Projet ADD L3

TALL OUSMANE

Décembre 2023

Question 1######.

On charge en mémoire les données et on en donne un aperçu:

```
# setwd("chemin_vers_répertoire_de_travail") # A modifier! Par exemple:
setwd("C:\\Users\\Tall\\Downloads" )
# En Windows remplacer \ par \\

x <- read.table("climats.txt", sep = ";",
                header = TRUE, row.names = 1)
# View(x)
head(x) # pour afficher les six premières colonnes
```

```
##             January February March April  May June July August September
## Amsterdam       2.9      2.5   5.7   8.2 12.5 14.8 17.1   17.1      14.5
## Athens          9.1      9.7  11.7  15.4 20.1 24.5 27.4   27.2      23.8
## Berlin         -0.2      0.1   4.4   8.2 13.8 16.0 18.3   18.0      14.4
## Brussels        3.3      3.3   6.7   8.9 12.8 15.6 17.8   17.8      15.0
## Budapest       -1.1      0.8   5.5  11.6 17.0 20.2 22.0   21.3      16.9
## Copenhagen     -0.4     -0.4   1.3   5.8 11.1 15.4 17.1   16.6      13.3
##             October November December Annual Amplitude Latitude Longitude
Area
## Amsterdam      11.4      7.0      4.4    9.9      14.6     52.2       4.5
West
## Athens         19.2     14.6     11.0   17.8      18.3     37.6      23.5
South
## Berlin         10.0      4.2      1.2    9.1      18.5     52.3      13.2
West
## Brussels       11.1      6.7      4.4   10.3      14.4     50.5       4.2
West
## Budapest       11.3      5.1      0.7   10.9      23.1     47.3      19.0
East
## Copenhagen      8.8      4.1      1.3    7.8      17.5     55.4      12.3
North
```

Question 2.

```
dim(x)
```

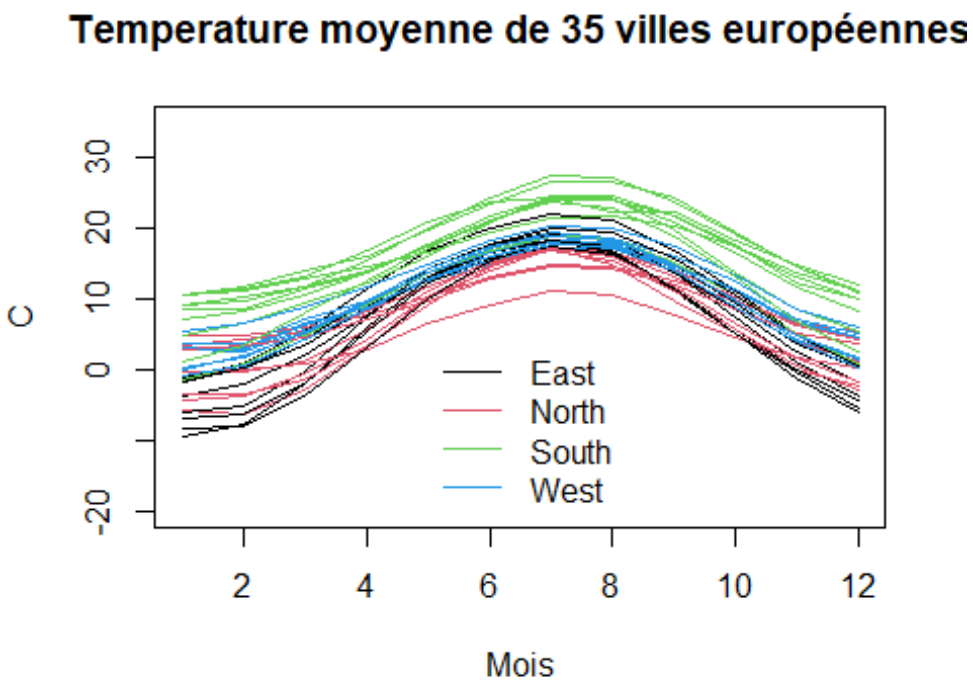
```
## [1] 35 17
```

```
str(x)
```

```
## 'data.frame':    35 obs. of  17 variables:
##  $ January  : num  2.9 9.1 -0.2 3.3 -1.1 -0.4 4.8 -5.8 -5.9 -3.7 ...
##  $ February : num  2.5 9.7 0.1 3.3 0.8 -0.4 5 -6.2 -5 -2 ...
##  $ March    : num  5.7 11.7 4.4 6.7 5.5 1.3 5.9 -2.7 -0.3 1.9 ...
##  $ April    : num  8.2 15.4 8.2 8.9 11.6 5.8 7.8 3.1 7.4 7.9 ...
##  $ May      : num  12.5 20.1 13.8 12.8 17 11.1 10.4 10.2 14.3 13.2 ...
##  $ June     : num  14.8 24.5 16 15.6 20.2 15.4 13.3 14 17.8 16.9 ...
##  $ July     : num  17.1 27.4 18.3 17.8 22 17.1 15 17.2 19.4 18.4 ...
##  $ August   : num  17.1 27.2 18 17.8 21.3 16.6 14.6 14.9 18.5 17.6 ...
##  $ September: num  14.5 23.8 14.4 15 16.9 13.3 12.7 9.7 13.7 13.7 ...
##  $ October  : num  11.4 19.2 10 11.1 11.3 8.8 9.7 5.2 7.5 8.6 ...
##  $ November : num  7 14.6 4.2 6.7 5.1 4.1 6.7 0.1 1.2 2.6 ...
##  $ December : num  4.4 11 1.2 4.4 0.7 1.3 5.4 -2.3 -3.6 -1.7 ...
##  $ Annual   : num  9.9 17.8 9.1 10.3 10.9 7.8 9.3 4.8 7.1 7.7 ...
##  $ Amplitude: num  14.6 18.3 18.5 14.4 23.1 17.5 10.2 23.4 25.3 22.1 ...
##  $ Latitude : num  52.2 37.6 52.3 50.5 47.3 55.4 53.2 60.1 50.3 50 ...
##  $ Longitude: num  4.5 23.5 13.2 4.2 19 12.3 6.1 25 30.3 19.6 ...
##  $ Area     : chr  "West" "South" "West" "West" ...
```

Il y a $p = 12$ variables quantitatives primaires, chacune correspondante à une temperature mensuelle moyenne. L'espace des individus est donc $\mathbb{R}^{12}$. Chaque individu correspond à une ville européene. Au total, il y a $n = 35$ villes.

```
plot(as.numeric(x[1,1:12]), type = 'l',
     col = as.factor(x$Area)[1], ylim = c(-20,35),
     ylab = "C", xlab = "Mois",
     main = "Temperature moyenne de 35 villes européennes")
for(i in 2:nrow(x))
points(as.numeric(x[i,1:12]),type='l',col=as.factor(x$Area)[i])
legend(x = 'bottom', lty = 1,
       col = 1:4, legend = c("East", "North", "South", "West"),
       bty = 'n')
```



Question 3.

##moeynne de chaque variable mean(x$January$)mean($xFebruary) mean(x$March$)mean($xApril) mean(xMay)mean($xJune) mean(x$July$)mean($xAugust) mean(x$September$)mean($xOctober) mean(x$November$)mean($xDecember)

##variance du nuage de points dans l'espace des individus apply(x[,1:12],2,var)

varT<-sum(apply(x[,1:12],2,var)) #moyene du nuage apply(x[,1:12],2,mean)

question4.

#Pour chaque région géographique (variable #supplémentaire #Area) calculer la température moyenne de #chaque mois, la variance de chaque mois, et la variance #totale du sous-nuage

apply(x[x$Area=="West",1:12],2,mean)

apply(x[x$Area=="West",1:12],2,var)

V1<-sum(apply(x[x$Area=="West",1:12],2,var))

V2<-sum(apply(x[x$Area=="East",1:12],2,var))

V3<-sum(apply(x[x$Area=="North",1:12],2,var))

V4<-sum(apply(x[x$Area=="South",1:12],2,var))

question5

#calculer les variances inter et intra de la #classification Cr n1<-sum(x$Area == 'West')n1n2 < −sum(x$Area=='East') n2 n3<-sum(x$Area == 'North')n3n4 < −sum(x$Area=='South') n4

n<-35

var_intra<-(n1/n)*V1+(n2/n)*V2+(n3/n)*V3+(n4/n)*V4

var_inter<-varT-var_intra

question6

#Realiser une acp # on choisira une acp standard car les donnees sont sur differentes echelles #et nous souhaitons donner la meme importances a toutes les variables #independamment de leur variance .

#stardiser les donnees x_numeric<-x[,sapply(x,is.numeric)] #centre et reduit autotiquement les donnees (selectionner uniquement les colonnes numeriques)

#pour realoser l'acp

library(FactoMineR)

acp<-PCA(x_numeric,graph=FALSE)

#choix du nombre de dimentions a retenir qui sera en fonction des valeurs propres plot(acp$eig[,1], type="b", main="Scree plot des valeurs propres")

#type="b" va nous tracer a la fois les lignes et les points pour chaque #observation

#Affichage des indivus dans le plan principal

install.packages("factoextra")

library(factoextra)

fviz_pca_ind(acp,ncol.ind="cos2",gradient.cols=c("#00AFBB","#E7B800","red"),legend.title= "Cos2")

#Affichages du cercle des correlations

fviz_pca_var(acp,ncol.var="contib",gradient.cols=c("#00AFBB","blue","red"),legend.title="co ntibutions")

#interpretation des axes #Pour l'acp , les axes sont interpretes comme suit : ##Dim1(79%):cette premiere dimention 79% de la variance de nos donnees . #cela signifie que cette coposante principle capture une grande partie de #l'information contenuedans l'ensemble des donnees original. #Les variables comme les mois d'été (juin, juillet, aout) sont positivement #corrélées avec cette dimention , indiquant qu'elle contribuent #significativement a la variabilité captée par cette composante . #la longitude semble

également étre un facteur important sur cet axe. ##Dim2(16.6%):la 2e dimention explique 16.6% de la variance . #elle est perpendiculere a la 1ere et capture une partie differente de #l'information . la variable "Latitude" semble etre le plus fortement correles #avec cette dimention ,suggeran que les differances de latitude sont une #sources secondaire de la variabilite des donnees .

#L'interpretation dans le contexte de nos donnees pourrait etre : #dim1 pourrait representer un gradient thermique saisonnier , avec des #temperatures plus elevees pendant les mois d'ete . #dim2 pourrait representer un geoggraphique nord-sud , comme l'indique la #correlation avec la latitude. #Les variables proches du cercle exterieur sont bien representer par les #dimentions de l'acp , tandis que celle plus proche du centre ont une #correlation pls faible avec les dimentions et sont donc moins representees.

question7

#7. Appliquer l'algorithme de Classification Ascendante Hiérarchique (CAH) de #Ward aux projections des #points sur le plan principal. Combien de classes est-il raisonnable de retenir?

Calcul des distances euclidiennes

distances <- dist(x[, 1:12], method = "euclidean")

Appliquer la CAH avec la méthode de Ward

cah <- hclust(distances, method = "ward.D2")

Afficher le dendrogramme

plot(cah, hang = -1) # 'hang = -1' permet de suspendre les étiquettes sous le dendrogramme

rect.hclust(cah, k = 3, border = "red")

#selon le critere de la variance expliquee , on choisira l'axe dim1 ou on a plus #79% de la variance

########Question8######## #. On retient la classification à trois classes CCAH,3. #• Afficher le dendogramme en mettant en évidence les trois classes avec des #rectangles; #• trouver la classe d'affectation de chaque individu; #• afficher le nuage de points dans le plan principal avec les couleurs #correspondantes aux classes #d'affectations; #• calculer les variances inter et intra correspondantes à CCAH,3 du nuage de #points initial, et les #comparer avec les variances inter et intra de Cr (voir point 5). Discuter les #résultas obtenus.

CAH avec la méthode de Ward

distances <- dist(x[, 1:2], method = “euclidean”) cah <- hclust(distances, method = “ward.D2”)

Affichage du dendrogramme avec trois classes

plot(cah, hang = -1) rect.hclust(cah, k = 3, border = “red”)

c3<-cutree(tree=cah,k=3)

plot(c3,col=1:4,pch=16,main=“Nuage de points dans le plan principal avec CAH,4”)

Attribution des classes à chaque individu

clusters <- cutree(cah, k = 3)

Affichage des individus sur le plan principal avec des couleurs pour chaque cluster

library(FactoMineR)

res.pca<-PCA(x_numeric,scale.unit=TRUE,ncp=5,graph=FALSE)

fviz_pca_ind(res.pca, col.ind =as.factor(clusters), # Couleurs selon la classe d’appartenance palette = c(“#2E9FDF”, “#00AFBB”, “#E7B800”), addEllipses = TRUE, legend.title = “Clusters”)

Supposons que ‘x_numeric’ contient nos données numériques originales

et que ‘clusters’ est un vecteur contenant la classe de chaque observation.

Convertir clusters en facteurs pour le calcul de la somme des carrés

clusters_factor <- as.factor(clusters)

Calcul de la somme des carrés totale

sst <- sum((x_numeric - colMeans(x_numeric))^2)

Calcul de la somme des carrés intra-classe (variance intra-classe)

ssw <- sum(sapply(split(x_numeric, clusters),function(cluster) {sum((cluster - colMeans(cluster))^2)}))

Calcul de la somme des carrés inter-classe (variance inter-classe)

ssb <- (-sst + ssw)

Calcul des variances intra et inter-classe

var_intra <- ssw / (length(clusters) - length(unique(clusters))) var_inter <- ssb / (length(unique(clusters)) - 1)

Affichage des résultats

cat("Variance intra-classe :", var_intra, "") cat("Variance inter-classe :", var_inter, "")

#Ce code calcule d'abord la somme des carrés totale pour l'ensemble des données #(SST), puis la somme des carrés intra-classe (SSW), qui est la somme des carrés #des écarts par rapport à la moyenne de chaque cluster. Ensuite, la somme des #carrés inter-classe (SSB) est obtenue en soustrayant SSW de SST. Les variances #sont ensuite calculées en divisant ces sommes de carrés par les degrés de #liberté appropriés.

les var inter et intra de cette classification sont largement superieures a #celles obtenues a la classification du point 5.

########Question9###### #. Répondre aux questions du point 8 en prenant la classification en 4 #classes #CCAH,4 obtenue par CAH

1. Afficher le dendrogramme pour 4 classes

plot(cah, hang = -1) # 'hang = -1' pour que les étiquettes pendent du dendrogramme rect.hclust(cah, k = 4, border = "red") #Ajouter des rectangles pour 4 clusters

2. Trouver la classe d'affectation pour chaque individu pour 4 clusters

clusters_cah <- cutree(cah, k = 4)

3. Afficher les individus sur le plan principal avec les couleurs des clusters

clusters<-as.factor(clusters_cah)

fviz_pca_ind(res.pca, col.ind = clusters, # Couleurs en fonction des clusters palette = c("red", "green", "blue", "yellow"), addEllipses = TRUE, # Ajouter des ellipses si vous le souhaitez legend.title = "Clusters_cah")

4. Calcul des variances inter et intra pour la classification à 4 clusters

Calcul de la somme des carrés totale (SST)

sst_cah <- sum((res.pca*ind*coord - colMeans(res.pca*ind*coord))^2)

Calcul de la somme des carrés intra-cluster (SSW) pour 4 clusters

ssw_cah <- sum(sapply(split(res.pca*ind*coord, clusters_cah), function(cluster) { sum((cluster - mean(cluster))^2) }))

Calcul de la somme des carrés inter-cluster (SSB) pour 4 clusters

ssb_cah <- sst - ssw

Calcul de la variance intra-cluster

var_intra_cah <- ssw / (nrow(res.pca*ind*coord) - length(unique(clusters)))

Calcul de la variance inter-cluster

var_inter_cah <- ssb / (length(unique(clusters)) - 1)

Affichage des variances

print(paste("Variance intra-cluster pour 4 classes:", var_intra)) print(paste("Variance inter-cluster pour 4 classes:", var_inter))

Comparaison avec les variances inter et intra de Ccah,3

var inter et intra pour 4 classes sont plus petites que si on considere 3 #classes.

########Question10 #######. Appliquer l'algorithme des centres mobiles au nuage de points #initial (c'est à dire aux individus dans #l'espace défini par les 12 variables primaires) pour obtenir une classification #Ck-means,4 avec 4 centres. #• Obtenir la classe de chaque pays; #• calculer les variances inter et intra correspondantes à Ck-means,4 du nuage #des points initial; #• comparer avec les résultats précedants et discuter.

#La question 10 porte sur l'application de l'algorithme des centres mobiles, #également connu sous le nom de K-means, à notre jeu de données. Voici les étapes #à suivre pour répondre à cette question :

#1. **Appliquer l'algorithme des centres mobiles (K-means)** : # - Vous choisir le nombre de centres (clusters) basé sur la question
#précédente ou d'autres critères (comme la méthode du coude). # - Vous exécuter l'algorithme K-means sur les données pour classer les #individus dans les clusters.

#2. **Obtenir la classe de chaque pays** : # - L'algorithme K-means vous fournira une affectation de cluster pour chaque #individu (pays dans votre cas).

#3. **Calculer les variances inter et intra** : #calculerons la variance intra-cluster (à l'intérieur des clusters) et #la variance inter-cluster (entre les clusters) pour évaluer la qualité de la #classification.

#4. **Comparer avec les résultats précédents** : #- nous comparons les variances obtenues avec celles de la classification #ascendante hiérarchique pour discuter des différences et peut-être justifier #la sélection d'une méthode sur l'autre.

Supposons que 'x_numeric' est notre dataframe des données numériques comme

#precedemment.

Appliquer K-means

set.seed(123) # Pour la reproductibilité des résultats kmeans_results <- kmeans(x_numeric, centers = 4)

Obtenir la classe de chaque observation

clusters_kmeans <- kmeans_results$cluster

Calculer les variances inter et intra clusters

SST (somme des carrés totale) reste le même que pour CAH

SSW (somme des carrés intra-cluster) et SSB (somme des carrés inter-cluster) doivent être recalculés pour K-means

ssw_kmeans <- sum(sapply(split(x_numeric, clusters_kmeans), function(cluster) { sum((cluster - colMeans(cluster))^2) })) ssb_kmeans <- (-sst + ssw_kmeans) var_intra_kmeans <- ssw_kmeans / (nrow(x_numeric) - length(unique(clusters_kmeans))) var_inter_kmeans <- ssb_kmeans / (length(unique(clusters_kmeans)) - 1)

#Pour faire la comparaison entre les résultats de la Classification Ascendante #Hiérarchique (CAH) et ceux de l'algorithme des centres mobiles (K-means), nous #devrons comparer plusieurs aspects :

#1. **Affectation des clusters** : #- Voir si les mêmes observations sont regroupées ensemble par les deux #méthodes.

#2. **Cohésion et séparation des clusters** : #- Utiliser les variances intra-cluster (SSW) et inter-cluster (SSB) calculées #pour chaque méthode. Les clusters bien séparés auront une variance #inter-cluster élevée et une variance intra-cluster faible.

#3. **Mesures statistiques** : #- Calculer des mesures comme l'indice de silhouette ou le coefficient de #Calinski-Harabasz pour les deux méthodes. Des valeurs plus élevées indiquent #généralement une meilleure performance de clustering.

#4. **Visualisation** : #- Comparer visuellement les clusters sur le plan principal des composantes de #l'ACP pour les deux méthodes.

#####Voici un exemple de code R pour comparer les variances :####

Supposons que nous avons déjà calculé SSW et SSB pour CAH et K-means

Calculer les variances pour CAH

var_intra_cah <- ssw_cah / (nrow(x_numeric) - length(unique(clusters_cah))) var_inter_cah <- ssb_cah / (length(unique(clusters_cah)) - 1)

Calculer les variances pour K-means

var_intra_kmeans <- ssw_kmeans / (nrow(x_numeric) - length(unique(clusters_kmeans))) var_inter_kmeans <- ssb_kmeans / (length(unique(clusters_kmeans)) - 1)

Comparer les variances

cat("CAH Variance intra-cluster:", var_intra_cah, "") cat("CAH Variance inter-cluster:", var_inter_cah, "") cat("K-means Variance intra-cluster:", var_intra_kmeans, "") cat("K-means Variance inter-cluster:", var_inter_kmeans, "")

Comparer visuellement les clusters

Pour CAH

fviz_cluster(list(data = x_numeric, cluster = clusters_cah), geom = "point")

Pour K-means

fviz_cluster(list(data = x_numeric, cluster = clusters_kmeans), geom = "point")

#Dans ce code, `ssw_cah` et `ssb_cah` sont les somme des carrés intra et inter #clusters pour CAH, et `ssw_kmeans` et `ssb_kmeans` pour K-means. `clusters_cah` #et `clusters_kmeans` sont les vecteurs des affectations de clusters pour chaque #méthode, respectivement.

#La visualisation peut être réalisée en utilisant la fonction `fviz_cluster` du #package `factoextra`, qui fournit une représentation graphique des clusters #basée sur les données de l'ACP ou sur les centres de clusters et leurs #affectations.

#La discussion de ces comparaisons devrait se concentrer sur les différences dans #la structure des clusters révélées par chaque méthode et sur la manière dont ces #différences peuvent être interprétées en fonction de votre connaissance du #domaine et des objectifs de l'étude.

install.packages('tinytex') tinytex::install_tinytex()