Traitement numérique des données

Mohamed Nadif

LIPADE, UFR MI, Université Paris Descartes, France

Outline

Plan du cours de la Partie 1

Partie 1

- Cours 1 : Types de données et statistique descriptives univariées
- Cours 2 : Visualisation des données quantitatives et qualitatives
- Cours 3 : Statistique descriptive bivariée et visualisation
- Cours 4 : Données mixtes
- Cours 5 : Premiers modèles et algorithmes
- Cours 6 : Evaluation 1

Intérêts

- Analyse des données
- Business Analytics
- Importance of Softwares
- Utilisation du logiciel **R**

Outline

Installation

- Télécharger une version à jour de R sur le site http://www.r-project.org/
- Le fichier d'installation est spécifique aux systèmes d'exploitation Windows, Unix, Linux, Macintosh.
- Pour windows, il suffit de double-cliquer sur le fichier téléchargé afin de lancer l'installation du logiciel.
- A l'issue de l'installation, le logiciel R peut donc être démarré en double-cliquant sur l'icône R.
- R studio

Environnement

- Pour lancer R sous windows, il suffit de double-cliquer sur l'icône R.
- Pour quitter R :
  - Utiliser la commande q(). Dans le menu, faire Fichier puis Sortir
  - L'environnement de travail est alors sauvegardé dans le fichier .Rdata
- Aide en ligne
  - Dans le menu, faire Aide puis Aide HTML
  - Taper help.start()
- Aide sur une commande : taper help(commande) ou ?commande

Langage

- R est un langage orienté objet
- Chaque objet est caractérisé par un nom, une classe et des attributs.
- Exemple de classes d'objets : vecteur, matrice, tableau, liste, etc. ...
- L'utilisateur peut effectuer des actions sur ces objets via des fonctions.

Opérations élémentaires

```
> 10 + 2 + 3 # addition (renvoie 15)
> 3 * 4 + 2.5 # addition et multiplication (renvoie 14.5)
> 1 + 3/2 # addition et division (renvoie 2.5)
> (1 + 3)/2 # renvoie 2
> 4^2 # puissance (renvoie 16)
```

Quelques fonctions usuelles

log(), sqrt(), abs(), sign(), exp(), sin(), asin(), cos(), acos(), tan(), atan()

Assignation

- On peut assigner des valeurs à des objets en utilisant les opérateurs = ou <-
- Exemple : assignons la valeur 5 à la variable x
  `> x = 5`
- on peut maintenant effectuer des calculs et définir d'autres variables à partir de la variable x
  `> y=sqrt(x + 4)`

Types de vecteurs

- Vecteur de valeurs numériques
  > c(2, 3, 5, 2, 7, 1)
  2 3 5 2 7 1
- Vecteur de valeurs logiques
  > c(T, F, F, F, T, T, F)
  TRUE FALSE FALSE FALSE TRUE TRUE FALSE
- Vecteur de chaînes de caractères > c("Bruxelles", "Paris", "Canberra", "Sydney")
  "Bruxelles" "Paris" "Canberra" "Sydney"
- On peut affecter un vecteur à une variable
  > x <- c(2,4,6,8)
  > x
  2 4 6 8
- Les fonctions usuelles appliquées à un vecteur s'appliquent à chaque élément de ce vecteur
  > x+2
  4 6 8 10
  > log(x)
  0.6931472 1.3862944 1.7917595 2.0794415

Vecteurs

- Les opérations usuelles appliquées à deux vecteurs de même taille sont effectuées élément par élément
  ```
  > y<-c(1,2,3,4)
  > x+y
  3 6 9 12
  > x*y
  2 8 18 32
  ```
- Produit scalaire de deux vecteurs
  ```
  > x %*% y
  60
  ```
- Séquence de nombres
  ```
  > x <- seq(10,34,by=1)
  > x
  10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
  > x=10:34
  > x
  10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
  ```

Création de vecteurs

- Dupliquer des nombres
  > rep(3,4)
  3 3 3 3
- Exercice : Taper les commandes suivantes et observer les résultats
  > seq(1,10,2)
  > seq(5,1)
  > seq(5,1,-2)
  > x<-c(1,2,3)
  > rep(x,3)
  > y<-x
  > rep(x,y)

Manipulation de vecteurs

- Extraction d'élément

```
> x<-1:5
> x
1 2 3 4 5
> x[3] # 3ème élément du vecteur x
3
> x[-3] # x privé de son 3ème élément
1 2 4 5
```

- Remplacement d'élément

```
> x[3] <- 10
> x
1 2 10 4 5
```

- Ajout d'élément

```
> x[6] <- 6
> x
1 2 10 4 5 6
> x <- c(x,7)
> x
1 2 3 10 4 5 6 7
```

Manipulation de vecteurs

- Utilisation d'opérateurs relationnels <, <=, >, >=, ==, !=
  > x <- c(3, 11, 8, 15, 12)
  > x > 8
  FALSE TRUE FALSE TRUE TRUE
  > x != 8
  TRUE TRUE FALSE TRUE TRUE
- Extraction d'éléments
  > x <- c(3, 11, 8, 15, 12)
  > x[c(2,4)]
  11 15
- Extraction d'éléments à l'aide d'opérateurs relationnels
  > x[x > 10]
  11 15 12
- Exemple de fonctions de vecteurs
  > length(x) # Nombre d'éléments du vecteur x
  > max(x) # Plus grande valeur de x
  > min(x) # Plus petite valeur de x
  > sum(x) # Somme des éléments de x
  > prod(x) # Produit des éléments de x
  > mean(x) # Moyenne des éléments de x
  > sd(x) # Ecart type des éléments de x

Manipulation de vecteurs

- Un facteur est un vecteur permettant la manipulation de données qualitatives. La longueur est donnée par length, le mode par mode et les modalités par levels. Les facteurs forment une classe d'objets et bénéficient de traitements particuliers pour certaines fonctions

```
> sexe=factor(c("M","F","F","M"))
> sexe
M F F M
levels: F M
> sexe=factor(c("M","F","F","M"),labels=c("Femme","Homme"))
> sexe
Homme Femme Femme Homme
Levels: Femme Homme
```

Création de Matrices

- Méthodes
  - Exemple 1
    > x <- matrix(1:6,2,3)

$$\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$

  - Exemple 2
    > x <- matrix(c(10,20,30,40,50,60),2,3)

$$\begin{pmatrix} 10 & 30 & 50 \\ 20 & 40 & 60 \end{pmatrix}$$

  - Exemple 3
    > x<- seq(1:9)
    > dim(x)<-c(3,3)

$$\begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix}$$

  - Agrégations de vecteurs par colonnes
    > cbind(1:4,5:8,9:12) # que ferait rbind(1:4,5:8,9:12) ?

$$\begin{pmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{pmatrix}$$

Tableaux de données

- Contrairement à une matrice, un tableau de données peut contenir plusieurs types de données (numériques, caractères, logiques, ...)
- Les vecteurs inclus dans le tableau doivent être de même longueur ; si un de ces segments est plus court il est alors recyclé un nombre entier de fois.
- Création d'un tableau de données :

```
> etudiants <- data.frame(nom=c("Alfred","Paul","Isabelle","Mathieu"),
+ age=c(21,26,23,20),sexe = c(rep("M",2),"F","M"))
> etudiants
nom age sexe
1 Alfred 21 M
2 Paul 26 M
3 Isabelle 23 F
4 Mathieu 20 M
```

- Autre manière de créer un tableau de données :

```
> nom <- c("Alfred","Paul","Isabelle","Mathieu")
> age <- c(21,26,23,20)
> sexe = c(rep("M",2),"F","M" )
> etudiants <- data.frame(nom,age,sexe)
```

Indexation des éléments d'un tableau de données

- Données
  > etudiants
  nom age sexe
  1 Alfred 21 M
  2 Paul 26 M
  3 Isabelle 23 F
  4 Mathieu 20 M
- Exemple 1
  > etudiants[2,]
  nom age sexe
  2 Paul 26 M
- Exemple 2
  > etudiants[etudiants$age>21,]
  nom age sexe
  2 Paul 26 M
  3 Isabelle 23 F

Importation des donnés

- Définir un répertoire de travail dans R
  > setwd("c:/mon repertoire R")
  > getwd()
  "c:/mon repertoire R"
- Importation des données dans R afin d'effectuer des calculs statistiques à partir de ces données. Ceci peut être effectué à l'aide des commandes suivantes :
  > data = read.table("h:/data.dat")
  > data = read.table("h:/data.dat", header=T) # avec nom de colonnes
- Pour prendre en compte les caractères de séparation des variables, on peut utiliser les commandes :
  > data = read.table("h:/data.dat", header=T, sep=",")
  > data = read.table("h:/data.dat", header=T, sep=" ")
- Exportation des résultats (row.names et col.names sont par défaut)
  > write.table(Resultat,"Res.csv",sep=";",row.names=TRUE,col.names=TRUE)
- write.csv est une laternative dans ce cas à write.table

Exemple de données

- Charger et visualiser les données iris
  > data(iris)
  > iris
- Classe des objets
  > class(iris)
  "data.frame"
  > class(iris$Species)
  [1] "factor"
  > class(iris$Sepal.Length)
  "numeric"
- Les variables présentes
  > names(iris)
  "Sepal.Length" "Sepal.Width" "Petal.Length" "Petal.Width" "Species"
- Modalités de la variable qualitative "Species"
  > levels(iris$Species)
  "setosa" "versicolor" "virginica"
- statistiques descriptives pour l'ensemble des variables
  > summary(iris)

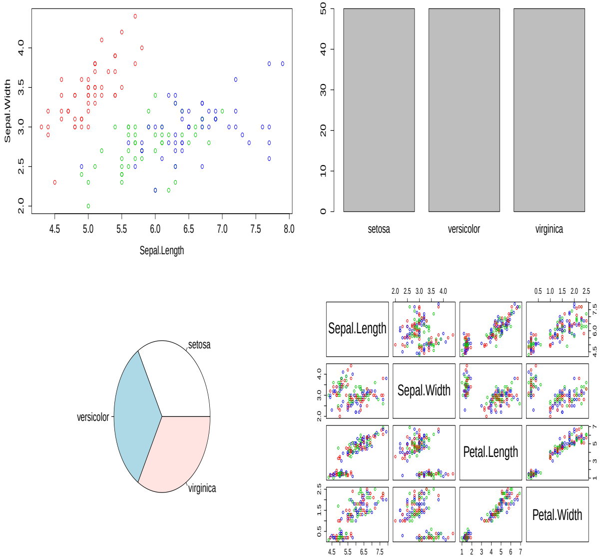
Quelques statistiques pour certaines variables

- Moyenne
  > mean(iris$Sepal.Length)
  5.843333
- Médiane
  > median(iris$Sepal.Length)
  5.8
- Maximum
  > max(iris$Sepal.Length) # max
  7.9
- Minimum
  > min(iris$Sepal.Length) # min
  4.3
- Ecart-type
  > sd(iris$Sepal.Length) # ecart-type
  0.8280661
- Covariance
  > cov(iris$Sepal.Length,iris$Petal.Length) # covariance
  1.274315
- Fréquence absolue
  > table(iris$Species) # nombre d'occurrences des modalités

Quelques statistiques et graphiques pour certaines variables

- Graphique (2 écritures équivalentes)
  > plot(iris[,1:2])
  > plot iris[c(1,2)]
- Même graphique mais en reportant les espèces
  > plot(iris[c(1,2)],col=c("red", "green3", "blue")[iris$Species])
- Diagramme en barre d'une variable qualitative (bâton)
  > barplot(table(iris[,"Species"]))
- Digramme circulaire
  > pie(table(iris[,"Species"]))
- Histogramme
  > hist(iris$Sepal.Length)
- Corrélation entre les variables
  > pairs(iris[1:4])
  > pairs(iris[1:4],class=iris$Species,col=c("red", "green3", "blue"))

Résultats


Sepal.Width
Sepal.Length
2.0 2.5 3.0 3.5 4.0
4.5 5.0 5.5 6.0 6.5 7.0 7.5 8.0
0 10 20 30 40 50
setosa
versicolor
virginica
setosa
versicolor
virginica
Sepal.Length
Sepal.Width
Petal.Length
Petal.Width

Mesures on 23 papillons

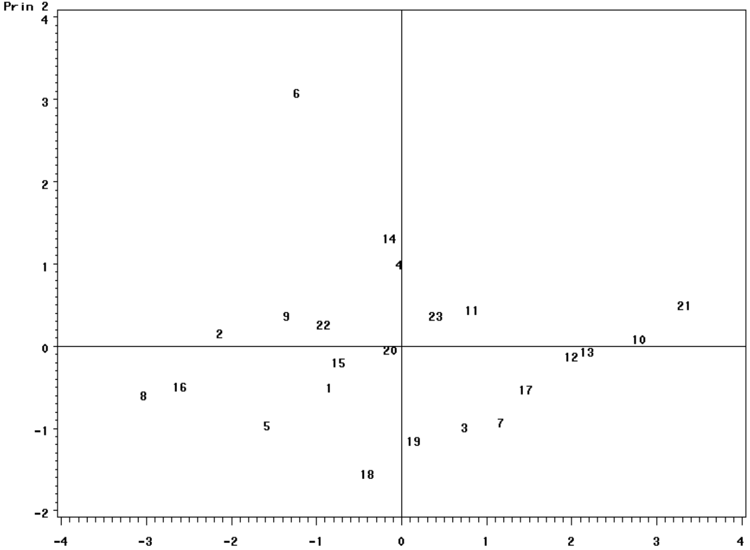
num	Z1	Z2	Z3	Z4
1	22	35	24	19
2	24	31	21	22
3	27	36	25	15
4	27	36	24	23
5	21	33	23	18
6	26	35	23	32
7	27	37	26	15
8	22	30	19	20
9	25	33	22	22
10	30	41	28	17
11	24	39	27	21
12	29	39	27	17
13	29	40	27	17
14	28	36	23	24
15	22	36	24	20
16	23	30	20	20
17	28	38	26	16
18	25	34	23	14
19	26	35	24	15
20	23	37	25	20
21	31	42	29	18
22	26	34	22	21
23	24	38	26	21

Mesures on 23 Butterflies

- Problèmes de visualisation et de classification
- Dimension 4
- Analyse Exlporatoire

Voir plus loin

ACP sur les papillons



Médiane

La liste des N données est rangée par ordre croissant

- Si N est impair ($N = 2n + 1$) la médiane est la donnée de rang $n + 1$
- Si N est pair ($N = 2n$) la médiane est la demi somme des données de rang n et de rang $n + 1$

Quartiles

- Le premier quartile Q1 est la plus petite donnée de la liste telle qu'au moins un quart des données de la liste sont inférieures ou égales à Q1.
- Le troisième quartile Q3 est la plus petite donnée de la liste telle qu'au moins les trois quarts des données de la liste sont inférieures ou égales à Q3.

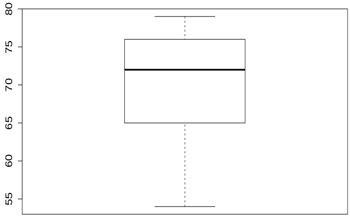
Exercice

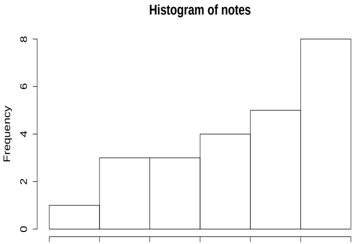
- On a relevé les notes de 24 élèves d'une classe lors d'un examen noté sur 100 points
  78 79 77 59 57 65 65 67 68 67 59 54 64 68 72 74 72 72 76 77 76 74 77 76
  1. Déterminer la médiane et les quartiles de cette série
  2. Dessiner la boite à moustache de cette série
  3. On peut comparer les résultats de cette classe avec les résultats d'une autre classe dont on sait que la note minimale est 47 , la note maximale est 85 , la médiane est 70, Q1 est 67 et Q3 est 76. Tracer sur le même graphique que dans la question 2 la boite à moustaches de cette nouvelle série.
  4. Que peut-on dire sur les différences entre les deux classes ?

Code R

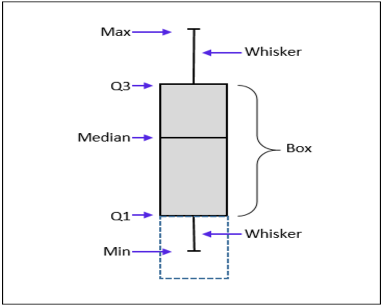
```
> notes=c(78,79,77,59,57,65,65,67,68,67,59,54,64,68,72,74,72,72,76,77,76,74,77,76)
> sort(notes)
> median(notes)
> mean(notes)
> sd(notes)
> quantile(notes, c(0.25, 0.5, 0.75))
> summary(notes)
> boxplot(notes)
> stem(notes)
> hist(notes)
```

Distribution des notes





Rappel



Délimitations des moustaches et Outliers

- L'extrémité de la moucstache inférieure est la valeur minimum dans les données qui est supérieure à la valeur frontière basse : Q1-1,5*(Q3-Q1)
- L'extrémité de la moustache supérieure est la valeur maximum dans les données qui est inférieure à la valeur frontière haute ; Q1+1,5*(Q3-Q1)
- Justification du choix de 1.5: Si une variable suit une distribution normale alors la zone délimitée par la boite et les moustaches devrait contenir 99.3% des observations. (1 implique 95.7% et 2 implique 99.9%)

Outline

Exercice 1

Description des données

Les données sur les naissances de 2006 sont constitués de 427 323 observations et 13 variables.

```
#Charger les données
> data(births2006.smpl)
> colnames(births2006.smpl)
"DOB_MM" "DOB_WK" "MAGER" "TBO_REC" "WTGAIN" "SEX" "APGAR5" "DMEDUC" "UPREVIS" "ESTGEST"
"DMETH_REC" "DPLURAL" "DBWT"
```

Variables	Description
DOB_MM	le mois de 1 à 12
DOB_WK	le jour de la semaine de 1 à 7
MAGER	
TBO_REC	
WTGAIN	le poids pris par la mère pendant la grossesse, NA indique une données manquante
SEX	le sexe du bébé "F" ou "M"
APGAR5	Score de 1 à 5 enregistré à la naissance
DMEDUC	Niveau d'instruction
UPREVIS	
ESTGEST	estimation en semaines de la grossesse, 99 indique une donnée manquante
DMETH_REC	naissance sans césarienne ou inconnue
DPLURAL	naissance unique ou plurielle
DBWT	le poids à la naissance

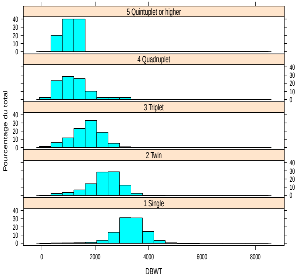
Code R

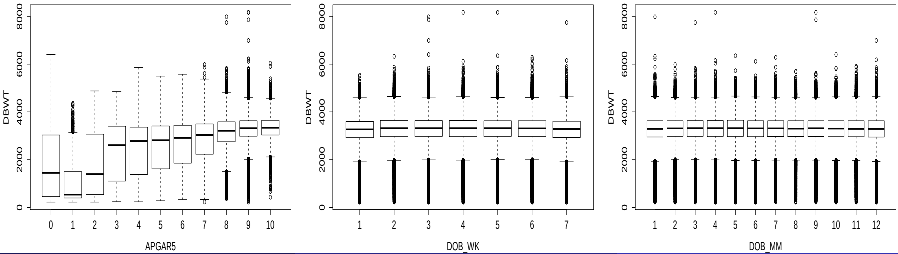
```
#Charger les packages utiles
> library(lattice)
> library(nutshell)
#Charger les données
> data(births2006.smpl)
#Taille des données
> dim(births2006.smpl)
#Affichage des 5 premières lignes
births2006.smpl[1:5,]
#Répartition par jour de la semaine
> repartition_jour=table(births2006.smpl$DOB_WK)
> repartition_jour
> sum(births_repartition)
#Répartition par mois de l'anné
> repartition_mois=table(births2006.smpl$DOB_MM)
> repartition_mois
> sum(repartition_mois)
#Diagramme en bâtons
> barchart(repartition_jour,ylab="Jour de la semaine",col="blue")
> barchart(repartition_mois,ylab="Mois de l'année ",col="green")
#Croisement de deux variables
> jour_sexe=table(births2006.smpl$DOB_MM,births2006.smpl$SEX)
> jour_sexe
```

Code R

```
#Visualisation simultanée
> barchart(jour_sexe,ylab="mois de l'année")
#Visualisation séparée
> barchart(jour_sexe,xlab="mois de l'année",groups=FALSE,horizontal=FALSE,col="black")
#Analyse de DBWT en fonction de DPLURAL
> summary(births2006.smpl$DPLURAL)
> histogram(~DBWT|DPLURAL,data=births2006.smpl,layout=c(1,5))
#Analyse de DBWT en fonction de APGAR5
> boxplot(DBWT~APGAR5,data=births2006.smpl,ylab="DBWT",xlab="APGAR5")
#Analyse de DBWT en fonction du jour de la semaine
> boxplot(DBWT~DOB_WK,data=births2006.smpl,ylab="DBWT",xlab="DOB_WK")
#Analyse de DBWT en fonction du mois de l'année
> boxplot(DBWT~DOB_MM,data=births2006.smpl,ylab="DBWT",xlab="DOB_WK")
#Moyenne par groupe sans tenant compte des données manquantes
> fac=births2006.smpl$DPLURAL
> res=births2006.smpl$DBWT
> t4=tapply(res,fac,mean,na.rm=TRUE)
> t4
#Visualisation
> barplot(t4,ylab="DBWT")
#Moyenne par groupe DPLURAL et SEX
> t5=tapply(res,INDEX=list(fac,births2006.smpl$SEX),mean,na.rm=TRUE)
> t5
#Visualisation
> barchart(t5,ylab="DBWT")
> barplot(t5,beside=TRUE,ylab="DBWT")
```

Résultats


5 Quintuplet or higher
4 Quadruplet
3 Triplet
2 Twin
1 Single
Pourcentage du total
DBWT


DBWT
APGAR5
DOB_WK
DOB_MM

Exercice 2

Description des données

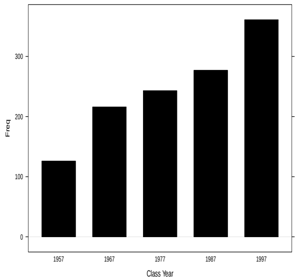
Le fichier contributions.csv contient les contributions reçues par un collège privé dans le Midwest. Le collège dispose d'un grand fond de dotations et, comme tous les collèges privés font, il tient des registres détaillés sur les dons d'anciens élèves. Ces contributions concernent 1230 anciens élèves et pour les années de 2000 à 2004.

Variables	Description
Gender	F ou M
Class.Yaer	1957, 1967, ..., 1997
Marital.Status	Etat civil
Major	Majeur dans une matière
Next Degree	études supérieures après le collège
FY04Giving	donation en 2004
FY03Giving	donation en 2003
FY02Giving	donation en 2002
FY01Giving	donation en 2001
FY00Giving	donation en 2000
AttendenceEvent	participation à l'événement

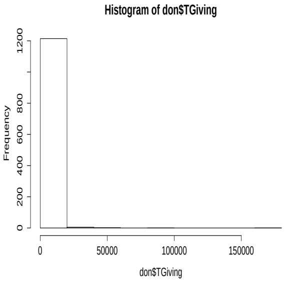
Code R

```
# Install packages
> library(lattice)
> don <- read.csv("C:/DataMining/Data/contribution.csv")
> don[1:5,]
> table(don$Class.Year)
> barchart(table(don$Class.Year),horizontal=FALSE,xlab="Class Year",col="black")
> don$TGiving=don$FY00Giving+don$FY01Giving+don$FY02Giving+don$FY03Giving+don$FY04Giving
# Attention aux données manquantes
> mean(don$TGiving,na.rm=TRUE)
> sd(don$TGiving,na.rm=TRUE)
> quantile(don$TGiving,probs=seq(0,1,0.05),na.rm=TRUE)
> quantile(don$TGiving,probs=seq(0.95,1,0.01),na.rm=TRUE)
> hist(don$TGiving)
> hist(don$TGiving[don$TGiving>0 & don$TGiving<=1000])
> boxplot(don$TGiving,horizontal=TRUE,xlab="Total Contribution")
> boxplot(don$TGiving,outline=FALSE,horizontal=TRUE,xlab="Total Contribution")
> ddd=don[don$TGiving>=30000,]
> ddd
> ddd1=ddd[,c(1:5,12)]
> ddd1
> ddd1[order(ddd1$TGiving,decreasing=TRUE),]
> boxplot(TGiving ~ Class.Year,data=don,outline=FALSE)
> boxplot(TGiving ~ Gender,data=don,outline=FALSE)
> boxplot(TGiving ~ Marital.Status,data=don,outline=FALSE)
> boxplot(TGiving ~ AttendenceEvent,data=don,outline=FALSE)
```

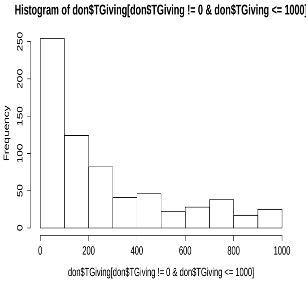
Résultats



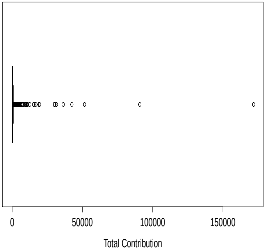
Freq
0
100
200
300
1957
1967
1977
1987
1997
Class Year



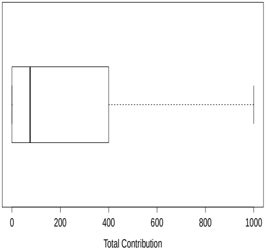
Histogram of don$TGiving
Frequency
0
200
400
600
800
1200
0
50000
100000
150000
don$TGiving



Histogram of don$TGiving[don$TGiving != 0 & don$TGiving <= 1000]
Frequency
0
50
100
150
200
250
0
200
400
600
800
1000
don$TGiving[don$TGiving != 0 & don$TGiving <= 1000]

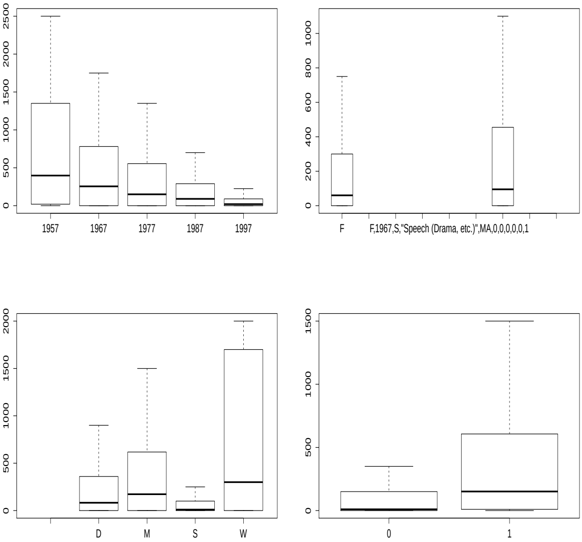


0
50000
100000
150000
Total Contribution



0
200
400
600
800
1000
Total Contribution

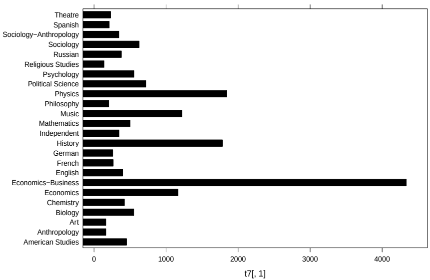
Résultats

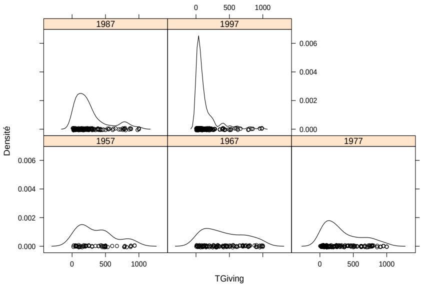

0
500
1000
1500
2000
2500
1957
1967
1977
1987
1997
0
200
400
600
800
1000
F
F,1967,S,"Speech (Drama, etc.)",MA,0,0,0,0,0,1
0
500
1000
1500
2000
D
M
S
W
0
500
1000
1500
0
1

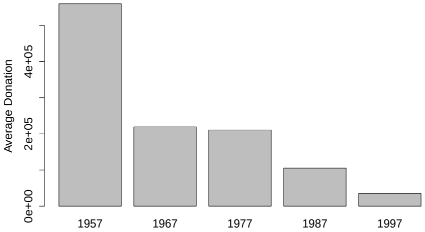
Code R

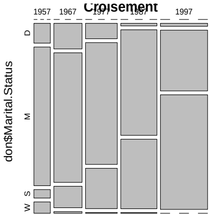
```
> t4=tapply(don$TGiving,don$Major,mean,na.rm=TRUE)
> t4
> t5=table(don$Major)
> t5
# fusionner les deux tables t4+t5
> t6=cbind(t4,t5)
# selection selon t5>10
> t7=t6[t6[,2]>10,]
# Tri suivant t4
> t7[order(t7[,1],decreasing=TRUE),]
> barchart(t7[,1],col="black")
> t4=tapply(don$TGiving,don$Next.Degree,mean,na.rm=TRUE)
> t4
> t5=table(don$Next.Degree)
> t5
> t6=cbind(t4,t5)
> t7=t6[t6[,2]>10,]
# tri selon t4
> t7[order(t7[,1],decreasing=TRUE),]
> barchart(t7[,1],col="black")
> densityplot(~TGiving|factor(Class.Year),data=don[don$TGiving<=1000 & don$TGiving>0,],col="black")
> t11=tapply(don$TGiving,don$Class.Year,FUN=sum,na.rm=TRUE)
> t11
> barplot(t11,ylab="Average Donation")
# Mosaique
> mosaicplot(table(don$Class.Year,don$Marital.Status),xlab="Class.Year",ylab="Marital.Status")
> mosaicplot(don$Class.Year don$Marital.Status,main="Croisement",xlab="Class.Year")
```

Résultats


Theatre
Spanish
Sociology−Anthropology
Sociology
Russian
Religious Studies
Psychology
Political Science
Physics
Philosophy
Music
Mathematics
Independent
History
German
French
English
Economics−Business
Economics
Chemistry
Biology
Art
Anthropology
American Studies
0
1000
2000
3000
4000
t7[, 1]


1987
1997
1957
1967
1977
Densité
0.000
0.002
0.004
0.006
0
500
1000
TGiving

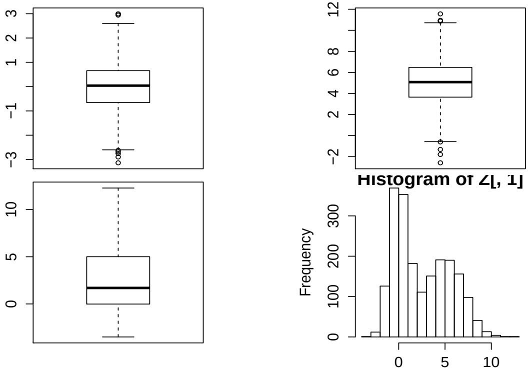

Average Donation
0e+00
2e+05
4e+05
1957
1967
1977
1987
1997


Croisement
1957
1967
1977
1987
1997
don$Marital.Status
D
M
S
W

Exercice 3

```
#============Simulation=========================
> uData <- rnorm(1000)
> ugroupe=rep(1,1000)
> uData
> X=cbind(uData,ugroupe)
> summary(X)
> sd(X[,1])
> boxplot(X[,1])
#=======
> vData <- rnorm(1000,mean=5,sd=2)
> vgroupe=rep(2,1000)
> Y=cbind(vData,vgroupe)
> mean(Y[,1])
> sd(Y[,1])
> boxplot(Y[,1])
#=======Visualisation
> par(mfrow=c(2,2))
> boxplot(uData,xlab="Variable uData")
> boxplot(vData,xlab="Variable vData")
> boxplot(Z[,1],xlab="Variable vData")
> hist(Z[,1])
```

Visualisation




```
#=======Commenter
> F=cbind(X[,1],Y[,1])
> dim(F)
> boxplot(F)
> pairs(F)
```

Outline

Types de variables

- Variable quantitative continue ou discrète: poids, taille etc.
  - Standardisation est souvent nécessaire (centered and reduced, etc.)
  - D'autres transformations utiles (log, exp, 1/x, TF-IDF, etc.)
- Variable binaire
  - Variable Nominale avec 2 catégories (modalités), la variable est *symmetrique*
  - Elle peut être considérée comme quantitative
  - Sinon, la variable est dite *asymmetrique*. Importance de 1, exemple: presence-absence d'une donnée en écologie.
- Variable qualitative ou catégorielle
  - Type nominal: generallement nous utilisons le codage *disjonctif complet*. Les catégories 1,2 and 3 sont codées respectivement par des variables binaires $(1, 0, 0)$, $(0, 1, 0)$ et $(0, 0, 1)$
  - Type ordinal: generallement nous utilisons le codage *disjonctif complet additif*. Dans ce les 3 categories 1,2 and 3 seront codées par des variables binaires $(1, 0, 0)$, $(1, 1, 0)$ et $(1, 1, 1)$. parfois, on utilise la transformation suivante $\frac{r_{ij}-1}{k_j-1} \in [0, 1]$ où $r_{ij} = 1, \ldots k_j$ est le rang de la valeur de la variable j et k_j est le nombre de valeurs distinctes, et j transformé paeut être prise pour une variable continue

Norme: E (Espace vectoriel) $||.|| : E \rightarrow \mathbb{R}^+$

- $\forall \boldsymbol{x} \in E, \lambda \in \mathbb{R}$, $||\lambda x|| = |\lambda| |x||$
- $\forall \boldsymbol{x} \in E$, $||\boldsymbol{x}|| = 0 \Rightarrow x = 0$
- $\forall \boldsymbol{x}, \boldsymbol{y} \in E$, $||\boldsymbol{x} + \boldsymbol{y}|| \leq ||\boldsymbol{x}|| + ||\boldsymbol{y}||$

Norme euclidienne et distance

- E espace euclidien, on définit la norme euclidienne $||\boldsymbol{x}||_M = \sqrt{< x, x >_M}$
- On peut montrer que can $d(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||$ est une ditance dans E
- $d_M(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_M = \sqrt{< \mathbf{x} - \mathbf{y}, \mathbf{x} - \mathbf{y} >_M} = \sqrt{(\mathbf{x} - \mathbf{y})^T M (\mathbf{x} - \mathbf{y})}$
- Par exemple, $M = I$ $d_M^2(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_j (x_{ij} - x_{i'j})^2$, $M = (1/s_j^2)$, $d_M^2(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_j (\frac{x_{ij}}{s_j} - \frac{x_{i'j}}{s_j})^2$

Autres distances

- distance de Manahattan: $d(\mathbf{x}_i, \mathbf{x}_{i'}) = \sum_{j=1}^p |x_{ij} - x_{i'j}|$
- distance de Minkowski: $d(\mathbf{x}_i, \mathbf{x}_{i'}) = (\sum_{j=1}^p \alpha_j |x_{ij} - x_{i'j}|^\lambda)^{1/\lambda}$ où λ et α_j sont positifs
- distance de Mahalanobis: (Σ est la matrice de variance) $d_{\Sigma^{-1}}^2(\mathbf{x}_i, \mathbf{x}_{i'}) = (\mathbf{x}_i - \mathbf{x}_{i'})^T \Sigma^{-1} (\mathbf{x}_i - \mathbf{x}_{i'})$
- etc.

Illustration: 4 mesures sur 23 papillons

ident	z1	z2	z3	z4
p8	22	30	19	20
p15	22	36	24	20
p22	26	34	22	21

Distances entre des papillons

- $d^2(p22, p15) = 4^2 + 2^2 + 2^2 + 1 = 25$, $d^2(p22, p8) = 4^2 + 4^2 + 3^2 + 1 = 42$
- p22 est plus proche de p15 que de p8

Données normalisés : comment ?

ident	z1	z2	z3	z4
p8	0.24176	0.32967	0.20879	0.21978
p15	0.21569	0.35294	0.23529	0.19608
p22	0.25243	0.33010	0.21359	0.20388

calcul de distances

- $d^2(p22, p15) \geq d^2(p22, p8)$, p22 est plus proche de p8 que de p15

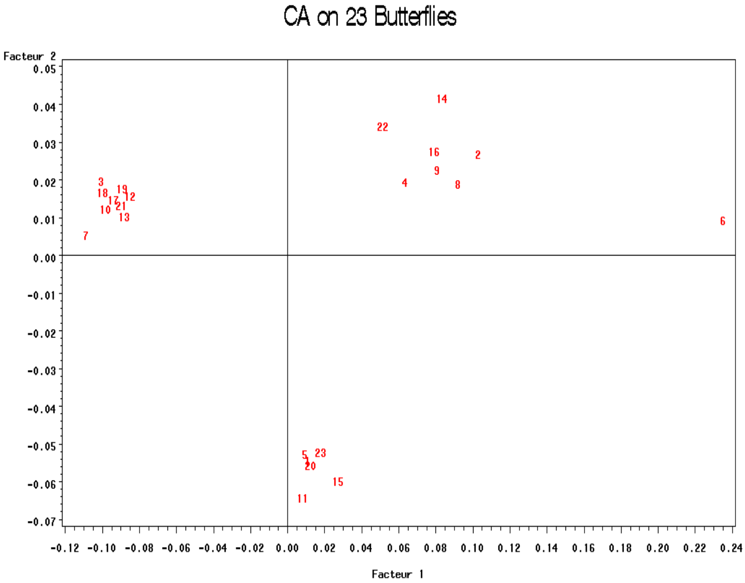
Illustration de la distance du χ^2

$$
\begin{array}{c|ccccc|c}
 & 1 & \dots & j & \dots & p & \\
\hline
1 & x_{1j} & \dots & x_{1j} & \dots & x_{1p} & x_{1.} \\
 & & & \vdots & \dots & & \\
i & x_{i1} & \dots & x_{ij} & \dots & x_{ip} & x_{i.} \\
 & & & \vdots & \dots & & \\
n & x_{n1} & \dots & x_{nj} & \dots & x_{np} & x_{n.} \\
\hline
 & x_{.1} & \dots & x_{.j} & \dots & x_{.p} & N
\end{array}
$$

	1	2	3	4	5	
1	5	4	6	1	0	16
2	6	5	4	0	1	16
3	1	0	1	7	5	14
4	1	1	0	6	5	13
5	4	5	3	4	5	21
6	5	4	4	3	4	20
	22	19	18	21	20	100

$$d_{\chi^2}(i,i') = \sum_{j=1}^{p} \frac{1}{x_{.j}} \left(\frac{x_{ij}}{x_{i.}} - \frac{x_{i'j}}{x_{i'.}} \right)^2$$

Analyse des correspondences sur l'ensemble des 23 papillons



Mesure de Dissimilarité

- $\forall \boldsymbol{x} \in \Omega,\ d(\boldsymbol{x}, \boldsymbol{x}) = 0$
- $\forall \boldsymbol{x}, \boldsymbol{y} \in \Omega,\ d(\boldsymbol{x}, \boldsymbol{y}) = d(\boldsymbol{y}, \boldsymbol{x})$

Exemple matrice de Dissimilarités

	a	b	c	d	e
a	0				
b	0.2	0			
c	1	1.05	0		
d	0.7	0.75	0.3	0	
e	1	0.8	1.5	1.3	0

Mesure de similarité

- $\forall \boldsymbol{x} \in \Omega,\ s(x, x) = s_{max}$
- $\forall \boldsymbol{x} \in \Omega,\ d(x, y) = s_{max} - s(x, y)$

Exemple de matrice de similarités

	X	Y	Z	T	W
X	40				
Y	20	40			
Z	15	39	40		
T	7	25	32	40	
W	10	38	30	10	40

Outline

Partition dure

- Soit Ω un ensemble fini,
- $\mathbf{z} = \{(z_1, z_2, ..., z_K); z_k \neq \emptyset; z_k \subset \Omega\}$ est une *partition* si
  - $\forall k \neq \ell,\ z_k \cap z_\ell = \emptyset$ et
  - $\cup_k z_k = \Omega$.
- Pour une telle partition $\mathbf{z}$ en K classes $z_1, \ldots, z_K$, chaque element de Ω appartient à une et une seule classe, cette partition $\mathbf{z}$ peut être également représentée par une matrice de classification binaire définie par :

$$\mathbf{z} = \begin{pmatrix} z_{11} & \cdots & z_{1K} \\ \vdots & \ddots & \vdots \\ z_{n1} & \cdots & z_{nK} \end{pmatrix}$$

  où $z_{ik} = 1$ si $i \in z_k$ et 0 sinon.
- La somme de ith ligne est égale à 1 et la somme de la kth colonne est égale à n_k representant ainsi la cardinalité de la classe of z_k. Ici, nous considérons une classification dure.

Partition floue

- Ensembles flous (Fuzzy sets) (Zadeh, 1965)
- Classification floue a été développé dans le début de 1970 par Ruspini généralisant la classification dure en considerant les coefficients d'appartenance $c_{ik} \in [0, 1]$

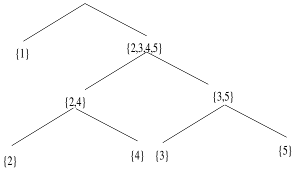
$$\mathbf{c} = \begin{pmatrix} c_{11} & \cdots & c_{1K} \\ \vdots & \ddots & \vdots \\ c_{n1} & \cdots & c_{nK} \end{pmatrix}$$

- Une partition floue est representée par une matrice de classificatin floue $\mathbf{c} = \{c_{ik}\}$ verfiant les conditions suivantes :
  - $\forall k, \sum_i c_{ik} > 0$
  - $\forall i, \sum_k c_{ik} = 1$
- La première condition considère qu'aucune classe n'est vide et la seconde indique qu'un élément exprime la décomposition de l'appartenance

Définition

- Soit Ω un ensemble fini et H un ensemble de sous-ensembles non vides de Ω
- H est une hiérarchie sur Ω si
  - $\Omega \in H$
  - $\forall \boldsymbol{x} \in \Omega,\ \{\boldsymbol{x}\} \in H$
  - $\forall h, h' \in H,\ h \cap h' = \emptyset$ ou $h \subset h'$ ou $h' \subset h$
- Exemple:
  - $\Omega = \{1, 2, 3, 4, 5\}$
  - $H = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{2, 4\}, \{3, 5\}, \{2, 3, 4, 5\}, \{1, 2, 3, 4, 5\}\}$

Exemple de représentations d'une hiérarchie



Ce type de représentations est rarement utilisé. On préfère associer un indice à une hiérarchie afin que la représentation soit facilement interprétable.

- L'indice (*index*) associée à une hiérarchie H est une fonction notée i de H à $\mathbb{R}^+$ verifiant les proprietés suivantes :
  - $h \subset h'$ et $h \neq h' \Rightarrow i(h) < i(h')$ (i est strictement croissante)
  - $\forall x \in \Omega \quad i(\{x\}) = 0$.
- Dans la suite nous notons (H, i) l'hiérarchie indicée
- Exemple : En associant aux classe {1},{2},{3},{4},{5},{2,4},{3,5},{2,3,4,5}, {1,2,3,4,5} de la précedente hiérarchie les valeurs 0,0,0,0,0,1,2,2.5,3.5, on obtient (H, i) qui peut être representée par un arbre appelé dendrogramme

Représentation de (H, i) par un dendrogramme

Si $\mathbf{z} = (z_1, z_2, \ldots, z_K)$ est une partition de Ω, H formé par des classes z_k, des singletons de Ω et Ω lui même constitue une hiérarchie. Inversement, il est possible d'associer à chaque niveau de (H, i) une partition. Par conséquent, (H, i) correspond alors à ensemble de classes emboités.

- Le nombre de hiérarchies et partitions possibles à définir sur Ω croit très vite lorsque la cardinalité Ω augmente
- Par exemple, le nombre de partitions de n objects en K classes est donnée par la formule suivante

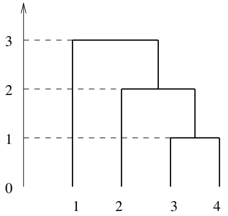
$$S(n,K) = \frac{1}{K!}\sum_{k=0}^{K}(-1)^{k-1}C_k^K k^n$$

- Quand n et K deviennent grands, nous avons $S(n,K) \approx \frac{K^n}{K!}$, par exemple $S(100,5) \approx 10^{67}$

(n,K)	1	2	3	4	5	6	7	8
1	1							
2	1	1						
3	1	3	1					
4	1	7	6	1				
5	1	15	25	10	1			
6	1	31	90	65	15	1		
7	1	63	301	350	140	21	1	
8	1	127	966	1701	1050	266	28	1

Ultrametrique associée à (H, i) : fonction φ

- L'application de la fonction φ sur l'hiérarchie (H_1, i_1) (a) implique l'ultrametrique δ_1 de (b)



a : (H_1, i_1)

	1	2	3	4
1	0			
2	3	0		
3	3	2	0	
4	3	2	1	0

b : $\delta_1 = \varphi(H_1, i_1)$

(H, i) associciée à une ultramétrique : fonction ψ

- L'application de ψ à l'ultramétrique δ_1 implique : nous avons $D_\delta = \{0, 1, 2, 3\}$. Les classes d'équivalence des 4 relations R_α are $R_0 : \{1\}, \{2\}, \{3\}, \{4\}$, $R_1 : \{1\}, \{2\}, \{3, 4\}$, $R_2 : \{1\}, \{2, 3, 4\}$ and $R_3 : \{1, 2, 3, 4\}$.
- L'hiérarchie obtenue est alors $\{\{1\}, \{2\}, \{3\}, \{4\}, \{3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$ et les indices associés aux classes de l'hiérarchie sont rspectivement $(0, 0, 0, 0, 1, 2, 3)$. On trouve alors (H_1, i_1)

Algorithme de Classification Ascendante Hiérarchique (CAH)

Construction d'une hiérarchie indicée

- pendant le processus de regroupement des classes dans l'approche ascendante, il est nécessaire de définir une distance entre les classes afin de fusionner les plus proches d'entre elles. En général, à partir de la mesure de dissimilarité sur Ω nous définissons un *distance* D entre les classes. En fait, D est une mesure de dissimilarité dite aussi critère d'agrégation. Nous verrons plus tard différentes manières de définir ces types de mesures. Maintenant, nous présentons brièvement les différentes étapes de l'algorithme:
  1. **Initialization**: Chaque objet étant une classe singleton, on calcule les dissimilarités entre ces objets.
  2. **Repeat**
     - fusionner les classes les plus proches au sens de D
     - calculer la distance entre la nouvelle classe obtenue par fusion et les autres classes non fusionnées
  3. **Until** le nombre de classes est égal à 1

Il est facile de montrer que l'ensemble des classes définies au cours des itérations forme une hiérarchie

Critères d'agrégation

- Differents critères d'agrégation D existent, les plus populaires sont
  - Single linkage ou Nearest Neighbor approach (Sibson, 1973)
  $$D(A,B) = \min\{d(i,i'), i \in A \text{ et } i' \in B\};$$
  - Complete linkage ou farthest Neighbor approach (Sorenson, 1948)
  $$D(A,B) = \max\{d(i,i'), i \in A \text{ et } i' \in B\};$$
  - Average linkage (Sokal and Michener, 1958)
  $$D(A,B) = \frac{\sum_{i \in A} \sum_{i' \in B} d(i,i')}{n_A.n_B}$$

  où n_E represente la cardinalité de la classe E.

Formules de Recurrence de Lance and Williams, 1967

- Pour les trois critères d'agrégation, il existe des relations de simplification des calculs, des distances entre les clusters, nécessaires à la classification hiérarchique (CAH), sans ces sortes de relations, il serait prohibitif dans le calcul du temps d'appliquer ce type d'algorithme Ces relations appelées généralement formules de recurrence de Lance and Williams, sont définies comme suit

$$D_{\min}: \qquad D(A, B \cup C) = \min\{D(A,B), D(A,C)\};$$

$$D_{\max}: \qquad D(A, B \cup C) = \max\{D(A,B), D(A,C)\};$$

$$D\text{average}: \qquad D(A, B \cup C) = \frac{n_B.D(A,B) + n_C.D(A,C)}{n_B + n_C}.$$

Remarque :

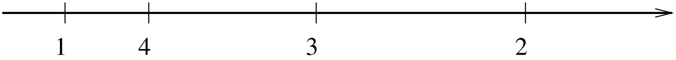
$$D(A, B \cup C) = \alpha_1 D(A,B) + \alpha_2 D(A,C) + \beta D(B,C) + \gamma |D(A,B) - D(A,C)|.$$

Donc $D_{\min}$ est obtenue en prenant $\alpha_1 = \alpha_2 = 0.5, \beta = 0, \gamma = -0.5$, $D_{\max}$ en prenant $\alpha_1 = \alpha_2 = 0.5, \beta = 0, \gamma = 0.5$ and Daverage en prenant $\alpha_1 = \frac{n_B}{n_B+n_C}$, $\alpha_2 = \frac{n_C}{n_B+n_C}$, $\beta = 0$, $\gamma = 0$

Exemple

- Ci-après, nous considérons 4 des points alignés, séparés successivement par les distances de 2, 4 et 5: Nous prenons comme mesure de dissemblance entre ces points, la distance euclidienne habituelle. Nous appliquons l'algorithme CAH selon les trois critères d'agrégation, les résultats sont rapportés dans ce qui suit

Data

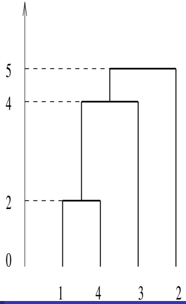


$D_{\min}$

	1	2	3	4
1	0			
2	11	0		
3	6	5	0	
4	2	9	4	0

	{1,4}	2	3
{1,4}	0		
2	9	0	
3	4	5	0

	{1,4,3}	2
{1,4,3}	0	
2	5	0

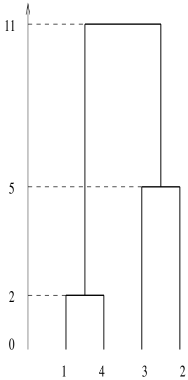


$D_{\max}$

	1	2	3	4
1	0			
2	11	0		
3	6	5	0	
4	2	9	4	0

	{1,4}	2	3
{1,4}	0		
2	11	0	
3	6	5	0

	{1,4}	{2,3}
{1,4}	0	
{2,3}	11	0

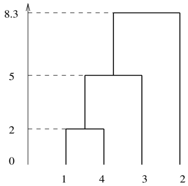


*D*average

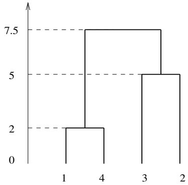
	1	2	3	4
1	0			
2	11	0		
3	6	5	0	
4	2	9	4	0

	{1,4}	2	3
{1,4}	0		
2	10	0	
3	5	5	0

	{1,4,3}	2
{1,4,3}	0	
2	8.3	0



	{1,4}	{2,3}
{1,4}	0	
{2,3}	7.5	0



Notons que dans ce cas, nous pouvons obtenir différentes solutions solutions si nous choisissons de fusionner les classes {1,4} et {3} ou les classes {2} and {4}.

Méthode de Ward Method

- Contrairement aux critères décrits précédents, le critère de Ward (Ward, 1963) nécessite que l'on dispose des données brutes et de la dissemblance entre les objets. Lorsque l'ensemble Ω à classifier est associé à un nuage de points dans $\mathbb{R}^p$ (chaque point a un poids égal à $\frac{1}{n}$) muni de la métrique euclidienne, le critère prend la forme suivante
$$D(A,B) = \frac{n_A n_B}{n_A + n_B} d^2(\boldsymbol{\mu}_A, \boldsymbol{\mu}_B)$$
où $\boldsymbol{\mu}_E$ represente le centre de l'ensemble E. La CAH associatée est souvant appelée la méthode de Ward (Ward, 1963). Il existe aussi une formule de recurrence
$$D(A, B \cup C) = \frac{(n_A + n_B) \times D(A,B) + (n_A + n_C) \times D(A,C) - n_A \times D(B,C)}{n_A + n_B + n_C},$$
qui peut être déduite de la forme de recurrence générale
$$D(A, B \cup C) = \alpha_1 D(A,B) + \alpha_2 D(A,C) + \beta D(B,C) + \gamma |D(A,B) - D(A,C)|,$$
avec $\alpha_1 = \frac{n_A+n_B}{n_A+n_B+n_C}$, $\alpha_2 = \frac{n_A+n_C}{n_A+n_B+n_C}$, $\beta = -\frac{n_A}{n_A+n_B+n_C}$ and $\gamma = 0$

Analysis of Flying Mileages Between Ten U.S. Cities

Atlanta	Chicago	Denver	Houston	LA	Miami	NewYork	SanFrancisco	Seattle	WashingtonD.C
0									
587	0								
1212	920	0							
701	940	879	0						
1936	1745	831	1374	0					
604	1188	1726	968	2339	0				
748	713	1631	1420	2451	1092	0			
2139	1858	949	1645	347	2594	2571	0		
2182	1737	1021	1891	959	2734	2408	678	0	
543	597	1494	1220	2300	923	205	2442	2329	0

Applications

- Single linkage
- Complete linkage
- Average linkage

Introduction

- Nous avons vu que le concept de (H, i) est équivalent à la notion de l'ultramétrique. L'algorithme CAH transforme une première mesure de dissimilitude d dans une nouvelle mesure de dissimilitude δ ayant la propriété d'une ultramétrique. Le but de la classification hiérarchique pourrait être ensuite posé en ces termes : Trouver la plus proche ultramétrique δ à la mesure de dissimilarité d.

- Il reste à choisir une distance entre d et δ sur Ω. Il s'agit d'un problème difficile, nous pouvons utiliser par exemple

$$\Delta(d, \delta) = \sum_{i,i' \in \Omega} (d(i, i') - \delta(i, i'))^2$$

ou

$$\Delta(d, \delta) = \sum_{i,i' \in \Omega} |d(i, i') - \delta(i, i')|.$$

Lien entre méthode de Ward et de l'inertie intraclasse

- Soit $\mathbf{z} = (z_1, \ldots, z_K)$ une partition et $\mathbf{z}'$ une partition obtenue à partir $\mathbf{z}$ fusioannt les classes $\mathbf{z}_k$ et $\mathbf{z}_\ell$. Nous pouvons montrer que :

$$W(\mathbf{z}') - W(\mathbf{z}) = \frac{n_k n_\ell}{n_k + n_\ell} d^2(\overline{x}_k, \overline{x}_\ell)$$

- La fusion de ces deux classes augmente nécessairement la variance intra-classe. Il est possible de propose une CAH qui fusionne à chaque étape les deux classes qui font augmentre le moins possible la variance intra-classe i.e. minimisant l'expression suivante :

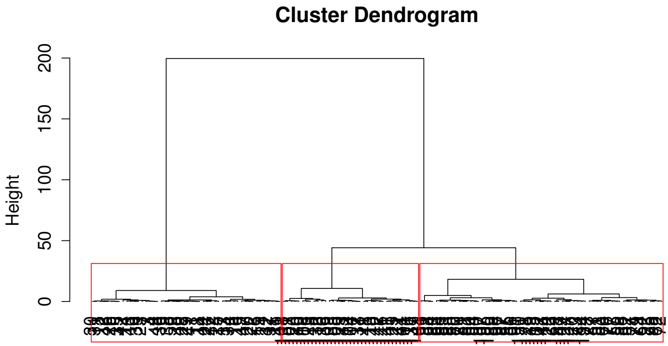
$$D(A, B) = \frac{n_k n_\ell}{n_k + n_\ell} d^2(\overline{x}_k, \overline{x}_\ell),$$

et nous obtenons le critère de Ward

Exemple de la méthode de Ward en 3 étapes

```
> d <- dist(iris[,-5], method = "euclidean") # distance matrix
> fit <- hclust(d, method="ward") # AHC
> plot(fit,hang=-1) # display dendogram
groups <- cutree(fit, k=3) # cut tree into 3 clusters
# draw dendogram with red borders around the 3 clusters
> rect.hclust(fit, k=3, border="red")
```

Exemple



Exemple de la méthode de Ward en 3 étapes

```
x = read.table("compact.txt")
plot(x,main="Jeu de données")
# CAH
# Matrice des distances (euclidiennes)
D <- dist(x, method = "euclidean")
# CAH - lien minimum
H1 <- hclust(D^2, method = "single")
classes1 <- cutree(H1, k=3)
# Coupe de l'arbre pour avoir la meilleure partition en deux classes
# CAH - lien maximum
H2 <- hclust(D^2, method = "complete")
classes2 <- cutree(H2, k=3)
# CAH - lien moyen
H3 <- hclust(D^2, method = "average")
classes3 <- cutree(H3, k=3)
# CAH - Ward
H4 <- hclust(D^2, method = "ward")
classes4 <- cutree(H4, k=3)
# Visualisation graphique des classes (pour les quatres méthodes)
par(mfrow=c(2,2))
plot(x[,1],x[,2],col=classes1,pch=classes1,main="CAH - lien minimum")
plot(x[,1],x[,2],col=classes2,pch=classes2,main="CAH - lien maximum")
plot(x[,1],x[,2],col=classes3,pch=classes3,main="CAH - lien moyen")
plot(x[,1],x[,2],col=classes4,pch=classes4,main="CAH - ward")
```

Outline

Description

- On garde les notations et nous décrions l'algorithme k-means lorsque l'ensemble à classifier Ω est mesuré par p variables quantitatives.
- Pour trouver la partition optimale $\mathbf{z}$ il suffit, par exemple, de minimiser la variance intra-classe $W(\mathbf{z})$

$$W(\mathbf{z}) = \sum_{k=1}^{K} \sum_{i \in \mathbf{z}_k} ||\boldsymbol{x}_i - \overline{x}_{\mathbf{z}_k}||^2.$$

Ceci est équivalent à maximiser la varince inter-classe

$$B(\mathbf{z}) = \sum_{k=1}^{K} \pi_k ||\overline{x}_{\mathbf{z}_k} - \overline{x}||^2,$$

avec π_k le poids de la classe z_k et $\overline{x}$ est le vecteur centre dee gravités des données. Cette equivalence est due à la décomposition de la variance totale I des données

$$I = \sum_{i=1}^{n} \pi_i ||\mathbf{x}_i - \overline{x}||^2 = W(\mathbf{z}) + B(\mathbf{z})$$

Description of k-means

- L'optimisation de $W(\mathbf{z})$ est équivalente à l'optimization à l'optimization de $W(\mathbf{z}, \boldsymbol{\mu})$ (Discrete sum-of-squares (SSQ))

$$W(\mathbf{z}, \boldsymbol{\mu}) = \sum_{k=1}^{K} \sum_{i \in z_k} ||\mathbf{x}_i - \boldsymbol{\mu}_k||^2 = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} ||\mathbf{x}_i - \boldsymbol{\mu}_k||^2$$

  où $z_{ik} \in \{0, 1\}$ and $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K)$ avec $\boldsymbol{\mu}_k$ de $\mathbb{R}^p$ represente le centre ou le prototype de la classe z_k.
- Cette optimisation peut être réalisée par k-meanset et les principales étapes sont les suivantes :

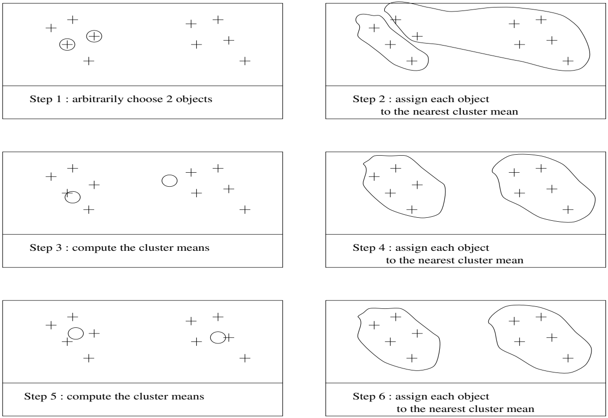
  1. Selectionner K objets de Ω qui forment les K premiers centres $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K$.
  2. Tant que non convergence
     1. affecter chaque objet de Ω à la classe dont le centre est le plus proche de cet objet. En cas de non unicité, l'objet est affecté à la classe dont l'indice est le plus faible (par exemple).
     2. Les centres de classes calculés deviennent les nouveaux centres.

Dans le processus des itérations, k-means offre une séquence $\boldsymbol{\mu}^{(0)}, \mathbf{z}^{(1)}, \boldsymbol{\mu}^{(1)}, \mathbf{z}^{(2)}, \ldots$ de partitions et de centres faisant décroitre le critère de variance intra-classe jusqu'à la convergence.

Description de la version k-means (Forgy, 1965)

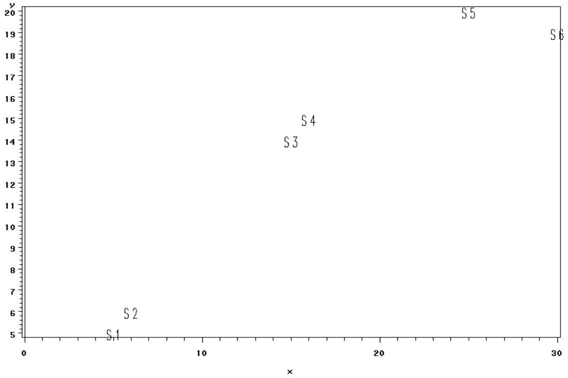
- L'algorithme k-means peut être illustrer de cette façon

Processus des itérations dans k-means.



L'algorithme répète ces itérations jusqu'à la convergence. On constate qu'il converge à la partition visible en 2 classes. Cependant attention à l'initialisation.

Exemple et problème



Sans faire de calcul

- Initialiser *k*-means avec S1, S4 et S6 identifier les 3 clusses.
- Initialiser *k*-means avec S4, S5 et S6 identifier les 3 clusses.

- Si le nombre de classes n'est pas conu, plusieurs solutions permettant de résoudre ce problème très difficile sont utilisés. Par exemple, la meilleure partition est demandée pour plusieurs nombres de classes et nous étudions la diminution du critère selon le nombre de classes pour sélectionner le nombre de classes en utilisant la méthod du *coude*. En effet, la qualité d'une partition peut être évaluée par le *Rsquare* dit Rcarré (RSQ)

  $$RSQ = 1 - \frac{W}{I} = \frac{B}{I}$$

  L'algorithme k-means est de compléxité linéaire ce qui implique une convergence rapide, il suffit donc de l'exécuter avec différents nombres de classes er d'observer grâce à la méthode de coude le ou les nombres de classes appropriés.

- Sachant que, selon les points de départ choisis, les résultats seront différents, il reste à exploiter ces différents résultats. Plusieurs solutions ont été proposées: nous aplliquons $k$means avec différentes initialisations aléatoires. Plusieurs d'autres stratégies sont possibles
  - Nous sélectionnons une *bonne* initialisation avec des informations supplémentaires ou avec une procédure automatique (les points fortement lointain, les régions à forte densité, etc.)
  - Nous devrions cependant faire un compromis entre le temps nécessaire de recherche de la configuration initiale et le temps nécessaire à l'algorithme lui-même.

Exemple

```
x <- read.table("billets.txt",header=T)
# K-means
vecteur_criteres = numeric(10)
for (k in 1:10)
vecteur_criteres[k]=kmeans(x[,1:7],k,nstart=100)$tot.withinss
plot(vecteur_criteres,type="b",ylab="Inertie-Intra",xlab="K")
# Le coude du critère est observé ici pour K=2 classes
classesKM = kmeans(x[,1:7],2,nstart=100)$cluster
```

- Liens entre k-means la méthode de Ward: Les deux méthodes sont assez similaires dans la mesure où elles cherchent à minimiser la variance intra-classe.

  1. Appliquer k-means pour classifier Ω en 50 classes (par exemple). En pratique, ce nombre de classes dépend de la taille des données, empirequement la valeur de $n^{\frac{1}{3}}$ est suggérée.
  2. Exécuter la méthode de Ward sur les centres des classes obtenues.
  3. A partir du dendrogramme proposer un nombre de classes approprié en utilisant par exemple le critère SPRSQ.
  4. Eventuellement, réappliquer k-means sur les classes obtenues pour améliorer SPRSQ.

- Interprétation des classes est une phase importante après la classification
  - Utiliser l'analyse exploratoire (moyennes, écart-types) pour décrire les classes
  - Utiliser les boites à moustaches pour décrire les classes en fonction de toutes les variables.
  - Utiliser les boites à moustaches pour décrire variable selon les classes.
  - Utiliser la visualisation (ACP, par exemple).
- Exemples avec package NbClust