ACT6100 H2019	Analyse de données Solutions 2

1. (a) Pour la matrice des distances euclidiennes, on a

$$\begin{aligned} d_1^2(x_1,x_1) &= d_1^2(x_2,x_2) \\ &= d_1^2(x_3,x_3) \\ &= 0^2+0^2 \\ &= 0 \\ d_1^2(x_1,x_2) &= d_1^2(x_2,x_1) \\ &= 1^2+0^2 \\ &= 1 \\ d_1^2(x_1,x_3) &= d_1^2(x_3,x_1) \\ &= 5^2+5^2 \\ &= 50 \\ d_1^2(x_2,x_3) &= d_1^2(x_3,x_2) \\ &= 4^2+5^2 \\ &= 41, \end{aligned}$$

et donc

$$D_1 = \begin{bmatrix} 0 & 1 & \sqrt{50} \\ 1 & 0 & \sqrt{41} \\ \sqrt{50} & \sqrt{41} & 0 \end{bmatrix}.$$

(b) L'utilisation d'un *agglomerative hierarchichal algorithm* implique qu'on doit débuter avec n groupes et regrouper pas à pas les groupes. Ici, l'analyse débute avec trois clusters : $P = \{x_1\}$, $Q = \{x_2\}$ et $R = \{x_3\}$. À partir de la matrice D_1 calculée en (a), on voit que la plus petite distance est entre les clusters P et Q. On doit donc les regrouper : $P \cup Q = \{x_1, x_2\}$ et calculer la distance entre ces deux clusters. On a

$$\begin{aligned} \delta(P\cup Q, R) &= \min(\delta(P,R), \delta(Q,R)) \\ &= \min(50, 41) \\ &= 41. \end{aligned}$$

La nouvelle matrice des distances est alors donnée par

$$D_2 = \begin{bmatrix} 0 & 41 \\ 41 & 0 \end{bmatrix}.$$

La dernière étape est de regrouper les deux clusters en un seul : $P \cup Q \cup R = \{x_1, x_2, x_3\}$.

(c) Le centre de gravité est donné par

$$\begin{aligned} g &= \sum_{i=1}^{3} p_i x_i \\ &= \left(\frac{1}{3}\right)(0,0) + \left(\frac{1}{3}\right)(1,0) + \left(\frac{1}{3}\right)(5,5) \\ &= (2, 5/3). \end{aligned}$$

Le plan est présenté à la figure 1.

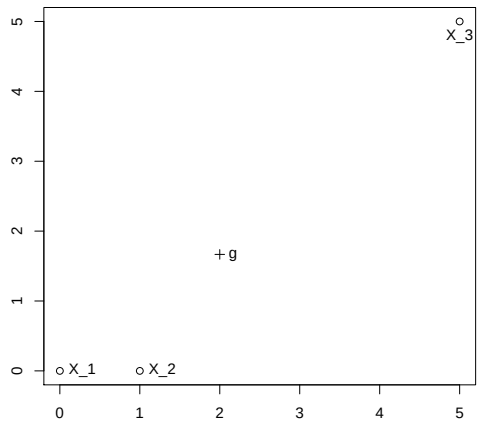

FIGURE 1 – Nuage de points et centre de gravité

2. On débute avec huit clusters, respectivement $X_1, \ldots, X_8$. Le premier regroupement se fait entre les deux clusters les plus près, c'est-à-dire entre les clusters X_3 et X_5 ($\delta(X_3, X_5) = 1$). La nouvelle matrice des distances est donnée par

$$D_2 = \begin{bmatrix} 0 & 10 & 50 & 73 & 98 & 41 & 65 \\ & 0 & 20 & 41 & 80 & 37 & 65 \\ & & 0 & 2 & 25 & 18 & 34 \\ & & & 0 & 17 & 20 & 32 \\ & & & & 0 & 13 & 9 \\ & & & & & 0 & 4 \\ & & & & & & 0 \end{bmatrix}.$$

Le second regroupement se fera entre les clusters X_{35} et X_4 car la distance y est la plus petite ($\delta(X_{35}, X_4) = 2$). La nouvelle matrice des distances est donnée par

$$D_3 = \begin{bmatrix} 0 & 10 & 50 & 98 & 41 & 65 \\ & 0 & 20 & 80 & 37 & 65 \\ & & 0 & 17 & 18 & 32 \\ & & & 0 & 13 & 9 \\ & & & & 0 & 4 \\ & & & & & 0 \end{bmatrix}.$$

Le troisième regroupement se fera entre les clusters X_7 et X_8 car la distance y est la plus petite ($\delta(X_7, X_8) = 4$). La nouvelle matrice des distances est donnée par

$$D_4 = \begin{bmatrix} 0 & 10 & 50 & 98 & 41 \\ & 0 & 20 & 80 & 37 \\ & & 0 & 17 & 18 \\ & & & 0 & 9 \\ & & & & 0 \end{bmatrix}.$$

Le quatrième regroupement se fera entre les clusters X_{78} et X_6 car la distance y est la plus petite ($\delta(X_{78}, X_6) = 9$). La nouvelle matrice des distances est donnée par

$$D_5 = \begin{bmatrix} 0 & 10 & 50 & 41 \\ & 0 & 20 & 37 \\ & & 0 & 17 \\ & & & 0 \end{bmatrix}.$$

Le cinquième regroupement se fera entre les clusters X_1 et X_2 car la distance y est la plus petite

($\delta(X_1, X_2) = 10$). La nouvelle matrice des distances est donnée par

$$D_6 = \begin{bmatrix} 0 & 20 & 37 \\ & 0 & 17 \\ & & 0 \end{bmatrix}.$$

Le sixième regroupement se fera entre les clusters X_{345} et X_{678} car la distance y est la plus petite ($\delta(X_{345}, X_{678}) = 17$). La nouvelle matrice des distances est donnée par

$$D_7 = \begin{bmatrix} 0 & 20 \\ & 0 \end{bmatrix}.$$

Le dernier regroupement se fera entre les clusters X_{12} et X_{345678}. La distance y est de $\delta(X_{12}, X_{345678}) = 20$. Le dendrogramme est présenté à la figure 2. On voit qu'il semble y avoir trois groupes dans les

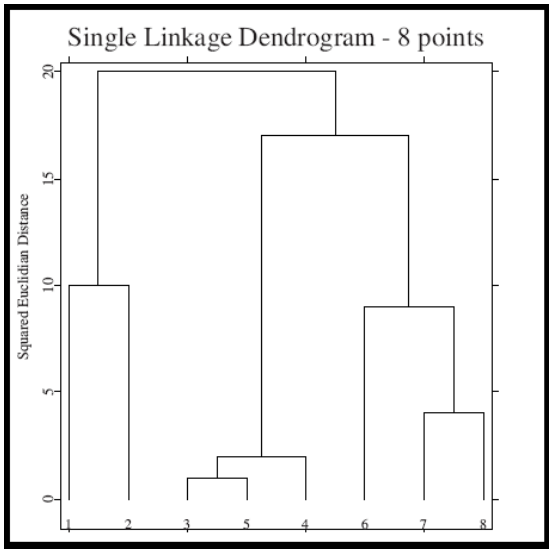

FIGURE 2 – Dendrogramme

données : $g_1 = \{X_1, X_2\}$, $g_2 = \{X_3, X_4, X_5\}$ et $g_3 = \{X_6, X_7, X_8\}$.

3. La différence principale entre les deux algorithmes se trouve dans le critère utilisé pour former les clusters :
   — *Agglomerative hierarchichal algorithm* : les deux clusters ayant la plus petite distance sont regroupés.
   — Algorithme de Ward : les deux clusters dont le regroupement causera la plus petite perte d'inertie (*between*) sont regroupés.

   L'inertie totale est donnée par

$$\begin{aligned} I_T &= \sum_{i=1}^{3} p_i d^2(x_i, g) \\ &= \left(\frac{1}{3}\right)(6.78) + \left(\frac{1}{3}\right)(3.78) + \left(\frac{1}{3}\right)(20.11) \\ &= 10.22. \end{aligned}$$

   Elle peut se décomposer en deux parties : l'inertie dans un groupe (*within*) et l'inertie entre les groupes (*between*). On doit également débuter avec n groupes et regrouper pas à pas les groupes. Ici, l'analyse débute avec trois clusters : $P = \{x_1\}$, $Q = \{x_2\}$ et $R\{x_3\}$.
   On trouve premièrement les centres de gravité des clusters initiaux

$$\begin{aligned} g_P &= \left(\frac{1}{p_P}\right)(p_P)(x_1) \\ &= x_1 \\ &= (0, 0) \\ g_Q &= (1, 0) \\ g_R &= (5, 5). \end{aligned}$$

On a ensuite

$$\begin{aligned}
\delta(P,Q) &= \frac{p_P p_Q}{p_p + p_Q} d^2(g_P, g_Q) \\
&= \frac{(1/3)(1/3)}{(1/3)+(1/3)}(1) \\
&= 1/6 \\
\delta(P,R) &= \frac{p_P p_R}{p_p + p_R} d^2(g_P, g_R) \\
&= \frac{(1/3)(1/3)}{(1/3)+(1/3)}(50) \\
&= 50/6 \\
\delta(Q,R) &= \frac{p_Q p_R}{p_Q + p_R} d^2(g_Q, g_R) \\
&= \frac{(1/3)(1/3)}{(1/3)+(1/3)}(41) \\
&= 41/6
\end{aligned}$$

La matrice des «distances» est alors donnée par

$$D_1 = \begin{bmatrix} 0 & 1/6 & 50/6 \\ 1/6 & 0 & 41/6 \\ 50/6 & 41/6 & 0 \end{bmatrix}.$$

La plus petite perte d'inertie est causée par le regroupement des clusters P et Q : $P \cup Q = \{x_1, x_2\}$. Le centre de gravité de ce nouveau cluster est

$$\begin{aligned}
g_{P\cup Q} &= \left(\frac{1}{2/3}\right)\left(\left(\frac{1}{3}\right)(0,0) + \left(\frac{1}{3}\right)(1,0)\right) \\
&= (0.5, 0).
\end{aligned}$$

La «distance» entre le cluster $P \cup Q$ et le cluster R est donnée par

$$\begin{aligned}
\delta(P\cup Q, R) &= \frac{((1/3)+(1/3))\delta(P,R) + ((1/3)+(1/3))\delta(Q,R) - (1/3)\delta(P,Q)}{(1/3)+(1/3)+(1/3)} \\
&= \frac{(2/3)(50/6) + (2/3)(41/6) - (1/3)(1/6)}{1} \\
&= 181/18.
\end{aligned}$$

La nouvelle matrice des «distances» est alors donnée par

$$D_2 = \begin{bmatrix} 0 & 181/18 \\ 181/18 & 0 \end{bmatrix}.$$

La dernière étape est de regrouper les deux clusters en un seul : $P \cup Q \cup R = \{x_1, x_2, x_3\}$. Les gains/pertes d'intertie sont décrits dans la Table 1. On remarque que pour chacune des étapes,

Groupe(s)	inertie totale (I_T)	inertie *between* (I_B)	inertie *within* (I_W)
P, Q et R	10.22	10.22	0
$P \cup Q$ et R	10.22	$181/18 = 10.05$	$1/6 = 0.17$
$P \cup Q \cup R$	10.22	0	10.22

TABLE 1 – Décomposition de l'inertie

$I_T = I_W + I_B$.

4. On réalise d'abord l'analyse de classification à l'aide de R et en utilisant l'algorithme de Ward et la mesure de distance euclidienne (les choix les plus communs). Le dendrogramme résultant est

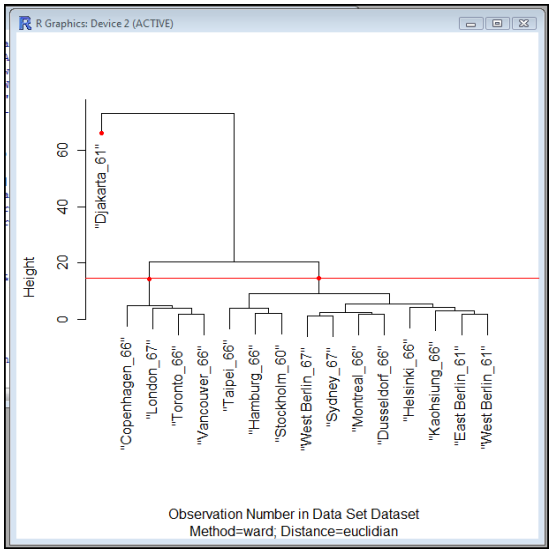

FIGURE 3 – Dendrogramme obtenu à partir de la base de données *LifeE.csv*, de l'algorithme de Ward et de la mesure de distance euclidienne.

présenté à la Figure 3. On observe qu'il semble y avoir trois groupes dans la base de données. Il peut être intéressant de vérifier que cette classification est robuste à un changement d'algorithme ou de mesure de distance. Après quelques essais, on remarque que les groupes semblent être toujours les mêmes, même si parfois on pourrait regrouper les deux groupes comprenant le plus grand nombre d'individus (voir Figure 4 par exemple).

(a) C'est totalement faux puisqu'on remarque la présence au sein d'un même groupe de villes physiquement très éloignées, par exemple Montréal et Sydney.

(b) C'est vrai. On remarque sur le dendrogramme que les villes de Düsseldorf et de Montréal sont regroupées (et donc considérées comme faisant partie d'un même groupe «homogène») bien avant que les villes de Montréal et de Toronto ne soit regroupées.

(c) C'est vrai. On note que
— l'inertie est une mesure d'**hétérogénéité** ;
— l'inertie *between* est une mesure de l'hétérogénéité entre les groupes ; et
— l'objectif d'une analyse de classification est d'obtenir des groupes les plus **homogènes** à l'interne et les plus **hétérogènes** entre eux.
Donc, en retirant d'un groupe contenant toutes les villes la ville de Djakarta qui est la plus éloignée des autres sur le dendrogramme, on **diminue** l'hétérogénéité interne des groupes et on augmente l'hétérogénéité entre les groupes. Ainsi, on augmente l'inertie *between* (I_B).

5. (a) On a

$$0.4g_1 + 0.6g_2 = g$$
$$0.4(1,3) + 0.6(6,2) = (4, 2.4),$$

il faut que

$$\begin{aligned} g_2 &= (g - 0.4g_1)\,/0.6 \\ &= ((4, 2.4) - 0.4(1,3))\,/0.6 \\ &= (6, 2). \end{aligned}$$

(b) Par le théorème de KÖNIG–HUYGENS, on a

$$\begin{aligned} d^2(g_1, g) &= (1-4)^2 + (3-2.4)^2 = 9.36 \\ d^2(g_2, g) &= (6-4)^2 + (2-2.4)^2 = 4.16 \\ I_{\text{between}} &= 0.4\, d^2(g_1, g) + 0.6\, d^2(g_2, g) = 6.24 \\ I_{\text{within}} &= I_1 + I_2 = 5 + 6 = 11 \\ I_{\text{total}} &= I_{\text{within}} + I_{\text{between}} = 17.24. \end{aligned}$$

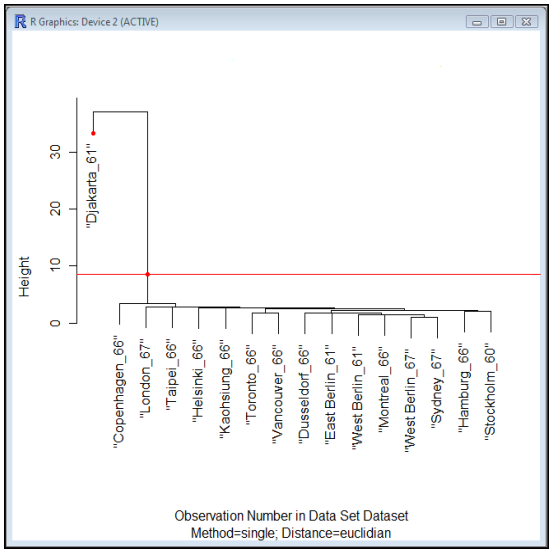

FIGURE 4 – Dendrogramme obtenu à partir de la base de données *LifeE.csv*, de l'algorithme avec lien simple et de la mesure de distance euclidienne.

6. La hauteur représente le niveau d'aggrégation, ici la perte de l'inertie *between* après l'aggrégation des clusters concernés. Dans l'algorithme de Ward, on cherche à aggréger les clusters afin de minimiser à chaque étape cette perte. Selon le théorème de Huygens, la somme de l'inertie *between* et *within* est constante.
7. (a) On a

$$\begin{aligned}
\sum_{i,i'\in C_k} (x_{ij} - x_{i'j})^2 &= \sum_{i\in C_k}\sum_{i'\in C_k} (x_{ij} - \bar{x}_{kj} + \bar{x}_{kj} - x_{i'j})^2 \\
&= \sum_{i\in C_k}\sum_{i'\in C_k} (x_{ij} - \bar{x}_{kj})^2 + (\bar{x}_{kj} - x_{i'j})^2 \\
&\quad + 2\,(x_{ij} - \bar{x}_{kj})\,(x_{i'j} - \bar{x}_{kj}) \\
&= \mathrm{card}(C_k) \sum_{i\in C_k} (x_{ij} - \bar{x}_{kj})^2 \\
&\quad + \mathrm{card}(C_k) \sum_{i'\in C_k} (x_{i'j} - \bar{x}_{kj})^2 \\
&\quad + \sum_{i\in C_k} (x_{ij} - \bar{x}_{kj}) \sum_{i'\in C_k} (x_{i'j} - \bar{x}_{kj}) \\
&= \mathrm{card}(C_k) \sum_{i\in C_k} (x_{ij} - \bar{x}_{kj})^2 \\
&\quad + \mathrm{card}(C_k) \sum_{i'\in C_k} (x_{i'j} - \bar{x}_{kj})^2 .
\end{aligned}$$

Par symétrie, on obtient alors

$$2 \sum_{i,i'\in C_k} (x_{ij} - x_{i'j})^2 = 2 \sum_{i\in C_k} (x_{ij} - \bar{x}_{kj})^2$$

et le résultat à démontrer s'obtient directement.

(b) Le résultat découle directement de la sous-question précédente :

$$\begin{aligned}
\sum_{k=1}^{K} W(C_k) &= \sum_{k=1}^{K} \frac{1}{\mathrm{card}(C_k)} \sum_{i,i'\in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \\
&= 2 \sum_{k=1}^{K} \sum_{j=1}^{p} \sum_{i\in C_k} (x_{ij} - \bar{x}_{kj})^2 .
\end{aligned}$$

(c) À partir du résultats de la sous-question précédente, on remarque que, puisque l'algorithme réassique chaque observation au centroïde le plus près selon la distance euclidienne ($\bar{x}_{kj}$), la valeur de la fonction objectif ne peut que diminuer.

8. (a) Le code est présenté à la Figure 5 et le dendrogramme à la Figure 6.

```
data <- (USArrests)
D <- dist(data, method = "euclidean")
n <- length(data[,1])
tree <- hclust(D^2/(2*n),method="ward.D")
plot(tree)
```

FIGURE 5 – Code informatique.

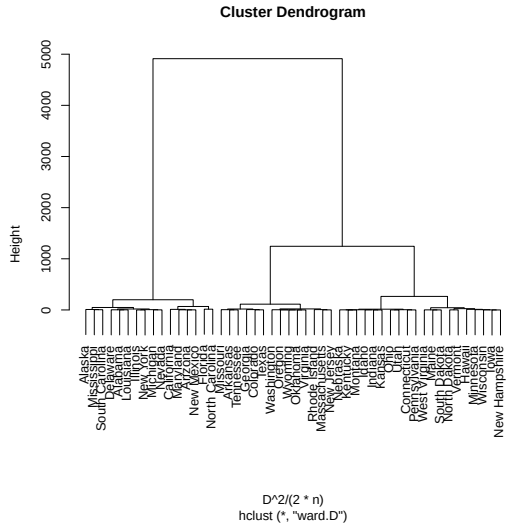

FIGURE 6 – Dendrogramme.

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

```
data$groupe <- cutree(tree, k = 10)
g <- aggregate(data[, 1:4], list(data$groupe), mean)[,2:5]
g

      Murder   Assault UrbanPop      Rape
1  11.471429 247.57143 74.28571 27.200000
2  13.500000 267.00000 46.66667 28.033333
3   9.950000 288.75000 77.00000 32.875000
4  11.500000 195.33333 66.16667 27.433333
5   5.590000 112.40000 65.60000 17.270000
6  14.200000 336.00000 62.50000 24.000000
7   2.980000  56.80000 65.60000 13.340000
8   3.866667  83.33333 45.00000  9.966667
9   5.750000 156.75000 74.00000 19.400000
10  1.500000  46.50000 38.00000  9.250000
```

FIGURE 7 – Code informatique.

(c) Le code est présenté à la Figure 8.

```
modele <- kmeans(data[,1:4], g)
```

FIGURE 8 – Code informatique.

```
### Installation des packages nécessaires
install.packages("BiocManager")
BiocManager::install("EBImage")
library(EBImage)

### Téléchargement des images
fnames <- paste0("cat.", 1001:1100, ".jpg")
original_dataset_dir <- "~/Dropbox/Cours/ACT6100/Images"

### Formatage des images (200 x 200)
n <- 200
XX <- matrix(NA, ncol = 100, nrow = n*n)

img_read <- function(x){
  f <- file.path(original_dataset_dir, fnames[x])
  y <- resize(readImage(f), w = n, h = n)
  XX[,x] <<- matrix(imageData(getFrame(y, i = 1))[1:n, 1:n], ncol = 1)
}
sapply(1:100, function(x) img_read(x))

### Classification
modele1 <- kmeans(t(XX), 3)

### Division des images en trois groupes
im1 <- XX[,which(modele1$cluster == 1)]
im2 <- XX[,which(modele1$cluster == 2)]
im3 <- XX[,which(modele1$cluster == 3)]
```

FIGURE 9 – Code informatique.

9. Le code est présenté à la Figure 9.