Clustering

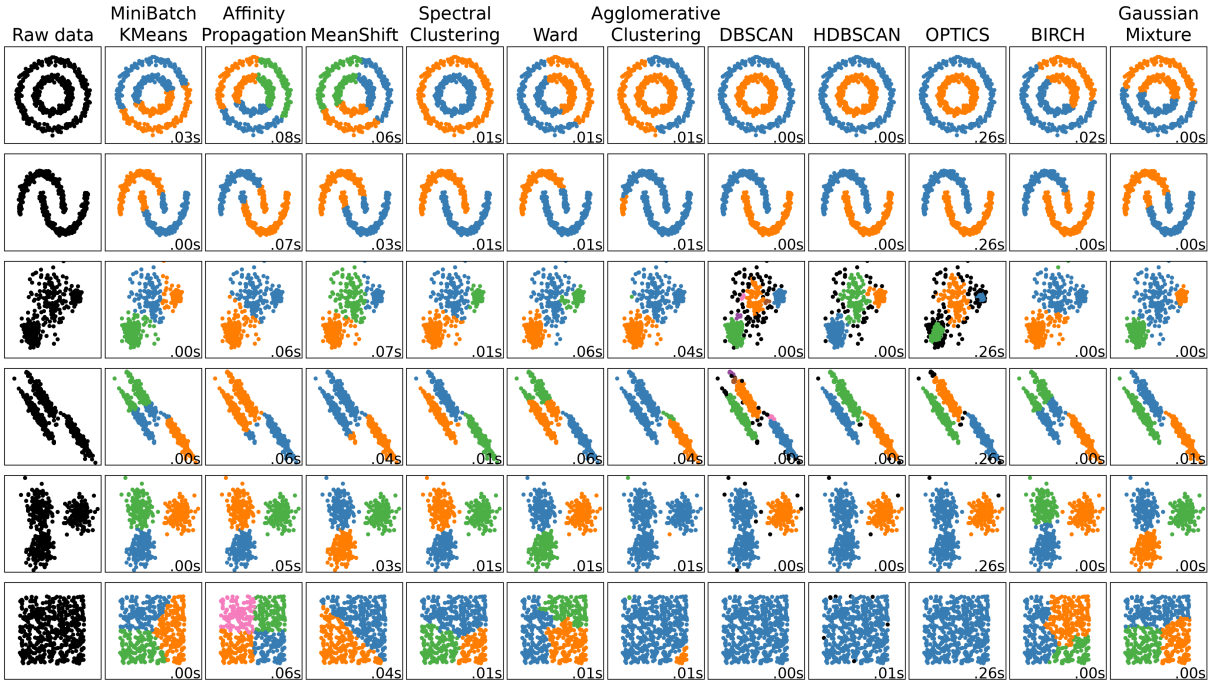
AUTEUR·RICE

Benjamin Charlier, Adrien Mazoyer

1 Objectifs

1.1 Panorama des méthodes

Rechercher dans un jeu de données une répartition des observations en différents groupes... Source: **documentation Scikit-learn**



1.2 Partition et hiérarchie

▸ **Définition**

- **Partition finie :** Une partition d'un ensemble fini E en K classes correspond à une famille de sous-ensembles $(\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(K)})$ non vides, deux à deux disjoints ($\mathcal{G}^{(i)} \cap \mathcal{G}^{(j)} = \emptyset$ pour $i \neq j$) et dont la réunion forme l'ensemble des individus ($\bigcup_{k=1}^{K} \mathcal{G}^{(k)} = E$).
- **Hiérarchie :** Une hiérarchie $\mathcal{H}$ sur un ensemble d'individus E est une famille de sous-ensembles non vides vérifiant :
  - $E \in \mathcal{H}$ (la classe réunissant tous les individus appartient à la hiérarchie),
  - $\forall x \in E, \{x\} \in \mathcal{H}$ (tous les singletons appartiennent à la hiérarchie),
  - $\forall (A, B) \in \mathcal{H}^2, A \cap B \in \{\emptyset, A, B\}$ (deux classes sont soit disjointes, soit incluses l'une dans l'autre).
- **Indice de hauteur :** Un indice $h : \mathcal{H} \to \mathbb{R}^+$ assigne une hauteur à chaque nœud de l'arbre (dendrogramme) en respectant la propriété de monotonie ($A \subset B \implies h(A) \leq h(B)$).

1.3 Principes généraux

Rechercher dans un jeu de données une partition des observations en différentes classes.

- **Problématiques centrales :**
  - Quelles observations regrouper au sein d'un même groupe ?
  - Comment régler les hyperparamètres (ex. nombre de groupes K, largeur de bande/fenêtre Δ, etc.) ?
  - Comportement en fonction du nombre de points, de clusters, de la dimension
- **Deux critères d'optimisation :**
  1. **Séparabilité inter-groupes :** les groupes doivent être les plus distincts possible les uns des autres.
  2. **Homogénéité intra-groupe :** chaque groupe doit être le plus homogène possible.

En pratique, nous évaluerons ces critères à l'aide de l'inertie : **minimiser l'inertie intra-groupe** (homogénéité) tout en **maximisant l'inertie inter-groupes** (séparabilité).

> ⚠ **Distinction essentielle**
>
> Ne pas confondre la **classification automatique** (*clustering* / apprentissage non supervisé) et la **classification supervisée**. Ici, nous ne disposons d'aucun étiquetage (*labels*) *a priori* sur les observations !

1.4 Contexte mathématique et notations

On dispose d'un ensemble de n observations $x_1, \ldots, x_n$ dans $\mathbb{R}^p$ (données quantitatives).

Chaque observation x_i est un vecteur de dimension p représenté par ses coordonnées :

$$x_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,p})^\top \in \mathbb{R}^p$$

L'objectif est de construire une partition de l'ensemble des observations $\{x_1, \dots, x_n\}$ en K groupes disjoints $\mathcal{G}^{(1)}, \dots, \mathcal{G}^{(K)}$.

Notations principales :

- **Centre de gravité global :** $g = \frac{1}{n} \sum_{i=1}^{n} x_i$
- **Effectif du groupe k :** $n_k = |\mathcal{G}^{(k)}| = \sum_{i=1}^{n} 1_{\{x_i \in \mathcal{G}^{(k)}\}}$
- **Centre de gravité du groupe k :** $g^{(k)} = \frac{1}{n_k} \sum_{x_i \in \mathcal{G}^{(k)}} x_i$

2 Distances et inerties

2.1 Distances entre observations

- **Distance euclidienne (norme L_2) :** $d_2(x_i, x_{i'}) = \sqrt{\sum_{j=1}^{p} (x_{i,j} - x_{i',j})^2}$
- **Distance de Manhattan (norme L_1) :** $d_1(x_i, x_{i'}) = \sum_{j=1}^{p} |x_{i,j} - x_{i',j}|$
- **Distance du maximum (norme L_∞) :** $d_\infty(x_i, x_{i'}) = \max_{j \in \{1,\dots,p\}} |x_{i,j} - x_{i',j}|$

Dans la suite du cours, la distance retenue sera la distance euclidienne : $d = d_2$

> ⓘ **Complexité spatiale et calculatoire**
>
> La première étape de nombreux algorithmes de partitionnement repose sur le calcul des distances deux à deux entre toutes les observations. Le stockage et la complexité algorithmique deviennent problématiques pour de grandes valeurs de n.

2.2 Décomposition de l'inertie

- **Inertie totale :** $\text{Inertie}_{\text{tot}} = \frac{1}{n} \sum_{i=1}^{n} d(x_i, g)^2$ avec g le barycentre de toutes les observations.
- **Inertie inter-classes :** $\text{Inertie}_{\text{inter}} = \sum_{k=1}^{K} \frac{n_k}{n} d(g^{(k)}, g)^2$
- **Inertie intra-classe :** $\text{Inertie}_{\text{intra}} = \frac{1}{n} \sum_{k=1}^{K} \sum_{x_i \in \mathcal{G}^{(k)}} d(x_i, g^{(k)})^2 = \sum_{k=1}^{K} \frac{n_k}{n} \text{Inertie}(\mathcal{G}^{(k)})$

Théorème de Huygens

L'inertie totale se décompose de manière unique en la somme des inerties intra-classe et inter-classes :

$$\text{Inertie}_{\text{tot}} = \text{Inertie}_{\text{intra}} + \text{Inertie}_{\text{inter}}$$

2.3 Inertie et qualité d'une partition

- **Conséquence directe :** L'inertie totale $\text{Inertie}_{\text{tot}}$ étant constante pour un jeu de données fixé, minimiser l'inertie intra-classe $\text{Inertie}_{\text{intra}}$ équivaut strictement à maximiser l'inertie inter-classes $\text{Inertie}_{\text{inter}}$.
- **Rapport d'inertie (Qualité du partitionnement) :** $Q = \dfrac{\text{Inertie}_{\text{inter}}}{\text{Inertie}_{\text{tot}}} = 1 - \dfrac{\text{Inertie}_{\text{intra}}}{\text{Inertie}_{\text{tot}}} \in [0, 1]$
- **Interprétation théorique :**
  - Si $Q = 0$, alors $g^{(k)} = g$ pour tout $k \in \{1, \dots, K\}$: la partition n'explique aucune variabilité.
  - Si $Q = 1$, alors $\text{Inertie}_{\text{intra}} = 0$: tous les individus au sein de chaque classe sont identiques ($x_i = g^{(k)}$).

Remarque importante

Le critère Q augmente mécaniquement avec le nombre de classes K. Pour $K = n$, chaque individu forme son propre groupe, d'où $\text{Inertie}_{\text{intra}} = 0$ et $Q = 1$, sans que la partition ne soit informative.

2.4 Distance entre groupes

- On considère une partition en K groupes $\mathcal{G}^{(1)}, \dots, \mathcal{G}^{(K)}$.
- On s'intéresse à la notion de dissimilarité ou de distance entre deux groupes $\mathcal{G}^{(k)}$ et $\mathcal{G}^{(k')}$, pour $k, k' \in \{1, \dots, K\}$.
- Chaque groupe $\mathcal{G}^{(k)}$ est pondéré par une masse w_k. Généralement

$$w_k = \frac{n_k}{n}$$

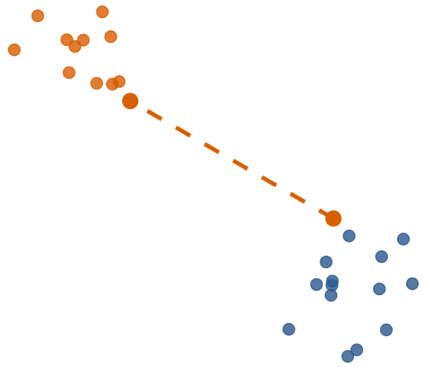
  avec $n_k = |\mathcal{G}^{(k)}|$ le nombre d'observations appartenant au groupe $\mathcal{G}^{(k)}$.
- Plusieurs métriques d'agrégation sont envisageables, chacune présentant des propriétés géométriques et topologiques distinctes.

2.5 Distances inter-groupes : saut simple et saut complet

- **Distance minimale (Saut simple) :**

$$d_{\min}(\mathcal{G}^{(k)}, \mathcal{G}^{(k')}) = \min_{x_i \in \mathcal{G}^{(k)}, x_{i'} \in \mathcal{G}^{(k')}} d(x_i, x_{i'})$$

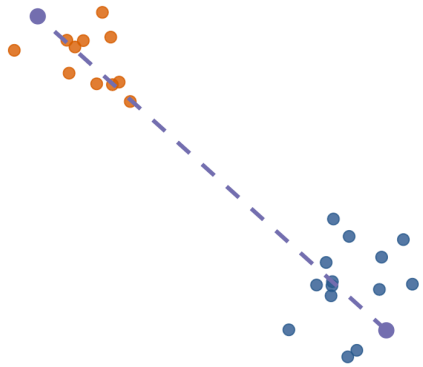
- *Avantages :* Capable de détecter des formes de classes non sphériques et irrégulières.
- *Inconvénients :* Sensible à l'effet de chaîne (*chaining effect*), tendance à créer de grands groupes allongés en fusionnant des sous-ensembles proches.



- **Distance maximale (Saut complet) :**

$$d_{\max}(\mathcal{G}^{(k)}, \mathcal{G}^{(k')}) = \max_{x_i \in \mathcal{G}^{(k)}, x_{i'} \in \mathcal{G}^{(k')}} d(x_i, x_{i'})$$

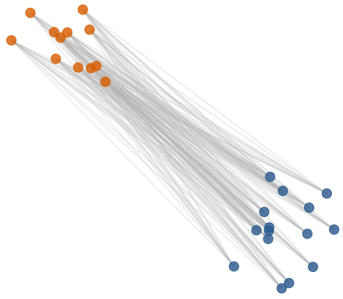
- *Avantages :* Produit des classes compactes et de diamètres homogènes.
- *Inconvénients :* Très sensible aux individus isolés (*outliers*).



2.6 Distances inter-groupes : moyenne et barycentrique

- **Distance moyenne (Average linkage) :**

$$d_{\text{mean}}(\mathcal{G}^{(k)}, \mathcal{G}^{(k')}) = \frac{1}{n_k n_{k'}} \sum_{x_i \in \mathcal{G}^{(k)}} \sum_{x_{i'} \in \mathcal{G}^{(k')}} d(x_i, x_{i'})$$

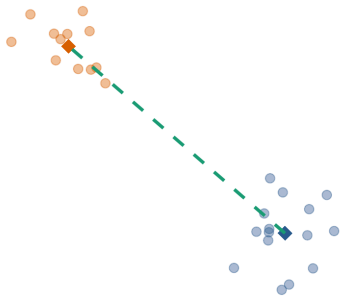


- Prend en compte l'ensemble des paires d'observations entre les deux groupes.
- Compromis robuste entre la distance minimale et la distance maximale.

- **Distance barycentrique (Centroid linkage) :**

$$d_{\text{bar}}(\mathcal{G}^{(k)}, \mathcal{G}^{(k')}) = d(g^{(k)}, g^{(k')})$$

avec $g^{(k)}$ le centre de gravité du groupe $\mathcal{G}^{(k)}$.



- *Avantages :* Simple à calculer et à interpréter, moins sensible aux observations isolées (*outliers*).
- *Inconvénients :* Peut présenter des inversions dans le dendrogramme (non-monotonie de l'indice de hauteur).

2.7 Algorithme de la Classification Ascendante Hiérarchique (CAH)

On choisit une distance entre observations d et un critère d'agrégation entre groupes δ.

- **Initialisation :** Chaque individu x_i forme son propre groupe : $\mathcal{P}_n = \{\mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(n)}\}$ avec $\mathcal{G}^{(i)} = \{x_i\}$.
- **Processus d'agrégation itératif :** À l'étape $m \in \{1, \ldots, n-1\}$:
  1. *Recherche :* de la paire $(\mathcal{G}^{(A)}, \mathcal{G}^{(B)})$ minimisant la distance inter-groupes $\delta(\mathcal{G}^{(A)}, \mathcal{G}^{(B)})$.
  2. *Fusion :* des deux groupes en un seul groupe $\mathcal{G}^{(A \cup B)} = \mathcal{G}^{(A)} \cup \mathcal{G}^{(B)}$.
  3. *Mise à jour :* des distances entre le nouveau groupe et l'ensemble des groupes restants.
- **Condition d'arrêt :** obtention d'un groupe unique contenant toutes les observations ($\mathcal{P}_1 = \{E\}$).

> ⓘ **Choix de la partition finale**
>
> Le choix du nombre final de classes K s'effectue *a posteriori* en coupant le dendrogramme à la hauteur souhaitée (souvent identifiée par le saut d'inertie le plus marqué).

2.8 Méthode de Ward et augmentation d'inertie

- **Initialisation :** on a n groupes, chacun contenant un individu.

- **Itérations :** on évalue à chaque étape l'augmentation de l'inertie intra en fusionnant deux groupes $\mathcal{G}^{(k)}$ et $\mathcal{G}^{(k')}$, qui s'exprime par :

$$\text{Inertie}(\mathcal{G}^{(k)} \cup \mathcal{G}^{(k')}) = \text{Inertie}(\mathcal{G}^{(k)}) + \text{Inertie}(\mathcal{G}^{(k')}) + \underbrace{\frac{n_k n_{k'}}{n_k + n_{k'}} d(g^{(k)}, g^{(k')})^2}_{\Delta\text{Inertie}_{\text{intra}}}$$

  On fusionne donc en premier les groupes qui minimisent l'augmentation de l'inertie intra-classe.

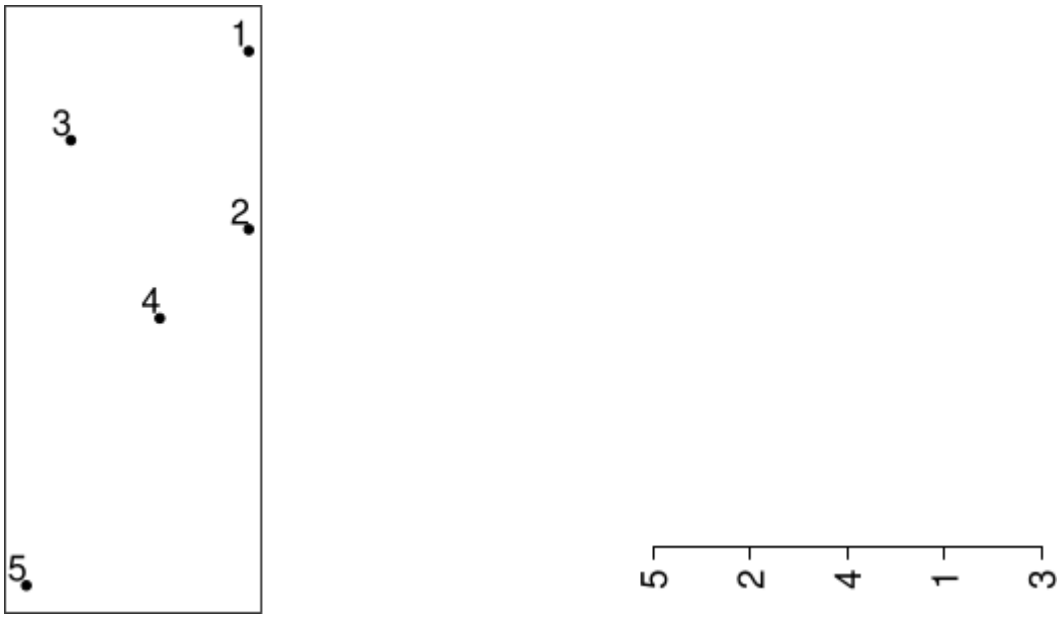
Distance de Ward entre deux groupes

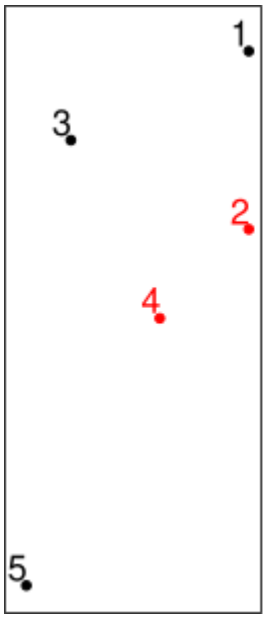
$$d_W(\mathcal{G}^{(k)}, \mathcal{G}^{(k')})^2 = \frac{w_k w_{k'}}{w_k + w_{k'}} d(g^{(k)}, g^{(k')})^2$$

avec $w_k = \frac{n_k}{n}$ et $w_{k'} = \frac{n_{k'}}{n}$ les poids de $\mathcal{G}^{(k)}$ et $\mathcal{G}^{(k')}$.

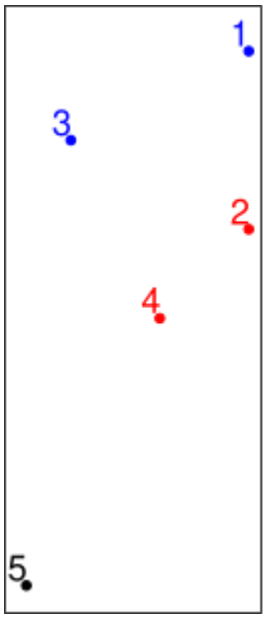
- **Propriété :** on minimise la perte d'inertie inter-classes garantissant des classes les plus homogènes possible à chaque étape.

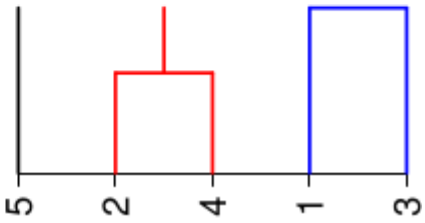
2.9 Illustration

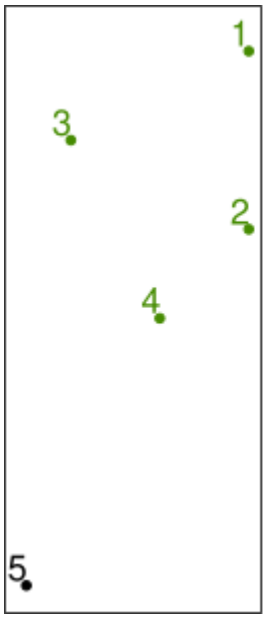


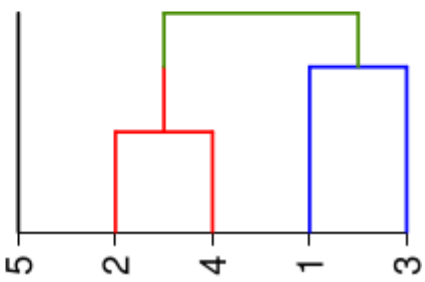

1
3
2
4
5

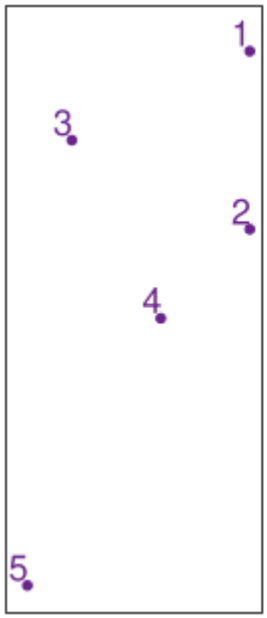

5
2
4
1
3

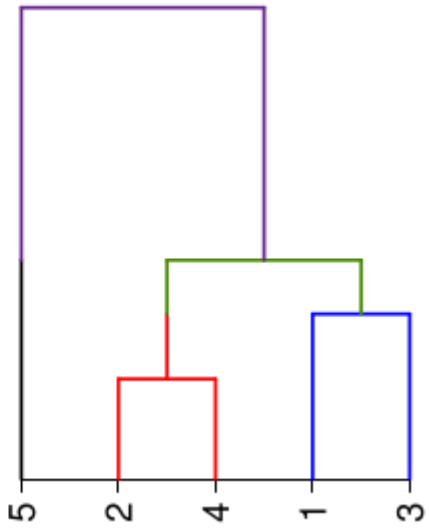

1
3
2
4
5


5
2
4
1
3


1
3
2
4
5

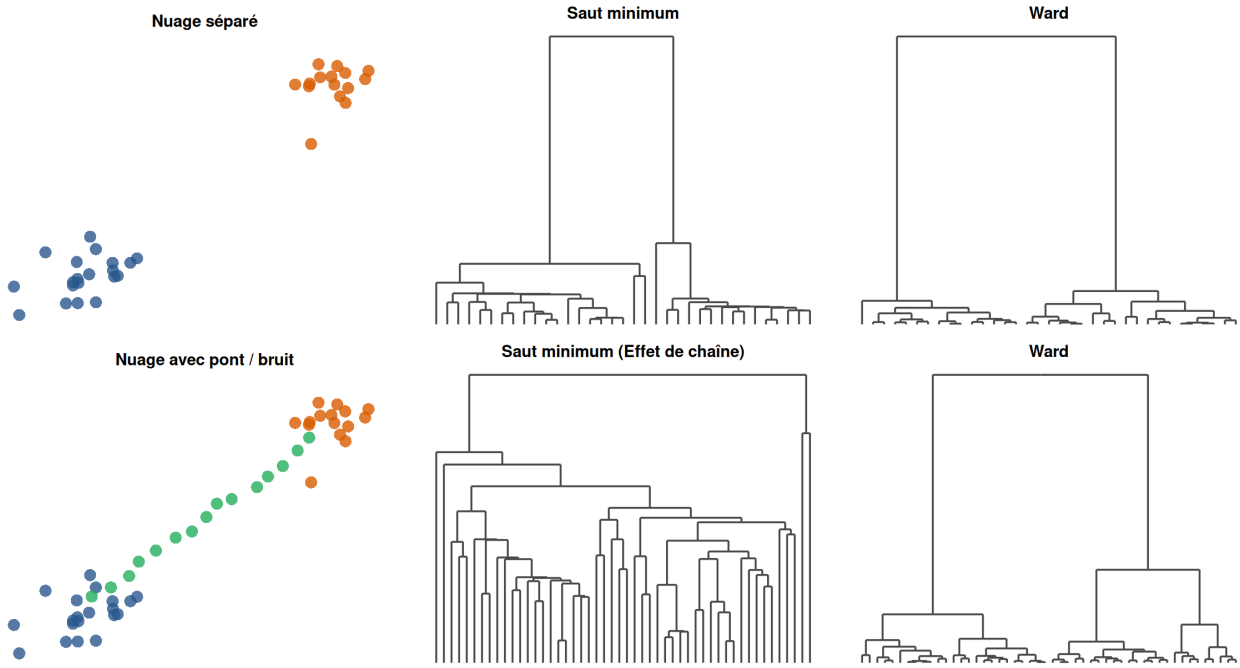

5
2
4
1
3





2.10 Propriétés de la distance de Ward

Favorise l'agrégation de classes de faibles poids et casse l'effet chaine du lien minimum.



2.11 Exemple

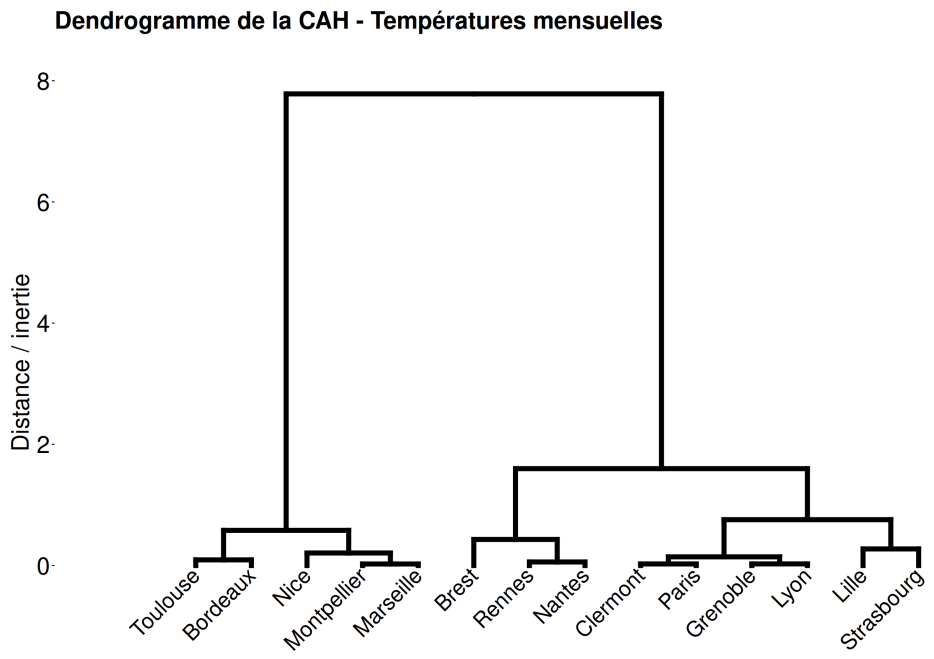
On reprend l'exemple utilisé pour l'ACP (les données ont déjà été normalisées)

Températures mensuelles centrées-réduites

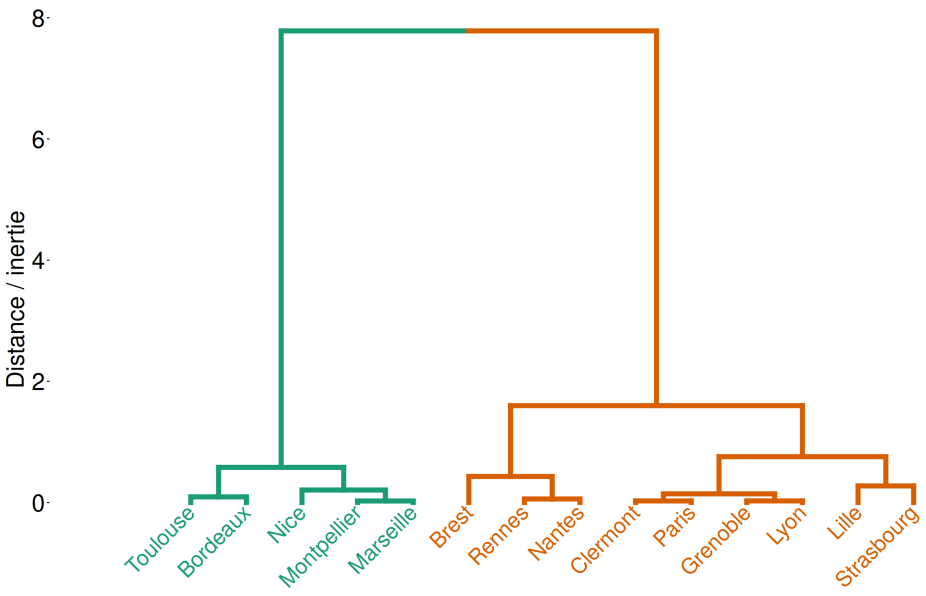
	Janv	Févr	Mars	Avri	Mai	Juin	juil	Août	Sept	Octo	Nove	Déce
Bordeaux	0.77	0.91	1.33	1.26	0.88	0.79	0.48	0.69	0.84	0.77	0.61	0.65
Brest	1.03	0.47	-0.34	-1.34	-1.95	-1.96	-2.01	-1.81	-1.29	-0.23	0.55	1.07
Clermont	-0.76	-0.68	-0.54	-0.55	-0.47	-0.33	-0.22	-0.26	-0.47	-0.68	-0.80	-0.70
Grenoble	-1.32	-0.95	-0.41	-0.33	0.00	-0.05	0.11	-0.06	-0.20	-0.57	-0.86	-1.38
Lille	-0.86	-1.11	-1.55	-1.56	-1.41	-1.45	-1.30	-1.26	-1.29	-1.12	-1.09	-0.76
Lyon	-1.01	-0.90	-0.41	-0.11	0.27	0.34	0.39	0.24	-0.09	-0.57	-0.75	-0.96
Marseille	0.72	0.91	1.13	1.40	1.55	1.64	1.61	1.59	1.56	1.44	1.23	1.02
Montpellier	0.77	0.97	1.06	1.26	1.15	1.24	1.33	1.34	1.23	1.22	1.12	0.81
Nantes	0.47	0.20	0.06	-0.19	-0.40	-0.38	-0.50	-0.51	-0.36	-0.12	0.10	0.29
Nice	1.74	1.95	1.66	1.62	1.48	1.24	1.33	1.44	1.77	2.00	1.97	1.70
Paris	-0.35	-0.46	-0.48	-0.26	-0.13	-0.22	-0.36	-0.46	-0.58	-0.57	-0.52	-0.34
Rennes	0.36	0.20	-0.28	-0.69	-0.94	-0.95	-0.93	-0.91	-0.74	-0.45	-0.13	0.23
Strasbourg	-1.88	-1.88	-1.81	-0.91	-0.33	-0.38	-0.41	-0.66	-1.07	-1.63	-1.77	-1.90
Toulouse	0.31	0.36	0.59	0.39	0.27	0.46	0.48	0.64	0.68	0.49	0.33	0.29

Rappel : si on emploie la distance euclidienne on a que $\text{Inertie}_{\text{tot}} = 12$

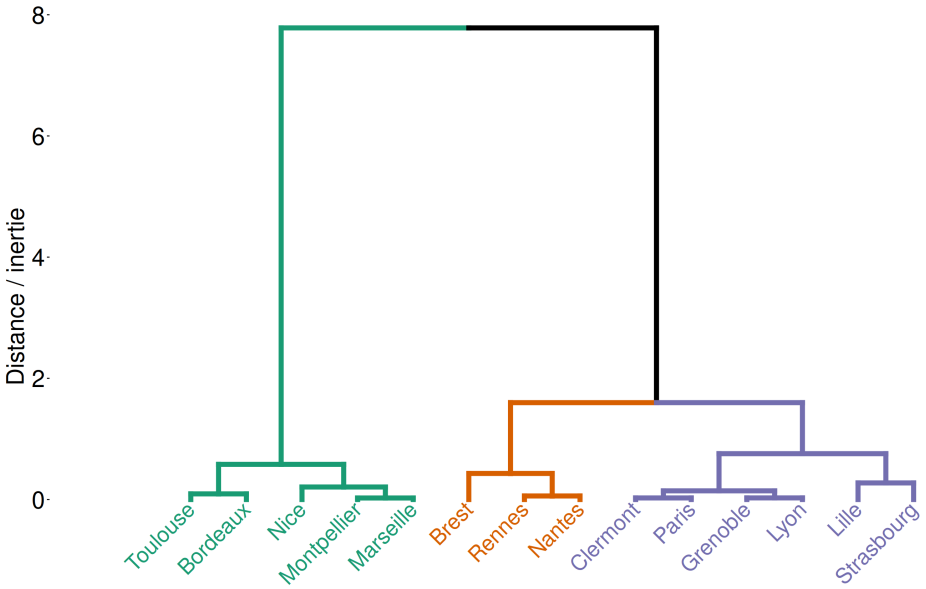
2.12 Exemple

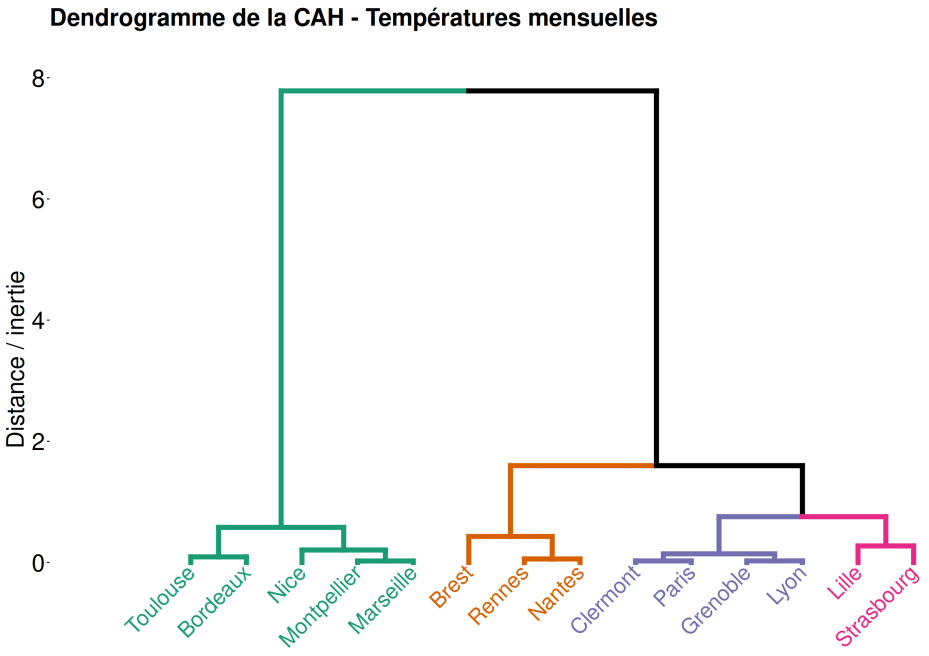


Dendrogramme de la CAH - Températures mensuelles



Dendrogramme de la CAH - Températures mensuelles





2.13 Évolution de l'inertie inter-classe

On trace l'évolution du gain d'inertie inter-classe lors qu'on passe de k à $k+1$ groupes, c'est à dire le saut de Ward associé



niveau	inertie	pct.cumul.inertie
1 → 2	7.783	64.9
2 → 3	1.598	78.2
3 → 4	0.756	84.5
4 → 5	0.580	89.3
5 → 6	0.430	92.9
6 → 7	0.273	95.2
7 → 8	0.206	96.9
8 → 9	0.143	98.1
9 → 10	0.093	98.9
10 → 11	0.058	99.4
11 → 12	0.026	99.6
12 → 13	0.026	99.8
13 → 14	0.025	100.0

2.14 Choix du nombre K de groupes

2.14.1 Critères d'aide à la décision

- **Analyse du dendrogramme :**
  On cherche la coupure qui sépare les branches les plus longues (hauteurs d'agrégation maximales).
- **Saut d'inertie inter-classe ($\Delta \text{Inertie}_{\text{inter}}$) :**
  On conserve les fusions qui apportent un gain d'inertie significatif (par exemple, un gain $> 10\%$ de l'inertie totale $\text{Inertie}_{\text{tot}}$).
- **Taille de l'échantillon n et variabilité :**
  Un effectif important ou une forte hétérogénéité des données justifient un nombre de classes K plus élevé.
- **Interprétabilité :**
  La partition retient sa pertinence si chaque classe possède une signature métier claire.

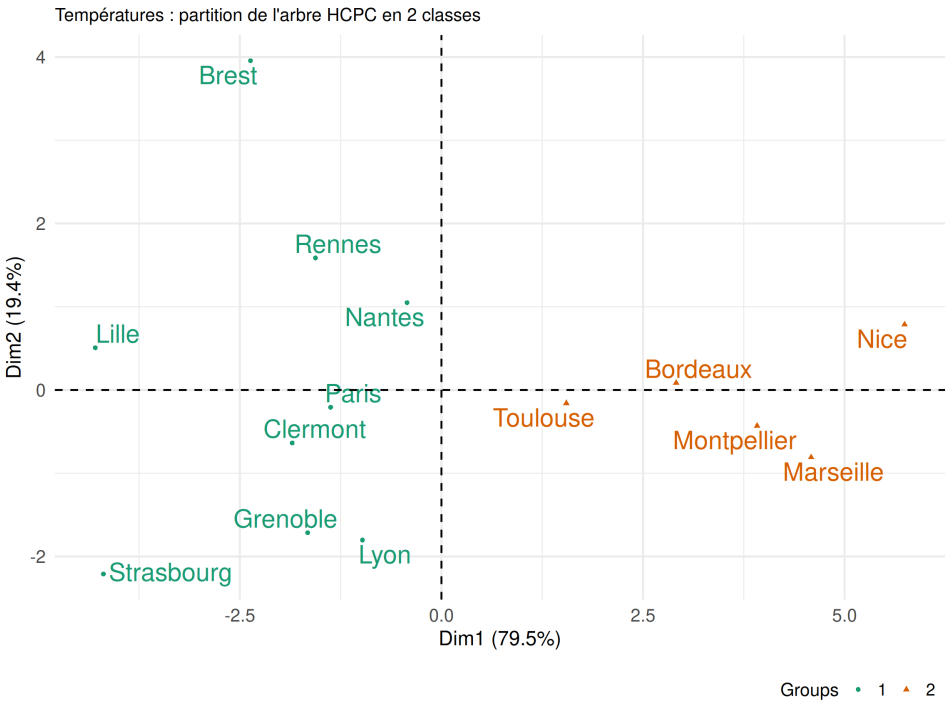
Règle du coude

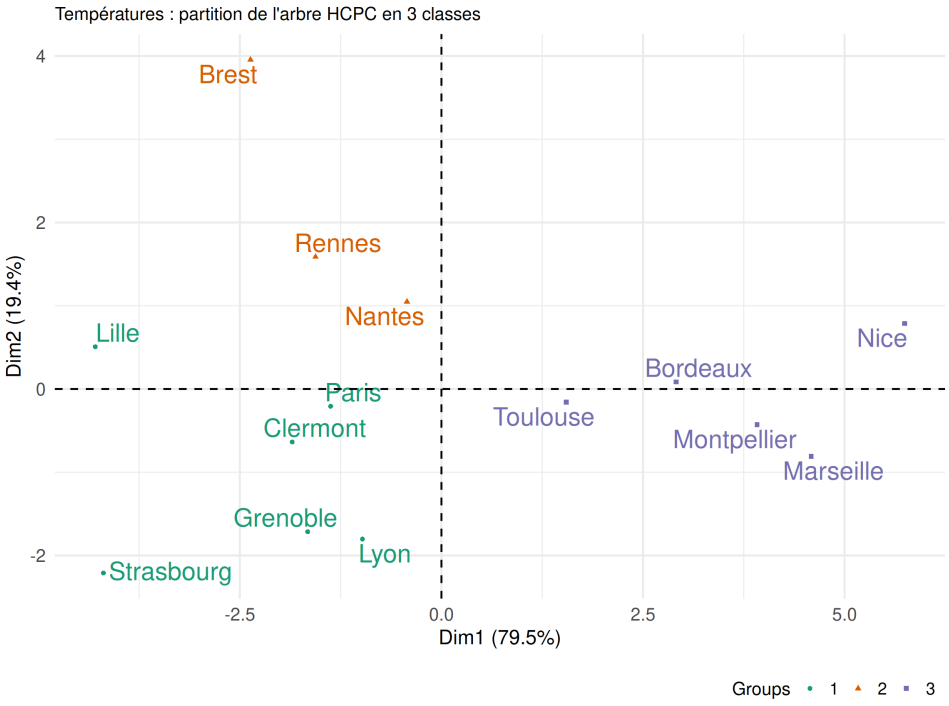
En traçant le gain d'inertie en fonction du nombre de groupes, on recherche un « coude » caractérisant une chute brutale du gain marginal d'inertie.

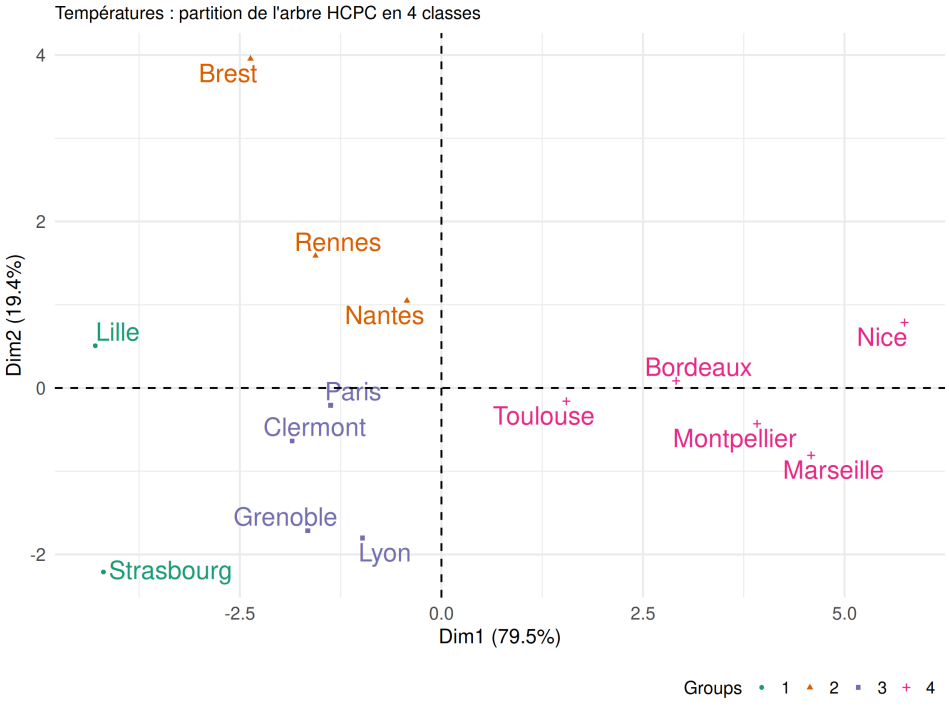
⚠ Arbitrage

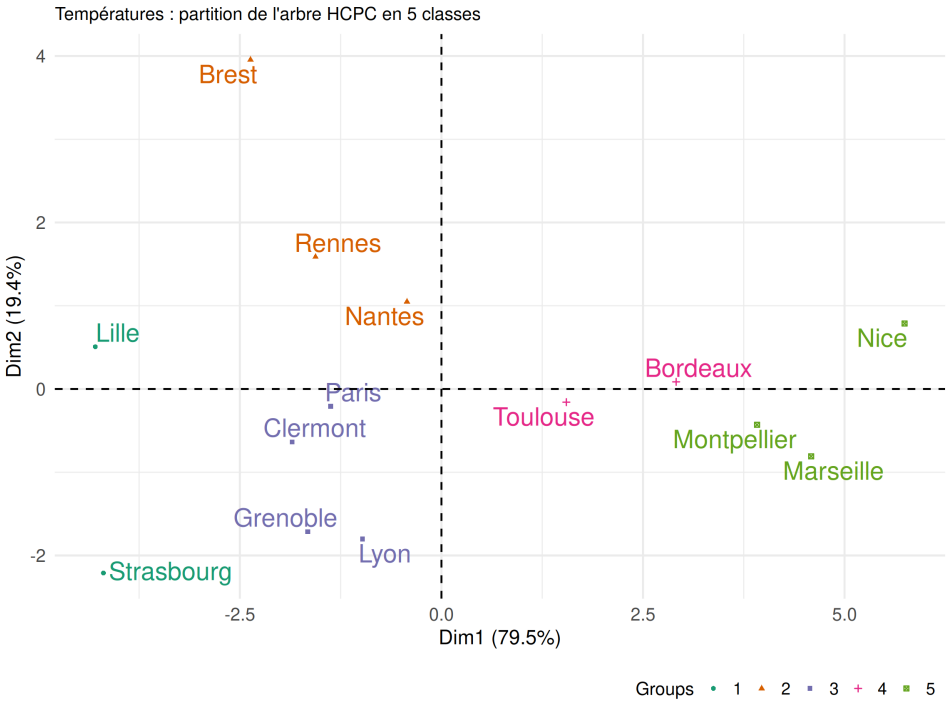
Un nombre de groupes trop faible simplifie à l'excès la structure des données ; un nombre trop élevé conduit à du sur-ajustement (*overfitting*).

2.15 Lien avec l'ACP




Températures : partition de l'arbre HCPC en 3 classes
Brest
Rennes
Nantes
Lille
Nice
Bordeaux
Paris
Toulouse
Clermont
Montpellier
Marseille
Grenoble
Lyon
Strasbourg
4
2
0
-2
-2.5
0.0
2.5
5.0
Dim2 (19.4%)
Dim1 (79.5%)
Groups 1 2 3


Températures : partition de l'arbre HCPC en 4 classes
Brest
Rennes
Nantes
Lille
Nice
Bordeaux
Paris
Toulouse
Clermont
Montpellier
Marseille
Grenoble
Lyon
Strasbourg
4
2
0
-2
-2.5
0.0
2.5
5.0
Dim2 (19.4%)
Dim1 (79.5%)
Groups 1 2 3 4



2.16 Implémentation logicielle de la CAH

2.16.1 En R

Plusieurs fonctions complémentaires selon l'usage :

- `hclust()` *(package* `stats`*)* :
  L'implémentation standard en R de base.
  (Alternative plus rapide pour grands jeux de données : `fastcluster::hclust`*)*
- `hcut()` *(package* `factoextra`*)* :
  Enveloppe autour de `hclust()`, simplifiant l'extraction et la visualisation directe des dendrogrammes avec `ggplot2`.
- `HCPC()` *(package* `FactoMineR`*)* :
  Conçue pour enchaîner une analyse factorielle (ACP/ACM) et une CAH, offrant des aides à la décision automatisées et une caractérisation des classes.

2.16.2 En Python

Deux bibliothèques principales :

- `scipy.cluster.hierarchy` :
  Offre les fonctions fondamentales pour calculer les matrices de liaison (`linkage`), tracer le dendrogramme (`dendrogram`) et effectuer les coupes (`fcluster`).
- `sklearn.cluster.AgglomerativeClustering` *(scikit-learn)* :
  L'API orientée ***Machine Learning*** intégrant `fit()` et `fit_predict()`, idéale pour s'insérer dans un ***pipeline*** de modélisation.

3 Méthode des K means

3.1 Algorithme des K-means (Centroïdes / Nuées dynamiques)

Algorithme de réallocation dynamique (Algorithme de Lloyd)

1. **Initialisation ($t = 0$) :**
   Fixer le nombre de classes K.
   Tirer aléatoirement K observations distinctes pour servir de centres initiaux : $\{\mu_1^{(0)}, \ldots, \mu_K^{(0)}\}$.
2. **Étape d'affectation (Recherche des voisins) :**
   Affecter chaque observation x_i au centre le plus proche :

$$\mathcal{G}_k^{(t)} = \left\{ x_i \;\middle|\; d(x_i, \mu_k^{(t)})^2 \leq d(x_i, \mu_\ell^{(t)})^2 \quad \forall \ell \in \{1, \ldots, K\} \right\}$$

3. **Étape de mise à jour (Recalcul des noyaux) :**
   Mettre à jour le centroïde de chaque groupe :

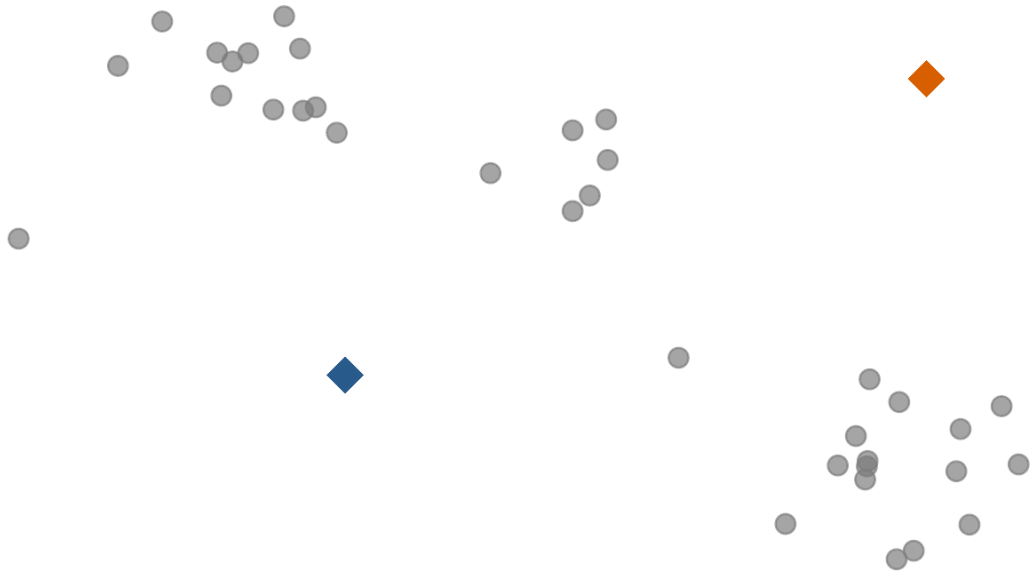
$$\mu_k^{(t+1)} = \frac{1}{|\mathcal{G}_k^{(t)}|} \sum_{x_i \in \mathcal{G}_k^{(t)}} x_i$$

4. **Critère d'arrêt :**
   Répéter jusqu'à la stabilisation des partitions ($\mathcal{G}_k^{(t+1)} = \mathcal{G}_k^{(t)}$) ou la convergence de l'inertie intra-classe.

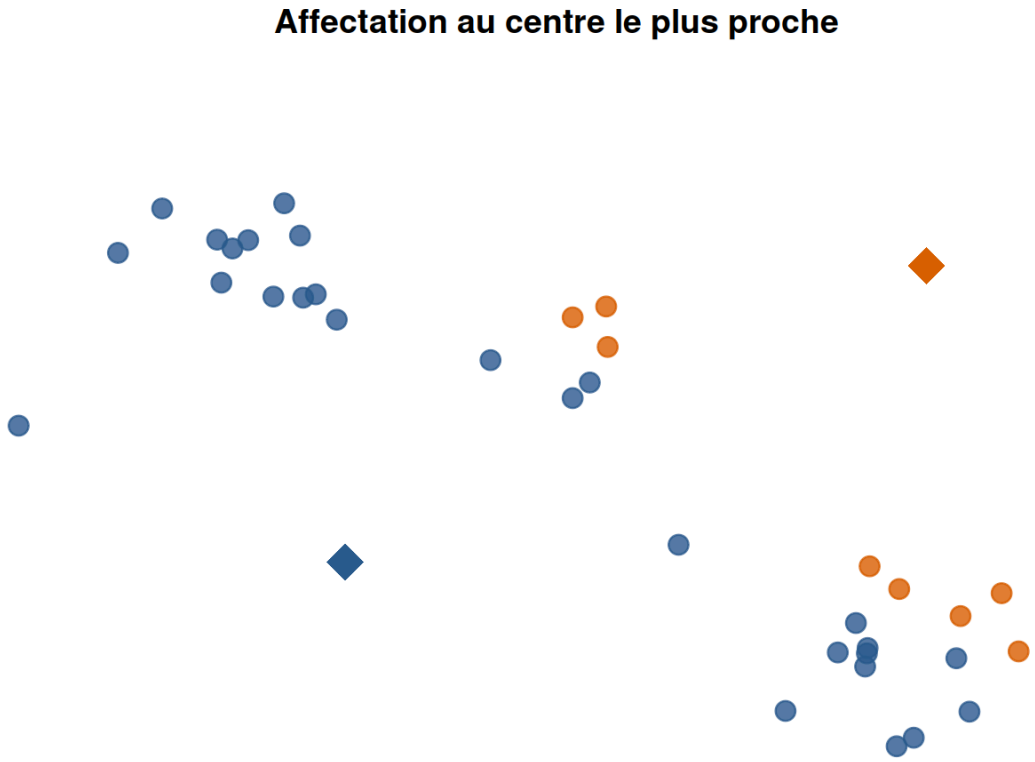
3.2 Illustration de l'algorithme des K-means

3.2.1 Initialisation

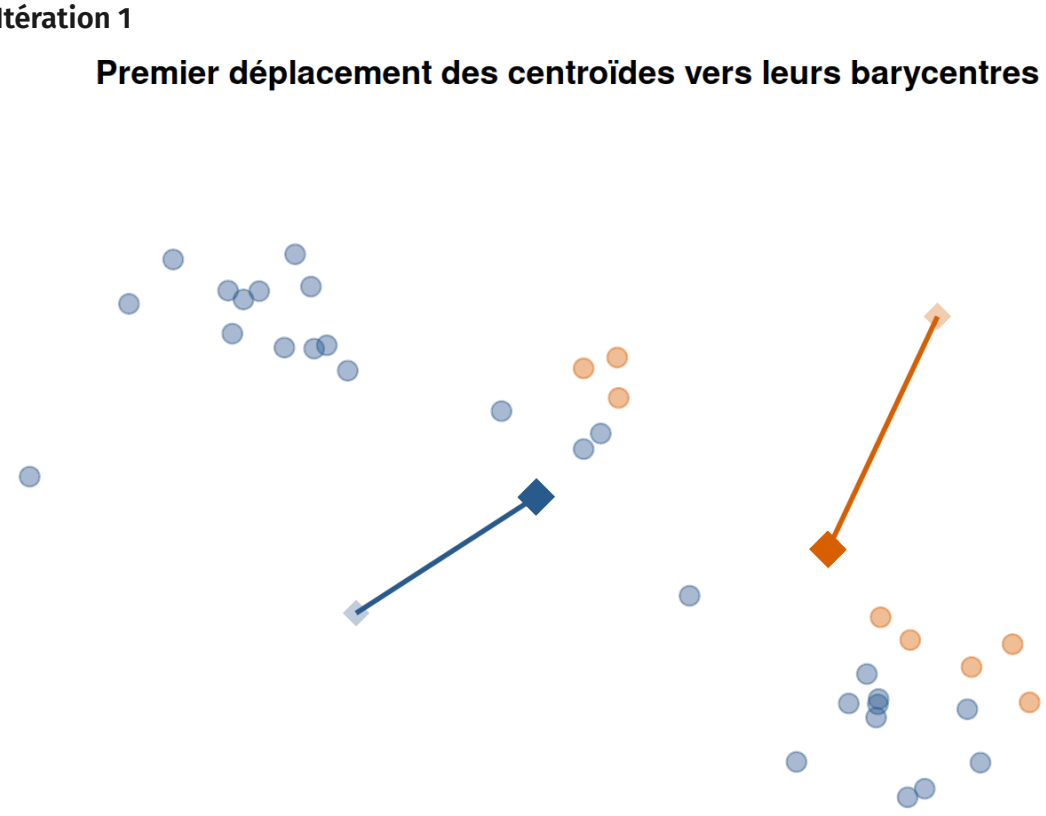
Choix aléatoire des centroïdes initiaux



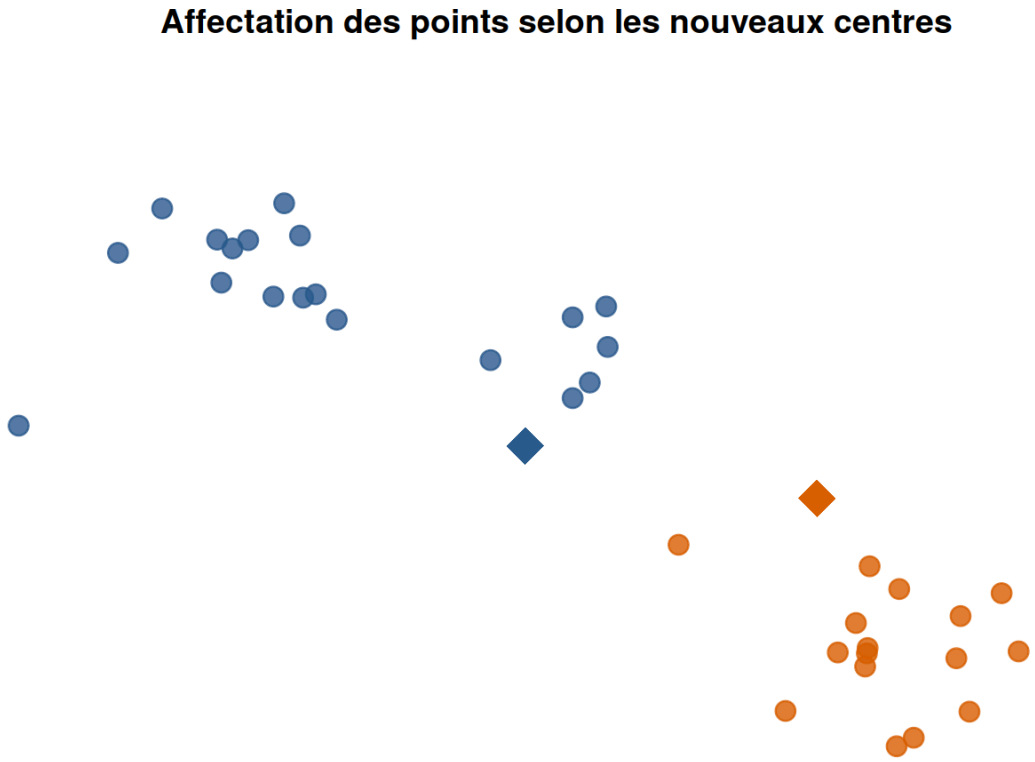
3.2.2 Itération 1



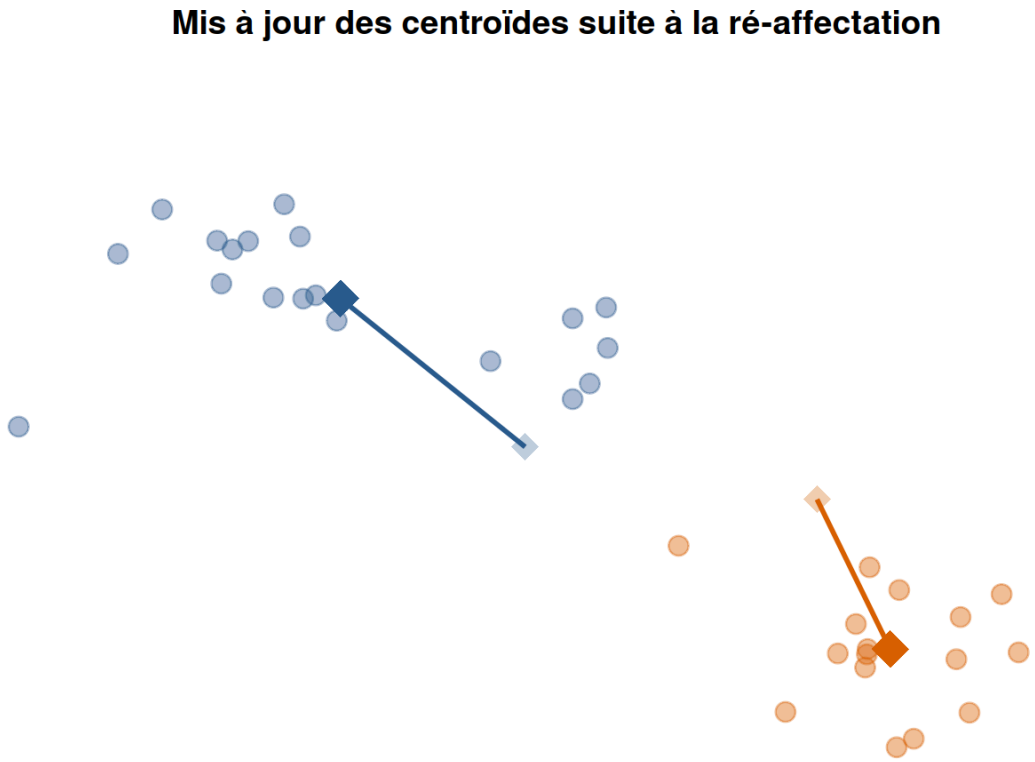
3.2.3 Itération 1



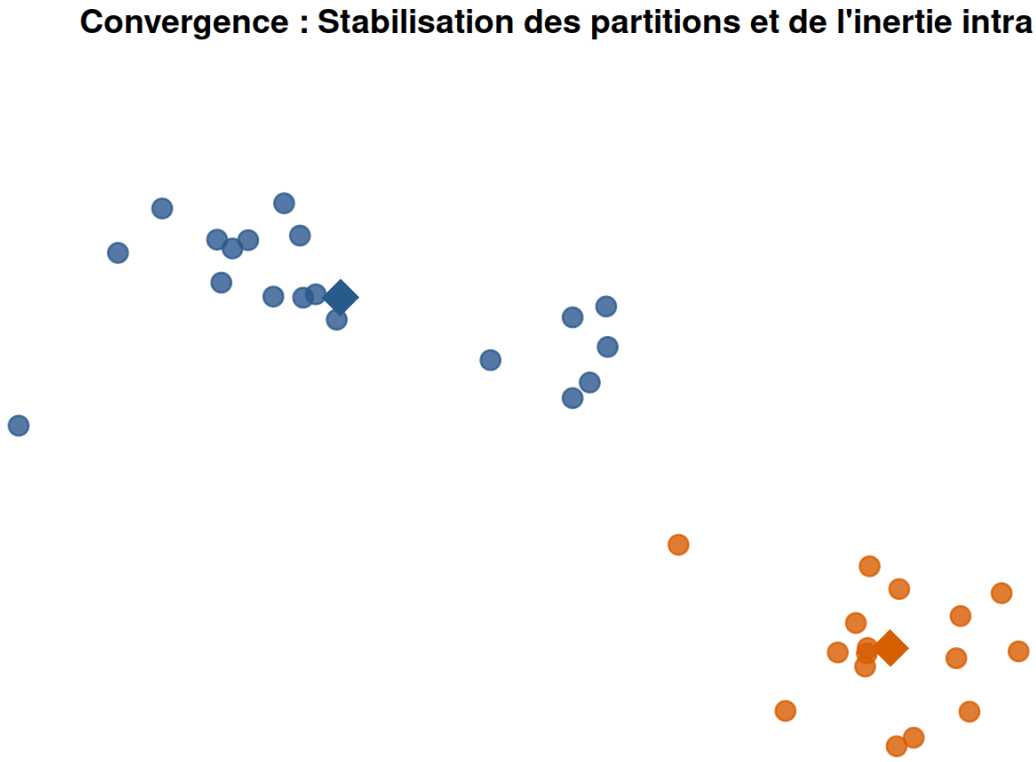
3.2.4 Itération 2



3.2.5 Itération 2



3.2.6 Arrêt de l'algorithme



3.3 Remarque

- **Sensibilité à l'initialisation :** L'algorithme converge toujours vers un optimum **local** de l'inertie intra-classe. Le partitionnement obtenu dépend du tirage aléatoire initial.
- **Bonne pratique :** Exécuter l'algorithme plusieurs fois (argument `nstart` en R) et retenir la solution minimisant l'inertie intra. Sélectionne des centroïdes initiaux de manière en favorisant des points éloignés les uns des autres.
- **Autre algorithme :** Algorithme de Hartigan-Wong

3.4 Synthèse : CAH vs. K-means

	CAH	K-means
Avantages	**+ Choix de** K ***a posteriori*** une fois l'arbre (dendrogramme) construit	**+ Algorithme rapide** et adapté aux grands jeux de données
	+ Traçabilité de l'ensemble des niveaux d'agrégation	**+ Réallocation dynamique :** un point peut changer de groupe au fil des itérations
Inconvénients	**- Numériquement coûteux** ($O(n^2)$ à $O(n^3)$)	**- Choix de** K **obligatoire** *a priori*
	- Irréversibilité : des observations une fois regroupées le restent	**- Instabilité :** dépend du choix des noyaux initiaux (minima locaux)
	- Sensibilité : dépend de la métrique et du critère d'agrégation	**- Contrainte géométrique :** tend à privilégier des classes sphériques / convexes

Quand privilégier la CAH ?

- Petit à moyen échantillon ($n \leq 1000$)
- Besoin d'explorer la structure hiérarchique globale des données
- Incertitude totale sur le nombre pertinent de groupes K

Quand privilégier les K-means ?

- Grand jeu de données (n élevé)
- Volonté d'effectuer une hybridation (ex. pré-clustering avant CAH)
- Nombre de classes K pressenti ou contraint par des impératifs métier

3.5 Combinaison des deux méthodes (Hybridation)

L'association de la CAH et du K-means permet de surmonter leurs limites respectives

3.5.1 Initialisation des K-means par la CAH

Permet d'éviter le tirage aléatoire des centroïdes initiaux et les minima locaux.

1. **Choix de K :** Exécuter une CAH et déterminer le nombre optimal de groupes K via le dendrogramme.
2. **Extraction des noyaux :** Calculer le barycentre de chacun des K groupes obtenus par la CAH.
3. **Initialisation RAD :** Lancer les K-means en utilisant ces K barycentres comme centres initiaux.

3.5.2 Pré-clustering pour grands jeux de données

Rend la CAH applicable en réduisant le nombre d'individus à traiter.

1. **Réduction (K élevé) :** Appliquer un K-means pour diviser les n observations en un nombre élevé de micro-groupes K
2. **Agrégation hiérarchique :** Appliquer la CAH directement sur les K **noyaux** des micro-groupes (pondérés par leurs effectifs).
3. **Partition finale :** Couper l'arbre pour obtenir la partition ultime.

4 Cas des données qualitatives et mixtes

4.1 Extension aux données qualitatives

Il est assez simple d'étendre les méthodes présentées dans le cas quantitatif au cas où les données sont qualitatives.

4.1.1 1. Partitionnement à partir du TDC

Le partitionnement repose sur la définition d'une distance entre les observations.

- Il suffit donc de considérer le **Tableau Disjonctif Complet (TDC)**
- On utilise ensuite la **distance du χ^2** (voir chapitre sur l'ACM)
- On applique enfin la **méthode de Ward** au tableau de distances ainsi obtenu

4.1.2 2. Partitionnement après ACM

Effectuer une **ACM** (Analyse Correspondances Multiples) permet de transformer un TDC en un tableau de données quantitatives (coordonnées des observations sur les axes principaux).

- On peut donc appliquer **n'importe quelle méthode de partitionnement** (CAH ou K-means) sur les variables synthétiques déduites de l'ACM.

4.2 Extension aux données mixtes

Lorsqu'on dispose de données mixtes (variables quantitatives et qualitatives), deux stratégies principales permettent de ramener le problème à un cadre homogène.

4.2.1 1. Rendre tout quantitatif

1. **Analyse factorielle :** Réaliser une **AFDM** (Analyse Factorielle de Données Mixtes), une ACM sur les variables qualitatives, ou une ACM sur l'ensemble des variables après discrétisation des variables quantitatives.
2. **Partitionnement :** Appliquer la méthode de classification sur les coordonnées des observations sur les axes principaux.

4.2.2 2. Rendre tout qualitatif

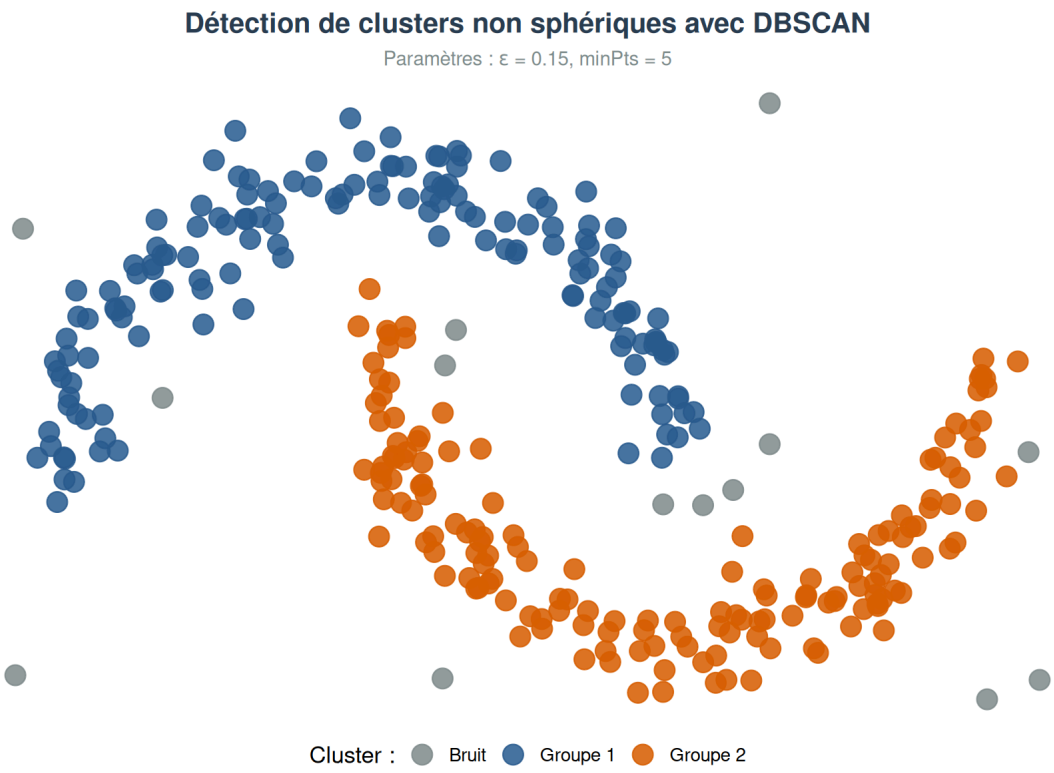
1. **Discrétisation :** Transformer l'ensemble des variables quantitatives en variables qualitatives ordinales (découpage en classes/quantiles).
2. **Partitionnement :** Effectuer la classification à partir de la distance du χ^2 appliquée au Tableau Disjonctif Complet (TDC) résultant.

5 Algorithme DBSCAN

5.1 En résumé

DBSCAN : *Density-Based Spatial Clustering of Applications with Noise*

- **Un cluster :** une zone caractérisée par une forte densité d'observations.
- **Paramètres fondamentaux :**
  - Un rayon $\varepsilon > 0$ de voisinage autour d'une observation
  - Un nombre minimal de points dans ce voisinage pour former un cluster (`minPts`)



5.2 Ingrédients de l'algorithme

- **ε-voisinage :** Pour $\varepsilon > 0$, le voisinage d'une observation $x \in \mathbb{R}^p$ est l'ensemble des points à une distance strictement inférieure à ε :

$$N_\varepsilon(x) = \{y \in \mathbb{R}^p \mid d(x, y) < \varepsilon\}$$

- **Point cœur :** Pour un paramètre $N_{\min}$ fixé, x est un **point cœur** si son ε-voisinage est suffisamment dense :

$$|N_\varepsilon(x)| \geq N_{\min}$$

- **Connexion par densité :** Deux points x et z sont connectés s'ils sont reliés par une chaîne de points cœurs $(y_0 = x, \ldots, y_{m+1} = z)$, où chaque point appartient au voisinage du précédent :

$$y_i \in N_\varepsilon(y_{i-1})$$

5.3 Algorithme DBSCAN

1. Initialisation - Sélectionner au hasard un point x non déjà visité - Déterminer son ε-voisinage $N_\varepsilon(x)$

2. Évaluation et expansion - **Si $|N_\varepsilon(x)| < N_{\min}$:** - x est étiqueté comme **bruit / anomalie** (classe 0). - **Sinon (x est un point cœur) :** - Création d'un nouveau cluster k. - x et tous ses voisins $y \in N_\varepsilon(x)$ reçoivent l'étiquette k. - **Propagation par densité :** Pour chaque point y ajouté, on évalue son propre voisinage $N_\varepsilon(y)$. S'il contient au moins $N_{\min}$ points, ses voisins rejoignent également le cluster k.

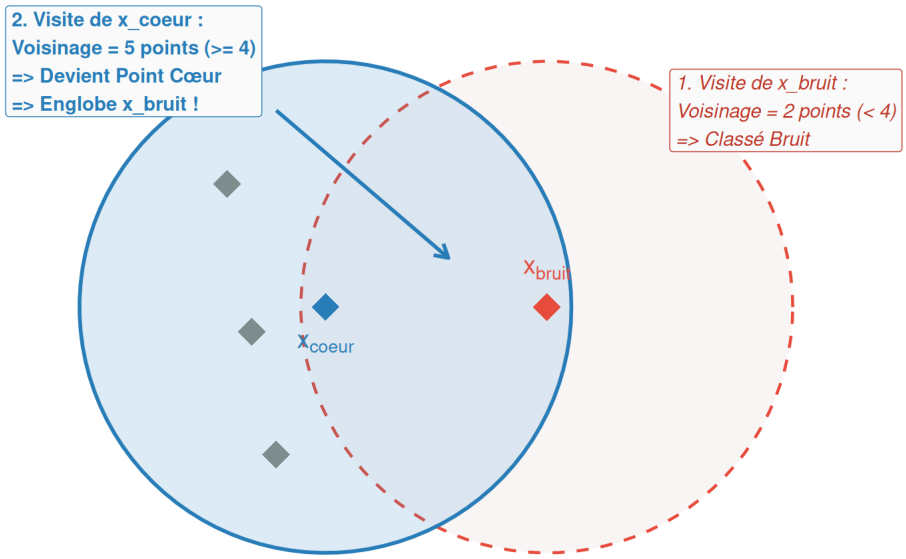
3. Itération - Répéter les étapes 1 et 2 pour chaque point non encore visité, jusqu'à ce que l'ensemble du jeu de données ait été exploré.

5.4 Subtilité de l'algorithme : repêchage du bruit

> ⚠ **Réaffectation dynamique**
>
> Un point initialement étiqueté comme **bruit** lors de l'exploration algorithmique peut finalement être rattaché à un autre cluster en tant que **point frontière**.

Mécanisme illustré ($N_{\min} = 4$) :

- Le point x_{bruit} est visité en premier. Son ε-voisinage ne contient pas assez de points. Il est provisoirement étiqueté comme **bruit**.
- Plus tard, l'algorithme visite $x_{\text{cœur}}$. Son voisinage est suffisamment dense ($\geq N_{\min}$) : c'est un **point cœur** qui forme un cluster.
- $\Rightarrow$ Puisque x_{bruit} appartient au voisinage de $x_{\text{cœur}}$, il est repêché et intègre ce nouveau cluster !



5.5 Commentaires et synthèses sur DBSCAN

- **Avantages :**
  - Peu coûteux numériquement (tant que la dimension p n'est pas trop grande).
  - Robuste aux données aberrantes (*outliers* / bruit).
  - Capable de repérer des formes de groupes très variées (non sphériques).
- **Inconvénients :**
  - Sensible au fléau de la dimensionnalité (peu adapté aux grandes dimensions).
  - Sensibilité au choix des hyperparamètres :
    - **Choix de ε :**

- ε trop grand $\Rightarrow$ risque de ne détecter qu'un unique cluster géant.
- ε trop faible $\Rightarrow$ surdétection d'anomalies (trop de bruit).

- **Choix de $N_{\min}$:**
  - $N_{\min}$ trop grand $\Rightarrow$ seuls les clusters extrêmement denses sont identifiés.
  - $N_{\min}$ trop faible $\Rightarrow$ fragmentation en de nombreux clusters peu denses.

5.6 Choix des paramètres

