CART

Muyao GUO

2025-10-07

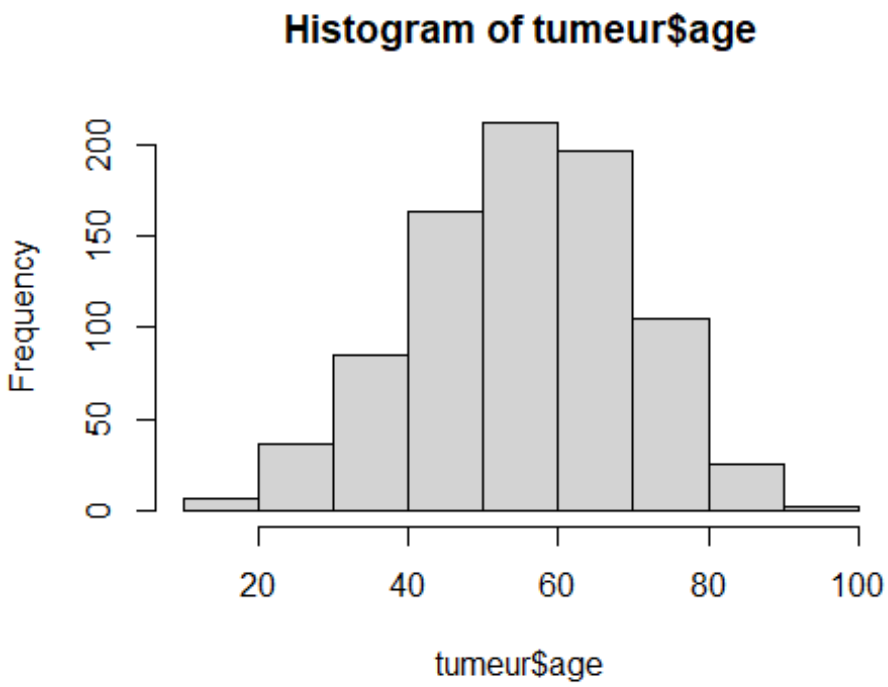
```
rm(list=objects())     # supprime les objets existant en session
graphics.off()         # supprime les graphiques existant en session
 # définit le répertoire en cours, à compléter

library(rpart)
# library(ggplot2)
library(rpart.plot)

#########################################################################
######

tumeur=read.table("mammographic.data",header=T)

str(tumeur)
```

```
## 'data.frame':    830 obs. of  5 variables:
##  $ age    : int  67 58 28 57 76 42 36 60 54 52 ...
##  $ shape  : int  3 4 1 1 1 2 3 2 1 3 ...
##  $ margin : int  5 5 1 5 4 1 1 1 1 4 ...
##  $ density: int  3 3 3 3 3 3 2 2 3 3 ...
##  $ Y      : int  1 1 0 1 1 1 0 0 0 0 ...
```

```
table(tumeur$shape)
```

```
##
##   1   2   3   4
## 190 180  81 379
```

```
hist(tumeur$age)
```




```r
summary(tumeur$density)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   3.000   3.000   2.916   3.000   4.000
```

```r
# 1   2   3   4
# 11  56 755   8

########################################
# les covariables autres que age sont discrètes : facteurs dans R

tumeur$shape=factor(tumeur$shape)
tumeur$margin=factor(tumeur$margin)
tumeur$density=factor(tumeur$density)
tumeur$Y=factor(tumeur$Y)

########################################
```

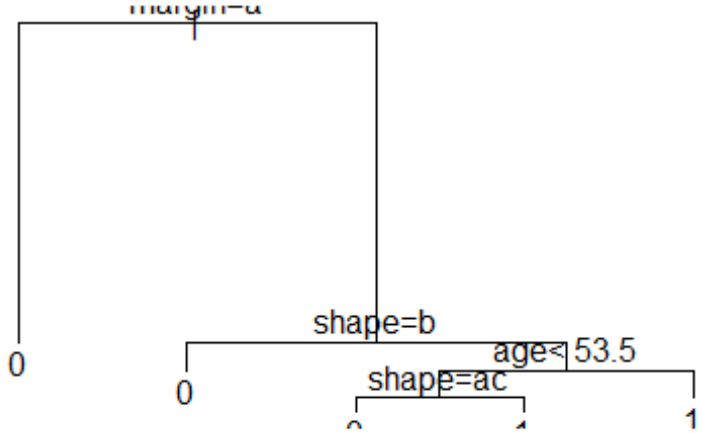
Q1

```r
dim(tumeur) # 4 covariables pour 830 individus
```

```
## [1] 830   5
```

```r
# $Y$ est binaire: modèle de classification
table(tumeur$Y)
```

```
##
##   0   1
## 427 403

# 0   1
# 427 403
# les nb de cas positifs/négatifs sont équilibrés
sum(tumeur$Y==1)/length(tumeur$Y)

## [1] 0.4855422

# 0.4855422 49% de lésions malignes

### Q2

t.rpart=rpart(Y~., data = tumeur)
#  rpart.plot(t.rpart)
# par défaut minsplit=20 et cp=0.01

plot(t.rpart)
text(t.rpart)
```




```
print(t.rpart) # savoir lire cet arbre sans figure

## n= 830
##
## node), split, n, loss, yval, (yprob)
```

```
##       * denotes terminal node
##
##  1) root 830 403 0 (0.5144578 0.4855422)
##    2) margin=1 320  38 0 (0.8812500 0.1187500) *
##    3) margin=2,3,4,5 510 145 1 (0.2843137 0.7156863)
##      6) shape=2 54  17 0 (0.6851852 0.3148148) *
##      7) shape=1,3,4 456 108 1 (0.2368421 0.7631579)
##       14) age< 53.5 118  48 1 (0.4067797 0.5932203)
##         28) shape=1,3 25   8 0 (0.6800000 0.3200000) *
##         29) shape=4 93  31 1 (0.3333333 0.6666667) *
##       15) age>=53.5 338  60 1 (0.1775148 0.8224852) *
```

Il y a 5 feuilles

1) root 830 403 0 (0.5144578 0.4855422)

le noeud racine contient 830 observations dont 403 positifs (lésions malignes)

la classe majoritaire est 0 (tumeurs bénignes) avec une proportion de 51%

noeud droit sous la racine

3) margin=2,3,4,5 510 145 1 (0.2843137 0.7156863)

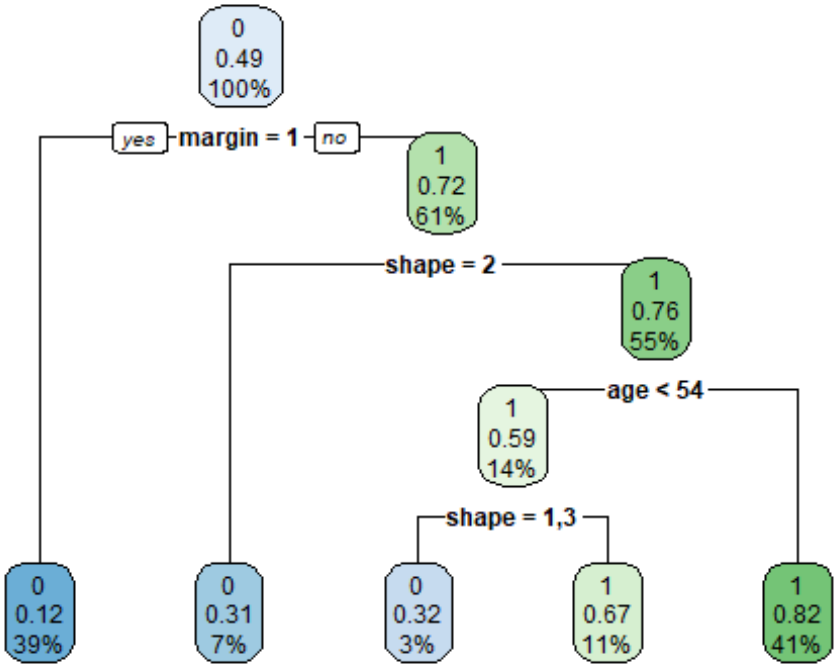
ce noeud contient 510 observations dont 145 positifs;

la classe majoritaire est 1 (tumeurs malignes) avec une proportion de 72%

La valeur de CP (par défaut 0.1) pénalise des arbres plus profonds et fournit ici un arbre à 5 feuilles; sur la branche gauche, le gain d'impureté calculé n'est pas suffisant pour créer un split (si l'on diminue minsplit, cela ne modifie pas l'arbre)

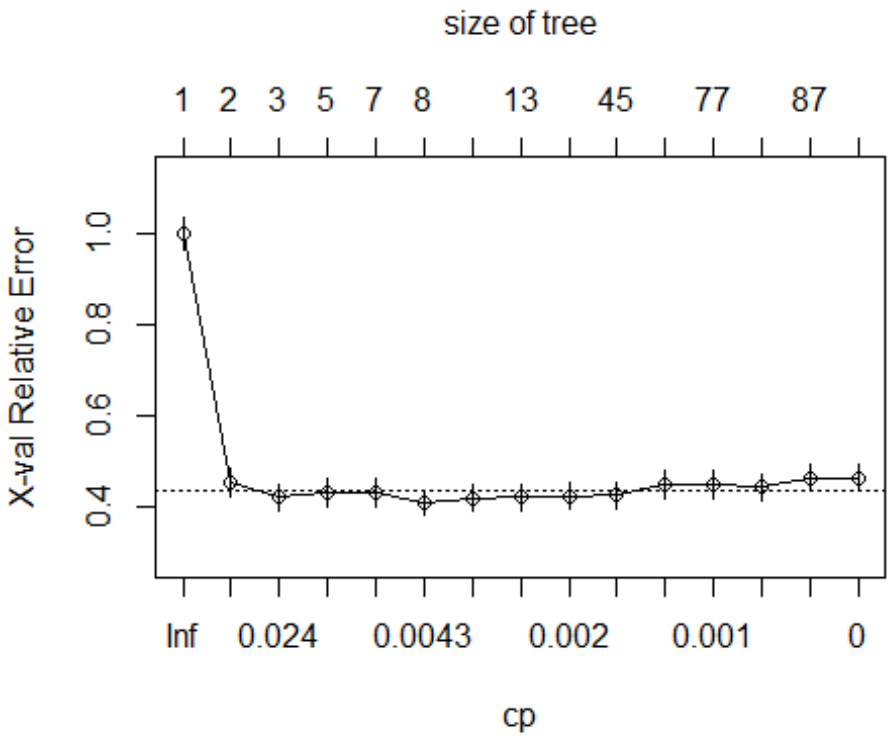
Tracé amélioré

```
rpart.plot(t.rpart)
```



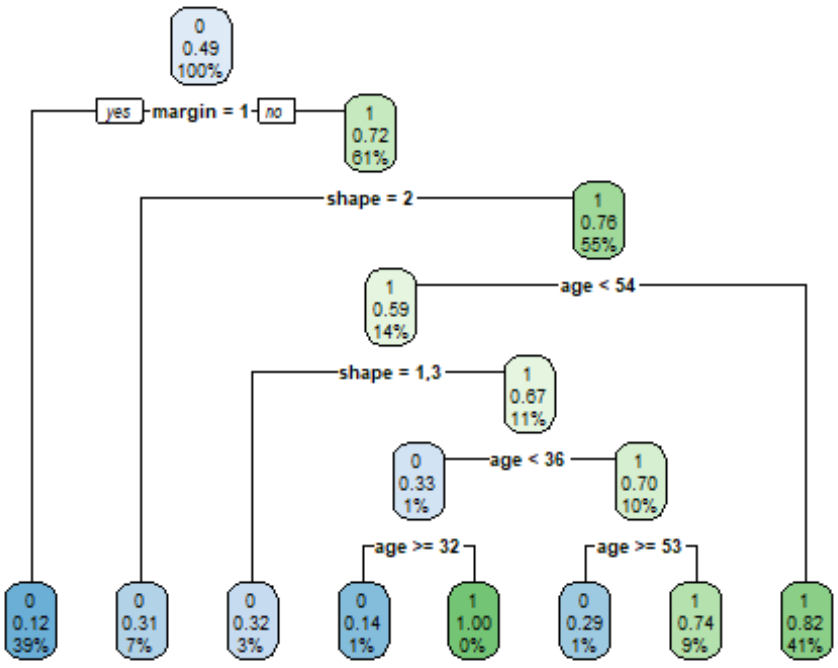

```
### Q3 arbre CART
set.seed(12345)

# On commence par calculer un arbre maximal
tmax <- rpart(Y~., data = tumeur,
              control = rpart.control(cp = 0,minsplit=3))

# On affiche la table qui fournit les valeurs du paramètre de complexit
é CP
# associées à une suite de sous-arbres
# compris entre l'arbre racine et l'arbre maximal de l'étape précédente

printcp(tmax)

##
## Classification tree:
## rpart(formula = Y ~ ., data = tumeur, control = rpart.control(cp = 0
,
##     minsplit = 3))
##
## Variables actually used in tree construction:
## [1] age     density margin  shape
##
## Root node error: 403/830 = 0.48554
##
## n= 830
```

```
##
##            CP nsplit rel error  xerror     xstd
## 1  0.54590571      0   1.00000 1.00000 0.035729
## 2  0.04962779      1   0.45409 0.45409 0.029637
## 3  0.01116625      2   0.40447 0.42184 0.028850
## 4  0.00744417      4   0.38213 0.43176 0.029099
## 5  0.00496278      6   0.36725 0.43176 0.029099
## 6  0.00372208      7   0.36228 0.40943 0.028530
## 7  0.00330852      9   0.35484 0.41935 0.028787
## 8  0.00248139     12   0.34491 0.42184 0.028850
## 9  0.00165426     30   0.30025 0.42432 0.028913
## 10 0.00148883     44   0.27295 0.42680 0.028976
## 11 0.00124069     49   0.26551 0.44913 0.029520
## 12 0.00082713     76   0.22829 0.44913 0.029520
## 13 0.00062035     82   0.22333 0.44417 0.029402
## 14 0.00035448     86   0.22084 0.46402 0.029866
## 15 0.00000000     93   0.21836 0.46402 0.029866

plotcp(tmax)
```



la valeur de CP optimale correspond au sous-arbre qui réalise le meilleur compromis biais/variance pour éviter le sur-apprentissage.
on choisit la valeur qui minimise l'erreur de classification (ici taux de mal classés) estimé par validation croisée.
cette estimation est fournie en colonne 4 "xerror" (dans une version renormalisée)

```r
# l'arbre cart est l'arbre élagué à la valeur de cp qui rend le taux de
 mal classés minimal
which.min(tmax$cptable[, 4])

## 6
## 6

# 6 --> ligne 6 correspondant à nsplit=7 soit 7 noeuds internes ou 8 fe
uilles
tmax$cptable[which.min(tmax$cptable[, 4]), 1]

## [1] 0.003722084

# cp= 0.003722084 pour ce run de validation croisée

# on élague l'arbre maximal pour afficher l'arbre CART
tcart <- prune(tmax, cp = tmax$cptable[which.min(tmax$cptable[, 4]), 1]
)
rpart.plot(tcart)
```



Q4 prédiction de l'arbre CART

```r
newdata1 <- data.frame(age=50,shape="3",margin="2",density="1")

predict(tcart, newdata = newdata1,type="class")
```

```
## 1
## 0
## Levels: 0 1

# 0 la prédiction est une lésion bénigne
# savoir retrouver cette prédiction dans l'arbre

prob = predict(tcart, newdata = newdata1)
#      0    1
# 1 0.68 0.32  probabilité d'une tumeur maligne estimée à 32%
```