CNS : Classification non supervisée - Clustering

Soit un ensemble d'individus $I = \{i = 1, \ldots, n\}$ de poids $\{p_1, \ldots, p_n\}$ sur lesquels on a observé des caractères qualitatifs et/ou quantitatifs, ou mesuré les distances deux à deux. Les données peuvent ainsi se présenter sous une multitude de formes, comme les tableaux rencontrés en ACP, AFC, ou ACM, les tableaux de distance...

La classification non supervisée désigne les méthodes ayant pour objectif de dresser ou de retrouver une typologie existante caractérisant un ensemble de n observations, à partir de p caractéristiques mesurées sur chacune des observations.

Table des matières

L'appartenance des observations à l'une des q populations n'est ainsi pas connue, le nombre q lui même est souvent inconnu. Dans les méthodes de partitionnement, comme la CAH ou les k-means, l'objectif est de constituer des classes d'éléments de I telles que :

- chaque classe soit composée d'individus semblables vis-à-vis des caractères (homogénéité intra-classes) ;
- les classes soient hétérogènes entre elles vis-à-vis des caractères (hétérogénéité inter-classes).

Dans un cadre euclidien, ces différents critères se résument à rechercher une partition maximisant l'inertie intra $\sum_{k=1}^{q} \sum_{i \in I(k)} p_i d^2(i, g_k)$ et minimisant l'inertie inter $\sum_{k=1}^{q} \pi_k d^2(g, g_k)$.

Il existe une large famille de méthodes dédiées à la classification non supervisée. On se limitera ici

- au partitionnement par k means,
- au partitionnement par la classification ascendante hiérarchique, CAH,
- au modèle de mélange.

Notations : On notera

- q le nombre de classes constituées,
- $k = 1, \ldots, q$ une classe parmi les q classes,
- $I(k)$ les indices des individus appartenant à la classe k,
- $\pi_k = \sum_{i \in I(k)} p_i$ le poids de la classe k,
- g_k le centre de gravité de la classe k,
- I_T, $I_{intra} = I_W$, $I_{inter} = I_B$ les inerties totales, intra et inter avec $I_T = I_W + I_B$

Les références suivantes peuvent compléter ce chapitre dont une partie en est tirée (agroparistech surtout) :

- Approche pragmatique de la classification : arbres hiérarchiques, partitionnements. J-P. Nakache, J. Confais. Technip 2004 (site internet)
- Finding Groups in Data : An Introduction to Cluster Analysis. Kaufman, L. and Rousseeuw, P.J. (1990). Wiley, New York.
- `http://www.jstatsoft.org/v01/i04`
- `http://www2.agroparistech.fr/IMG/pdf/ClassificationNonSupervisee-AgroParisTech.pdf`

1 Généralités

1.1 Dissimilarité, Similarité, distance,

Définition 1 (Dissimilarité)

Une dissimilarité est une fonction D qui à tout couple (i, i') associe une valeur dans $\mathbb{R}_+$ telle que :

- $D(i, i') = D(i', i)$ *(symétrie),*
- $D(i, i') = 0 \Rightarrow i = i'$ *(séparabilité).*

Définition 2 (distance)

Une distance est une fonction d qui à tout couple (i, i') associe une valeur dans $\mathbb{R}_+$ telle que :

- $d(i, i') = d(i', i)$ *(symétrie),*
- $d(i, i') = 0 \Rightarrow i = i'$ *(séparabilité),*
- $d(i, i') \leq d(i, i'') + d(i'', i')$ *(inégalité triangulaire).*

Remarque 1 *Une distance est ainsi aussi une dissimilarité. Elles d'autant plus grandes que les individus sont différents. A l'inverse, il est fréquent aussi d'utiliser un autre indicateur, appelé similarité, mesurant la ressemblance plutôt que la différence.*

Définition 3 (Similarité)

Une similarité est une fonction S qui à tout couple (i, i') associe une valeur dans $\mathbb{R}_+$ telle que :

- $S(i, i') = S(i', i)$ *(symétrie),*
- $S(i, i) \leq S(i, i')$.

Le choix de la distance est une question primordiale pour les méthodes exploratoires multivariées. Cést à cette étape qu'il est possible d'utiliser au mieux l'information a priori dont il dispose, afin de proposer une mesure pertinente de ressemblance ou dissemblance entre observations.

Remarque 2 *Plusieurs éléments interviennent ainsi dans le choix d'une distance :*

- *le choix des variables prises en compte,*
- *la normalisation des variables, en particulier en cas d'hétérogénéité,*
- *le choix du poids de chaque variable,*
- *et le choix de la distance ou dissimilarité.*

```
agri = data.frame(x=c(13.8,21.3,18.7,5.9,11.4,17.8,10.9,16.6,21,16.4,7.8,14),
                  y=c(2.7,5.7,3.5,22.2,10.9,6,14,8.5,3.5,4.3,17.4,2.3))
row.names(agri)=c('B','DK','D','GR','é','F','IRL','I','L','NL','P','UK')

round(dist(agri[1:5,], method="euclidean"),2)
        B     DK     D     GR
DK   8.08
D    4.96  3.41
GR  21.04 22.57 22.66
E    8.54 11.18 10.39 12.57
```

1.2 Exemples de distances, similarité, dissimilarité

- *une distance euclidienne quand les individus sont caractérisés par p variables quantitatives,* $d^2(i,i') = \sum_{j=1}^{p}(x_{ij} - x_{i'j})^2$
- *la similarité entre variables quantitatives peut être mesurées par la valeur absolue du coefficient de corrélation,* $S(j,j')\|r(j,j')|$,
- *la dissimilarité entre variables quantitative peut être mesurée par* $D^2(j,j') = 1 - r^2(j,j')$,
- *pour des profils, la distance du* χ^2,
- *pour des données binaires, en notant par 1 la présence d'un caractère et 0 son absence, utilisée en écologie par exemple, on note* $s_{i,i'}$ *le nombre de caractères communs,* $s_{i,0}$ *le nombre de caractères possédés par* i *et pas* i' *et* $s_{0,i'}$ *l'inverse. On dispose alors de différents indices de similarité :*
  - ⋆ *Indice de Jaccard :* $\dfrac{s_{i,i'}}{s_{i,i'} + s_{i,0} + s_{0,i'}}$
  - ⋆ *Indice de Occhiai :* $\dfrac{s_{i,i'}}{\sqrt{(s_{i,i'} + s_{i,0})(s_{i,i'} + s_{0,i'})}}$

Remarque 3 *: Il est souvent possible de passer d'une similarité à une dissimilarité, par exemple avec les indices de Jaccard en posant* $D(i,i') = 1 - S(i,i')$, $S(i,i) = 1$ *constant.*

1.3 Approche combinatoire

Disposant d'une collection de n objets, il est possible pour un nombre de classe fixé de calculer le nombre de partitions possibles, appelé nombre de Stirling $S_{n,q}$.

Proposition 1 (Nombre de Stirling $S_{n,q}$***)***
On dispose des propriétés suivantes :

- $S_{n,1} = S_{n,n} = 1$,
- $S_{n,n-1} = \dfrac{n(n-1)}{2}$,
- $S_{n,q} = S_{n-1,q-1} + qS_{n-1,q}$,
- $S_{n,q} = \dfrac{1}{q!}\sum_{l=1}^{q}\binom{l}{q}(-1)^{q-l}l^n$ *avec la formule d'inversion de Pascal,*
- $S_{n,q} \sim \dfrac{q^n}{n!}$ *quand* $n \to +\infty$

Il est alors possible de définir le nombre totale de partitions possibles en faisant varier q de 1 à n, on obtient le nombre de Bell B_n.

Proposition 2 (Nombre de Bell B_n)

On dispose des propriétés suivantes :

- $B_n = \sum_{q=1}^{n} S_{n,q}$,
- $B_n = B_0 + (n-1)B1 + \cdots + \binom{q}{n-1} B_q + \ldots B_{n-1}$, *avec* $B_0 = 1$,

Preuve :

Si à une partition, on peut associer un paramètre, en testant toutes les partitions, il serait alors possible de trouver la meilleure. La limite est liée à l'augmentation très forte de B_n avec n, par exemple $B_{10} = 115975$. Il serait impossible de tester toutes les possibilités pour des valeurs même relativement faibles de n. Les approches ne pourront être qu'algorithmique.

2 Méthode des centres mobiles

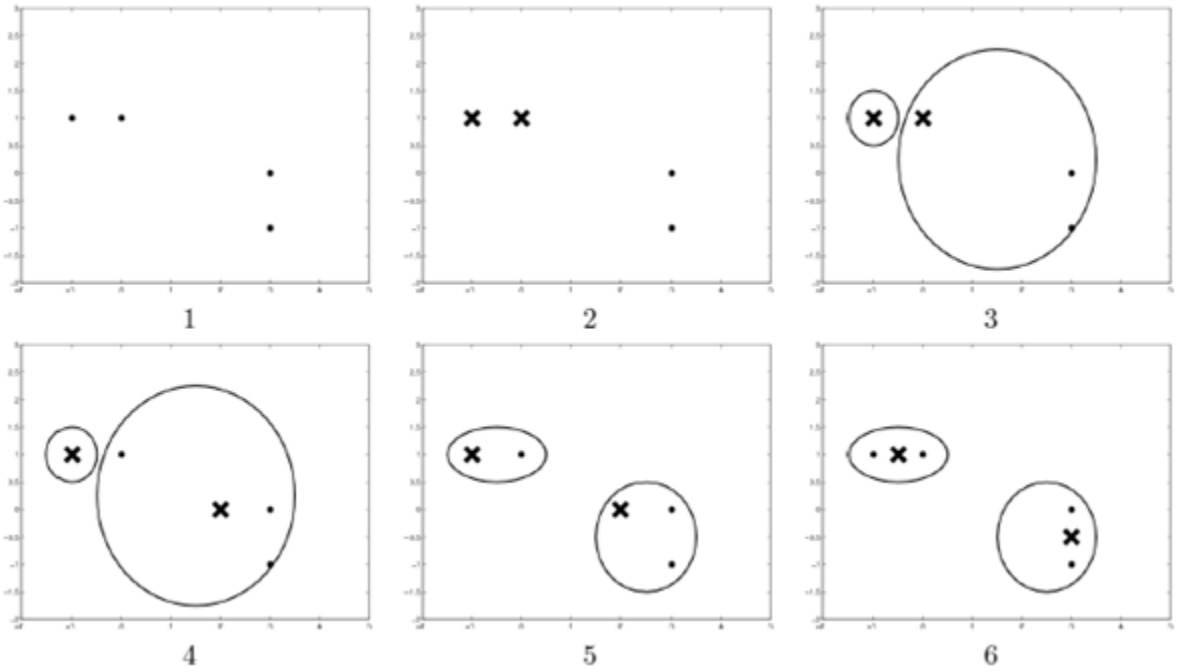
Pour utiliser la méthode des centres mobiles ou k−means, on suppose qu'il existe q classes et que les individus sont décrits par p variables quantitatives. Chaque individu peut être associé à un point de l'espace euclidien $\mathbb{R}^p$ disposant d'une distance d.

2.1 Principe

L'algorithme est le suivant :

- *On désigne q centres de classes parmi les n individus, notés $c_k^{(0)}$. Il est fréquent de choisir ces q centres initiaux par tirage au sort parmi les n individus, ou les choisir car représentatif de chaque classe a priori.*
- *Pour chaque individus, on l'associe au centre dont il est le plus proche. On construit ainsi une première partition notée $\{C_1^{(1)}, \ldots, C_q^{(1)}\}$ avec chaque classe $C_k^{(1)}$ définie par l'ensemble des individus dont $c_k^{(0)}$ est le centre le plus proche.*
- *On calcule alors le centre de gravité de chaque classe, noté $c_k^{(1)}$ et on recommence le partitionnement sur le même principe.*
- *L'algorithme s'arrête dès lors qu'un critère d'arrêt est satisfait comme la variation d'inertie intra ou que la partition n'évolue plus.*
- *Si au cours du processus une classe se vide, on lui affecte un individu au hasard.*

Expliquer les étapes suivantes



```
>data(iris)
>data=iris[,1:4]
>km=kmeans(data,3)
> names(km)
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
> km$centers
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     5.901613    2.748387     4.393548    1.433871
2     5.006000    3.428000     1.462000    0.246000
3     6.850000    3.073684     5.742105    2.071053
> km$withinss
[1] 39.82097 15.15100 23.87947
>fviz_cluster(km, data, ellipse.type = "norm")+
  theme_minimal()
```

2.2 Propriétés et limites

Proposition 3 (Convergence de l'algorithme) *L'inertie intraclasse de la suite des partitions* $(I^{(j)})$ $I_W^{(j)} = \sum_{k=1}^{q} \sum_{i \in C_k^{(j)}} p_i d^2(i, g_k^{(j)})$ *définies par l'algorithme décroît à chaque étape.*

La suite des partitions converge vers une partition stable dépendant du choix initial des centres initiaux correspondant à une partition optimale localement.

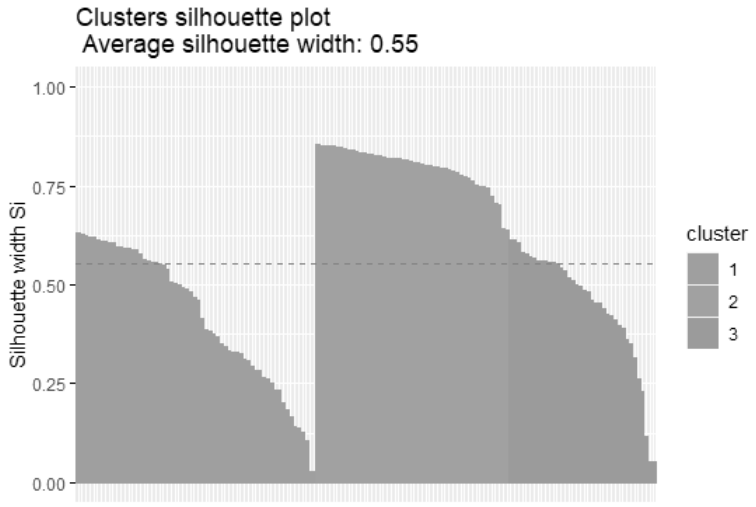
Preuve :

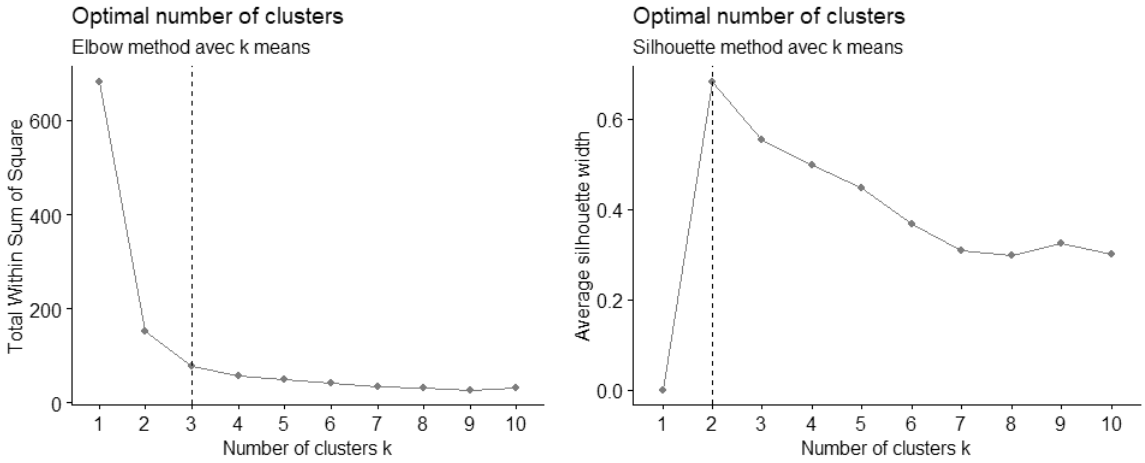
Remarque 4 *La première limite de cette méthode tient au choix du nombre q de classes. L'utilisation d'une CAH sur un échantillon peut permettre de préciser ce nombre, on parle alors de méthode mixte (CAH+ k-means). Il existe des méthodes pour déterminer le nombre de clusters idéal.*

- *La plus connu est la méthode du coude. On représente l'inertie intra finale en fonction du nombre de classe et on observe une décroissance forte au départ qui s'atténue ensuite.*
- *Le coefficient de silhouette se définit par* $s = \dfrac{b-a}{max(a,b)}$ *avec*
  - ⋆ *a la moyenne des distances aux autres individus du cluster (moyenne intra -cluster),*
  - ⋆ *b la moyenne au cluster le plus proche*

  e coefficient peut varier entre -1 et +1. Un coefficient proche de +1 signifie que l'observation est située bien à l'intérieur de son propre cluster, tandis qu'un coefficient proche de 0 signifie quélle se situe près d'une frontière ; enfin, un coefficient proche de -1 signifie que l'observation est associée au mauvais cluster. Comme pour l'inertie il est judicieux d'afficher l'évolution du coefficient en fonction du nombre de clusters comme ci-dessous

```
>fviz_silhouette(silhouette(km$cluster, dist(iris)))
>fviz_nbclust(data, kmeans, method = "wss") + geom_vline(xintercept = 3, linetyp
  labs(subtitle = "Elbow method avec k means")
>fviz_nbclust(data, kmeans, method = "silhouette")
```





Remarque 5 *La deuxième limite tient à la dépendance entre le résultat et le choix initial des centres. On peut répéter alors plusieurs fois l' algorithme et identifier des classes stables d'individus que l'on retrouve systématiquement ensemble appelées formes fortes. Il existe des fonctions permettant de comparer les partitions pour identifier les formes fortes.*

2.3 Méthodes voisines : nuées dynamiques ...

La méthode des k-means nécessite une structure euclidienne pour le calcul des centres de gravité et d'inertie.

Dans le cas contraire, on peut utiliser la méthode des nuées dynamiques reposant sur :

- *une dissimilarité D,*
- *une classe est représentée par un noyau pouvant être*
  - ⋆ *pour des données euclidiennes , un point, un sous-espace affine, on mesure la distance d'un point i au centre ou la distance du point i à sa projection dans l'espace affine.*
  - ⋆ *sinon un ensemble d'individus par classe, on mesure la distance par $D_k^2(i, C_k) = \sum_{l=1}^{n_k} p_{i_l} d^2(i, i_l)$, les points i_l étant les points du noyau de la classe C_k.*

Remarque 6 *Il existe d'autres méthodes voisines comme PAM, CLARA, FANNY... non développées ici.*

3 Classification ascendante hiérarchique

3.1 Hiérarchie et distance ultra-métrique

Définition 4 *Soit un ensemble I de n individus i et $\mathcal{H}$ un ensemble de parties de I. $\mathcal{H}$ définie une hiérarchie de I si*

- $I \in \mathcal{H}$
- $\forall i \in I,\ \{i\} \in \mathcal{H}$,
- $\forall\, h, h' \in \mathcal{H}^2, (h \cup h' \neq \emptyset) \Rightarrow (h \subset h' \ ou\ h' \subset h)$

Exemple 1 *Soit $I = \{a, b, c, d\}$. Construire 2 hiérarchies.*

Définition 5 *Soit un ensemble I de n individus i et $\mathcal{H}$ une hiérarchie de I. $\mathcal{H}$ est dite indicée s'il existe une application appelée indice $f : \mathcal{H} \to \mathbb{R}_+$ telle que*

- $\forall\, h \in \mathcal{H}, f(h) = 0 \Leftrightarrow h = \{i\}$
- $\forall\, h, h' \in \mathcal{H}^2, (h \subset h', h \neq h') \Rightarrow (f(h) < f(h')$

Exemple 2 *Construire un indice dans l'exemple précédent et représenter la hiérarchie sous forme de dendrogramme.*

Définition 6 *On appelle distance ultramétrique toute application $u : I \times I \to \mathbb{R}_+$ telle que*

- $u(i_1, i_2) = u(i_2, i_1)$,
- $u(i_1, i_2) = 0 \Leftrightarrow i_1 = i_2$,
- $u(i_1, i_2) \leq max(u((i_1, i_3), u(i_3, i_2))$

Remarque 7 *Tout triangle est isocèle et le petit côté est au plus égal au 2 côtés égaux.*

3.2 Principe

- *à l'étape initiale, les n individus constituent des classes à eux seuls.*
- *On calcule les distances deux à deux entre individus, et les deux individus les plus proches sont réunis en une classe.*
- *La distance entre cette nouvelle classe et les $n-2$ individus restants est ensuite calculée à l'aide d'un indice d'agrégation, et à nouveau les deux éléments (classes ou individus) les plus proches sont réunis.*
- *Ce processus est réitéré jusqu'à ce qu'il ne reste plus qu'une unique classe constituée de tous les individus.*

Il est ainsi nécessaire de définir deux distances ou dissimilarités :

- *la distance usuelle entre deux individus,*
- *et une distance entre classes.*

Il existe différentes méthodes de calcul de la distance/dissimilarité ou indice d'agrégation entre classes :

- *saut minimal "single" :* $\Delta_s(C_1, C_2) = \min_{i_1 \in C_1, i_2 \in C_2} d(i_1, i_2)$
- *saut maximal "complete" :* $\Delta_c(C_1, C_2) = \max_{i_1 \in C_1, i_2 \in C_2} d(i_1, i_2)$
- *saut moyen "UPGMA" :* $\Delta_u(C_1, C_2) = mean\{d(i_1, i_2), i_1 \in C_1, i_2 \in C_2\}$
- *indice de ward,* $\Delta_w = \left(I_{C_1} + I_{C2} - I_{C1 \cup C_2}\right)$ *variation de l'inertie intra par réunion des deux classes.*

Remarque 8 *Les individus sont appelés feuilles et les regroupements successifs noeuds, numérotés parfois* $C_{-(n+1)}$ *à* $C-(2n-1)$*. On représente cette CAH sous forme d'un dendrogramme (arbre, feuille, noeud, branche).*

Remarque 9 *: L'indice principalement utilisé est l'indice de Ward en étroite relation avec l'analyse factorielle. L'analyse conjointe de l'analyse factorielle et de la CAH permettent aux deux méthodes de s'enrichir en expliquant la nature des groupes ainsi constitués.*

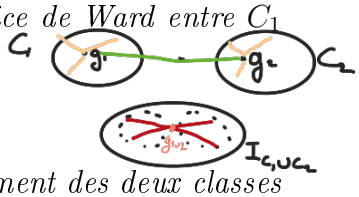
Remarque 10 *Les classifications ascendantes hiérachiques indicées obtenues peuvent être associées bijectivement à des distances ultramétriques.*

3.3 Indice de Ward - Propriétés et limites

Définition 7 (Indice de Ward) *Soit C_1, C_2, $C_1 \cup C_2$, des classes de poids respectifs p_1, p_2, p_1+p_2 et de centres de gravité g_1, g_2, $g_{1\cap 2} = \dfrac{p_1 g_1 + p_2 g_2}{p_1+p_2}$. L'indice de Ward entre C_1 et C_2 est :*

variation

$$\Delta_w(C_1, C_2) = \frac{p_1 p_2}{p_1+p_2} d^2(g_1, g_2).$$



Cet indice est égal à la variation d'inertie intra suite au regroupement des deux classes en une seule.

$I_{C_1 \cup C_2} = I_{C_1} + I_{C_2} + p_1 d^2(g_1, g_{C_1\cup C_2}) + p_2 d^2(g_2, g_{C_1\cup C_2})$

Intra

$\Delta(C_1, C_2)$ inter

Preuve :

$$\Delta(C_1,C_2) = p_1 \left\langle \frac{p_1 g_1 + p_2 g_2}{p_1+p_2} - g_1 , \frac{p_1 g_1 + p_2 g_2}{p_1+p_2} - g_1 \right\rangle + p_2 \left\langle \frac{p_1 g_1 + p_2 g_2}{p_1+p_2} - g_2 , \frac{p_1 g_1 + p_2 g_2}{p_1+p_2} - g_2 \right\rangle$$

$$= p_1 \left\langle p_2 \frac{g_2 - g_1}{p_1+p_2} , p_2 \frac{g_2 - g_1}{p_1+p_2} \right\rangle + p_2 \left\langle p_1 \frac{g_1 - g_2}{p_1+p_2} , p_2 \frac{g_1 - g_2}{p_1+p_2} \right\rangle$$

$$= \left(p_1 \frac{p_2^2}{(p_1+p_2)^2} + p_2 \frac{p_1^2}{(p_1+p_2)^2} \right) d^2(g_1, g_2)$$

$$= \frac{p_1 p_2}{p_1+p_2} d^2(g_1, g_2)$$

Remarque 11 *A chaque étape, on réunit ainsi les 2 classes conduisant à une augmentation minimale de l'inertie intra de la partition.*

Proposition 4 *La suite des indice d'agrégation servant au regroupement sont les variations d'inertie intra à chacune des étapes. On passe ainsi d'une inertie intra nulle à une inertie inter nulle. Les saut d'indice sont représentés sous forme d'éboulis et leur somme vaut l'inertie totale.*

$\forall i : p_i = 1$ →

Exemple 3 *Soit 4 points $A(1,1)$, $B(2,1)$ $C(3,3)$ et $D(4,4)$. Effectuer la CAH avec l'indice de Ward, représenter le dendrogramme et l'évolution de l'inertie intra et inter.*

d^2	B	C	D
A	1	8	18
B	0	5	13
C		0	2

$\times \frac{p_i p_j}{p_i + p_j} = \frac{1}{2}$

Δ	B	C	D
A	$\frac{1}{2}$	4	9
B	0	$\frac{5}{2}$	$\frac{13}{2}$
C		0	1

1°

$AB^2 = (x_B - x_A)^2 + (y_B - y_A)^2$

$C_{-5} = \{A, B\}$ avec $\Delta_{-5} = \frac{1}{2}$

$g_{-5}\left(\frac{3}{2}, 1\right)$ → le milieu de A et B

$\times \frac{1\times 2}{1+2} = \frac{2}{3}$ → car poids$(C_{-5}) = 2$

d^2	D	C_{-5}
C	2	25/4
D		61/4

Δ	D	C_{-5}
C	1	$\frac{25}{6}$
D		$\frac{61}{6}$

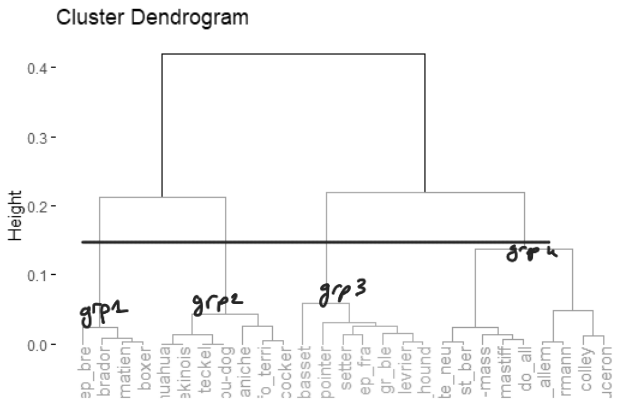
$C_{-6} = \{C, D\}$ avec $\Delta_{-6} = 1$

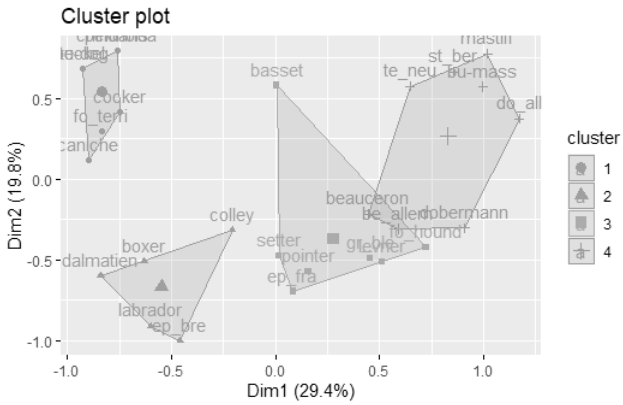
$g_{-6} = \left(\frac{7}{2}, \frac{7}{2}\right)$

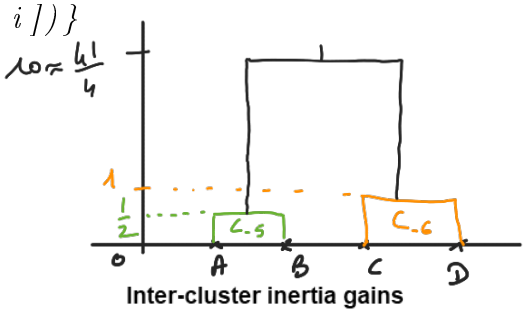
3.4 Etude d'un exemple avec FactomineR

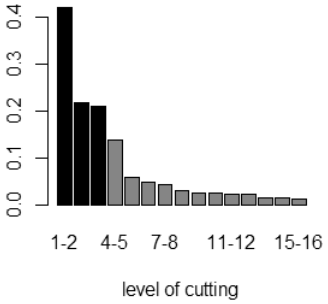
Reprenons l'exemple chiens vu en ACM.

```
data=read.table("chiens.txt",h=T)
for (i in 1:7) {data[,i]=as.factor(data[,i])}
acm=MCA(data,graph=FALSE)
cah=HCPC(acm,max=10,graph=FALSE)
fviz_dend(cah)
plot(cah, choice = "bar")
fviz_cluster(cah)
```









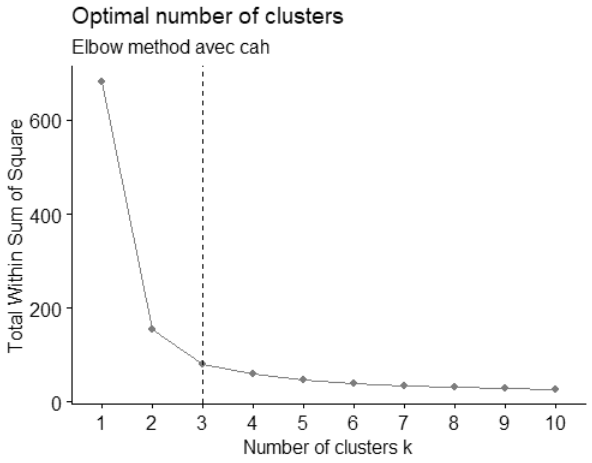
3.5 Choix du nombre de classes

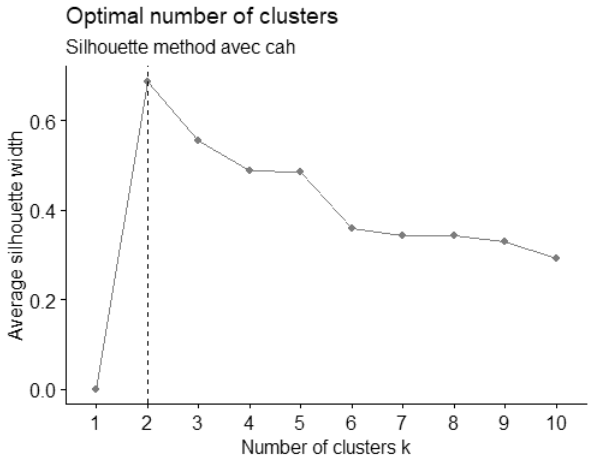
Concernant l'utilisation de l'algorithme des k-means, ou de l'algorithme CAH appliqué avec la distance de Ward, il est possible de tracer la courbe de l'inertie intra-classe I_W^k en fonction de k. On cherche alors à identifier les étapes où l'on observe une rupture dans cette courbe, synonyme d'une forte dégradation de l'inertie intra-classe. Cette dégradation résulte de la forte hétérogénéité des deux classes réunies lors de l'étape considérée, il est alors naturel de considérer un nombre de classes supérieur à celui pour lequel la rupture a lieu. Cette stratégie, parfois dénommée critère du coude, donne des résultats satisfaisants lorsqu'elle est appliquée à l'algorithme CAH où les partitions successives sont emboîtées. Lorsqu'appliquée à l'algorithme k-means (où les partitions successives ne sont pas emboîtées), l'identification des ruptures peut s'avérer plus difficile, rendant le critère du coude moins performant. Remarquons que l'objectif étant la mise en évidence de la structure sous-jacente des données, une méthode pragmatique pour le choix du nombre de classes est de choisir une partition dont il sera possible d'interpréter les classes.

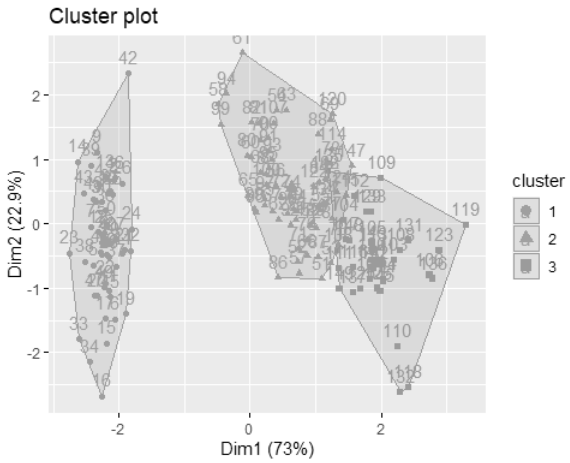
On retrouve les mêmes fonctions qu'avec k-means avec l'exemple iris.

```
>fviz_nbclust(data,hcut,method="silhouette")
```

```
>fviz_nbclust(data, hcut, method = "wss")
>cah=hcut(data, k = 3, hc_method = "ward")
>fviz_cluster(cah)
>fviz_silhouette(cah)
```







3.6 Caractérisation des groupes

Une fois les groupes établis, la caractérisation de ces groupes se fait à partir de plusieurs indicateurs.

- *Recherche des variables ou compoasantes principales les plus discriminantes.*

  Une analyse des variances des variables quantitatives ou des composantes principales avec le facteur groupe permet d'identifier celles jouant un rôle important. Cette analyse se résume à travers le calcul du coefficient de corrélation présenté en ACM associé au test F.

$$\eta^2(s) = \frac{Var(E(F_L^s|k))}{Var(F_L^s)}$$

```
> cah$desc.axes
Link between the cluster variable and the quantitative variables
================================================================
          Eta2      P-value
Dim.1 0.9146741 1.933064e-12
Dim.3 0.6940774 4.067031e-06
Dim.2 0.6339380 3.077434e-05
```

- *Recherches des variables/modalités les plus caractéristiques Une analyse des variances des variables quantitatives ou des composantes principales avec le facteur groupe permet d'identifier celles jouant un rôle important. Cette analyse se résume à travers le calcul du coefficient de corrélation présenté en ACM associé au test F.*

```
> cah$desc.var

Link between the cluster variable and the categorical variables (chi-sq
===========================================================================

            p.value df
fonction 1.738554e-08  6
poids    3.134227e-06  6
taille   1.550871e-05  6
affect   1.056919e-04  3
velocite 1.111861e-02  6

Description of each cluster by the categories
=============================================
$`1`
           Cla/Mod  Mod/Cl  Global      p.value   v.test
poids_1   87.50000 100.0  29.62963 9.008705e-06 4.439694
fonction_1 70.00000 100.0  37.03704 1.351306e-04 3.816915
taille_1  66.66667  85.7  33.33333 1.783724e-03 3.124063
```

- *Recherches des individus les plus caractéristiques.*

  Pour chaque groupe, on détermine les individus les plus repésentatifs par la distance la plus faible au centre de gravité. Pour chaque groupe formé, on appelle parangon l'individu dont les coordonnées sont les plus proches du centre de gravité du groupe. Le profil de cet individu caractérise alors le groupe auquel il appartient.

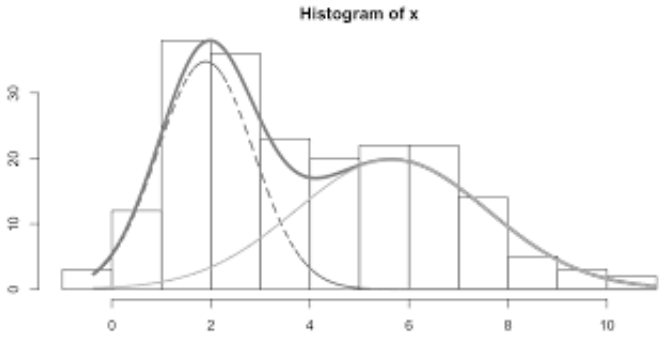
```
> cah$desc.ind
$para
Cluster: 1
 fo_terri     bu-dog     teckel chihuahua   pekinois
0.3939591 0.4251333 0.4251333 0.5182129 0.5182129
```

3.7 Méthodes mixtes

Il existe des cas où il peut être utile d'utiliser les deux algorithmes conjointement. Le premier cas est celui des grands jeux de données. L'algorithme de la CAH est un algorithme où les premières étapes sont les plus coûteuses, puisqu'elles nécessitent un grand nombre de calculs de distance. Pour de grands jeux de données, le temps de calcul de cet algorithme peut devenir prohibitif. On peut alors réduire le nombre d'individus initial en utilisant les k-means, par exemple pour passer de 1.000.000 d'individus à 10.000 classes, puis réaliser la CAH sur les centres des classes obtenues.La CAH permettant d'indentifier le nombre de classes pertinents. On reprend ensuite la méthode k-means. Le deuxième cas consiste à utiliser les k-means après la CAH : on constate qu'avec la CAH le classement de deux individus dans une même classe n'est plus remis en cause lors des étapes suivantes. Il peut alors être utile de réaliser la CAH, puis de permettre quelques réallocations des individus en faisant tourner l'algorithme des k-means en partant de la partition obtenue par CAH.

4 Méthode probabiliste : modèle de mélange

Le sens donné à l'homogénéité des groupes est ici différent : il ne se base plus sur des considérations géométriques mais s'appuie sur l'analyse de la distribution de probabilité de la population. Nous présentons ici les modèles les plus utilisés qui sont les modèles de mélanges de distributions. La notion d'homogénéité se traduit par le fait que les observations qui sont dans un même groupe sont issues d'une même distribution. On se limitera essentiellement au cas où le nombre de groupes q est connu a priori.



Dans ce chapitre, nous parlerons plutôt de populations que de groupes. Cette approche probabiliste présente deux avantages majeurs. D'une part, il permet d'avoir accès à des probabilités d'appartenance des individus aux différentes populations. C'est d'ailleurs à partir de ces probabilités que s'établit la classification. Il est en effet intéressant de disposer de ces probabilités en plus de la classification pour pouvoir par exemple comprendre le classement d'observations qui peut paraître suspect. D'autre part, le cadre formel de cette approche permet de proposer des solutions théoriques au problème du choix du nombre de populations, qui est en pratique inconnu. On peut en effet utiliser des critères classiques de sélection de modèles.

4.0.1 Modèle

On s'intéresse à n individus pour lesquels on dispose d'observations pour une variable x qui sont notées $\underline{x} = (x_1, \ldots, x_n)$. *On suppose qu'en réalité ces individus sont issus de q populations.*

Dans un premier temps, supposons connu ce nombre de populations. Du point de vue de la modélisation, on suppose que ces observations $\underline{x} = (x_1, \ldots, x_n)$ *sont des réalisations de n variables aléatoires, notées* $\underline{X} = (X_1, \ldots, X_n)$, *dont chacune est supposée être issue d'une distribution propre à la population à laquelle appartient l'individu associé. Pour le formaliser, on introduit une variable notée Z qui va servir de label pour chaque individu i à classer : à chaque* X_i *est associé un vecteur de dimension q, noté* $Z_i = (Z_{i_1}, \ldots, Z_{i_q})$ *tel que :*

$$Z_{ik} = \begin{cases} 1 & \textit{si l'individu } i \textit{ appartient à la population } k \\ 0 & \textit{sinon} \end{cases}$$

On note π_k *la probabilité que cette variable aléatoire prenne la valeur 1, c'est-à-dire la probabilité que l'individu appartienne à la population k :* $\pi_k = P(Z_{ik} = 1)$. *Cette probabilité est appelée probabilité a priori d'appartenance à la population k puisqu'elle ne prend pas en compte l'information dont on dispose, c'est-à-dire l'observation* x_i. *Elle représente donc tout simplement la probabilité qu'une observation prise au hasard appartienne à la population k. La somme des événements possibles vaut 1, c'est-à-dire que*

Cela revient à dire que les variables aléatoires Z_i *ont pour distribution une loi multinomiale de paramètres les probabilités a priori :*

$$Z_i \sim M(1; \pi_1, \ldots, \pi_q).$$

Une fois définie l'appartenance des individus aux populations, il s'agit de définir la distribution des observations dans chacune des populations : la distribution de X_i sachant que l'individu i appartient à la population k est notée

$$Xi|Z_{ik} = 1 \sim f_k(x_i),$$

où f_k est la distribution de probabilité attribuée à la population k.

Ici on se place dans un cadre paramétrique, c'est-à-dire que f_k est supposée appartenir à une famille de lois paramétrées :

$$f_k(.) = f(.; \theta_k),$$

où θ_k sont le(s) paramètre(s) de la distribution f dans la population k. Il faut donc choisir la famille à laquelle appartient f. Le choix de cette distribution aura des conséquences sur la classification finale. En pratique, ce choix se fait selon la même démarche que pour une modélisation plus classique : on peut s'aider de l'histogramme des observations comme des connaissances a priori que l'on a des observations.

On dispose donc de la loi du couple (X_i, Z_i) :

$$f(x_i, z_i) = P(z_i)f(x_i|z_i) = \pi_{z_i} f(x_i|z_i),$$

La distribution de X_i peut s'écrire sous la forme :

$$f(x_i; \phi) = P(Z_{i1} = 1) \times f(xi; \theta_1) + \cdots + P(Z_{iq} = 1) \times f(xi; \theta_q) = \sum_{k=1}^{q} \pi_k f(x_i; \theta_k) \qquad (4.1)$$

C'est ce qu'on appelle un mélange de distributions : c'est la somme des distributions des q populations pondérées par la taille de ces populations π_k (la proportion d'individus appartenant à ces populations). Ainsi dans ce modèle, chaque population k est caractérisée par

- *π_k qui représente la proportion d'individus appartenant à la population k,*
- *θ_k qui sont les paramètres de la distribution de la population k.*
- *soit $\phi_k = (\pi_k, \theta_k)$ avec $\phi = (\phi_1, \ldots, , \phi_k)$*

Les variables mises en jeu dans un modèle de mélange sont donc les X_i et Z_i, représentées sous forme d'un couple $Y_i = (X_i, Z_i)$, que l'on appelle données complètes. Cependant, seules les X_i sont observées et sont appelées données incomplètes. On souhaite donc reconstruire les variables d'appartenance Z_i.

4.0.2 Affectation

L'idée naturelle est de classer l'individu t dans la population dont il a le plus de chance d'être issu au vu de sa valeur x_i observée et des caractéristiques des populations. On s'intéresse donc à la probabilité que l'individu i appartienne à la population k sachant que l'on a observé pour cet individu la valeur x_i de la variable X. Cette probabilité est notée τ_{ik} :

$$\tau_{ik} = P(Z_{ik} = 1|X_i = x_i). \qquad (4.2)$$

Elle est appelée la probabilité a posteriori que l'individu soit dans la population k (contrairement à la probabilité a priori on prend en compte l'information dont on dispose). Par la formule de Bayes, elle s'écrit :

$$\tau_{ik} = \frac{P(Z_{ik} = 1)f(x_i; \theta_k)}{f(x_i)},$$

où $f(x_i)$ est donnée par (4.1). On obtient

$$\tau_{ik} = \frac{\pi_k f(x_i;\theta_k)}{\sum_{m=1}^{q} \pi_m f(x_i;\theta_m)} \tag{4.3}$$

Pour classer les individus, on utilise la règle du Maximum a posteriori (MAP) : on classe l'individu i dans la population k correspondant à la probabilité τ_{ik} maximale pour cet individu.

Si cette quantité est proche de 1 pour une population, on dira que l'individu est classé avec certitude dans cette population. Si par contre toutes les probabilités sont à peu près égales (par exemple 0.51 et 0.49 pour deux populations) alors il est plus difficile de classer avec certitude.

On peut remarquer que comme $f(x_i)$ ne dépend pas de la population, la règle du MAP consiste simplement à choisir la population k maximisant $\pi_k f(x_i;\theta_k)$. Ainsi, plus la population k est grande (valeur de π_k élevée), plus elle aura tendance à être attractive.

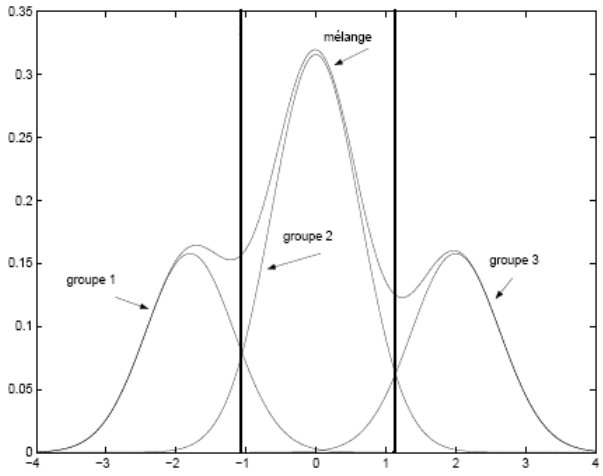
Pour pouvoir classer les individus dans les différentes populations, il faut connaître les caractéristiques de ces populations, à savoir les paramètres θ_k et π_k. On va donc chercher à les estimer.

4.0.3 Exemple d'un modèle de mélange de distributions gaussiennes

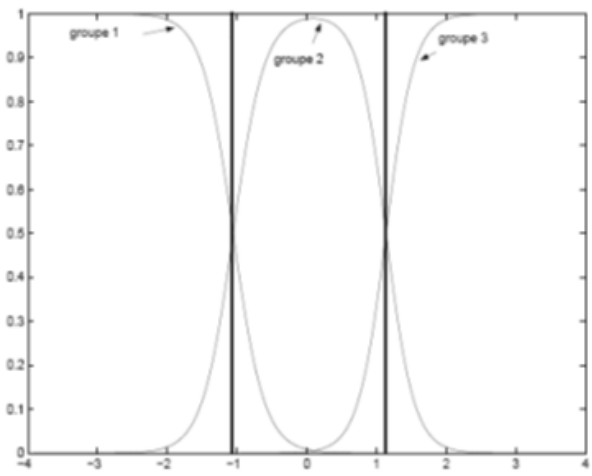
Bien que n'importe quelle loi puisse être utilisée pour modéliser les observations, la plus courante est la distribution gaussienne :

$$\theta_k = (\mu_k, \sigma_k^2)\ ,\ f_k(x;\theta_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} exp\big[-(x-\mu_k)^2/(2\sigma_k^2)\big].$$

Distribution
$f(x) = \pi_1 f(x;\theta_1) + \pi_2 f(x;\theta_2) + \pi_3 f(x;\theta_3)$



Probabilités a posteriori
$\tau ik = P(Z_{i,k} = 1|x)$



La figure de gauche représente la distribution du mélange avec en détail les distributions dans chaque population pondérées par leur proportion. Sur la figure de droite sont représentées les probabilités a posteriori d'appartenir aux 3 populations. Ce dernier graphique permet d'avoir une idée de la classification effectuée : les lignes verticales correspondent aux points frontières (aux valeurs de x pour lesquelles le classement change) : par exemple, tous les individus dont la valeur de x est inférieure à la valeur associée à la première ligne verticale (environ -1 sur la figure) seront classés dans la population 1. C'est d'ailleurs au niveau de ces lignes que le classement se fait avec beaucoup moins de certitude. En effet, c'est à ce niveau que les probabilités sont proches de 0.5. Comme on l'a vu dans le paragraphe précédent, la classification se fait à partir des probabilités $\tau ik = P(Z_{i,k} = 1|x)$ ou plus simplement à partir

des valeurs de $\pi_k f(x;\theta_k)$. C'est pourquoi on retrouve les points frontières sur la figure des distributions qui se situent au niveau du croisement des distributions pondérées.

4.0.4 Estimation des paramètres du modèle

Dans ce paragraphe, on cherche à estimer le paramètre ϕ du modèle par la méthode classique du maximum de vraisemblance. Cette méthode consiste à rechercher les valeurs des paramètres qui maximisent la vraisemblance (ou plutôt le logarithme de la vraisemblance) des données observées (c'est-à-dire des données incomplètes). Compte tenu de (4.1) *et de l'indépendance des observations, la vraisemblance s'écrit :*

$$\ell(\underline{X};\phi) = \prod_{i=1}^{n} f(X_i;\phi) = \prod_{i=1}^{n} \Big\{ \sum_{k=1}^{q} \pi_k f(X_i;\theta_k) \Big\},$$

et en passant au logarithme,

$$\ell\ell(\underline{X};\phi) = \sum_{i=1}^{n} \ln \Big\{ \sum_{k=1}^{q} \pi_k f(X_i;\theta_k) \Big\} \tag{4.4}$$

On cherche ensuite les valeurs des paramètres en annulant la dérivée de la logvraisemblance par rapport à tous ces paramètres. De par la forme particulièrement complexe de la log-vraisemblance (à cause du logarithme de la somme), on ne peut obtenir des expressions explicites de ces estimateurs. La solution est d'avoir recours à des algorithmes itératifs de recherche de maximum ou de minimum d'une fonction. Il existe plusieurs algorithmes qui ont cet objectif mais celui qui est utilisé dans le cadre des modèles de mélange est un algorithme appelé algorithme EM. Son succès tient dans le fait que cet algorithme est simple à mettre en oeuvre et qu'il mène à des formes explicites des estimateurs. Cet algorithme est présenté dans le paragraphe suivant.

4.0.5 Algorithme EM

Nous présentons dans ce paragraphe l'algorithme Expectation-Maximisation abrégé par EM. Puisque la vraisemblance des données incomplètes n'est pas simple à manipuler, l'idée est de travailler plutôt avec la vraisemblance des données complètes qui s'écrit :

$$\ell\ell(\underline{X},\underline{Z};\phi) = \ln \prod_{i=1}^{n} \prod_{k=1}^{q} \big[\pi_k f(X_i;\theta_k)\big]^{Z_{ik}} = \prod_{i=1}^{n} \prod_{k=1}^{q} Z_{ik} \ln \big[\pi_k f(X_i;\theta_k)\big] \tag{4.5}$$

Comme on le voit dans l'expression de la log-vraisemblance (4.5)*, les variables Z_{ik} apparaissent. Or ces variables ne sont pas observées puisque ce sont celles que l'on cherche à reconstruire. La stratégie consiste alors à remplacer les Z_{ik} non observés par la meilleure prédiction que l'on puisse en faire sachant les données observées X_i, donnée par $E(Z_{ik}|X_i = x_i) = \tau_{ik}$, dans l'expression* (4.5)*. On s'intéressera donc à la quantité suivante :*

$$\sum_{i=1}^{n} \sum_{k=1}^{q} \tau_{ik} \ln \big[\pi_k f(X_i;\theta_k)\big] \tag{4.6}$$

D'après (4.3)*, les probabilités a posteriori τ_{ik} dépendent des paramètres du modèle ϕ_k. Ainsi si on connaissait ces paramètres, on pourrait calculer les probabilités τ_{ik}, et inversement si on connaissait les probabilités τ_{ik}, on pourrait obtenir des valeurs des estimations des paramètres en maximisant la quantité précédente* (4.6)*. L'algorithme EM suit cette démarche. En effet, cet algorithme est un algorithme itératif qui, si on note $\phi^{(h)} = (\pi^{(h)}, \theta^{(h)})$ la valeur du paramètre courant, consiste en deux étapes à l'itération $(h+1)$:*

- **étape E (Estimation) :** *on calcule les probabilités a posteriori τ_{ik} à partir de la valeur courante du paramètre $\phi^(h)$:*

$$\tau_{ik}^{(h+1)} = \frac{\pi_k^{(h)} f(X_i; \theta_k^{(h)})}{\sum_{m=1}^{q} \pi_m^{(h)} f(X_i; \theta_m^{(h)})}$$

- **étape M (Maximisation) :** *on actualise les paramètres en maximisant la quantité donnée par (4.6) dans laquelle on a remplacé les τ_{ik} par les valeurs que l'on a obtenues à l'étape E :*

$$\phi^{(h+1)} = \underset{\phi}{Argmax} \sum_{i=1}^{n} \sum_{k=1}^{q} \tau_{ik}^{(h+1)} \ln \left[\pi_k f(X_i; \theta_k)\right].$$

Argmax signifie l'argument (valeur du paramètre ϕ) qui maximise la quantité d'intérêt. Il ne faut pas oublier qu'il existe une contrainte sur les proportions π_k, à savoir que $\sum_{k=1}^{q} \pi_k = 1$

Proposition 5 *L'estimateur de π_k à l'étape $(h+1)$ est :*

$$\pi_k^{(h+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(h+1)}}{n}.$$

Preuve :

$\phi^{(h+1)} = \text{Argmax} \underbrace{\sum_{i=1}^{n} \sum_{k=1}^{q} \tau_{ik}^{(h+1)} \ln\left[\pi_k f(X_i; \theta_k)\right]}_{a(\pi_k)}$

sous contrainte $\sum_{k=1}^{q} \pi_k = 1$ d'où $R(\pi_k) = \sum_{k=1}^{q} \pi_k - 1$

$\nabla a(\pi_k) - \lambda \nabla R(\pi_k) = 0$

$\sum_{i=1}^{n} \tau_{ik}^{(h+1)} \times \frac{1}{\pi_k} - \lambda = 0 \Rightarrow \pi_k = \frac{\sum_{i=1}^{n} \tau_{ik}^{(h+1)}}{\lambda}$

$\sum_{k=1}^{q} \sum_{i=1}^{n} \tau_{ik}^{(h+1)} = \lambda$ or $\sum_{k=1}^{q} \underbrace{\sum_{i=1}^{n} \tau_{ik}^{(h+1)}}_{1} = n = \lambda$

$\Rightarrow \pi_k = \frac{\sum_{i=1}^{n} \tau_{ik}^{(h+1)}}{n}$

Remarque 12 *Cette estimation a une interprétation naturelle : elle résume la contribution de chaque individu à la population k par leur probabilité a posteriori d'appartenance à cette population. Pour les paramètres θ_k, l'expression des estimateurs dépend de la distribution choisie. Notons cependant que ces estimateurs ne dépendent pas de la contrainte ajoutée*

4.0.6 Calculs pour un modèle de mélange de distributions gaussiennes

Proposition 6 *Les paramètres à estimer sont les moyennes et les variances de chaque population. à l'étape M de l'itération (h + 1), on obtient :*

$$\mu_k^{(h+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(h+1)} x_i}{\sum_{i=1}^{n} \tau_{ik}^{(h+1)}} \ , \ \sigma_k^{2\,(h+1)} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(h+1)} (x_i - \mu_k^{(h+1)})^2}{\sum_{i=1}^{n} \tau_{ik}^{(h+1)}}.$$

$$\phi_k(x_i, \theta_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} e^{\frac{-(x_i-\mu_k)^2}{2\sigma_k^2}}$$

Preuve :

$$a = \sum_{i=1}^{n} \sum_{k=1}^{q} \tau_{ik}^{(h+1)} \ln \pi_k \cdot \frac{1}{\sqrt{2\pi\sigma_k^2}} \cdot e^{\frac{-(x_i-\mu_k)^2}{2\sigma_k^2}}$$

$$\frac{\partial a}{\partial \mu_k} = \sum_{i=1}^{n} \tau_{ik} \left(\frac{2(x_i-\mu_k)}{2\sigma_k^2} \right) = \frac{1}{\sigma_k^2} \sum_{i=1}^{n} \tau_{ik} (\mu_k - x_i) = 0 \quad (\neq 0)$$

$$\Rightarrow \sum_{i=1}^{n} \tau_{ik}^{(h+1)} \mu_k = \sum_{i=1}^{n} \tau_{ik}^{(h+1)} x_i \Rightarrow \mu_k^{h+1} = \frac{\sum_{i=1}^{n} \tau_{ik}^{(h+1)} x_i}{\sum_{i=1}^{n} \tau_{ik}^{(h+1)}}$$

Remarque 13 *Ces estimations correspondent aux versions classiques des estimateurs des moyennes et variances mais pondérées par la probabilité a posteriori d'appartenance aux populations considérées.*

4.0.7 Propriétés de l'algorithme EM

Bien que l'algorithme EM ne consiste pas à maximiser directement la vraisemblance des données observées, il assure que l'on obtiendra des estimations du maximum de cette vraisemblance. Ce résultat est énoncé dans la propriété suivante qui montre que la log-vraisemblance des données incomplètes augmente à chaque itération de l'algorithme.

Proposition 7 *Soit une suite d'itérations de EM : $\phi^{(0)}, \ldots, \phi^{(h)}, \ldots$, on montre que la suite des $\ell\ell(\underline{X}; \phi^{(h)})$ est croissante*

Remarque 14 *Notons que cette valeur n'est pas forcément le maximum de vraisemblance mais peut correspondre à un maximum local.*

4.0.8 En pratique

Comme tous les algorithmes itératifs, l'algorithme EM nécessite

- **l'initialisation des paramètres** $\phi^{(0)}$ *ou des probabilités a posteriori* $\tau_{ik}^{(0)}$ *(dans le cadre particulier des modèles de mélange de distributions). En pratique, le plus simple est souvent de choisir ces probabilités. Si on ne dispose d'aucune information a priori, l'usage est de les choisir au hasard :* $\tau_{ik}^{(0)} = 1/q$.
- **une règle d'arrêt** *: on s'arrête quand les valeurs des paramètres ou de la quantité à maximiser entre deux itérations successives ne varie presque plus.*

Remarque 15 *Bien que cet algorithme soit très performant et souvent simple à mettre en oeuvre, son principal problème est sa forte dépendance aux valeurs initiales : pour différentes initialisations, on obtient différentes valeurs des estimations des paramètres, ce qui le rend peu stable. La raison de cette sensibilité est que l'algorithme converge vers des maxima locaux de la vraisemblance. Pour s'affranchir de ce problème, une solution consiste :*

- *soit à lancer l'algorithme à partir de plusieurs valeurs initiales et à retenir la meilleure,*
- *soit à lancer l'algorithme un grand nombre de fois à partir de plusieurs valeurs initiales, à moyenner les valeurs obtenues pour les probabilités a posteriori ou pour les paramètres (suivant le mode d'initialisation choisi), puis à lancer une dernière fois l'algorithme EM en utilisant les valeurs moyennes comme valeurs initiales.*

Remarque 16 *La procédure précédente permet d'obtenir une classification des n individus en q populations. Cependant, en pratique le nombre de populations est inconnu. Même si l'on dispose d'informations a priori sur les données, il est difficile de le fixer à l'avance. Il faut donc l'estimer. L'avantage du cadre probabiliste des modèles de mélange par rapport aux méthodes exploratoires présentées dans le chapitre précédent est que l'on dispose de critères théoriques pour choisir ce nombre. Ces critères sont appelés critères pénalisés.*

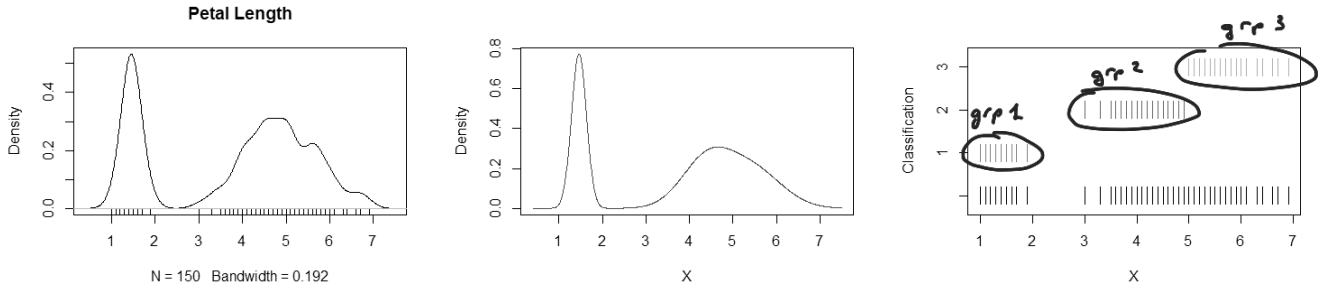
4.0.9 Etude d'un exemple avec R et mclust

On reprend l'exemple iris, en ne considérant dans un premier temps que la variable petal.Length.

On impose dans un premier temps un nombre $q = 3$ groupes.

```
data(iris)
X=iris$Petal.Length
plot(density(X,bw="SJ"),main="Petal Length")
rug(X)
install.packages("mclust")
library(mclust)
em = Mclust(X,3)
summary(em)
plot(em)
> em$parameters
$pro
[1] 0.3333221 0.3375687 0.3291092
$mean
       1        2        3
1.461986 4.386521 5.438728
$variance
$variance$sigmasq
[1] 0.02955091 0.33668292 0.46123150
> em$classification
  [1] 1 1 1 1 1 1 1
> em$uncertainty
  [1] 5.496119e-07 5.496119e-07
  [3] 3.247634e-07 1.266449e-06
  [5] 5.496119e-07 1.697349e-05
```

Distribution initiale *Distribution de modèle* *Classification*



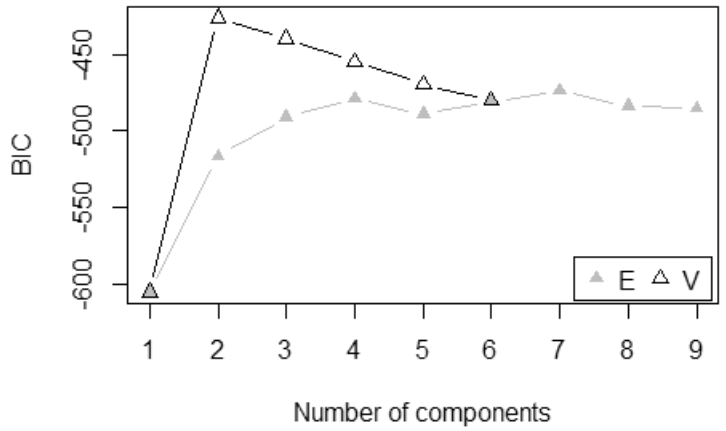
Supposons maintenant le nombre q de groupes inconnus. La fonction Mclust nous propose alors le nombre le plus vraisemblable en comparant les modèles avec différentes valeurs de q et gardant le plus vraisemblable, avec le critère BIC par exemple.

```
> em = Mclust(X)
> summary(em)
```

```
Gaussian finite mixture model fitted by EM algorithm
```

```
Mclust V (univariate, unequal variance)
model with 2 components:
log-likelihood   n  df        BIC        ICL
```

```
-200.5788 150 5 -426.2107 -426.2777
Clustering table:
 1  2
50 100
```
répartition dans les groupes

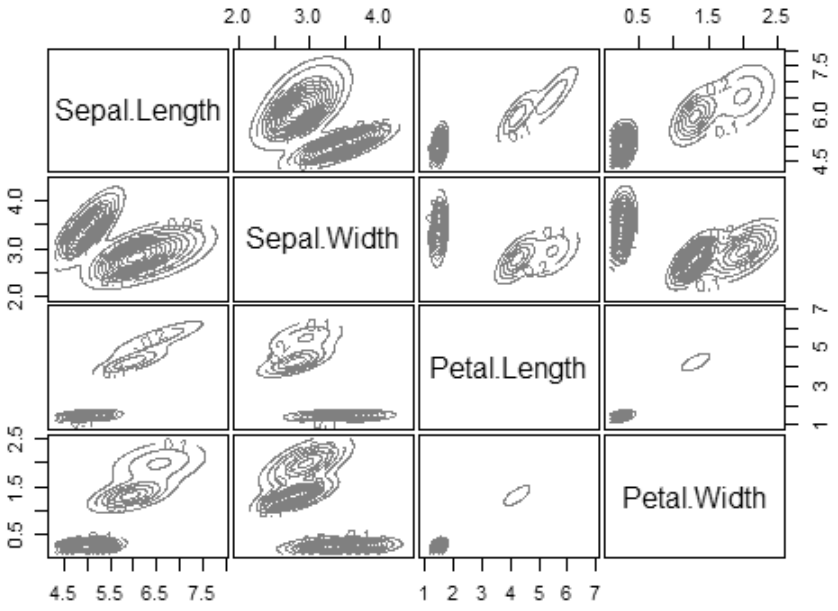


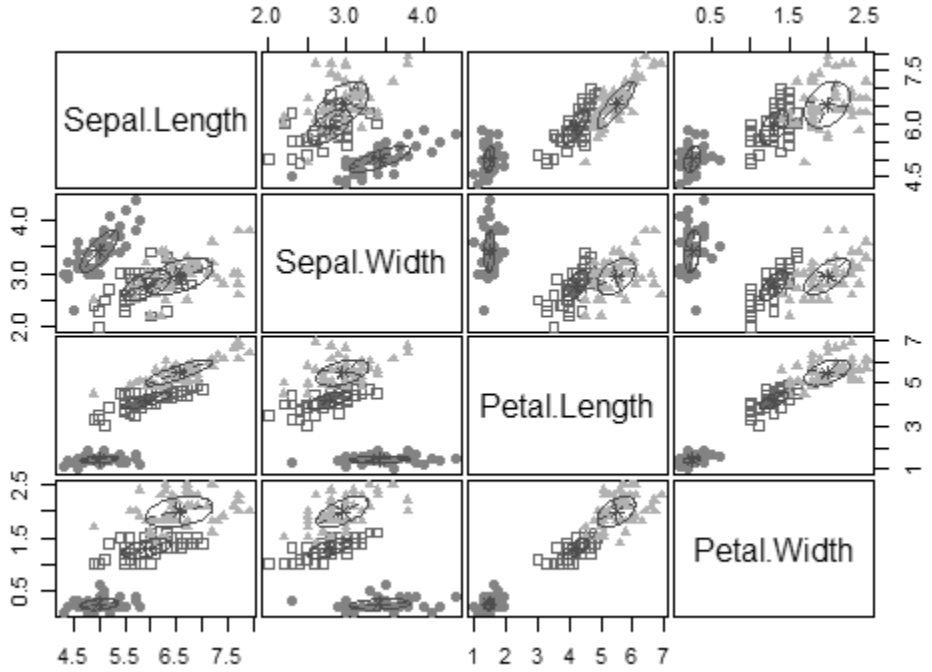
Reprenons maintenant les 4 variables en fixant $q = 3$.

```
> X=iris[,1:4]
> em=Mclust(X,3)
> summary(em)
Mclust VEV (ellipsoidal, equal shape)
model with 3 components:
 log-likelihood   n  df      BIC      ICL
       -186.074 150  38 -562.5522 -566.4673
Clustering table:
 1  2  3
50 45 55
```
nb de groupe.

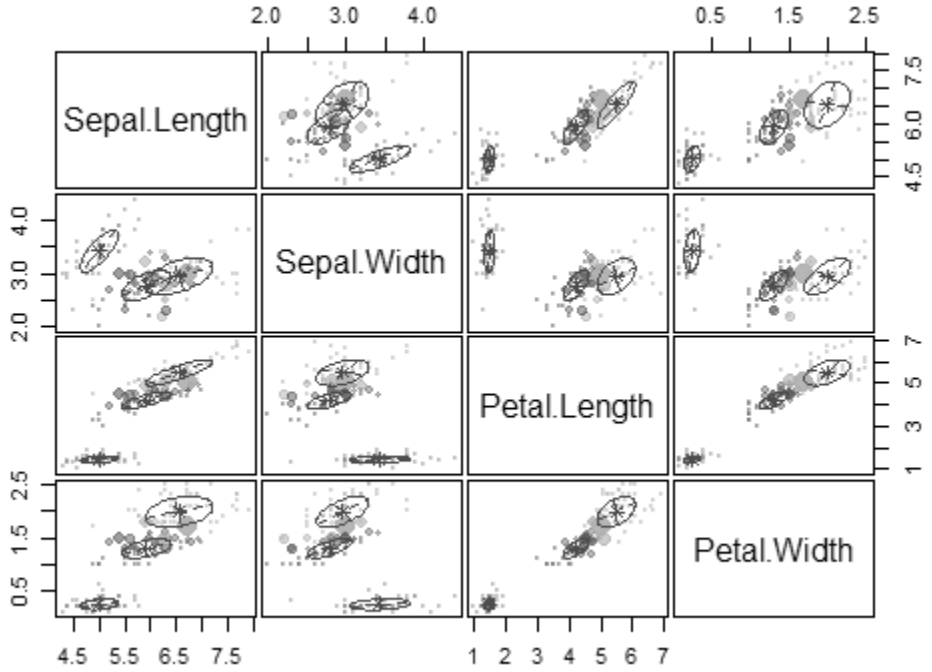
le bic d'avant est plus petit (abs) donc le 2 est meilleur comme nb de grp

Classification *Densité*

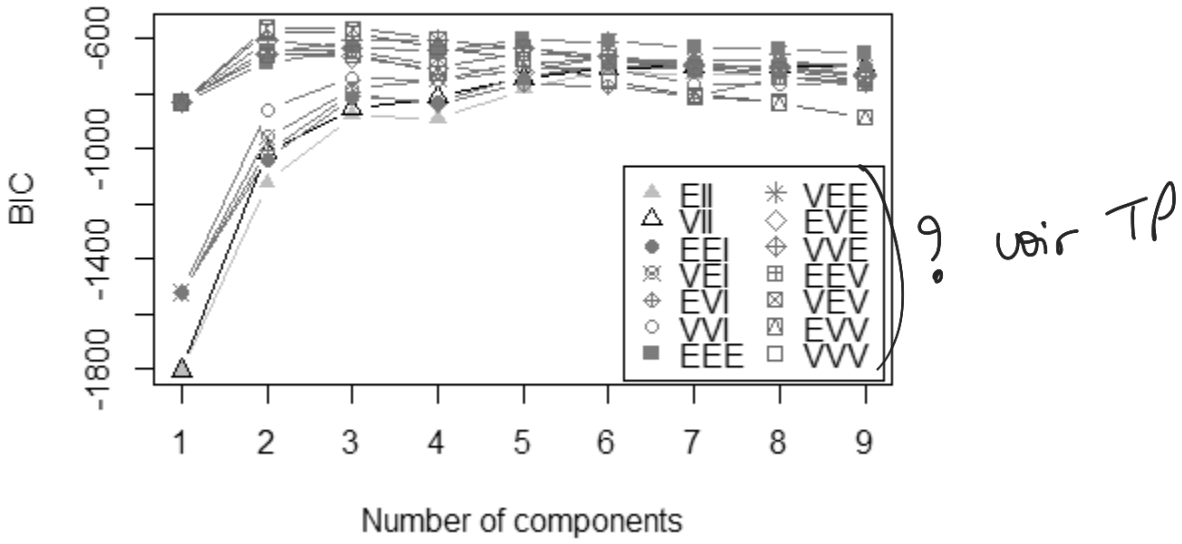




Classification flou



newpage Reprenons maintenant les 4 variables en ne fixant plus q.



Le choix porterait sur 2 ou 3 groupes.