TP2 Clustering FST

L'objectif de la séance est de faire un rappel des algorithmes de clustering. On les utilisera sur les données des iris.

N'oubliez pas de changer de répertoire. par ailleurs, chargez immédiatement les librairies cluster et FactoMineR avec les commandes library(cluster) puis library(FactoMineR).

K-means

La méthode des K-means est une méthode de partitionnement non hiérarchique en k classes, k fixé a priori. Rechargez les sauvegardes du TP précédent. En principe, vous avez immédiatement accès aux données d'iris. Sinon, rechargez-les.//

Nous allons partitionner le fichier des iris en 3 classes en utilisant uniquement les données quantitatives. J'appelle le résultat resclus1 (vous lui donnez le nom que vous voulez)

```
resclus1=kmeans(iris[,1:4],3)
```

Si vous tapez resclus1, vous allez afficher le contenu de l'objet resclus1. Cet objet a un certain nombre de composants cluster, centers, etc... En tapant resclus1$un des noms précédents, vous avez le contenu du composant.

- Examinez les groupes qui ont été trouvés. Que pensez-vous de ce partitionnement ? La commande suivante vous fournit un tableau croisant les 3 classes obtenues avec les espèces d'iris. Que pensez-vous du résultat ?
  ```
  table(resclus1$cluster,iris[,5])
  ```
- En fait, on n'a pas standardisé les variables avant d'effectuer les kmeans, ce qui donne plus d'importance aux variables qui ont une grande variance. On va donc centrer et normer les variables. Quel sera l'effet de cette standardisation. La commande suivante scale permet cela. On appelle le résultat stairis.
  ```
  stairis=scale(iris[,1:4])
  resclus2=kmeans(stairis[,1:4],3)
  table(resclus2$cluster,iris[,5])
  ```

Examinez ces nouveaux résultats. Refaites plusieurs fois un k-means sur les iris oroginaux et sur les iris standardisés. Que constatez-vous ? Vous pouvez aussi comparer vos résultats avec vos voisins et vous allez vous apercevoir que vous n'avez pas tout à fait la même chose. En effet les k-means sont sensibles aux tirages initiaux. Combien d'iris sont "mal classés" dans chacun des cas ?

–

– Au lieu de travailler sur les iris, on va prendre les deux premières coordonnées de l'ACP.et refaire un k means avec exactement la même séquence de traitements qu'auparavant.

```
ind2axe=res.pca$ind$coord
resclus3=kmeans(ind2axe[,1:2],3)
resclus3
table(resclus3$cluster, iris[,5])
```

Que constatez-vous ? Sur la qualité des groupes trouvés ? Que concluez-vous ?

CAH

On va faire une classification ascendante hiérarchique sur les données standardisées, on va utiliser la méthodes de Ward La commande est la suivante (avec ward) :

rescah1=agnes(scale(stairis),method="ward") puis pour afficher l'arbre, faites plot(rescah1) et appuyez sur "entrée". Le premier graphique ne vous intéresse pas . Refaites "entrée", vous aurez l'arbre de classification. Ce qui nous intéresse ici est la structure de l'arbre. En effet, on ne peut rien voir sur les extrémités des feuilles. On a envie de couper l'abre en 2 ou en 3 ou en 5. Essayez les différentes possibilités et examiner les résultats.

Par exemple classward=cutree(rescah1, k=2) vous coupe l'arbre en 2. Puis table(classward,iris[,5]) vous permet de comprendre les résultats. Que constatez-vous ?

Avec la CAH, il faut décider de couper l'arbre à un certain niveau. pour cela, on peut s'aider en examinant les valeurs de la distance ultramétrique (height).Vous pouvez l'afficher en faisant les deux commandes suivantes :

```
rescah2=as.hclust(rescah1)
```

```
plot(rev(rescah2$height),type="h", ylab="indice ")
```

as.hclust convertit un objet "clust" en classes.ET rev inverse l'ordre de l'index. Ne le faites pas et regardez ce que ça donne. Vous aurez le symétrique du dessin précédent.

On va refaire la même chose que précédemment en prenant les 2 premières coordonnées factorielles.

```
rescah3=agnes(ind2axe,method="ward")
plot(rescah3 )
rescah4=as.hclust(rescah3)
plot(rev(rescah4$height),type="h", ylab="indice ")
 classward=cutree(rescah3, k=2)  ou k=3 ou k=5
 table(classward,iris[,5])
```

Que concluez-vous ?