Non-parametric Statistics: Notes 9
Classification and Regression Trees

Gregory Matthews [1]

[1]Department of Mathematics and Statistics
Loyola University Chicago

Fall 2014

Outline

Some examples

- Basic idea: Want to predict a response OR class Y
- Use $X_1, X_2, \ldots, X_p$ to predict Y.
- We do this by growing a BINARY tree.
- At each node we apply a test to one of the predictors.
- Eventually, we stop doing these tests and we make a prediction.
- And the prediction is based on all data points that end up in this leaf.

- Fitting a global model to the data may be difficult.
- With recursive partitioning, we split the space into small regions where many different simple models can be fit.

Some vocabulary

- Root node: This is where the tree starts.
- Nodes: The questions or tests.
- Branches or Edges: The result of one of the binary tests.
- Terminal node or Leaves: The end of a branch where a prediction is made.

- An observation x falls into a leaf based on the values of the predictor variables.
- Once observations are classified into leaves we fit a very simple model in each leaf.
- For regression trees, the model is just the constant estimate of Y. (i.e. The prediction for the leaf is $\bar{Y}$ of all the observations in that leaf.)

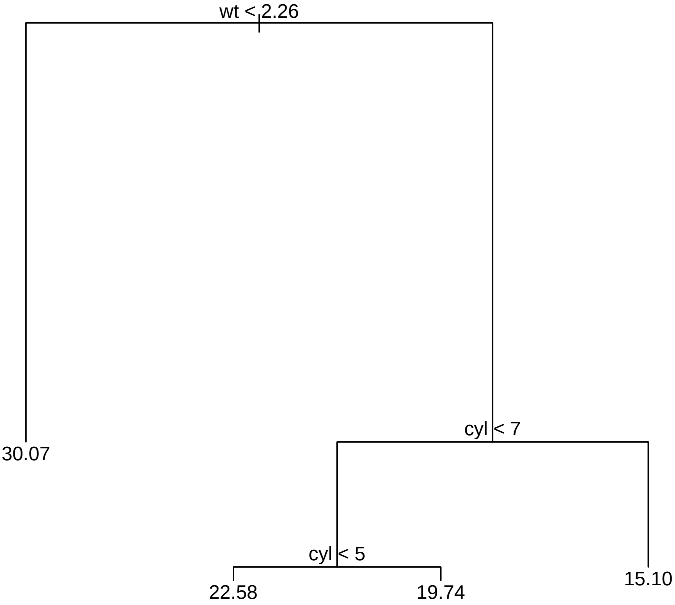
Advantage

- Computations are fast.
- Easily interpretable and it's clear what variables are important.
- Can deal with missing data.
- Can accomodate smooth and jagged responses.
- We have fast, reliable algorithms to learn these trees.

```r
library(tree)
(tree1<-tree(mpg~wt+cyl,data=mtcars))
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 32 1126.000 20.09
##    2) wt < 2.26 6   44.550 30.07 *
##    3) wt > 2.26 26  346.600 17.79
##      6) cyl < 7 12   42.120 20.92
##       12) cyl < 5 5    5.968 22.58 *
##       13) cyl > 5 7   12.680 19.74 *
##      7) cyl > 7 14   85.200 15.10 *
```

```r
mean(mtcars$mpg[mtcars$wt<2.26])

## [1] 30.06667

mean(mtcars$mpg[mtcars$wt>2.26 & mtcars$cyl>7])

## [1] 15.1

mean(mtcars$mpg[mtcars$wt>2.26 & mtcars$cyl<7 & mtcars$cyl<

## [1] 22.58

mean(mtcars$mpg[mtcars$wt>2.26 & mtcars$cyl<7 & mtcars$cyl>

## [1] 19.74286
```

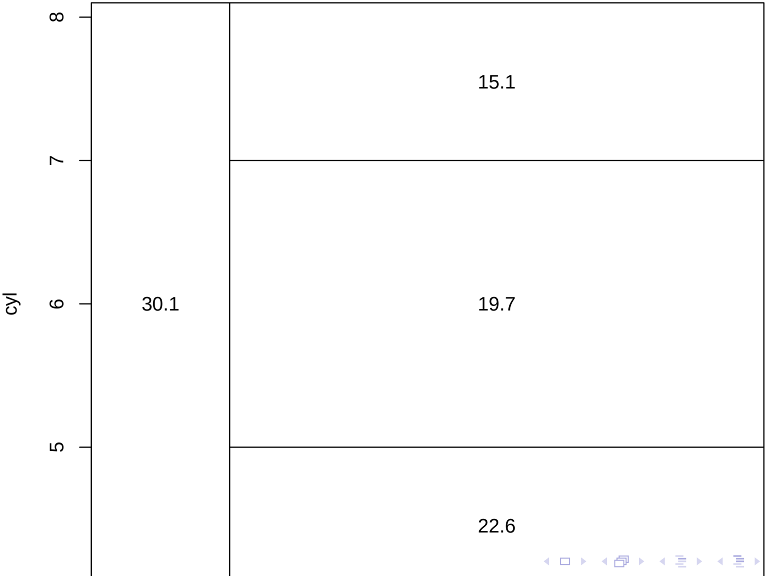


wt < 2.26
cyl < 7
cyl < 5
30.07
22.58
19.74
15.10

```r
predict(tree1)
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710
##            19.74286            19.74286            22.58000
##      Hornet 4 Drive   Hornet Sportabout             Valiant
##            19.74286            15.10000            19.74286
##          Duster 360           Merc 240D            Merc 230
##            15.10000            22.58000            22.58000
##            Merc 280           Merc 280C          Merc 450SE
##            19.74286            19.74286            15.10000
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
##            15.10000            15.10000            15.10000
## Lincoln Continental   Chrysler Imperial            Fiat 128
##            15.10000            15.10000            30.06667
##         Honda Civic      Toyota Corolla       Toyota Corona
##            30.06667            30.06667            22.58000
##    Dodge Challenger         AMC Javelin          Camaro Z28
##            15.10000            15.10000            15.10000
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2
##            15.10000            30.06667            30.06667
##        Lotus Europa      Ford Pantera L        Ferrari Dino
##            30.06667            15.10000            19.74286
##       Maserati Bora          Volvo 142E
##            15.10000            22.58000
```

```r
partition.tree(tree1)
```

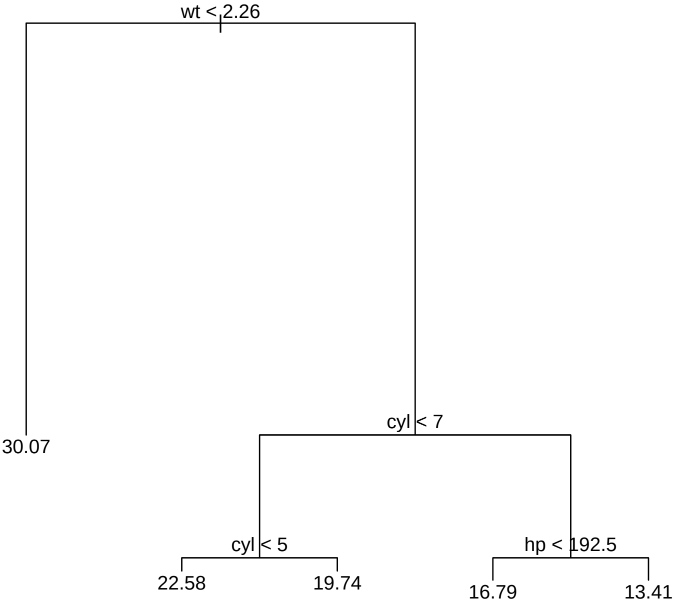


```r
library(tree)
(tree1<-tree(mpg~cyl+disp+hp+drat+wt+qsec+vs+am+gear+carb,data=mtcars))
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 32 1126.000 20.09
##    2) wt < 2.26 6   44.550 30.07 *
##    3) wt > 2.26 26  346.600 17.79
##      6) cyl < 7 12   42.120 20.92
##       12) cyl < 5 5    5.968 22.58 *
##       13) cyl > 5 7   12.680 19.74 *
##      7) cyl > 7 14   85.200 15.10
##       14) hp < 192.5 7   16.590 16.79 *
##       15) hp > 192.5 7   28.830 13.41 *
```

```r
mean(mtcars$mpg[mtcars$wt<2.26])

## [1] 30.06667

mean(mtcars$mpg[mtcars$wt>2.26 & mtcars$cyl>7 & mtcars$hp>192.5])

## [1] 13.41429
```

```
##
## Regression tree:
## tree(formula = mpg ~ cyl + disp + hp + drat + wt + qsec
##     am + gear + carb, data = mtcars)
## Variables actually used in tree construction:
## [1] "wt"  "cyl" "hp"
## Number of terminal nodes:  5
## Residual mean deviance:  4.023 = 108.6 / 27
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -4.067  -1.361   0.220   0.000   1.361   3.833
```


wt < 2.26
cyl < 7
30.07
cyl < 5
hp < 192.5
22.58
19.74
16.79
13.41

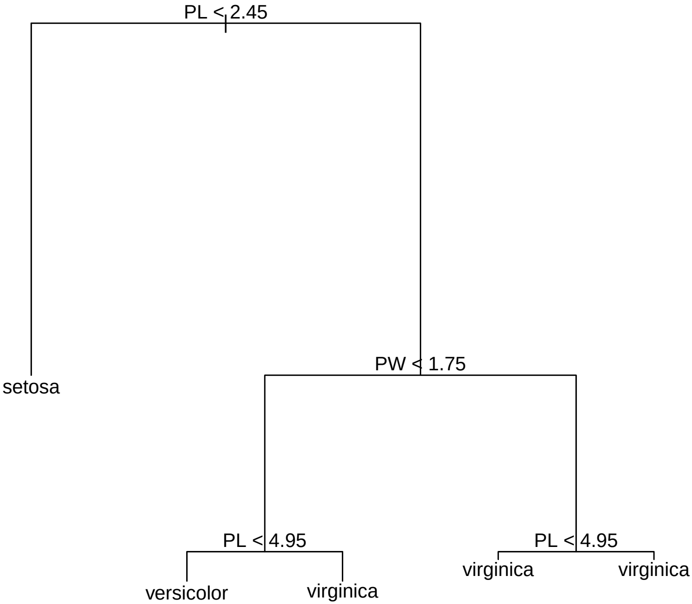
```r
predict(tree1)
```

```
##           Mazda RX4       Mazda RX4 Wag          Datsun 710
##            19.74286            19.74286            22.58000
##      Hornet 4 Drive   Hornet Sportabout             Valiant
##            19.74286            16.78571            19.74286
##          Duster 360           Merc 240D            Merc 230
##            13.41429            22.58000            22.58000
##            Merc 280           Merc 280C          Merc 450SE
##            19.74286            19.74286            16.78571
##          Merc 450SL         Merc 450SLC  Cadillac Fleetwood
##            16.78571            16.78571            13.41429
## Lincoln Continental   Chrysler Imperial            Fiat 128
##            13.41429            13.41429            30.06667
##         Honda Civic      Toyota Corolla       Toyota Corona
##            30.06667            30.06667            22.58000
##    Dodge Challenger         AMC Javelin          Camaro Z28
##            16.78571            16.78571            13.41429
##    Pontiac Firebird           Fiat X1-9       Porsche 914-2
##            16.78571            30.06667            30.06667
##        Lotus Europa      Ford Pantera L        Ferrari Dino
##            30.06667            13.41429            19.74286
##       Maserati Bora          Volvo 142E
##            13.41429            22.58000
```

```r
#Classification
library(tree)
names(iris)[3:4]<-c("PL","PW")
(tree2<-tree(Species~PL+PW,data=iris))
```

```
## node), split, n, deviance, yval, (yprob)
##       * denotes terminal node
##
##  1) root 150 329.600 setosa ( 0.33333 0.33333 0.33333 )
##    2) PL < 2.45 50   0.000 setosa ( 1.00000 0.00000 0.00000 ) *
##    3) PL > 2.45 100 138.600 versicolor ( 0.00000 0.50000 0.50000 )
##      6) PW < 1.75 54  33.320 versicolor ( 0.00000 0.90741 0.09259 )
##       12) PL < 4.95 48   9.721 versicolor ( 0.00000 0.97917 0.02083 ) *
##       13) PL > 4.95 6   7.638 virginica ( 0.00000 0.33333 0.66667 ) *
##      7) PW > 1.75 46   9.635 virginica ( 0.00000 0.02174 0.97826 )
##       14) PL < 4.95 6   5.407 virginica ( 0.00000 0.16667 0.83333 ) *
##       15) PL > 4.95 40   0.000 virginica ( 0.00000 0.00000 1.00000 ) *
```

```
##
## Classification tree:
## tree(formula = Species ~ PL + PW, data = iris)
## Number of terminal nodes:  5
## Residual mean deviance:  0.157 = 22.77 / 145
## Misclassification error rate: 0.02667 = 4 / 150
```


PL < 2.45
setosa
PW < 1.75
PL < 4.95
versicolor
virginica
PL < 4.95
virginica
virginica

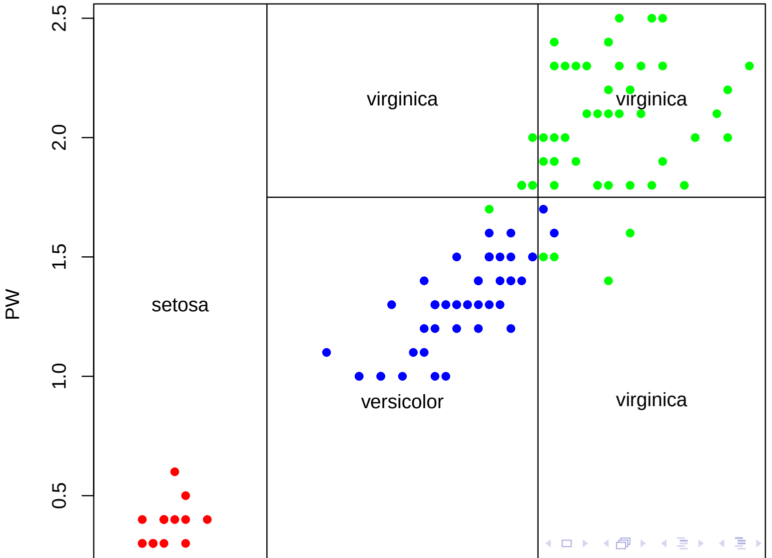
```r
predict(tree2)[1:10,]
```

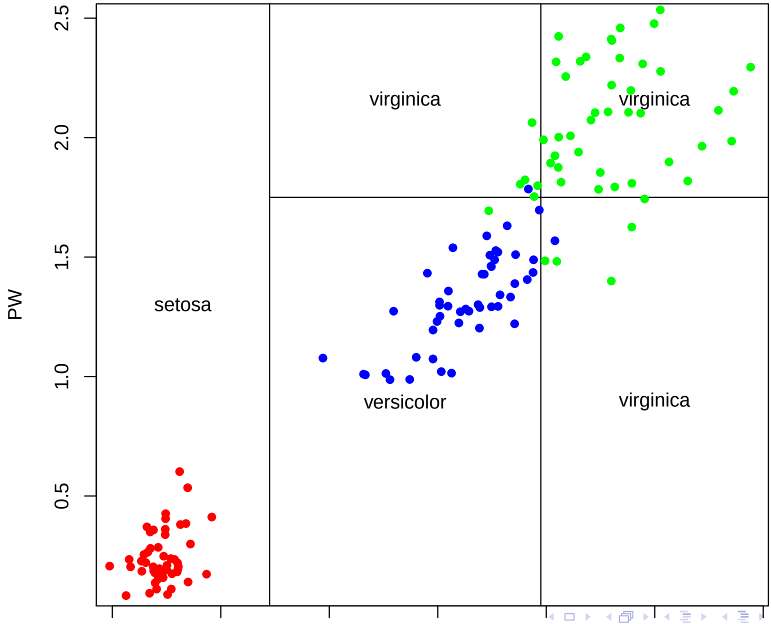
```
##    setosa versicolor virginica
## 1       1          0         0
## 2       1          0         0
## 3       1          0         0
## 4       1          0         0
## 5       1          0         0
## 6       1          0         0
## 7       1          0         0
## 8       1          0         0
## 9       1          0         0
## 10      1          0         0
```

```r
predict(tree2)[51:59,]
```

```
##    setosa versicolor  virginica
## 51      0  0.9791667 0.02083333
## 52      0  0.9791667 0.02083333
## 53      0  0.9791667 0.02083333
## 54      0  0.9791667 0.02083333
## 55      0  0.9791667 0.02083333
## 56      0  0.9791667 0.02083333
## 57      0  0.9791667 0.02083333
## 58      0  0.9791667 0.02083333
## 59      0  0.9791667 0.02083333
```

```r
partition.tree(tree2)
points(iris$PL,iris$PW,pch=16,col=c(rep("red",50),rep("blue",50),rep("green",50)))
```




2.5
2.0
1.5
1.0
0.5
PW
virginica
virginica
setosa
versicolor
virginica

- Once we have the leaves the models are determined
- So effort needs to be spent to find the best tree.

- Trees are controlled by how many leaves they have.
- By default in R, each leaf must contain a certain number of nodes and each split must reduce the MSE by a certain amount.

```r
#controlling the tree
tree.control(nobs,mincut=5,minsize=10,mindev=0.01)
```

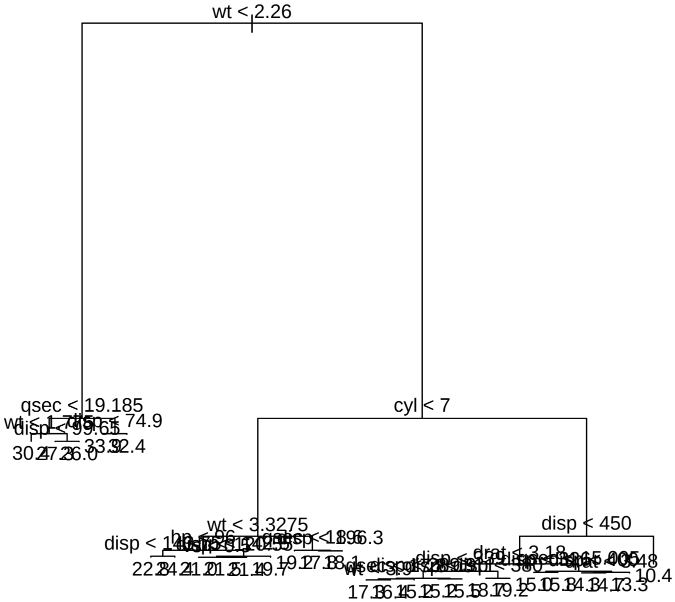
```
## Error in 4 * nobs:  non-numeric argument to binary operator
```

```r
#nobs: The number of observations in the training set.
#mincut: The minimum number of observations to include
#in either child node. This is a weighted quantity;
#the observational weights are used to compute the number.
#The default is 5.
#minsize: The smallest allowed node size: a weighted quantity.
#The default is 10.
#mindev: The within-node deviance must be at least this times
#that of the root node for the node to be split.
```

```r
library(tree)
(tree1_overfit<-tree(mpg~cyl+disp+hp+drat+wt+qsec+vs+am+gear+carb,data=mtcars,
                     control=tree.control(nobs=32,mincut=1,minsize=2,mindev=0.000001)))
```

```
## node), split, n, deviance, yval
##       * denotes terminal node
##
##   1) root 32 1.126e+03 20.09
##     2) wt < 2.26 6 4.455e+01 30.07
##       4) qsec < 19.185 4 1.491e+01 28.52
##         8) wt < 1.775 2 0.000e+00 30.40 *
##         9) wt > 1.775 2 8.450e-01 26.65
##          18) disp < 99.65 1 0.000e+00 27.30 *
##          19) disp > 99.65 1 0.000e+00 26.00 *
##       5) qsec > 19.185 2 1.125e+00 33.15
##        10) disp < 74.9 1 0.000e+00 33.90 *
##        11) disp > 74.9 1 0.000e+00 32.40 *
##     3) wt > 2.26 26 3.466e+02 17.79
##       6) cyl < 7 12 4.212e+01 20.92
##        12) wt < 3.3275 9 1.486e+01 21.78
##          24) hp < 96 3 1.707e+00 23.33
##            48) disp < 143.75 2 0.000e+00 22.80 *
##            49) disp > 143.75 1 0.000e+00 24.40 *
##          25) hp > 96 6 2.260e+00 21.00
##            50) hp < 142.5 5 2.320e-01 21.26
##             100) vs < 0.5 2 0.000e+00 21.00 *
##             101) vs > 0.5 3 6.667e-03 21.43
##               202) disp < 120.55 1 0.000e+00 21.50 *
##               203) disp > 120.55 2 0.000e+00 21.40 *
##            51) hp > 142.5 1 0.000e+00 19.70 *
##        13) wt > 3.3275 3 1.087e+00 18.37
##          26) qsec < 18.6 1 0.000e+00 19.20 *
##          27) qsec > 18.6 2 4.500e-02 17.95
##            54) disp < 196.3 1 0.000e+00 17.80 *
##            55) disp > 196.3 1 0.000e+00 18.10 *
##       7) cyl > 7 14 8.520e+01 15.10
##        14) disp < 450 12 3.366e+01 15.88
```

```
##
## Regression tree:
## tree(formula = mpg ~ cyl + disp + hp + drat + wt + qsec + vs +
##     am + gear + carb, data = mtcars, control = tree.control(nobs = 32,
##     mincut = 1, minsize = 2, mindev = 1e-06))
## Variables actually used in tree construction:
## [1] "wt"   "qsec" "disp" "cyl"  "hp"   "vs"   "drat"
## Number of terminal nodes:  27
## Residual mean deviance:  0 = 0 / 5
## Distribution of residuals:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0       0       0       0
```


wt < 2.26
qsec < 19.185
cyl < 7
disp < 450
wt < 3.3275
10.4

Details

Let's define some quantities:

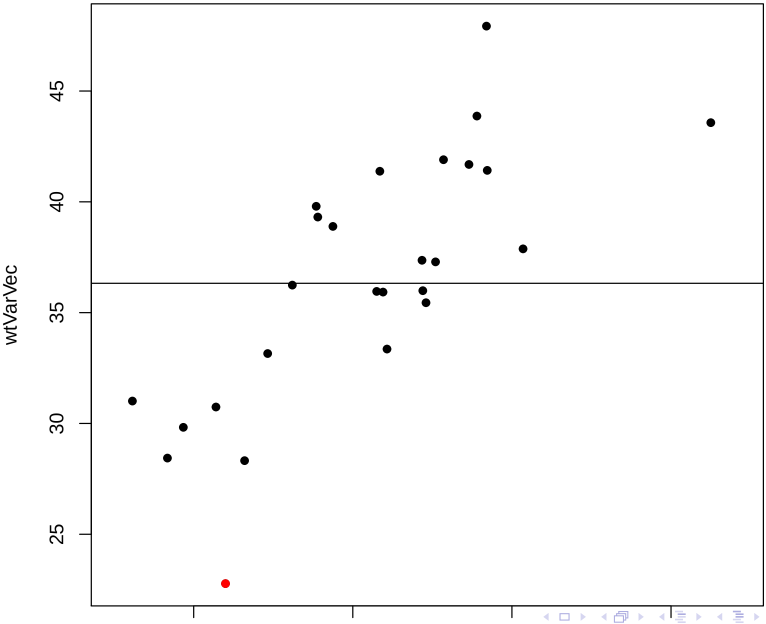
- $S = \Sigma_{c \in leaves(T)} \Sigma_{i \in C} (y_i - m_c)^2$
- $m_c = \frac{1}{n_c} \Sigma_{i \in C} y_i$
- We can re-write $S = \Sigma_{c \in leaves(T)} n_c V_c$ where V_c is the within-leaf variance.

Details

1. Start with a sinlge node containing all points. Calculate m_c and S.
2. If all the points in the node have the same value for all the independent variables, stop. Otherwise, search over all binary splits of all variables for the one which will reduce S as much as possible. If the largest decrease in S would be less than some threshold δ, or one of the resulting nodes would contain less than q points, stop. Otherwise, take that split, creating two new nodes.
3. In each new node, go back to step 1.

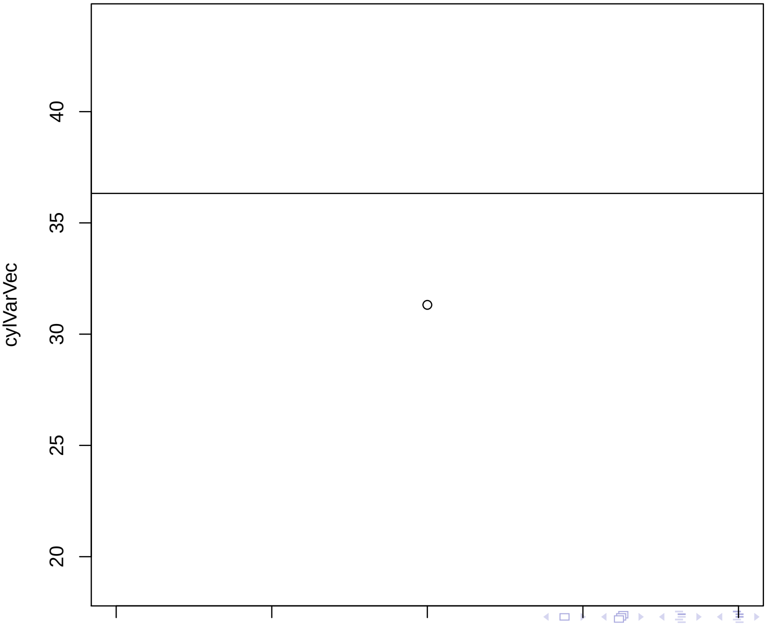
```
(tree1<-tree(mpg~wt+cyl,data=mtcars))

## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 32 1126.000 20.09
##    2) wt < 2.26 6   44.550 30.07 *
##    3) wt > 2.26 26  346.600 17.79
##      6) cyl < 7 12   42.120 20.92
##       12) cyl < 5 5    5.968 22.58 *
##       13) cyl > 5 7   12.680 19.74 *
##      7) cyl > 7 14   85.200 15.10 *

var(mtcars$mpg)

## [1] 36.3241
```

```
wtVec<-sort(unique(mtcars$wt))
wtVarVec<-rep(NA,length(wtVec))
for (i in 2:(length(wtVec)-1)){
cutoff<-wtVec[i]
wtVarVec[i]<-var(mtcars$mpg[mtcars$wt<=cutoff])+var(mtcars$
}
```



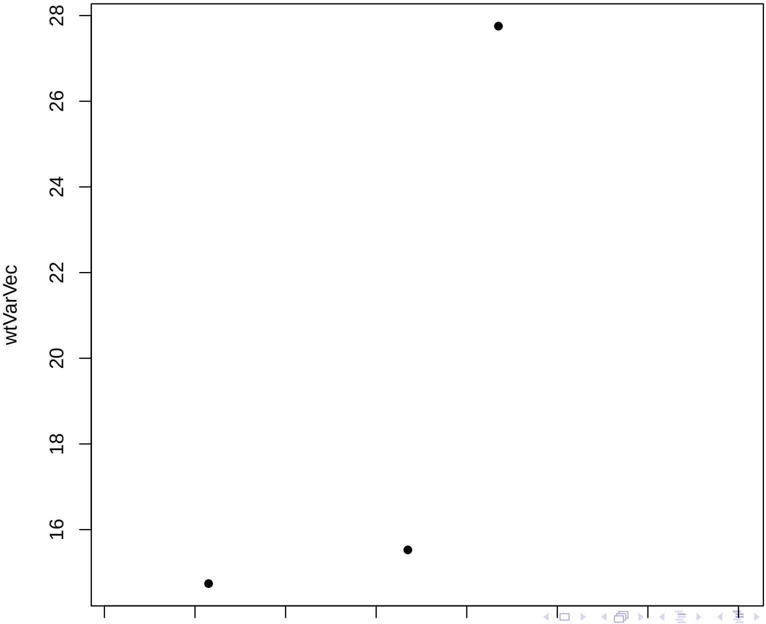
wtVarVec
45
40
35
30
25

```
cylVec<-sort(unique(mtcars$cyl))
cylVarVec<-rep(NA,length(cylVec))
for (i in 2:(length(cylVec)-1)){
cutoff<-cylVec[i]
cylVarVec[i]<-var(mtcars$mpg[mtcars$cyl<=cutoff])+var(mtcar
}
```


cylVarVec
40
35
30
25
20

```
#Now we split
split1<-mtcars[mtcars$wt<=2.2,]
var(split1$mpg)

## [1] 8.910667

split2<-mtcars[mtcars$wt>2.2,]
var(split2$mpg)

## [1] 13.86266
```

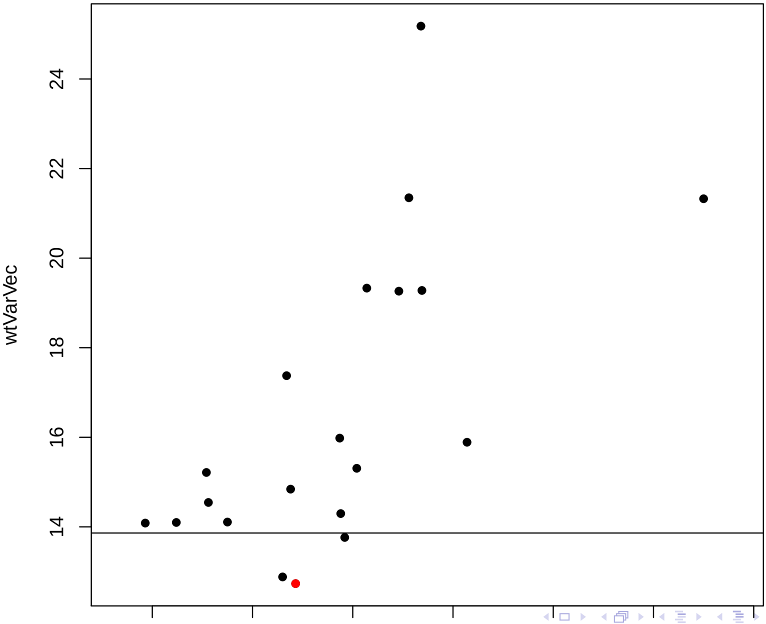
```
#Split 1
wtVec<-sort(unique(split1$wt))
wtVarVec<-rep(NA,length(wtVec))
for (i in 2:(length(wtVec)-1)){
cutoff<-wtVec[i]
wtVarVec[i]<-var(split1$mpg[split1$wt<=cutoff])+var(split1$
}
```


wtVarVec
16
18
20
22
24
26
28

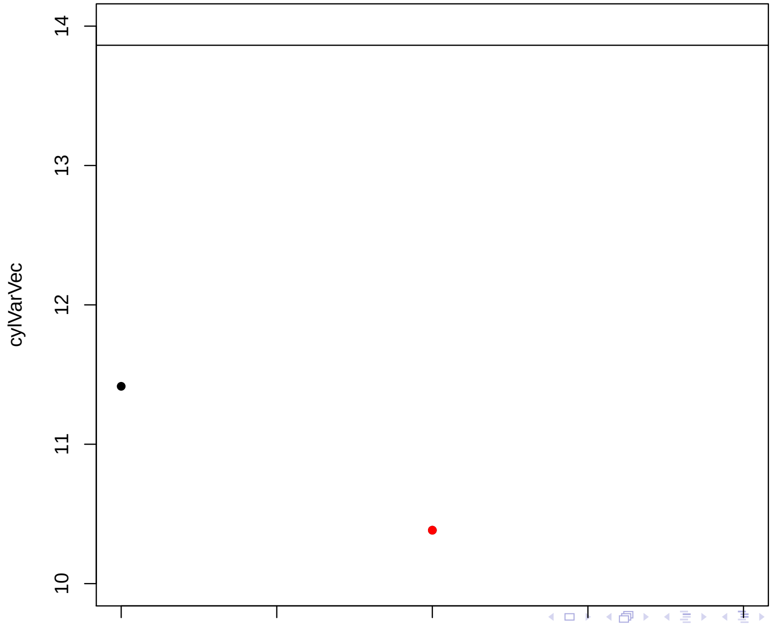
```
cylVec<-sort(unique(split1$cyl))
cylVarVec<-rep(NA,length(cylVec))
for (i in 2:(length(cylVec)-1)){
cutoff<-cylVec[i]
cylVarVec[i]<-var(split1$mpg[split1$cyl<=cutoff])+var(split
}
```

```
## Error in xy.coords(x, y, xlabel, ylabel, log):
'x' and 'y' lengths differ
## Error in int_abline(a = a, b = b, h = h, v = v,
untf = untf, ...):  plot.new has not been called yet
## Error in plot.xy(xy.coords(x, y), type = type,
...):  plot.new has not been called yet
```

```
#Split 2
wtVec<-sort(unique(split2$wt))
wtVarVec<-rep(NA,length(wtVec))
for (i in 2:(length(wtVec)-1)){
cutoff<-wtVec[i]
wtVarVec[i]<-var(split2$mpg[split2$wt<=cutoff])+var(split2$
}
```


wtVarVec
14
16
18
20
22
24

```
cylVec<-sort(unique(split2$cyl))
cylVarVec<-rep(NA,length(cylVec))
for (i in 1:(length(cylVec))){
cutoff<-cylVec[i]
cylVarVec[i]<-var(split2$mpg[split2$cyl<=cutoff])+var(split
}
```


cylVarVec
10
11
12
13
14