****

[Date]

DAGUET Louise

URCA

Cartographie des explorations sectorielles de la blockchain en France (2022).

Analyse des différentes utilisations de la blockchain dans plusieurs secteurs d’activité français en 2022.

Sommaire

[Introduction 3](#_Toc158048017)

[1. Cadre mathématique 4](#_Toc158048018)

[2. Mise en œuvre 6](#_Toc158048019)

[3. Exemples d’utilisation sous R et Python 8](#_Toc158048020)

[3.1. Sous R 8](#_Toc158048021)

[3.2. Sous Python 10](#_Toc158048022)

[4. Comparaison avec les K-Means 15](#_Toc158048023)

[4.1. Les différences entre K-Means et K-Medoids 15](#_Toc158048024)

[4.2. Comparaison sous R 16](#_Toc158048025)

[4.3. Comparaison sous Python 18](#_Toc158048026)

[Conclusion 20](#_Toc158048027)

[Bibliographie et webographie 21](#_Toc158048029)

[Table des figures 22](#_Toc158048030)

[Table des tableaux 23](#_Toc158048031)

# Introduction

Avant de se lancer dans la technique de la méthode des K-medoids, regardons comment cette dernière a été conçue et dans quel objectif. La méthode de K-medoids est une dérivée des K-means. Cette dernière voit le jour dans les années 1950 où l’algorithme fut proposé par Stuart Lloyd pour des fins de modulation par impulsions et codage. La méthode des K-means fut néanmoins dévoilée au public quelques années plus tard dans les années 1980 et connut des méthodes similaires comme la méthode de Lloyd-Forgy puis Hartignan-Wong. L’objectif de cet algorithme est de diviser les points en k groupes, k clusters de façon à minimiser la somme des carrés des distances entre un point et la moyenne des points de son cluster. Néanmoins, les K-means étant certes puissants, un problème de taille réduit son efficacité : les outliers. En effet, ces derniers peuvent facilement attirer la moyenne vers eux biaisant ainsi les résultats obtenus. C’est pour contrer ce problème qu’une solution fut proposée par Kaufman et Rousseeuw en 1987 : les K-medoids également appelés PAM (Partitioning Around Medoid). La différence majeure entre ces 2 algorithmes est que le point central de chaque cluster est un médoide : un point du jeu de donnée. Ainsi l’utilisation de points déjà existant renforce la robustesse de la méthode vis-à-vis des outliers. Elle est donc surtout utilisée en segmentation de clients, en analyse génomique ou encore en reconnaissance de formes.

D’un point de vue technique, la méthode des K-medoids se situe dans la même grande famille d’algorithmes que les K-means : les algorithmes non-supervisés. Ces méthodes sont utilisées lorsque nous disposons seulement des valeurs des variables explicatives et dont le but est d’essayer de découvrir une structure dans les données (par clustering ou segmentation par exemple). D’autres algorithmes sont également utilisés dans la clusterisation non supervisée comme la classification hiérarchique qui n’a pas besoin d’un nombre de cluster initial par exemple.

Ainsi, pour mettre en lumière cette méthode des K-medoids dans le contexte de l’apprentissage automatique, nous allons structurer ce développement avec une partie mathématique suivie d’une mise en œuvre et pour finir d’une comparaison avec des méthodes plus connues comme les K-means.

## Cadre mathématique

Cette méthode se base sur des résultats géométriques. En effet, on souhaite minimiser l’inertie intraclasse ou encore maximiser l’inertie interclasse, ce qui est équivalent. Pour cela, on calcule la somme des distances de chacun des points du cluster avec son médoïde qui n’est autre que le point médian du cluster. On peut pour cela utiliser n’importe quelle distance à choisir selon le contexte comme la distance euclidienne par exemple. On représente cela par la formule suivante donnée pour un seul cluster :

Où est l’inertie du cluster , est le nombre d’individus du cluster et est la distance entre le point et le centre de gravité .

Pour calculer l’inertie intraclasse, on réalise la moyenne des inerties de chaque cluster qu’on ne pondère plus par leur nombre d’individus. On divise plutôt la somme obtenue par le nombre total d’individus de sorte à obtenir la formule suivante :

Où est l’inertie intraclasse, le nombre de clusters et le nombre total d’individus.

L’inertie intraclasse est donc la moyenne des distances moyennes entre chaque point des clusters et leurs médoïdes. L’inertie interclasse représente l’inertie du nuage de points formés par les différents centres de gravité de chaque cluster. Elle est donnée par la formule suivante :

Où est l’inertie interclasse et est le centre de gravité du nuage de points entier.

Le but est donc de minimiser l’inertie intraclasse ou bien de maximiser l’inertie interclasse. Cela est équivalent car l’inertie totale du nuage de points est constante peu importe la partition et est donnée par :

Dans le cadre de la méthode des K-medoids, les centres de gravité sont en fait des points médians de chaque cluster. On les appelle médoïdes. Pour déterminer le médoïde d’un cluster, on cherche à trouver le point du cluster dont la distance avec tous les autres points du cluster est minimale. Mathématiquement, la distance est donnée par la formule suivante :

Où est la distance entre le médoïde du cluster et le point .

On peut maintenant définir la fonction de coût de la méthode des K-medoids pour la distance :

Le but sera de minimiser ce coût ce qui revient à minimiser l’inertie intraclasse.

## Mise en œuvre

Pour mettre en œuvre la méthode des K-Medoids, il existe les algorithmes PAM, CLARA et CLARANS. En pratique, c’est l’algorithme PAM (Partitioning Around Medoids) qui est largement utilisé et c’est donc celui-là que nous présenterons. L’étape préliminaire de celui-ci réside dans le choix de car il n’est pas capable de déterminer le nombre de clusters idéal, il faut donc lui spécifier. Pour trouver une valeur intéressante pour , il est possible d’utiliser la méthode du coude appliqué au graphique de l’évolution de l’inertie interclasse selon le nombre de clusters. Pour obtenir un tel graphique, il est nécessaire de lancer l’algorithme plusieurs fois pour récupérer les résultats. On présente maintenant les étapes de l’algorithme :

1. On initialise la méthode en sélectionnant k points du jeu de données aléatoirement afin de leur donner le rôle de médoïdes.
2. Pour chaque point du jeu de données, on calcule sa distance avec chacun des médoïdes puis on l’affecte au cluster du médoïde dont il est le moins éloigné.
3. On calcule le coût que l’on note .
4. On sélectionne aléatoirement un point qui n’est pas un médoïde puis on échange ce point avec le médoïde de son cluster. On recalcule ensuite le coût que l’on note .
5. Si , alors le point choisi devient le nouveau médoïde de son cluster et l’ancien médoïde devient un simple point.
6. On effectue de nouveau les étapes 2 à 8 jusqu’à ce qu’elles n’entrainent plus aucun changement.

La méthode des K-Medoids a une vitesse d’exécution comparable aux autres techniques de clustering et converge après un nombre fini d’itérations, elle est par ailleurs facile à implémenter. L’avantage de cette méthode est qu’elle permet toujours de réaliser un partitionnement même lorsque celui-ci est difficile à discerner visuellement. Son principal désavantage est que la partition finale est dépendante de l’initialisation. En effet, le choix des premiers médoïdes conditionne le clustering et on peut aboutir à des résultats parfois très différents selon l’initialisation.

**Avantages** :

* Facile à comprendre et simple à mettre en place.
* Rapide et convergent en un nombre fixe d'étapes.
* Moins sensible aux valeurs aberrantes que d'autres méthodes de partitionnement.

**Inconvénients** :

* Le principal inconvénient des algorithmes K-Medoids est leur inaptitude à regrouper des ensembles d'objets de forme non sphérique (de forme arbitraire). Cela est dû au fait qu'ils cherchent à minimiser les distances entre les objets non médians et le médian (le centre du cluster), privilégiant ainsi la compacité plutôt que la connectivité.

Les résultats peuvent varier d'une exécution à l'autre sur le même ensemble de données, car les premiers médoïdes sont choisis de manière aléatoire.

## Exemples d’utilisation sous R et Python

### Sous R

Nous allons ici réaliser un clustering K-Medoids sur la base USArrests. Pour cela, nous allons avoir besoin des deux packages *cluster* et *factoextra*. Une fois la base récupérée puis standardisée avec *scale*, nous pouvons lancer la recherche du nombre de clusters. Pour cela nous pouvons utiliser *fviz\_nbclust(df,pam,method=”silhouette”)* pour la méthode de la silhouette (figure 1) ou *fviz\_nbclust(df,pam,method=”wss”)* pour la méthode du coude (figure 2).

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Figure : Méthode de la silhouette. Cohorte : les 50 États des États-Unis. Lecture : le nombre optimal de clusters est 2.

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

Figure  : Méthode du coude. Cohorte : les 50 États des États-Unis. Lecture : le nombre optimal de clusters est 2.

Nous observons ici que le nombre de clusters recommandé est de deux. Nous pouvons désormais lancer la clusterisation avec *pam(df,2,metric=”manhattan”,stand = FALSE),* notons que nous aurions pu utiliser la distance euclidienne. Nous sommes dans la capacité maintenant de regarder les médoïdes ou les clusters, puis d’afficher le résultat de la clusterisation avec *fviz\_cluster(pam.res,df).*

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

Figure : États des États-Unis répartis dans deux clusters. Cohorte : les 50 États des États-Unis. Lecture : l’État du Maryland est dans le cluster 1 et celui de Montana dans le cluster 2.

Sur la figure 3, nous observons bien la présence des deux clusters autour des États du New Mexico et du Nebraska. À noter que dans notre cas, étant donné que nous avons plus de deux variables explicatives, le graphique est réalisé à partir du graphe de l’ACP pour réduire la dimension, d’où la présence des dimensions et des pourcentages.

### Sous Python

Cette fois, nous allons réaliser un clustering K-medoids sur la base simulée de Python Iris. Avant tout lancement d’algorithme, nous avons pris le soin comme pour l’application sous R de standardiser la base de données. À partir de ces données standardisées, nous avons, comme précédemment, utilisé la méthode du coude (figure 4) et la méthode de la silhouette (figure 6) pour déterminer combien de clusters nous devrions utiliser. Il semblerait que les deux méthodes se contredisent (2 clusters contre 3 clusters). C’est pourquoi nous allons développer l’algorithme des K-Means Medoids avec 2 et 3 clusters et ainsi nous évaluerons quel est définitivement le choix optimal. Mais avant cela nous avons visualisé les individus sur plan pour connaitre les potentiels « candidats » aux médoïdes lorsque l’on a 2 ou 3 clusters (figure 6 et 7). Pour simplifier, nous allons seulement prendre en compte la longueur et la largeur (en centimètres) du sépale des iris.

Une image contenant texte, ligne, capture d’écran, diagramme

Description générée automatiquement

Figure : Méthode du coude pour trouver le nombre de clusters optimal. Lecture : Selon la méthode du coude, le nombre de clusters optimal est 3.

Une image contenant texte, ligne, diagramme, capture d’écran

Description générée automatiquement

Figure : Méthode de la silhouette pour trouver le nombre de clusters optimal. Lecture : Selon la méthode de la silhouette, le nombre de clusters optimal est 2.

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

Figure : Visualisation de la distribution des individus normalisés classifiés en deux clusters.

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

Figure : Visualisation de la distribution des individus normalisés classifiés en trois clusters.

En appliquant les K-Medoids sur notre jeu de données, lorsque l’on choisit d’avoir 3 clusters (figure 9), nous apercevons une chose étonnante : le médoïdes du cluster 3 semblent être isolés de l’ensemble des autres individus de son cluster. Tandis que lorsque l’on veut avoir 2 clusters (figure 8), les médoïdes semblent être beaucoup mieux placés. A partir de cette visualisation graphique, nous pouvons déjà déterminer que 2 clusters semblent être optimaux (mais c’est à confirmer).

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

Figure : Clustering K-Medoids des espèces d’iris normalisées en deux clusters distincts avec leurs médoïdes associés.

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

Figure : Clustering K-Medoids des espèces d’iris normalisées en trois clusters distincts avec leurs médoïdes associés.

Pour définitivement départager entre 2 et 3 clusters, nous avons eu recours à l’indice de Davies-Bouldin qui cherche à mesurer la similarité entre clusters et cet indice est calculé ainsi :

Avec et sont respectivement l’inertie du cluster et l’inertie du cluster . Ainsi, pour chaque cluster , nous cherchons le cluster qui maximise l'indice de Davies-Bouldin. La meilleure partition est donc celle qui minimise la moyenne de cet indice calculé pour chaque cluster. En d'autres termes, nous souhaitons obtenir la plus petite valeur possible de Davies-Bouldin (DB). Dans notre cas, en appliquant l'indice de Davies-Bouldin pour 2 et 3 clusters, les résultats indiquent que 2 clusters constituent le meilleur choix ( pour et pour ).

## Comparaison avec les K-Means

La méthode des K-Medoids est une variante de la méthode des K-Means. Leur principale différence réside dans les centres de gravité : à la place des médoïdes on considère des centroïdes qui sont calculés pour être des points moyens des clusters. Ils ne sont donc pas forcément, et même très rarement, des points réels du jeu de données. Ils sont aussi très sensibles aux valeurs extrêmes à l’inverse des médoïdes, c’est ce qui fait la principale force de cette dernière méthode. La mise en œuvre de la méthode des K-Means passe par l’algorithme de Lloyd qui est très semblable à l’algorithme PAM. Il est constitué des étapes suivantes :

1. On initialise la méthode en sélectionnant k points du jeu de données aléatoirement afin de leur donner le rôle de centroïdes.
2. Pour chaque point du jeu de données, on calcule sa distance avec chacun des centroïdes puis on l’affecte au cluster du centroïde dont il est le moins éloigné.
3. Lorsque tous les points ont été affectés à un cluster, on recalcule les centroïdes en déterminant la moyenne des points de leur cluster.
4. On répète les deuxième et troisième étapes. La méthode converge lorsque la clusterisation n’évolue plus, c’est à dire qu’aucun des points n’est réaffecté à un nouveau cluster lorsqu’on refait l’étape 2.

À l’issue de ces quatre étapes, on obtient un partitionnement du jeu de données. Celui-ci n’est pas forcément optimal et est dépendant des centroïdes choisis initialement. On rencontre pour cela le même problème qu’avec la méthode des K-Medoids. Il existe une alternative pour résoudre ce problème appelée K-Means++ qui permet d’initialiser intelligemment les centroïdes pour diminuer ces risques.

### Les différences entre K-Means et K-Medoids

|  |  |  |
| --- | --- | --- |
| Caractéristiques | K-Means | K-Medoids |
| Complexité |  |  |
| Sensibilité aux valeurs aberrantes | Oui | Non |
| Facilité d’implémentation | Facile | Compliquée |
| Robustesse | Moyenne | Élevée |
| Temps d’exécution | Plus élevé | Moins élevé |

Tableau  : Comparaison des caractéristiques des K-Means et des K-Medoids. Source : « K-Medoid Clustering Algorithm- A Review », Noor Kamal Kaur, Usvir Kaur, Dr.Dheerendra Singh, International Journal of Computer Application and Technology (April 2014)

### Comparaison sous R

Regardons les différences que nous pouvons observer sur notre base USArrests avec deux et quatre clusters.

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

Figure : États des États-Unis répartis dans deux clusters par K-Medoids. Cohorte : Les 50 États des États-Unis. Lecture : L’État du Maryland est dans le cluster 1 et celui du Montana dans le cluster 2.

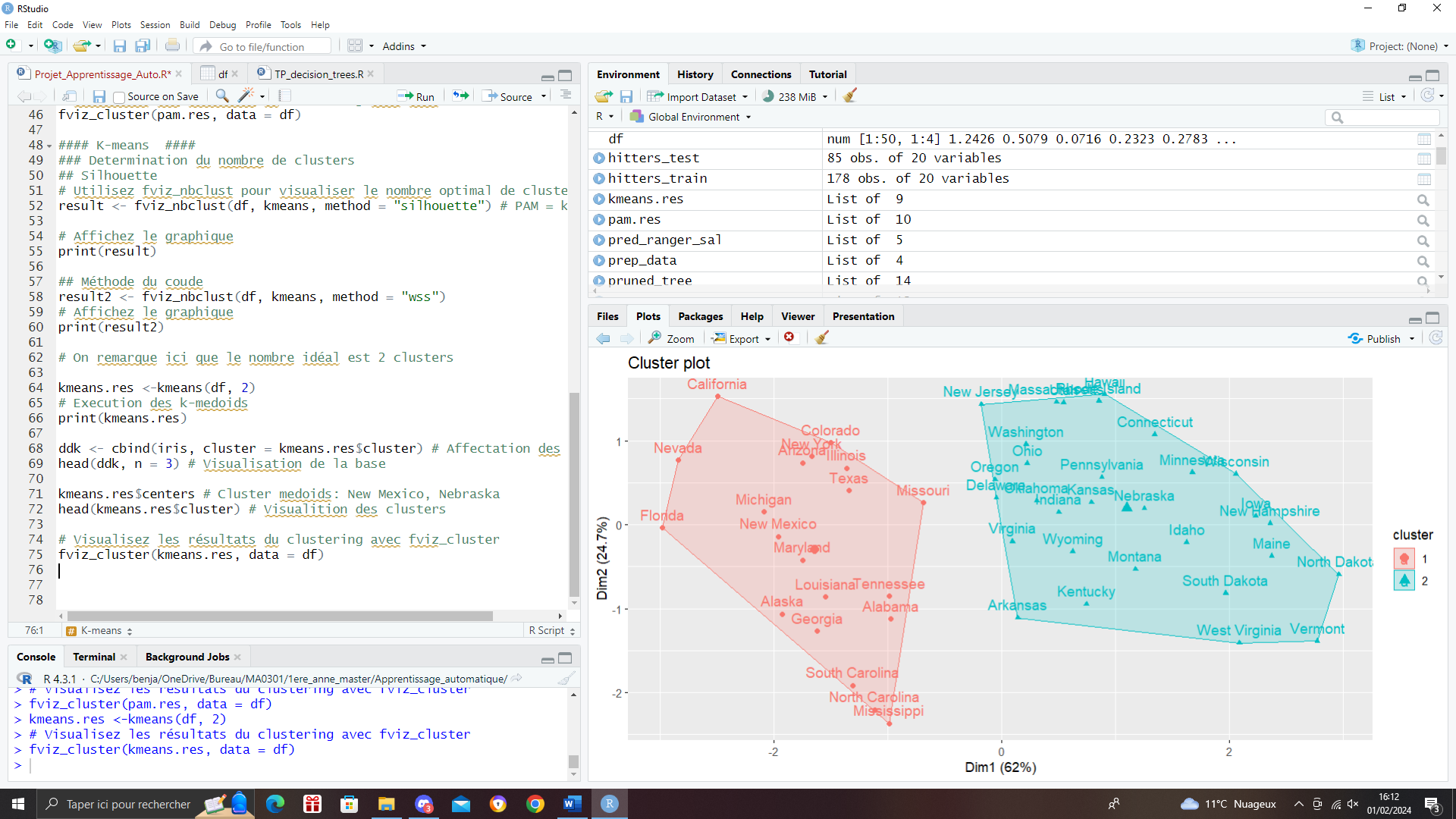


Figure : États des États-Unis répartis dans deux clusters par K-Means. Cohorte : Les 50 États des États-Unis. Lecture : L’État du Maryland est dans le cluster 1 et celui du Montana dans le 2.

Ainsi, nous pouvons observer que dans ce cas les résultats sont très semblables, les centroïdes de la figure 5 occupent des places très proches des médoïdes de la figure 4. Cela explique la ressemblance des formes des clusters.

Une image contenant texte, capture d’écran, logiciel, diagramme

Description générée automatiquement

Figure : États des États-Unis répartis dans quatre clusters par K-Medoids. Cohorte : les 50 États des États-Unis. Lecture : l’État du Maryland est dans le cluster 2 et celui du Montana dans le cluster 3.

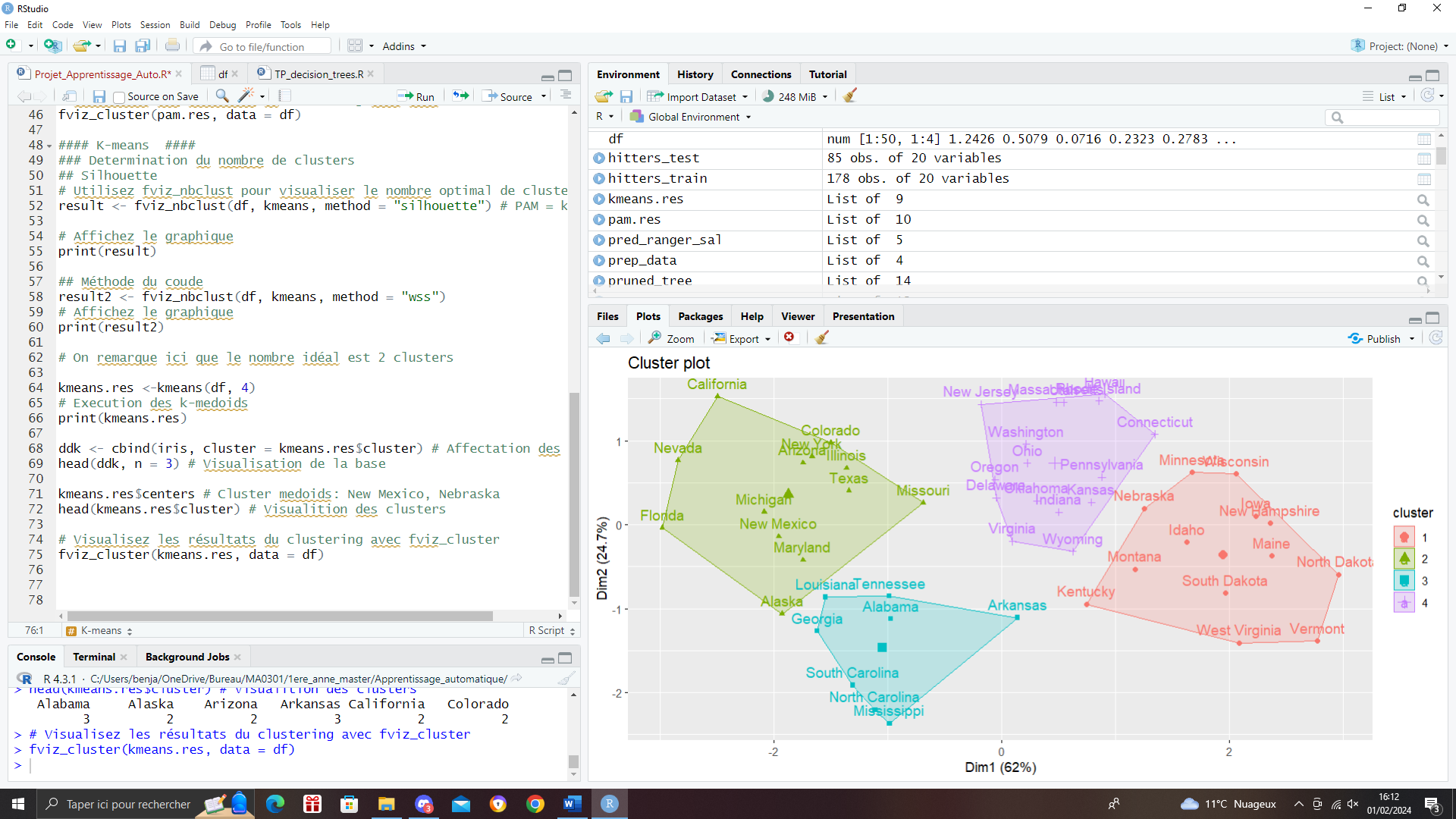


Figure : États des États-Unis répartis dans quatre clusters par K-Means. Cohorte : Les 50 États des États-Unis. Lecture : L’État du Maryland est dans le cluster 2 et celui de Montana dans le cluster 1.

Sur les figures 6 et 7, malgré le fait que nos méthodes de sélection préconisent deux clusters, nous observons les résultats avec quatre clusters pour comparer nos deux méthodes de clustering. Cette fois-ci, la différence est concrète : pour le partitionnement des K-Medoids, les individus sont plus rapprochés entre eux. Pour les K-Means, on constate par exemple que l’Arkansas est très loin de son centroïde ce qui tire ce dernier vers cet État. On observe moins ce genre de phénomène avec les K-Medoids prouvant ainsi une meilleure robustesse de cette méthode car elle n’est pas influencée par les valeurs extrêmes.

### Comparaison sous Python

Maintenant regardons les résultats de l’algorithme des K-Means pour la base Iris. Même si nous avons déterminer que 2 clusters était optimal pour les K-Medoids, nous allons tout de même voir où se trouve les centroïdes pour comparer avec les Médoïdes, de plus l’indice Davies-Bouldin que l’on avait calculé ne concernait seulement les médoïdes et non pas les centroïdes.

Lorsque l’on utilise seulement deux clusters (figure 14), les centroïdes ne sont pas très éloignés du point médoïdes, surtout pour le cluster 1. C’est également le cas lorsque l’on utilise 3 clusters (figure 15), surtout le cluster 3. De plus, avoir 3 clusters pour le K-Means semble tout aussi raisonnable que d’en avoir 2. Par ces comparaisons, si l’on veut utiliser 2 clusters pour le K-Medoids, il n’est pas nécessaire d’avoir autant lorsque l’on veut appliquer les K-Means. Ces deux algorithmes se valent mais selon le contexte et l’objet d’étude on peut privilégier tel ou tel algorithme.

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

Figure : Clustering K-Means des espèces d’iris normalisées en deux clusters distincts avec leurs centroïdes associés.

Une image contenant texte, capture d’écran, diagramme

Description générée automatiquement

Figure : Clustering K-Means des espèces d’iris normalisées en trois clusters distincts avec leurs centroïdes associés.

# Conclusion



Nous avons donc présenté dans ce document la méthode des K-Medoids qui permet de partitionner un jeu de données en k clusters. Ce nombre est au choix de l’utilisateur ce qui peut représenter un atout si on cherche un résultat précis comme un défaut si on n’a aucune idée du nombre de clusters idéal. Par ailleurs, cette méthode est assez aisée à mettre en œuvre et converge en un nombre fini d’itérations, son temps d’exécution est comparable aux autres méthodes du même type. La force de cette méthode réside aussi dans sa robustesse vis-à-vis des données extrêmes car les centres de gravité de chaque cluster sont représentés par des médoïdes qui n’y sont pas sensibles. Ces médoïdes sont en effet calculés pour minimiser l’inertie intragroupe en se basant sur une formule de distance mathématique, ils sont à chaque fois des points réels du jeu de données. C’est d’ailleurs le principal élément de différence avec la méthode des K-Means qui utilise des centroïdes. Les K-Medoids permettent donc de former des partitions satisfaisantes même si l’initialisation des médoïdes peut engendrer des résultats parfois très différents si elle n’est pas réalisée judicieusement.

# Bibliographie et webographie

« Calinski-Harabasz, Davies-Bouldin, Dunn et Silhouette - Complex systems and AI », 27 mars 2020. <https://complex-systems-ai.com/partitionnement-de-donnees/calinski-harabasz-davies-bouldin-dunn-et-silhouette/>.

GeeksforGeeks. « ML | K-Medoids Clustering with Solved Example », 17 mai 2019. <https://www.geeksforgeeks.org/ml-k-medoids-clustering-with-example/>.

ICHI.PRO. « Comprendre les algorithmes de clustering K-means, K-means ++ et K-medoids ». Consulté le 3 février 2024. <https://ichi.pro/fr/comprendre-les-algorithmes-de-clustering-k-means-k-means-et-k-medoids-190888206616522>.

Noor Kamal Kaur, Noor, Kaur Usvir, et Singh Dheerendra. « K-Medoid Clustering Algorithm-A Review ». *International Journal of Computer Application and Technology* 1 (avril 2014). <https://www.academia.edu/8446443/K_Medoid_Clustering_Algorithm_A_Review>.

OpenClassrooms. « Découvrez l’algorithme k-means ». Consulté le 3 février 2024. <https://openclassrooms.com/fr/courses/4525281-realisez-une-analyse-exploratoire-de-donnees/5177935-decouvrez-l-algorithme-k-means>.

OpenClassrooms. « Recherchez une bonne partition ». Consulté le 3 février 2024. <https://openclassrooms.com/fr/courses/4525281-realisez-une-analyse-exploratoire-de-donnees/5254143-recherchez-une-bonne-partition>.

OpenGenus IQ: Computing Expertise & Legacy. « K-Medoids Clustering », 31 décembre 2019. <https://iq.opengenus.org/k-medoids-clustering/>.

# Table des figures

[Figure 1 : Méthode de la silhouette. Cohorte : les 50 États des États-Unis. 8](#_Toc157893143)

[Figure 2 : Méthode du coude. Cohorte : les 50 États des États-Unis. 9](#_Toc157893144)

[Figure 3 : États des États-Unis répartis dans deux clusters. Cohorte : les 50 États des États-Unis. 9](#_Toc157893145)

[Figure 4 : Méthode du coude pour trouver le nombre de cluster optimal. 10](#_Toc157893146)

[Figure 5 : Méthode de la silhouette pour trouver le nombre de cluster optimal. 11](#_Toc157893147)

[Figure 6 : Visualisation de la distribution des individus normalisés classifier en deux clusters. 11](#_Toc157893148)

[Figure 7 : Visualisation de la distribution des individus normalisés classifier trois deux clusters. 12](#_Toc157893149)

[Figure 8 : Clustering K-Medoids des espèces d’iris normalisées en deux clusters distincts avec leurs médoïdes associés. 13](#_Toc157893150)

[Figure 9 : Clustering K-Medoids des espèces d’iris normalisées en trois clusters distincts avec leurs médoïdes associés. 13](#_Toc157893151)

[Figure 10 : États des États-Unis répartis dans deux clusters par K-Medoids. 16](#_Toc157893152)

[Figure 11 : États des États-Unis répartis dans deux clusters par K-Means. 17](#_Toc157893153)

[Figure 12 : États des États-Unis répartis dans quatre clusters par K-Medoids. 17](#_Toc157893154)

[Figure 13 : États des États-Unis répartis dans quatre clusters par K-Means. 18](#_Toc157893155)

[Figure 14 : Clustering K-Means des espèces d’iris normalisées en deux clusters distincts avec leurs centroïdes associés. 19](#_Toc157893156)

[Figure 15 : Clustering K-Means des espèces d’iris normalisées en trois clusters distincts avec leurs centroïdes associés. 19](#_Toc157893157)

# Table des tableaux

[Tableau 1 : Comparaison des caractéristiques des K-Means et des K-Medoids. 16](#_Toc157894352)