TP3 - Arbres de décision

Thomas Laurent

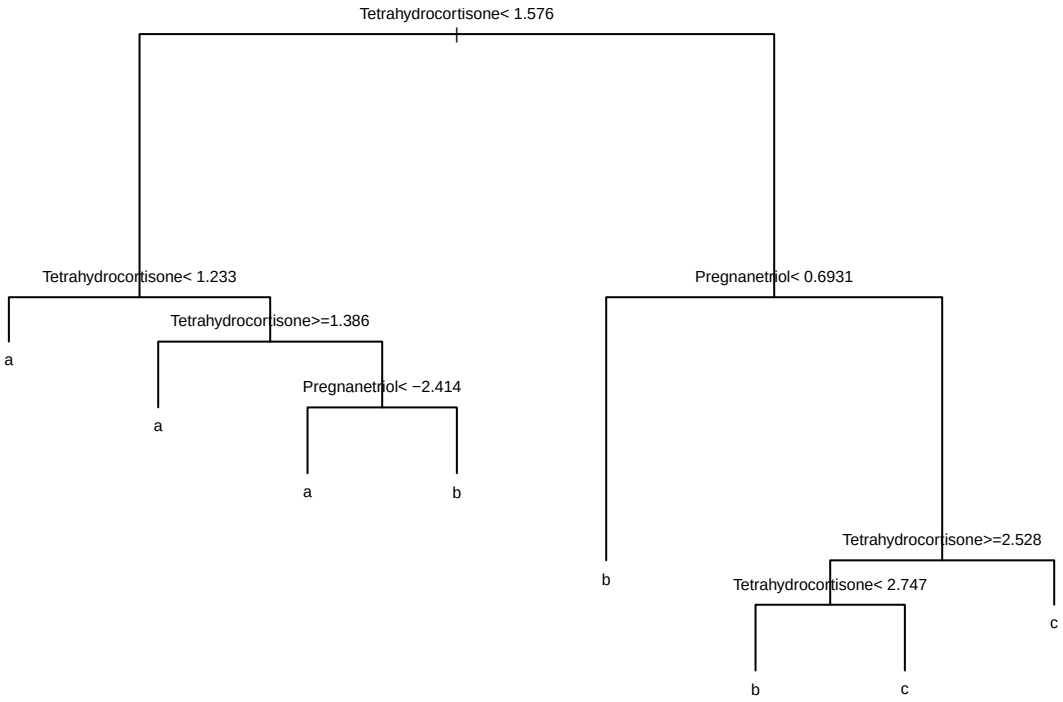
11/02/2018

Donnéees du syndrome de Cushing

On construit l'arbre de décision puis on procède à un élagage de l'arbre sur la base des valeurs de xerror et xstd.

```
###Chargement des donnees###
rm(list = ls(all = TRUE))
op=par(cex=0.5)

library(rpart)
library(MASS)
#Cushings data
data(Cushings)
cush <- Cushings[Cushings$Type!="u",]
cush$Type<-factor(cush$Type)
cush[,1:2] <- log(cush[,1:2])
```

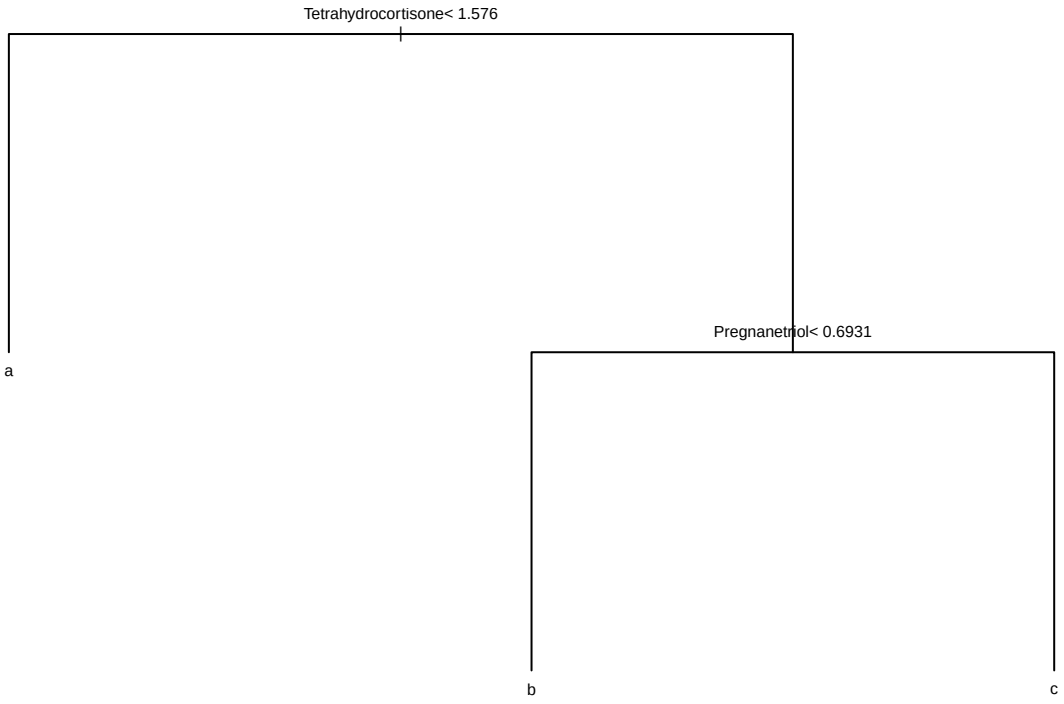
```
###Arbre de decision###
cush.trer<-rpart(Type~Tetrahydrocortisone+Pregnanetriol,cush,
cp=-Inf,control = rpart.control(minsplit=2,xval=10),method="class")
par(cex=0.5)
plot(cush.trer)
text(cush.trer)
```



```
cush.cpt<-printcp(cush.trer)
```

```
##
## Classification tree:
## rpart(formula = Type ~ Tetrahydrocortisone + Pregnanetriol, data = cush,
##     method = "class", control = rpart.control(minsplit = 2, xval = 10),
##     cp = -Inf)
##
## Variables actually used in tree construction:
## [1] Pregnanetriol       Tetrahydrocortisone
##
## Root node error: 11/21 = 0.52381
##
## n= 21
##
##         CP nsplit rel error  xerror    xstd
## 1 0.363636      0  1.000000 1.00000 0.20806
## 2 0.060606      2  0.272727 0.90909 0.20806
## 3 0.045455      5  0.090909 1.00000 0.20806
## 4 0.010000      7  0.000000 0.81818 0.20616
```

```
cush.pt<-prune(cush.trer,cp=0.07)
par(cex=0.5)
plot(cush.pt)
text(cush.pt)
```



Exercice (question 1)

$p(c|x)$ est déterminé par la proportion de chaque classe dans la feuille considérée. A la feuille Tetrahydrocortisone>= 1.575727, Pregnanetriol>= 0.6931472, le taux d'observations dans la classe c est égal à 5/6, ce qui

correspond aux résultats obtenus avec la fonction predict.

```
#Probabilites de classement pour tetrahydrocortisone=1.6 et pregnanetriol=0.7
cush.pt
```

```
## n= 21
##
## node), split, n, loss, yval, (yprob)
##       * denotes terminal node
##
## 1) root 21 11 b (0.2857143 0.4761905 0.2380952)
##   2) Tetrahydrocortisone< 1.575727 8  2 a (0.7500000 0.2500000 0.0000000) *
##   3) Tetrahydrocortisone>=1.575727 13  5 b (0.0000000 0.6153846 0.3846154)
##     6) Pregnanetriol< 0.6931472 7  0 b (0.0000000 1.0000000 0.0000000) *
##     7) Pregnanetriol>=0.6931472 6  1 c (0.0000000 0.1666667 0.8333333) *
```

```
summary(cush.pt)
```

```
## Call:
## rpart(formula = Type ~ Tetrahydrocortisone + Pregnanetriol, data = cush,
##     method = "class", control = rpart.control(minsplit = 2, xval = 10),
##     cp = -Inf)
##   n= 21
##
##          CP nsplit rel error    xerror      xstd
## 1 0.3636364      0 1.0000000 1.0000000 0.2080626
## 2 0.0700000      2 0.2727273 0.9090909 0.2080626
##
## Variable importance
## Tetrahydrocortisone       Pregnanetriol
##                  57                  43
##
## Node number 1: 21 observations,    complexity param=0.3636364
##   predicted class=b  expected loss=0.5238095  P(node) =1
##     class counts:     6    10     5
##    probabilities: 0.286 0.476 0.238
##   left son=2 (8 obs) right son=3 (13 obs)
##   Primary splits:
##       Tetrahydrocortisone < 1.575727  to the left,  improve=4.179487, (0 missing)
##       Pregnanetriol       < 0.6931472 to the left,  improve=3.761905, (0 missing)
##   Surrogate splits:
##       Pregnanetriol < -1.262864 to the left,  agree=0.762, adj=0.375, (0 split)
##
## Node number 2: 8 observations
##   predicted class=a  expected loss=0.25  P(node) =0.3809524
##     class counts:     6     2     0
##    probabilities: 0.750 0.250 0.000
##
## Node number 3: 13 observations,    complexity param=0.3636364
##   predicted class=b  expected loss=0.3846154  P(node) =0.6190476
##     class counts:     0     8     5
##    probabilities: 0.000 0.615 0.385
##   left son=6 (7 obs) right son=7 (6 obs)
##   Primary splits:
##       Pregnanetriol       < 0.6931472 to the left,  improve=4.487179, (0 missing)
```

```
##       Tetrahydrocortisone < 2.213739  to the left,  improve=3.296703, (0 missing)
##   Surrogate splits:
##       Tetrahydrocortisone < 2.213739  to the left,  agree=0.923, adj=0.833, (0 split)
##
## Node number 6: 7 observations
##   predicted class=b  expected loss=0  P(node) =0.3333333
##     class counts:     0     7     0
##    probabilities: 0.000 1.000 0.000
##
## Node number 7: 6 observations
##   predicted class=c  expected loss=0.1666667  P(node) =0.2857143
##     class counts:     0     1     5
##    probabilities: 0.000 0.167 0.833
```

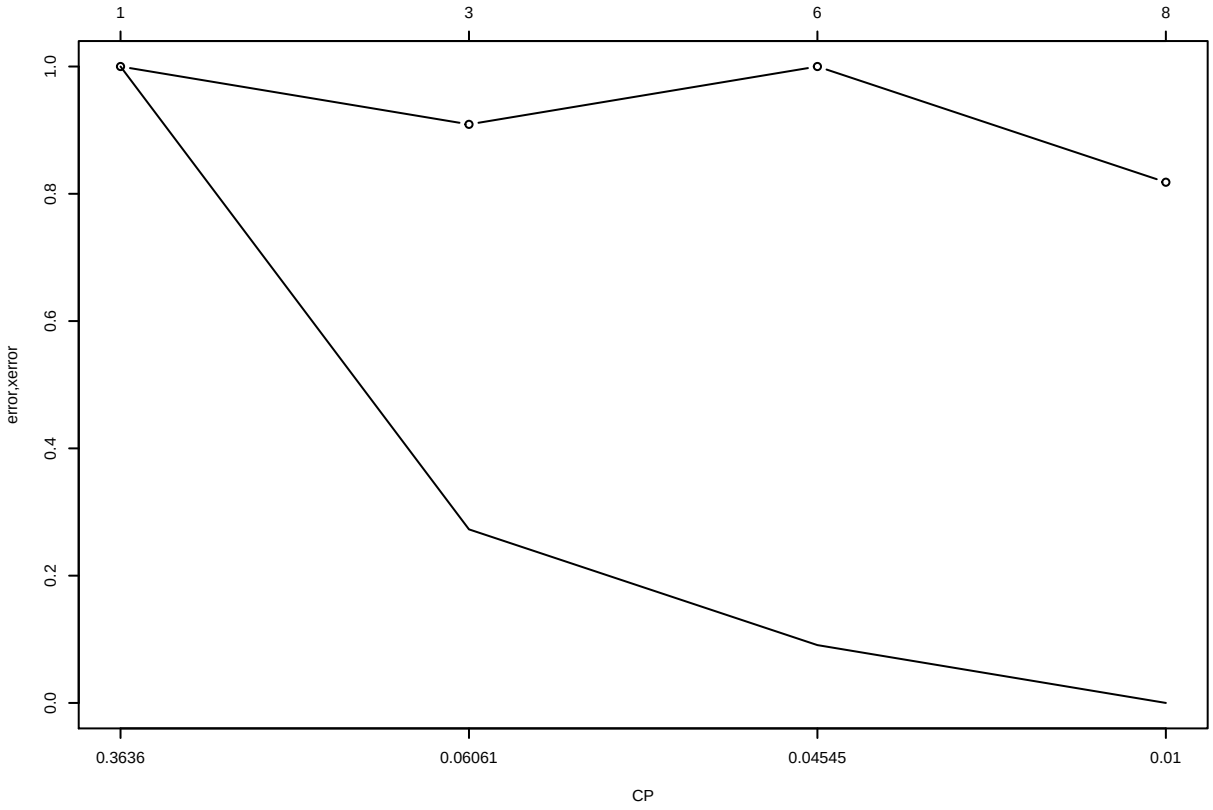
```
predict(cush.pt,data.frame(Tetrahydrocortisone=1.6,Pregnanetriol=0.7))
```

```
##   a         b         c
## 1 0 0.1666667 0.8333333
```

Exercice (question 2)

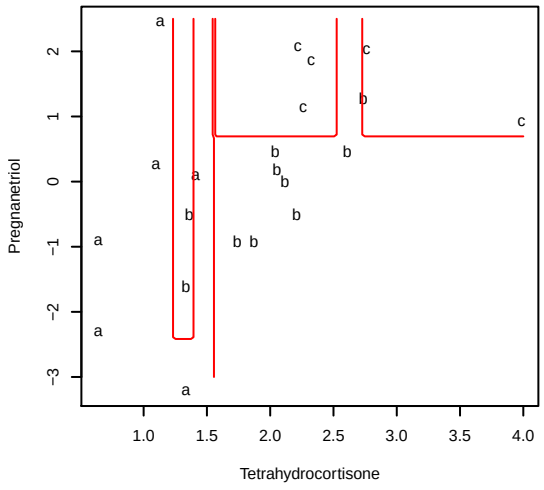
Pour la validation croisée, les erreurs moyennes sont calculées en faisant une moyenne géométrique à une valeur du coefficient de pénalisation fixée.

```
par(xaxt="n")
par(cex=0.5)
plot(1:nrow(cush.cpt),cush.cpt[,3],type='l',xlab="CP",ylab="error,xerror")
par(xaxt="s")
points(1:nrow(cush.cpt),cush.cpt[,4],type='b')
axis(1, at = 1:nrow(cush.cpt), labels = formatC(cush.cpt[,1], format="fg"))
axis(3, at = 1:nrow(cush.cpt), labels = formatC(cush.cpt[,2]+1, format="fg"))
```

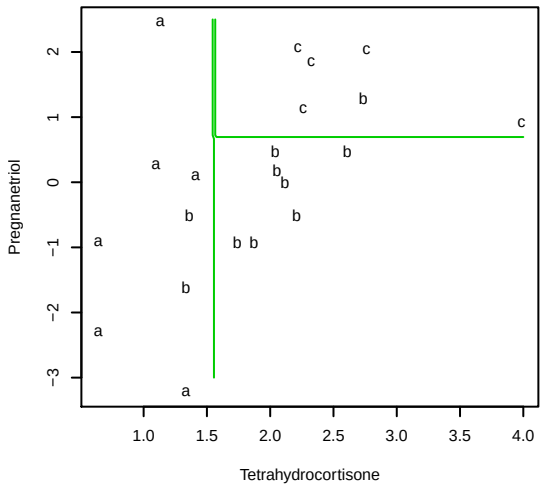


On trace ensuite les frontières pour l'arbre complet et l'arbre élagué.

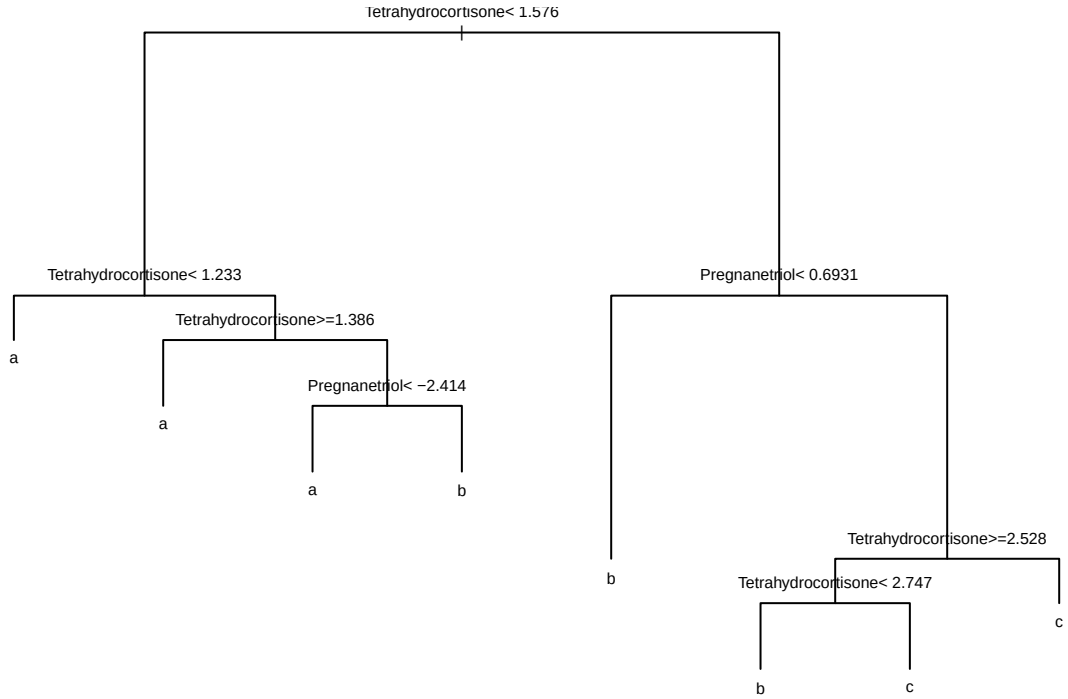
```
m<-100
x<-seq(0,4,length.out=m)
y<-seq(-3,2.5,length.out=m)
z<-data.frame(expand.grid(Tetrahydrocortisone=x,Pregnanetriol=y))

#Frontiere pour l'arbre complet
par(cex=0.5)
plot(cush[,1:2],pch=as.character(cush$Type))
cush.trerb<-predict(cush.trer,z)
contour(x,y,matrix(max.col(cush.trerb),m,m),levels=c(1.5,2.5),
add=T,d=F,lty=1,col=2)
```



```
#Frontiere pour l'arbre elague
par(cex=0.5)
plot(cush[,1:2],pch=as.character(cush$Type))
cush.ptb<-predict(cush.pt,z)
contour(x,y,matrix(max.col(cush.ptb),m,m),levels=c(1.5,2.5),
add=T,d=F,lty=1,col=3)
```




```
#Construction de l'arbre en utilisant le gain d'information
cush.tre<-rpart(Type~Tetrahydrocortisone+Pregnanetriol,cush,
cp=-Inf,control = rpart.control(minsplit=2,xval=10),
method="class",parms = list(split="information"))
par(cex=0.5)
plot(cush.tre)
text(cush.tre)
```



```
printcp(cush.tre)
```

```
##
## Classification tree:
## rpart(formula = Type ~ Tetrahydrocortisone + Pregnanetriol, data = cush,
##     method = "class", parms = list(split = "information"), control = rpart.control(minsplit = 2,
##         xval = 10), cp = -Inf)
##
## Variables actually used in tree construction:
## [1] Pregnanetriol       Tetrahydrocortisone
##
## Root node error: 11/21 = 0.52381
##
## n= 21
##
##         CP nsplit rel error  xerror    xstd
## 1 0.363636      0  1.000000 1.00000 0.20806
## 2 0.060606      2  0.272727 0.81818 0.20616
## 3 0.045455      5  0.090909 0.81818 0.20616
## 4 0.010000      7  0.000000 0.72727 0.20231
```

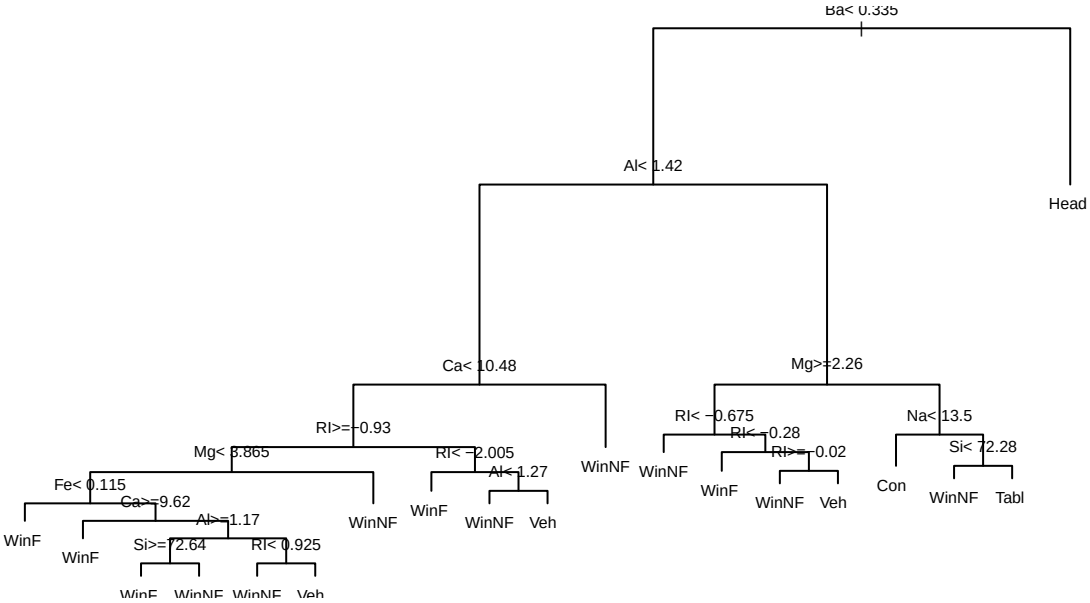
On remarque qu'avec le critère d'entropie, on obtient les mêmes résultats.

Jeux de données Verres Forensic

```
#Import des données
data(fgl)
names(fgl)[10]<-"Type"
set.seed(123)
fgl.trer<-rpart(Type~.,fgl,
cp=-Inf,control = rpart.control(minsplit=2,xval=10),method="class")
```

On trace l'arbre complet.

```
#Affichage de l'arbre
par(cex=0.5)
plot(fgl.trer)
text(fgl.trer)
```

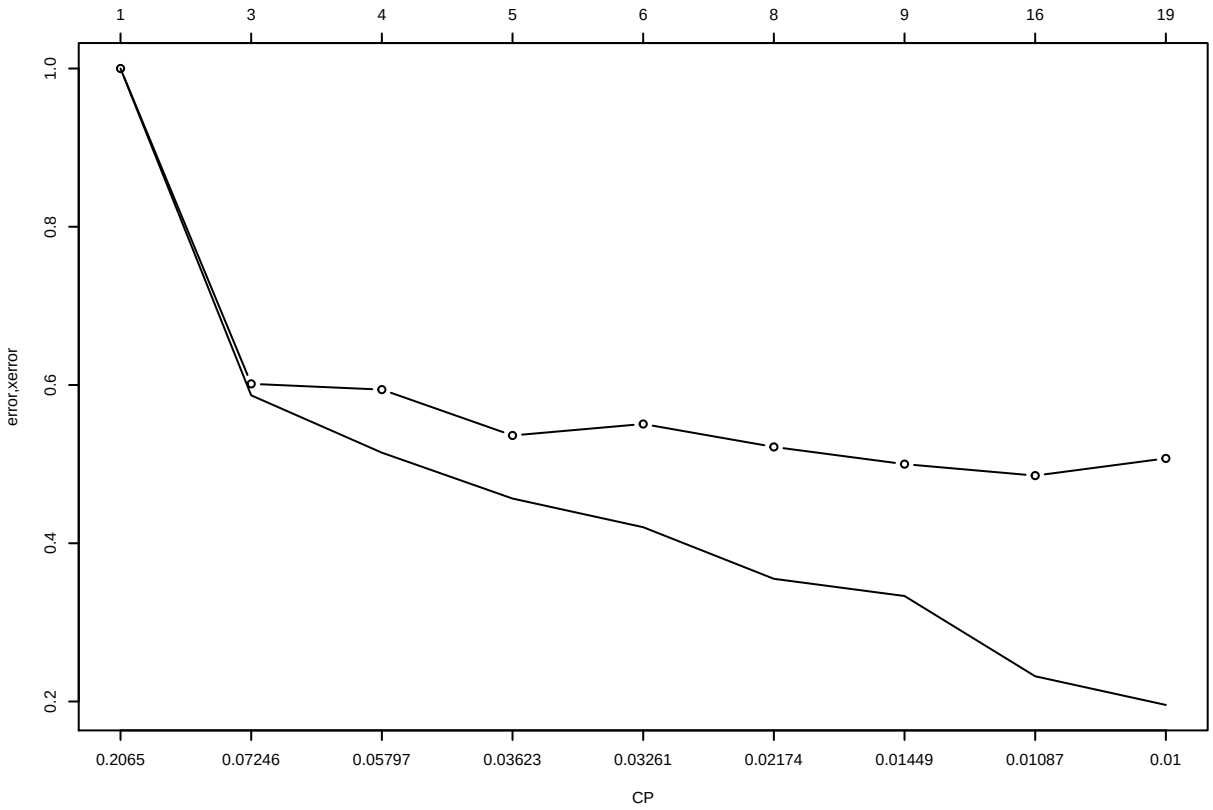



```
fgl.xv<-printcp(fgl.trer)
```

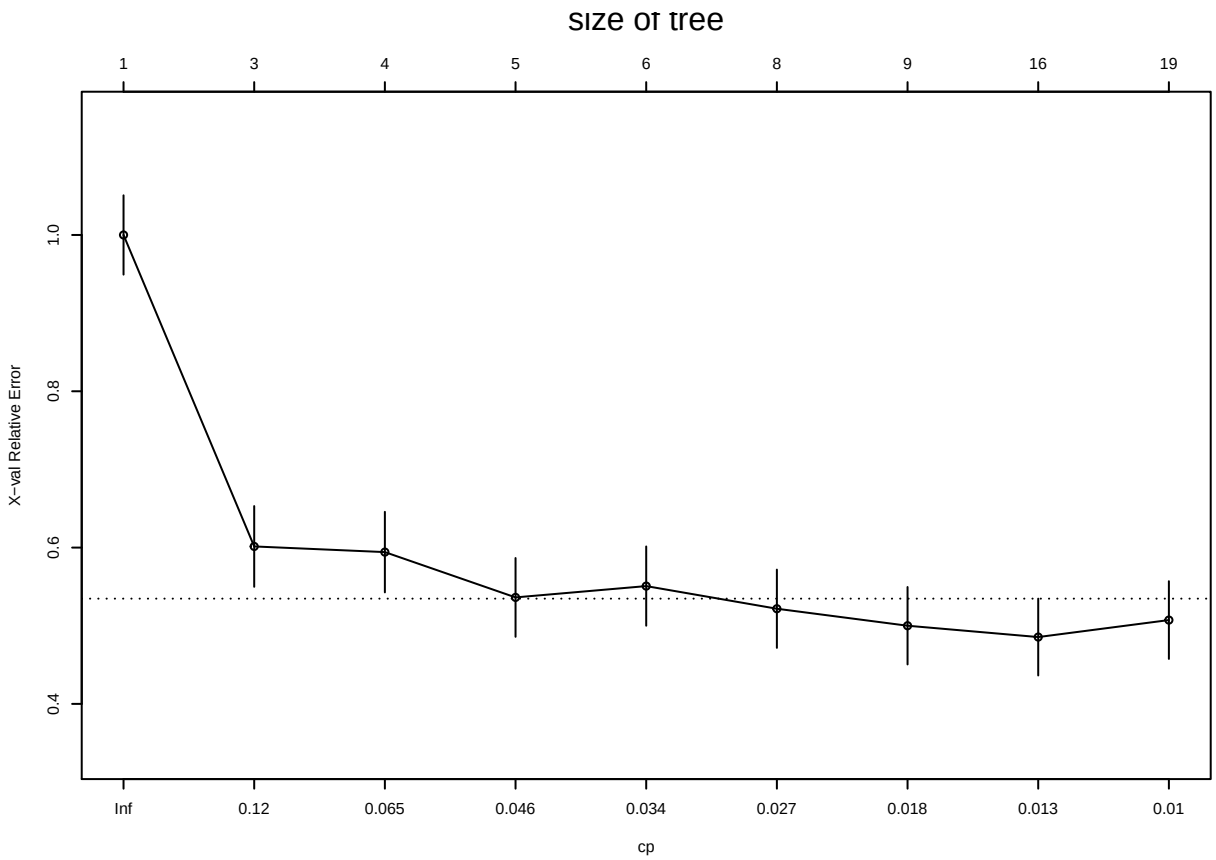
```
##
## Classification tree:
## rpart(formula = Type ~ ., data = fgl, method = "class", control = rpart.control(minsplit = 2,
##     xval = 10), cp = -Inf)
##
## Variables actually used in tree construction:
## [1] Al Ba Ca Fe Mg Na RI Si
##
## Root node error: 138/214 = 0.64486
##
## n= 214
##
##         CP nsplit rel error  xerror     xstd
## 1 0.206522      0   1.00000 1.00000 0.050729
## 2 0.072464      2   0.58696 0.60145 0.051652
## 3 0.057971      3   0.51449 0.59420 0.051536
## 4 0.036232      4   0.45652 0.53623 0.050419
## 5 0.032609      5   0.42029 0.55072 0.050729
## 6 0.021739      7   0.35507 0.52174 0.050087
## 7 0.014493      8   0.33333 0.50000 0.049548
## 8 0.010870     15   0.23188 0.48551 0.049160
## 9 0.010000     18   0.19565 0.50725 0.049733
```

En traçant les graphes des éboulis en fonction de CP, on choisit le découpage pour CP=0.27 (8 feuilles) car l'erreur moyenne est à un écart type du minimum de xerror.

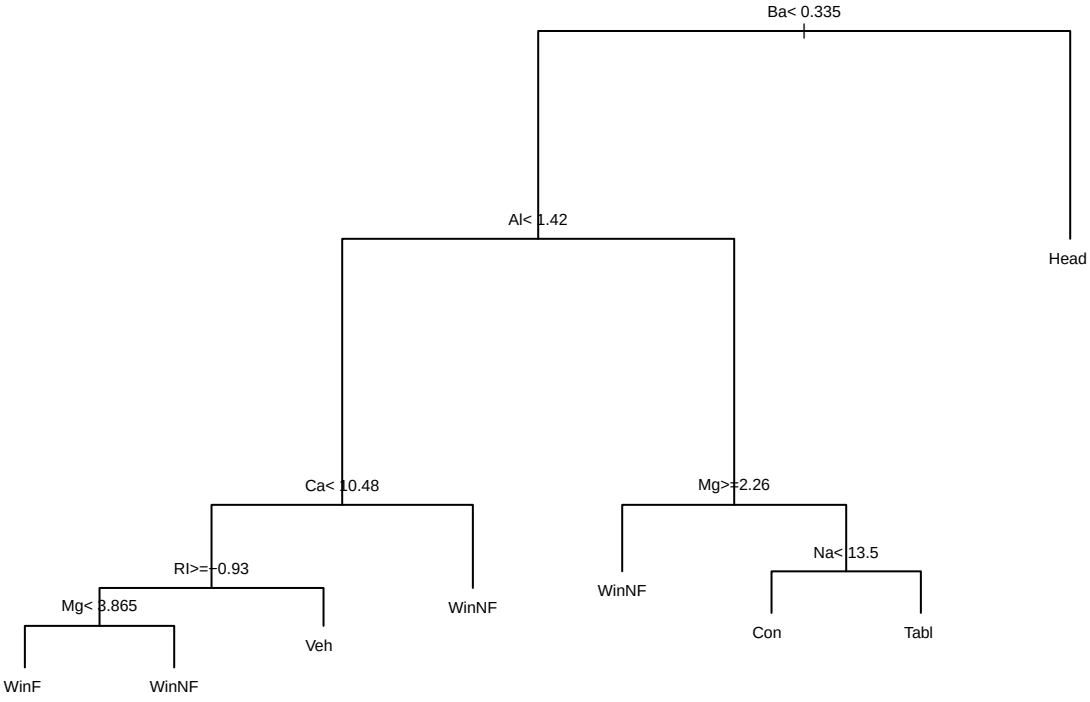
```
#Realisation du graphe des éboulis des coefficients CP
par(xaxt="n")
par(cex=0.5)
plot(1:nrow(fgl.xv),fgl.xv[,3],type='l',xlab="CP",ylab="error,xerror")
par(xaxt="s")
points(1:nrow(fgl.xv),fgl.xv[,4],type='b')
axis(1, at = 1:nrow(fgl.xv), labels = formatC(fgl.xv[,1], format="fg"))
axis(3, at = 1:nrow(fgl.xv), labels = formatC(fgl.xv[,2]+1, format="fg"))
```




```
par(cex=0.5)
plotcp(fgl.trer)
```




```
fgl.pt<-prune(fgl.trer,cp=0.027)
par(cex=0.5)
plot(fgl.pt)
text(fgl.pt)
```



```r
table(fgl$Type,predict(fgl.pt,type="class"))
```

```
##
##         WinF WinNF Veh Con Tabl Head
##   WinF    59     7   3   0    0    1
##   WinNF   11    57   4   1    2    1
##   Veh      5     5   7   0    0    0
##   Con      0     1   0  11    0    1
##   Tabl     1     2   1   0    5    0
##   Head     1     0   1   0    1   26
```

On regarde ensuite le taux de bien classés est supérieur à 75% ce qui semble convenable.