Classification and Regression Trees using `rpart`

Solon Karapanagiotis

27 July 2016

Introduction

The `rpart` package builds classification or regression tree (CART) models of a very general structure. We use the Automobile Data from 'Consumer Reports' 1990 found in the package. In contains data on 111 cars, taken from pages 235-255, 281-285 and 287-288 of the April 1990 Consumer Reports Magazine.

```
str(car90)
```

```
## 'data.frame':    111 obs. of  34 variables:
##  $ Country     : Factor w/ 10 levels "Brazil","England",..: 5 5 4 4 4 4 10 10 10 NA ...
##  $ Disp        : num  112 163 141 121 152 209 151 231 231 189 ...
##  $ Disp2       : num  1.8 2.7 2.3 2 2.5 3.5 2.5 3.8 3.8 3.1 ...
##  $ Eng.Rev     : num  2935 2505 2775 2835 2625 ...
##  $ Front.Hd    : num  3.5 2 2.5 4 2 3 4 6 5 5.5 ...
##  $ Frt.Leg.Room: num  41.5 41.5 41.5 42 42 42 42 42 41 41 ...
##  $ Frt.Shld    : num  53 55.5 56.5 52.5 52 54.5 56.5 58.5 59 58 ...
##  $ Gear.Ratio  : num  3.26 2.95 3.27 3.25 3.02 2.8 NA NA NA NA ...
##  $ Gear2       : num  3.21 3.02 3.25 3.25 2.99 2.85 2.84 1.99 1.99 2.33 ...
##  $ HP          : num  130 160 130 108 168 208 110 165 165 101 ...
##  $ HP.revs     : num  6000 5900 5500 5300 5800 5700 5200 4800 4800 4400 ...
##  $ Height      : num  47.5 50 51.5 50.5 49.5 51 49.5 50.5 51 50.5 ...
##  $ Length      : num  177 191 193 176 175 186 189 197 197 192 ...
##  $ Luggage     : num  16 14 17 10 12 12 16 16 16 15 ...
##  $ Mileage     : num  NA 20 NA 27 NA NA 21 NA 23 NA ...
##  $ Model2      : Factor w/ 21 levels "","      Turbo 4 (3)",..: 1 1 1 1 1 1 1 14 13 1 ...
##  $ Price       : num  11950 24760 26900 18900 24650 ...
##  $ Rear.Hd     : num  1.5 2 3 1 1 2.5 2.5 4.5 3.5 3.5 ...
##  $ Rear.Seating: num  26.5 28.5 31 28 25.5 27 28 30.5 28.5 27.5 ...
##  $ RearShld    : num  52 55.5 55 52 51.5 55.5 56 58.5 58.5 56.5 ...
##  $ Reliability : Ord.factor w/ 5 levels "Much worse"<"worse"<..: 5 5 NA NA 4 NA 3 3 3 NA ...
##  $ Rim         : Factor w/ 6 levels "R12","R13","R14",..: 3 4 4 3 3 4 3 3 3 3 ...
##  $ Sratio.m    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Sratio.p    : num  0.86 0.96 0.97 0.71 0.88 0.78 0.76 0.83 0.87 0.88 ...
##  $ Steering    : Factor w/ 3 levels "manual","power",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ Tank        : num  13.2 18 21.1 15.9 16.4 21.1 15.7 18 18 16.5 ...
##  $ Tires       : Factor w/ 30 levels "145","145/80",..: 16 20 20 8 17 28 13 23 23 22 ...
##  $ Trans1      : Factor w/ 4 levels "","man.4","man.5",..: 3 3 3 3 3 3 1 1 1 1 ...
##  $ Trans2      : Factor w/ 4 levels "","auto.3","auto.4",..: 3 3 2 2 3 3 2 3 3 3 ...
##  $ Turning     : num  37 42 39 35 35 39 41 43 42 41 ...
##  $ Type        : Factor w/ 6 levels "Compact","Large",..: 4 3 3 1 1 3 3 2 2 NA ...
##  $ Weight      : num  2700 3265 2935 2670 2895 ...
##  $ Wheel.base  : num  102 109 106 100 101 109 105 111 111 108 ...
##  $ Width       : num  67 69 71 67 65 69 69 72 72 71 ...
```

Use the following command for more info on the variables

```
?car90 #for more info
```

I have excluded 2 variables: `Tires` because it is factor with a very large number of levels whose printout does not fit well in the page size and `Disp2` because it is a transformation of the response.

```
car90new <- car90[,-3]
cars90new <- car90new[ ,-which(names(car90new) == "Tires")]
```

For illustration, I sample 30 data points as test set and use the rest as training set. However, when the number of samples is not large, a strong case can be made that a test set should be avoided because every sample may be needed for model building. Additionally, the size of the test set may not have sufficient power or precision to make reasonable judgements. Several researchers (Molinaro, Simon, and Pfeiffer 2005; Martin and Hirschberg 1996; Hawkins, Basak, and Mills 2003) show that validation using a single test set can be a poor choice. I ignore these issues here.

```
set.seed(186)
s <- sample(dim(cars90new)[1], 30)
test <- cars90new[s, ]
train <- cars90new[-s, ]
```

The goal is to predict the engine displacement (in cubic inches) on the basis of the 31 variables.

```
which(is.na(train$Disp)) # 2 missing values of the response
```

```
## [1] 10 14
```

Those 2 observations are not used in the analysis (more details below).

CART

The algorithm uses a two-stage procedure:
1. first the single variable is found which best splits the data into two groups ("best" is defined below). The data is separated, and then this process is applied separately to each sub-group, and so on recursively until the subgroups either reach a minimum size (5 for this data) or until no improvement can be made.
2. The second stage of the procedure consists of using cross-validation to trim back the full tree.

The "best" variable is chosen by sum of squares $SS_T - (SS_{right} + SS_{left})$, where $SS_T = \sum(y_i - \bar{y})$ is the sum of squares for the node, and SS_{right}, SS_{left} are the sums of squares for the right and left son, respectively. This is equivalent to choosing the split to maximize the between-groups sum-of-squares in a simple analysis of variance.

```
set.seed(1235)
controlrpart <- rpart.control(minsplit = 15, cp=0.01)
rpartTree <- rpart(Disp ~ ., data = train, control=controlrpart, method="anova") #the anova method lead
```

- setting the seed will make sure the results reproducible.
- `minsplit`: The minimum number of observations in a node for which the routine will even try to compute a split. Chosen to be 15 so the minimum number of observations in a terminal node would be 15/3=5.

- `cp`: complexity parameter (default=0.01). Any split that does not decrease the overall lack of fit by a factor of cp is not attempted. For instance, this means that the overall R-squared must increase by 0.01 at each step.
- for more details into the functions `rpart.control()` and `rpart()` visit CRAN.

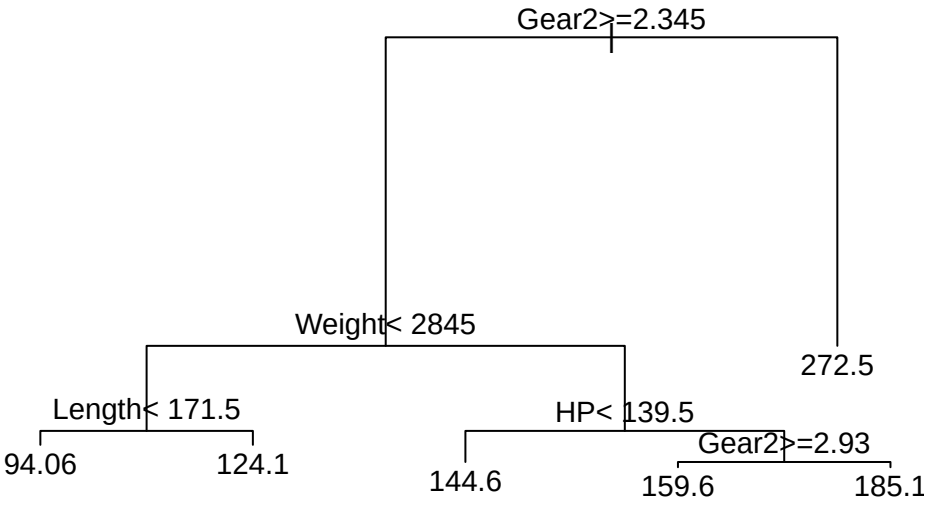
```
print(rpartTree)
```

```
## n=79 (2 observations deleted due to missingness)
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 79 262567.900 152.72150
##    2) Gear2>=2.345 69  76667.940 135.36230
##      4) Weight< 2845 33  14251.880 108.60610
##        8) Length< 171.5 17   3224.941  94.05882 *
##        9) Length>=171.5 16   3606.938 124.06250 *
##      5) Weight>=2845 36  17135.560 159.88890
##       10) HP< 139.5 18   3768.278 144.61110 *
##       11) HP>=139.5 18   4964.500 175.16670
##         22) Gear2>=2.93 7   1131.714 159.57140 *
##         23) Gear2< 2.93 11   1046.909 185.09090 *
##    3) Gear2< 2.345 10  21638.500 272.50000 *
```

- the tree was built on n=79 observations. 2 observations deleted due to missingness.
- The child nodes of node x are always $2x$ and $2x+1$. For example, the child nodes of node 2 are 4 and 5.
- Other items in the list are the definition of the split used to create a node, n=the number of subjects at the node, the loss or error at the node (in our case the deviance-least squares), and the predicted mean value for the node.

Plotting the tree

```
plot(rpartTree, compress = TRUE, margin = 0.05)
text(rpartTree, cex = 0.9)
```



We see that, for example, the highest engine displacement (i.e., 272.5 cubic inches) is predicted for a car with the overall gear ratio (`Gear2`), for automatic transmission, higher or equal to 2.345.

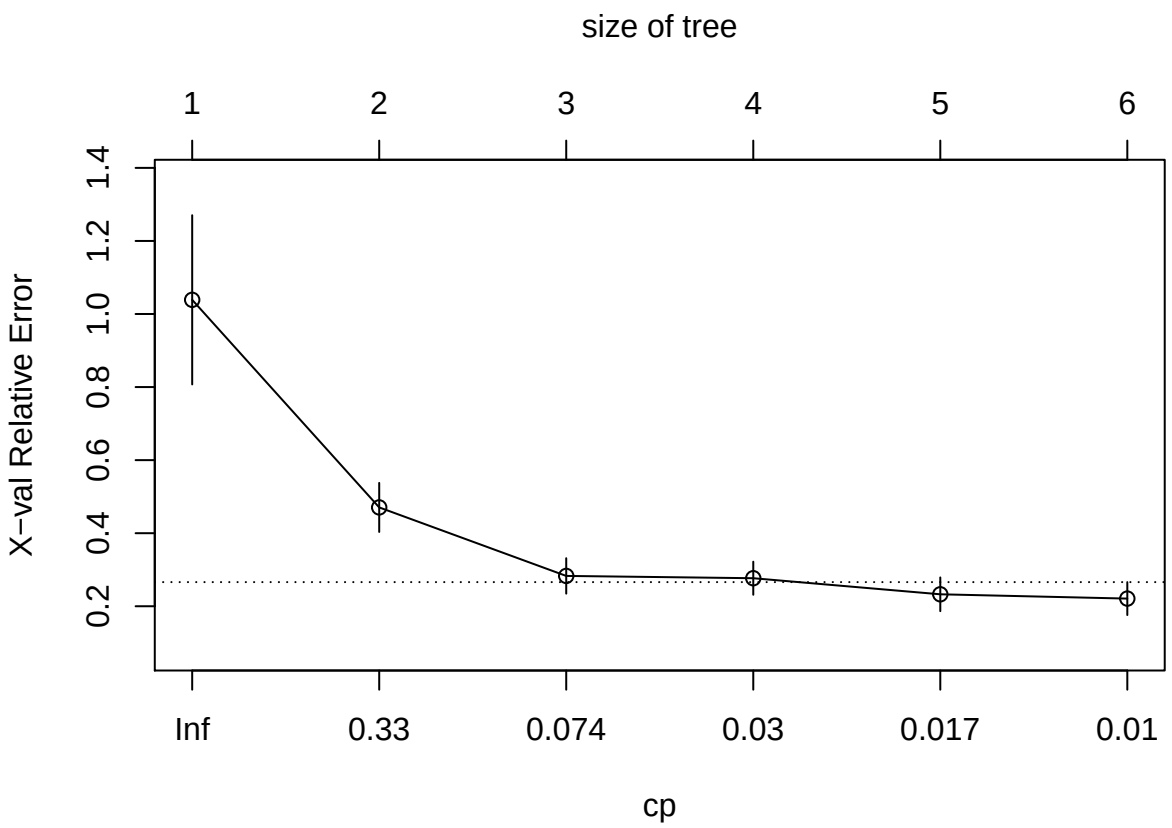
We have built a complete tree, possibly quite large and/or complex, and must now decide how much of that model to retain.

```
printcp(rpartTree)
```

```
##
## Regression tree:
## rpart(formula = Disp ~ ., data = train, method = "anova", control = controlrpart)
##
## Variables actually used in tree construction:
## [1] Gear2  HP     Length Weight
##
## Root node error: 262568/79 = 3323.6
##
## n=79 (2 observations deleted due to missingness)
##
##         CP nsplit rel error  xerror     xstd
## 1 0.625596      0   1.00000 1.03878 0.231623
## 2 0.172453      1   0.37440 0.47052 0.067199
## 3 0.032002      2   0.20195 0.28299 0.048540
## 4 0.028259      3   0.16995 0.27670 0.045279
## 5 0.010610      4   0.14169 0.23275 0.046220
## 6 0.010000      5   0.13108 0.22098 0.045112
```

- The complexity table is printed from the smallest tree (no splits) to the largest one (5 splits).
- The number of splits is listed, rather than the number of nodes. The number of terminal nodes is always 1 + the number of splits.
- The relative error is $1 - R^2$, similar to linear regression. The xerror is related to the PRESS statistic. The first split appears to improve the fit the most. The last split adds little improvement to the apparent error, and increases the cross-validated error.
- The 1-SE method for choosing simpler models finds the numerically optimal value and its corresponding standard error and then seeks the simplest model whose performance is within a single standard error of the numerically best value. The 1-SE rule would choose a tree with 4 splits (The minimal xerror is 0.22098, the xstd is 0.045112 so the tree with xerror smaller than 0.22098+0.045112 is the one with xerror 0.23275 which is a tree with 4 splits and final size (here the number of terminal nodes) equal to 5.

```
plotcp(rpartTree, minline = TRUE) # horizontal line is drawn 1SE above the minimum of the curve
```



Looking at the plot, we see that the best tree has 5 terminal nodes (4 splits), based on cross-validation (any number of splits within the "error bars"). This sub tree is extracted and saved in `rpartTree2`.

- We used the default cp value of 0.01 may have over pruned the tree, since the cross-validated error is barely at a minimum. A rerun with the cp threshold at .001 gave the same results! Run the code to verify.

```
set.seed(1235) #same as before
controlrpart2 <- rpart.control(minsplit = 15, cp=0.001)
rpartTree3 <- rpart(Disp ~ ., data =train, control=controlrpart2, method="anova")
plotcp(rpartTree3, minline = TRUE)
printcp(rpartTree3)
```

Returning back to `rpartTree2`. The `summary()` commands recognizes the cp option, which allows us to look at only the top few splits

```
summary(rpartTree2, cp = 0.0106)
```

```
## Call:
## rpart(formula = Disp ~ ., data = train, method = "anova", control = controlrpart)
##   n=79 (2 observations deleted due to missingness)
##
##           CP nsplit rel error    xerror       xstd
## 1 0.62559608      0 1.0000000 1.0387782 0.23162321
## 2 0.17245258      1 0.3744039 0.4705209 0.06719859
## 3 0.03200231      2 0.2019513 0.2829947 0.04853991
```

```
## 4 0.02825936      3 0.1699490 0.2767035 0.04527898
## 5 0.01061012      4 0.1416897 0.2327464 0.04621964
## 6 0.01000000      5 0.1310796 0.2209803 0.04511225
##
## Variable importance
##      Gear2     Weight     Length      Width    HP.revs    Luggage
##         19         16         15         15         10          8
##       Type       Tank   Frt.Shld         HP      Price    Rear.Hd
##          4          4          4          1          1          1
##   Steering Wheel.base   Front.Hd        Rim
##          1          1          1          1
##
## Node number 1: 79 observations,    complexity param=0.6255961
##   mean=152.7215, MSE=3323.644
##   left son=2 (69 obs) right son=3 (10 obs)
##   Primary splits:
##       Gear2   < 2.345   to the right, improve=0.6095606, (3 missing)
##       Weight  < 3197.5  to the left,  improve=0.4868311, (0 missing)
##       Length  < 198.5   to the left,  improve=0.4613309, (0 missing)
##       Width   < 69.5    to the left,  improve=0.4595483, (0 missing)
##       HP.revs < 4450    to the right, improve=0.4506438, (0 missing)
##   Surrogate splits:
##       Length  < 196     to the left,  agree=0.947, adj=0.6, (3 split)
##       Width   < 72.5    to the left,  agree=0.947, adj=0.6, (0 split)
##       HP.revs < 4450    to the right, agree=0.934, adj=0.5, (0 split)
##       Weight  < 3692.5  to the left,  agree=0.934, adj=0.5, (0 split)
##       Luggage < 18.5    to the left,  agree=0.921, adj=0.4, (0 split)
##
## Node number 2: 69 observations,    complexity param=0.1724526
##   mean=135.3623, MSE=1111.13
##   left son=4 (33 obs) right son=5 (36 obs)
##   Primary splits:
##       Weight   < 2845    to the left,  improve=0.5906055, (0 missing)
##       Steering splits as  LRL,         improve=0.4600793, (0 missing)
##       Tank     < 13.4    to the left,  improve=0.4374087, (0 missing)
##       Rim      splits as  LLRRR-,      improve=0.4330027, (0 missing)
##       HP       < 140.5   to the left,  improve=0.4145414, (0 missing)
##   Surrogate splits:
##       Frt.Shld < 54.25   to the left,  agree=0.841, adj=0.667, (0 split)
##       Width    < 67.5    to the left,  agree=0.841, adj=0.667, (0 split)
##       Type     splits as  L-RLLR,      agree=0.826, adj=0.636, (0 split)
##       Tank     < 15.95   to the left,  agree=0.812, adj=0.606, (0 split)
##       Length   < 179.5   to the left,  agree=0.797, adj=0.576, (0 split)
##
## Node number 3: 10 observations
##   mean=272.5, MSE=2163.85
##
## Node number 4: 33 observations,    complexity param=0.02825936
##   mean=108.6061, MSE=431.8751
##   left son=8 (17 obs) right son=9 (16 obs)
##   Primary splits:
##       Length     < 171.5   to the left,  improve=0.5206331, (0 missing)
##       Wheel.base < 96.5    to the left,  improve=0.5034260, (0 missing)
##       Weight     < 2437.5  to the left,  improve=0.4454525, (0 missing)
```

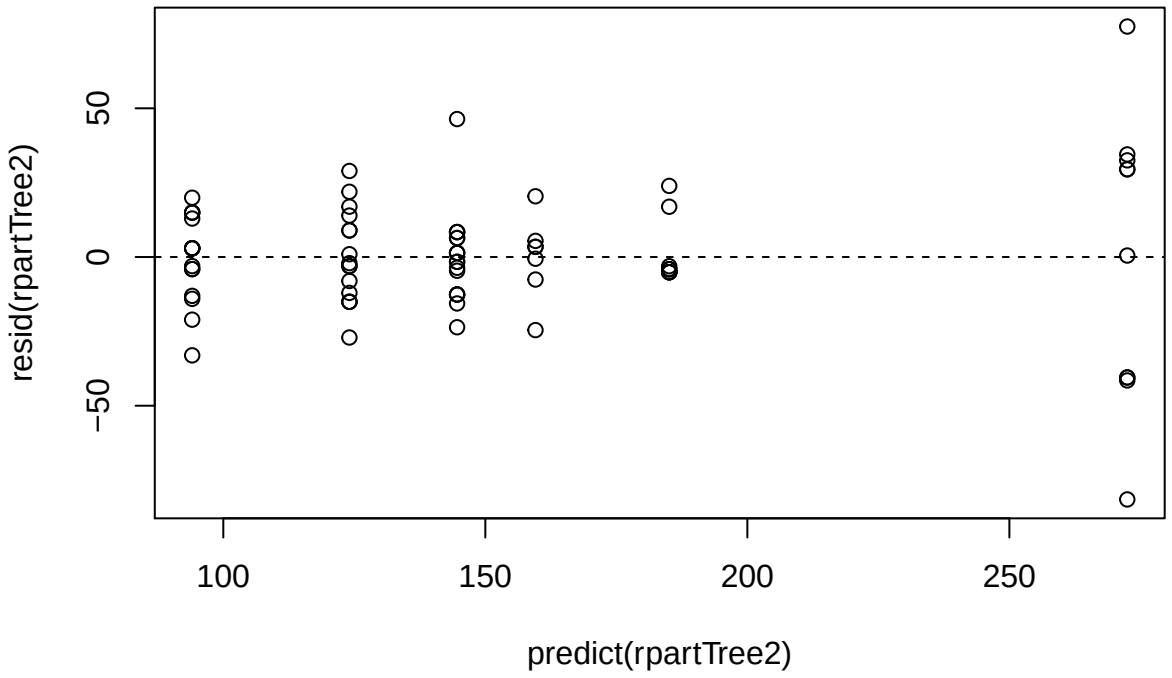
```
##       Tank       < 13.4    to the left,  improve=0.4383735, (0 missing)
##       Type       splits as  R--LL-,      improve=0.4311354, (0 missing)
##   Surrogate splits:
##       Steering   splits as  LRL,         agree=0.848, adj=0.688, (0 split)
##       Weight     < 2437.5  to the left,  agree=0.848, adj=0.688, (0 split)
##       Wheel.base < 96.5    to the left,  agree=0.848, adj=0.688, (0 split)
##       Tank       < 13.4    to the left,  agree=0.818, adj=0.625, (0 split)
##       Type       splits as  R--LL-,      agree=0.788, adj=0.562, (0 split)
##
## Node number 5: 36 observations,    complexity param=0.03200231
##   mean=159.8889, MSE=475.9877
##   left son=10 (18 obs) right son=11 (18 obs)
##   Primary splits:
##       HP      < 139.5   to the left,  improve=0.4903709, (0 missing)
##       Tank    < 18.25   to the left,  improve=0.3772651, (0 missing)
##       Weight  < 3465    to the left,  improve=0.3417571, (0 missing)
##       Gear2   < 2.88    to the right, improve=0.2223430, (0 missing)
##       Country splits as  -RLRRLL-LR,  improve=0.2211120, (1 missing)
##   Surrogate splits:
##       Price    < 15139.5 to the left,  agree=0.833, adj=0.667, (0 split)
##       Front.Hd < 3.25    to the right, agree=0.750, adj=0.500, (0 split)
##       Rim      splits as  --LRR-,      agree=0.750, adj=0.500, (0 split)
##       HP.revs  < 5450    to the left,  agree=0.722, adj=0.444, (0 split)
##       Rear.Hd  < 1.75    to the right, agree=0.722, adj=0.444, (0 split)
##
## Node number 8: 17 observations
##   mean=94.05882, MSE=189.7024
##
## Node number 9: 16 observations
##   mean=124.0625, MSE=225.4336
##
## Node number 10: 18 observations
##   mean=144.6111, MSE=209.3488
##
## Node number 11: 18 observations,    complexity param=0.01061012
##   mean=175.1667, MSE=275.8056
##   left son=22 (7 obs) right son=23 (11 obs)
##   Primary splits:
##       Gear2    < 2.93    to the right, improve=0.5611596, (0 missing)
##       Weight   < 3065    to the left,  improve=0.3920280, (0 missing)
##       Height   < 50.75   to the left,  improve=0.3715850, (0 missing)
##       RearShld < 54.25   to the left,  improve=0.3378034, (0 missing)
##       Rear.Hd  < 1.25    to the left,  improve=0.3222882, (0 missing)
##   Surrogate splits:
##       HP.revs      < 5750    to the right, agree=0.889, adj=0.714, (0 split)
##       Rear.Hd      < 1.25    to the left,  agree=0.833, adj=0.571, (0 split)
##       Rear.Seating < 26.5    to the left,  agree=0.833, adj=0.571, (0 split)
##       RearShld     < 54.25   to the left,  agree=0.833, adj=0.571, (0 split)
##       Height       < 50.25   to the left,  agree=0.778, adj=0.429, (0 split)
##
## Node number 22: 7 observations
##   mean=159.5714, MSE=161.6735
##
## Node number 23: 11 observations
```

```
##   mean=185.0909, MSE=95.17355
```

- The first split on `Gear2` partitions the 79 observations into groups of 69 and 10 (nodes 2 and 3) with mean of 135.36 and 272.5, respectively.
- The improvement listed is the percent change in SS for this split, i.e., $1 - (SS_{right} + SS_{left})/SS_{parent}$, which is the gain in R^2 for the fit.
- For explanations on the variable importance and surrogate splits we refer to An Introduction to Recursive Partitioning Using the RPART Routines by the authors of the package.

Finally, we look at the residuals from this model. There appears to be more variability in node 6 than in some of the other leaves.

```
plot(predict(rpartTree2), resid(rpartTree2))
abline(h = 0, lty = 2)
```



Missing data

In `rpart` the 2 observations with the missing values have been deleted. Given their small number compared to the 111 obserbvations the result is not expected to be altered substantially. Also, any observation with values for the dependent variable and at least one independent variable will participate in the modeling.

An Estimator of Prediction Error

A popular error measure of predictive performance is the root mean squared error (RMSE). It measures the average magnitude of the error, hence the lower its value the better. To obtain the predictions of the models we use the function `predict()`. It receives a model and a test dataset and retrieves the correspondent model predictions:

```
tree.predictions <- predict(rpartTree2, newdata=test)
```

The RMSE then can be obtained as follows:

```
(mse.tree <- sqrt(mean((tree.predictions - test$Disp)^2, na.rm = T)))
```

```
## [1] 31.23364
```

For further reading see "The Elements of Statistical Learning" by Friedman, Hastie, and Tibshirani (2001). It is available here. For less mathematically/statistically inclined audience "An Introduction to Statistical Learning" by James et al. (2013) is recommended.

References

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The Elements of Statistical Learning*. Vol. 1. Springer series in statistics Springer, Berlin.

Hawkins, Douglas M, Subhash C Basak, and Denise Mills. 2003. "Assessing Model Fit by Cross-Validation." *Journal of Chemical Information and Computer Sciences* 43 (2). ACS Publications: 579–86.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. Vol. 6. Springer.

Martin, J Kent, and Daniel S Hirschberg. 1996. "Small Sample Statistics for Classification Error Rates II: Confidence Intervals and Significance Tests." Information; Computer Science, University of California, Irvine.

Molinaro, Annette M, Richard Simon, and Ruth M Pfeiffer. 2005. "Prediction Error Estimation: A Comparison of Resampling Methods." *Bioinformatics* 21 (15). Oxford Univ Press: 3301–7.