Classifications de villes françaises à partir des températures mensuelles moyennes

------ **Documentation**
Documentation pour Kmeans (short) :
http://www.duclert.org/r-analyse-donnees/clustering-kmeans-R.php
Documentation pour CAH (short) :
http://www.duclert.org/r-analyse-donnees/clustering-hierarchique-R.php
Adresse du Mooc (Husson) : http://factominer.free.fr/course/MOOC_fr.html
Adresse de ressources Husson : https://husson.github.io/data.html
Adresse de ghub Husson : https://husson.github.io/

------ Exemple du cours
Exemple Temperatures - Classification

1- Importe les données en précisant que le nom des individus est dans la première colonne
```
library(vtable)
setwd("C:/Users/OHCE7285/OneDrive - orange.com/Bureau/Local/Univ/Cours/2024 Cours M2")
temperature<-read.csv("temperat.csv",header=TRUE,sep=";",row.names=1)
```
Donne un résume de la base de données (stats descriptives)
```
summary(temperature)
```

```
Janvier          Fevrier           Mars            Avril            Mai              Juin
Juillet          Aout
 Min.   :-9.300   Min.   :-7.900   Min.   :-3.700   Min.   : 2.900   Min.   : 6.50   Min.   :
9.30   Min.   :11.10   Min.   :10.60
 1st Qu.:-1.550   1st Qu.:-0.150   1st Qu.: 1.600   1st Qu.: 7.250   1st Qu.:12.15   1st
Qu.:15.40   1st Qu.:17.30   1st Qu.:16.65
 Median : 0.200   Median : 1.900   Median : 5.400   Median : 8.900   Median :13.80   Median
:16.90   Median :18.90   Median :18.30
 Mean   : 1.346   Mean   : 2.217   Mean   : 5.229   Mean   : 9.283   Mean   :13.91   Mean
:17.41   Mean   :19.62   Mean   :18.98
 3rd Qu.: 4.900   3rd Qu.: 5.800   3rd Qu.: 8.500   3rd Qu.:12.050   3rd Qu.:16.35   3rd
Qu.:19.80   3rd Qu.:21.75   3rd Qu.:21.60
 Max.   :10.700   Max.   :11.800   Max.   :14.100   Max.   :16.900   Max.   :20.90   Max.
:24.50   Max.   :27.40   Max.   :27.20
   Septembre        Octobre         Novembre         Decembre        Moyenne        Amplitude
Latitude       Longitude
 Min.   : 7.90   Min.   : 4.50   Min.   :-1.100   Min.   :-6.00   Min.   : 4.50   Min.
:10.20   Min.   :37.20   Min.   : 0.00
 1st Qu.:13.00   1st Qu.: 8.65   1st Qu.: 3.200   1st Qu.: 0.25   1st Qu.: 7.75   1st
Qu.:14.90   1st Qu.:43.90   1st Qu.: 4.35
 Median :14.80   Median :10.20   Median : 5.100   Median : 1.70   Median : 9.70   Median
:18.50   Median :50.00   Median : 9.40
 Mean   :15.63   Mean   :11.00   Mean   : 6.066   Mean   : 2.88   Mean   :10.27   Mean
:18.32   Mean   :48.77   Mean   :11.98
 3rd Qu.:18.25   3rd Qu.:13.30   3rd Qu.: 7.900   3rd Qu.: 5.40   3rd Qu.:12.65   3rd
Qu.:21.45   3rd Qu.:52.75   3rd Qu.:18.65
 Max.   :24.30   Max.   :19.40   Max.   :14.900   Max.   :12.00   Max.   :18.20   Max.
:27.60   Max.   :64.10   Max.   :30.30
    Region
 Length:35
 Class :character
 Mode  :character
```
Remarque : la table ci-dessus est illisible pour un rapport. Privilégier le rapport construit ci-dessous avec st (sumtable)

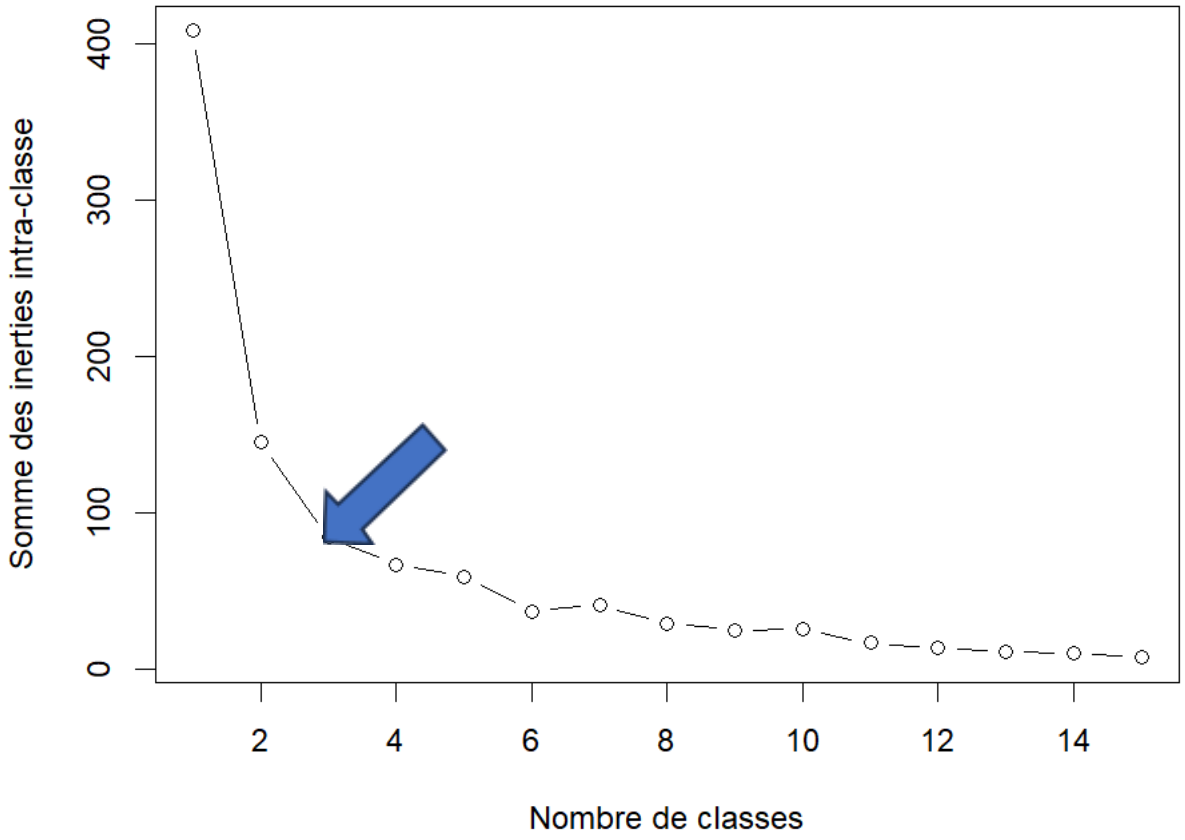
#Statistiques descriptives avec le package vtable
st(data = temperature)

Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
Janvier	35	1.3	5.5	-9.3	-1.5	4.9	11
Fevrier	35	2.2	5.5	-7.9	-0.15	5.8	12
Mars	35	5.2	4.9	-3.7	1.6	8.5	14
Avril	35	9.3	3.8	2.9	7.2	12	17
Mai	35	14	3.3	6.5	12	16	21
Juin	35	17	3.3	9.3	15	20	24
Juillet	35	20	3.6	11	17	22	27
Aout	35	19	3.7	11	17	22	27
Septembre	35	16	4.1	7.9	13	18	24
Octobre	35	11	4.3	4.5	8.6	13	19
Novembre	35	6.1	4.6	-1.1	3.2	7.9	15
Decembre	35	2.9	5	-6	0.25	5.4	12
Moyenne	35	10	4	4.5	7.8	13	18
Amplitude	35	18	4.5	10	15	21	28
Latitude	35	49	7	37	44	53	64
Longitude	35	12	8.9	0	4.3	19	30
Region	35						
... Est	8	23%					
... Nord	8	23%					
... Ouest	9	26%					
... Sud	10	29%					

#Cette table donne pour chaque variable quantitative l'effectif, la moyenne, l'écart-type, le minimum,
#le 25ème percentile, le 75ème percentile et le maximum. On trouve également pour la variable
#qualitative Région le % des villes pour chacune des modalités

# 3 - Classification de type K-means
# Préparation des données
temperature1 <- scale(temperature[,1:12]) # les variables quantitatives sont standardisées, cette
#étape n'est pas forcément obligatoire
Détermination du nombre de classes par le calcul de l'inertie intra classes
On construit ci-dessous 14 partitions construites à partir de K Means (de 2 à 15 classes) et on
calcule l'inertie intra-classes pour chacune d'elles (paramètre withinss)
for (i in 2:15) wss[i] <- sum(kmeans(temperature1,centers=i)$withinss)
# On représente le graphique indiquant pour le nombre de classes de chacune des classifications
#obtenues l'inertie intra-classes correspondante
plot(1:15, wss, type="b", xlab="Nombre de classes",ylab="Somme des inerties intra-classe")



L'objectif est d'avoir une inertie intra-classes faible sans avoir trop de classes. Le point d'inflexion
dans la figure ci-dessous indique qu'une partition est 3 classes donne de bons résultats.

# Mise en place de la classification de type K means permettant d'obtenir 3 classes comme déterminée ci-dessus
fit <- kmeans(temperature1, 3)

Contenu de fit :

K-means clustering with 3 clusters of sizes 8, 18, 9
3 classes ont été construites, d'affectif 8, 18 et 9

Cluster means:

	Janvier	Fevrier	Mars	Avril	Mai	Juin
1	-1.24873106	-1.31472999	-1.37590297	-1.2597957	-0.9695584	-0.7836064
2	-0.04768693	-0.03948801	-0.02758089	-0.1078785	-0.2173242	-0.3272347
3	1.20535702	1.24762491	1.27818665	1.3355754	1.2964782	1.3510083

	Juillet	Aout	Septembre	Octobre	Novembre	Decembre
1	-0.7372303	-0.8899019	-1.1147765	-1.2294885	-1.2242028	-1.226494
2	-0.3731721	-0.3075873	-0.2239363	-0.1497265	-0.1336069	-0.077617
3	1.4016601	1.4061985	1.4387852	1.3923317	1.3553941	1.245451

On donne ici les moyennes des variables **standardisées** pour chacune des classes. On voit déjà que
la classe 1 réunit des villes froides, la classe 2 des villes de températures moyennes et la classe 3
des villes chaudes.

Clustering vector:

```
    Amsterdam       Athenes        Berlin      Bruxelles
            2             3             2             2
     Budapest    Copenhague        Dublin      Helsinki
            2             2             2             1
         Kiev      Cracovie      Lisbonne       Londres
            1             2             3             2
       Madrid        Minsk        Moscou          Oslo
            3             1             1             1
        Paris       Prague     Reykjavik          Rome
            2             2             1             3
     Sarajevo        Sofia     Stockholm        Anvers
            2             2             1             2
    Barcelone      Bordeaux     Edimbourg     Francfort
            3             2             2             2
       Geneve        Genes         Milan       Palerme
            2             3             3             3
      Seville St Petersbourg        Zurich
            3             1             2
```

Il s'agit de la classe associée à chaque ville

```
Within cluster sum of squares by cluster:
[1] 22.47061 37.20052 24.54618
 (between_SS / total_SS =  79.4 %)
```

Inertie intra-classe de chacune des classes

```
Available components:

[1] "cluster"      "centers"      "totss"        "withinss"
[5] "tot.withinss" "betweenss"    "size"         "iter"
[9] "ifault"
```

Ensemble des informations disponibles via « fit »

Moyenne des variables par classe (information déjà fournie par « fit »)

```
aggregate(temperature1,by=list(fit$cluster),FUN=mean)
```

Affectation des individus à leur classe – rajoute au tableau de données une colonne correspondant
à la classe d'appartenance

```
temperatureKM <- data.frame(temperature, fit$cluster)
```

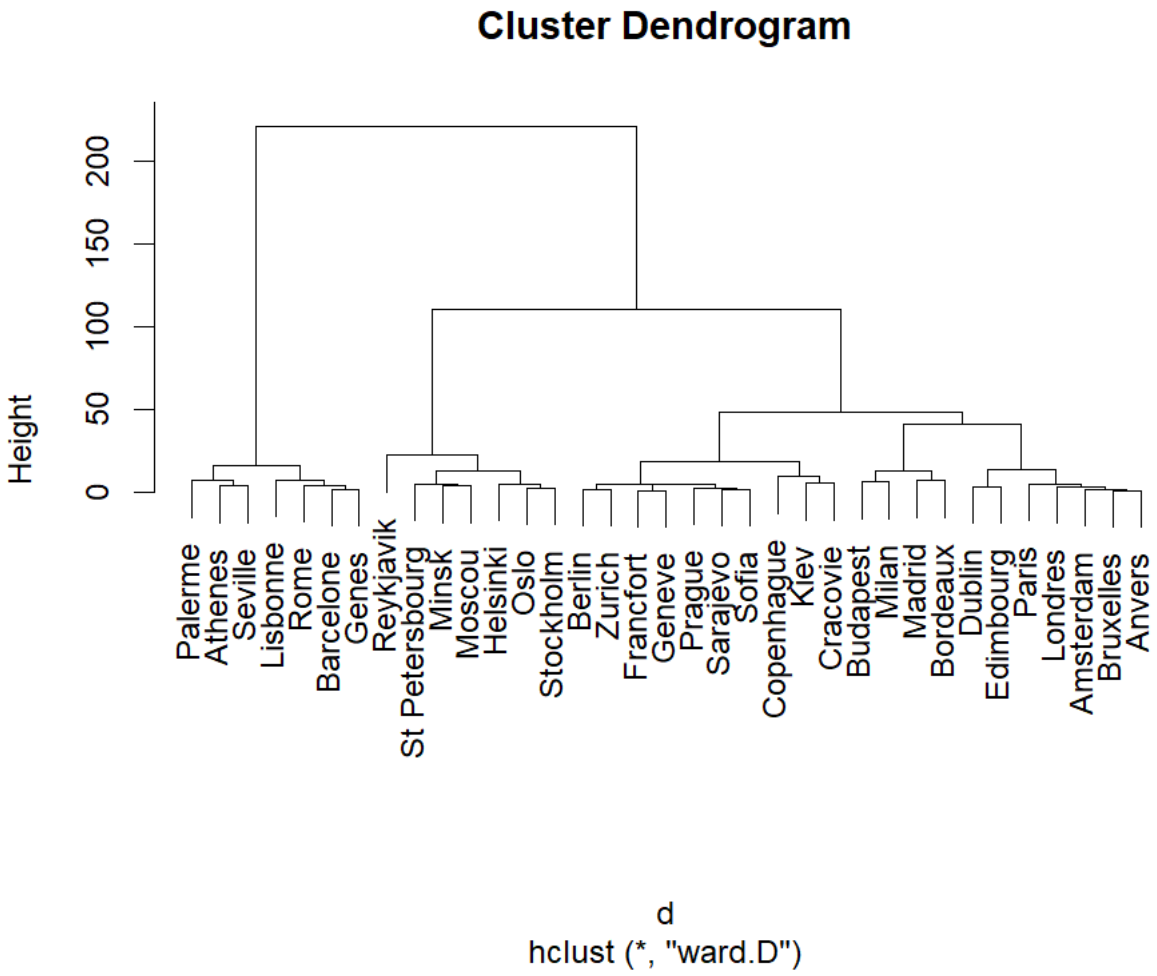
4 - CAH basée sur le critère de Ward

```
d <- dist(temperature[,1:12], method = "euclidean")
```

d est la matrice des distances entre chacune des villes

```
fit <- hclust(d, method="ward.D")
plot(fit)
```

Affiche le dendrogramme :

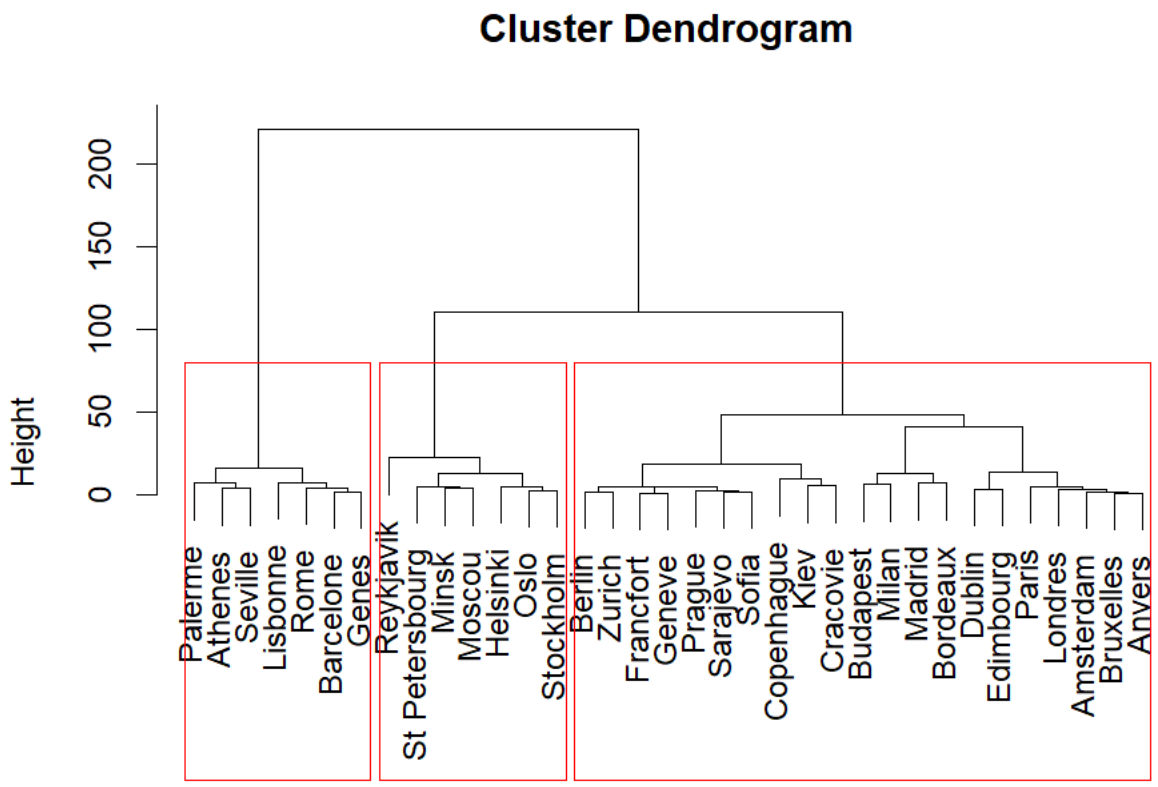


groups <- cutree(fit, k=3)

coupe l'arbre en 3 classes, groups donne la classe d'affectation de chaque ville

```
Amsterdam      Athenes      Berlin     Bruxelles
        1            2           1             1
   Budapest   Copenhague      Dublin     Helsinki
        1            1           1             3
       Kiev     Cracovie    Lisbonne      Londres
        1            1           2             1
     Madrid        Minsk      Moscou         Oslo
        1            3           3             3
      Paris       Prague   Reykjavik         Rome
        1            1           3             2
   Sarajevo       Sofia   Stockholm        Anvers
        1            1           3             1
  Barcelone     Bordeaux   Edimbourg     Francfort
        2            1           1             1
     Geneve        Genes       Milan      Palerme
        1            2           1             2
    Seville St Petersbourg      Zurich
        2            3           1
```

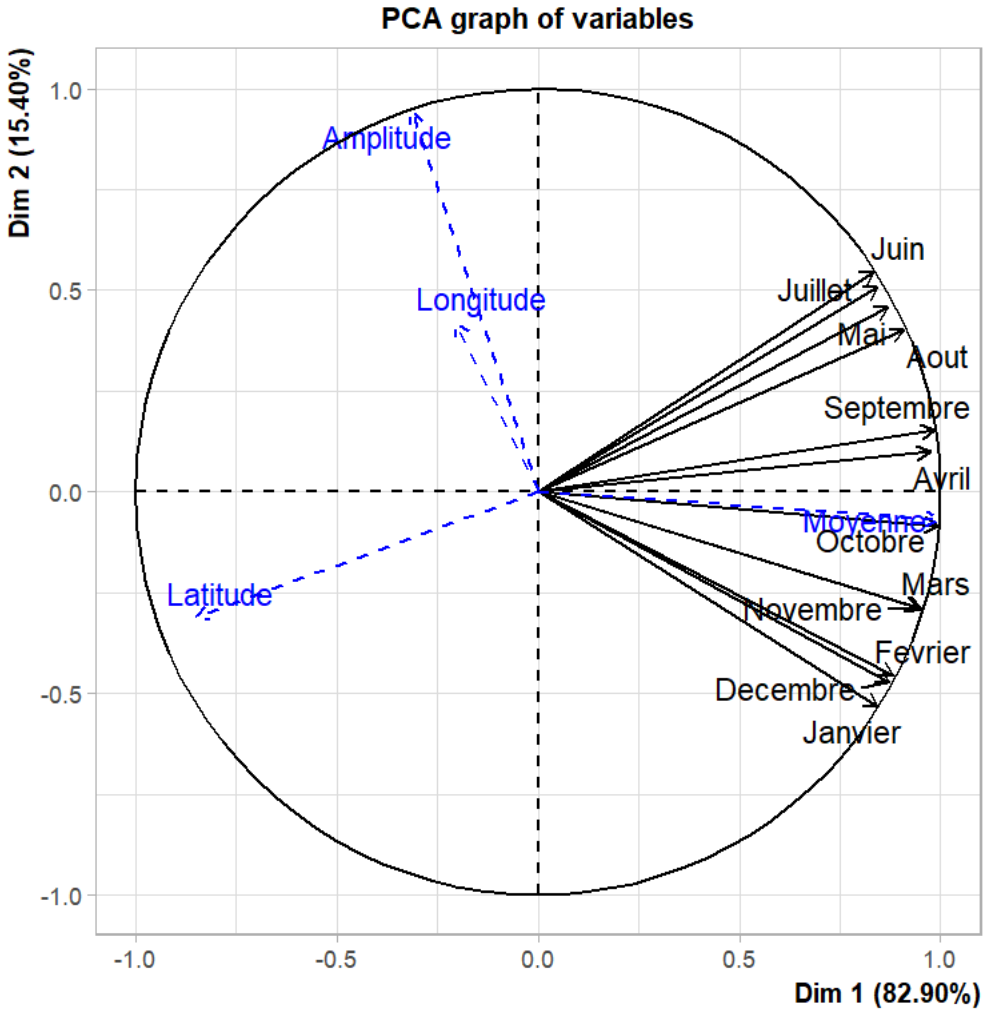
```
rect.hclust(fit, k=3, border="red")
# dessine le dendrogramme en délimitant chaque classe par un rectangle rouge
```



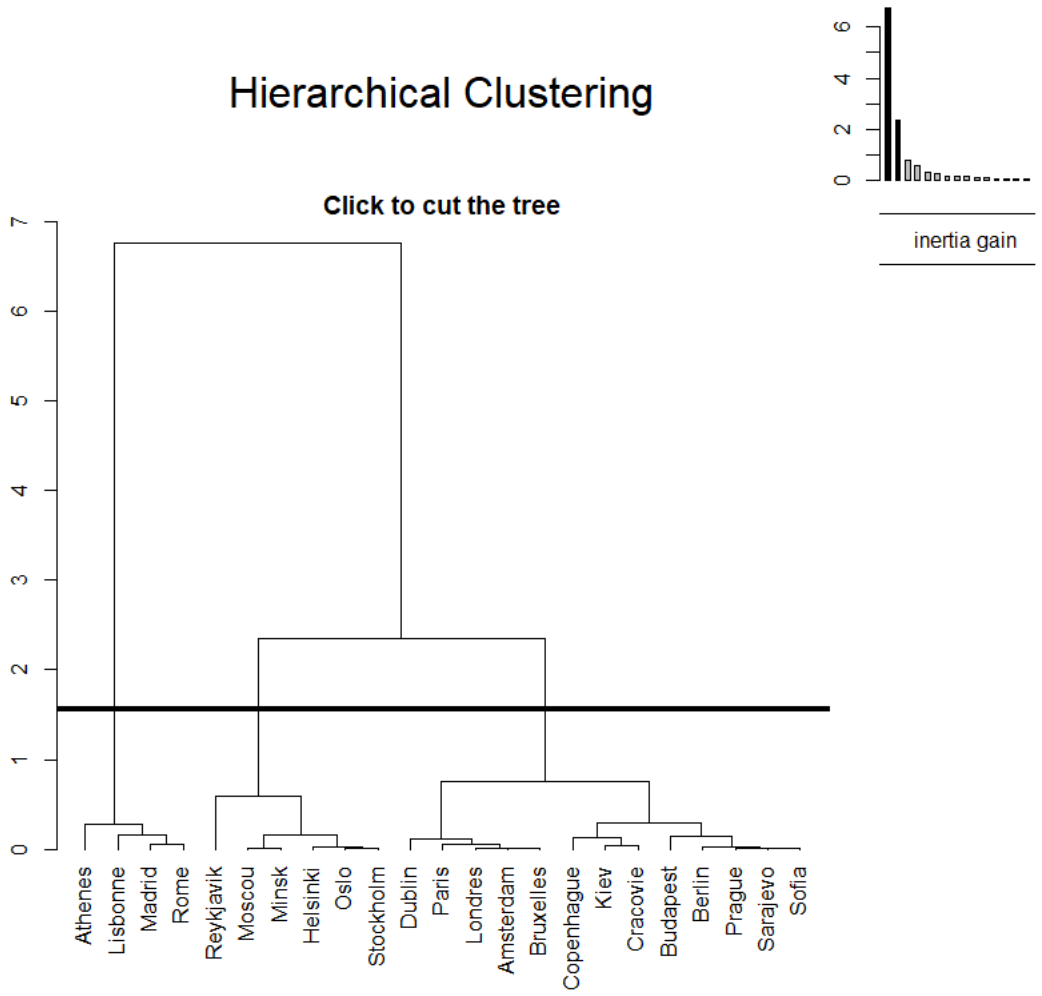

```
aggregate(temperature[,1:12],by=list(groups),FUN=mean)
# Donne la moyenne des variables par classe

Group.1   Janvier   Fevrier     Mars     Avril      Mai     Juin  Juillet
1       1 0.9380952  1.961905  5.180952  9.138095 13.58095 16.84762 18.90952
2       2 9.3857143 10.214286 12.228571 14.928571 18.58571 22.14286 24.74286
3       3 -5.4714286 -5.014286 -1.628571  4.071429 10.22857 14.38571 16.64286
     Aout Septembre   Octobre   Novembre  Decembre
1 18.40952  15.10476 10.519048  5.4761905  2.347619
2 24.41429  22.17143 18.028571 13.5285714 10.514286
3 15.25714  10.67143  5.428571  0.3714286 -3.157143


# 5 -La fonction utilisée dans Husson – utilise les  composantes issues de l'ACP
library(FactoMineR)
res.pca <- PCA(temperature,ind.sup=24:35,quanti.sup=13:16,quali.sup=17)
# Construit l'ACP du tableau de départ en prenant en compte les 12 derniers individus comme
# supplémentaires. Sont également supplémentaires les variables moyenne, amplitude, latitude et
# longitude ainsi que Region
```




```
res.hcpc <- HCPC(res.pca)
# Cette fonction s'applique sur les résultats de l'analyse en composantes principales et donne le
# dendrogramme suivant :
```



Le diagramme en haut à droite donne l'évolution de l'inertie intra-classes en partant d'une partition
à 1 classe. On voit un saut entre 2 et 3 classes. 3 classes semble un bon compromis.

res.hcpc <- HCPC(res.pca,t.levels="all")
# Permet d'obtenir un diagramme interactif, toutefois il semble instable