CART examples in R

Bin Li

1 Regression Tree Example: Body Fat Study

Overweight and obesity are considered to be major health problems because of their strong association with a higher risk of diseases of the metabolic syndrome, including diabetes mellitus and cardiovascular disease, as well as with certain forms of cancer. Obesity is frequently evaluated by using simple indicators such as body mass index, waist circumference, or waist-to-hip ratio. The bodyfat dataset contains 71 healthy German women with 9 common anthropometric measurements as input variables. The response variable is the women's body composition measured by Dual Energy X-Ray Absorptiometry (`DEXfat`). This reference method is very accurate in measuring body fat but finds little applicability in practical environments, mainly because of high costs and the methodological efforts needed. Therefore, a simple regression model for predicting `DEXfat` measurements of body fat is of special interest for the practitioner.

The `rpart` function from `rpart` library can be used to grow a regression tree. The response variable and the covariates are defined by a model formula in the same way as for `lm`. By default, a large initial tree is grown, we restrict the number of observations required to establish a potential binary split to at least ten (`minsplit = 10`). Besides `minsplit`, two other constraints in `rpart.control` are:

- `minbucket`: the minimum number of observations in any terminal node (also called leaf size). If only one of `minbucket` or `minsplit` is specified, the code either sets `minsplit` to `minbucket`*3 or `minbucket` to `minsplit`/3, as appropriate
- `maxdepth`: the maximum depth of any node of the final tree, with the root node counted as depth 0. Values greater than 30 rpart will give nonsense results on 32-bit machines

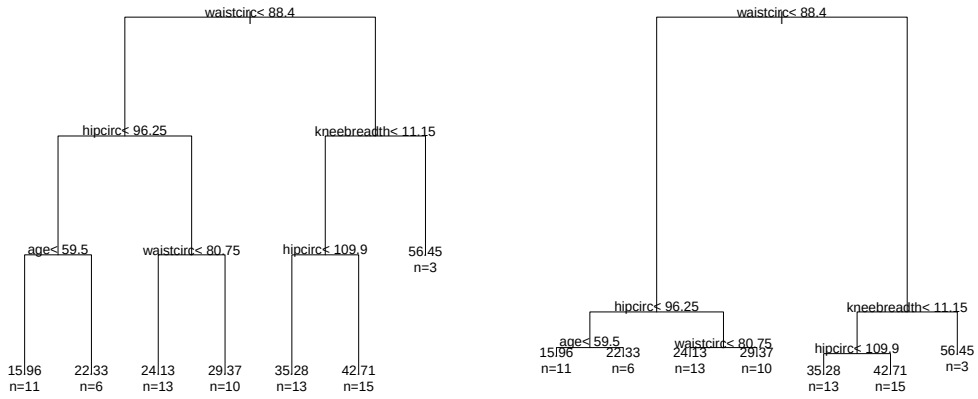
```
> library("rpart")
> data("bodyfat", package="TH.data")
> set.seed(1) #able to reproduce exact results for CV
> bodyfat_rpart <- rpart(DEXfat ~ age + waistcirc + hipcirc +
                         elbowbreadth + kneebreadth, data = bodyfat,
                         control = rpart.control(minsplit = 10))
> bodyfat_rpart

n= 71

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

 1) root 71 8535.98400 30.78282
```

```
2) waistcirc< 88.4 40 1315.35800 22.92375
  4) hipcirc< 96.25 17  285.91370 18.20765
    8) age< 59.5 11   97.00440 15.96000 *
    9) age>=59.5 6   31.45788 22.32833 *
  5) hipcirc>=96.25 23  371.86530 26.40957
   10) waistcirc< 80.75 13  117.60710 24.13077 *
   11) waistcirc>=80.75 10   98.99016 29.37200 *
3) waistcirc>=88.4 31 1562.16200 40.92355
  6) kneebreadth< 11.15 28  615.52590 39.26036
   12) hipcirc< 109.9 13  136.29600 35.27846 *
   13) hipcirc>=109.9 15   94.46997 42.71133 *
  7) kneebreadth>=11.15 3  146.28030 56.44667 *
```

We see that the first split (node 2 and 3) is on `waistcirc`, 40 subjects have waist circumference less than 88.4 with a mean response value of 22.9, whereas 31 observations have waist circumference greater than 88.4 with a mean response value of 40.9. The total RSS has been reduced from 8536 to $1315 + 1562 = 2877$.

The fitted tree diagram is shown below. For example, the rightmost leaf, which covers only 3 subjects, indicates the average body fat measured by DXA for the subjects with "waistcirc$\geq$88.4" and "kneebreadth$\geq$11.15" is 56.45. The key is observations that satisfy the condition shown for each node go to the left and observations that don't are element of the right branch in each node.

```
> #xpd=NA: otherwise on some devices the text is clipped
> par(mfrow=c(1,2),xpd=NA,cex=1.8)
> plot(bodyfat_rpart,uniform=T)
> text(bodyfat_rpart, use.n=TRUE)
> plot(bodyfat_rpart,uniform=F)
> text(bodyfat_rpart, use.n=TRUE)
```



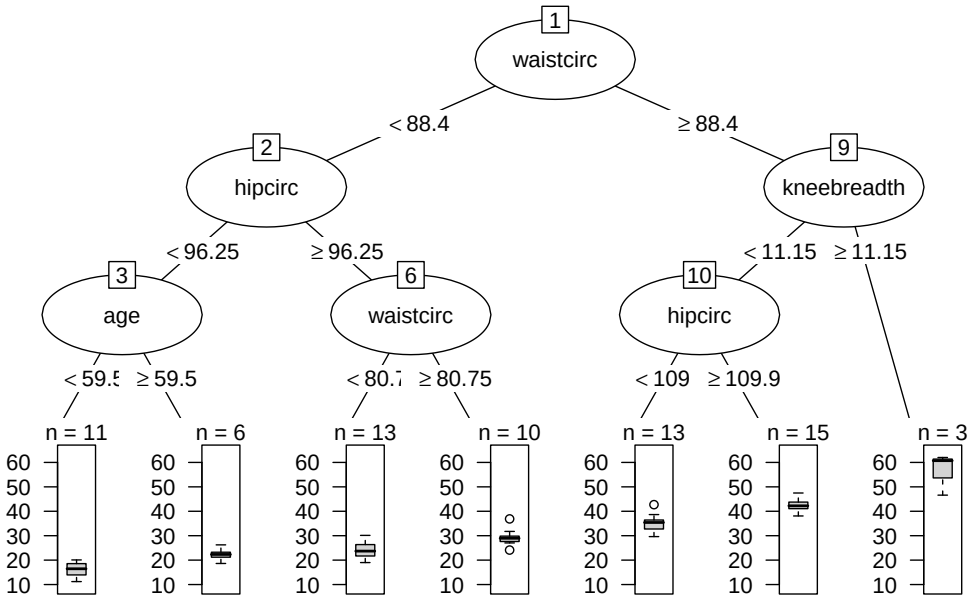
On the left panel, the depth of branches is held constant (by `uniform=T`) to improve readability. On the right, the depth is proportional to the improvement in fit.

Although the print method for `rpart` objects is informative; however, a graphical representation (here utilising functionality offered from package `partykit`, Hothorn and Zeileis, 2009) shown below

is more convenient. As expected, higher values for waist- and hip circumferences and wider knees correspond to higher values of body fat content. The rightmost terminal node consists of only three rather extreme observations.

```
> library("partykit")
```

```
> plot(as.party(bodyfat_rpart), tp_args = list(id = FALSE))
```



To determine if the tree is appropriate or if some of the branches need to be subjected to pruning we can use the cptable element of the rpart object:

```
> print(bodyfat_rpart$cptable)
```

```
          CP nsplit  rel error    xerror       xstd
1 0.66289544      0 1.00000000 1.0242885 0.16693802
2 0.09376252      1 0.33710456 0.4126456 0.09224327
3 0.07703606      2 0.24334204 0.4698277 0.09185447
4 0.04507506      3 0.16630598 0.3392513 0.06332402
5 0.01844561      4 0.12123092 0.2732376 0.06215470
6 0.01818982      5 0.10278532 0.2897925 0.06191890
7 0.01000000      6 0.08459549 0.2793817 0.06210235
```

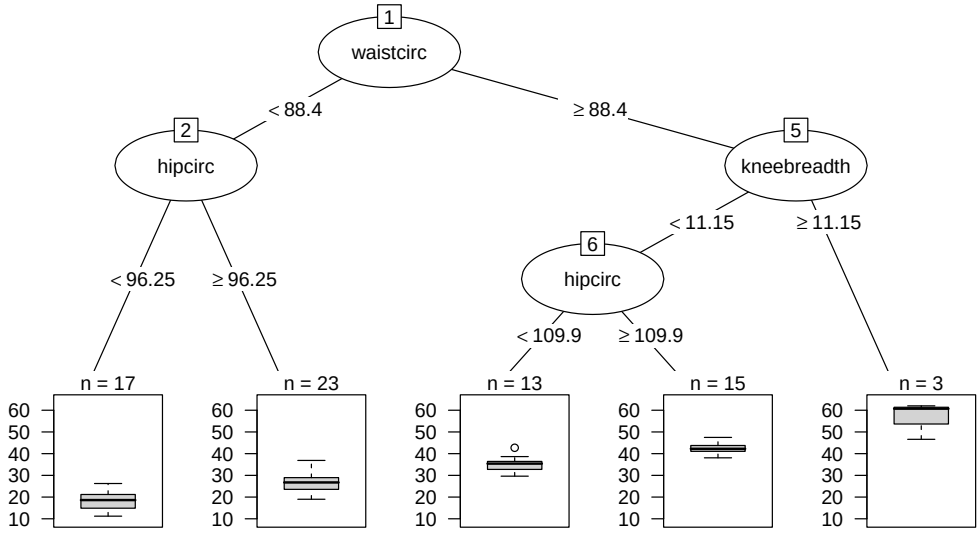
```
> opt <- which.min(bodyfat_rpart$cptable[,"xerror"])
```

The xerror column contains of estimates of cross-validated prediction error for different numbers of splits (nsplit). The best tree has four splits. Now we can prune back the large initial tree using

```
> cp <- bodyfat_rpart$cptable[opt, "CP"]
> bodyfat_prune <- prune(bodyfat_rpart, cp = cp)
```

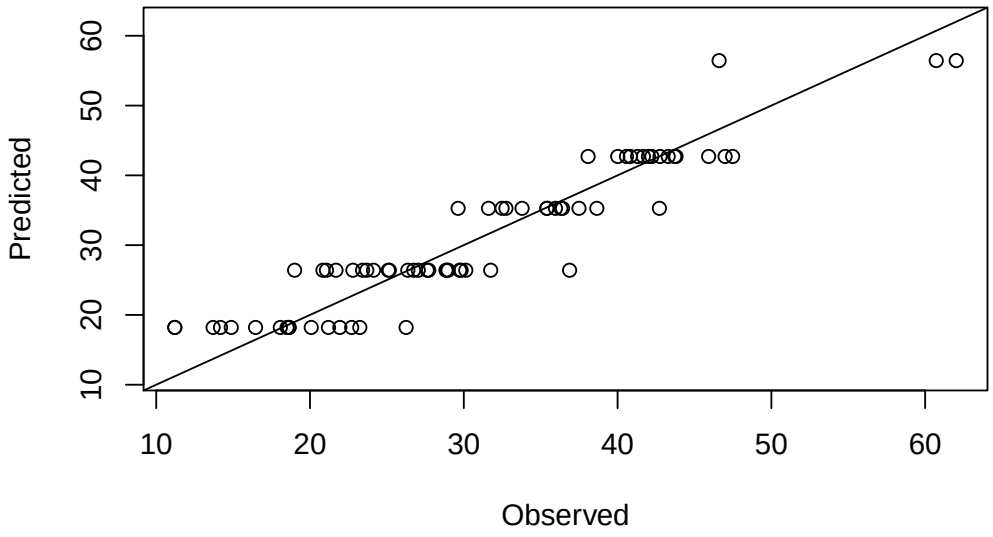
The result is shown below. Note that the inner nodes three and six have been removed from the tree. Still, the rightmost terminal node might give very unreliable extreme predictions.

```
> plot(as.party(bodyfat_prune), tp_args = list(id = FALSE))
```



Given this model, one can predict the (unknown, in real circumstances) body fat content based on the covariate measurements. Here, using the known values of the response variable, we compare the model predictions with the actually measured body fat as shown below. The three observations with large body fat measurements in the rightmost terminal node can be identified easily.

```
> DEXfat_pred <- predict(bodyfat_prune, newdata = bodyfat)
> xlim <- range(bodyfat$DEXfat)
> plot(DEXfat_pred ~ DEXfat, data = bodyfat, xlab = "Observed",
      ylab = "Predicted", ylim = xlim, xlim = xlim)
> abline(a = 0, b = 1)
```



2 Classification Tree Example: Glaucoma Eye Image Study

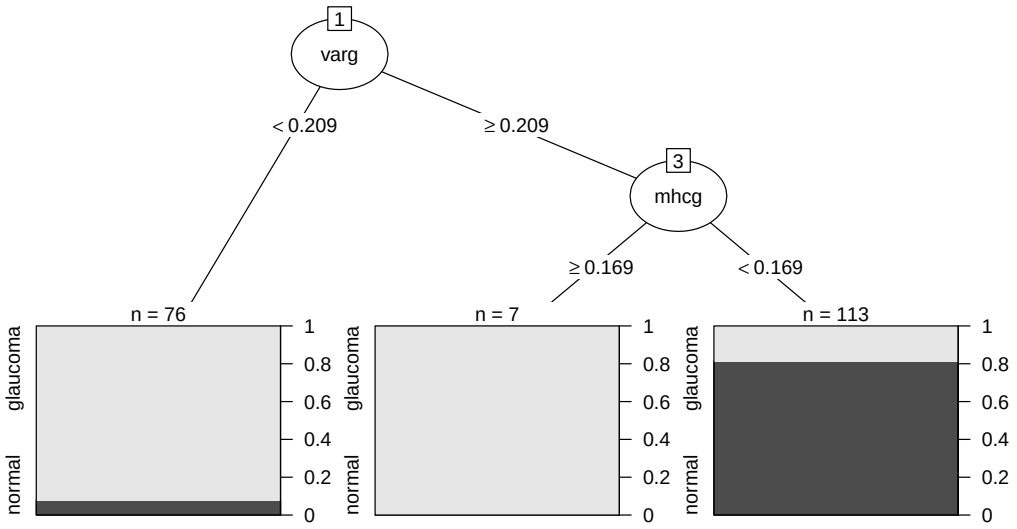
Here is another example: Glaucoma dataset. The dataset is from the investigation reported in Mardin et al. (2003) of whether laser scanner images of the eye background can be used to classify a patient's eye as suffering from glaucoma or not. Glaucoma is a neuro-degenerative disease of the optic nerve and is one of the major reasons for blindness in elderly people. The `GlaucomaM` dataset contains 196 observations in two classes. Namely 98 patients suffering glaucoma and 98 controls which have been matched by age and gender. 62 numeric variables are derived from a confocal laser scanning image of the optic nerve head, describing its morphology. Our aim is to construct a prediction model which is able to decide whether an eye is affected by glaucomateous changes based on the laser image data.

We start with a large initial tree and prune back branches according to the cross-validation criterion. The default is to use 10 runs of 10-fold CV and we choose 100 runs of 10-fold CV. Why?

```
> data("GlaucomaM", package = "TH.data")
> set.seed(0)
> glaucoma_rpart <- rpart(Class ~ ., data = GlaucomaM,
                          control = rpart.control(xval = 100))
> glaucoma_rpart$cptable

          CP nsplit rel error    xerror       xstd
1 0.65306122      0 1.0000000 1.5714286 0.05861799
2 0.07142857      1 0.3469388 0.3877551 0.05647630
3 0.01360544      2 0.2755102 0.3775510 0.05590431
4 0.01000000      5 0.2346939 0.4489796 0.05960655

> opt <- which.min(glaucoma_rpart$cptable[,"xerror"])
> cp <- glaucoma_rpart$cptable[opt, "CP"]
> glaucoma_prune <- prune(glaucoma_rpart, cp = cp)
```

The pruned tree consists of three leaves only; the class distribution in each leaf is depicted using a barplot. For most eyes, the decision about the disease is based on the variable varg, a measurement of the volume of the optic nerve above some reference plane. A volume larger than 0.209 mm^3 indicates that the eye is healthy, and damage of the optic nerve head associated with loss of optic nerves (varg smaller than 0.209 mm^3) indicates a glaucomateous change.

```
> plot(as.party(glaucoma_prune), tp_args = list(id = FALSE))
```



As we discussed earlier, the choice of the appropriatly sized tree is not a trivial problem. For the glaucoma data, the above choice of three leaves is very unstable across multiple runs of cross-validation. As an illustration of this problem we repeat the very same analysis as shown above and record the optimal number of splits as suggested by the cross-validation runs.

```
> nsplitopt <- vector(mode = "integer", length = 25)
> for (i in 1:length(nsplitopt)) {
   set.seed(i)
   cp <- rpart(Class ~ ., data = GlaucomaM)$cptable
   nsplitopt[i] <- cp[which.min(cp[,"xerror"]), "nsplit"]
 }
> nsplitopt

 [1] 5 5 1 1 2 5 1 2 1 5 1 1 1 1 2 2 2 1 2 2 1 5 1 1 2

> table(nsplitopt)

nsplitopt
 1  2  5
12  8  5
```

Although for 12 runs of cross-validation a simple tree with one split only is suggested, larger trees would have been favored in 13 of the cases. This short analysis shows that we should not trust the pruned tree shown above too much.

3 Classification Tree Example: Identifying Kangaroo Species

This is a study involving the identification of the sex and species of a historical specimen of kangaroo. The training data consisting of 148 cases with the following variables: three possible species, *Giganteus*, *Melanops* and *Fuliginosus*, the sex of the animial and 18 skull measurements. The data were published in Andrews and Herzberg (1985). The test sample is a historical specimen from a museum in Leiden which had the following skull measurements in the same order as in the data:

```
1115 NA 748 182 NA NA 178 311 756 226 NA NA NA 48 1009 NA 204 593
```

Here we only focus on identifying the species of a historical specimen of kangaroo. We start by reading in and specifying the museum case.

```
> library("faraway")
> data(kanga)
> x0 <- c(1115,NA,748,182,NA,NA,178,311,756,226,NA,NA,NA,48,1009,NA,204,593)
> str(kanga)

'data.frame':        148 obs. of  20 variables:
 $ species             : Factor w/ 3 levels "fuliginosus",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ sex                 : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
 $ basilar.length      : int  1312 1439 1378 1315 1413 1090 1294 1377 1296 1470 ...
 $ occipitonasal.length: int  1445 1503 1464 1367 1500 1195 1421 1504 1439 1563 ...
 $ palate.length       : int  882 985 934 895 969 740 872 954 878 987 ...
 $ palate.width        : int  NA 230 NA 230 NA NA 239 248 208 236 ...
 $ nasal.length        : int  609 629 620 564 645 493 606 660 630 672 ...
 $ nasal.width         : int  241 222 233 207 247 189 226 240 215 231 ...
 $ squamosal.depth     : int  180 150 135 158 161 122 155 159 NA 185 ...
 $ lacrymal.width      : int  394 416 403 394 426 350 396 417 387 429 ...
 $ zygomatic.width     : int  782 824 778 801 823 673 780 812 759 856 ...
 $ orbital.width       : int  249 233 244 224 241 234 237 240 248 227 ...
 $ .rostral.width      : int  227 248 240 242 252 185 238 245 219 268 ...
 $ occipital.depth     : int  531 632 575 568 607 462 577 614 584 659 ...
 $ crest.width         : int  153 141 144 116 120 188 149 128 151 103 ...
 $ foramina.length     : int  88 100 107 79 99 90 101 91 117 94 ...
 $ mandible.length     : int  1086 1158 1131 1090 1175 901 1084 1149 1069 1240 ...
 $ mandible.width      : int  131 148 116 132 131 101 124 129 121 132 ...
 $ mandible.depth      : int  179 181 169 189 197 138 168 175 159 196 ...
 $ ramus.height        : int  591 643 610 594 654 476 578 628 578 683 ...
```

Although CART can handle missing values in the input variables, for this special situation, where we want to classify one particular kangaroo, we just exclude all variables that are missing in the test case. We also drop sex since we will not modeling it.

```
> 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 mandible.length
1            394             782           249              88            1086
2            416             824           233             100            1158
  mandible.depth ramus.height
1            179          591
2            181          643
```

We still have missing values in the training set. Let's first check where the missing values occur:

```
> apply(kanga,2,function(x) sum(is.na(x)))
```

```
        species  basilar.length   palate.length    palate.width squamosal.depth
              0               1               1              24               1
 lacrymal.width zygomatic.width   orbital.width foramina.length mandible.length
              0               1               0               0              12
 mandible.depth    ramus.height
              0               0
```

We observe that the majority of missing values occur in just two variables: mandible length and palate width. Suppose we remove all the missing values and compute the pairwise correlation of these two variables with others.

```
> 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
```

We see that these two variables are highly correlated with other variables in the data. There is probably not much additional information in these two variables and we can reasonably discard them. So we do this and then remove the remaining missing cases.

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

[1] 144  10

> dim(kanga)
```

```
[1] 148  12

> dim(na.omit(kanga))

[1] 112  12
```

After excluding these two variables, we only lose four cases, whereas we would lose 36 cases by including these two variables. Now let's fit a classification.

```
> set.seed(2)
> kt <- rpart(species~., data=newko, cp=0.001)
> printcp(kt)

Classification tree:
rpart(formula = species ~ ., data = newko, cp = 0.001)

Variables actually used in tree construction:
[1] basilar.length  foramina.length lacrymal.width  ramus.height
[5] squamosal.depth zygomatic.width

Root node error: 95/144 = 0.65972

n= 144

        CP nsplit rel error  xerror     xstd
1 0.178947      0   1.00000 1.18947 0.051918
2 0.105263      1   0.82105 0.96842 0.060672
3 0.050000      2   0.71579 0.96842 0.060672
4 0.021053      6   0.51579 0.87368 0.062416
5 0.010526      7   0.49474 0.89474 0.062120
6 0.001000      8   0.48421 0.88421 0.062275
```

The cross-validation error reaches a minimum for the six-split tree. We select this tree:

```
> ktp <- prune(kt,cp=0.0211)
> ktp

n= 144

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 144 95 fuliginosus (0.34027778 0.33333333 0.32638889)
   2) zygomatic.width>=923 37 13 fuliginosus (0.64864865 0.16216216 0.18918919) *
   3) zygomatic.width< 923 107 65 giganteus (0.23364486 0.39252336 0.37383178)
     6) zygomatic.width>=901 16  3 giganteus (0.12500000 0.81250000 0.06250000) *
     7) zygomatic.width< 901 91 52 melanops (0.25274725 0.31868132 0.42857143)
      14) foramina.length< 98.5 58 33 melanops (0.36206897 0.20689655 0.43103448)
```

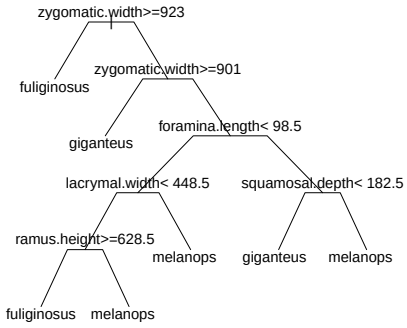
```
      28) lacrymal.width< 448.5 50 29 fuliginosus (0.42000000 0.24000000 0.34000000)
        56) ramus.height>=628.5 33 14 fuliginosus (0.57575758 0.18181818 0.24242424) *
        57) ramus.height< 628.5 17  8 melanops (0.11764706 0.35294118 0.52941176) *
      29) lacrymal.width>=448.5 8  0 melanops (0.00000000 0.00000000 1.00000000) *
    15) foramina.length>=98.5 33 16 giganteus (0.06060606 0.51515152 0.42424242)
      30) squamosal.depth< 182.5 26 10 giganteus (0.07692308 0.61538462 0.30769231) *
      31) squamosal.depth>=182.5 7  1 melanops (0.00000000 0.14285714 0.85714286) *
```

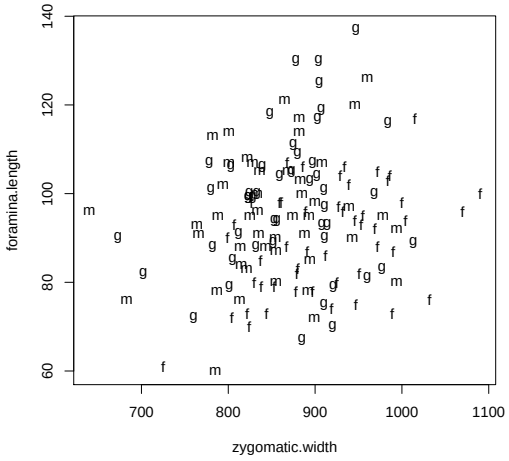
```
> par(mfrow=c(1,2))
> plot(ktp,compress=T,uniform=T,branch=0.4,margin=0.1)
> text(ktp)
> plot(foramina.length ~ zygomatic.width, data=newko,pch=substring(species,1,1))
> (tt <- table(actual=newko$species,predicted=predict(ktp,type="class")))
```

```
            predicted
actual       fuliginosus giganteus melanops
  fuliginosus          43         4        2
  giganteus            12        29        7
  melanops             15         9       23
```

```
> 1-sum(diag(tt))/sum(tt)
```

```
[1] 0.3402778
```





This tree is not particularly successful as the CV error is estimated as 87% of just guessing the species. Some of the terminals are quite pure, for example, Node 29 and 31, while others retain much uncertainty, for example, Node 56 and 57. We see that the overall error rate is 34%. We see that we can generally correctly identify *fuliginosus*, but more likely to be in error in distinguishing *melanops* and *giganteus*.

A look at the right panel explains why we may have difficulty in classification. Any single measure will reflect mostly the overall size of the skull not the 'shape' or the relative dimensions of the skull. One possibility is to allow splits on linear combinations of variables, which is allowed in some classification tree software implementations. An alternative idea is to apply the method to the principal component (PC) scores rather than the raw data. PCs seek out the main directions of variation in the data and might generate more effective predictors for classification.

```
> pck <- princomp(newko[,-1])
> pcdf <- data.frame(species=newko$species,pck$scores)
> dim(pcdf)

[1] 144  10

> pcdf[1:2,1:6]

    species     Comp.1   Comp.2    Comp.3     Comp.4     Comp.5
1 giganteus -271.21047 20.39092  8.478316 -12.270721  0.2167112
2 giganteus  -98.39458 45.46744 -1.823037   7.515042 -9.2393780

> set.seed(1)
> kt2 <- rpart(species~.,pcdf,cp=0.001)
> printcp(kt2)

Classification tree:
rpart(formula = species ~ ., data = pcdf, cp = 0.001)

Variables actually used in tree construction:
[1] Comp.2 Comp.3 Comp.4 Comp.7

Root node error: 95/144 = 0.65972

n= 144

        CP nsplit rel error  xerror     xstd
1 0.400000      0   1.00000 1.11579 0.055672
2 0.178947      1   0.60000 0.64211 0.062416
3 0.042105      2   0.42105 0.47368 0.058549
4 0.010526      3   0.37895 0.53684 0.060412
5 0.001000      5   0.35789 0.56842 0.061152
```

We find a much smaller CV error (0.474). Before we can predict the test case, we need to do some work to remove the missing values, unused variables and apply the PC transformation:

```
> nx0 <- x0[!is.na(x0)]
> nx0 <- nx0[-c(3,9)]
> nx0 <- (nx0-pck$center)/pck$scale
> test.sample <- as.data.frame(nx0 %*% pck$loadings)
> test.sample

     Comp.1   Comp.2    Comp.3   Comp.4   Comp.5   Comp.6    Comp.7    Comp.8
1 -499.9313 -74.8336 -37.63232 23.16925 3.956425 16.58442 -54.01662 -35.99481
    Comp.9
1 16.70477
```

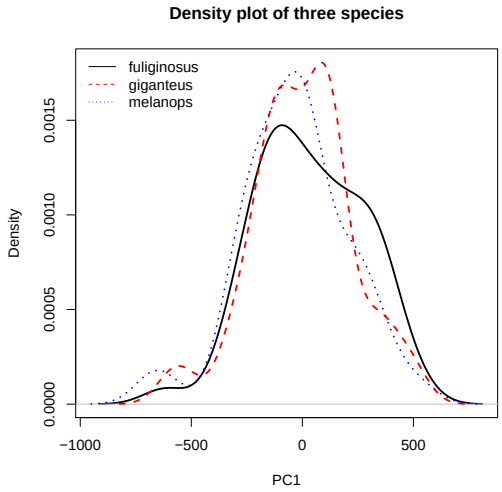
Our chosen tree is:

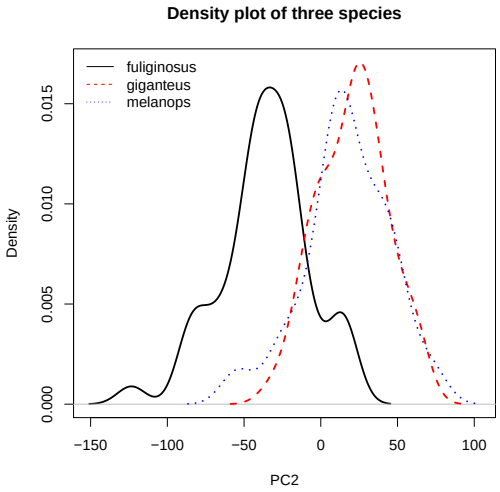
```
> ktp2 <- prune.rpart(kt2,0.0421)
> ktp2

n= 144

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 144 95 fuliginosus (0.34027778 0.33333333 0.32638889)
   2) Comp.2< -15.12579 49  8 fuliginosus (0.83673469 0.04081633 0.12244898) *
   3) Comp.2>=-15.12579 95 49 giganteus (0.08421053 0.48421053 0.43157895)
     6) Comp.4>=-9.512962 63 24 giganteus (0.11111111 0.61904762 0.26984127)
      12) Comp.3>=-18.9955 55 17 giganteus (0.09090909 0.69090909 0.21818182) *
      13) Comp.3< -18.9955 8  3 melanops (0.25000000 0.12500000 0.62500000) *
     7) Comp.4< -9.512962 32  8 melanops (0.03125000 0.21875000 0.75000000) *

> predict(ktp2,newdata=test.sample,tye="class")

  fuliginosus  giganteus melanops
1   0.8367347 0.04081633 0.122449
```

It is interesting that the first PC is not used. Typically the first PC represents an overall average or total size. Other PCs represent contrasts between variables which would describe shape features in this case. We see that the test case is classified as *fuliginosus*, which agrees with the experts. We can also compute the overall error rate as before:

```
> (tt <- table(newko$species,predict(ktp2,type="class")))

             fuliginosus giganteus melanops
  fuliginosus          41         5        3
  giganteus             2        38        8
  melanops              6        12       29

> 1-sum(diag(tt))/sum(tt)

[1] 0.25
```

We see the error rate has been reduced to 25%. The following figure shows the density plot of three species. We can see why the first PC is not used in the tree. It would be worth considering other combinations of predictors in an attempt to reduce the error rate further.


Density plot of three species
fuliginosus
giganteus
melanops
Density
0.0000
0.0005
0.0010
0.0015
−1000
−500
0
500
PC1


Density plot of three species
fuliginosus
giganteus
melanops
Density
0.000
0.005
0.010
0.015
−150
−100
−50
0
50
100
PC2