CLassfication automatique

TP 1

Notions de distances, similarités et inerties

On considère 6 individus $\{w_1, w_2, w_3, w_4, w_5, w_6\}$ représentés par deux variables X_1 et X_2

	X1	X2
w1	2.0	2
w2	7.5	4
w3	3.0	3
w4	0.5	5
w5	6.0	4
w6	1.5	7

1) Matrice des distances entres les individus

```
# La fonction dist permet de calculer la distance entre les différents individus
#   d'une matrice ou d'un vecteur
dist.manhattan <- dist(data.tp1,method = "manhattan")  # Distance à  la norme L1
xtable(as.matrix(dist.manhattan))
```

	w1	w2	w3	w4	w5	w6
w1	0.0	7.5	2.0	4.5	6.0	5.5
w2	7.5	0.0	5.5	8.0	1.5	9.0
w3	2.0	5.5	0.0	4.5	4.0	5.5
w4	4.5	8.0	4.5	0.0	6.5	3.0
w5	6.0	1.5	4.0	6.5	0.0	7.5
w6	5.5	9.0	5.5	3.0	7.5	0.0

```
dist.euclidean <- dist(data.tp1,method = "euclidean")  # Distance à  la norme L2
xtable(as.matrix(dist.euclidean))
```

	w1	w2	w3	w4	w5	w6
w1	0.000000	5.852350	1.414214	3.354102	4.472136	5.024938
w2	5.852350	0.000000	4.609772	7.071068	1.500000	6.708204
w3	1.414214	4.609772	0.000000	3.201562	3.162278	4.272002
w4	3.354102	7.071068	3.201562	0.000000	5.590170	2.236068
w5	4.472136	1.500000	3.162278	5.590170	0.000000	5.408327
w6	5.024938	6.708204	4.272002	2.236068	5.408327	0.000000

Bien que les valeurs soient différentes, l'ordre de grandeur des distances entres les individus ici reste inchangé quelque soit la distance appliquée à notre jeu de données.

2) Fonction calculant pour un ensemble de données les coordonnées de son barycentre.

```
barycentre<-function(vect){
  # apply permet d'appliquer ici la fonction mean aux colonnes de vect.
  return(apply(vect,2, mean))
}
```

On l' applique à la partition : $\{\{w_1, w_3\}; w_4; \{w_2, w_5\}; w_6\}$

```
# Barycentre w_1, w_3:
G1_3 <- barycentre(data.tp1[c(1,3),])
# w_4 est son propre barycentre
G4   <- data.tp1[4,]
#Barycentre w_2, w_5

G2_5 <- barycentre(data.tp1[c(2,5),])

G6   <- data.tp1[6,]
barycentres <- rbind(G1_3,G4,G2_5,G6)
xtable(barycentres)
```

	X1	X2
G1_3	2.50	2.5
G4	0.50	5.0
G2_5	6.75	4.0
G6	1.50	7.0

3) Calcul de distance avec le barycentre

```
distances<-function(vect){
    bar      <- barycentre(vect)    # calcul du barycentres de l'ensemble de points.
    vect.2 <- rbind(vect,bar)       # On le rajoute à l'ensemble des points afin de calcul
                                    # les distances globales avec l'ensemble des points.
    z      <- dist(vect.2,method = "euclidean")
    return (as.matrix(z))
}

D1  <- distances(data.tp1[c(1,3),])
xtable(D1)
```

	w1	w3	bar
w1	0.0000000	1.4142136	0.7071068
w3	1.4142136	0.0000000	0.7071068
bar	0.7071068	0.7071068	0.0000000

```
D2 <- distances(data.tp1[c(2,5),])

xtable(D2)
```

	w2	w5	bar
w2	0.00	1.50	0.75
w5	1.50	0.00	0.75
bar	0.75	0.75	0.00

4) Calcul de l'inertie totale

```
n <- nrow(data.tp1)
Inertie.totale <- sum((distances(data.tp1)^2)[n+1,1:n])/n
```

L'inertir totale vaut: 8.76

Question 5: Calcul d'inertie inter-classe et intra-classe

```
# barycentre global
G             <- barycentre(data.tp1)

# Matrice des barycentres
BAR           <- rbind(G1_3,G4,G2_5,G6,G)
nbar          <- nrow(BAR)

# distance au carrée entre les différents barycentres et le barycentre global
ecarts        <-(as.matrix(dist(BAR))^2)[nbar,1:nbar-1]
#effectifs de chaque classe
effectifs     <- c(2,1,2,1)

Inertie.inter <- sum(ecarts*effectifs)/sum(effectifs)


temp.1        <- (D1^2)[nrow(D1),1:nrow(D1)-1] # Vecteur de distance entre
                                               # w_1, w_3 et leur barycentre
                                               #
temp.2        <- (D2^2)[nrow(D2),1:nrow(D2)-1] # Vecteur de distance entre
                                              # w_2, w_5 et leur barycentre

# w_4 et w_6 representant des singletons,la distance au barycentre vaut 0.
vect.sum.dist <- rbind(mean(temp.1),0,mean(temp.2),0)

Inertie.Intra <- sum(vect.sum.dist*effectifs)/sum(effectifs)

Intertie.exp <- (1-(Inertie.Intra/Inertie.totale))*100
```

L'inertie inter-classe vaut: 8.4. L'inertie intra-classe vaut: 0.35. Le pourcentage d'inertie expliqué est : 95.96%.

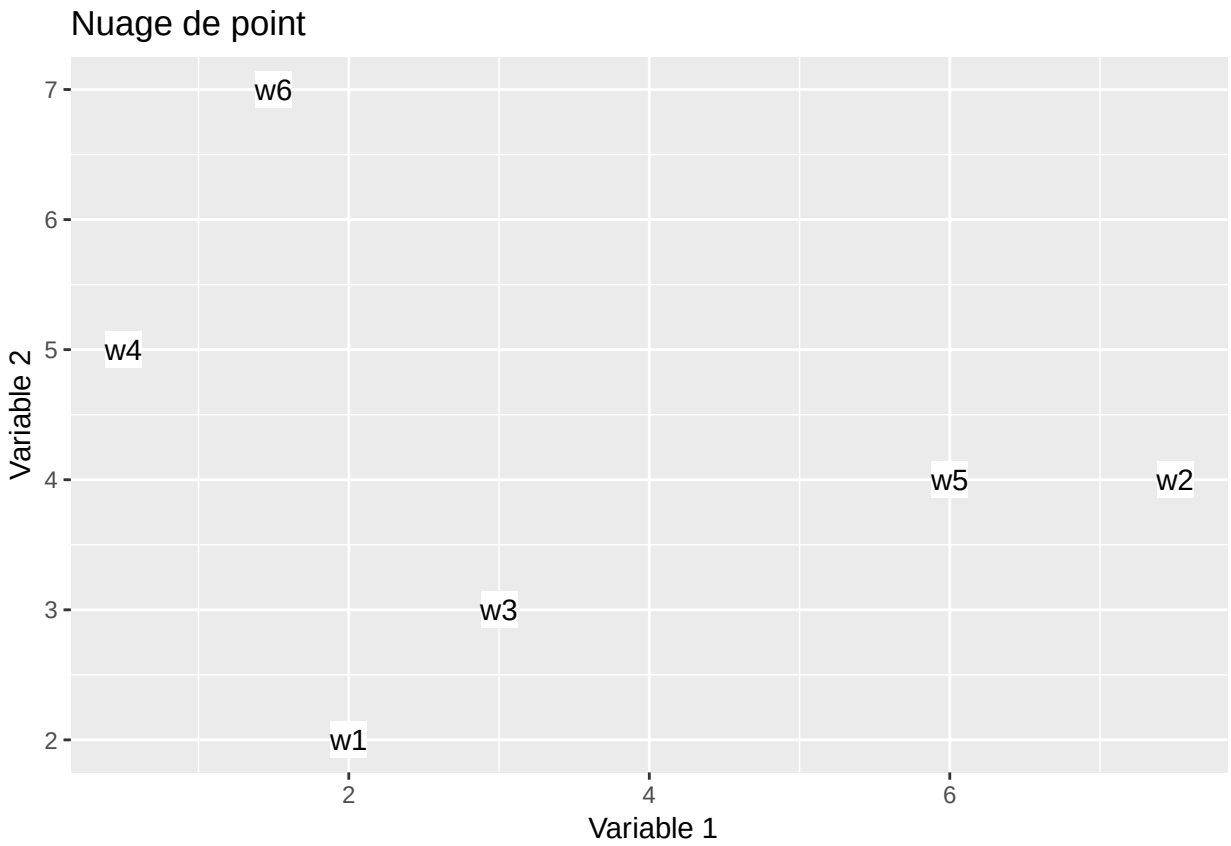
On retrouve bien en sommant ces deux interties l'inertie totale (théoreme de huyguens)

6) Représentation graphique, choix de la segmentation - Calcul d'inertie inter-classe et intra-classe

```
nuage.points <-as.data.frame(data.tp1)
nuage.points
```

	X1	X2
w1	2.0	2
w2	7.5	4
w3	3.0	3
w4	0.5	5
w5	6.0	4
w6	1.5	7

```
p <- ggplot(data= nuage.points, aes(x= nuage.points$X1,
            y=nuage.points$X2,label=rownames(nuage.points)))
p <- p + geom_point(size=2,colour="red")
p <- p + ggtitle("Nuage de point")
p <- p + xlab(label=" Variable 1")
p <- p + ylab(label="Variable 2")
p <- p + geom_point(shape=15,color="white",size=6)+geom_text()
p
```



La représenation graphique suggère de considérer la partition: $\{\{w_1, w_3\}; \{w_2, w_5\}; \{w_4, w_6\}\}$

```r
G4_6 <- barycentre(data.tp1[c(4,6),])

# barycentre global
G            <- barycentre(data.tp1)

# Matrice des barycentres
BAR          <- rbind(G1_3,G4_6,G2_5,G)
xtable(BAR)
```

	X1	X2
G1_3	2.500000	2.500000
G4_6	1.000000	6.000000
G2_5	6.750000	4.000000
G	3.416667	4.166667

```r
nbar         <- nrow(BAR)

# distance au carrée entre les différents barycentres et le barycentre global
ecarts       <- (as.matrix(dist(BAR))^2)[nbar,1:nbar-1]

#effectifs de chaque classe
effectifs    <- c(2,2,2)

Inertie.inter <- sum(ecarts*effectifs)/sum(effectifs)

D3           <- distances(data.tp1[c(4,6),])

temp.3       <- (D3^2)[nrow(D3),1:nrow(D3)-1] # Vecteur de distance entre
                                              # w_4, w_6 et leur barycentre

vect.sum.dist <- rbind(mean(temp.1),mean(temp.2),mean(temp.3))

Inertie.Intra <- sum(vect.sum.dist*effectifs)/sum(effectifs)

Intertie.exp  <- (1-(Inertie.Intra/Inertie.totale))*100
```

le barycentre de la nouvelle classe à pour coordonnées: (1, 6).

L'inertie inter-classe vaut: 7.99.

L'inertie intra-classe vaut: 0.77.

Le pourcentage d'inertie expliquée est : 91.2%.