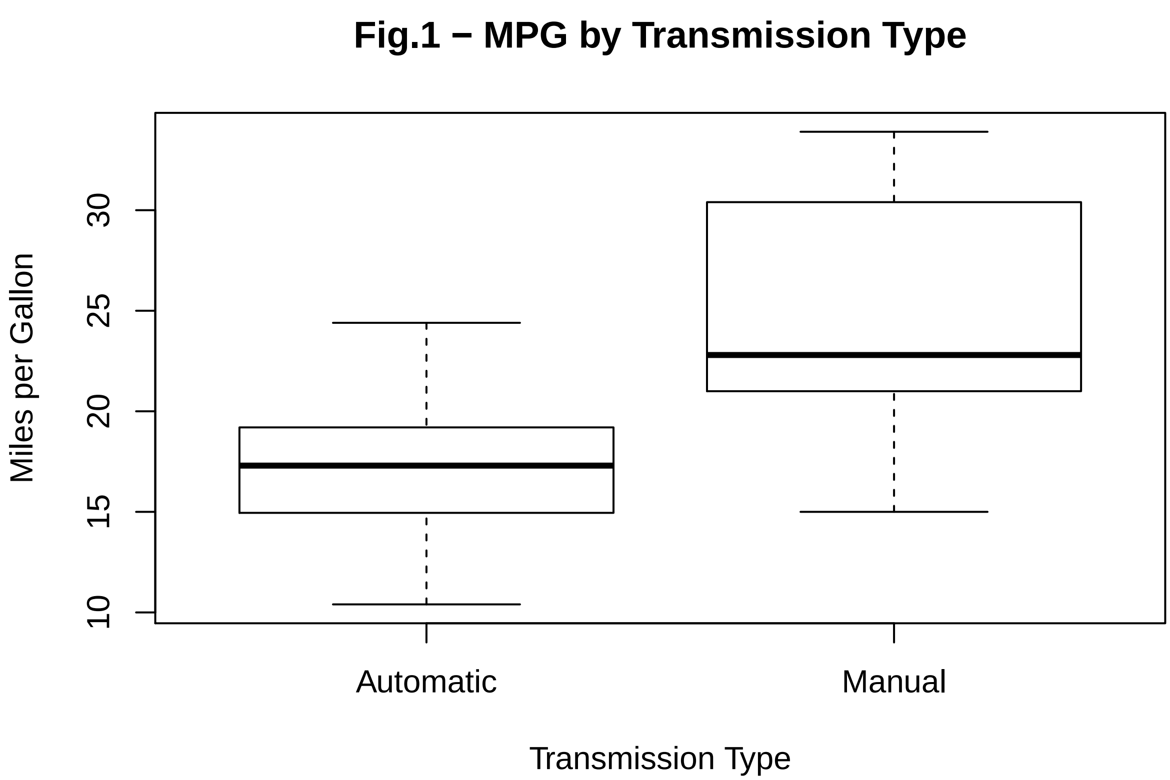

Fig. 2 - Residual and QQ plots of MPG by Transmission Type

```r
par(mfrow=c(1,2))
# Residuals plot
plot(resid(fit), main="Residual Plot (mpg ~ am)")
abline(a=0, b=0)
# Normal Probability Plot
```

```r
qqnorm(rstandard(fit),
       ylab="Standardized Residuals",
       xlab="Normal Scores")
qqline(rstandard(fit))
```

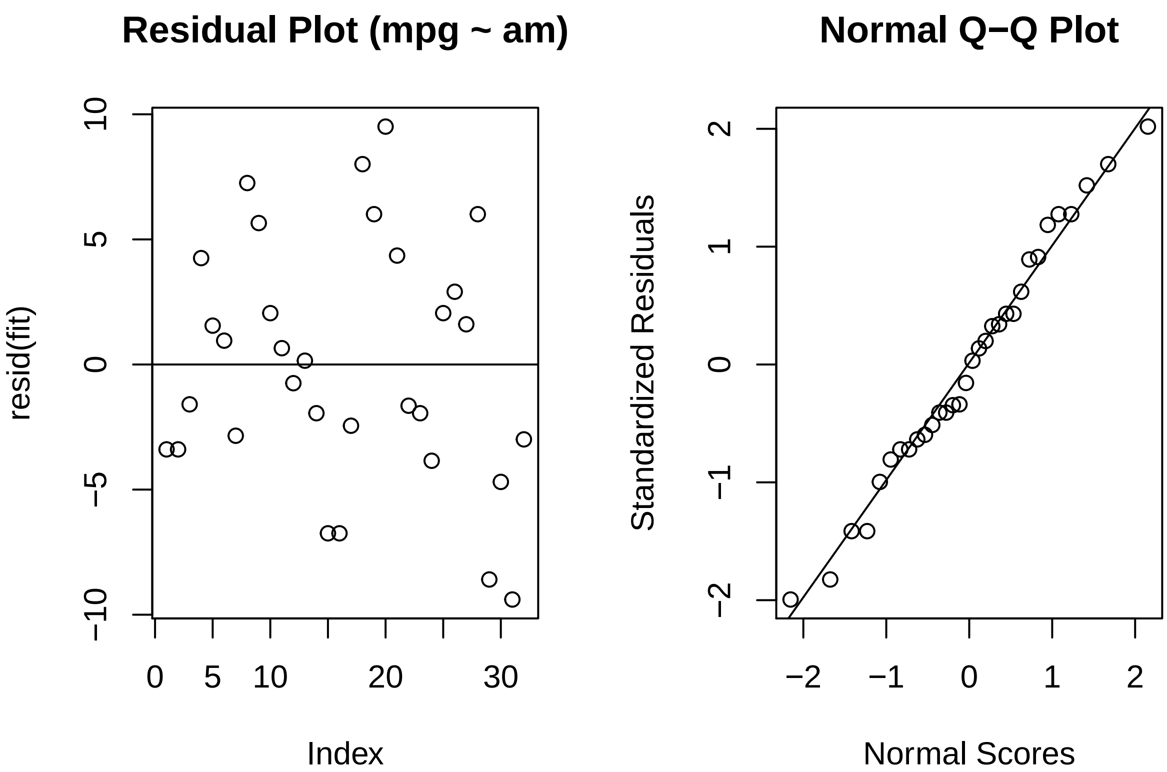

Fig. 3 - Simple backwards elimination stepwise by highest p-value

1. Start with a full model, as it provides an unbiased variance estimate for MPG due to including all variables. It may contain regressors with high colinearity and litle unique contribution to **mpg**.
2. Eliminate one regressor variable at a time (whichever has the highest p-value from the T-test) and refit.
3. Stop eliminating when no regressor has a p-value higher than α or when our adjusted R^2 stops going up.

These intermediate steps proceed as follows:

```r
showp <- function(b) {summary(b)$coeff[,4]}
showp(fitb1)
```

```
## (Intercept)         cyl        disp          hp        drat          wt
##  0.51812440  0.91608738  0.46348865  0.33495531  0.63527790  0.06325215
##        qsec          vs  am(Manual)        gear        carb
##  0.27394127  0.88142347  0.23398971  0.66520643  0.81217871
```

```r
showp(fitb2)
```

```
## (Intercept)        disp          hp        drat          wt        qsec
##  0.42659327  0.45380797  0.30615002  0.59214373  0.05715727  0.23291993
##          vs  am(Manual)        gear        carb
##  0.84325850  0.19768373  0.60753821  0.78325783
```

```r
showp(fitb3)
```

```
## (Intercept)        disp          hp        drat          wt        qsec
##  0.41985460  0.45897019  0.30398892  0.56300717  0.05049085  0.13194532
##  am(Manual)        gear        carb
##  0.19282690  0.56921947  0.74695821
```

```r
showp(fitb4)
```

```
## (Intercept)        disp          hp        drat          wt        qsec
## 0.433339841 0.213420001 0.134763097 0.581507634 0.002717119 0.049814778
##  am(Manual)        gear
## 0.171042438 0.619640616
```

```r
showp(fitb5)
```

```
## (Intercept)        disp          hp        drat          wt        qsec
## 0.338475309 0.244054196 0.149381426 0.462401185 0.002536163 0.049550895
##  am(Manual)
## 0.079692318
```

```r
showp(fitb6)
```

```
## (Intercept)        disp          hp          wt        qsec  am(Manual)
## 0.152378367 0.298972150 0.156387279 0.002075008 0.043907652 0.027487809
```

```r
showp(fitb7)
```

```
## (Intercept)          hp          wt        qsec  am(Manual)
## 0.072149342 0.223087932 0.001141407 0.075731202 0.045790788
```

```r
showp(fman)
```

```
##  (Intercept)           wt         qsec   am(Manual)
## 1.779152e-01 6.952711e-06 2.161737e-04 4.671551e-02
```

Fig. 4 - Residuals from backwards elimination (fman aka fit5)

```
par(mfrow=c(2,2))
plot(fman)
```

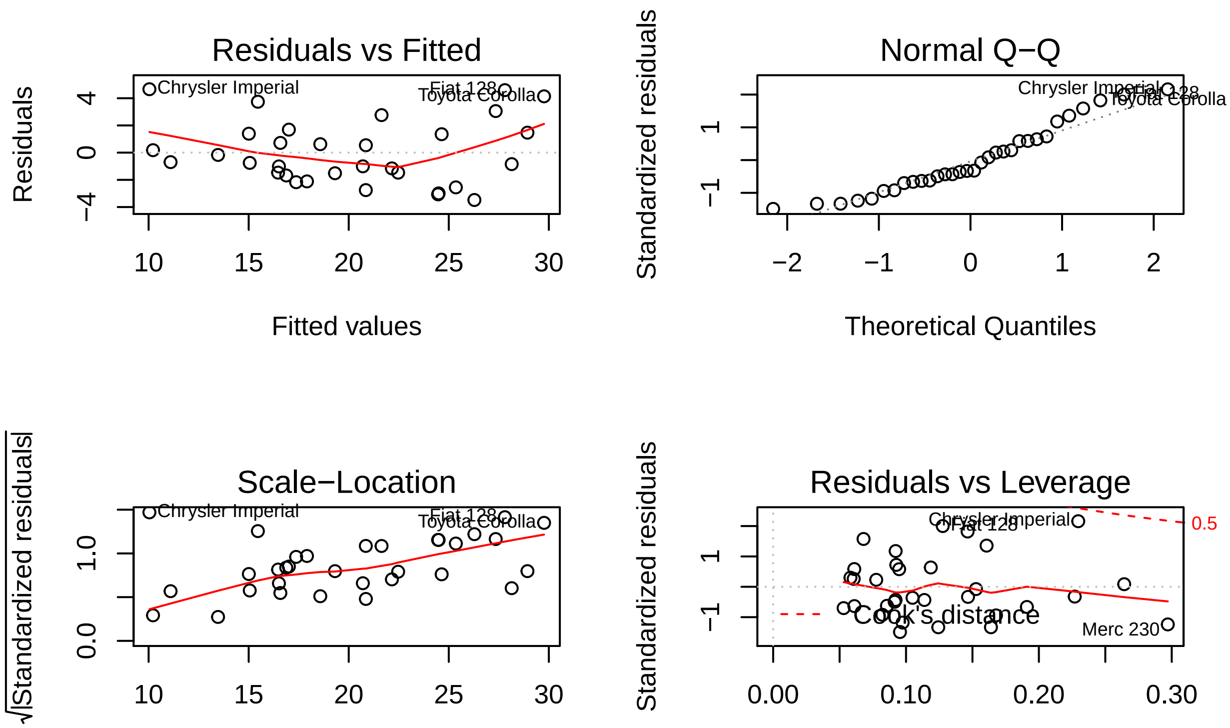

Fig. 5 - Residuals from fit2

```r
par(mfrow=c(2,2))
plot(fit2)
```