TP 2: Classification automatique

Exercice 1. Treillis de Galois

1) Utilisez le logiciel Galicia (mis sur le site Moodle du cours) pour calculer le treillis de ce tableau.

2) Interprétez brièvement mais soigneusement les classes les plus intéressantes (celles qui ne sont pas "trop petites"). Vous zoomerez sur les parties du treillis qui vous paraissent les plus intéressantes, et expliquerez en quoi elles vous ont intéressé/e.

Caractéristiques :	Enfants		Ados-adultes		Longueur (minutes)				Distraction		Éducation		Images réelles		Images de synthèse	
Films	O	N	O	N	< 30	30-60	60-180	> 180	O	N	O	N	O	N	O	N
Série policière (SP)		1	1					1	1			1	1			1
Série humoristique (SH)	1		1					1	1			1	1			1
Long métrage d'animation (LA)	1		1				1		1			1		1	1	
Court métrage d'animation (CA)	1			1	1				1			1		1	1	
Clip chanson (CC)	1		1		1				1			1	1		1	
Documentaire nature/géographie (DN)	1		1			1			1		1		1			1
Documentaire histoire/politique (DH)		1	1			1				1	1		1		1	
Documentaire art/culture (DA)		1	1			1			1		1		1		1	
Documentaire sciences (DS)		1	1			1			1		1		1		1	
Film de *fantasy* (FF)	1		1				1		1			1	1		1	
Film horreur/science fiction (FH)		1	1				1		1			1	1		1	
Film dramatique (FD)		1	1				1		1			1	1			1
Film policier (FP)		1	1				1		1			1	1			1
Film comique (FC)	1		1				1		1			1	1			1

Exercice 2. Classification hiérarchique de parcelles forestières tropicales

*NB. L'interface **R-studio** de R est fortement conseillée.*

Charger dans le logiciel les données relatives au peuplement arboré de la forêt du bassin du Congo (**Datagenus.csv**). Inspectez le fichier et corrigez-en les erreurs triviales s'il en est. Ces données fournissent sur 1000 parcelles de cette forêt: les variables de comptage de 27 espèces d'arbres (*gen1*, ..., *gen27*), la surface de la parcelle, le type forestier (*forest*) tel qu'identifié par les écologues . On ne tiendra pas compte des autres variables. Calculer la densité de peuplement de chaque espèce par unité de surface pour les 1000 parcelles. Les parcelles seront traduits en nuage dans l'espace des 27 densités de peuplement, $\mathbb{R}^{27}$.

1 - Préparation des données.

Chargez les données dans un dataframe.

a) Faut-il standardiser (centrer-réduire) les densités de peuplement ? Justifiez votre réponse. Indice: formulez la distance euclidienne entre deux parcelles et examinez-y la contribution de chaque variable.

b) Si nécessaire (seulement), standardisez les densités de peuplement.

2 - CAH des parcelles sur les densités de peuplement.

Le dataframe contenant les densités de peuplement produit à la question 1 est ci-dessous nommé *datapeuple*.

Il est demandé de procéder à la classification des parcelles décrites par leur peuplement arboré, de sélectionner le nombre de classes des partitions prometteuses. Plusieurs indices (Ward, saut maximum, saut moyen et saut minimum) pourront être utilisés, et leurs résultats comparés notamment à l'aide de l'indice de Rand. Pour faire tout ça, on *pourra* utiliser les commandes données ci-après (attention, ces commandes ne sont données qu'à titre d'exemple pour vous aider si besoin est. Il ne s'agit pas de les exécuter dans l'ordre sans réfléchir, mais de les *adapter aux besoins de votre démarche !*).

a) création de la matrice des distances euclidiennes:

```
> dp=dist(datapeuple, method="euclidean")
```

b) CAH avec Ward :

```
> CAHDP = hclust(d=dp, method = "ward.D")
```

c) Dendrogramme de la hiérarchie indicée:

```
> plot(CAHDP)
```

d) Coupure de l'arbre et fabrication de la variable de classe correspondant à la partition obtenue, par exemple k=2 classes:

```
> PDP2 = cutree(tree = CAHDP, k=2)
```

e) Calcul du R^2 des variables avec la variable de classe. On va stocker tous les R^2 dans un seul vecteur: R2.

```
> R2_PDP2 = cbind(rep(0 , ncol(datapeuple)))
```

Puis, on calcule les R^2 de toutes les variables avec la variable de classe et on met les résultats dans R2:

```
> for (i in cbind(1:ncol(datapeuple))) {R2_PDP2[i] =
summary(lm(datapeuple[,i]~as.factor(PDP2)))$r.squared}
```

On peut réassigner les noms des variables aux éléments de ce vecteur:

```
> row.names(R2_PDP2) = colnames(datapeuple)
```

f) Calcul du R^2 de la partition:

On rappelle que, les variables ayant toutes le même poids, le R^2 de la partition est égal à la moyenne des R^2 des variables. On se sert de cette propriété pour le calculer:

```
> R2G_PV2 = mean(R2_PV2)
```

3 - Optimisation d'une partition avec les K-means

Chacune des partitions jugée prometteuse à l'issue de la CAH doit être optimisée avec la méthode des K-means. Après avoir optimisé chacune de ces partitions, vous en interpréterez les classes

E.g. pour la partition en 2 classes:

- Transformation d'une variable qualitative en matrice d'indicatrices:

```
> IC2DP = data.frame(model.matrix(~as.factor(PDP2)-1))
```

- Calcul matriciel des centres de gravité de classes de la CAH:

Démontrez que les centres de gravité (CentresC2) sont obtenus par la formule programée ci-dessous, que vous écrirez mathématiquement:

```
> mIC2DP = as.matrix(IC2DP)
> mDP = as.matrix(datapeuple)
> CentresC2 = solve(t(mIC2DP) %*% mIC2DP)  %*%  t(mIC2DP)%*% mDP
```

- K-means à partir de ces centres initiaux:

```
> KMDP2 = kmeans(datapeuple, CentresC2)
```

- La variable de classe ainsi produite est dans:

```
> KMDP2$cluster
```

- Boxplot d'une variable x^j conditionnellement à la variable de classe:

```
> boxplot(datapeuple[,j]~as.factor(KMDP2$cluster))
```