

## Quelques commandes R

R Version 1.6.2

### Lancement de R

R	Lancement d'une session interactive (ou menu démarrer... sous windows)
R --vanilla < file	Lancement de R et execution des commandes contenues dans <i>file</i>
R --help	description des options de commande

### Arrêt de R

q()	sortie de R
INTERRUPT	(e.g. C-c ou icone stop) arrêt de la commande en cours et retour au niveau principal

### Aide

help.start()	démarrage de l'aide html
help(command)	aide en ligne de <i>command</i>
apropos(command)	recherche d'aide à propos de <i>command</i>
help.search("command")	recherche de <i>command</i> dans les packages R

### Assignement

var <- expr	Assigne l'expression à la variable
expr -> var	Assigne l'expression à la variable

### Types élémentaires

NA	valeur manquante
TRUE FALSE	booléen (logique)
numeric	reel ou entier
complex	complexe
character	caractère

### Caractères spéciaux

\\	backslash
\n	newline, ASCII code 10
\t	tabulation, ASCII code 9
#	commentaires (jusqu'à la fin de ligne)

### Vecteurs

Les vecteurs ne sont pas des matrices et n'ont qu'1 dimension. Ils sont constitués d'un seul type avec ou non des NA (objet homogène)

c(x, y, ...)	saisie d'un vecteur
rep(c(0,1),10)	répétition d'un motif
10:(-1)	séquence de 10 à -1
seq(1,20,by=0.5)	séquence par pas
seq(1,20,length=13)	séquence par longueur finale
x[c(6,1,9)]	affichage des coordonnées 6, 1 et 9
x[c(-6,-2)]	affichage des coordonnées <i>autres</i> que 6 et 2
x[c(TRUE,FALSE)]	affichage de la coordonnée 1 et pas de la 2
x[x>0]	affichage des coordonnées positives

### Arithmétique

x + y	addition (elt par elt)
x - y	soustraction (elt par elt)

x %*% y	multiplication matricielle
x %o% y	produit exterieur
x * y	multiplication (elt par elt)
x ^ y	élévation à la puissance (elt par elt)
x / y	division (elt par elt)
- x	moins unaire
x %/% y	partie entière de la division (elt par elt)
x %% y	reste de la division (elt par elt)

### Autres opérations (numérique)

c(x, y)	concaténation de <i>x</i> et de <i>y</i>
length(x)	longueur de <i>x</i>
min(x)	minimum de <i>x</i>
max(x)	maximum de <i>x</i>
range(x)	étendue de <i>x</i>
mean(x)	moyenne de <i>x</i>
median(x)	médiane de <i>x</i>
IQR(x)	écart inter-quartile de <i>x</i>
quantile(x)	quantile de <i>x</i>
var(x)	variance de <i>x</i>
sd(x)	écart type de <i>x</i>
rank(x)	rang des éléments de <i>x</i>
sort(x)	éléments de <i>x</i> ordonnés
order(x)	coordonnées du plus petit élément de <i>x</i> , puis du 2eme plus petit...
which(x==min(x))	indice du minimum <i>x</i>
sum(x)	somme (des éléments) de <i>x</i>
prod(x)	produit de <i>x</i>
diff(x)	différence des éléments consécutifs de <i>x</i>
cumsum(x)	somme cumulée de <i>x</i>

### Autres opérations (caractères)

paste(x, y)	concaténation de <i>x</i> et <i>y</i> (elt par elt)
substring(x,2,3)	extraction du 2eme et 3eme caractère (elt par elt)

### Facteurs

Les facteurs sont les mesures de variables qualitatives. Ils ont une longueur (le nombre de mesures) et des modalités (levels). Ce sont des vecteurs de caractères "spéciaux".

factor(x)	transforme <i>x</i> en facteur
ordered(x)	transforme <i>x</i> en facteur ordonné
levels(fa)	modalités de <i>fa</i>
codes(fa)	transforme <i>fa</i> en numérique

### Opérations combinées avec des facteurs

ave(x,fa)	moyenne du vecteur <i>x</i> par modalité de <i>fa</i>
split(df,fa)	sépare le vecteur <i>df</i> par modalité de <i>fa</i>
by(x,fa,median)	applique la fonction <i>median</i> par modalité de <i>fa</i> au data-frame <i>x</i> .

### Matrices

Les matrices possèdent 2 dimensions. Elles sont constituées d'un seul type avec ou non des NA (objet homogène).

matrix(1:2,1,2)	matrice à 1 ligne 2 colonnes
matrix(0,10,20)	matrice de 0 à 10 ligne 20 colonnes
as.matrix(1:10)	matrice unicolonne du vecteur 1:10
diag(1:10)	matrice diagonale d'ordre 10 dont les valeurs sont 1:10

diag(10)	identité d'ordre 10
----------	---------------------

### Sélection et matrices

x[c(6,1,9),]	affichage des lignes 6, 1 et 9
x[,c(6,1,9)]	affichage des colonnes 6, 1 et 9
x[1,-2]	affichage de la ligne 1 sans la colonne 2
x[x[,3]<2,]	affichage des lignes dont la colonne 3 a une valeur inférieure à 2

### Autres opérations (matrices)

length(x)	longueur de <i>x</i> , i.e. son nombre d'éléments
diag(x)	extraction de la diagonale de <i>x</i>
nrow(x)	nombre de lignes de <i>x</i>
ncol(x)	nombre de colonnes de <i>x</i>
dim(x)	dimension de <i>x</i>
cbind(x,y)	concaténation par colonne de <i>x</i> et <i>y</i>
rbind(x,y)	concaténation par ligne de <i>x</i> et <i>y</i>
qr(x)	décomposition QR de <i>x</i>
chol(x)	décomposition de Cholesky de <i>x</i>
svd(x)	décomposition en valeur singulière <i>x</i>
eigen(x)	valeurs propres et vecteurs propres de <i>x</i>
det(x)	déterminant de <i>x</i>
solve(x)	inversion de <i>x</i>
apply(x,1,sum)	somme par ligne de <i>x</i>
apply(x,2,var)	variance par colonne de <i>x</i>

### Listes

Les listes sont des objets hétérogènes composés d'une collection d'objets homogènes de longueur pas obligatoirement identique.

list(nomx=x, y)	listes à 2 éléments <i>x</i> de nom <i>nomx</i> et <i>y</i> sans nom.
as.list(1:10)	liste à 10 éléments de longueur 1

### Divers sur les listes

li[[2]]	extraction de la 2eme composante de la liste <i>li</i>
li\$nomx	extraction de la composante <i>nomx</i> de la liste <i>li</i>
dimnames(x) <- list(nomlign, nomcol)	noms des lignes et des colonnes de la matrice <i>x</i> .
lapply(li, f)	application de la fonction <i>f</i> à chacune des composantes de la liste
unlist(li)	transformation en vecteur de la liste

### Data-frame

Les data-frames sont des objets hétérogènes composés d'une collection d'objets homogènes de longueur identique.

data.frame(x)	Converti <i>x</i> en data-frame.
cbind.data.frame(df, df2)	Concatène par colonne.
names(df)	nom des colonnes
df[,1]	extraction de la colonne 1
df[[1]]	extraction de la composante 1 (colonne 1 en général)
df[, "nomcol"]	extraction de la colonne <i>nomcol</i>
summary(df)	résumé colone par colonne (min,max,moyenne etc...)

## Opérations sur les data-frame

<code>lapply(df,f)</code>	application de la fonction <i>f</i> à chacune des composantes de la liste
<code>apply(df,1,f)</code>	application de la fonction <i>f</i> à chacune des lignes
<code>aggregate(df,li,f)</code>	application de la fonction <i>f</i> à chacune des modalités de l'interaction des facteurs contenus dans la liste <i>li</i>

## Objets : question et conversion

<code>is.factor(x)</code>	renvoie un boolean; vrai si <i>x</i> est un facteur
<code>is.matrix(x)</code>	de même avec une matrice
<code>is.vector(x)</code>	de même avec vecteur
<code>as.factor(x)</code>	conversion explicite en facteur
<code>as.matrix(x)</code>	conversion explicite en matrice
<code>as.vector(x)</code>	conversion explicite en vecteur

## Entrée sortie

```
df <- read.table("c:/doc.txt",header=T,sep=" ")
# chargement d'un fichier (nombreuses options)
write.table(x,"c:/sortie.txt",row.names=F)
# exportation de l'objet x dans le fichier
Voir aussi scan(), write()
source("c:/fichiercommandes.txt")
# importe et execute les commandes contenues dans le fichier
```

## Programmation

```
for (i in vecteur) { listecommandes }
# boucle sur tous les éléments du vecteur
while (condition) { listecommandes }
# boucle tant que la condition est vraie
repeat { listecommandes }
# boucle infinie (couplée avec l'ordre break pour sortir de la boucle)
if (condition) {
  listecommandes
} else {
  listalternative }
# Structure conditionnelle
```

## Fonction

```
fonction <- function(a,b=1) {
  listecommandes
  return(nom1=resultat1,nom2=resultat2) }
# Fonction avec 2 arguments en entrée a et b. b a comme
# valeur par défaut 1 renvoyant une liste à deux composantes
# (nommées nom1 et nom2).
fonction(3,1) ou fonction(3)
res <- fonction(3,2)
# appel de la fonction
```

## Lois de probilités et nombres aléatoires

Pour avoir un quantile utiliser le préfixe **q**. Pour avoir une probabilité utiliser le préfixe **p**, pour avoir une densité utiliser le préfixe **d** et pour générer des nombres aléatoire utiliser le préfixe **r**. Lois disponibles : norm, poiss, bino, logis,...

`qt(0.95,12)` quantile à 95% d'une loi de student (t) (préfixe **q**) à 12 ddl

<code>ppois(3,2)</code>	probabilité cumulée jusqu'à 3 d'une loi de poisson (poiss) (préfixe <b>p</b> )
<code>df(3,6,30)</code>	densité au point 3 d'une loi de fisher (f) (préfixe <b>d</b> ) à (6,30) ddl
<code>rnorm(10,1,2)</code>	génération de 10 nombres aléatoires (préfixe <b>r</b> ) selon une loi normale (norm) de moyenne 1 d'écart type 2
<code>sample(10,1:30)</code>	tirage sans remise de 10 nombres parmi les coordonnées du vecteur 1:30

## Formules

<code>y~x1+x2</code>	y expliqué par <b>x1</b> et <b>x2</b>
<code>y~x1+x2+x1:x2</code>	y expliqué par <b>x1</b> et <b>x2</b> et leur interaction raccourci de la précédente
<code>y~x1*x2</code>	y expliqué par <b>x1</b> nidé dans <b>x2</b>
<code>y~x1%in%x2</code>	y expliqué par <b>x1</b> conditionnellement à <b>x2</b>
<code>y~x1 x2</code>	y expliqué par <b>x1</b> sans moyenne générale (intercept)
<code>y~I(x1*x2)</code>	y expliqué par la variable résultant du produit <b>x1x2</b> ; <b>I()</b> protège pour éviter l'interprétation comme <b>x1+x2+x1:x2</b>
<code>y~I(x1&gt;0)</code>	y expliqué par la variable binaire 0 quand <b>x1</b> est négatif ou nul, 1 sinon
<code>y~exp(x1*1932)</code>	y expliqué par la variable <b>x1</b> multiplié par 1932 et dont on en prend l'exponentielle

## Graphiques

<code>plot(x,y-options-)</code>	dessin des points aux abscisses contenues dans <i>x</i> et aux ordonnées dans <i>y</i> (voir les options ci-dessous et l'aide)
<code>matplot(x-options-)</code>	dessin des colonnes de la matrice (voir les options ci-dessous et l'aide)
<code>coplot(y~x1 x2, data=x -options-)</code>	dessin de <i>y</i> fonction de <b>x1</b> conditionnellement à <b>x2</b> (voir les options ci-dessous et l'aide)
<code>,lty=1 2 ...</code>	type de ligne (pointillé, continue...) -option-
<code>,pch=1 2 ... ou pch="o"</code>	type de point (rond...) -option-
<code>,col=0 1 2 ...</code>	numéro de couleur-option-
<code>,type="l" "p" "b" "n"</code>	tracé de ligne ou point ou les deux (both) ou rien (none)-option-
<code>abline(v=0)</code>	ajoute une ligne verticale en 0
<code>abline(h=0)</code>	ajoute une ligne horizontale en 0
<code>abline(0,1)</code>	ajoute une droite d'ordonnée à l'origine 0 et de coef. directeur 1
<code>text(1,1,"blabla")</code>	ajoute le texte "blabla" en (1,1)
<code>lines(x,y,col=2)</code>	ajout d'une ligne (brisée) de coordonnées (x,y) de couleur 2

Voir aussi `interaction.plot`, `stars`, `pairs`, `piechart`, `mtext`, `symbols`,...

## Analyses statistiques

`lm()` `aov()` `anova()` `glm()` `rpart()` `nlme()` `nls()`  
modèle linéaire, analyse de variance, modèle linéaire généralisé, arbres régression/segmentation, modèles mixtes, régression non-linéaire selon les librairies

```
resmodele <- lm(y~x1+x2+x1:x2,data=df)
summary(resmodele)
residuals(resmodele)
predict(resmodele)
plot(resmodele)
# exemple d'une régression linéaire, explication de y par x1
# x2 et leur interaction. Les données sont contenues dans le
# data-frame df. Vecteurs des résidus, de l'estimation de y par
# le modèle et dessin du modèle.
chisq.test(), fisher.test(), friedman.test(),
mantelhaen.test(), t.test(),...
```

## Librairies - packages

De nombreux programmes sont disponibles sur <http://cran.r-project.org> et sur ses sites miroirs.

`install.packages(tree)` Installe le package **tree** ; attention aux droits d'écriture sur votre disque...

`library()` liste des packages disponibles sur votre plateforme

## Chemin

R cherche tout objet dans des répertoires donnés dans un ordre donné. Cette liste ordonnée est appelée un chemin (path).

`search()` Liste du chemin

`attach(df,pos=2)` attache le data frame *df* en position 2 sur le chemin.

`detach(df)` détache le data frame *df*.

## Liste non exhaustive pour la programmation

`assign(x,pos=1)` assigne l'objet *x* dans l'environnement 1 du chemin

`parse(texte)` transforme le texte en expression (non évaluée)

`eval(expr)` évalue l'expression ; par exemple `eval(parse(text=texte))`

`deparse(expr)` transforme une expression en texte

`substitute(expr)` substitue dans une expression les valeurs des variables ; par exemple `deparse(substitute(x))`

`browser()` stoppe l'exécution et permet de fureter, utile pour déboguer

---

Edition 0.2 for R Version 1.6.2. 2003, P.A. Cornillon (pac@uhb.fr).  
Comme d'habitude, "the author assumes no responsibility for any errors on this card" et l'aide en ligne *a toujours raison*.

Roland Pesch (pesch@cygnus.com) a codé les macros T<sub>E</sub>X, originellement pour la carte de référence GDB et John W. Eaton (jwe@che.utexas.edu) les a modifiées pour Octave.

R itself is free software; you are welcome to distribute copies of it under the terms of the GNU General Public License. There is absolutely no warranty for R.