Motor Trend magazine - Data analysis of influence on MPG for Automatic vs. Manual Transmission.

by jmvilaverde

Thursday, June 18, 2015

Executive summary (First paragraph)

For all models evaluated that have a P-value Manual transmission is better for MPG...

Data adquisition

1. Data adquisition and Initial structure analysis

```
data(mtcars)

#Show colnames of mtcars
names(mtcars)
```

1.1. Variables

```
##  [1] "mpg"  "cyl"  "disp" "hp"   "drat" "wt"   "qsec" "vs"   "am"   "gear"
## [11] "carb"
```

?mtcars

Format: *A data frame with 32 observations on 11 variables.*

Variable	Units	Values
mpg	**Miles/(US) gallon**	
cyl	Number of cylinders (4,6,8)	4, 6, 8
disp	Displacement (cu.in.)	
hp	Gross horsepower	
drat	Rear axle ratio	
wt	Weight (lb/1000)	
qsec	1/4 mile time	
vs	V/S -> V motor or straight motor	0, 1
am	**Transmission (0 = automatic, 1 = manual)**	0, 1
gear	Number of forward gears	3, 4, 5
carb	Number of carburetors	1, 2, 3, 4, 6, 8

```
#Show structure of mtcars
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 ...
```

2.Initial analysis

One predictor Selected variables:

- Predictor: X = am, Transmission with values 0 for automatic, 1 for manual.
- Outcome: Y = mpg, Miles/(US) gallon

Linear regression model formula: Y = $\beta_0 + \beta_1 X$ -> mpg = $\beta_0 + \beta_1 am$

Create initial model:

- Calculate β_0 (Intercept): $\hat{\beta}_0 = \bar{Y} + \hat{\beta}_1 \bar{X}$
- Calculate β_1 (Slope): $\hat{\beta}_1 = Cor(Y, X) \frac{Sd(Y)}{Sd(X)}$

Fitting the best line: $\Sigma_n^{i=1}(Y_i - (\beta_0 + \beta_1 X_i))^2$ to minimize the distance.

```
#Theoretical formula
y <- mtcars$mpg
x <- mtcars$am
beta1 <- cor(y,x)*(sd(y)/sd(x))
beta0 <- mean(y) - beta1 * mean(x)

#R function
model.Initial <- lm(mpg ~ am, data=mtcars)
coeffs <- coef(model.Initial)
summary(model.Initial)
```

```
##
## Call:
## lm(formula = mpg ~ am, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.3923 -3.0923 -0.2974  3.2439  9.5077
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   17.147      1.125  15.247 1.13e-15 ***
## am             7.245      1.764   4.106 0.000285 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.902 on 30 degrees of freedom
## Multiple R-squared:  0.3598, Adjusted R-squared:  0.3385
## F-statistic: 16.86 on 1 and 30 DF,  p-value: 0.000285
```

```
#Comparation
rbind(c(beta0, beta1), coeffs)
```

```
##        (Intercept)       am
##           17.14737 7.244939
## coeffs    17.14737 7.244939
```

For linear model $Y = \beta_0 + \beta_1 X$ -> $mpg = \beta_0 + \beta_1 am$

- Intercept $\beta_0 = 17.1473684$
- Slope $\beta_1 = 7.2449393$

  With this linear model we have a P-value under 0.05 on Intercept and all the coefficents and overall model. R^2 is 0.3598 for the model, this low value indicates that the model is not a good fit to the data.

  For this linear regression model, the expected difference in MPG is 7.2449393 when the car have manual transmission in comparation to the same car with automatic transmission. The Intercept 17.1473684 is the expected MPG of a automatic transmission car.

Multiple predictor In multivariable regression analysis you must evaluate the consecuences to throwing variables that aren't related to the outcome and consecuences to omitting variables that are related to the outcome.

Multivariable linear model formula:

- $Y_i = \beta_1 X_{1i} + \beta_2 X_{2i} + ... + \beta_p X_{pi} + \epsilon_i = \Sigma_{k=1}^{p} X_{ik} \beta_j + \epsilon_i$

Model Complete: $mpg = \beta_{cyl} cyl + \beta_{dips} disp + \beta_{hp} hp + \beta_{drat} drat + \beta_{wt} wt + \beta_{qsec} qsec + \beta_{vs} vs + \beta_{am} am + \beta_{gear} gear + \beta_{carb} carb$

```
model.Complete <- lm(mpg ~ ., data=mtcars)
beta.am.Complete <- coef(model.Complete)["am"]
summary(model.Complete)
```

```
##
## Call:
## lm(formula = mpg ~ ., data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4506 -1.6044 -0.1196  1.2193  4.6271
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) 12.30337   18.71788   0.657   0.5181
## cyl         -0.11144    1.04502  -0.107   0.9161
## disp         0.01334    0.01786   0.747   0.4635
## hp          -0.02148    0.02177  -0.987   0.3350
## drat         0.78711    1.63537   0.481   0.6353
## wt          -3.71530    1.89441  -1.961   0.0633 .
## qsec         0.82104    0.73084   1.123   0.2739
## vs           0.31776    2.10451   0.151   0.8814
## am           2.52023    2.05665   1.225   0.2340
## gear         0.65541    1.49326   0.439   0.6652
## carb        -0.19942    0.82875  -0.241   0.8122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.65 on 21 degrees of freedom
## Multiple R-squared:  0.869,  Adjusted R-squared:  0.8066
## F-statistic: 13.93 on 10 and 21 DF,  p-value: 3.793e-07
```

With this linear model we have a P-value over 0.05 on Intercept and all the coefficents, and under 0.05 for overall model. R^2 is 0.869 for the model, this high value indicates that the model is a good fit to the data.

For linear regression model *Complete*, the expected difference in MPG is 2.5202269 when the car have manual transmission in comparation to the same car with automatic transmission.

```
#Confidence Interval
sumCoef.Complete <- summary(model.Complete)$coef
confInterval.Model.Complete <- sumCoef.Complete[9,1] + c(-1,1) * qt(0.975, df=model.Complete$df) * sumC
```

Model to be proposed. Define a model based on Variation Inflation Factor analysis to check collinearity between variables:

```
library(car)
vif.Complete <- vif(model.Complete)
vif.Complete
```

```
##       cyl      disp        hp      drat        wt      qsec        vs
## 15.373833 21.620241  9.832037  3.374620 15.164887  7.527958  4.965873
##        am      gear      carb
##  4.648487  5.357452  7.908747
```

VIF	Indication
1	No correlation
$1 < VIF < 5$	moderate correlation
$VIF > 5$	strong correlation

Remove variables with strong correlation: Use the other variables:

Proposed model: $mpg = \beta_{prop.hp} hp + \beta_{prop.drat} drat + \beta_{prop.vs} vs + \beta_{prop.am} am$

```
model.Prop <- lm(formula = mpg ~ drat + vs + am, data = mtcars)
beta.am.Prop <- coef(model.Prop)["am"]
vif.Prop <- vif(model.Prop)
sqrt(vif.Prop)
```

```
##     drat       vs       am
## 1.608598 1.144700 1.465209
```

```
summary(model.Prop)
```

```
##
## Call:
## lm(formula = mpg ~ drat + vs + am, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.9892 -2.6090  0.2629  2.1127  6.2924
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    8.327      6.017   1.384 0.177316
## drat           1.985      1.883   1.054 0.300772
## vs             6.235      1.421   4.387 0.000148 ***
## am             4.669      1.838   2.540 0.016898 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.485 on 28 degrees of freedom
## Multiple R-squared:  0.6981, Adjusted R-squared:  0.6657
## F-statistic: 21.58 on 3 and 28 DF,  p-value: 1.922e-07
```

Bad summary > For linear regression model *Prop*, the expected difference in MPG is 4.6687251 when the car have manual transmission in comparation to the same car with automatic transmission.

Obtain a model by R function step:

```
step(model.Complete, trace=FALSE)
```

```
##
## Call:
## lm(formula = mpg ~ wt + qsec + am, data = mtcars)
##
## Coefficients:
## (Intercept)           wt         qsec           am
##       9.618       -3.917        1.226        2.936
```

```
model.Step <- lm(formula = mpg ~ wt + qsec + am, data = mtcars)
step(model.Complete, direction="forward", trace=FALSE)
```

```
##
## Call:
```

```
## lm(formula = mpg ~ cyl + disp + hp + drat + wt + qsec + vs +
##     am + gear + carb, data = mtcars)
##
## Coefficients:
## (Intercept)          cyl         disp           hp         drat
##    12.30337     -0.11144      0.01334     -0.02148      0.78711
##          wt         qsec           vs           am         gear
##    -3.71530      0.82104      0.31776      2.52023      0.65541
##        carb
##    -0.19942
```

```
summary(model.Step)
```

```
##
## Call:
## lm(formula = mpg ~ wt + qsec + am, data = mtcars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.4811 -1.5555 -0.7257  1.4110  4.6610
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.6178     6.9596   1.382 0.177915
## wt           -3.9165     0.7112  -5.507 6.95e-06 ***
## qsec          1.2259     0.2887   4.247 0.000216 ***
## am            2.9358     1.4109   2.081 0.046716 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.459 on 28 degrees of freedom
## Multiple R-squared:  0.8497, Adjusted R-squared:  0.8336
## F-statistic: 52.75 on 3 and 28 DF,  p-value: 1.21e-11
```

Do a comparation between the 3 models

```
library(car)
anova(model.Initial, model.Prop, model.Step, model.Complete)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ am
## Model 2: mpg ~ drat + vs + am
## Model 3: mpg ~ wt + qsec + am
## Model 4: mpg ~ cyl + disp + hp + drat + wt + qsec + vs + am + gear + carb
##   Res.Df    RSS Df Sum of Sq       F    Pr(>F)
## 1     30 720.90
## 2     28 339.99  2    380.91 27.1164 1.517e-06 ***
## 3     28 169.29  0    170.70
## 4     21 147.49  7     21.79  0.4432    0.8636
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

View Figure 2 for plot with regression line.

3.Basic regression model with additive Gaussian errors. Into point 2 is obtained a linear regression model that fits the data, but doesn't take in consideration the impact of the others variables. We are going to analyze gaussian errors for Initial Model.

Selected variables:

- **Predictor**: X = am, Transmission with values 0 for automatic, 1 for manual.
- **Outcome**: Y = mpg, Miles/(US) gallon

Probabilistic model for linear regression:

- $Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$ -> $mpg_i = \beta_0 + \beta_1 am_i + \epsilon_i$
- ϵ_i are assumed iid $N(\mu_i, \sigma^2)$.
- Note, $E[Y_i|X_i = x_i] = \mu_i = \beta_0 + \beta_1 x_i$ and $Var(Y_i|X_i = x_i) = \sigma^2$.
- $\hat{\beta}_0 = \bar{Y} + \hat{\beta}_1 \bar{X}$ and $\hat{\beta}_1 = Cor(Y, X)\frac{Sd(Y)}{Sd(X)}$.

Residuals analysis:

- Observed outcome i is Y_i at a predictor value X_i.
- Predicted outcome i is $\hat{Y}_i$ at a predictor value X_i is $\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$.
- Residual is the difference between observed an predicted: $e_i = Y_i - \hat{Y}$, the vertical distance between the observed data point and the regression line.
- Least squares minimizes $\Sigma_{i=1}^{n} e_i^2$.
- e_i can be thought of as estimates of the ϵ_i.

```
#Calculate residuals
e.ModelInitial <- y-(beta0+beta1*x)
#R function for residuals: resid(model.Initial)

#Calculate predicted y (mpg)
yhat <- predict(model.Initial)
#Calculate max difference between residual and observed Y - predicted Y (y hat)
max(abs(e.ModelInitial -(y - yhat)))
```

```
## [1] 3.552714e-15
```

```
#Calculate residuals
e.ModelComplete <- resid(model.Complete)

#Calculate predicted y (mpg)
yhat <- predict(model.Complete)
#Calculate max difference between residual and observed Y - predicted Y (y hat)
max(abs(e.ModelComplete -(y - yhat)))
```

```
## [1] 6.57252e-14
```

```
#Calculate residuals
e.Model.Step <- resid(model.Step)

#Calculate predicted y (mpg)
yhat <- predict(model.Step)
#Calculate max difference between residual and observed Y - predicted Y (y hat)
max(abs(e.Model.Step -(y - yhat)))
```

```
## [1] 6.439294e-14
```

Figure 3 Plot of residuals:

Formula to estimate residual variation:

- ML estimate of σ^2 is $\frac{1}{n}\Sigma_{i=1}^{n} e_i^2$
- For $E[\hat{\sigma}^2] = \sigma^2$ most people use $\frac{1}{n-2}\Sigma_{i=1}^{n} e_i^2$

```
#Calculate variation
n <- length(y)
var.e.Model.Initial <- sqrt((1/(n-2))*sum((e.ModelInitial^2)))
var.e.Model.Initial
```

```
## [1] 4.902029
```

```
#R function to calculate residual variation
summary(model.Initial)$sigma
```

```
## [1] 4.902029
```

Model Initial has a residual variation of 4.9020288.

Total variation. Formula Total variation = Residual variation + Regression Variation:

- $\Sigma_{i=1}^{n}(Y_i - \bar{Y})^2 = \Sigma_{i=1}^{n}(Y_i - \hat{Y}_i)^2 + \Sigma_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2$

Define the percent of total variation described by the model as:

- $R^2 = \frac{\Sigma_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2}{\Sigma_{i=1}^{n}(Y_i - \bar{Y})^2} = 1 - \frac{\Sigma_{i=1}^{n}(Y_i - \hat{Y}_i)^2}{\Sigma_{i=1}^{n}(Y_i - \bar{Y})^2}$

Relation between R^2 and r (the correlation):

Recall that: $(\hat{Y}_i - \bar{Y}) = \hat{\beta}_1(X_i - \bar{X})$ so that

- $R^2 = \frac{\Sigma_{i=1}^{n}(\hat{Y}_i - \bar{Y})^2}{\Sigma_{i=1}^{n}(Y_i - \bar{Y})^2} = \hat{\beta}_1^2 \frac{\Sigma_{i=1}^{n}(X_i - \bar{X})}{\Sigma_{i=1}^{n}(Y_i - \bar{Y})^2} = Cor(Y, X)^2$

```
#Calculate R2
R2.Model.Initial <- sum((yhat - mean(y))^2)/sum((y-mean(y))^2)
R2.Model.Initial
```

```
## [1] 0.8496636
```

```
#R function for R2
cor(y,x)^2
```

```
## [1] 0.3597989
```

Inference in regression

Create confidence intervals and perform hypothesis tests.

```r
#Calculation of coefficients
sigma <- var.e.Model.Initial
ssx <- sum((x - mean(x))^2)
seBeta0 <- (1/n + mean(x)^2/ssx) ^ 0.5 * sigma
seBeta1 <- sigma / sqrt(ssx)
tBeta0 <- beta0 / seBeta0
tBeta1 <- beta1 / seBeta1
pBeta0 <- 2 * pt(abs(tBeta0), df=n-2, lower.tail=FALSE)
pBeta1 <- 2 * pt(abs(tBeta1), df=n-2, lower.tail=FALSE)
coefTable <- rbind(c(beta0, seBeta0, tBeta0, pBeta0), c(beta1, seBeta1, tBeta1, pBeta1))
colnames(coefTable) <- c("Estimate","Std.Error", "t value", "P(>|t|)")
rownames(coefTable) <- c("(Intercept)", "x")
coefTable
```

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

```r
#R function for calculate coefficients
summary(model.Initial)$coef
```

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

Final Analysis

Is an automatic or manual transmission better for MPG?

For model Initial, model Complete and model Step:

> The manual transmission is better for MPG.

Quantify the MPG difference between automatic and manual transmissions.

Model Initial:

```r
sumCoef <- summary(model.Initial)$coef
confInterval.Model.Initial <- sumCoef[1,1] + c(-1,1) * qt(0.975, df=model.Initial$df) * sumCoef[1,2]
```

> With 95% confidence, we estimate that a manual transmission results in a 14.8506236 to 19.4441132 increase in MPG comparing to use of automatic transmission for the Model Initial.

Model Complete:

```r
sumCoef.Complete <- summary(model.Complete)$coef
confInterval.Model.Complete <- sumCoef.Complete[1,1] + c(-1,1) * qt(0.975, df=model.Complete$df) * sumC
```

> With 95% confidence, we estimate that a manual transmission results in a -26.6225974 to 51.2293458 increase in MPG comparing to use of automatic transmission for the Model Complete.

Model Proposed:

```
sumCoef.Step <- summary(model.Step)$coef
confInterval.Model.Step <- sumCoef.Step[1,1] + c(-1,1) * qt(0.975, df=model.Step$df) * sumCoef.Step[1,2]
```

```
sumCoef.Prop <- summary(model.Prop)$coef
confInterval.Model.Prop <- sumCoef.Prop[1,1] + c(-1,1) * qt(0.975, df=model.Prop$df) * sumCoef.Prop[1,2]
```

> With 95% confidence, we estimate that a manual transmission results in a -4.6382995 to 23.8738605 increase in MPG comparing to use of automatic transmission for the Model Step.

Main Body + Apendix only figures (not more than 5)

Figure :

```
pairs(mtcars, panel = panel.smooth, main = "mtcars data", col=3+(mtcars$am>0))
```

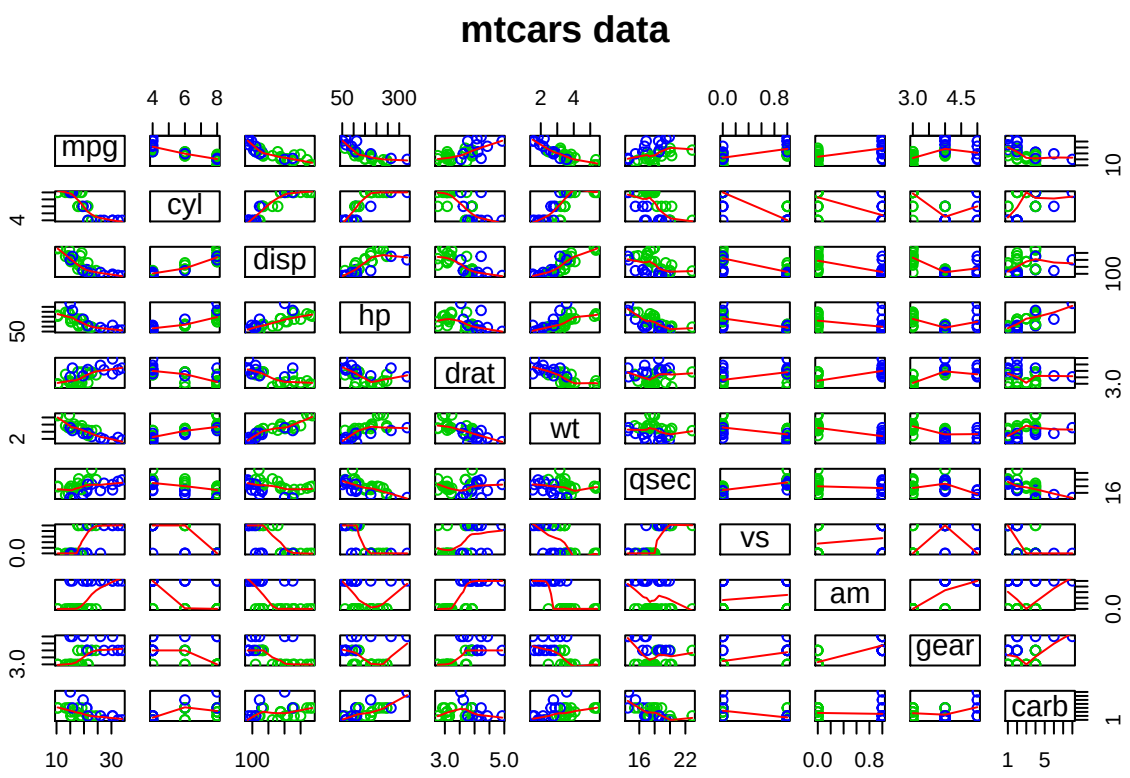
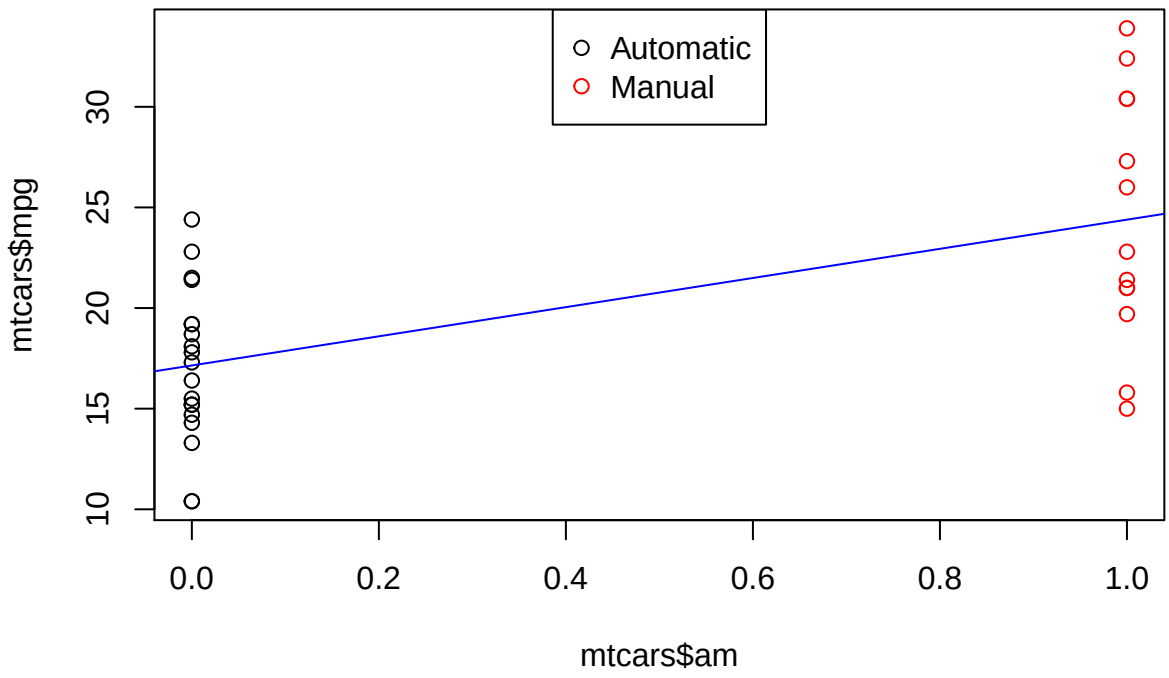
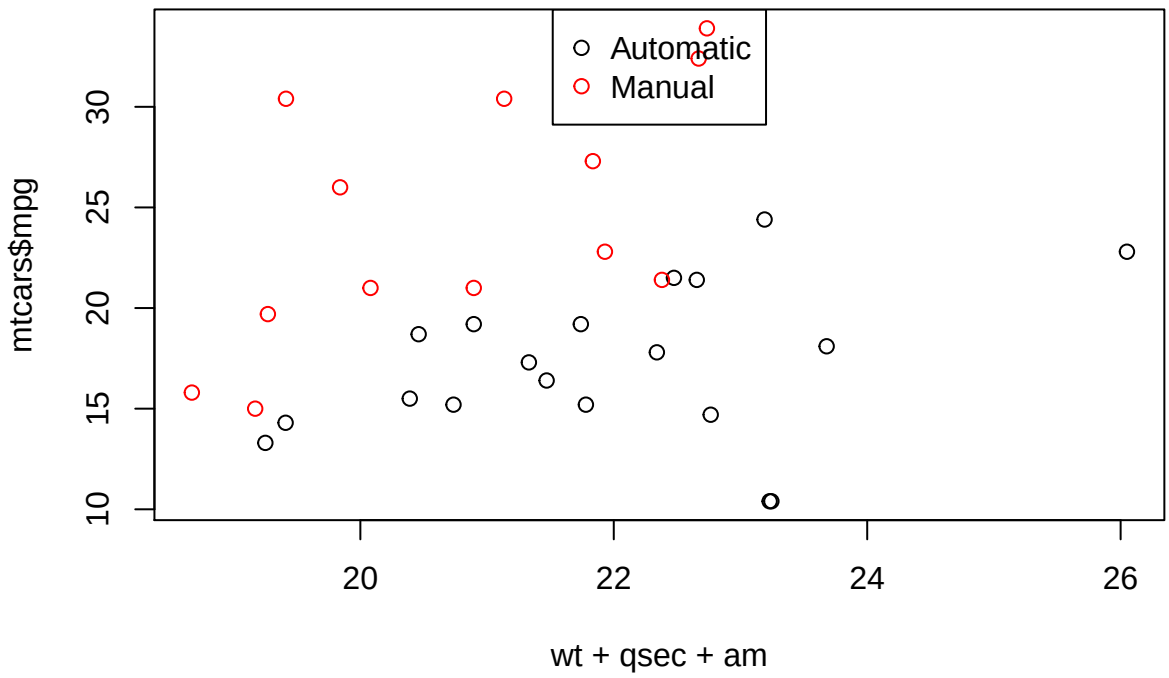


Figure 2:

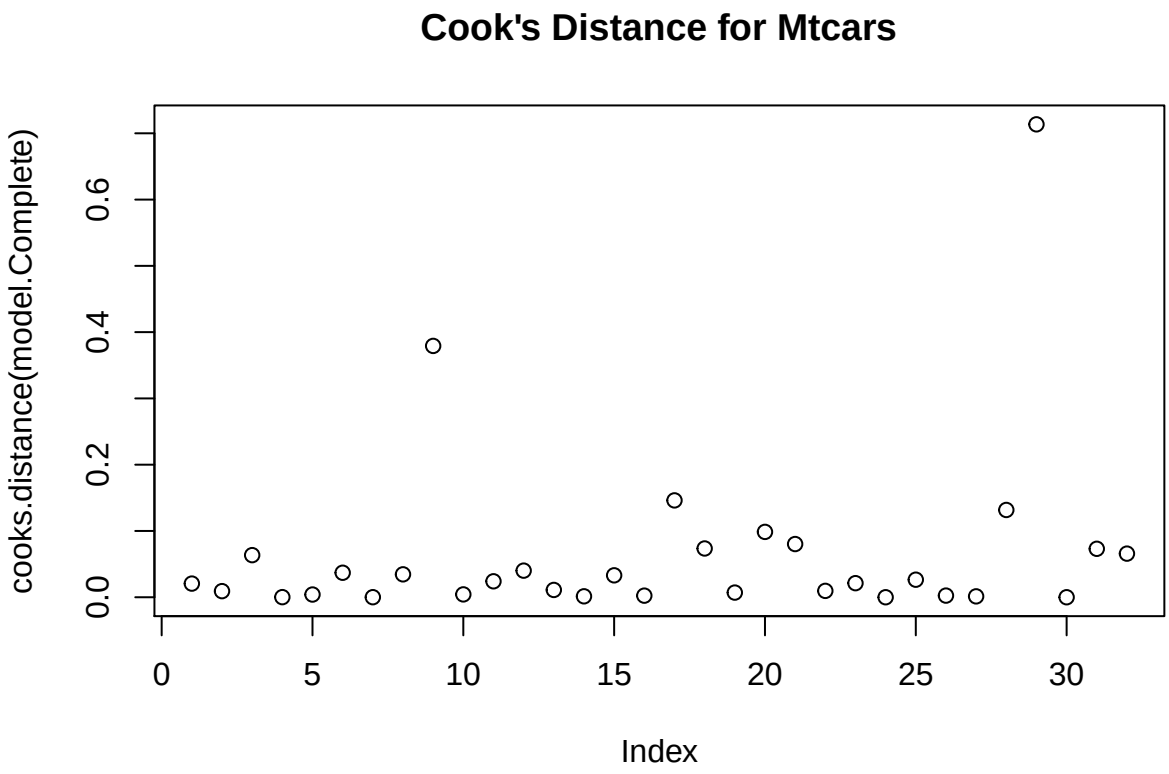
```
plot(x=mtcars$am, y=mtcars$mpg,col=mtcars$am+1)
legend("top",c("Automatic","Manual"),col=c(1,2),pch=1)
abline(model.Initial, col="blue")
```



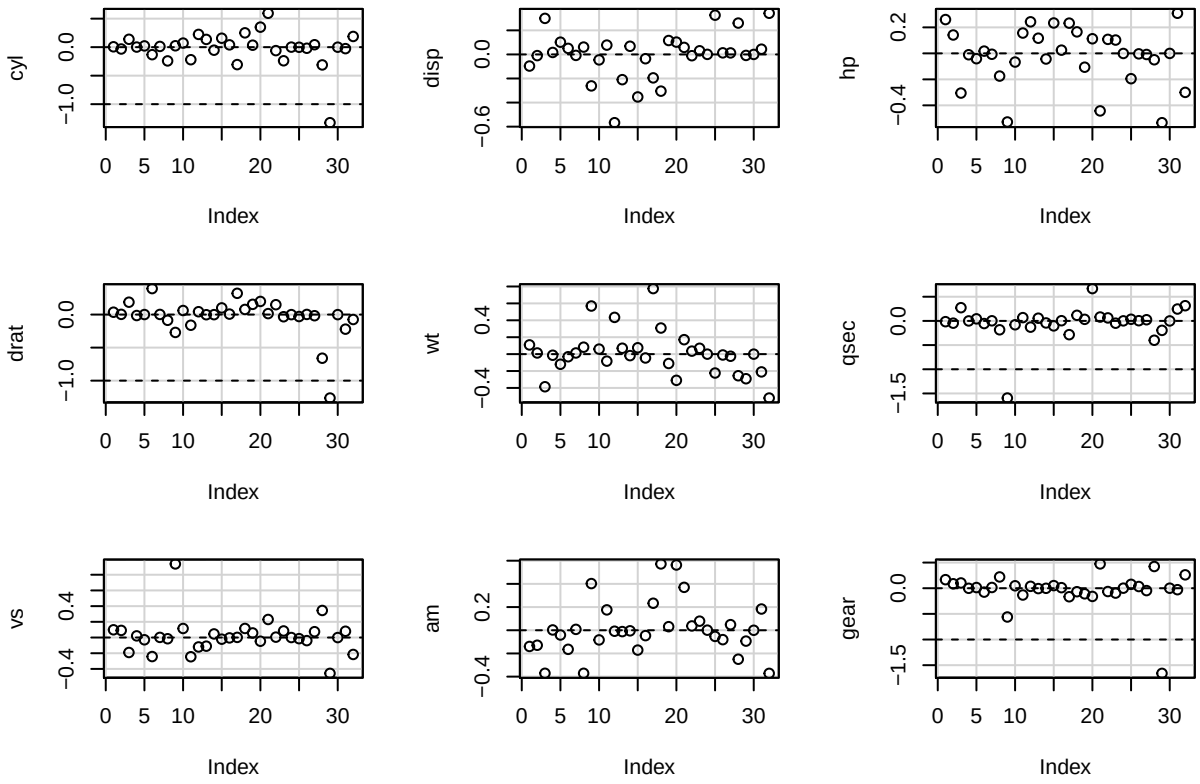

```
with(mtcars, plot(x=wt + qsec + am, y=mtcars$mpg,col=mtcars$am+1))
legend("top",c("Automatic","Manual"),col=c(1,2),pch=1)
```

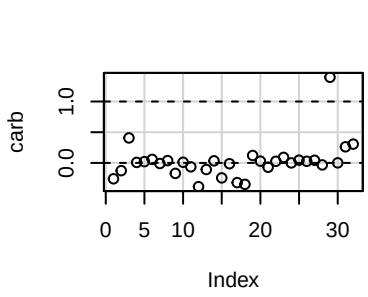



```
plot(cooks.distance(model.Complete), main="Cook's Distance for Mtcars")
```

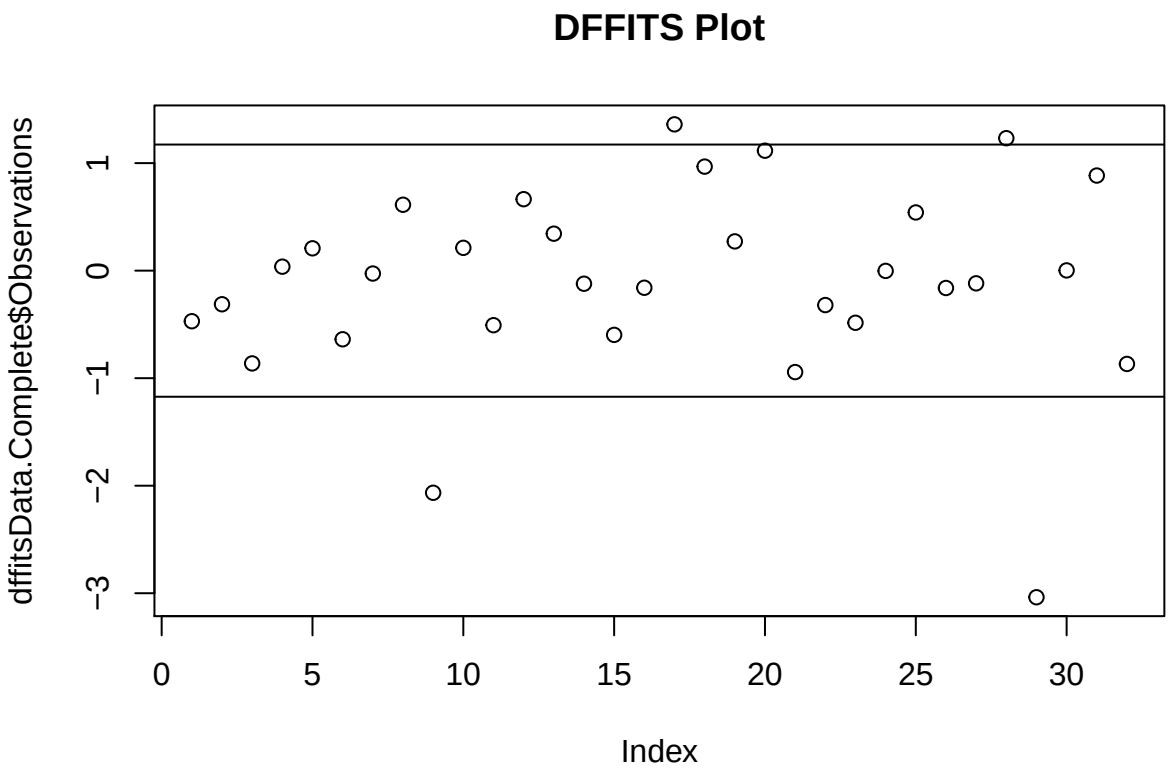



```
dfbetasPlots(model.Complete, main="Plot of Mtcars DFBETAS")
```


cyl
disp
hp
drat
wt
qsec
vs
am
gear
Index



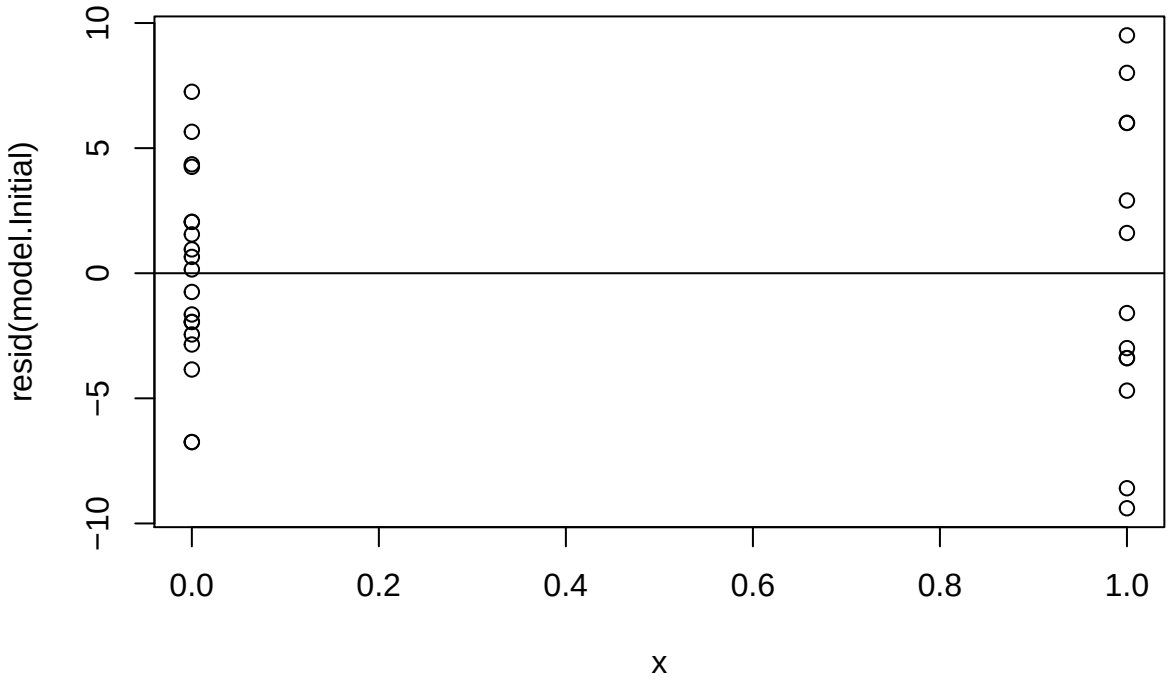

```
#Dffits
dffitsData.Complete <- as.data.frame(dffits(model.Complete))
names(dffitsData.Complete) <- c("Observations")
cutoff <- 2*sqrt(11/length(mtcars$mpg))
plot(dffitsData.Complete$Observations, main="DFFITS Plot")
abline(h=cutoff)
abline(h=-cutoff)
```



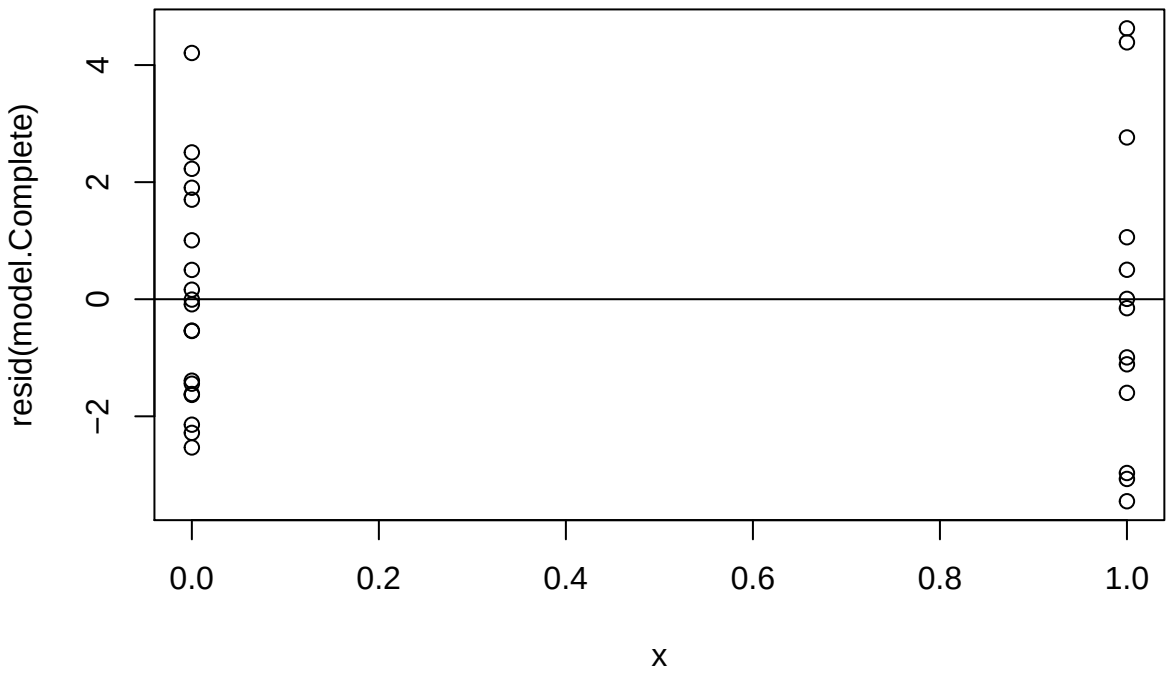

```
labels=row.names(mtcars)
```

Figure 3 Plot of residuals:

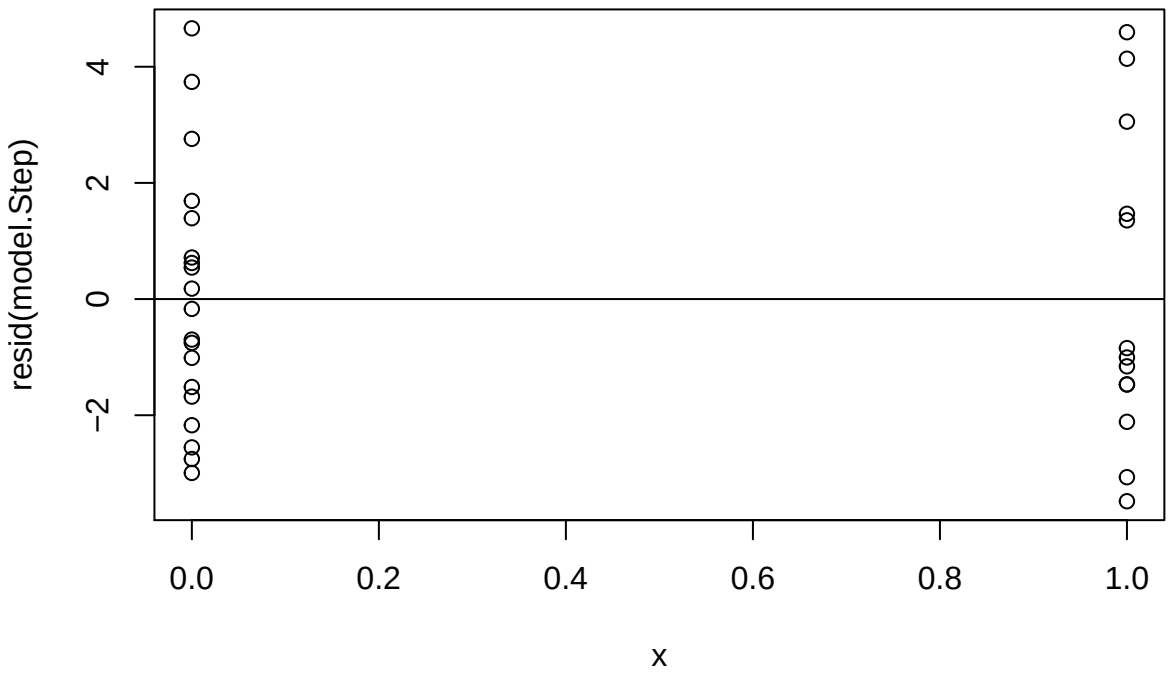
```
plot(x, resid(model.Initial))
abline(h=0)
```




```
plot(x, resid(model.Complete))
abline(h=0)
```




```
plot(x, resid(model.Step))
abline(h=0)
```


resid(model.Step)
4
2
0
−2
0.0
0.2
0.4
0.6
0.8
1.0
x