

# Part 4 - Clustering

C. Maugis-Rabusseau and P. Neuvial

Institut de Mathématiques de Toulouse

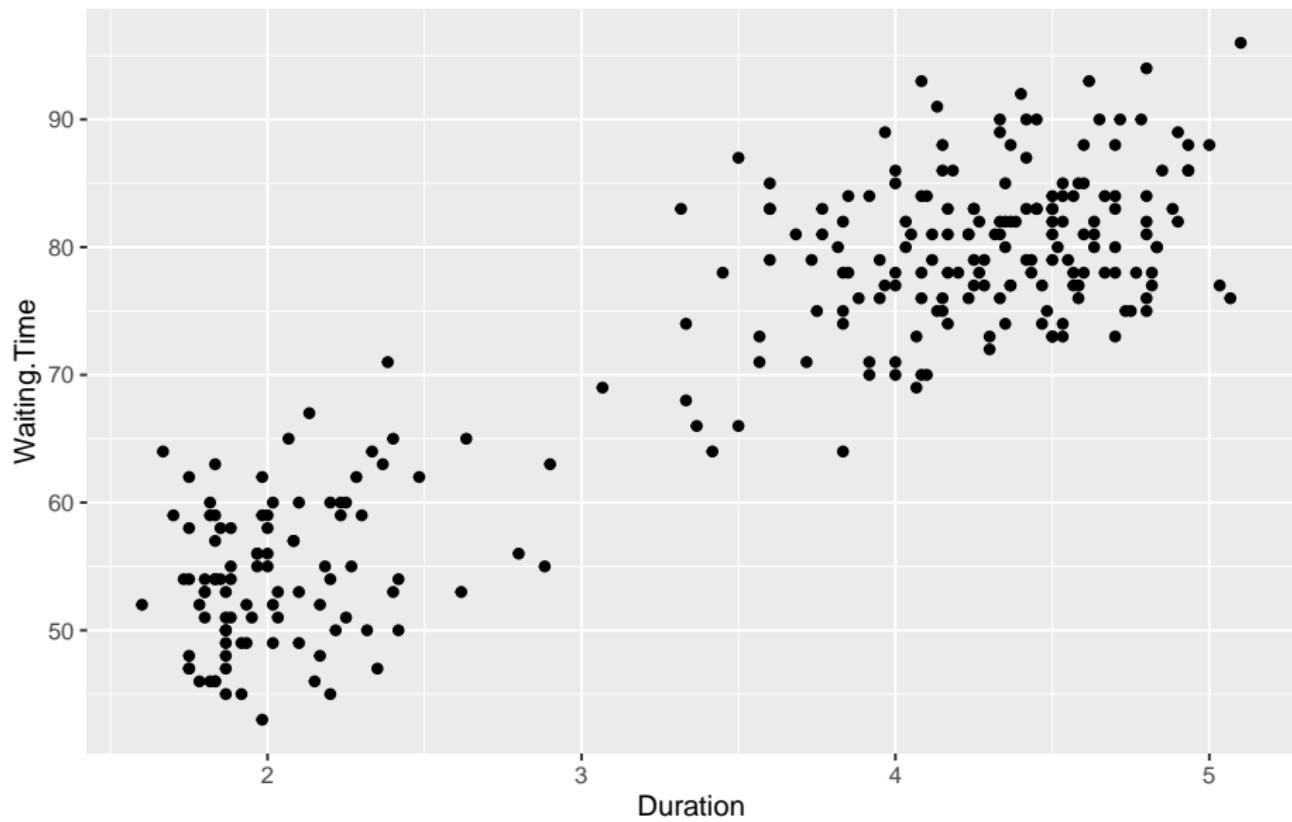
2021-06-29

- 1 Introduction
- 2 Dissimilarités, distances et inerties
- 3 Clustering par partitionnement
- 4 Clustering hiérarchique
- 5 Clustering par modèles de mélanges

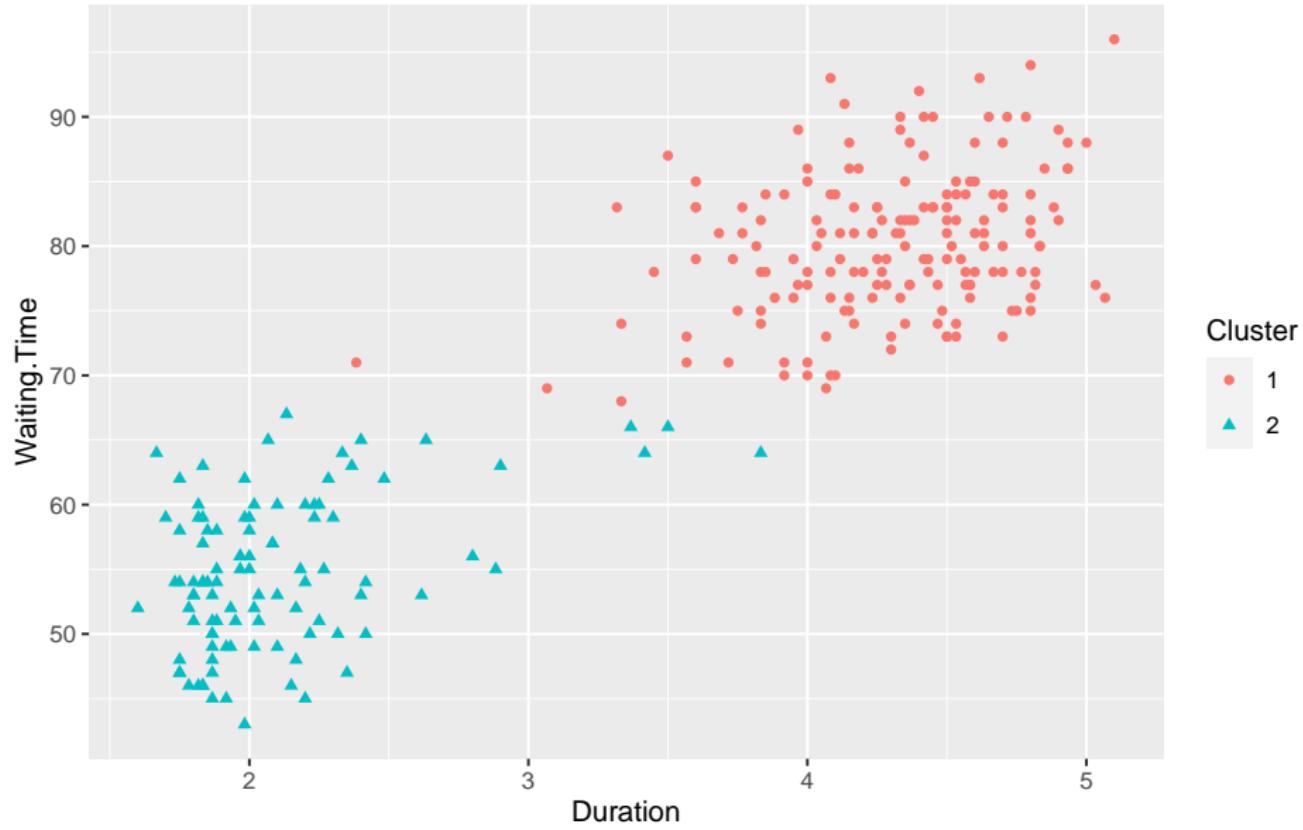
## Section 1

### Introduction

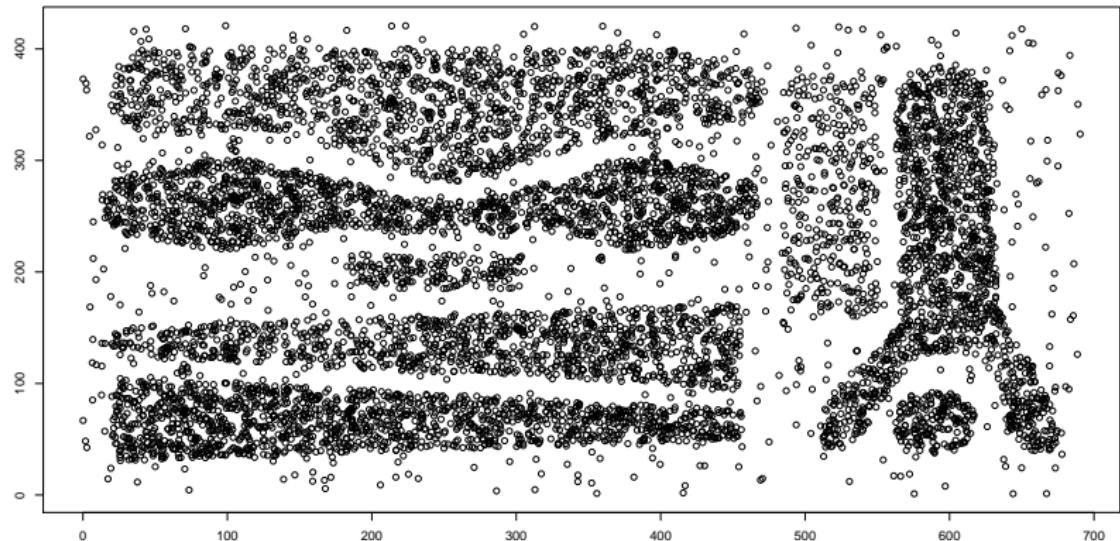
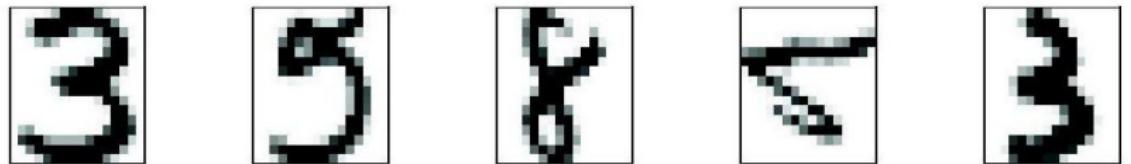
# Exemple des geysers



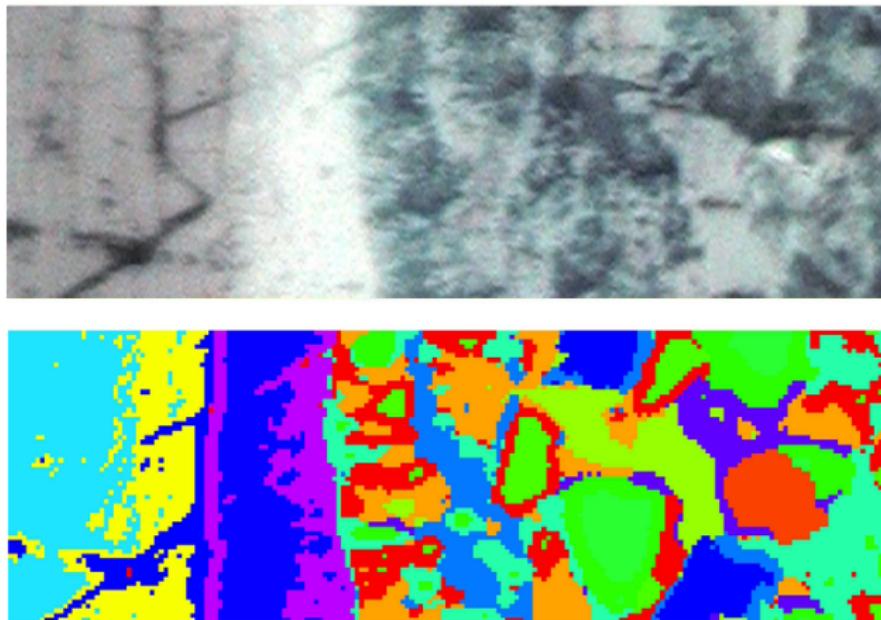
# Exemple des geysers



# Exemple de classification non supervisée de formes

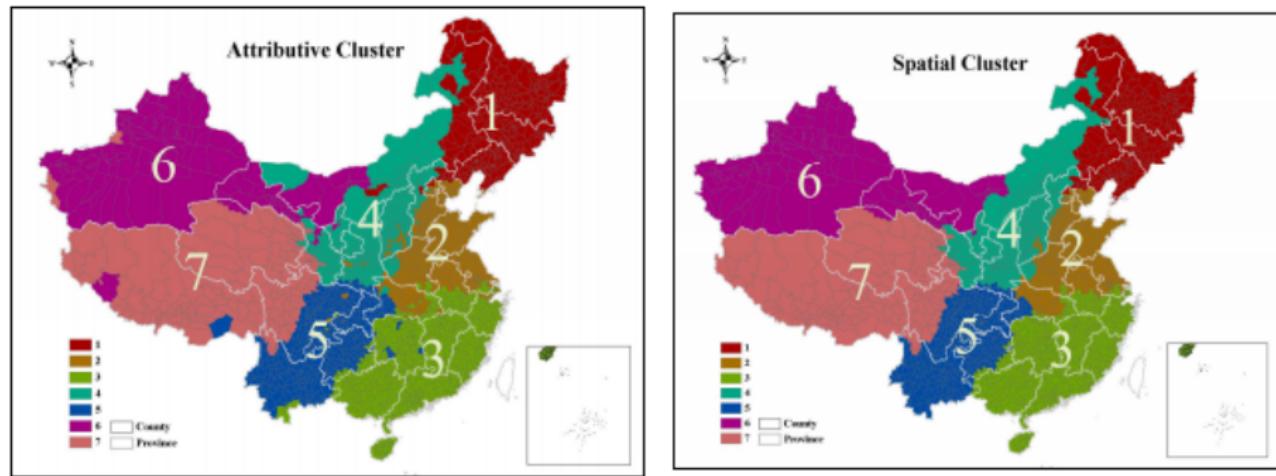


# Exemple spatial



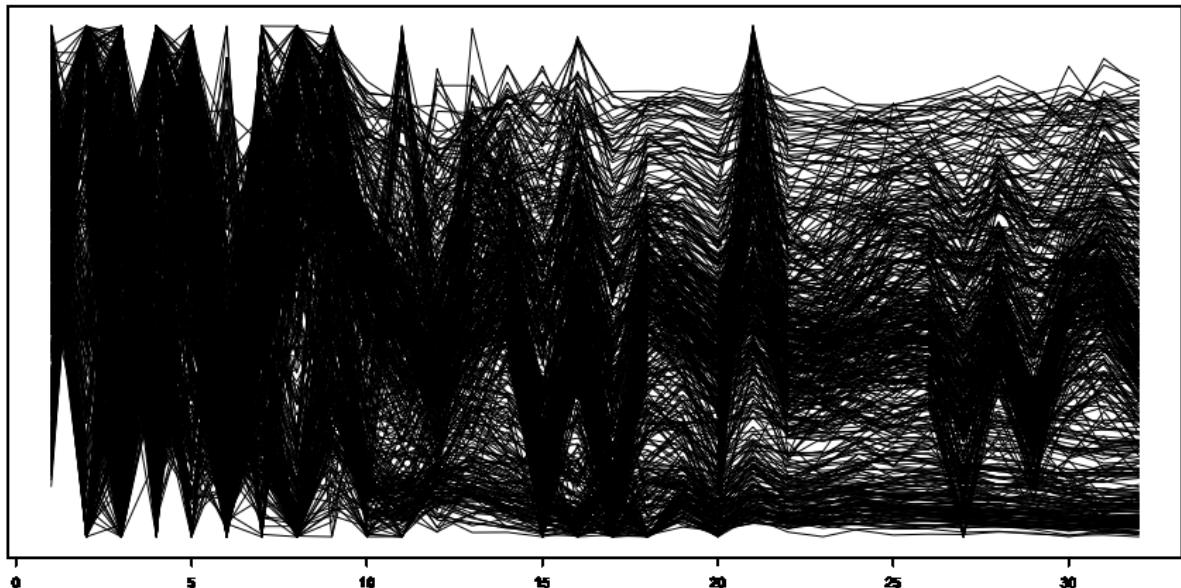
*Le Pennec and Cohen (2011)*

# Exemple spatial

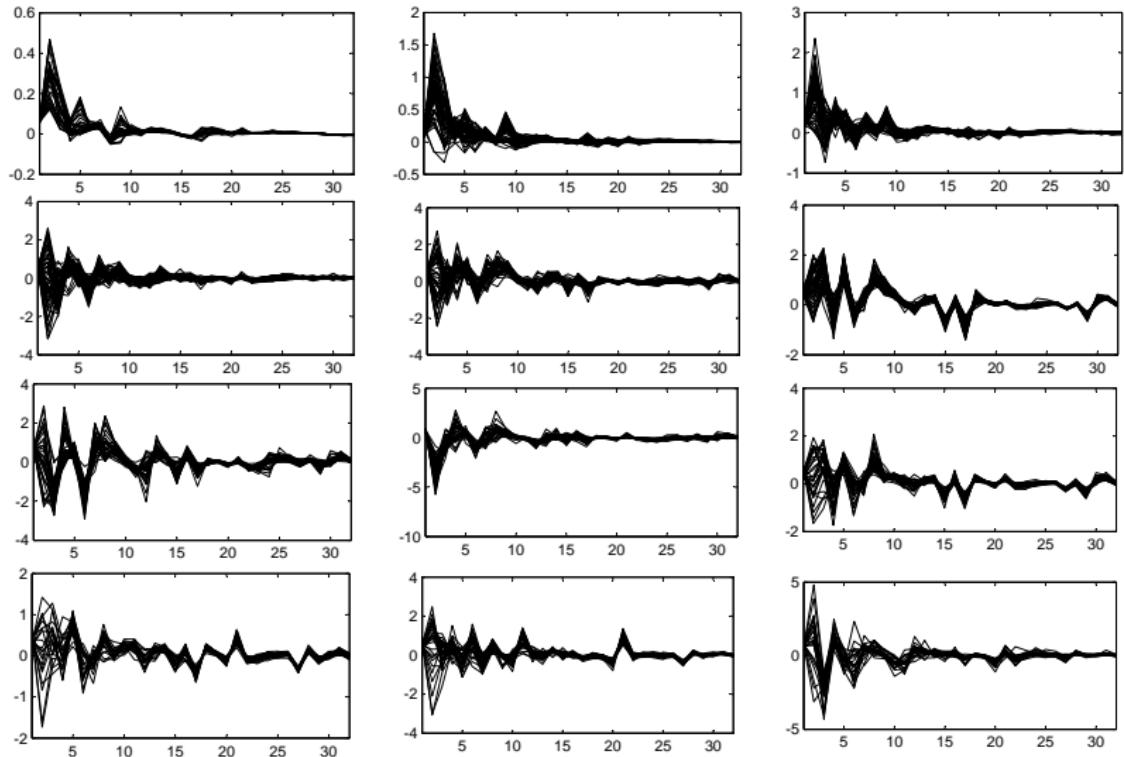


Wang et al. (2010)

# Classification non supervisée de gènes



# Classification non supervisée de gènes



# Objectif

- On observe  $n$  individus décrits par  $p$  variables

$$\underline{\mathbf{x}} = (x_{ij})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq p}} \text{ avec } x_i \in \mathcal{X}$$

- Classification : organisation d'un ensemble d'individus hétérogènes en un ensemble de classes homogènes
- Non supervisée : on ne dispose d'aucune partition a priori des individus et on ne connaît pas le nombre de classes  $K$ .



Déterminer  $K$  classes  $\{\mathcal{C}_1, \dots, \mathcal{C}_K\}$  à partir des  $n$  individus telles qu'une classe est une collection d'individus **similaires** entre eux et **dissimilaires** aux individus des autres classes (classes bien séparées).

- Impossibilité d'une recherche exhaustive

# Catégories de méthodes

- Méthodes par partitionnement (ex: Kmeans)
- Méthodes hiérarchiques (ex: CAH)
- Méthodes basées sur des modèles probabilistes (ex: mélanges)
- Méthodes basées sur des voisinages (densité) (ex: DBSCAN)
- Méthodes basées sur des graphes
- ...

## Section 2

### Dissimilarités, distances et inerties

# Définitions

- **Dissimilité entre individus** :  $d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}^+$  telle que
  - $\forall (x_i, x_\ell) \in \mathcal{X}^2, d(x_i, x_\ell) = d(x_\ell, x_i)$  (symétrie)
  - $d(x_i, x_\ell) = 0 \Leftrightarrow x_i = x_\ell$
- **Similarité (normée) entre individus** :  $s : \mathcal{X} \times \mathcal{X} \rightarrow [0, 1]$  telle que
  - $\forall (x_i, x_\ell) \in \mathcal{X}^2, s(x_i, x_\ell) = s(x_\ell, x_i)$  (symétrie)
  - $s(x_i, x_\ell) = 1 \Leftrightarrow x_i = x_\ell$
- **Distance entre individus** : dissimilité  $d$  + inégalité triangulaire

$$\forall (x_i, x_\ell, x_m) \in \mathcal{X}^3, d(x_i, x_m) \leq d(x_i, x_\ell) + d(x_\ell, x_m)$$

- **Distance ultramétrique** : distance  $d$  + inégalité ultratriangulaire

$$\forall (x_i, x_\ell, x_m) \in \mathcal{X}^3, d(x_i, x_m) \leq \max [d(x_i, x_\ell), d(x_\ell, x_m)]$$

## Subsection 1

### Variables quantitatives

# Dissimilités pour variables quantitatives

- $\forall i \in \{1, \dots, n\}, x_i \in \mathcal{X} = \mathbb{R}^p$
- Distance de **Minkowski** (norme  $L_q$ )

$$d(x_i, x_\ell) = \left( \sum_{j=1}^p |x_{ij} - x_{\ell j}|^q \right)^{\frac{1}{q}}$$

Ex: **Manhattan** ( $q=1$ ), norme **euclidienne** usuelle ( $q=2$ )

- Norme  $L_\infty$

$$d(x_i, x_\ell) = \max_{1 \leq j \leq p} |x_{ij} - x_{\ell j}|$$

$\implies$  invariantes par translation mais sensibles à l'échelle des variables.

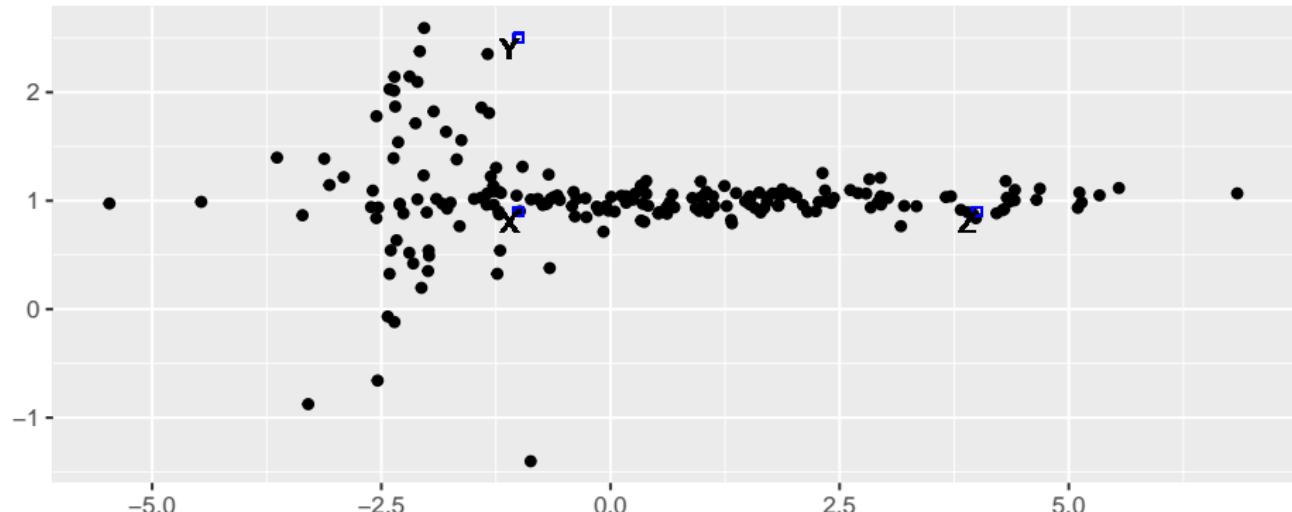
# Dissimilités pour variables quantitatives

- Distances définies comme des formes quadratiques

$$d^2(x_i, x_\ell) = (x_i - x_\ell)' M (x_i - x_\ell)$$

- Norme euclidienne usuelle :  $M = I_p$  ( $d^2(x_i, x_\ell) = \|x_i - x_\ell\|^2$ )
- $M = \text{diag} \left( \frac{1}{\sigma_1^2}, \dots, \frac{1}{\sigma_p^2} \right)$  où  $\sigma_j^2 = \Sigma_{jj} = \text{var}(x^{(j)})$
- $M = \text{diag} \left( \frac{1}{s_1^2}, \dots, \frac{1}{s_p^2} \right)$  où  $s_j$  déviation absolue moyenne
- Distance de Mahalanobis :  $M = \Sigma^{-1}$  où  $\Sigma_{jq} = \text{cov}(x^{(j)}, x^{(q)})$ .

## Exemple pour la distance de Mahalanobis



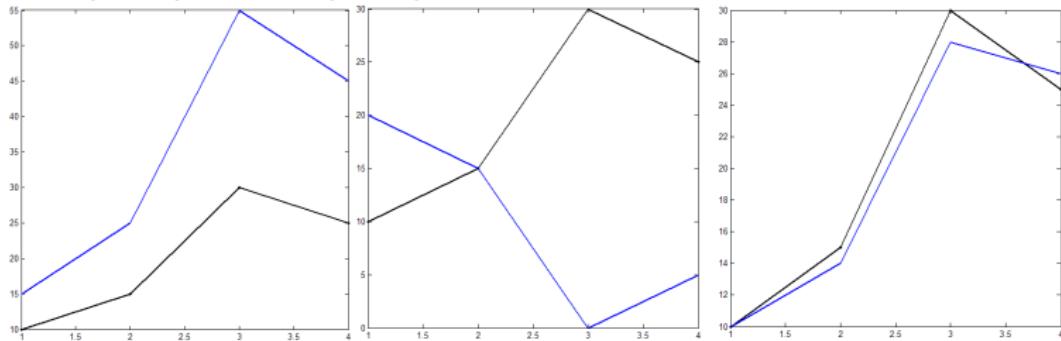
$$\|X - Y\|_2 = 1.6 \quad \|X - Y\|_{2,\Sigma^{-1}} = 4.78$$

$$\|X - Z\|_2 = 5 \quad \|X - Z\|_{2,\Sigma^{-1}} = 2.56$$

# Coefficient de corrélation de Pearson

- Coefficient de corrélation de Pearson  $\rho(x_i, x_\ell)$
- Exemple de dissimilarités

- $d(x_i, x_\ell) = 1 - \rho(x_i, x_\ell)$
- $d(x_i, x_\ell) = 1 - |\rho(x_i, x_\ell)|$
- $d(x_i, x_\ell) = 1 - \rho(x_i, x_\ell)^2$



$$\begin{aligned}d_1(x_g, x_{g'}) &= 1 - \rho(x_g, x_{g'}) = 0 \\d_2(x_g, x_{g'}) &= 1 - |\rho(x_g, x_{g'})| = 0 \\d_3(x_g, x_{g'}) &= 1 - \rho(x_g, x_{g'})^2 = 0 \\\|x_g - x_{g'}\|_2 &= 33.92\end{aligned}$$

$$\begin{aligned}d_1(x_g, x_{g'}) &= 1 - \rho(x_g, x_{g'}) = 2 \\d_2(x_g, x_{g'}) &= 1 - |\rho(x_g, x_{g'})| = 0 \\d_3(x_g, x_{g'}) &= 1 - \rho(x_g, x_{g'})^2 = 0 \\&\|x_g - x_{g'}\|_2 = 37, 42\end{aligned}$$

$$\begin{aligned}d_1(x_g, x_{g'}) &= 1 - \rho(x_g, x_{g'}) = 0.01 \\d_2(x_g, x_{g'}) &= 1 - |\rho(x_g, x_{g'})| = 0.01 \\d_3(x_g, x_{g'}) &= 1 - \rho(x_g, x_{g'})^2 = 0.02 \\&\|x_g - x_{g'}\|_2 = 2.45\end{aligned}$$

# Exemple des Iris

- 3 espèces d'iris : setosa(50), versicolor(50) et virginica(50)
- Mesures en centimètres de longueur du sépale, largeur du sépale, longueur du pétale et largeur du pétale



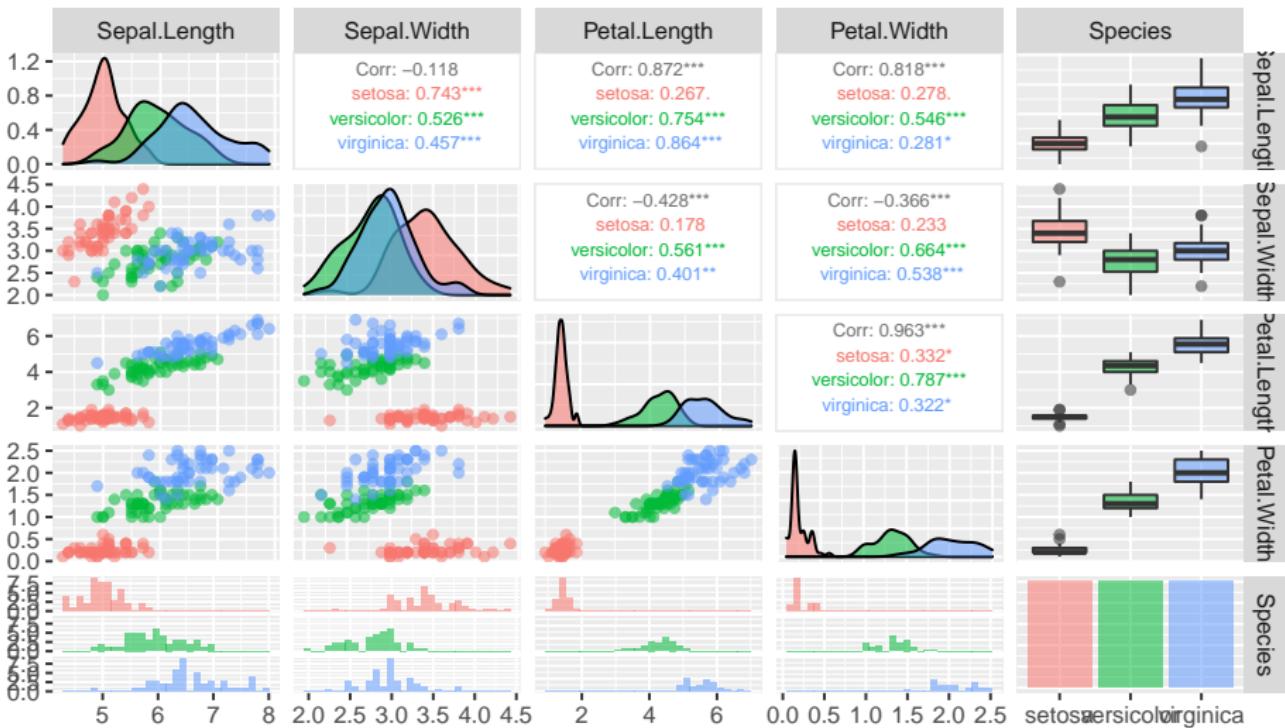
FIG. 2 – *I.setosa*, *I.versicolor*, *I.Virginica*

```
data(iris)  
head(iris)
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa

# Exemple des Iris

```
ggpairs(iris, columns = 1:5, aes(color = Species, alpha = 0.5),  
        upper = list(continuous = wrap("cor", size = 2.5)))
```



# Exemple des Iris

```
irisaux=iris[c(1,2,51,52,101,102),1:4]
round(dist(irisaux,method="manhattan"),2)
```

```
1   2   51   52 101
2   0.7
51  6.7 6.8
52  6.0 6.1 0.9
101 8.3 8.6 3.2 2.7
102 6.9 6.6 2.6 2.1 2.6
```

```
#round(dist(irisaux,method="minkowski",p=2),2)
round(dist(irisaux,method="euclidian"),2)
```

```
1   2   51   52 101
2   0.54
51  4.00 4.10
52  3.62 3.69 0.64
101 5.28 5.34 1.84 1.81
102 4.21 4.18 1.45 1.06 1.33
```

```
round(1-cor(t(irisaux)),2)
```

```
1   2   51   52 101 102
1  0.00 0.00 0.21 0.21 0.49 0.43
2  0.00 0.00 0.17 0.17 0.43 0.37
51 0.21 0.17 0.00 0.00 0.07 0.04
52 0.21 0.17 0.00 0.00 0.07 0.04
101 0.49 0.43 0.07 0.07 0.00 0.00
102 0.43 0.37 0.04 0.04 0.00 0.00
```

## Subsection 2

### Variables qualitatives

# Dissimilités pour variables binaires

- Table de contingence entre 2 individus  $x_i$  et  $x_\ell$ :

	1	0
1	$m_{11}$	$m_{10}$
0	$m_{01}$	$m_{00}$

- variable binaire **symétrique** = pas d'influence sur le choix du codage 0 – 1 (ex: sexe d'une personne)

- Appariement simple :  $s(x_i, x_\ell) = \frac{m_{11} + m_{00}}{m_{11} + m_{00} + m_{10} + m_{01}}$
- Rogers et Tanimoto :  $s(x_i, x_\ell) = \frac{m_{11} + m_{00}}{m_{11} + m_{00} + 2(m_{10} + m_{01})}$
- Sokal et Sneath :  $s(x_i, x_\ell) = \frac{2(m_{11} + m_{00})}{2(m_{11} + m_{00}) + m_{10} + m_{01}}$

- variable binaire **asymétrique** = les valeurs 0-1 n'ont pas la même importance (ex: présence ou non du groupe sanguin AB<sup>-</sup>)

- Jaccard :  $s(x_i, x_\ell) = \frac{m_{11}}{m_{11} + m_{10} + m_{01}}$
- Dice :  $s(x_i, x_\ell) = \frac{2m_{11}}{2m_{11} + m_{10} + m_{01}}$

# Exemple de Kaufman et Rousseeuw (90)

Personne	Sexe (Homme=1,Femme=0)	Marié (Oui=1,Non=0)	Cheveux clairs (1) ou bruns (0)	yeux bleus (1) ou marron (0)	Lunettes (Oui=1,Non=0)	V'sage rond (1) ou ovale (0)	Pessimiste (1) ou optimiste (0)	Du matin (0) ou du soir (1)	Enfant unique (Oui=1,Non=0)	Gaucher (1) ou droitier (0)
Ilan	1	0	1	1	0	0	1	0	0	0
Peter	1	1	0	0	1	0	1	1	0	0
Jacqueline	0	1	0	0	1	0	0	0	0	0
Lieve	0	1	0	0	0	0	0	1	1	0
Leon	1	1	0	0	1	1	0	1	1	0
Kim	0	0	1	0	0	0	1	0	0	1
Talia	0	0	0	1	0	1	0	0	0	0
Tina	0	0	0	1	0	1	0	0	0	0

Extrait de Kaufman et Rousseeuw (90), p24

		$s(Lieve, Jacqueline)$	$s(Ilan, Peter)$
	Appariement simple	$7/10 = 0.7$	$> 5/10 = 0.5$
	Rogers et Tanimoto	$7/13 = 0.53$	$> 5/15 = 0.33$
	Sokal et Sneath	$14/17 = 0.82$	$> 10/15 = 0.667$
	Jaccard	$1/4 = 0.25$	$< 2/7 = 0.29$
	Dice	$2/5 = 0.4$	$< 4/9 = 0.44$

## (Dis)similarité pour variables nominales

- Variables avec  $M > 2$  modalités (ex: couleur des yeux, statut marital)
- Coefficient d'**appariement simple** :  $s(x_i, x_\ell) = \frac{u}{p}$  où  $u$  = nombre de variables où  $x_i$  et  $x_\ell$  ont la même modalité
- Transformer la variable nominale en variables binaires (une par modalité) + distance du  $\chi^2$

$$\underline{x} = \begin{pmatrix} 2 & 2 & 1 \\ 1 & 3 & 4 \\ 2 & 1 & 1 \\ \vdots & \vdots & \vdots \end{pmatrix} \Rightarrow Z = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ \vdots & \vdots \end{pmatrix}$$

Distance du  $\chi^2$  entre individus :

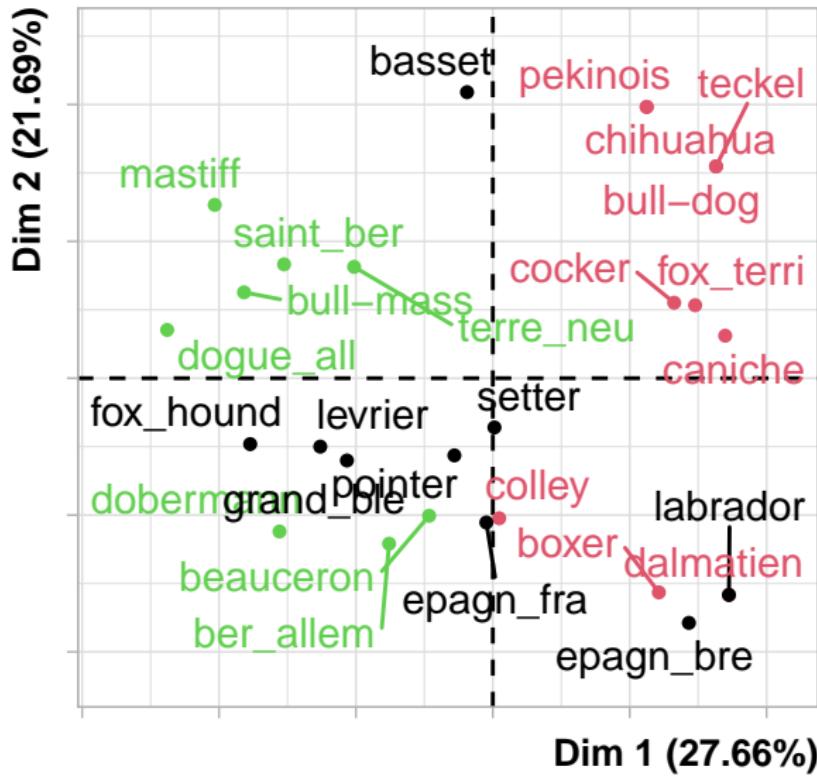
$$d^2(x_i, x_\ell) = \frac{n}{p} \sum_{j=1}^{\tilde{p}} \frac{(Z_{ij} - Z_{\ell j})^2}{Z_{\cdot j}} \text{ avec } Z_{\cdot j} = \frac{1}{n} \sum_{i=1}^n Z_{ij}$$

# Exemple des chiens

- Données : 27 races de chiens décrites par 6 variables
  - taille : petite (1), moyenne (2), grande (3)
  - poids : petite (1), moyenne (2), grande (3)
  - vélocité : petite (1), moyenne (2), grande (3)
  - intelligence : petite (1), moyenne (2), grande (3)
  - affectation : faible (1), forte (2)
  - agressivité : faible (1), forte (2)
- 1 autre variable “fonction” : compagnie (1), chasse (2), utilité (3)

# Exemple des chiens

MCA factor map



# Exemple des chiens

Avec l'appariement simple :

	beauceron	caniche	epagn_bre
beauceron	1.00	0.17	0.33
caniche	0.17	1.00	0.67
epagn_bre	0.33	0.67	1.00

Avec la distance du Khi-deux :

	beauceron	caniche	epagn_bre
beauceron	0.00	0.51	0.45
caniche	0.51	0.00	0.25
epagn_bre	0.45	0.25	0.00

Avec la distance euclidienne usuelle sur les coordonnées de l'ACM

	beauceron	caniche
caniche	1.702559	
epagn_bre	1.381553	1.227103

## Subsection 3

### Variables mixtes

# Stratégies pour les variables mixtes

- 1ère stratégie : tout transformer en variables de même nature
- 2ème stratégie : **métrique de Gower**

$$d(x_i, x_\ell) = \sum_{j=1}^p \delta_{i\ell}^{(j)} d_{i\ell}^{(j)} / \sum_{j=1}^p \delta_{i\ell}^{(j)}$$

avec

$$\delta_{i\ell}^{(j)} = \begin{cases} 0 & \text{si } \begin{cases} x_{ij} \text{ ou } x_{\ell j} \text{ est manquante} \\ x_{ij} = x_{\ell j} = 0 \text{ et } j \text{ variable binaire asymétrique} \end{cases} \\ 1 & \text{sinon.} \end{cases}$$

et

$$d_{i\ell}^{(j)} = \begin{cases} \mathbb{1}_{x_{ij} \neq x_{\ell j}} & \text{si } j \text{ variable binaire ou nominale} \\ \frac{|x_{ij} - x_{\ell j}|}{\max_{1 \leq h \leq n} x_{hj} - \min_{1 \leq h \leq n} x_{hj}} & \text{si } j \text{ est quantitative} \end{cases}$$

## Subsection 4

**En pratique sous R**

# Quelques commandes de R

- `dist()` : `method= "euclidian", "manhattan", "minkowski", "maximum", "canberra", "binary"`
- `daisy()` [`library(cluster)`] : `metric= "euclidian", "manhattan", "gower"`
- `dist.binary()` [`library(ade4)`] : 10 méthodes sont implémentées dont Jaccard, Simple matching coefficient, Sokal&Sneath, Rogers&Tanimoto, Dice, ...
- `dist.quant()` [`library(ade4)`] : 3 méthodes de la forme  $\|x - y\|_A$  : canonical ( $A=I$ ), Joreskog ( $A=1/\text{diag}(\text{cov})$ ), Mahalanobis ( $A=\text{inv}(\text{cov})$ )
- `dist.ktab()` [`library(ade4)`] : pour les variables mixtes (coefficient de Gower)
- ...

## Subsection 5

### Inertie

# Définitions

- Soit  $d$  une **distance euclidienne** entre individus. Soit  $\mathcal{P} = \{\mathcal{C}_1, \dots, \mathcal{C}_K\}$  une partition des individus en  $K$  classes.
- **Inertie totale** :  $I_T = \frac{1}{n} \sum_{i=1}^n d(x_i, c)^2$   
où  $c = \frac{1}{n} \sum_{i=1}^n x_i$  est le centre de gravité du nuage de points
- **Inertie interclasse** :  $I_{inter} = \frac{1}{n} \sum_{k=1}^K |\mathcal{C}_k| \times d(m_k, c)^2$   
où  $m_k = \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} x_i$  est le centre de gravité de la classe  $\mathcal{C}_k$

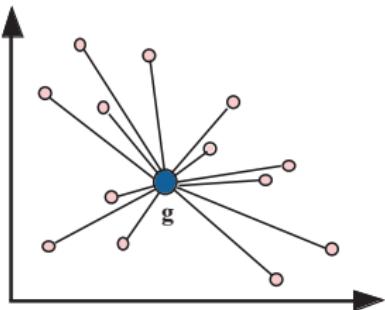
⇒ variance des centres des classes

- **Inertie intra-classe** :  $I_{intra} = \frac{1}{n} \sum_{k=1}^K \sum_{i \in \mathcal{C}_k} d(x_i, m_k)^2$

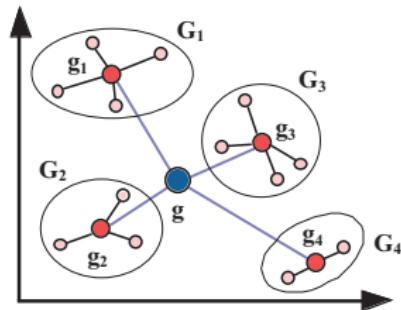
⇒ variance des points d'une même classe

# Propriété de Huygens

$$I_T = I_{inter} + I_{intra}$$



Bisson (2001)

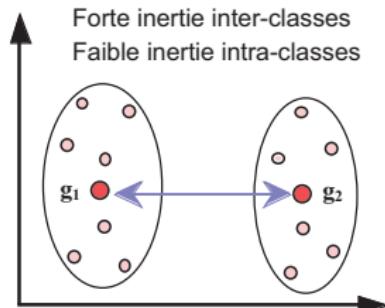


# Objectif

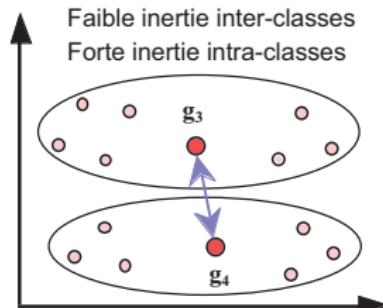
on veut minimiser l'inertie intra-classe



maximiser l'inertie inter-classe



Bisson (2001)



## Section 3

### Clustering par partitionnement

## Subsection 1

### Méthodes de type K-means

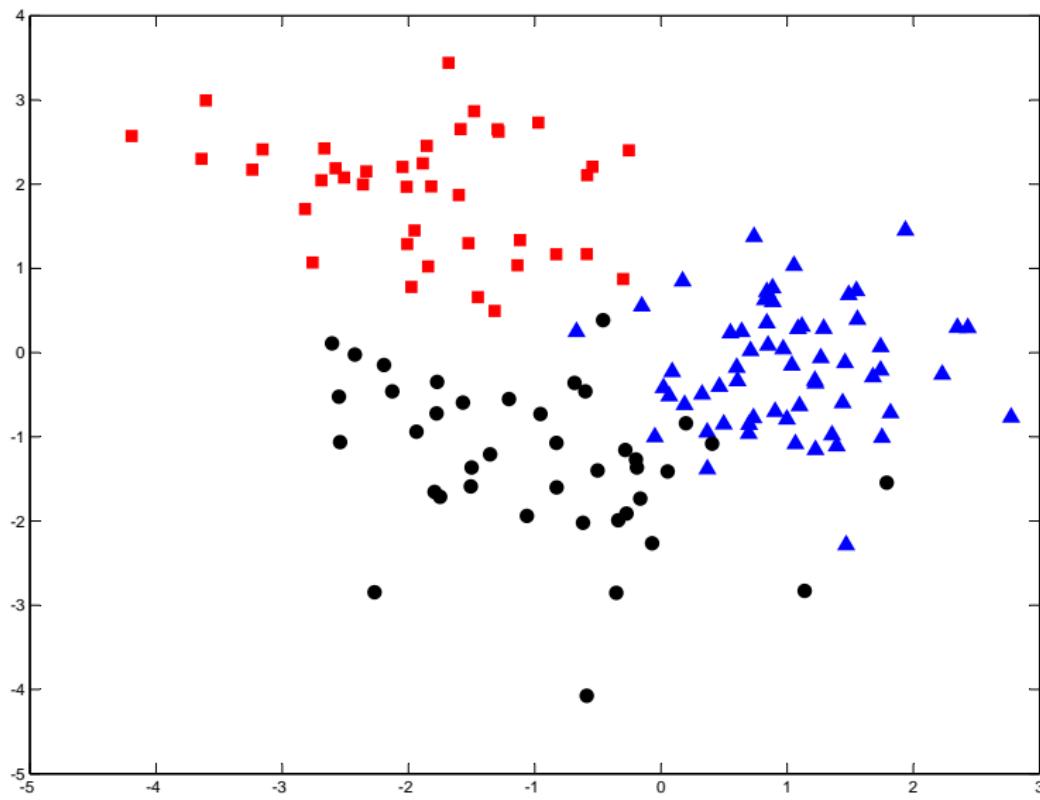
# Méthode des centres mobiles (Forgy, 65)

- On considère des **variables quantitatives** et  $d$  une distance (euclidienne)
- But : trouver une partition de l'ensemble des individus telle que l'inertie intra-classe soit minimale.
- Principe général :
  - Initialisation :
    - choix du nombre de classes  $K$
    - choix de  $K$  noyaux initiaux  $c_1^{(0)}, \dots, c_K^{(0)}$
  - On itère les deux étapes suivantes : à l'itération  $t$
  - Affectation des individus à la classe la plus proche :

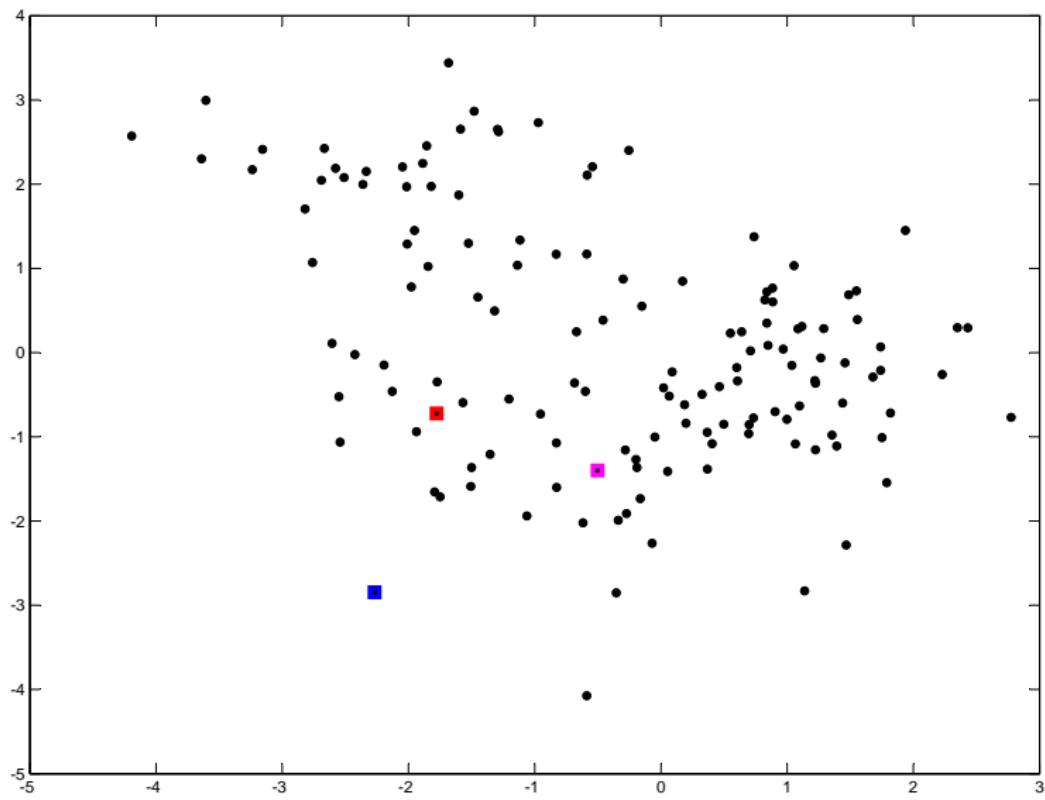
$$i \in \mathcal{C}_k^{(t)} \text{ si } d\left(x_i, c_k^{(t-1)}\right) = \min_{1 \leq k' \leq K} d\left(x_i, c_{k'}^{(t-1)}\right)$$

- Mise à jour des noyaux (centres de gravité) :  $c_1^{(t)}, \dots, c_K^{(t)}$
- Arrêt de l'algorithme quand la classification n'est plus modifiée

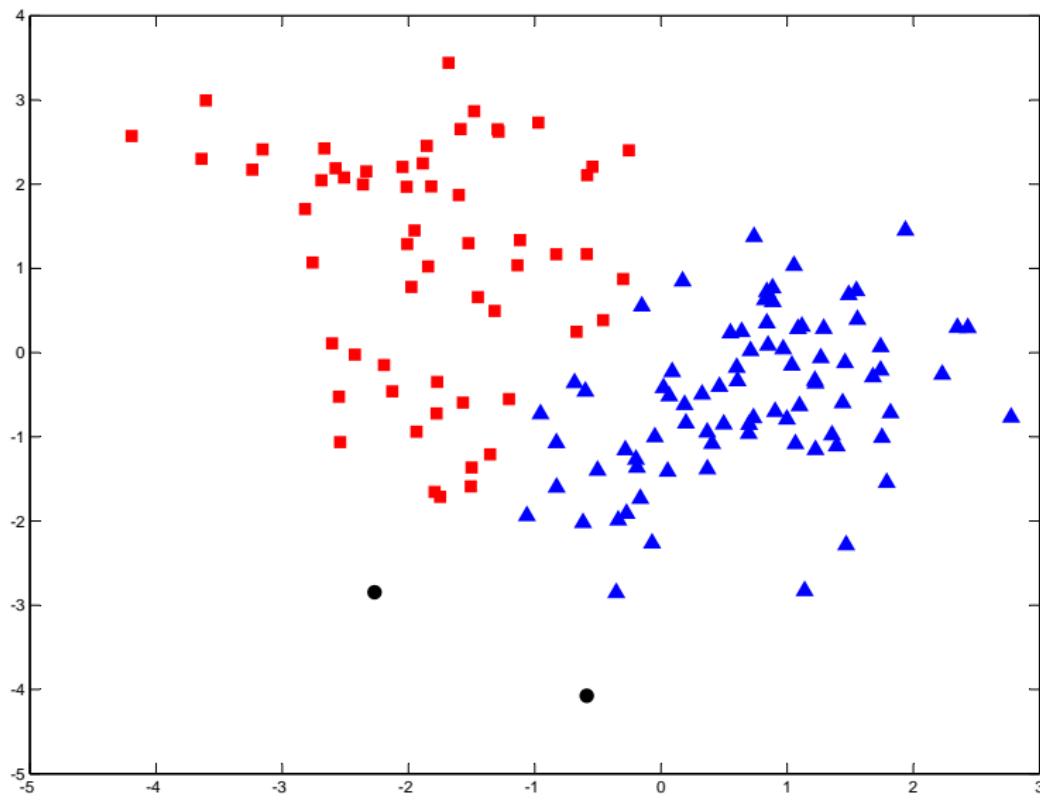
# Exemple



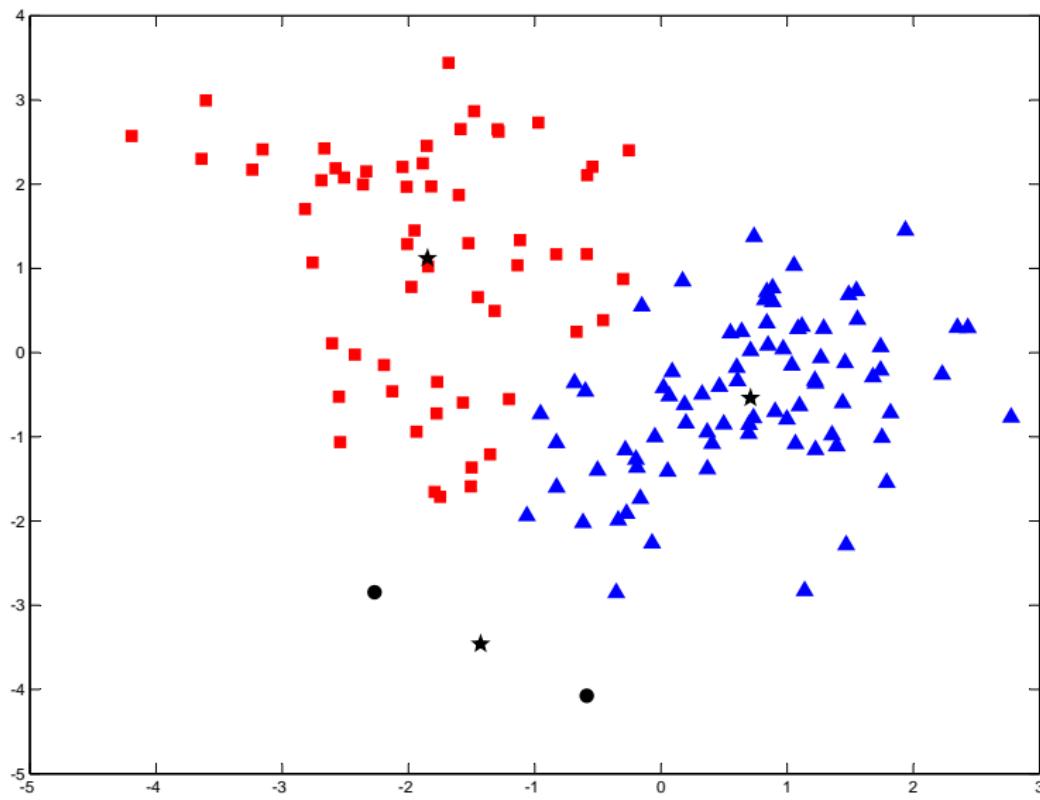
# Exemple



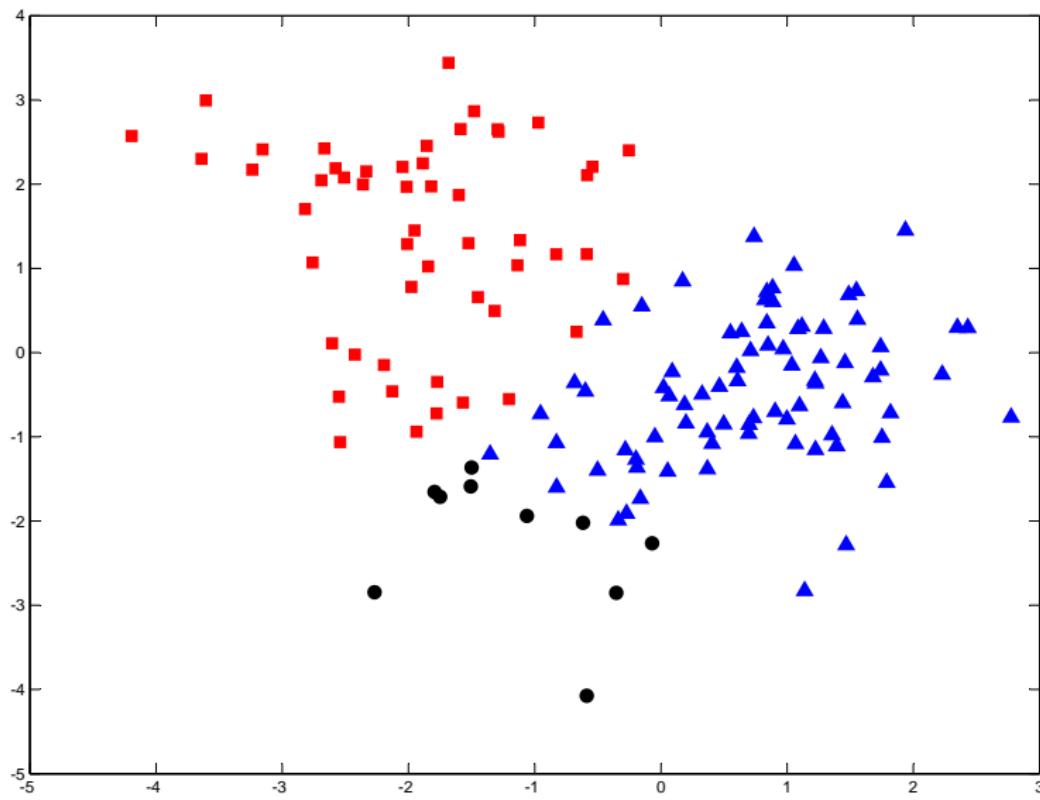
# Exemple



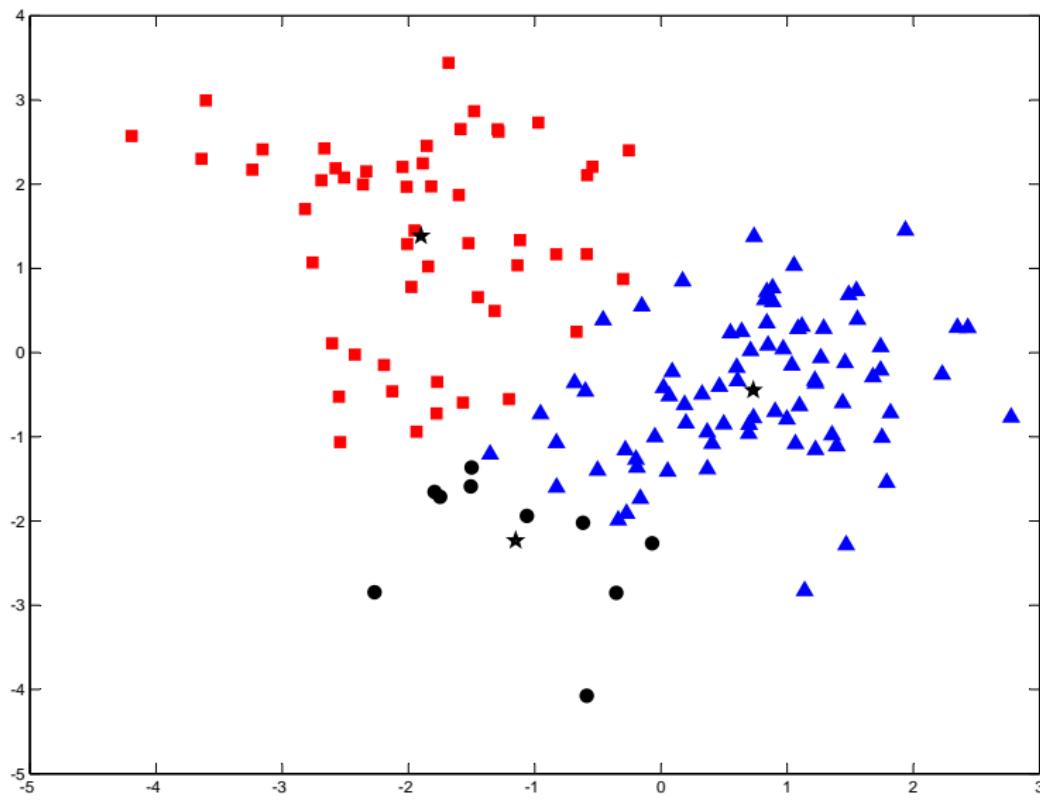
# Exemple



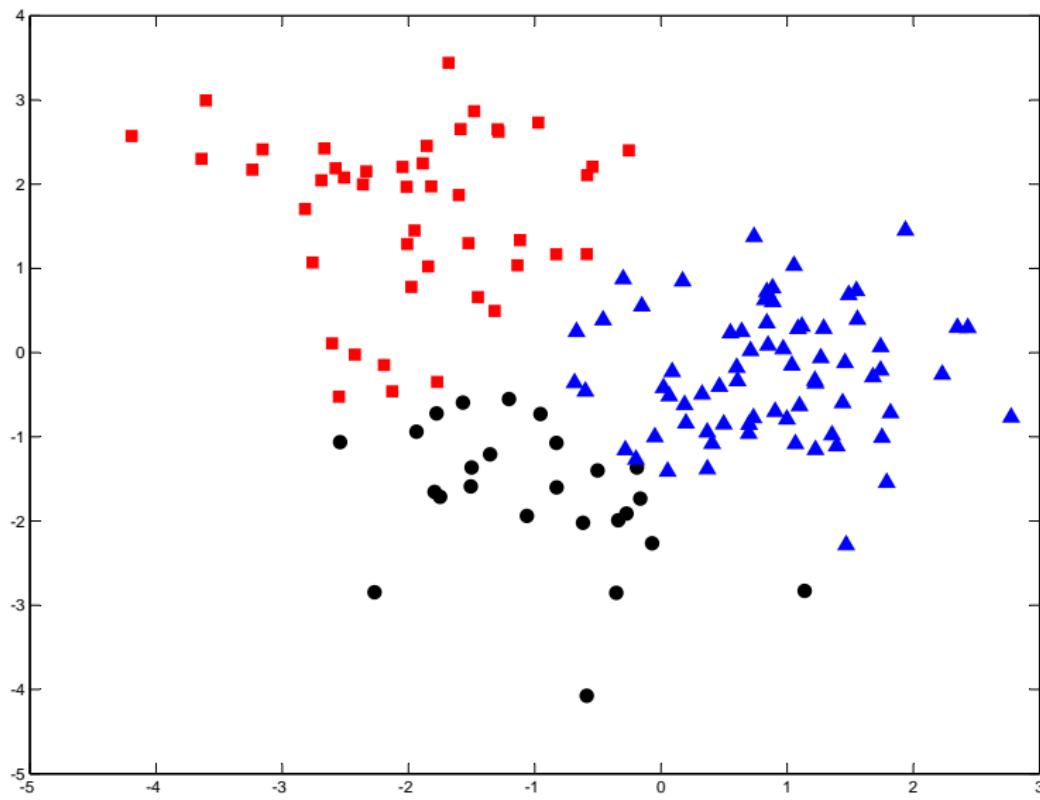
# Exemple



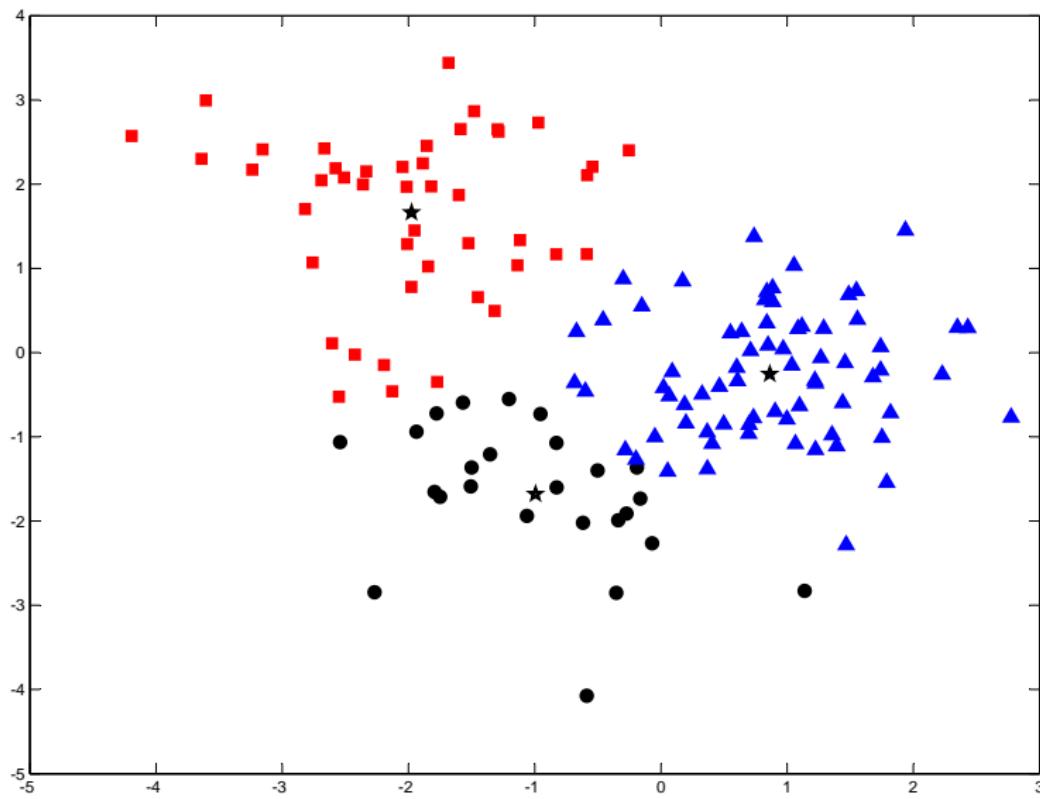
# Exemple



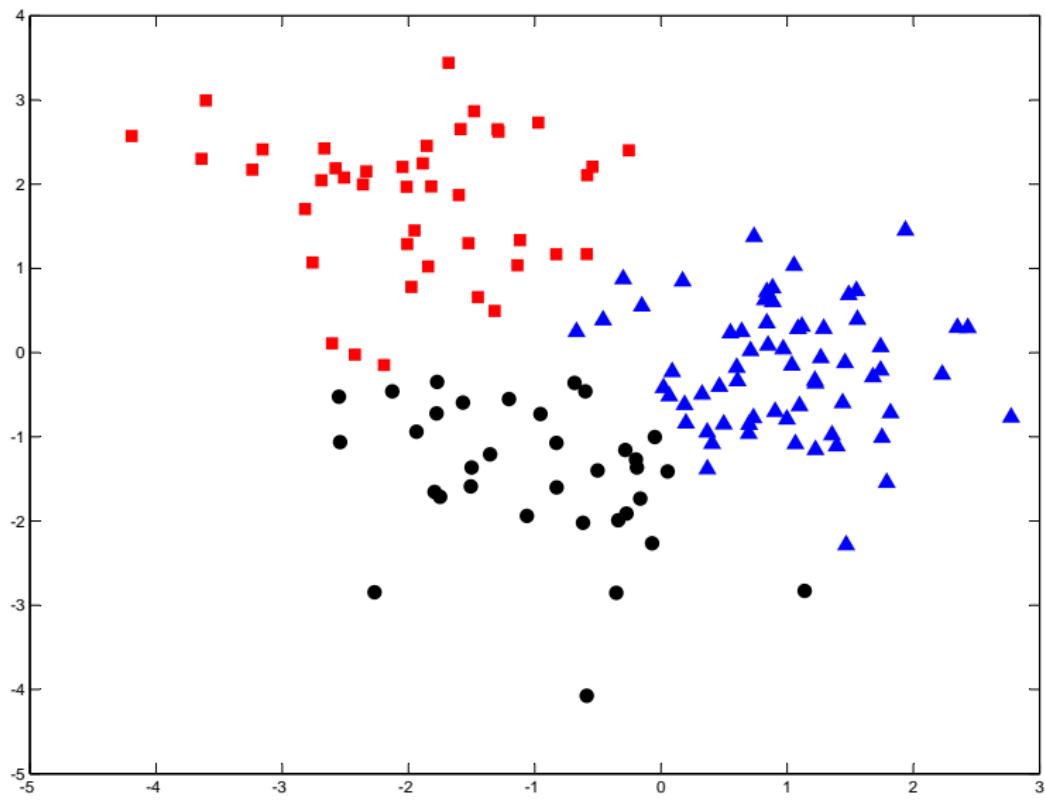
# Exemple



# Exemple



# Exemple



# Plus / Moins

- Avantages :
  - Relativement efficace (rapide)
  - Tend à réduire l'inertie intra-classe à chaque itération
  - Forme des classes compactes, bien séparées
- Inconvénients :
  - Spécification du nombre de classes  $K$
  - Influence du choix des noyaux initiaux
  - Convergence vers un minimum local
  - Non détermination de classes non convexes
  - Non applicable à des variables qui ne sont pas numériques
  - Peut produire des classes vides
  - Influence des outliers

- **K-means** (MacQueen, 67) : A chaque itération,
  - on tire un point  $x_i$  au hasard
  - on détermine le noyau le plus proche
  - on remet à jour le noyau
- **Nuées dynamiques** (Diday, 71) : les noyaux sont des sous-ensembles de points de cardinal  $q$  fixé

# Quelques commandes sous R

- `km=kmeans(x, centers=, iter.max=, algorithm=)`
  - *centers*: vecteur indiquant soit les noyaux initiaux, soit le nombre de classes choisies
  - *iter.max*: nombre d'itérations maximale (jusqu'à convergence par défaut)
  - *algorithm*: "Forgy" (centres mobiles), "MacQueen" ( $K$ -means), nuées dynamiques par défaut
- `names(km) [1] "cluster" "centers" "withinss" "size"`
  - *cluster*: vecteur des labels
  - *centers*: matrice des centres de gravité des classes
  - *withinss*: vecteur des sommes des carrés des écarts intra-classes pour chaque classe
  - *size*: vecteur des effectifs par classe.
- `clusplot()` [`library(cluster)`]: 2-dimensional clustering plot
- `fviz_cluster()` [`library(factoextra)`]: ggplot visualisation de clustering

# Exemple des iris

```
data(iris)
kmiris=kmeans(iris[,1:4],centers=3)
table(iris$Species,kmiris$cluster)
```

	1	2	3
setosa	50	0	0
versicolor	0	48	2
virginica	0	14	36

```
classError(kmiris$cluster,iris$Species)$errorRate
```

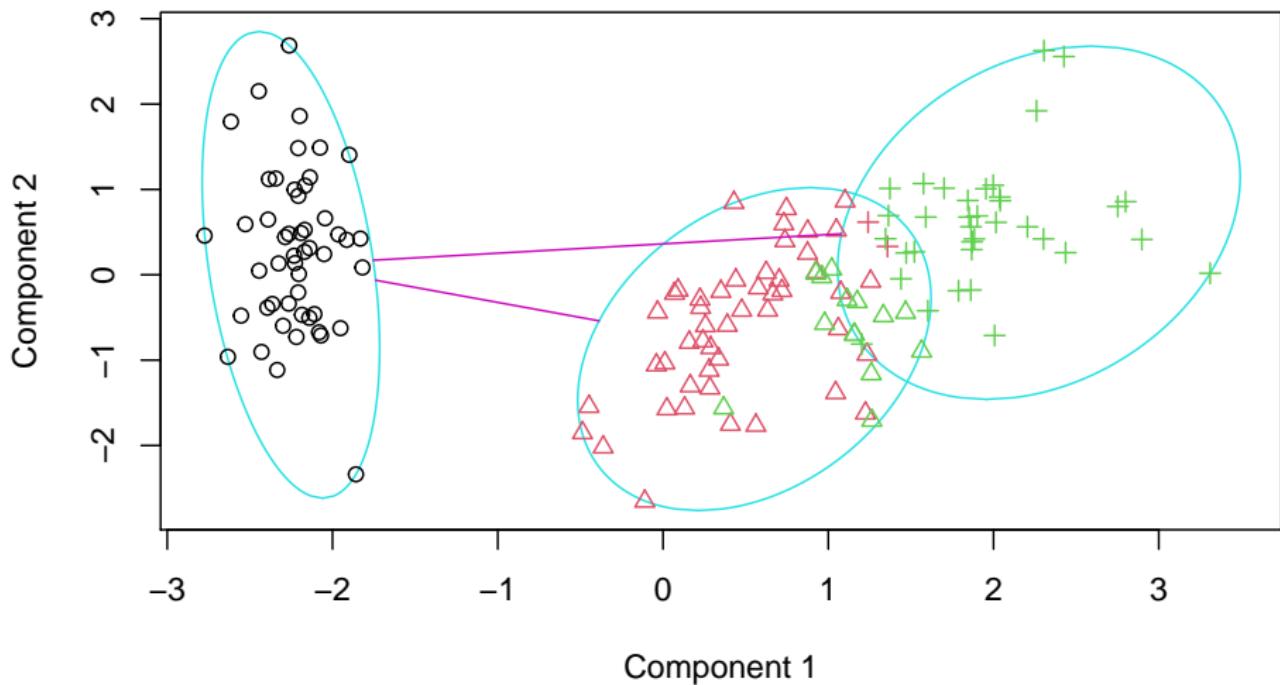
```
[1] 0.1066667
```

```
adjustedRandIndex(iris$Species,kmiris$cluster)
```

```
[1] 0.7302383
```

# Exemple des iris

```
clusplot(iris[,1:4], kmiris$cluster, col.p=iris$Species, plotchar=T, main="")
```

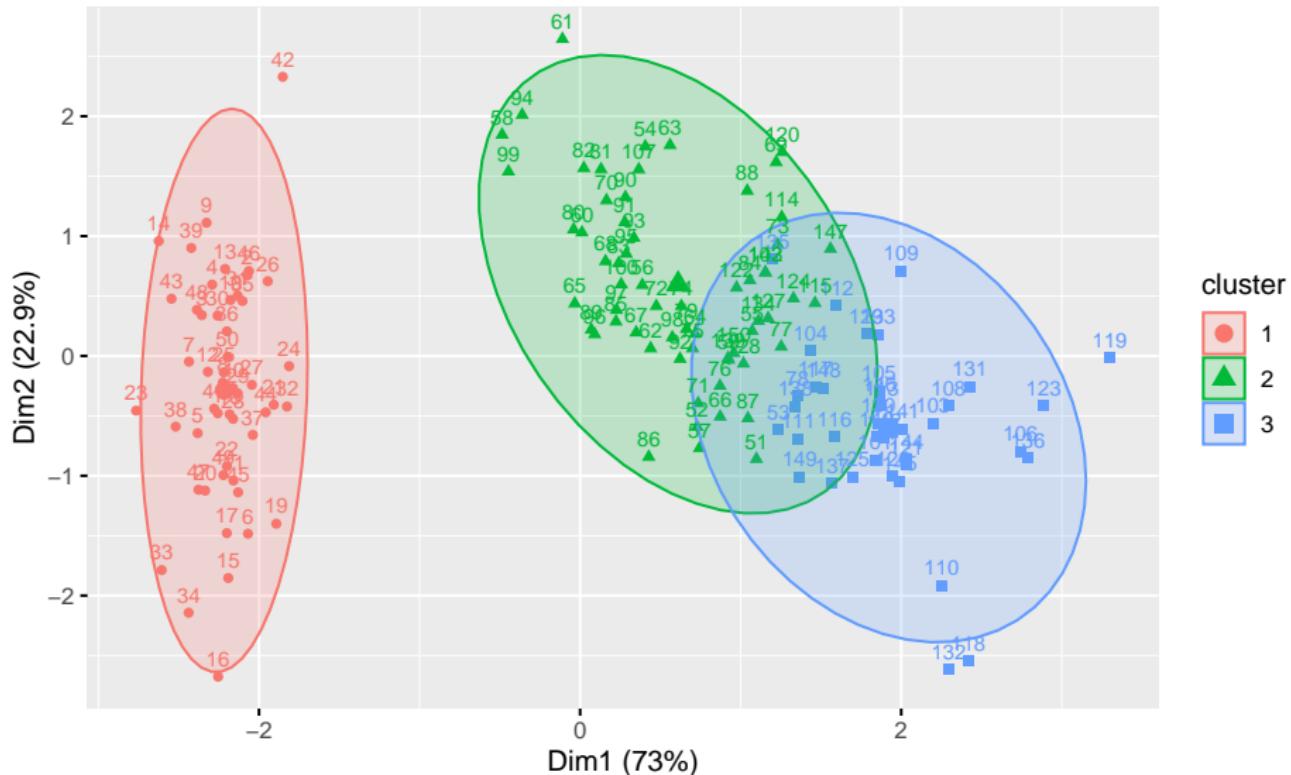


Component 1

These two components explain 95.81 % of the point variability.

# Exemple des iris

```
fviz_cluster(kmiris,data=iris[,1:4],ellipse.type="norm",labelsize=8)+  
  ggtitle("")
```



# Extensions

- Pour variables **qualitatives** :

- Dans les méthodes de type *K*-means, chaque classe est représentée par son centre de gravité, non adapté pour des variables qualitatives
- Algorithme ***K*-modes** (Huang, 98) : modif. de la dissimilarité et les modes remplacent les centres de gravité

- Pour variables **mixtes** :

- Algorithme ***K*-prototype** (Huang, 97) : si chaque individu  $i$  est décrit par deux types de variables ( $x_i = (x_i^{(1)}, x_i^{(2)})$ ) alors

$$d(x_i, x_\ell) = d_1(x_i^{(1)}, x_\ell^{(1)}) + \gamma d_2(x_i^{(2)}, x_\ell^{(2)})$$

où  $d_1$  (resp.  $d_2$ ) est la dissimilarité sur les variables quantitatives (resp. qualitatives) et  $\gamma$  un paramètre à fixer pour éviter de favoriser un type de variables

# Exemple des chiens

```
library(klaR)
clkmodes<-kmodes(chiens[,-7],3,iter.max=100,weight=FALSE)
adjustedRandIndex(chiens[,7],clkmodes$cluster)
```

```
[1] 0.3314785
```

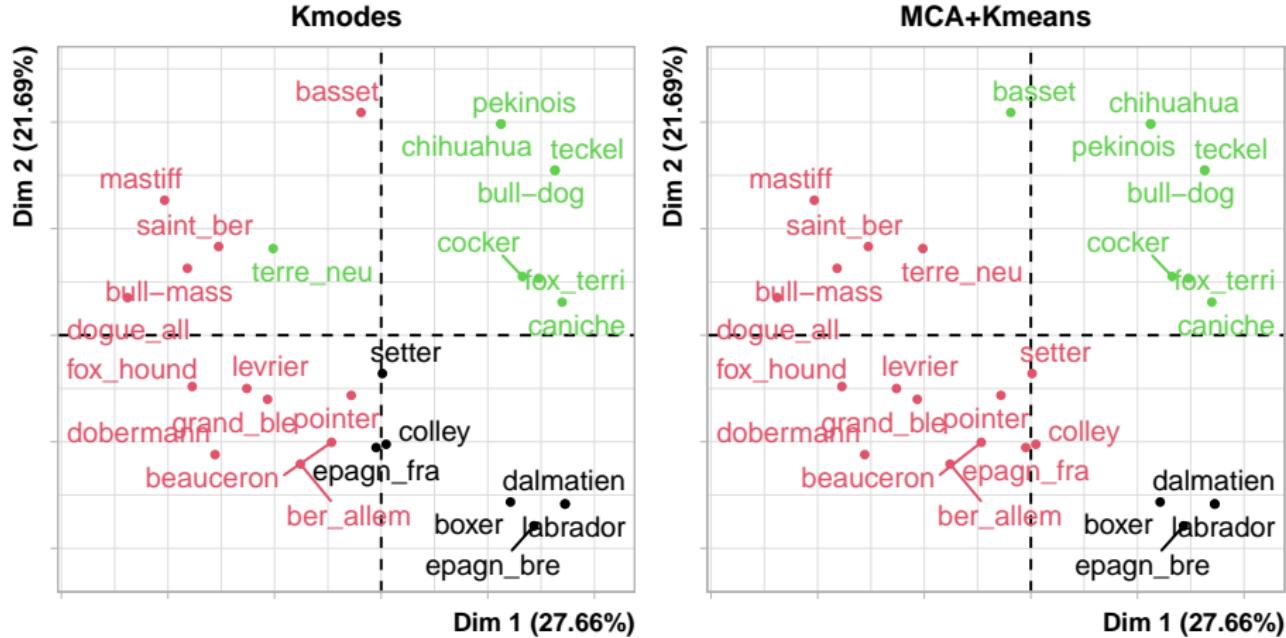
```
coeff<-MCA(chiens,quali.sup=7,graph=F)$ind$coord
clkmeans<-kmeans(coeff,3,nstart=50)
adjustedRandIndex(chiens[,7],clkmeans$cluster)
```

```
[1] 0.2824869
```

```
adjustedRandIndex(clkmodes$cluster,clkmeans$cluster)
```

```
[1] 0.4844012
```

# Exemple des chiens



## Subsection 2

### Méthodes de type K-médoides

# Méthode des K-médoïdes

- Initialisation : choix aléatoire de  $K$  points (médoïdes) parmi les  $n$
- A chaque itération  $t$  :
  - on associe chaque point  $x_i$  à son plus proche médoïde
  - pour chaque médoïde  $m_k^{(t-1)}$  : pour chaque point  $x_i$  qui n'est pas un médoïde, on échange  $x_i$  avec  $m_k^{(t-1)}$  et on calcule le coût total de la configuration
  - on sélectionne la configuration la moins coûteuse

# PAM (Partitioning Around Medoids)

Cet algorithme a été proposé par Kaufman et Rousseeuw

- Choix de  $K$  médoïdes  $c_1, \dots, c_K$  parmi les  $n$  individus
- Sélectionner au hasard un médoïde  $c_k$  et un autre objet (non médoïde)  $x_i$
- Calculer la qualité de la nouvelle partition si les rôles de  $c_k$  et  $x_i$  sont inversés
- Échanger  $c_k$  et  $x_i$  si la qualité est supérieure
- Et retourner en (2) jusqu'à stabilité de la qualité de la partition.

# Méthode des K-médoïdes

- Chaque classe est représentée par un individu de la classe donc pas de limitation sur le type de variables prises en compte
- Algorithme efficace pour de petits jeux de données
- PAM est plus robuste que  $K$ -means en présence de bruit ou d'outliers (un médoïde est moins influencé par un outlier que la moyenne)

# Commandes sous R

- `pam(x, k, diss = , metric = , ...)` [library(cluster)]  
metric = “euclidian” ou “manhattan”, sinon x matrice de dissimilarité
- `clara(x, k, metric = , ...)` [library(cluster)]  
metric = “euclidian” ou “manhattan”, sinon x matrice de dissimilarité  
fonction à privilégier si données de grande dimension

# Exemple des iris

```
A<-pam(iris[,1:4], 3, metric = "euclidean")
A$medoids
```

```
  Sepal.Length Sepal.Width Petal.Length Petal.Width
[1,]         5.0       3.4        1.5       0.2
[2,]         6.0       2.9        4.5       1.5
[3,]         6.8       3.0        5.5       2.1
```

```
A$id.med
```

```
[1] 8 79 113
```

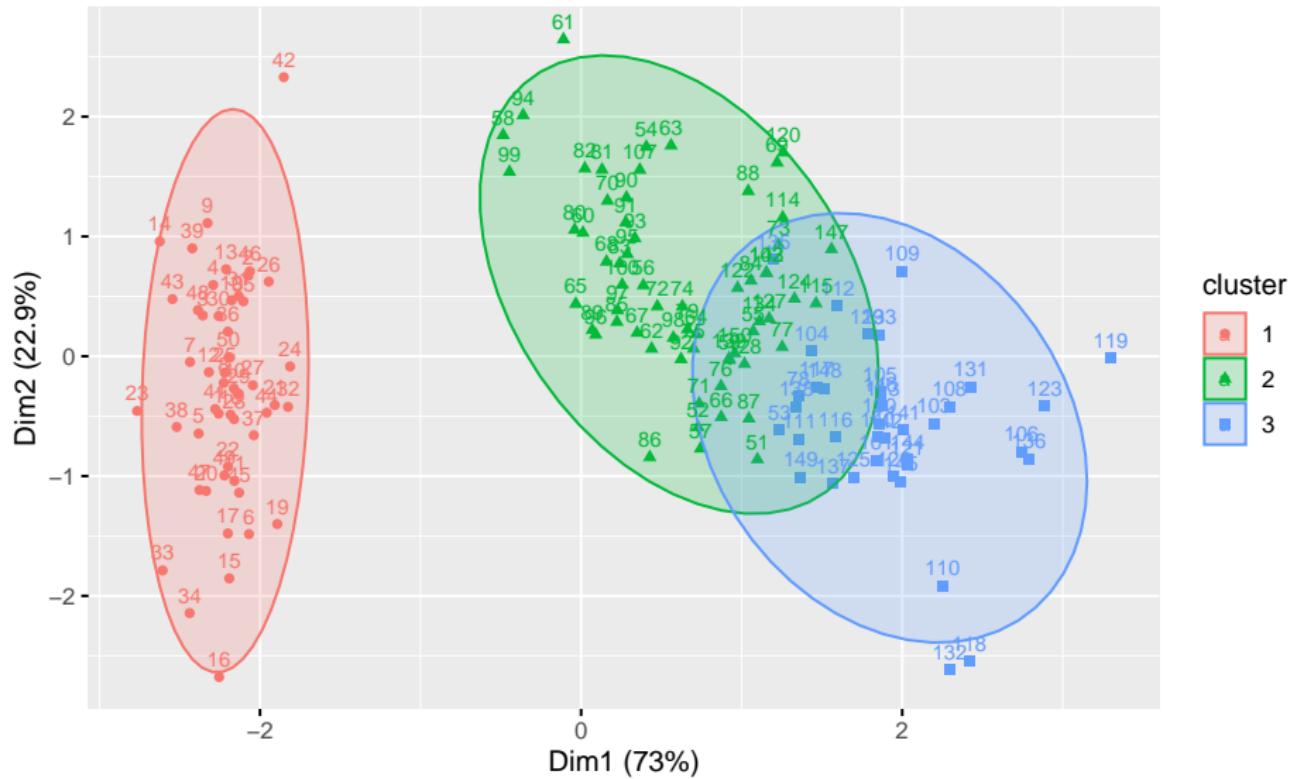
```
classError(A$clustering,iris$Species)$errorRate
```

```
[1] 0.1066667
```

```
table(iris$Species,A$clustering)
```

	1	2	3
setosa	50	0	0
versicolor	0	48	2
virginica	0	14	36

# Exemple des iris

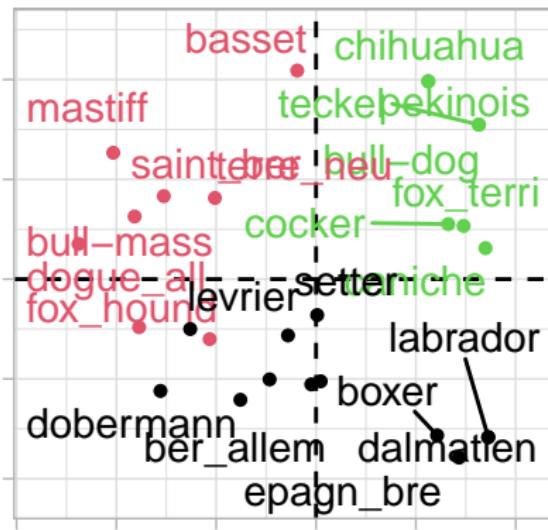


# Exemple des chiens

```
Clpam<-pam(daisy(chiens[,-7],metric="gower"),3)
table(Clpam$clustering,chiens[,7])
```

	chasse	compagnie	utilite
1	6	3	3
2	3	0	5
3	0	7	0

MCA factor map



## Subsection 3

### Choix du nombre de classes

# Problématique

- On a rarement des connaissances complémentaires pour choisir  $K$
- Pour chaque valeur de  $K \in \{2, \dots, K_{\max}\}$ , on obtient une classification et on sélectionne finalement celle où on observe un saut important de l'inertie intra-classe ("coude")
- Choix par d'autres indices basés sur les inerties intra- et inter-
- Choix par maximisation d'un indice:  
Ex: silhouette, Gap statistique, ...
- Autres critères de type sélection de modèles (voir les mélanges)
- ...

# Critères fondés sur les inerties

- **R-Square :**

$$K \mapsto RSQ(K) = 1 - \frac{I_{intra}(\mathcal{P}_K)}{I_{totale}} = \frac{I_{inter}(\mathcal{P}_K)}{I_{totale}}$$

On retient l'endroit où la courbe  $K \mapsto RSQ(K)$  forme un coude.

- **Semi-Partial R-Square :**

$$K \mapsto SPRSQ(K) = \frac{I_{inter}(\mathcal{P}_K) - I_{inter}(\mathcal{P}_{K-1})}{I_{totale}}$$

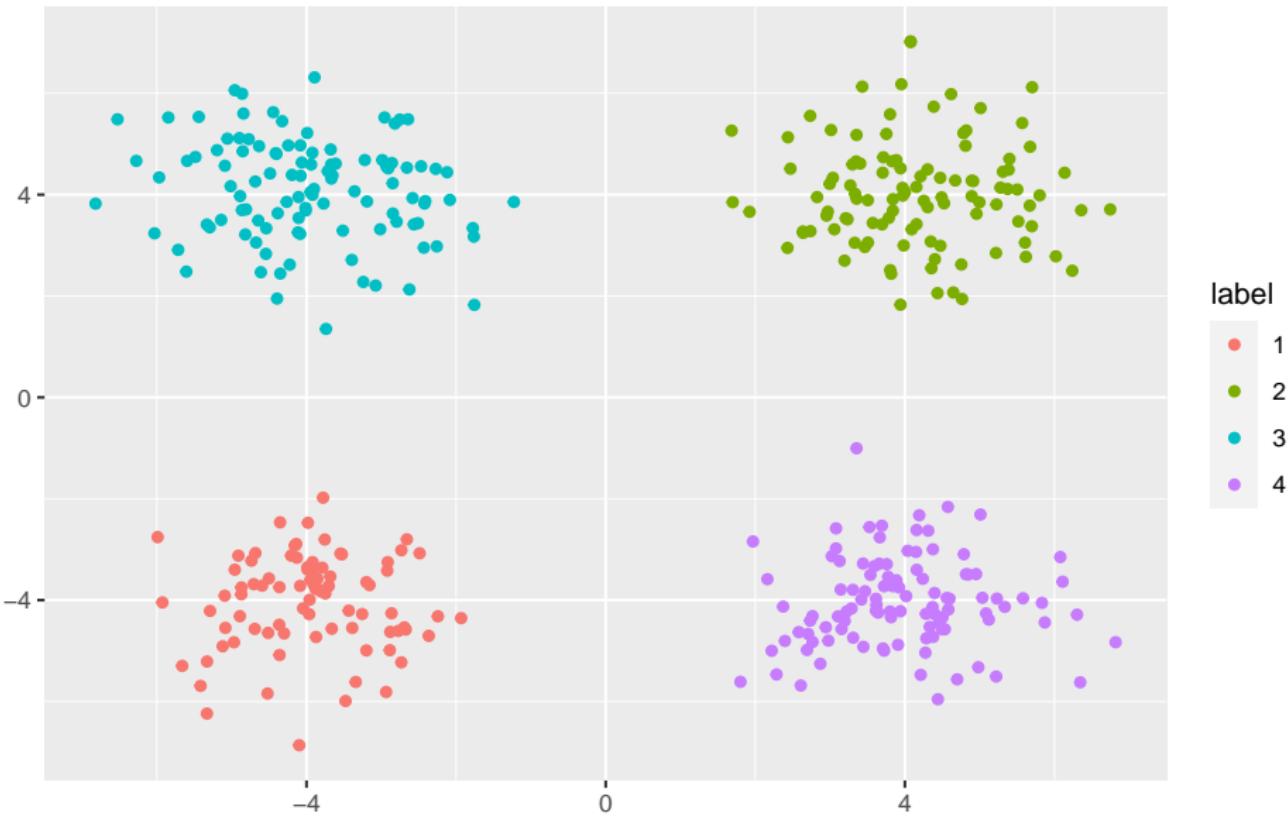
On retient l'endroit où on a la plus forte réduction du SPRSQ.

- **Calinski-Harabasz (CH) :**

$$K \mapsto PseudoF(K) = \frac{I_{inter}(\mathcal{P}_K)/(K-1)}{I_{intra}(\mathcal{P}_K)/(n-K)}$$

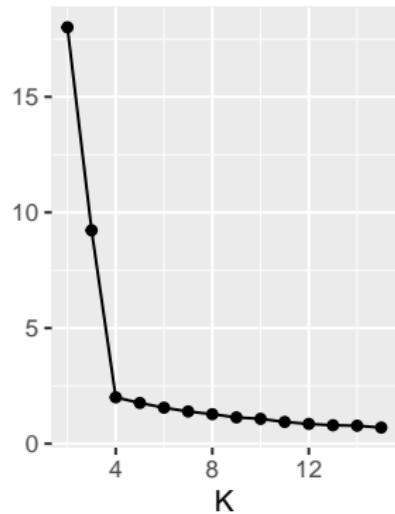
On cherche un pic sur cette courbe

# Exemple Données simulées jouet

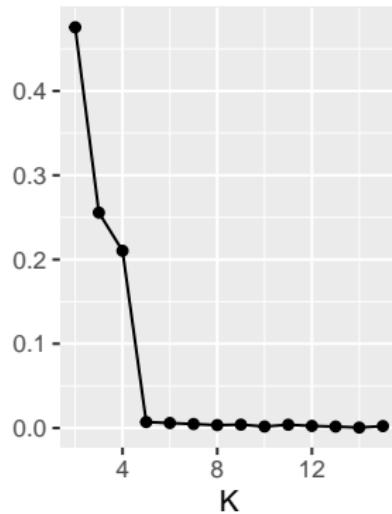


# Exemple - Critères fondés sur les inerties

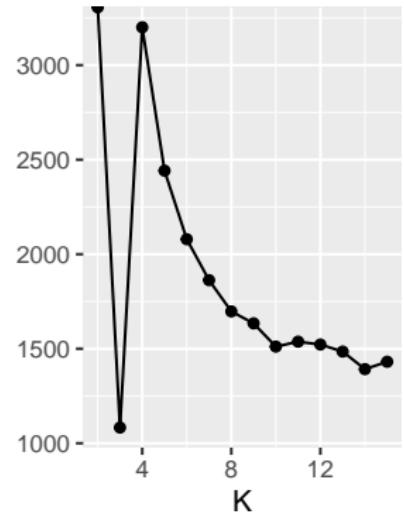
lintra



SPRSQ



Calinski–Harabasz



# Silhouette

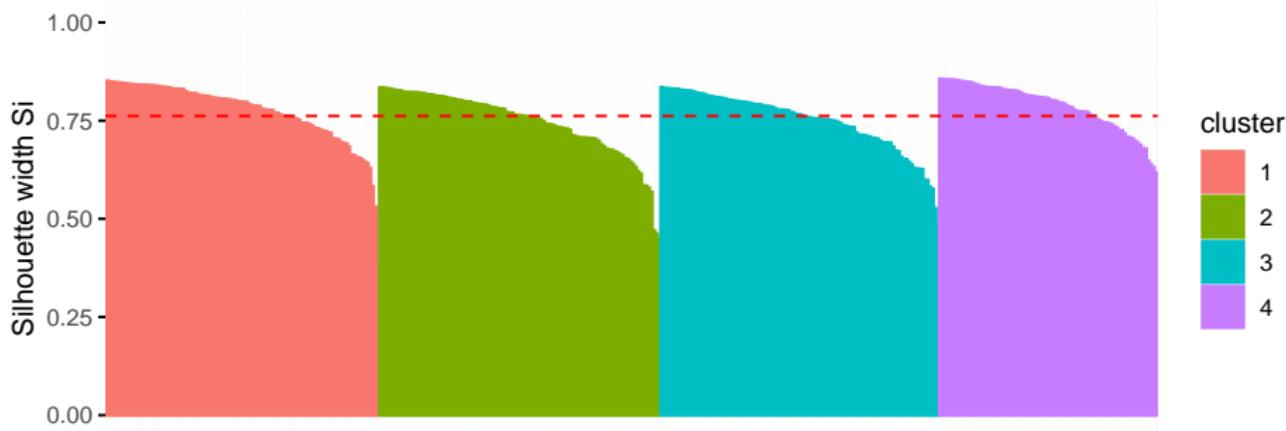
- Pour toute valeur  $K \in \{1, \dots, K_{\max}\}$ , on calcule le critère  $S(K)$  :
  - Soit  $\{\mathcal{C}_1, \dots, \mathcal{C}_K\}$  une classification de  $\{1, \dots, n\}$
  - $\forall i \in \{1, \dots, n\}, \exists! k \in \{1, \dots, K\}; i \in \mathcal{C}_k$ 
    - $a(i) = \frac{1}{|\mathcal{C}_k|-1} \sum_{\substack{\ell \in \mathcal{C}_k \\ \ell \neq i}} d(x_i, x_\ell)$
    - $b(i) = \min_{k' \neq k} \frac{1}{|\mathcal{C}_{k'}|} \sum_{\ell \in \mathcal{C}_{k'}} d(x_i, x_\ell)$
    - $s(i) = \frac{b(i)-a(i)}{\max(b(i), a(i))} \in [-1, 1]$
  - $S(K) = \frac{1}{n} \sum_{i=1}^n s(i)$
- Le nombre de classes retenu :  $\hat{K} = \underset{1 \leq K \leq K_{\max}}{\operatorname{argmax}} S(K)$
- Sous R, commande `silhouette()` [`library(cluster)`]

# Exemple - Silhouette

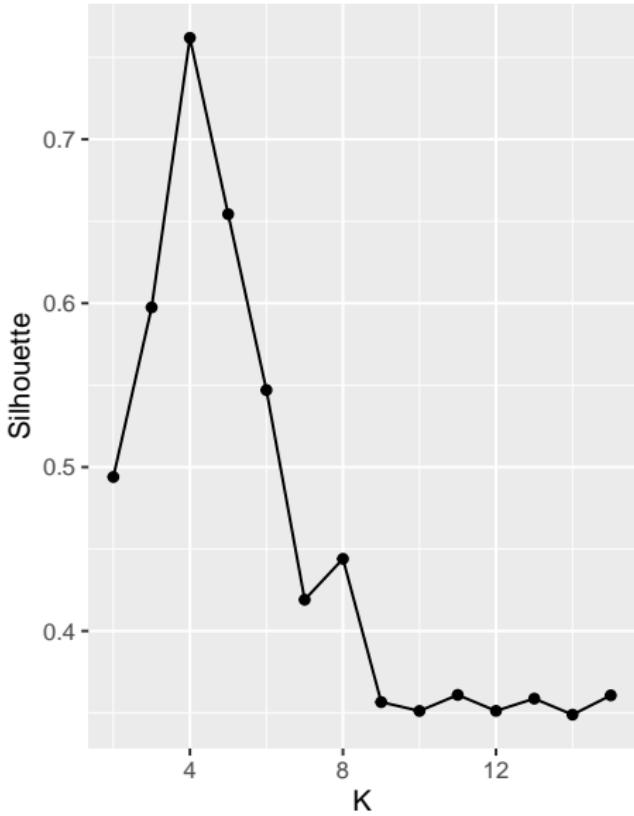
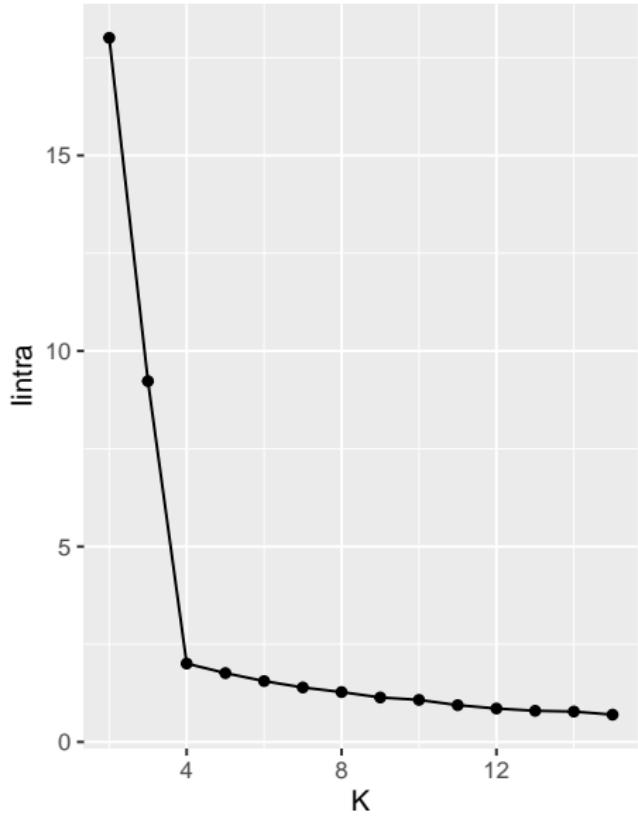
```
Classif<-kmeans(datasimu[,1:2],4,nstart=10)
aux<-silhouette(Classif$cl, daisy(datasimu[,1:2]))
library(factoextra)
fviz_silhouette(aux)+theme(plot.title = element_text(size =9))
```

	cluster	size	ave.sil.width
1	1	104	0.77
2	2	107	0.74
3	3	106	0.75
4	4	83	0.79

Clusters silhouette plot  
Average silhouette width: 0.76



# Exemple - Silhouette



## Gap statistique (Tibshirani et al.,01)

- Pour chaque  $K$  dans  $\{1, \dots, K_{\max}\}$ :

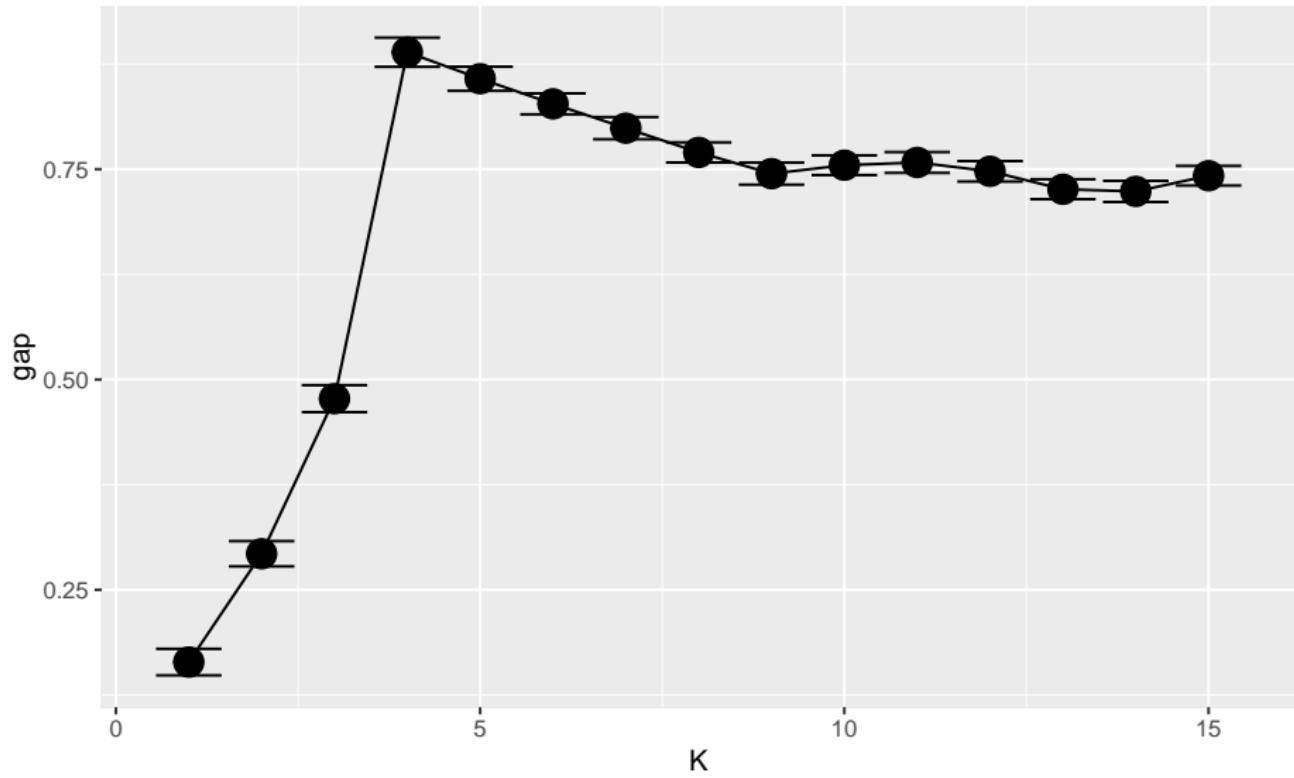
- Calculer la matrice de dispersion intra-classe

$$W_K = \frac{1}{n} \sum_{k=1}^K \sum_{i \in C_k} (x_i - m_k)'(x_i - m_k) \text{ et son déterminant } \det(W_K)$$

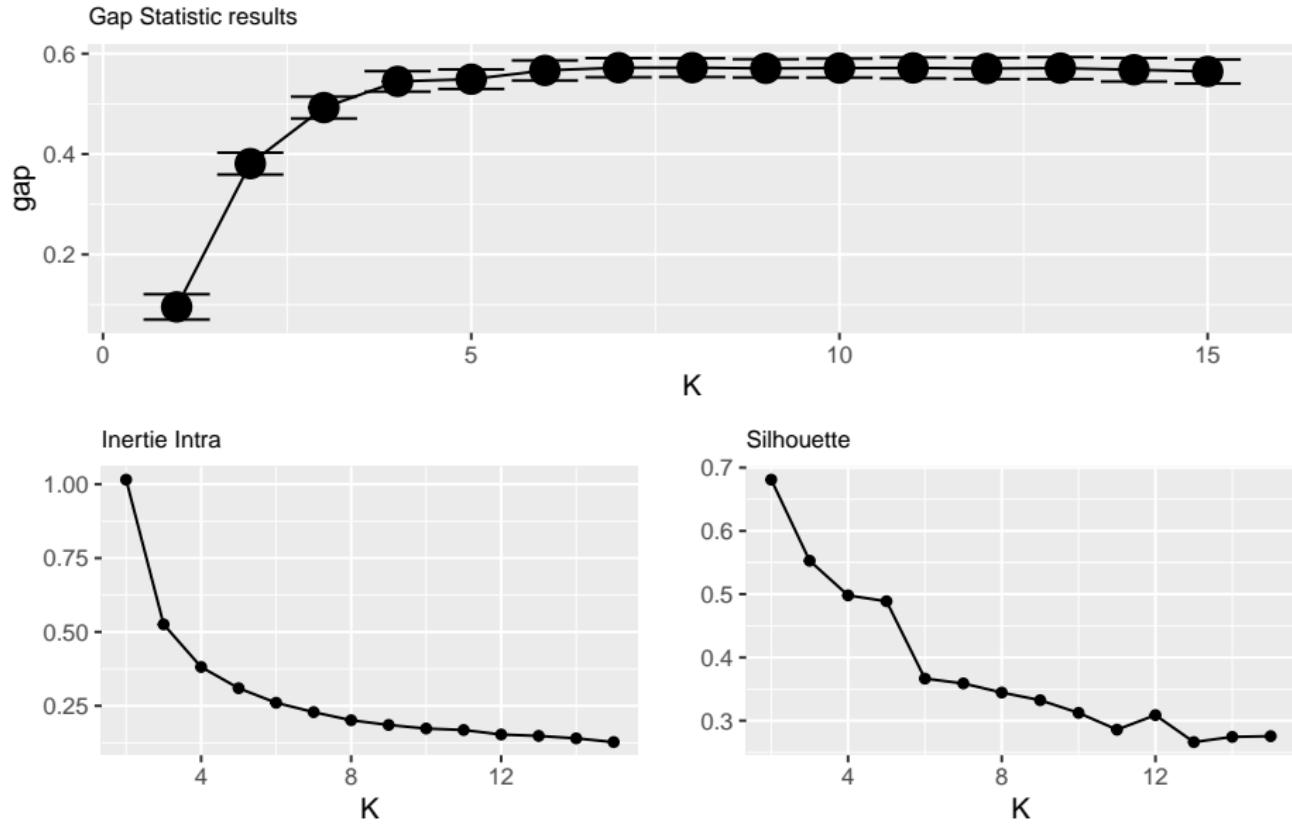
- Construire la courbe des  $\ln(\det(W_K))$  en fonction de  $K$
  - Comparer cette courbe à celle obtenue à partir des données uniformément réparties
- L'estimation du nombre de classes à retenir est la valeur  $K$  correspondant au plus grand écart entre les deux courbes
- Sous R, commande `clusGap()` [`library(cluster)`]

# Exemple - Gap Statistic

Gap Statistic results



# Exemple des Iris



## Section 4

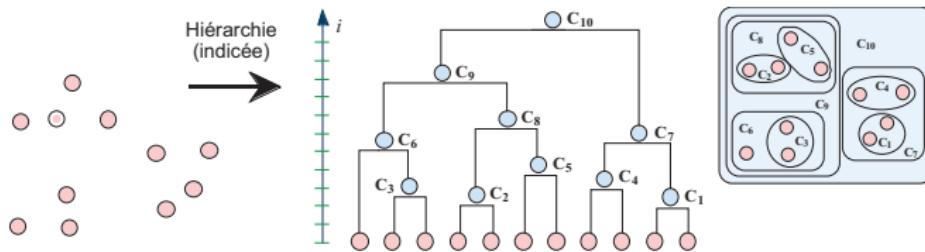
### Clustering hiérarchique

## Subsection 1

### Principe

# Algorithme général de CAH

- Initialisation :  $\mathcal{P}_n = \{\{x_1\}, \dots, \{x_n\}\}$
- Étapes agrégatives :
  - on part de la partition précédente  $\mathcal{P}_K = \{\mathcal{C}_1, \dots, \mathcal{C}_K\}$  en  $K$  classes
  - on agrège les deux classes  $\mathcal{C}_k$  et  $\mathcal{C}_{k'}$  qui minimisent une **mesure d'agrégation**  $D(\mathcal{C}_k, \mathcal{C}_{k'})$  :  $\mathcal{C}_{k \cup k'} = \mathcal{C}_k \cup \mathcal{C}_{k'}$  on obtient ainsi une partition en  $K - 1$  classes
- On recommence l'étape d'agrégation jusqu'à obtenir une partition en une seule classe



# Les choix à faire

- Choix d'une **dissimilarité**  $d$  entre les points
- Choix d'une **mesure d'agrégation**  $D$  entre classes
- Construction d'un dendrogramme (la représentation n'est pas unique)
- **Critère pour la coupure du dendrogramme** pour en déduire une classification des données
- Dans ce cours on ne parlera que de la classification ascendante hiérarchique.

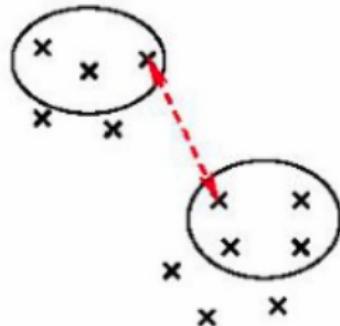
## Subsection 2

### Mesures d'agrégation entre classes

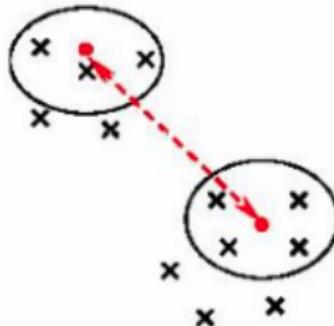
# Single / Complete / average

- **Lien simple** (*Single linkage*) :  $D(\mathcal{C}_k, \mathcal{C}_{k'}) = \min_{i \in \mathcal{C}_k, \ell \in \mathcal{C}_{k'}} d(x_i, x_\ell)$
- **Lien complet** (*Complete linkage*) :  $D(\mathcal{C}_k, \mathcal{C}_{k'}) = \max_{i \in \mathcal{C}_k, \ell \in \mathcal{C}_{k'}} d(x_i, x_\ell)$
- **Lien moyen** (*Average linkage*)  $D(\mathcal{C}_k, \mathcal{C}_{k'}) = \frac{1}{|\mathcal{C}_k||\mathcal{C}_{k'}|} \sum_{i \in \mathcal{C}_k} \sum_{\ell \in \mathcal{C}_{k'}} d(x_i, x_\ell)$

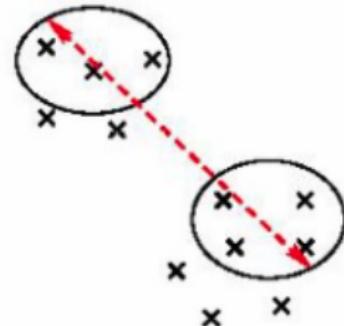
- Simple linkage



- Average linkage



- Complete linkage



# Mesures d'agrégation de Ward

- Mesure d'agrégation de Ward

$$D(\mathcal{C}_k, \mathcal{C}_{k'}) = \frac{|\mathcal{C}_k||\mathcal{C}_{k'}|}{n(|\mathcal{C}_k| + |\mathcal{C}_{k'}|)} d(m_k, m_{k'})^2$$

où  $m_k$  (resp.  $m_{k'}$ ) centre de gravité de  $\mathcal{C}_k$  (resp.  $\mathcal{C}_{k'}$ ) et  $d$  est une distance euclidienne.

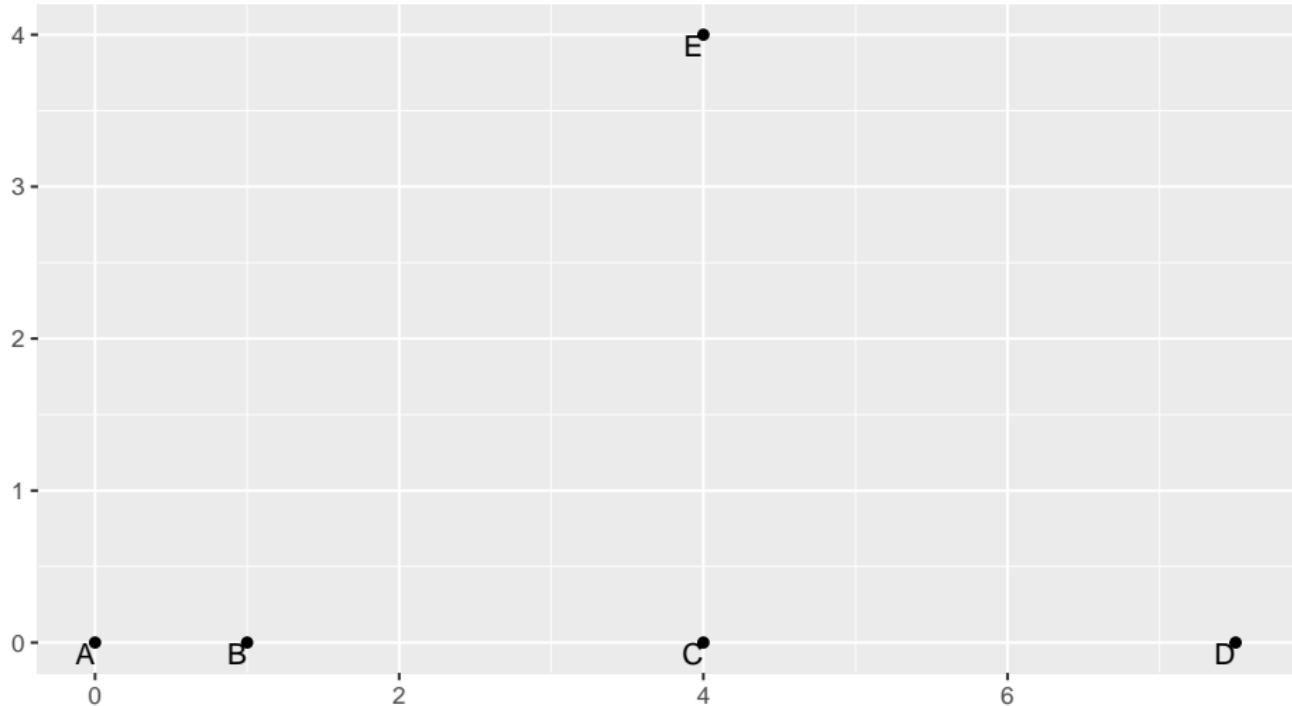
- Proposition : Si l'on rassemble les deux classes  $\mathcal{C}_k$  et  $\mathcal{C}_{k'}$  en une classe notée  $\mathcal{C}_{k \cup k'}$  alors l'inertie interclasse diminue (l'inertie intraclasse augmente) de :

$$\frac{|\mathcal{C}_k||\mathcal{C}_{k'}|}{n(|\mathcal{C}_k| + |\mathcal{C}_{k'}|)} d(m_k, m_{k'})^2$$

- Méthode de Ward : Elle consiste à choisir à chaque étape les deux classes dont le regroupement implique une augmentation minimale de l'inertie intraclasse.

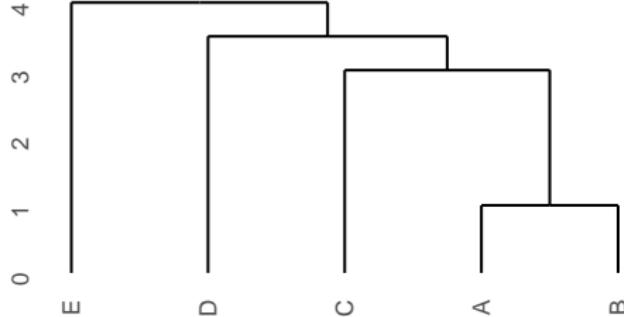
# Exemple jouet

- Données : 5 points de  $\mathbb{R}^2$
- $d$  = la distance euclidienne usuelle



# Exemple jouet

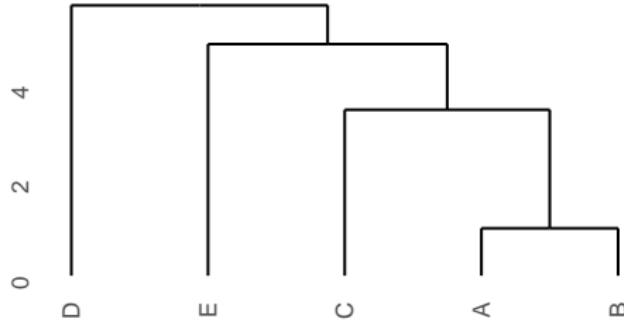
Single linkage



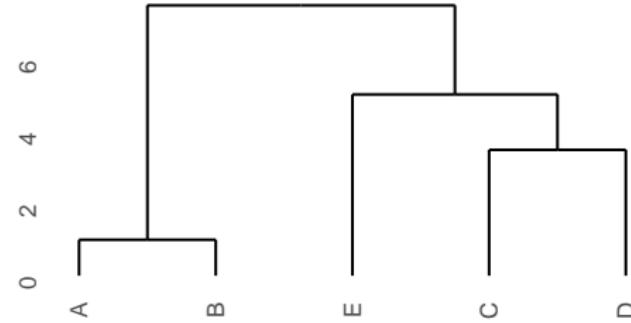
Complete linkage



Average linkage



Ward linkage



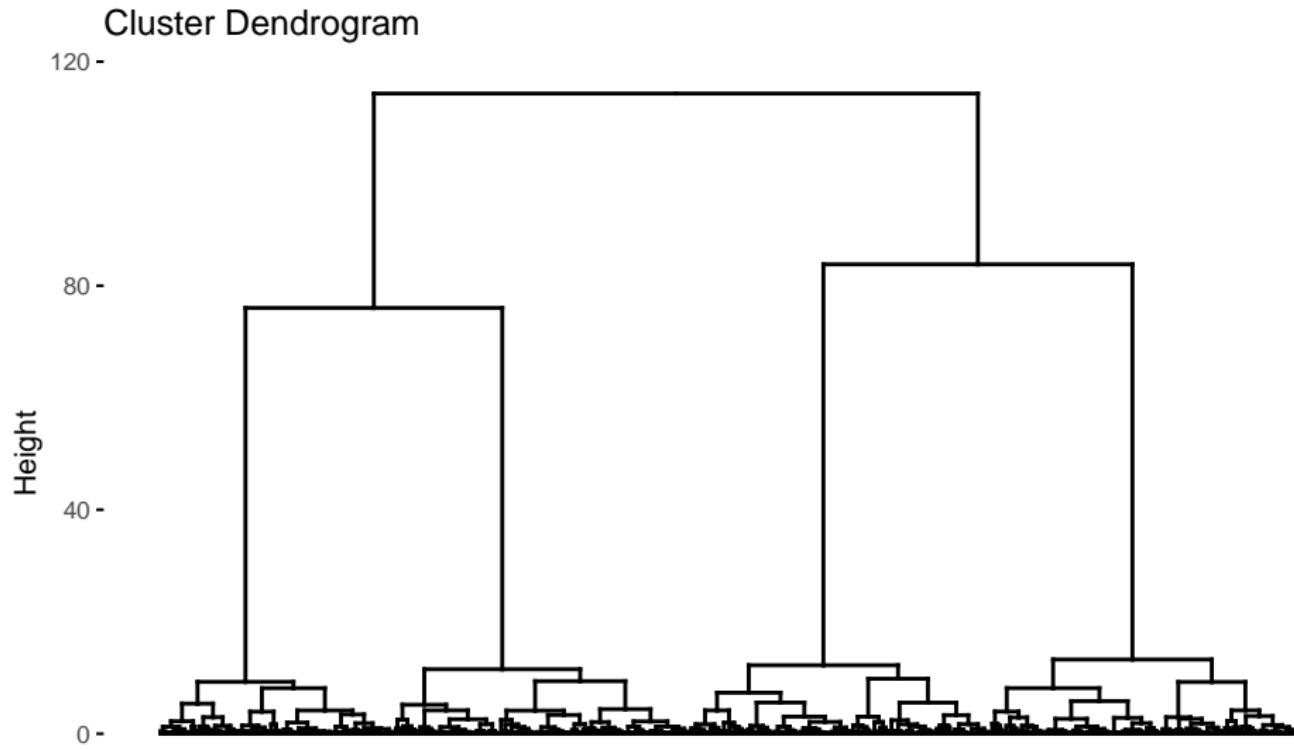
## Subsection 3

### Coupe du dendrogramme

# Comment faire ?

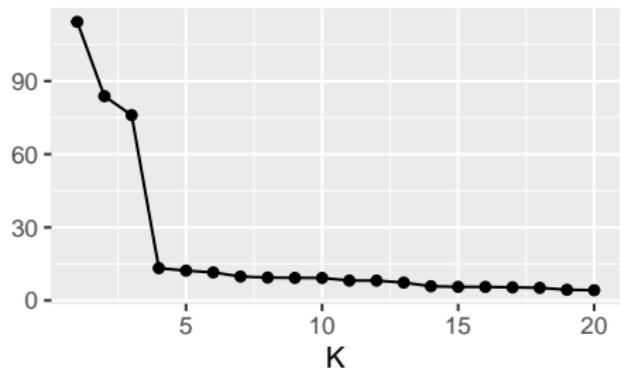
- Le choix du niveau de coupure du dendrogramme détermine le nombre de classes et ces classes sont alors uniques
- On peut définir la coupure du dendrogramme en déterminant à l'avance le nombre de classes dans lesquels on désire répartir l'ensemble des données
- Le choix du niveau de coupure peut être facilité par l'examen des indices croissants de niveau de l'arbre hiérarchique
- On peut aussi faire ce choix en utilisant les indices vus dans la section précédente ( $R^2$ , CH, Silhouette, Gap Statistic, ...)

# Exemple des données simulées

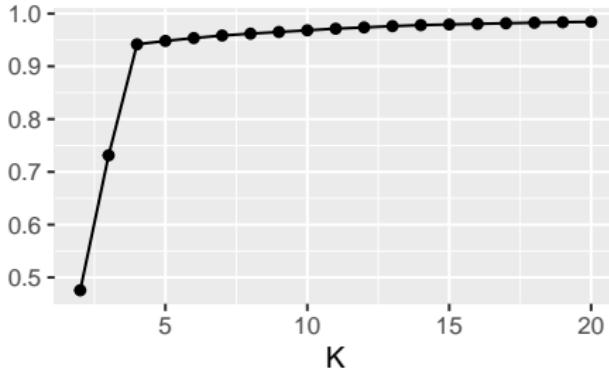


# Exemple des données simulées

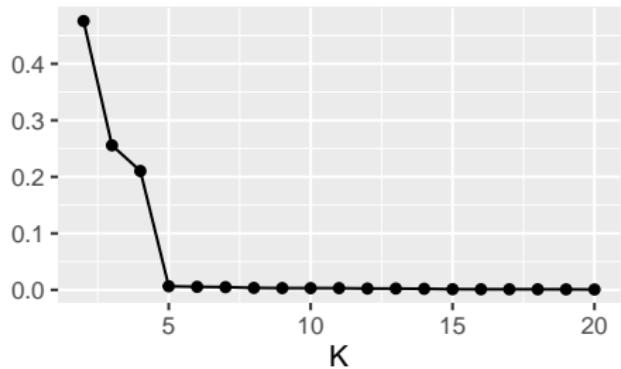
height



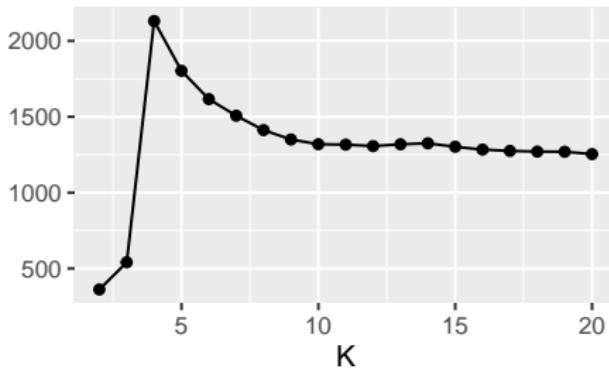
RSQ



SPRSQ

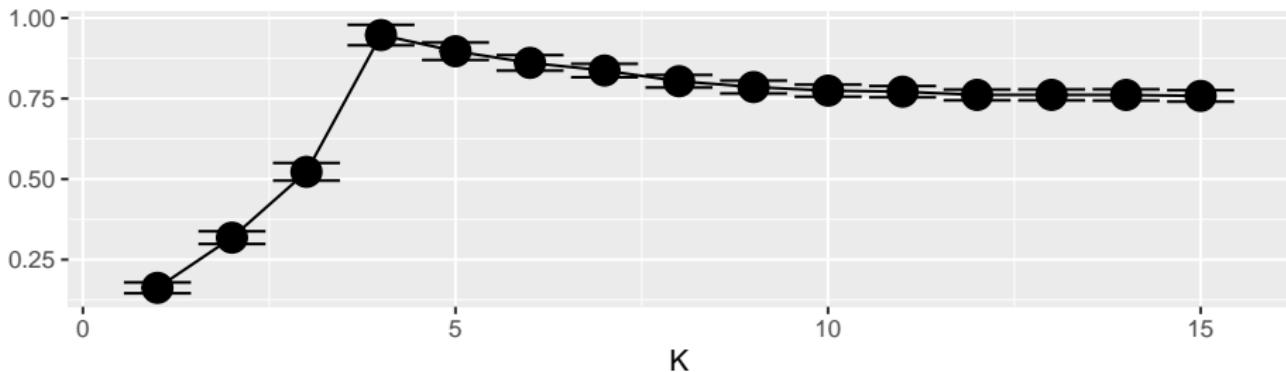


Calinski–Harabasz

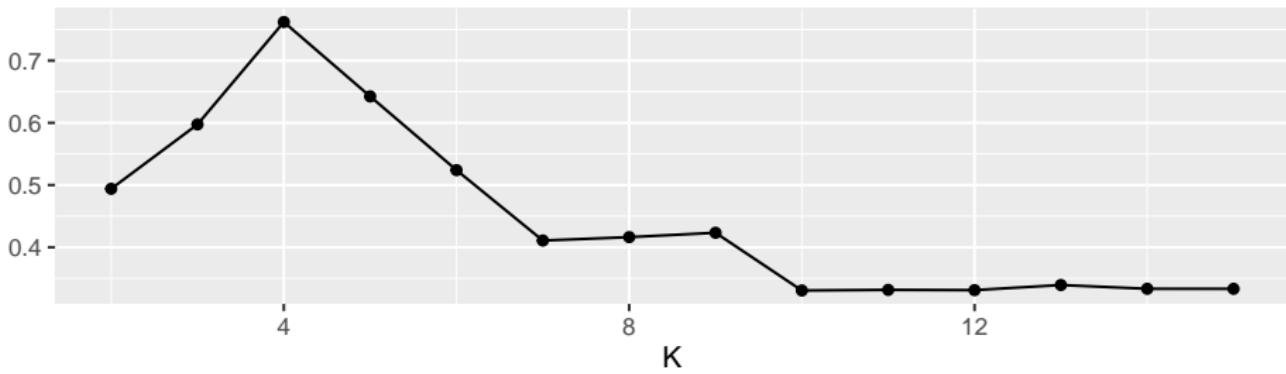


# Exemple des données simulées

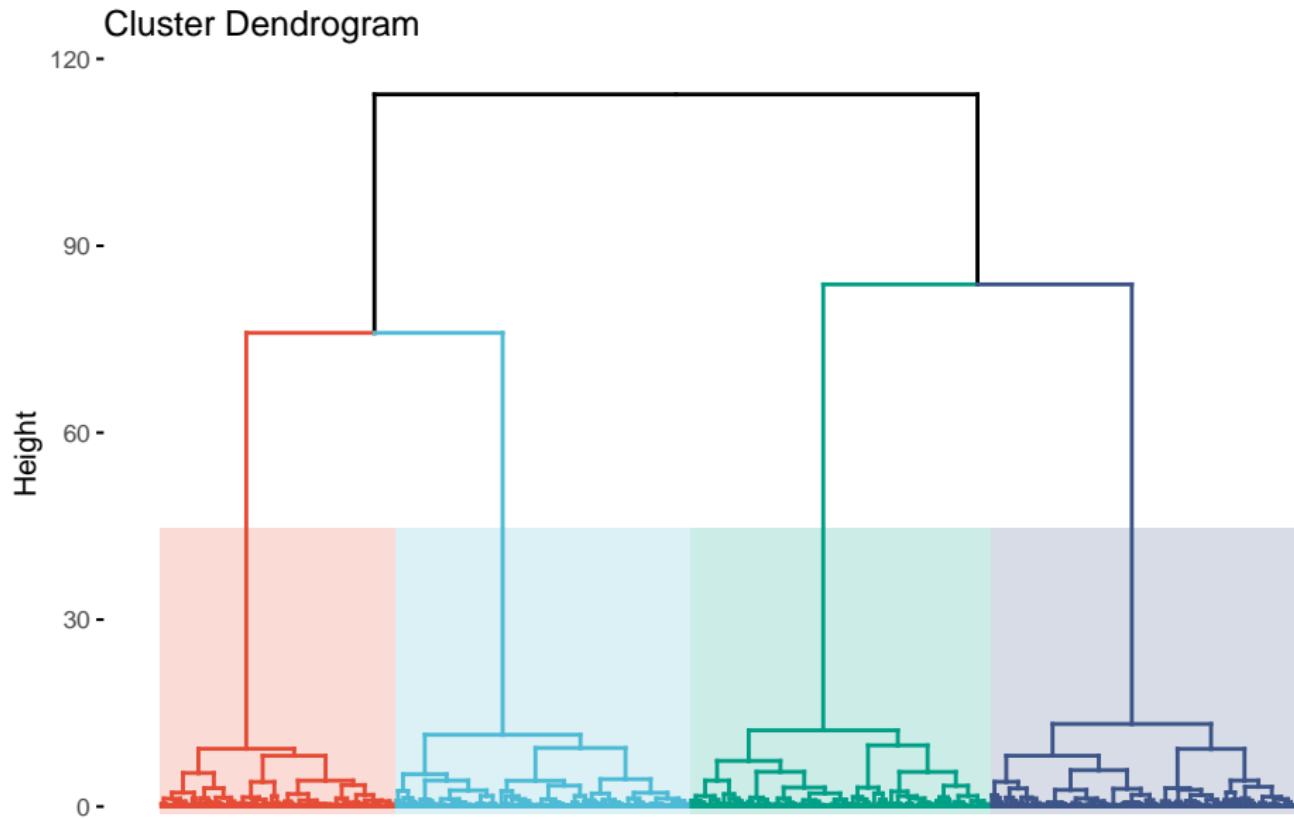
Gap Statistic results



Silhouette



# Exemple des données simulées



## Subsection 4

### Applications

# Quelques commandes avec R

- `hc=hclust(d,method=)`
  - *d* : tableau de distances comme produit par `dist()`
  - *method* : méthode d'agrégation “ward.D2”, “single”, “complete”, “average”, ...
- `names(hc) [1]` “merge” “height” “order” “labels” “method” “call” “dist.method”
- `plot(hc,hang=,...)` ou `ggdendrogram(hc,...)` ou `fviz_dend()` : permet de dessiner le dendrogramme
- `cutree(hc,k=..)` pour obtenir la classification en *k* classes
- `HCPC()` [`library(FactoMineR)`] : permet de combiner PCA et CAH

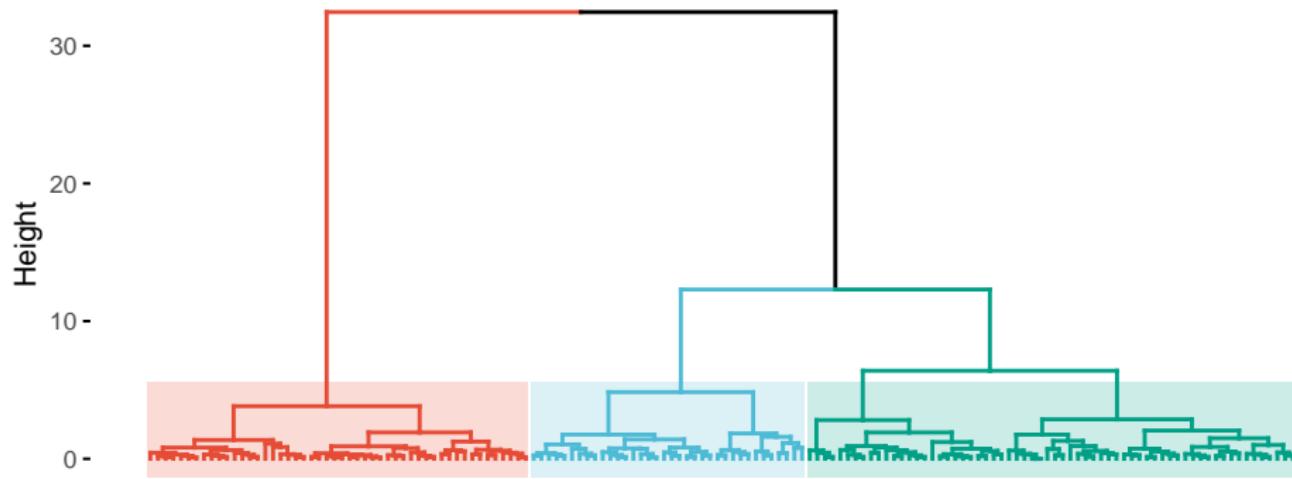
# Exemple des iris

```
hirisward<-hclust(dist(iris[1:4],method="euclidean"),method="ward.D2")
clward<-cutree(hirisward,3)
table(clward,iris$Species)
```

```
clward setosa versicolor virginica
 1      50          0          0
 2      0          49         15
 3      0          1         35
```

```
fviz_dend(hirisward,k=3,rect = TRUE, rect_fill = TRUE,palette = "npg",rect_border = "npg",show_labels = FALSE)
```

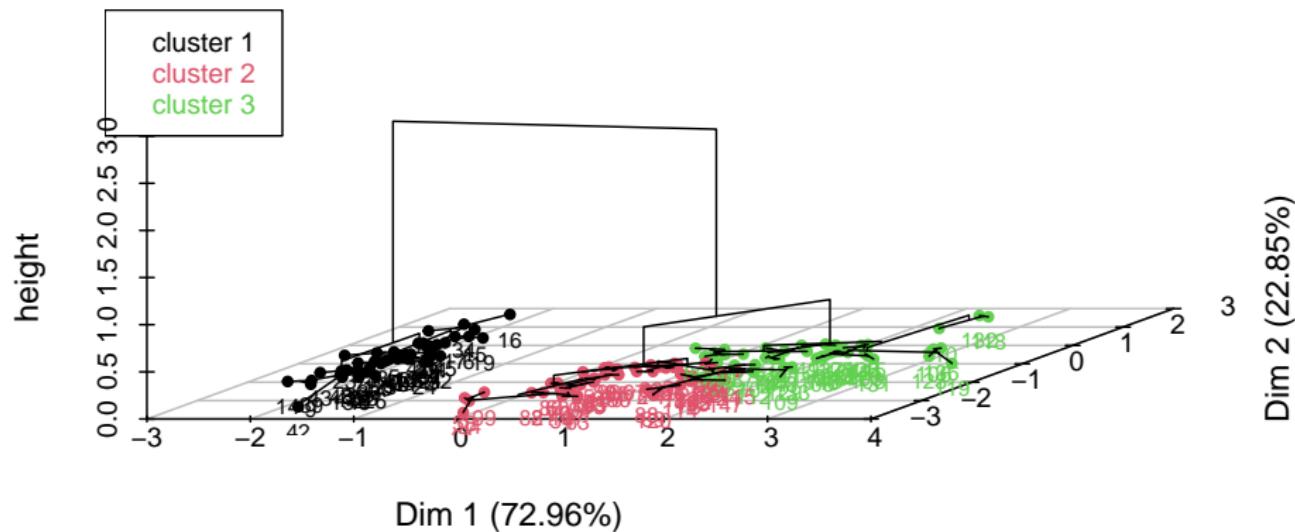
Cluster Dendrogram



# Exemple des iris

```
res.pca <- PCA(iris[,1:4], ncp=2, graph=FALSE)
res.hcpc <- HCPC(res.pca, graph=F)
plot(res.hcpc, choice = "3D.map")
```

Hierarchical clustering on the factor map



## Subsection 5

### Conclusion

# Avantages et inconvénients CAH

- Avantages :
  - Méthode flexible pour le niveau de finesse de la classification
  - Prise en compte facile de distances et d'indices de similarité de n'importe quel type
  - Rapide d'exécution et reproductible
- Inconvénients :
  - Choix de la coupure de l'arbre
  - La partition obtenue à une étape dépend de celle à l'étape précédente
  - Les algorithmes fournissent toujours des classes à partir de n'importe quelles données

## Section 5

# Clustering par modèles de mélanges

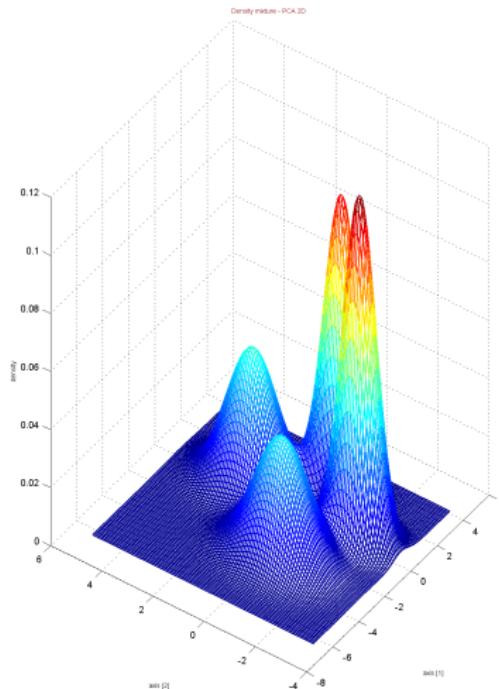
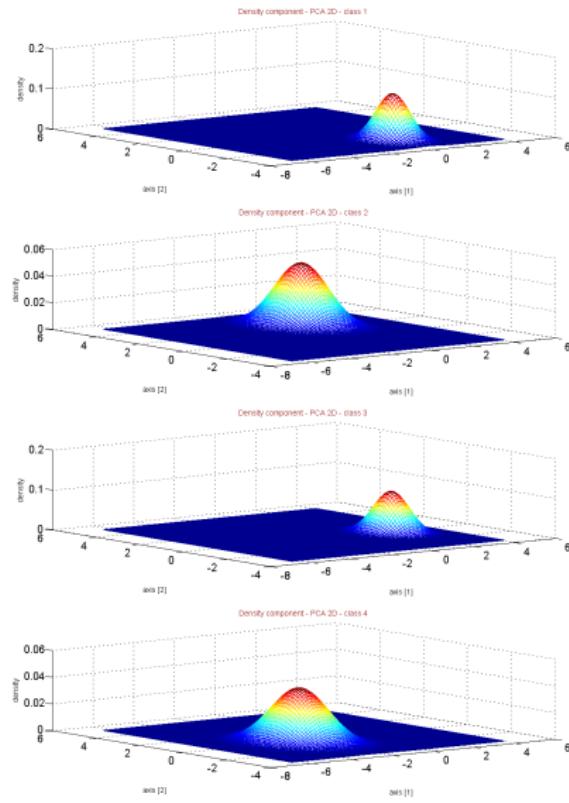
## Subsection 1

### Principe

# Hypothèses des mélanges finis

- On suppose que les données proviennent d'une population contenant plusieurs sous-populations
- Chaque sous-population est modélisée indépendamment des autres **(choix d'une loi de distribution pour chaque sous-population)**.
- La population totale est alors vue comme un mélange de ces sous-populations. Le modèle résultant est un modèle de mélange fini.
- C'est une approche probabiliste !

# Schéma



## Définition d'un mélange fini [McLachlan and Peel,00]

- Un modèle de mélange à  $K$  composants est de la forme

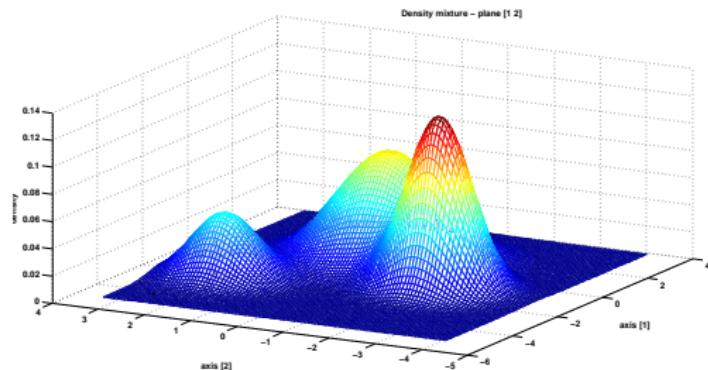
$$f(.|\theta_K) = \sum_{k=1}^K \pi_k f_k(.|\alpha_k)$$

- $(\pi_1, \dots, \pi_K)$  sont les proportions du mélange

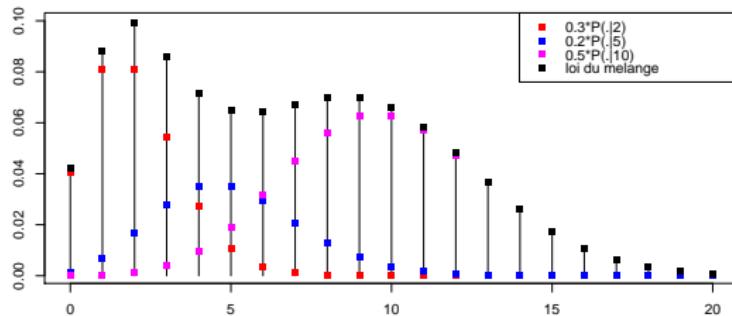
$$\forall k \in \{1, \dots, K\}, \pi_k \in [0, 1] \text{ et } \sum_{k=1}^K \pi_k = 1$$

- $f_k(.|\alpha_k)$  est la densité de la  $k$ ème sous-population
- $\theta_K = (\pi_1, \dots, \pi_K, \alpha_1, \dots, \alpha_K)$
- Le choix des  $f_k$  dépend de la nature des données

# Exemples



## Mélanges de lois gaussiennes (normales)



## Mélanges de lois de Poisson

# Les grandes étapes

- ① Collection de modèles :

$$\forall K \in \mathbb{N}^*, \mathcal{S}_K = \left\{ x \in \mathbb{R}^p \mapsto f(x|\theta_K) = \sum_{k=1}^K \pi_k f_k(\cdot|\alpha_k) \right\}$$

Mélanges à  
K=2 classes

Mélanges à  
K=3 classes

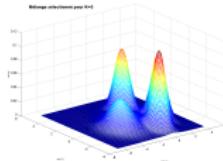
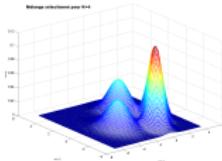
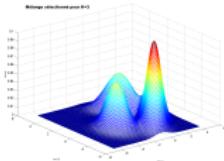
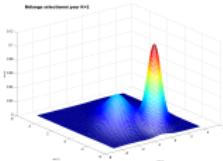
Mélanges à  
K=4 classes

Mélanges à  
K=5 classes

...

⇒ Choix initial de la modélisation

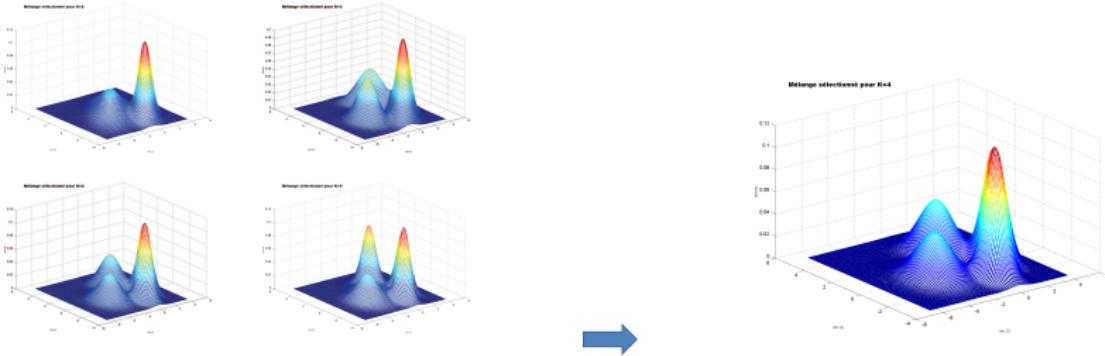
- ② Dans chaque modèle  $\mathcal{S}_K$  : on détermine le mélange qui s'ajuste le mieux aux données:  $f(\cdot|\hat{\theta}_K)$



⇒ Besoin d'un algorithme d'estimation des paramètres ( $\hat{\theta}_K$ )

# Les grandes étapes

- ③ Choisir le “meilleur” mélange parmi  $f(.|\hat{\theta}_2), f(.|\hat{\theta}_3), \dots, f(.|\hat{\theta}_{K_{\max}})$



⇒ Besoin d'un critère de sélection de modèles pour déterminer  $\hat{K}$  et donc choisir  $f(.|\hat{\theta}_{\hat{K}})$ .

- ④ Règle du “MAP” pour en déduire une classification des données : on attribue un individu  $i$  à la classe pour laquelle il a la plus forte probabilité d'appartenance.

## Etape 2 : Estimation du maximum de vraisemblance

- On désire déterminer le vecteur des paramètres  $\theta_K = (\pi_1, \dots, \pi_K, \alpha_1, \dots, \alpha_K)$  qui maximise la logvraisemblance :

$$\mathcal{L}(\underline{\mathbf{x}}|\theta_K) = \mathcal{L}(x_1, \dots, x_n|\theta_K) = \sum_{i=1}^n \ln \left[ \sum_{k=1}^K \pi_k f_k(x_i|\alpha_k) \right]$$

- Ce problème de maximisation ne possède généralement pas de solution analytique
- Utilisation d'algorithmes itératifs pour l'estimation

# Algorithmes de type EM

- Algorithme EM [Dempster et al., 77]

EM = Expectation Maximization

- Algorithme CEM [Celeux and Govaert, 92]

CEM = Classification EM

- Algorithme SEM [Celeux and Diebolt, 85]

SEM = Stochastique EM

- Algorithme SAEM [Delyon et al., 99]

SAEM = Stochastic Approximation EM

- Algorithme MCEM [Wei and Tanner, 90]

MCEM = Monte Carlo EM

## Etape 3 : Critère de sélection de modèle

$$\hat{K} = \underset{K}{\operatorname{argmin}} \operatorname{crit}(K) = \underset{K}{\operatorname{argmin}} -\mathcal{L}(\underline{\mathbf{x}}|\hat{\theta}_K) + \operatorname{pen}(K)$$

- **AIC** (Akaike Information Criterion) [Akaike, 73]

$$\operatorname{crit}(K) = -\mathcal{L}(\underline{\mathbf{x}}|\hat{\theta}_K) + \nu_K$$

- **BIC** (Bayesian Information Criterion) [Schwarz, 78]

$$\operatorname{crit}(K) = -\mathcal{L}(\underline{\mathbf{x}}|\hat{\theta}_K) + \frac{\nu_K}{2} \ln(n)$$

- **ICL** (Integrated Completed Likelihood) [Biernacki et al., 00]

$$\operatorname{crit}(K) = -\mathcal{L}(\underline{\mathbf{x}}|\hat{\theta}_K) + \frac{\nu_K}{2} \ln(n) + \operatorname{Ent}(K)$$

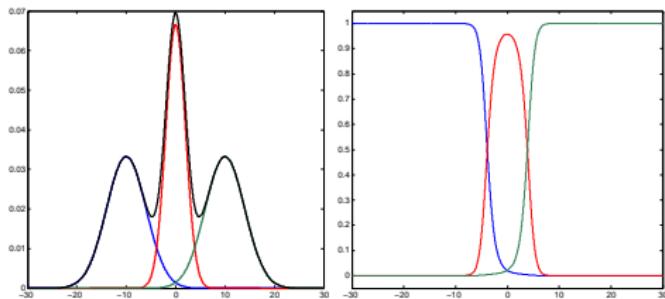
où  $\nu_K$  est nombre de paramètres libres des mélanges de  $\mathcal{S}_K$

$\operatorname{Ent}(K) = - \sum_{i=1}^n \sum_{k=1}^K \hat{z}_{ik} \ln[\hat{t}_{ik}(\hat{\theta}_K)]$  est un terme d'entropie

## Etape 4 : Règle du MAP

- Principe : chaque individu est affecté à la classe pour laquelle il a la plus forte probabilité d'appartenance conditionnellement à l'estimation des paramètres
- Probabilité conditionnelle d'appartenance de  $i$  à  $\mathcal{C}_k$  avec le vecteur de paramètres  $\theta$

$$\begin{aligned} t_{ik}(\theta) &= P(z_{ik} = 1 | x_i, \theta) \\ &= \frac{\pi_k f_k(x_i | \alpha_k)}{\sum_{\ell=1}^K \pi_\ell f_\ell(x_i | \alpha_\ell)} \end{aligned}$$



- Règle du maximum a posteriori avec  $\hat{\theta}_{\hat{K}}$ :

$$i \in \mathcal{C}_k \text{ si } t_{ik}(\hat{\theta}_{\hat{K}}) > t_{i\ell}(\hat{\theta}_{\hat{K}}) \forall \ell \neq k$$

## Subsection 2

### Les mélanges gaussiens multivariés

# Mélanges gaussiens multivariés

- Données  $\underline{x} = (x_1, \dots, x_n)$  avec  $x_i \in \mathbb{R}^p$
- Les observations sont supposées être un échantillon de loi

$$f(\cdot | \theta_{K,m}) = \sum_{k=1}^K \pi_{k,m} \phi(\cdot | \mu_k, \Sigma_{k,m})$$

- $\phi(\cdot | \mu_k, \Sigma_k)$  la densité d'une loi  $\mathcal{N}_p(\mu_k, \Sigma_k)$
- $\theta_{K,m} = (\pi_{1,m}, \dots, \pi_{K,m}, \mu_1, \dots, \mu_K, \Sigma_{1,m}, \dots, \Sigma_{K,m})$
- Collection de modèles:  $(\mathcal{S}_{(K,m)})_{(K,m) \in \mathbb{N}^* \times \mathcal{M}}$  avec
  - $\mathcal{S}_{(K,m)} = \{x \in \mathbb{R}^p \mapsto f(x | \theta_{K,m}); \theta_{K,m} \in \Theta_{K,m}\}$
  - $\mathcal{M}$  = ensemble de formes de mélanges gaussiens

# Formes $m$ des mélanges gaussiens

- 14 formes (contraintes sur la décomposition en valeur propre) réparties en 3 familles (sphérique, diagonale, générale)
- Proportions supposées égales ou libres

⇒ 28 formes de mélanges gaussiens possibles



# Exemple des iris

```
library(mclust)
# Selection avec BIC
modBIC=mclustBIC(iris[,-5],G=1:9)
summary(modBIC)
```

Best BIC values:

	VEV,2	VEV,3	VVV,2
BIC	-561.7285	-562.5522369	-574.01783
BIC diff	0.0000	-0.8237748	-12.28937

# Récup du modèle sélectionné par BIC

```
mBIC1 = Mclust(iris[,-5], x = modBIC)
summary(mBIC1,parameters=F)
```

---

Gaussian finite mixture model fitted by EM algorithm

---

Mclust VEV (ellipsoidal, equal shape) model with 2 components:

log-likelihood	n	df	BIC	ICL
-215.726	150	26	-561.7285	-561.7289

Clustering table:

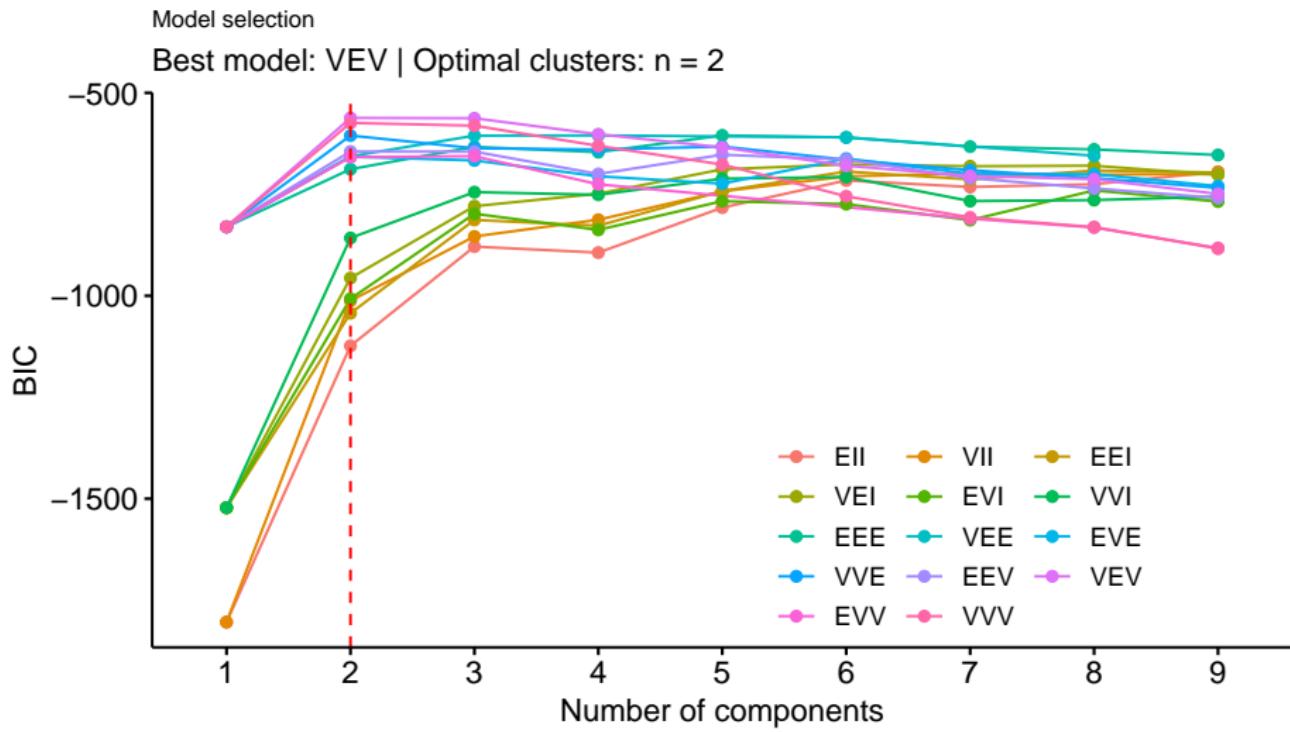
1	2
50	100

# pour accéder aux paramètres

```
# mBIC1$parameters$pro; mBIC1$parameters$mean; mBIC1$parameters$variance$sigma
```

# Exemple des iris

```
#plot(modBIC)
fviz_mclust_bic(mBIC1)+theme(plot.title = element_text(size =9))
```



# Exemple des iris

```
table(mBIC1$classification,iris$Species)
```

	setosa	versicolor	virginica
1	50	0	0
2	0	50	50

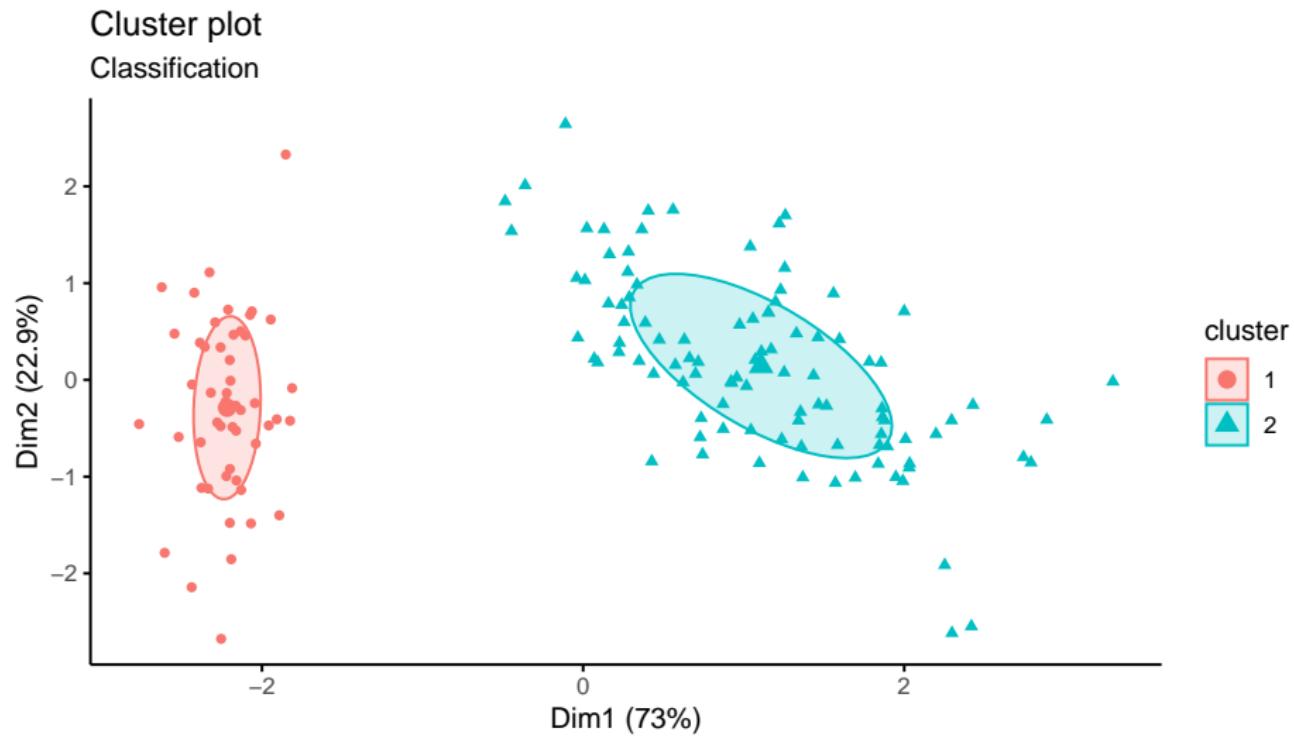
```
# On peut faire la même chose avec mclustICL pour le critere ICL
modICL=mclustICL(iris[,-5],G=1:9)
summary(modICL)
```

Best ICL values:

	VEV,2	VEV,3	VVV,2
ICL	-561.7289	-566.467287	-574.01910
ICL diff	0.0000	-4.738411	-12.29022

# Exemple des iris

```
#plot(mBIC1, what="classification")
fviz_mclust(mBIC1,what="classification",geom="point")
```



# Exemple des iris

```
mBIC5=Mclust(iris[,-5],G=5)
summary(mBIC5)
```

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

```
Mclust EEE (ellipsoidal, equal volume, shape and orientation) model with 5
components:
```

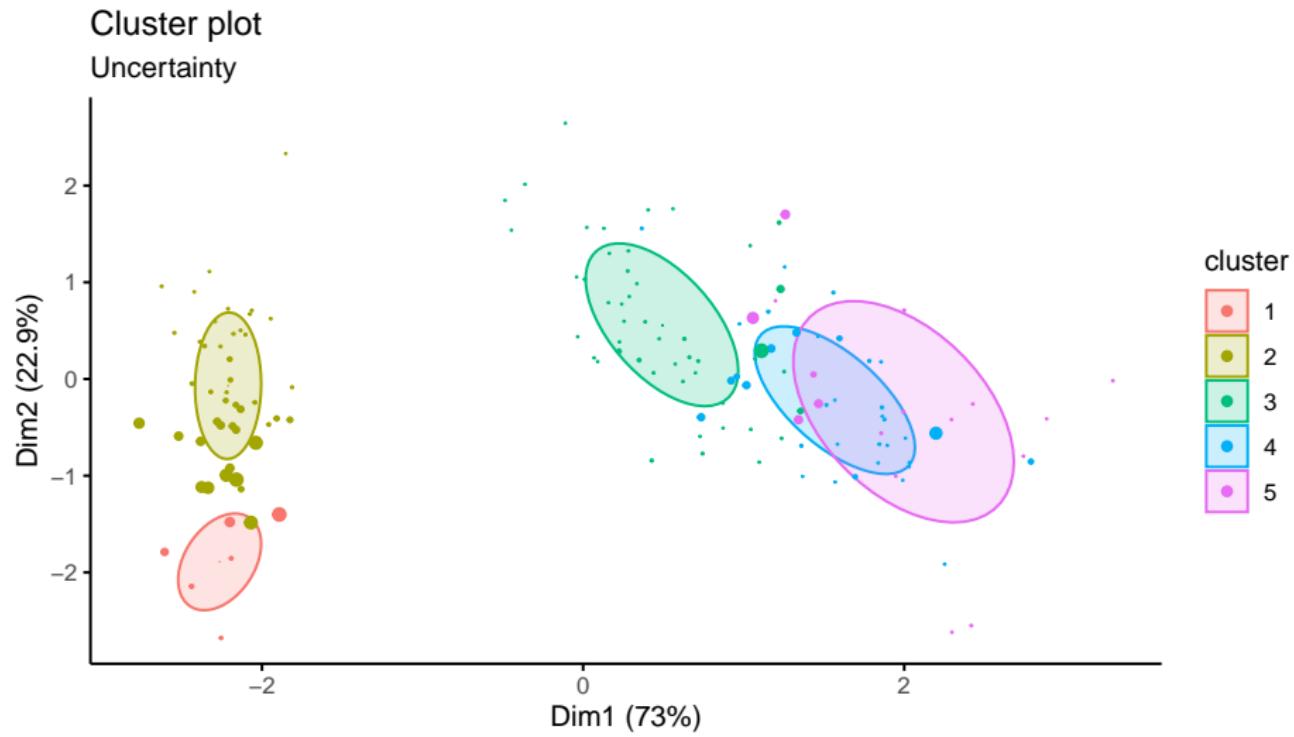
log-likelihood	n	df	BIC	ICL
-217.2257	150	34	-604.8131	-619.7786

Clustering table:

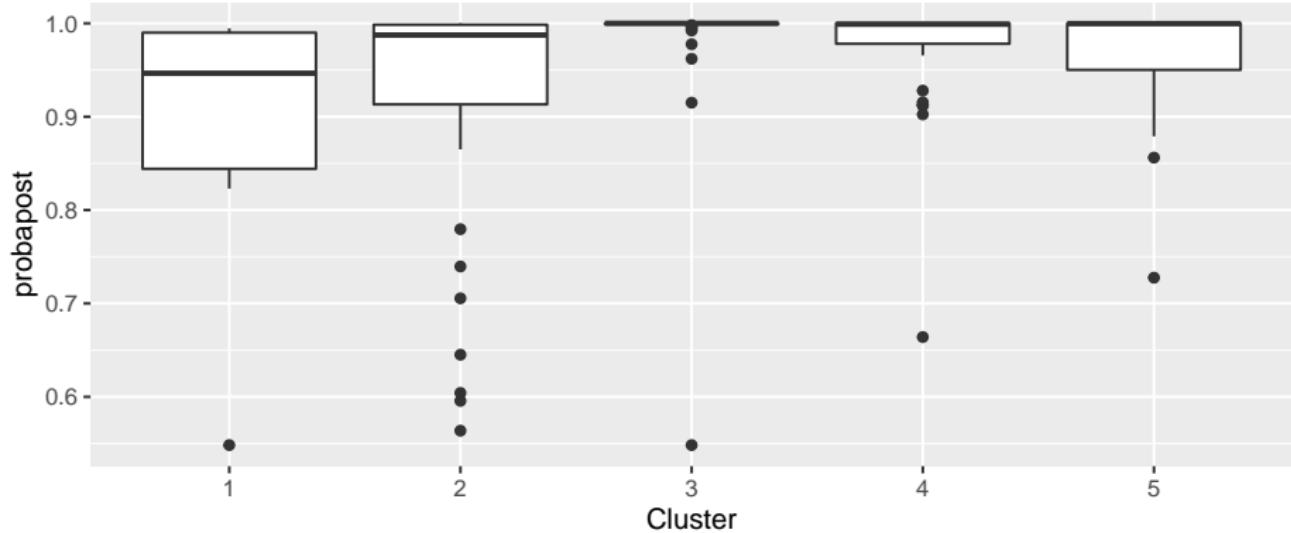
1	2	3	4	5
6	44	49	35	16

# Exemple des iris

```
fviz_mclust(mBIC5,what="uncertainty")
```



# Exemple des iris



### Subsection 3

**En pratique avec R**

# Quelques librairies sous R

- Librairie `mclust` [Scrucca et al.]

*mixtures of multivariate Gaussian*

- Librairie `Rmixmod` [Biernacki et al.]

*mixtures of multivariate Gaussian or multinomial components*

- Librairie `mixture` [McNicholas P.D. et al.]

*Gaussian, Student's t, generalized hyperbolic, variance-gamma or skew-t mixtures*

- Librairie `mvnMF` [Horbik and Grün]

*mixtures of von Mises-Fisher distributions*

- et bien d'autres !