

Clustering

Hierarchical clustering et Kmeans

Anne Badel, Frédéric Guyon & Jacques van Helden

2020-03-10

Contents

Les données	3
Les données dans l'ordinateur (1)	4
Les iris de Fisher	4
Les données dans l'ordinateur (2)	4
Les données dans l'ordinateur (2)	4
Représentons ces données : une fleur (1)	5
Représentons ces données : une fleur (2)	5
Représentons ces données : une fleur (3)	6
Représentons ces données : toutes les fleurs (4)	6
Représentons ces données : une variable à la fois (1)	7
Représentons ces données : deux variables à la fois (2)	7
Il faut tenir compte de toutes les dimensions	7
Clustering et classification (termes anglais)	8
Clustering et classification (termes anglais)	8
Clustering	9
Méthodes	9
Géométrie et distances (1)	9
Géométrie et distances (2)	9
Géométrie et distances (3)	9
Distances	10
Distances utilisées dans R (1)	10
Distances utilisées dans R (2)	10
Autres distances non géométriques (pour information)	10
Distances plus classiques en génomique	11

Avec R (1)	11
Avec R (2)	11
Distances entre groupes (1)	12
Distances entre groupes (2)	12
Distances entre groupes (3)	12
Distances entre groupes (4)	13
Les données	13
Visualisation des données	14
Visualisation des données - coloration par espèces	14
Visualisation des données	15
Nettoyage des données (1) : données manquantes	16
Nettoyage des données (2) : variables constantes	16
Normalisation	16
On peut visuellement regarder l'effet de la normalisation :	16
par un plot des données	16
... par une boîte à moustaches (boxplot)	17
... par une image	17
La matrice de distances	18
La classification hiérarchique : principe	18
Notion importante, cf distances	19
L'algorithme : étape 1	19
au départ	19
identification des individus les plus proches	20
construction du dendrogramme	20
étape j :	20
calcul des nouveaux représentants 'BE' et 'CD'	21
calcul des distances de l'individu restant 'A' aux points moyens	21
A est plus proche de ...	22
dendrogramme	22
pour finir	23

dendrogramme final	23
Je ne fais pas attention à ce que je fais ...	23
En utilisant une autre métrique	25
En utilisant un autre critère d'aggrégation	26
En conclusion	26
Les heatmap	26
Les k-means	27
L'algorithme	27
étape 1 :	27
Choix des centres provisoires	28
Calcul des distances aux centres provisoires	28
Affectation à un cluster	29
Calcul des nouveaux centres de classes	29
Etape j :	29
Fin :	29
Arrêt :	29
Un premier k-means en 5 groupes	30
Comment déterminer le nombre de clusters ? (1)	30
Comment déterminer le nombre de clusters ? (2)	31
Comment déterminer le nombre de clusters ? avec la classification hiérarchique	31
Comment déterminer le nombre de clusters ? avec les kmeans	32
Comparaison de clustering: Rand Index	32
Comparaison de clustering: Adjusted Rand Index	32
Comparaison des résultats des deux clustering	32
Comparaison de clustering: Rand Index	33
Comparaison de clustering: Adjusted Rand Index	33
Comparaison des résultats des deux classifications	33
Pros et cons des différents algorithmes	33
... par une projection sur une ACP	34
Cas d'étude : TCGA Breast Invasive Cancer (BIC)	35
TP : analyse de données d'expression	35

Les données

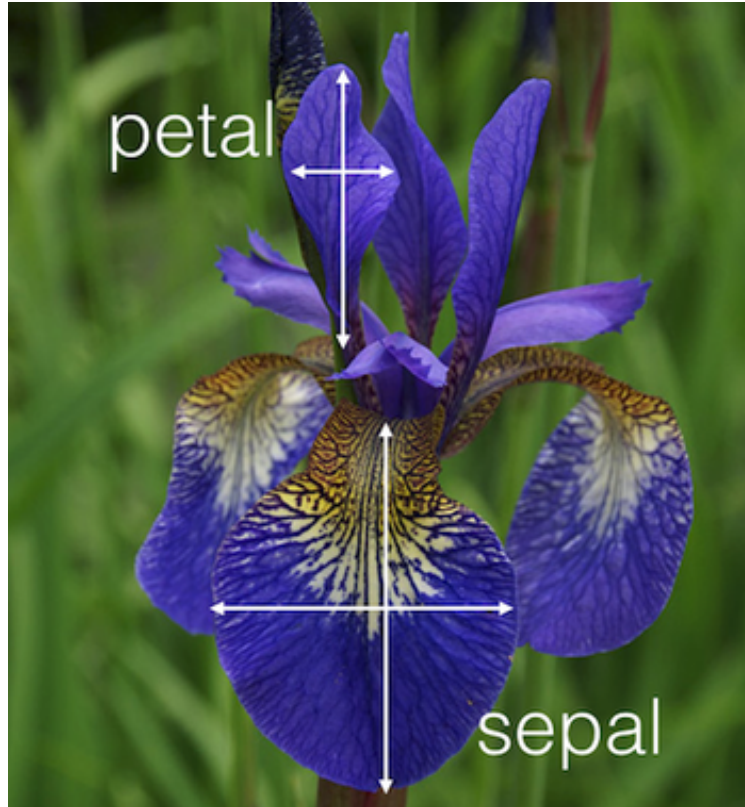
- Comment sont représentées les données dans l'ordinateur ?
- Comment représenter les données dans l'espace ?
- Comment découvrir des "clusters" dans les données ?

- classification hiérarchique
- kmeans
- comment déterminer le nombre de groupe optimal ?
- comment comparer deux classifications ?

Les données dans l'ordinateur (1)

Les iris de Fisher

Ces données sont un classique des méthodes d'apprentissage



Les données dans l'ordinateur (2)

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

Les données dans l'ordinateur (2)

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2

3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2
6	5.4	3.9	1.7	0.4

- 1 ligne = 1 fleur = 1 vecteur
- 1 colonne = 1 variable = 1 vecteur
- l'ensemble des données = 1 échantillon = 1 data.frame

Représentons ces données : une fleur (1)

```
mes.iris[1,]
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	5.1	3.5	1.4	0.2



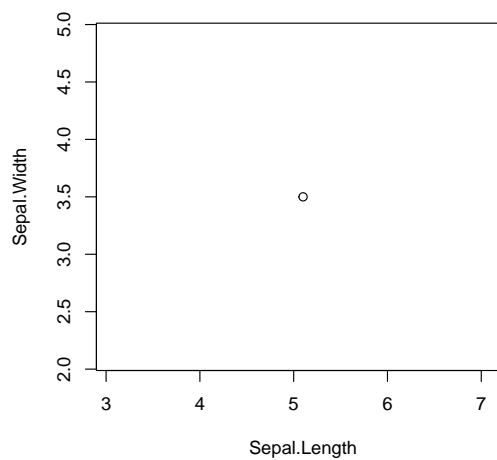
Comment représenter cette fleur ?

- par un point !

Dans quel espace de représentation ?

Représentons ces données : une fleur (2)

```
plot(mes.iris[1,1:2])
```



Dans le plan, un point de coordonnées :

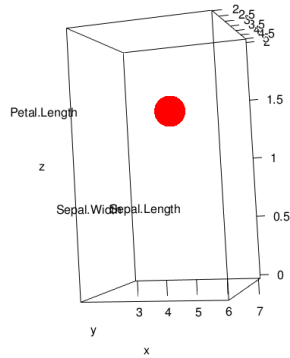
- $x = 5.1$
- $y = 3.5$

représenté par un vecteur $v2 = (5.1, 3.5)$ dans \mathbb{R}^2

Représentons ces données : une fleur (3)

Dans l'espace, un point de coordonnées :

- $x = 5.1$
- $y = 3.5$
- $z = 1.4$



représenté par un vecteur $v3 = (5.1, 3.5, 1.4)$ dans \mathbb{R}^3

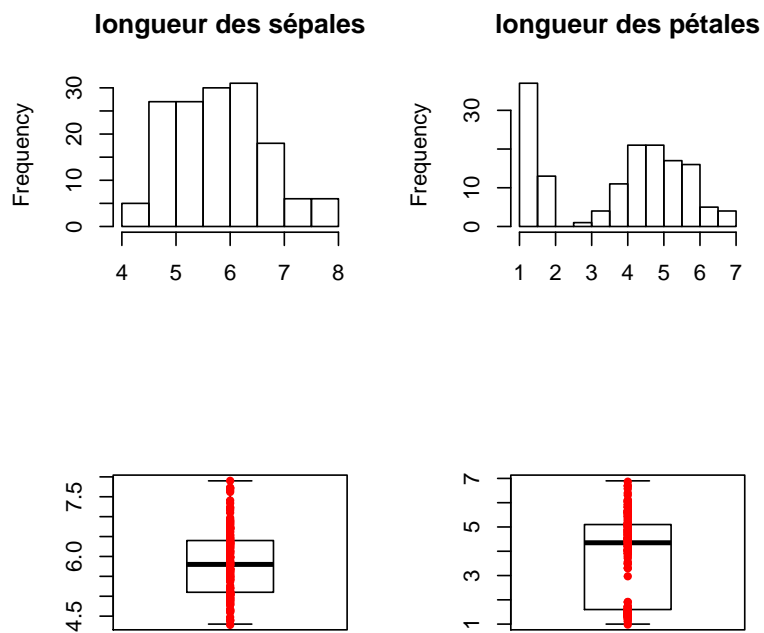
Représentons ces données : toutes les fleurs (4)

= un nuage de points dans un espace à 4 dimensions

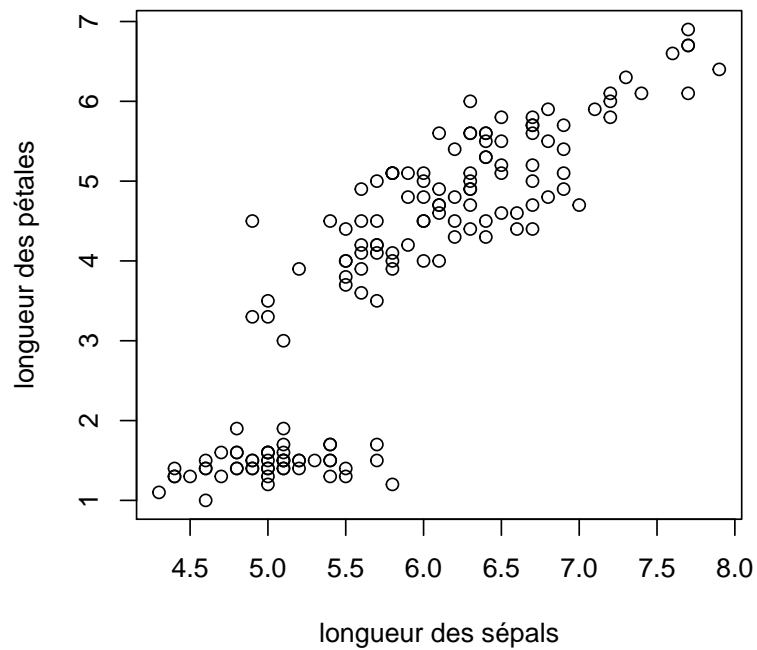
- chaque point est représenté par un vecteur dans \mathbb{R}^4
- le nuage de points est représenté par une matrice à n et p (= 4 dimensions)
 - n = nombre de lignes = nombre d'individus = taille de l'échantillon
 - p = nombre de colonnes = nombre de variables décrivant l'échantillon

= PAS de représentation possible (pour l'instant)

Représentons ces données : une variable à la fois (1)



Représentons ces données : deux variables à la fois (2)



Il faut tenir compte de toutes les dimensions

c'est à dire de toutes les variables à notre disposition

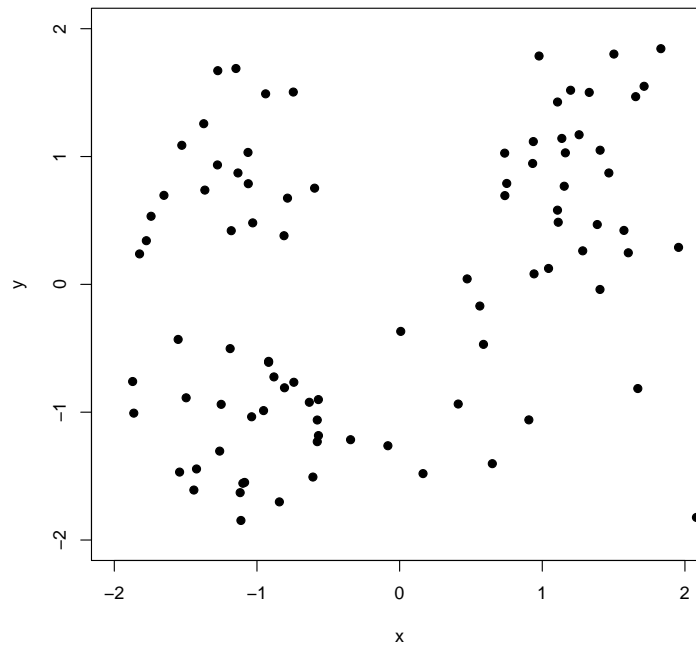


Figure 1: Y a-t-il des groupes ?

Clustering et classification (termes anglais)

On a une **information** sur nos données

- variables quantitatives = vecteur de réels

Clustering : on cherche à mettre en évidence des groupes dans les données

- le clustering appartient aux méthodes dites **non supervisées**, ou descriptives

Clustering et classification (termes anglais)

On a une **information** sur nos données

Clustering : on cherche à mettre en évidence des groupes dans les données

Classification :

- on connaît le partitionnement de notre jeu de données
 - variables quantitatives = vecteur de réels
 - ET
 - variable qualitative = groupe (cluster) d'appartenance = vecteurs de entiers / niveau d'un facteur
 - on cherche à prédire le groupe (la classe) de nouvelles données
- la classification appartient aux méthodes dites **supervisées**, ou prédictives

Clustering

Méthodes

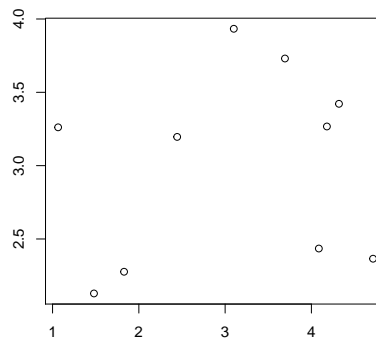
Trois grands principes de méthodes basées sur:

- La géométrie
- Les probabilités (statistique)
- Les graphes

En fait, trois façons de voir les mêmes algorithmes

Géométrie et distances (1)

On considère les données comme des points de \mathbb{R}^n



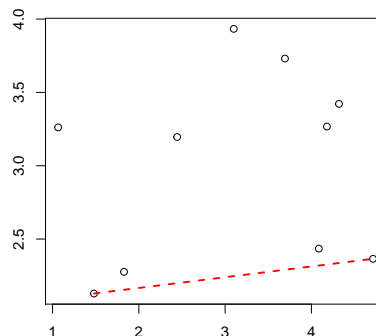
\mathbb{R}^n : espace Euclidien à n dimensions, où

- chaque dimension représente une des variables observées;
- un individu est décrit comme un vecteur à n valeurs, qui correspond à un point dans cet espace.

Géométrie et distances (2)

On considère les données comme des points de R^n (*)

- géométrie donnée par distances
- distances = dissimilarités imposées par le problème
- dissimilarités \rightarrow permettent visualisation de l'ensemble des points



Géométrie et distances (3)

Sur la base d'une distance (souvent euclidienne)

- Clustering :
 - Méthode agglomérative ou hierarchical clustering
 - Moyennes mobiles ou K-means : séparation optimale des groupes connaissant le nombre de groupes

Distances

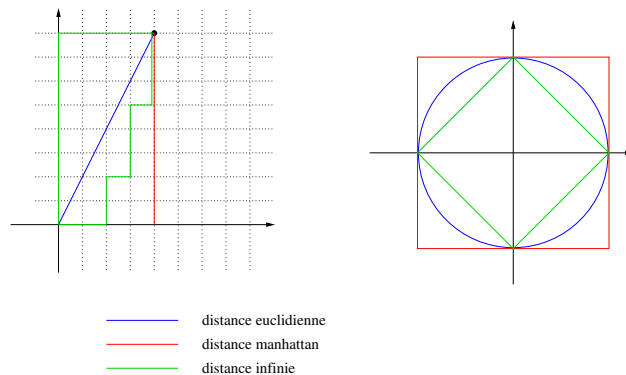
Définition d'une distance : fonction positive de deux variables

1. $d(x, y) \geq 0$
2. $d(x, y) = d(y, x)$
3. $d(x, y) = 0 \iff x = y$
4. **Inégalité triangulaire** : $d(x, z) \leq d(x, y) + d(y, z)$

Si 1,2,3 : dissimilarité

Distances utilisées dans R (1)

- distance euclidienne ou distance L_2 : $d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$
- distance de manhattan ou distance L_1 : $d(x, y) = \sum_i |x_i - y_i|$
- distance du maximum ou L-infinis, L_∞ : $d(x, y) = \max_i |x_i - y_i|$



Distances utilisées dans R (2)

- distance de Minkowski l_p :

$$d(x, y) = \sqrt[p]{\sum_i |x_i - y_i|^p}$$

- distance de Canberra (x et y valeurs positives):

$$d(x, y) = \sum_i \frac{|x_i - y_i|}{x_i + y_i}$$

- distance binaire ou distance de Jaccard ou Tanimoto: proportion de propriétés communes

Autres distances non géométriques (pour information)

Utilisées en bio-informatique:

- Distance de **Hamming**: nombre de remplacements de caractères (substitutions)

- Distance de **Levenshtein**: nombre de substitutions, insertions, deletions entre deux chaînes de caractères

$$d("BONJOUR", "BONSOIR") = 2$$

- Distance d'**alignements**: distances de Levenshtein avec poids (par ex. matrices BLOSSUM)
- Distances d'**arbre** (Neighbor Joining)
- Distances **ultra-métriques** (phylogénie UPGMA)

Distances plus classiques en génomique

Il existe d'autres mesures de distances, plus ou moins adaptées à chaque problématique :

- **Jaccard** (comparaison d'ensembles): $J_D = \frac{A \cap B}{A \cup B}$
- Distance du χ^2 (comparaison de tableau d'effectifs)

Ne sont pas des distances, mais indices de dissimilarité :

- **Bray-Curtis** (en écologie, comparaison d'abondance d'espèces)
- **Jensen-Shannon** (comparaison de distributions)

Note : lors du TP, sur les données d'expression RNA-seq, nous utiliserons le **coefficient de corrélation de Spearman** et la distance dérivée, $d_c = 1 - r$ ou $d_c = \sqrt{2 \times (1 - r)}$

Avec R (1)

- on utilise la fonction `dist()` avec l'option `method = "euclidean", "manhattan", ...`

4.32	1.48	1.83	4.71	3.10
3.42	2.13	2.28	2.37	3.93

distance euclidienne : 4.07

distance de manhattan = 10.73

Avec R (2)

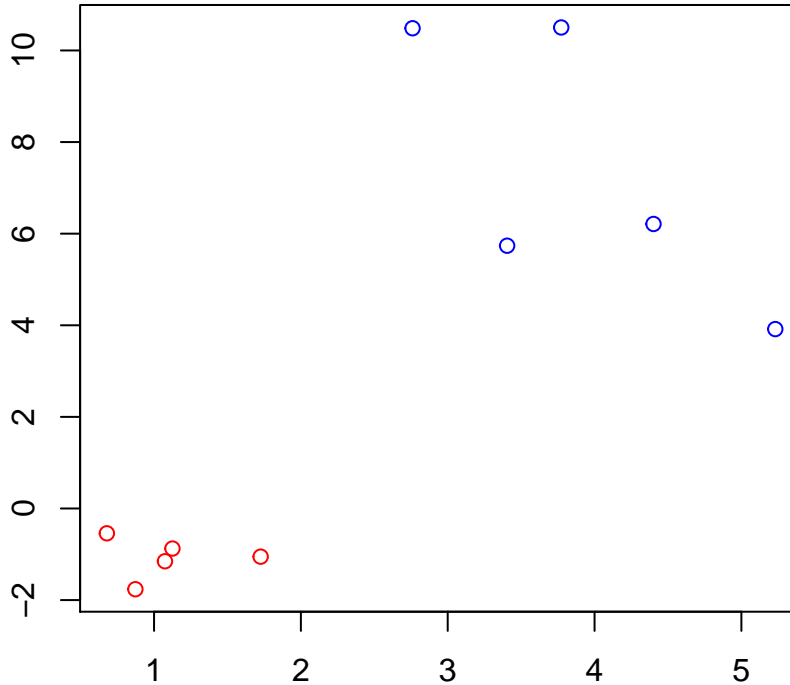
- ou pour des distances particulières, par exemple l'indice de Jaccard :

v.a	0	1	0	0	0	0	0
v.b	0	1	0	0	0	1	0
v.c	0	1	0	0	0	0	0

```

      v.a      v.b
v.b 0.3333333
v.c 0.0000000 0.3333333
```

Distances entre groupes (1)



Distances entre groupes (2)

- **Single linkage** : éléments les plus proches des 2 groupes

$$D(C_1, C_2) = \min_{i \in C_1, j \in C_2} D(x_i, x_j)$$

- **Complete linkage** : éléments les plus éloignés des 2 groupes

$$D(C_1, C_2) = \max_{i \in C_1, j \in C_2} D(x_i, x_j)$$

Distances entre groupes (3)

- **Group average** : distance moyenne

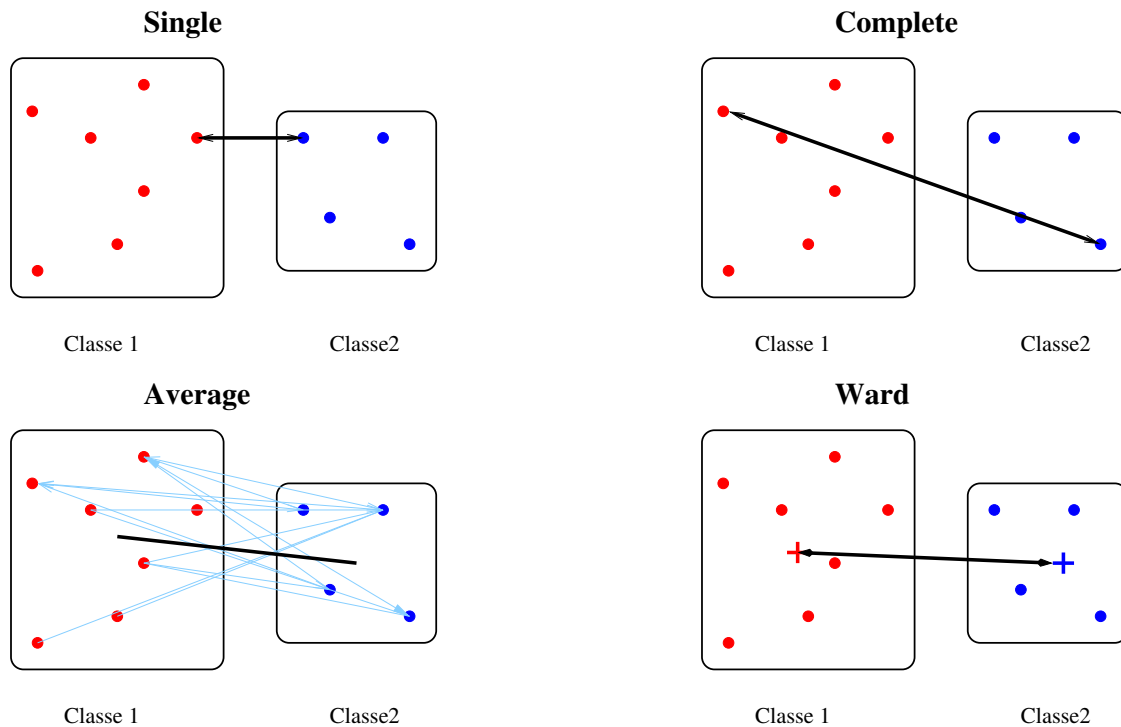
$$D(C_1, C_2) = \frac{1}{N_1 N_2} \sum_{i \in C_1, j \in C_2} D(x_i, x_j)$$

- **Ward**

$$d^2(C_i, C_j) = I_{intra}(C_i \cup C_j) - I_{intra}(C_i) - I_{intra}(C_j)$$

$$D(C_1, C_2) = \sqrt{\frac{N_1 N_2}{N_1 + N_2}} \|m_1 - m_2\|$$

Distances entre groupes (4)



Les données

Ces données sont un classique des méthodes d'apprentissage

Dans un premier temps, regardons les données

```
dim(mes.iris)
```

```
[1] 150  4
```

```
head(mes.iris)
```

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

```
str(mes.iris)
```

```
'data.frame':  150 obs. of  4 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
```

```
summary(mes.iris)
```

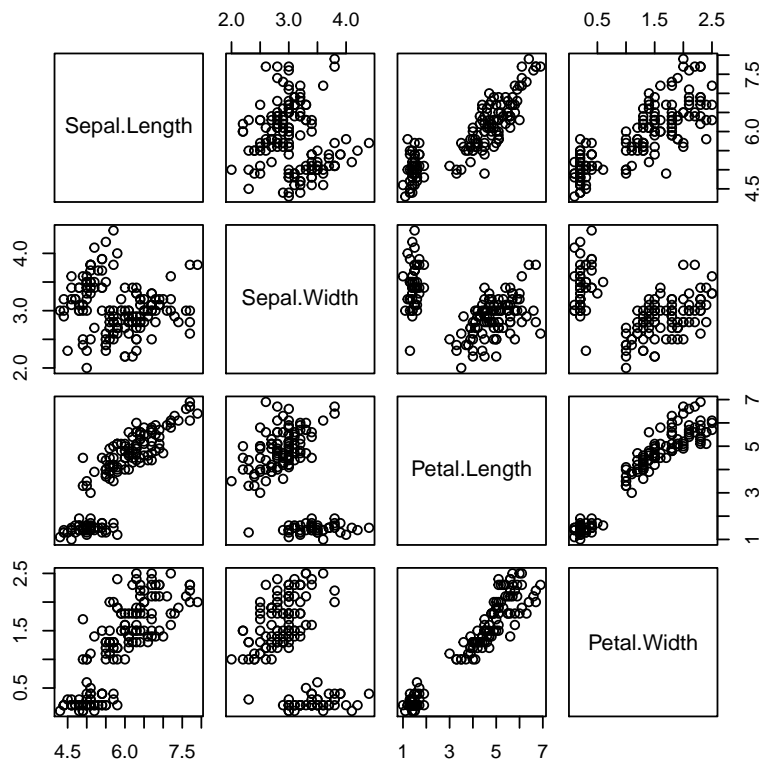
Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
Min. :4.300	Min. :2.000	Min. :1.000	Min. :0.100
1st Qu.:5.100	1st Qu.:2.800	1st Qu.:1.600	1st Qu.:0.300
Median :5.800	Median :3.000	Median :4.350	Median :1.300
Mean :5.843	Mean :3.057	Mean :3.758	Mean :1.199
3rd Qu.:6.400	3rd Qu.:3.300	3rd Qu.:5.100	3rd Qu.:1.800
Max. :7.900	Max. :4.400	Max. :6.900	Max. :2.500

Visualisation des données

On peut ensuite essayer de visualiser les données

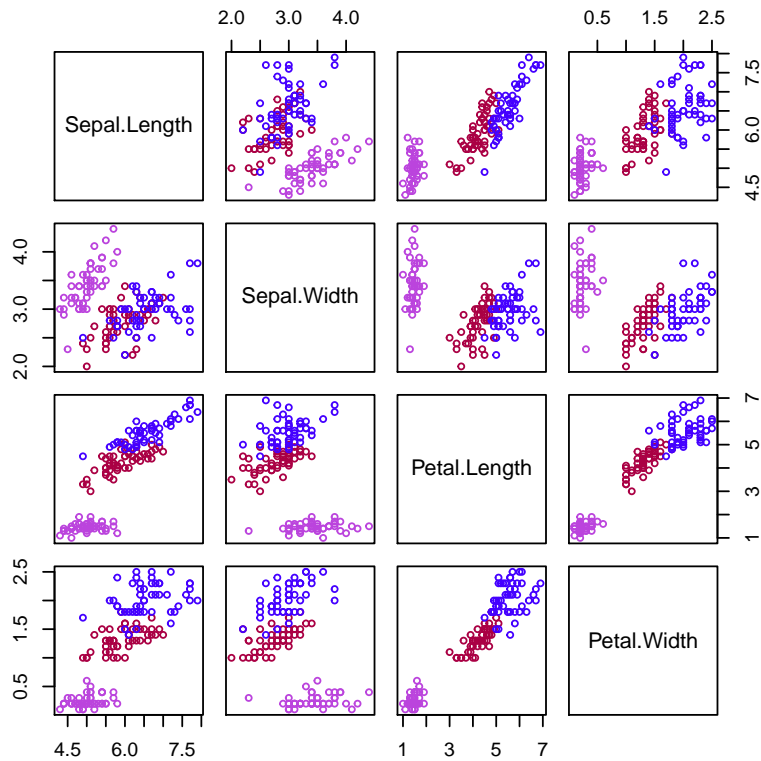
- par un plot

```
plot(mes.iris)
```



Visualisation des données - coloration par espèces

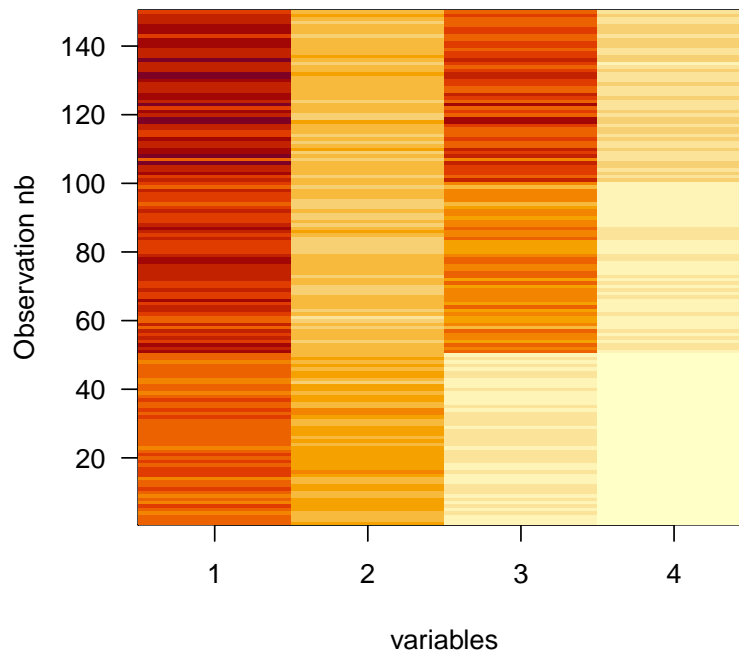
```
species.colors <- c(setosa = "#BB44DD", virginica = "#AA0044", versicolor = "#4400FF")
plot(mes.iris, col = species.colors[iris$Species], cex = 0.7)
```



Visualisation des données

- par la fonction `image()`

```
image(1:nb.var, 1:nb.iris, t(as.matrix(mes.iris)), xlab = "variables", ylab = "Observation nb", las = 1)
```



Nettoyage des données (1) : données manquantes

Avant de commencer à travailler, il est nécessaire de commencer par vérifier que :

- il n'y a pas de données manquantes

```
sum(is.na(mes.iris))
```

```
[1] 0
```

Nettoyage des données (2) : variables constantes

- aucune variable n'est constante (aucune variable n'a une variance nulle)

```
iris.var <- apply(mes.iris, 2, var)
kable(iris.var, digits = 3, col.names = "Variance")
```

	Variance
Sepal.Length	0.686
Sepal.Width	0.190
Petal.Length	3.116
Petal.Width	0.581

```
sum(apply(mes.iris, 2, var) == 0)
```

```
[1] 0
```

Normalisation

Afin de pouvoir considérer que toutes les variables sont à la même échelle, il est parfois nécessaire de normaliser les données.

- soit
 - en centrant (ramener la moyenne de chaque variable à 0)

```
mes.iris.centre <- scale(mes.iris, center=TRUE, scale=FALSE)
```

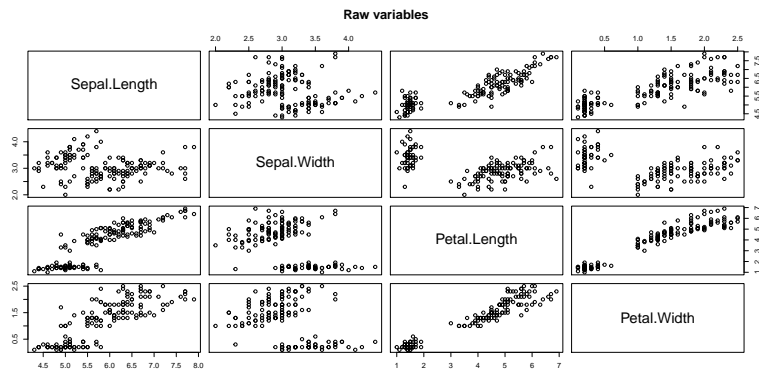
- soit
 - en centrant (ramener la moyenne de chaque variable 0)
 - et mettant à l'échelle (ramener la variance de chaque variable à 1)

```
mes.iris.scaled <- scale(mes.iris, center=TRUE, scale=TRUE)
```

On peut visuellement regarder l'effet de la normalisation :

par un plot des données

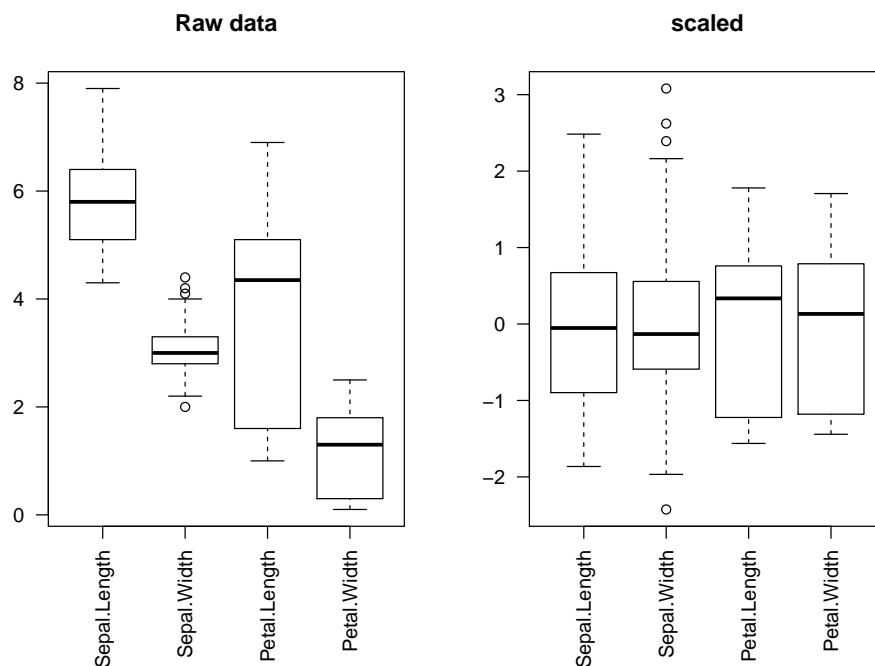
```
plot(mes.iris, main = "Raw variables")
```

! ne pas faire si “grosses” données

... par une boîte à moustaches (boxplot)

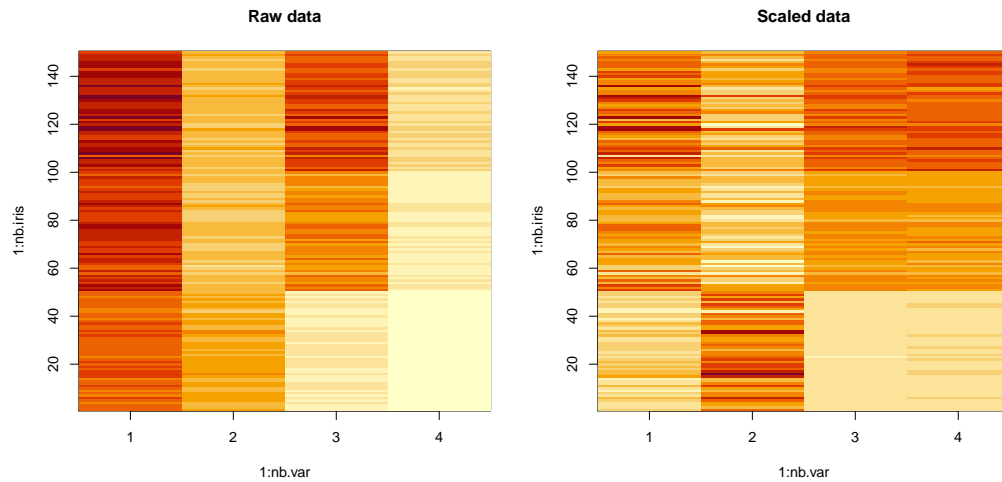
```
par(mfrow = c(1,2))
par(mar = c(7, 4.1, 4.1, 1.1)) # adapt margin sizes for the labels
boxplot(mes.iris, main = "Raw data", las = 2)
boxplot(mes.iris.scaled, main = "scaled", las = 2)
```



```
par(mar = c(5.1, 4.1, 4.1, 2.1)) # Restore original margin sizes
```

... par une image

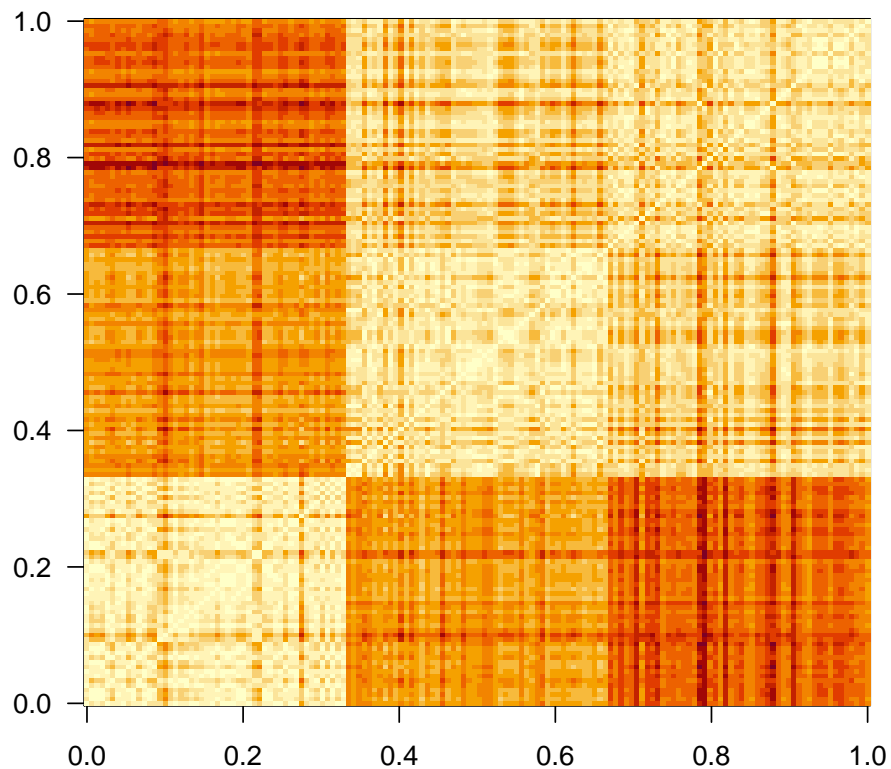
```
par(mfrow=c(1,2))
image(1:nb.var, 1:nb.iris, t(as.matrix(mes.iris)), main="Raw data")
image(1:nb.var, 1:nb.iris, t(as.matrix(mes.iris.scaled)), main="Scaled data")
```



La matrice de distances

Nous utilisons ici la distance euclidienne sur données **normalisées**.

Données normalisées



La classification hiérarchique : principe

classification hiérarchique : mettre en évidence des liens hiérarchiques entre les individus

- classification hiérarchique **ascendante** : partir des individus pour arriver à des classes / cluster
- classification hiérarchique **descendante** : partir d'un groupe qu'on subdivise en sous-groupes /clusters jusqu'à arriver à des individus.

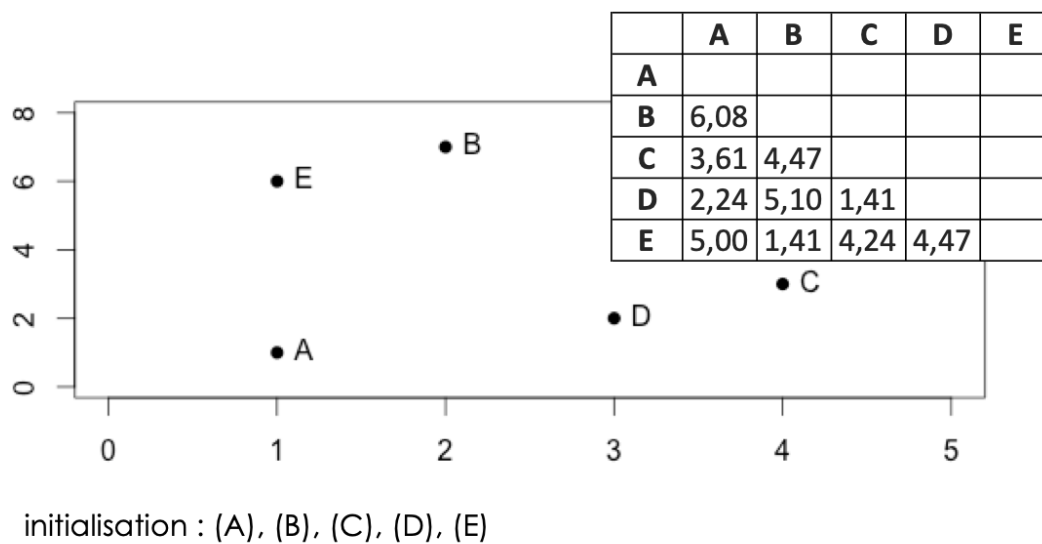
Notion importante, cf distances

- ressemblance entre individus = distance
- ressemblance entre groupes d'individus = critère d'aggrégation
 - lien simple
 - lien complet
 - lien moyen
 - critère de Ward

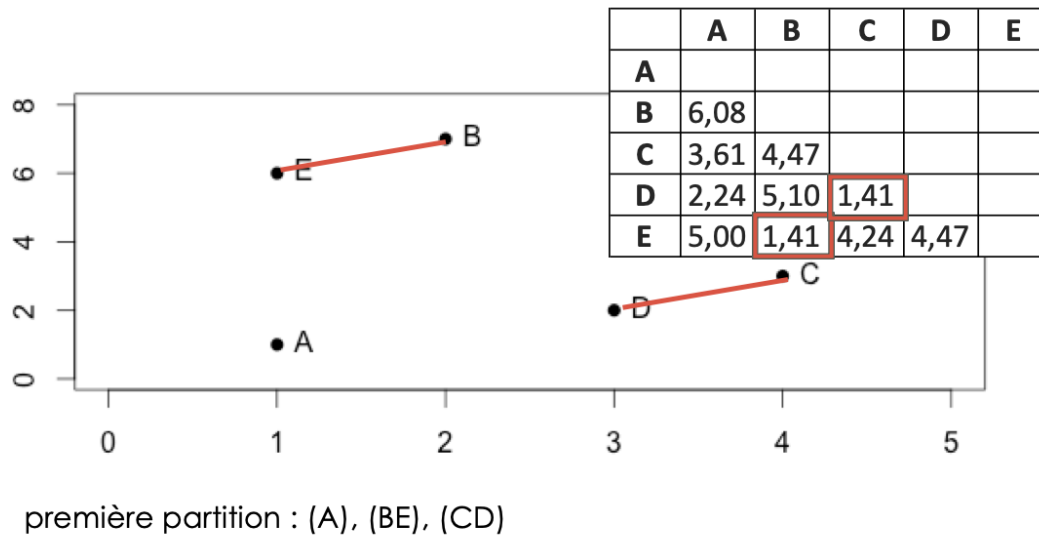
L'algorithme : étape 1

- départ : n individus = n clusters distincts
- calcul des distances entre tous les individus
 - choix de la métrique à utiliser en fonction du type de données
- regroupement des 2 individus les plus proches => (n-1) clusters

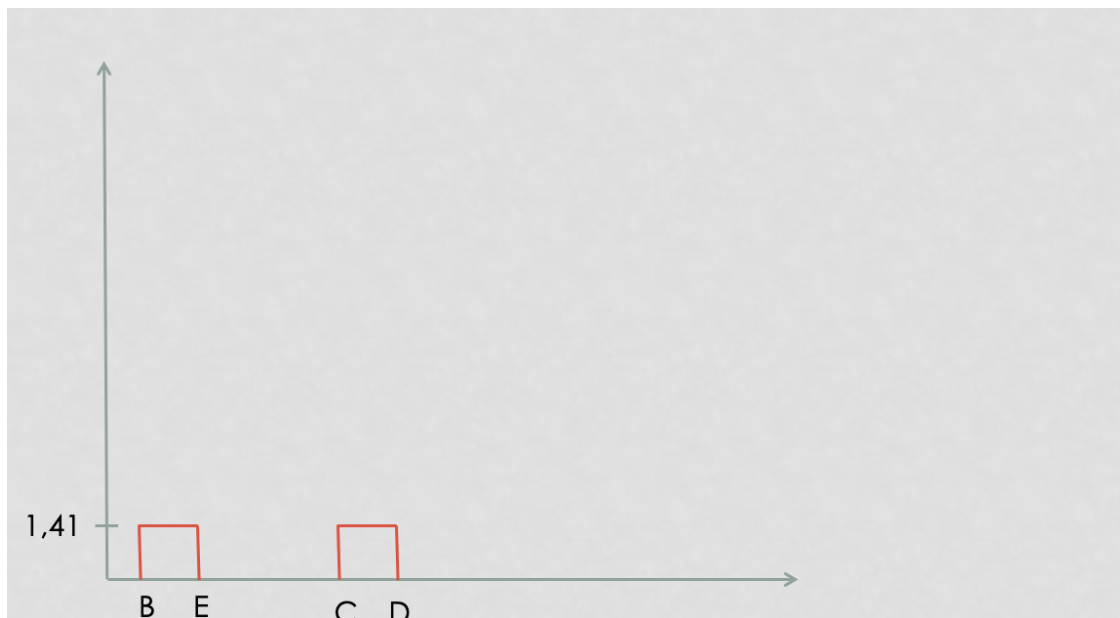
au départ



identification des individus les plus proches



construction du dendrogramme

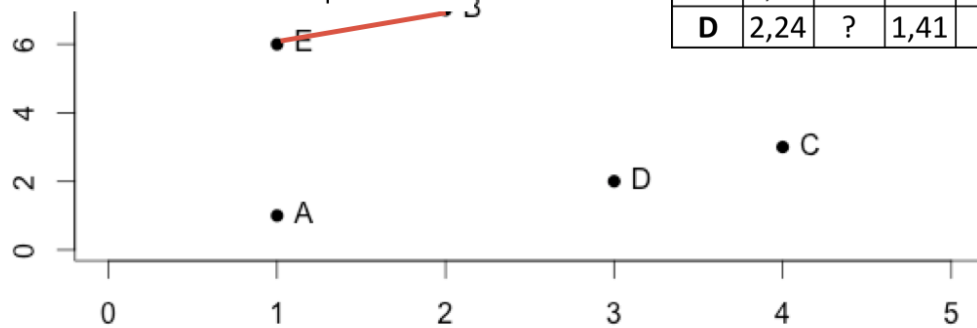


étape j :

- calcul des dissemblances entre chaque groupe obtenu à l'étape $(j - 1)$
- regroupement des deux groupes les plus proches $\Rightarrow (n - j)$ clusters

calcul des nouveaux représentants 'BE' et 'CD'

par le lien moyen,
calcul de BE
calcul de la distance de BE
à tous les points



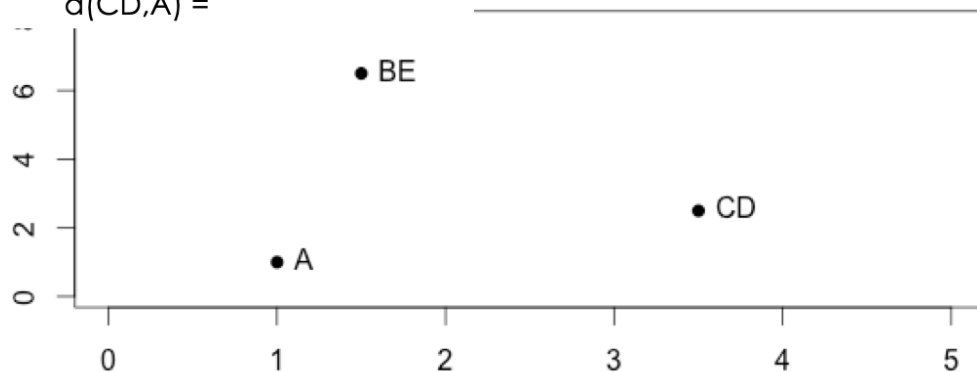
	A	BE	C	D
A				
BE	?			
C	3,61	?		
D	2,24	?	1,41	

par le lien moyen,
calcul de CD
calcul de la distance de CD

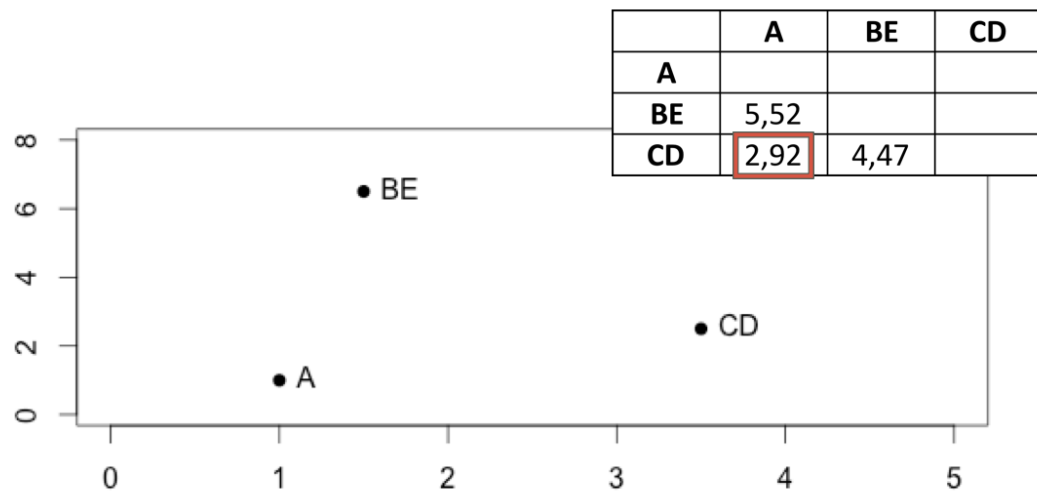
calcul des distances de l'individu restant 'A' aux points moyens

calcul des distances
 $d(BE, A) =$
 $d(BE, CD) =$
 $d(CD, A) =$

$$d(BE, A) = \frac{d(B, A) + d(E, A)}{2} = \frac{6.08 + 5}{2} = 5.52$$



A est plus proche de ...

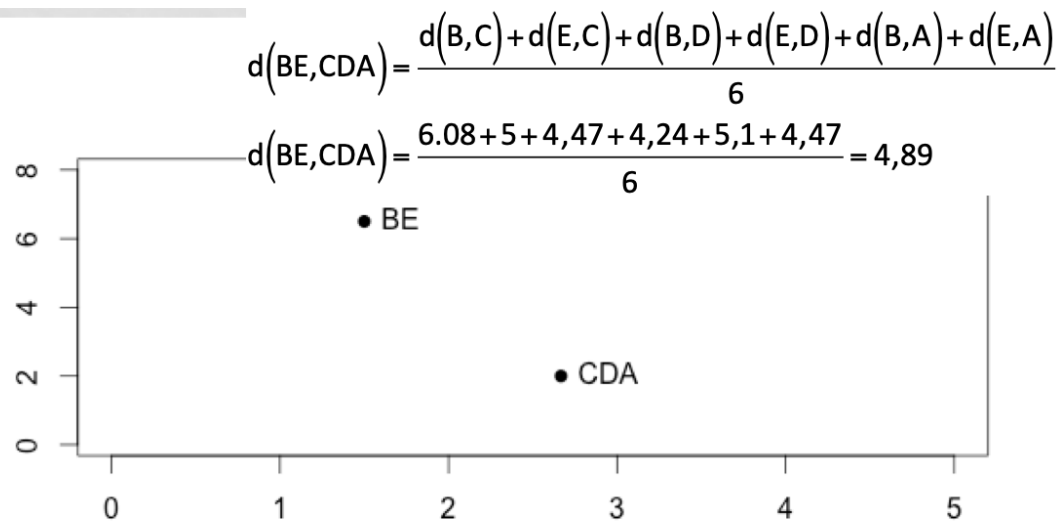


deuxième partition : (BE), (CDA)

dendrogramme

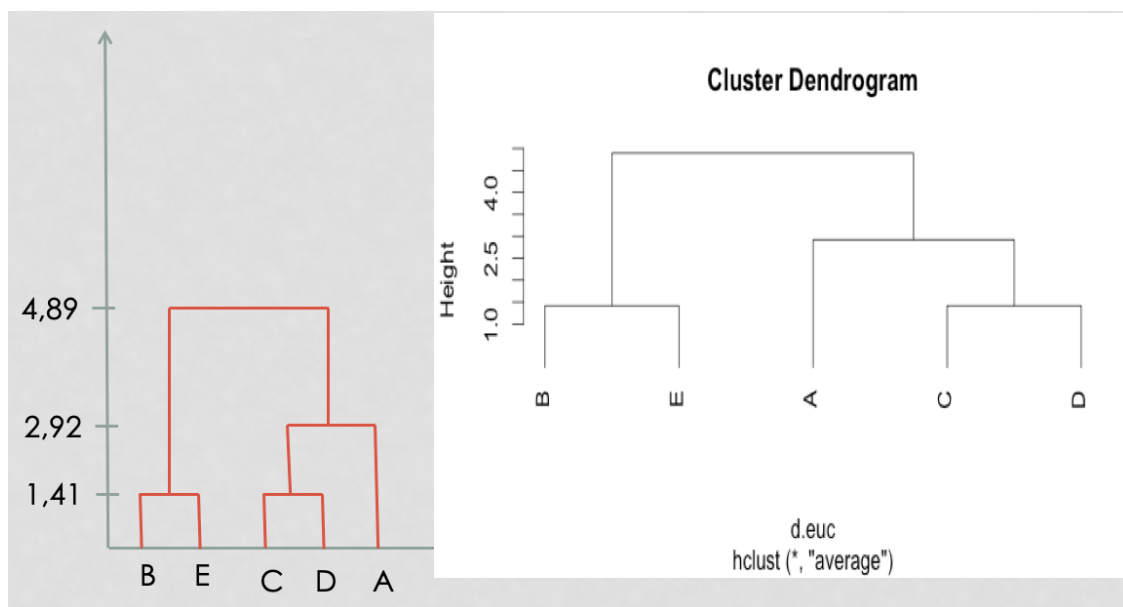


pour finir



- à l'étape $(n - 1)$, tous les individus sont regroupés dans un même cluster

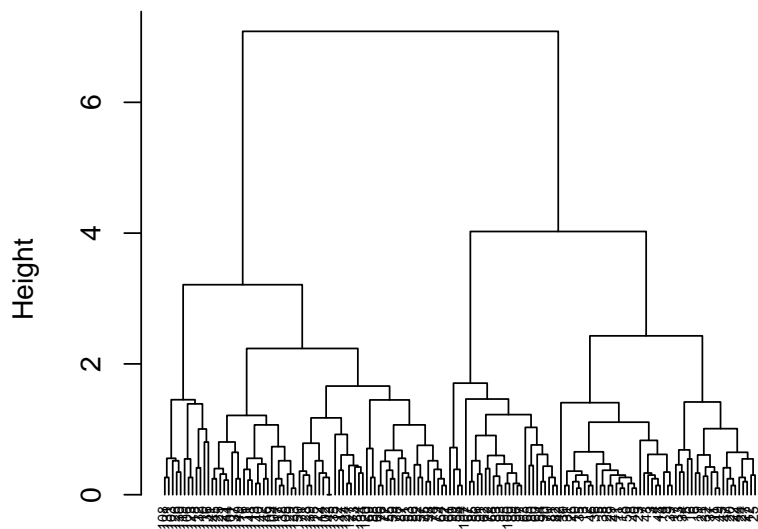
dendrogramme final



Je ne fais pas attention à ce que je fais ...

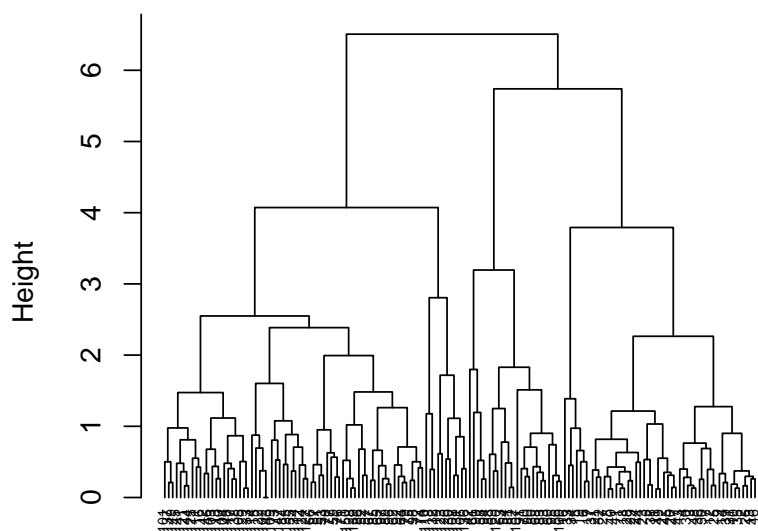
... c'est à dire aux options des fonctions `dist()` et `hclust()`

Cluster Dendrogram



iris.euc
hclust (*, "complete")

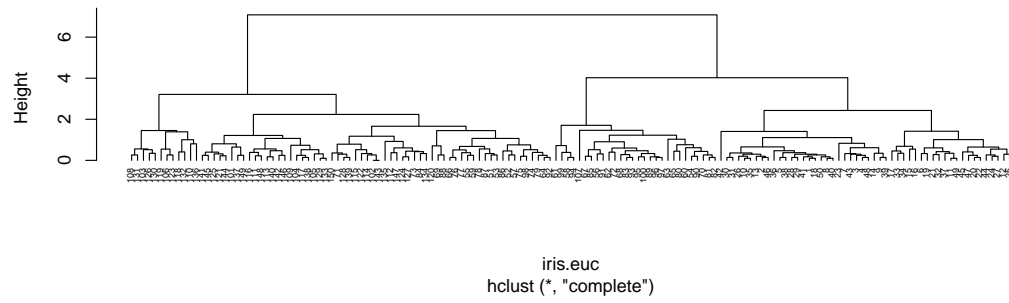
Cluster Dendrogram



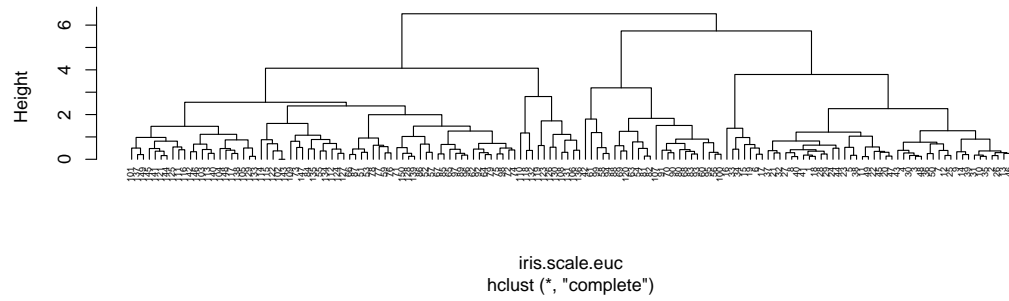
iris.scale.euc
hclust (*, "complete")

```
par(mfrow = c(2, 1))
plot(iris.hclust, hang = -1, cex = 0.5, main = "Données brutes")
plot(iris.scale.hclust, hang = -1, cex = 0.5, main = "Normalisées")
```


Données brutes

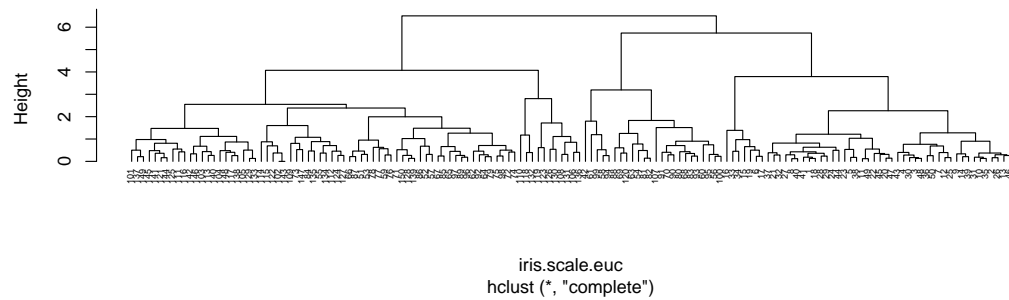


Normalisées



En utilisant une autre métrique

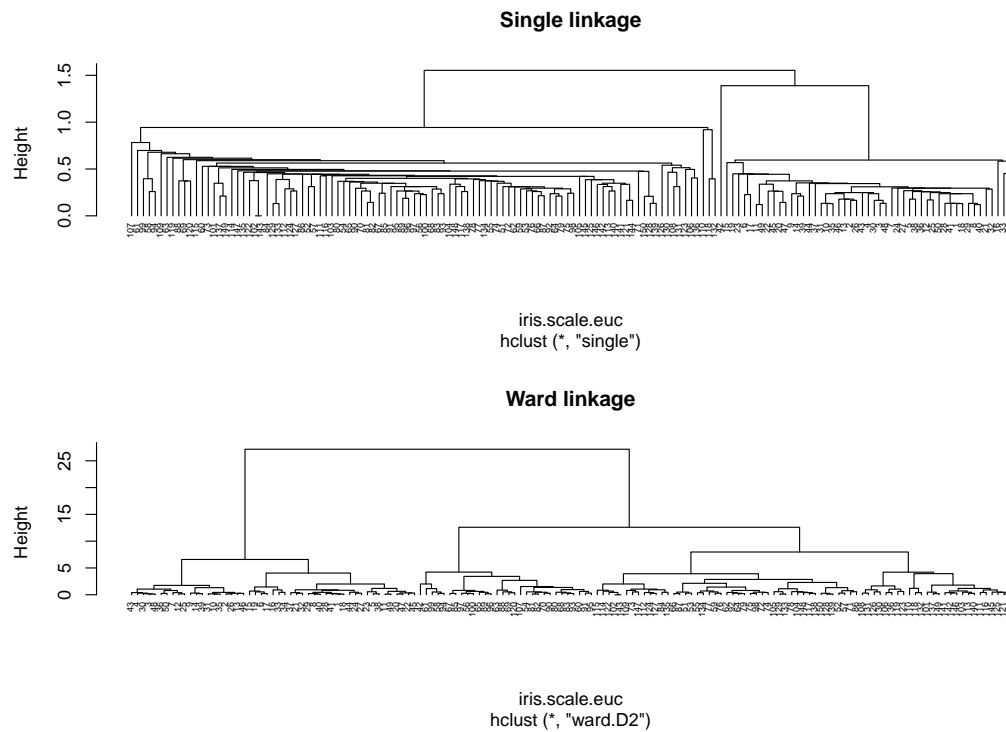
Euclidian dist



Manhattan dist



En utilisant un autre critère d'aggrégation

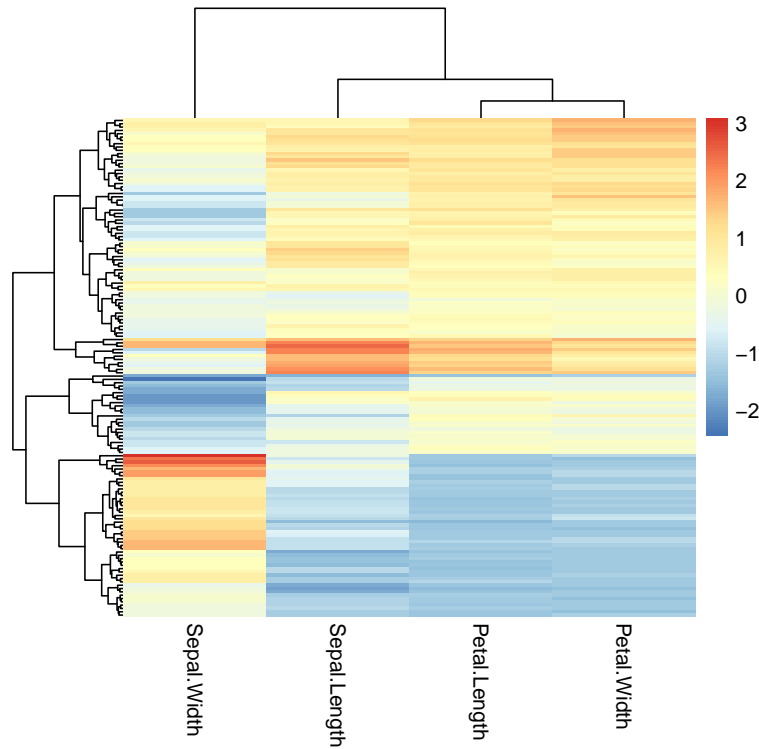


En conclusion

- Faire attention au données
 - données manquantes
 - données invariantes
 - données normalisées
 - Choisir la distance et le critère d'aggrégation adaptés à nos données
-

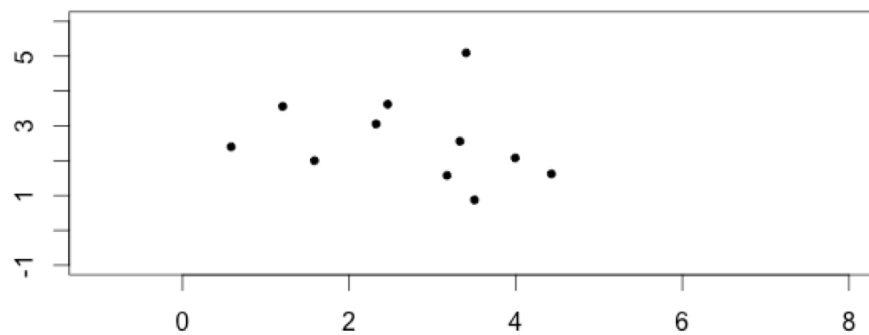
Les heatmap

```
pheatmap::pheatmap(mes.iris.scaled)
```



Les k-means

Les individus dans le plan



=> faire apparaitre des classes / des clusters

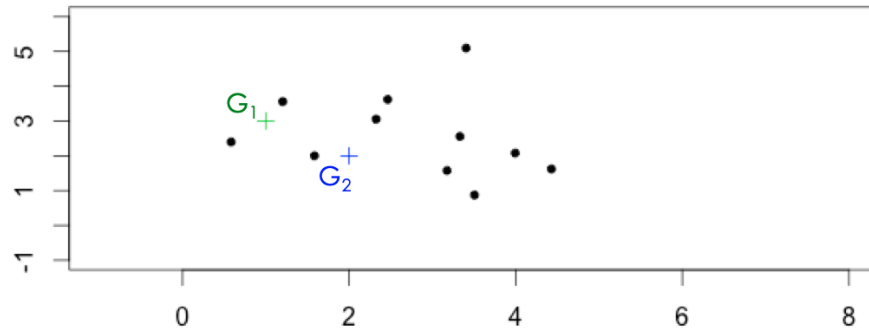
L'algorithme

étape 1 :

- k centres provisoires tirés au hasard
- k clusters créés à partir des centres en regroupant les individus les plus proches de chaque centre
- obtention de la partition P_0

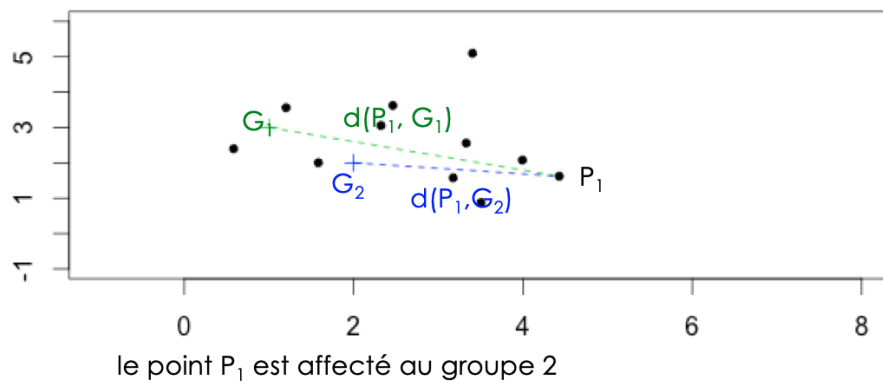
Choix des centres provisoires

combien de cluster ?
les deux centres initiaux (G_1 et G_2) sont choisis au hasard

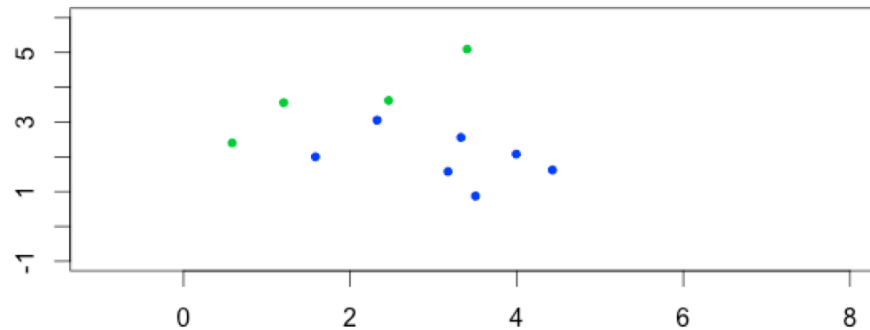


Calcul des distances aux centres provisoires

- calcul des distances de chaque point aux centres G_1 et G_2 ,



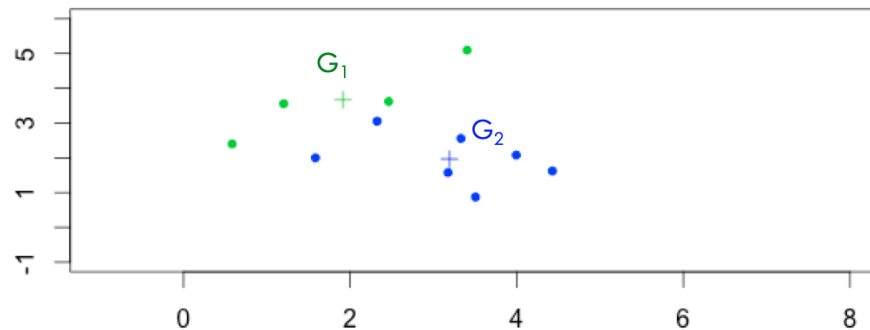
Affectation à un cluster



Calcul des nouveaux centres de classes

Etape j :

- construction des centres de gravité des k clusters construits à l'étape $(j - 1)$
- k nouveaux clusters créés à partir des nouveaux centres suivant la même règle qu'à l'étape 0
- obtention de la partition P_j

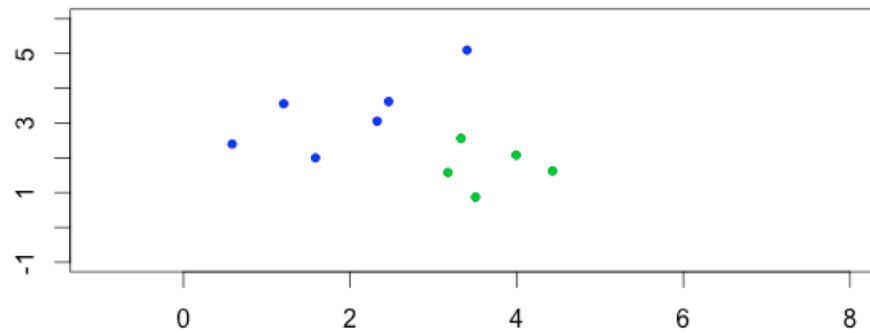


Fin :

- l'algorithme converge vers une partition stable

Arrêt :

- lorsque la partition reste la même, ou lorsque la variance intra-cluster ne décroît plus, ou lorsque le nombre maximal d'itérations est atteint.



Un premier k-means en 5 groupes

```
iris.scale.kmeans5 <- kmeans(mes.iris.scaled, center=5)
iris.scale.kmeans5
```

K-means clustering with 5 clusters of sizes 29, 48, 28, 23, 22

Cluster means:

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width
1	1.3926646	0.2323817	1.1567451	1.21327591
2	0.3804044	-0.3896455	0.6067908	0.56390985
3	-0.7467198	1.4252951	-1.2932659	-1.21734309
4	-0.3516137	-1.3285553	0.1026061	0.01228268
5	-1.3477916	0.1187465	-1.3100027	-1.29316224

Clustering vector:

```
[1] 3 5 5 5 3 3 5 3 5 5 3 5 5 3 3 3 3 3 3 3 5 5 5 3 3 3 5 5 3 3 3 5 5 3 3 5 5 3 3 5 5 3 3 5 3 3 5 3 3
[148] 2 1 2
```

Within cluster sum of squares by cluster:

```
[1] 26.891293 27.830133 13.761588 13.686590 8.032603
(between_SS / total_SS = 84.9 %)
```

Available components:

```
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss" "betweenss"    "size"
```

Comment déterminer le nombre de clusters ? (1)

Ces méthodes non supervisées, sont sans *a priori* sur la structure, le nombre de groupe, des données.

rappel : un cluster est composé

- d'individus qui se ressemblent
- d'individus très différents des individus de ceux des autres clusters

Comment déterminer le nombre de clusters ? (2)

- si les individus d'un même cluster sont proches
 - homogénéité maximale à l'intérieur de chaque cluster => variance intra faible
- si les individus de 2 clusters différents sont éloignés => variance inter forte
 - hétérogénéité maximale entre chaque cluster

Comment déterminer le nombre de clusters ? avec la classification hiérarchique

La coupure de l'arbre à un niveau donné construit une partition. la coupure doit se faire :

- après les agrégations correspondant à des valeurs peu élevées de l'indice
- avant les agrégations correspondant à des niveaux élevés de l'indice, qui dissocient les groupes bien distincts dans la population.

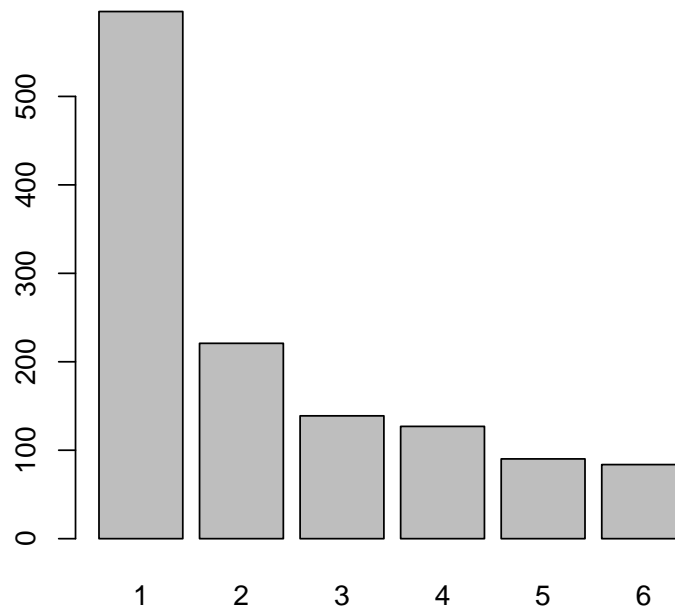
```
plot(iris.scale.hclust.ward, hang=-1, cex=0.5)
```

Cluster Dendrogram



Comment déterminer le nombre de clusters ? avec les kmeans

variance intra en fonction du nombre de cluster



Comparaison de clustering: Rand Index

Mesure de similarité entre deux clustering

à partir du nombre de fois que les classifications sont d'accord

$$R = \frac{m + s}{t}$$

- m=nombre de paires dans la même classe dans les deux classifications
- s=nombre de paires séparées dans les deux classifications
- t=nombre de paires totales

Comparaison de clustering: Adjusted Rand Index

$$ARI = \frac{RI - ExpectedRI}{MaxRI - ExpectedRI}$$

- ARI=RI normalisé
- Prend en compte la taille des classes
- ARI=1 pour classification identique
- ARI $\simeq 0$ pour classification aléatoire (peut être <0)
- Adapté pour nombre de classe différent entre les deux classifications et taille de classe différente

Comparaison des résultats des deux clustering

- par une table de confusion

0	0	29
0	0	20
0	29	1
21	24	0
26	0	0

Comparaison de clustering: Rand Index

Mesure de similarité entre deux clustering

à partir du nombre de fois que les classifications sont d'accord

$$R = \frac{m + s}{t}$$

- m = nombre de paires dans la même classe dans les deux classifications
- s = nombre de paires séparées dans les deux classifications
- t = nombre de paires totales

Comparaison de clustering: Adjusted Rand Index

$$ARI = \frac{RI - ExpectedRI}{MaxRI - ExpectedRI}$$

- ARI=RI normalisé
- Prend en compte la taille des classes
- ARI=1 pour classification identique
- ARI \simeq 0 pour classification aléatoire (peut être <0)
- Adapté pour nombre de classe différent entre les deux classifications et taille de classe différente

Comparaison des résultats des deux classifications

- rand index et adjusted rand index

```
clues::adjustedRand(cluster.hclust5, cluster.kmeans3)
```

	Rand	HA	MA	FM	Jaccard
	0.7848770	0.4637776	0.4730527	0.6167001	0.4299265

Pros et cons des différents algorithmes

Algorithme	Pros	Cons
Hiérarchique	<p>L'arbre reflète la nature imbriquée de tous les sous-clusters</p> <p>Permet une visualisation couplée dendrogramme (groupes) + heatmap (profils individuels)</p> <p>Choix a posteriori du nombre de clusters</p>	Complexité quadratique (mémoire et temps de calcul) → quadruple chaque fois qu'on double le nombre d'individus

Algorithme	Pros	Cons
K-means	Rapide (linéaire en temps), peut traiter des jeux de données énormes (centaines de milliers de pics ChIP-seq)	Positions initiales des centres est aléatoire → résultats changent d'une exécution à l'autre Distance euclidienne (pas appropriée pour transcriptome par exemple)

```
## Print the complete list of libraries + versions used in this session
sessionInfo()
```

R version 3.6.1 (2019-07-05)

Platform: x86_64-apple-darwin15.6.0 (64-bit)

Running under: macOS Mojave 10.14.6

Matrix products: default

BLAS: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib

LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib

locale:

[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:

[1] stats graphics grDevices utils datasets methods base

other attached packages:

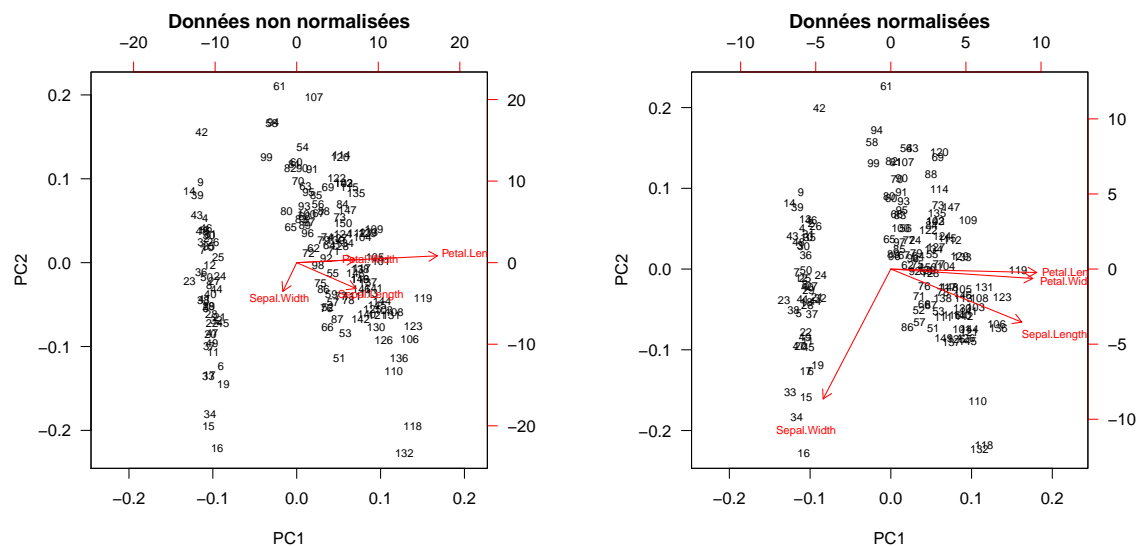
[1] pheatmap_1.0.12 vegan_2.5-6 lattice_0.20-38 permute_0.9-5 rgl_0.100.30 RColor

loaded via a namespace (and not attached):

[1] ggrepel_0.8.1 Rcpp_1.0.2 assertthat_0.2.1 zeallot_0.1.0
 [13] rlang_0.4.0 lazyeval_0.2.2 rstudioapi_0.10 miniUI_0.1.1.1
 [25] shiny_1.4.0 compiler_3.6.1 httpuv_1.5.2 xfun_0.10
 [37] dplyr_0.8.3 later_1.0.0 MASS_7.3-51.4 leaps_3.1
 [49] promises_1.1.0 scatterplot3d_0.3-41 xml2_1.2.2 vctr_0.2.0
 [61] yaml_2.2.0 colorspace_1.4-1 cluster_2.1.0 rvest_0.3.4

... par une projection sur une ACP

```
par(mfrow = c(1,2))
biplot(prcomp(mes.iris), las = 1, cex = 0.7,
       main = "Données non normalisées")
biplot(prcomp(mes.iris, scale = TRUE), las = 1, cex = 0.7,
       main = "Données normalisées")
```



Cas d'étude : TCGA Breast Invasive Cancer (BIC)

- Présentation du cas d'étude (Jacques van Helden A COMPLETER)

TP : analyse de données d'expression

- TP clustering : [html] [pdf] [Rmd]
- Première partie : chargement des données

Contact: anne.badel@univ-paris-diderot.fr