Cars Data - Regression Tree Analysis

Dr. Phil

December 9, 2019

```
# Set global figure size
knitr::opts_chunk$set(fig.width=6, fig.height=4.0)
```

Introduction

This report examines data on cars manufactured between 1970 and 1982. The primary goal is to determine what factors are most predictive of mileage, measured in miles per gallon (MPG). The predictor variables include number of cylinders, Displacement, Horsepower, Weight (lbs), Acceleration, Year produced, and country of origin (Europe, US, Japan). There are 394 observations in the data set.

```
library(readr)
cars2 <- read_csv("~/Fall 2017/Math 327 Fall 2017/Other Data/
cars.csv")

## Parsed with column specification:
## cols(
##   Model = col_character(),
##   MPG = col_double(),
##   n.Cyl = col_double(),
##   Displc = col_double(),
##   Hpower = col_double(),
##   Weight = col_double(),
##   Accel = col_double(),
##   Year = col_double(),
##   Origin = col_character()
## )

carscom = cars2 [complete.cases(cars2),]
attach (carscom)
```

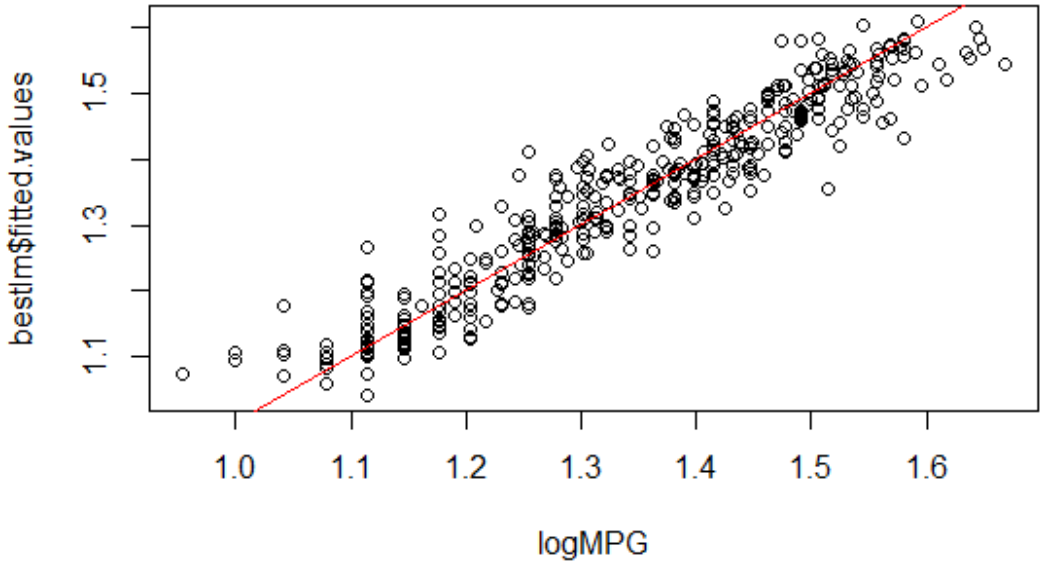
Final Linear Regression Model

This was my final linear regression model:

```r
logMPG = log10 (MPG)
Hpower.c = Hpower - mean (Hpower, na.rm=T)
Weight.c = Weight - mean (Weight)
Year.c = Year - mean (Year)
Hpower.Weight = Hpower.c * Weight.c
Hpower.Year = Hpower.c * Year.c
Weight.Year = Weight.c * Year.c
bestlm = lm (logMPG ~ Hpower.c + Weight.c + Year.c + Origin +
                      Hpower.Weight + Hpower.Year + Weight.Year)
summary (bestlm)

##
## Call:
## lm(formula = logMPG ~ Hpower.c + Weight.c + Year.c + Origin +
##     Hpower.Weight + Hpower.Year + Weight.Year)
##
## Residuals:
##       Min         1Q    Median         3Q       Max
## -0.157073 -0.029009  0.001747  0.027918  0.158632
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.343e+00  7.074e-03 189.798  < 2e-16 ***
## Hpower.c      -1.051e-03  1.637e-04  -6.417 4.11e-10 ***
## Weight.c      -9.923e-05  6.767e-06 -14.664  < 2e-16 ***
## Year.c         1.317e-02  8.017e-04  16.428  < 2e-16 ***
## OriginJapan   -1.921e-03  8.384e-03  -0.229 0.818847
## OriginUS      -2.255e-02  7.675e-03  -2.938 0.003499 **
## Hpower.Weight  4.985e-07  9.610e-08   5.188 3.46e-07 ***
## Hpower.Year   -1.222e-04  3.632e-05  -3.364 0.000847 ***
## Weight.Year    3.964e-06  1.786e-06   2.219 0.027052 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04864 on 383 degrees of freedom
## Multiple R-squared:  0.8937, Adjusted R-squared:  0.8915
## F-statistic: 402.6 on 8 and 383 DF,  p-value: < 2.2e-16
```

```
plot (logMPG, bestlm$fitted.values)
abline (0, 1, col='red')
```



Analysis Method - Recursive Partitioning

Start with modeling MPG vs all predictors.

```
#This code demonstrates recursive partition analysis
library (rpart)
tree1 = rpart (MPG ~ n.Cyl + Displc + Hpower + Weight + Accel + Year +
                 Origin, maxdepth=5)
print (tree1$cptable)
```

```
##           CP nsplit rel error    xerror       xstd
## 1 0.58033113      0 1.0000000 1.0070138 0.06194362
## 2 0.11060764      1 0.4196689 0.4486656 0.03868203
## 3 0.04877326      2 0.3090612 0.3783010 0.03243685
## 4 0.04245216      3 0.2602880 0.3264706 0.03044149
## 5 0.02976827      4 0.2178358 0.2845419 0.02925698
```

```
## 6 0.01873528       5 0.1880675 0.2410507 0.02610509
## 7 0.01866186       6 0.1693323 0.2318149 0.02586353
## 8 0.01000000       7 0.1506704 0.2177656 0.02548789
```

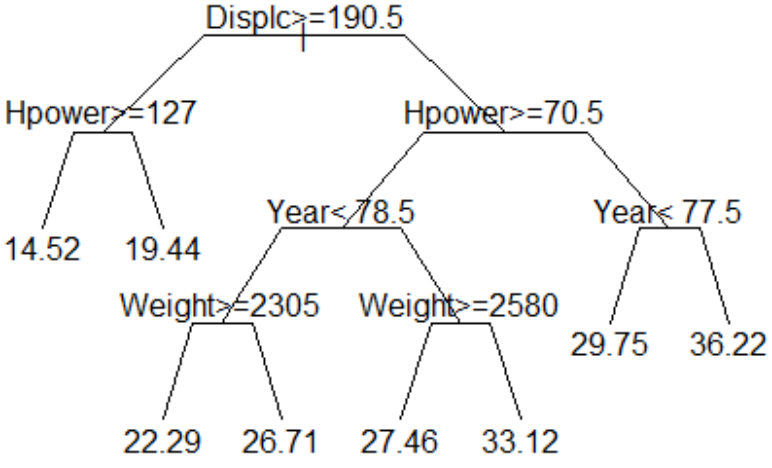
Find the tree with the smallest xerror:

```
opt1 = which.min (tree1$cptable [,"xerror"])
opt1
```

```
## 8
## 8
```

Plot the tree:

```
par (mfrow=c(1,1))
plot(tree1, uniform = TRUE, margin = 0.1, branch = 0.5,
     compress = TRUE)
text(tree1)
```



This tree says that displacement is the single most important predictor. Cars have the highest MPG when Displacement < 190.5, Horsepower < 70.5, and Year > 77.5. That group of cars has an average MPG of 36.22 miles/gallon.

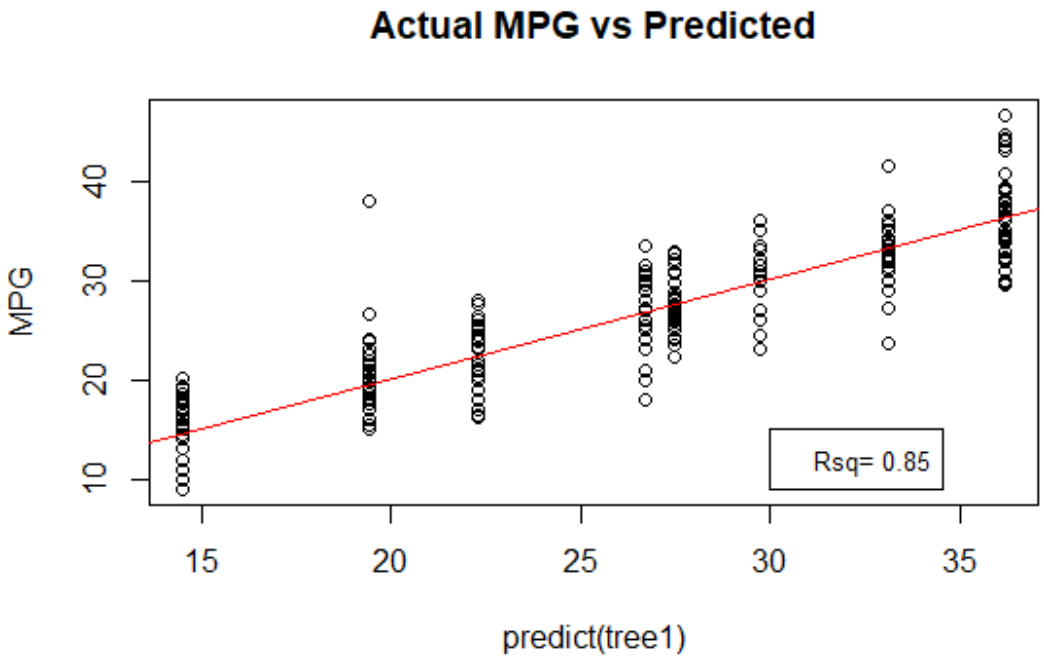
The cars with the lowest MPG are ones with Displacement >= 190.5 and Horsepower >= 127, regardless of year. Those cars have an average MPG of 14.52 miles/gallon.

Year matters when Discplacement < 190.5 and Horsepower < 70.5. Newer cars have higher MPG on average.

Car weight matters when Discplacement < 190.5, Horsepower < 70.5, and Year < 78.5. Heavier cars have lower MPG on average.

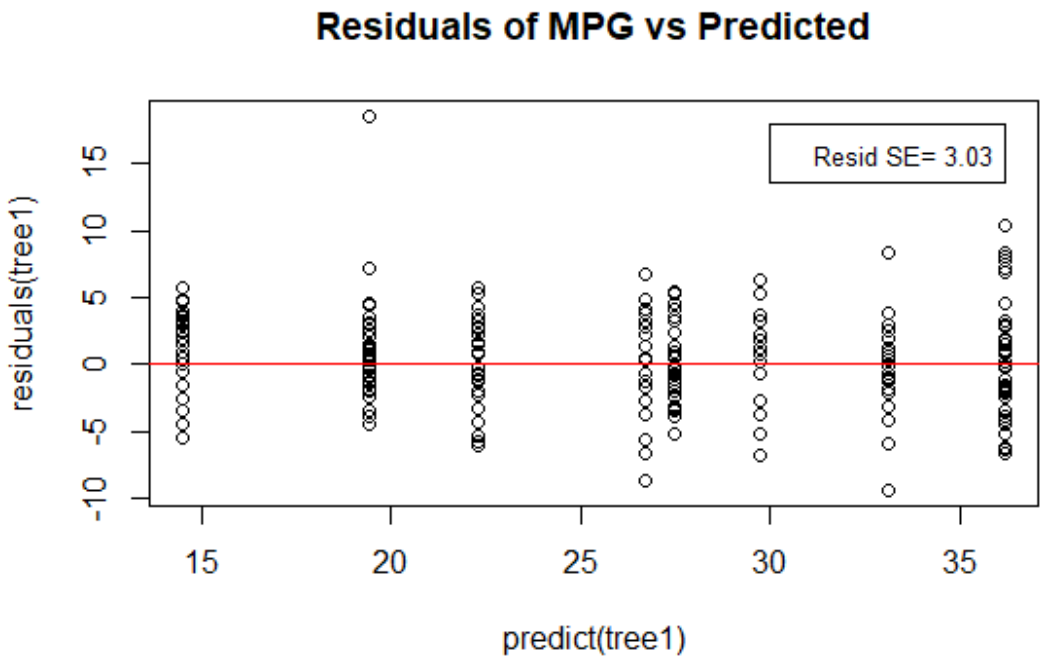
Plot logMPG vs predicted.

```
plot (predict(tree1), MPG, main="Actual MPG vs Predicted")
abline (0, 1, col='red')
rsq1 = cor (predict(tree1), MPG)^2
legend (30, 15, c(paste("Rsq=", round (rsq1, 2))), cex=0.8)
```



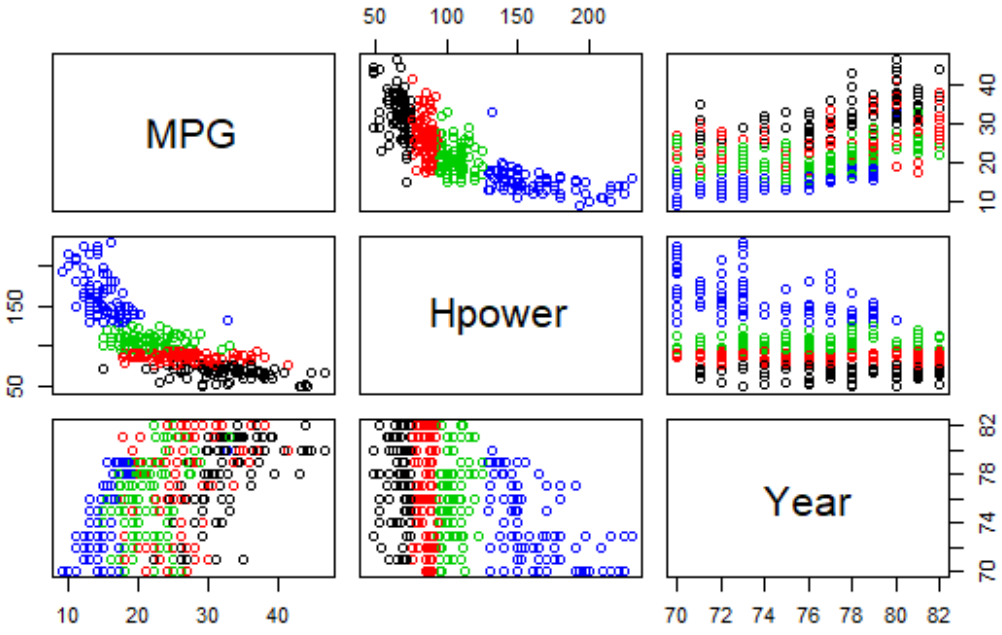
Residual plot

```
plot (predict(tree1), residuals(tree1), main="Residuals of MPG vs
Predicted")
abline (0, 0, col='red')
resid.se = sd (residuals (tree1))
legend (30, 18, c(paste ("Resid SE=", round (resid.se, 2))), cex=0.8)
```



Interaction between Horsepower and Year:

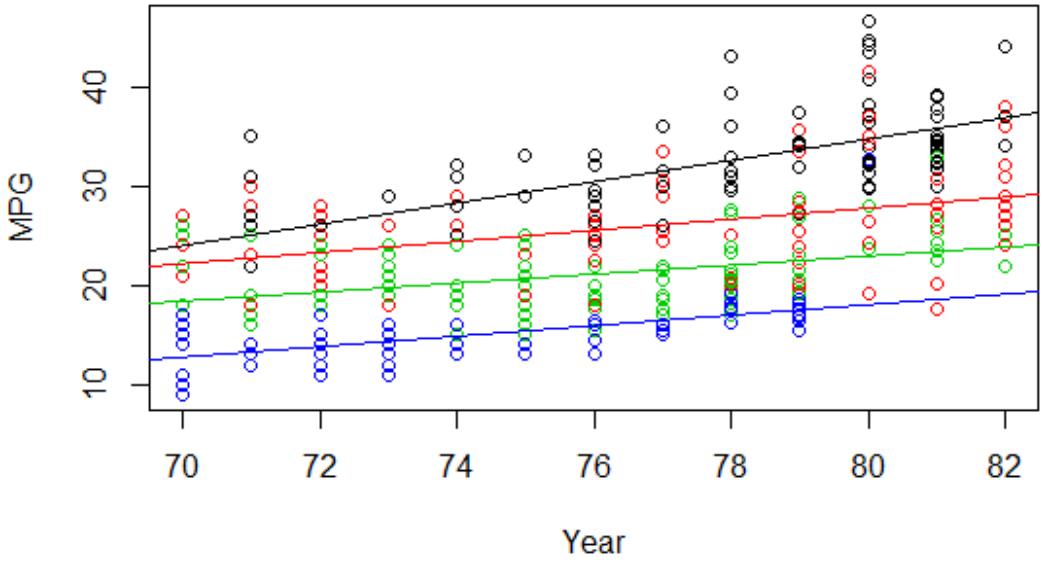
```
Hpower.levels = cut(Hpower, breaks=quantile(Hpower), 1:4)
pairs (cbind.data.frame (MPG, Hpower, Year), col=Hpower.levels)
```

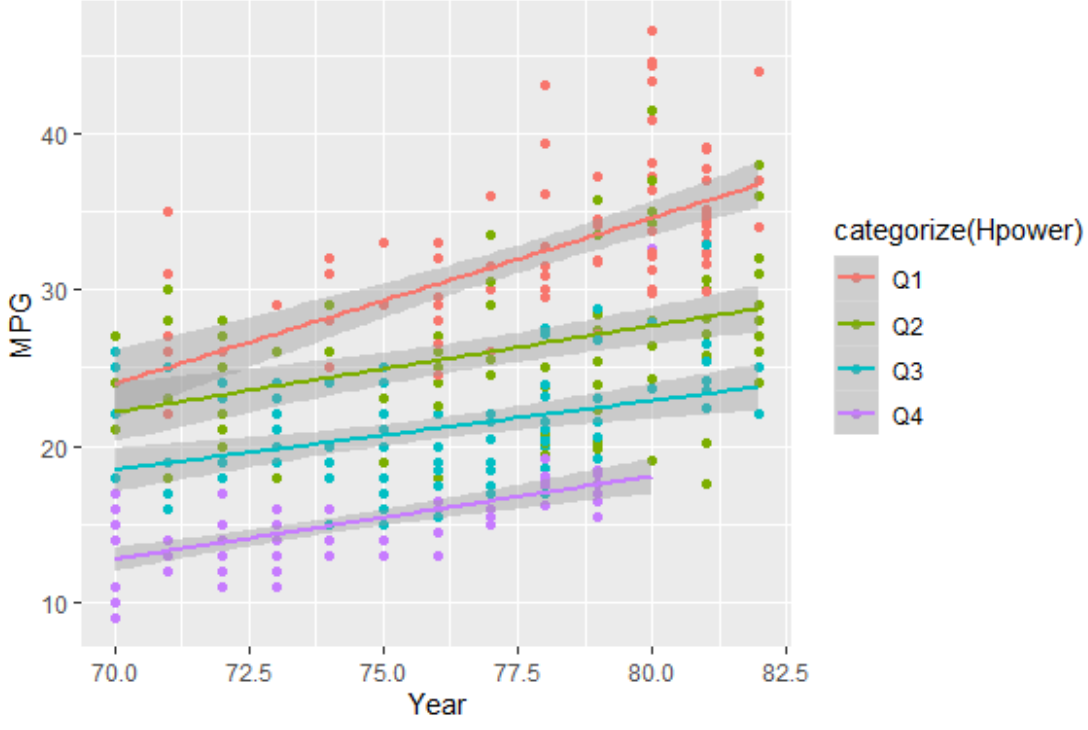



```
plot (Year, MPG, col=Hpower.levels)
abline (lm (MPG[Hpower.levels==1] ~ Year[Hpower.levels==1]), col=1)
abline (lm (MPG[Hpower.levels==2] ~ Year[Hpower.levels==2]), col=2)
abline (lm (MPG[Hpower.levels==3] ~ Year[Hpower.levels==3]), col=3)
abline (lm (MPG[Hpower.levels==4] ~ Year[Hpower.levels==4]), col=4)

categorize = function (x) {
  quartiles = summary (x) [c(2, 3, 5)]
  result = rep ("Q1", length (x))
  result [(quartiles[1] < x) & (x <= quartiles [2])] = "Q2"
  result [(quartiles[2] < x) & (x <= quartiles [3])] = "Q3"
  result [quartiles[3] < x] = "Q4"
  return (result)
}

ggplot2::qplot (x=Year, y=MPG, color=categorize (Hpower)) +
ggplot2::geom_smooth (method="lm")
```


MPG
10
20
30
40
70
72
74
76
78
80
82
Year


MPG
10
20
30
40
70.0
72.5
75.0
77.5
80.0
82.5
Year
categorize(Hpower)
Q1
Q2
Q3
Q4

The linear regression for these data used log(MPG), so do a tree regression with log(MPG).

```
tree2 = rpart (logMPG ~ n.Cyl + Displc + Hpower + Weight + Accel +
Year +
                  Origin, maxdepth=5)
print (tree2$cptable)
```

```
##           CP nsplit rel error    xerror       xstd
## 1 0.62737506      0 1.0000000 1.0053191 0.05528606
## 2 0.07748983      1 0.3726249 0.4222656 0.03265932
## 3 0.07445893      2 0.2951351 0.3820881 0.03260036
## 4 0.02946272      3 0.2206762 0.2692002 0.02554621
## 5 0.02280696      4 0.1912135 0.2398357 0.02451626
## 6 0.01718722      5 0.1684065 0.2112190 0.02265143
## 7 0.01000000      6 0.1512193 0.1859679 0.01861852
```

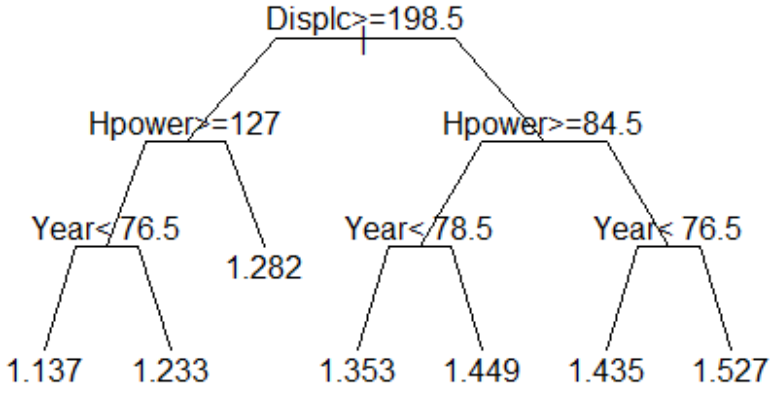
Find the tree with the smallest xerror:

```
opt2 = which.min (tree2$cptable [,"xerror"])
opt2
```

```
## 7
## 7
```

Plot the tree:

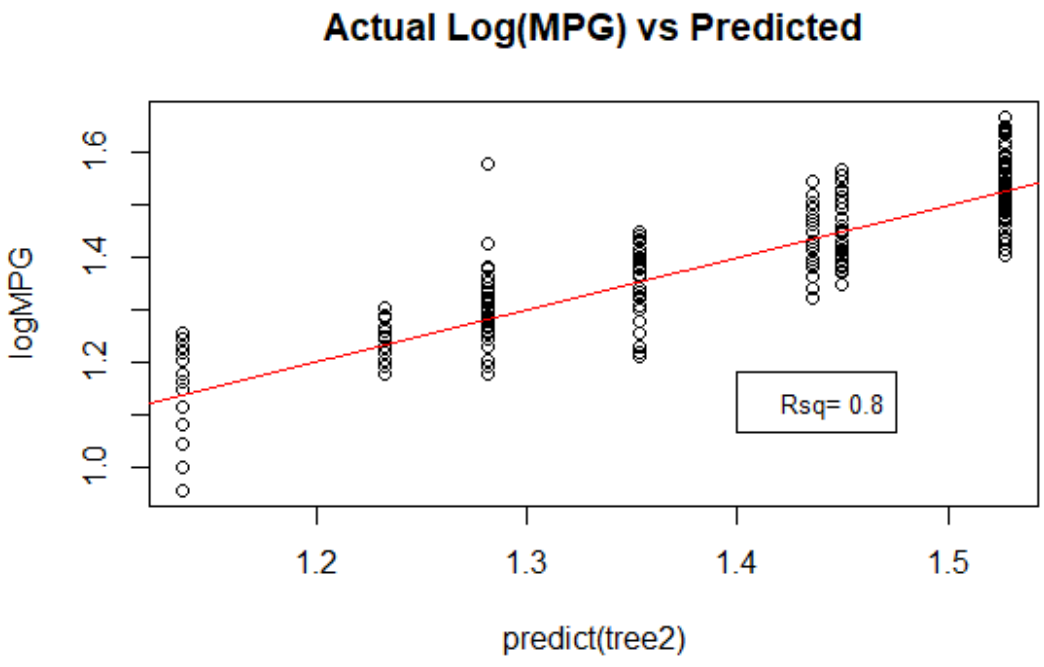
```
par (mfrow=c(1,1))
plot(tree2, uniform = TRUE, margin = 0.1, branch = 0.5,
     compress = TRUE)
text(tree2)
```



The Log(MPG) trees says that displacement is the single most important predictor. Cars with the highest MPG have Displacement < 198.5, Horsepower < 84.5, and Year > 76.5. The average MPG of those cars is `round (10^1.527, 1)`.

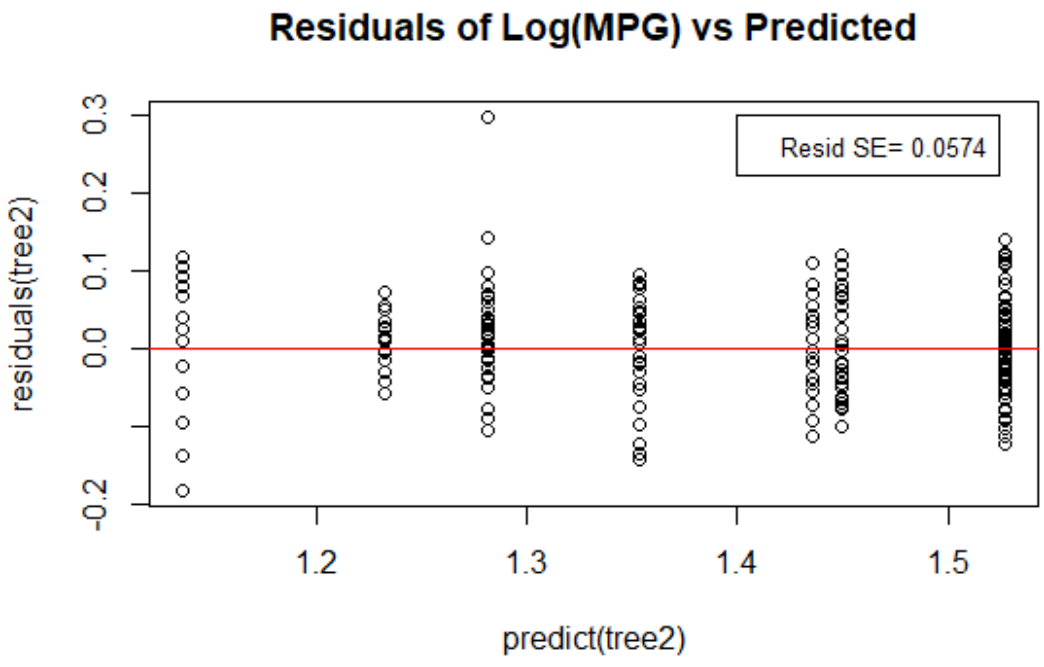
Plot logMPG vs predicted

```
plot (predict(tree2), logMPG, main="Actual Log(MPG) vs Predicted")
abline (0, 1, col='red')
rsq1 = cor (predict(tree2), MPG)^2
legend (1.4, 1.18, c(paste("Rsq=", round (rsq1, 2))), cex=0.8)
```



Residual plot

```
plot (predict(tree2), residuals(tree2), main="Residuals of Log(MPG) vs
Predicted")
abline (0, 0, col='red')
resid.se = sd (residuals (tree2))
legend (1.4, 0.3, c(paste ("Resid SE=", round (resid.se, 4))),
cex=0.8)
```



The large positive outlier is a diesel:

```
max.res.car = which.max(residuals(tree2))
carscom [max.res.car, ]
```

```
## # A tibble: 1 x 9
##   Model                   MPG n.Cyl Displc Hpower Weight Accel
Year Origin
##   <chr>                 <dbl> <dbl>  <dbl>  <dbl>  <dbl> <dbl>
<dbl> <chr>
## 1 oldsmobile cutlass c~    38     6    262     85   3015    17
82 US
```

```
predict (tree2) [max.res.car]
```

```
##     382
## 1.28168
```

```
10^predict (tree2) [max.res.car]
```

```
##      382
## 19.12845
```

Summary

The R^2 for the Log MPG tree is 0.8, which is lower than the R^2 for the final linear regression model, 0.89.

The residual SE for the tree is 0.057, which is higher than the residual SE for the final linear regression model, 0.049.