Une étude de l'académie nationale des Sciences des Etats-Unis d'Amérique a montré que la classification des carcinomes pulmonaires humains par les profils de l'expression de l'ARNm révèle des sous-classes d'adénocarcinome distinctes.

Description des données :

Cet ensemble de données contient 56 variables mesurées sur 12625 gènes en utilisant Affymetrix GeneChip 95av2 (dévouée à l'acquisition, l'analyse et la gestion les informations génétiques complexes). Parmi les 56 variables 20 mesurent les carcinoïdes pulmonaires (carcinoïdes), 13 sont relatives aux métastases du cancer du côlon (colon), 17 au fonctionnement pulmonaire normal (Normal) et 6 au carcinome du poumon à petites cellules (SmallCell).

Les libellés des lignes indiquent les noms des gènes.

Travail à faire :

1. L'objectif ici est de retrouver la classification donnée par l'académie des Sciences Américaine.
   Appliquer les méthodes de classification pour retrouver la vraie configuration des données et déterminer le nombre optimal de classes de gènes ainsi que l'appartenance des gènes aux classes.
2. Comparer les méthodes de classification vues en cours pour prédire les classes des gènes en utilisant la validation croisée pour déterminer les valeurs optimales des paramètres dans chaque méthode. i.e nombre de voisins les plus proches dans la KNN, nombre d'arbres bootstrap dans le bagging, …

```r
#On transforme le fichier en un fichier csv au préalable
#On ouvre le fichier à l'aide de read.table
#On indique que la première colonne et la première lignes correspondent aux noms respectives des variables/individus
dat = read.table("C:/Users/kouro/OneDrive/Bureau/CC2 Classification/data.csv", header=T, sep=';', row.names=1)
#On cherche a normaliser et standardiser les données à l'aide de scale pour qu'elles aient tous la même importance
data=scale(dat)
#On affiche le début du tableaux des observations
head(data)
#On affiche la fin du tableaux des observations
tail(data)
#On affiche un récapitulatif sur le min, max, mean,... de chacun des variables
summary(data)
```

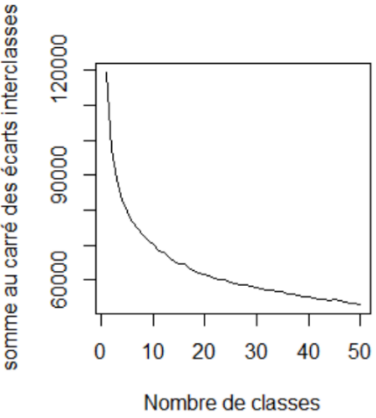
Question 1 : On cherche à retrouver la classification donnée par l'académie des Sciences Américaines. Pour cela, nous devons identifier la vraie configuration des données et déterminer le nombre optimal de classe de gènes ainsi que l'appartenance des gènes aux classes. Nous disposons ici de 56 variables, 20 pour les carcinoïdes, 13 pour les métastases et 17 pour le fonctionnement pulmonaire normal ainsi que 6 autres carcinomes du poumon.

Le partitionnement en kmeans est un problème d'optimisation qui cherche à diviser des points en k groupes, appelés clusters, de façon à minimiser la somme totale des carrés des distances dans chaque classe, ce qu'on appelle aussi la variance intra-classe.

```r
wss = vector()
for (i in 1:50){
  set.seed(i)
  wss[i] =kmeans(data, centers=i)$tot.withinss
}
```

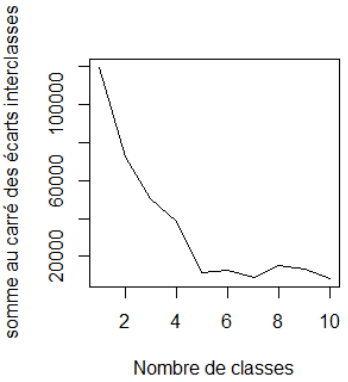
Pour connaître le nombre de classe optimale, nous utilisons une boucle afin de calculer, pour un nombre de classe égale à i, le tot.withinss qui correspond à la somme totale des carrés des distances dans chaque classe. Ainsi, la boucle va réaliser un kmeans sur l'ensemble de toutes les observations en variant le nombre de classe de 1 à 50. On impose un seed afin de garder la même valeur dès lors qu'on relance le programme.

```r
plot(1:50,wss,type="l",xlab="Nombre de classes",ylab="somme au carré des écarts intraclasses")
```



Après avoir lancé le programme, on souhaite déterminer la classe optimale en coupant la courbe là où la différence d'inertie intra-classe est la plus élevée. Afin de faciliter cette analyse, on décide d'afficher une courbe représentant la somme au carré des écarts intra-classes à l'aide de la fonction « plot ». On remarque, par la suite, que cette 'cassure' de la courbe est comprise entre 2 et 10 classes.

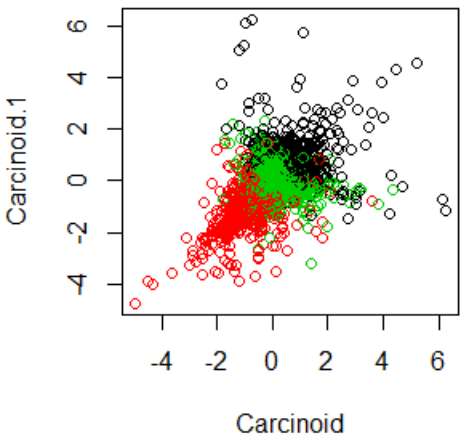
```r
wss = vector()
for (i in 1:10){
  set.seed(i)
  wss[i] =kmeans(data, centers=i)$withinss
}
plot(1:10,wss,type="l",xlab="Nombre de classes",ylab="somme au carré des écarts intraclasses")
```



On relance donc le programme avec maintenant un i qui varie de 1 à 10.

On remarque plus facilement ici qu'on a une 'cassure', c'est-à-dire un changement de la pente de la variance intra-classe à partir de i=3, ce qui d'après la méthode du coude (Elbow method) correspond au nombre de classes optimales.

```r
#On créer 3 classes
cent <- rbind(data[1,], data[2,],data[3,])
#classification automatique à l'aide de la méthode des centres mobiles
cl <- kmeans(data, centers=cent)
#On affiche les 3 classes à l'aide de couleurs différentes
plot(as.matrix(data),col=cl$cluster,axes=T)
```



On sait donc que le nombre optimal de classe est égale à 3, on divise donc nos données en 3 classes que l'on regroupe et classe automatiquement à l'aide de la méthode du kmeans ou méthode des centres mobiles. On affiche donc ces 3 classes :

On obtient ainsi une représentation des trois classes, déterminées précédemment. Chacune d'entre elles est représentée par une couleur différente, avec, en abscisse Carcinoïde et en ordonnée la variable Carcinoïd.1

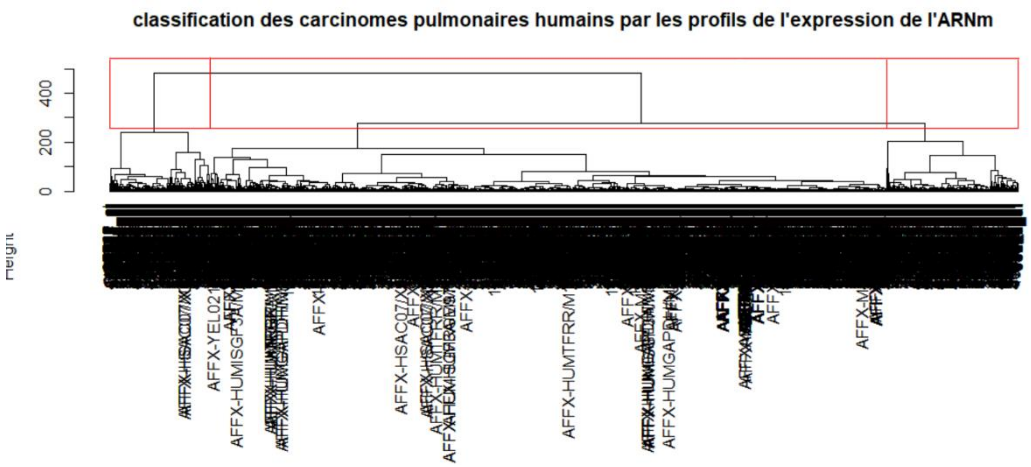
Ensuite, on a deux possibilités :

-1ère possibilité :

```r
      #1ère possibilité:
#On peut ensuite calculer le tableau des distances euclidienne :
dist = dist(data)
#On représente à l'aide de la fonction hclust les différentes classes en utilisant la méthode 'ward.D2'
hc = hclust(dist, method = "ward.D2") # représenter le clustering
#On affiche à l'aide de plot le dendogramme
plot(hc,hang = -1, main = "classification des carcinomes pulmonaires humains par les profils de l'expression de l'ARNm")
#On cherche à obtenir nos 3 classes en les représentant sur le dendogramme
rect.hclust(hc,3)
# on coupe le dendrogramme en 3 classes
class = cutree(hc, k = 3)
#On regroupe toutes ces informations dans un tableau
tt = table(row.names(data),class)
tt
#On attribut chacun des individus à sa classe
c1 = tt[,1]==1
class1 = row.names(data)[c1]
c2 = tt[,2]==1
class2 = row.names(data)[c2]
c3 = tt[,3]==1
class3 = row.names(data)[c3]
```

On calcule le tableau des distances euclidiennes, puis on utilise l'algorithme CAH avec la méthode de 'ward.D2' qui permet d'obtenir une agrégation qui minimise la perte d'information (permet de compacter les clusters).

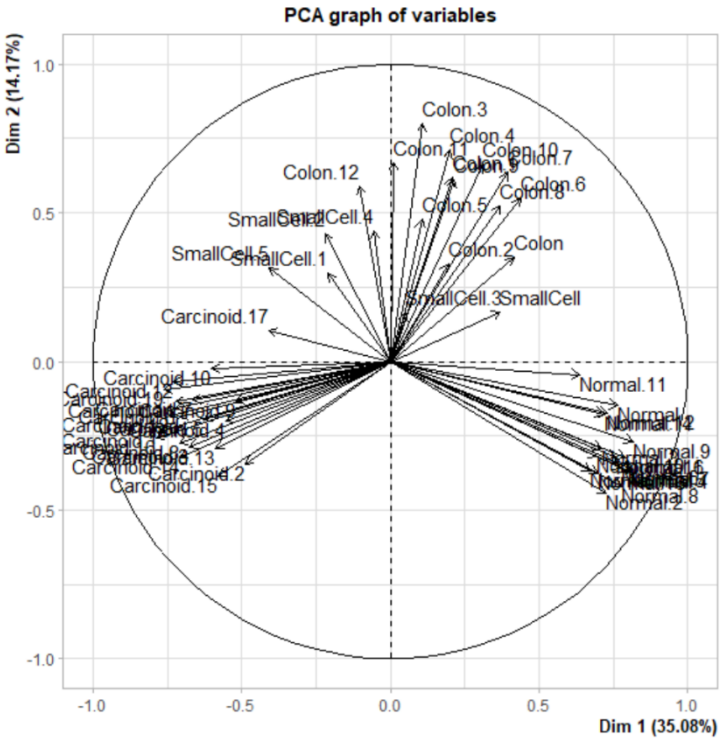
On affiche le dendrogramme que l'on divise en 3 classes optimales à l'aide de cutree, puis, on encadre ce dernier en rouge. On peut maintenant afficher pour chaque individu le groupe auquel il appartient. Pour cela on crée le tableau tt, qui va permettre de regrouper les informations pour pouvoir ensuite attribuer à chaque individu sa classe.



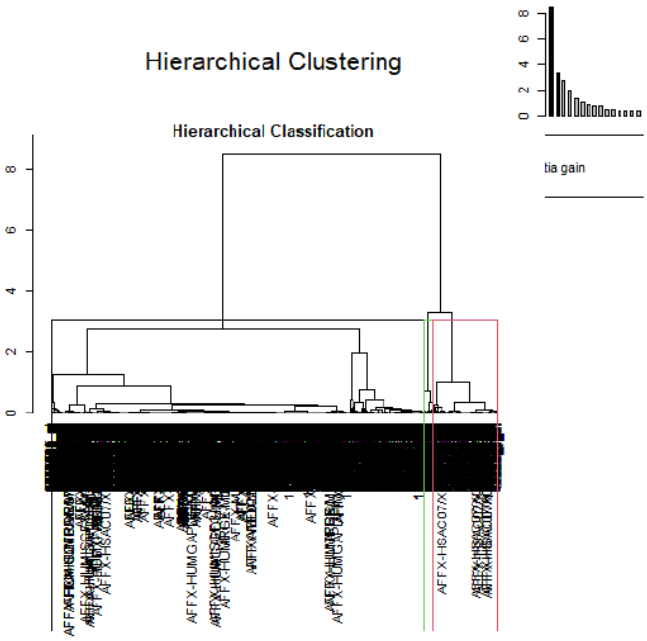
-2ème possibilité :

On effectue une classification hiérarchique sur l'ensemble des variables à l'aide du package FactoMineR, on regarde ensuite les valeurs propres associées à cette ACP.

```
install.packages("FactoMineR")
library(FactoMineR)
#On effectue une ACP
results = PCA(data)
#On regarde les valeurs propres de cette ACP
results$eig
```

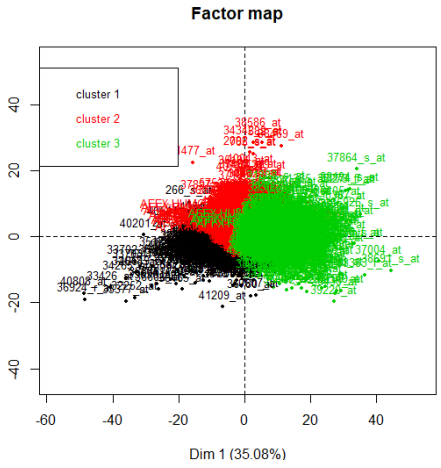


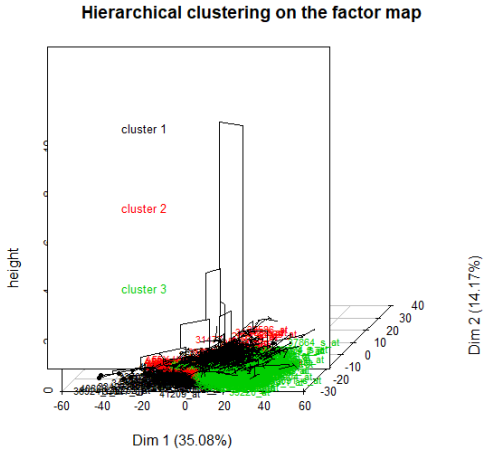
On souhaite conserver uniquement les composantes principales qui sont significatives, c'est-à-dire que leurs valeurs propres sont supérieures à 1. D'après ce critère, on remarque que seul 9 variables sont explicatives.



À l'aide de la fonction HCPC on obtient notre arbre de classification qui est élagué par R lorsque le saut d'inertie est maximal. On obtient ainsi un dendrogramme qui est découpé en 3 classes pour l'optimiser. L'arbre hiérarchique suggère une partition en trois classes :

On obtient un arbre hiérarchique tridimensionnel et un plan factoriel où chaque individu est coloré en fonction de la classe à laquelle il appartient.





Nous allons donc continuer notre analyse avec 3 classes, ce qui paraît être le choix optimal afin de réduire ces données tout en conservant le maximum d'information.