SY09 : Rapport TP2

Classification automatique

CATHELAIN Valentin MARECHAL Anaig

22 juin 2016

Résumé

Ce deuxième TP nous propose d'analyser les données de trois jeux, *Iris*, *Crabs* et *Mutations*. Afin de visualiser correctement les données, nous avons utilisé plusieurs méthodes :

L'ACP , vue au TP précédent

L'AFTD , qui tout comme l'ACP permet de visualiser les données dans un espace multidimensionnel, mais pour sa part à partir d'un tableau de proximités.

La classification hiérarchique , qui peut être ascendante ou descendante et permet d'obtenir une représentation des individus grâce à leurs dissimilarités en les regroupant en classes, ici sous forme d'arbre.

La méthode des centres mobiles ou encore *kmeans*, qui permet de classer les individus en un nombre k de clusters en fonction de leurs proximités.

En utilisant toutes ces méthodes, nous allons réaliser différentes classifications des trois jeux de données dans une démarche polythétique. Nous allons ainsi pouvoir les expérimenter et nous questionner sur leurs différents points faibles et points forts.

1 Exercice 1 : Visualisation des données

1.1 Données Iris

Nous avons tout d'abord affiché les données, sans tenir compte de l'espèce, dans le premier plan factoriel. Nous avons donc réalisé une ACP grâce à la fonction *princomp*, puis nous avons affiché les données dans le premier plan grâce à un *biplot*.
Sur la figure 1, on remarque que **deux groupes** sont distinctement visibles, à droite et à gauche.
Sur la figure 2, nous avons ensuite distingué les différents individus suivant l'espèce .

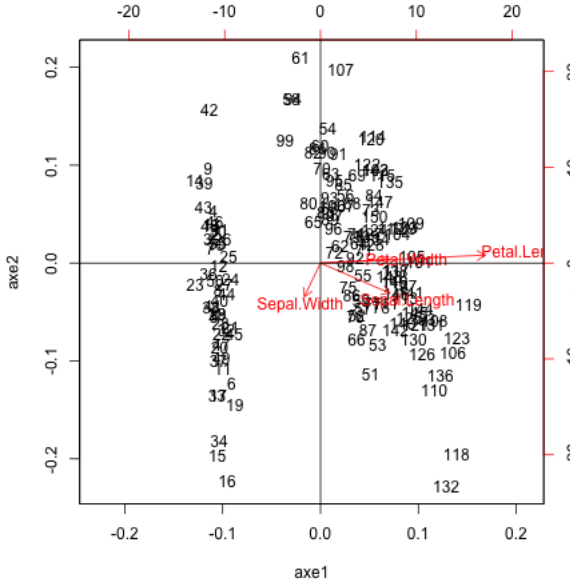


FIGURE 1 – Répartition des données Iris dans le premier plan factoriel

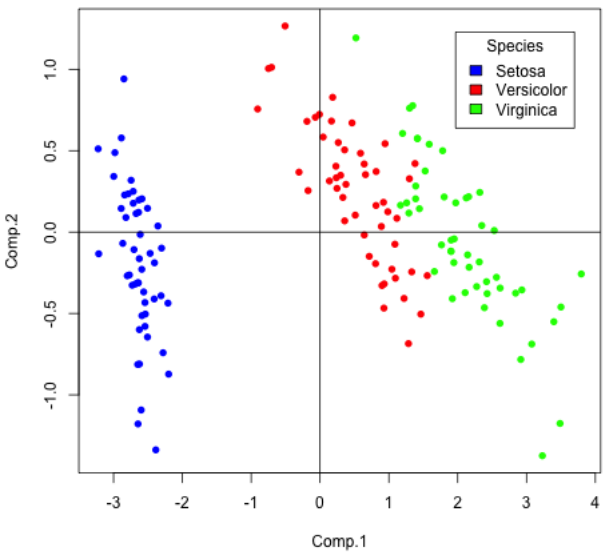


FIGURE 2 – Répartition des données Iris dans le premier plan factoriel selon l'espèce

On remarque alors cette fois sur la figure 2 que le jeu de données comporte en réalité **trois groupes** : un à gauche, que nous avions repéré, et deux à droite, qui paraissaient confondus. Ces groupes sont assez distincts avec des couleurs différentes mais n'étaient pas visibles dans notre premier graphique à cause de la proximité entre les espèces Virginica et Versicolor. Si on ajoute des instances, on peut donc globalement prédire que l'espèce en bleu sera toujours bien séparée des deux autres, mais cette observation est plus discutable pour les groupes vert et rouge. Si l'on recherche une partition des données, on n'aura donc pas de découpage parfait car cela risque d'être un peu arbitraire pour les deux espèces de droite. La meilleure solution serait alors de trouver un bon compromis.

1.2 Données Crabs

De la même façon, nous avons observé les données *Crabs*, tout d'abord sans tenir compte ni du sexe ni de l'espèce, ici représenté sur la figure 3. On remarque de façon claire que **deux groupes** se démarquent, un sur la gauche, l'autre sur la droite.

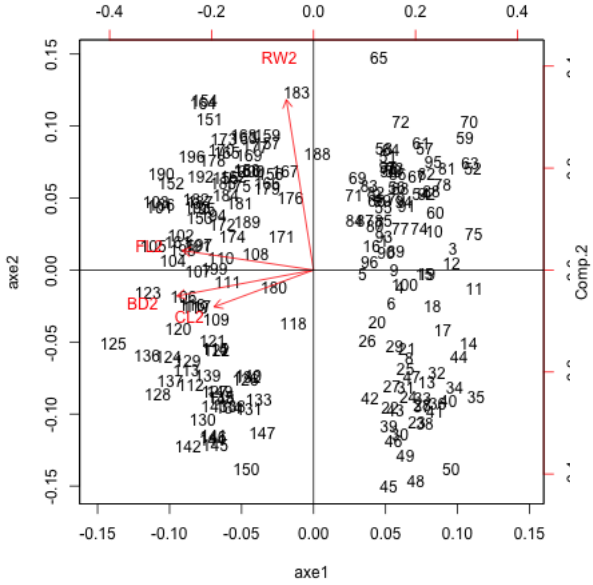


FIGURE 3 – Répartition des données Crabs dans le premier plan factoriel

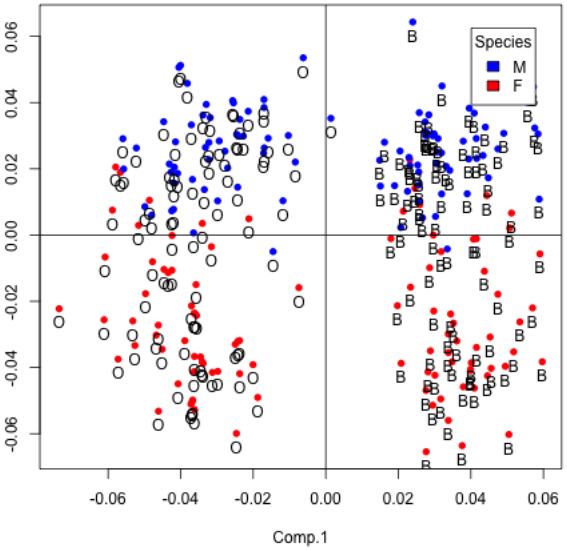


FIGURE 4 – Répartition des données Crabs dans le premier plan factoriel selon l'espèce et le sexe

Nous avons ensuite sur la figure 4 distingué les crabes par espèce (voir texte en coordonnées) et par sexe (voir couleur). On note que les deux groupes visibles sur la première figure correspondaient aux espèces des crabes, le groupe de gauche représentant l'espèce O, le groupe de droite l'espèce B. Cependant, il nous aurait été impossible de distinguer clairement un groupe pour les mâles et les femelles, qui se situent respectivement en haut et en bas mais dont la délimitation est très floue.

1.3 Données Mutations

Enfin, nous avons procédé à la visualisation des données *Mutations*. Pour ce jeu, nous avons utilisé la méthode d'AFTD et non pas d'ACP car on le présente comme un tableau de dissimilarités. On utilise donc la commande R *cmdscale* en précisant deux dimensions, voir figure 5, puis on utilise la fonction *Shepard* pour obtenir les distances nécessaires à l'obtention de diagrammes, ici figure 6. Sur la figure 5 de la représentation de l'AFTD, on observe trois groupes distincts, le moule à gateau d'un côté, la levure à farine et les champignon de la peau d'un autre et le reste des animaux dans le dernier groupe.

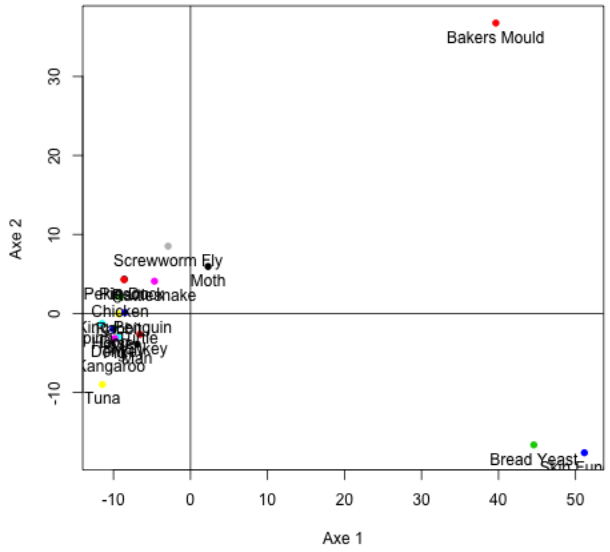


FIGURE 5 – Représentation euclidienne des données en d=2 variables par AFTD

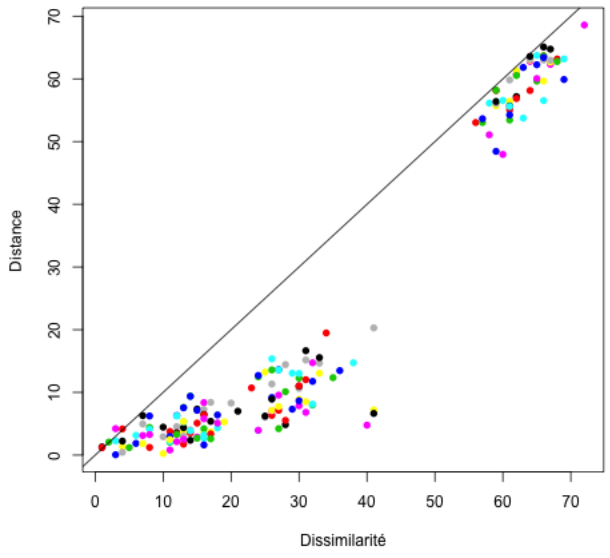


FIGURE 6 – Diagramme de Shepard pour d=2

Le diagramme de Shepard nous permet d'évaluer la qualité de l'AFTD. On remarque ici que les points du diagramme ne se situent pas exactement sur la diagonale. On peut en déduire que la

qualité de la représentation n'est pas optimale. Nous avons donc cherché d'autres représentations avec un nombre de variables plus important, allant de 3 à 5, qui correspondent au nombre de composantes principales.

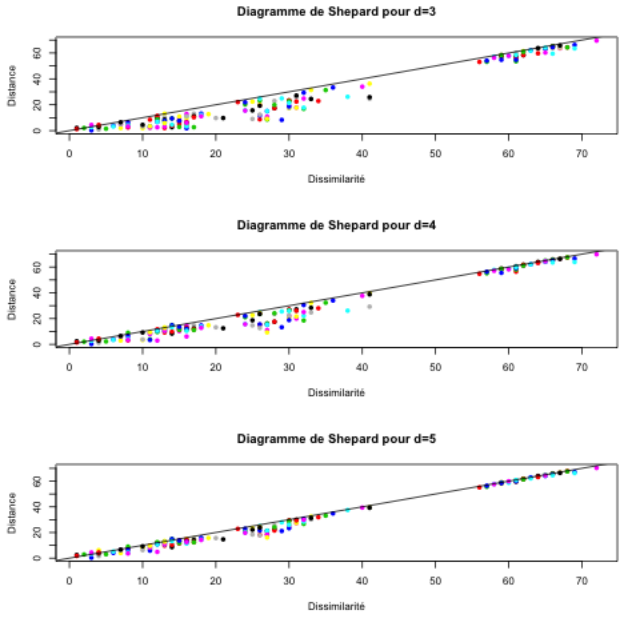


FIGURE 7 – Diagramme de Shepard pour 3, 4 et 5 variables

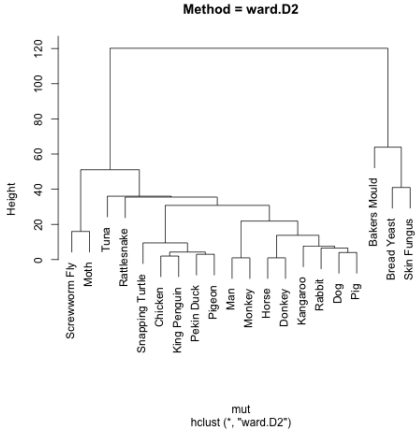
A partir de ces 3 nouveaux diagrammes, on remarque que plus le nombre de variables est important, meilleure est la qualité de l'AFTD.

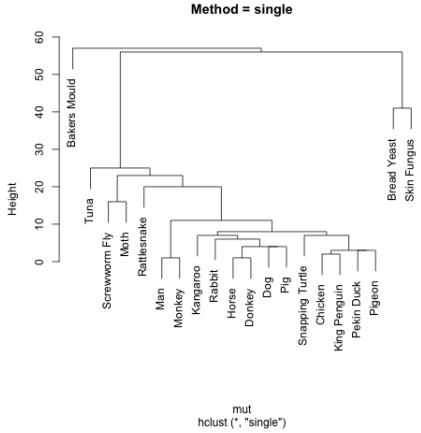
2 Exercice 2 : Classification hiérarchique

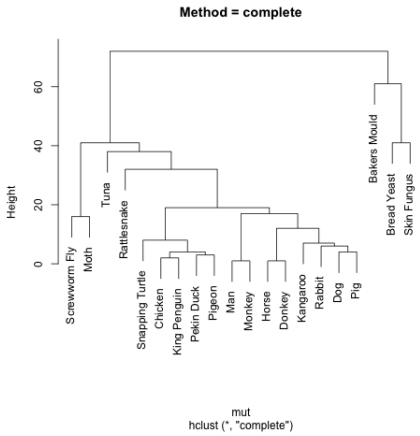
2.1 Données Mutation

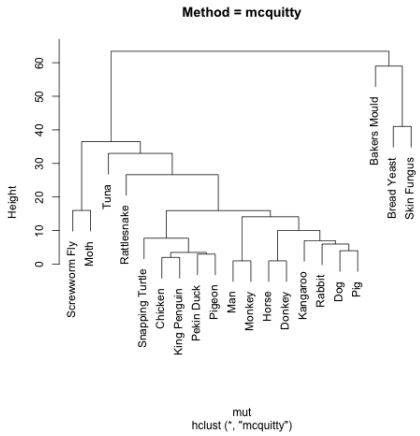
La fonction *hclust* permet d'effectuer une classification hiérarchique, ici dite ascendante car on part de singletons pour remonter au général, l'ensemble complet. Elle nous propose 8 critères d'aggrégations différents qui vont rendre la classification plus ou moins lisible, grâce au paramètre "method" : ward.D2, single, complete, average, mcquitty, median et centroid.

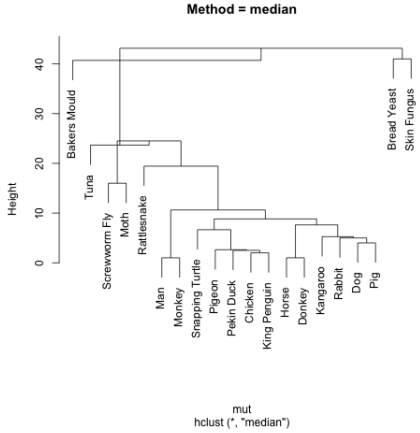
- La méthode *ward* est la plus classique et consiste à réduire au maximum l'inertie intra-classe à chaque agrégation, en utilisant la distance entre les barycentres au carré, pondérés par les effectifs des deux clusters. $D(A,B) = \frac{n_A n_B}{n_A+n_B} d^2(g_A, g_B)$ avec g centre de gravité.
- La méthode *Single Linkage* ou critère du lien minimum utilise la plus petite distance entre deux classes. $D(A,B) = min\{d(i,i'), i \in A \text{ et } i' \in B\}$
- *Complete linkage* ou critère du lien maximum utilise la plus grande distance entre deux classes (à éviter en cas de valeurs aberrantes !). $D(A,B) = max\{d(i,i'), i \in A \text{ et } i' \in B\}$
- Le critère *unweighted pair-group average linkage* ou critère de la distance moyenne utilise la moyenne de la distance entre chaque point des deux clusters. $D(A,B) = \frac{\sum_{i \in A} \sum_{i' \in B} d(i,i')}{n_A n_B}$
- Le critère *median* utilise la médiane de la distance entre chaque point des deux clusters.
- Le critère *centroid* utilise les distance entre les différents barycentres.
- La méthode *McQuitty*, lorsque deux groupes sont réunis, calcule la distance entre le nouveau groupe et un autre groupe comme la moyenne des distances entre cet autre groupe et les groupes à réunir prochainement.

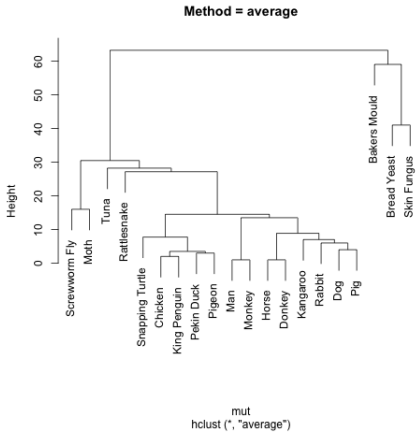


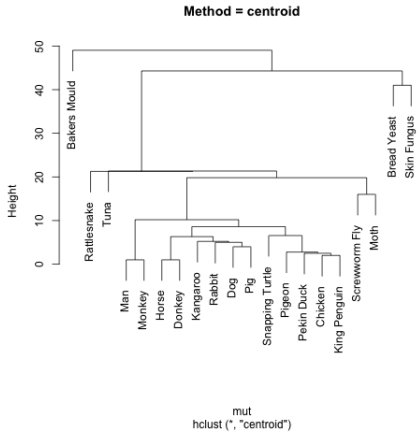



Method = complete
Height
0
20
40
60
Screwworm Fly
Moth
Tuna
Rattlesnake
Snapping Turtle
Chicken
King Penguin
Pekin Duck
Pigeon
Man
Monkey
Horse
Donkey
Kangaroo
Rabbit
Dog
Pig
Bakers Mould
Bread Yeast
Skin Fungus
mut
hclust (*, "complete")


Method = mcquitty
Height
0
10
20
30
40
50
60
Screwworm Fly
Moth
Tuna
Rattlesnake
Snapping Turtle
Chicken
King Penguin
Pekin Duck
Pigeon
Man
Monkey
Horse
Donkey
Kangaroo
Rabbit
Dog
Pig
Bakers Mould
Bread Yeast
Skin Fungus
mut
hclust (*, "mcquitty")


Method = median
Height
0
10
20
30
40
Bakers Mould
Tuna
Screwworm Fly
Moth
Rattlesnake
Man
Monkey
Snapping Turtle
Pigeon
Pekin Duck
Chicken
King Penguin
Horse
Donkey
Kangaroo
Rabbit
Dog
Pig
Bread Yeast
Skin Fungus
mut
hclust (*, "median")


Method = average
Height
0
10
20
30
40
50
60
Screwworm Fly
Moth
Tuna
Rattlesnake
Snapping Turtle
Chicken
King Penguin
Pekin Duck
Pigeon
Man
Monkey
Horse
Donkey
Kangaroo
Rabbit
Dog
Pig
Bakers Mould
Bread Yeast
Skin Fungus
mut
hclust (*, "average")


Method = centroid
Height
0
10
20
30
40
50
Bakers Mould
Rattlesnake
Tuna
Man
Monkey
Horse
Donkey
Kangaroo
Rabbit
Dog
Pig
Snapping Turtle
Pigeon
Pekin Duck
Chicken
King Penguin
Screwworm Fly
Moth
Bread Yeast
Skin Fungus
mut
hclust (*, "centroid")

Les critères *ward*, *complete*, *average* et *McQuitty* présentent une classification relativement similaire, montrant d'un côté le moule à gâteau, la levure à farine et les champignon de la peau qui sont bien séparés des autres espèces. Ces critères présentent donc de bons résultats. Cependant, les méthode *single* et *centroid* nous paraissent ici encore meilleures car le moule à gâteau n'a aucun noeud en commun, et donc aucune propriété commune, avec la levure et le champignon. Or sur la représentation de l'AFTD, le moule a gâteau est en effet isolé à l'opposé de la levure et du champignon. De plus, les espèces qui ressortaient bien du groupe sur le diagramme AFTD comme le thon, la mouche et le papillon de nuit sont ici un peu plus haut situés que les autres espèces (mais ce phénomène est visible sur tous les dendogrammes).
Autre phénomène notable, on observe pour le critère *median* des *inversions*. Cette méthode ne peut donc pas être retenue car elle ne respecte pas la condition selon laquelle le critère d'agrégation doit être une fonction *strictement croissante* (les indices doivent être croissants avec le niveau de la hiérarchie).

2.2 Données Iris

Pour effectuer une classification hiérarchique des données *Iris*, nous avons tout d'abord calculé les distances associées avec la fonction R *dist*. Parmi les différents critères d'agrégation testés pour ces distances, c'est le critère *ward* qui nous a paru le plus pertinent. En effet, on sait que le jeu de données peut se décomposer en *trois classes* qui correspondent aux trois espèces Setosa, Virgnica et Versicolor. Or sur notre dendogramme figure 8, on retrouve bien trois classes facilement distinguables.

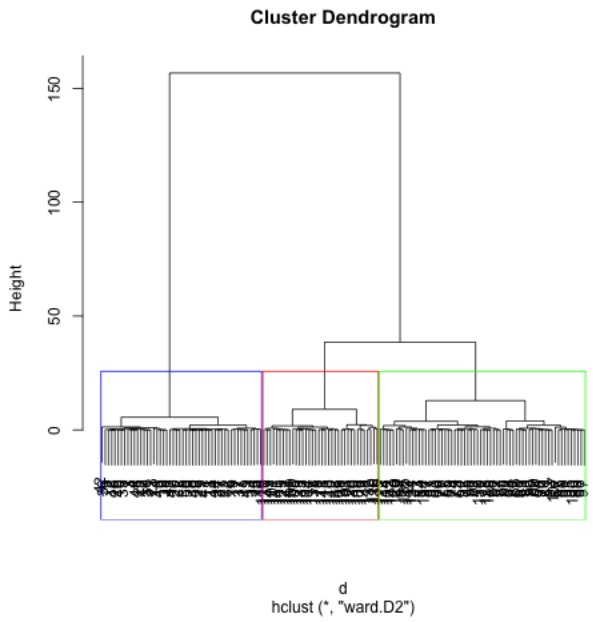


FIGURE 8 – Classification ascendante hiérarchique des données Iris avec le critère ward

Nous avons ensuite effectué la classification hiérarchique descendante de ces données. La différence avec la CAH est que l'on part de l'ensemble général des données que l'on divise jusqu'à obtenir des singletons.

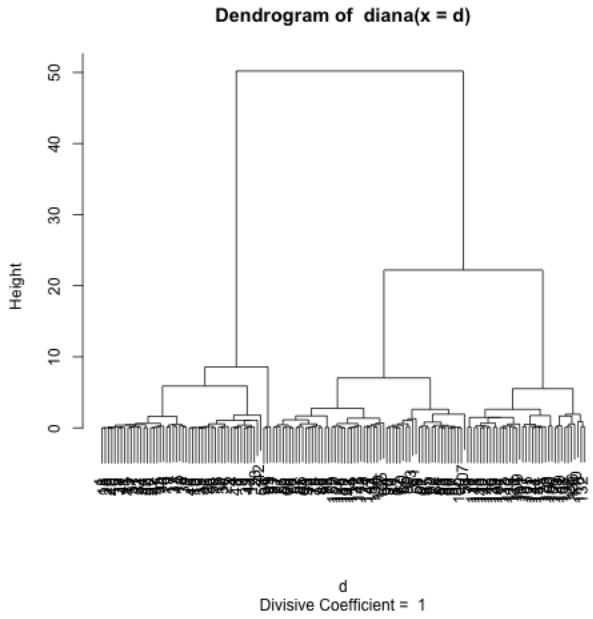


FIGURE 9 – Classification descendante hiérarchique des données Iris

Comme avec la classification ascendante, on obtient ici un résultat correct car on peut bien distinguer trois classes principales parmi les iris.

3 Exercice 3 : Méthode des centres mobiles

Dans cet exercice, nous appliquons la méthode des centres mobiles qui permet de créer plusieurs clusters d'individus en fonction de leur proximité.
Fonctionnement de l'algorithme des centres mobiles pour k=2 : Deux points sont tout d'abord choisis aux hasard dans l'ensemble d'individus et représentent les nouveaux centres. Chacun des autres points est affecté à l'un de ces centres en fonction de leur proximité, ce qui créé deux groupes. Les centres de gravité de ces deux groupes sont calculés et deviennent les nouveaux centres. On affecte une nouvelle fois chaque point au centre le plus proche.On répète finalement ces deux dernières étapes jusqu'à ce que l'algorithme converge.

3.1 Données Iris

On tente tout d'abord une première partition des données pour K=2, K=3 et K=4 grâce à la fonction R *kmeans*.

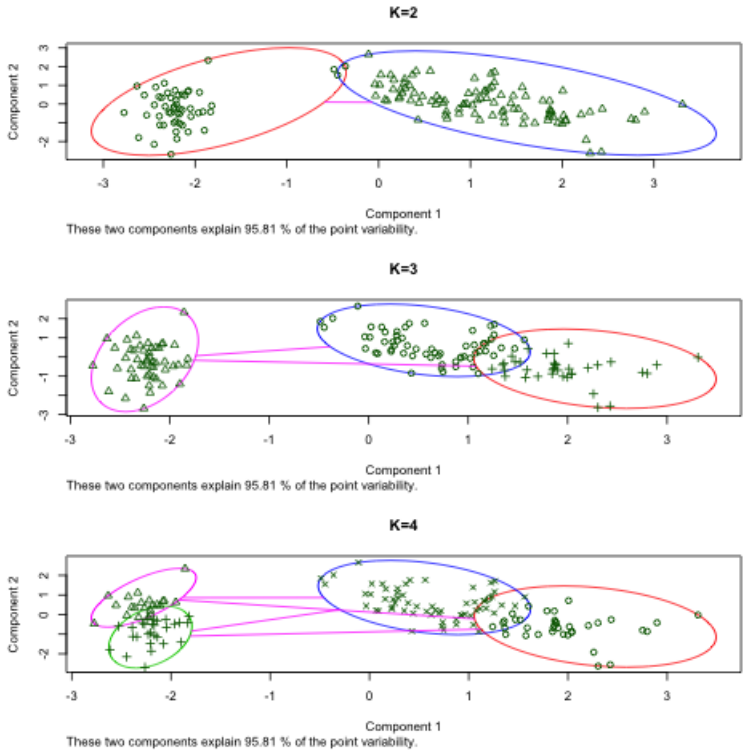


FIGURE 10 – Kmeans des iris pour 2, 3 et 4 classes

Bien que nous savons que les données *Iris* se décomposent en 3 espèces, l'algorithme nous donne ici quand même des résultats pour 2 et 4 classes. Ainsi, pour un nombre de classes égale à 2, la fonction a regroupé les espèces Versicolor et Virginica, qui, comme nous l'avons vu précédemment, sont très proches, et pour un nombre de classes égale à 4, la fonction a décomposé l'espèce Setosa en deux groupes en fonction de ses caractéristiques.

Pour étudier la stabilité du résultat de la partition, nous avons effectué plusieurs classifications des données pour K=3 groupes. On trouve deux types de classification, représentées sur la figure 11 ci-dessous. La première classification a une inertie intra-classe de **79**, tandis que la deuxième a une inertie intra-classe de **143**. Or ici pour améliorer la qualité des résultats, on cherche à minimiser l'inertie intra-classe. En effet, elle représente l'homogénéité des classes et est donc équivalente à maximiser l'inertie inter-classe, c'est-à-dire la séparation entre les différentes classes. Nous retiendrons donc la première classification.

Le fait que deux représentations différentes soient trouvées s'explique par le fait que, comme nous l'avons vu dans l'algorithme, les premiers centres sont tout d'abord sélectionnés au hasard parmi les points. Selon les points sélectionnés au départ, la représentation peut donc changer. Pour éviter cette situation, il est possible d'indiquer à la fonction quels points sélectionner en prenant des points situés vers le centre de gravité de chacun des groupes que l'on souhaite distinguer.

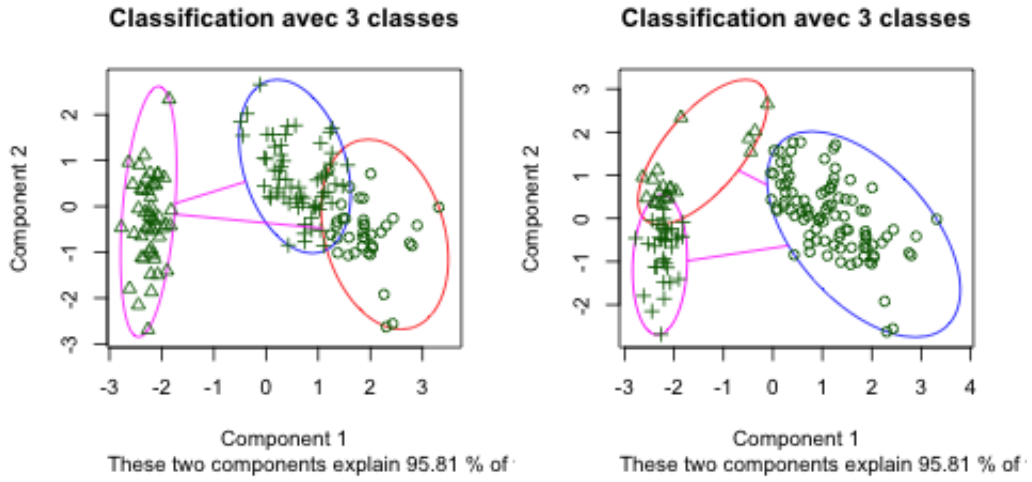


FIGURE 11 – Différents kmeans des iris pour 3 classes

Nous allons essayer de déterminer le nombre de classe optimal par en comparant les valeurs d'inertie intra-classe obtenues pour chaque classe .
Pour chaque classe $k = 3, \ldots, 10$, on lance 100 classifications différentes. On stocke ensuite les valeurs minimales des inerties intra-classe pour chaque k.

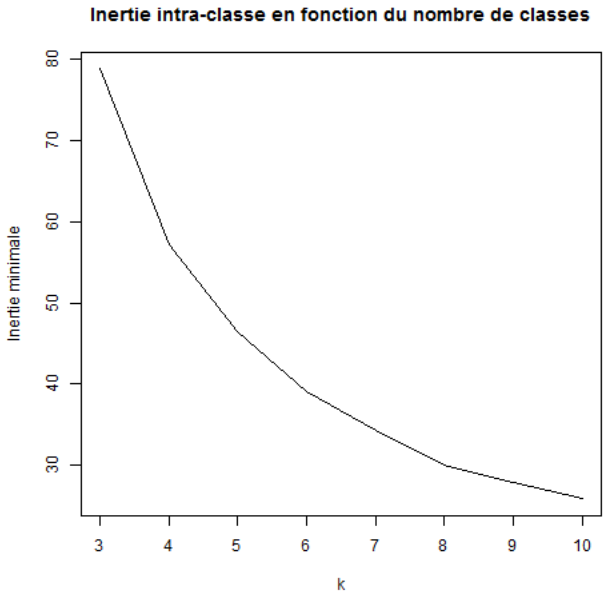


FIGURE 12 – Inerties intra-classes pour $k = 3, \ldots, 10$

On remarque sur la figure 12 que l'inertie décroît de façon exponentielle, mais malheureusement on ne remarque pas de "décrochement" réel, qui nous permettrait de conjecturer sur le nombre de classe optimal.
Nous relançons donc 100 classifications mais cette fois pour $k = 1, \ldots, 10$.

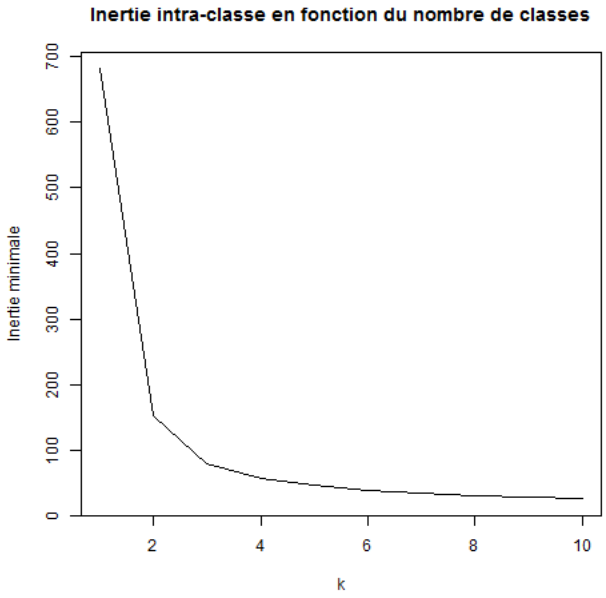


FIGURE 13 – Inerties intra-classes pour $k = 1, \ldots, 10$

On faisant apparaître les inerties de $k = 1, 2$ sur la figure 13, on a cette fois une grosse différence entre les valeurs de l'inertie intra-classe de $k = 2$ et de $k = 3$. Cette méthode du coude (par ailleurs complètement empirique) confirme donc que le nombre de classes idéal est $k = 3$, ce qui correspond à la réalité.

Finalement, on peut comparer la partition obtenue avec les centres mobiles et celle obtenue avec la partition réelle en 3 espèces.

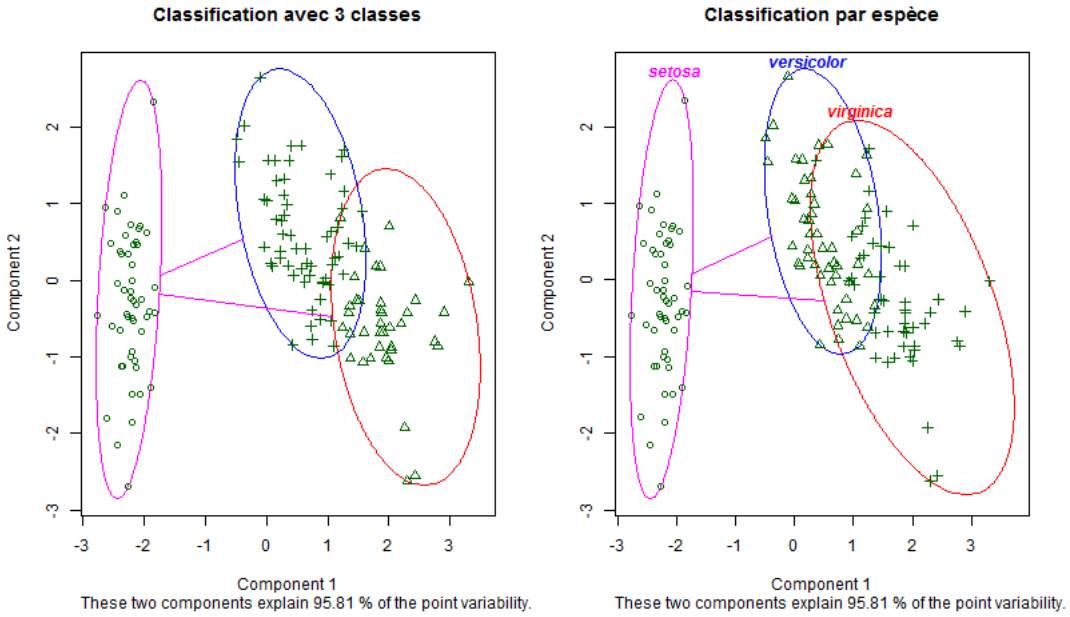


FIGURE 14 – Comparaison des partitions

Tout d'abord, pour la partition réelle (à droite sur la figure 14 , on remarque qu'il y a un groupe bien distinct, l'espèce Setosa, alors que les deux autres espèces de sont pas facilement dissociables. On retrouve cette séparation sur la figure de gauche, entre le cluster rose et les deux autres dans la partition obtenue par les centres mobiles. Cependant dans les deux autres clusters, le fait que les deux classes ne soient pas clairement séparées pose des problèmes pour certains points, qui ont changé de groupe.

3.2 Données Crabs

Lorsque nous lançons une classification avec 2 classes, il y a deux classifications possibles qui ressortent 15. Cela parait logique puisque l'initialisation de la fonction *kmeans* est aléatoire et peut donc donner des résultats différents si on peut distinguer plusieurs classifications.

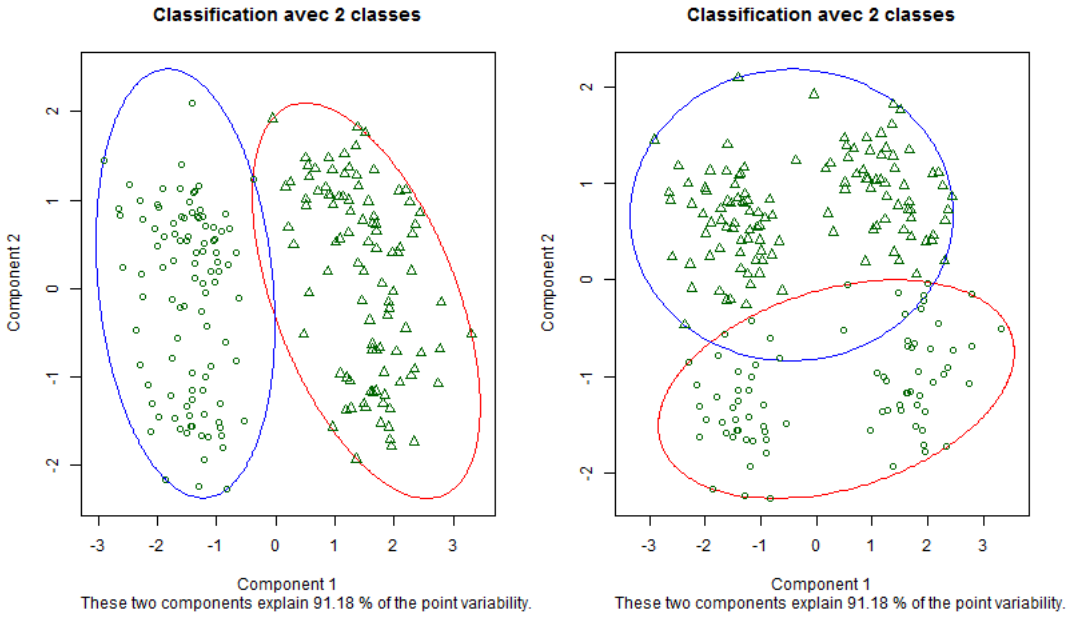


FIGURE 15 – Différents *kmeans* pour $k = 2$

Ces deux types de classification correspondent en fait aux deux distinctions que l'on peut effectuer grâce aux caractéristiques : selon **le sexe** et selon **l'espèce**.

Si l'on effectue une classification avec $k = 4$, on obtient 4 groupes qui sont en fait l'intersection des 4 groupes différents que l'on a pu obtenir précédemment. Mais les groupes ne sont pas tous bien distincts et la fonction *kmeans* peut nous renvoyer la classification 2 (figure 16 à droite). Pour la suite nous forceront *kmeans* à obtenir la classification 1.

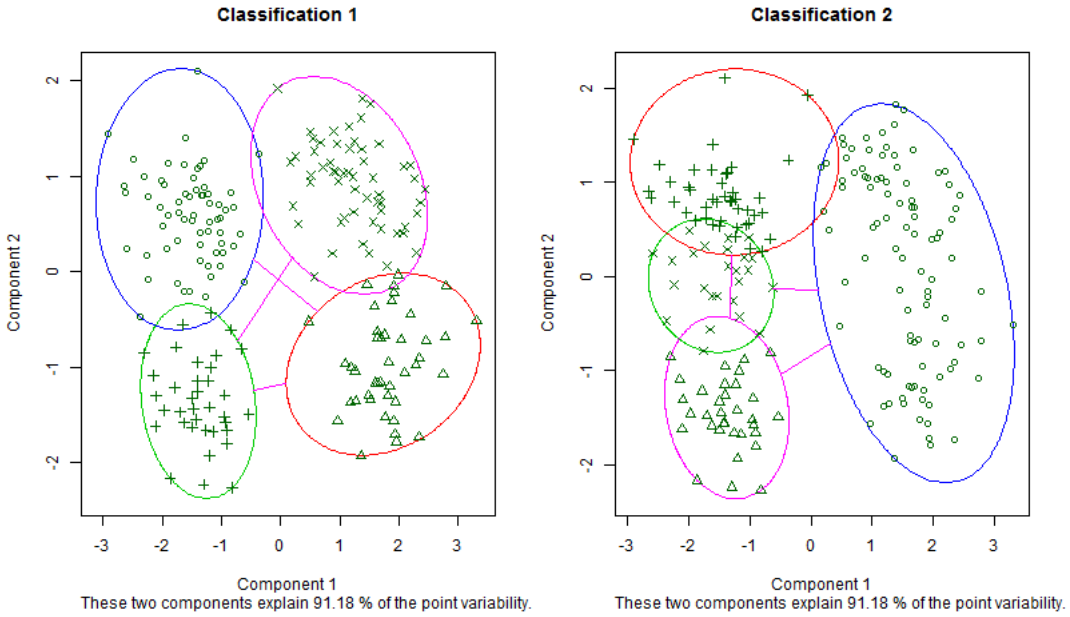


FIGURE 16 – Différents *kmeans* pour $k = 2$

Ils correspondent donc à ces 4 groupes :
— Crabe femelle, espèce O,
— Crabe femelle, espèce B,
— Crabe mâle, espèce O,
— Crabe mâle, espèce B,

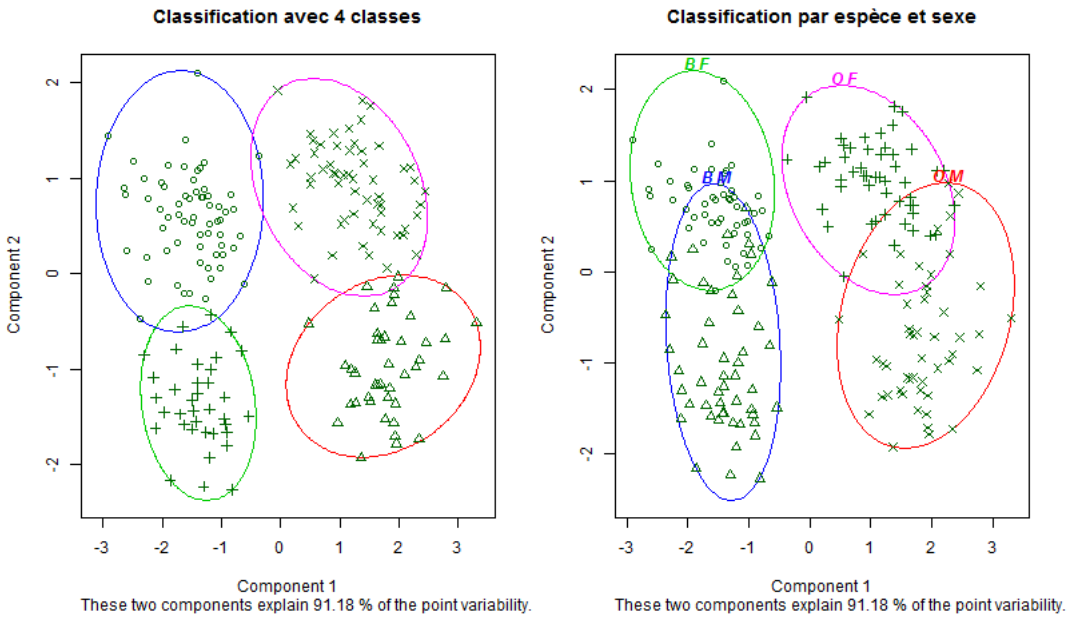


FIGURE 17 – Différents *kmeans* pour $k = 2$

On constate sur la figure 17 quelques différences entre la classification avec les centres mobiles et la réalité. Une partie de crabes mâles appartiennent aux groupes qui correspondant aux femelles. Une fois de plus cette différence s'explique par le fait que dans la réalité, une partie des individus n'est pas clairement identifiable grâce à ces caractéristiques.

3.3 Données Mutations

Les méthode des centres mobiles sur les données mutations donne de nombreux résultats différents. Nous avons testé pour $k = 3$ (figure 18).

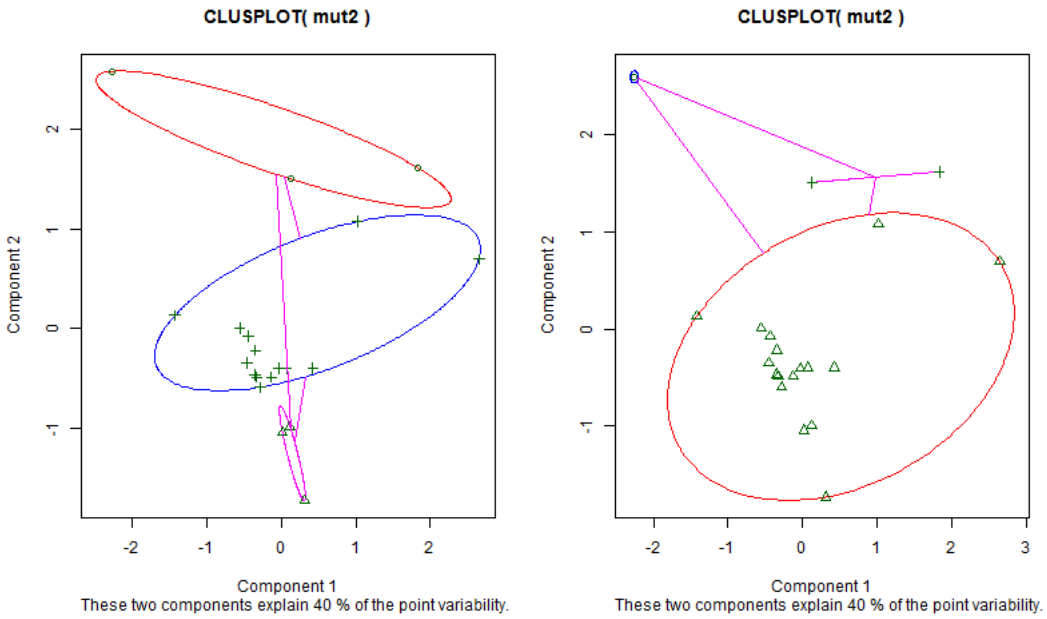


FIGURE 18 – Différents *kmeans* pour $k = 3$

Nous pouvons étudier plus en détail la fréquence à laquelle ces résultats apparaissent. Pour cela, nous lançons la fonction *kmeans* 100 fois, pour $k = 2, \ldots, 5$. Pour en déduire la stabilité, nous étudions la répartition de l'inertie intra-classe pour chaque k. On obtient la répartition suivante :

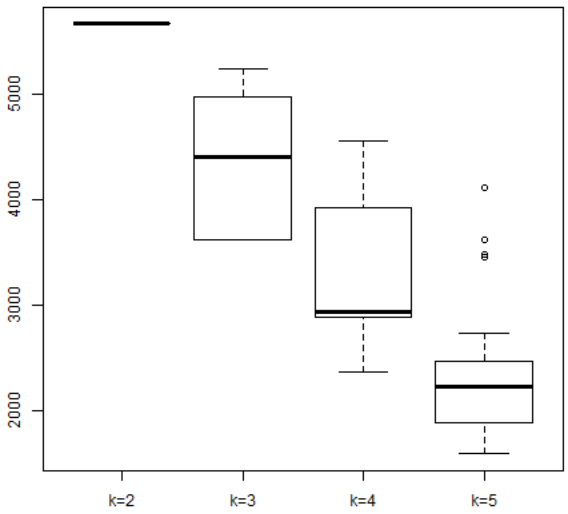


FIGURE 19 – Répartition de l'inertie intra-classe selon k

On remarque que l'écart-type est très important pour $k = 3$ et $k = 4$ (pour $k = 3$ on obtient en fait 6 valeurs différentes) et pour $k = 5$ on observe plusieurs données aberrantes. On se penchant sur les 100 inerties intra-classes obtenues pour $k = 2$ on constate qu'elles ont toutes la même valeur. On peut donc conclure que la seule classification qui soit stable et celle en $k = 2$, bien qu'elle ait une inertie intra-classe élevée.

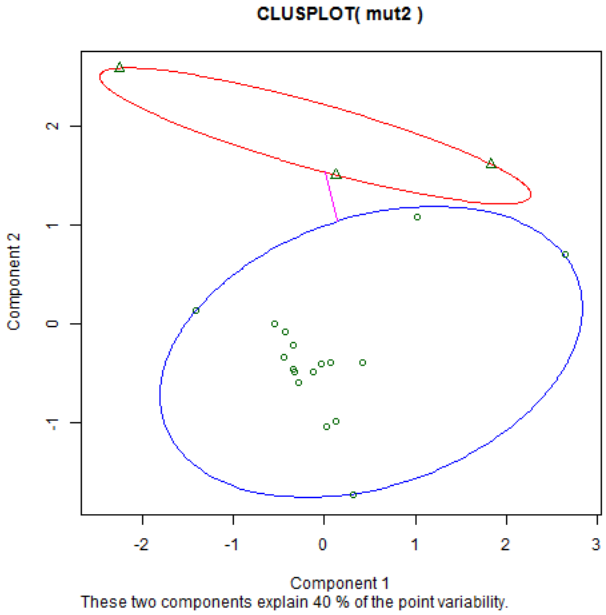


FIGURE 20 – Classification pour $k = 2$

4 Conclusion

La classification automatique des données, tout comme l'ACP étudiée au TP précédent, nous permet une réduction des données et ainsi l'obtention d'une représentation simplifiée de nos données de départ pour les analyser plus aisément. Ainsi, on peut organiser nos données en classes qui nous offrent la possibilité de distinguer les différents individus d'une population.
Nous avons vu que la classification hiérarchique comme la méthode des centres mobiles sont des méthodes extrêmement puissantes qui permettent de regrouper les individus en fonction de leurs caractéristiques. Mais ces méthodes s'avèrent limitées lorsque qu'il est difficile de distinguer les individus. D'où la nécessité de visualiser dans un premier temps les données grâce à la représentation simplifiée.