**SY09 Printemps 2009 – UTC**

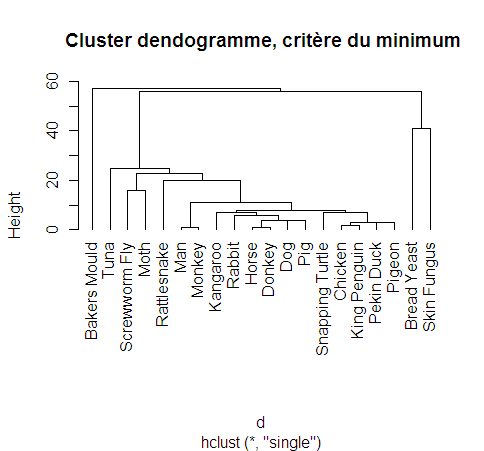
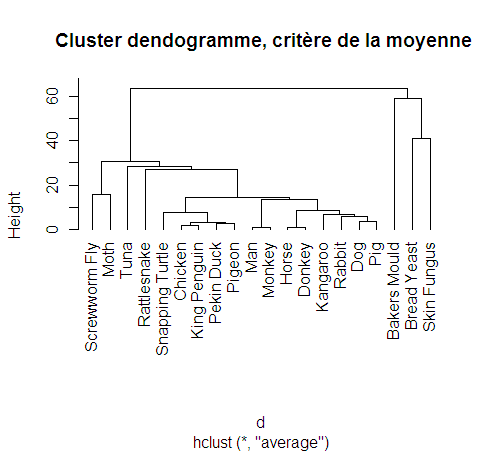
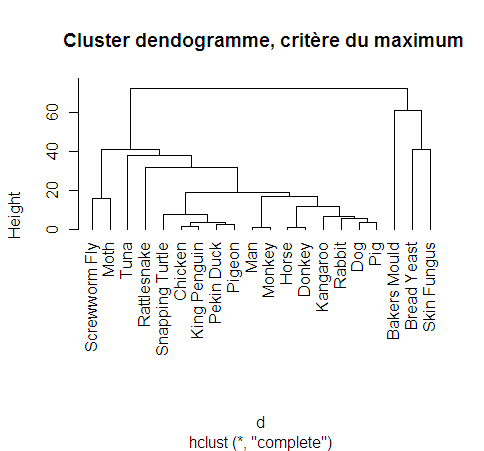
**Classification automatique**

Le but de ce TP va être de d’utiliser la méthode de classification automatique pour traiter plusieurs jeux de données, de manière à comparer ces résultats à ceux obtenus avec des précédentes méthodes.

Exercice I : Classification hiérarchique

1. *Classification hiérarchique ascendante*

Elle est réalisée grâce à la fonction hclust sur R. Les différents critères d’agrégation disponibles sont « average » (= moyenne, ce critère consiste à calculer la distance moyenne entre tous les éléments d’une classe et tous les éléments de l’autre classe), « single » (= minimum, la distance entre 2 classes c1 et c2 est définie par la plus petite distance séparant un individu de c1 et un individu de c2), « complete » (= maximum, la distance entre 2 classes c1 et c2 est définie par la plus grande distance séparant un individu de c1 et un individu de c2), et enfin « ward » qui correspond au critère de Ward. Cependant, le jeu de données  « mutation » est un tableau de distance, nous ne pouvons donc pas appliquer le critère de Ward, puisqu’il nécessite un espace euclidien.

Nous allons tester les différents critères grâce à la représentation en dendogramme. Sur ces graphiques, nous observons la composante height, qui correspond à un vecteur des distances entre les classes aux différents stades de la classification.

La difficulté du choix du critère d'agrégation réside dans le fait que ces critères peuvent déboucher sur des résultats différents. Nous comparons ces résultats de manière à s'assurer que les variations restent faibles.

Dans tous les cas, deux groupes se distinguent ici : Bakers mould, bread yeast, skin fungus d’un côté ; et d’un autre côté, les autres.

Puis à partir de la classe constituée du plus grand nombre d’éléments, les représentations commencent à diverger selon la méthode utilisée, même si la répartition en classes finales reste assez similaire.

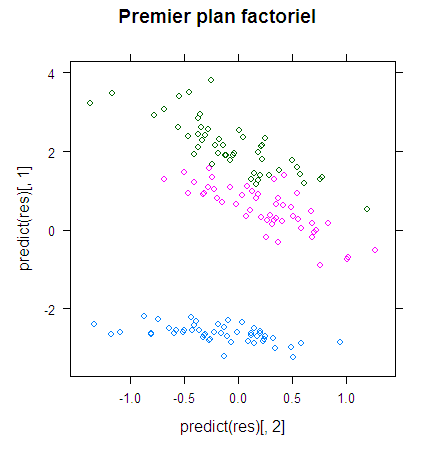
Ainsi, nous pouvons difficilement conclure quant à une représentation idéale pour ce jeu de donnée. Nous pouvons cependant exclure la méthode du critère d’agrégation minimum, puisque nous observons que, dès le début, « Backers Mould » est séparé de « Bread Yeast » et de « Skin Fungus ».

1. *Classification hiérarchique des données Iris*

Nous nous intéressons désormais à la comparaison entre la partition obtenue par classification hiérarchique ascendante, et la représentation plane obtenue par l’analyse en composante principale.

Les analyses factorielles et les techniques de classification sont complémentaires, et on est le plus souvent amené à les utiliser conjointement pour analyser un problème donné. En reliant un module de l’A.C.P à un module de C.A.H, les variables utilisées par le module de C.A.H. seraient les coordonnées des individus sur les premiers axes factoriels.

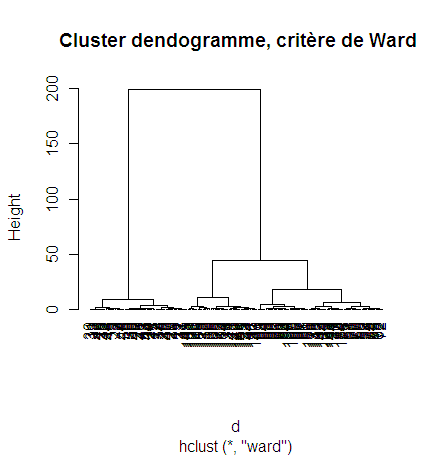
* R**eprésentation dans le premier plan factoriel**



A gauche figure donc la présentation dans le premier plan factoriel des donnes Iris.

Nous observons bien la séparation en **3 classes** (suivant les différentes couleurs : Setosa, Versicolor, Virginica), comme explicitement mentionné dans le tableau de données.

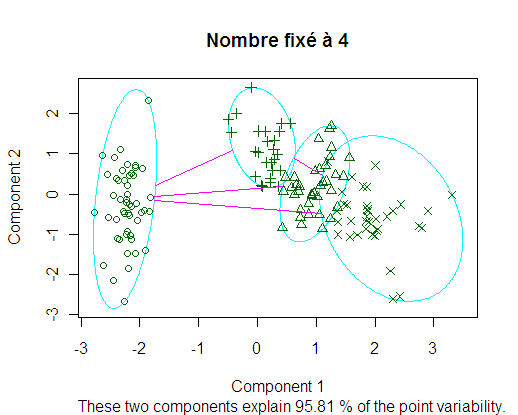
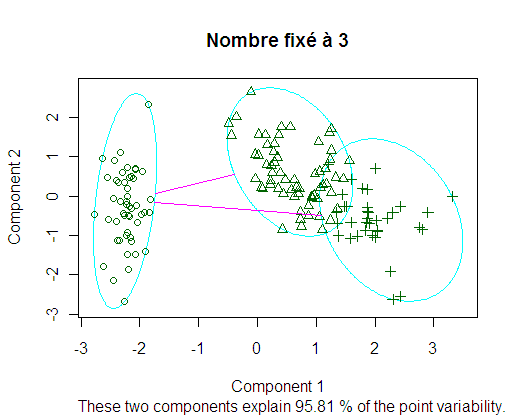
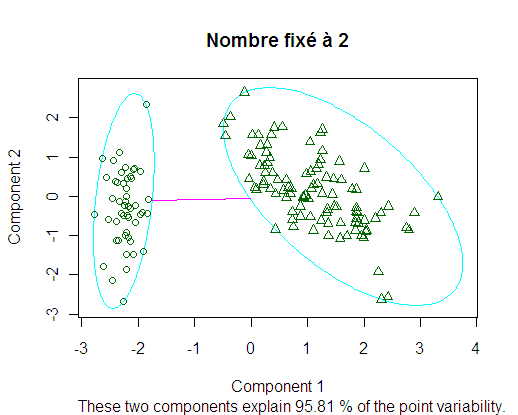
* **Classification hiérarchique ascendante**



Puisque le jeu de données Iris constitue un espace euclidien, nous décidons d’utiliser le critère d’agrégation de **Ward** (en effet, un rapide coup d’œil sur la représentation des autres critères d’agrégation utilisés précédemment ne permet pas d’obtenir une meilleure classification et une aussi claire représentation).

A la vu de cette représentation, nous aurions tendance à conclure en une classification débouchant sur **4 classes**.

Puis, en utilisant la fonction cutree, nous pouvons visualiser la composition des partitions en choisissant leur nombre. Cela permet en effet d’imposer un **niveau de coupe** dans les dendogrammes. La fonction clusplot nous permet ensuite de représenter une partition sur une projection (c’est-à-dire : cut <- cutree(h,3) ; clusplot(donnees$num,cut)).

Pour un nombre de partition fixé à 2 : 2 groupes sont formés distinctement.

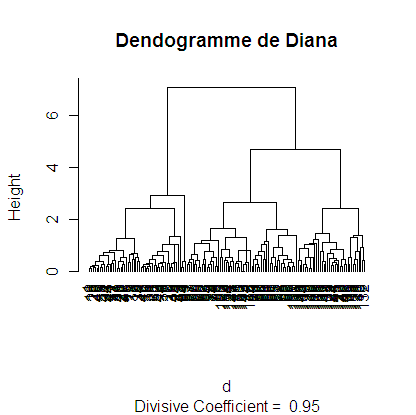
Pour un nombre de partition fixé à 3 : **3 groupes** sont également bien formés et bien **distincts**.

Cependant, pour un nombre de partition fixé à 4, la distinction entre les groupes et beaucoup moins significatives que leur des deux précédentes représentations.

Le dendogramme permettrait alors une classification en 3 groupes distincts, se qui correspondrait alors aux 3 classes (setosa, versicolor, virginica), également visualisées dans la représentation dans le premier plan factoriel.

En gardant un nombre de partition fixé à 3, la partition obtenue par classification hiérarchique ascendante, et la représentation plane obtenue par l’analyse en composante principale, convergent vers le même résultat : soit une classification en **3 classes**.

1. *Classification hiérarchique descendante*



En comparaison avec la C.A.H, à la vue de cette représentation, nous pouvons plus facilement conclure quant à une classification débouchant en 3 classes ; ce qui n’était pas le cas lors de la précédente représentation avec le critère d’agrégation de Ward.

Dans ce premier exercice, la classification automatique se rapprochant le plus des résultats obtenus lors des précédents méthodes serait donc la classification hiérarchique descendante, où nous obtenons une classification en 3 classes distinctes significativement.

Exercice II : Les centres mobiles

Le but de cet exercice est de tester les performances de l'algorithme des centres mobiles sur deux jeux de données réelles : Iris et Crabs.

**Données Iris**

* **Partition en K = {2, 3, 4}**

On réalise une classification des données en 2,3 et 4 classes, à l’aide de la fonction kmeans. On réalise des tableaux de contingence :

|  |  |  |  |
| --- | --- | --- | --- |
|  | K=2 | | |
|  | setosa | versicolor | virginica |
| 1 | 0 | 47 | 50 |
| 2 | 50 | 3 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | K=3 | | |
|  | setosa | versicolor | virginica |
| 1 | 50 | 0 | 0 |
| 2 | 0 | 2 | 36 |
| 3 | 0 | 48 | 14 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | K=4 | | |
|  | setosa | versicolor | virginica |
| 1 | 0 | 0 | 27 |
| 2 | 50 | 0 | 0 |
| 3 | 0 | 23 | 22 |
| 4 | 0 | 27 | 1 |

A partir de la partition en 2 classes, on obtient deux classes bien distinctes: une classe contient toute l’espèce setosa et l’autre classe contient versicolor et virginica. Pour une partition en 3 classes, on obtient 3 classes avec à peu près la même cardinalité (1/3 chacune) avec toujours une classe qui contient l’espèce setosa. Pour k=4, l’espèce setosa est toujours contenue par une classe, ensuite la moitié de l’espèce versicolor et de celle de virginica sont dans une même classe. Le reste des espèces est dans des classes distinctes.

On peut en conclure que l’espèce setosa est très éloignée des autres espèces.

* **Etude de la stabilité du résultat de la partition**

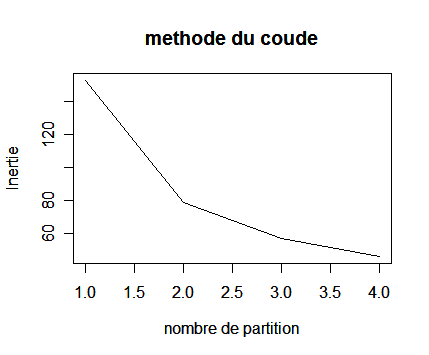
On étudie la stabilité du résultat de partitionnement de kmeans en effectuant 50 fois des classifications en K=3 classes de jeux de donnée.

On remarque que les valeurs obtenues peuvent changer. Les 2 valeurs obtenues pour la partition sont 142,75 et 78,85. La valeur d’inertie intra-classes permettant une plus grande stabilité est la valeur minimum (78,85).

L’explication est la suivante : initialement, la méthode des kmeans prend au hasard un nombre de points du jeu de données égal au nombre de classes désiré. L’algorithme est ensuite appliqué à ces centres initiaux. Ce mécanisme engendre le constat suivant : d’une exécution à l’autre, les points de départ de la méthode ne sont pas les mêmes. La fonction kmeans n’est donc pas stable, le résultat dépend donc des premiers centres mobiles choisis.

* **Détermination du nombre de classes optimal**

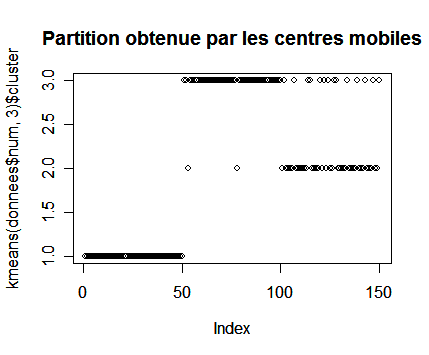
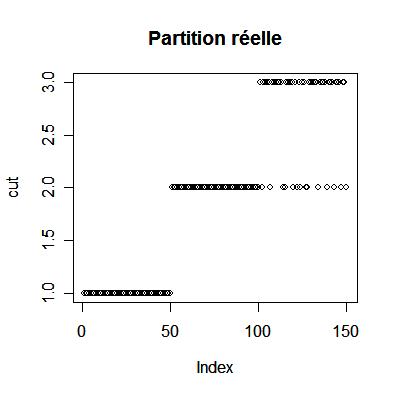
La méthode du coude nous aide à proposer un nombre de classes plausible pour la méthode des kmeans.



On voit clairement ici le ”coude” pour un partitionnement en 3 classes. Dans la mesure où nous voulons l’information sur l’espèce, un partitionnement en 3 classes est largement suffisant puisque au-delà de 3 classes, le partitionnement ne réduit plus de manière significative l’inertie intra-classe.

* **Comparaison entre les 2 méthodes**

On compare la partition obtenue par les centres mobiles avec la partition réelle des iris en trois groupes:



Les résultats sont comparés avec la réalité. 100% des Setosa sont reconnus par la partition des centres mobiles, 96% des Versicolors sont identifiés, et seulement 72% des Virginia sont dans la bonne classe. Ce qui donne un pourcentage global de réussite de 89,3%. Ceci prouve que cette méthode n’est pas la plus optimale pour étudier ce jeu de données, en comparaison avec la partition réelle.

**Données Crabs**

On compare les résultats de la partition obtenue par les centres mobiles avec la partition réelle des iris en trois groupes. Voici le résultat avec F = femelle et M = mâle et B = bleu et O = orange:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | F&B | F&O | M&B | M&O |
| 1 | 0 | 49 | 0 | 8 |
| 2 | 50 | 1 | 13 | 0 |
| 3 | 0 | 0 | 0 | 42 |
| 4 | 0 | 0 | 37 | 0 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | F&B | F&O | M&B | M&O |
| 1 | 50 | 1 | 16 | 0 |
| 2 | 0 | 0 | 34 | 0 |
| 3 | 0 | 49 | 0 | 12 |
| 4 | 0 | 0 | 0 | 38 |

Partition obtenue avec les centres mobiles Partition réelle des Crabs

On remarque que les résultats sont un peu différents mais que la répartition dans les classes est quasiment la même dans les 2 cas. Seulement quelques spécimens (4 au maximum) différent par rapport à la partition réelle. Pour ces données, la méthode des centres mobiles est aussi efficace qu’une partition réelle.

Ainsi, en conclusion de ce TP, nous pouvons dire qu’il n’y a pas une méthode mieux que les autres à proprement dite, et que parfois même une partition réelle s’avère plus efficace qu’une méthode de classification automatique. C’est à nous de choisir la plus juste, ou celle qui déforme le moins la réalité.