

Initiation au machine learning

Laurent Rouvière

2 janvier 2024

Table des matières

1 Présentation du cours	1
1.1 Intelligence Artificielle - Apprentissage Automatique - Apprentissage Profond	3
1.2 L'IA : nouveaux enjeux	8
1.3 Apprentissage : quelques exemples	15
2 Apprentissage supervisé	20
2.1 Écriture du problème	20
2.2 Distances	23
2.3 L'algorithme des k plus proches voisins	28
2.4 Evaluation de la performance - choix de k	32
2.5 Une autre méthode : les arbres	40
3 Apprentissage non supervisé	44
3.1 Écriture du problème	44
3.2 La classification ascendante hiérarchique (CAH)	47

1 Présentation du cours

Présentation

- *Enseignant* : Laurent Rouvière laurent.rouviere@univ-rennes2.fr
 - *Recherche* : statistique non paramétrique, apprentissage statistique.
 - *Enseignement* : statistique et probabilités (Université, école d'ingénieur, formation continue).

- **Responsabilités** : parcours “Science des données, Intelligence Artificielle” du Master MAS
- **Consulting** : énergie (ERDF), finance, marketing.
- **Prérequis** : Mathématiques (algèbre) et informatique (python).
- **Objectifs** :
 - présenter les notions d’**Intelligence Artificielle**, **Machine Learning**, **Apprentissage**
 - comprendre l’utilité des **mathématiques** et de l’**informatique** dans ces domaines
 - développer **quelques algorithmes** à partir d’outils de L1.

Documents de cours

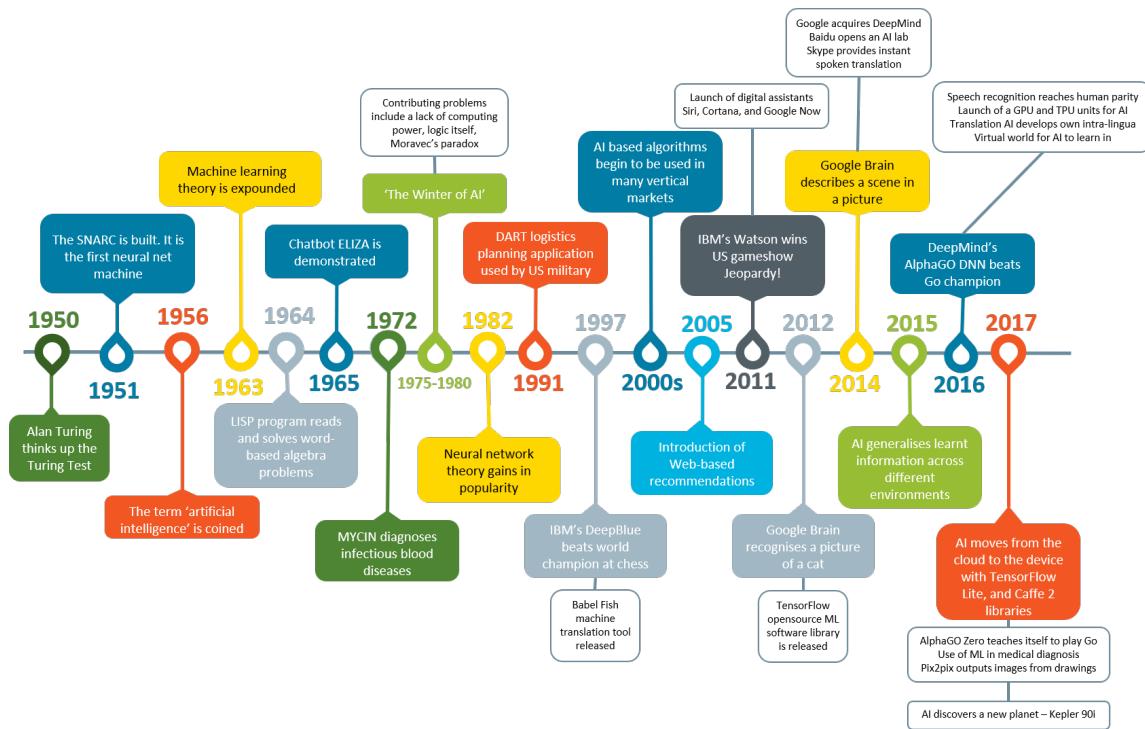
- **Slides** disponibles à l’url https://lrouviere.github.io/page_perso/apprentissage_sup.html#intro-ML-L1Miashs
- Exercices et TP dans cursus.

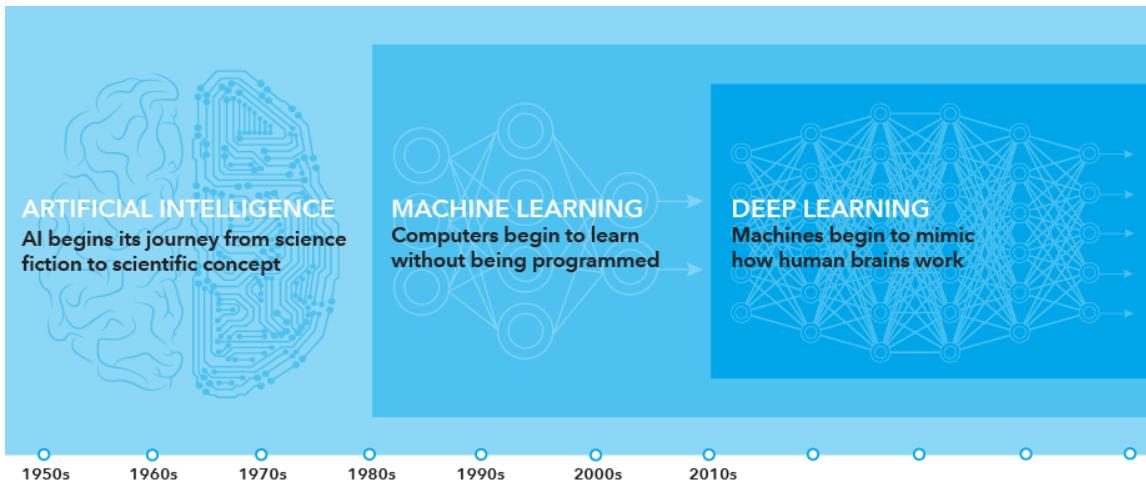
Pourquoi ce cours ?

1. Mettre une dose de **Mathématiques et d’Informatique Appliqués** dans une filière MIASHS.
2. Comprendre la **dualité/complémentarité** maths/info pour répondre à des **problèmes concrets**.
3. Aider par rapport aux **futurs choix** de cursus universitaire.

1.1 Intelligence Artificielle - Apprentissage Automatique - Apprentissage Profond

Un peu d'historique





IA - début ([source](#))

- Historiquement, les débuts de l'IA remontent à [Alan Turing](#) dans les années [1950](#)

Test de Turing

- Faire parler une personne avec d'autres personnes et un ordinateur (à l'aveugle).
- Identifier si **vraie personne** ou **ordinateur**.
- Le mot veut **tout dire et ne rien dire**.
- Se réduit à des **règles déterministes**

```
si (condition 1 et|ou condition 2...) alors (action 1)
sinon (action 2)
```
- Les choses ont évolué par la suite notamment grâce au développement des **ressources informatiques** et de la **puissance de calcul**.

Quelques définitions

Intelligence artificielle (Wikipédia)

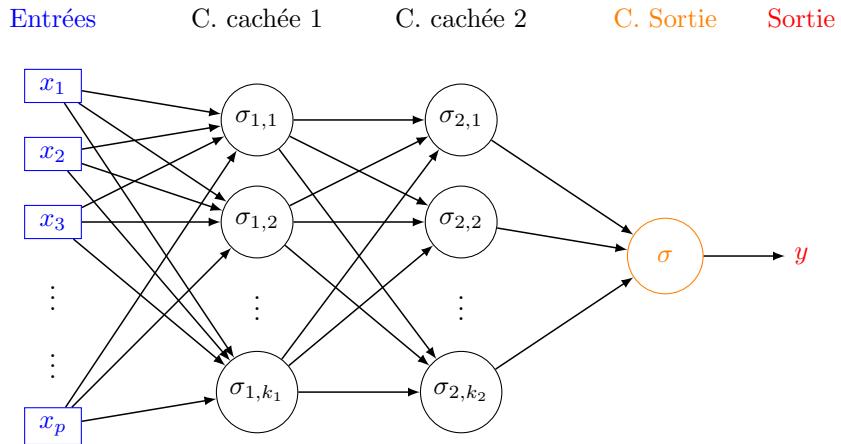
Ensemble de théories et de techniques mises en œuvre en vue de réaliser des **machines capables de simuler l'intelligence humaine**.

Apprentissage automatique - Machine Learning (Wikipédia)

Champ d'étude de l'intelligence artificielle qui se fonde sur des approches mathématiques et statistiques pour donner aux ordinateurs la capacité d'« apprendre » à partir de données, c'est-à-dire d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune.

Apprentissage profond - Deep Learning

Technologie du ML qui permet de mimer les actions du cerveau humain grâce à des réseaux de neurones composés de dizaines voire de centaines de couches de neurones, chacune recevant et interprétant les informations de la couche précédente.



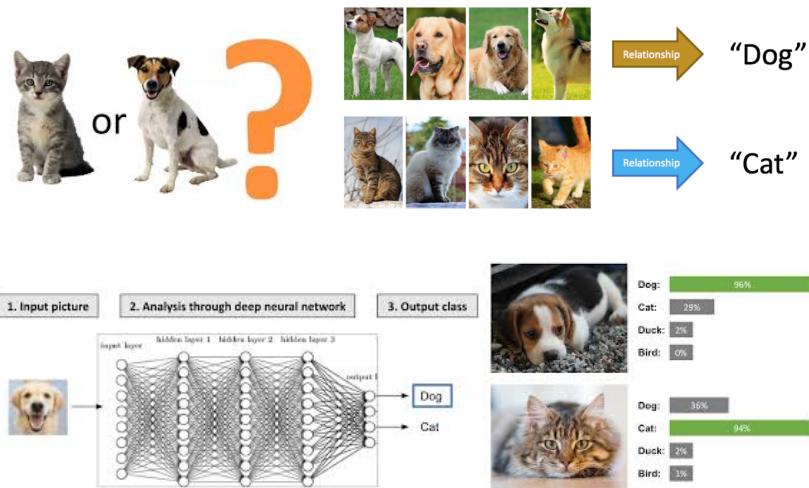
Quelques dates

- 1950 : Alan Turing
- 1959 : apparition du *Machine Learning* (Arthur Samuel, jeu de dames) \implies création d'un programme pour IBM qui s'améliore en jouant.
- Années 80 : développement du ML \implies l'ordinateur commence à déduire des règles à suivre en analysant seulement des données.
- 1997 : *Deep Blue* \implies premier ordinateur à vaincre Garry Kasparov aux échecs.
- 2000 : naissance de l'*apprentissage profond (deep learning)*.
- 2012 : *identification de visages humains* par des réseaux de neurones développés par Google.

2015 nouveaux défis

- *Perception visuelle*
- *Compréhension du langage naturel* écrit ou parlé
- *Analyse automatique du langage*
- *Prise de décision autonome*

Perception visuelle et prise de décision



Compréhension du langage

The screenshot shows the DeepL Translator interface. At the top, there's a navigation bar with links for "DeepL Traducteur", "DeepL Pro", "Pourquoi DeepL?", "API", "Forfaits et tarifs", "Applications", and "GRATUIT". On the right side of the bar are buttons for "Commencer l'essai gratuit", "Connexion", and a menu icon.

Below the navigation bar, there are two tabs: "Traduire du texte" (selected) and "Traduire des fichiers". The "Traduire du texte" tab has a sub-section for "29 langues".

The main translation area shows a French input "Introduction au machine learning est mon cours préféré." and its English translation "Introduction to Machine Learning is my favorite course." The source text has a red dotted underline under "machine learning". There are small icons for copy, like, and share below the input text.

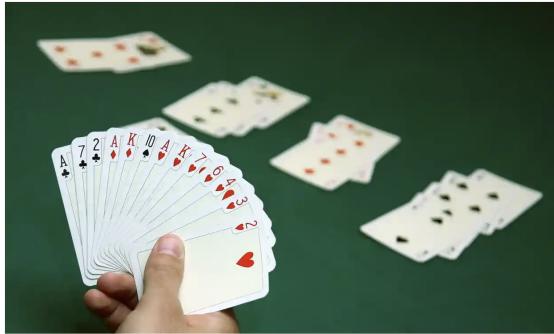
At the bottom left, there's a "Dictionnaire" section with the instruction "Cliquez sur un mot pour en savoir plus à son sujet." and a small icon.

1.2 L'IA : nouveaux enjeux

Compétitions hommes/machines

Artificial intelligence beats eight world champions at bridge

Victory marks milestone for AI as bridge requires more human skills than other strategy games



The AI, Nook, was able to read its opponents and explain its decision-making. Photograph: switas/Getty Images/iStockphoto

An artificial intelligence has beaten eight world champions at bridge, a game in which human supremacy has resisted the march of the machines until now.

The victory represents a new milestone for AI because in bridge players work with incomplete information and must react to the behaviour of several other players - a scenario far closer to human decision-making.

Artificial Intelligence Easily Beats Human Fighter Pilot in DARPA Trial

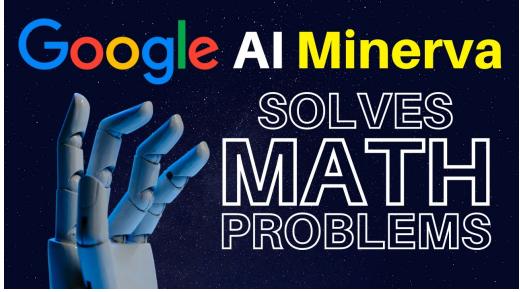
Aug. 20, 2020 | By Brian W. Everstine



SHARE ARTICLE

In the battle of artificial intelligence versus a human fighter pilot, it wasn't even close. The artificial intelligence algorithm, developed by Heron Systems, swept a human F-16 pilot in a simulated dogfight 5-0 in the Defense Advanced Research Projects Agency's [AlphaDogfight Trials](#) on Aug. 20. The company beat out seven other companies before going head to head with "Banger," a pilot from the District of Columbia Air National Guard and a recent graduate of the Air Force Weapons School's F-16 Weapons Instructor Course. The pilot, whose full name was not provided, is an operational fighter pilot with more than 2,000 hours in the F-16.

Banger and Heron Systems' AI fought in five different basic fighter maneuver scenarios with the simulated fight only using the Fighting Falcon's guns, and each time the AI was able to out maneuver and take out Banger. The algorithm operated within the limits of the F-16—meaning it did not pull Gs beyond what a real-world aircraft could do. However, Banger said after the event that the jet was not limited by the training and thinking that is engrained in an



Question: What are the eigenvalues of a 2-by-2 matrix $M = \begin{pmatrix} 1 & 2 \\ -1 & 5 \end{pmatrix}$?

Model output:
The characteristic polynomial of M is $|M - \lambda I|$, where I is the 2-by-2 identity matrix. So the characteristic polynomial is $|M - \lambda I| = \begin{vmatrix} 1 - \lambda & 2 \\ -1 & 5 - \lambda \end{vmatrix} = (1 - \lambda)(5 - \lambda) + 2 = \lambda^2 - 6\lambda + 7$. This is a quadratic polynomial, so the eigenvalues are the solutions of $\lambda^2 - 6\lambda + 7 = 0$. We can solve this by completing the square:

$$\begin{aligned} \lambda^2 - 6\lambda + 7 &= 0 \\ \Rightarrow \quad \lambda^2 - 6\lambda + 9 - 9 + 7 &= 0 \\ \Rightarrow \quad (\lambda - 3)^2 - 2 &= 0 \\ \Rightarrow \quad (\lambda - 3)^2 &= 2 \\ \Rightarrow \quad \lambda - 3 &= \pm\sqrt{2} \\ \Rightarrow \quad \lambda &= 3 \pm \sqrt{2} \end{aligned}$$

So the eigenvalues are $3 + \sqrt{2}$ and $3 - \sqrt{2}$.

Question: Assume that the variance of the first n natural numbers is 10, and the variance of the first m even natural numbers is 16. Compute $m + n$.

Model output: Let σ^2 be the variance of the first n even natural numbers, and τ^2 be the variance of the first n natural numbers. Then, we have $\tau^2 = \frac{n^2 - 1}{12}$ and $\sigma^2 = \frac{m^2 - 1}{3}$. We solve the system of equations:

Question: For every $a, b, b \neq a$ prove that $\frac{a^2 + b^2}{2} > \left(\frac{a+b}{2}\right)^2$.

Model output:

$$\begin{aligned} \frac{a^2 + b^2}{2} &> \left(\frac{a+b}{2}\right)^2 \\ \iff \frac{a^2 + b^2}{2} &> \frac{a^2 + b^2 + 2ab}{4} \\ \iff a^2 + b^2 &> a^2 + b^2 + 2ab \\ \iff 2a^2 + 2b^2 &> a^2 + b^2 + 2ab \\ \iff a^2 + b^2 &> 2ab \\ \iff a^2 + b^2 - 2ab &> 0 \\ \iff (a - b)^2 &> 0 \end{aligned}$$

which is true, because the square of a real number is positive.

ChatGPT

- Dernière expérimentation d'*OpenAI*.
- Outil conversationnel en *langage naturel* qui va produire des *réponses et des textes* demandés par des commandes écrites dans des langues prises en charge.
- Il sait aussi faire des *maths*...

Exemples

The screenshot shows a digital notebook interface with two entries:

Différences entre intelligence artificielle et machine Learning en français

L'intelligence artificielle (IA) et le machine learning (apprentissage automatique en français) sont souvent confondus, mais ils ne sont pas exactement la même chose. L'IA est un domaine de l'informatique qui vise à créer des ordinateurs qui simulent l'intelligence humaine. Le machine learning, en revanche, est une technique utilisée pour mettre en œuvre l'IA.

Voici quelques différences clés entre l'IA et le machine learning :

- L'IA est une discipline qui vise à créer des ordinateurs capables de simuler l'intelligence humaine, tandis que le machine learning est une technique utilisée pour mettre en œuvre l'IA.
- L'IA peut utiliser différentes techniques pour atteindre ses objectifs, comme le raisonnement symbolique, la logique floue, etc. Le machine learning, en revanche, se base principalement sur l'analyse de données et l'apprentissage automatique.
- L'IA peut être utilisée pour résoudre des problèmes complexes qui nécessitent une grande quantité de données et de calculs, tels que la reconnaissance de la parole ou la traduction automatique. Le machine learning, quant à lui, est utilisé pour automatiser des tâches qui nécessitent de grandes quantités de données et de calculs, mais qui ne nécessitent pas nécessairement une compréhension profonde du domaine.

En résumé, l'IA est un domaine large qui vise à créer des ordinateurs capables de simuler l'intelligence humaine, tandis que le machine learning est une technique utilisée pour mettre en œuvre l'IA en utilisant l'analyse de données et l'apprentissage automatique.

Démontrer que racine 2 n'est pas un rationnel

Pour démontrer que la racine carrée de 2 n'est pas un rationnel, nous allons utiliser le fait que tous les nombres rationnels peuvent être écrits sous la forme d'une fraction irréductible, c'est-à-dire une fraction dont le numérateur et le dénominateur ne ont pas de diviseurs communs autres que 1.

Commençons par supposer que la racine carrée de 2 est un rationnel, c'est-à-dire qu'elle peut être écrite sous la forme d'une fraction irréductible a/b , où a et b sont des entiers. Nous savons alors que $a^2 = 2 * b^2$.

Si a et b sont tous les deux pairs, alors a^2 et b^2 sont tous les deux pairs, ce qui signifie que $2 * b^2$ est pair, ce qui signifie que a^2 est pair. Cela signifie que a est pair, ce qui signifie que a et b ont au moins 2 comme diviseur commun, ce qui contredit le fait que la fraction a/b est irréductible.

Si a est pair et b est impair, alors a^2 est pair et b^2 est impair, ce qui signifie que $2 * b^2$ est impair, ce qui signifie que a^2 est impair. Cela signifie que a est impair, ce qui signifie que a et b n'ont aucun diviseur commun, ce qui signifie que la fraction a/b est irréductible, ce qui contredit notre hypothèse initiale selon laquelle la racine carrée de 2 était un rationnel.

Enfin, si a et b sont tous les deux impairs, alors a^2 et b^2 sont tous les deux impairs, ce qui signifie que $2 * b^2$ est pair, ce qui signifie que a^2 est pair. Cela signifie que a est pair, ce qui signifie que a et b ont au moins 2 comme diviseur commun, ce qui contredit le fait que la fraction a/b est irréductible.

En résumé, dans tous les cas, nous arrivons à la conclusion que la fraction a/b ne peut être irréductible, ce qui signifie que la racine carrée de 2 ne peut être un rationnel. Par conséquent, nous avons démontré que la racine carrée de 2 n'est pas un rationnel.

Génération d'image

Generative Adversarial Networks (GAN)

- Type de **réseau de neurones** utilisé en apprentissage automatique pour générer de nouvelles données de **manière convaincante**, de manière à ce qu'elles soient difficiles à distinguer des données réelles.
- Nombreuses applications, notamment la **génération d'images**, la **génération de texte** et la création de **contenu vidéo**.

DALL.E

- “new AI system that can create realistic images and art from a description in natural language”.
- voir <https://openai.com/dall-e-2/>

Exemple <https://www.whichfaceisreal.com/>

Click on the person who is real.



Exemple <https://www.whichfaceisreal.com/>

You are **correct**. The image on the right is real.

[Play again.](#)



Autres...



- *Apprentissage automatique* ou *machine learning* : sous domaine de l'Intelligence Artificielle
- On rencontre différents types de problème en *apprentissage*, notamment

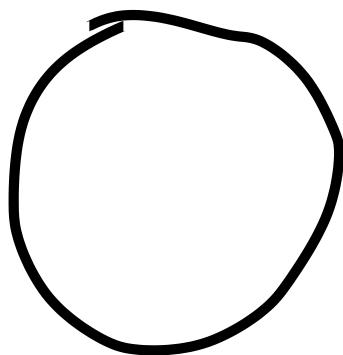
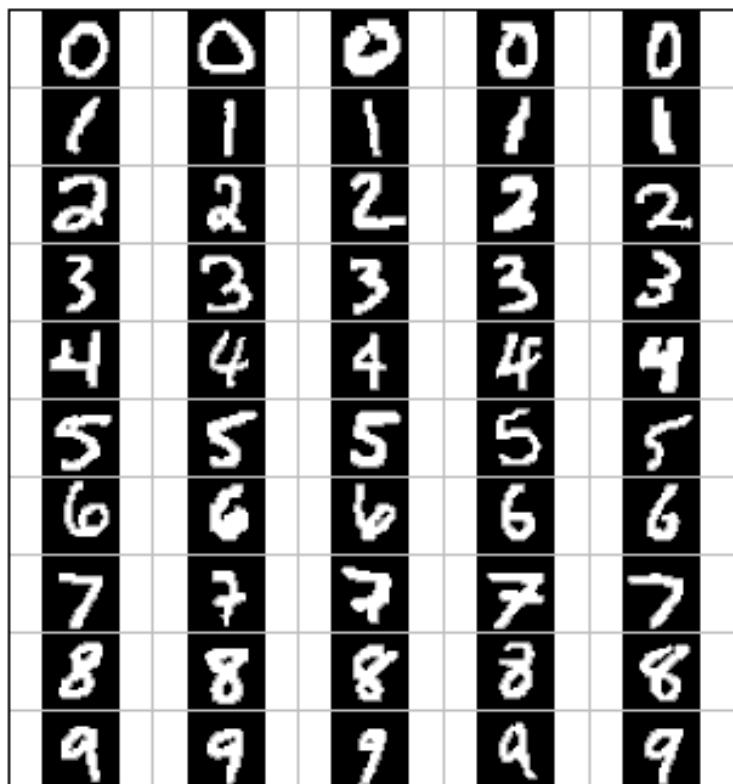


Dans ce cours

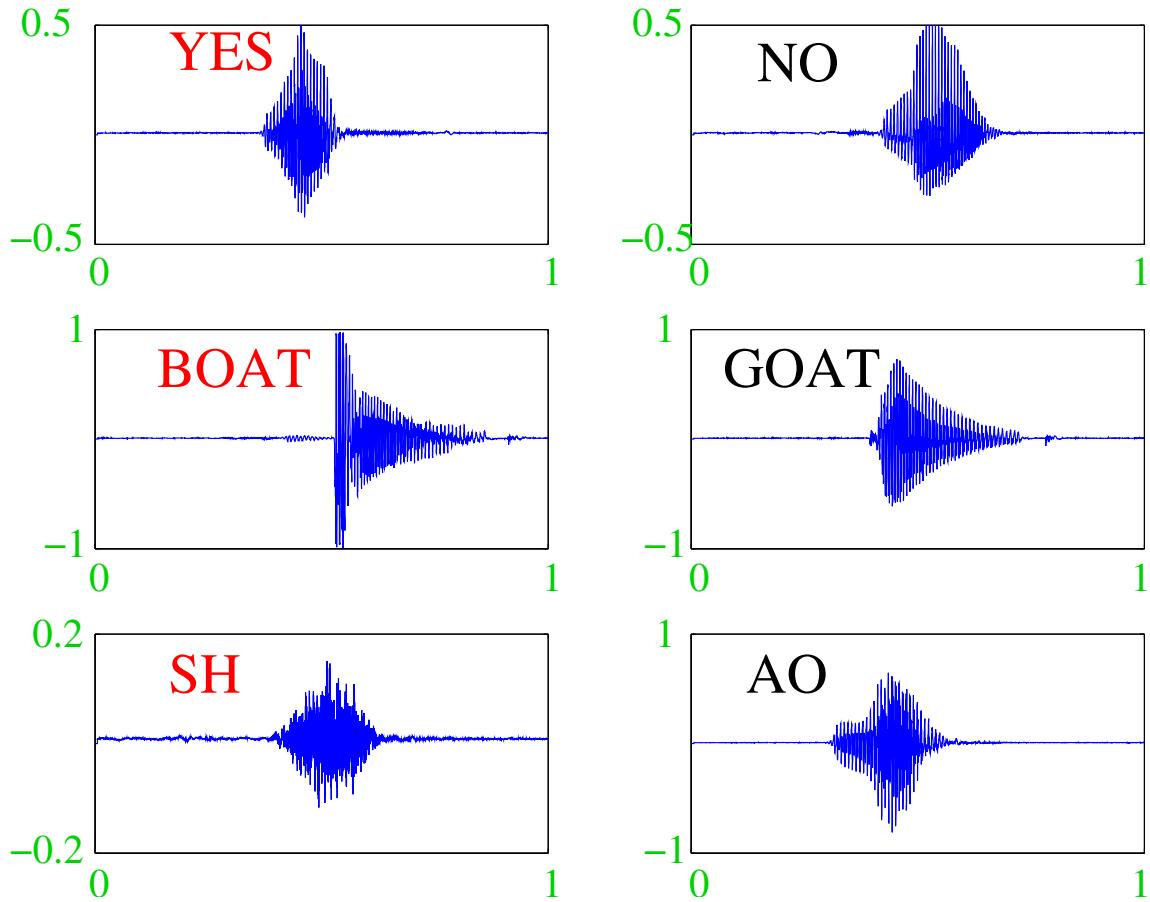
On se concentrera sur l'apprentissage supervisé et non supervisé.

1.3 Apprentissage : quelques exemples

Reconnaissance de l'écriture



Reconnaissance de la parole



Prévision des pics d'ozone

- On a mesuré pendant 112 jours la *concentration maximale* en ozone max03 ;
- On dispose également d'autres variables météorologiques (température, nébulosité, vent...).

```
## # A tibble: 6 x 11
##   max03    T9    T12    T15    Ne9    Ne12    Ne15    Vx9    Vx12    Vx15 max03v
##   <int> <dbl> <dbl> <dbl> <int> <int> <int> <dbl> <dbl> <dbl> <dbl> <int>
## 1    87  15.6  18.5  18.4     4     4     8  0.695 -1.71 -0.695     84
## 2    82   17   18.4  17.7     5     5     7 -4.33  -4     -3      87
## 3    92  15.3  17.6  19.5     2     5     4  2.95   1.88  0.521     82
## 4   114  16.2  19.7  22.5     1     1     0  0.985  0.347 -0.174     92
```

```
## 5    94  17.4 20.5 20.4     8     8     7 -0.5 -2.95 -4.33    114
## 6    80  17.7 19.8 18.3     6     6     7 -5.64 -5      -6      94
```

Détection de spam

- Sur 4601 mails, on a pu identifier *1813 spams*.
- On a également mesuré sur chacun de ces mails la présence ou absence de *57 mots*.

```
## # A tibble: 8 x 9
##   make address all num3d our over remove internet type
##   <dbl>   <dbl> <dbl> <dbl> <dbl> <dbl>   <dbl> <fct>
## 1 0.09   0.49  0.59    0  0.39  0.19    0        0 spam 
## 2 0       0      0      0  0      0      0        0 spam 
## 3 0       0      0      0  0      0      0        0 nonspam
## 4 0       0      1.58   0  1.58   0      1.58    0 spam 
## 5 0       0      0      0  0      0      0        0 nonspam
## 6 0       0      3.57   0  0      0      0        0 nonspam
## 7 0       0      0      0  0      0      0        0 nonspam
## 8 0       0      0      0  0      0      0        0 nonspam
```

Les iris de Fisher

Question

Pour 3 espèces d'iris différentes, est-il possible d'*expliquer* (ou de *prédire*) l'appartenance à une des espèces connaissant les longueurs et largeurs de sépales ?



Les données

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width   Species
## 1         4.3       3.0        1.1       0.1     setosa
## 2         5.0       3.3        1.4       0.2     setosa
## 3         7.7       3.8        6.7       2.2 virginica
## 4         4.4       3.2        1.3       0.2     setosa
## 5         5.9       3.0        5.1       1.8 virginica
## 6         6.5       3.0        5.2       2.0 virginica
## 7         5.5       2.5        4.0       1.3 versicolor
## 8         5.5       2.6        4.4       1.2 versicolor
```

La criminalité aux États-Unis

```
##           Murder Assault UrbanPop Rape
## Alabama      13.2     236      58 21.2
## Alaska       10.0     263      48 44.5
## Arizona       8.1     294      80 31.0
## Arkansas      8.8     190      50 19.5
## California    9.0     276      91 40.6
## Colorado      7.9     204      78 38.7
```

Le problème

Regrouper les états en fonction des types de crimes commis.

Supervisé vs non supervisé

- Les exemples précédents peuvent être divisés en *2 catégories de problème*.

Catégorie 1

Expliquer et/ou prédire une quantité par d'autres (écriture, parole, ozone, spam...)

⇒ Apprentissage Supervisé

Catégorie 2

Regrouper des individus en fonction de leur proximité (criminalité)

⇒ Apprentissage Non Supervisé

Début de modélisation - Supervisé

On observe deux types de quantité

À expliquer/prédire

- On a toujours une seule quantité à expliquer.
- Elle est appelée variable à expliquer ou label ou encore étiquette.
- On la note y .

Explicatives

- On en a souvent plusieurs.
- Elles sont appelées variables explicatives ou covariables.
- On les note x .

Exemples

Explicatives x	À expliquer y
Image	Chiffre
Courbe	Mot
Pétales, sépales	Espèce
Données météo	Ozone

Début de modélisation - Non Supervisé

- On n'observe pas de label ou d'étiquette ici (pas de y !).
- On dispose “juste” de plusieurs variables (notées x).
- Et on souhaite regrouper les individus proches au sens de ces variables.

2 Apprentissage supervisé

2.1 Écriture du problème

Les données

- Elles correspondent à des *mesures* faites sur des *individus (statistiques)*.
- On note n le nombre d'individus statistique.
- Ces données sont réparties en deux parties.

À expliquer

- Pour chacun des n individus, on dispose *d'une valeur* observée de la variable à expliquer.
- On les note $y_i, i = 1, \dots, n$.

Explicative

- Pour chacun des n individus, on dispose de *p valeurs* qui correspondent aux *variables explicatives* V_1, \dots, V_p .
- On utilise donc un *vecteur* pour représenter ces variables :

$$x_i = (x_{i1}, \dots, x_{ip}), \quad i = 1, \dots, n$$

⇒ x_{ij} correspond à la mesure de la variable j pour l'individu i .

Le tableau de données

On peut ainsi résumer les données dans un tableau de la forme

$$\mathcal{D}_n = \underbrace{\begin{pmatrix} x_{11} & \dots & x_{1p} & y_1 \\ x_{21} & \dots & x_{2p} & y_2 \\ \vdots & & \vdots & \vdots \\ x_{n1} & \dots & x_{np} & y_n \end{pmatrix}}_{\text{Explicative}} \quad \underbrace{\qquad}_{\text{À expliquer}}$$

Le problème

À partir de cette base (d'entraînement) **prédirer la meilleure valeur de y** pour tout nouvel individu $x = (x_1, \dots, x_p)$.

L'exemple des iris

- $n = 150, p = 4$
- Extrait de la **base d'entraînement** :

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
## 1	4.3	3.0	1.1	0.1	setosa
## 2	5.0	3.3	1.4	0.2	setosa
## 3	7.7	3.8	6.7	2.2	virginica
## 4	4.4	3.2	1.3	0.2	setosa
## 5	5.9	3.0	5.1	1.8	virginica
## 6	6.5	3.0	5.2	2.0	virginica
## 7	5.5	2.5	4.0	1.3	versicolor
## 8	5.5	2.6	4.4	1.2	versicolor

- On a par exemple

$$x_1 = (4.3, 3, 1.1, 0.1) \quad \text{et} \quad y_1 = \text{setosa}$$

Le problème

- On a un **nouvel individu** (un nouvel iris)

$$x = (5.8, 2.4, 2.2, 1.3)$$

- On ne connaît **pas** y (son espèce)
- Que **prédirer** ? Setosa, versicolor ou virginica ?

Remarque

La quantité à prédire (y) est ici une **variable qualitative**.

L'exemple de l'ozone

- $n = 112, p = 10$

```
## # A tibble: 6 x 11
##   max03    T9    T12    T15    Ne9   Ne12   Ne15    Vx9    Vx12    Vx15 max03v
##   <dbl> <dbl>
## 1    87  15.6  18.5  18.4     4     4     8  0.695 -1.71 -0.695    84
## 2    82   17    18.4  17.7     5     5     7 -4.33  -4     -3     87
## 3    92  15.3  17.6  19.5     2     5     4  2.95   1.88  0.521    82
## 4   114  16.2  19.7  22.5     1     1     0  0.985  0.347 -0.174    92
## 5    94  17.4  20.5  20.4     8     8     7 -0.5   -2.95 -4.33    114
## 6    80  17.7  19.8  18.3     6     6     7 -5.64  -5     -6     94
```

- On a par exemple

$$x_1 = \quad \text{et} \quad y_1 =$$

Le problème

- On veut prédire la concentration en ozone de demain à partir des prévisions météo

$$x = (11.8, 13.4, \dots, 81)$$

- On ne connaît pas y .
- Quelle valeur prédire ?

Remarque

La quantité à prédire (y) est ici une *variable quantitative (ou continue)*.

Algorithme de prévision

- On veut renvoyer une valeur de y (la prévision) pour tout nouvel individu x .
- Mathématiquement, il faut donc trouver une fonction f qui va de l'espace *des variables explicatives* (généralement \mathbb{R}^p) dans l'espace où la *variable à expliquer* prend ses valeurs (par exemple \mathbb{R}) :

$$f : \mathbb{R}^p \rightarrow \mathbb{R}$$

- Bien entendu, on ne va pas construire cette fonction “au hasard”, on va utiliser les données d’entraînement \mathcal{D}_n .

Définition

Un *algorithme de prévision* est une fonction f construite à partir des données d’entraînement \mathcal{D}_n qui, à toute nouvelle observation x , renvoie une unique prévision $f(x)$.

Exemples

- Données *iris* : un *algorithme de prévision* est une fonction

$$f : \quad \rightarrow$$

construite à partir de la *base de données* de dimension 150x5.

- Données *ozone*, un *algorithme de prévision* est une fonction

$$f : \quad \rightarrow$$

construite à partir de la *base de données* de dimension 112x11.

2.2 Distances

Rappels

- La base de données \mathcal{D}_n est constituée d’individus $(x_i, y_i), i = 1, \dots, n$.
- On veut prédire la valeur de y pour un nouvel individu x à l’aide d’un *algorithme de prévision*.

Importance des distances

- La plupart des algorithmes vont reposer sur des *calculs de distance* entre le nouvel x et les x_i de la base d’entraînement.
- L’idée sous jacente est que le y *inconnu* du nouvel x doit être proche des valeurs de y des x_i qui sont proches de x .
- Il est donc *crucial* de trouver une *distance pertinente* pour le problème d’intérêt (c’est un problème récurrent en statistique).

Distance euclidienne

- Les x étant dans \mathbb{R}^p , une idée naturelle est d'utiliser la *distance euclidienne*.

Définition

Étant donné $x = (x_1, \dots, x_p)$ et $x' = (x'_1, \dots, x'_p)$, la *distance euclidienne* entre x et x' est définie par

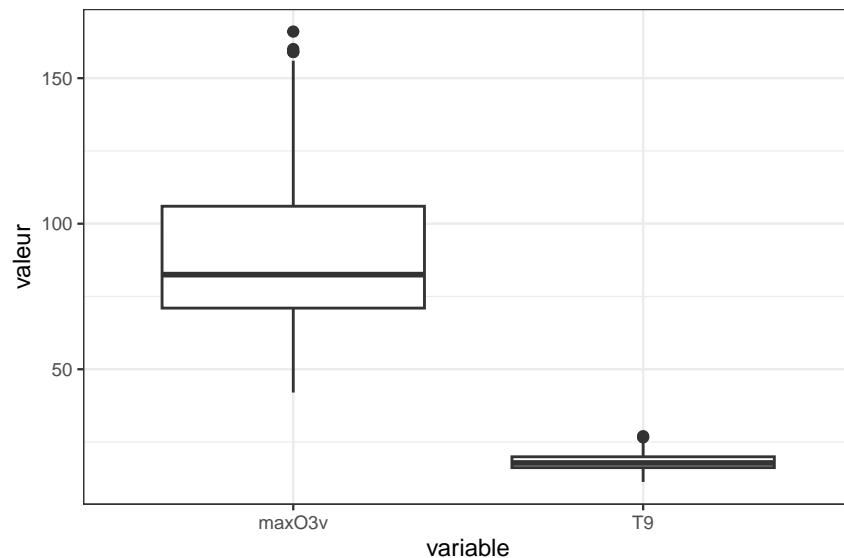
$$d(x, x') = \sqrt{\sum_{j=1}^p (x_j - x'_j)^2}.$$

Un exemple en dimension 3

Si $x = (1.1, -0.8, 3.4)$ et $x' = (0.8, -2.1, 2.8)$ alors

Des problèmes d'échelle...

- On considère les données ozone et on visualise les valeurs des variables T9 et maxO3v à l'aide d'un *boxplot* :



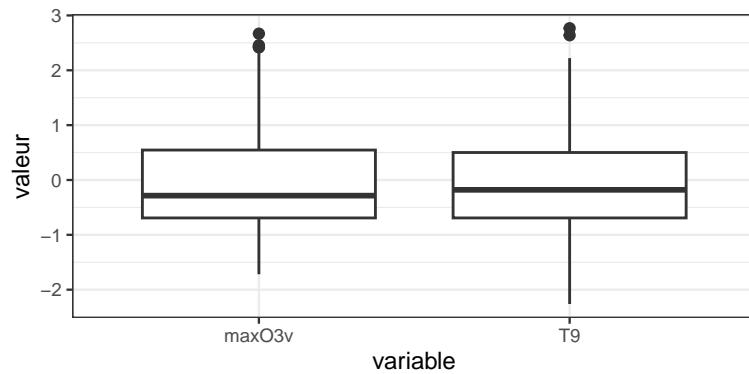
Remarque

- Les valeurs de `max03v` sont *beaucoup plus dispersées* que celles de `T9`.
- Si on utilise ces deux variables pour calculer les distances euclidiennes, `max03v` va *beaucoup plus contribuer à la valeur de la distance* que `T9`.
- Il est souvent nécessaire de *normaliser* les valeurs des variables explicatives en les *divisant par leur écart-type*.
- Cela permet de mettre toutes les variables à la *même échelle*.
- On parle de *réduire les données*.

```
apply(ozone[,-1], 2, sd)
##      T9      T12      T15      Ne9      Ne12      Ne15      Vx9      Vx12
## 3.122726 4.042321 4.530859 2.594916 2.281860 2.332259 2.632742 2.795673
##      Vx15      max03v
## 2.810198 28.276853
```

Réduction

```
ozone_reduit <- scale(ozone[,-1], center=TRUE)
apply(ozone_reduit[,-1], 2, sd)
##      T12      T15      Ne9      Ne12      Ne15      Vx9      Vx12      Vx15      max03v
##          1          1          1          1          1          1          1          1          1
```



⇒ Les 2 variables ont bien été *mises à la même échelle*, elles auront donc la *même contribution* dans le calcul de la distance euclidienne.

Autre exemple

- 2 individus avec deux variables

	Taille (en cm)	Poids (en kg)
x_1	175	70
x_2	158	55

- Nouvel individu

	Taille (en cm)	Poids (en kg)
x	180	56

Question

De qui x est-il le plus proche ?

- On fait la même chose en considérant les *poids en grammes*.
- On a alors

	Taille (en cm)	Poids (en g)
x_1	175	70 000
x_2	158	55 000
x	180	56 000

Problème

Les distances changent en fonction de l'**échelle de la variable** !

```
D1
##      [,1] [,2]
## [1,] 175   70
## [2,] 158   55
D2
##      [,1] [,2]
## [1,] 175 70000
## [2,] 158 55000
```

- Ici encore il faut *réduire les données*

```
scale(D1)
##           [,1]      [,2]
## [1,] 0.7071068 0.7071068
## [2,] -0.7071068 -0.7071068
## attr(,"scaled:center")
## [1] 166.5 62.5
## attr(,"scaled:scale")
## [1] 12.02082 10.60660
scale(D2)
##           [,1]      [,2]
## [1,] 0.7071068 0.7071068
## [2,] -0.7071068 -0.7071068
## attr(,"scaled:center")
## [1] 166.5 62500.0
## attr(,"scaled:scale")
## [1] 12.02082 10606.60172
```

Exercice

1. Expliquer comment ont été obtenues les valeurs dans le tableau ci-dessus ? Par exemple la valeur 0.7071068.
2. De quel individu x est-il le plus proche ?

2.3 L'algorithme des k plus proches voisins

Présentation

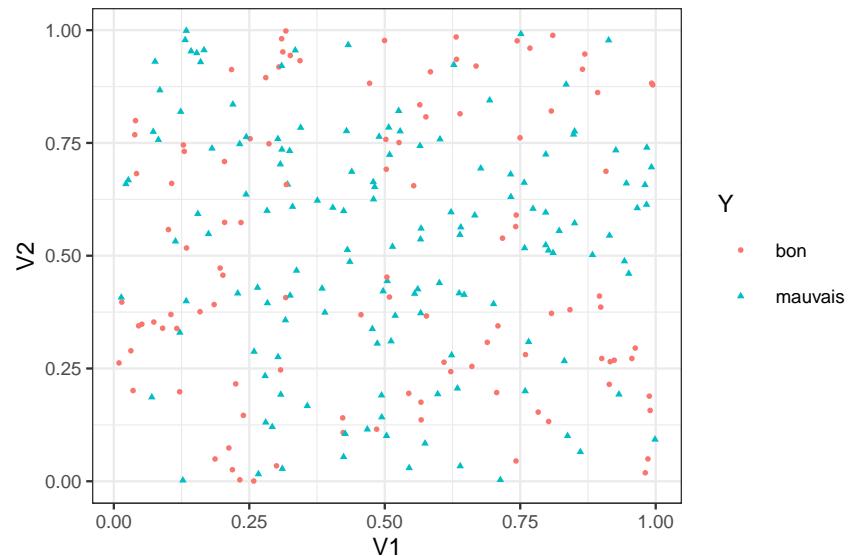
- Appartient aux *algorithmes de référence* en machine learning.

Idée

Prédire le label du nouvel x en se basant sur les **labels des individus** de l'échantillon d'apprentissage qui sont **proches de x** .

- On propose d'illustrer la méthode avec un jeu de données simple où on a :
 - $n = 250$ individus
 - $p = 2$ variables continues
 - y prend 2 valeurs **bon** ou **mauvais**.

Visualisation des données



Le problème

- Les données

```

## # A tibble: 6 x 3
##      V1     V2 Y
## <dbl> <dbl> <fct>
## 1 0.114 0.532 mauvais
## 2 0.622 0.597 mauvais
## 3 0.609 0.264 bon
## 4 0.623 0.280 mauvais
## 5 0.861 0.0651 mauvais
## 6 0.640 0.563 mauvais

```

- On veut *prédir le label* pour une nouvelle observation *x*, par exemple

```

newx
## # A tibble: 1 x 2
##      V1     V2
## <dbl> <dbl>
## 1 0.125 0.38

```

L'algorithme des *k*-ppv

- L'utilisateur (nous) doit *choisir un entier* plus petit que n , on propose de choisir ici $k = 3$.
- On calcule ensuite *la distance (euclidienne)* entre le nouvel individu x et tous les individus x_i de l'échantillon d'apprentissage :

```
#> [1] 0.1523245 0.5424321 0.4980004 0.5084031 0.8004570 0.5468673
```

- On déduit l'individu *le plus proche* de x :

```

## # A tibble: 1 x 3
##      V1     V2 Y
## <dbl> <dbl> <fct>
## 1 0.134 0.399 mauvais

```

- On fait de même avec le *deuxième et le troisième* plus proche voisin (puisque $k = 3$) :

```

## # A tibble: 3 x 3
##      V1     V2 Y
## <dbl> <dbl> <fct>
## 1 0.134 0.399 mauvais
## 2 0.105 0.370 bon
## 3 0.159 0.376 bon

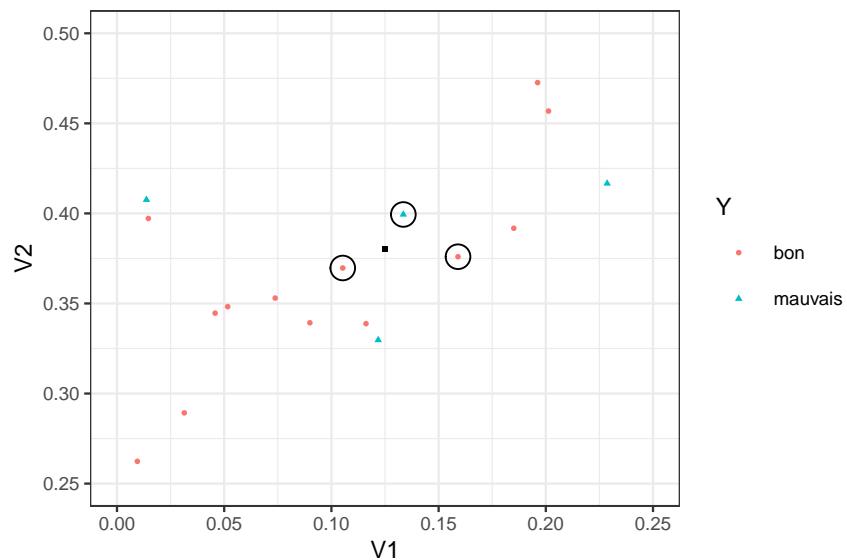
```

La prévision

- Elle s'effectue en **regardant les labels** des 3 ppv et en faisant un **vote à la majorité**.
- On a ici 2 **bon** contre 1 **mauvais** \implies l'algorithme va donc prédire **bon**.
- On note

$$f(x) = \text{bon}$$

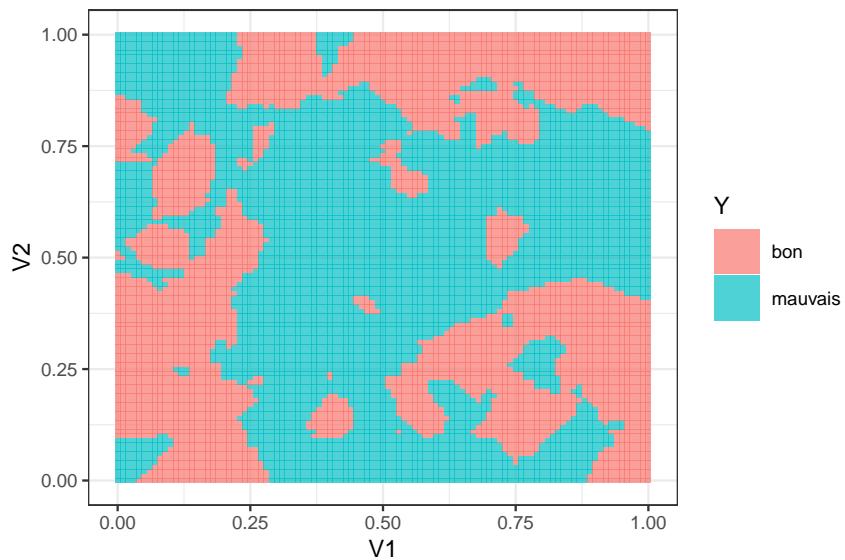
Visualisation



Calcul de l'algorithme de prévision

- On propose de *visualiser en globalité* l'algorithme des 3 plus proches voisins.
- Pour ce faire on propose de calculer pour **tous les points x** du carré $[0, 1]^2$ les **prévisions** de cet algorithme.
- On colorie ensuite *chaque point* en fonction de la **valeur** prédite.

Visualisation de l'algorithme de prévision



Label y continu ?

- L'exemple précédent concerne une variable à expliquer y *qualitative* (bon ou mauvais).

Question

Que se passe t-il si y est *quantitative* (ou continue) ?

Réponse

Le principe est le *même* :

- On cherche les k plus proches voisins du nouvel x .
- **Seule différence** : au lieu de faire un vote à la majorité on calcule la *moyenne* des y_i !

Exemple de l'ozone

- Données (*réduite*)

```

##          T9        T12        T15        Ne9        Ne12
## [1,] -0.8840720 -0.7487743 -0.93308537 -0.35784253 -0.446064660
## [2,] -0.4357457 -0.7735125 -1.08758145  0.02752635 -0.007825696
## [3,] -0.9801419 -0.9714186 -0.69030582 -1.12858029 -0.007825696
## [4,] -0.6919321 -0.4519151 -0.02817977 -1.51394917 -1.760781551

```

- Nouvelle observation à prédire qu'il faut *réduire* aussi (Comment ?)

```

##          T9        T12        T15        Ne9        Ne12
## 1 -0.04079159 -0.226522 -0.09194006  0.2416202  0.259987

```

- On veut prédire à partir de l'algorithme des *3 plus proches voisins*.

1. On calcule la *distance* entre le nouvel x et les x_i de l'échantillon d'apprentissage.

```

## [1] 1.840051 2.176668 2.698325 3.613709 2.316838 2.930628

```

2. On en déduit les *3 plus proches voisins*

```

## # A tibble: 3 x 5
##   max03      T9      T12      T15     Ne9
##   <int>    <dbl>    <dbl>    <dbl>   <dbl>
## 1    77 -0.692 -0.180 -0.116  0.413
## 2    88 -0.788 -0.600 -0.249  0.413
## 3    88 -0.468 -0.303 -0.425  0.413

```

3. On prédit la concentration en ozone par la *concentration moyenne des 3 plus proches voisins* :

```

(77+88+88)/3
## [1] 84.33333

```

2.4 Evaluation de la performance - choix de k

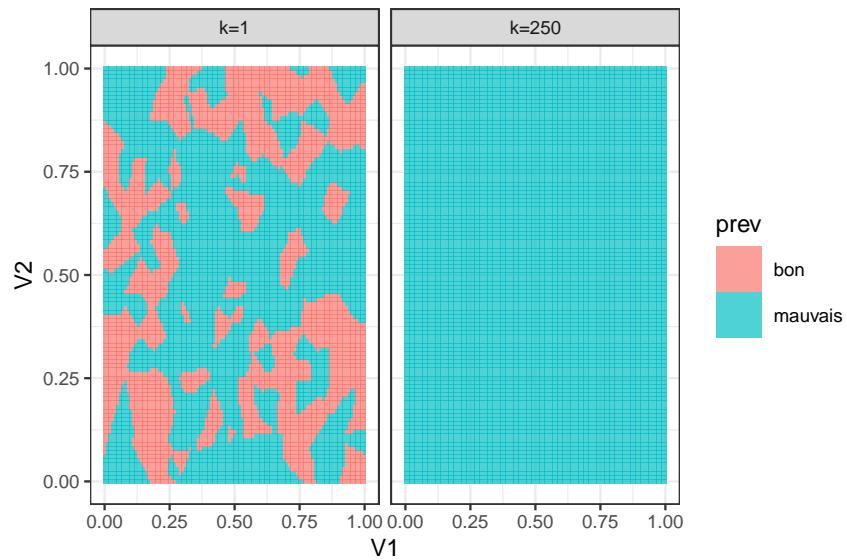
- Dans les exemples précédents on a utilisé l'algorithme des k -ppv avec $k = 3$.
- Ce paramètre doit être *choisi par le datascientist*.

Plusieurs questions

- La valeur de ce paramètre est-elle importante pour la qualité des prévisions (prédirer est facile, on veut bien prédire !).
- Si oui, comment choisir cette valeur ?

- On reprend l'exemple jouet en dimension 2 (bon-mauvais)
- On propose de visualiser les cas extrêmes :
 - $k = 1 \implies$ on utilise uniquement le plus proche voisin, c'est-à-dire que la prévision est calculée avec une seule observation ;
 - $k = n$ (ici 250) \implies la prévision est calculée avec toutes les observations.

Visualisation des cas extrêmes



Commentaires $k = 250$

- L'algorithme utilise toutes les données d'apprentissage pour prédire.
- Quel que soit l'endroit où on veut prédire, il va donc toujours renvoyer la même prévision !

- Cette prévision correspond au *label majoritaire* dans la base de données :

```
##      bon mauvais
##      108     142
```

Vocabulaire

- L'algorithme utilise *trop peu les données* pour prédire.
- On dit qu'il *sous-apprend*.

Commentaires $k = 1$

- L'algorithme utilise *uniquement le plus proche voisin* pour prédire.
- Les prévisions sont donc calculées à partir d'*une seule observation*.
- Il a donc tendance à prédire *exactement ce qui est observé*.

Vocabulaire

- L'algorithme dépend *trop des données* pour calculer les prévisions
- On dit qu'il *sur-apprend* (*overfitting* en anglais).

- Le *sous/sur-apprentissage* est un *problème récurrent en machine learning*.
- Sous-apprendre est *mauvais*, sur-apprendre est *mauvais*.
- Il faut donc *éviter ces deux situations*.
- Pour les k -ppv, cela s'effectue en *sélectionnant la meilleure valeur de k* .

Comment ?

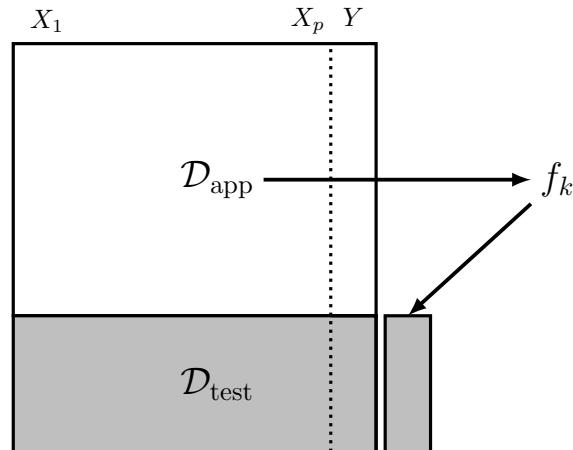
- Ce n'est *pas au dataScientist de décider* de la meilleure valeur de k .
- *Les données doivent la sélectionner automatiquement !*
- Le dataScientist doit néanmoins dire aux données *comment trouver cette meilleure valeur*.

Choisir k

- La manière de choisir k est *naturelle*.
- Vu qu'on ne souhaite pas choisir une valeur de k “au hasard” on va *toutes les tester*, c'est-à-dire
 1. Entraîner l'algorithme avec $k = 1, 2, \dots, n$;
 2. Evaluer les *performances* pour chaque valeur de k ;
 3. Choisir la valeur qui possède la *meilleure performance*.

Couper l'échantillon en 2

- Il n'est *pas recommandé* d'utiliser les mêmes données pour *entraîner* les algorithmes et pour les *tester*.
- Une “astuce” est donc de *scinder* la base de données \mathcal{D}_n en *deux* :
 - une *base d'apprentissage* \mathcal{D}_{app} pour *entraîner* les algorithmes pour chaque valeur de k ;
 - une *base test* $\mathcal{D}_{\text{test}}$ pour *tester* les algorithmes et en déduire la meilleure valeur de k .



Comment découper ?

L'échantillon d'*apprentissage* est généralement *privilégié*, on utilise souvent :

- 3/4 des données pour l'apprentissage et 1/4 pour le test.
- ou 2/3 des données pour l'apprentissage et 1/3 pour le test.

Application

- On coupe l'échantillon en deux :

```
dim(dapp);dim(dtest)
## [1] 150   3
## [1] 100   3
```

- On entraîne la *règle des kppv pour chaque valeur de k* sur les données d'*apprentissage* et on *calcule les prévisions* sur les *données test* :

```
for (k in 1:150){
  prev[,k] <- kknn(Y~.,train=dapp,test=dtest,k=k,
                     kernel="rectangular")$fitted.values
}
```

Question ?

Pourquoi *k* ne va t-il pas au delà de 150 dans la boucle ?

- On a ainsi pour *chaque valeur de k* les prévisions de *chaque individu* de l'échantillon *test* :

```
prev[1:5,1:8]
## # A tibble: 5 x 8
##   `k=1`  `k=2`  `k=3`  `k=4`  `k=5`  `k=6`  `k=7`  `k=8`
##   <fct>  <fct>  <fct>  <fct>  <fct>  <fct>  <fct>
## 1 bon    bon    bon    bon    bon    bon    bon    bon
## 2 mauvais mauvais mauvais mauvais bon    bon    bon    bon
## 3 mauvais mauvais bon    bon    bon    bon    bon    bon
## 4 bon    bon    bon    bon    bon    bon    bon    bon
## 5 mauvais mauvais mauvais mauvais mauvais mauvais mauvais
```

Comment calculer la performance ?

- On va, pour chaque valeur de k , confronter les prévisions aux valeurs observées et...
- Compter le **nombre (ou la proportion)** d'erreur.
- Il restera à choisir la valeur de k où on s'est le *moins trompé*.

Sélection de k

- Pour $k = 1$ on a par exemple

```
## # A tibble: 4 x 2
##   `k=1`    obs
##   <fct>    <fct>
## 1 bon      mauvais
## 2 mauvais  bon
## 3 mauvais  mauvais
## 4 bon      bon
```

- La *proportion d'erreur* est donc

```
mean(prev[,1] != dtest[,3])
## [1] 0.41
```

Interprétation

Avec $k = 1$ l'algorithme se **trompe dans 41% des cas**.

- Il reste à faire la même chose pour **chaque valeur de k**

```
## # A tibble: 1 x 8
##   `k=1` `k=2` `k=3` `k=4` `k=5` `k=6` `k=7` `k=8`
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  0.41  0.41  0.36  0.36  0.31  0.31  0.31  0.31
```

- Et à *choisir la valeur* de k qui possède **la plus petite erreur**

```
## # A tibble: 1 x 2
##   k     erreur
##   <chr> <dbl>
## 1 k=5    0.31
```

Conclusion

On choisira donc **$k=5$** . Le pourcentage d'erreur de prévision pour cette valeur est estimé à **0.31**.

Remarques importantes

- La procédure est totalement *automatique* dans le sens où le datascientist ne choisit rien.
- Seules *les données décident*.
- On peut maintenant utiliser l'algorithme des k -ppv avec $k=5$ pour prédire de nouveaux individus.

Validation hold out

- La méthode utilisée pour choisir k s'appelle *validation hold out*.
- Elle s'applique également à *tous les autres algorithmes de Machine Learning*.
- Dans le cas des k -ppv, on peut la résumer ainsi.

Algorithm

1. Séparer les données en un échantillon d'apprentissage \mathcal{D}_{app} et un échantillon test $\mathcal{D}_{\text{test}}$.
2. Pour $k = 1, \dots, n$
 - a. Entraîner l'algorithme des k -ppv sur \mathcal{D}_{app} ;
 - b. Prédire le label des individus de $\mathcal{D}_{\text{test}}$;
 - c. En déduire l'erreur pour la valeur k en confrontant les prévisions aux valeurs observées.
3. Choisir la valeur de k qui minimise l'erreur.

Choisir k : y continue

- Exemple de l'*ozone* : `y=max03`.
- La méthodologie est similaire.

Seul changement

- La manière d'évaluer la performance.

- y qualitative : proportion d'erreur de prévision.
- y continue \Rightarrow prévisions et observations sont des nombres \Rightarrow utiliser une distance (euclidienne par exemple).

Exemple

- Pour $k = 1$ on obtient les prévisions suivantes :

```
## # A tibble: 3 x 2
##   `k=1`    obs
##   <dbl> <int>
## 1     88    101
## 2     88     72
## 3     79     83
```

- L'erreur pour $k = 1$ s'obtient donc avec

$$\sqrt{(88 - 101)^2 + (88 - 72)^2 + \dots}$$

- On trouve

```
## [1] 95.36247
```

- On fait de même pour les autres valeurs de k :

```
## # A tibble: 1 x 8
##   `k=1` `k=2` `k=3` `k=4` `k=5` `k=6` `k=7` `k=8`
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  95.4  97.2  94.7  96.7  97.0  100.  105.  104.
```

- Et on choisit la valeur de k qui possède la plus petite erreur

```
## # A tibble: 1 x 2
##   k     erreur
##   <chr> <dbl>
## 1 k=48    82.9
```

- On utilisera donc ici **k=48**.

2.5 Une autre méthode : les arbres

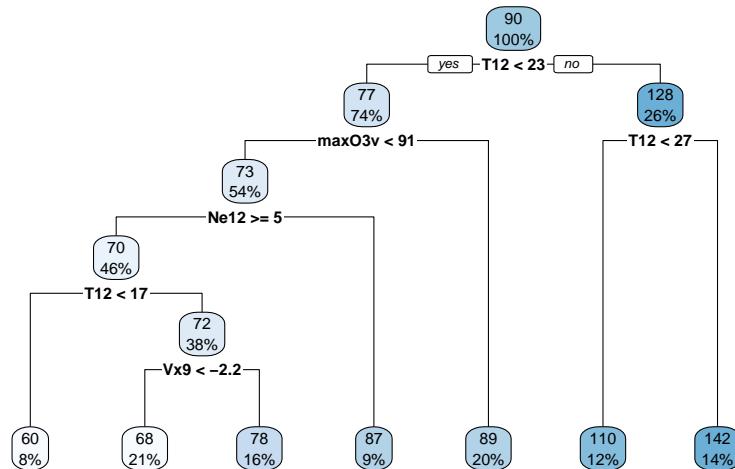
- Algorithme de *Machine Learning* qui permet de faire de la *prévision*.

Idée

- Trouver une suite de **questions pertinentes** à poser aux données à prédire.
- Prédire en fonction des **réponses à ces questions**.
- L'algorithme peut se visualiser sous la forme d'un *arbre*.

Exemple de l'ozone

```
arbre <- rpart(maxO3~., data=ozone)
rpart.plot(arbre)
```



Calcul de prévision

- Nouvelle observation

```
xnew
## # A tibble: 1 x 10
##      T9    T12    T15    Ne9   Ne12   Ne15    Vx9    Vx12    Vx15 maxO3v
##      <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  18.2  20.6  22.2  5.56  5.61  5.28 -1.26 -1.83 -2.27  93.5
```

- *Question 1* : $T12 < 23$? \Rightarrow Réponse : Oui !
- *Question 2* : $\max(03v) < 91$? \Rightarrow Réponse : Non !

La prévision

- `xnew` se trouve dans le **5ème bloc (appelé feuille)** de la dernière ligne.
- On regarde la valeur de ce bloc pour obtenir la *prévision*, elle est donc de 89

- Les logiciels possèdent bien entendu des *fonctions* pour **calculer** directement les **prévisions** :

```
predict(arbre,newdata=xnew)
## 1
## 89
```

Algorithme automatique

- **Question importante** : comment est construit l'arbre ? C'est-à-dire
 1. comment trouver les *bonnes questions* ?
 2. comment calculer les *valeurs prédites* ?
- Comme pour les k -ppv, ce n'est **pas le data scientist** qui choisit ces paramètres “au hasard”.
- Ce sont les *données* qui choisissent **automatiquement** les meilleures questions et les meilleures valeurs prédites.
- L'algorithme sera présenté en détails **plus tard dans le cursus**.

Exercice

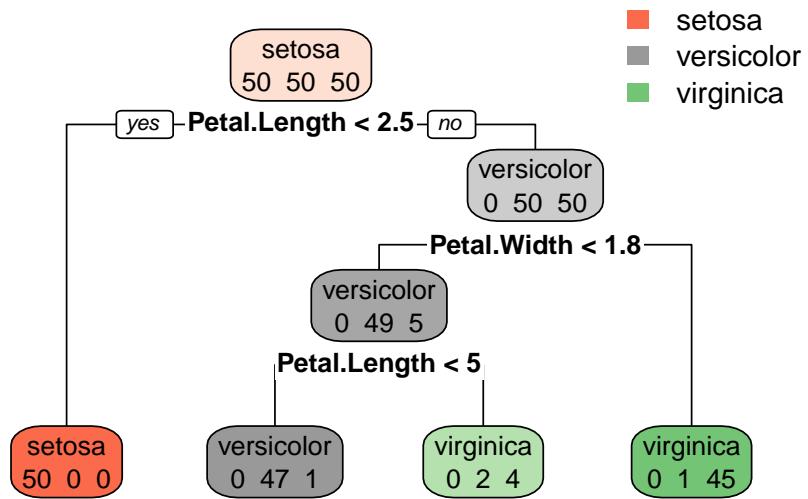
- On considère les données suivantes

	V_1	V_2	Y
1	1	2	4
2	3	1	6
3	2	3	5
4	4	2	1
5	1	3	8

1. Quelles sont les valeurs de n et p ? La variable à expliquer est-elle **continue** ou **qualitative** ?
2. On considère le nouvel individu $x = (3, 2)$. Calculer la valeur prédictive de Y par l'**algorithme des 2-ppv**.

Exercice

- On a construit l'**arbre** suivant pour les iris de Fisher



- On considère les individus suivants :

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1       4.9      3.1        1.5       0.1
## 2       5.7      2.6        3.5       1.0
## 3       7.7      3.8        6.7       2.2
```

1. Quelles sont les *espèces prédites par l'arbre* pour ces 3 iris ?
2. Proposez un indicateur numérique qui permette de *quantifier la confiance* en la prévision.

3 Apprentissage non supervisé

Rappels

Wikipédia

- L'apprentissage non supervisé désigne la situation d'apprentissage automatique où les données ne sont *pas étiquetées* (pas de Y).
- Il s'agit donc de découvrir les structures sous-jacentes à ces données non étiquetées en regroupant les individus en fonction de *leurs proximités*.

3.1 Écriture du problème

Rappel : supervisé

- Les *données* :

$$\mathcal{D}_n = \begin{pmatrix} V_1 & \dots & V_p & Y \\ x_{11} & \dots & x_{1p} & y_1 \\ x_{21} & \dots & x_{2p} & y_2 \\ \vdots & & \vdots & \vdots \\ x_{n1} & \dots & x_{np} & y_n \end{pmatrix}$$

Le problème

- Expliquer/prédire Y par V_1, \dots, V_p .
- Prédire le y d'un nouvel $x = (x_1, \dots, x_p)$.

Non supervisé

- Les *données* (pas de Y) :

$$\mathcal{D}_n = \begin{pmatrix} V_1 & \dots & V_p \\ x_{11} & \dots & x_{1p} \\ x_{21} & \dots & x_{2p} \\ \vdots & & \vdots \\ x_{n1} & \dots & x_{np} \end{pmatrix}$$

Le problème

- Regrouper de façon **homogène** les individus.
- Regrouper les **lignes** du tableau.

Exemple

```
dim(USArrests)
## [1] 50  4
head(USArrests)
##           Murder Assault UrbanPop Rape
## Alabama     13.2    236      58 21.2
## Alaska      10.0    263      48 44.5
## Arizona      8.1    294      80 31.0
## Arkansas     8.8    190      50 19.5
## California   9.0    276      91 40.6
## Colorado     7.9    204      78 38.7
```

Description

- $n = 50$ individus.
- $p = 4$ variables :
 - **Murder** : nombre d'*arrestations pour meurtre* pour 100 000 habitants.
 - **Assault** : nombre d'*arrestations pour aggression* pour 100 000 habitants.

- UrbanPop : pourcentage de la population urbaine.
- Rape : nombre d'*arrestations pour viol* pour 100 000 habitants.

Le problème “métier”

Le problème métier

- Existe t-il des *similitudes entre certains états* en terme de **criminalité** ?
- Peut-on *regrouper les états* en fonction de ces **similitudes** ?

Le problème science des données

Regrouper les lignes du tableau de sorte que

- 2 états x_1 et x_2 soient dans *le même groupe* si ils sont **proches**.
- 2 états x'_1 et x'_2 soient dans *deux groupes différents* si ils sont **éloignés**.

L'approche

Questions

1. Comment **affecter** les états dans des groupes ?
2. Comment choisir le **nombre de groupes** ?
3. Comment **interpréter** les groupes ?

Réponses

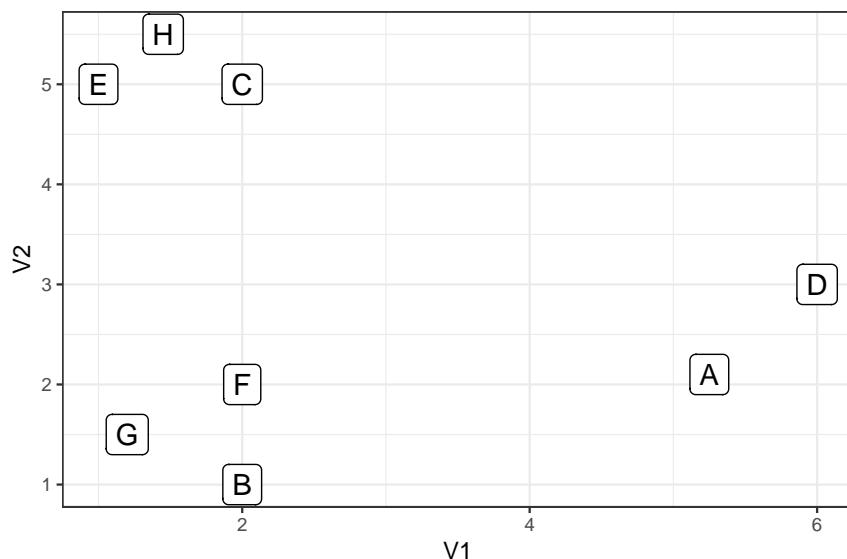
- La notion de **distance** va encore être **primordiale** pour les questions 1 et 2 :
 - 2 états doivent être dans le *même groupe* si la distance entre ces deux états est **petite** ;
 - 2 états doivent être dans *deux groupes différents* si la distance entre ces deux états est **grande**.
- 1 état = 1 **vecteur** (de dimension 4) \implies on va (encore) devoir calculer des **distances entre vecteurs**.

3.2 La classification ascendante hiérarchique (CAH)

Principe de la méthode

- Algorithme *itératif* qui permet de constituer des groupes (ou clusters).
- Au début *chaque individu forme un groupe* (on a donc n groupes) puis
- on va *regrouper les individus un à un* en fonction de leurs proximités

Un exemple avec 8 données

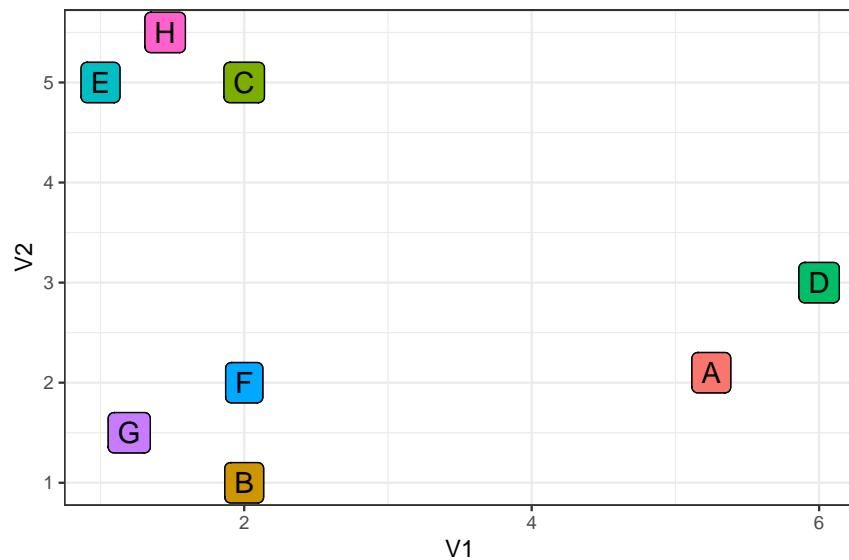


Calcul des distances

- L'algorithme va se baser sur les *distances entre les 8 individus*.
- On obtient après calcul :

```
##      A     B     C     D     E     F     G
## B 3.43
## C 4.36 4.00
## D 1.17 4.47 4.47
## E 5.15 4.12 1.00 5.39
## F 3.25 1.00 3.00 4.12 3.16
## G 4.09 0.94 3.59 5.03 3.51 0.94
## H 5.10 4.53 0.74 5.19 0.67 3.54 4.01
```

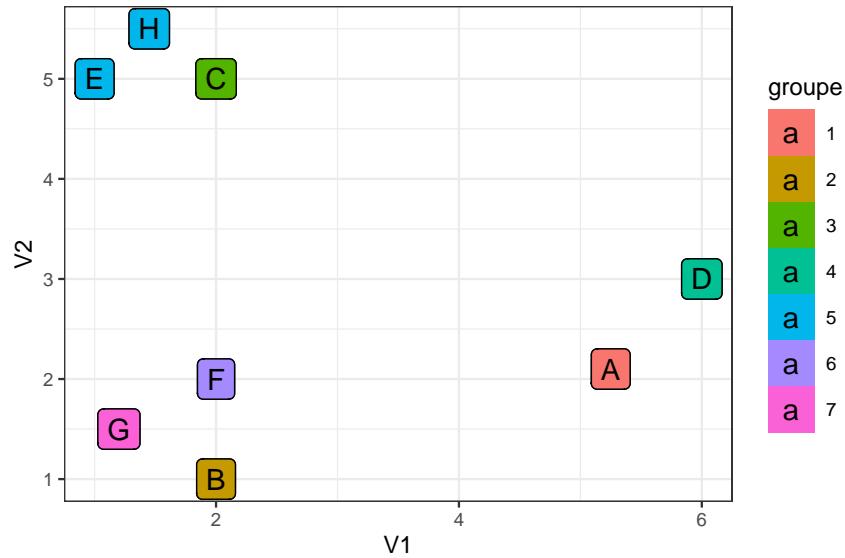
Étape 1



- 8 groupes de taille 1

Étape 2

- Recherche des *deux individus les plus proches*.
- E et H avec une distance de 0.67.
- On *assemble ces deux individus* dans un *même groupe*.



- *7 groupes* : 6 de taille 1 et 1 de taille 2 ($\{E, H\}$).

Étape 3

Objectif

- Obtenir **6 groupes**;
- **2 possibilités** :
 - Assembler 2 individus isolés.
 - Assembler 1 individu au groupe $\{E, H\}$

Problème

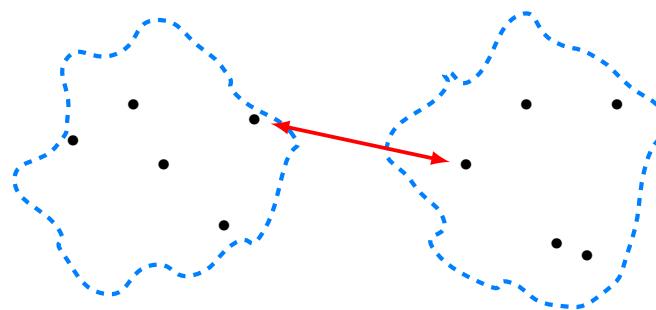
- On sait calculer des distances entre 2 individus.
- On *ne sait pas* calculer des distances entre **2 groupes d'individus**.

La distance du saut minimum

- Elle permet de calculer des *distances entre deux groupes d'individus*.

Définition

La *distance du saut minimum* entre 2 groupes C_1 et C_2 est définie comme la **distance minimale** entre 1 individu de G_1 et 1 individu de G_2 .



Exercice

- Compléter le tableau suivant en calculant la distance du saut minimum entre le groupe $\{E, H\}$ et les autres individus.

	dist avec $\{E, H\}$
A	
B	
C	
D	
F	
G	

Mise à jour de la matrice de distance

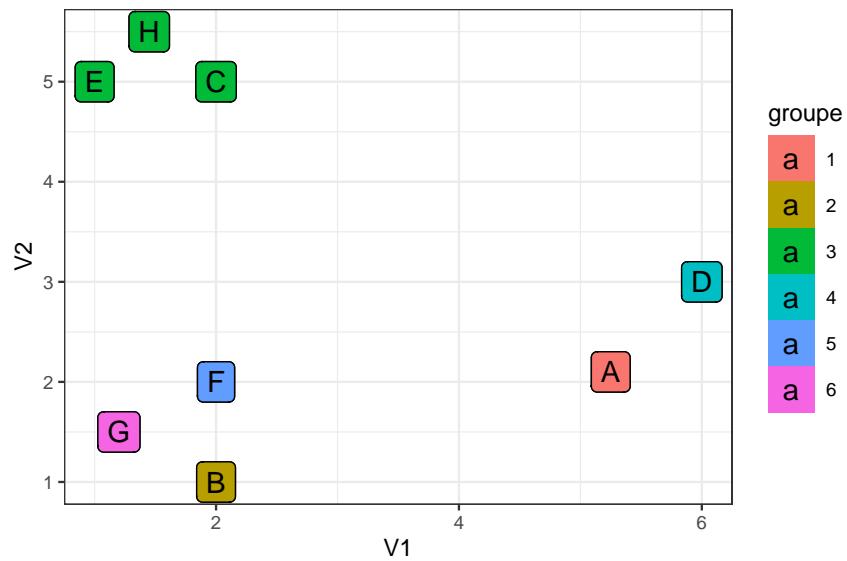
- On calcule les *distances entre les 7 groupes* de l'étape 2

	A	B	C	D	$\{E, H\}$	F
B						
C						
D						
$\{E, H\}$						
F						
G						

Décision étape 3

On assemble ... et

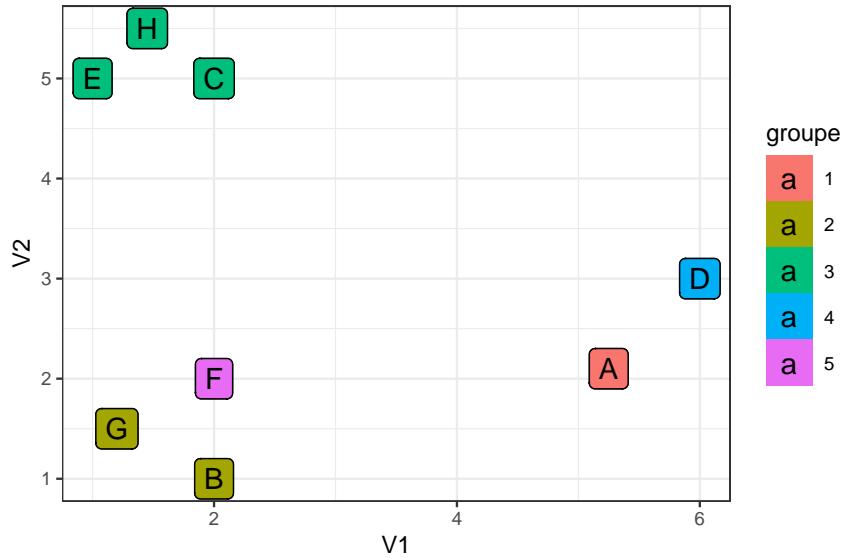
Les 6 groupes



Itération

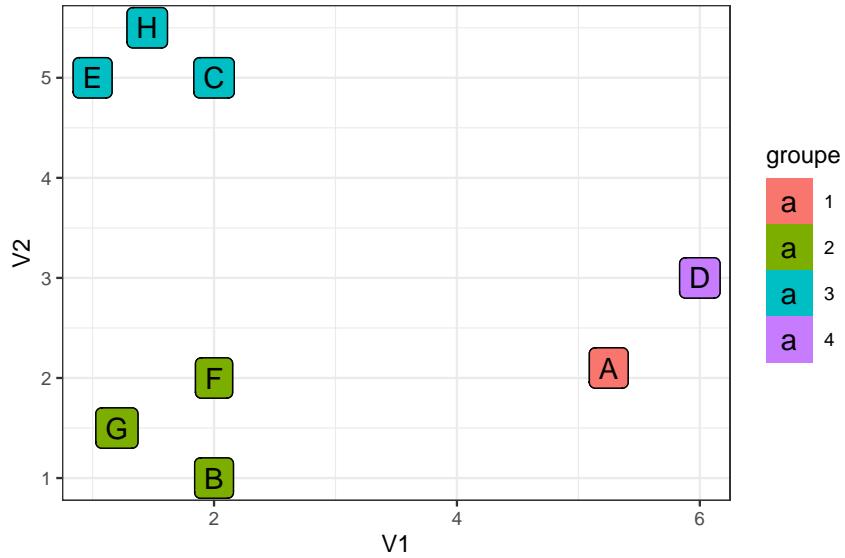
Le procédé est **répété** jusqu'à **obtenir un seul groupe**.

Étape 4 - 5 groupes



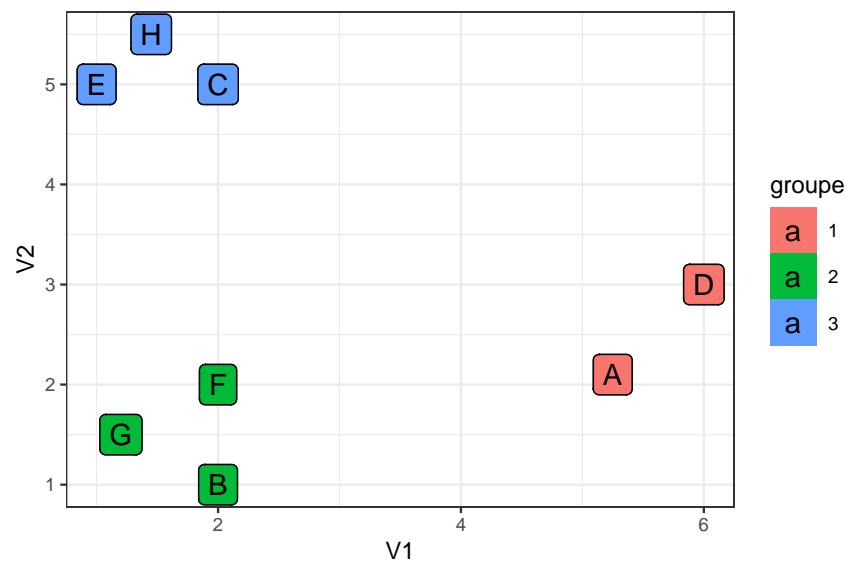
$$d(B, G) = 0.94.$$

Étape 5 - 4 groupes



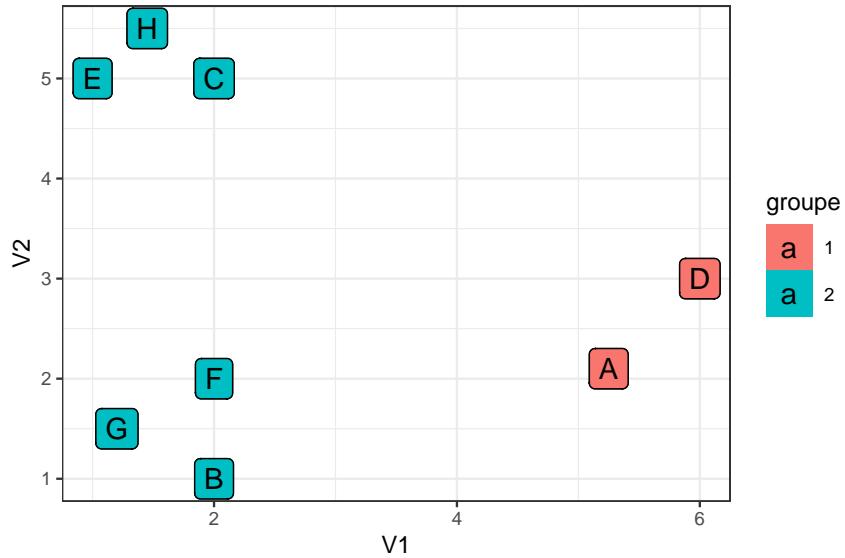
$$d(F, \{B, G\}) = 0.94.$$

Étape 6 - 3 groupes



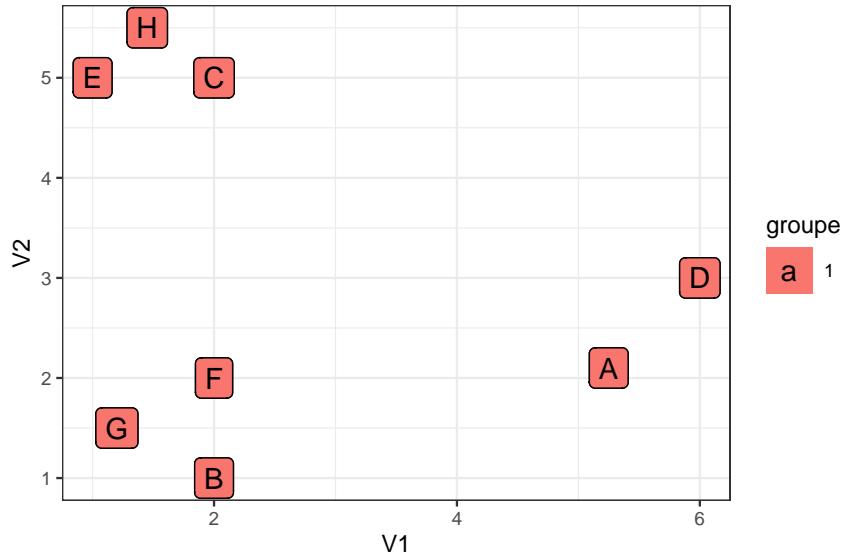
$$D(A, D) = 1.17.$$

Étape 7 - 2 groupes



$$d(\{C, E, H\}, \{B, F, G\}) = 3.00.$$

Étape 8 - 1 groupe



$$d(\{C, E, H, B, F, G\}, \{A, D\}) = 3.25.$$

Commentaires

- Ce procédé *ne répond pas* entièrement à la question initiale.
- Il permet d'obtenir uniquement *une suite de groupes* : 8 groupes, puis 7, puis 6...
- On peut visualiser cette suite à l'aide d'un graphique appelé *dendrogramme*.

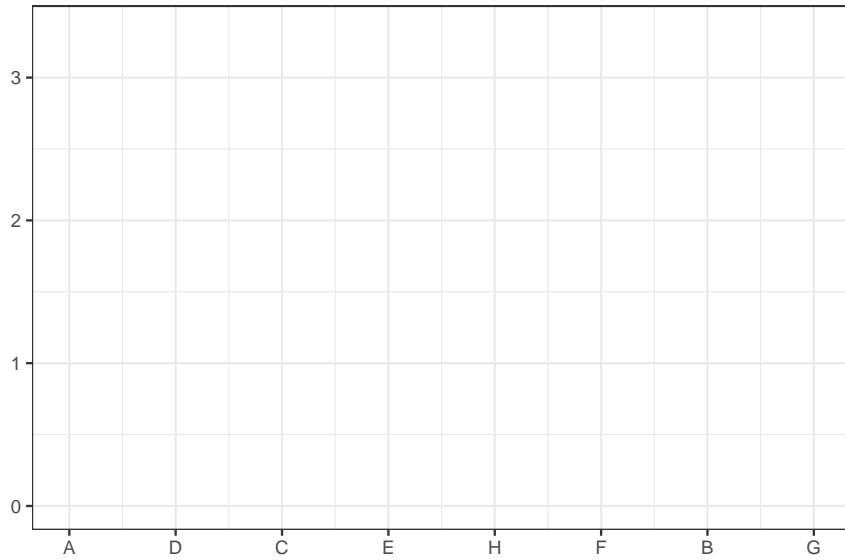
Le dendrogramme

- Outil de *visualisation* qui permet de représenter le process *d'agglomération de la CAH*.

Comment ?

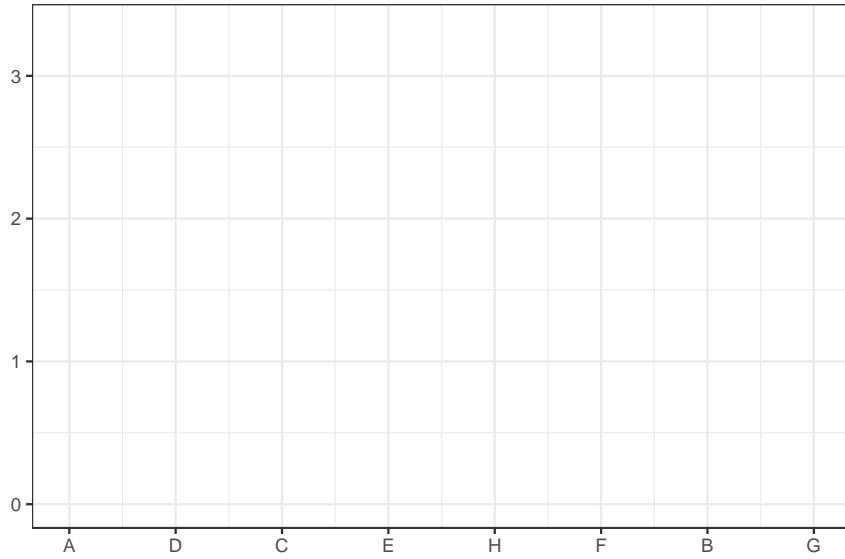
1. Les *individus* sont placés sur l'axe des *abscisses* (étape 1).
2. Les deux individus les plus proches sont assemblés (étape 2) en utilisant un *segment de hauteur égale à la distance entre ces deux individus*.
3. Les deux groupes sont ensuite assemblés (étape 3) en utilisant un *segment de hauteur égale à la distance entre ces deux groupes*.
4. ...

Étape 1



Exercice

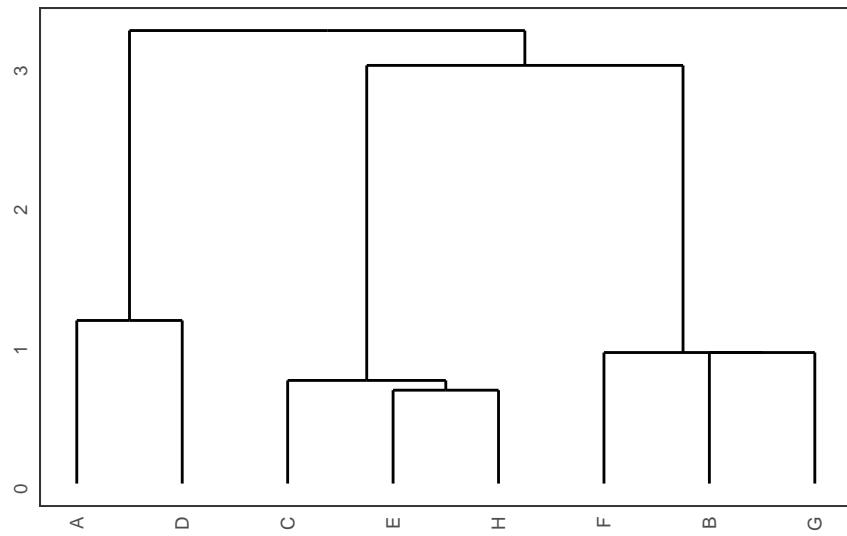
- Construire le dendrogramme.



Fonctions R

- Il existe bien entendu des *fonctions* qui permettent d'obtenir le **dendrogramme**, par exemple

```
ggdendrogram(hc)
```

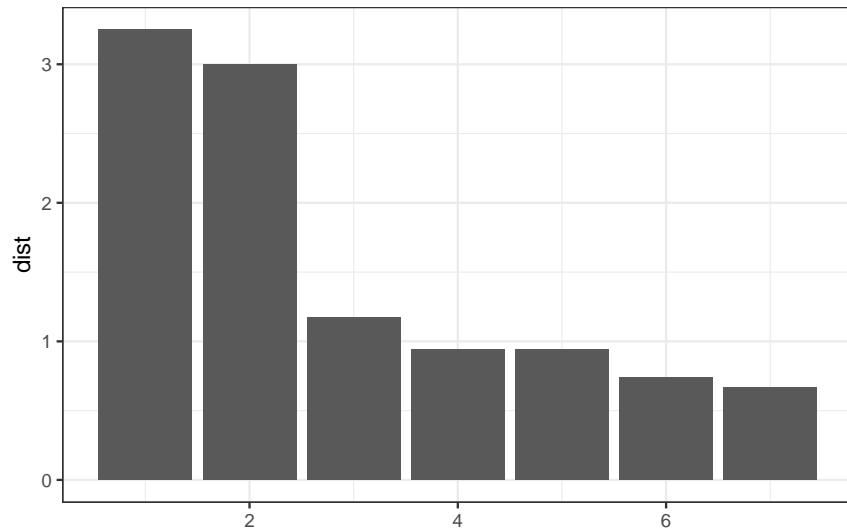


Choix du nombre de classes

- Le dendrogramme permet de *visualiser le process d'agglomération* des individus.
- Il ne permet *pas d'obtenir faire le clustering* (d'obtenir les groupes, ni même le nombre de groupes) mais...
- on l'utilise pour choisir les groupes car *la hauteur des segments* correspond à la *distance entre deux groupes*.

Règle empirique

- Sélectionner la coupure qui correspond à un *saut important*.
- *Outils* : diagramme en barres des distances entre groupes.

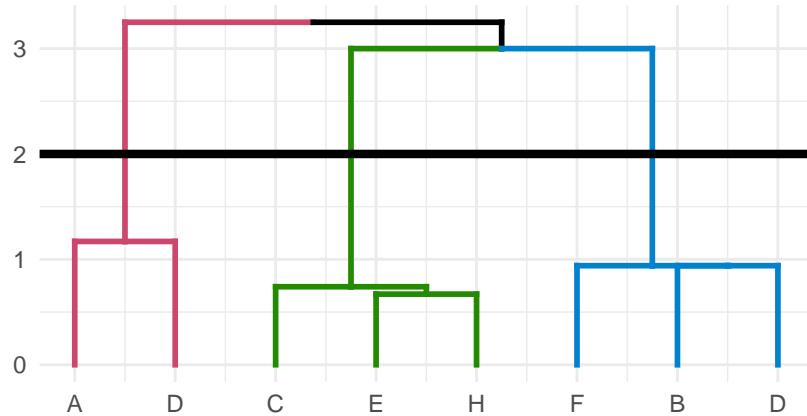


Commentaires

- *Lecture* : la 3ème barre représente la distance entre les deux groupes assemblés lorsqu'on passe de 4 à 3 groupes \Rightarrow on peut l'interpréter comme une *mesure de la difficulté à passer à 3 groupes*.
- *Saut important* entre les 2ème et 3ème barres \Rightarrow relativement *facile de passer à 3 groupes* mais *plus difficile de passer à 2 groupes*.

Application

- Le saut est important lorsqu'on passe de *3 à 2 groupes* \implies on choisit donc *3 groupes*.



Conclusion

On choisit *3 groupes* : $\{\{A,D\}, \{C,E,H\} \text{ et } \{F,B,D\}\}$

Un exemple sur des données réelles

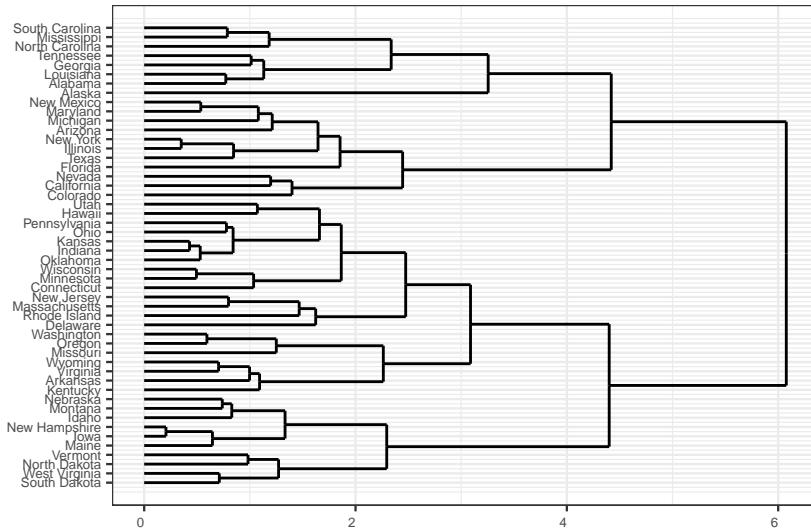
```
head(USArrests)
##          Murder Assault UrbanPop Rape
## Alabama    13.2     236      58 21.2
## Alaska     10.0     263      48 44.5
## Arizona     8.1     294      80 31.0
## Arkansas    8.8     190      50 19.5
## California   9.0     276      91 40.6
## Colorado    7.9     204      78 38.7
```

La méthode étant basée sur des distances, on commence par *centrer-réduire les données* :

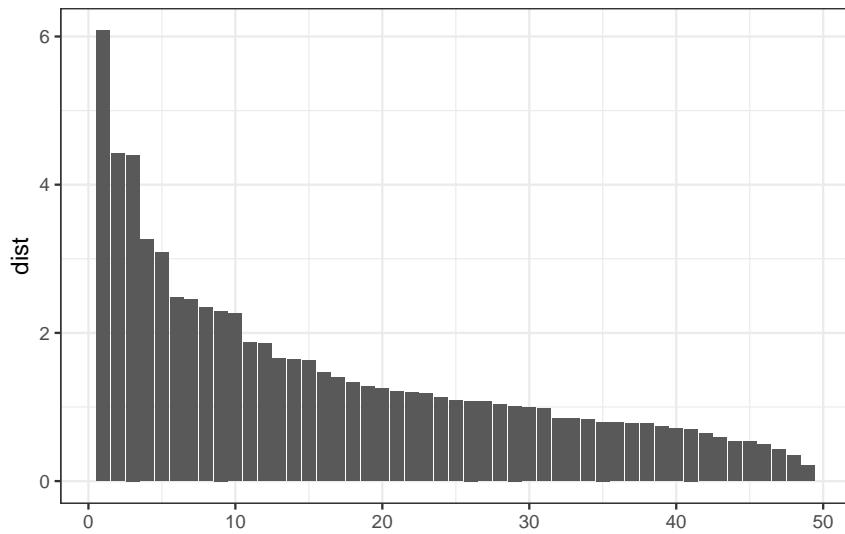
```
USArrests2 <- scale(USArrests); head(USArrests2)
##          Murder Assault UrbanPop      Rape
## Alabama  1.24256408  0.7828393 -0.5209066 -0.003416473
## Alaska   0.50786248  1.1068225 -1.2117642  2.484202941
## Arizona  0.07163341  1.4788032  0.9989801  1.042878388
## Arkansas 0.23234938  0.2308680 -1.0735927 -0.184916602
## California 0.27826823  1.2628144  1.7589234  2.067820292
## Colorado  0.02571456  0.3988593  0.8608085  1.864967207
```

- On effectue la CAH et on visualise le dendrogramme :

```
hc_USA <- hclust(dist(USArrests2))
ggdendrogram(hc_USA, rotate = TRUE, theme_dendro = FALSE) +
  xlab("") + ylab("") + theme(axis.text = element_text(size=5))
```

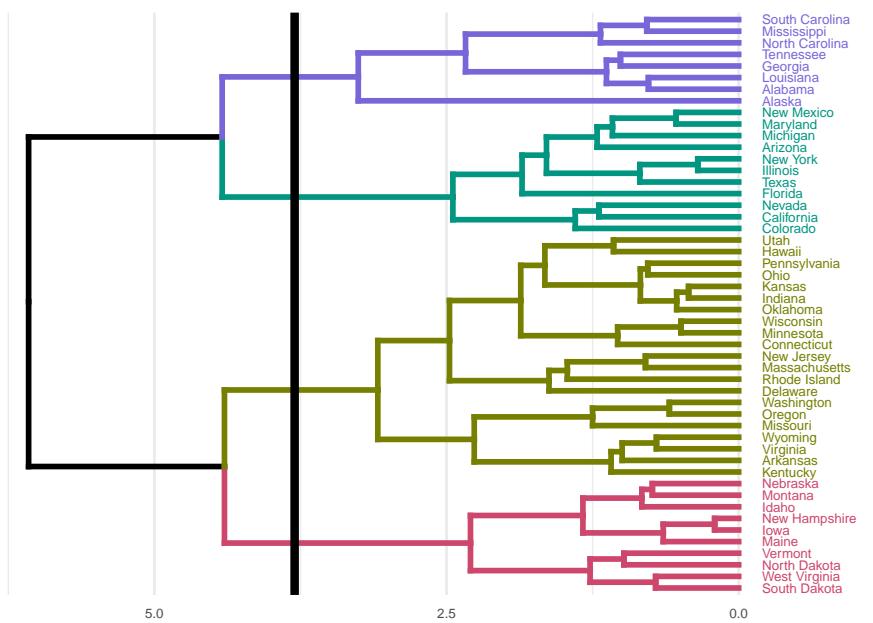


Choix du nombre de groupes



Commentaires

- Saut important pour passer de *4 à 3 groupes* ou de *6 à 5 groupes* \Rightarrow choisir 4 ou 6 groupes.
- On choisit 4 groupes.



Les groupes

```
## # A tibble: 6 x 2
##   etat     groupe
##   <chr>    <fct>
## 1 Alabama  1
## 2 Alaska   1
## 3 Arizona  2
## 4 Arkansas 3
## 5 California 2
## 6 Colorado  2
```

Interprétation

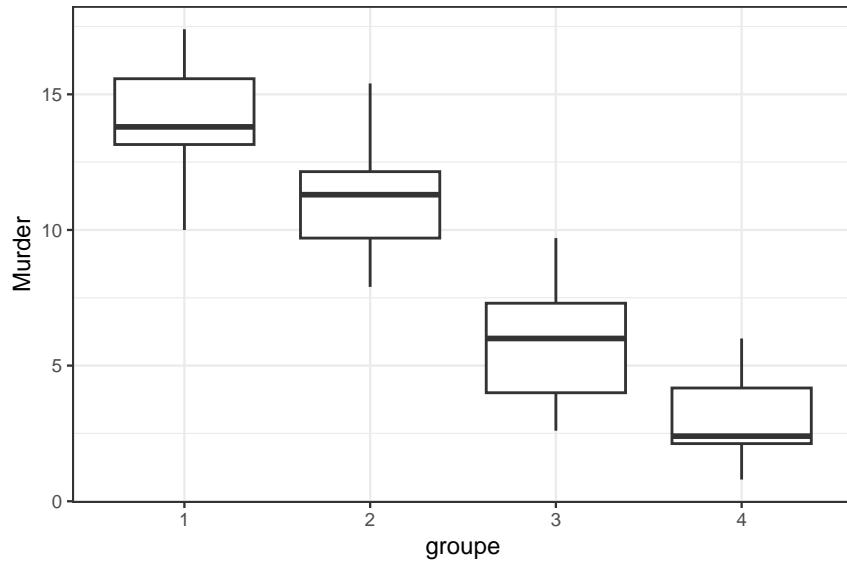
- Le travail n'est pas fini \implies faire des groupes n'est pas une fin en soi.
- Il faut savoir les interpréter.
- Comment ? Avec les outils de statistiques descriptives standards.

Les groupes

- *Groupe 1* : Alabama, Alaska, Georgia, Louisiana, Mississippi, North Carolina, South Carolina, Tennessee
- *Groupe 2* : Arizona, California, Colorado, Florida, Illinois, Maryland, Michigan, Nevada, New Mexico, New York, Texas
- *Groupe 3* : Arkansas, Connecticut, Delaware, Hawaii, Indiana, Kansas, Kentucky, Massachusetts, Minnesota, Missouri, New Jersey, Ohio, Oklahoma, Oregon, Pennsylvania, Rhode Island, Utah, Virginia, Washington, Wisconsin, Wyoming
- *Groupe 4* : Idaho, Iowa, Maine, Montana, Nebraska, New Hampshire, North Dakota, South Dakota, Vermont, West Virginia

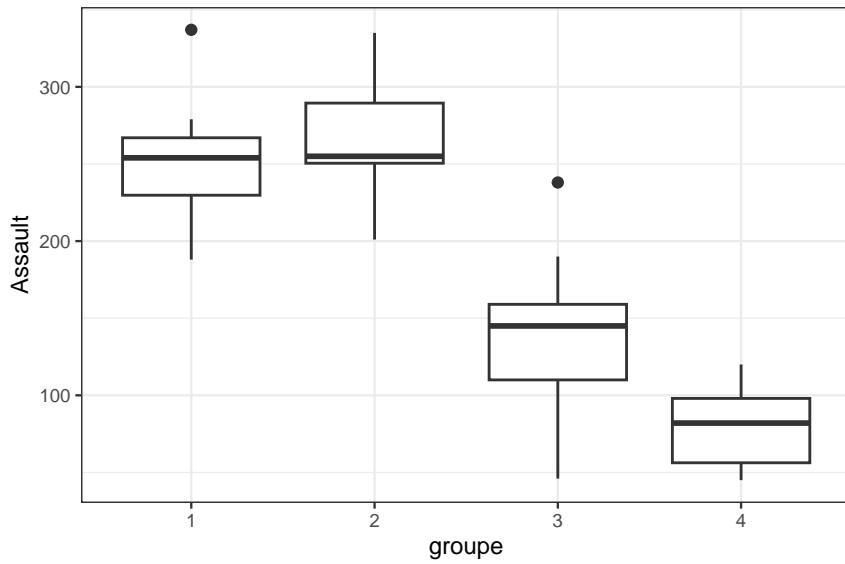
Arrestations pour meurtre

```
ggplot(res1)+aes(x=groupe,y=Murder)+geom_boxplot()
```



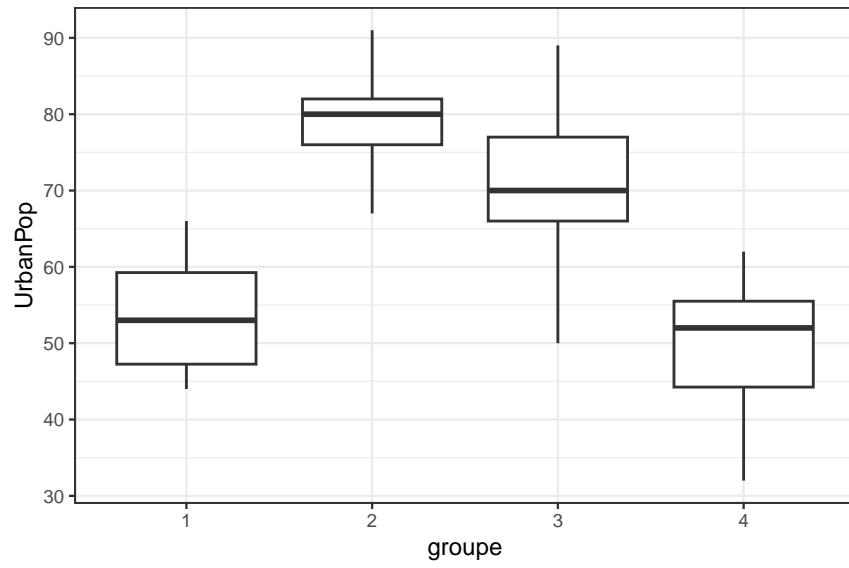
Arrestations pour aggression

```
ggplot(res1)+aes(x=groupe,y=Assault)+geom_boxplot()
```



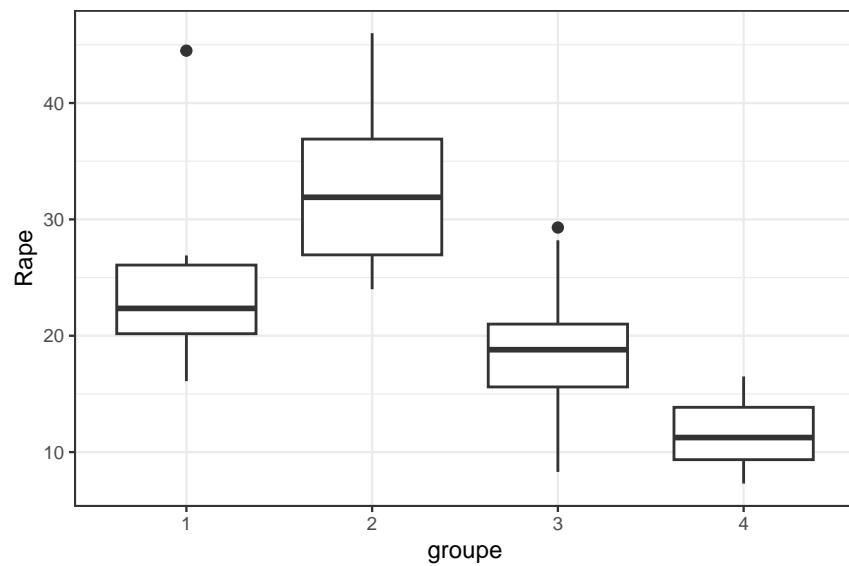
Pourcentage de la population urbaine

```
ggplot(res1)+aes(x=groupe,y=UrbanPop)+geom_boxplot()
```



Arrestations pour viol

```
ggplot(res1)+aes(x=groupe,y=Rape)+geom_boxplot()
```



Interprétation

- *Groupe 1* : états ruraux avec un *fort niveau d'insécurité* (notamment au niveau des meurtres) : Alabama, Alaska, Géorgie...
- *Groupe 2* : états urbains avec un *très fort niveau d'insécurité* (dans tous les domaines) : Texas, Californie, Floride, New York...
- *Groupe 3* : états urbains avec un *niveau d'insécurité modéré* : Arkansas, Connecticut, Virginie...
- *Groupe 4* : états ruraux avec un *faible niveau d'insécurité* : Iowa, Montana, Vermont...

Conclusion

- Deux problèmes appartenant à la partie *Machine Learning* de l'Intelligence Artificielle.
- *Apprentissage supervisé* : expliquer/prédire une variable par d'autres.
- *Apprentissage non supervisé* : regrouper les individus de façon homogène.
- Deux algorithmes automatiques basés sur des distances (penser à réduire les variables) qui répondent à ces problématiques.
⇒ dans les deux cas, ce sont uniquement les *données* et l'*expérience du datascientist* qui permettent de répondre au problème.

Les deux algorithmes

- Supervisé : *k-plus proches voisins*.
 - Calculer les *prévisions* en se basant sur les *observations proches* de x .
 - Choix de k très important ⇒ problème de *sous-apprentissage* et *sur-apprentissage*.
 - Procédure *automatique* pour choisir k en *coupant* les données en deux.
- Non supervisé : *classification ascendante hiérarchique*.
 - Les groupes se forment de *façon itérative* en assemblant les objets les plus proches.
 - Procédure *automatique* pour choisir le *nombre de groupes*.
 - *Interprétation des groupes* à l'aide d'outils de *statistiques descriptives* et de *data-mining*.