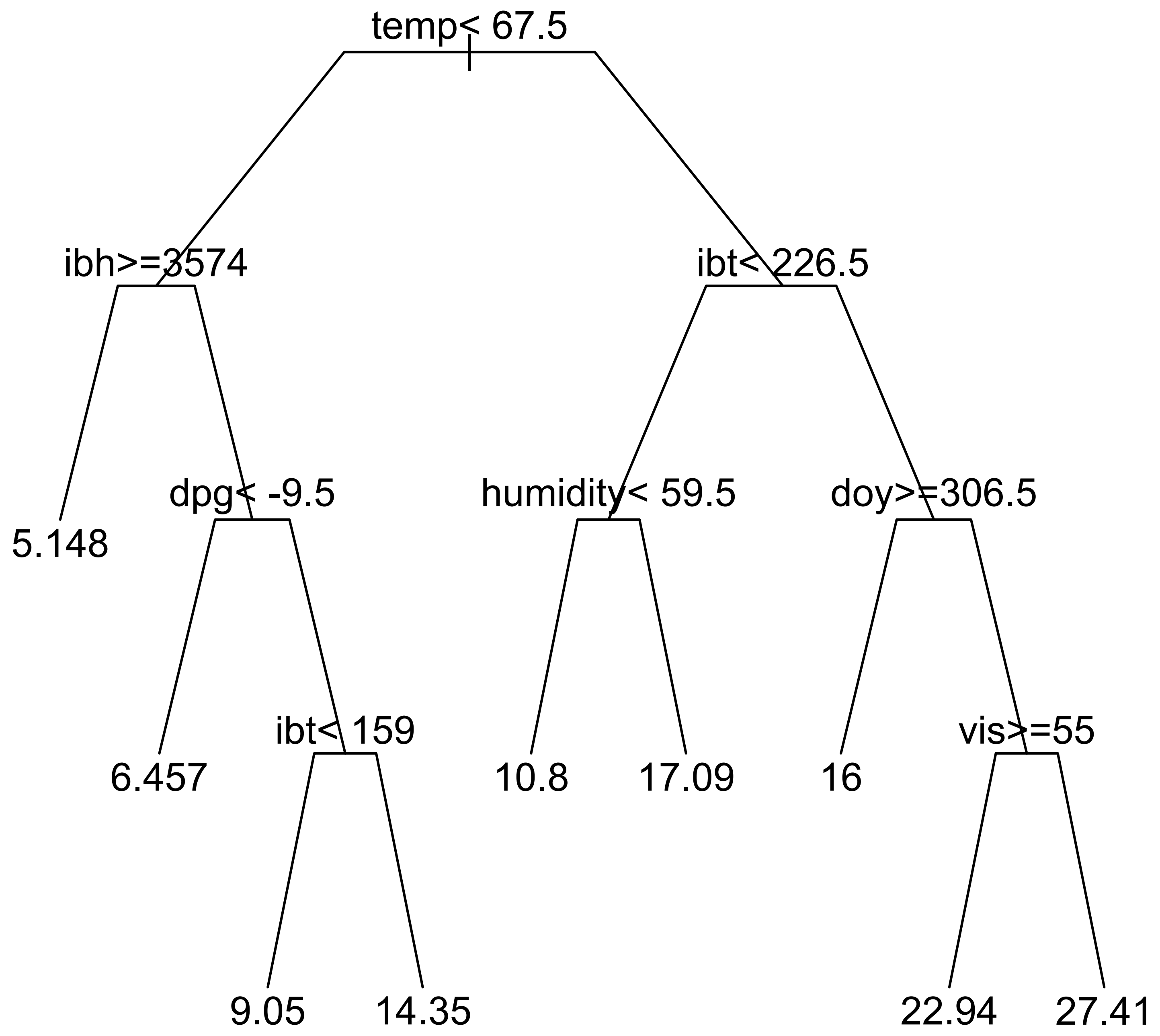
```
# Revisit the Ozone Data.

> ozone[1:5,]
  O3   vh wind humidity temp  ibh dpg ibt vis doy
1  3 5710    4       28   40 2693 -25  87 250  33
2  5 5700    3       37   45  590 -24 128 100  34
3  5 5760    3       51   54 1450  25 139  60  35
4  6 5720    4       69   35 1568  15 121  60  36
5  4 5790    6       19   45 2631 -33 123 100  37

> library(rpart)
> roz = rpart(O3 ~ .,ozone)
> roz
n= 330

node), split, n, deviance, yval
      * denotes terminal node

 1) root 330 21115.4100 11.775760
   2) temp< 67.5 214  4114.3040  7.425234
     4) ibh>=3573.5 108   689.6296  5.148148 *
     5) ibh< 3573.5 106  2294.1230  9.745283
      10) dpg< -9.5 35   362.6857  6.457143 *
      11) dpg>=-9.5 71  1366.4790 11.366200
        22) ibt< 159 40   287.9000  9.050000 *
        23) ibt>=159 31   587.0968 14.354840 *
   3) temp>=67.5 116  5478.4400 19.801720
     6) ibt< 226.5 55  1276.8360 15.945450
      12) humidity< 59.5 10   167.6000 10.800000 *
      13) humidity>=59.5 45   785.6444 17.088890 *
     7) ibt>=226.5 61  2646.2620 23.278690
      14) doy>=306.5 8   398.0000 16.000000 *
      15) doy< 306.5 53  1760.4530 24.377360
        30) vis>=55 36  1149.8890 22.944440 *
        31) vis< 55 17   380.1176 27.411760 *
```



```
> x0 = apply(ozone[,-1],2,median)
> predict(roz, data.frame(t(x0)))
       1
14.35484
```

## Regression Tree

1. How to fit a tree?

Starting from the root, check all possible splits for each variable, pick the one giving you the smallest RSS, then apply that split; then repeat this procedure on the left and right child nodes.

2. When to stop?

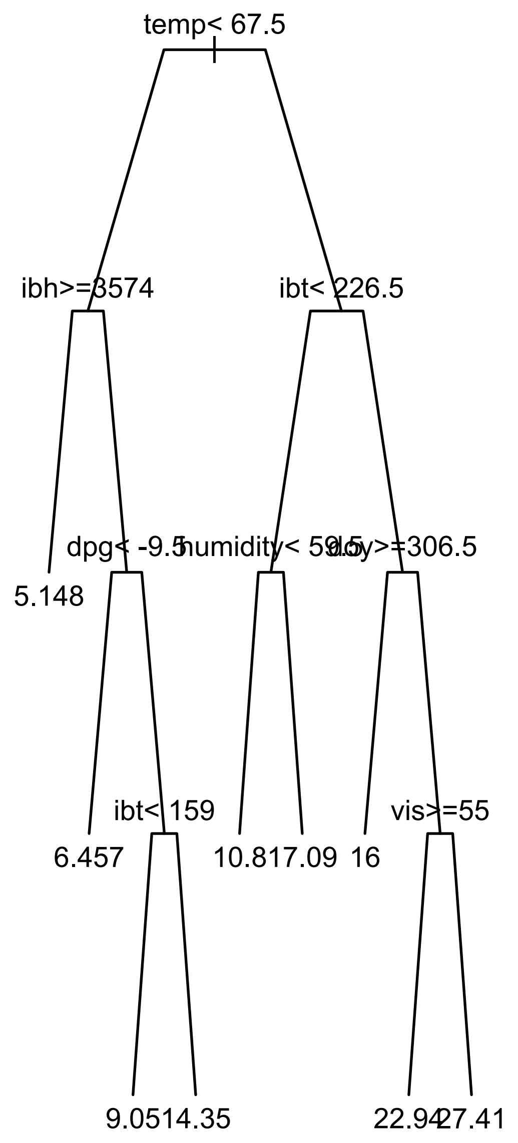
```
rpart.control(minsplit = 20, maxdepth = 30,
          cp = 0.01, minbucket = round(minsplit/3),...)
```

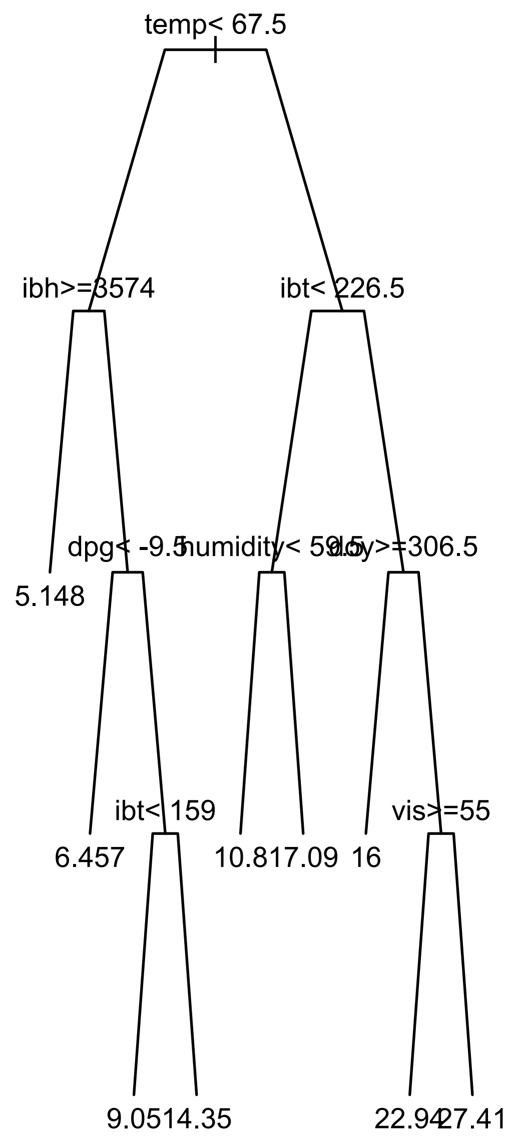
3. Model selection: pruning based on cost-complexity.

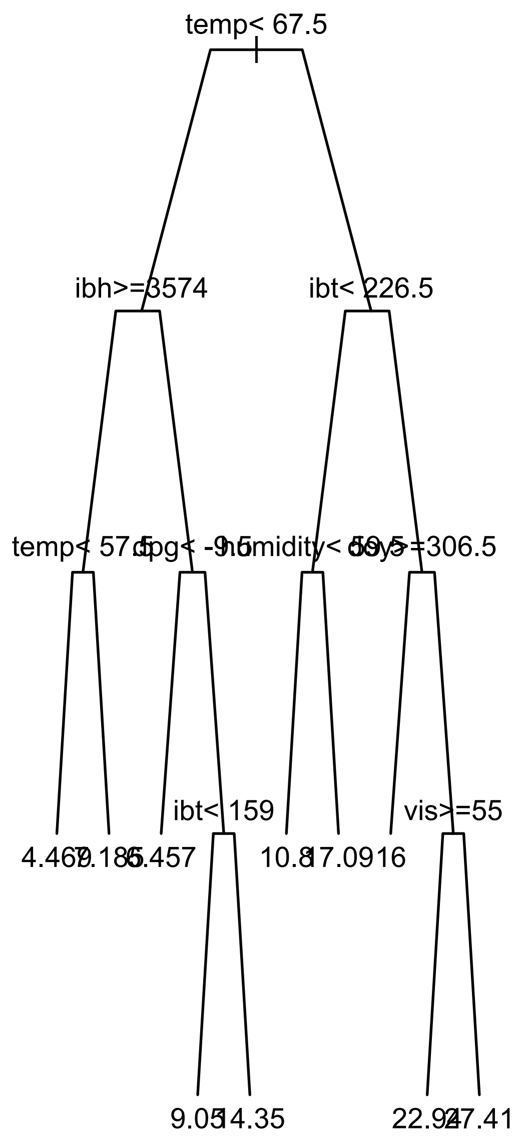
$$\mathrm{CC(T)} = \mathrm{RSS}(T) + \lambda \dim(T)$$

```
> roze = rpart(O3 ~ .,ozone, cp=0.001)
> printcp(roze)
          CP nsplit rel error  xerror     xstd
1  0.5456993      0   1.00000 1.01118 0.077468
2  0.0736591      1   0.45430 0.48154 0.041648
3  0.0535415      2   0.38064 0.40854 0.037994
4  0.0267557      3   0.32710 0.36686 0.035444
5  0.0232760      4   0.30034 0.34907 0.034853
6  0.0231021      5   0.27707 0.34213 0.034340
7  0.0153249      6   0.25397 0.35256 0.036782
8  0.0109137      7   0.23864 0.32955 0.034742
9  0.0070746      8   0.22773 0.33197 0.035263
10 0.0059918      9   0.22065 0.34639 0.037149
11 0.0059317     10   0.21466 0.35062 0.038163
12 0.0049709     12   0.20280 0.35473 0.038641
13 0.0047996     15   0.18789 0.36027 0.039335
14 0.0044712     16   0.18309 0.36079 0.039332
15 0.0031921     17   0.17861 0.36010 0.039849
16 0.0022152     19   0.17223 0.36198 0.039881
17 0.0020733     20   0.17002 0.35998 0.039858
18 0.0020297     22   0.16587 0.36016 0.039855
19 0.0014432     23   0.16384 0.35853 0.039846
20 0.0011322     24   0.16240 0.36055 0.039840
21 0.0011035     25   0.16126 0.36175 0.040008
22 0.0010000     26   0.16016 0.36275 0.040010
```

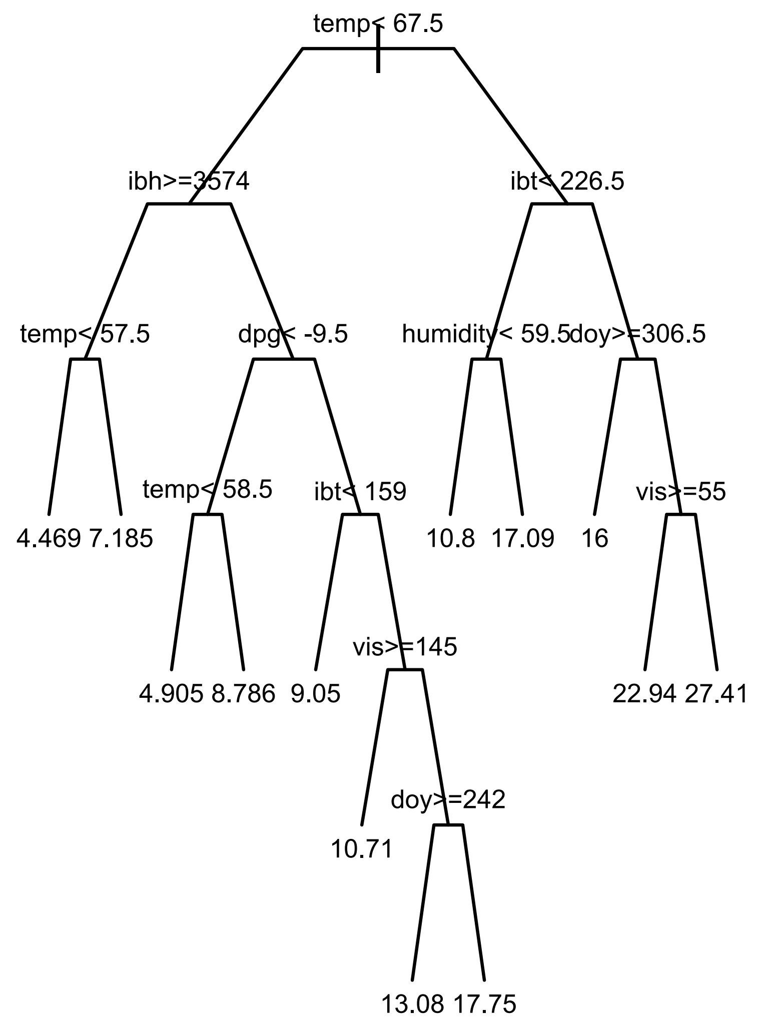
```
# The optimal tree (based on the cost-complexity criterion)
# stays the same for a range of CP values.
# The CP values listed on the previous slide give us
# the break-points where the returned optimal tree changes.

> tmp1=rpart(O3~., ozone, cp=0.01)
> tmp2=rpart(O3~., ozone, cp=0.008)
> tmp3=rpart(O3~., ozone, cp=0.007)

> junk=rpart(O3~., ozone, cp=0.6)
> junk
n= 330

node), split, n, deviance, yval
      * denotes terminal node

1) root 330 21115.41 11.77576 *
> var(ozone$O3)
[1] 64.18057
> dim(ozone)
[1] 330  10
> var(ozone$O3)*329
[1] 21115.41
```

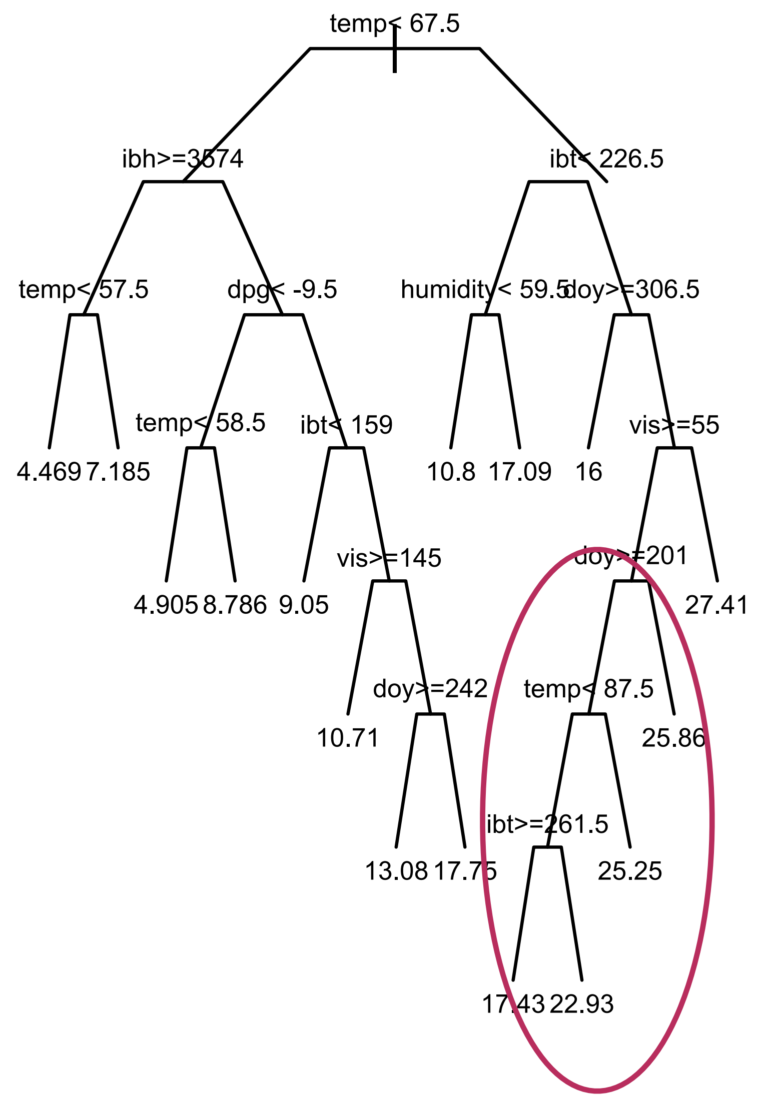

temp< 67.5
ibh>=3574
ibt< 226.5
5.148
dpg< -9.5
humidity< 59.5
doy>=306.5
6.457
ibt< 159
10.8
17.09
16
vis>=55
9.05
14.35
22.94
27.41

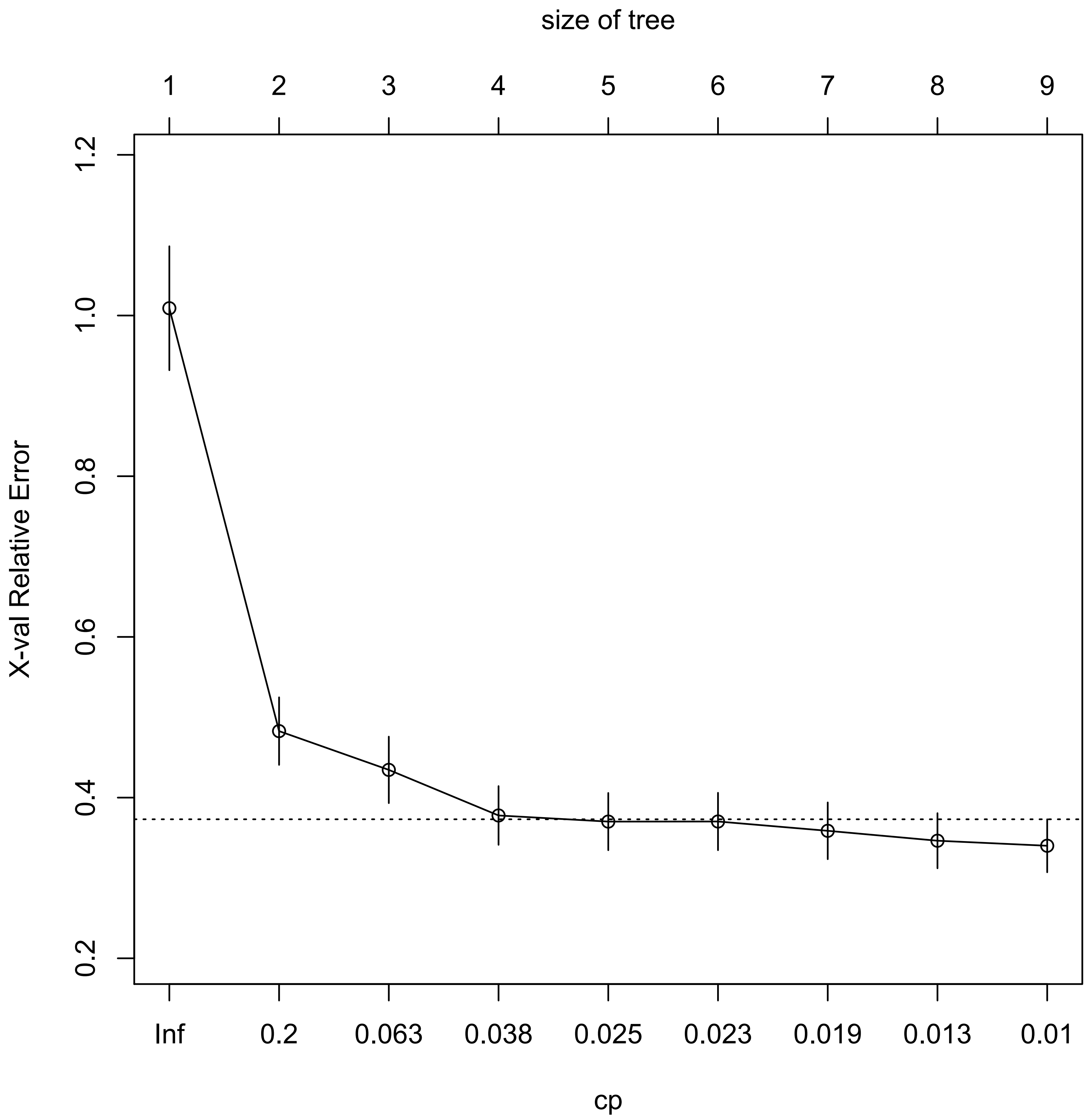

temp< 67.5
ibh>=3574
ibt< 226.5
5.148
dpg< -9.5
humidity< 59.5
doy>=306.5
6.457
ibt< 159
10.8
17.09
16
vis>=55
9.05
14.35
22.94
27.41


temp< 67.5
ibh>=3574
ibt< 226.5
temp< 57.5
dpg< -9.5
humidity< 59.5
doy>=306.5
4.467
7.186
6.457
ibt< 159
10.8
17.09
16
vis>=55
9.05
14.35
22.94
27.41

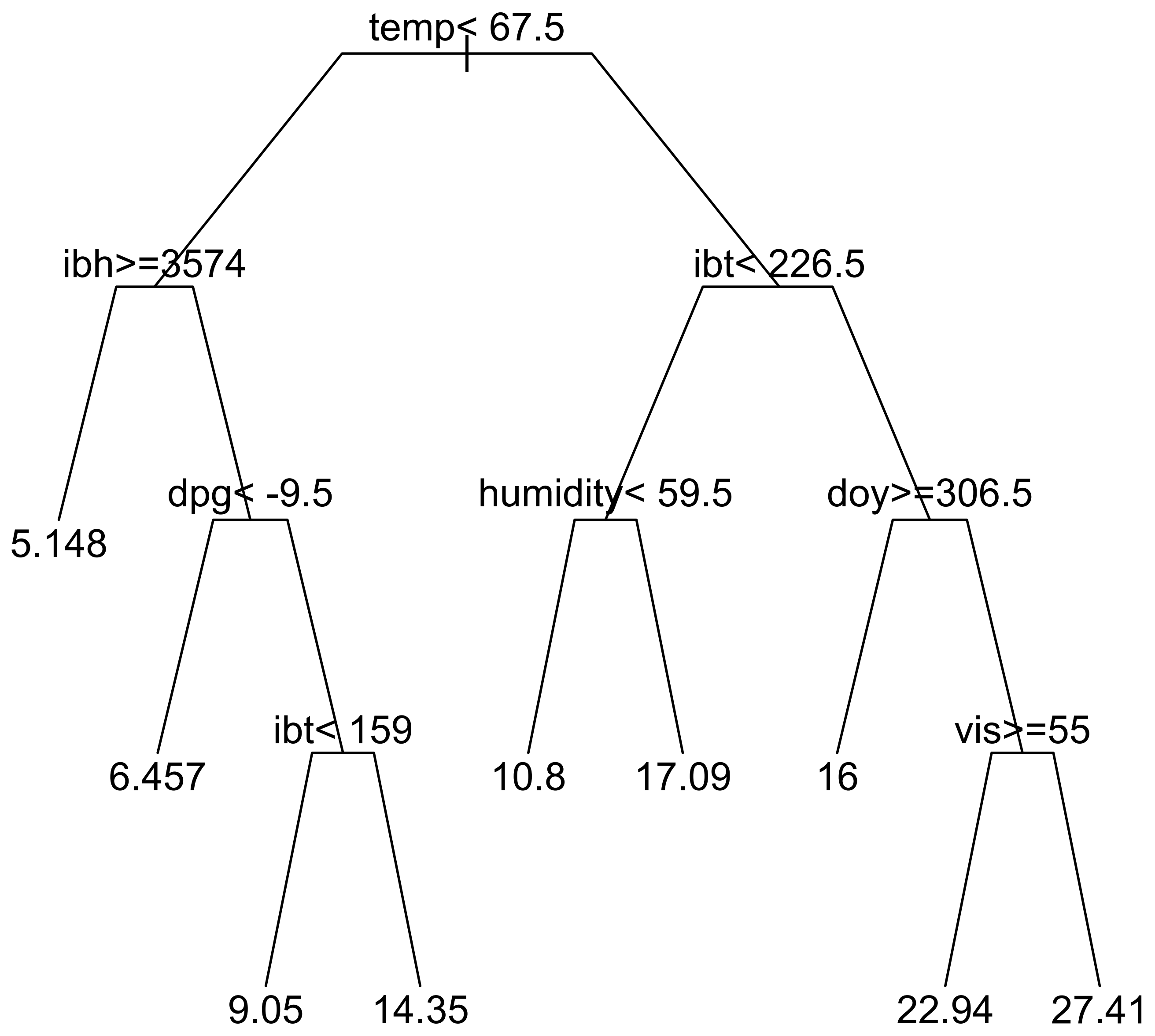
```
# The size (or equivalently the number of splits) is not
# continuous in terms of integers: we could have tree size
# jumps from 12 to 10, skipping size 11.

> tmp1=rpart(O3~., ozone, cp=0.0055)
> tmp2=rpart(O3~., ozone, cp=0.0049)

> par(mfrow=c(1,2))
> plot(tmp1,compress=T,uniform=T,branch=0.4,margin=.10)
> text(tmp1, cex=0.5)
> plot(tmp2,compress=T,uniform=T,branch=0.4,margin=.10)
> text(tmp2, cex=0.5)
```

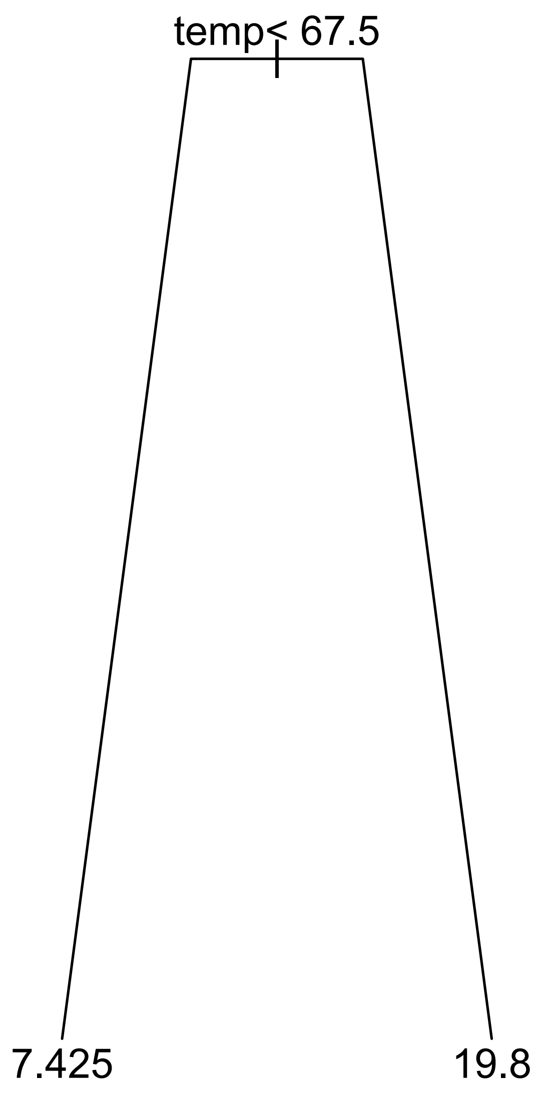

temp< 67.5
ibh>=3574
ibt< 226.5
temp< 57.5
dpg< -9.5
humidity< 59.5
doy>=306.5
4.469
7.185
temp< 58.5
ibt< 159
10.8
17.09
16
vis>=55
4.905
8.786
9.05
vis>=145
22.94
27.41
10.71
doy>=242
13.08
17.75


temp< 67.5
ibh>=3574
ibt< 226.5
temp< 57.5
dpg< -9.5
humidity< 59.5
doy>=306.5
temp< 58.5
ibt< 159
vis>=55
4.469
7.185
10.8
17.09
16
vis>=145
doy>=201
4.905
8.786
9.05
27.41
doy>=242
temp< 87.5
10.71
25.86
ibt>=261.5
13.08
17.75
25.25
17.43
22.93


size of tree
1
2
3
4
5
6
7
8
9
X-val Relative Error
0.2
0.4
0.6
0.8
1.0
1.2
Inf
0.2
0.063
0.038
0.025
0.023
0.019
0.013
0.01
cp

```
> plotcp(roz)
```





```
prune.rpart(roze,0.01091)

# Not correct. Should use a
# Cp value like 0.012 to get
# a tree with nsplit=7.
```

```
prune.rpart(roze, 0.408)
```

# Classification Tree

Classification trees work similarly to regression trees except the RSS is no longer a suitable criterion for splitting the nodes. Instead we use purity measures.

At node *j*, define

1. Deviance:

$$D_j = -2 \sum_{k=1}^{K} n_k \log \hat{p}_{jk}, \quad \hat{p}_{jk} = \frac{n_{jk}}{n_j}$$

2. Gini Index:

$$1 - \sum_{k} \hat{p}_{jk}^2, \quad \hat{p}_{jk} = \frac{n_{jk}}{n_j}$$

We have some training data consisting of 148 cases with the following variables: three possible species (*Giganteus*, *Melanops* and *Fuliginosus*), sex, and 18 skull measurements (Andrews and Herzberg, 1985).

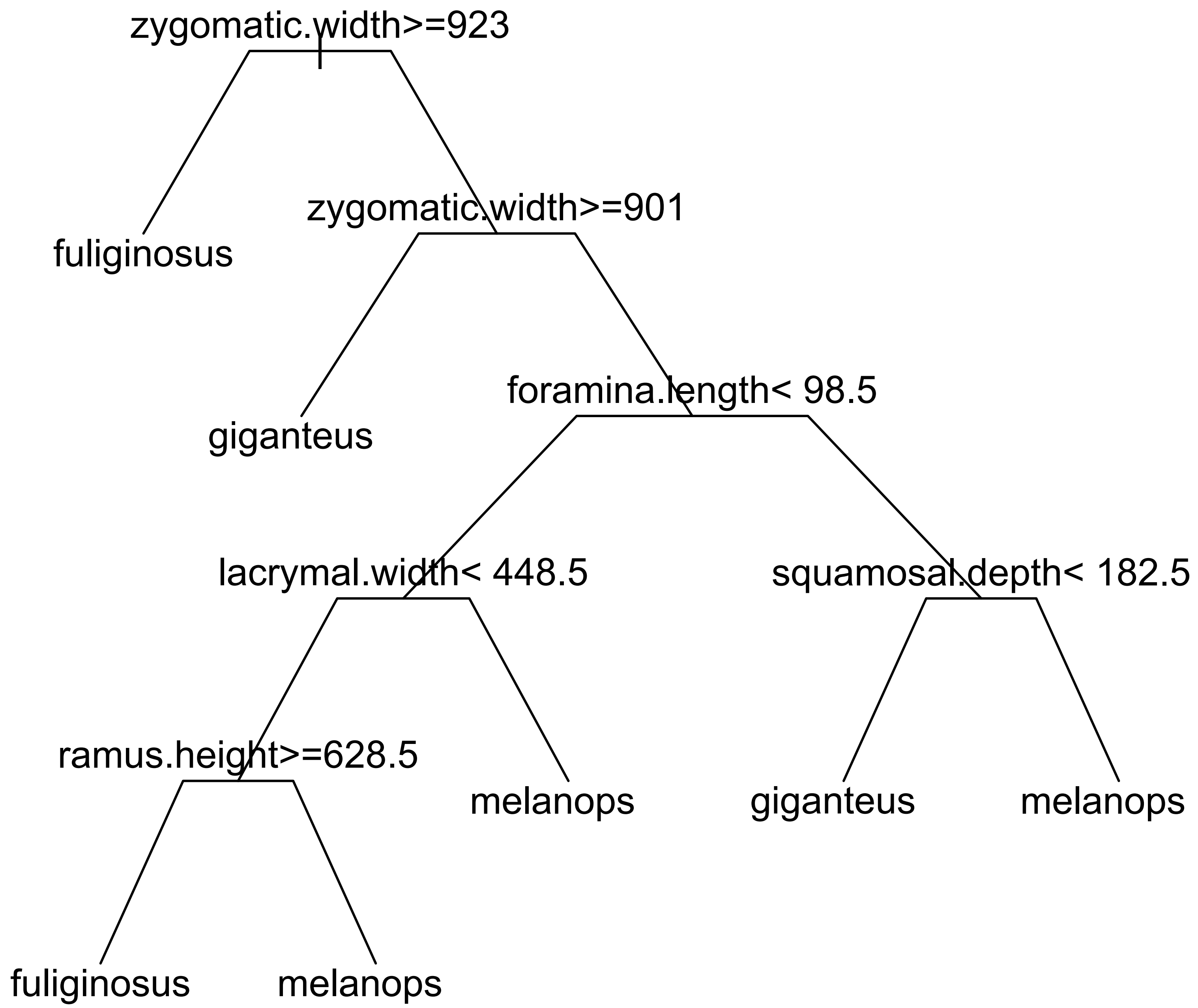
The goal is to identify the species of a historical specimen from the Rijksmuseum van Natuurlijkee in Leiden.

```
> kanga[1:3,]
      species  sex basilar.length occipitonasal.length palate.length
1 giganteus Male           1312                 1445           882
2 giganteus Male           1439                 1503           985
3 giganteus Male           1378                 1464           934

.........................................
.........................................
.........................................

  mandible.depth ramus.height
1            179          591
2            181          643
3            169          610
> x0=c(1115, NA, 748, 182, NA, NA, 178, 311, 756, 226, NA, NA,
NA, 48, 1009, NA, 204, 593)
> kanga = kanga[,c (T, F, !is.na(x0))]

> kanga[1:2,]
    species basilar.length palate.length palate.width squamosal.depth
1 giganteus           1312           882           NA             180
2 giganteus           1439           985          230             150
  lacrymal.width zygomatic.width orbital.width foramina.length
1            394             782           249              88
2            416             824           233             100
  mandible.length mandible.depth ramus.height
1            1086            179          591
2            1158            181          643
```

```
> apply(kanga,2,function(x) sum(is.na(x)))
        species  basilar.length   palate.length    palate.width
              0               1               1              24
squamosal.depth  lacrymal.width zygomatic.width   orbital.width
              1               0               1               0
foramina.length mandible.length  mandible.depth    ramus.height
              0              12               0               0

> round(cor(kanga[,-1],use="pairwise.complete.obs")[,c(3, 9)],
2)
                palate.width mandible.length
basilar.length          0.77            0.98
palate.length           0.81            0.98
palate.width            1.00            0.81
squamosal.depth         0.69            0.80
lacrymal.width          0.77            0.92
zygomatic.width         0.78            0.92
orbital.width           0.12            0.25
foramina.length         0.19            0.23
mandible.length         0.81            1.00
mandible.depth          0.62            0.85
ramus.height            0.73            0.94

> newko = na.omit (kanga [, -c(4, 10)])
> dim(newko)
[1] 144  10
```

```
> kt = rpart(species ~ ., data=newko,cp=0.01)
> printcp(kt)

Classification tree:
Root node error: 95/144 = 0.65972

n= 144
        CP nsplit rel error  xerror     xstd
1 0.178947      0   1.00000 1.22105 0.049992
2 0.105263      1   0.82105 0.96842 0.060672
3 0.050000      2   0.71579 0.84211 0.062767
4 0.021053      6   0.51579 0.80000 0.063060
5 0.010526      7   0.49474 0.76842 0.063152
6 0.010000      8   0.48421 0.83158 0.062859

> ktp = prune(kt,cp=0.011)
> table (newko$species, predict(ktp,type="class"))

              fuliginosus giganteus melanops
  fuliginosus          43         4        2
  giganteus            12        26       10
  melanops             15         4       28

> (43+26+28)/144
[1] 0.6736111
```

Training error rate



Plot of the Classification Tree.

## Grow a Forest

In an R package called "randomForest", we grow many (500, by default) classification/regression trees.

For a new sample, each tree gives a prediction, then the forest would use the average (for regression) or majority voting (for classification) as its prediction.

To reduce redundancy/correlation among trees, i) each tree is built based on a random subset of the data, and ii) each split is selected on a random subset of variables.

```
> library(randomForest);
> rfModel = randomForest(species~., data = newko)

>table(newko$species,  predict(rfModel,newko))
             fuliginosus giganteus melanops
  fuliginosus          49         0        0
  giganteus             0        48        0
  melanops              0         0       47
```

Training error rate

```
> table(newko$species, rfModel$predicted)

             fuliginosus giganteus melanops
  fuliginosus          37         4        8
  giganteus             5        21       22
  melanops             12        19       16
```

Test (CV) error rate