ACT6100	Analyse de données
H2019	Solutions 6

1. (a) Pour l'index Gini, on a

$$\begin{aligned}
G_R &= \sum_{g\in\mathcal{G}} \widehat{p}_{Rg}\left(1-\widehat{p}_{Rg}\right) \\
&= \left(\frac{2}{6}\right)\left(1-\frac{2}{6}\right)+\left(\frac{3}{6}\right)\left(1-\frac{3}{6}\right)+\left(\frac{1}{6}\right)\left(1-\frac{1}{6}\right) \\
&\approx 0.6111 \\
G_{R_1} &= \left(\frac{2}{4}\right)\left(1-\frac{2}{4}\right)+\left(\frac{1}{4}\right)\left(1-\frac{1}{4}\right)+\left(\frac{1}{4}\right)\left(1-\frac{1}{4}\right) \\
&\approx 0.625 \\
G_{R_2} &= 0 \\
G_{AGG} &= \left(\frac{4}{6}\right)(0.6111)+\left(\frac{2}{6}\right)(0) \\
&\approx 0.4167.
\end{aligned}$$

Pour l'entropie, on a

$$\begin{aligned}
D_R &= -\sum_{g\in\mathcal{G}} \widehat{p}_{Rg}\ln\left(\widehat{p}_{Rg}\right)\mathbb{I}_{\{\widehat{p}_{Rg}\neq 0\}} \\
&= -\left(\left(\frac{2}{6}\right)\ln\left(\frac{2}{6}\right)(1)+\left(\frac{3}{6}\right)\ln\left(\frac{3}{6}\right)(1)+\left(\frac{1}{6}\right)\ln\left(\frac{1}{6}\right)(1)\right) \\
&\approx 1.0114 \\
G_{R_1} &= -\left(\left(\frac{2}{4}\right)\ln\left(\frac{2}{4}\right)(1)+\left(\frac{1}{4}\right)\ln\left(\frac{1}{4}\right)(1)+\left(\frac{1}{4}\right)\ln\left(\frac{1}{4}\right)\right) \\
&\approx 1.0397 \\
G_{R_2} &= 0 \\
G_{AGG} &= \left(\frac{4}{6}\right)(1.0114)+\left(\frac{2}{6}\right)(0) \\
&\approx 0.6931.
\end{aligned}$$

(b) L'index Gini et l'entropie étant définis par des sommes, on va vérifier que les termes des sommes sont numériquement semblables lorsque p est près de 0 ou de 1 (j'omets volontairement les chapeaux et les indices pour ne pas alourdir inutilement la notation). On a

$$\begin{aligned}
G &= \sum p\left(1-p\right) \\
D &= \sum p\left(-\ln\left(p\right)\right)\mathbb{I}_{\{p\neq 0\}} \\
\lim_{p\to 0^+} p(1-p) &= 0 \\
\lim_{p\to 0^+} p\ln(p) &= \lim_{p\to 0^+}\frac{\ln(p)}{1/p} \\
&= \lim_{p\to 0^+}\frac{1/p}{-1/p^2} \\
&= \lim_{p\to 0^+} -p = 0.
\end{aligned}$$

Si on fait le développement de Taylor d'ordre 1 de la fonction $-\ln(p)$ autour du point $a=1$, on obtient

$$\begin{aligned}
-\ln(p) &= -\ln(1)+\frac{-1/1}{1!}(p-1)+\ldots \\
&\approx -(p-1).
\end{aligned}$$

Ainsi, autour du point $a = 1$, on a $-\ln(p) \approx (1-p)$ et

$$-p\ln(p)\mathbb{I}_{\{p\neq 0\}} \approx p(1-p).$$

2. (a) Puisqu'il y a deux variables explicatives dans le modèle, l'espace est $\mathbb{R}^2$. La Figure 1 présente les éléments demandés.

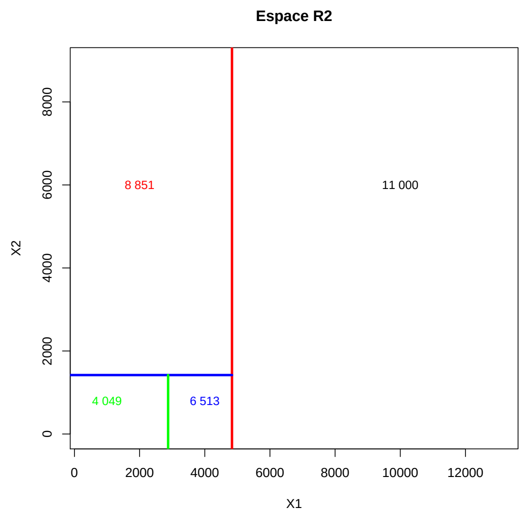

Figure 1 – Espace des variables explicatives pour l'arbre de décision 1.

(b) Puisqu'il y a une variable explicative dans le modèle, l'espace est $\mathbb{R}$. La Figure 2 présente les éléments demandés.

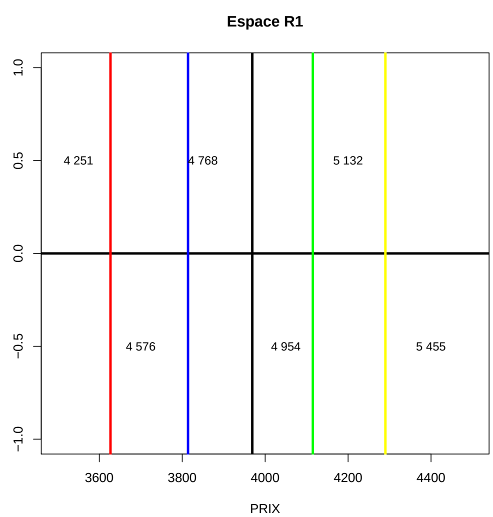

Figure 2 – Espace de la variable explicative pour l'arbre de décision 2.

(c) Puisqu'il y a deux variables explicatives dans le modèle, l'espace est $\mathbb{R}^2$. Comme il s'agit d'un arbre de classification, la prédiction faite dans chacune des classes est le mode. La Figure 3 présente les éléments demandés.

3. **Partie I**

(a) Le code est présenté à la Figure 4. Le graphique des erreurs relatives est illustré à la Figure 5 et l'arbre final est illustré à la Figure 6.

(b) Le code est présenté à la Figure 7.

(c) Le code est présenté à la Figure 8. L'arbre final est illustré à la Figure 9.

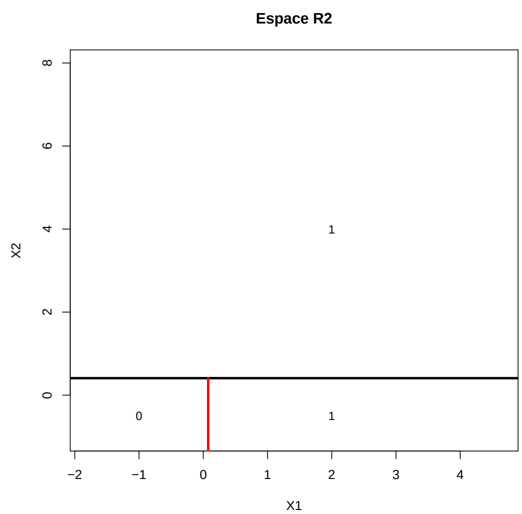

FIGURE 3 – Espace de la variable explicative pour l'arbre de décision 3.

```
library(CASdatasets)
data(freMPL2)

### Construction de l'arbre complet
arbreA <- rpart(ClaimInd ~ Exposure + DrivAge + BonusMalus, control =
                rpart.control(cp = 0), data = freMPL2)

### Graphique permettant de déterminer la valeur optimale du par. de complexité
CP <- arbreA$cptable[which.min(arbreA$cptable[,4]),1]
plotcp(arbreA)

### Élagage de l'arbre
arbreAA <- prune(arbreA,cp = CP)

### Graphique de l'arbre final
prp(arbreAA, extra = 1)
```

FIGURE 4 – Code informatique.

Régions	**Exposure**	**BonusMalus**	$\widehat{p}$
R_1	$(-\infty, 0.34)$	$(-\infty, \infty)$	0.02
R_2	$[0.34, 0.77)$	$(-\infty, 75)$	0.043
R_3	$[0.34, 0.77)$	$[75, \infty)$	0.069
R_4	$[0.77, \infty)$	$(-\infty, 77)$	0.07
R_5	$[0.77, \infty)$	$[77, \infty)$	0.1

TABLE 1 – Régions.

Partie II

(a) À partir de la Figure 6, on peut construire la Table 1. On définit une variable indicatrice I avec $I = 1$ si un sinistre survient et $I = 0$ si aucun sinistre ne survient. On a $I \sim \text{Bernoulli}(p)$ où,

$$\begin{aligned} p &= \sum_{j=1}^{5} m_j \mathbb{I}_{\{\mathbf{X} \in R_j\}} \\ &= (0.02)\mathbb{I}_{\{\mathbf{X} \in R_1\}} + (0.043)\mathbb{I}_{\{\mathbf{X} \in R_2\}} + (0.069)\mathbb{I}_{\{\mathbf{X} \in R_3\}} + (0.07)\mathbb{I}_{\{\mathbf{X} \in R_4\}} + (0.1)\mathbb{I}_{\{\mathbf{X} \in R_5\}}. \end{aligned}$$

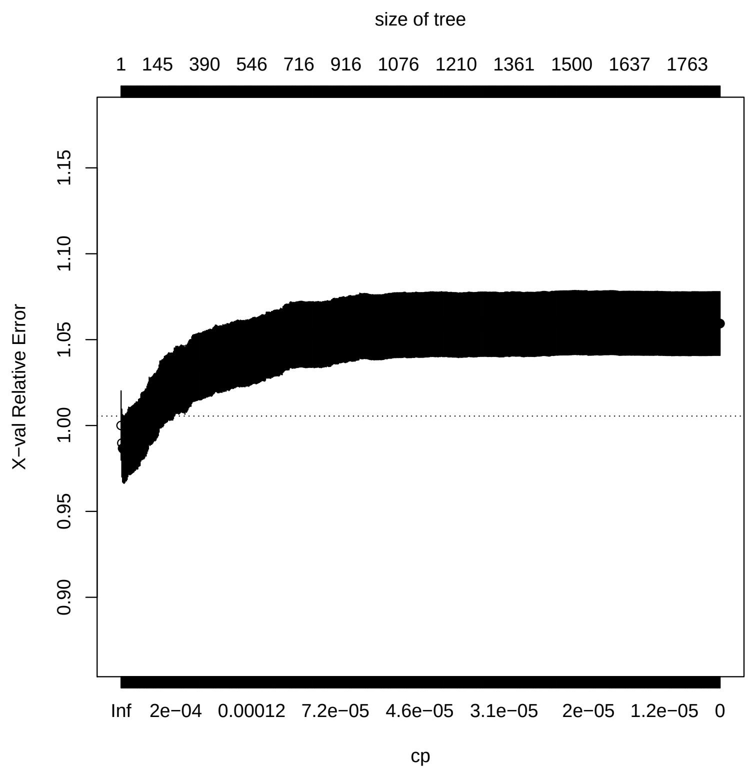

FIGURE 5 – Erreurs relatives en fonction du paramètre de complexité.

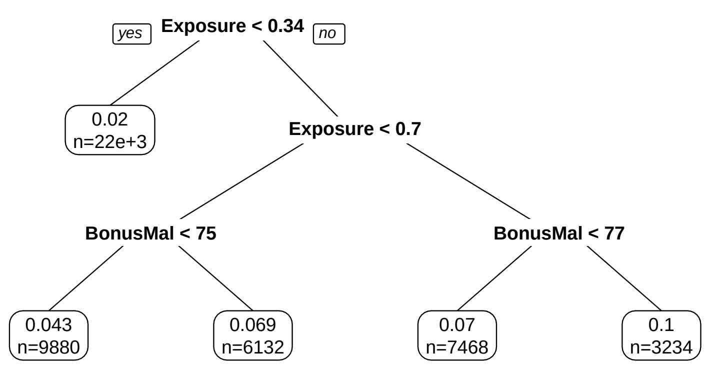

FIGURE 6 – Arbre final.

(b) À partir de la Figure 9, on peut construire la Table 2. On définit une variable Y représentant

```
freMPL2 <- freMPL2[freMPL2$ClaimAmount > 0,]

set.seed(101)
n <- length(freMPL2$Exposure)
indice <- sample(1:n, round(0.2*n,0), replace = FALSE)

dataTRAIN <- freMPL2[-indice, ]
dataVALID <- freMPL2[indice, ]
```

FIGURE 7 – Code informatique.

```
FUN <- function(x)
{
  AAA <- rpart(ClaimAmount ~ DrivAge + BonusMalus,
               data = dataTRAIN, control=rpart.control(cp = x))
  sum((predict(AAA, newdata = dataVALID) - dataVALID$ClaimAmount)^2)
}
vMSE <- sapply((0:5000)/100000, function(x) FUN(x))

CP <- mean(((0:5000)/100000)[(vMSE == min(vMSE))])
CP
0.01409

arbre <- rpart(ClaimAmount ~ DrivAge + BonusMalus, data = dataTRAIN,
               control=rpart.control(cp = CP))

prp(arbre, extra = 1)
```

FIGURE 8 – Code informatique.

Régions	**DriverAge**	**BonusMalus**	$\widehat{Y}$
R_1	$(-\infty, \infty)$	$(-\infty, 101)$	1 822
R_2	$[39, \infty)$	$[101, 111)$	19 000
R_3	$(-\infty, 39)$	$[101, 111)$	1 467
R_4	$(-\infty, \infty)$	$[111, \infty)$	1 456

TABLE 2 – Régions.

la sévérité d'un sinistre. On a

$$\begin{aligned}\widehat{Y} &= \sum_{j=1}^{4} m_j \mathbb{I}_{\{\mathbf{X} \in R_j\}} \\ &= (1\,822)\mathbb{I}_{\{\mathbf{X} \in R_1\}} + (19\,000)\mathbb{I}_{\{\mathbf{X} \in R_2\}} + (1\,467)\mathbb{I}_{\{\mathbf{X} \in R_3\}} + (1\,456)\mathbb{I}_{\{\mathbf{X} \in R_4\}}.\end{aligned}$$

(c) Le code est présenté à la Figure 10. La fonction de répartition du cout final est illustrée à la Figure 11. Il est à noter que l'approche présentée a un petit défaut : les simulations pour la fréquence et pour la sévérité sont considérées comme indépendantes. Ainsi, il n'est pas certain que les valeurs de la variable **BonusMalus** seront cohérentes. D'autres approches peuvent être imaginées.

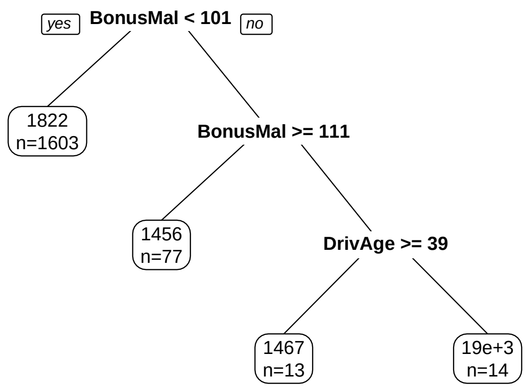

FIGURE 9 – Arbre final.

```
n <- 1000
prop_freq <- c(48295 - 9880 - 6132 - 7468 - 3234, 9880, 6132, 7468, 3234)/48295
p_freq <- c(0.02, 0.043, 0.069, 0.07, 0.1)

prop_sev <- c(1603, 77, 13, 14)/1707
y_sev <- c(1822, 1456, 1467, 19000)

FUN <- function(x)
{
P <- sample(p_freq, size = n, replace = TRUE, prob = prop_freq)
I <- rbinom(n, 1, P)
Y <- sample(y_sev, size = sum(I), replace = TRUE, prob = prop_sev)
S <- sum(Y)
S
}
S <- sapply(1:10000, function(y) FUN(y))
plot(ecdf(S), main = "", xlab = "S", ylab = "F(s)")
```

FIGURE 10 – Code informatique.

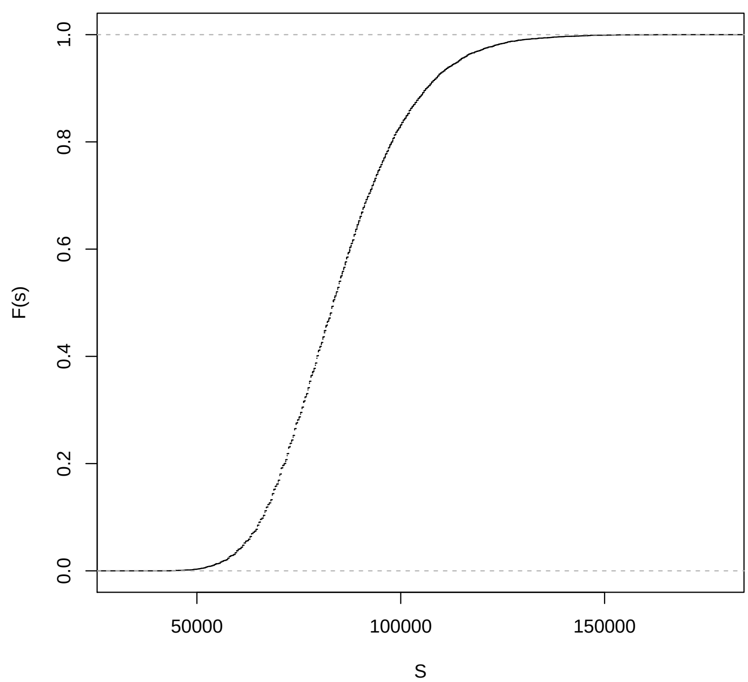

FIGURE 11 – Fonction de répartition empirique du cout total pour un portefeuille de taille 1 000.