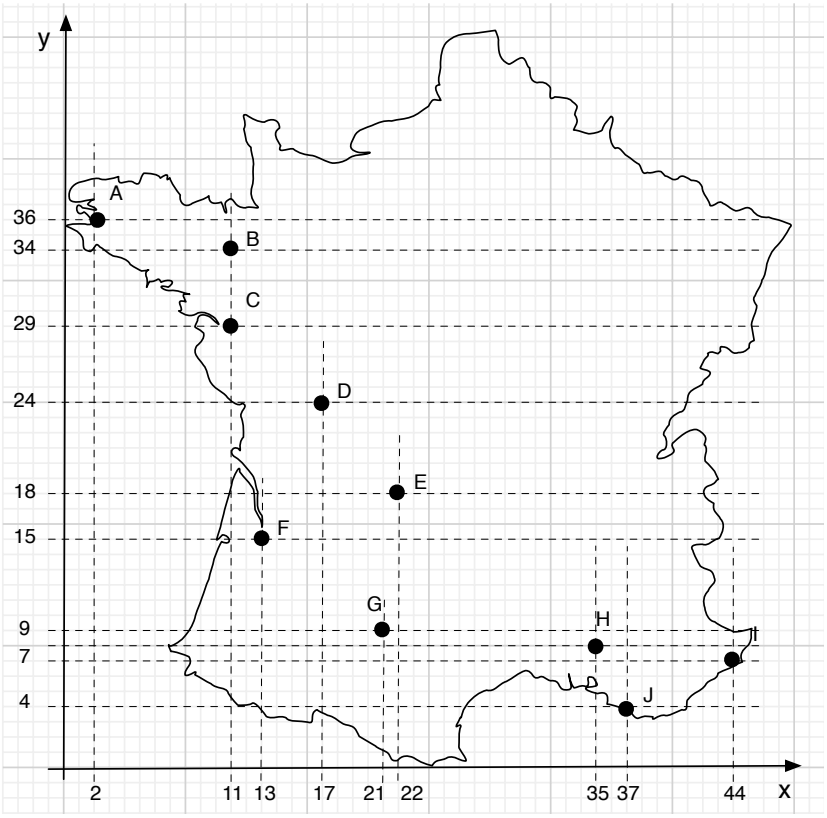
TD1 : Classification hiérarchique ascendante

Objectif : faire exécuter l'algorithme de classification hiérarchique ascendante à la main puis avec le logiciel R. Le travail est à faire en binôme, un compte rendu est à rendre en fin de TD.

Données : la carte ci-dessous, les points de A à I représentent des villes. Les axes x et y fournissent les coordonnées des villes sur la carte.



Questions :

1- Donner les attributs de cet individu
2- Choisir une distance entre individus
3- Calculer le centre de gravité gl : $g_l = \frac{1}{N_l} \sum_{i \in C_l} x_i$
4- Calculer l'inertie : $\mathcal{I} = \sum_{i=1}^{i=N} d^2(x_i, g)$
5- Avec Excel faire les étapes de l'algorithme ci-dessous jusqu'à obtenir 3 classes. On utilisera le saut minimum pour la mise à jour des distances.
6- Pour chaque itération calculer l'inertie intraclasse et interclasse avec w_i le rapport entre le nombre d'individus de la classe i sur le nombre total d'individus.

$$\mathcal{I}_W = \sum_{i=1}^{i=K} w_i \mathcal{I}_i \qquad \mathcal{I}_B = \sum_{i=1}^{i=K} w_i d^2(g_i, g)$$

Algorithme :

- chaque individu correspond à une classe
- calculer une matrice des distances entre tous les points (10x10)
- chercher la plus petite distance *d(i, j)* avec *i !=j* et *i < j*
- tant que *(d(i, j) < seuil) et (le nombre de classes est > 1)* faire
    - fusionner la classe *i* et la classe *j* dans la classe *i* en renommant la colonne *i* et la ligne *i* de la matrice des distances en *ij*
    - mettre à jour les distances de la ligne *i* : *d(i, k) = min(d(i, k) , d(j, k)*
    - mettre à jour les distances de la colonne *i* : *d(k, i) = min(d(k, i) , d(k, j)*
    - supprimer la colonne *j* et la ligne *j*
    - chercher la plus petite distance *d(i, j) avec i !=j et i < j*
- 5- Sous le logiciel R, exécutez les commandes suivantes. Lancer R dans un terminal, copier les commandes une à une. Sauvegarder les graphiques.

```
data <- read.table("carte.csv", header=TRUE, sep="\t", dec = ",", row.names=3)
data
plot(data$x, data$y)
dist <- dist(data)
hac <- hclust(dist, method='single')
plot(hac)
groups <- cutree(hac, k=4)
rect.hclust(hac, k=4, border="red")
foo <- cbind(data, classe=groups)
foo
plot(foo$x, foo$y, col=palette()[foo$classe])
q(y)
```

En utilisant les aides (tutoriel R et autres) sur internet, commenter les lignes suivantes.

Vérifier que vous obtenez la même classification que dans la question 4.
Tester le saut maximum (*complete*) et le saut moyen (*average*).

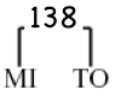
Exemple sur un autre jeu de données (http://ouestgenopuces.univ-rennes1.fr/formations/analyse_partie2c.pdf)

Matrices des distances

BA : Bari
FL : Florence
MI : Milan
NA : Naples
RM : Rome
TO : Turin

	BA	FL	MI	NA	RM	TO
BA	-					
FL	662	-				
MI	877	295	-			
NA	255	468	754	-		
RM	412	268	564	219	-	
TO	996	400	138	869	669	-

L(MI, TO) = 138



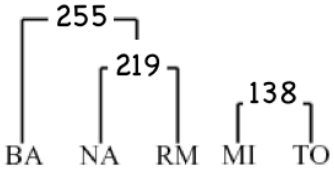
	BA	FL	MI/ TO	NA	RM
BA	-				
FL	662	-			
MI/ TO	877	295	-		
NA	255	468	754	-	
RM	412	268	564	219	-

HCL single linkage

	BA	FL	MI/ TO	NA	RM
BA	-				
FL	662	-			
MI/ TO	877	295	-		
NA	255	468	754	-	
RM	412	268	564	219	-

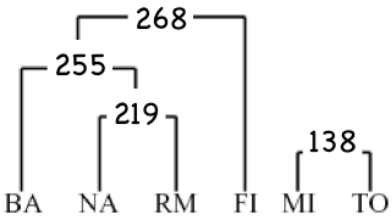
L(NA, RM) = 219

L(BA, NA/RM) = 255



	BA	FL	MI/ TO	NA/ RM
BA	-			
FL	662	-		
MI/ TO	877	295	-	
NA/ RM	255	268	564	-

L(BA/NA/RM, FL) = 268



	BA/NA/ RM	FL	MI/TO
BA/NA/ RM	-		
FL	268	-	
MI/TO	564	295	-

	BA/NA/ RM/FL	MI/TO
BA/NA/ RM/FL	-	
MI/TO	295	-

L(BA/NA/RM/FI, MI/TO) = 295