Chapitre II
Classification automatique

Licence 3 MIASHS - Université de Bordeaux

Marie Chavent

Introduction

Comment définir des groupes d'individus ou de variables qui se ressemblent ?

Exemple : données quantitatives décrivant 8 eaux minérales sur 13 variables.

```
load("eaux.RData")
print(data[,1:5])
```

```
##            saveur.amère saveur.sucrée saveur.acide saveur.salée saveur.alcaline
## St Yorre            3.4           3.1          2.9          6.4             4.8
## Badoit              3.8           2.6          2.7          4.7             4.5
## Vichy               2.9           2.9          2.1          6.0             5.0
## Quézac              3.9           2.6          3.8          4.7             4.3
## Arvie               3.1           3.2          3.0          5.2             5.0
## Chateauneuf         3.7           2.8          3.0          5.2             4.6
## Salvetat            4.0           2.8          3.0          4.1             4.5
## Perrier             4.4           2.2          4.0          4.9             3.9
```

- A partir des distances entre individus : quelle mesure de distance ?
- A partir des liaisons entre les variables : quelle mesure de liaison ?

Dépend de la nature des données : quantitatives, qualitatives ou mixtes.

A partir des distances euclidiennes entre individus ?

```
print(dist(data),digit=2)
```

```
##             St Yorre Badoit Vichy Quézac Arvie Chateauneuf Salvetat
## Badoit           4.1
## Vichy            7.9    4.8
## Quézac           2.9    5.3   9.7
## Arvie            3.0    1.8   5.5    4.7
## Chateauneuf      2.9    1.8   5.7    4.3   1.3
## Salvetat         4.0    1.2   5.4    4.9   1.8         1.6
## Perrier          8.2   10.6  14.7    6.2  10.1         9.9      10.3
```

A partir des corrélations entre les variables ?

```
print(cor(data[,1:5]),digit=2)
```

```
##                 saveur.amère saveur.sucrée saveur.acide saveur.salée saveur.alcaline
## saveur.amère            1.00         -0.83         0.78        -0.67           -0.96
## saveur.sucrée          -0.83          1.00        -0.61         0.49            0.93
## saveur.acide            0.78         -0.61         1.00        -0.44           -0.82
## saveur.salée           -0.67          0.49        -0.44         1.00            0.56
## saveur.alcaline        -0.96          0.93        -0.82         0.56            1.00
```

On applique une méthode de classification automatique.

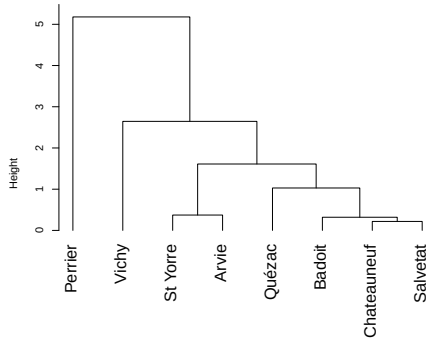
Partition en 4 classes des individus.

```
##             P4
## St Yorre     1
## Badoit       2
## Vichy        3
## Quézac       2
## Arvie        1
## Chateauneuf  2
## Salvetat     2
## Perrier      4
```

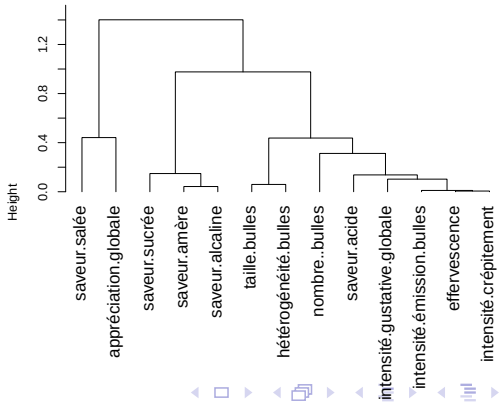
Partition en 3 classes des variables.

```
##                           P3
## saveur.amère               1
## saveur.sucrée              1
## saveur.acide               2
## saveur.salée               3
## saveur.alcaline            1
## appréciation.globale       3
## intensité.émission.bulles  2
## nombre..bulles             2
## taille.bulles              2
## hétérogénéité.bulles       2
## effervescence              2
## intensité.gustative.globale 2
## intensité.crépitement      2
```

Hiérarchie des individus



Hiérarchie des variables



Il existe de nombreux algorithmes de classification automatique qui se distinguent par :

- la nature des objets à regrouper : les individus ou les variables,
- la nature des données : quantitatives, qualitatives ou mixtes,
- la nature de la structure de classification : partition ou hiérarchie,
- la nature de l'approche utilisée : approche géométrique (distance, dissimilarité, similarité) ou approche probabiliste (modèles de mélange).

Dans ce chapitre, on s'intéresse à la classification d'individus décrits par des données quantitatives, à l'aide d'approches géométriques utilisant les distances.

Plan

On considère un ensemble $\Omega = \{1, ..., i, ..., n\}$ de n individus décrits par p variables quantitatives dans une matrice $\mathbf{X}$:

$$\mathbf{X} = \begin{array}{c} \\ 1 \\ \vdots \\ i \\ \vdots \\ n \end{array} \begin{array}{c} \begin{array}{ccc} 1 \cdots & j & \cdots p \end{array} \\ \begin{bmatrix} & \cdot & \\ & \vdots & \\ \cdots & x_{ij} \in \mathbb{R} & \cdots \\ & \vdots & \\ & \cdot & \end{bmatrix} \end{array}.$$

- Un individu $i \in \Omega$ est décrit par un vecteur $\mathbf{x}_i \in \mathbb{R}^p$ (ligne $\mathbf{X}$).
- Un poids w_i est associé à chaque individu i. On prendra souvent $w_i = \frac{1}{n}$.

On dispose comme en ACP, d'un nuage pondérés de n points-individus de $\mathbb{R}^p$.

Distance, dissimilarité ou similarité ?

Un indice de similarité $s : \Omega \times \Omega \to \mathbb{R}^+$ vérifie $\forall i, i' \in \Omega$:

$$\begin{aligned} s(i, i') &\geq 0, \\ s(i, i') &= s(i', i), \\ s(i, i) &= s(i', i') = smax \geq s(i, i') \end{aligned}$$

Exemple : lorsque les données sont binaires, on peut construire le taleau croisé entre deux individus i et i' :

		individu i'	
		1	0
individu i	1	a	b
	0	c	d

Il existe alors plusieurs indices de similarité normalisés ($smax = 1$) :

$$\text{Jaccard} : \frac{a}{a+b+c} \qquad \text{Russel et Rao} : \frac{a}{2a+b+c+d}$$

$$\text{Dice ou Czekanowski} : \frac{2a}{2a+b+c} \qquad \text{Ochiai} : \frac{a}{\sqrt{a+b}\sqrt{a+c}}$$

Un indice de dissimilarité $d : \Omega \times \Omega \to \mathbb{R}^+$ vérifie :

$$d(i, i') \geq 0, \quad d(i, i') = d(i', i), \quad d(i, i) = 0$$

Remarque : il est facile de transformer un indice de similarité s en un indice de dissimilarité d en posant :

$$d(i, i') = smax - s(i, i')$$

Une distance est une dissimilarité qui vérifie en plus l'inégalité triangulaire :

$$d(i, j) \leq d(i, k) + d(k, j) \ \forall i, j, k \in \Omega.$$

Distances classiques entre deux vecteurs $\mathbf{x}_i$ et $\mathbf{x}_{i'}$ de $\mathbb{R}^p$:

- distance euclidienne simple :

$$d^2(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

- distance euclidienne normalisée :

$$d^2(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^{p} \frac{1}{s_j^2} (x_{ij} - x_{i'j})^2,$$

où ${s_j}^2 = \frac{1}{n} \sum_{i=1}^{n} (x_{ij} - \bar{x}^j)^2$ et $\bar{x}^j = \frac{1}{n} \sum_{i=1}^{n} x_{ij}$.

- distance de city-block ou de Manhattan : $d(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^{p} \mid x_{ij} - x_{i'j} \mid$
- distance de Chebychev ou du max : $d(\mathbf{x}_i, \mathbf{x}_{i'}) = \max_{j=1,\ldots,p} \mid x_{ij} - x_{i'j} \mid$

- En général, on utilise la distance euclidienne simple lorsque toutes les variables ont la même échelle de mesure. En effet, si une variable a une variance bien plus forte, la distance euclidienne simple va accorder beaucoup plus d'importance à la différence entre les deux individus sur cette variable qu'à la différence entre les deux individus sur les autres variables.
- Dans le cas d'échelles de mesures trop différentes, il est préférable d'utiliser la distance euclidienne normalisée afin de donner la même importance à toutes les variables. Cela revient à calculer la distance euclidienne simple sur les données standardisées (centrées-réduites).

Exercice : Démontrer (de nouveau) ce résultat.

Partition ou hiérarchie ?

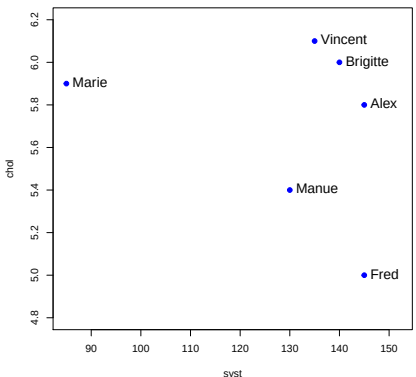
Une partition P_K de Ω en K classes est un ensemble $(C_1, \ldots, C_k, \ldots, C_K)$ de classes non vides, deux à deux disjointes et dont la réunion forme Ω :

$$C_k \neq \emptyset \; \forall k \in \{1, \ldots K\},$$
$$C_k \cap C_{k'} = \emptyset \; \forall k \neq k'$$
$$C_1 \cup \ldots \cup C_K = \Omega$$

Exemple : si $\Omega = \{1, \ldots, 7\}$, $P_3 = (C_1, C_2, C_3)$ avec $C1 = \{7\}$, $C_2 = \{5, 4, 6\}$ et $C_3 = \{1, 2, 3\}$ est une partition en trois classes de Ω.

Exercice : proposer une partition en 3 classes des 6 individus ci-dessous.

```
##          syst chol
## Brigitte  140  6.0
## Marie      85  5.9
## Vincent   135  6.1
## Alex      145  5.8
## Manue     130  5.4
## Fred      145  5.0
```



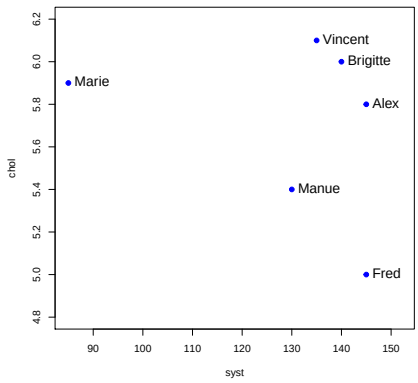
Une hiérarchie H de Ω est un ensemble de classes non vides (appelées paliers) qui vérifient :

- $\Omega \in H$,
- $\forall i \in \Omega,\ \{i\} \in H$ (la hiérarchie contient tous les singletons),
- $\forall A, B \in H, A \cap B \in \{A, B, \emptyset\}$ (deux classes de la hiérarchie sont soit disjointes soit contenues l'une dans l'autre).

Exemple : $H = \{\{1\}, \ldots, \{7\}, \{4,5\}, \{2,3\}, \{4,5,6\}, \{1,2,3\}, \{4,5,6,7\}, \Omega\}$.

Exercice : proposer une hiérarchie des 6 individus ci-dessous.

```
##          syst chol
## Brigitte  140  6.0
## Marie      85  5.9
## Vincent   135  6.1
## Alex      145  5.8
## Manue     130  5.4
## Fred      145  5.0
```



Inertie totale, intra ou inter ?

On note :

$$\mu_k = \sum_{i \in C_k} w_i \quad \text{le poids de } C_k$$

$$\mathbf{g}_k = \frac{1}{\mu_k} \sum_{i \in C_k} w_i \mathbf{x}_i \quad \text{le centre de gravité de } C_k$$

$$d^2(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2 \quad \text{la distance entre deux individus}$$

L'inertie totale T du nuage des n individus est

$$T = \sum_{i=1}^{n} w_i d^2(\mathbf{x}_i, \mathbf{g})$$

où $\mathbf{g}$ est le centre de gravité du nuage des individus.

L'inertie totale est indépendante de la partition.

L'inertie inter-classe B de la partition P_K est l'inertie des centres de gravité des classes pondérées par μ_k et mesure donc la séparation des classes.

$$B = \sum_{k=1}^{K} \mu_k d^2(\mathbf{g}_k, \mathbf{g})$$

L'inertie intra-classe W de la partition P_K est la somme des inerties des classes et mesure donc l'hétérogénéité des classes.

$$W = \sum_{k=1}^{K} I(C_k)$$

$$I(C_k) = \sum_{i \in C_k} w_i d^2(\mathbf{x}_i, \mathbf{g}_k)$$

Une bonne partition a une inertie inter-classe grande et une inertie intra-classe petite.

Remarques

L'inertie totale, intra-classe et inter-classe est aussi appellée

- variance intra et inter-classe quand $w_i = \frac{1}{n}$.
- somme des carrés intra et inter-classe quand $w_i = 1$.

De plus,

- Quand $w_i = \frac{1}{n}$, $T = s_1^2 + \ldots + s_p^2$ où s_j^2 est la variance empirique de la $j$ème variable.
- Quand $w_i = \frac{1}{n}$ et les données sont standardisées, $T = p$.

Les inerties (total, intra et inter) calculées sur les données standardisées sont égales aux inerties calculées sur le tableau des p composantes principales de l'ACP.

On a la relation fondamentale suivante :

$$T = W + B$$

Minimiser l'inertie intra-classe c'est à dire l'hétérogénéité des classes est équivalent à maximiser l'inertie inter-classe, c'est à dire la séparation entre les classes.

Cette relation se déduit du théorème de Huygens (à démontrer) :

$$\forall \mathbf{a} \in \mathbb{R}^p,\ I_{\mathbf{a}} = I_{\mathbf{g}} + (\sum_{i=1}^{n} w_i) d^2_{\mathbf{M}}(\mathbf{a}, \mathbf{g});$$

où $\mathbf{M}$ est une métrique de $\mathbb{R}^p$ et $I_{\mathbf{a}}$ est l'inertie par rapport à un point $\mathbf{a} \in \mathbb{R}^p$ définie par :

$$I_{\mathbf{a}} = \sum_{i=1}^{n} w_i d^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{a}).$$

Le pourcentage de l'inertie totale expliquée par une partition P_k est :

$$(1 - \frac{W}{T}) \times 100$$

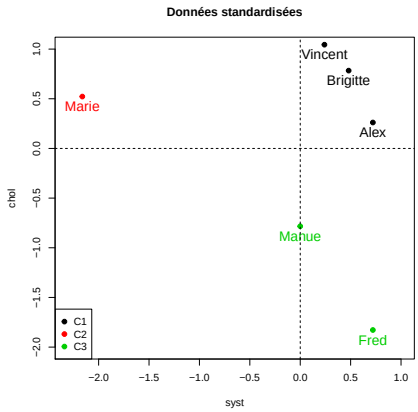
Ce critère varie entre zero et cent. Il vaut :

- 100 pour la partition en n classes des singletons,
- 0 pour la partition réduite à une classe Ω.

Ce critère augmente quand le nombre de classes K augmente. Il ne permet donc que de comparer deux partitions ayant le même nombre de classe : si le pourcentage d'inertie expliquée d'une partition est supérieur au pourcentage d'inertie expliquée d'une autre partition ayant le même nombre de classe, on considérera que la première partition est meilleure que la seconde, au sens du critère d'inertie.

Exercice : calculer l'inertie totale, intra-classe et inter-classe de la partition ci-dessous (avec les poids $w_i = \frac{1}{n}$). En déduire le pourcentage d'inertie expliquée.

```
##           syst  chol
## Brigitte  0.48  0.78
## Marie    -2.16  0.52
## Vincent   0.24  1.04
## Alex      0.72  0.26
## Manue     0.00 -0.78
## Fred      0.72 -1.83
## g1        0.48  0.70
## g2       -2.16  0.52
## g3        0.36 -1.31
```




```
## [1] "distances au carré"
##         Brigitte   Marie Vincent    Alex   Manue    Fred      g1      g2
## Marie     7.0490
## Vincent   0.1259  6.0420
## Alex      0.3304  8.3759  0.8444
## Manue     2.6853  6.3776  3.3986  1.6101
## Fred      6.8759 13.8304  8.4808  4.3636  1.6101
## g1        0.0076  7.0111  0.1789  0.2471  2.4202  6.4289
## g2        7.0490  0.0000  6.0420  8.3759  6.3776 13.8304  7.0111
## g3        4.3781  9.7015  5.5372  2.5844  0.4025  0.4025  4.0220  9.7015
```


Plan

Une bonne partition de Ω possède des classes

- homogènes : les individus dans une même classe se ressemblent,
- séparées : les individus de deux classes différentes ne se ressemblent pas.

Soit

$$\mathcal{H} : \mathcal{P}_K(\Omega) \to \mathbb{R}^+$$

un critère qui mesure l'hétérogénéité d'une partition P_K.

Par exemple le diamètre d'une partition est le plus grand diamètre de ses classes :

$$\mathcal{H}(P_K) = \max_{k=1,\dots,K} \max_{i,i' \in C_k} \underbrace{d(\mathbf{x}_i, \mathbf{x}_{i'})}_{\text{diam}(C_k)}.$$

Ce critère se calcule avec une matrice de dissimilarité.

Un autre critère d'hétérogénéité d'une partition est l'inertie intra-classe (aussi notée W) :

$$\mathcal{H}(P_K) = \sum_{k=1}^{K} I(C_k)$$

Ce critère se calcul avec un tableau de données quantitatives.

Ce critère est très populaire car avec ce critère des classes homogènes (petite inertie intra-classe W) correspondent à des classes bien séparées (grand inertie inter-classe B) d'après la relation fondamentale $T = W + B$.

L'objectif des méthodes de partitionnement est donc de trouver la partition P_K de Ω qui minimise $\mathcal{H}(P_K)$.

Ω est un ensemble fini donc $\mathcal{P}_K(\Omega)$ est fini. La partition P_K qui minimise $\mathcal{H}(P_K)$ peut être trouvée par énumération complète. Mais c'est impossible en pratique car :

$$card(\mathcal{P}_K(\Omega)) \sim \frac{K^n}{K!}.$$

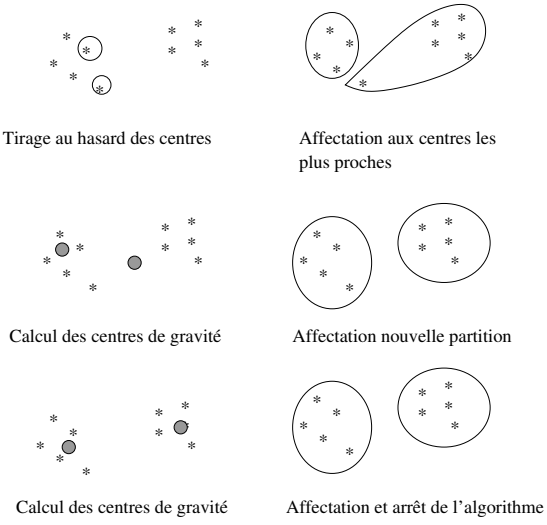
Par exemple, calculez approximativement le nombre de partitions en $K = 3$ classes de $n = 100$ individus.

Les méthodes de partitionnement sont alors souvent des heuristiques du type :

- on part d'une solution réalisable c'est à dire d'une partition P_K^0,
- à l'étape m+1, on cherche une partition $P_K^{m+1} = g(P_K^m)$ telle que $\mathcal{H}(P_K^{m+1}) \leq \mathcal{H}(P_K^m)$.

Lorsque $\mathcal{H}(P_K)$ estl'inertie intra-classe cette heuristique est l'algorithme des k-means. L'algorithme des k-means part d'une partition initiale et répète :

- une étape de représentation où les centres de gravité des classes sont calculés,
- une étape d'affectation chaque individu est affecte à la classe dont centre de gravité est le plus proche (plus petite distance euclidienne).



Plus précisément, l'algorithme des centres mobiles est le suivant :

(a) Initialisation
On se donne une partition $P_K = (C_1, \ldots, C_K)$ et on calcule $\mathbf{g}_1, \ldots, \mathbf{g}_K$

(b) Etape d'affectation
$test \leftarrow 0$
Pour tout i de 1 à n faire

déterminer la classe $k*$ telle que

$$k* = \arg \min_{k=1,\ldots,K} d(\mathbf{x}_i, \mathbf{g}_k)$$

déterminer la classe ℓ de i

si $k* \neq \ell$

$test \leftarrow 1$
$C_{k*} \leftarrow C_{k*} \cup \{i\}$
$C_\ell \leftarrow C_\ell \backslash \{i\}$

(c) Etape de représentation
Pour tout k de 1 à K calculer le centre de gravité de la nouvelle classe C_k

(d) Si $test = 0$ FIN, sinon aller en (b)

Propriétés de l'algorithme

- L'algorithme converge vers une partition réalisant un minimum local de l'inertie intra-classe (à démontrer).
- La partition finale dépend de la partition initiale. Si on relance l'algorithme avec une autre initialisation, la partition finale peut être différente. En pratique,
  - on lance N fois l'algorithme avec des initialisations aléatoires différentes.
  - on retient parmi les N partitions finales, celle ayant le pourcentage d'inertie expliquée le plus grand.
- La complexité de l'algorithme est $o(KpnT)$ où T est le nombre d'itérations. L'algorithme va donc pouvoir s'appliquer à des grands jeux de données et être répété plusieurs fois sans que cela "coûte trop cher".

On applique cet algorithme à un petit exemple pour illustrer deux aspects méthodologiques :

- Pourquoi faut-il parfois standardiser les données ?
- Comment interpréter les classes à l'aide d'une ACP ?

Le jeu de donnée indique la quantité de protéines consommée dans 9 types d'aliments dans 25 pays européens : 25 individus et 9 variables quantitatives.

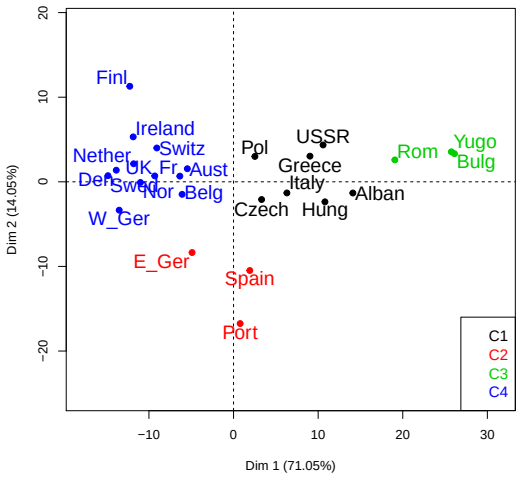
```
# devtools::install_github("chavent/PCAmixdata")
# This needs the devtools package to be installed :
# install.packages("devtools")
library(PCAmixdata)
data(protein)
print(protein[1:5,])

##       Red.Meat White.Meat Eggs Milk Fish Cereals Starchy.Foods Nuts Fruite.veg.
## Alban     10.1        1.4  0.5  8.9  0.2      42           0.6  5.5         1.7
## Aust       8.9       14.0  4.3 19.9  2.1      28           3.6  1.3         4.3
## Belg      13.5        9.3  4.1 17.5  4.5      27           5.7  2.1         4.0
## Bulg       7.8        6.0  1.6  8.3  1.2      57           1.1  3.7         4.2
## Czech      9.7       11.4  2.8 12.5  2.0      34           5.0  1.1         4.0
```

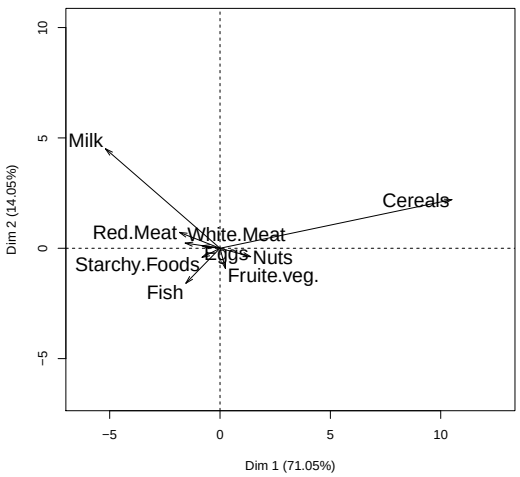

L'algorithme des *k*means est appliqué à ce jeu de données avec :

- $K = 4$ classes,
- $N = 5$ répétitions de l'algorithme.

```
#Z <- scale(protein)*sqrt(6/5)
res <- kmeans(protein,centers=4,nstart=5)
P4 <- res$cluster
#partition en 4 classes
print(P4)

##   Alban    Aust    Belg    Bulg   Czech     Den   E_Ger    Finl      Fr  Greece    Hung Ireland
##       1       4       4       3       1       4       2       4       4       1       1       4
##   Italy  Nether     Nor     Pol    Port     Rom   Spain    Swed   Switz      UK    USSR   W_Ger
##       1       4       4       1       2       3       2       4       4       4       1       4
##    Yugo
##       3

#Pourcentage d'inertie expliquée par la partition
print(res$betweenss/res$totss*100,digits=3)

## [1] 75.8
```

Une ACP non normée est réalisée pour visualiser et interpréter cette partition.





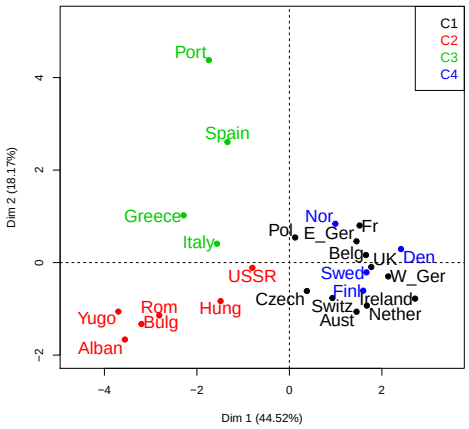

```
## Ecart-type des variables
##      Red.Meat    White.Meat          Eggs          Milk          Fish       Cereals Starchy.Foods
##          3.35          3.69          1.12          7.11          3.40         10.97          1.63
##          Nuts   Fruite.veg.
##          1.99          1.80
```

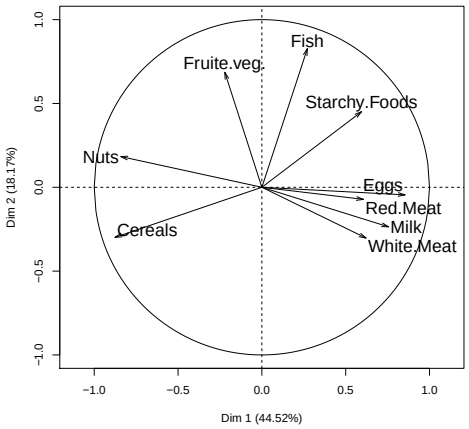
Donnez une interprétation de la partition en 4 classes des 25 pays européens.

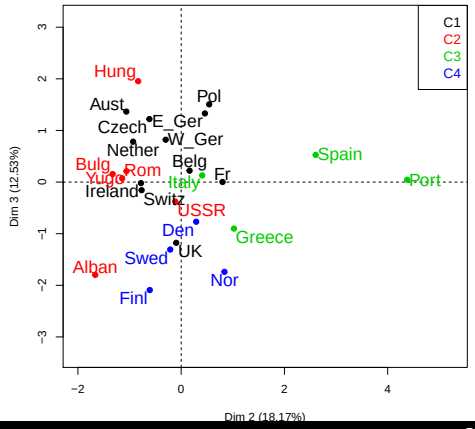
L'algorithme des *k*-means est maintenant appliqué auxdonnées standardisées.

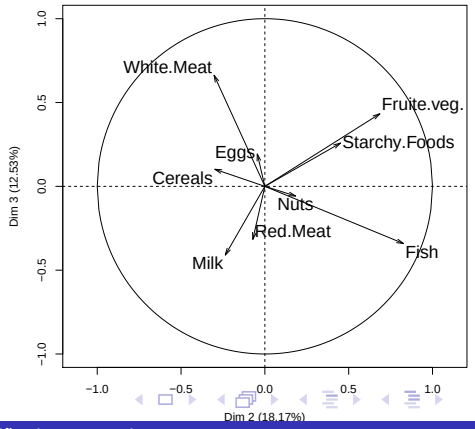
```
n <- nrow(protein)
Z <- sweep(protein,2,STATS=colMeans(protein),FUN="-")
Z <- sweep(Z,2,STATS=apply(Z,2,sd)*sqrt((n-1)/n),FUN="/")
res <- kmeans(Z,centers=4,nstart=5)
P4 <- res$cluster
#partition en 4 classes
print(P4)

##   Alban    Aust    Belg    Bulg   Czech     Den   E_Ger    Finl      Fr  Greece    Hung Ireland
##       2       1       1       2       1       4       1       4       1       3       2       1
##   Italy  Nether     Nor     Pol    Port     Rom   Spain    Swed   Switz      UK    USSR   W_Ger
##       3       1       4       1       3       2       3       4       1       1       2       1
##    Yugo
##       2

#Pourcentage d'inertie expliquée par la partition
print(res$betweenss/res$totss*100,digits=3)

## [1] 59.8
```

Donnez une interprétation de cette partition en 5 classes à partir des graphiques de l'ACP normée ci-après.









Plan

La structure classificatoire est maintenant la hiérarchie (voir section 1).

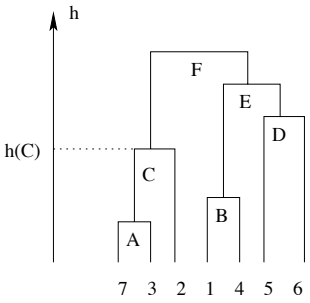
Une hiérarchie binaire est une hiérarchie H de Ω dont chaque classe est la réunion de deux classes. Le nombre de classes (mis à part les singletons) d'une hiérarchie binaire vaut $n-1$.

Une hiérarchie indicée est un couple (H, h) où H est une hiérarchie binaire et h est une fonction de H dans $\mathbb{R}^+$ telle que :

$$\forall A \in H, h(A) = 0 \Leftrightarrow A \text{ est un singleton}$$
$$\forall A, B \in H, A \neq B, A \subset B \Rightarrow h(A) \leq h(B) \text{ (h croissante)}$$

Un dendrogramme (ou arbre hiérarchique) est la représentation graphique d'une hiérarchie indicée et la fonction h mesure la hauteur des classes dans ce dendrogramme.

Par exemple $H = \{\{1\}, \ldots, \{7\}, \{7,3\}, \{2,7,3\}, \{1,4\}, \{5,6\}, \{1,4,5,6\}, \Omega\}$ peut être indicée pour obtenir le dendrogramme suivant :



Une hiérarchie indicée définie ainsi une séqunce de partitions emboîtées de 2 à n classes. Ces partitions sont obtenues en coupant le dendrogramme par une séquence de lignes horizontales.

Par exemple, coupez le dendrogramme ci-dessus pour obtenir une partition en 2 classes et une partition en 4 classes.

Comme la fonction h est croissante, il n'y a pas d'inversion : si $C = A \cup B$ la classe C plus haute que les classes A et B dans le dendrogramme.

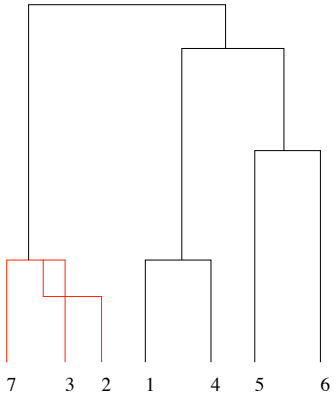


FIGURE: Exemple d'inversion dans le dendrogramme d'une hiérarchie

On peut noter qu'une hiérarchie indicée a plusieurs représentations équivalentes. En effet l'ordre de la représentation des n individus en bas de la hiérarchie peut être modifiée et le nombre de représentations possibles est 2^{n-1}.

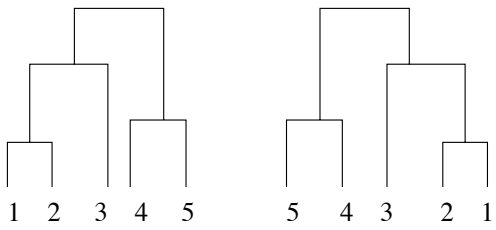


FIGURE: Deux représentations équivalentes de la même hiérarchie indicée

L'algorithme de classification ascendante hiérarchique :

(a) Initialisation
La partition initiale est la partition des singletons $P_n = (C_1, \ldots, C_n)$ avec $C_k = \{k\}$.

(b) Etape agrégative
On agrège les deux classes C_k et $C_{k'}$ de la partition P_K en K classes obtenue à l'étape précédente, qui minimisent une mesure d'agrégation $D(C_k, C_{k'})$. Une nouvelle partition P_{K-1} en $K-1$ classes est ainsi obtenue.

(c) Répeter l'étape (b) jusqu'à obtenir la partition en une classe $P_1 = (\Omega)$

Cet algorithme dépend du choix de la mesure d'agrégation :

$$D : \mathcal{P}(\Omega) \times \mathcal{P}(\Omega) \to \mathbb{R}^+$$

qui mesurer la dissimilarité entre deux classes.

Par exemple trois mesures de dissimilarité entre deux classes A et B de Ω sont représentées ci-dessous. Elles dépendent du choix de la dissimilarité d entre deux individus.

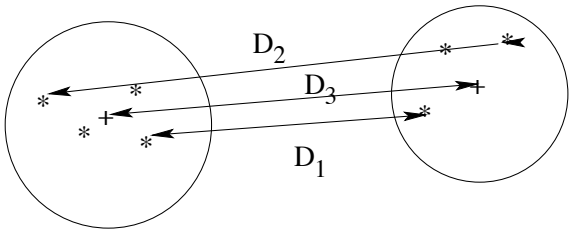


FIGURE: Trois mesures de dissimiarité entre classes

La mesure d'agrégation du lien minimum est :

$$D_1(A, B) = \min_{i \in A, i' \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})$$

La mesure d'agrégation du lien maximum est :

$$D_2(A, B) = \max_{i \in A, i' \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})$$

La mesure d'agrégation de Ward dite de variance minimum est :

$$D_3(A, B) = \frac{\mu_A \mu_B}{\mu_A + \mu_B} d^2(\mathbf{g}_A, \mathbf{g}_B)$$

où

$$\mu_A = \sum_{i \in A} w_i \text{ est le poids de la classe,}$$

$$\mathbf{g}_A = \frac{1}{\mu_A} \sum_{i \in A} w_i \mathbf{x}_i \text{ est le centre de gravité de la classe.}$$

La fonction h de la hiérarchie indicée (H, h) est généralement définie par :

$$h(A \cup B) = D(A, B)$$

où $h(A \cup B)$ est la hauteur de la classe $A \cup B$ dans le dendrogramme de H.

La propriété $A \subset B \Rightarrow h(A) \leq h(B)$ est alors satisfaite pour les trois mesures d'agrégation D précédentes (lien minimum, lien maximum et Ward) mais elle n'est pas satisfaite pour :

- la mesure du lien moyen :

$$D(A, B) = \frac{1}{n_A n_B} \sum_{i \in A} \sum_{i' \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})$$

- la mesure des centroides :

$$D(A, B) = d^2(\mathbf{g}_A, \mathbf{g}_B)$$

et des inversions peuvent alors être observées.

Propriétés de la mesure d'agrégation de Ward.

On peut montrer que :

$$\begin{aligned} D_3(A,B) &= \frac{\mu_A \mu_B}{\mu_A + \mu_B} d^2(\mathbf{g}_A, \mathbf{g}_B) \\ &= I(A \cup B) - I(A) - I(B) \end{aligned} \tag{1}$$

De plus lorsqu'on agrège deux classes A et B d'une partition P_K en K classes pour obtenir une partition P_{K-1} en $K-1$ classes, on a :

$$\mathcal{H}(P_{K-1}) - \mathcal{H}(P_K) = I(A \cup B) - I(A) - I(B) \tag{2}$$

On déduit de (1) and (2) que :

- La mesure d'agrégation de Ward est l'augmentation de l'inertie intra-classe lorsque A et B sont agrégées dans une nouvelle partition.
- La somme des hauteurs du dendrogramme de Ward est égale à l'inertie totale T.

La classification ascendante hiérarchique de Ward agrège à chaque étape les deux classes qui donnent la partition de plus petite inertie intra-classe. Le critère optimisé est donc le même qu'avec la méthode des k-means.

Algorithme de CAH de Ward.

(a) Initialisation : construire la matrice $\Delta = (\delta_{ij})_{n\times n}$ des mesures de Ward entre les singletons :

$$\delta_{ij} = D_3(\{i\},\{j\}) = \frac{w_i w_j}{w_i + w_j} d^2(\mathbf{x}_i, \mathbf{x}_j).$$

(b) Etape agrégative

- Agréger les deux classes A et B de P_K qui minimisent $D_3(A,B)$ pour constuire P_{K-1}.
- Calculer la mesure de Ward entre $A\cup B$ et les autres classes de P_{K-1} avec la formule de Lance et Williams :

$$D_3(A\cup B, C) = \frac{\mu_A+\mu_C}{\mu_A+\mu_B+\mu_C} D_3(A,C) + \frac{\mu_B+\mu_C}{\mu_A+\mu_B+\mu_C} D_3(B,C) - \frac{\mu_C}{\mu_A+\mu_B+\mu_C} D_3(A,B)$$

(c) Recommencer l'étape (b) jusqu'à obtenir la partition en une classe.

L'algorithme de Ward prend donc en entrée :

- le vecteur $\mathbf{w} = (w_i)_{i=1,\dots,n}$ des poids des individus,
- la matrice $\mathbf{D}$ des distances Euclidiennes entre les individus.

Avec la fonction `hclust` de R :

- `hclust(d=`Δ`, method="ward.D")` lorsque les poids sont uniformes,
- `hclust(d=`Δ`, method="ward.D", members=`$\mathbf{w}$`)` sinon.

Lorsque les poids sont $w_i = \frac{1}{n}$, la matrice Δ des mesures de Ward entre les singletons en entrée sont :

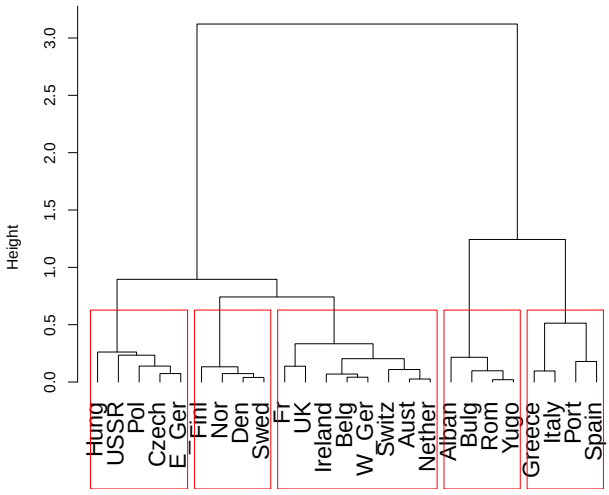
$$\Delta = \frac{\mathbf{D}^2}{2n}$$

où $\mathbf{D}^2 = (d_{ij}^2)_{n\times n}$.

Example des données protéines standardisées.

```
D <- dist(Z)
tree <- hclust(D^2/(2*n),method="ward.D")
plot(tree,hang=-1,main="",sub="",xlab="",cex=1.5)
rect.hclust(tree,k=5)
```

```
  sum(tree$height) # sum of the heights

## [1] 9
```



Le dendrogramme suggère de couper l'arbre en 3 ou 5 classes.

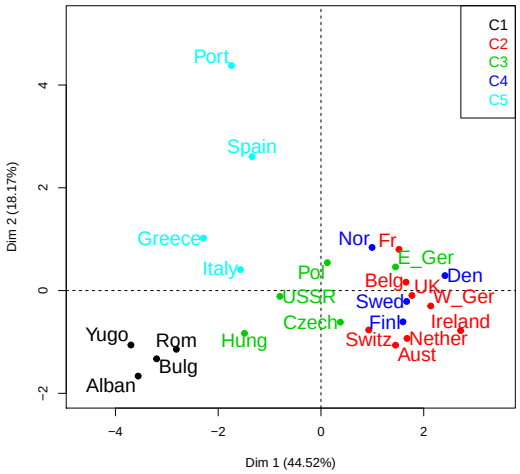
Les peuvent être interprétées, via l'ACP ou via des statistiques descriptives appropriées.

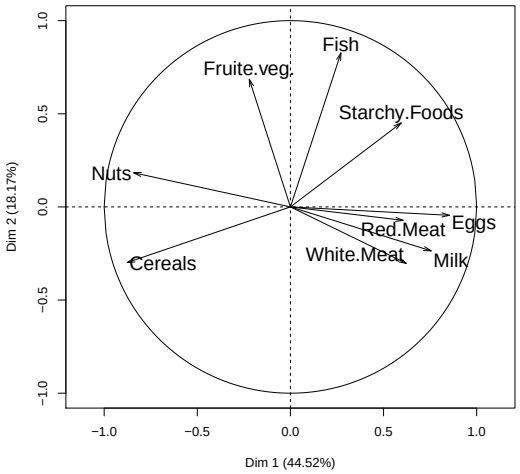
L'arbre est coupé pour obtenir une partition en 5 classes des 25 pays.

```
P5 <- cutree(tree,k=5)
P5

##   Alban    Aust    Belg    Bulg   Czech     Den   E_Ger    Finl      Fr  Greece    Hung Ireland
##       1       2       2       1       3       4       3       4       2       5       3       2
##   Italy  Nether     Nor     Pol    Port     Rom   Spain    Swed   Switz      UK    USSR   W_Ger
##       5       2       4       3       5       1       5       4       2       2       3       2
##    Yugo
##       1

K <- 5
T <- sum(tree$height)
W <- sum(tree$height[1:(n-K)])

#Pourcentage d'inertie expliquée
(1-W/T)*100

## [1] 67
```

Interprétation des classes via l'ACP.

```
P5 <- as.factor(P5)
levels(P5) <- paste("C",1:5,sep="")
library(FactoMineR)
res <- PCA(data.frame(P5,protein),quali.sup=1,graph=FALSE)
```





Plan

Interprétation des résultats

On peut interpréter les classes d'une partition à partir :

- des variables actives : variables utilisées dans le processus de clustering,
- de variables illustratives : utilisées uniquement pour la description des classes.

Ces variables peuvent être quantitatives ou qualitatives.

En pratique, on interprétera souvent les classes par :

- les modalités des variables qualitatives : une modalité est-elle plus fréquente dans la classe, la classe contient-elle tous les individus possédant cette modalité, etc ?
- les variables quantitatives : la moyenne dans la classe est-elle différent de la moyenne chez tous les individus, etc ?

La fonction `catdesc` du package R FactoMineR.

```
#Countries in C1:
pays <- rownames(protein)
pays[which(P5=="C1")]

## [1] "Alban" "Bulg"  "Rom"   "Yugo"

res <- catdes(data.frame(P5,Z),num.var=1)
res$quanti$C1

##              v.test Mean in category Overall mean sd in category Overall sd p.value
## Cereals         3.8              1.8      2.4e-16           0.54          1 0.00017
## Nuts            2.2              1.0     -1.6e-17           0.41          1 0.02972
## Fish           -2.3             -1.1      6.3e-17           0.12          1 0.02342
## Milk           -2.4             -1.1     -2.1e-16           0.15          1 0.01862
## Starchy.Foods  -3.1             -1.5      1.4e-16           0.70          1 0.00190
## Eggs           -3.4             -1.6      3.1e-17           0.39          1 0.00070
```

La première colonne donne les `v.test` (valeurs-test) des variables quantitatives dans la classe.
La dernière colonne donne une `p.value` et par défaut seules les variables avec une `p.value` plus grande que 0.05 sont affichées.

La valeur-test $t_k(X)$ d'une variable quantitative X dans une classe C_k mesure la différence entre la moyenne de X dans C_k et la moyenne de X dans toutes les données, divisé par l'écart-type de la moyenne de n_k individus tirés aléatoirement sans remise (calculé à partir de la variance empirique de X^j notée $\sigma^2(X^j)$ ci-dessous) :

$$t_k(X^j) = \frac{\bar{X}_k^j - \bar{X}^j}{\sqrt{\frac{(n-n_k)}{n-1}\frac{\sigma^2(X^j)}{n_k}}} \tag{3}$$

Si la valeur-test d'une variable dans une classe est grande (en valeur absolue), cette variable caractérise la classe.

De plus, sous l'hypothèse nulle que les n_k individus de C_k sont tirés au hasard sans remise, la statistique $t_k(X)$ est approximativement $\mathbb{N}(0,1)$. Si la `p.value` de ce test est petite (plus petite que 0.05 par exemple), cette variable caractérise la classe.


```
res <- catdes(data.frame(P5,Z),num.var=1)
res$quanti[2:5]
```

```
## $C2
##            v.test Mean in category Overall mean sd in category Overall sd p.value
## Red.Meat      3.5             1.03      1.7e-16           0.94          1 0.00052
## Eggs          3.2             0.96      3.1e-17           0.52          1 0.00125
## White.Meat    2.5             0.76      8.6e-18           0.70          1 0.01091
## Cereals      -2.4            -0.70      2.4e-16           0.28          1 0.01832
##
## $C3
##               v.test Mean in category Overall mean sd in category Overall sd p.value
## Starchy.Foods      2              0.8      1.4e-16           0.59          1   0.049
##
## $C4
##             v.test Mean in category Overall mean sd in category Overall sd p.value
## Milk           2.9             1.37     -2.1e-16           0.59          1  0.0033
## Fish           2.5             1.18      6.3e-17           0.51          1  0.0115
## Nuts          -2.1            -0.98     -1.6e-17           0.18          1  0.0371
## Fruite.veg.   -2.4            -1.14     -4.5e-17           0.28          1  0.0150
##
## $C5
##             v.test Mean in category Overall mean sd in category Overall sd p.value
## Fruite.veg.    3.6              1.7     -4.5e-17           0.31          1 0.00038
## Nuts           2.9              1.3     -1.6e-17           0.70          1 0.00423
## Fish           2.1              1.0      6.3e-17           1.20          1 0.03214
## White.Meat    -2.4             -1.1      8.6e-18           0.22          1 0.01554
```

La fonction `catdesc` décrit aussi les classes avec les modalités des variables qualitatives.

```
zone <- c("east","west","west","east","east","north","east","north","west","south",
          "east","west","south","west","north","east","south","east","south","north",
          "west","west","east","west","east")
res <- catdes(data.frame(P5,zone),num.var=1)
res$category$C1

##           Cla/Mod Mod/Cla Global p.value v.test
## zone=east      44     100     36    0.01    2.6
```

Cla/Mod = proportion de la modalité s dans la classe k
$= \frac{n_{ks}}{n_s}$

Mod/Cla = proportion de la classe k dans la modalité s
$= \frac{n_{ks}}{n_k}$

Global = proportion de la modalité s dans toutes les données
$= \frac{n_s}{n}$


```
res$category

## $C1
##           Cla/Mod Mod/Cla Global p.value v.test
## zone=east      44     100     36    0.01    2.6
##
## $C2
##           Cla/Mod Mod/Cla Global p.value v.test
## zone=west     100     100     32 9.2e-07    4.9
## zone=east       0       0     36 1.2e-02   -2.5
##
## $C3
##           Cla/Mod Mod/Cla Global p.value v.test
## zone=east      56     100     36  0.0024      3
##
## $C4
##            Cla/Mod Mod/Cla Global p.value v.test
## zone=north     100     100     16 7.9e-05    3.9
##
## $C5
##            Cla/Mod Mod/Cla Global p.value v.test
## zone=south     100     100     16 7.9e-05    3.9
```