

Rendu TP2

Oumaima Talouka, Zineb Slam

9 mai 2017

Dans ce TP2 nous allons travailler sur la méthode de classification automatique qui est une méthode non supervisée ; c'est à dire nous n'avons à priori aucune connaissance sur le nombre ni sur le nom des classes qui peuvent exister. Nous utiliserons en majorité la méthode des centres mobiles dont nous analyserons et critiquerons les résultats

1 Exercice 1 : Visualisation des données

Nous disposons de 3 jeux de données : **Iris**, **Crabs** et un jeu de données **Mutations** de dissimilarités des espèces

1.1 Iris

Les iris est un jeu de données de la librairie *MASS* avec **150 individus** et 4 variables (Sepal.Length, Sepal.Width, Petal.Length et Petal.Width). La variable *Z* de réponse est l'espèce d'appartenance. Le but est d'identifier l'espèce de chaque individu en fonction des 4 variables. Pour commencer nous allons centrer et réduire les données puis effectuer une ACP pour pouvoir réduire la dimension du dataset et passer de 4 dimensions à 2 dimensions. Après avoir fait appel à la fonction *prcomp* nous affichons les données dans le premier plan factoriel sans tenir compte de l'espèce. Nous notons d'abord que les 2 premiers plans factoriels expliquent **95.8%** des données (*inertie expliquée*). Ensuite nous utilisons la fonction *autoplot* pour colorer chaque individu selon son appartenance à une espèce et ce dans le même plan factoriel.

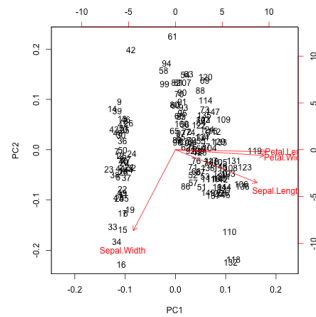


FIGURE 1 – Représentation des Iris dans les 2 premiers plan factoriel

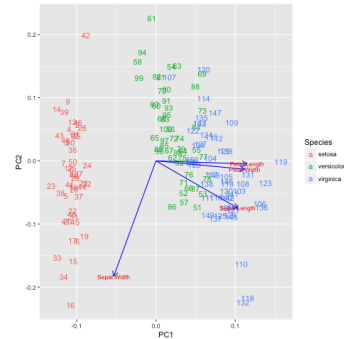


FIGURE 2 – Représentation des Iris dans les 2 premiers plan factoriel en colorant les espèces

Nous observons donc que selon la figure 1 qu'il existe au moins 2 classes bien distinctes. Néanmoins la figure 2 nous permet de constater l'existence de 3 classes : **setosa**, **versicolor** et **virginica**. Dans la première partie nous avons confondu les classes versicolor et virginica qui sont très proches et donc indistinguables à l'œil nu. L'appartenance à l'espèce setosa est majoritairement expliquée par la variable Sepal.Width, tandis que les 2 autres espèces par les 3 autres variables. Enfin Petal.Length, Sepal.Length et Petal.Width sont des variables fortement corrélées. On devrait donc s'attendre à 3 classes pour ce jeu de données

1.2 Crabs

Nous allons procéder de la même manière que dans la question précédente mais cette fois pour les données Crabs2. Le jeu de données Crabs2 compte 200 individus et 4 variables (**FL2**, **RW2**, **CL2**, **BD2**) et deux variables de réponses le sexe et l'espèce. Nous utilisons la fonction *interaction* pour merger le sexe et l'espèce et obtenir une seule variable de réponse Z. **91.2%** des données sont expliquées par les 2 premières composantes principales. Les 2 figures ci-dessous représentent les données Crabs dans les 2 premiers plan factoriel sans distinguer les espèces (à gauche) puis en les colorant (à droite).

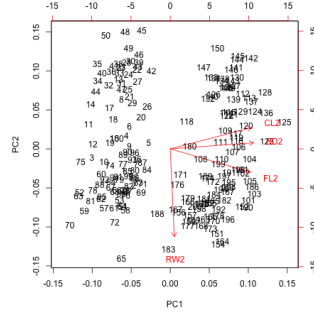


FIGURE 3 – Représentation des Crabs2 dans les 2 premiers plan factoriel

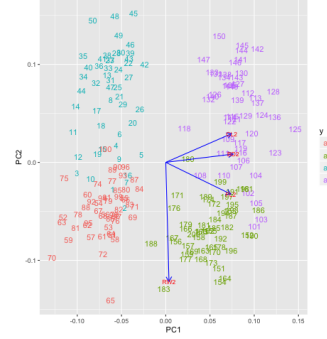


FIGURE 4 – Représentation des Crabs2 dans les 2 premiers plan factoriel en colorant les espèces

Selon la figure 3 on peut observer qu'il existe au moins 2 classes celle de gauche et celle de droite voir 4 classes si on distingue 2 classes dans celle de droite et celle de gauche. Néanmoins on peut remarquer que ces classes ne sont pas très séparées. La figure 4 montre qu'il existe bien 4 classes B.F, O.F, B.M et O.M. (B et O pour distinguer les 2 espèces et M et F pour le sexe). La deuxième composante PC2 qui est concentrée dans la variable RW2 permet de distinguer le sexe, tandis que CL2, FL2, BD2 encodées dans PC1 permettent de distinguer l'espèce.

On pourrait alors mettre l'hypothèse que le fait que le sexe ne soit pas aussi distinguable que l'espèce est du au fait qu'il est expliqué que par une seule variable tandis que l'espèce par 3.

1.3 Mutations

Nous disposons dans cette étude d'une matrice de dissimilarités entre 20 individus (Homme, Singe, Kangourou, Cheval...). Grâce à la méthode **AFTD** nous allons essayer de réduire la dimension du jeu de données. Nous essayerons de trouver au fur et à mesure le bon nombre de variables à choisir.

Nous allons alors commencer par réduire le nombre de dimensions à 2. Pour évaluer cette représentation nous utiliserons le pourcentage d'inertie expliquée cumulée et le Diagramme de Shepard. Notons que dans le Diagramme de Shepard un axe représente la dissimilarité calculée par l'AFTD alors que l'autre les dissimilarités initiales de la matrice 20x20. Ainsi si la dissimilarité calculée par l'AFTD est exacte le Diagramme de Shepard sera une fonction $y=x$.

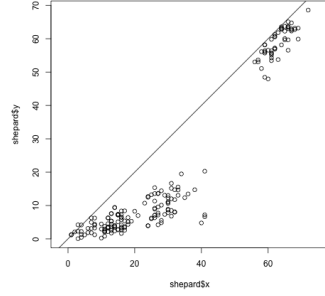


FIGURE 5 – Diagramme de Shepard des Mutations avec 2 variables

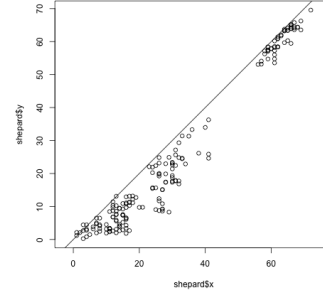


FIGURE 6 – Diagramme de Shepard des Mutations avec 3 variables

On remarque d'après les 2 premières présentations ci dessous que le choix de 2 variables n'est pas très représentatif de nos données. En effet les distances calculées par la représentation en 2 dimensions restent loines des distances initiales. Avec 3 variables les dissimilarités calculées se rapprochent de leur valeur exacte mais on compte toujours des points éloignés Dans la partie suivante nous allons améliorer cette représentation avec 4 puis 5 variables.

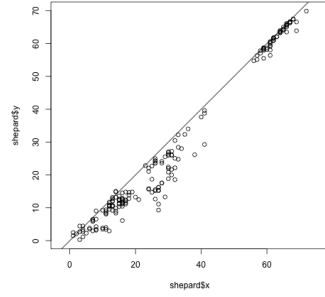


FIGURE 7 – Diagramme de Shepard des Mutations avec 4 variables

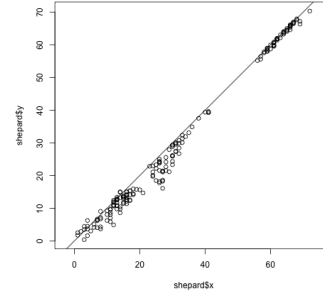


FIGURE 8 – Diagramme de Shepard des Mutations avec 5 variables

Nous remarquons qu'avec 4 variables certaines dissimilarités sont exactes mais d'autres restent fausses. Avec 5 variables on se rapproche plus du cas initial ou toutes les dissimilarités calculées sont plus proche de la droite $y=x$. Ces résultats étaient prévisibles si on calcule le pourcentage d'inertie expliquée cumulé grâce à l'option `eig = TRUE` dans la fonction `cmscale` pour récupérer les valeurs propres.

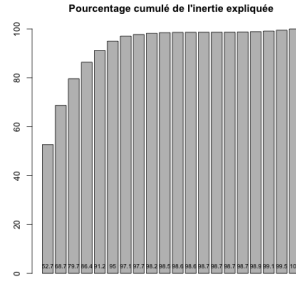


FIGURE 9 – Pourcentage d’inertie cumulee expliquee par AFTD

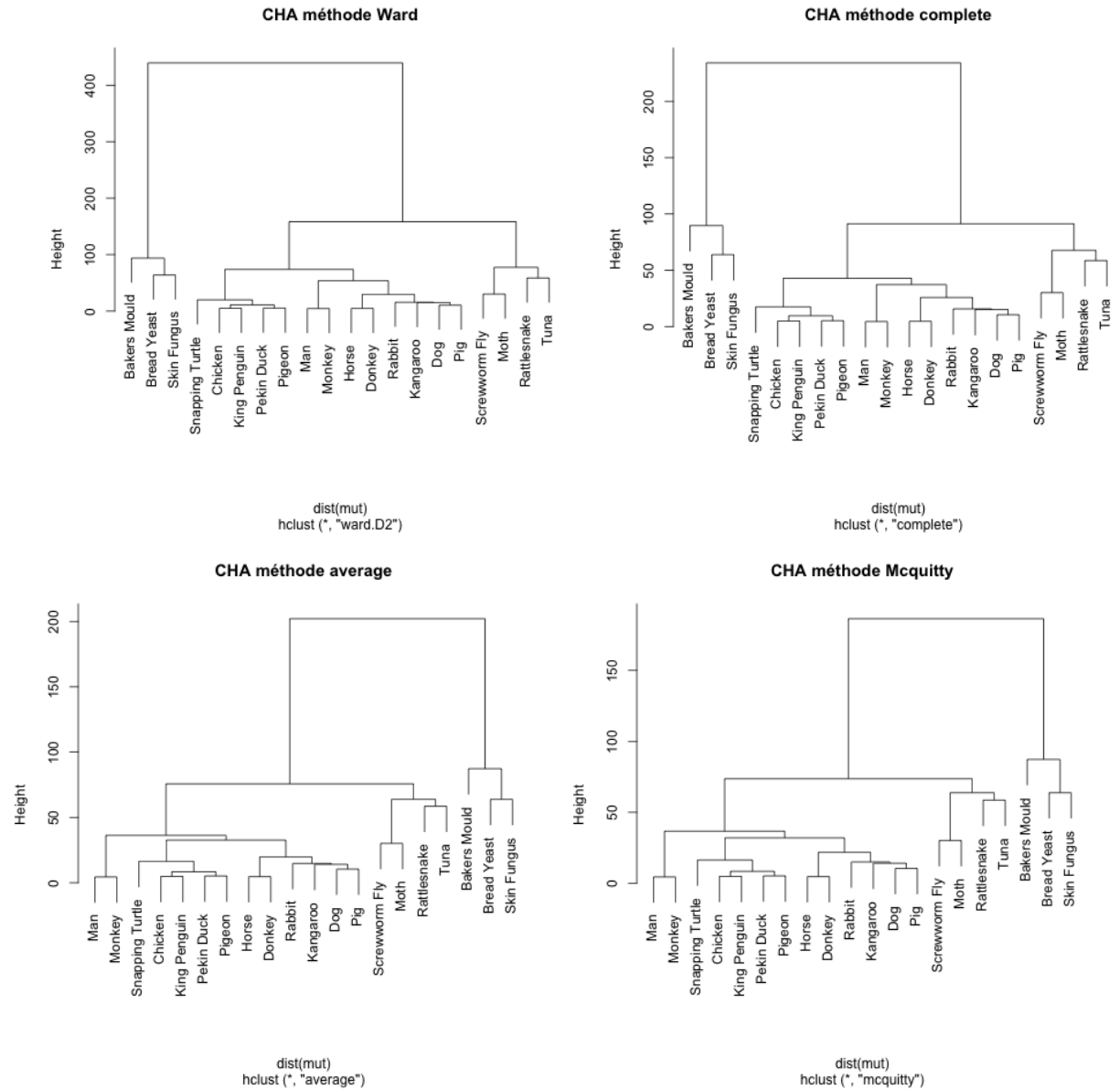
On remarque bien qu’avec 2 variables seulement 68% des données sont expliquées et à partir de 4 le pourcentage est supérieur a 86%.

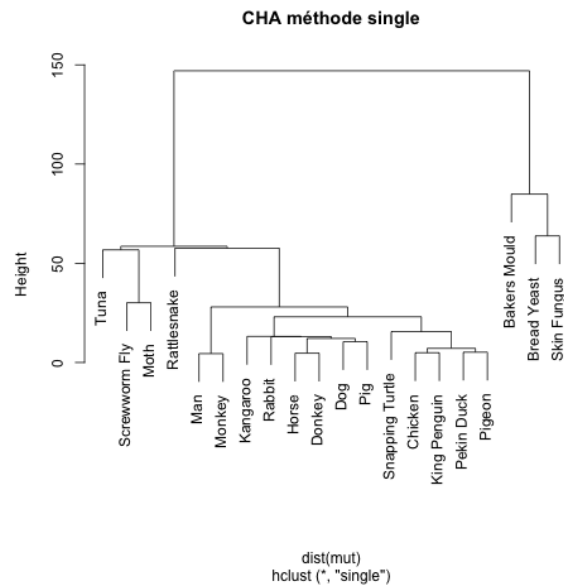
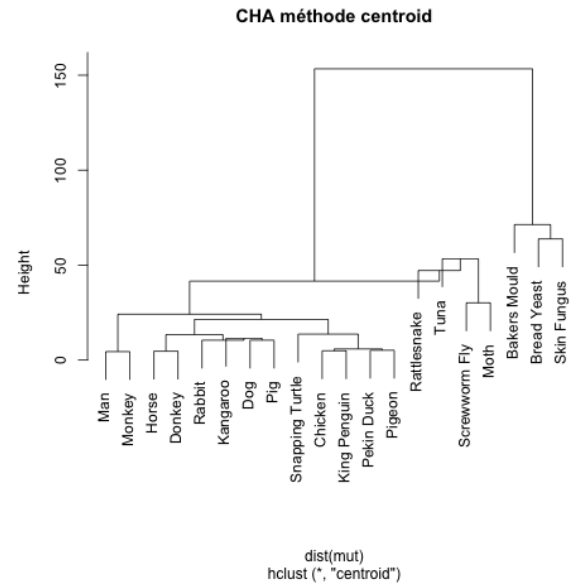
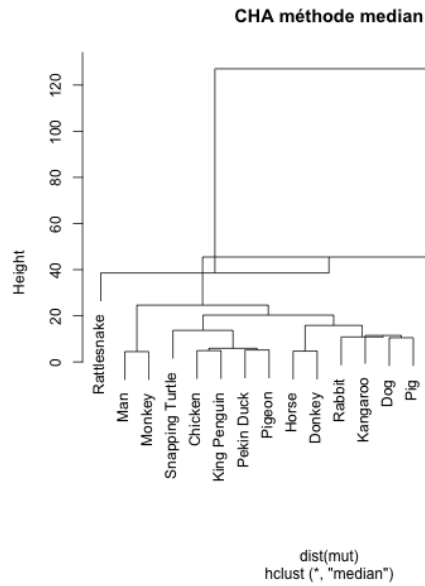
1.3.1 Conclusion

Pour conclure l’AFTD nous a permis de passer d’une représentation de dissimilarités de dimension 20 à une autre de dimension 5 avec un pourcentage d’inertie expliquée de 91%. Nous n’avons pas pu choisir 2 dimensions car la représentation n’est pas fiable. On peut supposer que le fait que cette méthode n’a pas été aussi performante avec ce jeu de données est parce que nos dissimilarités ne sont pas des distances Euclidiennes mais des différences entre les acides aminés dans les chaînes chromosomiques.

2 Exercice 2 : Classification hiérarchique

2.1 Question 1 : CAH Donnees Mutations





Nous remarquons que les méthodes de Ward et complète présentent des CAH identiques. De même la moyenne et Mcquitty d'un autre cote puis la méthode des médianes et des centroides. La methode la plus fiable a choisir est celle de Ward car elle minimise les inerties intra-classes des variables quantitatives.

2.2 Question 2 : CAH Donnees Iris

Nous utiliserons de critere d'aggregation Ward pour effectuer la CAH des donnees Iris. Nous mettons en avant aussi les 3 categories d'especes.

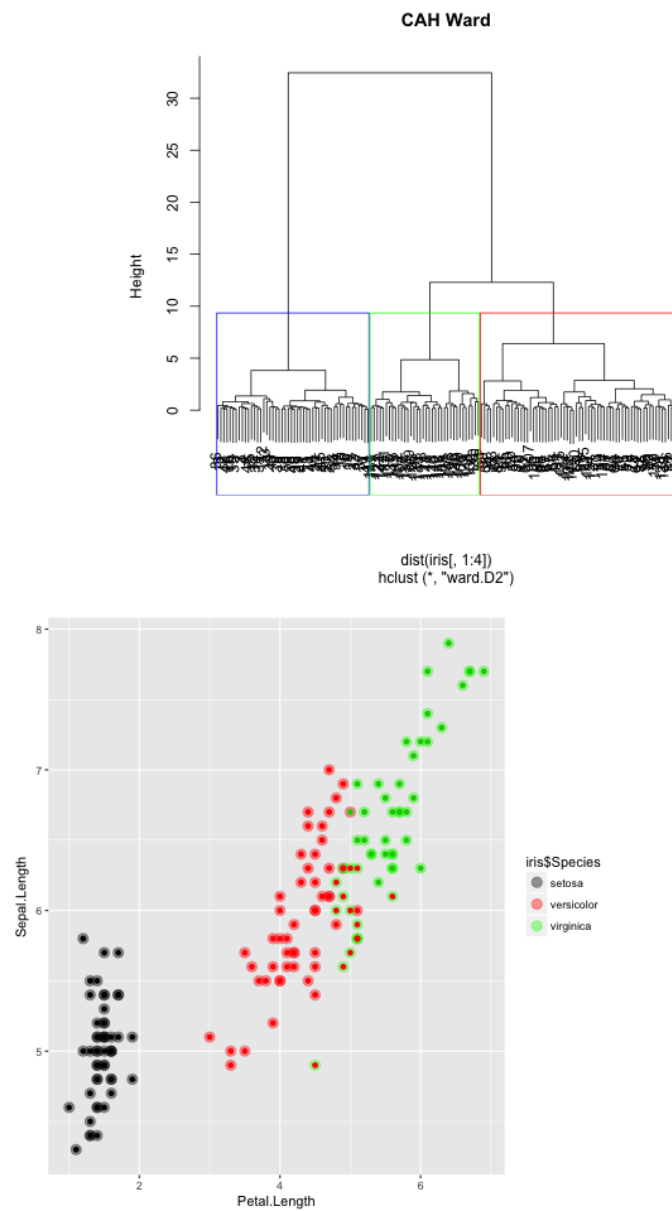


FIGURE 10 – Représentation des données Iris avec la méthode Ward

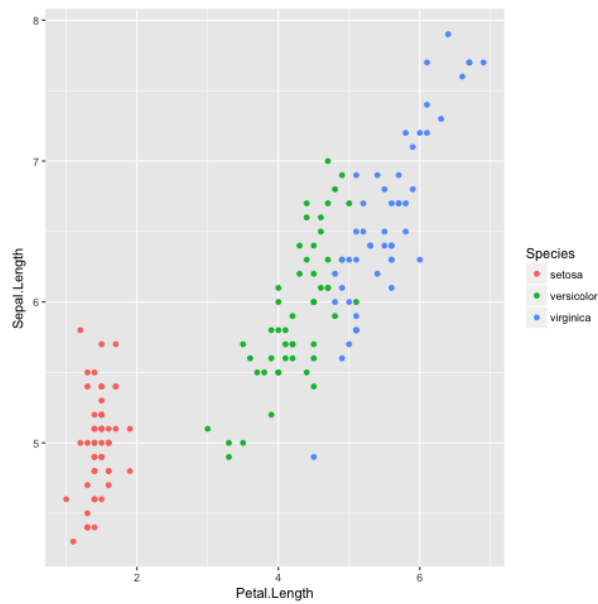


FIGURE 11 – Représentation des données normales Iris
 Nous observons grâce au ggplot le chevauchement des espèces Versicolor et Virginica. Elles sont difficilement différenciables sans couleurs contrairement à l'espèce Setosa.

2.3 Question 3 : CDH Données Iris

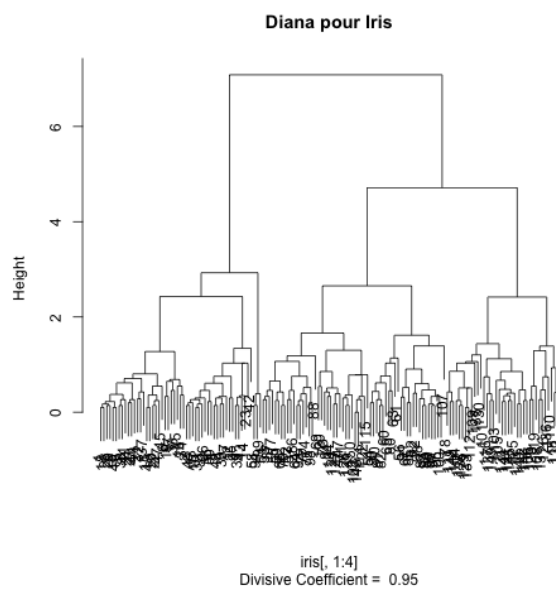


FIGURE 12 – Représentation de la classification descendante données Iris
 Nous n'avons pas observé des différences flagrantes entre la classification hiérarchique ascendante et descendante

3 Exercice 3 : Méthodes des centres mobiles

Le but de cet exercice est de manipuler la méthode des kmeans sur les trois jeux de données Iris2, Crabs2 et mutations2 et analyser la qualité de la méthode et ses limites. On utilisera l'index de Rand ajustée ainsi que les représentations graphiques pour juger des performances des kmeans.

3.1 Iris2

3.1.1 Question 1

Nous commençons par classer nos méthodes en 2, 3 puis 4 classes et les représenter en fonction de 2 variables **Petal.Length** et **Sepal.Width**. Nous avons choisi ces deux variables car comme on a vu dans la première partie ces dernières sont bien expliquées par les 2 premières composantes principales. Les résultats sont représentés ci-dessous :

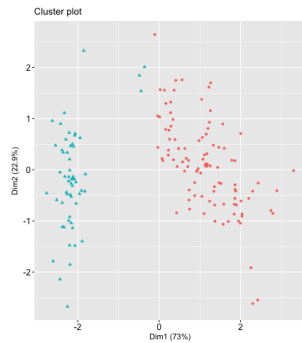


FIGURE 13 – Représentation de Iris2 en 2 classes

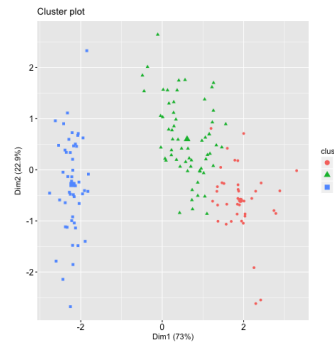


FIGURE 14 – Représentation des Iris2 en 3 classes

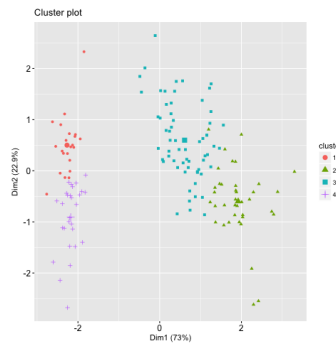


FIGURE 15 – Representation des Iris2 en 4 classes

On remarque qu'avec $k=2$ classes nous obtenons des individus qui appartiennent à la classe 1 alors qu'ils sont plus proche de la classe 2. Tandis qu'avec $k=3$ on a des individus qui appartiennent à une classe mais qui sont très éloignés. La représentation avec $k=4$ classes semble plus correcte de plus avec une connaissance à priori des données nous savons que c'est le nombre de classe correct.

3.1.2 Question 2

A présent nous avons effectué plusieurs opérations de kmeans avec $k=3$, et on remarque qu'à chaque opération nous obtenons des résultats différents. En effet ceci est normal car à la première itération de l'algorithme, celui ci choisit au hasard 3 points ($k=3$) appartenant aux individus et va construire à partir de ces points les 3 classes. Pour toujours obtenir le même résultat on peut faire un appel au préalable à la fonction `set.seed(10)` sur R.

3.1.3 Question 3

Dans cette question nous cherchons à déterminer le nombre de classes minimum dans le jeu de données Iris2. Pour cela on utilise la méthode du coude. Tout d'abord on effectue 100 opérations kmeans pour chaque $k = 2, 3, \dots, 10$ en calculant l'inertie intra-classe. Ensuite on choisit l'inertie minimum pour représenter l'inertie de chaque k , ce qui s'écrit comme $\hat{I}_k = \min_{i=1..100} I_{k_i}$. Pour cela on utilise une matrice 10×100 ou on sauvegarde l'inertie à chaque opération puis on récupère l'inertie intra-classe minimale de chaque ligne. Le code en R détaillé et commenté de cet algorithme est à trouver en annexe. La méthode du coude consiste à choisir le point d'inflexion sur la courbe tel que l'inertie intra-classe ne diminue pas de manière aussi rapide en rajoutant des clusters. Comme il est représenté dans la figure 16; et comme on s'y attendait aussi, le nombre de classes minimum du jeu de données Iris2 est 3.

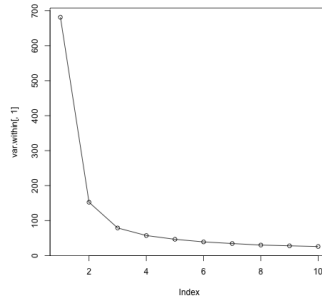


FIGURE 16 – Variation de l'Inertie intra-classe en fonction du nombre de classes

3.1.4 Question 4 : Analyse des résultats pour k= 3

Pour analyser la qualité de classification de la méthode des kmeans nous utilisons l'*index de Rand* ajusté et la représentation graphique. L'index de Rand compte le nb de paires de points qui sont classés de la même manière dans les 2 partitions. Cet indice de concordance est calculé en R grâce à la fonction *adjustedRandIndex*. L'index de Rand est compris entre 0 et 1, plus l'index est proche de 1, plus la classification de la méthode correspond à la classification initiale. La diagonale de la matrice de confusion représente les points qui sont bien classés. On observe que les individus appartenant à l'espèce *Setosa* sont tous bien classés alors que 96% des individus de *Versicolor* ont été mal classés et 28% de *Virginica* sont mal classés aussi. Nous obtenons un **index de Rand= 0.73** ce qui montre que notre classification se rapproche de la vraie classification de la colonne espèces. Pour pouvoir comparer graphiquement les 2 classements nous avons fait une interaction entre la colonne espèce la colonne *iris.kmeans\$cluster* que nous avons représenté grâce à *ggplot*.

Espece	setosa	versicolor	virginica
1	50	0	0
2	0	2	36
3	0	48	14

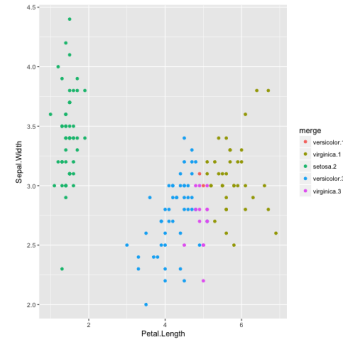


FIGURE 17 – Comparaison de la classification originale et celle des kmeans

La figure 17 rejoint donc bien la remarque faite précédemment comme quoi les espèces *Versicolor* et *Virginica* ne sont pas bien classés. En effet ceci trouve sens si on revient à la figure 2, puisqu'on voit bien que l'espèce *Setosa* ne dépend que d'une seule composante PC1 tandis que les 2 autres espèces sont dépendantes des 2 et ne sont pas linéairement séparable.

3.1.5 Conclusion

Dans cette exercice nous avons utilisé la méthode des kmeans pour classifier notre jeu de données Iris. Nous retenons que cette méthode est efficace quand on a des classes bien distinctes et linéairement séparables mais n'est pas très flexible.

3.1.6 Mutations

Après avoir réalisé l'AFTD pour réduire la dimension de nos jeux de données de 20 à 5, nous utilisons l'algorithme des centres mobiles pour classer les individus en 3 classes. Notons qu'avec 5 variables on explique 91% des données.

Tout d'abord on effectue 1000 itérations en appelant la fonction *kmeans* et on enregistre dans un vecteur l'inertie intra-classe. Ensuite grâce à la fonction *unique* de R on a pu recensé 6 inerties différentes. Ainsi avec 3 classes on peut classer les espèces de 6 manières différentes. Nous avons représentons ci-dessous 3 de ces classifications.

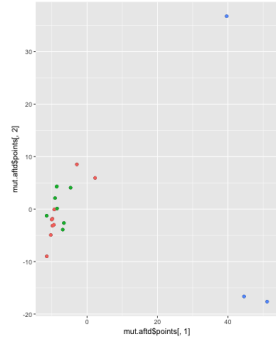


FIGURE 18 – Representation de Mutations2 en 3 classes

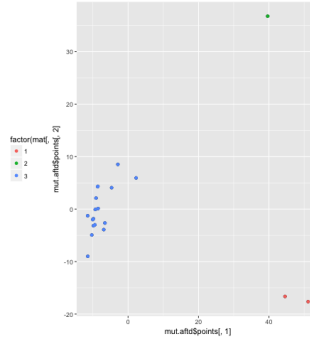


FIGURE 19 – Representation de Mutations2 en 3 classes

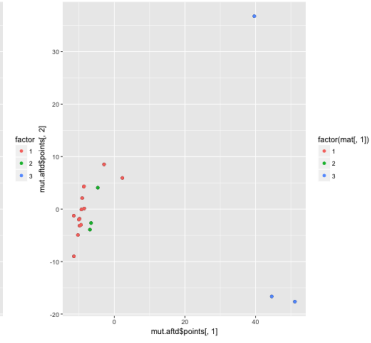


FIGURE 20 – Representation de Mutations2 en 3 classes

Pour obtenir les différentes classifications il nous a fallu tourner plusieurs fois l'algorithme. C'est aussi une des limites du *kmeans* puisqu'on ne peut pas prédire les résultats, de plus ces derniers peuvent changer d'une itération à l'autre.

4 Conclusion

Ce TP nous a permis d'utiliser l'ACP et le AFTD pour réduire les dimensions et classer les données grâce à l'algorithme des centres mobiles. L'AFTD semble bien marcher mais elle est plus intéressante à utiliser quand les distances sont Euclidiennes. La méthode des centres mobiles est une méthode simple et performante mais n'est pas très flexible quand on a à faire à des classes non linéairement séparables. L'un des gros problèmes de cette méthode est qu'elle fournit des résultats différents.