# Unsupervised Classification (Clustering)

## Introduction to Machine Learning

Mathilde Mougeot

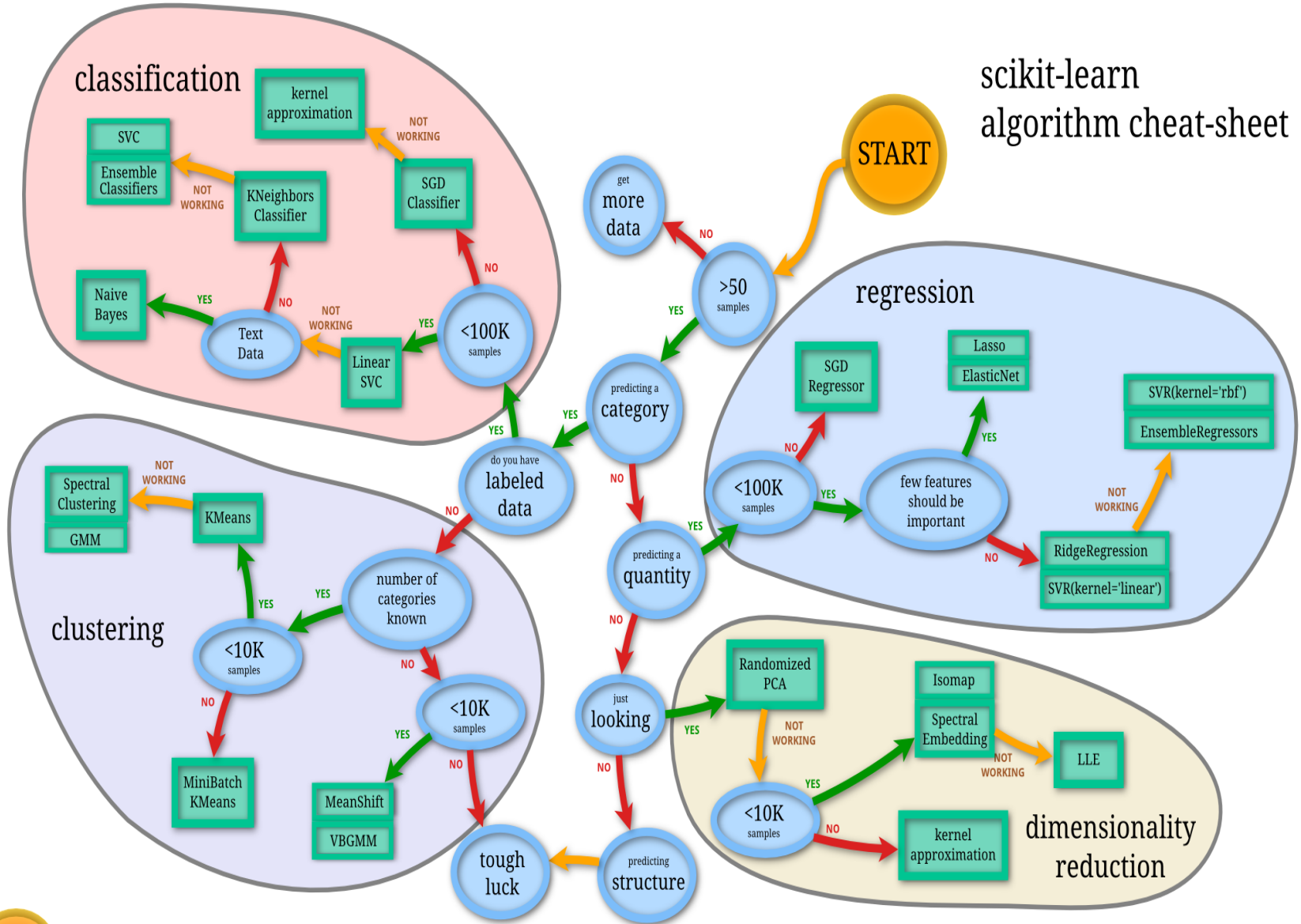
ENSIIE

2020

# At this stage

- Outline
  - Observations $(X_1, \ldots, X_n)$ in $\mathbb{R}^d$ , $d$ may be large
  - Target $(Y_1, \ldots, Y_n)$ labels (classification) or continuous (regression)
  - Goal : Predictive modeling : understand the link between $X \mapsto Y$, reduce $d$.

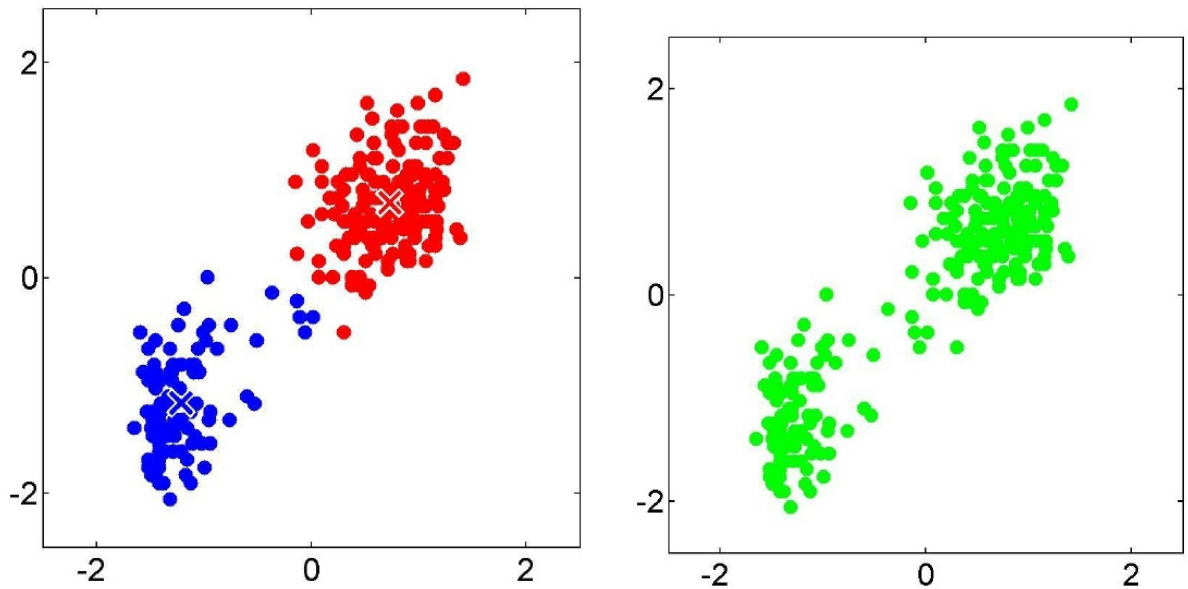
- Methods
  1. **Classification** models (Parametric/ Non Parametric)
     Linear Quadratic Discriminant Analysis, Logistic, KNN CART, Bagging, Random Forest
  2. **Regression** models (Parametric/ Non Parametric)...
     Linear models, Linear models with penalization : LASSO, Ridge, KNN, CaRt, Bagging, Random Forest...
  3. Clustering (unsupervised Classification)


scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
predicting a category
YES
NO
do you have labeled data
YES
NO
<100K samples
YES
Linear SVC
NOT WORKING
Text Data
NO
YES
Naive Bayes
KNeighbors Classifier
NOT WORKING
Ensemble Classifiers
NOT WORKING
SVC
kernel approximation
NOT WORKING
SGD Classifier
NO
classification
predicting a quantity
YES
NO
<100K samples
NO
YES
SGD Regressor
Lasso
ElasticNet
few features should be important
YES
NO
RidgeRegression
SVR(kernel='linear')
NOT WORKING
SVR(kernel='rbf')
EnsembleRegressors
regression
number of categories known
YES
NO
<10K samples
YES
KMeans
NOT WORKING
Spectral Clustering
GMM
<10K samples
NO
MiniBatch KMeans
YES
MeanShift
VBGMM
NO
clustering
just looking
YES
Randomized PCA
NOT WORKING
<10K samples
YES
Isomap
Spectral Embedding
NOT WORKING
LLE
NO
kernel approximation
dimensionality reduction
NO
predicting structure
tough luck
Back

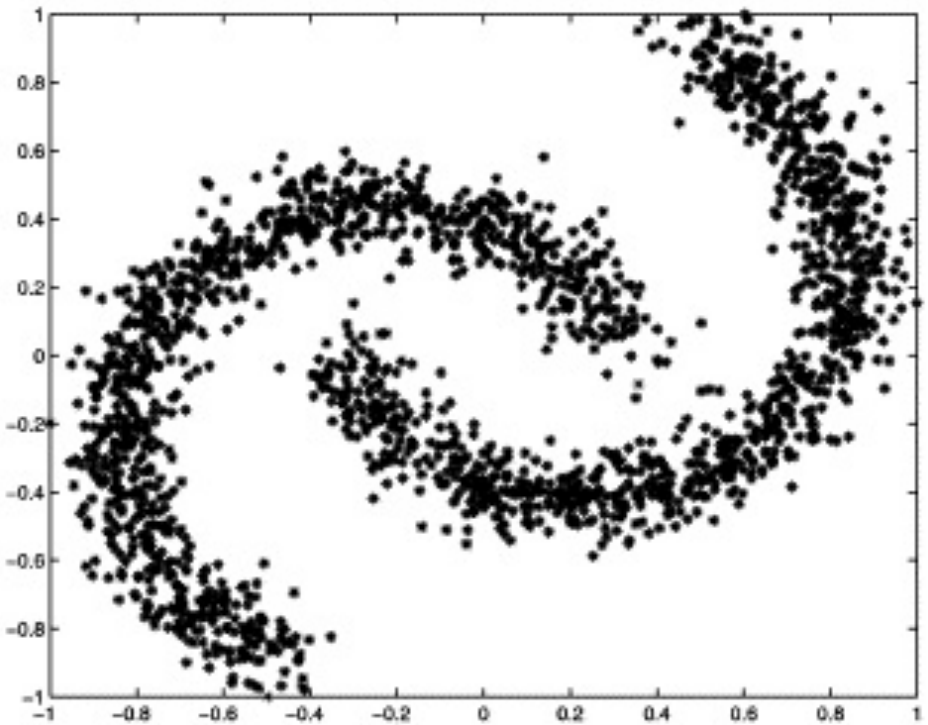
# Outline

# Motivations

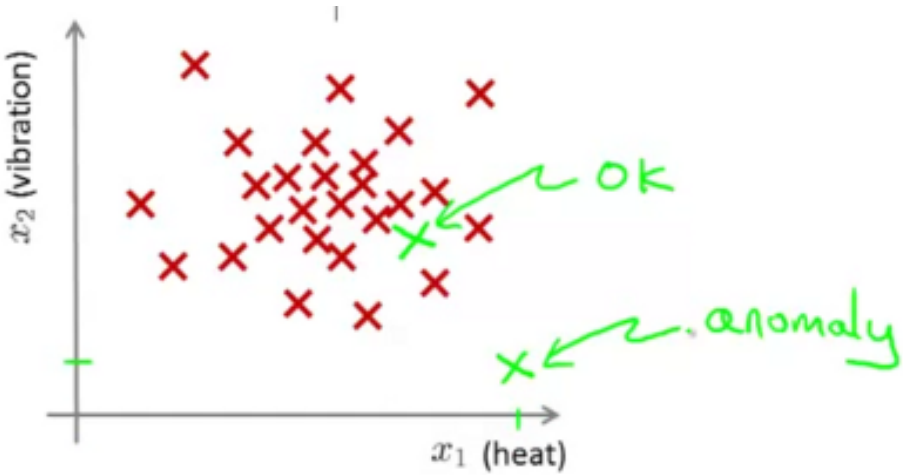
# Unsupervised data (1) - Clustering

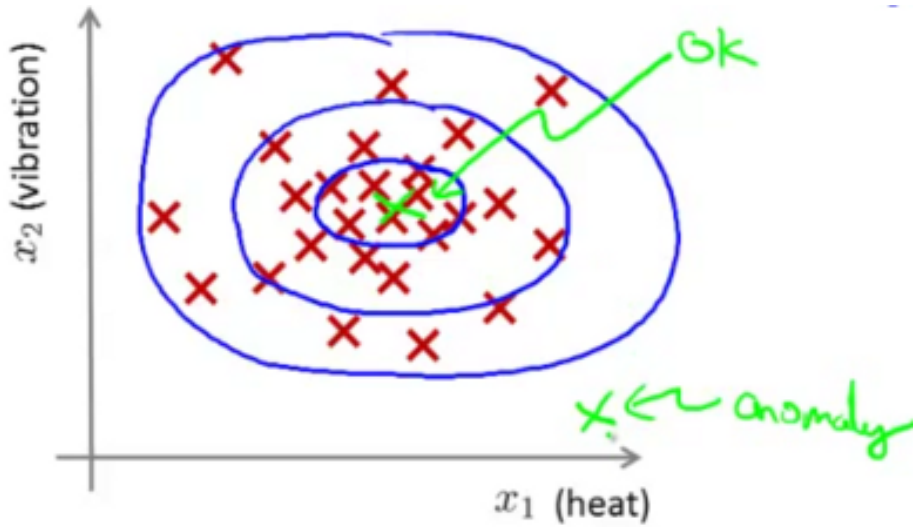


# Clustering can be difficult !

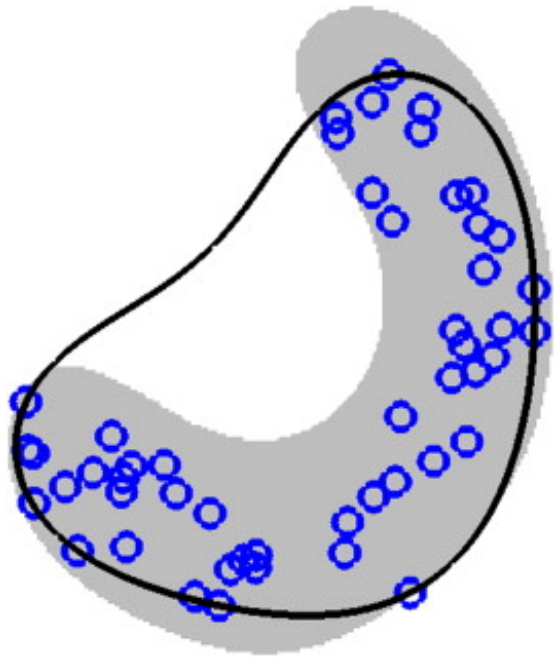


# Unsupervised data (2) - Anomaly/Mode detection



# Unsupervised data (3) - Novelty detection



# Distance-based clustering

-

# Notations

# Clustering input : distance matrix

- Data matrix
  - Individual index $i \in \{1, \ldots, n\}$
  - Feature index $p \in \{1, \ldots, d\}$
  - Measurements $x_{ip}$
- Distance matrix
  - $p$-th feature distance between individuals $i$ and $j = d_p(x_i, x_j)$
  - Distance between individuals $i$ and $j$ :

$$D(x_i, x_j) = \sum_{p=1}^{d} w_p d_p(x_i, x_j)$$

where $w_p$ $p$-th feature importance, $w_p > 0$

# Examples of distances

- Quantitative features
  - Squared distance or absolute difference
  - 1-correlation
- Discrete ordinal variables
  - Equidistance encoding
- Categorical variables
  - Zero-one distance
- What if missing values ?

# Cluster dispersion functions

- Encoder function $C : \{1, \ldots, n\} \mapsto \{1, \ldots, K\}$ (point to cluster)
- Within-cluster dispersion

$$W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} D(x_i, x_j)$$

- Between-cluster dispersion

$$B(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)\neq k} D(x_i, x_j)$$

- Total dispersion : $T = W(C) + B(C)$

# Clustering method #1 - Brute force

- Combinatorial assignment
- Number of possibilities for assigning $n$ points to $K$ clusters

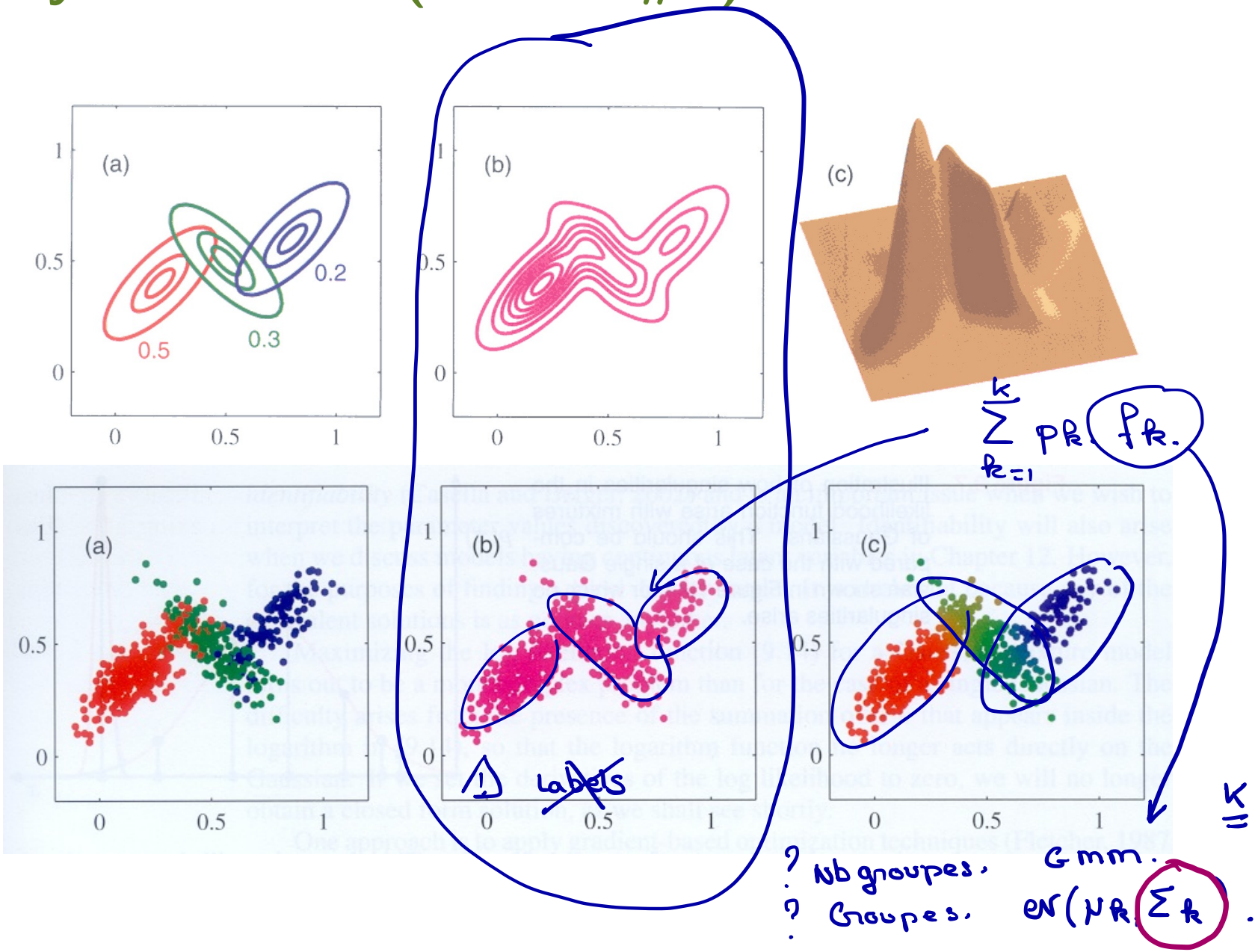
$$S(n, K) = \frac{1}{K!} \sum_{k=1}^{K} (-1)^{K-k} \left( \begin{array}{c} K \\ k \end{array} \right) k^n$$

  Example : $S(19, 4) \simeq 10^{10}$

- Question : Limited search vs. approximate solution

# Parametric approach

# Density estimation (Course #1)



# Reminder on gaussian mixture models

- Random vector $X$ on $\mathbb{R}^d$ with $K$ components
- Gaussian densities $f_k$ , $k = 1, \ldots, K$,
- Component parameters $(\mu_k, \Sigma_k)$,
- Mixture parameter $p = (p_1, \ldots, p_K)$ in the simplex
- Distribution of $X$ = Mixture density

$$f_X(x) = \sum_{k=1}^{K} p_k f_k(x) \quad , \quad \forall x \in \mathbb{R}^d$$

- For estimation, use EM algorithm...

# Complexity of the problem depends on...

- The dimension $d$
- The number of clusters $K$
- The number of samples $n$
- The smallest mixture coefficient "$\min_j p_j$"
- How separated the clusters are...

# High dimension

# Distance between observations

| – | $X^1$ | $X^2$ | ... | $X^j$ | ... | $X^d$ |
|---|---|---|---|---|---|---|
| 1 | $x_{11}$ | | ... | $x_{1j}$ | | $x_{1d}$ |
| 2 | | | | | | |
| ... | | | | | | |
| $\rightarrow i$ | $x_{i1}$ | | ... | $x_{ij}$ | | $x_{id}$ |
| ... | | | | | | |
| $n$ | $x_{n1}$ | | ... | $x_{nj}$ | | $x_{nd}$ |

- For two observations $(x_i, x_k)$, $x_i \in \mathbb{R}^d$, $x_k \in \mathbb{R}^d$

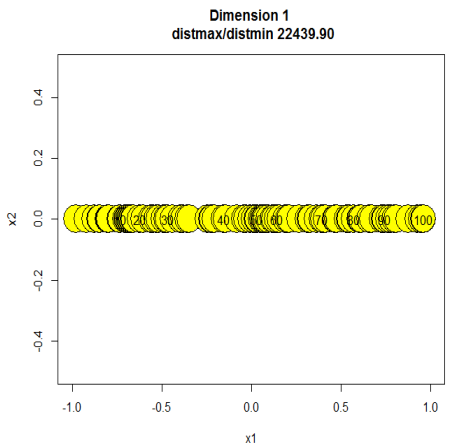
$\rightarrow$ Euclidian distance $\ell_2$ between two observations

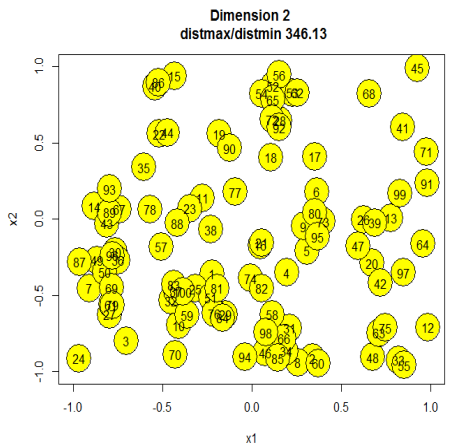
$$||x_i - x_k||_{\ell_2} = \sqrt{\sum_{j=1}^{d}(x_i(j) - x_k(j))^2}$$

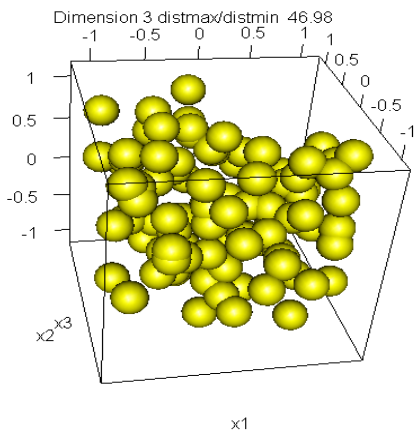
# Dimension curse

- Evaluation of the distance between two observations in dimension $d$
- Illustrations : $n = 100$ observations uniformly distributed, 1, 2, 3, ...
- Indicator : $\frac{max_{i \neq j} ||x_i - x_k||_{\ell_2}}{min_{i \neq j} ||x_i - x_k||_{\ell_2}}$

$d = 1$ | $d = 2$ | $d = 3$



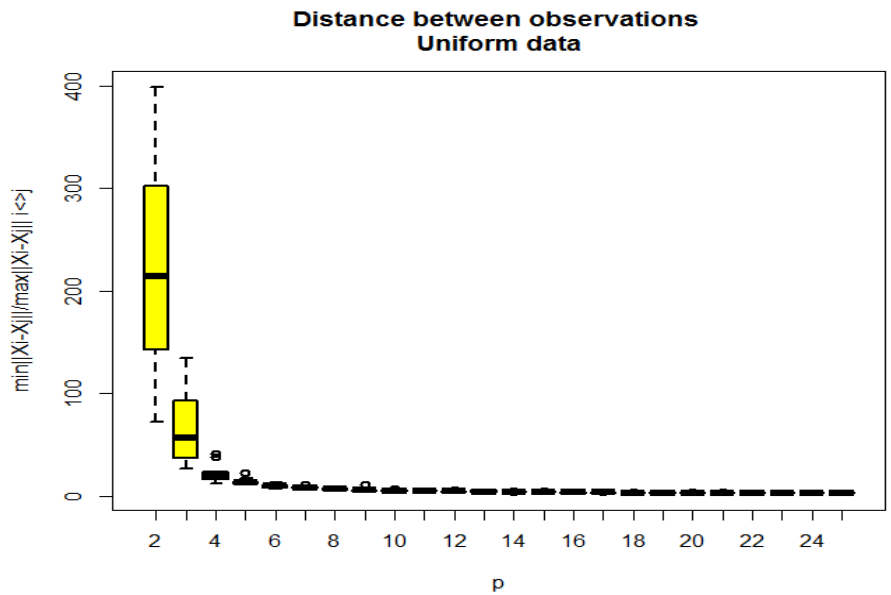




22 435 | 346 | 47

# Dimension curse

Ratio study $\frac{max_{i \neq j} ||x_i - x_j||_{\ell_2}}{min_{i \neq j} ||x_i - x_j||_{\ell_2}}$ function of the dimension $d$

Illustration : $n = 100$ observations uniformly distributed ($K = 100$ repetitions)



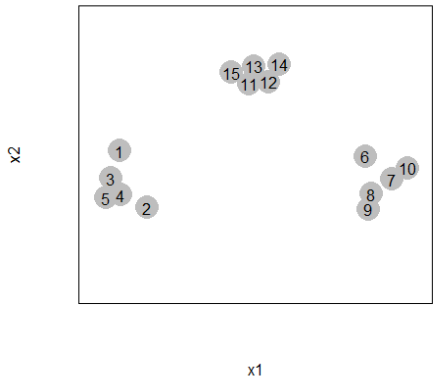

$\rightarrow$ The value of the ratio tends to $\sim 1$ when $d$ increases.

$\rightarrow$ The euclidian distance loses its discrimination ability in high dimension

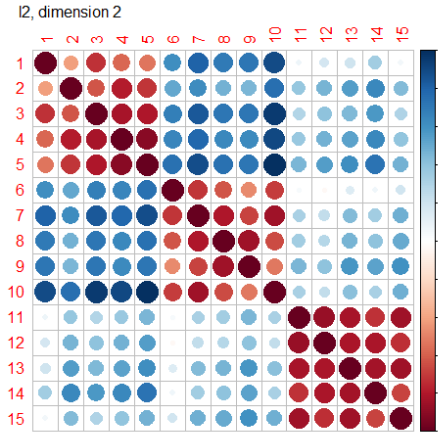
$\rightarrow$ Serious problem especially for segmentation tasks...

# Data segmentation (d=2)

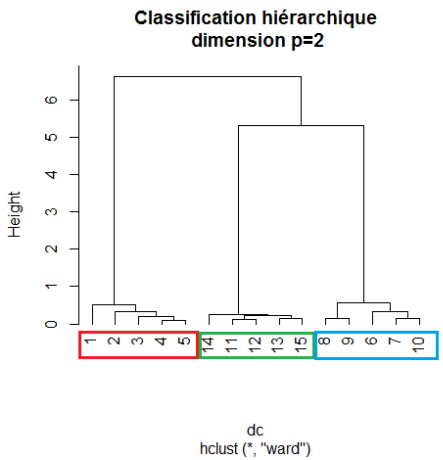
## Observations



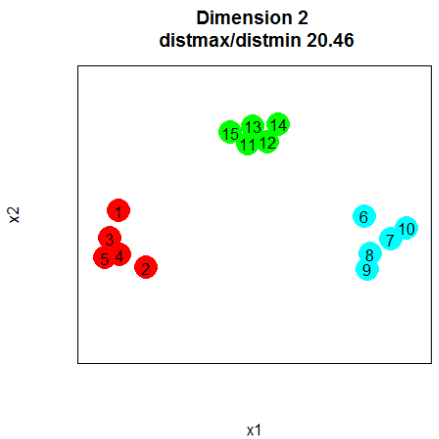

## distance matrix




## HAC



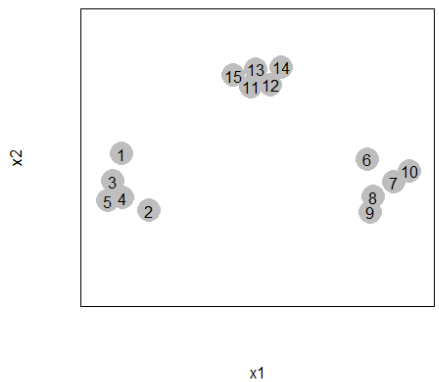

## 3 classes Clustering



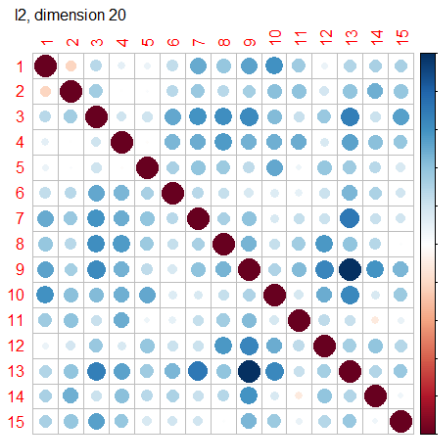
# Data segmentation (d=20)

data are embedded in a high dimensional space $d = 20 = 2 + 18$

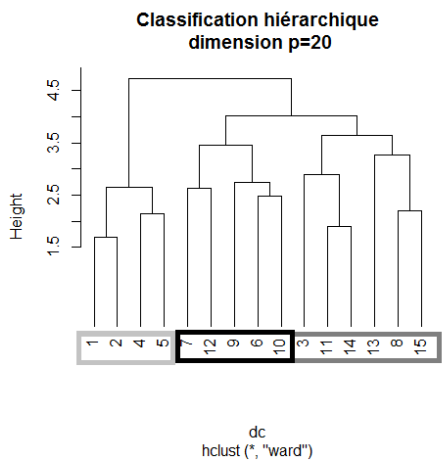
Observations



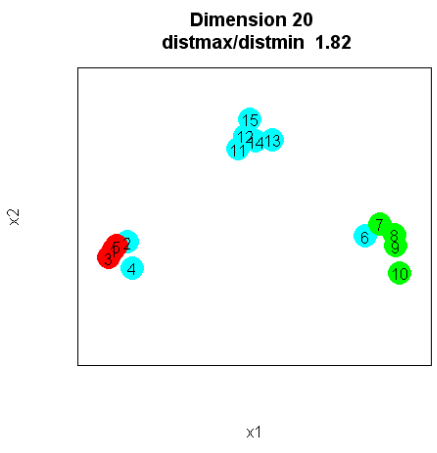

distance matrix




HAC




3 classe Clustering



# Dimension reduction

Find good representations of the data initially coded in large dimensions

- Features : a small number of discriminant features
  based on data expertise or automatic extraction.
- Compress Sensing : sparse representation ($S$) of x based on a linear combinaison of $p$ vectors.
- Manifold estimation : $x$ is represented in a low-dimensional space using the Laplacian eigenvectors on the variety, estimated from a graph of neighborhoods using the examples

$\rightarrow$ Mathematical tools at the interface of harmonic analysis, geometry, probability and statistics.

# Model based Clustering

# Model-based clustering

Set of observations $\{x_i, x_i \in \mathbb{R}^d, 1 \leq i \leq N\}$

**Assumptions :** Mixture of $K$ gaussian : $f_X(x) = \sum_{k=1}^{K} \pi_k f_k(x)$
$\mu_k$ (means), $\Sigma_k$ (covariances), $\pi_k$ (mixing coefficients)
$1 \leq k \leq K$

**Find ? :** $\mu_k$, $\Sigma_k$, $\pi_k$

**using the EM Algorithm :**

1. **Initialization**
2. **E Step :** Expectation Step
3. **M Step :** Maximization Step
4. **LogLikelihood computation**

# EM for gaussian mixture (1/4)

1. **Initialization** :
   $\mu_k$ (means), $\Sigma_k$ (covariances), $\pi_k$ (mixing coefficients)
   Compute Log Likelihood

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\}$$

2. **E Step :**
3. **M Step :**
4. **Evaluate the log likelihood :**

# EM for gaussian mixture (2/4)

1. **Initialization** :$\mu_k$, $\Sigma_k$, $\pi_k$, Compute Log Likelihood
2. **E Step :** Evaluate the responsibilities using current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j, \Sigma_j)}$$

3. **M Step :**
4. **Evaluate the log likelihood :**

# EM for gaussian mixture (3/4)

1. **Initialization** :$\mu_k$, $\Sigma_k$, $\pi_k$, Compute Log Likelihood
2. **E Step :** $\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j,\Sigma_j)}$
3. **M Step :** Re-estimate the parameters using the current responsibilities :
   - $\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$
   - $\sum_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$
   - $\pi_k^{new} = \frac{N_k}{N}$ where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$
4. **Evaluate the log likelihood :**

# EM for gaussian mixture (4/4)

1. **Initialization** :$\mu_k$, $\Sigma_k$, $\pi_k$, Compute Log Likelihood
2. **E Step :** $\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(x_n|\mu_k,\Sigma_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(x_n|\mu_j,\Sigma_j)}$
3. **M Step :** Re-estimate the parameters using the current responsibilities :
   - $\mu_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) x_n$
   - $\sum_k^{new} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk})(x_n - \mu_k^{new})(x_n - \mu_k^{new})^T$
   - $\pi_k^{new} = \frac{N_k}{N}$ where $N_k = \sum_{n=1}^{N} \gamma(z_{nk})$
4. **Evaluate the log likelihood :**

$$\ln p(X|\mu, \Sigma, \pi) = \sum_{n=1}^{N} \ln\{\sum_{k=1}^{K} \pi_k \mathcal{N}(x_n|\mu_k, \Sigma_k)\}$$

REPEAT 2,3,4 UNTIL CONVERGENCE

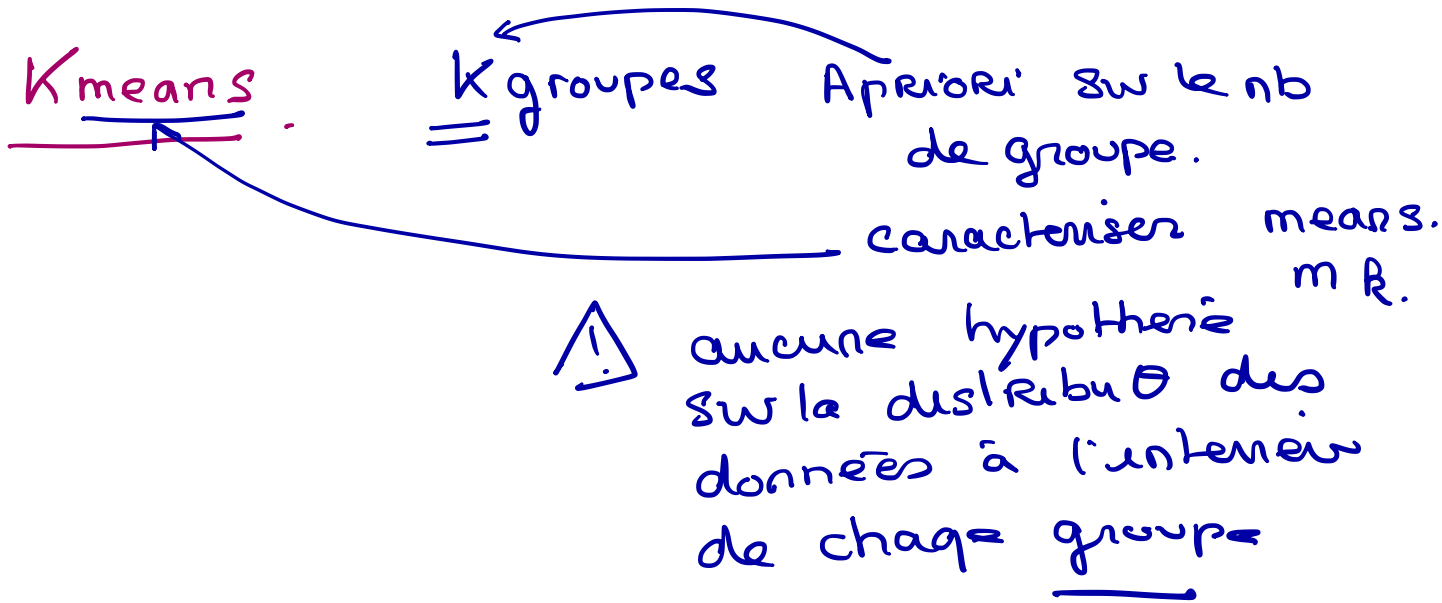
# Model-based clustering

- The mixture density assumption (gaussian) should be valid
- How to chose the number of clusters ? (value of $K$ )
  penalization of the likelihood

# Illustration

Image segmentation based on model-based clustering

# Centroid-based clustering



# The celebrated $K$-means (1)

$\mathcal{D}_n = \{ x_i, \quad x_i \in \mathbb{R}^d, 1 \le i \le n \}$.

- Use for : Quantitative features
- Squared Euclidean distance
- Barycenter cluster $k$ with $n_k$ elements

$$\bar{x}_k = n_k^{-1} \sum_{C(i)=k} x_i$$

$n_k$ : nb d'observations $\in C_k$.

- Note that : for any subset $I \subset \{1, \dots, n\}$ of individuals

$$\bar{x}_I = \arg\min_m \sum_{i \in I} \|x_i - m\|^2$$

$m$.

$\sum \| x_i - m \|^2$

# The celebrated $K$-means (2)

- Optimization criterion

l'ensemble des groupes k.

$$W(C) = \sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2$$

- Solution

$$C^* = \arg\min_{C} W(C) = \arg\min_{C, m_1, \ldots, m_K} \sum_{k=1}^{K} \sum_{C(i)=k} \|x_i - m_k\|^2 + \underline{P(k)}$$

$m_1^*, C_1^*$

$m_2^*, C_2^*$

$\vdots$

$m_K^*, C_K^*$

# Clustering method #3a - $K$-means algorithm

on suppose $K$ connu.

$C_1^{(0)}, C_2 \dots C_K^{(0)}$.

$m_1^{(0)}, m_2^{(0)} \dots m_K^{(0)}$.

**Parameter :** encoder range $K$

**Initialization :** initial encoder $C^{(0)}$, and centers $m_k^{(0)}$

**Step 1 :** Fix encoder $C$, compute the centers

- $$m_k = \frac{1}{n_k} \sum_{C(i)=k} x_i$$

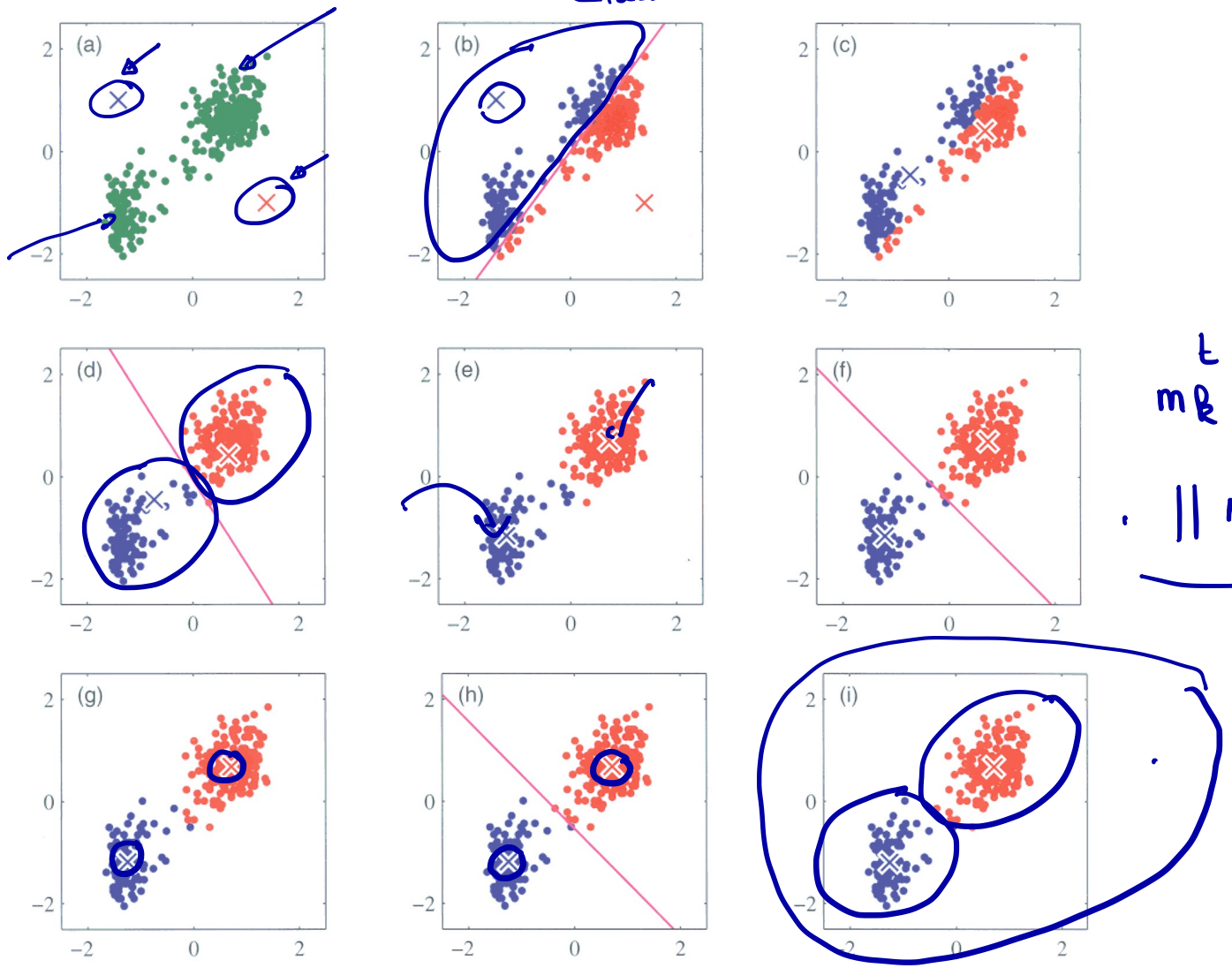
**Step 2 :** Fix the centers $m_1, \dots, m_K$, assign with new encoder

(*) $$C(i) = \arg\min_{1 \le k \le K} \|x_i - m_k\|^2$$

**Iteration :** repeat Steps 1 & 2 until $C$ does not change anymore.

= . stabilité des groupes
. ↑ Nb d'itérations fixées.

# $K$-means algorithm - How it works



# A variation : $K$-medoids

- Use for : any type of features
- Arbitrary distance
- Centers $\{m_1, \ldots, m_K\}$ belong to the data set $\{x_1, \ldots, x_n\}$

# Clustering method #3b - $K$-medoids algorithm

**Parameter :** encoder range $K$

**Initialization :** initial encoder $C^{(0)}$, and centers $m_k^{(0)}$

**Step 1 :** Fix encoder $C$, compute the centers $m_k = x_{i_k^*}$ with

$$i_k^* = \underset{C(i)=k}{\arg\min} \sum_{C(j)=k} D(x_i, x_j)$$

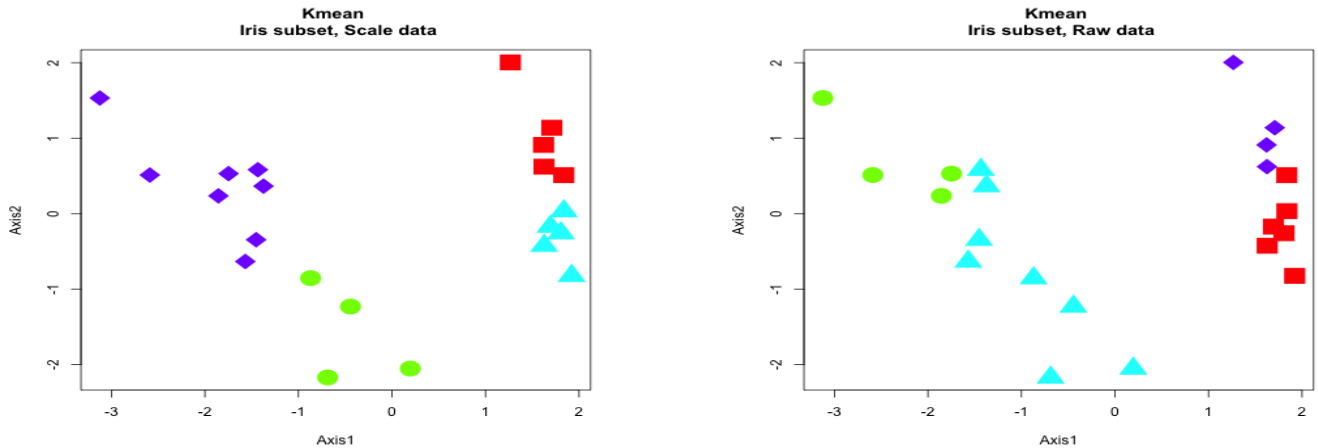
**Step 2 :** Fix the centers $m_1, \ldots, m_K$, assign with new encoder

$$C(i) = \underset{1 \leq k \leq K}{\arg\min}\, D(x_i, m_k)$$

**Iteration :** repeat Steps 1 & 2 until $C$ does not change anymore.

# Kmeans Clustering. Illustration

Use of PCA to project the observations, and to represent the clusters.



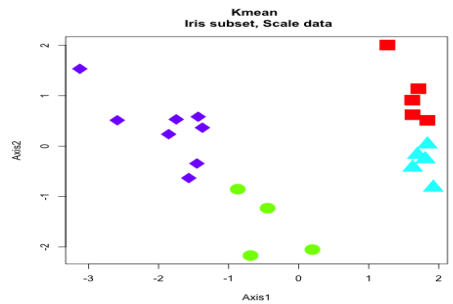

Subset (#30) of Iris Data Set

Impact of Scaling data

(left :scaled vs right :raw)

# Kmeans Clustering. R instructions.

```
# tab : dataframe
pca=dudi.pca(tab,scannf=FALSE,nf=2) ;
K=4 ;
mycol=rainbow(K) ; mypch=c(1,3,4,8)
#Scale data
res=kmeans(tab,centers=4) ;
plot(pca$li,col=mycol[res$cluster],pch=mypch[res$cluster])
```

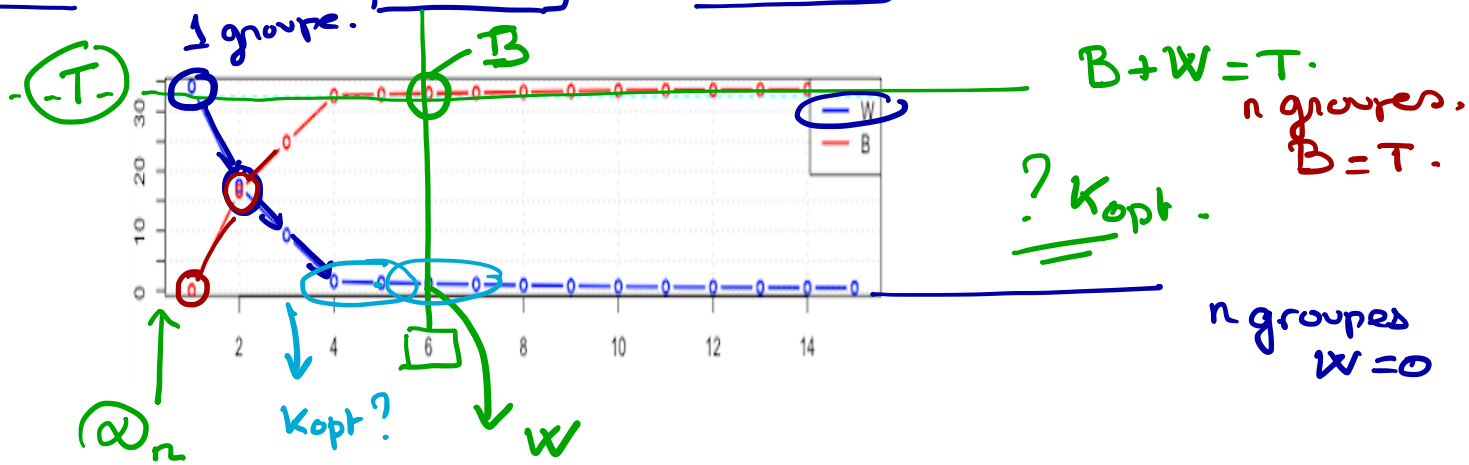


# Calibrating the number of clusters

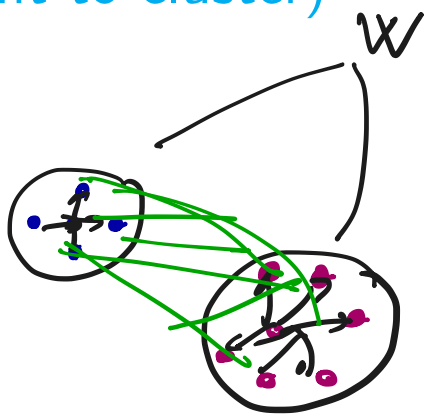
$K$ ?

# Calibrating the number of clusters

Most of the criteria rely on the Within (and Between) cluster dispersion.




- Encoder function $C : \{1, \ldots, n\} \mapsto \{1, \ldots, K\}$ (point to cluster)
- Total dispersion : $T = W(C) + B(C)$
- Within-cluster dispersion :
  $W(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)=k} D(x_i, x_j)$
- Between-cluster dispersion :
  $B(C) = \frac{1}{2} \sum_{k=1}^{K} \sum_{C(i)=k} \sum_{C(j)\neq k} D(x_i, x_j)$



# Calibrating the number of clusters (1)

- Calinski & Harabasz (1974)

$$\text{maximize} \quad F_{CH}(K) = \frac{B(C_K)/(K-1)}{W(C_K)/(n-K)} \ , \quad \forall K > 1 \ .$$

- Hartigan (1975)

$$\text{take smallest } K \geq 1 \text{ such that } \quad F_H(K) \leq 10 \ ,$$

where

$$F_H(K) = \left( \frac{W(C_K)}{W(C_{K+1})} - 1 \right) / (n-K-1) \ .$$

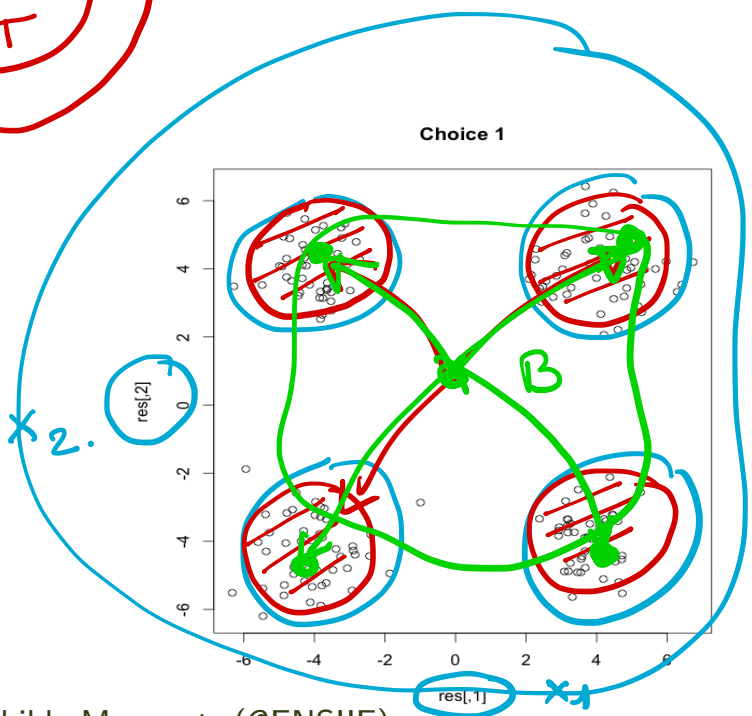
# Calibrating the number of clusters (2)

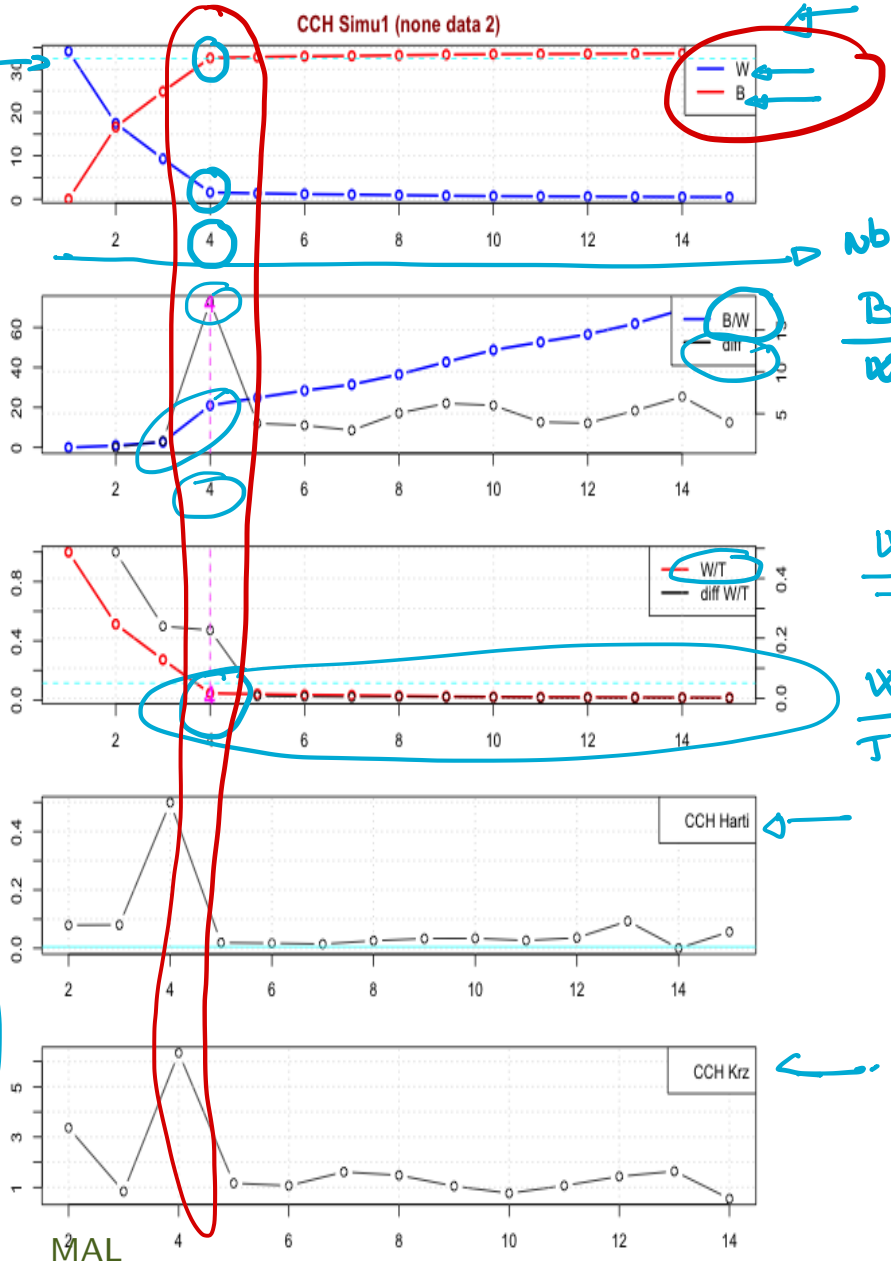
- Krzanowski & Lai (1985)

$$\text{maximize} \quad F_{KL}(K) = \left| \frac{\Delta(K)}{\Delta(K+1)} \right|$$

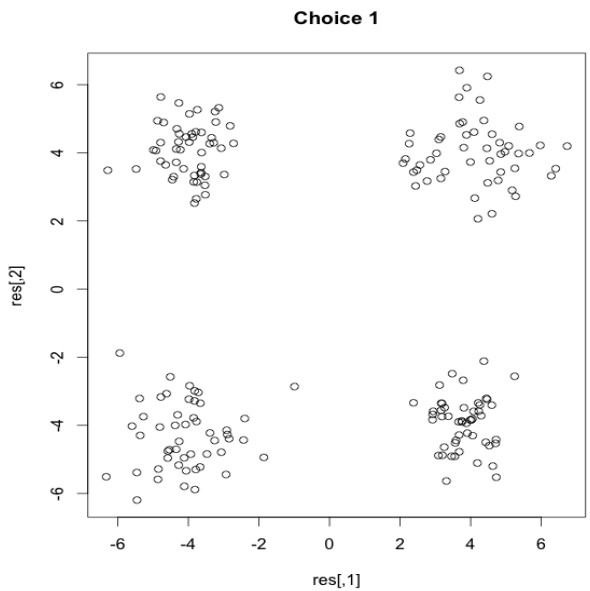
where $\Delta(K) = (K-1)^{2/d} W(C_{K-1}) - K^{2/d} W(C_K)$ and $d$ is the dimension of input data.

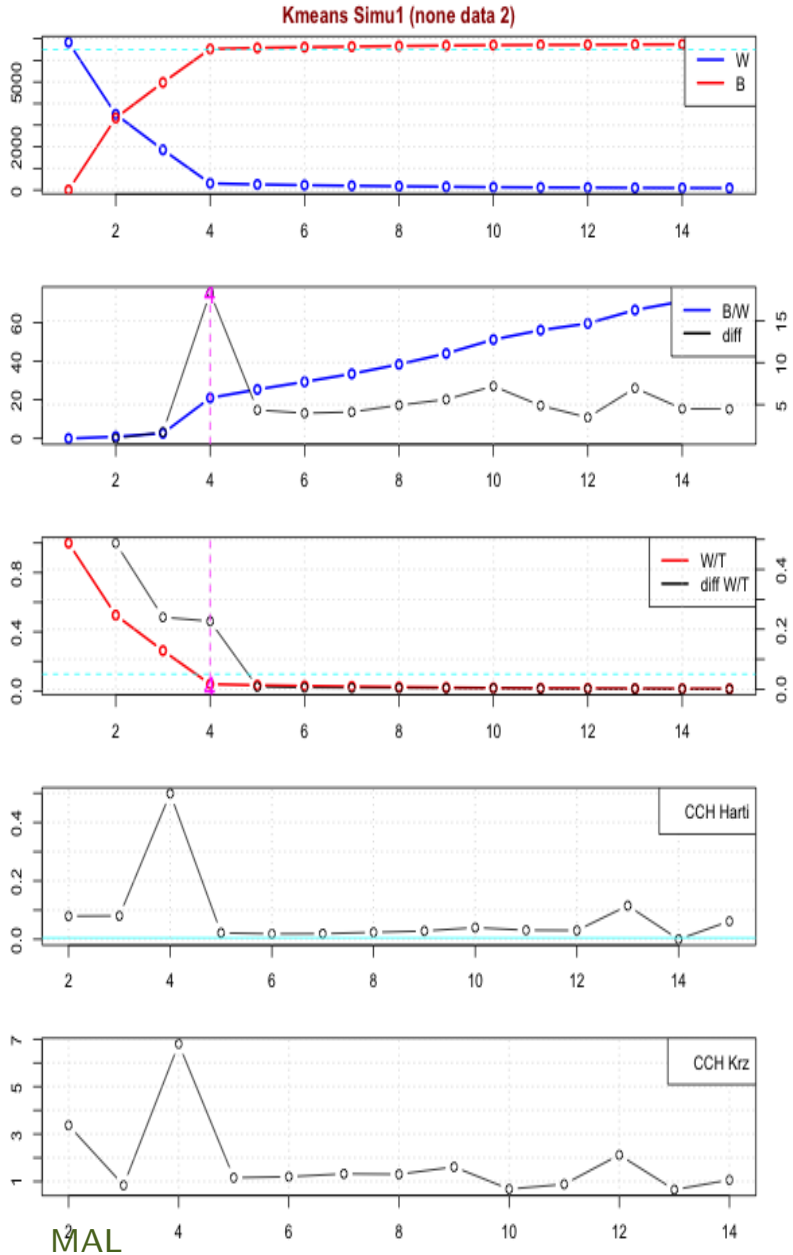
# Illustration. Calibrating the number of clusters



# Illustration. Calibrating the number of clusters



# Calibrating the number of clusters (3)

- Rousseeuw (1987) - Silhouette statistic

indicateur visuel.

$$F_S(K) = \sum_{i=1}^{n} \left( \frac{b(i) - a(i)}{\max(a(i), b(i))} \right) ,$$

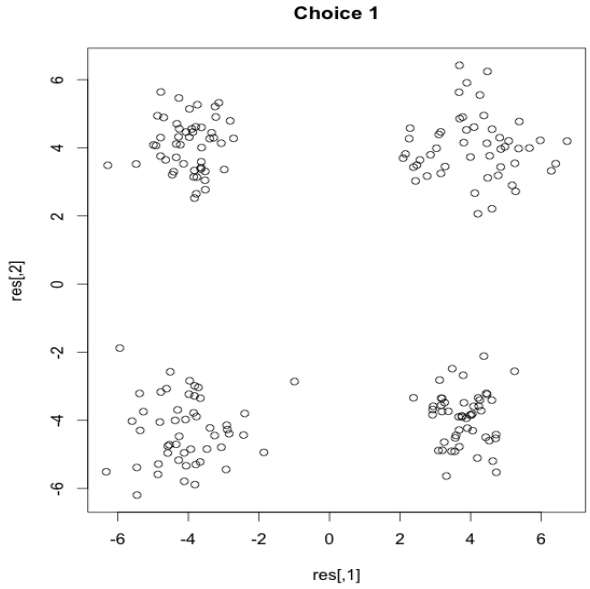
where for $i \in C_k$

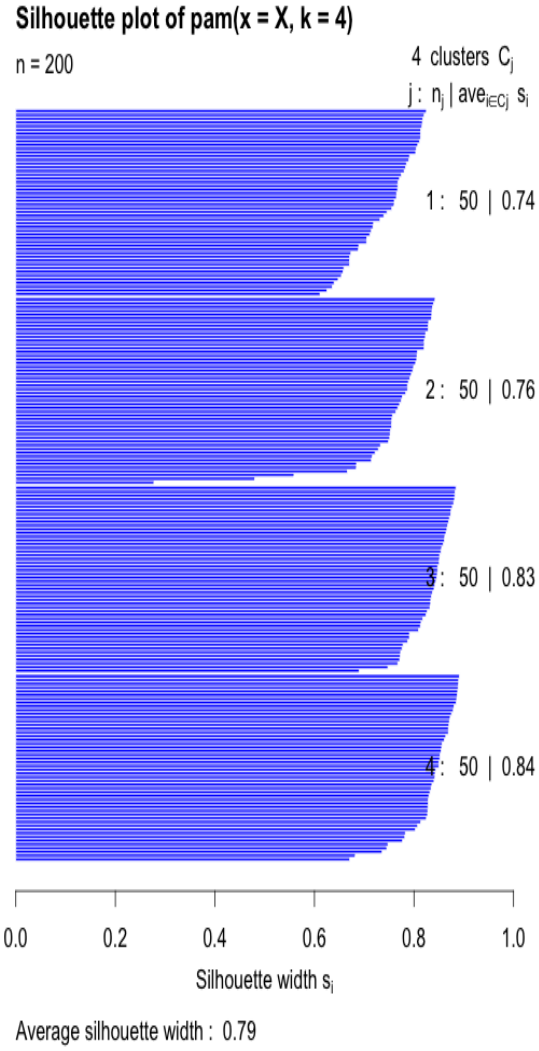
$$a(i) = \frac{1}{n_k} \sum_{C(j)=k} \|x_j - x_i\|_2^2 ,$$

and, if $\ell = \ell(i)$ is the next nearest cluster of the point $x_i$ :

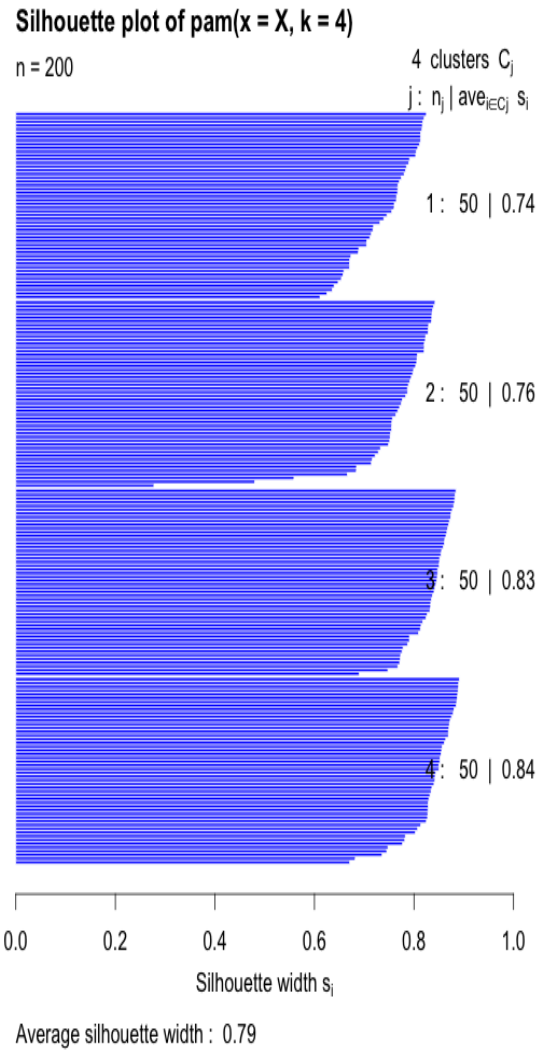
$$b(i) = \frac{1}{n_\ell} \sum_{C(j)=\ell} \|x_j - x_i\|_2^2 .$$

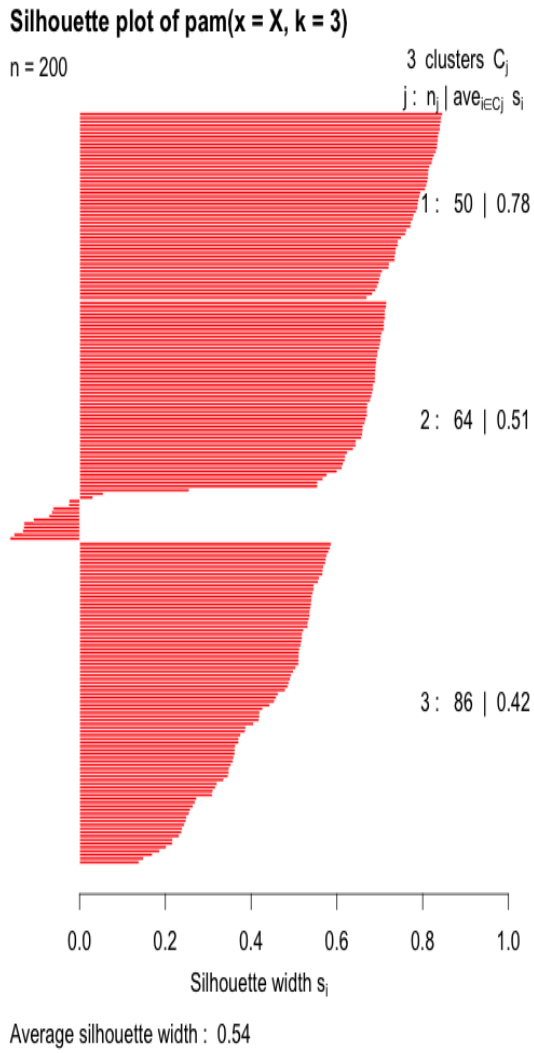
# Calibrating the number of clusters. Illustration.

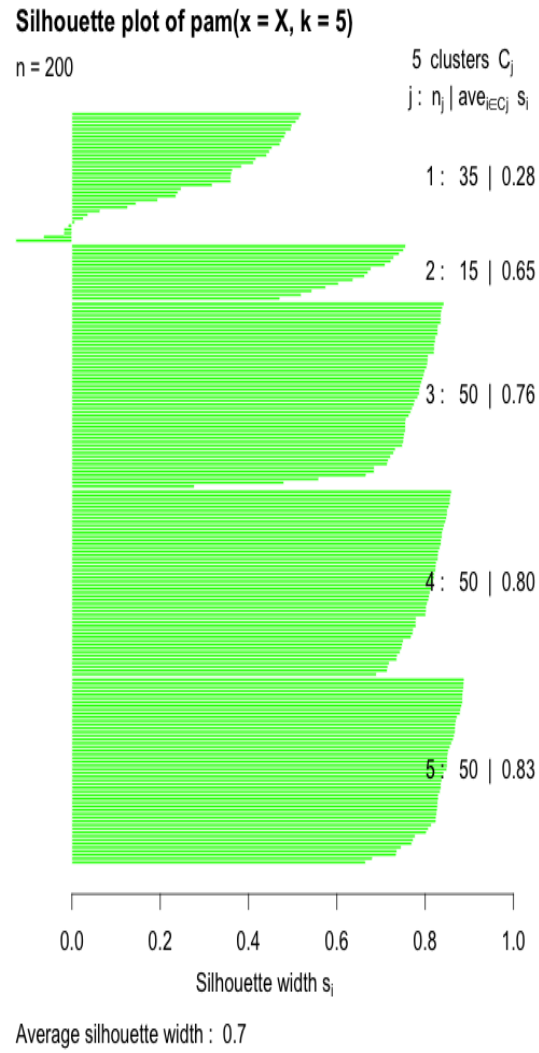


# Calibrating the number of clusters. Illustration.

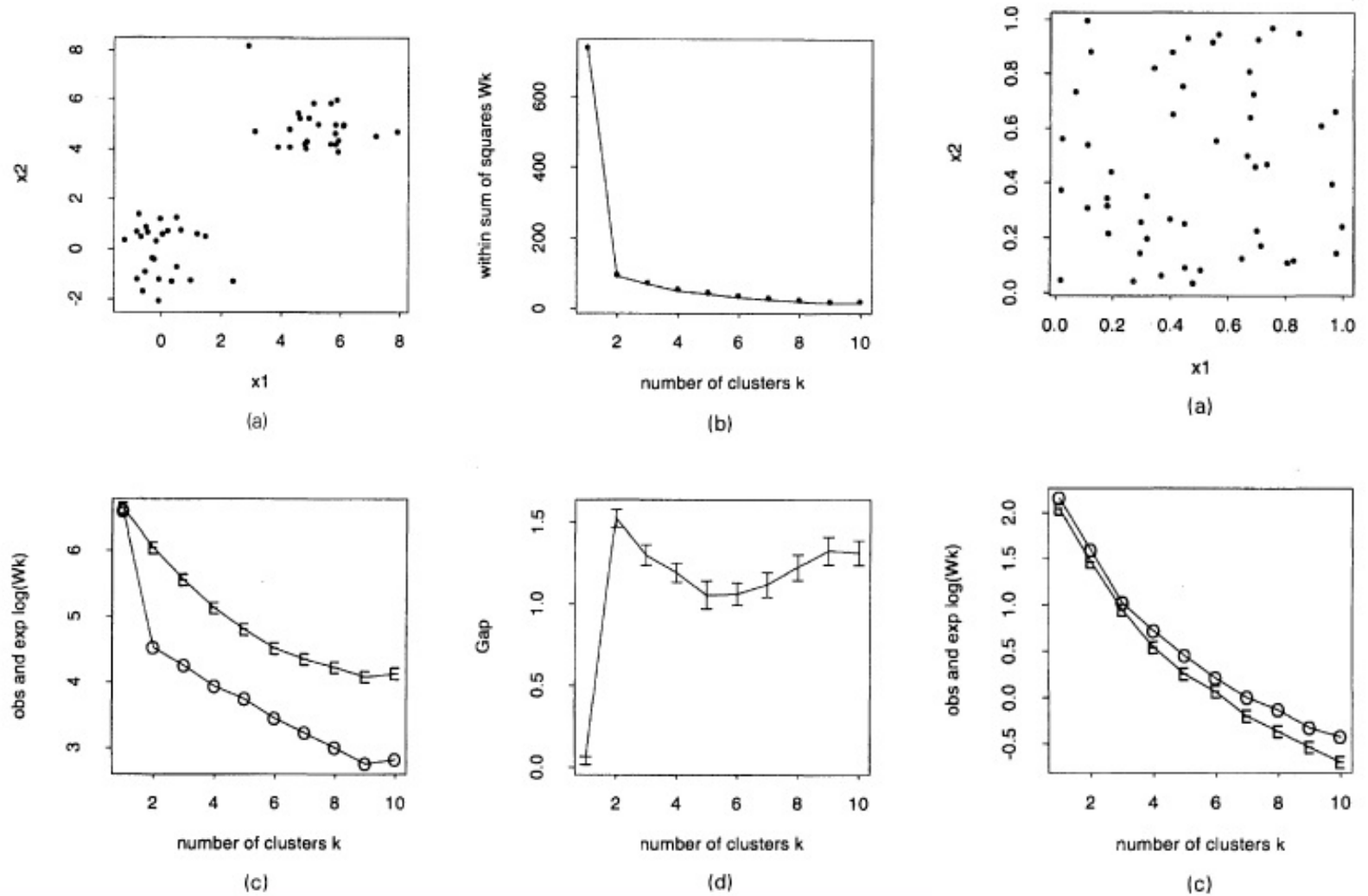


# Calibrating the number of clusters (4)

- Tibshirani, Walther, Hastie (2001) - Gap statistic



# Theoretical analysis

- An information-theoretic formulation
- References on Vector Quantization
- Work of Pollard (1981, 1982), but also Linder (2001)
- Proofs of strong consistency of $K$-means clustering

# K-means clustering vs EM Algorithm

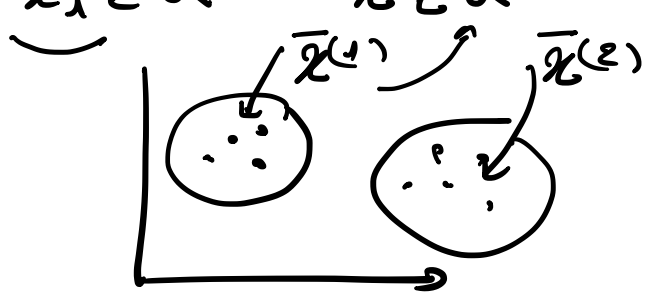
- **Distortion measure :**

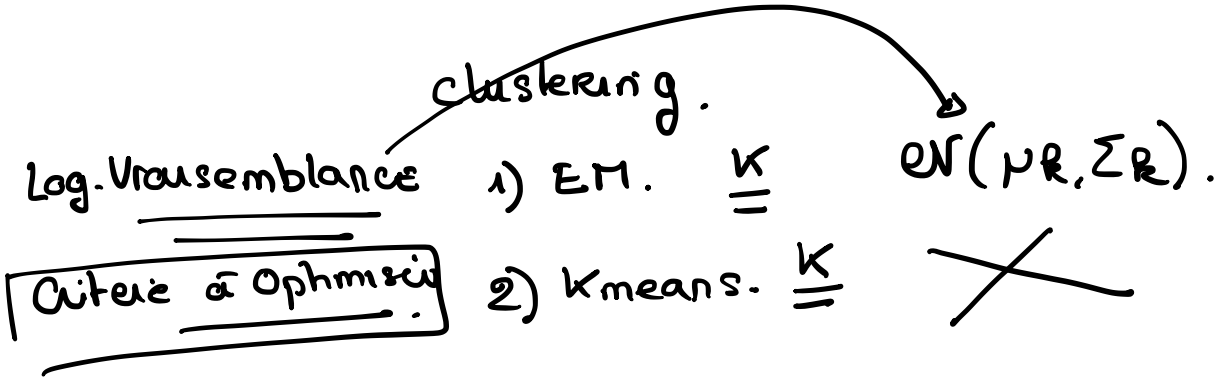
$$J = \sum_{i=1}^{n} \sum_{k=1}^{k} r_{ik} ||x_i - \mu_k||^2$$

$r_{ik} = 1$ if $k = \arg\min_j ||x_i - \mu_j||^2$ (=0 otherwise)

- **K-means algorithm :**
  1. E-Step : $r_{ik}$ computation
  2. M-step : $\mu_k$ computation

REPEAT 1, 2 UNTIL convergence

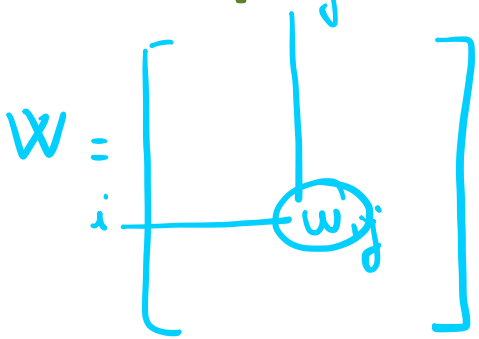


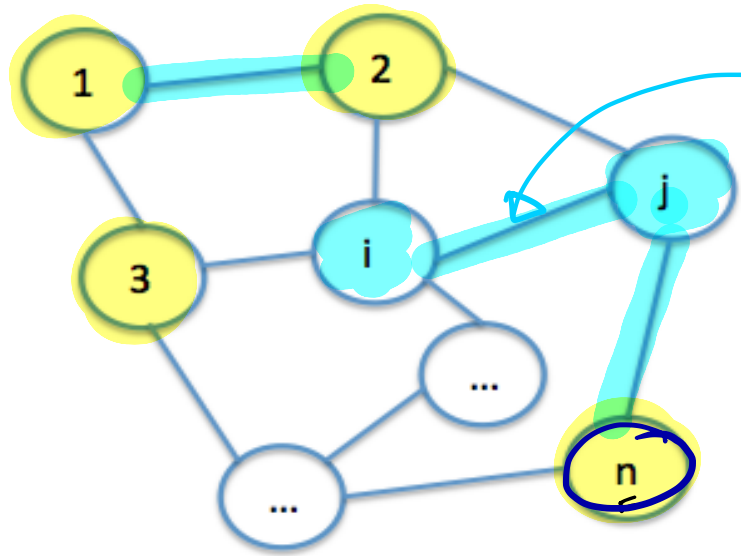



# Graph-based clustering

clustering Spectral

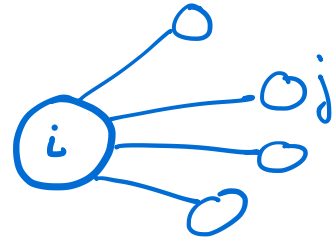
# A Graph



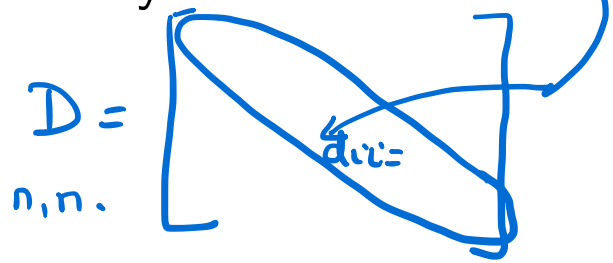

- $G = (V, E)$ is a graph with vertex set $V = \{v_1, \ldots, v_n\}$ and Edges.
- $w_{i,j}$ : positive weight between node i and node j.
  If $w_{ij} = 0$ then vertices $v_i$ and $v_j$ are not connected
- Weighted adjacency matrix $W = (w_{ij})_{1 \leq i,j \leq n}$ with positive coefficients
- Undirected graph means $W$ symmetric : $w_{i,j} = w_{j,i}$.

# Graph definitions

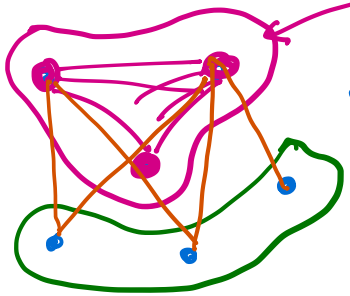



- The degree of a vertex (node) of index $i$ defined by : $\deg_i = \sum_{j=1}^{n} w_{ij}$.
- The Degree matrix is a diagonal matrix defined by

$$D = \mathrm{diag}(\deg_1, \ldots, \deg_n)$$




- For the subset $A \subset \{1, \ldots, n\}$ :



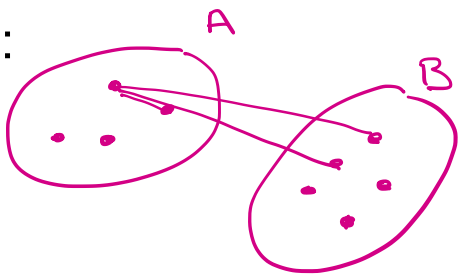
$|A|$ = cardinality of $A$ # noeuds de A.

$\mathrm{vol}(A) = \sum_{i \in A} \deg_i$

- Let $A$ and $B$ two disjoint subsets of nodes of $\{1, \ldots, n\}$

The Mincut distance between $A$ and $B$ is defined by :

$$W(A, B) = \sum_{i \in A, j \in B} w_{ij}$$



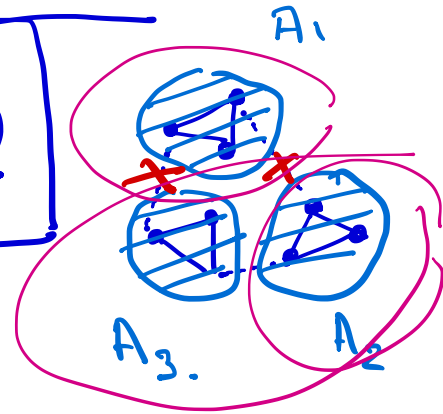
# Graph cut formulation

**In order to understand the underlaying structure of the graph, we are interested in discovering subsets of nodes characterized by edges within group with high weights and edges between groups with low weights and edges.**

- Considering K disjoint subsets, the MinCut criterion is defined by :

$$\text{MinCut}(A_1, \ldots, A_K) = \frac{1}{2} \sum_{i=1}^{K} W(A_i, \overline{A}_i)$$

where $\overline{A}_i = V - A_i$




- Drawback : Often leads to a cluster such that $|A_1| = 1$ if $K = 2$.

# Alternative criteria to MinCut

- To guarantee large enough clusters, other normalized criteria are introduced :

$$\text{RatioCut}(A_1, \ldots, A_K) = \frac{1}{2} \sum_{i=1}^{K} \frac{W(A_i, \overline{A}_i)}{|A_i|}$$

$$\text{NCut}(A_1, \ldots, A_K) = \frac{1}{2} \sum_{i=1}^{K} \frac{W(A_i, \overline{A}_i)}{\text{vol}(A_i)}$$

- Drawback : NP-hard optimization problems

# Spectral clustering

- The Spectral clustering algorithm proposes a relaxation of the RatioCut and Ncut minimizations.
- The computation of the eigenvectors of the Graph Laplacian operator approximate the RatioCut solution.

# Graph Laplacian (1)

Notations :

- $W$ is the adjacency matrix $W = (w_{ij})_{1 \leq i,j \leq n}$ with $w_{ij} \geq 0$ .
- $D = \mathrm{diag}(\deg_1, \ldots, \deg_n)$ is the diagonal degree matrix :, $\deg_i = \sum_{j=1}^{n} w_{ij}$.

- Definition - Unnormalized graph Laplacian matrix :

$$L = D - W$$

- Property 1 - For any vector $f \in \mathbb{R}^n$

$$f^T L f = \frac{1}{2} \sum_{i,j} w_{ij} (f_i - f_j)^2$$

- Property 2 - $L$ symmetric, positive
- Property 3 - Smallest eigenvalue of $L$ is 0
- Property 4 - Relation between the spectrum of $L$ and the number of connected components of the graph.

# Building a similarity graph

In practice :

Input : Similarities $s(x_i, x_j)$ or distances $D(x_i, x_j)$, $\forall i, j$

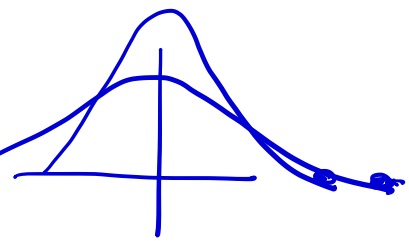
- Following options are often applied on the similarity matrix, and appeared to be very useful based on the (introduction of thresholding or non-linearities)

→ Option 1 - $\epsilon$-neighborhood graph, ($\epsilon$ hyper parameter)

→ Option 2- $k$-nearest neighbor graph, ($k$ hyper parameter)

→ Option 3 - Fully connected graph, ($\sigma$ hyper parameter)

$$s(x_i, x_j) = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$$



# Clustering method - Unnormalized Spectral Clustering Algorithm

**Input :** Similarity matrix $S$. The number $K$ of clusters is given (as Kmeans)

**Preprocessing :** a similarity graph with adjacency matrix $W$ and Compute the unnormalized Laplacian $L$

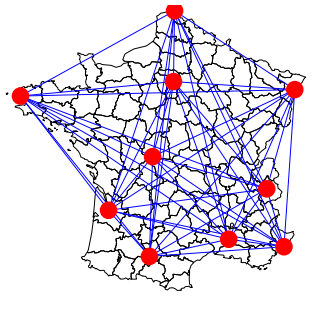
**Solve eigenvalue problem :** compute the first $K$ eigenvectors of $L = D - W$

**Clustering in feature space:** Let $U \in \mathbb{R}^{n \times K}$ be the matrix containing the vectors $u_1, \ldots, u_K$ as columns (eigenvectors with normalisation)

- For $i = 1, \ldots, n$ , let $y_i \in \mathbb{R}^K$ be the vector corresponding to the $i$-th row of $U$
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^K$ with the $K$-means algorithm into clusters encoded by a partition $A_1, \ldots, A_K$

# Spectral clustering

Full connected graph with $n$ nodes.



$\mathcal{D}_n = \{x_i \; x_i \in \mathbb{R}^d\}$

$n = 10$

$z_i$ : feature. lié aux obs $x_i$.

Weight between two nodes $(Z_i, Z_j)$ :

$$w_{i,j} = e^{\frac{-||Z_i - Z_j||_2^2}{2\mu^2}},$$

$\mu$ heat parameter

Normalized Graph Laplacian : $L = D - W$

$$L = I - D^{-1/2} W D^{-1/2}$$

$L' = D^{-1/2} L D^{-1/2}$

$= I -$

$L \in \mathbb{R}^{N \times N}$

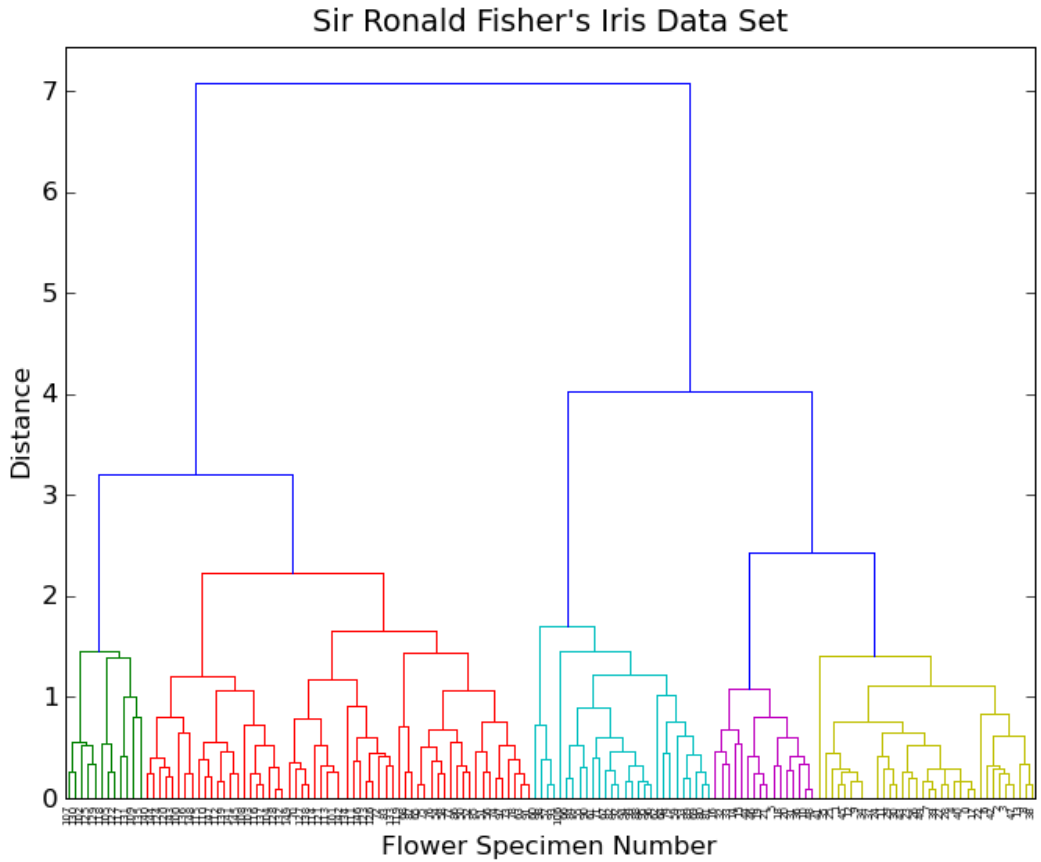
Ng et al. Algorithm (2002) :

**Input :** Fix $k$ nb . clusters

1. Compute the first $k$ eigenvectors $u_1, \ldots u_k$ of $L$ corresponding to the "k" smallest eigenvalues,
2. let $U \in \mathbb{R}^{n \times k}$ be the matrix of column vectors $u_1, \ldots, u_k$
3. Form the matrix $T \in \mathbb{R}^{n \times k}$ $t_{i,j} = u_{i,j} / (\sqrt{\sum_k u_{ik}^2})$. Let $y_i \in R^k$ $i^{th}$ row of T.
4. Cluster $\{y_i\}$, $1 \leq i \leq n$ with the k-means into clusters $C_1, \ldots, C_k$

# Hierarchical clustering

# Dendrogram


Sir Ronald Fisher's Iris Data Set
Distance
0
1
2
3
4
5
6
7
Flower Specimen Number

# Clustering method #2a - Bottom-up heuristic

**Input :** number $K$ of clusters, distance matrix $D(x_i, x_j)$, cluster distance $\Delta$

**Initial step :** find the pair $(x_i, x_j)$ the minimal element in the distance matrix and form cluster $A_1 = \{i, j\}$, the remaining $x_k$'s form clusters with one element $A_2, \ldots, A_{n-1}$

**Agglomeration step :** Consider $A_1, \ldots, A_{n-1}$ clusters and find the pair $(k^*, \ell^*)$ such that

$$(k^*, l^*) = \underset{k \neq l}{\arg\min}\, \Delta(A_k, A_l)$$

and merge these clusters into $A = A_{k^*} \cup A_{l^*}$.

**Stopping criterion :** Iterate until the target number of clusters is reached.

# Linkage distance

- $A, B \subset \{1, \ldots, n\}$
- Single linkage

$$\Delta(A, B) = \min_{i \in A, j \in B} \{D(x_i, x_j)\}$$

- Complete linkage

$$\Delta(A, B) = \max_{i \in A, j \in B} \{D(x_i, x_j)\}$$

- Centroid linkage

$$\Delta(A, B) = D(\overline{x}_A, \overline{x}_B)$$

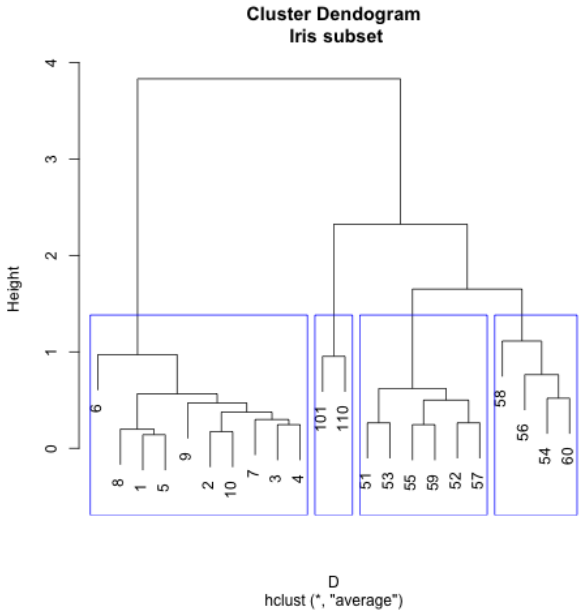
- Average linkage

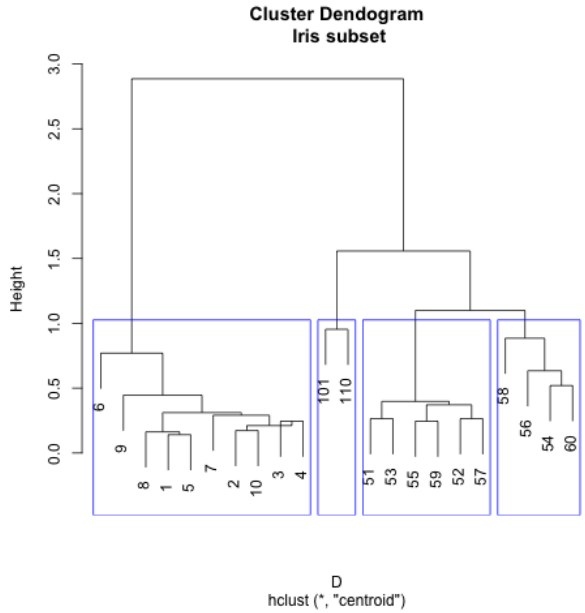
$$\Delta(A, B) = \frac{1}{|A| \cdot |B|} \sum_{i \in A} \sum_{j \in B} D(x_i, x_j)$$

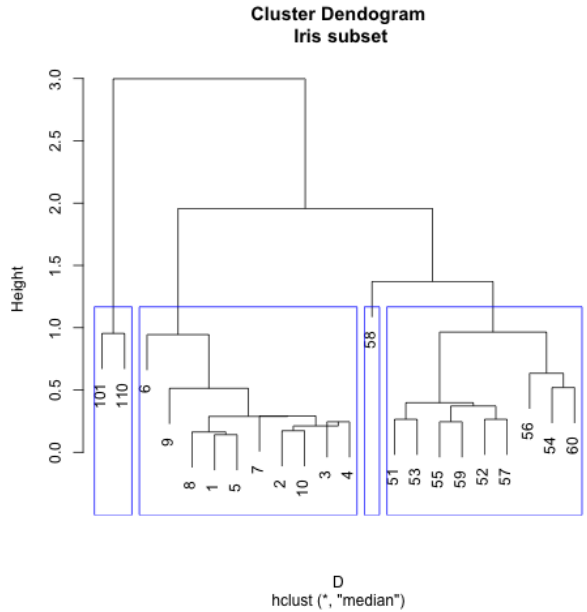
- Ward linkage

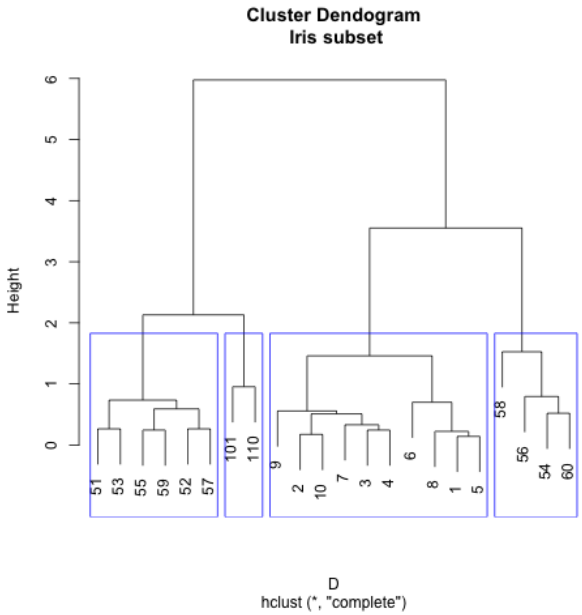
$$\Delta(A, B) = \frac{|A| + |B|}{|A| \cdot |B|} D(\overline{x}_A, \overline{x}_B)$$

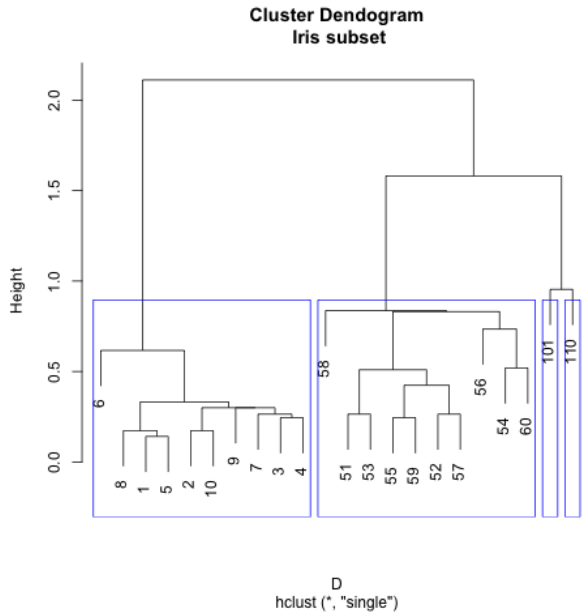
# Hierachical Clustering. Impact of Linkage. Illustration

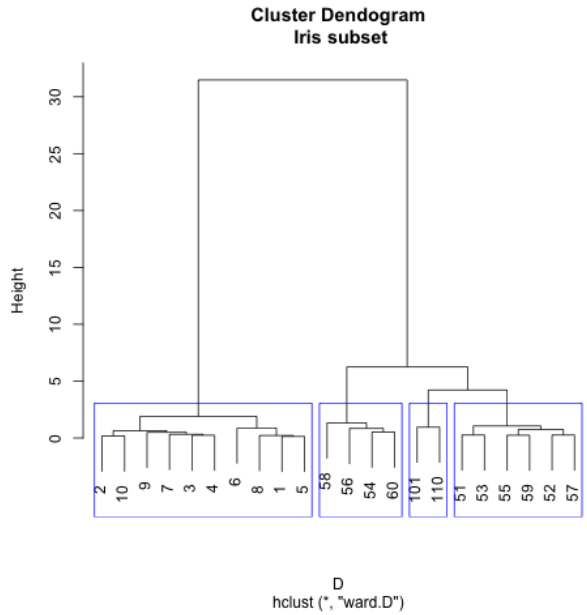


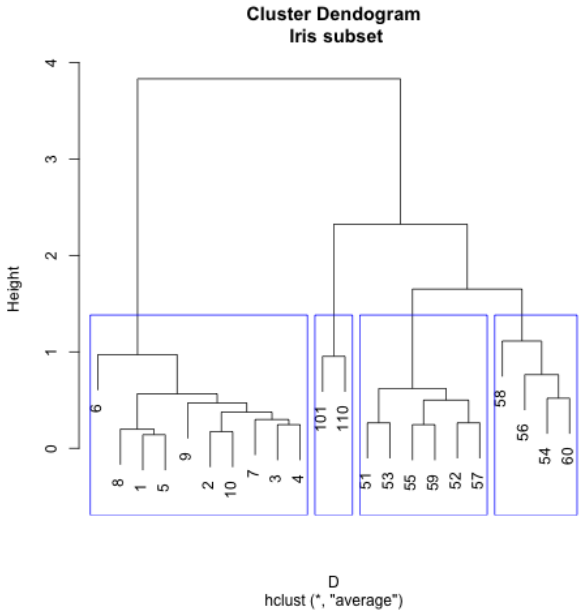



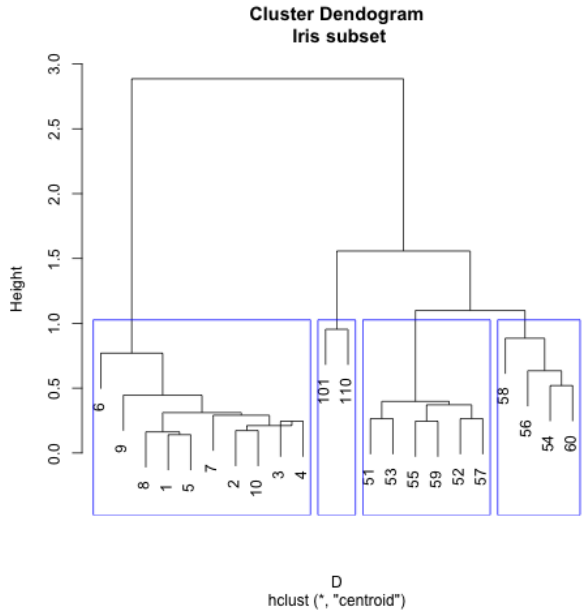



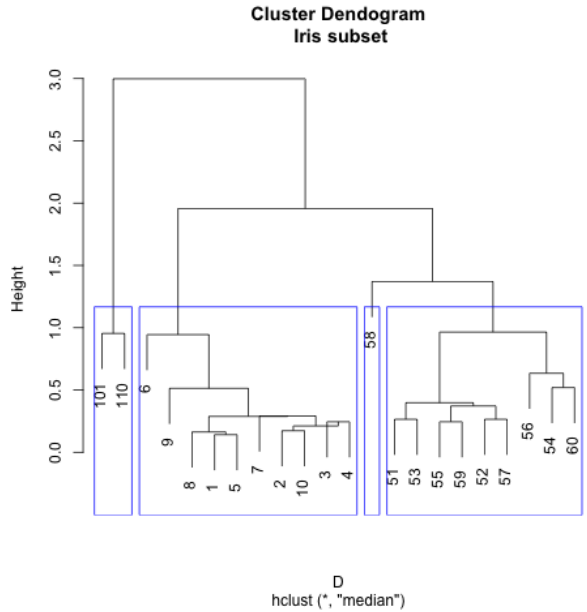



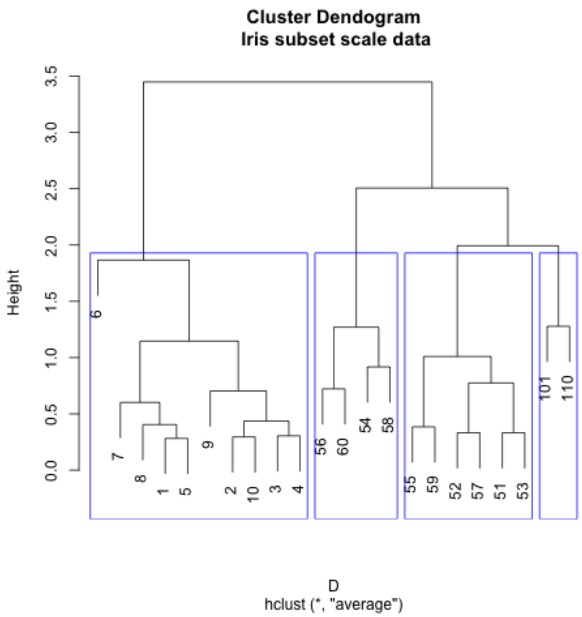



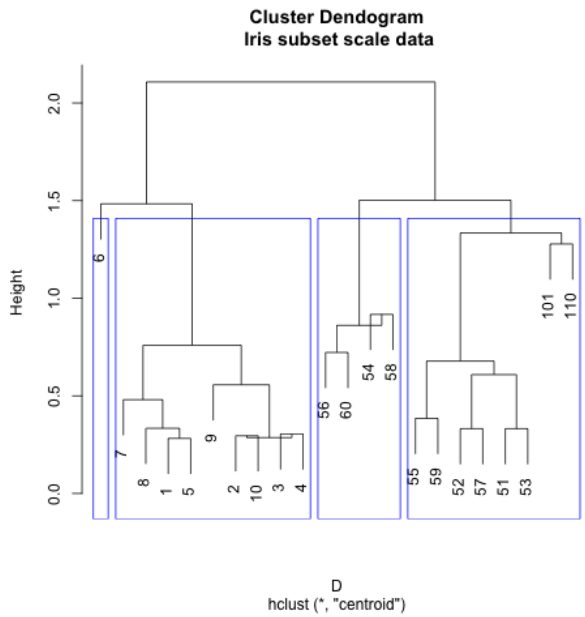
# Hierachical Clustering. Impact of Scaling. Illustration

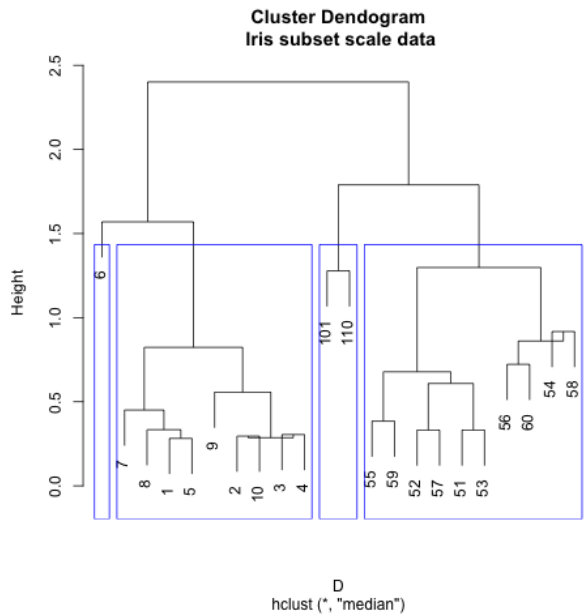


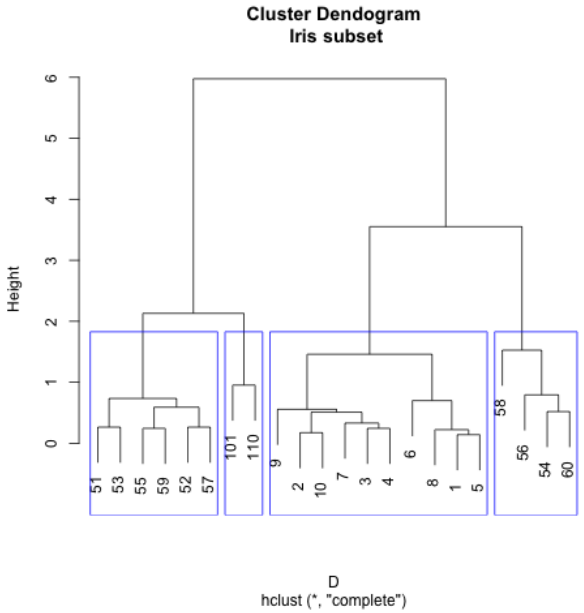



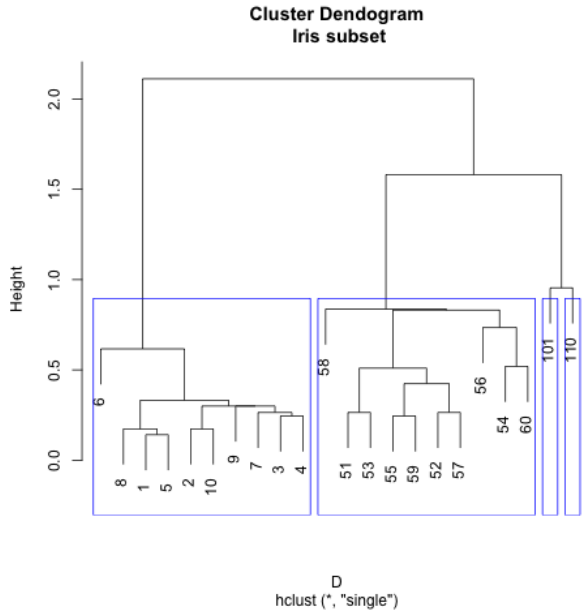



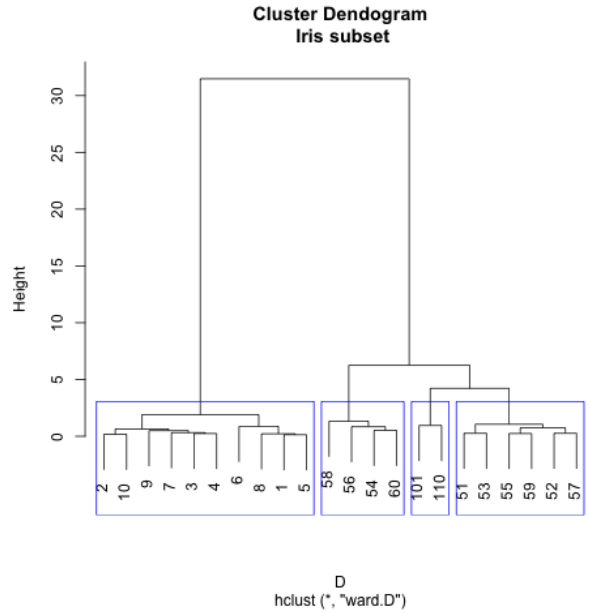



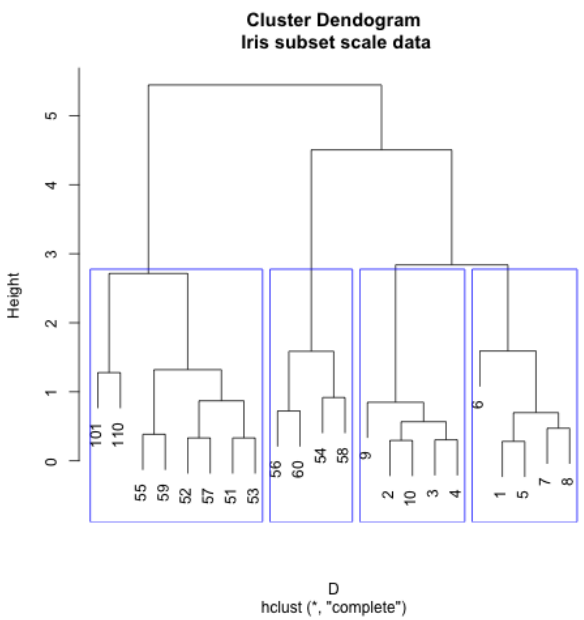



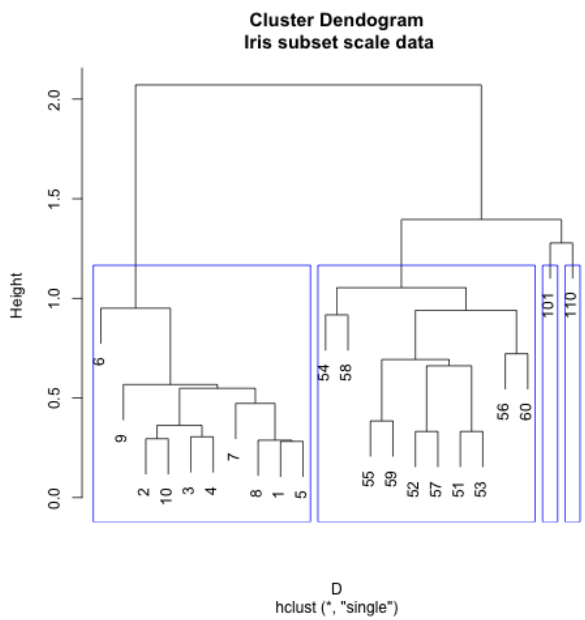
# Hierachical Clustering. Impact of Scaling. Illustration

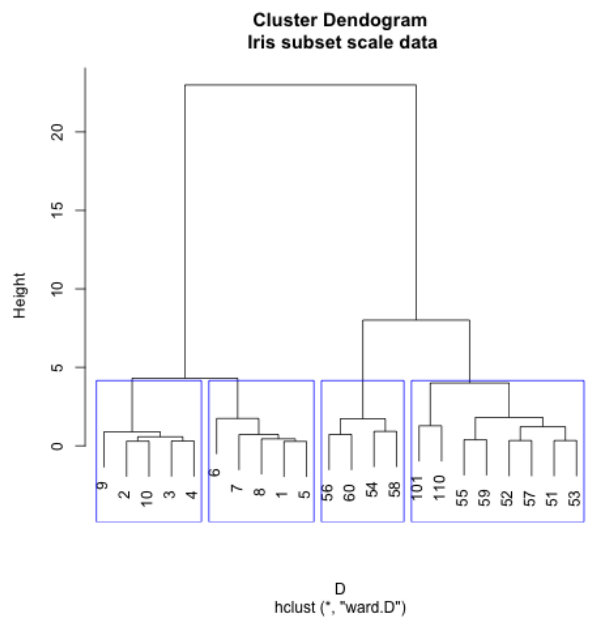


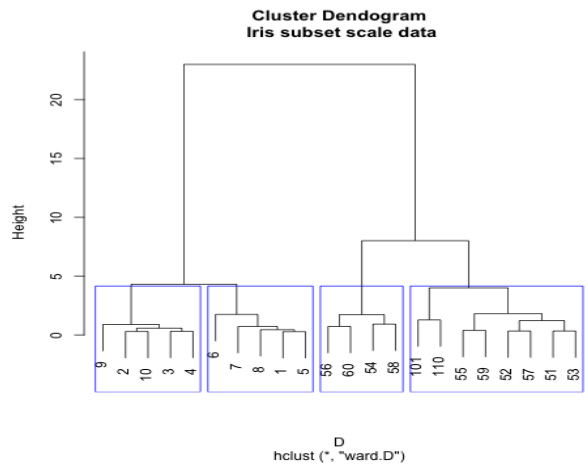



# Hierachical Clustering. R instructions.

```
# Tab : dataframe
help(Hclust) ; D=dist(tab) ;
HC=hclust(D,method="ward.D") ;
plot(HC) ;
rect.hclust(HC, k = 4) ;
members <- cutree(HC, k = 4) ;
```



# Clustering method #2b -Top-down heuristic

**Parameters :** Threshold $t$

**Initial step :** find the pair $(x_i, x_j)$ having the maximal element in the distance matrix denoted $d_M$

**Division step :** if $d_M > t$ consider each of them as a center and affect remaining points to the closest center

**Iteration :** Iterate until $d_M < t$ within each cluster.

# What about theory ?

- Pessimistic result (Kleinberg, 2003)

  There is no clustering function satisfying scale invariance, richness and consistency.

- Optimistic result (von Luxburg, Ben-David, 2005)

  Some stability for clustering can be guaranteed.