Regression Models Project

```
## Warning: package 'ggplot2' was built under R version 3.1.2
```

Executive Summary

In this report, we examine the `mtcars` dataset as provided by R. The `mtcars` data was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles. Using linear regression models and t.test we can form some conclusions on the relationships between Miles/gallon `mtcars$mpg` and all the other variables in the dataset. In particular we will look to answer some questions in the following order:

- "Is an automatic or manual transmission `mtcars$am` (0 = automatic, 1 = manual) better for MPG?"
- "Can we quantify the MPG difference between automatic and manual transmissions?"
- "Using the rest of the data can we come up with a feasible linear model that best explains MPG?"

Based on the analysis in the report we can summarize the following:

- Manual cars on average give higher MPG of **7.245** when we exclude all other variables
- The final model using a mix of the `step()` model and intution is: $mpg_e = 30.947 + 11.555 * am - 0.027 * hp - 2.516 * wt - 3.578 * wt * am$
- Results show the possibility of one outlier in the 'Maserati Bora' model which dents the effectiveness of the above model.

Exploring the Data

We first need to understand and clean the data before we perform any sort of linear regression. Figure 1.1 in the Appendix section shows pair graphs between all the variables. Using this plot in conjunction with `?mtcars` one can argue that some of the variables should be factorized. However, given the limited amount of observations along with the reasoning that for a varible like `cyl`, the number of cylinders can be different from observed of just 4, 6 or 8 (i.e. we can technically have 5 or 7 cylinders) then the only variables we `factorize()` are `vs` & `am` - whether engine is v-shaped (1 or 0) or whether car is automatic or manual (0 or 1) respectively.

Is an automatic or manual transmission better for MPG?

We perform basic exploratory data analysis via a boxplot for the two different types of transmission (0 =automatic, 1 = manual). Figure 2.1 (appendix) shows that there is a significant difference in mpg between 2 types of transmissions with manual cars having a higher mean MPG over that of automatic cars. We can take this a step further by conducting a Welch Two Sample t-test below (results in appendix Figure 2.2):

```
test <- t.test(mpg~am, data=mtcars)
```

We find that **`test$p.value` = 0.0014**, therefore we can reject null hyptothesis that the means are the same. Furthermore, the difference in means between the two types of transmissions is **7.245**, and we can conclude that manual transmission has on average higher MPG.

Quantifying MPG difference between transmissions

In order to quantify the difference in MPG between transmission and its significance we look to linearly regress `mtcars$mpg` against `mtcars$am` with the following model (results in appendix Figure 3.1):

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

Although the coefficient of `am` is significant in explaining `mpg`, the R-squared is only **0.3598**, and consequently is not enough to explain the variation in `mpg`. Therefore we have to explore the other variables in the dataset to see if we can come up with a more viable model

Model Building

We start off by regressing MPG against all the other variables:

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

From Figure 3.2 we see that by including all the variables, the **r-squared** is 0.869, however now none of the coefficients are significant resulting in a poor model.

Instead we look to build a model using the backwards `step()` function (which removes the least significant variables one by one) as a building block and then use a little bit of intuition and understanding of cars to finalize and make sense of the model

```
fit_step <- step(fit_all)
summary(fit_step)
```

Results from the `step()` algorithm show that the optimal model is: $mpg_e = b_0 + b_1 * wt + +b_2 * qsec + b_3 * am$, with **R-squared** of 0.8497 (di), however the problem with using quarter mile time (`qsec`) as a predictor is that this variable is actually an outcome of the parts that make a car rather than what explains the MPG for a particular model. So instead we substitute `qsec` with `hp` with the intuitive reasoning that higher horsepower may affect mpg, but higher horsepower usually means lower quarter mile times; this can be confirmed by looking at the correlation between the two variables which consequently is **-0.708**.

Now we analyze the model with `qsec` replaced: $mpg_e = b_0 + b_1 * wt + +b_2 * hp + b_3 * am$

```
fit1 <- lm(mpg ~ wt + am + hp, data = mtcars)
summary(fit1)
```

Results in Fig 3.4, show a pretty good **R-squared** = 0.8399 and with the exception of `am` all coefficients are significant giving us a pretty good final model.

Interactions

Before finalizing the model and reporting results, we look at the different variables for possible interactions. Fig 4.1 & Fig 4.2 are plots of `mpg` vs `wt` and `vs` respectively with transmission type in different colors. We see that there is a signficantly different slope in Fig 4.1 with the different transmission types, but not in Fig 4.2 suggestion an interaction of $wt * am$ to explore:

```
fit_final <- lm(mpg ~ am + hp + wt + wt:am, mtcars)
summary(fit_final)
```

Fig 4.3 shows this model has an **R-squared** = 0.8696 with all coefficients stastically significant. We can therefore conclude using `anova()` in fig 4.4 that `fit_final` is the best model given the level of significance for each additional variable:

$$mpg_e = 30.947 + 11.555 * am - 0.027 * hp - 2.516 * wt - 3.578 * am * wt$$

- Therefore, a manual car that has weight = 1000lbs will have a higher mpg of 11.555 - -3.578 = 7.977 higher mpg given same hp.
- A manual car that weighs 5000lbs will have a lower mpg of -6.335.

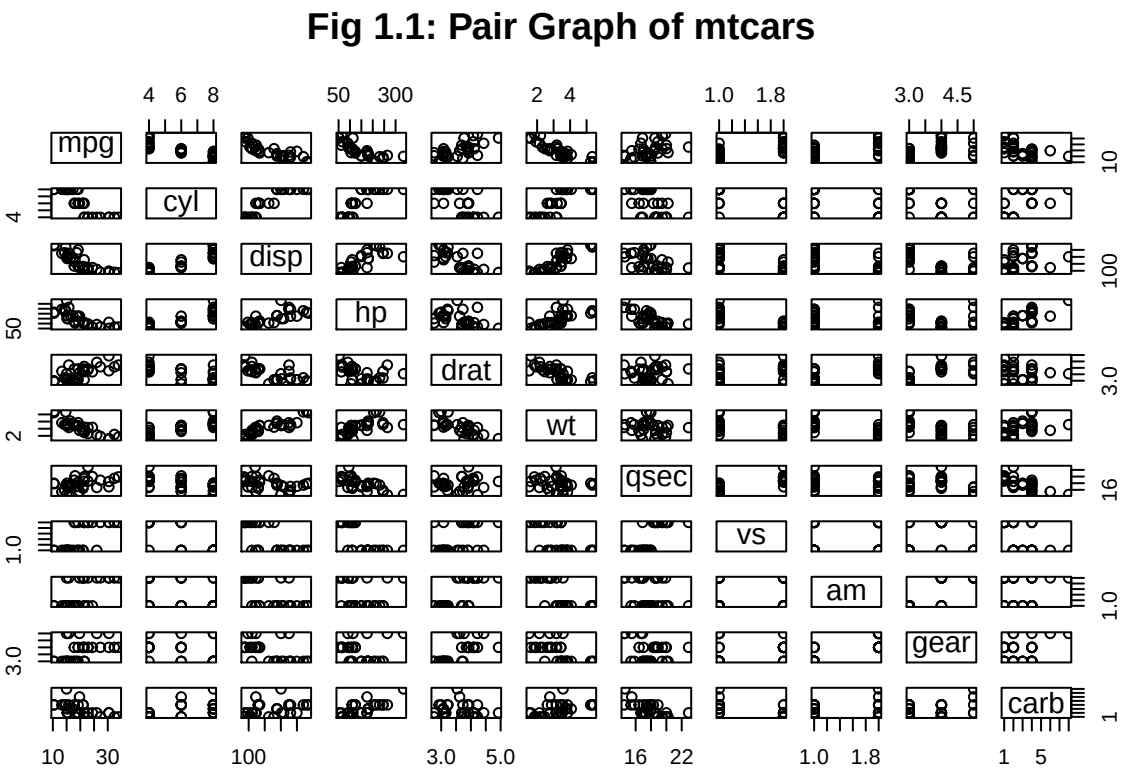
Residual Plot & Diagnostics

Fig 5.1 shows the residual plot & diagnostic:

- Residual plot shows no obvious pattern
- Q-Q plot shows for the most part that standardized residuals lie on the line.
- Scale-Location plot shows random distribution of points.
- Residual vs leverage for the most part has no outliers with exception of the 'Maserati Bora'. We show the influence in fig 4.4 using `dfbetas(fit_final)["Maserati Bora",1]`. This unfortunately shows some weakness in the model.

Appendix

```
pairs(mtcars,  main="Fig 1.1: Pair Graph of mtcars")
```




```
boxplot(mpg~am, data=mtcars,xlab="Transmission, 0 = Automatic, 1 = Manual", ylab="Miles/Gallon",
        main=" Fig 2.1 MPG v Transmission")
```

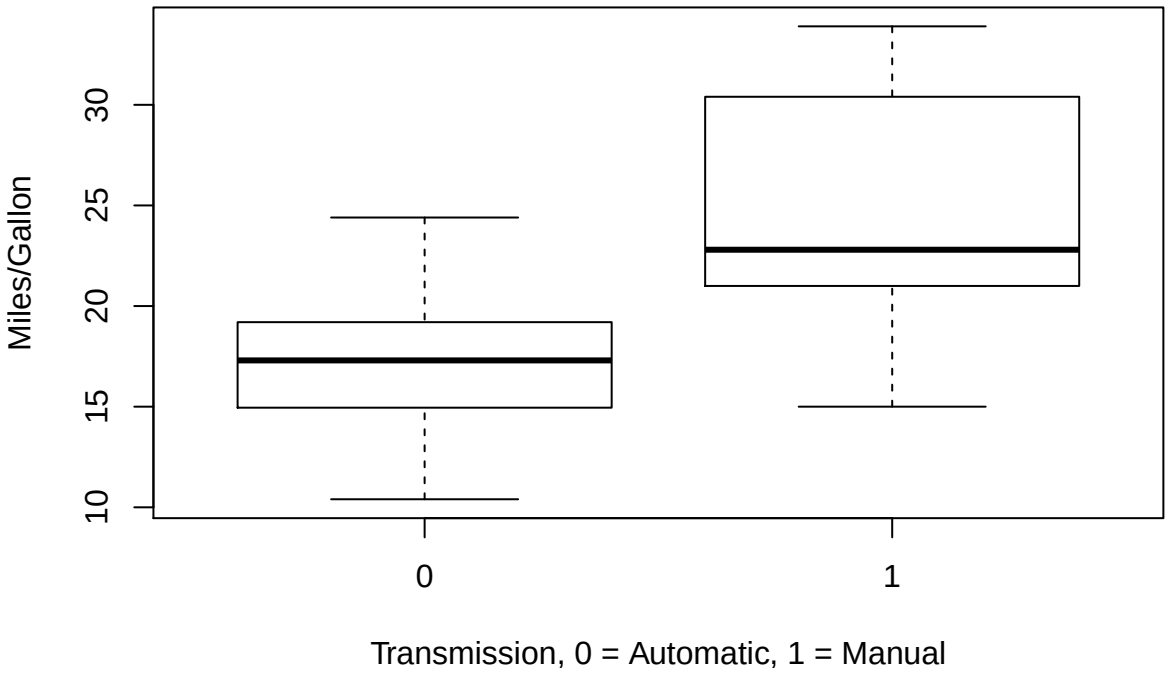


Figure 2.2

```
t.test(mpg~am, data=mtcars)
```

```
##
##  Welch Two Sample t-test
##
## data:  mpg by am
## t = -3.7671, df = 18.332, p-value = 0.001374
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -11.280194  -3.209684
## sample estimates:
## mean in group 0 mean in group 1
##        17.14737        24.39231
```

Figure 3.1

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

```
##              Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 17.147368   1.124603 15.247492 1.133983e-15
## am1          7.244939   1.764422  4.106127 2.850207e-04
```

Figure 3.2

```
##                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
## vs1          0.31776281  2.10450861  0.1509915 0.88142347
## am1          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
```

Figure 3.3

```
fit_step <- step(fit_all)
```

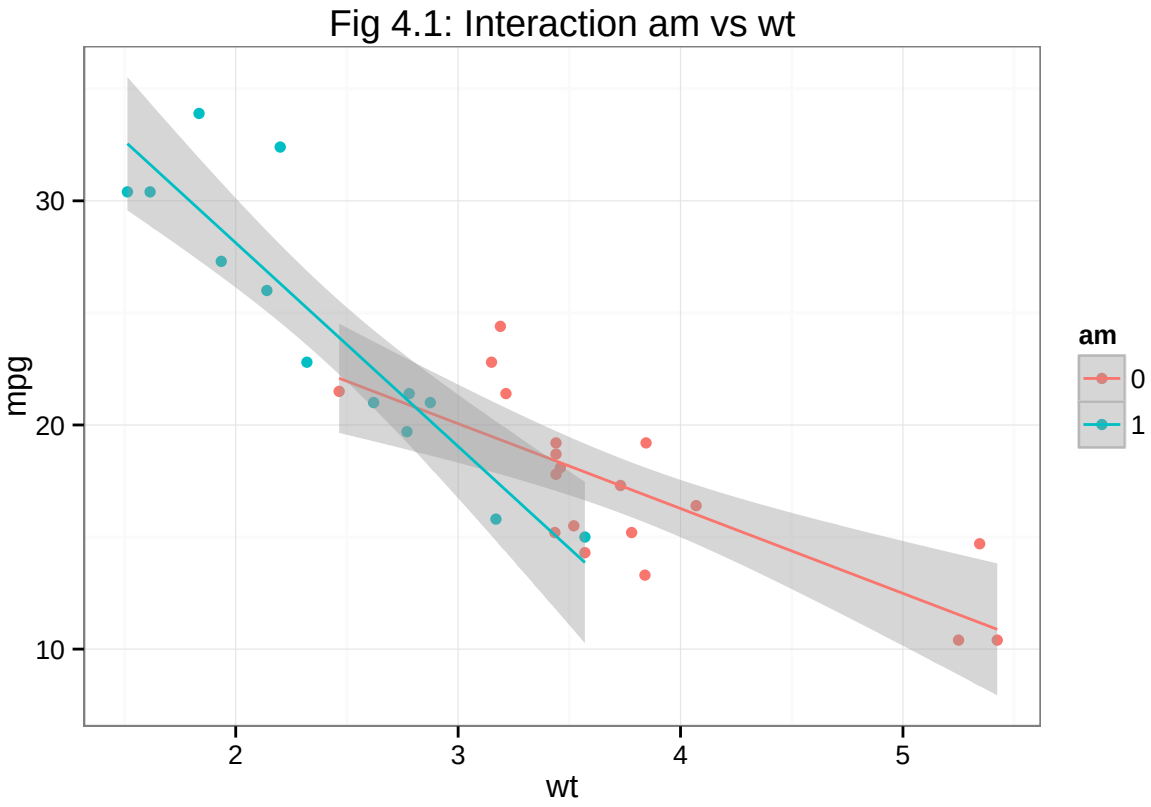
```
summary(fit_step)$coef
```

```
##              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
## am1          2.935837  1.4109045  2.080819 4.671551e-02
```

Figure 3.4

```
##
## Call:
## lm(formula = mpg ~ wt + am + hp, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4221 -1.7924 -0.3788  1.2249  5.5317
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.002875   2.642659  12.867 2.82e-13 ***
## wt          -2.878575   0.904971  -3.181 0.003574 **
## am1          2.083710   1.376420   1.514 0.141268
## hp          -0.037479   0.009605  -3.902 0.000546 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.538 on 28 degrees of freedom
## Multiple R-squared:  0.8399, Adjusted R-squared:  0.8227
## F-statistic: 48.96 on 3 and 28 DF,  p-value: 2.908e-11
```

```
g1 <- ggplot(data=mtcars, aes(x=wt, y=mpg, colour=am)) + geom_point() + stat_smooth(method="lm") + labs(tit
g1
```




```
g1 <- ggplot(data=mtcars, aes(x=hp, y=mpg, colour=am)) +
geom_point() + stat_smooth(method="lm") + labs(title="Fig 4.2: Interaction am vs hp") + theme_bw()
g1
```

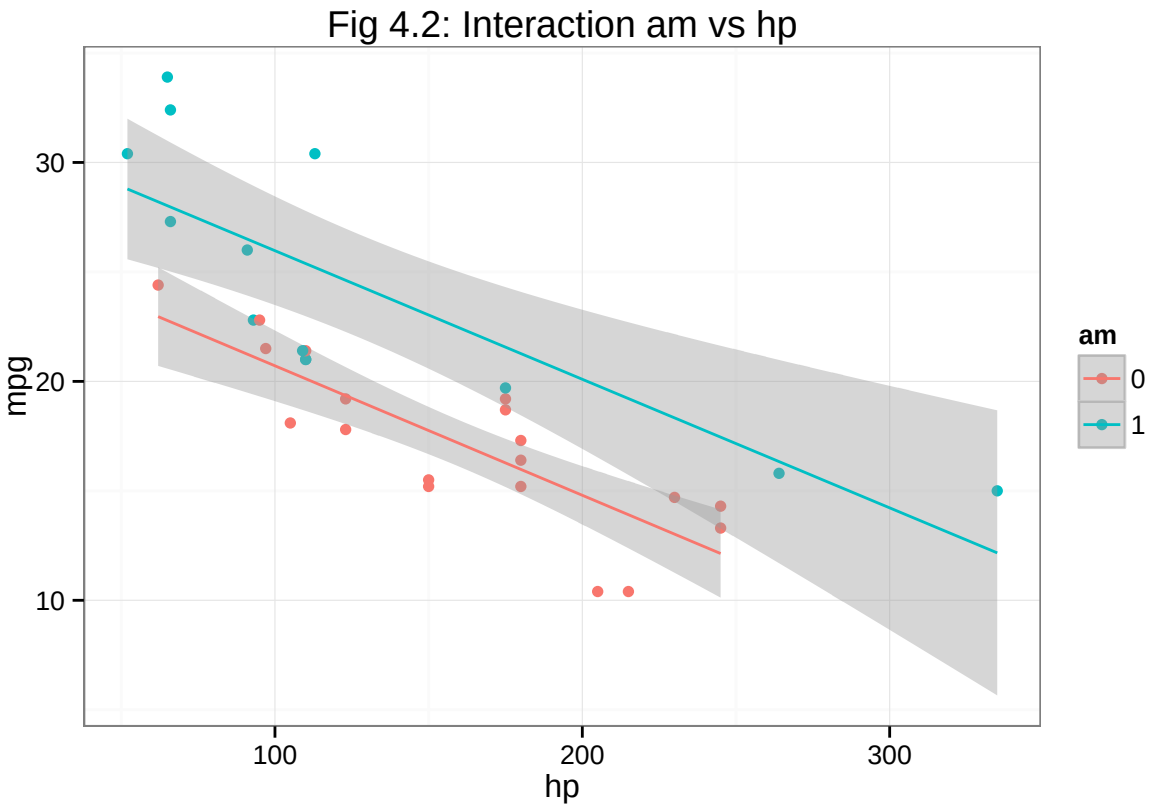


Figure 4.3

```
##                Estimate  Std. Error   t value     Pr(>|t|)
## (Intercept) 30.94733319 2.723410935 11.363446 8.546944e-12
## am1         11.55481296 4.023276579  2.871991 7.844579e-03
## hp          -0.02694935 0.009795903 -2.751084 1.047673e-02
## wt          -2.51558550 0.844496532 -2.978799 6.051842e-03
## am1:wt      -3.57790980 1.442795585 -2.479845 1.967639e-02
```

Figure 4.4

```
anova(fit_am, fit1, fit_final)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ am
## Model 2: mpg ~ wt + am + hp
## Model 3: mpg ~ am + hp + wt + wt:am
##   Res.Df    RSS Df Sum of Sq       F    Pr(>F)
## 1     30 720.90
## 2     28 180.29  2    540.61 49.6998 8.909e-10 ***
## 3     27 146.85  1     33.45  6.1496   0.01968 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
dfbetas(fit_final)["Maserati Bora",1]
```

```
## [1] 0.0601626
```