HAX711X - Analyse des Données Multidimensionnelles
DM2 Classification automatique

SCAIA Matteo et MARIAC Damien

1er décembre 2024

Table des matières

1 Treillis de Galois

1.1 Création du treillis

Dans cette question, nous allons analyser le treillis de Galois construit à partir des données fournies par le sujet afin de modéliser les relations entre les films et leurs caractéristiques.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Default N...	E_O	E_N	A_O	A_N	L<30	30<L<60	60<L<180	L>180	D_O	D_N	Edu_O	Edu_N	Ir_o	Ir_N	Is_O	Is_N
Série polic...	0	X	X	0	0	0	0	X	X	0	0	X	X	0	0	X
Série hum...	X	0	X	0	0	0	0	X	X	0	0	X	X	0	0	X
Long métr...	X	0	X	0	0	0	X	0	X	0	0	X	0	X	X	0
Court mét...	X	0	0	X	X	0	0	0	X	0	0	X	0	X	X	0
Clip chans...	X	0	X	0	X	0	0	0	X	0	0	X	X	0	X	0
Document...	X	0	X	0	0	X	0	0	X	0	X	0	X	0	0	X
Document...	0	X	X	0	0	X	0	0	0	X	X	0	X	0	X	0
Document...	0	X	X	0	0	X	0	0	X	0	X	0	X	0	X	0
Document...	0	X	X	0	0	X	0	0	X	0	X	0	X	0	X	0
Film de fa...	X	0	X	0	0	0	X	0	X	0	0	X	X	0	X	0
Film horre...	0	X	X	0	0	0	X	0	X	0	0	X	X	0	X	0
Film dram...	0	X	X	0	0	0	X	0	X	0	0	X	X	0	0	X
Film polici...	0	X	X	0	0	0	X	0	X	0	0	X	X	0	0	X
Film comi...	X	0	X	0	0	0	X	0	X	0	0	X	X	0	0	X

Figure 1 – Tableau des relations binaires

À l'aide du logiciel Galicia, nous obtenons le treillis de Galois suivant :

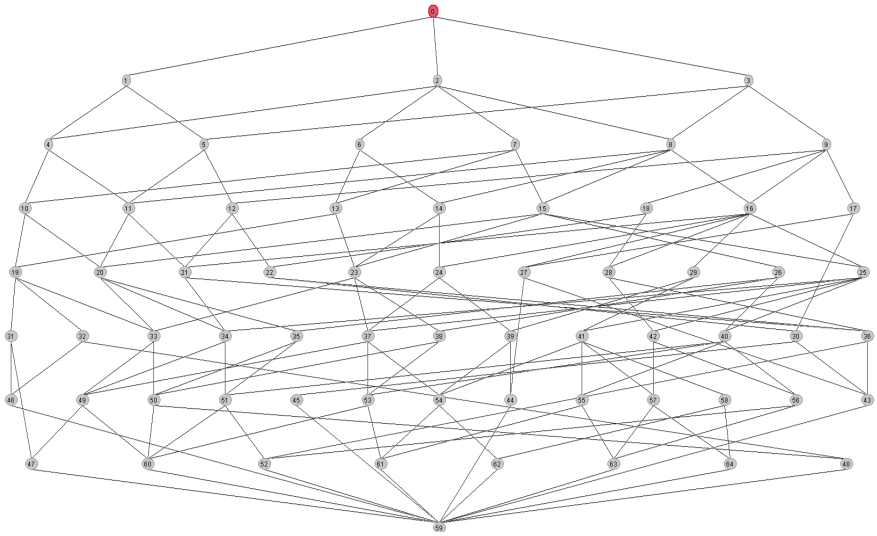

Figure 2 – Treillis de Galois

1.2 Interprétation

Dans un treillis de Galois, les nœuds situés aux extrémités correspondent soit à l'ensemble de tous les individus (en haut), soit à l'ensemble de toutes les caractéristiques (en bas). Ces nœuds étant trop généraux ou trop spécifiques, leur analyse n'est pas nécessaire.

Le nœud 57 regroupe les films dramatiques (FD), les films policiers (FP) et les séries policières (SP), caractérisés par les propriétés suivantes : ils s'adressent aux adolescents et aux adultes (A = Oui), ont un objectif distractif (D = Oui), ne sont pas éducatifs (ED = Non), ne ciblent pas les enfants (E = Non), utilisent des images réelles (IR = Oui) et n'incluent pas d'images de synthèse (IS = Non). Ce regroupement correspond à des œuvres qui partagent des thématiques et des intentions narratives du genre "thriller".
Par ailleurs, ce nœud est lié à la classe 42, qui partage les mêmes caractéristiques mais inclut également des œuvres utilisant des images de synthèse. Cela permet d'y intégrer des films d'horreur (FH).
Cette classe peut être interprétée comme un regroupement de film conçues pour provoquer du suspense ou des "frissons".

Le nœud 16 représente une classe large regroupant plusieurs types de films et séries partageant diverses caractéristiques : un public adulte, une vocation distractive, et des images réelles. Ce nœud est particulièrement

intéressant, car il se connecte à plusieurs autres classes. Ces caractéristiques sont partagées par des genres variés. De ce fait, cette classe regroupe de nombreux types de films et séries, tels que les clips musicaux, les documentaires artistiques, les documentaires sur la nature, les documentaires scientifiques, les films comiques, les films dramatiques, les films de fantasy, les films d'horreur, ainsi que les séries humoristiques et policières.

On remarque également le nœud 17, qui regroupe tous les documentaires. Les caractéristiques de cette classe sont qu'elle s'adresse aux adultes (A = Oui), a une vocation éducative (ED = Oui), utilise des images réelles (IR = Oui) et correspond à des productions de courte durée (30-60 minutes). Ces caractéristiques décrivent ce que sont les documentaires.

2 Classification hiérarchique de parcelles forestières tropicales

Charger dans le logiciel les données relatives au peuplement arboré de la forêt du bassin du Congo (Datagenus.csv). Inspectez le fichier et corrigez-en les erreurs triviales s'il en est. Ces données fournissent sur 1000 parcelles de cette forêt : les variables de comptage de 27 espèces d'arbres (gen1, ..., gen27), la surface de la parcelle, le type forestier (forest) tel qu'identifié par les écologues . On ne tiendra pas compte des autres variables. Calculer la densité de peuplement de chaque espèce par unité de surface pour les 1000 parcelles. Les parcelles seront traduits en nuage dans l'espace des 27 densités de peuplement.

2.1 Préparation des données

Nous traduisons les observations dans l'espace euclidien, où la distance servira de mesure de dissimilarité globale entre les parcelles. Cette mesure repose sur la comparaison des écarts sur les différentes dimensions (densités des espèces). Puisque toutes les dimensions (espèces) sont supposées avoir une importance équivalente, aucune ne doit dominer le calcul.
On note x_i la i-ième parcelle. La distance euclidienne entre deux parcelles x_1 et x_2 est :

$$d(x_1, x_2) = \sqrt{\sum_{j=1}^{27} (x_1^j - x_2^j)^2}$$

où x_1^j et x_2^j représentent respectivement la densité de l'espèce j dans les parcelles 1 et 2.
En analysant la contribution de chaque espèce, nous obtenons le tableau suivant :

Table 1 – Contribution des espèces (gen1 à gen11)

	gen1	gen2	gen3	gen4	gen5	gen6	gen7	gen8	gen9	gen10	gen11
Pourcentage	0.23	0.00	0.03	0.00	0.01	0.05	0.00	64.90	0.07	0.48	1.03

Nous voyons que les variables ne contribuent pas tous de la même manière. Les contributions des espèces gen1 à gen27 à la distance euclidienne ne sont pas uniformes. Par exemple, gen8 représente 64.90 % de la distance, ce qui montre que les variations dans cette dimension dominent les calculs de dissimilarité.
Nous standardisons les densités de peuplement et on note $\mathbf{Z}$ la matrice associée :

$$z_i^j = \frac{x_i^j - \overline{x}^j}{\sigma_{x^j}}$$

avec x_i^j la densité de l'espèce j sur la parcelle i, $\overline{x}^j$ la moyenne des densités pour l'espèce j, et σ_{x^j} l'écart-type de l'espèce j.
La standardisation assure que chaque dimension contribue de manière équivalente à la mesure de la dissimilarité, indépendamment de son échelle initiale ou de sa variabilité.

2.2 CAH des parcelles sur les densités de peuplement

2.2.1 Classification ascendante hiérarchique

Dans cette section, nous procéderons à la classification des parcelles en fonction de leur peuplement arboré. L'objectif est de déterminer le nombre optimal de classes pertinentes pour cette partition. À partir des données standardisées représentant les densités de peuplement obtenues à la question 2.1, nous utiliserons le code en langage R fourni dans le sujet pour effectuer une classification ascendante hiérarchique (CAH). Cette méthode nous permettra d'identifier les classes les plus adaptées à la caractérisation des parcelles. Nous allons utiliser divers indice d'agrégation (Ward, maximum, moyen) qui permettent de mesurer la difficulté d'agrégation de deux classes. L'indice de saut minimum étant à part nous ne l'utiliserons pas. En effet, ses résultats sont trop différents. Il permet de dépister les chaines/continuité.

• Indice de Ward
L'indice de Ward est définie par :

Soient A et B deux classes de centre de gravité $\bar{x}_A$ et $\bar{x}_B$, et de poids w_A et w_B.

$$\mu_{\text{Ward}}(A, B) = \frac{w_A w_B}{w_A + w_B} \|\bar{x}_B - \bar{x}_A\|^2$$

Avec cette indice, nous obtenons le dendogramme suivant :

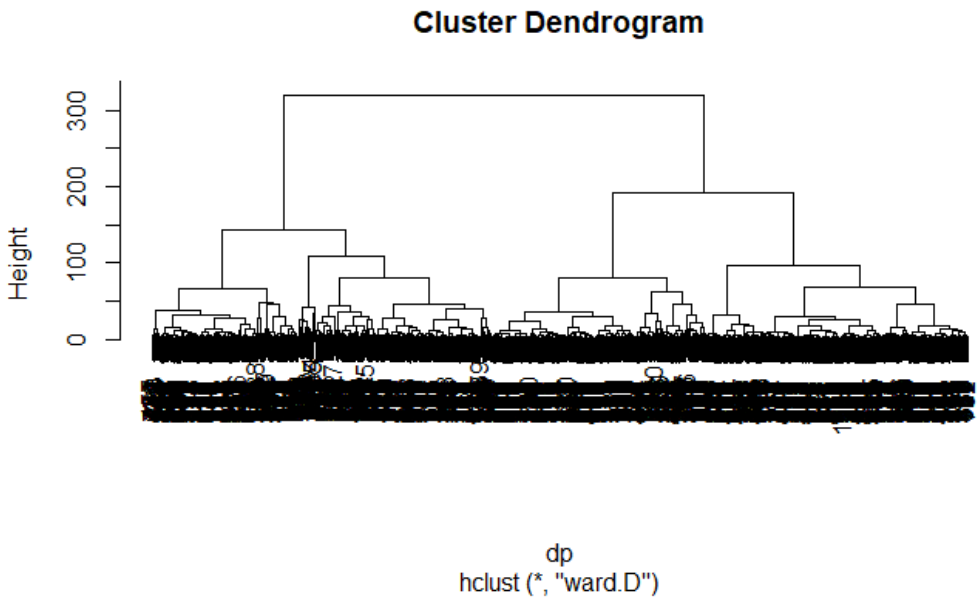

Figure 3 – Dendogramme (indice de Ward)

On remarque que l'on peut partitionner notre arbre en 4 classes. Pour confirmer notre choix d'utiliser 4 classes, nous pouvons utiliser l'histogramme des niveaux de μ_{ward}.

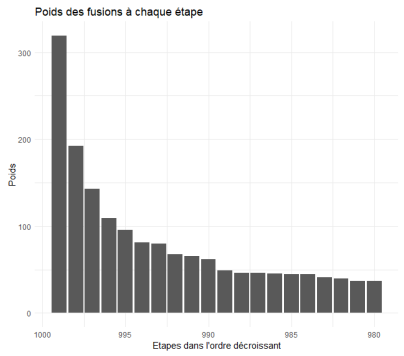

Figure 4 – Histogramme des niveaux μ_{Ward}

La figure 4 illustre que la différence de coût d'agrégation entre une partition en quatre classes et une partition en cinq classes est relativement faible. Par conséquent, une partition en quatre classes semble appropriée.

Les R^2 associés à chaque classe sont ensuite calculés :

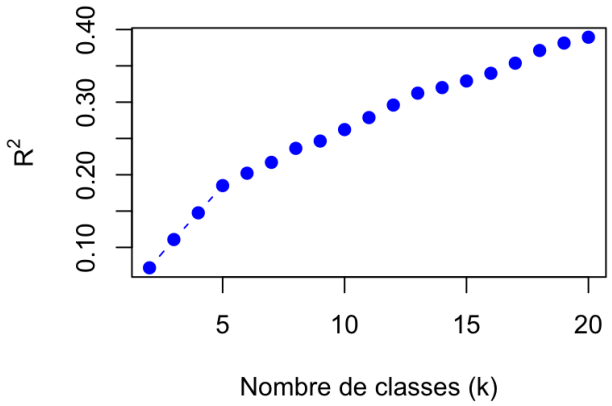

Figure 5 – Evolution du R2 en fonction du nombre de classe

À partir de ce graphique, nous constatons que les partitions potentielles sont en 3, 4 et 5 classes, avec des R^2 respectifs de 0.111, 0.148 et 0.185. Nous décidons donc d'opter pour une partition en 4 classes.
Ainsi, nous obtenons la répartition suivante :

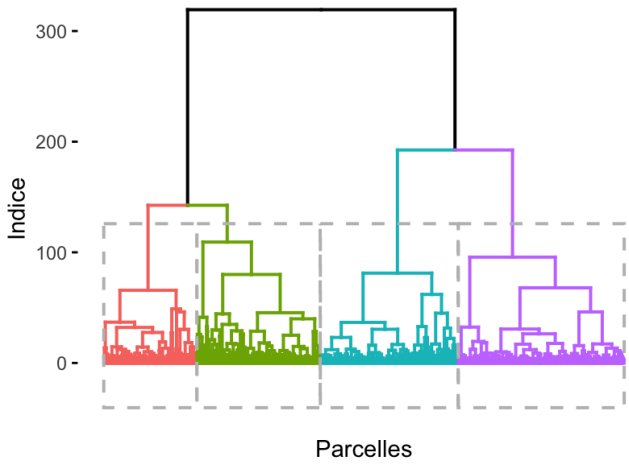

Figure 6 – Dendogramme indice de Ward

- Indice du saut maximum

La méthode du saut maximum consiste à mesurer la similitude avec la paire la plus eloignée L'indice de saut maximum est définie par :

Soient A et B deux classes :

$$\mu(A, B) = \max_{a \in A, b \in B} d(a, b)$$

Avec l'indice de saut maximum, on obtient le dendogramme suivant :

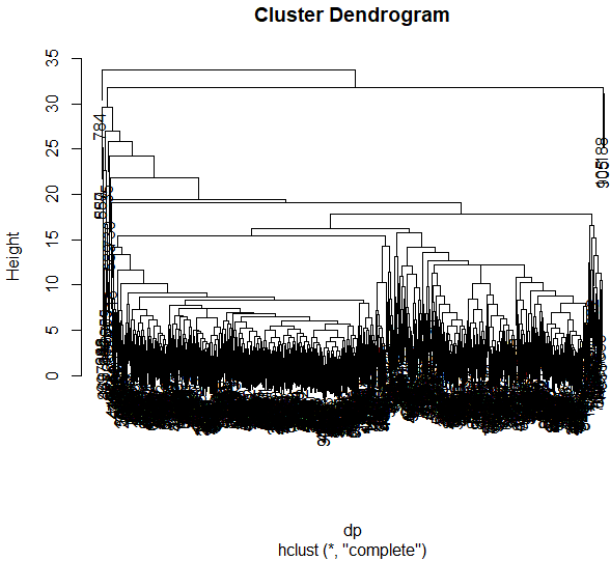

Figure 7 – Dendrogramme indice du saut max

Avec son histogramme associé

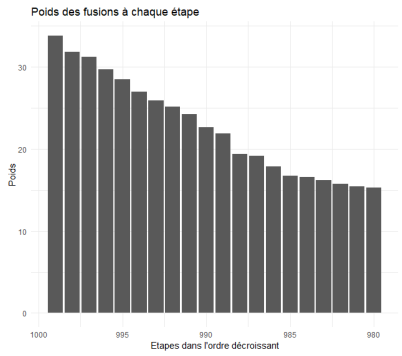

Figure 8 – Histogramme des niveaux saut maximum

L'examen de la figure 7 révèle la présence de points atypiques, notamment les parcelles 905, 105 et 188, qui se distinguent par leur position en haut à droite du graphique. L'interprétation du nombre optimal de classes à partir de cette figure s'avère délicate. Afin de clarifier cette question, nous avons tracé l'histogramme présenté en figure 8. Celui-ci permet de constater qu'un partitionnement en quatre classes semble le plus pertinent pour notre analyse.

- Comparaison

Pour évaluer le degré de similarité entre deux partitions P et P', nous utilisons l'indice de Rand. Cette mesure repose sur l'examen de toutes les paires d'éléments présentes dans les partitions.
Nous obtenons les résultats suivant :

Comparaison des partitions	**Indice de Rand**
P_{WARD} et P_{MAX}	0.2639
P_{WARD} et P_{MOYEN}	0.2629
P_{MAX} et P_{MOYEN}	0.9980
P_{MIN} et P_{MOYEN}	0.9960

Table 2 – Comparaison des partitions à l'aide de l'indice de Rand

Les indices de Rand montrent que les partitions obtenues par les méthodes *Ward* et *Max* (0.2639), ainsi que *Ward* et *Moyen* (0.2629), sont relativement peu similaires, indiquant des différences notables dans le regroupement des éléments. En revanche, les partitions *Max* et *Moyen* (0.9980) ainsi que *Min* et *Moyen* (0.9960) sont presque identiques, suggérant une forte similarité dans les regroupements effectués par ces méthodes. Ainsi, les partitions *Max*, *Min*, et *Moyen* se ressemblent beaucoup, tandis que *Ward* produit une partition distincte.

2.3 Optimisation d'une partition avec les K-means

Après avoir identifié une partition en 4 classes prometteuse avec la CAH. Nous poursuivons l'analyse en optimisant ces groupements à l'aide de la méthode des K-means. La commande kmeans de R nous permet de calculer une partition à partir d'un jeu de données et de centres de gravité. Montrons que les centres de gravité sont bien obtenus par la formule que nous avons implementée.

Avec les notations du cours, nous savons que les centres de gravités correspondent aux moyennes par classes de chaque variable. Ainsi, nous pouvons utiliser la formule démontrée en cours.

Notons :

— $X \in \mathbb{R}^{1000\times 27}$ la matrice des variables quantitatives.
— $M \in \mathbb{R}^{1000\times 4}$ la matrice indicatrice des modalités.
— $W = \frac{1}{1000} I_{1000}$ la matrice des poids des individus.
— $C \in \mathbb{R}^{4\times 27}$ la matrice des centres de gravité.

On a alors,

$$
\begin{aligned}
C &= (M'WM)^{-1}M'WX = (M'\frac{1}{1000}I_{1000}M)^{-1}M'\frac{1}{1000}I_{1000}X \\
&= \frac{1000}{1000}(M'M)^{-1}M'X \\
&= (M'M)^{-1}M'X
\end{aligned}
$$

Ce que l'on voulait démontrer.

Après l'implémentation de la méthode des K-means, nous observons une valeur de $R^2 = 0.180$ Cette valeur indique une amélioration de la partition des classes.

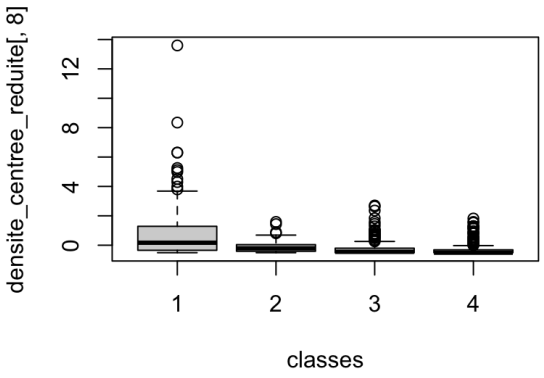

Figure 9 – Boites à moustache pour l'espèce 10 avant K-means

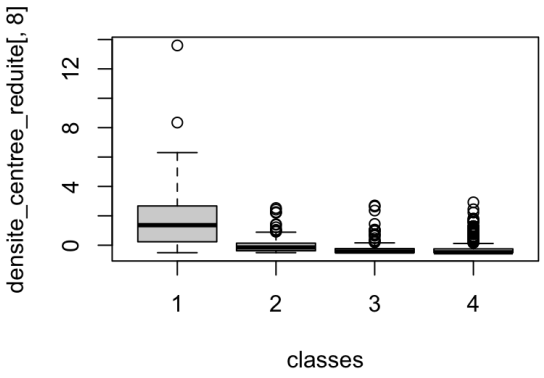

Figure 10 – Boites à moustache pour l'espèce 10 après K-means

À partir des deux figures précédentes, nous observons un des effets de la méthode des K-means. La différence la plus marquante concerne la classe 1, où l'on constate une diminution des valeurs atypiques et un changement de la médiane.
Nous souhaitons désormais interpréter les classes obtenues. À cet effet, nous utiliserons l'indice de Tschuprow pour examiner l'association entre les variables, à savoir le type forestier et le type géologique, avec les classes formées. Nous rappelons que l'indice de Tshuprow est donné par :

$$T^2 = \frac{\phi^2(X,Y)}{\sqrt{I-1}\sqrt{J-1}}$$

avec I le nombre de modalité de X une variable nominale et J le nombre de modalité de Y une variable nominale et ϕ^2 le coefficient de contingence qui mesure la liaison entre deux variables.
Nous obtenons alors, pour le type forestier, une valeur de 0.325, et pour le type géologique, une valeur de 0.369. Bien que l'indice ne révèle pas une forte association, il montre tout de même une certaine relation entre le type forestier et les classes.

Nous analysons la distribution des types forestiers et géologiques au sein de chaque classe dans le but d'extraire des informations pertinentes.

	1	2	3	4
Type forestier 1	0.500	0.167	0.432	0.208
Type forestier 2	0.106	0.008	0.054	0.190
Type forestier 3	0.058	0.004	0.014	0.028
Type forestier 4	0.086	0	0.027	0.007
Type forestier 5	0.144	0.047	0.140	0.263
Type forestier 6	0.058	0.024	0.086	0.152
Type forestier 7	0.048	0.750	0.248	0.151

Table 3 – Proportion type forestier par classe

	1	2	3	4
Type géologique 1	0.038	0.019	0.158	0.145
Type géologique 2	0.087	0.004	0.131	0.052
Type géologique 3	0.240	0.758	0.018	0.161
Type géologique 5	0.221	0.012	0.423	0.191
Type géologique 6	0.413	0.206	0.270	0.450

Table 4 – Proportion type géologique par classe

(Le type géologique 4 n'est pas présent dans les données de base.)

Globalement, les classes 1 et 4 présentent des similitudes dans leurs caractéristiques géologiques et forestières. Toutefois, elles se distinguent par une prédominance du type géologique 1 dans la classe 4. De plus, elles diffèrent en ce qui concerne la proportion du type forestier 1, qui est plus présente dans la classe 1 que dans la classe 4.
La classe 2 se caractérise par une forte présence du type forestier 7 et du type géologique 3, tandis qu'elle affiche une faible proportion du type forestier 3 et une absence de type forestier 4.
Quant à la classe 3, elle montre une faible proportion du type géologique 3, représentant seulement 1,8 % de sa composition.
Ces observations mettent en évidence des variations spécifiques dans la distribution des types géologiques et forestiers au sein des différentes classes.

3 ANNEXE

```
rm(list = ls())

datanul <- read.csv(file='C:/Users/damie/desktop/MASTER/ADM/tp/Datagenus.csv', sep=';')
data <- datanul[1:1000,]

espece <- paste0("gen", 1:27)
densite <- data[espece] / data$surface
densite <- as.matrix(densite)

n <- nrow(densite)
p <- ncol(densite)

moyennes_especes <- colMeans(densite)
sd_especes <- sqrt(colSums((densite - matrix(moyennes_especes, n, p, byrow = TRUE))^2) / (n))

tableau <- (densite - matrix(moyennes_especes, n, p, byrow = TRUE)) / matrix(sd_especes, n, p, byrow = TRUE)

dp = dist(tableau, method="euclidean")
CAHDP = hclust(d=dp, method = "ward.D")
plot(CAHDP)

PDP2 = cutree(tree = CAHDP, k=4)
rect.hclust(CAHDP, 4, border="blue")

R2_PDP2 = cbind(rep(0 , ncol(tableau)))
for (i in 1:ncol(tableau)) {
  R2_PDP2[i] = summary(lm(tableau[,i]~as.factor(PDP2)))$r.squared
}
row.names(R2_PDP2) = colnames(tableau)
R2G_PDP2 = mean(R2_PDP2)

# Partie K-means
IC2DP = data.frame(model.matrix(~as.factor(PDP2)-1))
mIC2DP = as.matrix(IC2DP)
mDP = as.matrix(tableau)
CentresC2 = solve(t(mIC2DP) %*% mIC2DP) %*% t(mIC2DP)%*% mDP
KMDP2 = kmeans(tableau, CentresC2)
KMDP2$cluster

boxplot(tableau[, 1]~as.factor(KMDP2$cluster), main="Boxplot pour une variable spécifique par Cluster K-means")

R2_KMDP2 = cbind(rep(0, ncol(tableau)))
for (i in 1:ncol(tableau)) {
  R2_KMDP2[i] = summary(lm(tableau[, i] ~ as.factor(KMDP2$cluster)))$r.squared
}
row.names(R2_KMDP2) = colnames(tableau)
R2G_KMDP2 = mean(R2_KMDP2)


print(paste("Le R^2 global après K-means est de :", R2G_KMDP2))



data$cluster <- KMDP2$cluster
forest_by_cluster <- table(data$forest, data$cluster)
print(addmargins(forest_by_cluster))
```

```
proportions <- prop.table(forest_by_cluster, 2)
print(proportions)


#Tshuprow

chisq_test <- chisq.test(table(type_forestier, KMDP2$cluster))
Tschuprow_T <- sqrt(chisq_test$statistic / (n * sqrt((length(unique(type_forestier))-1)
    *(4-1))))
print(Tschuprow_T)

chisq_test <- chisq.test(table(type_geo, KMDP2$cluster))
Tschuprow_T <- sqrt(chisq_test$statistic / (n * sqrt((length(unique(type_geo))-1)*(4-1))
    ))
print(Tschuprow_T)
```

```
rm(list = ls())
library(factoextra)

# On charge le dataframe.
tab <- read.csv("./Datagenus.csv", sep = ";")
# On enleve la ligne qui pose probleme.
data <- tab[1 :1000,]

# Sélectionner les colonnes des especes
espece <- paste0("gen", 1:27)   # Calcule de la densité de peuplement pour chaque espece
densite <- data[espece] / data$surface
densite <- as.matrix(densite)   # Convertir la dataframe densité en matrice densité

#Partie 1 preparation des donnees

# Statistiques descriptives pour chaque espece
summary(densite)
# On s'apercoit que les moyennes, le min et le max differe entre chaque espece
# donc la distance euclidienne entre chaque variable peut etre consequente.

# Calcul direct de la matrice des distances
# Initialiser une matrice pour stocker les contributions par variable
n = 1000 # Nombre de parcelle
p = 27   # Nombre de caractéristique
contributions <- array(0, dim = c(n, n, p))
# Calcul des distances et contributions
for (i in 1:n) {
  for (j in i:n) {
    # Différence au carré pour chaque variable
    diff_carre <- (densite[i, ] - densite[j, ])^2
    # Distance totale (somme des différences au carré)
    dist_carre <- sum(diff_carre)
    # Contribution de chaque variable
    if (dist_carre != 0) {
      contributions[i, j, ] <- round((diff_carre / dist_carre)*100,2)

    }
  }
}
print(contributions[1, 2,])
# On standardise !
# Centrage et réduction
n <- nrow(densite) # Nombre de parcelles
p <- ncol(densite) # Nombre d'espèces
# Calcul des moyennes et écarts-types par espèce
```

```
moyennes_especes <- colMeans(densite)
sd_especes <- sqrt(colSums((densite - matrix(moyennes_especes, n, p, byrow = TRUE))^2) / (n))
# Centrage et réduction des densités
densite_centree_reduite <- (densite - matrix(moyennes_especes, n, p, byrow = TRUE)) / matrix(sd_especes, n, p, byrow = TRUE)
```

```
# Partie 2 CAH des parcelles sur les densites de peuplement.

#-----------------------------------Pour l'indice de Ward ----------------------------------
#a) création de la matrice des distances euclidiennes:
dp=dist(densite_centree_reduite, method="euclidean")
#b) CAH avec Ward
CAHDP = hclust(d=dp, method = "ward.D")
plot(CAHDP , main="",xlab ="Parcelles", ylab="Indice" ,sub="")
# Nombre de classe observe
k=4
rect.hclust(CAHDP, k, border="blue")  # Pour k classes
#Coupure de l'arbre et fabrication de la variable de classe correspondant à la partition obtenue.
PDP2 = cutree(tree = CAHDP, k)
WARD = cutree(tree=CAHDP,k)
#Calcul du R2 des variables avec la variable de classe. On va stocker tous les R2 dans un seul vecteur
R2_PDP2 = cbind(rep(0 , ncol(densite)))
#Puis, on calcule les R2 de toutes les variables avec la variable de classe et on met les résultats dans R2:
for (i in cbind(1:ncol(densite))) {
  R2_PDP2[i] = summary(lm(densite[,i]~as.factor(PDP2)))$r.squared
}
#On peut réassigner les noms des variables aux éléments de ce vecteur:
row.names(R2_PDP2) = colnames(densite)
#f) Calcul du R2 de la partition:
R2G_PV2 = mean(R2_PDP2)

# Calcul de la matrice des distances
dp = dist(densite_centree_reduite, method = "euclidean")

# CAH avec la méthode de Ward
CAHDP = hclust(d = dp, method = "ward.D")


# Création d'une liste pour stocker les parcelles dans chaque classe
classes_parcelles_ward <- list()
# Remplir la liste avec les indices des parcelles dans chaque classe
for (i in 1:k) {
  classes_parcelles_ward[[i]] <- which(PDP2 == i)
}
# Création d'une liste pour stocker les résultats par classe
resultats_forest_par_classe_ward <- list()

# Parcourir chaque classe et compter le nombre de parcelles par type de géologie
for (i in 1:k) {
  # Obtenir les indices des parcelles dans la classe i
  parcelles_classe_i <- classes_parcelles_ward[[i]]

  # Extraire les types de géologie correspondants pour ces parcelles
```

```
  forest_parcelles_i <- data$forest[parcelles_classe_i]

  # Calculer la fréquence des types de géologie dans cette classe
  resultats_forest_par_classe_ward[[i]] <- table(forest_parcelles_i)
}

fviz_dend(CAHDP,
          k = 4,
          show_labels = FALSE,
          rect = TRUE,
          xlab = "Parcelles",  # Nom de l'axe X
          ylab = "Indice",      # Nom de l'axe Y
          main =""
)

#-------------------------------------Pour l'indice du saut maximal
#CAH avec saut maximal
CAHDP2 = hclust(d=dp, method = "complete")
plot(CAHDP2, main="Dendogramme (indice du saut maximal)",xlab ="Parcelles", ylab="Indice" ,sub="")
rect.hclust(CAHDP2, k, border="blue")  # Pour k classes
#Coupure de l'arbre et fabrication de la variable de classe correspondant à la partition obtenue.
PDP2_2 = cutree(tree = CAHDP2, k)
MAX = cutree(tree = CAHDP2, k)
#Calcul du R2 des variables avec la variable de classe. On va stocker tous les R2 dans un seul vecteur
R2_PDP2_2 = cbind(rep(0 , ncol(densite)))
#Puis, on calcule les R2 de toutes les variables avec la variable de classe et on met les résultats dans R2:
for (i in cbind(1:ncol(densite))) {
  R2_PDP2_2[i] = summary(lm(densite[,i]~as.factor(PDP2_2)))$r.squared
}
#On peut réassigner les noms des variables aux éléments de ce vecteur:
row.names(R2_PDP2_2) = colnames(densite)
#f) Calcul du R2 de la partition:
R2G_PV2_2 = mean(R2_PDP2_2)

# Création d'une liste pour stocker les parcelles dans chaque classe
classes_parcelles_max <- list()
# Remplir la liste avec les indices des parcelles dans chaque classe
for (i in 1:k) {
  classes_parcelles_max[[i]] <- which(PDP2_2 == i)
}
# Création d'une liste pour stocker les résultats par classe
resultats_forest_par_classe_max <- list()

# Parcourir chaque classe et compter le nombre de parcelles par type de géologie
for (i in 1:k) {
  # Obtenir les indices des parcelles dans la classe i
  parcelles_classe_i <- classes_parcelles_max[[i]]

  # Extraire les types de géologie correspondants pour ces parcelles
  forest_parcelles_i <- data$forest[parcelles_classe_i]

  # Calculer la fréquence des types de géologie dans cette classe
  resultats_forest_par_classe_max[[i]] <- table(forest_parcelles_i)
}
fviz_dend(CAHDP2,
          k = 4,
          show_labels = FALSE,
          rect = TRUE,
```

```
          xlab = "Parcelles",  # Nom de l'axe X
          ylab = "Indice",      # Nom de l'axe Y
          main=""
)
#-------------------------------------Pour l'indice du saut minimal
CAHDP3 = hclust(d=dp, method = "single")
plot(CAHDP3, main="Dendogramme (indice du saut minimal)",xlab ="Parcelles", ylab="Indice
    " ,sub="")
#Coupure de l'arbre et fabrication de la variable de classe correspondant à la partition
     obtenue.
PDP2_3= cutree(tree = CAHDP3, k)
MIN= cutree(tree = CAHDP3, k)
#Calcul du R2 des variables avec la variable de classe. On va stocker tous les R2 dans
    un seul vecteur
R2_PDP2_3 = cbind(rep(0 , ncol(densite)))
#Puis, on calcule les R2 de toutes les variables avec la variable de classe et on met
    les résultats dans R2:
for (i in cbind(1:ncol(densite))) {
  R2_PDP2_3[i] = summary(lm(densite[,i]~as.factor(PDP2_3)))$r.squared
}
#On peut réassigner les noms des variables aux éléments de ce vecteur:
row.names(R2_PDP2_3) = colnames(densite)
#f) Calcul du R2 de la partition:
R2G_PV2_3 = mean(R2_PDP2_3)
hist(CAHDP3$height)

# Création d'une liste pour stocker les parcelles dans chaque classe
classes_parcelles_min <- list()
# Remplir la liste avec les indices des parcelles dans chaque classe
for (i in 1:k) {
  classes_parcelles_min[[i]] <- which(PDP2_3 == i)
}
# Création d'une liste pour stocker les parcelles dans chaque classe
classes_parcelles_min <- list()
# Remplir la liste avec les indices des parcelles dans chaque classe
for (i in 1:k) {
  classes_parcelles_min[[i]] <- which(PDP2_3 == i)
}
# Création d'une liste pour stocker les résultats par classe
resultats_forest_par_classe_min <- list()

# Parcourir chaque classe et compter le nombre de parcelles par type de géologie
for (i in 1:k) {
  # Obtenir les indices des parcelles dans la classe i
  parcelles_classe_i <- classes_parcelles_min[[i]]

  # Extraire les types de géologie correspondants pour ces parcelles
  forest_parcelles_i <- data$forest[parcelles_classe_i]

  # Calculer la fréquence des types de géologie dans cette classe
  resultats_forest_par_classe_min[[i]] <- table(forest_parcelles_i)
}

#-------------------------------------Pour l'indice du saut moyen
CAHDP4 = hclust(d=dp, method = "average")
plot(CAHDP4, main="Dendogramme (indice du saut moyen)",xlab ="Parcelles", ylab="Indice"
    ,sub="")
#Coupure de l'arbre et fabrication de la variable de classe correspondant à la partition
     obtenue.
PDP2_4= cutree(tree = CAHDP4, k)
MOYEN= cutree(tree = CAHDP4, k)
#Calcul du R2 des variables avec la variable de classe. On va stocker tous les R2 dans
```

```
    un seul vecteur
R2_PDP2_4 = cbind(rep(0 , ncol(densite)))
#Puis, on calcule les R2 de toutes les variables avec la variable de classe et on met
    les résultats dans R2:
for (i in cbind(1:ncol(densite))) {
  R2_PDP2_4[i] = summary(lm(densite[,i]~as.factor(PDP2_4)))$r.squared
}
#On peut réassigner les noms des variables aux éléments de ce vecteur:
row.names(R2_PDP2_4) = colnames(densite)
#f) Calcul du R2 de la partition:
R2G_PV2_4 = mean(R2_PDP2_4)
hist(CAHDP4$height)
# Création d'une liste pour stocker les parcelles dans chaque classe
classes_parcelles_moy <- list()
# Remplir la liste avec les indices des parcelles dans chaque classe
for (i in 1:k) {
  classes_parcelles_moy[[i]] <- which(PDP2_4 == i)
}
# Création d'une liste pour stocker les parcelles dans chaque classe
classes_parcelles_moy <- list()
# Remplir la liste avec les indices des parcelles dans chaque classe
for (i in 1:k) {
  classes_parcelles_moy[[i]] <- which(PDP2_4 == i)
}
# Création d'une liste pour stocker les résultats par classe
resultats_forest_par_classe_moy <- list()

# Parcourir chaque classe et compter le nombre de parcelles par type de géologie
for (i in 1:k) {
  # Obtenir les indices des parcelles dans la classe i
  parcelles_classe_i <- classes_parcelles_moy[[i]]

  # Extraire les types de géologie correspondants pour ces parcelles
  forest_parcelles_i <- data$forest[parcelles_classe_i]

  # Calculer la fréquence des types de géologie dans cette classe
  resultats_forest_par_classe_moy[[i]] <- table(forest_parcelles_i)
}
fviz_dend(CAHDP4,
          k = 4,
          show_labels = FALSE,
          rect = TRUE,
          xlab = "Parcelles",  # Nom de l'axe X
          ylab = "Indice",      # Nom de l'axe Y
          main=""
)
#------------------INDICE DE RAND--------------------------
# Fonction pour calculer l'indice de Rand
rand_index <- function(partition1, partition2) {
  n <- length(partition1)

  # Initialisation des compteurs
  C1 <- 0
  C2 <- 0
  D1 <- 0
  D2 <- 0

  # Boucle sur toutes les paires possibles
  for (i in 1:(n - 1)) {
    for (j in (i + 1):n) {
      same_in_P <- (partition1[i] == partition1[j])
      same_in_P_prime <- (partition2[i] == partition2[j])
```

```
276
277       if (same_in_P && same_in_P_prime) {
278         C1 <- C1 + 1
279       } else if (!same_in_P && !same_in_P_prime) {
280         C2 <- C2 + 1
281       } else if (same_in_P && !same_in_P_prime) {
282         D1 <- D1 + 1
283       } else if (!same_in_P && same_in_P_prime) {
284         D2 <- D2 + 1
285       }
286     }
287   }
288
289   # Calcul de l'indice de Rand
290   rand_index_value <- (C1 + C2) / (C1 + C2 + D1 + D2)
291
292   return(rand_index_value)
293 }
294
295 indice_rand_1 <- rand_index(WARD, MAX )
296 cat(indice_rand_1)
297 #indice_rand_2 <- rand_index(WARD, MIN)
298 indice_rand_3 <- rand_index(WARD, MOYEN)
299 cat(indice_rand_3)
300 #indice_rand_4 <- rand_index(MAX, MIN)
301 indice_rand_5 <- rand_index(MAX, MOYEN)
302 cat(indice_rand_5)
303 indice_rand_6 <- rand_index(MIN, MOYEN)
304 cat(indice_rand_6)
305
306 # Nous allons maintenant optimiser avec la methode du Kmeans
307
308 #- Transformation d'une variable qualitative en matrice d'indicatrices:
309 IC2DP = data.frame(model.matrix(~as.factor(PDP2)-1))
310 #- Calcul matriciel des centres de gravité de classes de la CAH:
311 mIC2DP = as.matrix(IC2DP)
312 mDP = as.matrix(densite)
313 CentresC2 = solve(t(mIC2DP) %*% mIC2DP) %*% t(mIC2DP)%*% mDP
314 #- K-means à partir de ces centres initiaux:
315 KMDP2 = kmeans(densite_centree_reduite, CentresC2)
316 #- La variable de classe ainsi produite est dans:
317 KMDP2$cluster
318
319 # Extraire les indices des parcelles appartenant au cluster
320 liste_cluster_indices <- list()
321 for(i in 1:k){
322   liste_cluster_indices[[i]] <- which(KMDP2$cluster == i) # indice des parcelles
        appartenant au i eme cluster
323 }
324
325 #- Boxplot d'une variable xj conditionnellement à la variable de classe:
326 boxplot(densite_centree_reduite[,8]~as.factor(PDP2), main="",xlab ="classes")
327 boxplot(densite_centree_reduite[,8]~as.factor(KMDP2$cluster), xlab ="classes")
328
329
330
331 # Voir l'opti des classes et ce qu'il y a dedans.
332 # Création d'une liste pour stocker les parcelles dans chaque classe
333 classes_parcelles_ward_kmeans <- list()
334 # Remplir la liste avec les indices des parcelles dans chaque classe
335 for (i in 1:k) {
336   classes_parcelles_ward_kmeans[[i]] <- which(KMDP2$cluster == i)
```

```
}
# Création d'une liste pour stocker les résultats par classe
resultats_forest_par_classe_ward_kmeans <- list()

# Parcourir chaque classe et compter le nombre de parcelles par type de géologie
for (i in 1:k) {
  # Obtenir les indices des parcelles dans la classe i
  parcelles_classe_i <- classes_parcelles_ward_kmeans[[i]]

  # Extraire les types de géologie correspondants pour ces parcelles
  forest_parcelles_i <- data$forest[parcelles_classe_i]

  # Calculer la fréquence des types de géologie dans cette classe
  resultats_forest_par_classe_ward_kmeans[[i]] <- table(forest_parcelles_i)
}
```