METHODES DE CLUSTERING
Méthode des kmeans

Sébastien Quinault / IRIAF / 2019

INTRODUCTION

Apprentissage supervisé / non supervisé

Les algorithmes prédictifs se classent en deux grandes catégories :

- **supervisés**

On dispose d'un échantillon de données du passé avec la valeur cible à prédire connue. On apprend à prédire les valeurs cibles des données futures

- **non supervisés**

On dispose d'un échantillon de données du passé sans valeur cible. On extrait des patterns (caractéristiques communes) que l'on cherchera à détecter sur les données futures

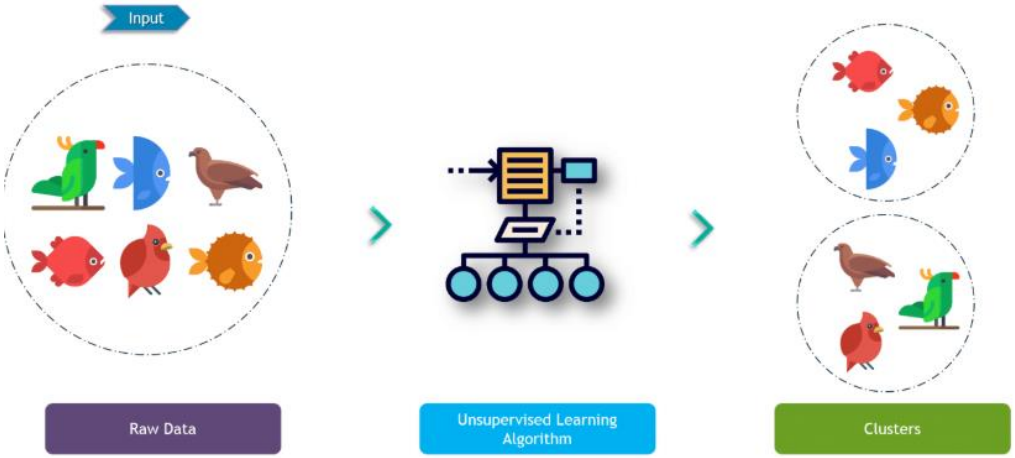
Exemples

Supervisé :

un label en input

Non supervisé :

pas de label en input



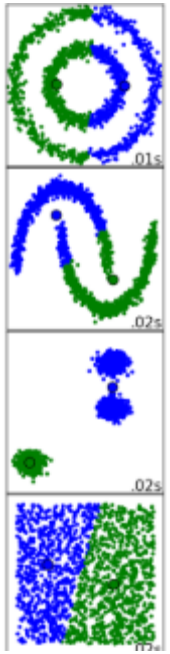
LA METHODE DES KMEANS

La méthode des kmeans

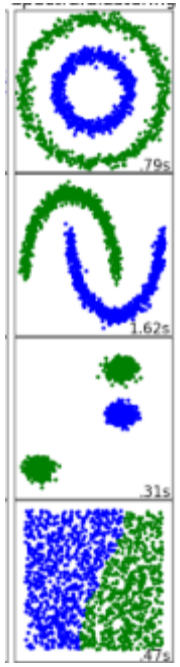
Méthode **d'apprentissage non supervisé** utilisée sur des **données non labellisés**

L'objectif est simple : regrouper des observations similaires ensemble dans un nombre (k) de clusters prédéfinis.

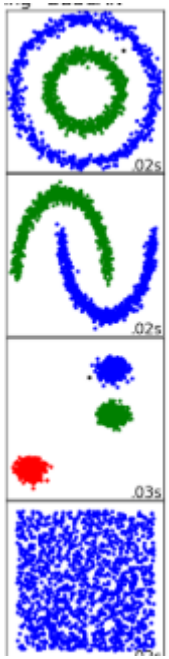
Autres algorithmes de clustering



kmeans



Spectral clustering



DBSCAN

Cas d’usage

Segmentation client :

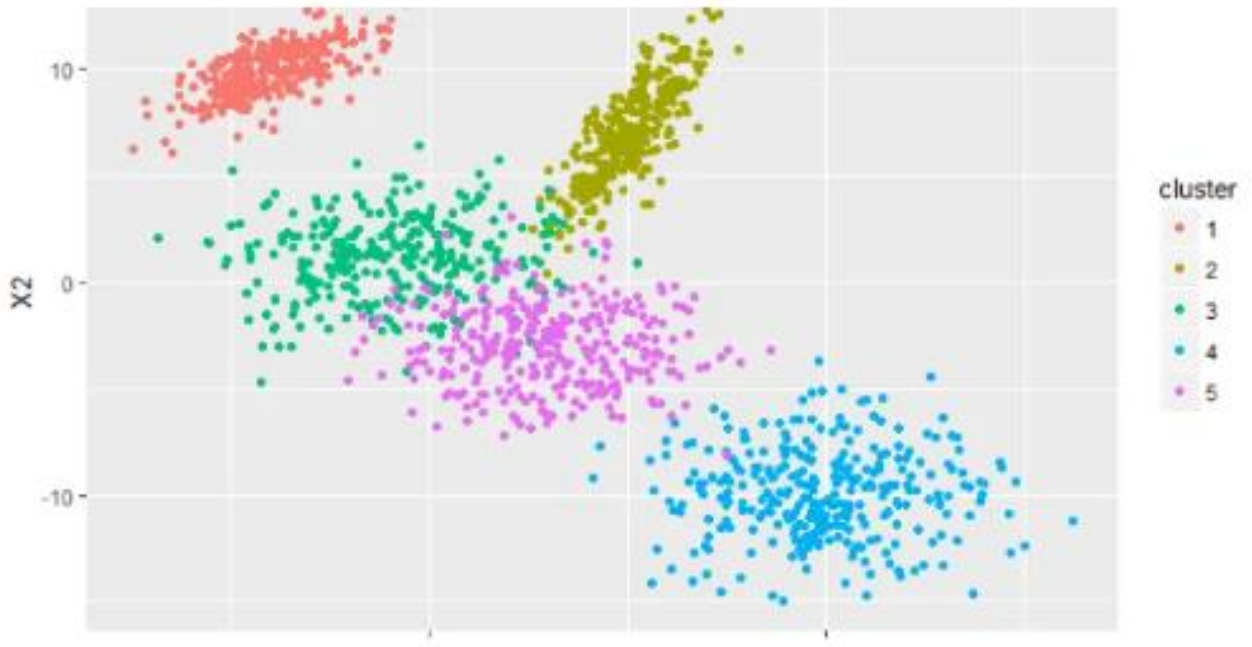
- Habitudes de consommation
- Comportement du conducteur/ du client

Détection d’anomalies :

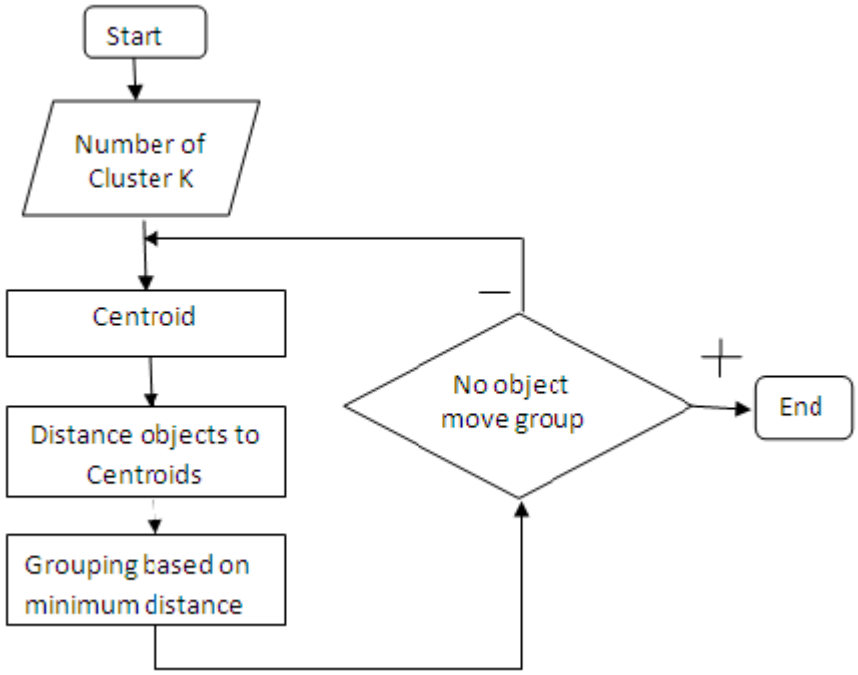
- fraudes
- pannes

Objectif

Définir ***k*** groupes homogènes parmi les observations

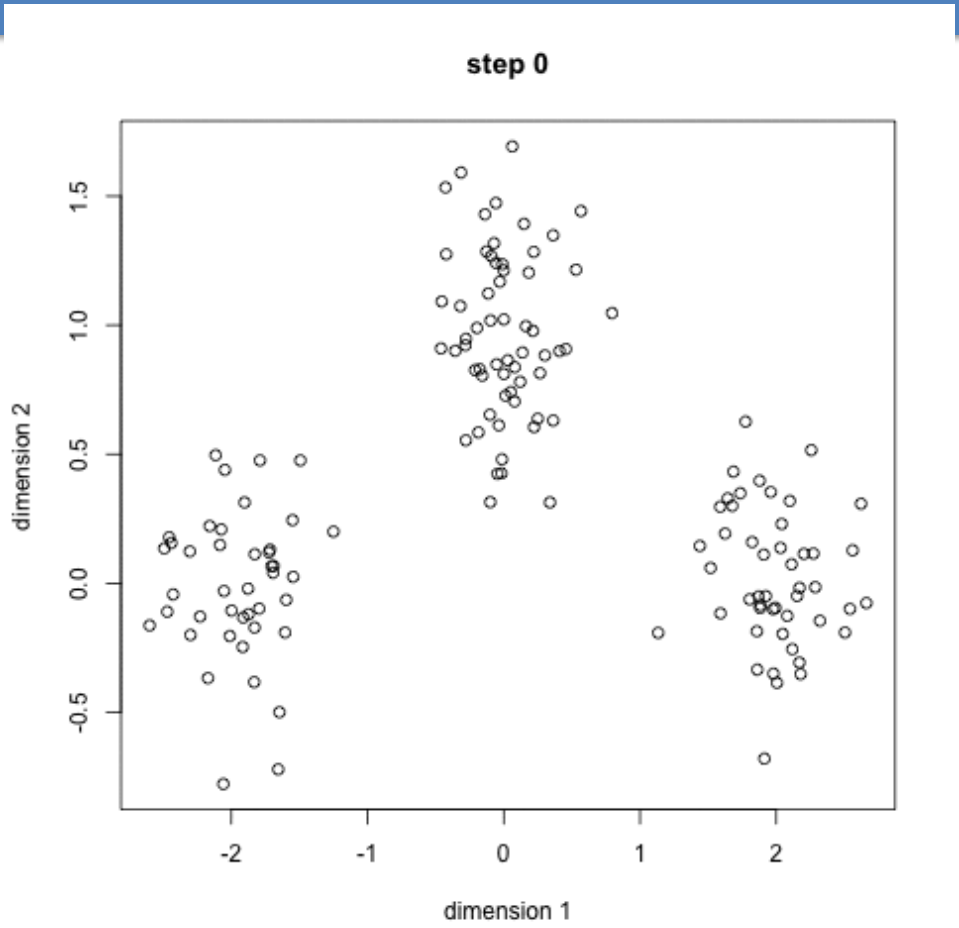


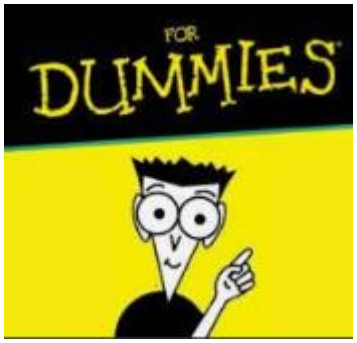
L'algorithme des kmeans



Step 1 :
On choisit k éléments aléatoires dans le plan – ce sont les « centres » des clusters
Step 2 :
on affecte chaque individu à un cluster
Step 3 :
Chaque groupe constitué permet de recalculer un élément « centre » (centre de gravité)
Step 4 :
On réaffecte les indivdus aux nouveaux centres définis au step 3
On itère jusqu'à ce que les groupes d'individus soient stables *(ie aucun indivdu ne change plus de groupe)*

En image





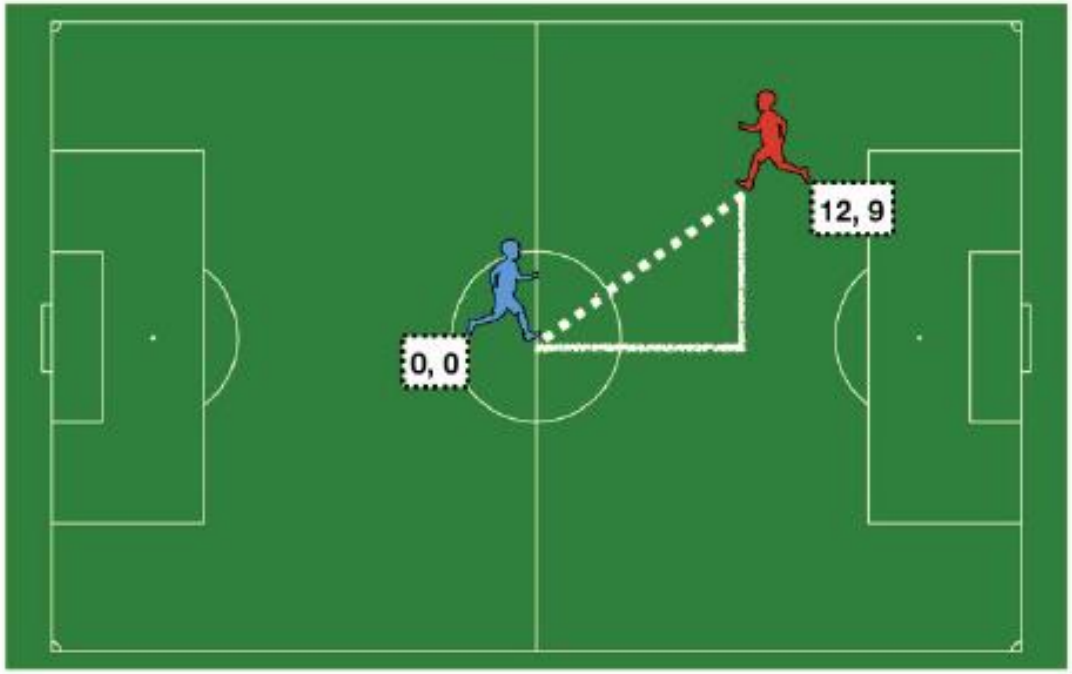
On choisit un nombre de clusters et l'algorithme fait le boulot

Le groupe le plus proche ?

L'algorithme s'appuie sur la mesure de la distance euclidienne d'un point à son centre le plus proche.

Calcul de la distance

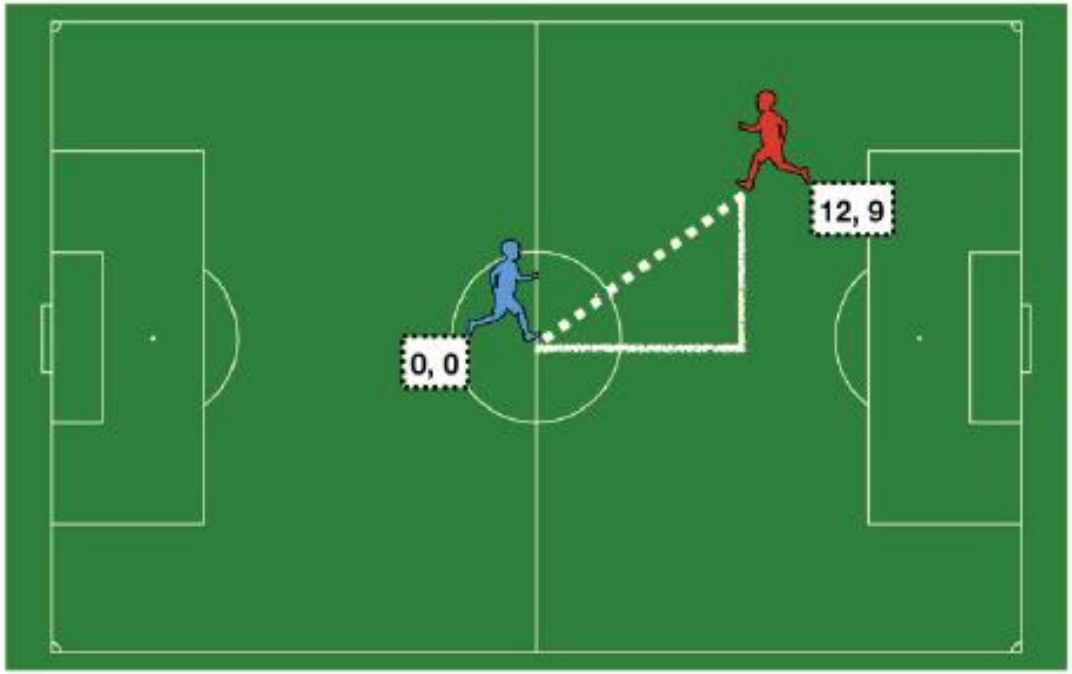
	X	Y
Blue	0	0
Red	12	9



Réponse ?

Calcul de la distance

	X	Y
Blue	0	0
Red	12	9



$$\sqrt{(xRed - xBlue)^2 + (yRed - yBlue)^2}$$

=15

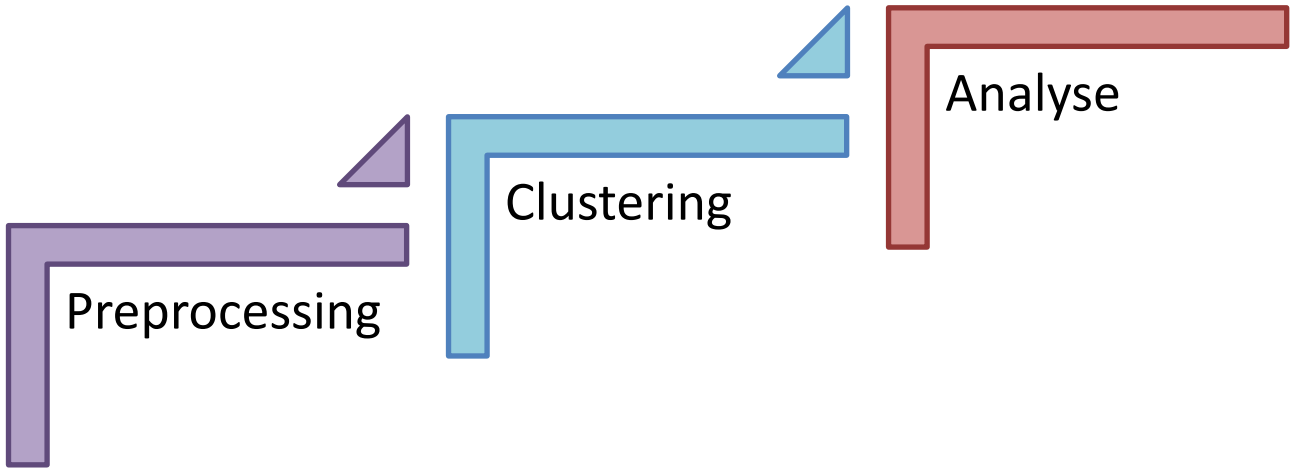
Minimiser la variance intra

L'objectif d'affecter un individu à son « centre » le plus proche revient à minimiser la variance intra du centroïd.

Formule variance intra $$\sum_{i=1}^{p} \sum_{j=1}^{n_i} (y_{ij} - \overline{y_i})^2$$

LE FLUX DE TRAITEMENT

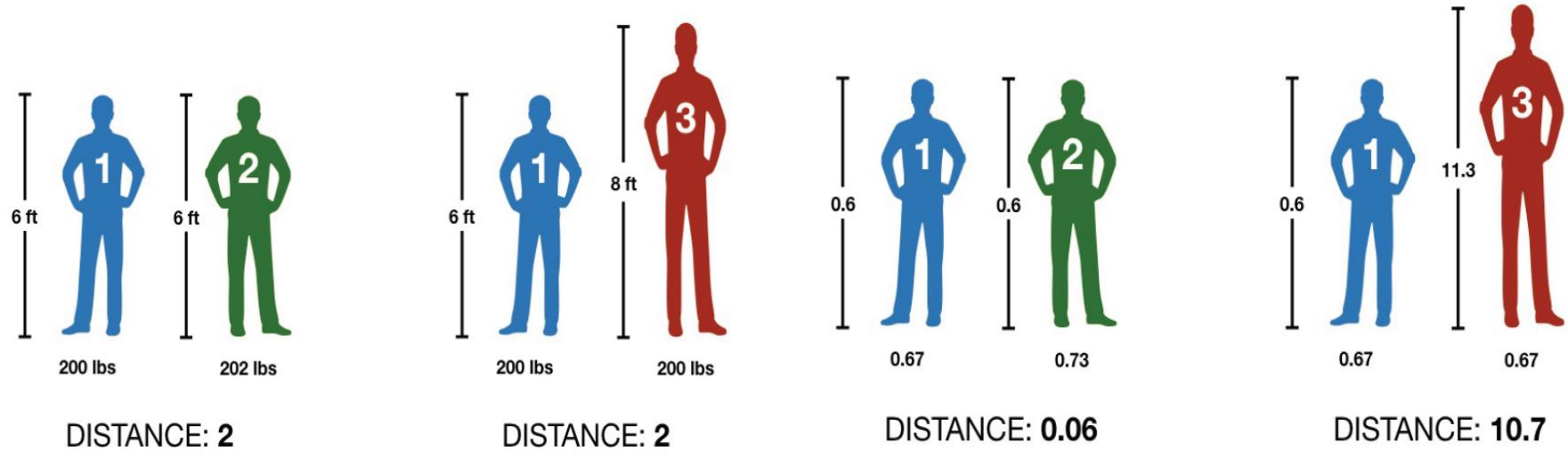
APPROCHE STANDARD POUR UN CLUSTERING



PRE PROCESSING

Standardiser les données

- L'affectation des individus aux clusters repose sur la distance, il est donc important de comparer des variables de même ordre de grandeur



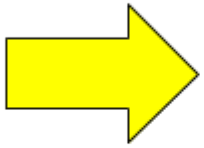
Avant scaling

Après scaling

Variables qualitatives

Possible d'utiliser des variables qualitatives en les recodant (dummy encoding)

Color
Red
Red
Yellow
Green
Yellow



Red	Yellow	Green
1	0	0
1	0	0
0	1	0
0	0	1

ACP *(facultatif)*

- Dans le cas de dataset avec un nombre important de variables (featuresà, il peut être intéressant de procéder à une ACP pour réduire le nombre de dimensions.
- On réalisera le clustering sur les coordonnées des individus sur chaque composante principale retenue

CHOIX DE K ET PERFORMANCE DU CLUSTERING

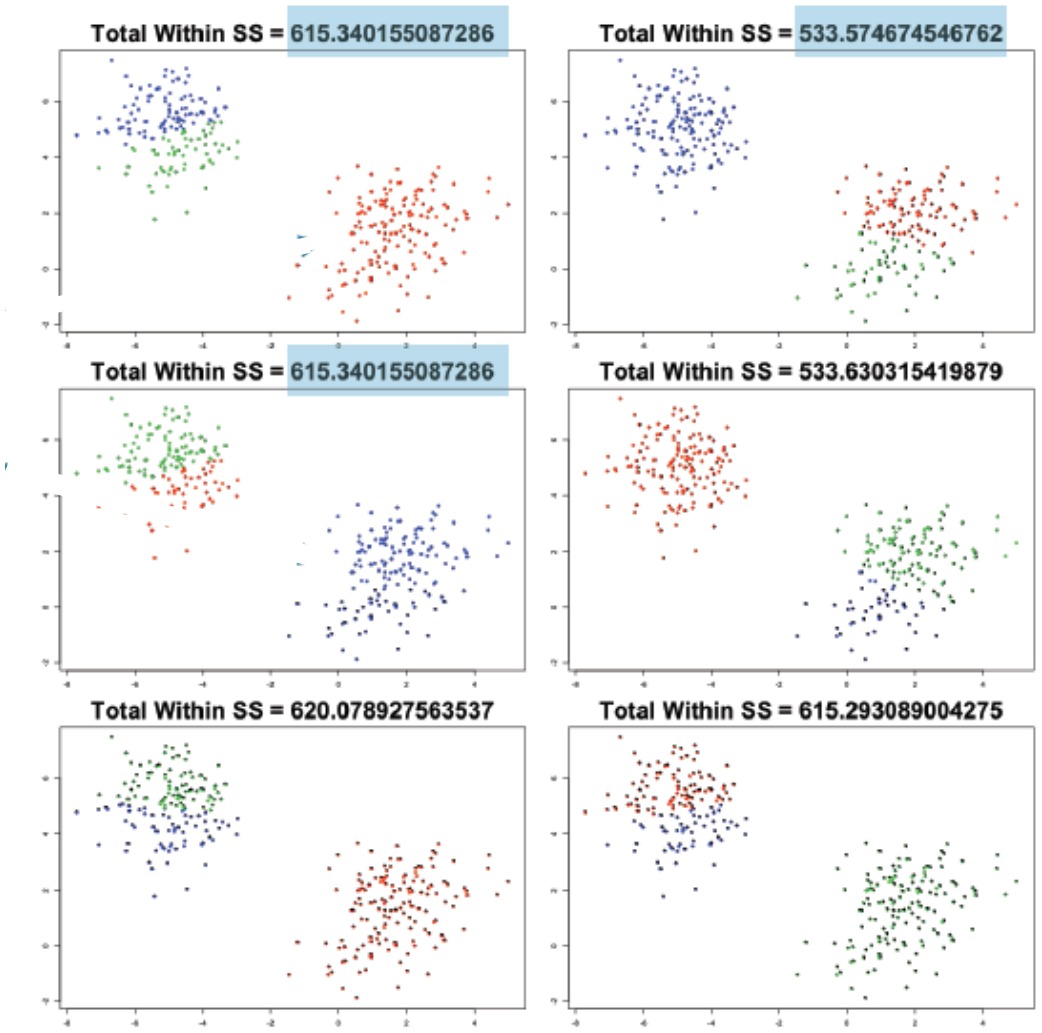
Variance intra (*total within SS*)

Une mesure de la performance du modèle est la variance intra totale, *ie* la somme des variances intra de chaque cluster.

Plus cette variance est faible, plus les groupes sont compacts, meilleur est le modèle.

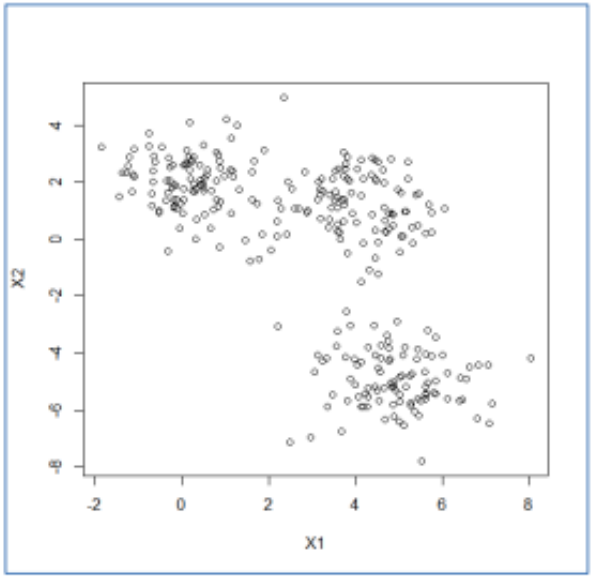
Variance intra

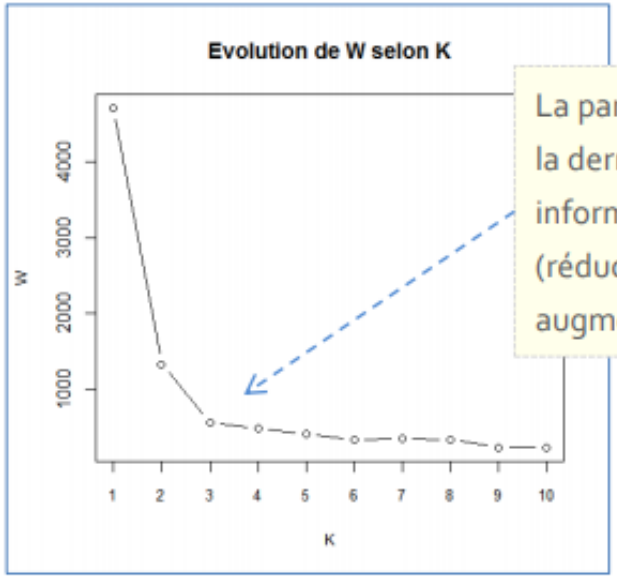
Le choix des clusters initiaux étant aléatoire, il faut procéder à plusieurs exécutions pour trouver le modèle qui minimise la variance *intra (total within Sum of Squares)*



Choix du nombre de classes k
Le coude (elbow)

Principe : Une stratégie simple pour identifier le nombre de classes consiste à faire varier K et surveiller l'évolution de l'inertie intra-classes W. L'idée est de visualiser le « coude » où l'adjonction d'une classe ne correspond à rien dans la structuration des données.





La partition en K = 3 classes est la dernière à induire un gain informationnel significatif (réduction inertie intra ➔ augmentation de l'inertie inter)

Choix du nombre de classes k
La silhouette

Permet de vérifier la pertinence de l'affectation d'un individu à une classe en calculant :

- sa distance moyenne aux individus de sa classe (C)
- sa distance moyenne aux individus de la classe la plus proche (N)

$$s(i) = \begin{cases} 1 - C(i)/N(i), & \text{if } C(i) < N(i) \\ 0, & \text{if } C(i) = N(i) \\ N(i)/C(i) - 1, & \text{if } C(i) > N(i) \end{cases}$$

Plus s est proche de 1, plus l'affectation est bonne

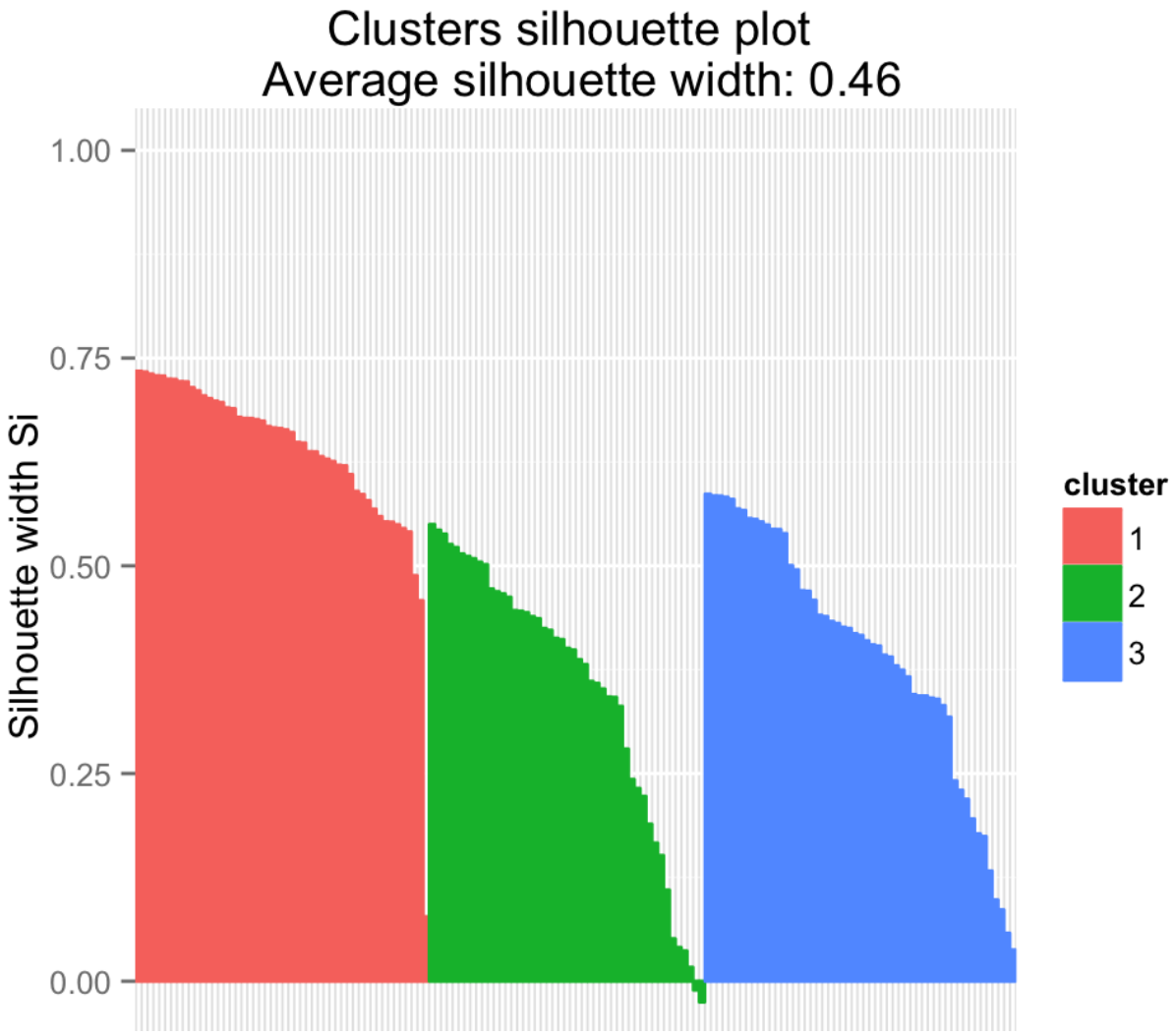
Si s est proche de 0, l'individu est à la frontière de deux clusters

Si s est <0, l'individu est mal classé

Choix du nombre de classes k
La silhouette

Un baton-> un individu

Des mal classés ?



Choix du nombre de classes k
La silhouette

La moyenne des silhouettes des individus permet de définir une métrique de performance :

- plus cette valeur est proche de 1 meilleur est le modèle

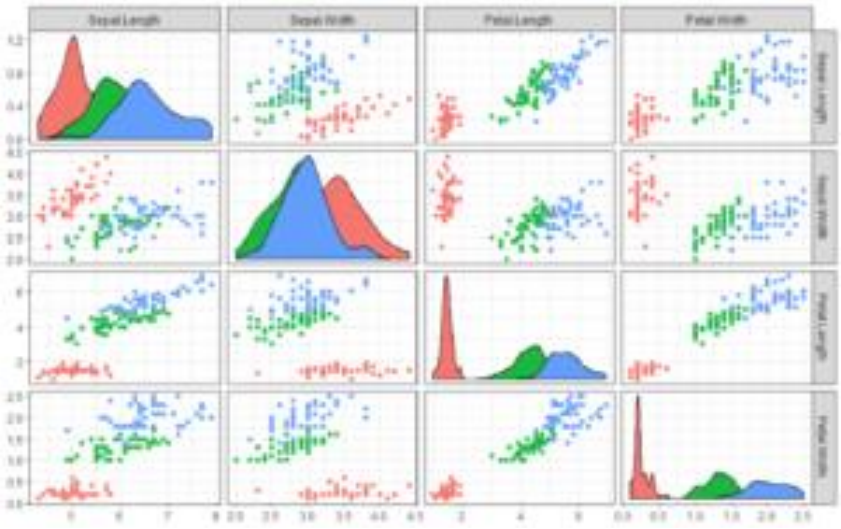
On peut ainsi calculer les silhouettes pour k:1…n clusters et choisir la valeur de k qui maximise la silhouette moyenne

DONNER DU SENS AU CLUSTERING

Analyser les caractéristiques des clusters

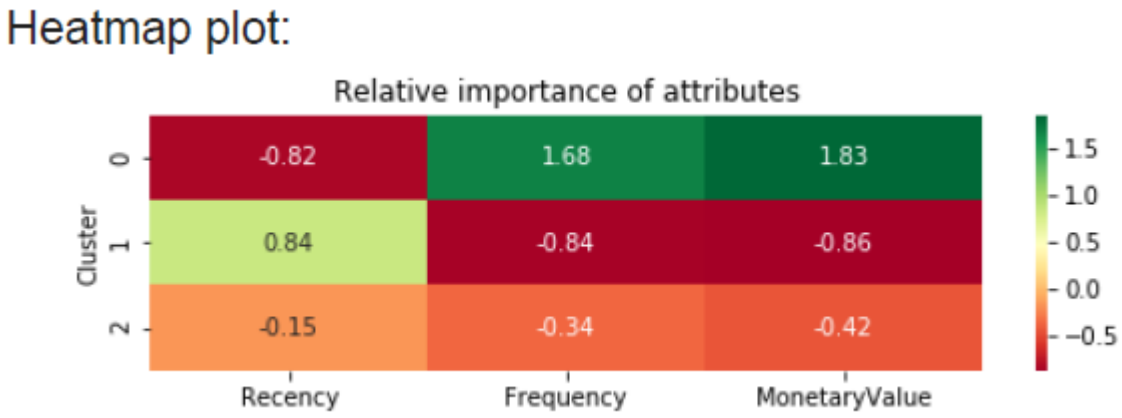
Pour chaque cluster

- ✓ sortir les statistiques descriptives
- ✓ étudier les différences entre les clusters



Analyser les caractéristiques des clusters

Identifier les variables marquantes de chaque cluster



Écart relatif de la moyenne du cluster à la moyenne de la population

CRITIQUE DE L'ALGORITHME

Inconvénients de l'algorithme

1. Le nombre de classe doit être fixé au départ

2. Le résultat dépend du tirage aléatoire initial des centres des classes

3. Labels des classes pas stables d'une exécution à l'autre

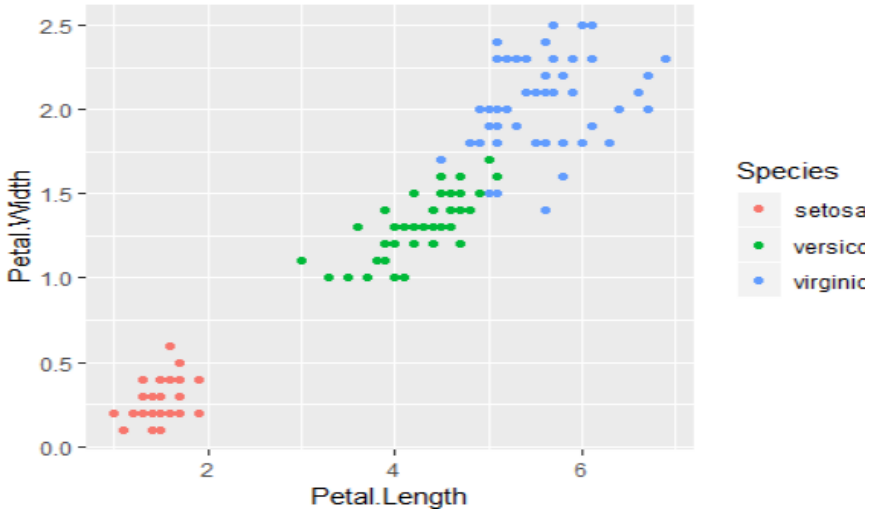
PROGRAMMATION R

R

```
library(datasets)
library(ggplot2)

head(iris)
ggplot(iris, aes(Petal.Length, Petal.Width, color =
Species)) + geom_point()
```

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



R

```
set.seed(20)
irisCluster <- kmeans(iris[, 1:4], 3, nstart = 20)
```

```
K-means clustering with 3 clusters of sizes 38, 62, 50

Cluster means:
  Sepal.Length Sepal.Width Petal.Length Petal.Width
1     6.850000    3.073684     5.742105    2.071053
2     5.901613    2.748387     4.393548    1.433871
3     5.006000    3.428000     1.462000    0.246000

Clustering vector:
  [1] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
 [41] 3 3 3 3 3 3 3 3 3 3 2 2 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 2
 [81] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 2 1 1 1 1 2 1 1 1 1 1 2 2 1 1 1 2
[121] 1 2 1 2 1 1 2 2 1 1 1 1 1 2 1 1 1 1 2 1 1 1 2 1 1 1 2 1 1 2
Levels: 1 2 3

Within cluster sum of squares by cluster:
[1] 23.87947 39.82097 15.15100
 (between_SS / total_SS =  88.4 %)
```

QUIZZ

WWW.KAHOOT.IT


89
CF
GRIEZMANN
86 PAC 87 DRI
85 SHO 48 DEF
82 PAS 72 PHY
87
GK
LLORIS
87 DIV 89 REF
82 HAN 64 SPE
68 KIC 83 POS
86
CM
POGBA
74 PAC 84 DRI
79 SHO 68 DEF
84 PAS 87 PHY
85
CB
UMTITI
75 PAC 72 DRI
64 SHO 85 DEF
72 PAS 82 PHY
85
ST
MBAPPÉ
93 PAC 87 DRI
80 SHO 38 DEF
76 PAS 74 PHY

A VOTRE TOUR