Effects of Transmission Type on Miles per Gallon

Toby Popenfoose

June 20, 2015

Executive Summary:

This analysis is of the mtcars dataset described in https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/mtcars.html. There are 19 cars with manual transmissions and 13 with automatic transmissions in the dataset. This report is of the inference from modeling the dataset to determine and quantify the difference in miles per gallon for an automatic versus a manual transmission. The multivariable linear model shows there is a statistically significant difference at a 95% level between manual and automatic transmissions. Accounting for weight and quarter mile time, the predicted difference in miles per gallon is 2.94 miles per gallon better with an automatic transmission than with a manual transmission. The lower 95% confidence interval is 0.05 miles per gallon and the upper 95% confidence interval is 5.83 miles per gallon due to the difference in transmissions. The multivariable linear model using transmission, weight and quarter second time explains 85% of the model variation compared to only 36% explained in the simple linear model of just transmission to predict miles per gallon. When testing the two models, the multivariable model is statistically significantly better than the simple linear model.

Exploratory Data Analysis:

There were no missing values found.

For EDA I used the str, summary, cor and table functions. See the roughDraft report on github for the particulars.

```r
str(mtcars)
```

```
## 'data.frame':    32 obs. of  11 variables:
##  $ mpg : num  21 21 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 ...
##  $ cyl : num  6 6 4 6 8 6 8 4 4 6 ...
##  $ disp: num  160 160 108 258 360 ...
##  $ hp  : num  110 110 93 110 175 105 245 62 95 123 ...
##  $ drat: num  3.9 3.9 3.85 3.08 3.15 2.76 3.21 3.69 3.92 3.92 ...
##  $ wt  : num  2.62 2.88 2.32 3.21 3.44 ...
##  $ qsec: num  16.5 17 18.6 19.4 17 ...
##  $ vs  : num  0 0 1 1 0 1 0 1 1 1 ...
##  $ am  : num  1 1 1 0 0 0 0 0 0 0 ...
##  $ gear: num  4 4 4 3 3 3 3 4 4 4 ...
##  $ carb: num  4 4 1 1 2 1 4 2 2 4 ...
```

```r
summary(mtcars)
```

```
##       mpg             cyl             disp             hp       
##  Min.   :10.4   Min.   :4.00   Min.   : 71.1   Min.   : 52.0  
##  1st Qu.:15.4   1st Qu.:4.00   1st Qu.:120.8   1st Qu.: 96.5  
##  Median :19.2   Median :6.00   Median :196.3   Median :123.0  
##  Mean   :20.1   Mean   :6.19   Mean   :230.7   Mean   :146.7  
##  3rd Qu.:22.8   3rd Qu.:8.00   3rd Qu.:326.0   3rd Qu.:180.0  
##  Max.   :33.9   Max.   :8.00   Max.   :472.0   Max.   :335.0  
##       drat            wt            qsec            vs       
##  Min.   :2.76   Min.   :1.51   Min.   :14.5   Min.   :0.000  
```

```
##  1st Qu.:3.08   1st Qu.:2.58   1st Qu.:16.9   1st Qu.:0.000
##  Median :3.69   Median :3.33   Median :17.7   Median :0.000
##  Mean   :3.60   Mean   :3.22   Mean   :17.8   Mean   :0.438
##  3rd Qu.:3.92   3rd Qu.:3.61   3rd Qu.:18.9   3rd Qu.:1.000
##  Max.   :4.93   Max.   :5.42   Max.   :22.9   Max.   :1.000
##        am             gear           carb
##  Min.   :0.000   Min.   :3.00   Min.   :1.00
##  1st Qu.:0.000   1st Qu.:3.00   1st Qu.:2.00
##  Median :0.000   Median :4.00   Median :2.00
##  Mean   :0.406   Mean   :3.69   Mean   :2.81
##  3rd Qu.:1.000   3rd Qu.:4.00   3rd Qu.:4.00
##  Max.   :1.000   Max.   :5.00   Max.   :8.00
```

```r
cor(mtcars)
```

```
##          mpg     cyl    disp      hp     drat      wt    qsec      vs
## mpg   1.0000 -0.8522 -0.8476 -0.7762  0.68117 -0.8677  0.4187  0.6640
## cyl  -0.8522  1.0000  0.9020  0.8324 -0.69994  0.7825 -0.5912 -0.8108
## disp -0.8476  0.9020  1.0000  0.7909 -0.71021  0.8880 -0.4337 -0.7104
## hp   -0.7762  0.8324  0.7909  1.0000 -0.44876  0.6587 -0.7082 -0.7231
## drat  0.6812 -0.6999 -0.7102 -0.4488  1.00000 -0.7124  0.0912  0.4403
## wt   -0.8677  0.7825  0.8880  0.6587 -0.71244  1.0000 -0.1747 -0.5549
## qsec  0.4187 -0.5912 -0.4337 -0.7082  0.09120 -0.1747  1.0000  0.7445
## vs    0.6640 -0.8108 -0.7104 -0.7231  0.44028 -0.5549  0.7445  1.0000
## am    0.5998 -0.5226 -0.5912 -0.2432  0.71271 -0.6925 -0.2299  0.1683
## gear  0.4803 -0.4927 -0.5556 -0.1257  0.69961 -0.5833 -0.2127  0.2060
## carb -0.5509  0.5270  0.3950  0.7498 -0.09079  0.4276 -0.6562 -0.5696
##            am    gear     carb
## mpg   0.59983  0.4803 -0.55093
## cyl  -0.52261 -0.4927  0.52699
## disp -0.59123 -0.5556  0.39498
## hp   -0.24320 -0.1257  0.74981
## drat  0.71271  0.6996 -0.09079
## wt   -0.69250 -0.5833  0.42761
## qsec -0.22986 -0.2127 -0.65625
## vs    0.16835  0.2060 -0.56961
## am    1.00000  0.7941  0.05753
## gear  0.79406  1.0000  0.27407
## carb  0.05753  0.2741  1.00000
```

```r
table(mtcars$am)
```

```
##
##  0  1
## 19 13
```

It appears that two variables, am and vs, do not make sense as numeric. I will transform am and vs to factors so the modeling functions will behave better.

```r
mtcars$am <- factor(mtcars$am)
mtcars$vs <- factor(mtcars$vs)
```

Simple Linear Model:

A naive starting point would be to just do a simple linear regression of just the independent variable am to predict mpg. After doing a simple linear model with just 'am' for the predictor, only 36% of the variation is explained by the model suggesting there are other confounder variables to be accounted for.

```
fit1 <- lm(mpg ~ am, mtcars)
confint(fit1, level=.95)  # 95% confidence interval
```

```
##              2.5 % 97.5 %
## (Intercept) 14.851  19.44
## am1          3.642  10.85
```

```
summary(fit1)
```

```
##
## Call:
## lm(formula = mpg ~ am, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.392 -3.092 -0.297  3.244  9.508
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    17.15       1.12   15.25  1.1e-15
## am1             7.24       1.76    4.11  0.00029
##
## Residual standard error: 4.9 on 30 degrees of freedom
## Multiple R-squared:  0.36,   Adjusted R-squared:  0.338
## F-statistic: 16.9 on 1 and 30 DF,  p-value: 0.000285
```

```
tidy(fit1, conf.int=TRUE)
```

```
##          term estimate std.error statistic   p.value conf.low conf.high
## 1 (Intercept)   17.147     1.125    15.247 1.134e-15   14.851     19.44
## 2         am1    7.245     1.764     4.106 2.850e-04    3.642     10.85
```

```
glance(fit1)
```

```
##   r.squared adj.r.squared sigma statistic  p.value df logLik   AIC   BIC
## 1    0.3598        0.3385 4.902     16.86 0.000285  2 -95.24 196.5 200.9
##   deviance df.residual
## 1    720.9          30
```

```
augment(fit1)
```

```
##               .rownames  mpg am .fitted .se.fit  .resid    .hat .sigma
## 1             Mazda RX4 21.0  1   24.39   1.360 -3.3923 0.07692  4.943
```

```
## 2        Mazda RX4 Wag 21.0  1   24.39   1.360 -3.3923 0.07692  4.943
## 3           Datsun 710 22.8  1   24.39   1.360 -1.5923 0.07692  4.976
## 4       Hornet 4 Drive 21.4  0   17.15   1.125  4.2526 0.05263  4.919
## 5    Hornet Sportabout 18.7  0   17.15   1.125  1.5526 0.05263  4.977
## 6              Valiant 18.1  0   17.15   1.125  0.9526 0.05263  4.983
## 7           Duster 360 14.3  0   17.15   1.125 -2.8474 0.05263  4.956
## 8            Merc 240D 24.4  0   17.15   1.125  7.2526 0.05263  4.790
## 9             Merc 230 22.8  0   17.15   1.125  5.6526 0.05263  4.868
## 10            Merc 280 19.2  0   17.15   1.125  2.0526 0.05263  4.970
## 11           Merc 280C 17.8  0   17.15   1.125  0.6526 0.05263  4.984
## 12          Merc 450SE 16.4  0   17.15   1.125 -0.7474 0.05263  4.984
## 13          Merc 450SL 17.3  0   17.15   1.125  0.1526 0.05263  4.986
## 14         Merc 450SLC 15.2  0   17.15   1.125 -1.9474 0.05263  4.972
## 15  Cadillac Fleetwood 10.4  0   17.15   1.125 -6.7474 0.05263  4.817
## 16 Lincoln Continental 10.4  0   17.15   1.125 -6.7474 0.05263  4.817
## 17   Chrysler Imperial 14.7  0   17.15   1.125 -2.4474 0.05263  4.964
## 18            Fiat 128 32.4  1   24.39   1.360  8.0077 0.07692  4.740
## 19         Honda Civic 30.4  1   24.39   1.360  6.0077 0.07692  4.849
## 20      Toyota Corolla 33.9  1   24.39   1.360  9.5077 0.07692  4.635
## 21       Toyota Corona 21.5  0   17.15   1.125  4.3526 0.05263  4.916
## 22    Dodge Challenger 15.5  0   17.15   1.125 -1.6474 0.05263  4.976
## 23         AMC Javelin 15.2  0   17.15   1.125 -1.9474 0.05263  4.972
## 24          Camaro Z28 13.3  0   17.15   1.125 -3.8474 0.05263  4.932
## 25    Pontiac Firebird 19.2  0   17.15   1.125  2.0526 0.05263  4.970
## 26           Fiat X1-9 27.3  1   24.39   1.360  2.9077 0.07692  4.954
## 27       Porsche 914-2 26.0  1   24.39   1.360  1.6077 0.07692  4.976
## 28        Lotus Europa 30.4  1   24.39   1.360  6.0077 0.07692  4.849
## 29      Ford Pantera L 15.8  1   24.39   1.360 -8.5923 0.07692  4.701
## 30        Ferrari Dino 19.7  1   24.39   1.360 -4.6923 0.07692  4.903
## 31       Maserati Bora 15.0  1   24.39   1.360 -9.3923 0.07692  4.644
## 32          Volvo 142E 21.4  1   24.39   1.360 -2.9923 0.07692  4.952
##       .cooksd .std.resid
## 1  0.02161671   -0.72028
## 2  0.02161671   -0.72028
## 3  0.00476270   -0.33809
## 4  0.02206695    0.89130
## 5  0.00294147    0.32541
## 6  0.00110733    0.19966
## 7  0.00989269   -0.59677
## 8  0.06418272    1.52006
## 9  0.03898776    1.18472
## 10 0.00514102    0.43021
## 11 0.00051971    0.13678
## 12 0.00068155   -0.15664
## 13 0.00002843    0.03199
## 14 0.00462725   -0.40814
## 15 0.05555149   -1.41416
## 16 0.05555149   -1.41416
## 17 0.00730845   -0.51294
## 18 0.12045197    1.70025
## 19 0.06779763    1.27560
## 20 0.16980457    2.01874
## 21 0.02311696    0.91226
```

```
## 22 0.00331137   -0.34527
## 23 0.00462725   -0.40814
## 24 0.01806153   -0.80636
## 25 0.00514102    0.43021
## 26 0.01588167    0.61738
## 27 0.00485518    0.34136
## 28 0.06779763    1.27560
## 29 0.13868158   -1.82438
## 30 0.04135920   -0.99630
## 31 0.16570811   -1.99424
## 32 0.01681944   -0.63535
```

Multivariate Linear Model:

With 10 possible predictor variables there are 2^10= 1,024 different models. To save time, I used stepFit and regsubsets from the leaps package for variable selection using AIC in stepFit and BIC in leaps. See Figures 4 and 5 for graphs of the results. Based on the multiple models both methods tested, I have chose to use additional variables of weight and quarter mile time. Using transmission type, weight and quarter mile time explains 85% of the variance.

```
leapsFit <- regsubsets(mpg ~ ., data=mtcars, nvmax=8)
leapsSummary <- summary(leapsFit); leapsSummary
```

```
## Subset selection object
## Call: regsubsets.formula(mpg ~ ., data = mtcars, nvmax = 8)
## 10 Variables  (and intercept)
##      Forced in Forced out
## cyl      FALSE      FALSE
## disp     FALSE      FALSE
## hp       FALSE      FALSE
## drat     FALSE      FALSE
## wt       FALSE      FALSE
## qsec     FALSE      FALSE
## vs1      FALSE      FALSE
## am1      FALSE      FALSE
## gear     FALSE      FALSE
## carb     FALSE      FALSE
## 1 subsets of each size up to 8
## Selection Algorithm: exhaustive
##          cyl disp hp  drat wt  qsec vs1 am1 gear carb
## 1  ( 1 ) " " " "  " " " "  "*" " "  " " " " " "  " "
## 2  ( 1 ) "*" " "  " " " "  "*" " "  " " " " " "  " "
## 3  ( 1 ) " " " "  " " " "  "*" "*"  " " "*" " "  " "
## 4  ( 1 ) " " " "  "*" " "  "*" "*"  " " "*" " "  " "
## 5  ( 1 ) " " "*"  "*" " "  "*" "*"  " " "*" " "  " "
## 6  ( 1 ) " " "*"  "*" "*"  "*" "*"  " " "*" " "  " "
## 7  ( 1 ) " " "*"  "*" "*"  "*" "*"  " " "*" "*"  " "
## 8  ( 1 ) " " "*"  "*" "*"  "*" "*"  " " "*" "*"  "*"
```

```
names(leapsSummary)
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```r
leapsSummary$rsq
```

```
## [1] 0.7528 0.8302 0.8497 0.8579 0.8637 0.8667 0.8681 0.8687
```

```r
coef(leapsFit, 3)
```

```
## (Intercept)          wt        qsec         am1
##       9.618      -3.917       1.226       2.936
```

```r
stepFit <- step(lm(mpg ~ ., data=mtcars))
```

```
## Start:  AIC=70.9
## mpg ~ cyl + disp + hp + drat + wt + qsec + vs + am + gear + carb
##
##        Df Sum of Sq RSS  AIC
## - cyl   1      0.08 148 68.9
## - vs    1      0.16 148 68.9
## - carb  1      0.41 148 69.0
## - gear  1      1.35 149 69.2
## - drat  1      1.63 149 69.2
## - disp  1      3.92 151 69.7
## - hp    1      6.84 154 70.3
## - qsec  1      8.86 156 70.8
## <none>               148 70.9
## - am    1     10.55 158 71.1
## - wt    1     27.01 174 74.3
##
## Step:  AIC=68.92
## mpg ~ disp + hp + drat + wt + qsec + vs + am + gear + carb
##
##        Df Sum of Sq RSS  AIC
## - vs    1      0.27 148 67.0
## - carb  1      0.52 148 67.0
## - gear  1      1.82 149 67.3
## - drat  1      1.98 150 67.3
## - disp  1      3.90 152 67.7
## - hp    1      7.36 155 68.5
## <none>               148 68.9
## - qsec  1     10.09 158 69.0
## - am    1     11.84 159 69.4
## - wt    1     27.03 175 72.3
##
## Step:  AIC=66.97
## mpg ~ disp + hp + drat + wt + qsec + am + gear + carb
##
##        Df Sum of Sq RSS  AIC
## - carb  1      0.69 148 65.1
## - gear  1      2.14 150 65.4
## - drat  1      2.21 150 65.4
## - disp  1      3.65 152 65.8
```

```
## - hp    1      7.11 155 66.5
## <none>              148 67.0
## - am    1     11.57 159 67.4
## - qsec  1     15.68 164 68.2
## - wt    1     27.38 175 70.4
##
## Step:  AIC=65.12
## mpg ~ disp + hp + drat + wt + qsec + am + gear
##
##        Df Sum of Sq RSS  AIC
## - gear  1       1.6 150 63.5
## - drat  1       1.9 150 63.5
## <none>              148 65.1
## - disp  1      10.1 159 65.2
## - am    1      12.3 161 65.7
## - hp    1      14.8 163 66.2
## - qsec  1      26.4 175 68.4
## - wt    1      69.1 218 75.3
##
## Step:  AIC=63.46
## mpg ~ disp + hp + drat + wt + qsec + am
##
##        Df Sum of Sq RSS  AIC
## - drat  1       3.3 153 62.2
## - disp  1       8.5 159 63.2
## <none>              150 63.5
## - hp    1      13.3 163 64.2
## - am    1      20.0 170 65.5
## - qsec  1      25.6 176 66.5
## - wt    1      67.6 218 73.4
##
## Step:  AIC=62.16
## mpg ~ disp + hp + wt + qsec + am
##
##        Df Sum of Sq RSS  AIC
## - disp  1       6.6 160 61.5
## <none>              153 62.2
## - hp    1      12.6 166 62.7
## - qsec  1      26.5 180 65.3
## - am    1      32.2 186 66.3
## - wt    1      69.0 222 72.1
##
## Step:  AIC=61.52
## mpg ~ hp + wt + qsec + am
##
##        Df Sum of Sq RSS  AIC
## - hp    1       9.2 169 61.3
## <none>              160 61.5
## - qsec  1      20.2 180 63.3
## - am    1      26.0 186 64.3
## - wt    1      78.5 239 72.3
##
## Step:  AIC=61.31
```

```
## mpg ~ wt + qsec + am
##
##        Df Sum of Sq RSS  AIC
## <none>              169 61.3
## - am    1      26.2 195 63.9
## - qsec  1     109.0 278 75.2
## - wt    1     183.3 353 82.8
```

```r
summary(stepFit)
```

```
##
## Call:
## lm(formula = mpg ~ wt + qsec + am, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.481 -1.556 -0.726  1.411  4.661
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.618      6.960    1.38  0.17792
## wt            -3.917      0.711   -5.51 0.000007
## qsec           1.226      0.289    4.25  0.00022
## am1            2.936      1.411    2.08  0.04672
##
## Residual standard error: 2.46 on 28 degrees of freedom
## Multiple R-squared:  0.85,   Adjusted R-squared:  0.834
## F-statistic: 52.7 on 3 and 28 DF,  p-value: 0.0000000000121
```

```r
finalFit <- lm(mpg ~ am + wt + qsec, data=mtcars)
confint(finalFit, level=.95)  # 95% confidence interval
```

```
##                2.5 % 97.5 %
## (Intercept) -4.63830 23.874
## am1          0.04573  5.826
## wt          -5.37333 -2.460
## qsec         0.63457  1.817
```

```r
summary(finalFit)
```

```
##
## Call:
## lm(formula = mpg ~ am + wt + qsec, data = mtcars)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.481 -1.556 -0.726  1.411  4.661
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    9.618      6.960    1.38  0.17792
```

```
## am1            2.936      1.411    2.08  0.04672
## wt            -3.917      0.711   -5.51 0.000007
## qsec           1.226      0.289    4.25  0.00022
##
## Residual standard error: 2.46 on 28 degrees of freedom
## Multiple R-squared:  0.85,   Adjusted R-squared:  0.834
## F-statistic: 52.7 on 3 and 28 DF,  p-value: 0.0000000000121
```

```r
tidy(finalFit, conf.int=TRUE)
```

```
##          term estimate std.error statistic     p.value conf.low conf.high
## 1 (Intercept)    9.618    6.9596     1.382 0.177915165 -4.63830    23.874
## 2         am1    2.936    1.4109     2.081 0.046715510  0.04573     5.826
## 3          wt   -3.917    0.7112    -5.507 0.000006953 -5.37333    -2.460
## 4        qsec    1.226    0.2887     4.247 0.000216174  0.63457     1.817
```

```r
glance(finalFit)
```

```
##   r.squared adj.r.squared sigma statistic         p.value df logLik   AIC
## 1    0.8497        0.8336 2.459     52.75 0.0000000000121  4 -72.06 154.1
##     BIC deviance df.residual
## 1 161.4    169.3          28
```

```r
augment(finalFit)
```

```
##              .rownames  mpg am    wt  qsec .fitted .se.fit  .resid    .hat
## 1            Mazda RX4 21.0  1 2.620 16.46   22.47  0.7197 -1.4705 0.08567
## 2        Mazda RX4 Wag 21.0  1 2.875 17.02   22.16  0.7435 -1.1582 0.09143
## 3           Datsun 710 22.8  1 2.320 18.61   26.28  0.7598 -3.4811 0.09548
## 4       Hornet 4 Drive 21.4  0 3.215 19.44   20.86  0.6849  0.5426 0.07759
## 5    Hornet Sportabout 18.7  0 3.440 17.02   17.01  0.7486  1.6904 0.09268
## 6              Valiant 18.1  0 3.460 20.22   20.85  0.7677 -2.7541 0.09747
## 7           Duster 360 14.3  0 3.570 15.84   15.05  0.9417 -0.7539 0.14667
## 8            Merc 240D 24.4  0 3.190 20.00   21.64  0.7466  2.7581 0.09219
## 9             Merc 230 22.8  0 3.150 22.90   25.35  1.3401 -2.5536 0.29704
## 10            Merc 280 19.2  0 3.440 18.30   18.58  0.6054  0.6213 0.06063
## 11           Merc 280C 17.8  0 3.440 18.90   19.31  0.6080 -1.5143 0.06115
## 12          Merc 450SE 16.4  0 4.070 17.40   15.01  0.6076  1.3920 0.06105
## 13          Merc 450SL 17.3  0 3.730 17.60   16.58  0.5931  0.7152 0.05818
## 14         Merc 450SLC 15.2  0 3.780 18.00   16.88  0.5663 -1.6793 0.05304
## 15  Cadillac Fleetwood 10.4  0 5.250 17.98   11.10  1.1715 -0.6976 0.22701
## 16 Lincoln Continental 10.4  0 5.424 17.82   10.22  1.2639  0.1800 0.26422
## 17   Chrysler Imperial 14.7  0 5.345 17.42   10.04  1.1783  4.6610 0.22963
## 18            Fiat 128 32.4  1 2.200 19.47   27.81  0.8784  4.5947 0.12763
## 19         Honda Civic 30.4  1 1.615 18.52   28.93  0.8470  1.4681 0.11866
## 20      Toyota Corolla 33.9  1 1.835 19.90   29.76  0.9406  4.1380 0.14635
## 21       Toyota Corona 21.5  0 2.465 20.01   24.49  0.9956 -2.9936 0.16393
## 22    Dodge Challenger 15.5  0 3.520 16.87   16.51  0.7464 -1.0124 0.09214
## 23         AMC Javelin 15.2  0 3.435 17.30   17.37  0.7051 -2.1724 0.08223
## 24          Camaro Z28 13.3  0 3.840 15.41   13.47  0.9608 -0.1693 0.15268
## 25    Pontiac Firebird 19.2  0 3.845 17.05   15.46  0.6413  3.7398 0.06803
```

```
## 26          Fiat X1-9 27.3  1 1.935 18.90   28.14  0.7963 -0.8444 0.10489
## 27       Porsche 914-2 26.0  1 2.140 16.70   24.64  0.7573  1.3554 0.09485
## 28        Lotus Europa 30.4  1 1.513 16.90   27.35  0.9855  3.0546 0.16065
## 29      Ford Pantera L 15.8  1 3.170 14.50   17.91  1.0071 -2.1136 0.16775
## 30        Ferrari Dino 19.7  1 2.770 15.50   20.71  0.8296 -1.0061 0.11382
## 31       Maserati Bora 15.0  1 3.570 14.60   16.47  1.0746 -1.4696 0.19098
## 32          Volvo 142E 21.4  1 2.780 18.60   24.47  0.8668 -3.0672 0.12428
##    .sigma   .cooksd .std.resid
## 1   2.486 0.0091619   -0.62542
## 2   2.493 0.0061443   -0.49419
## 3   2.403 0.0584744   -1.48858
## 4   2.502 0.0011099    0.22975
## 5   2.481 0.0133031    0.72174
## 6   2.441 0.0375326   -1.17901
## 7   2.499 0.0047336   -0.33191
## 8   2.441 0.0351912    1.17731
## 9   2.434 0.1620827   -1.23866
## 10  2.501 0.0010966    0.26070
## 11  2.486 0.0065777   -0.63558
## 12  2.489 0.0055485    0.58422
## 13  2.500 0.0013872    0.29971
## 14  2.482 0.0068974   -0.70185
## 15  2.499 0.0076443   -0.32268
## 16  2.504 0.0006542    0.08537
## 17  2.286 0.3475974    2.15973
## 18  2.318 0.1464019    2.00067
## 19  2.486 0.0136146    0.63600
## 20  2.351 0.1421983    1.82147
## 21  2.423 0.0869043   -1.33149
## 22  2.496 0.0047376   -0.43212
## 23  2.466 0.0190521   -0.92224
## 24  2.504 0.0002521   -0.07480
## 25  2.390 0.0452966    1.57550
## 26  2.498 0.0038600   -0.36299
## 27  2.489 0.0087942    0.57940
## 28  2.420 0.0879746    1.35596
## 29  2.464 0.0447393   -0.94227
## 30  2.496 0.0060670   -0.43467
## 31  2.484 0.0260598   -0.66451
## 32  2.423 0.0630462   -1.33300
```

Diagnostics and Inference:

The residual plots in Figure 6 show there is little heteroscadicity, they are mostly normal, there does not appear to be any significant outliers and there does not appear to be any data points with high influence or high leverage.

```
anova(fit1, finalFit)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ am
## Model 2: mpg ~ am + wt + qsec
```

```
##   Res.Df RSS Df Sum of Sq    F       Pr(>F)
## 1     30 721
## 2     28 169  2       552 45.6 0.0000000016
```

The results of the anova are statistically significant at the 95% level and suggests that the multivariable model is better by looking at the residual sum of squares.

```r
vif(finalFit)
```

```
##    am    wt  qsec
## 2.541 2.483 1.364
```

The variance inflation factor suggests there is only moderate correlation between the variables selected which suggests there is not a large amount of collinearity between the variables.

```r
PRESS(fit1)
```

```
## [1] 830.3
```

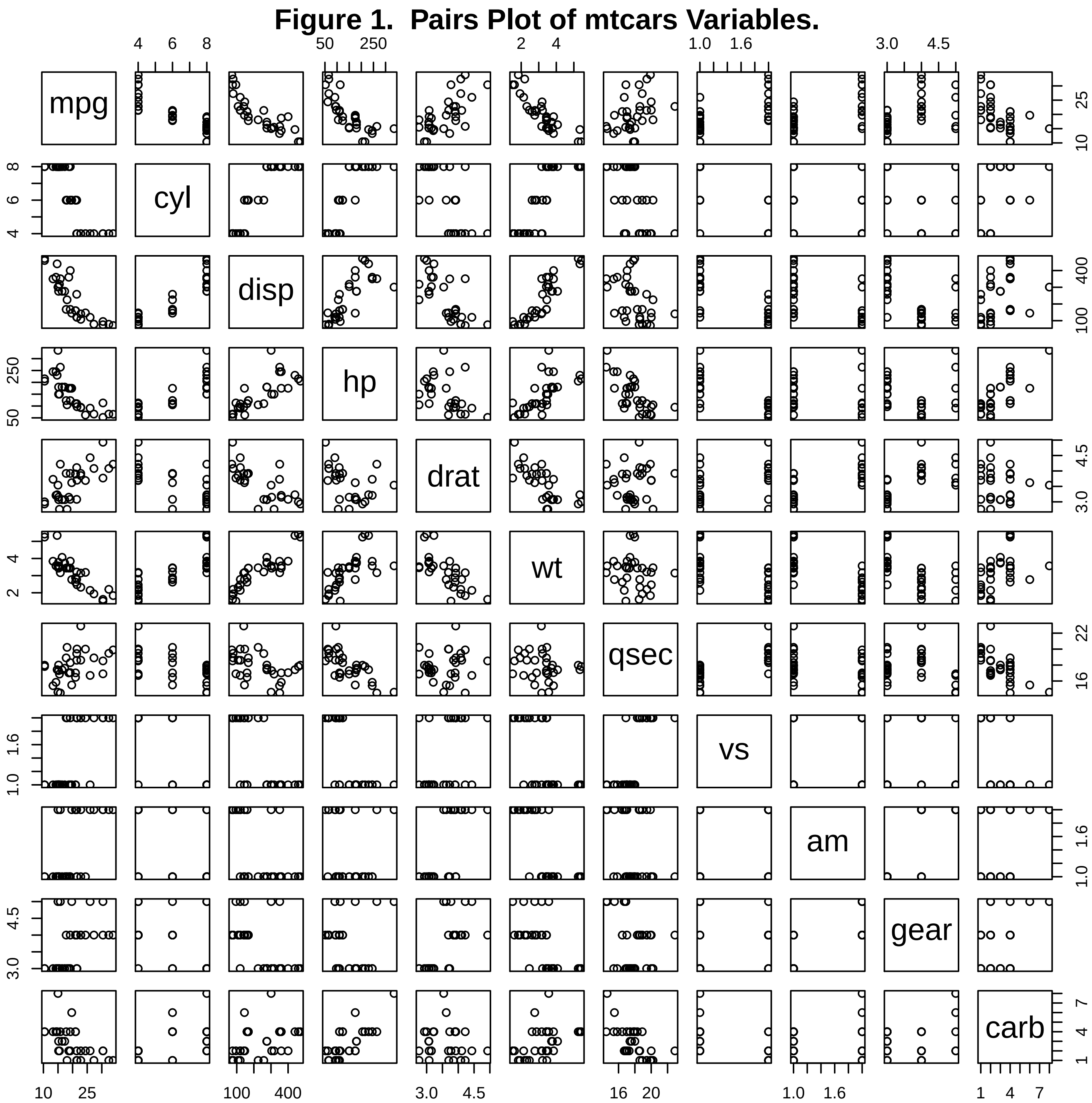
```r
PRESS(finalFit)
```

```
## [1] 231.3
```

The PRESS statistic is less for the multivariable linear model which suggest is has higher predictive ability due to less predictive error than the simple linear model.

Appendix:

```r
pairs(mtcars, main = "Figure 1.  Pairs Plot of mtcars Variables.", line.main=1.5, oma=c(2,2,3,2))
```




```
ggpairs(mtcars[ ,c(1,6,7,9)], lower=list(continuous="smooth"), colour="am", alpha=0.4,
        params = c(binwidth=.4), title="Figure 2.  Ggpairs Plot of Selected mtcars Variables.")
```

Figure 2. Ggpairs Plot of Selected mtcars Variables.

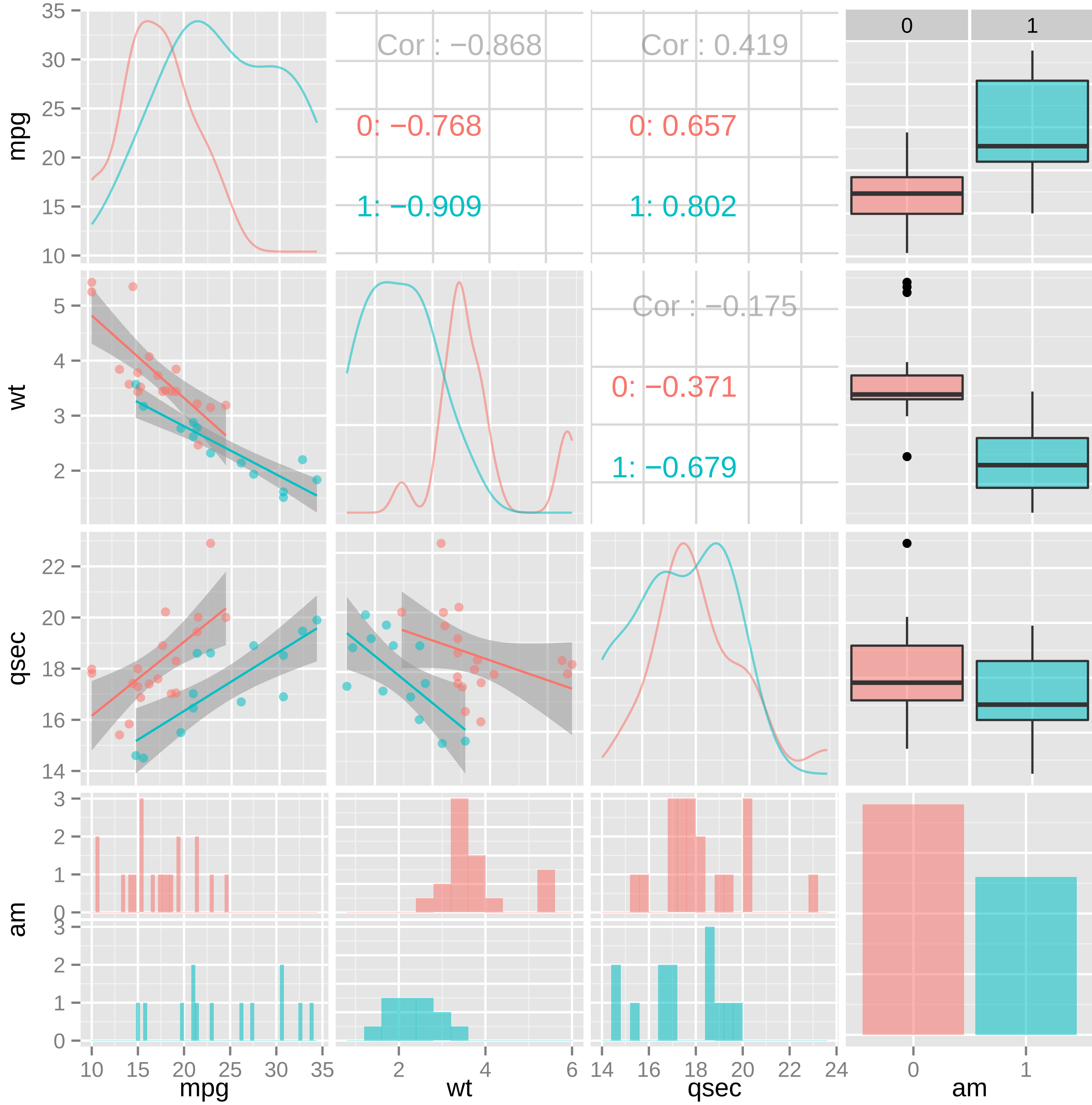


Figure 3. Residuals and Diagnostics Plots for Simple Linear Model.

```
par(mfrow=c(3,2)); plot(fit1, which=1:6)
```

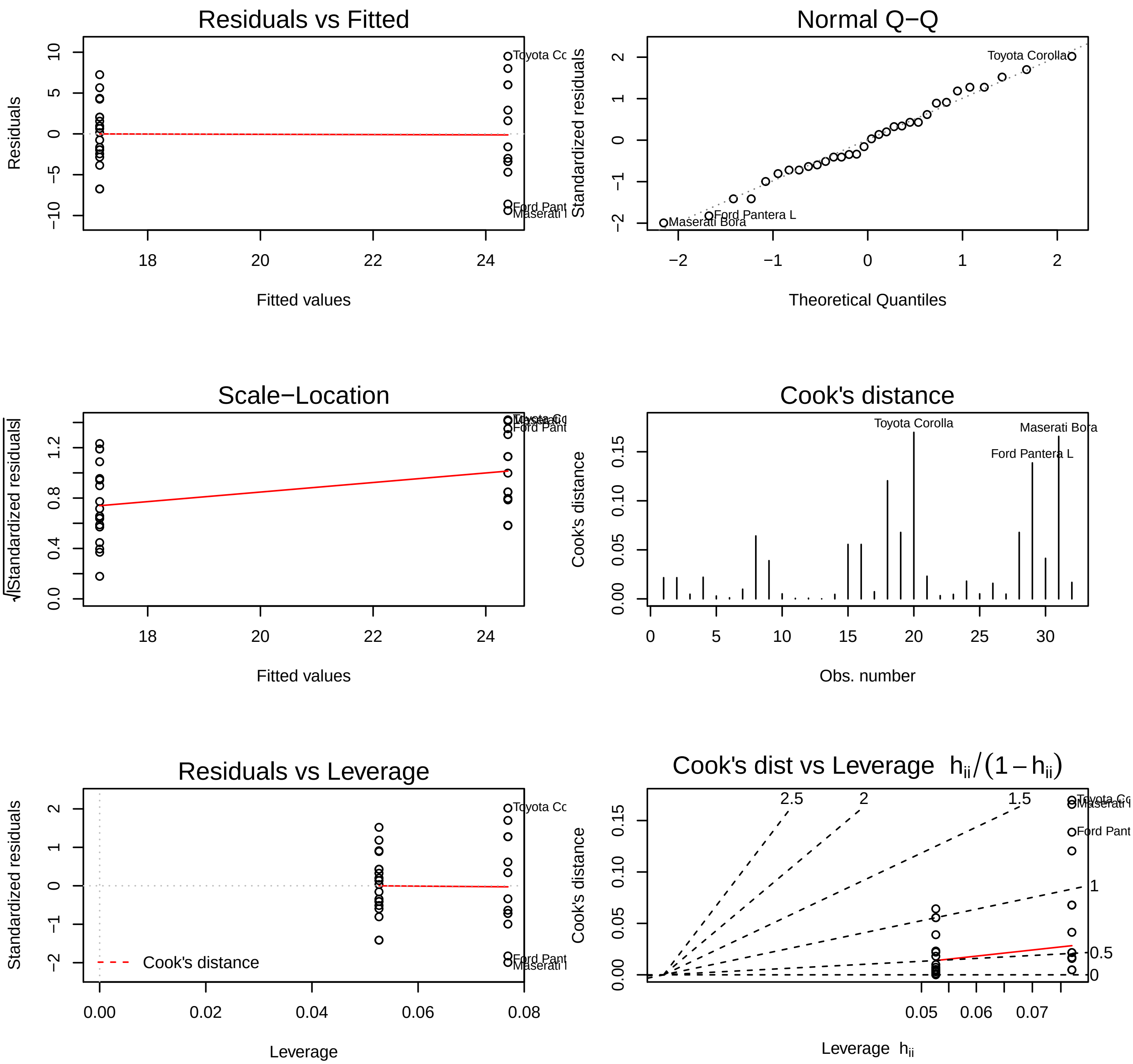


Figure 4. Multivariate Variable Selection.

```
par(mfrow=c(2,2))
plot(leapsSummary$rss, xlab="Number of Variables", ylab="RSS", type="l")
plot(leapsSummary$adjr2, xlab="Number of Variables", ylab="Adjusted RSq", type="l")
points(which.max(leapsSummary$adjr2),
       leapsSummary$adjr2[which.max(leapsSummary$adjr2)], col="red", cex=2, pch=20)
plot(leapsSummary$cp,xlab="Number of Variables", ylab="Cp", type="l")
points(which.min(leapsSummary$cp), leapsSummary$cp[which.min(leapsSummary$cp)],
       col="red", cex=2, pch=20)
plot(leapsSummary$bic, xlab="Number of Variables", ylab="BIC", type="l")
points(which.min(leapsSummary$bic), leapsSummary$bic[which.min(leapsSummary$bic)],
```

```
        col="red", cex=2, pch=20)
```

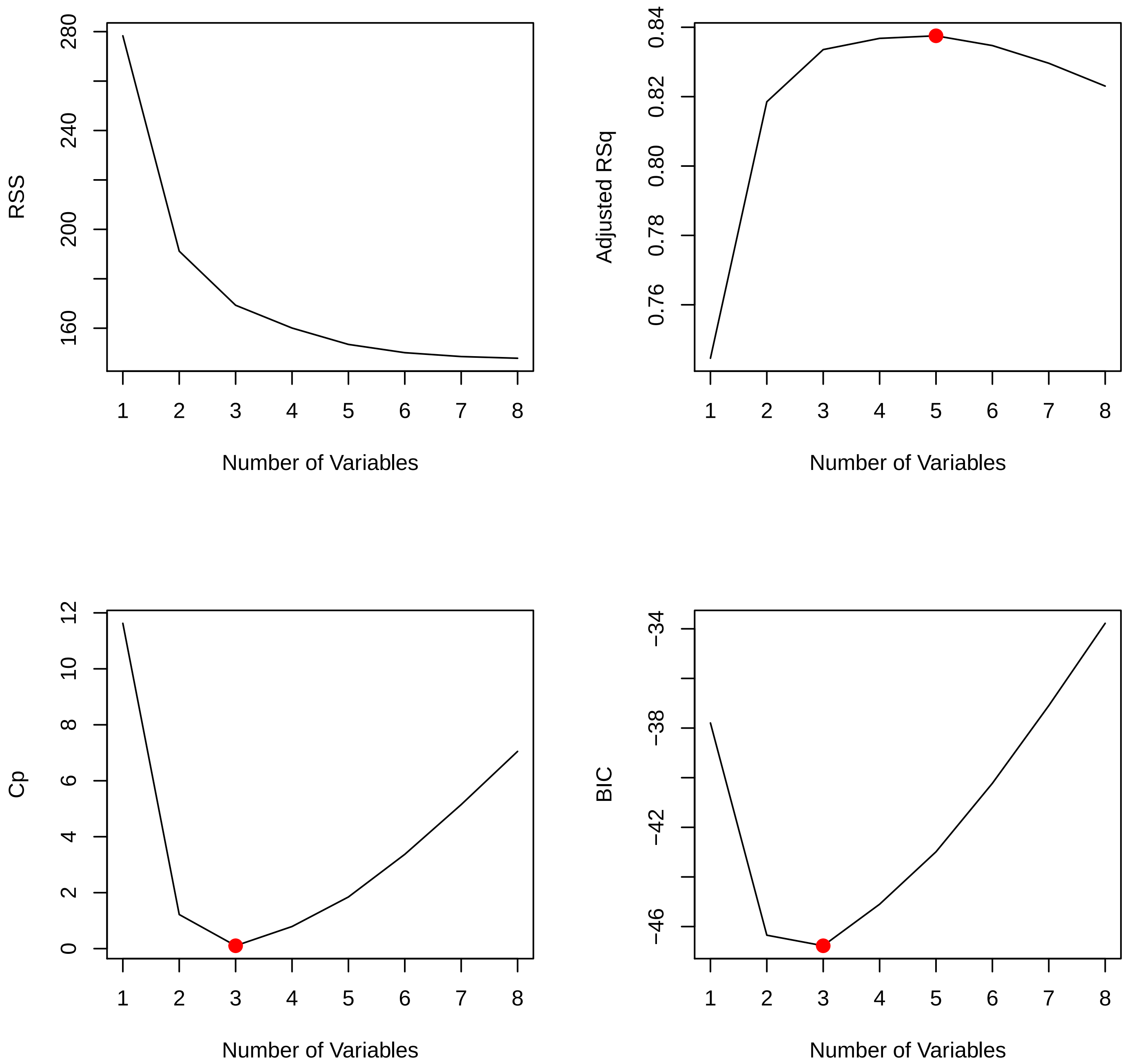


Figure 5. Multivariate Variables.

```
par(mfrow=c(2,2))
plot(leapsFit, scale="r2")
plot(leapsFit, scale="adjr2")
plot(leapsFit, scale="Cp")
plot(leapsFit, scale="bic")
```

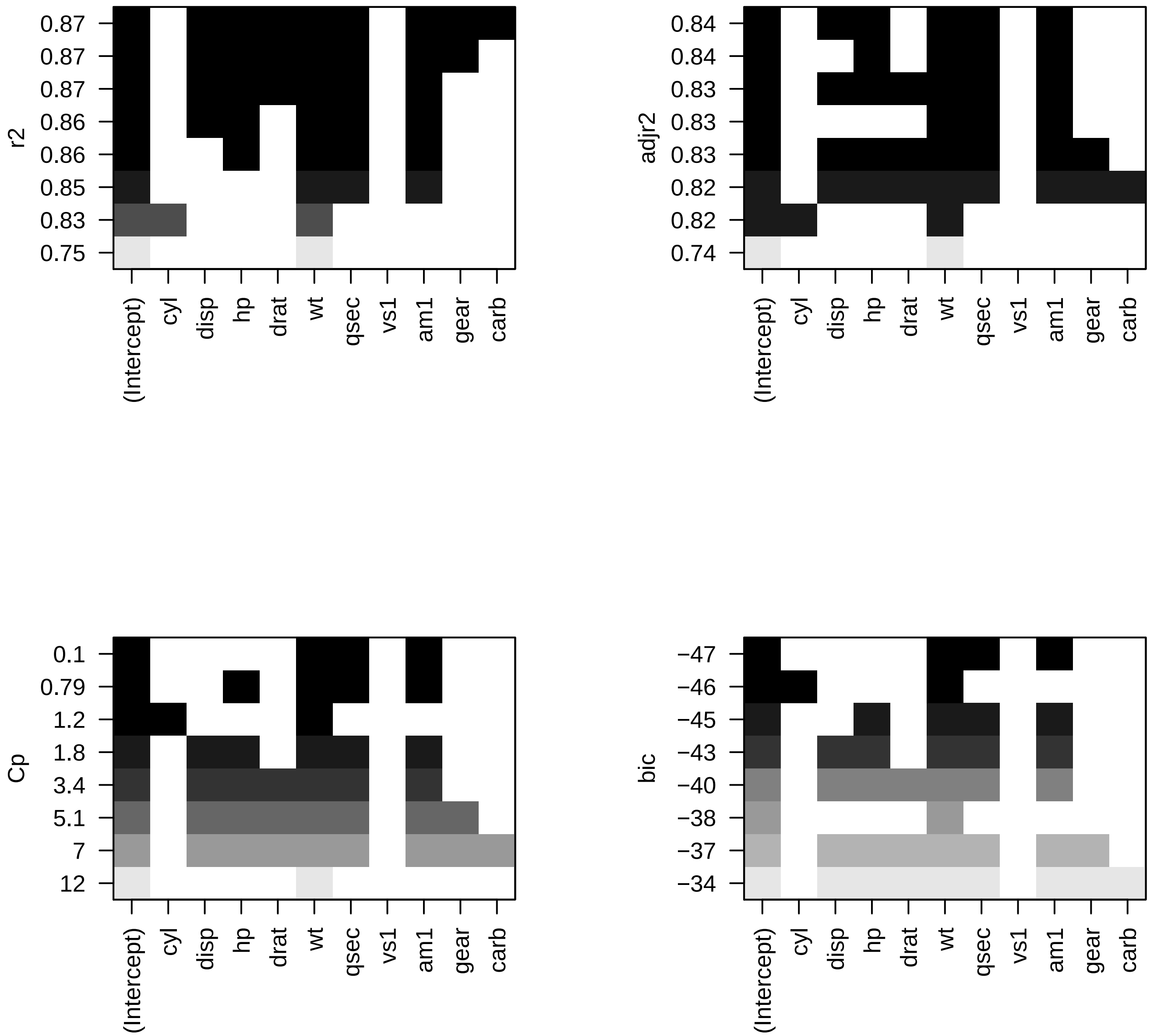
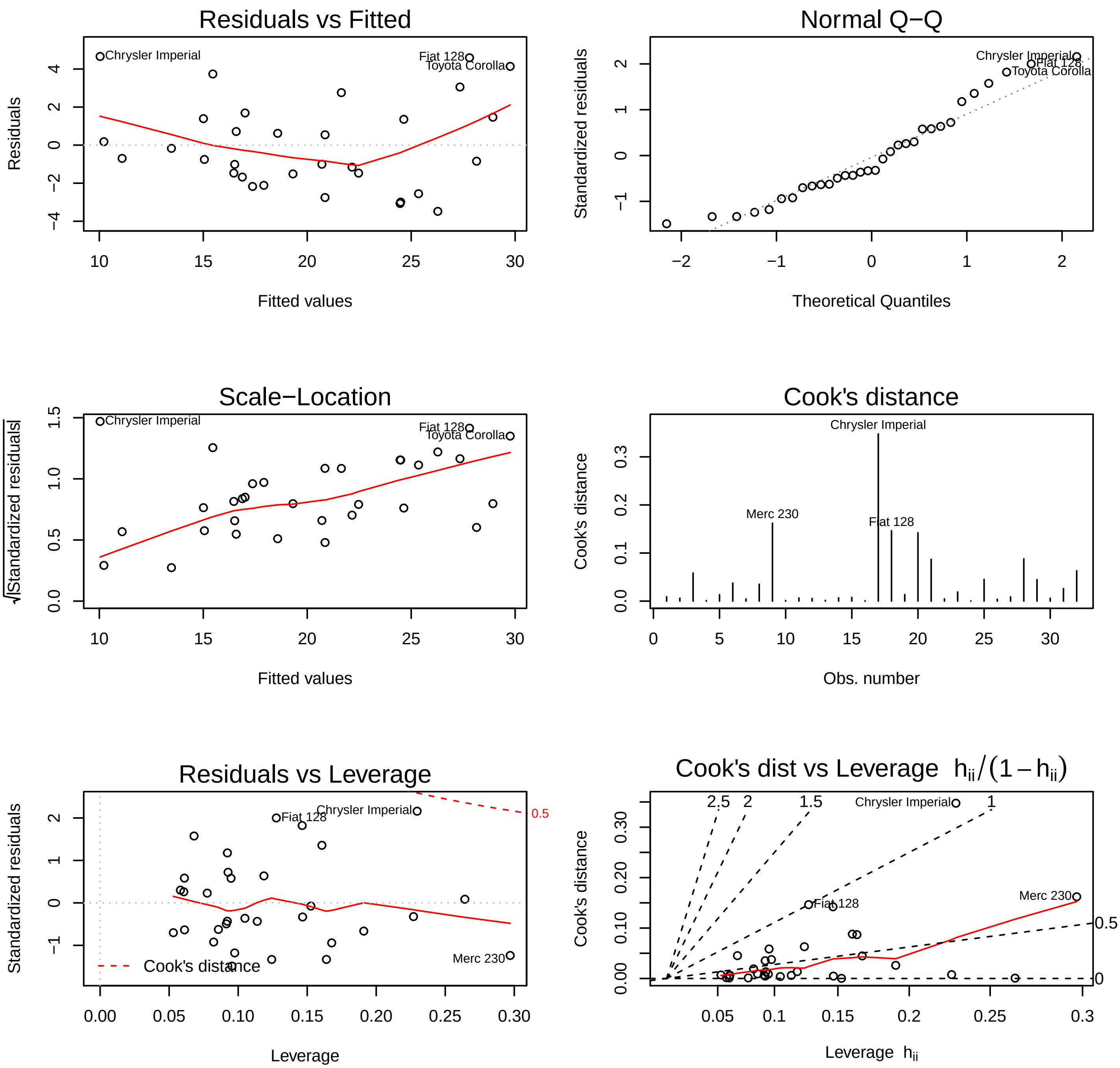


Figure 6. Residuals and Diagnostics Plots for Final Multivariable Linear Model.

```r
par(mfrow=c(3,2)); plot(finalFit, which=1:6)
```


Residuals vs Fitted
Residuals
Fitted values
Chrysler Imperial
Fiat 128
Toyota Corolla
Normal Q−Q
Standardized residuals
Theoretical Quantiles
Scale−Location
√|Standardized residuals|
Cook's distance
Obs. number
Merc 230
Residuals vs Leverage
Leverage
Cook's distance
0.5
Cook's dist vs Leverage $h_{ii}/(1-h_{ii})$
Leverage h_{ii}
2.5
2
1.5
1
0