

Classification : CAH

Yanice

11 décembre 2016

Contents

Classifications automatique : CAH	1
I. Distance entre individu - matrice de dissimilarité initiale	2
II. Classification hiérarchique	2
II.1. Méthode de regroupement - Critère	2
II.2. Tracé des dendogrammes	3
III. Critère pour l'aide à la décision du nombre de classe	3
III.1. Critère de Coude	3
III.2. Coefficient de corrélation multiple	4
III.3. Pseudo F	6

Classifications automatique : CAH

Remarque : méthode ACP, AFC, etc... : **méthode d'ordination**. ACP : L'étude des individus et études des variables et liaisons des infos entre elles ces méthodes sont essentiellement basées pour variables et/ou individus. entre individus on parle de **typologie** (création de carte factorielle), entre variable on parle de **corrélation**.

On a face à ses méthodes, **les méthodes de classifications** : Elles ne partent pas d'un tableau individus variables, mais individus individus, définis par les distances entre individus. L'objectif est de crée une variable qualitative qui permette de créer des **dendogrammes** (des représentations du types arbres (voir page 4)).

Attention à ne pas confondre avec : **l'Analyse en Coordonnées principales (PCO)**; Cette analyse part d'un tableau individus individus et avec on représente une carte factorielle, je me sert de ce que je vois pour essayer de définir des groupes (qui n'existent pas en soit).

- Pour les méthodes d'ordinnations, on peut avoir une variable qualitative et on s'en sert pour essayer de voir si on peut s'ens servir pour interpréter la typologie des individus.
- En classification, on cherche à créer une variable qualitative, je fais une classification, j'obtiens un dendrogramme et je le coupe à un niveau ou ça a un sens pour moi : **la valuation : h** et je définie une variable qualitative à postériori (c'est à dire que je ne la connais pas).

Classification : découpage a postériori, méthode d'ordination, variable qualitative qui nous illustre la typologie entre les individus.



I. Distance entre individu - matrice de dissimilarité initiale

```

objet=#data.frame
##-----Centrage et reduction -----
#Pour éviter que les variables a fortes variance pesent sur le résultats:
objet=scale(objet,center=T,scale=T)

##-----Distance-----
#Distance euclidienne
distance=dist(objet)
#Distance entre variable dichotomique
distance=dist.binary(objet)
#Distance entre variables quantitatives
distance=dist.quant(objet)
#Distance entre vecteur de proportion avec somme des lignes qui vaut 1
distance=dist.prop(objet)

#Matrice de distance
MatriceDistance=as.matrix(ditance)

```

II. Classification hiérarchique

Chaque procédé qui définit M : le critère choisis pour le regroupement et h la fonction de valuation donne une classification hiérarchique particulière.

Procédé :

- 1) Partant de la matrice de dissimilarité on regroupe les deux valeurs **minimales** toujours en premier.
- 2) On recalcule la matrice de distance à l'aide du critère de regroupement choisis
- 3) et on refait le processus

II.1. Méthode de regroupement - Critère

```

##----- Les critères -----

#Lien simple = d(a,b)=min
LienSimple=hclust(distance,"single")#classification avec le min
#Lien complet = max
LienComplet=hclust(distance,method="complete")
#Lien moyen
LienMoyen=hclust(distance,method="average")
#Critère de Ward : meilleur critère avec carré des distance
LienWard=hclust(distance,method="ward.D2")

##-----Les Sorties-----
##
Lien=LienSimple

```

```

#Etape de regroupement :
Lien$merge
  #En ligne : étape i
  # le signe - : pour dire que c'est un singleton
  #Signe +: regroupement de classe (groupe)

#Fonction de valuation h():
Lien$height
  #Longueur des branches

#Ordre d'apparition des feuilles
Lien$order

#Retrouver la méthode
Lien$method

#Retrouver la distance utilisée
Lien$dist.method

##-----Affichage dendrogramme-----
plot(Lien)

##-----Recherche d'une partition -----
#matérialisation des groupes
rect.hclust(Lien,k)#k:nombre de groupe
#Découpage
NouvGroupe=cutree(Lien,k=) #nombre de groupe ou h=hauteur de coupe
# Donne la liste des éléments et leur groupe

#liste des groupes
print(sort(NouvGroup))

```

II.2. Tracé des dendogrammes

```

##-----Tracé du dendrogramme avec un trait la où on souhaite couper-----
plot(Lien,main="ligne1 \n ligne 2")
abline(h=#hauteur souhaitée,col="red",lwd=1.5)

##-----Coupage-----
#Découpage
NouvGroupe=cutree(Lien,k=) #nombre de groupe (voir critère plus bas)
#liste des groupes
print(sort(NouvGroup))

```

III. Critère pour l'aide à la décision du nombre de classe

III.1. Critère de Coude

On applique le critère du coude aux hauteurs du dendrogramme **décroissantes**.

```
##-----hauteurs des dendrogramme par ordre décroissant-----
HauteurDecroissantH=rev(Lien$height)

##-----Tracé des hauteurs successives-----
plot(HauteurDecroissantH,type="l",main="hauteur du dendrogramme décroissantes",ylab="hexo2$height",xlab="")

##-----Rajout des points à chaque coude-----
points(1:length(HauteurDecroissantH),HauteurDecroissantH)

##-----Détermination des coudes -----
diff(HauteurDecroissantH,lag=1,differences = 2)#différences=2 cest comme ca
```

Les points de coudes sont les Points ou ca présente du sens de couper mais ça ne dit pas qu'il y en a un meilleur que l'autre. Couper avec deux groupes ce n'est pas le meilleur choix. L'objectif c'est de réussir à avoir des groupes à caractériser. On peut couper avant chaque changement de signe.

III.2. Coefficient de corrélation multiple

Il est fréquent de calculer le ratio de l'inertie interclasse sur l'inertie totale pour juger de la qualité d'une partition. **Cette grandeur devrait être idéalement proche de 1.**

Dans le cas d'une hiérarchie découlant d'un dendrogramme, il est possible de calculer le R2 pour chaque partition et de s'aider du graphique de ces valeurs pour faire un choix équilibré entre la part d'inertie expliquée par les classes et un petit nombre de classes.

Mieux vaut essayer de calculer l'inertie des centres de classes pondérés.

calculer les valeurs de R2 pour les différentes partitions issues du découpage de l'objet :

R2: Proportion de la variance expliquée par les classes.

R2 semi partiel ?

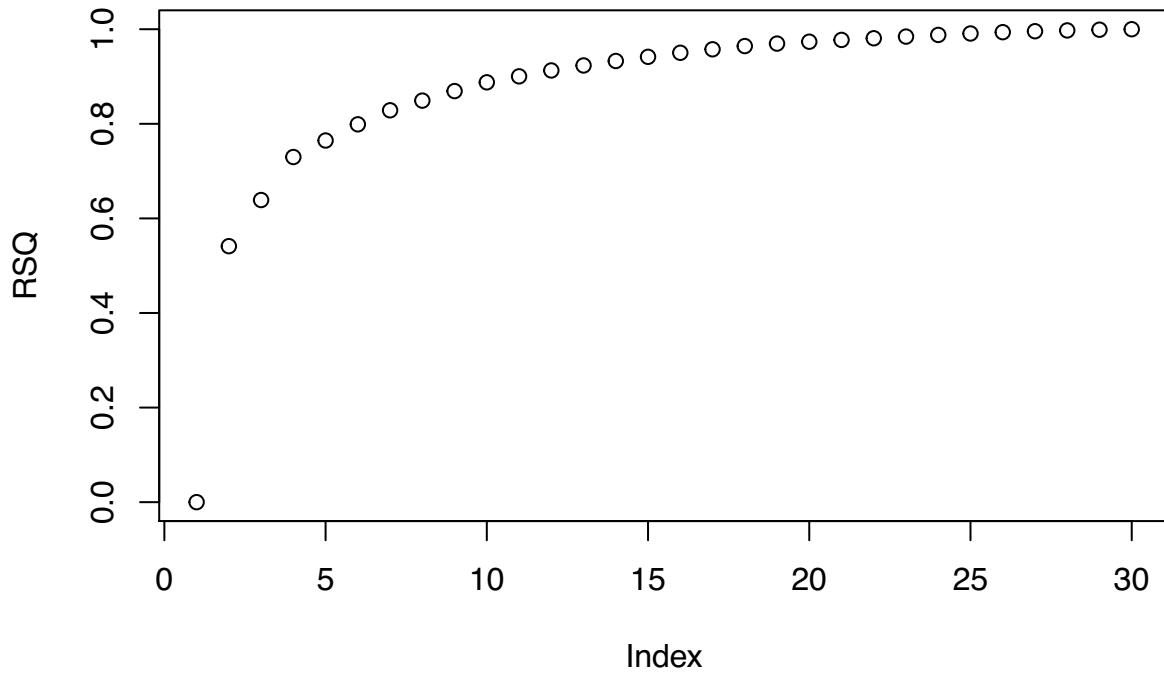
```
library(ade4)
data(doubs)
objet=doubs$fish
Distobjet=dist(objet)
Lien=hclust(Distobjet,'ward.D2')#Contient les hauteurs qui nous intéressent

##-----Calcul du R2 pour les différentes partitions du découpage-----

RSQ<-rep(0,nrow(objet))#objet : élément sur lequel on applique les distances etc

sum(scale(objet,scale=FALSE)^2)->SQTot
for (i in 1:nrow(objet)) {
  Cla<-as.factor(cutree(Lien,i))
  sum(t((t(sapply(1:ncol(objet), function(i)tapply(objet[,i],Cla,mean))-apply(objet,2,mean))^2) * as.vector(table(Cla)))/SQTot->RSQ[i]
}

plot(RSQ)
```

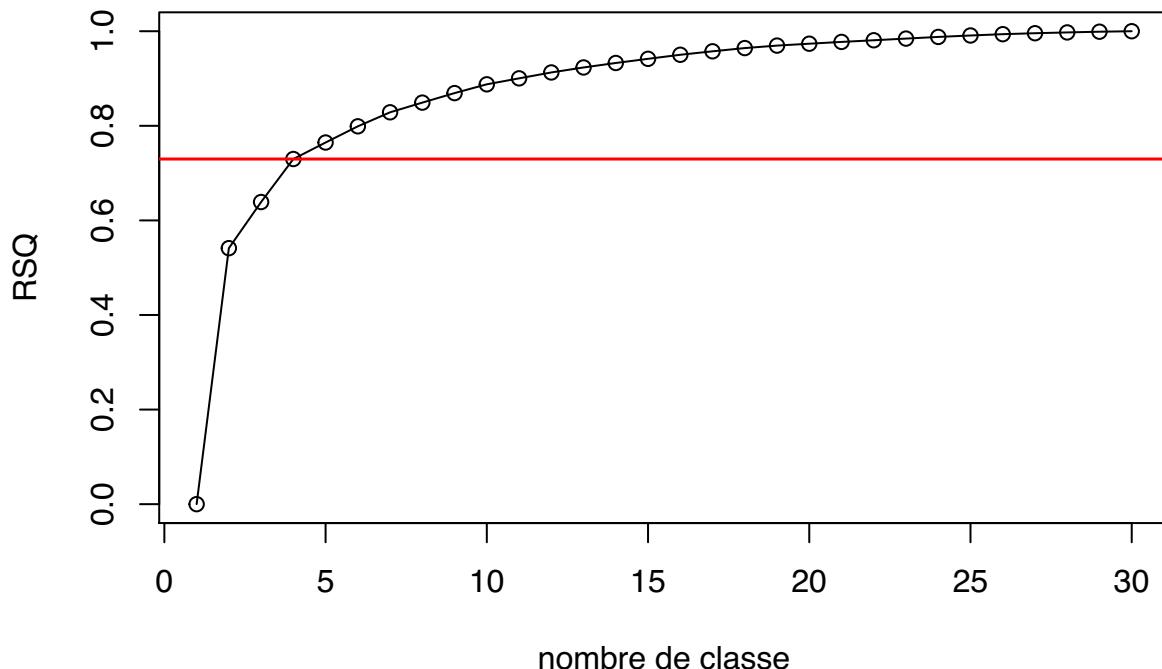


```
#RSQ=km$betweenss/km$totss
#R2 semi partiel

##-----Graphique du RSQ -----
plot(sort(RSQ),type='l',xlab="nombre de classe",ylab="RSQ",main="Valeurs de R2 pour différentes partitions")
points(sort(RSQ))

#Voilà la valeur du RSQ que l'on aurait avec seulement 4 classes différentes
abline(h=RSQ[4], col="red", lwd=1.5)
```

Valeurs de R2 pour différentes partitions



Ce code calcule, pour chaque partition, le centre de gravité de chaque classe, le centre de gravité global et les utilise pour calculer l'inertie interclasse, ramenée à l'inertie totale. la fonction tapply est utilisée pour calculer les centres de gravité des classes par variable et la fonction sapply permet de faire ce calcul sur l'ensemble des variables.

variance inter : R2 variance intra : 1-R2

III.3. Pseudo F

Il sert à mesurer la séparation des classes.
k : nombre de classes de la partition considérée et n nombre d'objet de l'ensemble E.

C'est la mesure de la séparation entre toutes les classes.

$$pseudoF = \frac{\frac{R^2}{k-1}}{\frac{1-R^2}{n-k}}$$

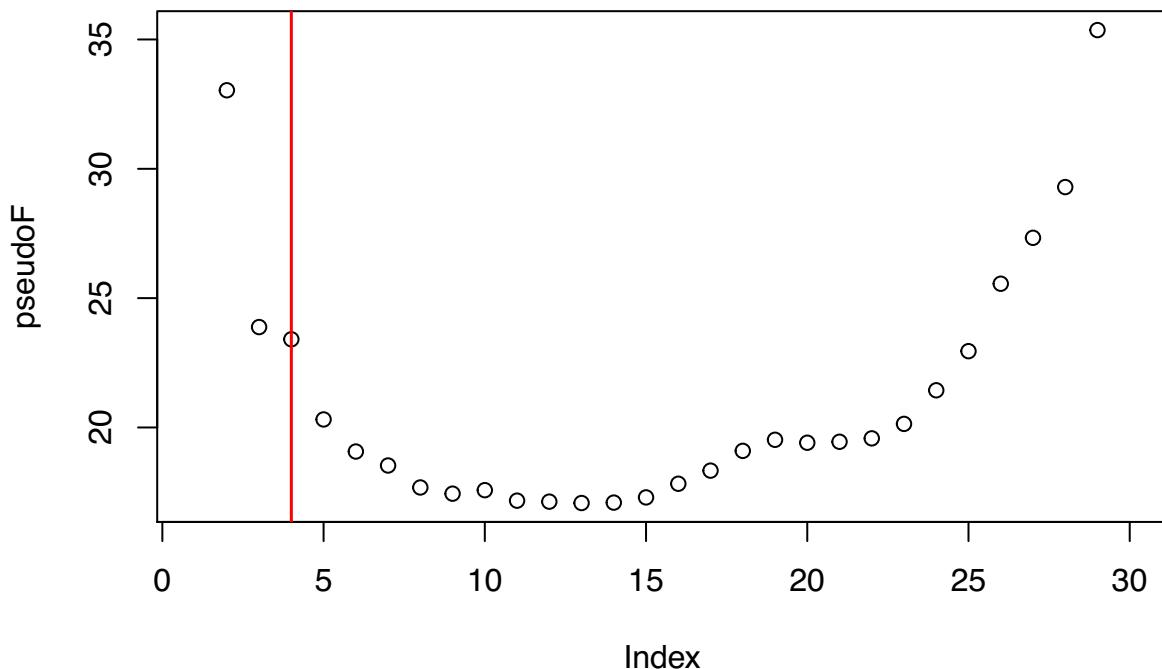
```
pseudoF<-rep(0,nrow(objet))
pseudoF=c()

for (i in 1:nrow(objet)) {
#      Cla<-as.factor(cutree(Lien,i))#choix d'avoir i groupe , chaque élément est identifié a son groupe
  pseudoF[i]=(RSQ[i]/(i-1))/((1-RSQ[i])/(nrow(objet)-i))
}
```

```
plot(pseudoF)

k <- 4 #Nous gardons 4 classes
n <- 30 #Nous avons 30 individus

#Voila la valeur du F_ratio que l'on aurait seulement avec 4 classes différentes
abline(v=4, col = "red", lwd = 1.5)
```



On cherche une grande valeur suivi d'une décroissance ##