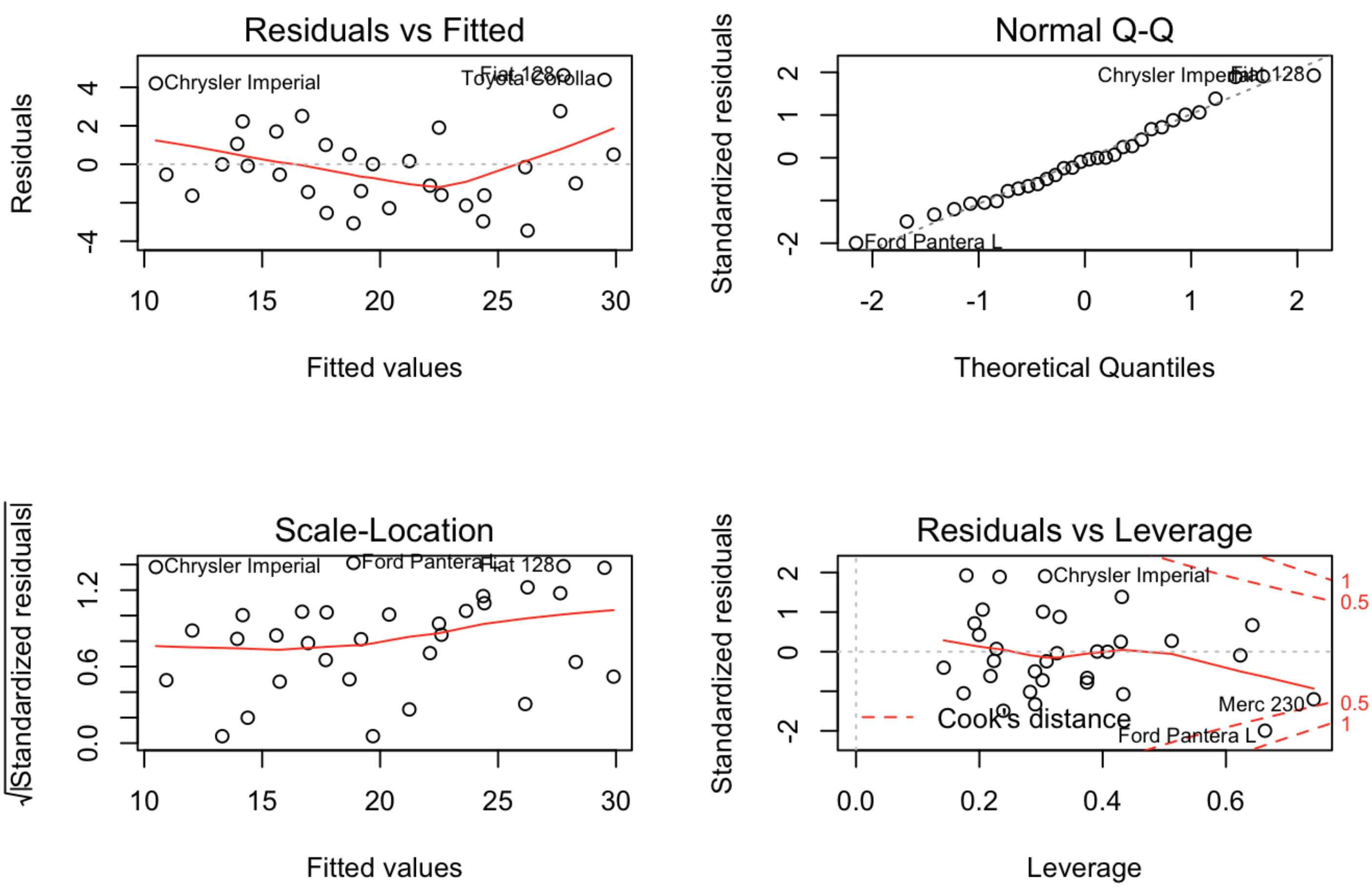
Executive Summary

This document reviews 1974 *Motor Trend* US magazines' data to investigate if the type of tranmission used has an effect on miles per gallon (mpg) yielded. It looks at exactly what type of effect (positive or negative), and then explores an approach to find a possible value or range of values this effect has.

```r
library(datasets)
library(lattice)

# Exploratory phase, looking at data and residuals - may be fruitless
fit <- lm(mpg ~ ., mtcars)
par(mfrow = c(2,2))
plot(fit)
```




```r
## Exploratory, looking at Correlation between variables; am, mpg and other variables
amMpgCor <- cbind(cor(mtcars)[,"am"], cor(mtcars)[,"mpg"])
colnames(amMpgCor) <- c("am_Cor", "mpg_Cor")
print(amMpgCor["am","mpg_Cor"])
```

```
## [1] 0.5998324
```

Is an automatic or manual transmission better for MPG

Given the data here, it appears there is a definite **correlation** between am and mpg. It appears there is a **positive** correlation between miles per gallon and **manual** transmission (am = 1 is manual; am = 0 is automatic). In short, **manual transmission is better for MPG than automatic.**

Quantify the MPG difference between automatic and manual transmissions

We can use the t test to identify a range.

```
unfitConf <- t.test(mpg ~ am, mtcars)
print(c(unfitConf$conf.int[1],unfitConf$conf.int[2]))
```

```
## [1] -11.280194  -3.209684
```

So can say that we can be **95% confident that manual tranmission increases miles per gallon somewhere between 3.2096842 and 11.2801944.**

Unfortunately, the data is multivariate, and the correlation is not quite as simple. We can see there are many factors that affect mpg (see appendix for amMpgCor output). Ideally, 100% should be accounted for by am, but given that the cov we calculated is about 0.6, the relationship is not necessarily direct.

So, in order to see if we need to factor the other variables, we will take our fitted regression line with all variables, and compare it to a line with just mpg vs am.

```
unfit <- lm(mpg ~ am, mtcars)
semifit <- lm(mpg ~ am + wt + cyl + disp, mtcars )
anova(fit, semifit, unfit)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ cyl + disp + hp + drat + wt + qsec + vs + am + gear + carb
## Model 2: mpg ~ am + wt + cyl + disp
## Model 3: mpg ~ am
##   Res.Df    RSS Df Sum of Sq       F    Pr(>F)
## 1     21 147.49
## 2     27 188.43 -6    -40.93  0.9713    0.4684
## 3     30 720.90 -3   -532.47 25.2708 3.641e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

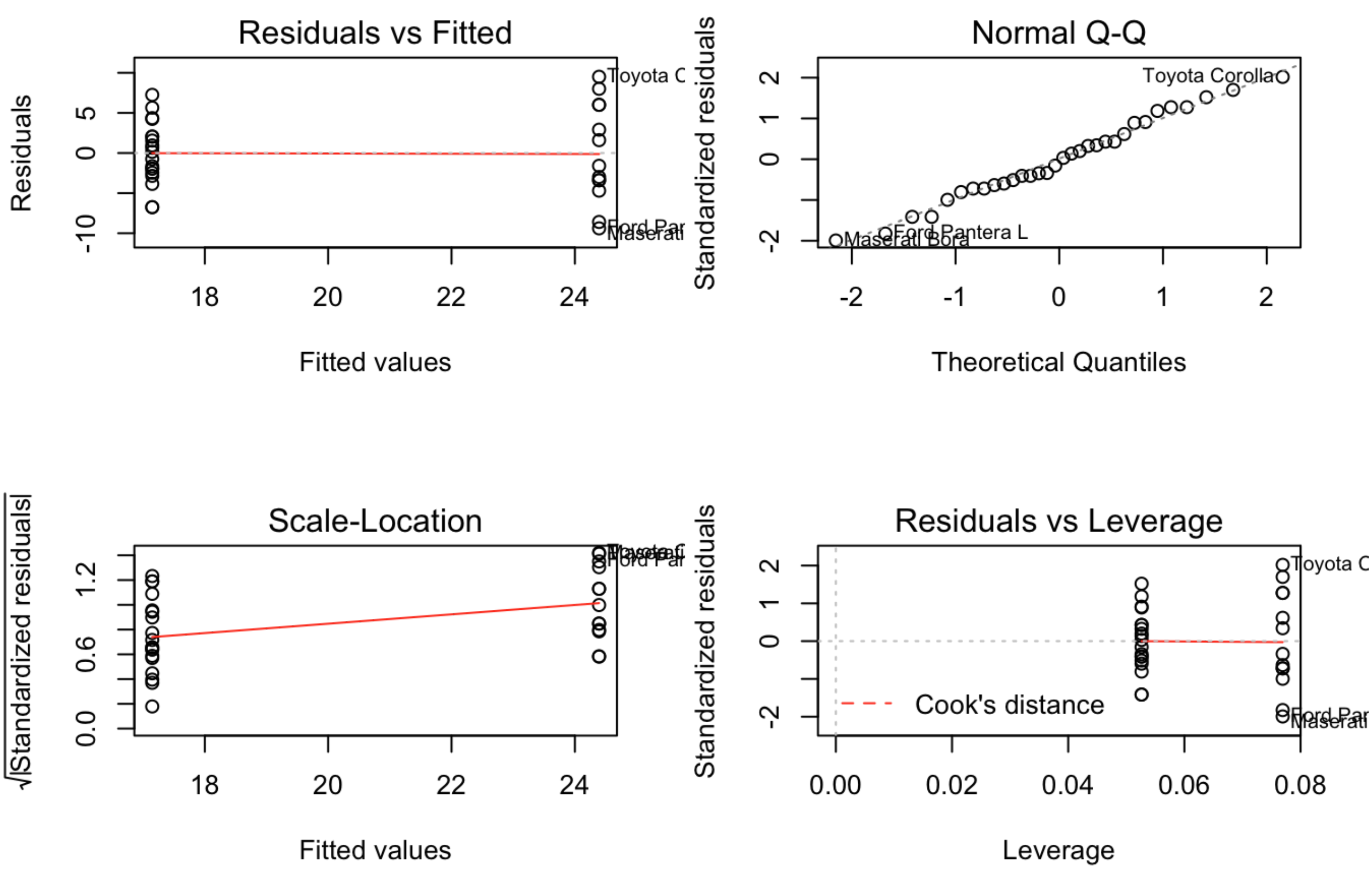
This confirms we cannot ignore other variations, and and could find an equation close to the “mpg vs am” plot (see appendix) that accounts for the variances to get a stronger, tighter confidence interval. As you can see, “semifit” matches the original data nicely, yet is closer to “mpg vs am” than “fit” is. However, exploration of this “semifit” data would require additional length than allowed.

Appendix

```
print(amMpgCor)
```

```
##           am_Cor    mpg_Cor
## mpg   0.59983243  1.0000000
## cyl  -0.52260705 -0.8521620
## disp -0.59122704 -0.8475514
## hp   -0.24320426 -0.7761684
## drat  0.71271113  0.6811719
## wt   -0.69249526 -0.8676594
## qsec -0.22986086  0.4186840
## vs    0.16834512  0.6640389
## am    1.00000000  0.5998324
## gear  0.79405876  0.4802848
## carb  0.05753435 -0.5509251
```

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



```r
# Also performed the following; cannot fit into appendix or here
# for(var in colnames(mtcars[,2:ncol(mtcars)])) {
#   print(xyplot(mpg ~ mtcars[,var] | am, mtcars, xlab = var))
# }
```