Classification and Regression Tree Example

Packages used

```
##   ade4 mvpart  rpart  vegan
##  1.6-2  1.6-2  4.1-8 2.0-10
```

Data

Load the `rpart` package and take a look at data on cars from *Consumer Reports* (price, reliability, mileage, etc.):

```
library(rpart)
summary(cu.summary)
```

(n.b.: do **not** load the `mvpart` package in this section – it will load a different version of the `rpart()` function. If you load it by accident, use `det ach("package:mvpart")` to get rid of it)

Grow tree

```
(fit <- rpart(Mileage~Price + Country + Reliability + Type,
        method="anova", xval =100, data=cu.summary))
```

If you know you want to use all the variables other than the response variable (`Mileage` in this case) as predictors, you can use the formula `Mileage~.`.

If your data are binary and so you want to build a classification tree rather than a regression tree, use `method="class"`.

Display results (basic information plus a table of "complexity parameter" CP, number of splits, relative error, cross-validation error (`xerror`) and the standard deviation of the cross-validation error (`xstd`).

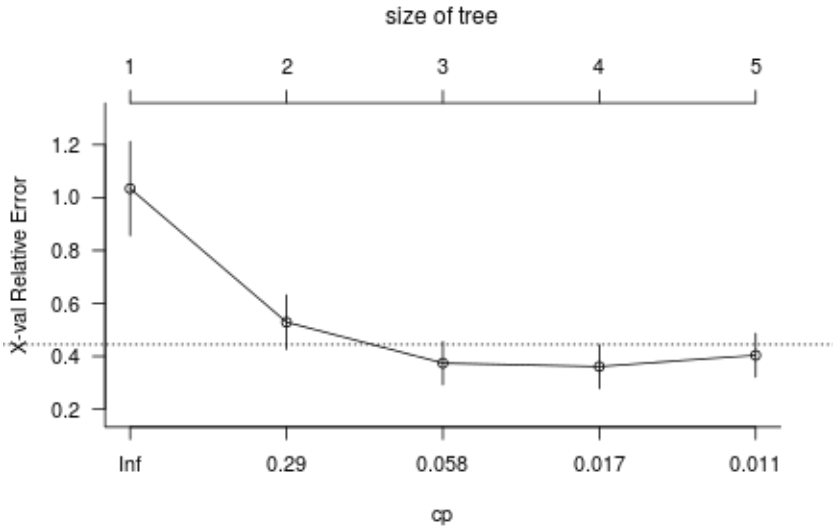
```
printcp(fit)
```

```
##
## Regression tree:
## rpart(formula = Mileage ~ Price + Country + Reliability + Type,
##     data = cu.summary, method = "anova", xval = 100)
##
## Variables actually used in tree construction:
```

```
## [1] Price Type
##
## Root node error: 1355/60 = 23
##
## n=60 (57 observations deleted due to missingness)
##
##      CP nsplit rel error xerror  xstd
## 1 0.623      0      1.00   1.03 0.178
## 2 0.132      1      0.38   0.53 0.103
## 3 0.025      2      0.25   0.37 0.082
## 4 0.012      3      0.22   0.36 0.083
## 5 0.010      4      0.21   0.40 0.083
```

We can plot the cross-validation error as a function of the number of splits:

```
plotcp(fit)
```

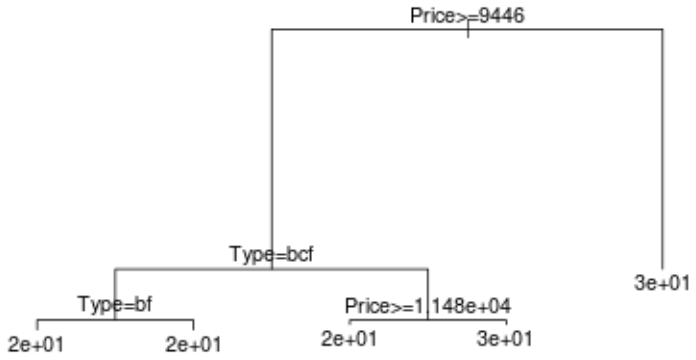


Or get detailed information about the tree:

```
summary(fit) # detailed summary of splits
```

We can also plot the tree itself (`plot` gives just the branches, `text` adds labels):

```
plot(fit)
text(fit)
```

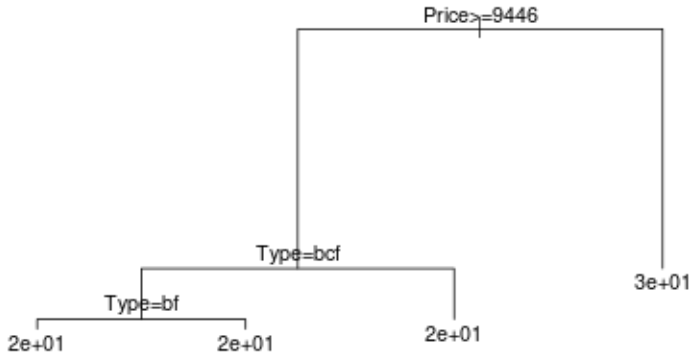


Now we can prune the tree back to its optimal size (based on cross-validation error):

```
best.tree <- which.min(fit$cptable[,"xerror"])
cpval <- fit$cptable[best.tree,"CP"]
pfit <- prune(fit, cp=cpval)
```

The resulting tree is simpler (only 4 splits; note that all the trees above 3 splits have about the same `xerror`):

```
plot(pfit)
text(pfit)
```




```
summary(pfit)
```

Pick your own tree size

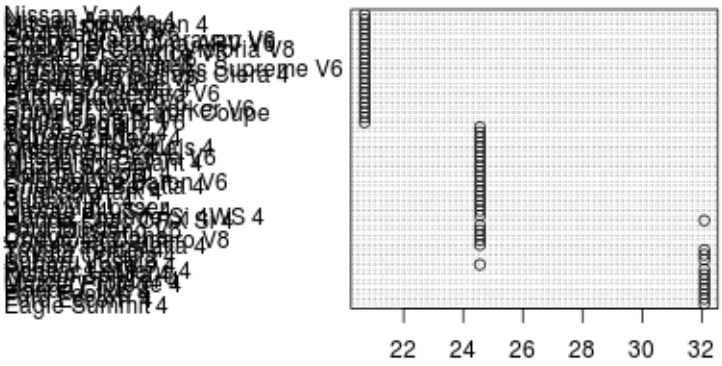
```
dfit <- rpart(Mileage~., method="anova",
              maxdepth=2, data=cu.summary)

plot(dfit)
text(dfit)
```


```
summary(dfit)
```

(Code modified from Quick-R: Accessing the Power of R)

We can also make predictions:

```
dotchart(predict(dfit))
```



Multivariate Regression Tree Example

Data

Data information

```
library(ade4)
data(doubs)
env <- doubs$env   ## site (row) * env variable (column)
spe <- doubs$fish  ## site (row) * species (column)
```

Transform response variables

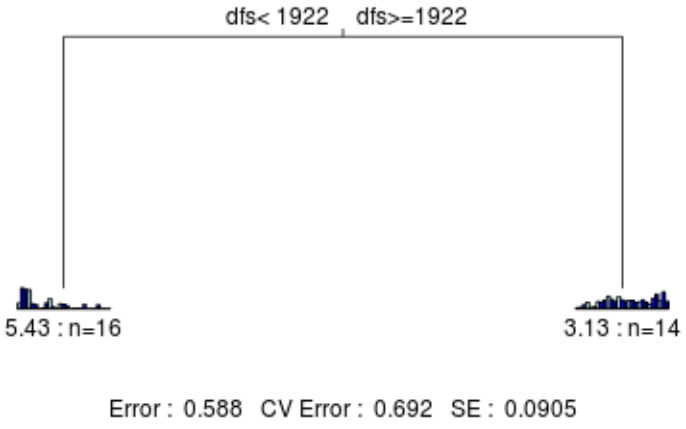
The transformation consists of expressing each fish density as a proportion of the sum of all densities in the analytical unit and taking the square root of the resulting value (Legendre and Gallagher 2001).The square-root portion of the transformation decreases the importance of the most abundant species.

```
library(vegan)
spe.norm <- as.matrix(decostand(spe, "hellinger"))
```

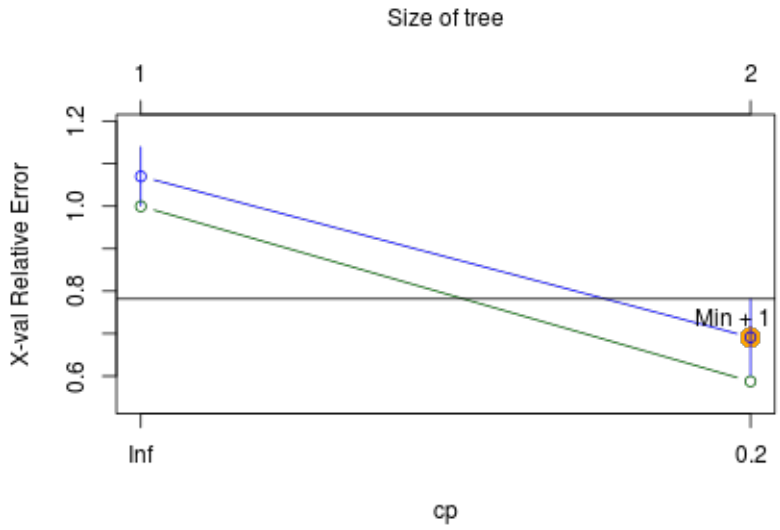
Multivariate Regression Tree

```
library(mvpart)
spe.ch.mvpart<-mvpart(spe.norm ~ ., env,
```

```
                    xv="1se",
                    xval=nrow(spe),
                    xvmult=100, which=4)
```




```
summary(spe.ch.mvpart)
printcp(spe.ch.mvpart)
plotcp(spe.ch.mvpart)
```



Or we can use "pick"

```
spe.ch.mvpart<-mvpart(data.matrix(spe.norm) ~., env,
                    xv="pick", xval=nrow(spe),
```

```
                    xvmult=100, which=4)
summary(spe.ch.mvpart)
printcp(spe.ch.mvpart)
```

- `xv` = Selection of tree by cross-validation:
  - `"1se"` - gives best tree within one SE of the overall best,
  - `"min"` - the best tree
  - `"pick"` - pick the tree size interactively,
  - `"none"` - no cross-validation.
- `xval` = Number of cross-validations or vector defining cross-validation groups (here we use as many rows there are in the dataset (*leave-one-out cross-validation*) because it is a small dataset)
- `xvmult` = Number of multiple cross-validations.
- `which` = Which split labels and where to plot them: 1=centered, 2 = left, 3 = right and 4 = both.

(Modified R Code from *Numerical Ecology with R*, Borcard et al. 2012)