

Introduction à R

- > Qu'est-ce que R ?
- > Objets sous R et fonctions basiques pour leur manipulation
- > Représentations graphiques
- > Éléments de programmation sous R
 - > création de fonctions
 - > structures de contrôle
- > Outils usuels pour l'analyse de données
 - > statistiques descriptives
 - > modèles de régression

Introduction à R

R is a language and environment for statistical computing and graphics.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering, ...) and graphical techniques, and is highly extensible.

Introduction à R

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes :

- > an effective data handling and storage facility,
- > a suite of operators for calculations on arrays, in particular matrices,
- > a large, coherent, integrated collection of intermediate tools for data analysis,
- > graphical facilities for data analysis and display either on-screen or on hardcopy, and
- > a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

Introduction à R

R is designed around a true computer language, and it allows users to add additional functionality by defining new functions.

Many users think of R as a statistics system. We prefer to think of it of an environment within which statistical techniques are implemented. R can be extended (easily) via packages. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

I. Les objets sous R

Nombres

5

5*3+1

a <- 5*3+1 # équivalent à a=5*3+1 et à 5*3+1 -> a

a

Booléens

B <- TRUE # équivalent à B<-TRUE. R n'est pas sensible aux espaces

B

a==2

a<2

a>=16

Chaînes de caractères

C <- "mot"

C

I. Les objets sous R

Vecteurs

```
B <- c(7,9,45,85,14)
```

```
B
```

```
C <- 8:36 # équivalent à C <- c(8:36)
```

```
C
```

```
E <- c() # déclare l'existence d'un objet (e.g. une variable). On peut aussi  
faire D <- NULL
```

```
E <- c(B,C)
```

```
E
```

```
E <- c(7,9,45,85,14,8:36,a,78)
```

```
E
```

```
E[1] # renvoie la première valeur
```

```
length(E) # donne la longueur de E
```

```
# Lister les objets en mémoire:
```

```
ls()
```

```
# Supprimer un ou plusieurs objets
```

```
rm(a)
```

```
rm(B,C)
```

```
# rm(list=ls()) efface tous les objets en mémoire
```

I. Les objets sous R

Exercice : créer un objet x contenant successivement les valeurs de 1 à 10, puis les valeurs 3, 4 et 5

Exercice: extraire de la 2ème à la 6ème valeur de x, puis les 9è, 11è valeurs. Stocker le résultat dans un objet X. Extraire ensuite seulement la dernière valeur de x (juste en l'affichant, i.e. sans en faire un objet)

I. Les objets sous R

Quelques exemples d'opérations arithmétiques

```
xlog <- log(x)
```

```
xlog10 <- log(x,base=10)
```

Est-ce que `log(x)` et `log(x,base=10)` sont identiques?

```
xlog==xlog10
```

Exercice: faire la somme des valeurs n°4 à 8 de x et de l'avant-dernière valeur de x

Exercice: multiplier la 2eme valeur de x par la 3eme valeur de y <- 1:4

Exercice: mettre à 0 à toutes les valeurs de x inférieures ou égales à 5

`x <- x[x>=5]` # ne retient que les valeurs supérieure ou égales à 5 => l'objet final aura une taille inférieure à l'objet initial

```
x <- c(1:10,3:5)
```

```
x[x<=5] <- 0
```

```
x
```


I. Les objets sous R

Fonction which(): donne les indices (les positions dans l'objet) des valeurs vérifiant le critère qu'on spécifie dans la fonction

```
which(x==0)
```

Fonction seq()

?seq # donne l'aide associée à la fonction, équivalent à help(seq)

```
seq(1,10,by=2)
```

```
seq(1,10,length.out=5)
```

Exercice: pour une valeur sur 5 de l'objet x, ajouter 0.5 et prendre le log10 de la valeur obtenue x+0.5 (les valeurs de x seront modifiées)

Exercice: créer un objet X ne contenant qu'une valeur sur 2 de x

I. Les objets sous R

Générer des valeurs aléatoires

y <- rnorm(10) # génère 10 valeurs aléatoires tirées dans une distribution normale centrée-réduite

y

Exercice: où sont les valeurs négatives de y ?

Matrices

A0 <- matrix(data=rnorm(25),nrow=5,ncol=5)

dim(A0)

A0[1,1] # sélectionne la valeur de la 1ere ligne et de la 1ere colonne de A

A0[2,] # deuxième ligne (implicitement, toutes les colonnes sont sélectionnées)

A0[,3] # idem pour les colonnes

I. Les objets sous R

Créer une matrice vide de taille déterminée (à remplir ultérieurement)

```
A <- matrix(data=NA,nrow=4,ncol=4) # NA = not available
```

A

Exercice: remplir la matrice A (sans valeurs) avec des 0 en ligne 1, des valeurs aléatoires en ligne 2, une suite de valeurs de 4 à 56 en ligne 3, un zéro en 1ere et dernière positions de la ligne 4 et un 4 et un 5 en 2e et 3e positions de la ligne 4

Exercice: dans la matrice A, extraire 1 valeur sur 2 de la colonne 3, la 3eme valeur de la colonne 2, toute la colonne 1, les 1ere et 3e valeurs de la colonne 4. Ranger toutes ces valeurs dans un objet a de type vecteur les unes à la suite des autres (dans l'ordre énoncé précédemment).

I. Les objets sous R

Exercice: fonction `sort()` -> trier des valeurs ?`sort`. Modifier la matrice A en triant dans l'ordre décroissant les valeurs de la 3e colonne de A

Exercice: Fonctions `cbind()` et `rbind()`. `cbind()` et `rbind` permettent de créer des matrices. ?`cbind`. Créer une matrice B de valeurs aléatoires à l'aide de la fonction `cbind()`, puis une matrice C de valeurs aléatoires avec `rbind()`

Que renvoie `is.matrix(Y)` ?

I. Les objets sous R

Data frames

?data.frame

```
A <- c(21,20,23,19,21,26)
```

```
B <- c("h","f","f","f","h","h")
```

```
Dn <- data.frame(Age=A,Genre=B)
```

```
Dn
```

nom des colonnes

```
colnames(Dn)
```

```
row.names(Dn) # les lignes Dn n'ont pas de nom
```

```
row.names(Dn) <- c("Jean","Marie","Valérie","Chloé","Paul","Pierre")
```

```
Dn
```

```
colnames(Dn) <- c("Age","Sexe") # modification du nom des colonnes
```

I. Les objets sous R

Exercice: afficher l'âge de Chloé

Exercice: afficher dans Dn, les valeurs de la colonne 1 pour lesquelles la colonne nommée "Sexe" correspond à "h"

Exercice: Afficher le sexe des personnes dont l'âge est strictement inférieur à 23 ans

I. Les objets sous R

Exercice: afficher l'âge de Chloé

Exercice: afficher dans Dn, les valeurs de la colonne 1 pour lesquelles la colonne nommée "Sexe" correspond à "h"

Exercice: Afficher le sexe des personnes dont l'âge est strictement inférieur à 23 ans

Statistiques élémentaires sur une data frame

mean(Dn\$Age)

min(Dn\$Age)

max(Dn\$Age)

median(Dn\$Age)

sd(Dn\$Age)

II. Représentations graphiques simples

```
x <- 501:600
y <- rnorm(100)
plot(x,y)
plot(y)
plot(x,y,type="l",lwd=2,col="blue",lty="solid",xlab="Abscisses",ylab="Ordon
nées", main="Mon premier graphe", xlim=c(400,650), ylim=c(-4,4))
abline(h=0,col="red",lty="dashed",lwd=2)
abline(v=501,col="red",lty="dashed",lwd=2)
abline(v=600,col="red",lty="dashed",lwd=2)
lines(x,z<-2*rnorm(100),col="darkgreen",lwd=2)
```


II. Représentations graphiques simples

La figure s'enregistre dans le répertoire de travail courant. Pour le connaître, on utilise la commande:

```
getwd()
```

pour changer de répertoire de travail:

```
setwd("/home/nico/Documents/Enseignements/LICENCE/L2SVT-STE/info")
```

```
getwd()
```

On peut aussi générer des fichiers TIFF, JPEG, BMP, EPS...

```
?tiff
```

```
?jpeg
```

```
?bmp
```

II. Représentations graphiques simples

sauvegarder une figure

pdf("/home/nico/Documents/Enseignements/LICENCE/L2SVT-STE/info/figure.pdf") # ouvre "pour création" un fichier pdf

plot(x,y,type="l",lwd=2,col="blue",lty="solid",xlab="Abscisses",ylab="Ordonnées", main="Mon premier graphe", xlim=c(400,650), ylim=c(-4,4))

abline(h=0,col="red",lty="dashed",lwd=2)

abline(v=501,col="red",lty="dashed",lwd=2)

abline(v=600,col="red",lty="dashed",lwd=2)

lines(x,z<-2*rnorm(100),col="darkgreen",lwd=2)

dev.off() # finalise/ferme/enregistre le fichier pdf

III. Fonctions et structures de contrôle dans R

Fonctions

```
aire <- function(longueur, largeur) {  
  result <- longueur * largeur  
  return(result)  
}  
A <- aire(3,2)
```

```
vol <- function(hauteur, longueur, largeur) {  
  result <- hauteur*aire(longueur, largeur)  
  return(result)  
}  
V <- vol(6,3,2)
```

III. Fonctions et structures de contrôle dans R

Exercice: créer une fonction `surfcy` qui calcule la surface et le volume d'un cylindre pour des dimensions données du cylindre

III. Fonctions et structures de contrôle dans R

Exercice: créer une fonction surfcy qui calcule la surface et le volume d'un cylindre pour des dimensions données du cylindre

```
surfcy <- function(r,h) {
```

```
  S <- 2*pi*r*h
```

```
  V <- pi*r^2*h
```

```
  return(c(S,V))
```

```
}
```

```
surfcy(4,8)
```

III. Fonctions et structures de contrôle dans R

Exercice à rendre: générer une série de 100 valeurs aléatoires de moyenne 12 et d'écart-type 20, puis créer une fonction `centred()` pour standardiser la série (i.e. pour ramener la moyenne à 0 et l'écart-type à 1)

III. Fonctions et structures de contrôle dans R

Alternatives: if

```
x <- 3
```

```
if (x>0) cat("Positif\n") else cat("Négatif\n")
```

```
if (x>0) { cat("Positif\n") } else { cat("Négatif\n") } # les accolades sont nécessaires s'il faut exécuter plusieurs opérations au lieu d'une seule.
```

Exemple:

```
if (x>0) {cat("Super!\n");cat("Positif\n")} else {cat("Zut!\n");cat("Négatif\n")}
```

Une application de if: calcul de la factorielle d'un nombre:

```
facto <- function(n) {  
  if (n==0) return(1) else return(n*facto(n-1))  
}
```

```
facto(0)
```

```
facto(3)
```

III. Fonctions et structures de contrôle dans R

Boucles: exécuter plusieurs fois à la suite une opération

for

```
for (i in 1:10) cat("Bonjour\n")
```

```
for (i in 1:10) {
```

```
  print(i)
```

```
  cat("Bonjour\n")
```

```
}
```

Exercice: à l'aide d'une boucle for, déterminer à partir d'une série de 100 valeurs aléatoires positives ou nulles, les valeurs des sommes cumulées de cette série

III. Fonctions et structures de contrôle dans R

Boucles: exécuter plusieurs fois à la suite une opération

for

```
for (i in 1:10) cat("Bonjour\n")
```

```
for (i in 1:10) {  
  print(i)  
  cat("Bonjour\n")  
}
```

Exercice: à l'aide d'une boucle for, déterminer à partir d'une série de 100 valeurs aléatoires positives ou nulles, les valeurs des sommes cumulées de cette série

```
x <- rnorm(100)
```

```
x[x<=0] <- 0
```

```
y <- x[1]
```

```
for (i in 2:100) {
```

```
  y[i] <- x[i]+y[i-1]
```

```
}
```

```
plot(x,type="l")
```

```
plot(y,type="l")
```

III. Fonctions et structures de contrôle dans R

```
# while
```

```
i <- 1
```

```
while (i <= 10) {
```

```
  print(i)
```

```
  cat("Bonjour\n")
```

```
  i <- i+1
```

```
}
```

Exercice: le même que précédemment mais avec while

III. Fonctions et structures de contrôle dans R

while

```
i <- 1
while (i <= 10) {
  print(i)
  cat("Bonjour\n")
  i <- i+1
}
```

Exercice: le même que précédemment mais avec while

```
x <- rnorm(100)
x[x<=0] <- 0
y <- x[1]
i <- 2
while (i <= 100) {
  y[i] <- x[i]+y[i-1]
  i <- i+1
}
```

III. Fonctions et structures de contrôle dans R

```
# repeat  
i <- 1  
repeat {  
  print(i)  
  cat("Bonjour\n")  
  i <- i+1  
  if (i>10) break  
}
```

Exercice: encore le même mais avec repeat

III. Fonctions et structures de contrôle dans R

```
# repeat
i <- 1
repeat {
  print(i)
  cat("Bonjour\n")
  i <- i+1
  if (i>10) break
}
```

```
# Exercice: encore le même mais avec repeat
x <- rnorm(100)
x[x<=0] <- 0
y <- x[1]
i <- 2
repeat {
  y[i] <- x[i]+y[i-1]
  i <- i+1
  if (i>length(x)) break
}
```


