Classification automatique

STID - 2A

Maxime FRANCOISE

2020 – 2021

Plan

- **Introduction à la classification**
- **Classification Ascendante Hiérarchique (CAH)**
  - Hiérarchie de parties
  - Dissemblance entre parties
  - Choix du nombre de classes et profilage
  - Utilisation de l'outil logiciel
- **Centres mobiles (k-means, nuées dynamiques)**
  - Partitionnement
  - Classification mixte
  - Utilisation de l'outil logiciel
- **Classification sur facteurs principaux**

Introduction à la classification

Classification

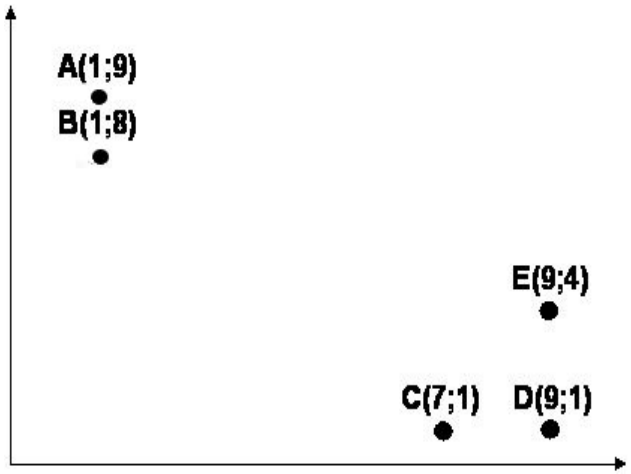
- **Cadre** : analyse de données multidimensionnelles.
- **Données** : table de données (pour l'instant **numériques**) de taille $n \times p$ (n individus décrits par p variables), ou $n \times (p+1)$ (n individus décrits par p variables + 1 étiquette)
- **But** : effectuer un regroupement des n individus en k ($k << n$) groupes de manière à rassembler dans chaque groupe les individus "les plus semblables" selon les modalités des p variables descriptives.
- **Principe général** : table = échantillon de n points $(x_1, \ldots, x_n)$ d'un sous-ensemble de $\mathbb{R}^p$, associés ou non à un ensemble d'étiquettes $(y_1, \ldots, y_n)$ à valeurs dans un ensemble fini discret ou réel.
- **Classification** : déterminer des regroupements de points (=**partition**) en associant entre eux les points les plus semblables selon un certain critère (en général assimilé à une distance).

Exemple simple sans étiquette

- On considère la table de données suivante :

Nom \ Variable	Var_1	Var_2
A	1	9
B	1	8
C	7	1
D	9	1
E	9	4

- Dans $\mathbb{R}^2$, on obtient la représentation suivante :



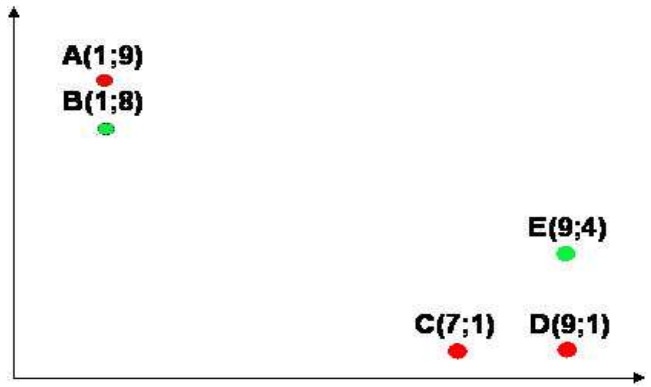
- **Question** : comment construire la partition la plus optimale possible ?

Exemple simple avec étiquette

- On considère la même table de données à laquelle on a rajouté des étiquettes :

Variable / Nom	*Couleur*	Var_1	Var_2
A	rouge	1	9
B	vert	1	8
C	rouge	7	1
D	rouge	9	1
E	vert	9	4

- Dans $\mathbb{R}^2$, on obtient la représentation suivante :



- **Question** : comment construire la partition la plus optimale possible pour regrouper au mieux les individus de même couleur ?

Deux types de Classification

- **Apprentissage non-supervisé** (**Clustering**/**Classification**) : pas d'étiquette. k classes sont construites suivant les modalités des p variables descriptives afin de représenter les données.
  $\Longrightarrow$ Profilage des classes.

- **Apprentissage supervisé** (**Classification**/**Classement**) : l'étiquette est connue pour les n individus de la table. k classes sont construites suivant les modalités des p variables explicatives de manière à regrouper au mieux les données ayant la même étiquette.
  $\Longrightarrow$ Un nouvel individu recevra l'étiquette de la classe dans laquelle il tombe (*prédicteur*).

Exemples d'application

- **Marketing** : création de profils clients permettant de
  - cibler les offres promotionnelles suivant certains critères (non-supervisé),
  - prédire s'ils seront mauvais payeurs (supervisé),
  - ...
- **Médias** :
  - création de profils utilisateurs permettant de personnaliser une page web (non supervisé),
  - filtres spam (supervisé),
  - ...
- **Economie** : création de profils pays suivant différents critères tels que
  - les échanges économiques,
  - le taux d'armement,
  - le niveau d'éducation,
  - ...
- **Et bien d'autres...**

Apprentissage non-supervisé : Présentation des données

- Echantillon de n individus décrits par p variables quantitatives.
- Individu $i \longrightarrow p$ valeurs $x_i^{(1)}, \ldots, x_i^{(p)}$, correspondant aux valeurs prises par les variables $X_i^{(1)}, \ldots, X_i^{(p)}$ sur l'échantillon.
- Présentation sous forme de **tableau** à double entrée, avec en général les individus en ligne et les variables en colonne :

Individu \ **Variable**	$X^{(1)}$	$X^{(2)}$	$\ldots$	$X^{(p)}$
1	$x_1^{(1)}$	$x_1^{(2)}$	$\ldots$	$x_1^{(p)}$
2	$x_2^{(1)}$	$x_2^{(2)}$	$\ldots$	$x_2^{(p)}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
n	$x_n^{(1)}$	$x_n^{(2)}$	$\ldots$	$x_n^{(p)}$

- Tableau souvent associé à une **matrice** à n lignes et p colonnes.

Apprentissage non-supervisé : Exemple de données

Consommation de protéines en Europe : pour chacun des 25 pays de l'union européenne, relevé de la consommation moyenne journalière des 9 types de protéines.

Prot. / **Pays**	viandr	viandb	oeuf	lait	poisson	céréale	féculent	...
Bulgaria	7,8	6,0	1,6	8,3	1,2	56,7	1,1	...
Yugoslavia	4,4	5,0	1,2	9,5	0,6	55,9	3,0	...
Romania	6,2	6,3	1,5	11,1	1,0	49,6	3,1	...
Germany	11,4	12,5	4,1	18,8	3,4	18,6	5,2	...
France	18,0	9,9	3,3	19,5	5,7	28,1	4,8	...
Norway	9,4	4,7	2,7	23,3	9,7	23,0	4,6	...
Greece	10,2	3,0	2,8	17,6	5,9	41,7	2,2	...
...	...	...	...	...	...	...	...	...

Quelle classification, et donc quel profilage des pays, en fonction de la consommation de protéines ?

Apprentissage non-supervisé : Objectifs et méthodes

- **Objectifs** :
  - Construire un ou plusieurs ensembles de **classes homogènes** (les éléments d'une classe sont plus ressemblants entre eux qu'avec un élément d'une autre classe).
  - **Caractériser les classes** d'un point de vue descriptif.
- **Méthode générale** : Classification des n individus en k classes telles que :
  - l'homogénéité soit maximale à l'intérieur de chaque classe,
  - l'hétérogénéité soit maximale d'une classe à l'autre.
- **Attention** : les **classes** et le **nombre** k de classes sont **inconnus**.

Classification = partition ?

- On construit des classes **disjointes**.
- Si tout individu est classé, on aboutit à la notion de **partition** : tout individu appartient à une classe et une seule.
- Les éléments d'une même classe sont équivalents et donc indiscernables. Il suffit alors d'utiliser un **représentant** pour chaque classe (par exemple le point moyen) dans la suite des traitements.

Partitions optimales

- **Idée naïve** : rechercher une partition optimale, par exemple celle qui maximise un critère quantifiant le degré d'homogénéité des classes.
- **Problème** : examen de toutes les partitions possibles d'un ensemble à n éléments.
- **Nombre de partitions** d'un ensemble à n éléments (*nombre de Bell*) :
  En notant
  - $P_{n,q}$ le nombre de parties à q éléments,
  - P_n le nombre de partitions d'un ensemble à n éléments,

  on obtient
  - $P_n = \sum_{q=1}^{n} P_{n,q}$ avec
  - $P_{n,q} = P_{n-1,q-1} + qP_{n-1,q}$.

  $\Rightarrow$ Problème *np*-complet (explosion combinatoire).
  $\Rightarrow$ Obligation de chercher un **optimum local**.

Les trois approches proposées

- **Méthodes hiérarchiques** : exemple du cas ascendant
  - Construction d'une *hiérarchie* de partitions par *agrégation* successive d'individus les plus "proches".
  - Choix du nombre de classes.
  - Obtention de la classification.

- **Méthodes de partitionnement** : exemple des centres mobiles
  - k fixé a priori.
  - Agrégation séquentielle autour de k "centres".
  - Calcul des barycentres des k groupes, devenant les nouveaux "noyaux".
  - Itération jusqu'à convergence.

- **Méthode mixte** (hiérarchique+partitionnement)

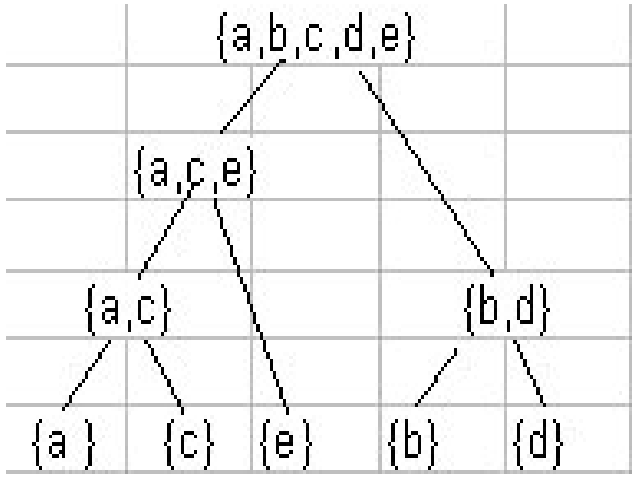
Méthodes étudiées dans ce cours

- Classification ascendante hiérarchique (CAH),
- Centres mobiles (k-means, nuées dynamiques).

Classification Ascendante Hiérarchique (CAH)

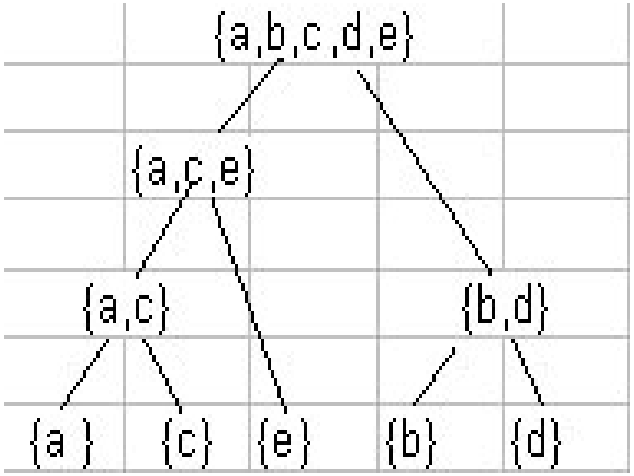
Hiérarchie de Parties

- **Hiérarchie de parties** : ensemble de parties “emboîtées”.
- Cette représentation traduit la façon dont elles sont construites :
  - soit par réunion successive de parties (*ascendante*),
  - soit par division successive (*descendante*).
- La relation d’inclusion conduit à une représentation graphique du type :



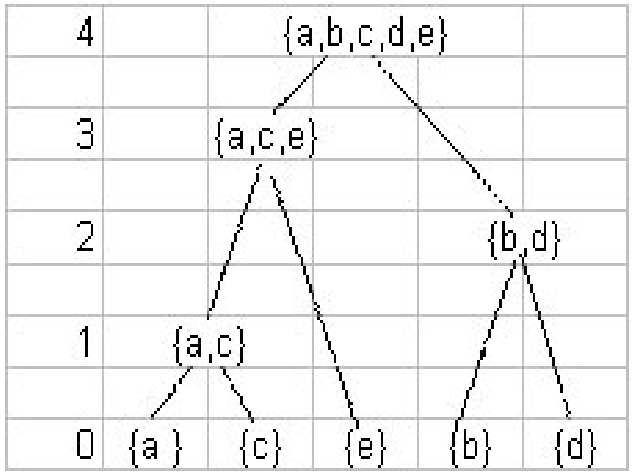
Représentation graphique

- **Ordre de formation** : traduction des différents niveaux de ressemblance entre les parties.
- Ordre utilisé pour **ordonner** le graphique suivant l'axe vertical :



Représentation graphique (2)

- **Indice de diamètre** : indice **numérique** quantifiant la variabilité entre les parties



- $\Longrightarrow$ **hiérarchie indicée**.
- Représentation graphique = **dendrogramme**.

CAH : l'algorithme de base

- **Etape 1 :**
  - Calcul des distances entre chaque élément de l'échantillon de données.
  - Regroupement des 2 individus les plus proches $\Rightarrow$ obtention de $n-1$ classes.
  - L'indice de diamètre des parties à 1 élément est 0 et celui de la partie obtenue par fusion des 2 éléments est proportionnel à la distance entre ces 2 éléments.

- **Etape j ($1 \leqslant j \leqslant n-1$) :**
  - Calcul des **dissemblances** entre chaque partie obtenue à l'étape $j-1$.
  - Regroupement des deux parties les plus proches $\Rightarrow$ obtention de $n-j$ classes.
  - L'indice de diamètre de la nouvelle partie obtenue est proportionnel à la dissemblance entre les 2 parties dont elle est issue.

- **Fin de l'algorithme :** A l'étape $n-1$, lorsque tous les éléments sont regroupés dans une seule partie.

Dissemblance : les choix

- **Agrégation des parties :** choix d'une **distance** δ permettant de quantifier la **dissemblance** entre les parties. δ repose elle-même sur le choix d'une distance d entre les éléments de $\mathbb{R}^p$, en général d =distance euclidienne.
- **Dissemblances** : Exemples
  - **saut minimal** (lien simple) :

$$\delta_{min}(A, B) = \min\{d(x, y) \ ; \ x \in A, \ y \in B\}$$

  - **saut maximal** (lien complet) :

$$\delta_{max}(A, B) = \max\{d(x, y) \ ; \ x \in A, \ y \in B\}$$

  - **saut moyen** (lien moyen) :

$$\delta_{moy}(A, B) = \frac{\sum_{\{x \in A, \ y \in B\}} d(x, y)}{card(A) \times card(B)}$$

Critère de Ward

Critère nécessitant l'utilisation de la distance euclidienne

- d distance entre 2 points = distance euclidienne.
- Les classes sont représentées par leur **centre de gravité**.
- La fusion de 2 classes est représentée par le remplacement des 2 points par leur centre de gravité muni de la somme des masses.
- Dans ce cadre, on ne regarde plus les distances pour choisir les classes à fusionner, mais la décomposition de **l'inertie** selon les classes d'une partition.

Inertie inter et inertie intra

- n individus dans k groupes $P_1, \ldots, P_k$,
- G : isobarycentre des n individus, $G =$,
- n_j : effectif du groupe P_j,
- G_j : isobarycentre du groupe P_j, $G_j =$.
- **Inertie intra-classe** : Somme des inerties des points du groupe P_j au barycentre G_j.

$$I_{intra}(k) = \frac{1}{n} \sum_{j=1}^{k} \sum_{\{i \ ; \ x_i \in P_j\}} d^2(x_i, G_j)$$

- **Inertie inter-classe** : Inertie des barycentres G_j au barycentre G.

$$I_{inter}(k) = \frac{1}{n} \sum_{j=1}^{k} n_j d^2(G_j, G)$$

Inertie totale

- **Inertie totale** : Inertie des n points au barycentre G.

$$I_G = \frac{1}{n} \sum_{i=1}^{n} d^2(x_i, G)$$

- **Relation de Huygens** : L'inertie totale est la somme des inerties intra et inter classes.

$$I_G = I_{intra}(k) + I_{inter}(k)$$

- **2 cas extrêmes** :
  - $k = n$: $I_{intra}(n) = 0$ et $I_{inter}(n) = I_G$.
  - $k = 1$: $I_{intra}(1) = I_G$ et $I_{inter}(1) = 0$.

Qualité d'une partition (LMP)

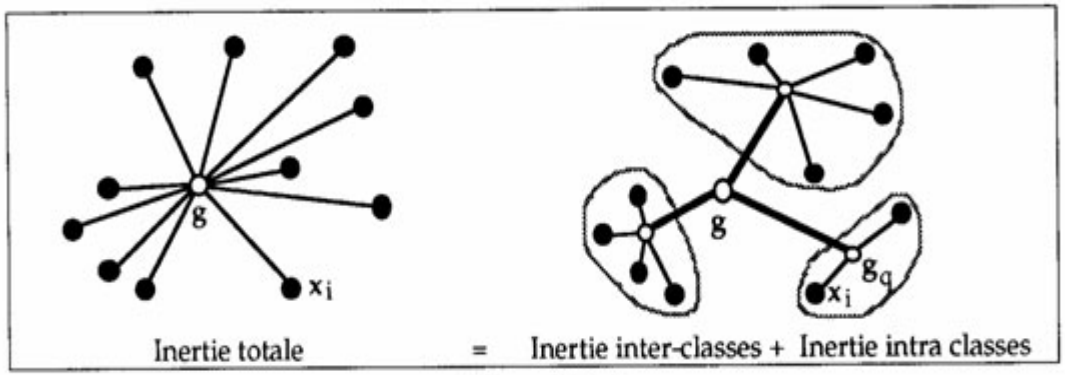


Figure 2.2 - 9
Décomposition de l'inertie selon la relation de Huygens

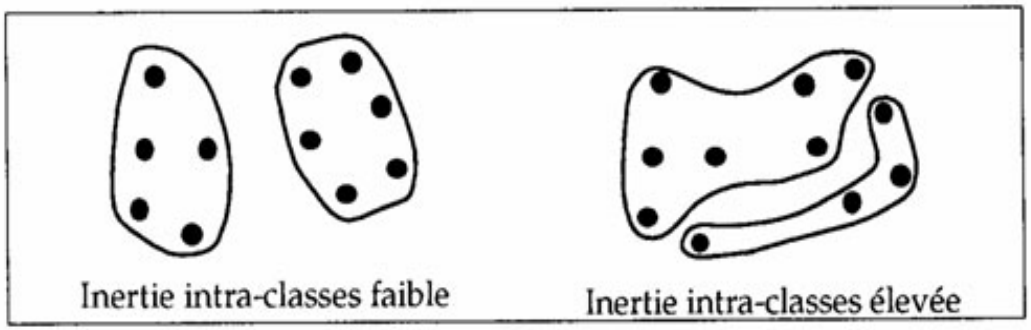


Figure 2.2 - 10
Qualité globale d'une partition

Algorithme d'agrégation par critère de Ward

- **Principe général** : Au départ de l'algorithme, l'inertie inter classes est maximale (n classes). A la fin, celle-ci est nulle (1 classe).
  $\Rightarrow$ On minimise à chaque étape la perte d'inertie inter-classes (ou on minimise le gain d'inertie intra-classes).

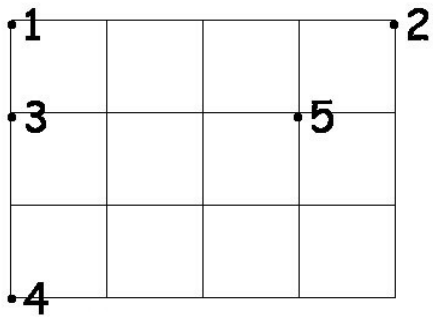
- **A chaque étape** : On regroupe les 2 classes pour lesquelles la perte d'inertie inter-classes est minimale. C'est-à-dire, on regroupe les classes j et j' pour lesquelles la perte

$$\Delta_{jj'} = \frac{n_j n_{j'}}{n_j + n_{j'}} d^2(G_j, G_{j'})$$

  est minimale.

Un exemple simple

- Cinq points dans un plan



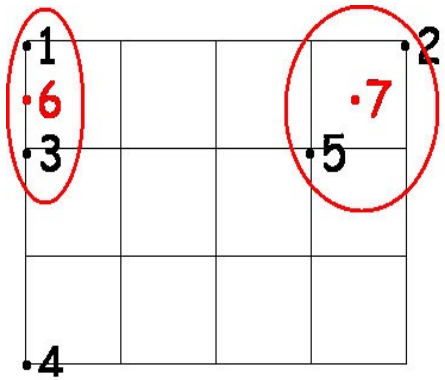
$I_{tot} = 4,24$

- Carrés des distances euclidiennes

	1	2	3	4	5
1	0	16	1	9	10
2		0	17	25	2
3			0	4	9
4				0	13
5					0

- Regroupement 1 et 3 $\Rightarrow$ nouvel individu 6

Un exemple simple (2)



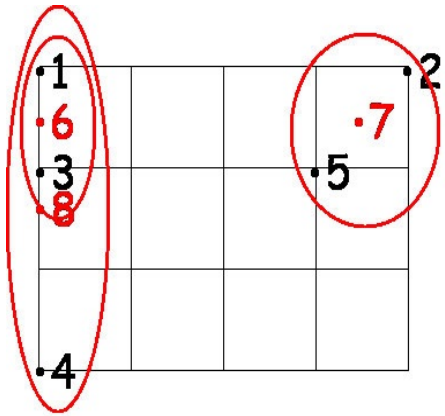
Iintra = 0,1 Iinter = 4,14

- Carrés des distances euclidiennes pondérés (arrondis)

	2	4	5	6
2	0	25	2	6
4		0	13	6
5			0	9
6				0

- Regroupement 2 et 5 $\Rightarrow$ nouvel individu 7

Un exemple simple (3)



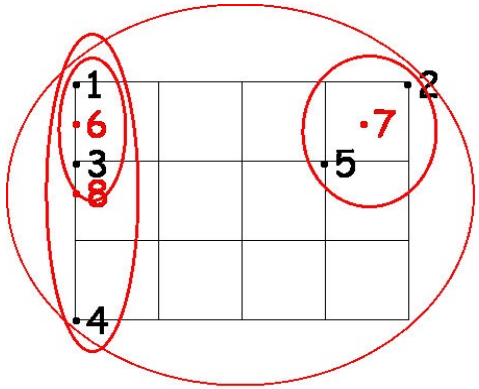
Iintra = 0,3 Iinter = 3,94

- Carrés des distances euclidiennes (arrondis)

	4	6	7
4	0	6	19
6		0	12
7			0

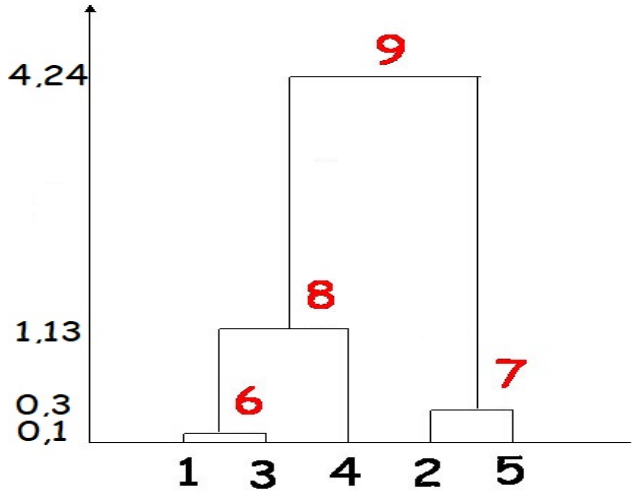
- Regroupement 4 et 6 $\Rightarrow$ nouvel individu 8

Un exemple simple (4)



Iintra = 1,134 Iinter = 3,106

- Dendrogramme



Choix du nombre de classes

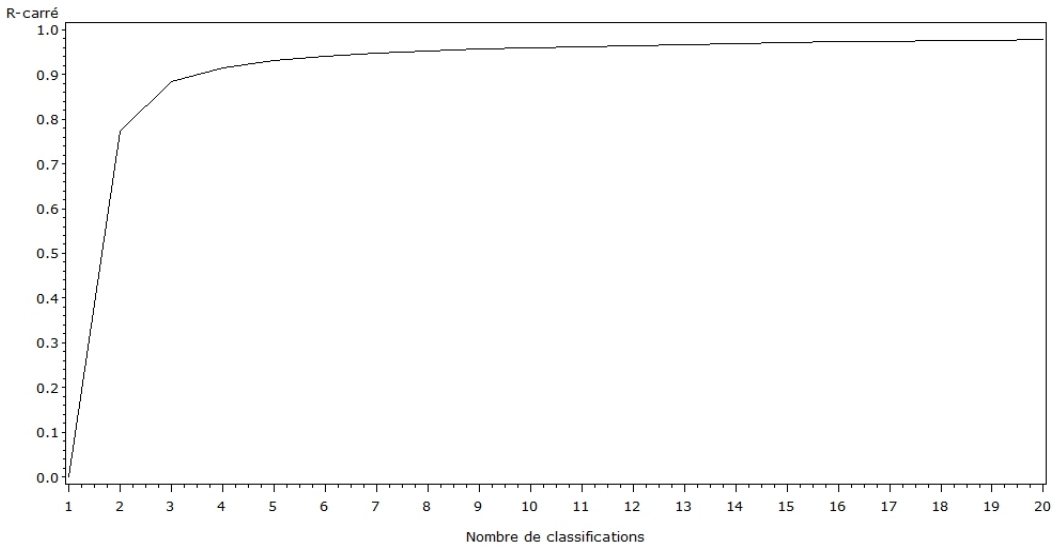
- **Coupure de l'arbre** à un niveau donné de l'indice $\Longrightarrow$ **partition**.
- La coupure doit se faire :
  - **après** les agrégations correspondant à des valeurs **peu élevées** de l'indice,
  - **avant** les agrégations correpondant à des niveaux **élevés** de l'indice (dissocient les groupes bien distincts dans la population).
- **Règle empirique** : sélection d'une coupure lors d'un saut important de l'indice par **inspection visuelle** de l'arbre.
- Ce saut traduit le passage brutal entre des classes d'une certaine homogénéité de l'ensemble à des classes beaucoup moins homogènes.
- Dans la plupart des cas, il y a **plusieurs paliers** et donc **plusieurs choix de partitions** possibles.

Choix du nombre de classes : R^2

- **Critère numérique** de choix du nombre de classes : pour $k = 1, \ldots, n$,

$$\begin{aligned} R^2(k) &= \frac{\text{Inertie inter-classes}}{\text{Inertie totale}} \\ &= \frac{I_{inter}(k)}{I_G} \end{aligned}$$

- **Repérage** du point k où il y a **rupture** de pente dans le R^2 :

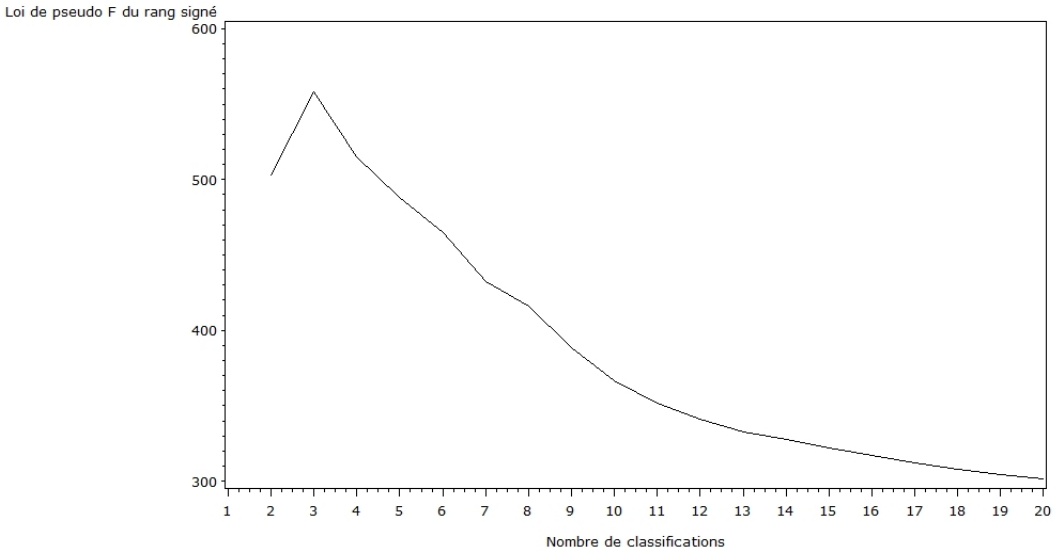


Choix du nombre de classes : *PseudoF*

- Autre **critère numérique** (plus stable) : pour $k = 1, \ldots, n$,

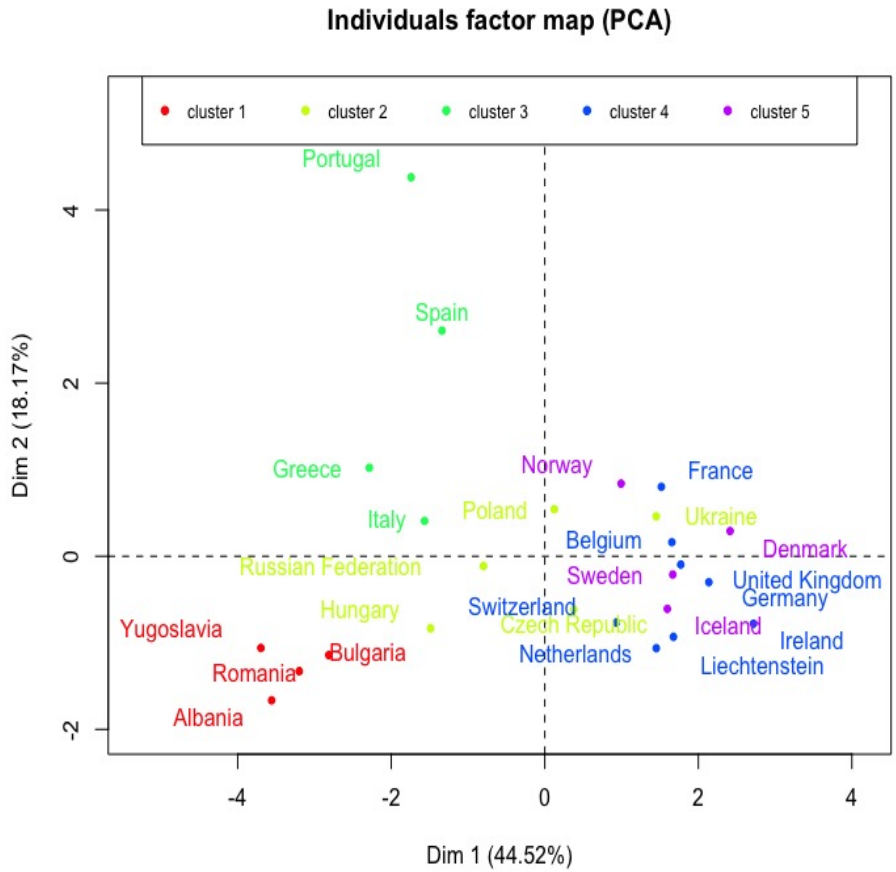
$$\begin{aligned} PseudoF(k) &= \frac{\text{Inertie inter-classes}/(k-1)}{\text{Inertie intra-classes}/(n-k)} \\ &= \frac{R^2(k)/(k-1)}{(1-R^2(k))/(n-k)} \end{aligned}$$

- **Repérage** du point k où *PseudoF* est **maximal** :



Visualisation de la partition

- Données **numériques** $\Longrightarrow$ ACP
- **Intérêt** : **visualisation des groupes** dans le premier plan factoriel (concentrant la plus grosse part de variance).



Caratérisation des classes

- Une fois les classes choisies, on peut sélectionner les variables caractérisant chaque classe à l'aide
  - des représentations des groupes dans **les** plans factoriels,
  - **et** de **valeurs test**.
- Pour des variables **continues** on utilise l'écart de la moyenne dans la classe j à la moyenne générale renormalisé par l'écart-type dans la classe j :

$$t_j(X) = \frac{\overline{X}_j - \overline{X}}{\sigma_j(X)}$$

- Si X est illustrative, on peut effectuer un test sur $t_j(X)$.
- Si X est active, $t_j(X)$ est utilisé comme **mesure de similarité** entre la variable et la classe.
- Pour des variables **nominales** (illustratives), on regarde l'abondance de chaque modalité dans la classe j.

CAH : Exemple d'application sous R

En utilisant la commande `read.table` :

```
proteine=read.table(file="proteines.dat", dec=",",
col.names=c("country", "viandr","viandb ", "oeuf", "lait", "poisson",
"cereals", "feculent", "oleagine", "fruitleg"))
```

Résumé des observations :

```
> proteine
```

	country	viandr	viandb.	oeuf	lait	poisson	cereals	feculent	oleagine	fruitleg
1	Bulgaria	7.8	6.0	1.6	8.3	1.2	56.7	1.1	3.7	4.2
2	Yugoslavia	4.4	5.0	1.2	9.5	0.6	55.9	3.0	5.7	3.2
3	Romania	6.2	6.3	1.5	11.1	1.0	49.6	3.1	5.3	2.8
4	Russian Federation	9.3	4.6	2.1	16.6	3.0	43.6	6.4	3.4	2.9
5	Albania	10.1	1.4	0.5	8.9	0.2	42.3	0.6	5.5	1.7
6	Greece	10.2	3.0	2.8	17.6	5.9	41.7	2.2	7.8	6.5
7	Hungary	5.3	12.4	2.9	9.7	0.3	40.1	4.0	5.4	4.2
8	Italy	9.0	5.1	2.9	13.7	3.4	36.8	2.1	4.3	6.7
9	Poland	6.9	10.2	2.7	19.3	3.0	36.1	5.9	2.0	6.6
10	Czech Republic	9.7	11.4	2.8	12.5	2.0	34.3	5.0	1.1	4.0
11	Spain	7.1	3.4	3.1	8.6	7.0	29.2	5.7	5.9	7.2
12	France	18.0	9.9	3.3	19.5	5.7	28.1	4.8	2.4	6.5
13	Liechtenstein	8.9	14.0	4.3	19.9	2.1	28.0	3.6	1.3	4.3
14	Portugal	6.2	3.7	1.1	4.9	14.2	27.0	5.9	4.7	7.9
15	Belgium	13.5	9.3	4.1	17.5	4.5	26.6	5.7	2.1	4.0
16	Iceland	9.5	4.9	2.7	33.7	5.8	26.3	5.1	1.0	1.4
17	Switzerland	13.1	10.1	3.1	23.8	2.3	25.6	2.8	2.4	4.9
18	Ukraine	8.4	11.6	3.7	11.1	5.4	24.6	6.5	0.8	3.6
19	United Kingdom	17.4	5.7	4.7	20.6	4.3	24.3	4.7	3.4	3.3
20	Ireland	13.9	10.0	4.7	25.8	2.2	24.0	6.2	1.6	2.9
21	Norway	9.4	4.7	2.7	23.3	9.7	23.0	4.6	1.6	2.7
22	Netherlands	9.5	13.6	3.6	23.4	2.5	22.4	4.2	1.8	3.7
23	Denmark	10.6	10.8	3.7	25.0	9.9	21.9	4.8	0.7	2.4
24	Sweden	9.9	7.8	3.5	24.7	7.5	19.5	3.7	1.4	2.0
25	Germany	11.4	12.5	4.1	18.8	3.4	18.6	5.2	1.5	3.8

CAH : Exemple sous R

- `proteinestd = scale(proteine[,-1])`

  standardise les données,

- `distance=dist(proteinestd,method="euclidean")`

  crée la matrice symétrique des distances euclidiennes utilisée pour construire le dendrogramme,

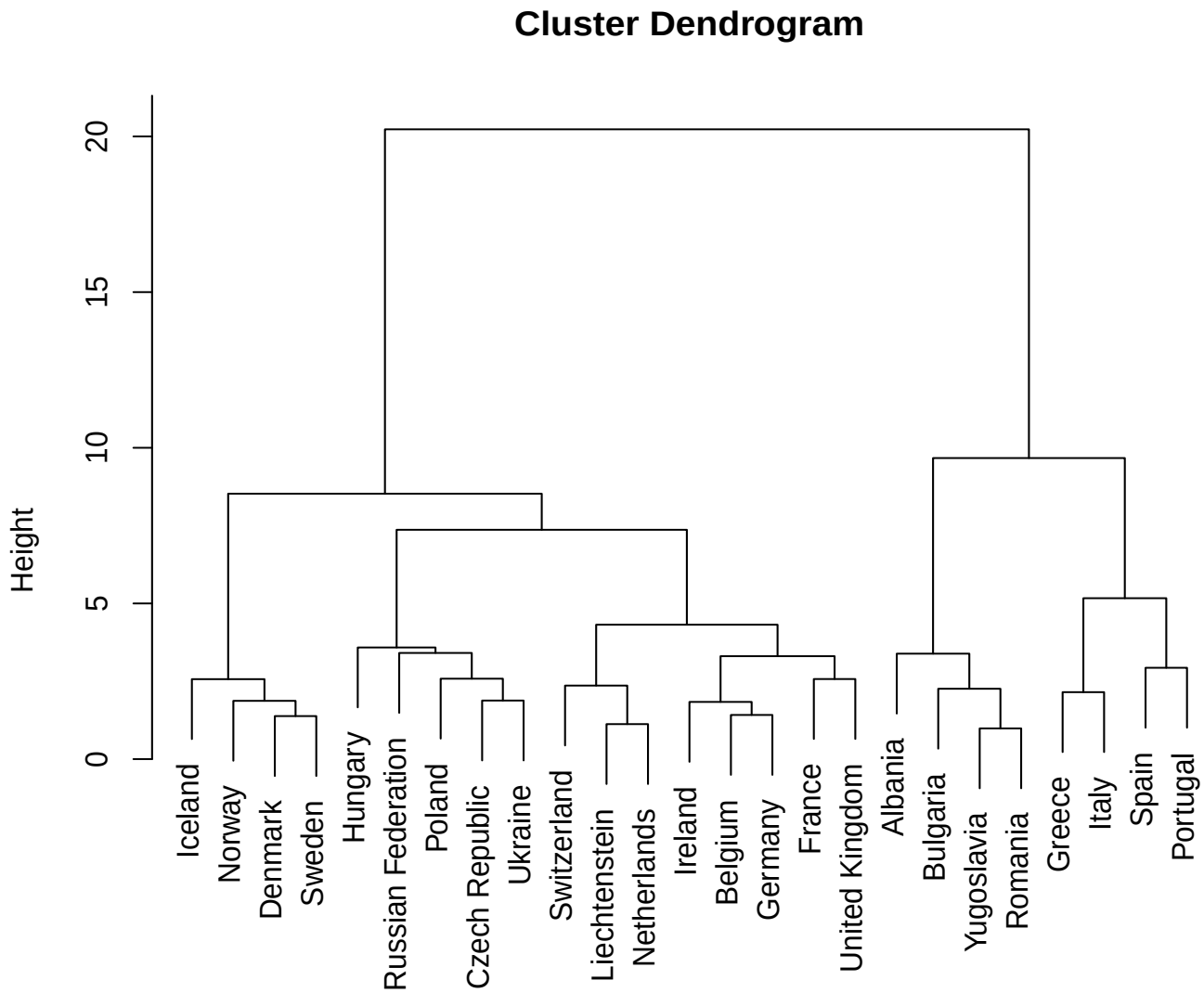
- `dendro=hclust(distance, method="ward.D2")`

  crée le dendrogramme en utilisant la méthode de Ward,

- `plot(dendro, labels=proteine$country)`

  affiche le dendrogramme avec les labels "country" au niveau des feuilles.

CAH : Exemple sous R (2)



CAH : Exemple sous R (3)

```
class34=cutree(dendro,k=3:4)
rownames(class34)=proteine$country
```

crée une matrice détaillant la répartition des pays dans les partitions à 3 et 4 classes.

```
> class34
                   3 4
Bulgaria           1 1
Yugoslavia         1 1
Romania            1 1
Russian Federation 2 2
Albania            1 1
Greece             3 3
Hungary            2 2
Italy              3 3
Poland             2 2
Czech Republic     2 2
Spain              3 3
France             2 2
Liechtenstein      2 2
Portugal           3 3
Belgium            2 2
Iceland            2 4
Switzerland        2 2
Ukraine            2 2
United Kingdom     2 2
Ireland            2 2
Norway             2 4
Netherlands        2 2
Denmark            2 4
Sweden             2 4
Germany            2 2
```

CAH : Exemple sous R (4)

```
class4=array(cutree(dendro,k=4))
rownames(class4)=proteine$country
```

crée un vecteur détaillant la répartition des pays dans la partition à 4 classes.

```
> class4
          Bulgaria          Yugoslavia             Romania Russian Federation
                 1                   1                   1                   2
            Albania              Greece             Hungary               Italy
                 1                   3                   2                   3
             Poland      Czech Republic               Spain              France
                 2                   2                   3                   2
      Liechtenstein            Portugal             Belgium             Iceland
                 2                   3                   2                   4
        Switzerland             Ukraine      United Kingdom             Ireland
                 2                   2                   2                   2
             Norway         Netherlands             Denmark              Sweden
                 4                   2                   4                   4
            Germany
                 2
```

CAH : Exemple sous R (5)

On peut calculer des résumés statistiques à partir des sorties :

- `repartition` donne la répartition des observations dans chaque classe,

```
> repartition
                  1  2 3 4
Nb observations 4 13 4 4
```

- `moyenne` : moyenne des observations par classe et par variable,

```
> moyenne
                1         2      3      4
viandr      7.125 11.176923  8.125  9.850
viandb.     4.675 10.407692  3.800  7.050
oeuf        1.200  3.546154  2.475  3.150
lait        9.450 18.346154 11.200 26.675
poisson     0.750  3.130769  7.625  8.225
cereals    51.125 28.946154 33.675 22.675
feculent    1.950  5.000000  3.975  4.550
oleagine    5.050  2.246154  5.675  1.175
```

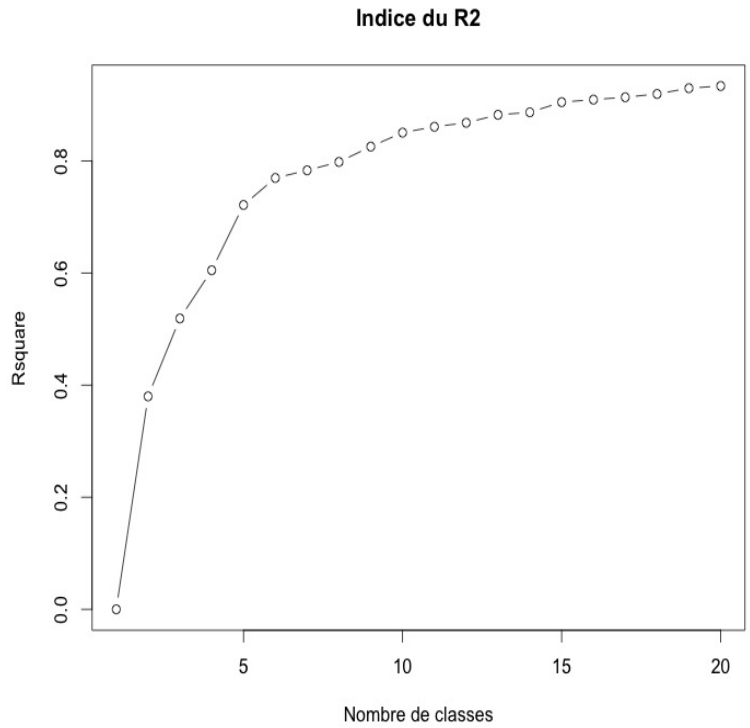
CAH : Exemple sous R (6)

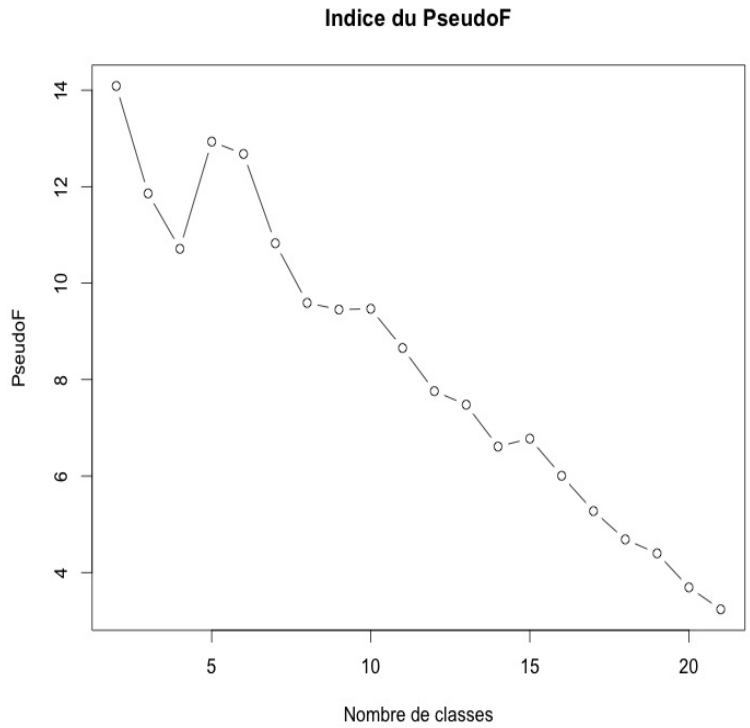
`ecart` : écart-type des observations par classe et par variable

```
> ecart
                1         2         3         4
viandr   2.4212600 3.8306122 1.8099263 0.5446712
viandb.  2.2529610 2.7617813 0.9128709 2.8734416
oeuf     0.4966555 0.8120092 0.9251126 0.5259911
lait     1.2041595 4.8818739 5.5874860 4.7415715
poisson  0.4434712 1.5112824 4.6349218 1.9482898
cereals  6.6854942 7.3718627 6.8006740 2.8241518
feculent 1.2871156 1.1387127 2.1093048 0.6027714
oleagine 0.9146948 1.2319466 1.5713582 0.4031129
fruitleg 1.0340052 1.1757703 0.6238322 0.5619905
```

CAH : Exemple sous R (7)

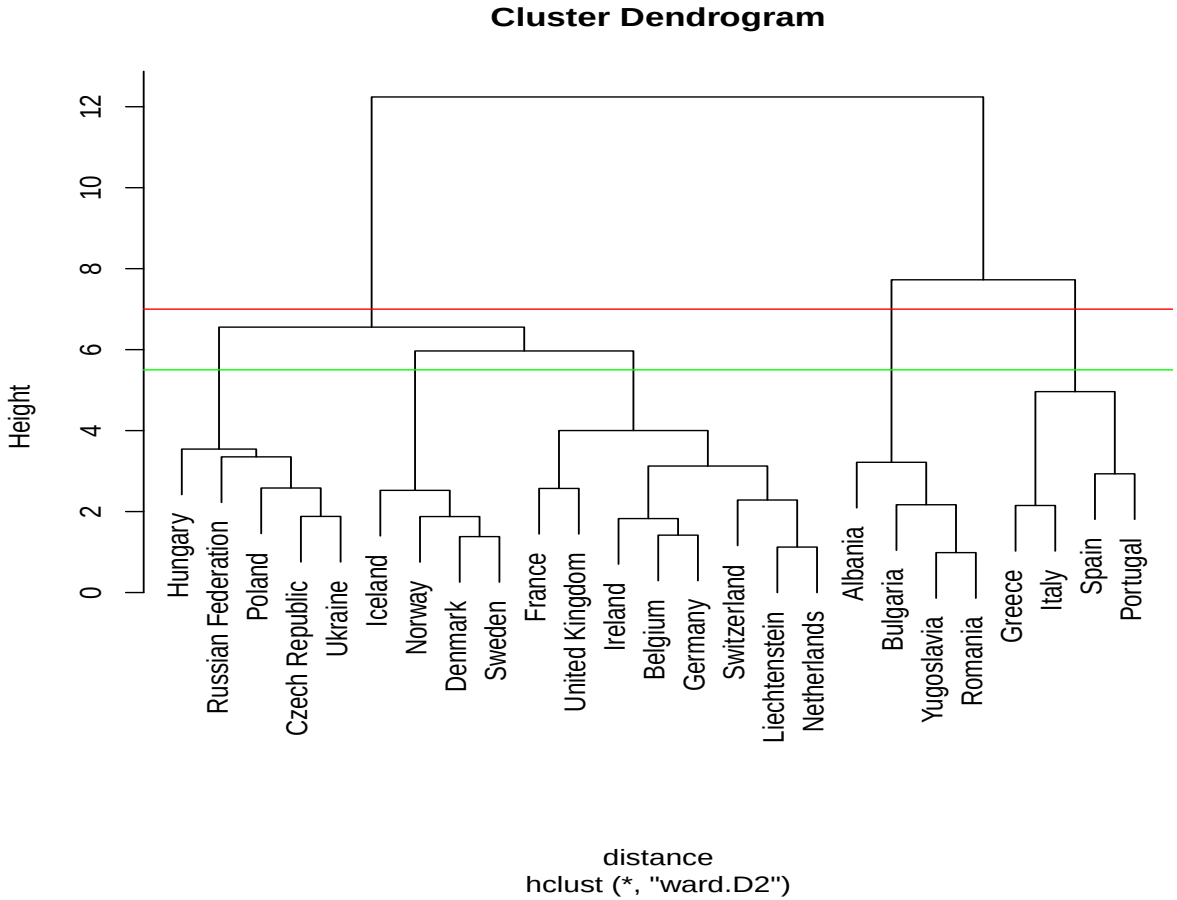
R^2 et *PseudoF* :





CAH : Exemple sous R (8)

```
plot(dendro, labels=proteine$country)
abline(h=5.5, col="green")
abline(h=7, col="red")
```



Partition en 5 classes : Résultat

- **Classe 1** : Bulgarie, Yougoslavie, Roumanie, Albanie,
- **Classe 2** : Russie, Pologne, Ukraine, Hongrie, Rep. Tchèque
- **Classe 3** : Grèce, Italie, Espagne, Portugal,
- **Classe 4** : Lichtenstein, Belgique, Suisse, UK, Irlande, Pays-Bas, Allemagne, France,
- **Classe 5** : Islande, Norvège, Suède, Danemark.
- $\Longrightarrow$ Classification facilement intreprétable en termes **géographiques**

Package Cluster

- Le package `cluster` permet de lancer la CAH sans passer par la matrice de distances.
- Commandes :
  - `library(cluster)`

    Charge le package dans la session courante.
  - La fonction `agnes` permet de faire une CAH directement à partir des données (voir les options sur l'aide en ligne).

CAH : Exemple sous SAS

Obs	country	viandr	viandb	oeuf	lait	poisson	cereals	feculent	oleagine	fruitleg
1	Bulgaria	7.8	6.0	1.6	8.3	1.2	56.7	1.1	3.7	4.2
2	Yugoslavia	4.4	5.0	1.2	9.5	0.6	55.9	3.0	5.7	3.2
3	Romania	6.2	6.3	1.5	11.1	1.0	49.6	3.1	5.3	2.8
4	Russian Federat	9.3	4.6	2.1	16.6	3.0	43.6	6.4	3.4	2.9
5	Albania	10.1	1.4	0.5	8.9	0.2	42.3	0.6	5.5	1.7
6	Greece	10.2	3.0	2.8	17.6	5.9	41.7	2.2	7.8	6.5
7	Hungary	5.3	12.4	2.9	9.7	0.3	40.1	4.0	5.4	4.2
8	Italy	9.0	5.1	2.9	13.7	3.4	36.8	2.1	4.3	6.7
9	Poland	6.9	10.2	2.7	19.3	3.0	36.1	5.9	2.0	6.6
10	Czech Republic	9.7	11.4	2.8	12.5	2.0	34.3	5.0	1.1	4.0
11	Spain	7.1	3.4	3.1	8.6	7.0	29.2	5.7	5.9	7.2
12	France	18.0	9.9	3.3	19.5	5.7	28.1	4.8	2.4	6.5
13	Liechtenstein	8.9	14.0	4.3	19.9	2.1	28.0	3.6	1.3	4.3
14	Portugal	6.2	3.7	1.1	4.9	14.2	27.0	5.9	4.7	7.9
15	Belgium	13.5	9.3	4.1	17.5	4.5	26.6	5.7	2.1	4.0
16	Iceland	9.5	4.9	2.7	33.7	5.8	26.3	5.1	1.0	1.4
17	Switzerland	13.1	10.1	3.1	23.8	2.3	25.6	2.8	2.4	4.9
18	Ukraine	8.4	11.6	3.7	11.1	5.4	24.6	6.5	0.8	3.6
19	United Kingdom	17.4	5.7	4.7	20.6	4.3	24.3	4.7	3.4	3.3
20	Ireland	13.9	10.0	4.7	25.8	2.2	24.0	6.2	1.6	2.9
21	Norway	9.4	4.7	2.7	23.3	9.7	23.0	4.6	1.6	2.7
22	Netherlands	9.5	13.6	3.6	23.4	2.5	22.4	4.2	1.8	3.7
23	Denmark	10.6	10.8	3.7	25.0	9.9	21.9	4.8	0.7	2.4
24	Sweden	9.9	7.8	3.5	24.7	7.5	19.5	3.7	1.4	2.0
25	Germany	11.4	12.5	4.1	18.8	3.4	18.6	5.2	1.5	3.8

- Pour chaque pays
- Les consommations des 9 protéines.

Procédure Cluster

- Lancer la **CAH** directement à l'aide de la procédure `Cluster`,

```
/*********************************************/
/* 3. CAH sur les données standardisées */

proc cluster data=proteines outtree=proteines_cah_ward method=WARD STANDARD Pseudo Rsquare noprint;
var viandr - - fruitleg;
id country;
run;
```

- **Options** :
  - `method` : dissemblance utilisée pour regrouper les parties,
  - `STANDARD` : centre et réduit les données (évite les problèmes d'échelle),
  - `Pseudo, Rsquare` : calcul du PseudoF et du R^2,
  - `var` : variables sur lesquelles la CAH est effectuée,
  - `id` : nom des individus (le cas échéant).

Sortie de la procédure `Cluster`

	Name of Observation or Cluster	Parent of Observation or Cluster	Number of Clusters	Frequency of Cluster	Semi-Partial R-Squared	Root-Mean-Square Standard Deviation	Semi-Partial R-Squared	R-Squared	Pseudo F Statistic
1	CL1		1	25	0.3468712254	1	0.3468712254	0	.
2	CL2	CL1	2	8	0.1381375374	0.9493165052	0.1381375374	0.3468712254	12.215107484
3	CL3	CL1	3	17	0.0994926304	0.7651257169	0.0994926304	0.4850087627	10.359586735
4	CL4	CL3	4	12	0.0824272295	0.6802779539	0.0824272295	0.5845013931	9.847228568
5	CL5	CL2	5	4	0.0570082359	0.8372325145	0.0570082359	0.6669286226	10.011797289
6	CL6	CL4	6	8	0.0370631427	0.592418119	0.0370631427	0.7239368585	9.9649668809
7	CL7	CL3	7	5	0.0290642552	0.6870763482	0.0290642552	0.7610000012	9.552301318
8	CL8	CL7	8	4	0.0260000569	0.6300141553	0.0260000569	0.7900642564	9.1395940835
9	CL9	CL2	9	4	0.023953517	0.5447436836	0.023953517	0.8160643134	8.8733657755
10	CL10	CL6	10	6	0.0226035466	0.4898783002	0.0226035466	0.8400178303	8.7511609584
11	CL11	CL5	11	2	0.0199100856	0.6912612055	0.0199100856	0.8626213769	8.7908140355
12	CL12	CL8	12	3	0.0154308753	0.5323307905	0.0154308753	0.8825314625	8.8789028192
13	CL13	CL6	13	2	0.015303976	0.6060490274	0.015303976	0.8979623379	8.8003029384
14	CL14	CL4	14	4	0.0147463932	0.4674722586	0.0147463932	0.9132663139	8.9096156105
15	CL15	CL10	15	3	0.0120910754	0.4244334421	0.0120910754	0.928012707	9.2081003731
16	CL16	CL9	16	3	0.0108881827	0.3970847738	0.0108881827	0.9401037824	9.4173270448
17	CL17	CL5	17	2	0.010701464	0.506789044	0.010701464	0.9509919652	9.7024086831
18	CL18	CL12	18	2	0.0081837972	0.4431829567	0.0081837972	0.9616934291	10.337427839
19	CL19	CL14	19	3	0.0081509043	0.3883796478	0.0081509043	0.9698772264	10.732491391
20	CL20	CL10	20	3	0.0077262856	0.3854420402	0.0077262856	0.9780281307	11.713879248
21	CL21	CL20	21	2	0.0046541783	0.3342159162	0.0046541783	0.9857544163	13.839438747
22	CL22	CL19	22	2	0.0044189916	0.325662091	0.0044189916	0.9904085946	14.751429597
23	CL23	CL15	23	2	0.0029209035	0.2647672251	0.0029209035	0.9948275861	17.484848197
24	CL24	CL16	24	2	0.0022515104	0.2324569844	0.0022515104	0.9977484896	19.267230178

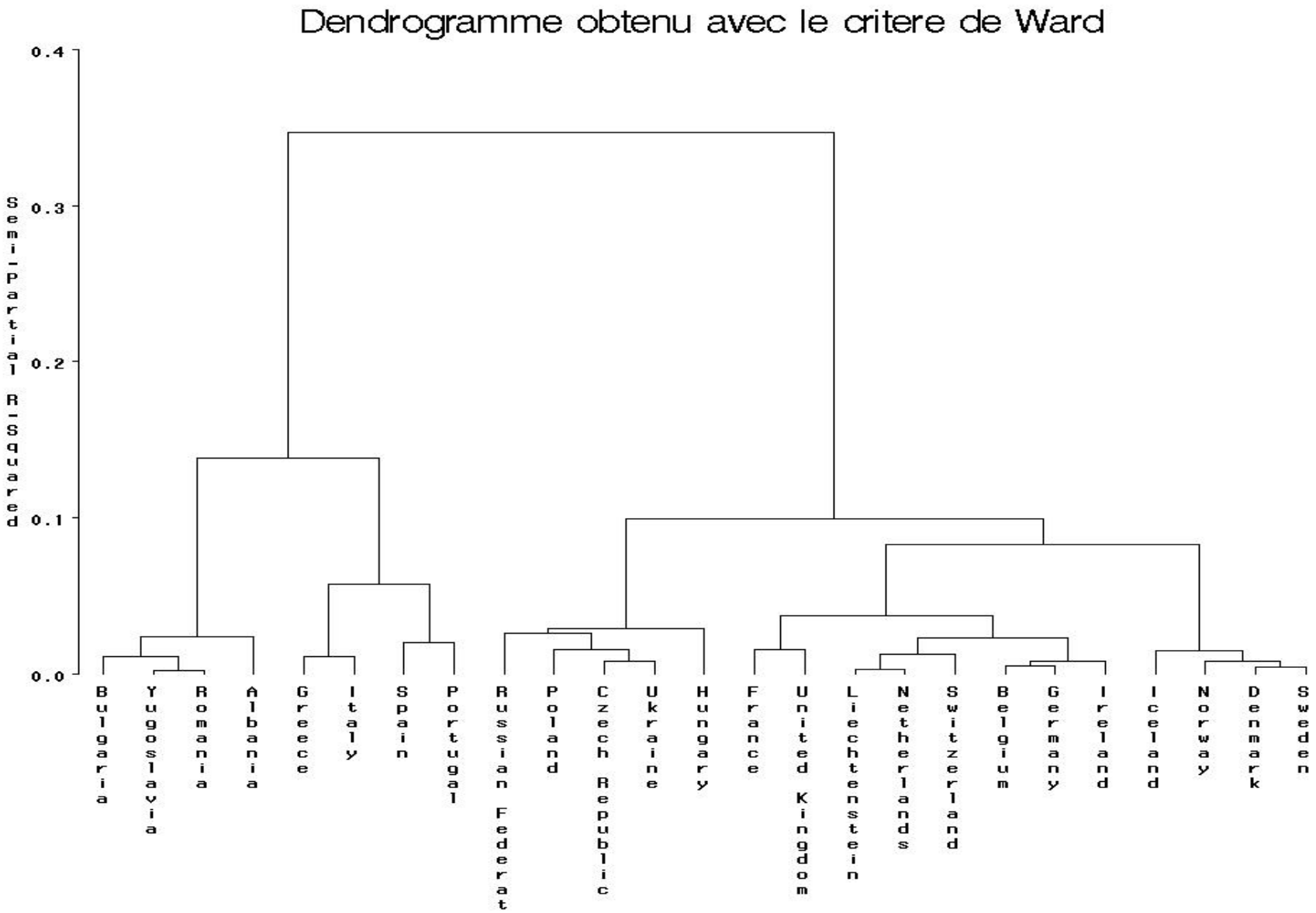
- **Chaîne des agrégations** : ordre d'agrégation des parties,
- **NCL** : nombre de parties (**clusters**) correspondant à l'étape d'agrégation,
- **RSQ, PSF** : R^2 et PseudoF correspondants à l'étape d'agrégation.

Dendrogramme et choix du nombre de classes

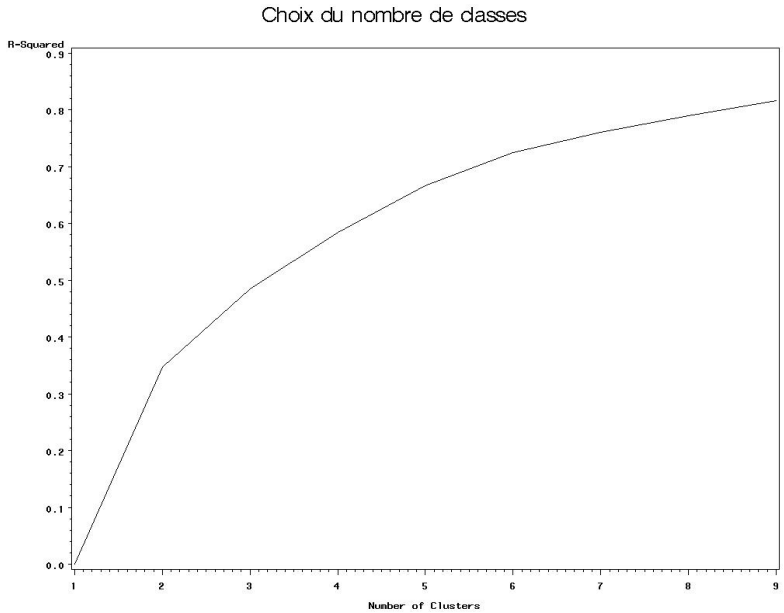
```
/*************************************************/
/* 4. Dendrogramme et choix du nombre de classes */
/* Dendrogramme */
title "Dendrogramme obtenu avec le critere de Ward";
proc tree data = proteines_cah_ward;
run;
/* Graphe du R2 et du pseudoF */
symbol1 i = join;
title "Choix du nombre de classes";
proc sort data = proteines_cah_ward;
    by _NCL_;
proc gplot data = proteines_cah_ward (where = (_NCL_ < 10));
    plot (_RSQ_ _PSF_) * _NCL_;
run;
quit;
```

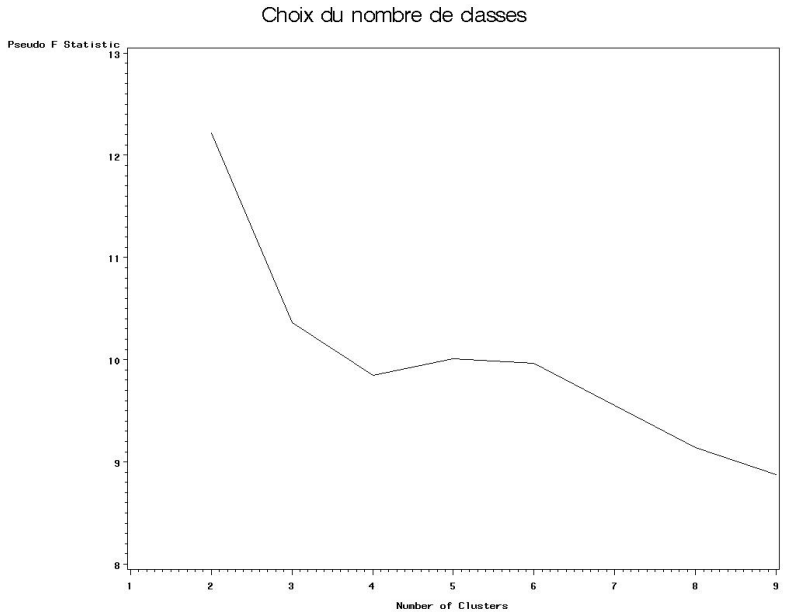
- **Sort** : range les valeurs de R^2 et PseudoF dans l'ordre de `NCL`,
- **plot** : trace R^2 et PseudoF en fonction du nombre de classes `NCL`.

Dendrogramme



R^2 et PseudoF





Partition des données en 2, 3 ou 4 classes

```
/*****************************************************************/
/* 5. Dendrogramme et extraction de la partition à 4, 3 et 2 classes */

proc tree data=proteines_cah_ward out=proteines_cah_ward_4 ncl=4 noprint;
run;

proc tree data=proteines_cah_ward out=proteines_cah_ward_3 ncl=3 noprint;
run;

proc tree data=proteines_cah_ward out=proteines_cah_ward_2 ncl=2 noprint;
run;
```

Option `ncl =` : choisit le nombre de classes dans le dendrogramme.
Résultat de la partition en 3 et 4 classes :

Obs	_NAME_	CLUSTER	CLUSNAME
1	Yugoslavia	1	CL9
2	Romania	1	CL9
3	Bulgaria	1	CL9
4	Albania	1	CL9
5	Liechtenstein	2	CL3
6	Netherlands	2	CL3
7	Denmark	2	CL3
8	Sweden	2	CL3
9	Belgium	2	CL3
10	Germany	2	CL3
11	Ireland	2	CL3
12	Norway	2	CL3
13	Czech Republic	2	CL3
14	Ukraine	2	CL3
15	Switzerland	2	CL3
16	Iceland	2	CL3
17	France	2	CL3
18	United Kingdom	2	CL3
19	Poland	2	CL3
20	Russian Federat	2	CL3
21	Hungary	2	CL3
22	Greece	3	CL5
23	Italy	3	CL5
24	Spain	3	CL5
25	Portugal	3	CL5

Obs	_NAME_	CLUSTER	CLUSNAME
1	Yugoslavia	1	CL9
2	Romania	1	CL9
3	Bulgaria	1	CL9
4	Albania	1	CL9
5	Liechtenstein	2	CL4
6	Netherlands	2	CL4
7	Denmark	2	CL4
8	Sweden	2	CL4
9	Belgium	2	CL4
10	Germany	2	CL4
11	Ireland	2	CL4
12	Norway	2	CL4
13	Switzerland	2	CL4
14	Iceland	2	CL4
15	France	2	CL4
16	United Kingdom	2	CL4
17	Czech Republic	3	CL7
18	Ukraine	3	CL7
19	Poland	3	CL7
20	Russian Federat	3	CL7
21	Hungary	3	CL7
22	Greece	4	CL5
23	Italy	4	CL5
24	Spain	4	CL5
25	Portugal	4	CL5

Partition en 4 classes : Résultat

Obs	_NAME_	CLUSTER	CLUSNAME
1	Yugoslavia	1	CL9
2	Romania	1	CL9
3	Bulgaria	1	CL9
4	Albania	1	CL9
5	Liechtenstein	2	CL4
6	Netherlands	2	CL4
7	Denmark	2	CL4
8	Sweden	2	CL4
9	Belgium	2	CL4
10	Germany	2	CL4
11	Ireland	2	CL4
12	Norway	2	CL4
13	Switzerland	2	CL4
14	Iceland	2	CL4
15	France	2	CL4
16	United Kingdom	2	CL4
17	Czech Republic	3	CL7
18	Ukraine	3	CL7
19	Poland	3	CL7
20	Russian Federat	3	CL7
21	Hungary	3	CL7
22	Greece	4	CL5
23	Italy	4	CL5
24	Spain	4	CL5
25	Portugal	4	CL5

- **Classe 1** : Bulgarie, Yougoslavie, Roumanie, Albanie,
- **Classe 2** : Lichtenstein, Belgique, Islande, Suisse, UK, Irlande, Norvège, Pays-Bas, Danemark, Suède, Allemagne, France,
- **Classe 3** : Russie, Pologne, Ukraine, Hongrie, Rep. Tchèque
- **Classe 4** : Grèce, Italie, Espagne, Portugal.
- La classification obtenue est facilement intreprêtable en termes géographiques

Statistiques

La procédure MEANS

CLUSTER	N Obs	Variable	Moyenne	Écart-type
1	4	viandr	9.08	3.89
		viandb	7.78	2.73
		oeuf	2.65	1.46
		lait	15.08	7.78
		poisson	4.05	4.26
		cereals	40.28	18.61
		feculent	3.65	2.04
		oleagine	3.05	2.15
		fruitleg	3.45	0.82
2	12	viandr	9.59	3.20
		viandb	7.23	3.72
		oeuf	2.70	1.04
		lait	16.55	7.14
		poisson	3.05	2.01
		cereals	35.44	8.53
		feculent	4.03	1.88
		oleagine	3.53	2.19
		fruitleg	4.05	1.82
3	5	viandr	12.06	4.13
		viandb	8.72	3.41
		oeuf	3.74	0.65
		lait	19.48	6.82
		poisson	5.16	2.29
		cereals	23.88	4.83
		feculent	5.12	0.95
		oleagine	2.56	1.91
		fruitleg	4.48	2.27
4	4	viandr	8.50	1.56
		viandb	9.00	5.56
		oeuf	2.93	1.38
		lait	17.88	8.80
		poisson	7.13	5.87
		cereals	25.10	2.81
		feculent	4.58	0.97

$\Longrightarrow$ Profilage des pays en fonction de leur consommation de protéines.

Centres Mobiles

Références

- **Algorithme** dû principalement à Forgy (1965) : E. Forgy, *Cluster analysis of multivariaet data : Efficiency versus interpretability of classification*, Biometrics, 21 :768 :780, 1965
- **Prémisses ou variantes** : Thorndike (1953), *k-means* MacQueen (1967), Ball & Hall (1967)
- **Généralisation marquante** : Algorithme des **nuées dynamiques** proposé par Diday (1971)
- Adapté aux **grands ensembles de données** :
  - traitement séquentiel possible
  - ne requiert pas le calcul d'une matrice $n \times n$ de distances entre individus

Algorithme (LMP)

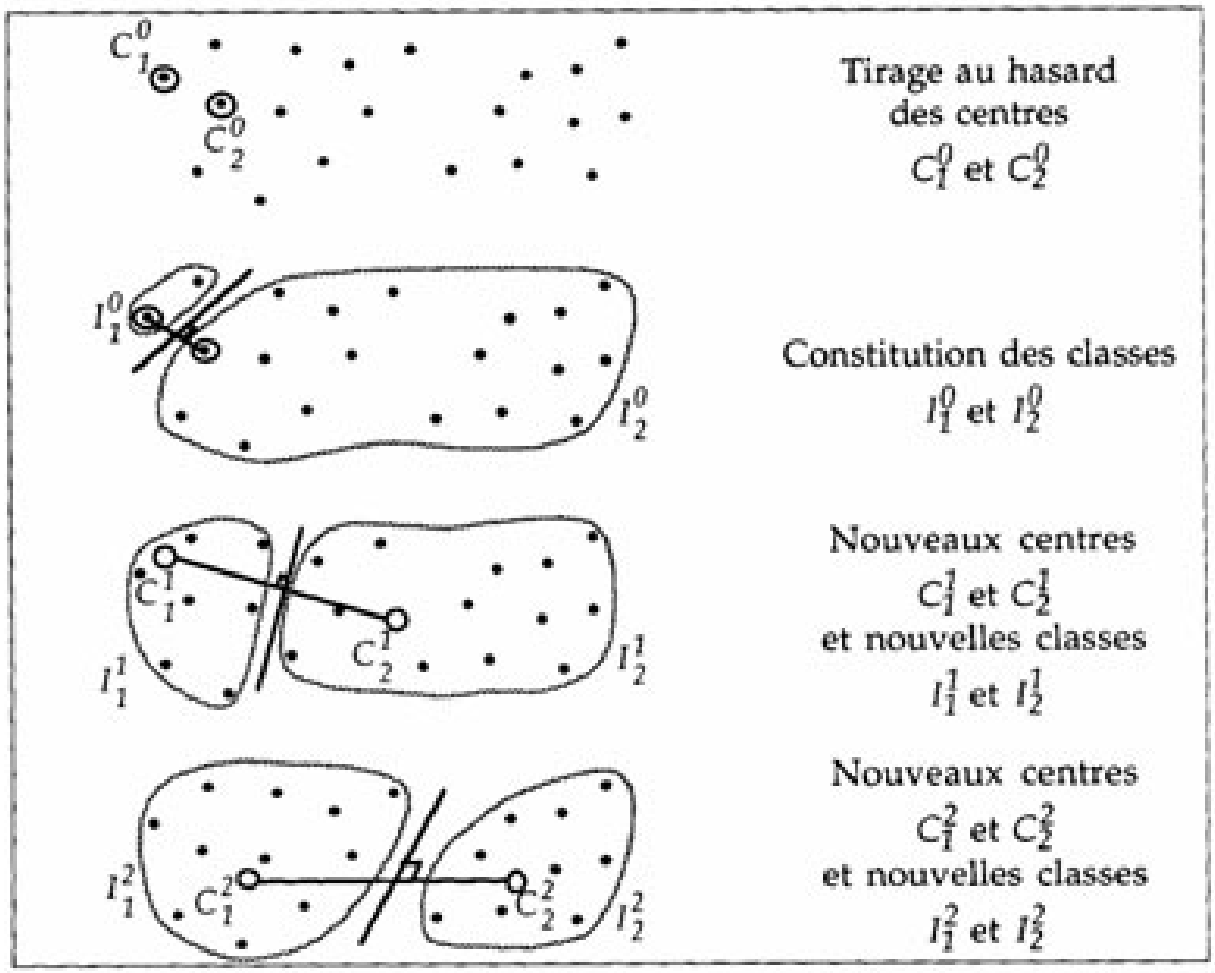


Figure 2.1 - 1
Etapes de l'algorithme

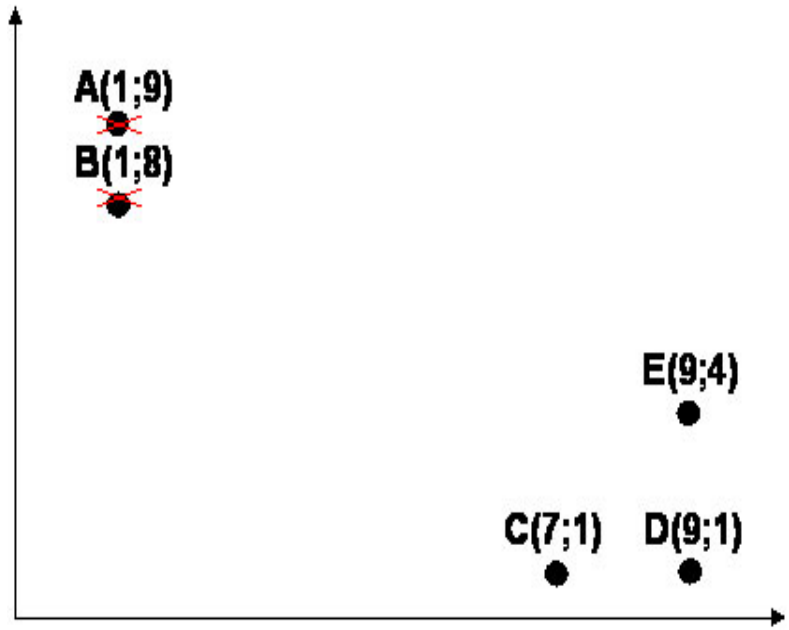
Description de l'algorithme

- n individus, décrits par p variables, à partitionner en k classes (k **fixé**)
- **Etape 0 :**
  - k centres provisoires $\{C_1^0, \ldots, C_k^0\}$ tirés au hasard.
  - k classes $\{P_1^0, \ldots, P_k^0\}$ crées à partir des centres en regroupant les données les plus proches de chaque centre.
  - Obtention de la partition $\mathcal{P}^0$.
- **Etape j :**
  - $\{C_1^j, \ldots, C_k^j\}$ centres de gravité des k classes $\{P_1^{j-1}, \ldots, P_k^{j-1}\}$ construites à l'étape $j-1$.
  - k nouvelles classes $\{P_1^j, \ldots, P_k^j\}$ créees à partir des nouveaux centres suivant la même règle qu'à l'étape 0.
  - Obtention de la partition $\mathcal{P}^j$.
- **Fin de l'algorithme :** l'algorithme converge vers une partition stable. Arrêt lorsque la partition reste la même, ou lorsque la variance intra-classes ne décroît plus, ou encore lorsque le nombre maximal d'itérations est atteint.

Exemple simple

$k = 2$ classes, centres initiaux choisis au hasard, par exemple A et B.

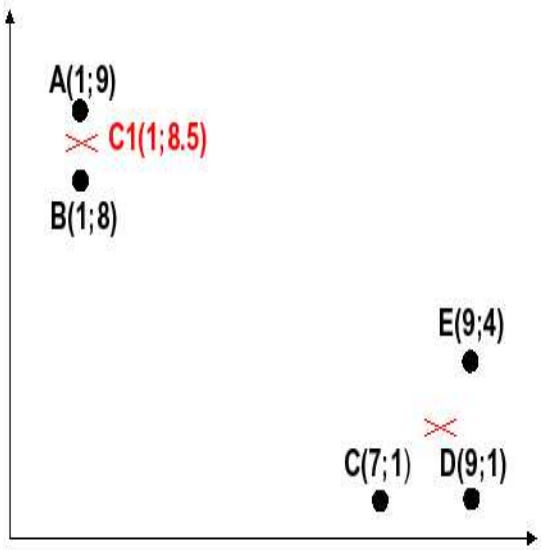
Etape 0 :



- C1=A attire le point A
- C2=B attire les points B, C, D et E.
- $\Longrightarrow$ **Nouveaux centres** : C1 (1 ; 9) et C2 (6,5 ; 3,5)

Exemple simple (2)

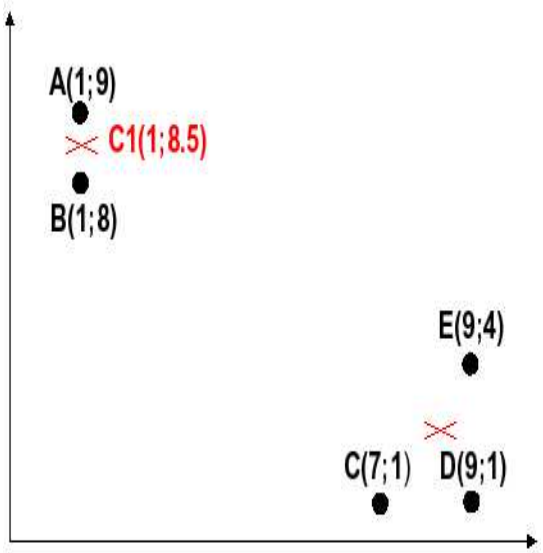
Etape 1 :



- C1 attire les point A et B
- C2 attire les points C, D et E.
- $\Longrightarrow$ **Nouveaux centres** : C1 (1 ; 8,5) et C2 (8,3 ; 2)

Exemple simple (3)

Etape 2 :



Mêmes classes que précédemment $\Longrightarrow$ la méthode s'arrête.

Atouts et limites des méthodes

- **Atouts et limites de la CAH**
  - **Atout** :
    - Fournit à la fois les classes et leur nombre
  - **Limites** :
    - Souvent malaisé de choisir la coupure significative sur le dendrogramme
    - Partition non-optimale en raison de sa structure hiérarchique
    - Fort coût algorithmique lorsque n devient grand
- **Atouts et limites des centres mobiles**
  - **Atouts** :
    - Coût algorithmique faible
    - Traitement séquentiel
  - **Limites** :
    - Nombre de classes fixé a priori
    - Partition obtenue fortement dépendante des centres provisoires des classes
- **Idée** : Mixer les 2 méthodes (CAH et centres mobiles)

Classification mixte

- **Etape 1 : Partitionnement préliminaire (si n grand)**
  Partitionnement en q classes, avec $n >> q >> k$ le nombre de classes final désiré, en utilisant la méthode des **centres mobiles** ($q \simeq 10$ ou 100)
- **Etape 2 : Classification ascendante hiérarchique**
  CAH sur les q éléments (centres) obtenus à l'étape 1
- **Etape 3 : Optimisation**
  - Partition finale obtenue par coupure de l'arbre de la CAH
  - Homogénéité des classes optimisée par réaffectation par la technique des centres mobiles (consolidation)
- **Attention :** Méthode qui peut être instable sur les échantillons de petite taille.

Centres Mobiles : Exemple sous R

```
> proteine
                country viandr viandb. oeuf lait poisson cereals feculent oleagine fruitleg
1              Bulgaria    7.8     6.0   1.6   8.3     1.2    56.7     1.1      3.7      4.2
2            Yugoslavia    4.4     5.0   1.2   9.5     0.6    55.9     3.0      5.7      3.2
3               Romania    6.2     6.3   1.5  11.1     1.0    49.6     3.1      5.3      2.8
4  Russian Federation      9.3     4.6   2.1  16.6     3.0    43.6     6.4      3.4      2.9
5               Albania   10.1     1.4   0.5   8.9     0.2    42.3     0.6      5.5      1.7
6                Greece   10.2     3.0   2.8  17.6     5.9    41.7     2.2      7.8      6.5
7               Hungary    5.3    12.4   2.9   9.7     0.3    40.1     4.0      5.4      4.2
8                 Italy    9.0     5.1   2.9  13.7     3.4    36.8     2.1      4.3      6.7
9                Poland    6.9    10.2   2.7  19.3     3.0    36.1     5.9      2.0      6.6
10       Czech Republic    9.7    11.4   2.8  12.5     2.0    34.3     5.0      1.1      4.0
11                Spain    7.1     3.4   3.1   8.6     7.0    29.2     5.7      5.9      7.2
12               France   18.0     9.9   3.3  19.5     5.7    28.1     4.8      2.4      6.5
13        Liechtenstein    8.9    14.0   4.3  19.9     2.1    28.0     3.6      1.3      4.3
14             Portugal    6.2     3.7   1.1   4.9    14.2    27.0     5.9      4.7      7.9
15              Belgium   13.5     9.3   4.1  17.5     4.5    26.6     5.7      2.1      4.0
16              Iceland    9.5     4.9   2.7  33.7     5.8    26.3     5.1      1.0      1.4
17          Switzerland   13.1    10.1   3.1  23.8     2.3    25.6     2.8      2.4      4.9
18              Ukraine    8.4    11.6   3.7  11.1     5.4    24.6     6.5      0.8      3.6
19       United Kingdom   17.4     5.7   4.7  20.6     4.3    24.3     4.7      3.4      3.3
20              Ireland   13.9    10.0   4.7  25.8     2.2    24.0     6.2      1.6      2.9
21               Norway    9.4     4.7   2.7  23.3     9.7    23.0     4.6      1.6      2.7
22          Netherlands    9.5    13.6   3.6  23.4     2.5    22.4     4.2      1.8      3.7
23              Denmark   10.6    10.8   3.7  25.0     9.9    21.9     4.8      0.7      2.4
24               Sweden    9.9     7.8   3.5  24.7     7.5    19.5     3.7      1.4      2.0
25              Germany   11.4    12.5   4.1  18.8     3.4    18.6     5.2      1.5      3.8
```

- Pour chaque pays
- Les consommations des 9 protéines.

Fonction kmeans

- `proteinestd = scale(proteine[,-1])`

  Centre et réduit les données.

- `nclass=4`
  `km = kmeans(proteinestd, nclass, nstart=10)`

  Lance l'algorithme des centres mobiles pour une partition en `nclass = 4` classes sur les variables quantitatives centrées et réduites. On prend la moyenne des classifications effectuées avec `nstart = 10` initiations différentes afin de stabiliser les résultats.

- Détail des sorties :
  - `km$cluster` : répartition des observations dans les parties

```
> km$cluster
 [1] 2 2 2 2 2 3 2 3 1 1 4 1 1 4 1 1 1 1 1 1 1 1 1 1 1
```

Sorties de la fonction `kmeans`

- `km$size` : répartition des données dans chaque classe

```
> km$size
[1] 15  6  2  2
```

- `km$centers` : coordonnées des centres de classes (qui sont aussi les moyennes dans chaque classe pour chaque variable)

```
> km$centers
      viandr    viandb.       oeuf       lait    poisson    cereals   feculent   oleagine   fruitleg
1  0.45173726  0.5063957  0.57622628  0.5837801  0.1183432 -0.6100043  0.3533068 -0.7043759 -0.2195240
2 -0.79014185 -0.5267887 -1.16557572 -0.9047559 -0.9504683  1.4383272 -0.7604664  0.8870168 -0.5373533
3 -0.06811911 -1.0411250 -0.07694947 -0.2057585  0.1075669  0.6380079 -1.3010340  1.4997366  1.3659270
4 -0.94948480 -1.1764767 -0.74802044 -1.4583242  1.8562639 -0.3779572  0.9326321  1.1220326  1.8925628
```

Attention : centres calculés sur les données **standardisées**.

Sorties de la fonction kmeans (2)

- km$withinss : variances intra-classe

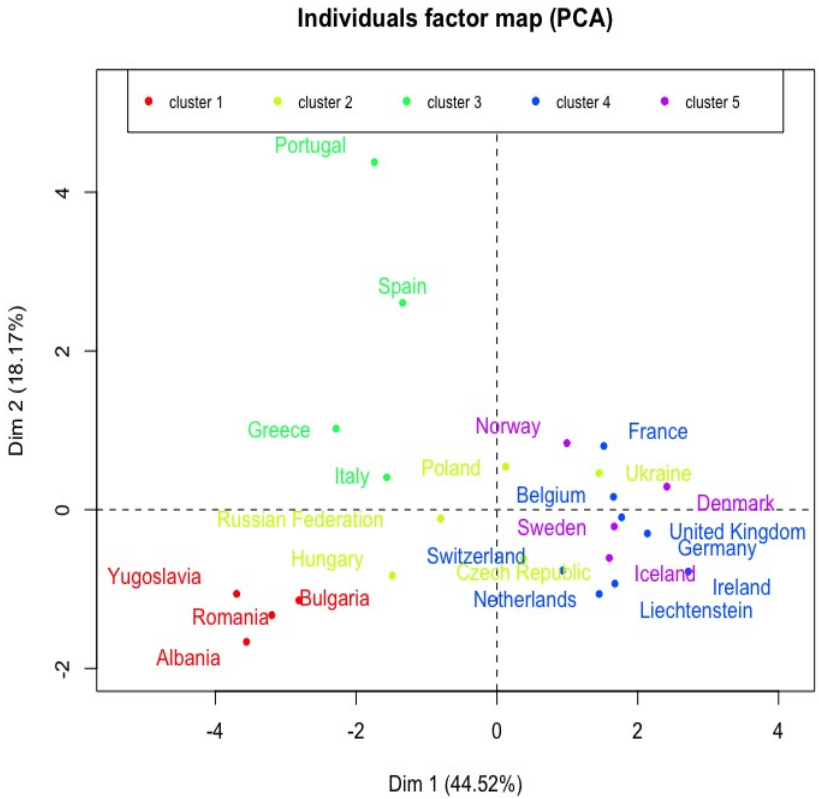
```
> km$withinss
[1] 62.969328 24.091128  2.311516  4.300578
```

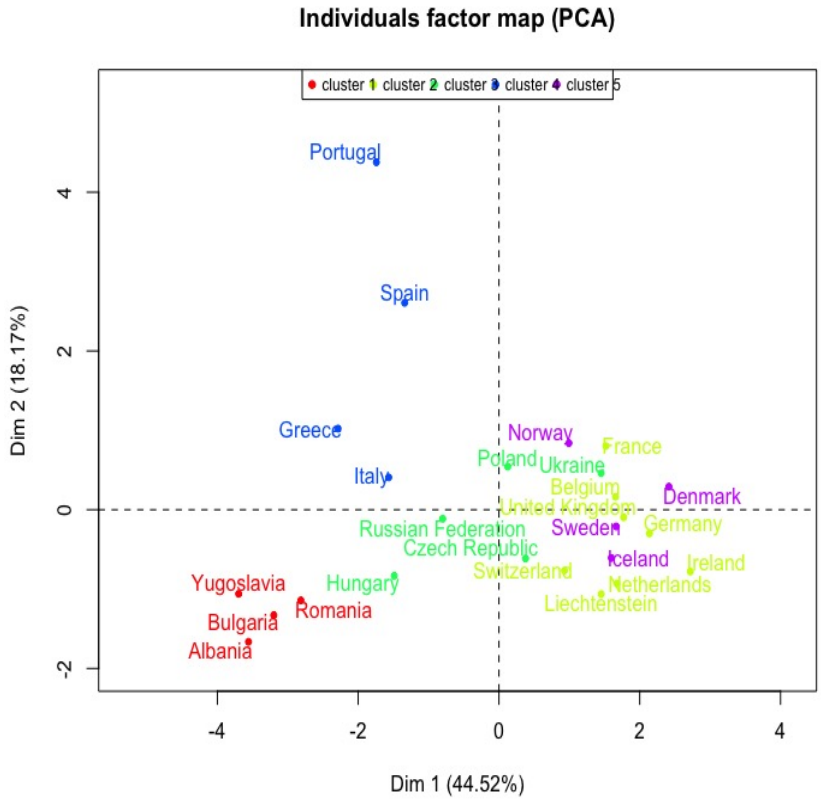
- moyenne et ecartcm : moyennes (si kmeans pas appliquée sur données brutes) et écart-types pour chaque variable dans chaque classe

```
> moyenne
                1        2     3     4
viandr   11.340000  7.183333  9.60  6.65
viandb.   9.766667  5.950000  4.05  3.55
oeuf      3.580000  1.633333  2.85  2.10
lait     21.260000 10.683333 15.65  6.75
poisson   4.686667  1.050000  4.65 10.60
cereals  25.553333 48.033333 39.25 28.10
feculent  4.853333  3.033333  2.15  5.80
oleagine  1.673333  4.833333  6.05  5.30
fruitleg  3.740000  3.166667  6.60  7.55
```

```
> ecartcm
                1         2          3         4
viandr   3.2244158 2.2639935 0.84852814 0.6363961
viandb.  2.8974537 3.6098476 1.48492424 0.2121320
oeuf     0.6991832 0.8140434 0.07071068 1.4142136
lait     5.5146299 3.0465828 2.75771645 2.6162951
poisson  2.6492227 1.0310189 1.76776695 5.0911688
cereals  4.7925339 7.1402147 3.46482323 1.5556349
feculent 1.0063134 2.0944371 0.07071068 0.1414214
oleagine 0.7116045 1.0073066 2.47487373 0.8485281
fruitleg 1.4613106 0.9479803 0.14142136 0.4949747
```

Comparaison avec la CAH





Centres Mobiles : Exemple sous SAS

Obs	country	viandr	viandb	oeuf	lait	poisson	cereals	feculent	oleagine	fruitleg
1	Bulgaria	7.8	6.0	1.6	8.3	1.2	56.7	1.1	3.7	4.2
2	Yugoslavia	4.4	5.0	1.2	9.5	0.6	55.9	3.0	5.7	3.2
3	Romania	6.2	6.3	1.5	11.1	1.0	49.6	3.1	5.3	2.8
4	Russian Federat	9.3	4.6	2.1	16.6	3.0	43.6	6.4	3.4	2.9
5	Albania	10.1	1.4	0.5	8.9	0.2	42.3	0.6	5.5	1.7
6	Greece	10.2	3.0	2.8	17.6	5.9	41.7	2.2	7.8	6.5
7	Hungary	5.3	12.4	2.9	9.7	0.3	40.1	4.0	5.4	4.2
8	Italy	9.0	5.1	2.9	13.7	3.4	36.8	2.1	4.3	6.7
9	Poland	6.9	10.2	2.7	19.3	3.0	36.1	5.9	2.0	6.6
10	Czech Republic	9.7	11.4	2.8	12.5	2.0	34.3	5.0	1.1	4.0
11	Spain	7.1	3.4	3.1	8.6	7.0	29.2	5.7	5.9	7.2
12	France	18.0	9.9	3.3	19.5	5.7	28.1	4.8	2.4	6.5
13	Liechtenstein	8.9	14.0	4.3	19.9	2.1	28.0	3.6	1.3	4.3
14	Portugal	6.2	3.7	1.1	4.9	14.2	27.0	5.9	4.7	7.9
15	Belgium	13.5	9.3	4.1	17.5	4.5	26.6	5.7	2.1	4.0
16	Iceland	9.5	4.9	2.7	33.7	5.8	26.3	5.1	1.0	1.4
17	Switzerland	13.1	10.1	3.1	23.8	2.3	25.6	2.8	2.4	4.9
18	Ukraine	8.4	11.6	3.7	11.1	5.4	24.6	6.5	0.8	3.6
19	United Kingdom	17.4	5.7	4.7	20.6	4.3	24.3	4.7	3.4	3.3
20	Ireland	13.9	10.0	4.7	25.8	2.2	24.0	6.2	1.6	2.9
21	Norway	9.4	4.7	2.7	23.3	9.7	23.0	4.6	1.6	2.7
22	Netherlands	9.5	13.6	3.6	23.4	2.5	22.4	4.2	1.8	3.7
23	Denmark	10.6	10.8	3.7	25.0	9.9	21.9	4.8	0.7	2.4
24	Sweden	9.9	7.8	3.5	24.7	7.5	19.5	3.7	1.4	2.0
25	Germany	11.4	12.5	4.1	18.8	3.4	18.6	5.2	1.5	3.8

- Pour chaque pays
- Les consommations des 9 protéines.

Procédure Fastclus

Si nécessaire : standardiser les données

```
/*************************************************/
/* 2. Standardisation des données   */

proc standard data=proteines out=proteinestd mean=0 std=1;
var viandr -- fruitleg;
run;
```

Lancer la procédure sur les données avec $k = 4$ classes

```
/*************************************************/
/* 3. Centres mobiles à 4 classes sur données standardisées */

proc fastclus data=proteinestd out=proteines_cm_4 maxclusters=4 noprint maxiter=30 cluster=CM4 vardef=n;
var viandr -- fruitleg;
run;
```

Sortie de la procédure Fastclus

	viandr	viandb	oeuf	lait	poisson	cereals	feculent	oleagine	fruitleg	Cluster	Distance to Cluster Seed
1	-0.605901566	-0.51325352	-1.195401094	-1.240180769	-0.906383469	2.2280160572	-1.943595511	0.3162641218	0.0354786226	3	1.3390666847
2	-1.62171287	-0.783956853	-1.553305613	-1.071295508	-1.082722666	2.1551216991	-0.780865199	1.3234747007	-0.518874856	3	1.1821421418
3	-1.083930415	-0.43204252	-1.284877223	-0.846115159	-0.965163201	1.5810786286	-0.719668867	1.1220325849	-0.740616247	3	0.9701051511
4	-0.15774952	-0.892238187	-0.748020445	-0.072057712	-0.37736588	1.0343709423	1.2998100959	0.1651825349	-0.685180899	2	2.8635887338
5	0.0812649043	-1.758488853	-2.179638521	-1.155738138	-1.20028213	0.9159176103	-2.249577172	1.2227536428	-1.350405073	3	1.9698907964
6	0.1111417073	-1.32536352	-0.121687536	0.0686800063	0.4749402355	0.8612468417	-1.270435856	2.3810458086	1.3104916229	1	1.9173666821
7	-1.352821642	1.2192478134	-0.032211407	-1.043147964	-1.170892264	0.7154581254	-0.168901877	1.1723931139	0.0354786226	1	2.3345134735
8	-0.247379929	-0.75688652	-0.032211407	-0.480197093	-0.259806416	0.4147688979	-1.331632189	0.6184272955	1.4213623185	1	1.1662751826
9	-0.874792793	0.62370048	-0.211163666	0.3079341265	-0.37736588	0.3509863346	0.9938284349	-0.53986487	1.3659269707	2	2.4825489369
10	-0.038242308	0.94854448	-0.121687536	-0.649082354	-0.671264541	0.1869740287	0.4430614451	-0.993109631	-0.075392073	2	1.8680716629
11	-0.815039187	-1.217082187	0.1467408528	-1.197959454	0.7982287621	-0.277727505	0.8714357705	1.4241957586	1.6985390577	4	1.4663864579
12	2.4415323453	0.54248948	0.3256931123	0.33608167	0.4161605034	-0.377957247	0.3206687807	-0.338422755	1.3104916229	2	2.6190944971
13	-0.277256732	1.6523731467	1.2204544099	0.3923767572	-0.641874675	-0.387069042	-0.413687206	-0.892388573	0.0909139705	2	1.9880944705
14	-1.083930415	-1.135871187	-1.642781742	-1.718689009	2.9142991183	-0.47818699	0.9938284349	0.8198694112	2.0865864926	4	1.4663864579
15	1.097076208	0.38006748	1.0415021504	0.0546062345	0.0634821107	-0.514634169	0.8714357705	-0.489504341	-0.075392073	2	1.1288602852
16	-0.097995914	-0.811027187	-0.211163666	2.3345572625	0.4455503694	-0.541969553	0.5042577773	-1.04347016	-1.516711117	2	2.7170545989
17	0.9775689958	0.5966301467	0.1467408528	0.9412538565	-0.583094943	-0.605752116	-0.903257863	-0.338422755	0.4235260575	2	1.835153158
18	-0.426640748	1.0026851467	0.6835976314	-0.846115159	0.3279909052	-0.696870064	1.3610064281	-1.144191218	-0.297133464	2	2.0620739151
19	2.262271527	-0.59446452	1.578358929	0.4908931596	0.0047023786	-0.724205448	0.2594724485	0.1651825349	-0.463439508	2	2.5322364728
20	1.2165834202	0.5695598134	1.578358929	1.222729292	-0.612484809	-0.751540833	1.1774174315	-0.741306986	-0.685180899	2	1.9079422531
21	-0.127872717	-0.865167853	-0.211163666	0.8708849976	1.5917551457	-0.84265878	0.1982761163	-0.741306986	-0.796051595	2	2.2969576944
22	-0.097995914	1.5440918134	0.5941215016	0.8849587694	-0.52431521	-0.897329549	-0.046509212	-0.640585928	-0.241698117	2	1.5466027228
23	0.2306489196	0.78612248	0.6835976314	1.1101391178	1.6505348778	-0.942888523	0.3206687807	-1.194551747	-0.962357638	2	1.9934814685
24	0.0215112982	-0.02598752	0.5046453718	1.0679178025	0.9451780924	-1.161571597	-0.352490873	-0.842028044	-1.18409903	2	1.8135011489
25	0.469663344	1.2463181467	1.0415021504	0.2375652676	-0.259806416	-1.24357775	0.5654541095	-0.791667515	-0.186262769	2	1.3383667944

- **Cluster** : numéro de la partie à laquelle l'individu appartient,
- **Distance to Cluster Seed** : distance de l'individu au centre de classe correspondant.

Résultats

Affichage de la classification

```
/*************************************************************************/
/* 4. Description des classes */

/* impression des classes et des pays */
proc print data=proteines_cm_4 (keep=country CM4);
run;
```

Obs	country	CM4
1	Bulgaria	3
2	Yugoslavia	3
3	Romania	3
4	Russian Federat	2
5	Albania	3
6	Greece	1
7	Hungary	1
8	Italy	1
9	Poland	2
10	Czech Republic	2
11	Spain	4
12	France	2
13	Liechtenstein	2
14	Portugal	4
15	Belgium	2
16	Iceland	2
17	Switzerland	2
18	Ukraine	2
19	United Kingdom	2
20	Ireland	2
21	Norway	2
22	Netherlands	2
23	Denmark	2
24	Sweden	2
25	Germany	2

Résultats (2)

Statistiques et Profils

```
/* fusion pour récupération des classes avec le jeu de données de départ */
data proteines_class4;
    merge proteines proteines_cm_4(keep = CM4);
run;
title "Description des classes";
proc means data = proteines_class4 mean std maxdec=2;
var viandr -- fruitleg;
class CM4;
run;
```

La procédure MEANS

Cluster	N Obs	Variable	Moyenne	Écart-type
1	3	viandr	8.17	2.55
		viandb	6.83	4.93
		oeuf	2.87	0.06
		lait	13.67	3.95
		poisson	3.20	2.81
		cereals	39.53	2.50
		feculent	2.77	1.07
		oleagine	5.83	1.79
		fruitleg	5.80	1.39
2	16	viandr	11.21	3.16
		viandb	9.44	3.08
		oeuf	3.49	0.77
		lait	20.97	5.45
		poisson	4.58	2.59
		cereals	26.68	6.46
		feculent	4.95	1.05
		oleagine	1.78	0.81
		fruitleg	3.69	1.43
3	4	viandr	7.13	2.42
		viandb	4.68	2.25
		oeuf	1.20	0.50
		lait	9.45	1.20
		poisson	0.75	0.44
		cereals	51.13	6.69
		feculent	1.95	1.29
		oleagine	5.05	0.91
		fruitleg	2.98	1.03
4	2	viandr	6.65	0.64

Comparaison avec la CAH

- Lancer la CAH et enregistrer la classification à 4 classes,
- fusionner les 2 classifications (centres mobiles et CAH),
- lancer une procédure `freq` sur les 2 colonnes correspondantes.

```
/*************************************************************************/
/* 5. Comparaison avec la CAH */

proc cluster data=proteines outtree=proteines_cah_ward method=WARD STANDARD noprint;
var viandr - - fruitleg;
id country;
run;
proc tree data=proteines_cah_ward out=proteines_cah_ward_4 ncl=4 noprint;
run;
proc print data=proteines_cah_ward_4;
run;
/* fusion des 2 classifications */
data compar_proteines;
merge proteines_cm_4 (keep = CM4) proteines_cah_ward_4 (keep = CLUSTER);
run;
title "Comparaison avec la CAH";
proc freq data = compar_proteines;
tables CM4*CLUSTER / nopercent nocol norow;
run;
```

Comparaison avec la CAH : Résultats

Obs	country	CM4
1	Bulgaria	3
2	Yugoslavia	3
3	Romania	3
4	Russian Federat	2
5	Albania	3
6	Greece	1
7	Hungary	1
8	Italy	1
9	Poland	2
10	Czech Republic	2
11	Spain	4
12	France	2
13	Liechtenstein	2
14	Portugal	4
15	Belgium	2
16	Iceland	2
17	Switzerland	2
18	Ukraine	2
19	United Kingdom	2
20	Ireland	2
21	Norway	2
22	Netherlands	2
23	Denmark	2
24	Sweden	2
25	Germany	2

Obs	_NAME_	CLUSTER	CLUSNAME
1	Yugoslavia	1	CL9
2	Romania	1	CL9
3	Liechtenstein	2	CL4
4	Netherlands	2	CL4
5	Denmark	2	CL4
6	Sweden	2	CL4
7	Belgium	2	CL4
8	Germany	2	CL4
9	Ireland	2	CL4
10	Norway	2	CL4
11	Czech Republic	3	CL7
12	Ukraine	3	CL7
13	Greece	4	CL5
14	Italy	4	CL5
15	Bulgaria	1	CL9
16	Switzerland	2	CL4
17	Iceland	2	CL4
18	France	2	CL4
19	United Kingdom	2	CL4
20	Poland	3	CL7
21	Spain	4	CL5
22	Portugal	4	CL5
23	Albania	1	CL9
24	Russian Federat	3	CL7
25	Hungary	3	CL7

Comparaison avec la CAH : Résultats

Obs	country	CM4
1	Bulgaria	3
2	Yugoslavia	3
3	Romania	3
4	Russian Federat	2
5	Albania	3
6	Greece	1
7	Hungary	1
8	Italy	1
9	Poland	2
10	Czech Republic	2
11	Spain	4
12	France	2
13	Liechtenstein	2
14	Portugal	4
15	Belgium	2
16	Iceland	2
17	Switzerland	2
18	Ukraine	2
19	United Kingdom	2
20	Ireland	2
21	Norway	2
22	Netherlands	2
23	Denmark	2
24	Sweden	2
25	Germany	2

Obs	_NAME_	CLUSTER	CLUSNAME
1	Yugoslavia	1	CL9
2	Romania	1	CL9
3	Liechtenstein	2	CL4
4	Netherlands	2	CL4
5	Denmark	2	CL4
6	Sweden	2	CL4
7	Belgium	2	CL4
8	Germany	2	CL4
9	Ireland	2	CL4
10	Norway	2	CL4
11	Czech Republic	3	CL7
12	Ukraine	3	CL7
13	Greece	4	CL5
14	Italy	4	CL5
15	Bulgaria	1	CL9
16	Switzerland	2	CL4
17	Iceland	2	CL4
18	France	2	CL4
19	United Kingdom	2	CL4
20	Poland	3	CL7
21	Spain	4	CL5
22	Portugal	4	CL5
23	Albania	1	CL9
24	Russian Federat	3	CL7
25	Hungary	3	CL7

La procédure FREQ

Table de CM4 par CLUSTER

CM4(Cluster) CLUSTER

FREQUENCE	1	2	3	4	Total
1	0	3	0	0	3
2	2	7	4	3	16
3	2	2	0	0	4
4	0	0	1	1	2
Total	4	12	5	4	25

Sauvegarde des centres de classes

Enregistrer les coordonnées des centres de classes : utiliser l'option `MEAN` =<*nom de la table*> dans la procédure `fastclus`.

Sortie :

	Optimization Criterion	Cluster	Frequency of Cluster	Root-Mean-Square Standard Deviation	Maximum Distance from Cluster Seed	Nearest Cluster	Distance to Nearest Cluster	viandr	viandb	oeuf	lait	
1	0.6481265387	1	3	0.6232069377	2.3345134735	3	2.7397511615	-0.496353288	-0.287667409	-0.062036783	-0.48488835	-0.
2	0.6481265387	2	16	0.7057115424	2.8635887338	1	3.3984103866	0.4136443382	0.4189810842	0.4934608556	0.542790193	0.0
3	0.6481265387	3	4	0.4717618686	1.9698907964	1	2.7397511615	-0.807569986	-0.871935437	-1.553305613	-1.078332394	-1.
4	0.6481265387	4	2	0.488795486	1.4663864579	1	3.5619968771	-0.949484801	-1.176476687	-0.748020445	-1.458324232	1.8

- **Cluster** : numéro de la partie,
- **Frequency of Cluster** : nombre d'observations dans la partie,
- **Standard Deviation** : écart-type de la partie.

Attention : coordonnées calculées sur les données **standardisées**.

Classification sur Facteurs Principaux

Passer aux coordonnées factorielles

- **Sélection d'un sous-espace factoriel** de dimension suffisante $q < p$
- **Classification** des n individus représentés par leurs composantes sur les q premiers **axes factoriels**.

Passer aux coordonnées factorielles

- **Sélection d'un sous-espace factoriel** de dimension suffisante $q < p$
- **Classification** des n individus représentés par leurs composantes sur les q premiers **axes factoriels**.
- **Remarque** : équivalence entre les classifications des n individus sur l'ensemble
  - des p variables,
  - des p facteurs issus d'une analyse factorielle,

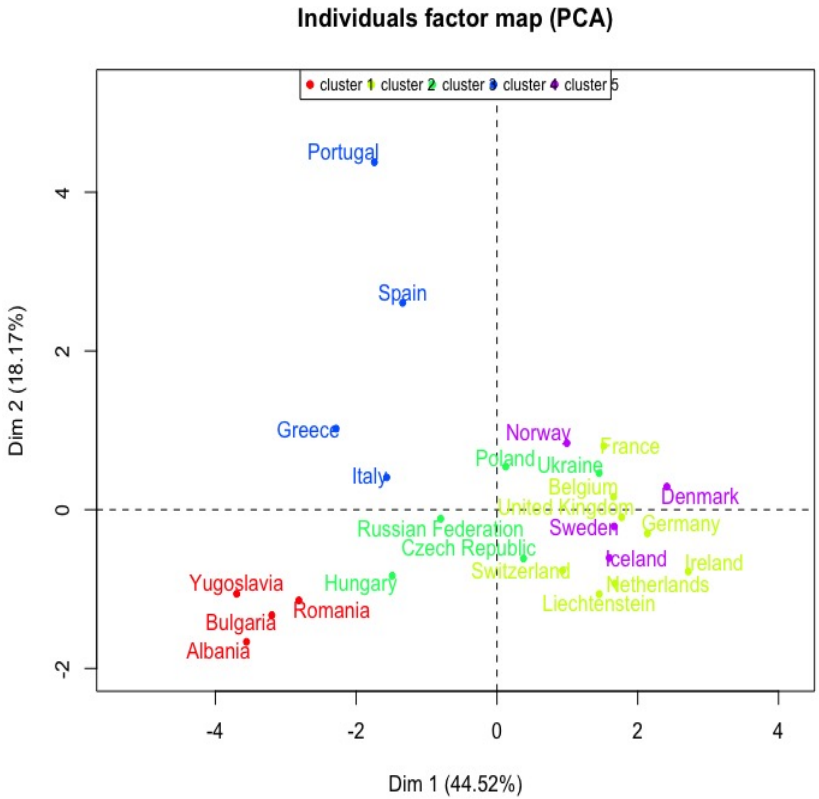
  seule la représentation des données change (changement de base).

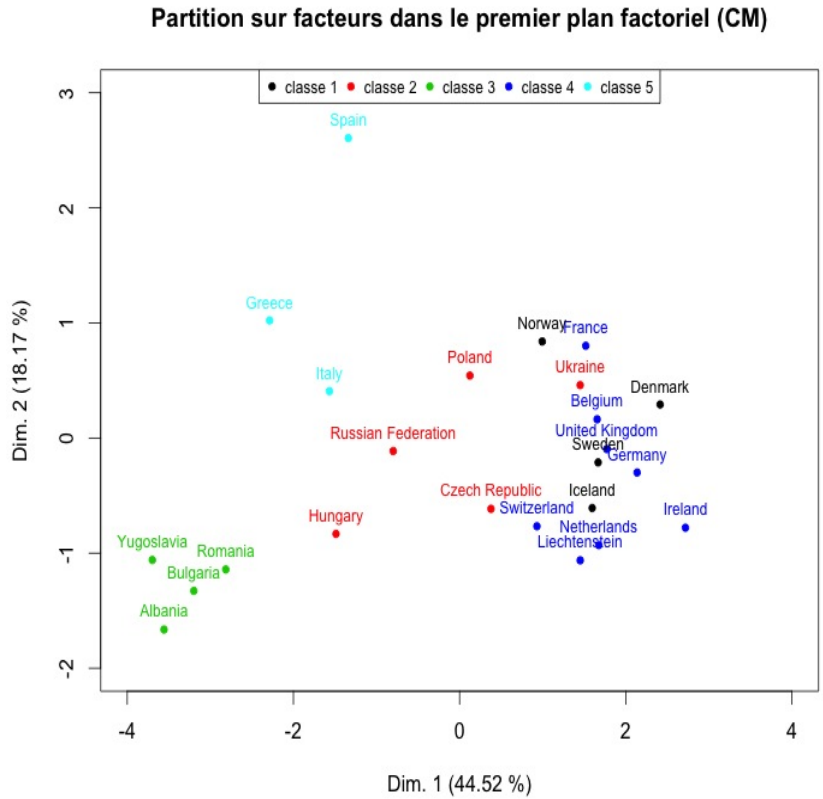
Pourquoi les coordonnées factorielles ?

- **Variables quantitatives** : **Analyse en Composantes Principales (ACP)**.
  - **Elimination des fluctuations aléatoires** constituant l'essentiel de la variance prise en compte par les $(p - q)$ derniers axes.
  - **Lissage** des données.
  - Production de classes **plus homogènes**.
- **Variables qualitatives** : **Analyse des Correspondances Multiples (ACM)**.
  - **Transformation des variables** en variables quantitatives via un **tableau disjonctif complet**.
- **Remarque :** ACM nécessaire pour gérer les variables qualitatives, par contre ACP pas toujours utile pour les variables quantitatives.

Exemple sous R : Classification sur Facteurs Principaux

Classification sur `proteine.acp$ind$coord[,1:4]` $\Longrightarrow$ 5 classes.



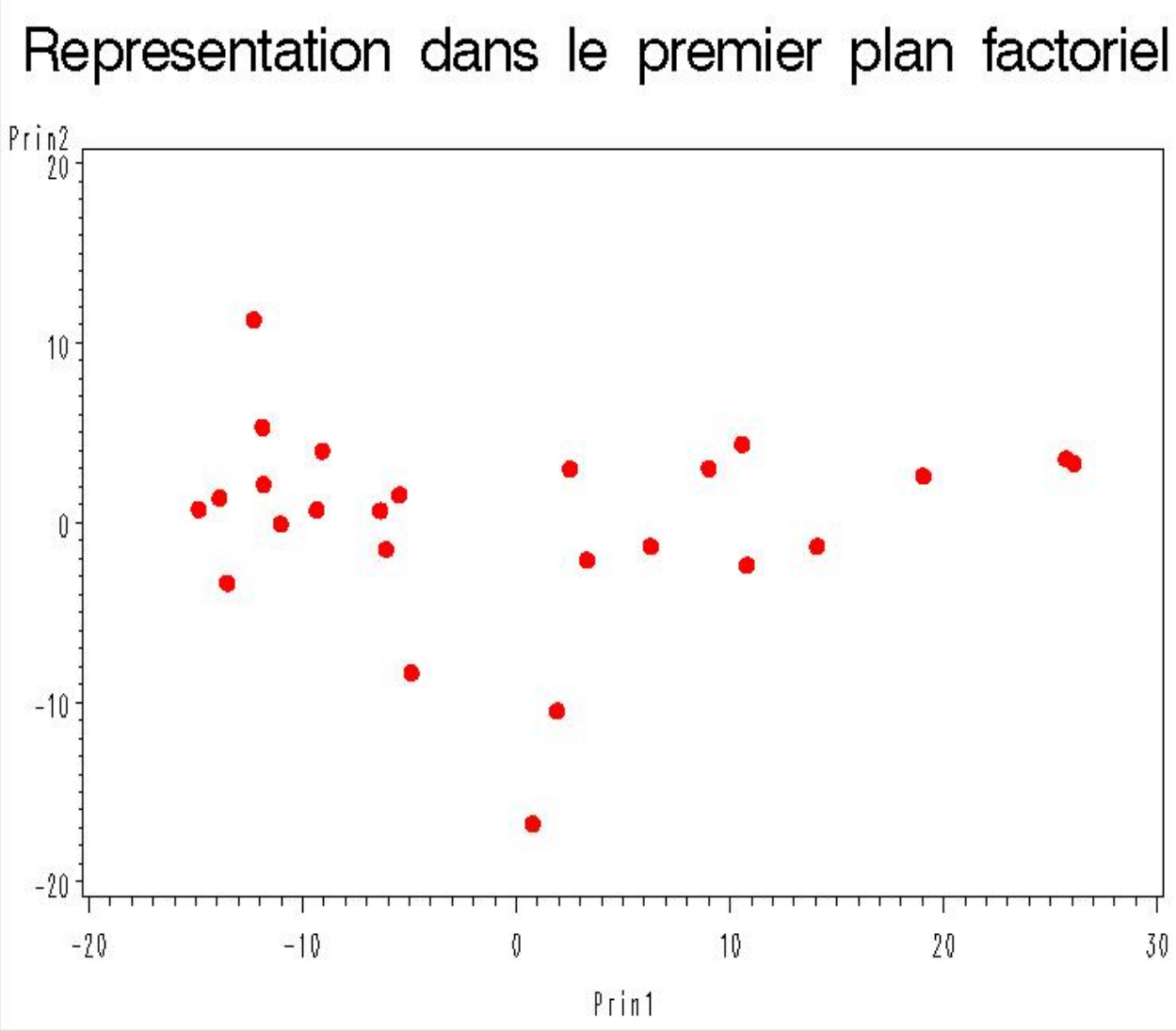


Exemple sous SAS

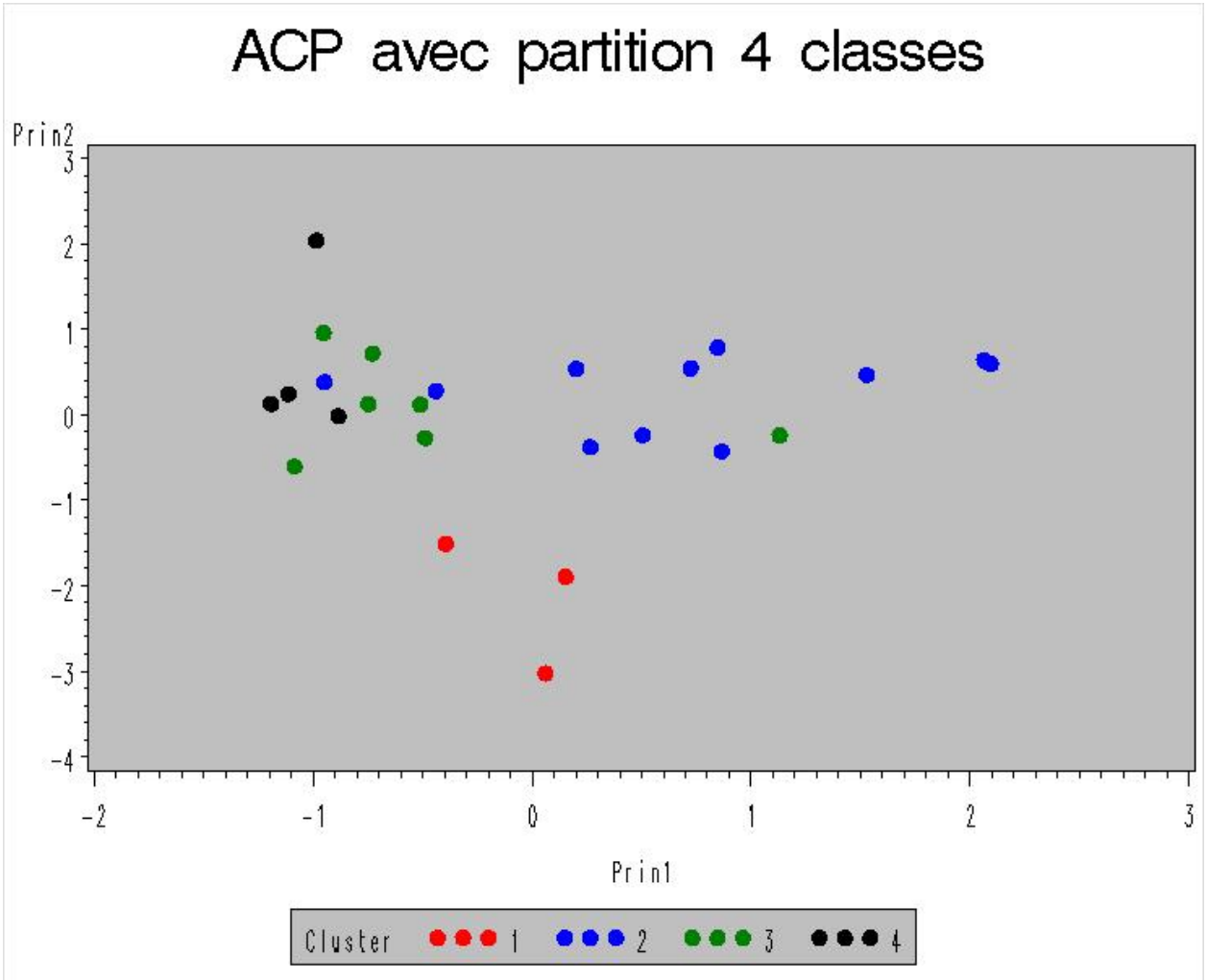
Obs	country	viandr	viandb	oeuf	lait	poisson	cereals	feculent	oleagine	fruitleg
1	Bulgaria	7.8	6.0	1.6	8.3	1.2	56.7	1.1	3.7	4.2
2	Yugoslavia	4.4	5.0	1.2	9.5	0.6	55.9	3.0	5.7	3.2
3	Romania	6.2	6.3	1.5	11.1	1.0	49.6	3.1	5.3	2.8
4	Russian Federat	9.3	4.6	2.1	16.6	3.0	43.6	6.4	3.4	2.9
5	Albania	10.1	1.4	0.5	8.9	0.2	42.3	0.6	5.5	1.7
6	Greece	10.2	3.0	2.8	17.6	5.9	41.7	2.2	7.8	6.5
7	Hungary	5.3	12.4	2.9	9.7	0.3	40.1	4.0	5.4	4.2
8	Italy	9.0	5.1	2.9	13.7	3.4	36.8	2.1	4.3	6.7
9	Poland	6.9	10.2	2.7	19.3	3.0	36.1	5.9	2.0	6.6
10	Czech Republic	9.7	11.4	2.8	12.5	2.0	34.3	5.0	1.1	4.0
11	Spain	7.1	3.4	3.1	8.6	7.0	29.2	5.7	5.9	7.2
12	France	18.0	9.9	3.3	19.5	5.7	28.1	4.8	2.4	6.5
13	Liechtenstein	8.9	14.0	4.3	19.9	2.1	28.0	3.6	1.3	4.3
14	Portugal	6.2	3.7	1.1	4.9	14.2	27.0	5.9	4.7	7.9
15	Belgium	13.5	9.3	4.1	17.5	4.5	26.6	5.7	2.1	4.0
16	Iceland	9.5	4.9	2.7	33.7	5.8	26.3	5.1	1.0	1.4
17	Switzerland	13.1	10.1	3.1	23.8	2.3	25.6	2.8	2.4	4.9
18	Ukraine	8.4	11.6	3.7	11.1	5.4	24.6	6.5	0.8	3.6
19	United Kingdom	17.4	5.7	4.7	20.6	4.3	24.3	4.7	3.4	3.3
20	Ireland	13.9	10.0	4.7	25.8	2.2	24.0	6.2	1.6	2.9
21	Norway	9.4	4.7	2.7	23.3	9.7	23.0	4.6	1.6	2.7
22	Netherlands	9.5	13.6	3.6	23.4	2.5	22.4	4.2	1.8	3.7
23	Denmark	10.6	10.8	3.7	25.0	9.9	21.9	4.8	0.7	2.4
24	Sweden	9.9	7.8	3.5	24.7	7.5	19.5	3.7	1.4	2.0
25	Germany	11.4	12.5	4.1	18.8	3.4	18.6	5.2	1.5	3.8

- Pour chaque pays
- Les consommations des 9 protéines.

Représentation dans le premier plan factoriel



Représentation dans le premier plan factoriel (2)



Comparaison avec la classification mixte

Classification mixte sur les 4 premiers facteurs principaux (85.8% de la variance expliquée).
Comparaison avec la classification sans ACP préalable :

La procédure FREQ

FREQUENCE — Table de CM4 par CLUSTER

CM4(Cluster)	CLUSTER(Cluster) 1	2	3	4	Total
1	1	6	3	0	10
2	2	5	0	0	7
3	0	5	1	0	6
4	0	0	0	2	2
Total	3	16	4	2	25