

# Aide Mémoire R

Pour les différents types d'éléments , nous utilisons les abréviations suivantes : **obj** un objet de R, **don** un data.frame, **fac** un facteur, **fich** fichier externe à R, **m** une matrice, **v** un vecteur

## Aide en ligne

**?topic**  
aide sur la fonction **topic**  
**help.search("topic")**  
chercher dans l'aide le mot **topic**  
**apropos("topic")**  
tous les objets dans l'aide contenant "topic" dans leur nom  
**example(topic)**  
exécute l'exemple dde la page d'aide sur la fonction **topic**

## Environnement

**options()**  
options globales de l'interprète, comme la largeur de l'affichage ou le nombre de chiffre significatifs  
**getwd()**  
le répertoire de travail  
**setwd()**  
changer le répertoire de travail  
**system.time()**  
temps d'évaluation d'une routine R  
**library("paquet")**  
chargement du paquet de fonctions additionnels  
**data(don)**  
chargement d'une table de données **don**  
**source("script.R")**  
execute le script R  
**save(x, y, file="fich.RData")**  
sauver des objets **x** y de R dans un fichier externe  
**load("fich.RData")**  
chargement d'une base de données de R issu d'un fichier externe

## Manipulation des objets

**obj <- expr**  
assignement de l'expression **expr** à **obj**  
**mode(obj)**  
le mode **obj**  
**length(obj)**  
le nombre d'éléments dans **obj**  
**class(obj)**  
la classe de **obj**  
**attributes(obj)**  
les attributs de **obj**  
**is.na(obj) is.null(obj) is.numeric(obj)**  
test la classe ou le type; pour une liste complète **methods(is)**  
**as.numeric(obj) as.data.frame(obj) as.ordered(obj)**  
convertit un objet selon la classe ou le type spécifié; pour une liste complète **methods(as)**  
**str(obj)**  
affiche la structure interne de **obj**

**summary(obj)**  
résumé du contenu de **obj**  
**print(obj)**  
affiche le contenu de **obj**  
**plot(obj)**  
représentation graphique du contenu de **obj**  
**methods(class=class(obj))**  
lister toutes les méthodes permettant de traiter les objets de la classe de l'objet **obj**  
**ls()**  
liste des objets  
**rm(obj)**  
effacer l'objet **obj**

## Les vecteurs atomiques

**v[n]** n<sup>e</sup> élément  
**v[-n]** tous *sauf* le n<sup>e</sup> élément  
**v[1:n]** les n premiers éléments  
**v[-(1:n)]** tous *sauf* les n premiers éléments  
**v[c(1,4,2)]** éléments 1, 4, 2  
**v["nom"]** élément nommé "nom"  
**v[x > 3]** tous les éléments plus grand que 3  
**v[x > 3 & x < 5]** tous les éléments compris entre 3 et 5  
**v[x %in% v1]** éléments appartenant à **v1**  
**c(v1, v2)**  
concaténation des 2 vecteurs **v1** et **v2**  
**names(v)**  
nom des éléments de **v**  
**rep(v, times)**  
réplique le vecteur **v** **times** fois

**1:n**  
génère une suite d'entiers allant de 1 à n  
**seq(from,to,by=)**  
génère une suite de **drom** à **to** en spécifiant le pas  
**seq(from,to,length=)**  
génère une suite de **drom** à **to** en spécifiant la longueur

## Les matrices

**m[i, j]** élément à la ligne **i**, colonne **j**  
**m[i,]** ligne **i**  
**m[, j]** colonne **j**  
**m[, c(1,3)]** colonnes 1 et 3  
**m["nom",]** ligne nommée "nom"  
**matrix(v,nrow=,ncol=)**  
construction d'une matrice à partir d'un vecteurs en précisant la dimension  
**rbind() cbind()**

**concaténation verticale et horizontale de matrices**  
**dim(m) nrow(m) ncol(m)**  
dimension, nombre de lignes, de colonnes de **x**;  
**dimnames(m)**  
noms des lignes et des colonnes de **m**  
**t(m)**  
matrice transposée  
**%\*%**  
multiplication matricielle  
**solve(m,b) solve(m)**  
résolution de **m %\*% x = b** en **x**, inverse de **m**  
**eigen(m)**  
valeurs propres, vecteurs propres  
**rowSums(m) colSums(m)**  
somme de chaque ligne, de chaque colonne  
**rowMeans(m) colMeans(m)**  
moyenne de chaque ligne, de chaque colonne  
**apply(m, 1, sum)**  
applique la fonction **mean** à chaque ligne

## Les listes

**x[[n]]** n<sup>e</sup> élément de la liste  
**x\$V1** élément de la liste dont le nommée "V1"  
**list(V1=obj1,V2=obj2)**  
construction d'une liste à partir de deux objets  
**names(x)**  
nom des objets de la liste

## Les data.frame

**data.frame(V1=v1,V2=v2)**  
construction d'un data.frame à partir de deux vecteurs  
Indexation : **id.** que les liste et les matrices  
**names(dat)**  
nom des variables du data.frame  
**rownames(dat)**  
nom des individus du data.frame  
**rbind() cbind()**  
concaténation verticale et horizontale de data.frame  
**merge(dat1,dat2,by=)**  
fusionne 2 data.frames en utilisant leurs noms de colonnes en commun  
**reshape(dat,...)**  
réorganisation en largeur et en longueur d'un data.frame

**read.table(fich)**  
importation d'un tableau de données à partir d'un fichier externe au format tabulaire  
sep=           séparateurs  
header=       première ligne comme nom des variables  
dec=           symbole pour les décimales  
na.strings=   codage des valeurs manquantes  
Autres paquets utiles : foreign xlsx

**read.csv2(fich)**  
importation d'un tableau de données à partir d'un fichier csv

**read.fwf(fich,widths=)**  
importation d'un tableau en spécifiant la largeur des colonnes

**write.table(dat, file=fich)**  
écrire dans un fichier externe un data.frame.

Les facteurs

**factor(v, levels=)**  
transforme un vecteur x en facteur

**levels(fac)**  
les niveaux du facteur fac

**nlevels(fac)**  
le nombre de niveaux du facteur fac

**cut(x,breaks)**  
découpe le numeric x en intervalle (facteur)

Manipulation des données

**head(don) tail(don)**  
premières et dernières lignes de la table

**v %in% v.ref**  
renvoie TRUE lorsque l'élément de v est dans v.ref

**subset(don, cond, select=c(V1, V2))**  
sélection des lignes de don vérifiant les conditions, et des colonnes d'intérêt

**which(v == 0)**  
renvoie les indices de v qui satisfont la condition, l'option arr.ind=TRUE permet de renvoyer les indices-couples lignes-colonnes lorsque v est une matrice.

**which.max(v), which.min(v)**  
indice du maximum (minimum) de v

**replace(v, ind, new.val)**  
remplace la valeur dans la variable v aux indices ind par la valeur new.val

**with(don, expr)**  
évaluer une expression faisant appel aux variables de don

**complete.cases(don)**  
indices des observations ne contenant pas NA

**rev(x)**  
inverse l'ordre des éléments de v

**sort(x)**  
tri les éléments de v dans l'ordre croissant;

**order(x)**  
indice des éléments de v triés dans l'ordre croissant;

**unique(x)**  
les modalités prise par v

**table(x)**  
les effectifs pour chaque modalité de v

**apply(don,margin,fun)**  
applique une fonction aux marges du tableau don

**lapply(liste,fun)**  
applique une fonction aux éléments de la liste list

**split(v,fac)**  
coupe une série statistique en sous séries selon le facteur

**by(v,fac,sum)**  
liste des sommes des valeurs de v par valeur de type fac

**aggregate(don,by=fac,sum)**  
tableau des sommes des valeurs des colonnes de don par valeur de type fac

**tapply(v,fac,sum)**  
matrice des sommes des valeurs de v par valeur de type fac

Autres packages utiles doBy

Math

**sin,cos,tan,asin,acos,atan,atan2,log,log10,exp**

**range(x) min(x) max(x)**  
minimum et maximum de x

**sum(x) prod(x)**  
somme et produits des éléments de x

**mean(x) median(x)**  
moyenne et médiane des éléments de x

**rank(x)**  
rangs des éléments de x

**var(x) sd(x)**  
variance et écart-type de x

**cov(x, y) cor(x, y)**  
covariance et corrélation linéaire entre x et y

**round(x, n)**  
arrondis les éléments de x à n décimales

**cumsum(v) cumprod(v)**  
sommes et produits cumulées des éléments de x

**cummin(v) cummax(v)**  
minima et maxima cumulés des éléments de x

**union(x,y) intersect(x,y) setdiff(x,y) setequal(x,y)**  
**is.element(e1, set)**  
opération d'ensembles sur les vecteurs

**convolve(x,y)**  
convolution entre deux suite de vecteurs

Chaînes de caractères

**paste(...)** **cat(...)** **sprintf(...)**  
différentes façon de concaténer des vecteurs après conversion en caractères

**substr(v,start,stop)**  
extrait une sous-chaîne de caractères de v

**grep(pattern,v)**  
renvoie les indices des éléments de v dans lesquels on trouve le patron pattern

**tolower(v) toupper(v)**  
met en minuscules, en majuscules v

**nchar(v)**  
nombre de caractères

Fonctions graphiques principales

**plot()**  
fonction générique pour représenter les objets  
add=           superpose les graphiques  
axes=           affichage des axes  
type=           type du tracé  
pch=,lty=       symboles, lignes  
col=,cex=       couleur, épaisseur  
xlim=, ylim=   bornes des axes  
xlab=, ylab=   légendes des axes  
main=           titre

**plot(x,y)**  
nuage de points, plus de détail avec l'aide ?plot.default

**sunflowerplot(x, y)**  
idem que plot(x,y) mais les points qui se superposent exactement sont représentés avec des pétales de fleurs

**hist(x)**  
histogramme des effectifs de x

**barplot(x)**  
diagramme en barres de x

**mosaicplot(x)**  
représentation graphique d'un tableau de contingence

**image(x)**  
représentation graphique d'un tableau de contingence, la valeur des effectifs modulent la couleur

**pie(x)**  
diagramme circulaire

**boxplot(x)**  
diagramme de Tukey

**stripchart(x)**  
superpose les valeurs répétées de x sur une ligne

**coplot(x~y | z)**  
nuage de points de x et y conditionnellement à z

**pairs(tab)**  
tableau des nuages de points des colonnes de tab

**ts.plot(x)**  
représentation de séries temporelles multivariées observées à la même fréquences, mais dont le départ peut être différent

**qqplot(x,y) qqnorm(x)**  
graphe quantiles-quantiles de x et y, ou de x associé à la loi normale

**contour(x, y, z)**  
lignes de niveau de la matrice z selon les coordonnées x et y

**image(x, y, z)**  
id. avec une échelle couleur

**persp(x, y, z)**  
id. mais en perspectives

Fonctions graphiques secondaires

**points(x, y) lines(x, y)**  
ajoute des points/lignes (utiliser type= pour spécifier le type de tracé)

**rug(v)**  
ajoute le tapis des valeurs sur une des axes

**text(x, y, labels, ...)**  
ajoute du texte labels aux coordonnées (x,y)

**abline(a,b)**  
ajoute une droite affine de pente b et d’intercept a

**abline(h=y) abline(v=x)**  
ajoute une ligne horizontale/verticale

**segments(x0, y0, x1, y1), arrows(x0, y0, x1, y1)**  
ajoute des segments/flèches allant de (x0,y0) à (x1,y1)

**polygon(x, y)**  
colorie un polygone dont les sommets sont définis dans l’ordre par les coordonnées x et y

**legend(x, y, legend)**  
ajoute une légende aux coordonnées (x,y)

**axis(side, vect)**  
spécifier les axes du graphiques (en bas (side=1), à gauche (2), en haut (3), à droite (4))

Fonctions graphiques interactives

**locator(n, type="n", ...)**  
retourne les coordonnées (x,y) après avoir cliquer n fois sur la fenêtre graphique avec la souris

**identify(x,y, ...)**  
obtenir d’un simple clic l’identité de l’individu à partir d’un nuage de points

Options graphiques

**par(...)**  
définit les paramètres graphiques à venir, pour plus de détails ?par  
col couleur du tracé, voir colors()  
rgb(), hsv(), gray(), et rainbow()  
font= style du texte  
mar= marge en ligne du graphe dans la figure  
mfcol=, mfrow= partition de la fenêtre graphique

**layout(...)**  
partitionne la fenêtre graphique en spécifiant les largeurs et les hauteurs de chaque case, et l’ordre de remplissage

Statistique inférentielle

**optim(p, fun, ...)** **nlm(fun,p)**  
minimisation de la fonction fun par un algorithme itératif à choisir; p est le point d’initialisation

**lm(y~x)**  
régression linéaire de y avec x

**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

**?Distributions**  
fonction de probabilités disponibles, elle comprennent pour chaque

loi un générateur aléatoire **rloi**, la fonction de répartition **ploi**, la densité **dloi**, la fonction quantile **qloi**.

Programmation

**function( arglist ){ expr; return(value) }**  
définition d’une fonction

**if(cond) expr**  
**if(cond) cons.expr else alt.expr**  
**ifelse(test, yes, no)**  
**for(var in seq) expr**  
**while(cond) expr**  
**repeat expr**  
**break**  
**next**

**== != < > <= >=**  
opérateurs de comparaisons

**| & ! all() any() xor()**  
opérateurs logiques OU ET NOT

**do.call(fun, args)**  
exécute une fonction avec la liste des arguments à entrer à l’intérieur

**source("fich.R")**  
exécute un script

KnitR

Le format du fichier est .Rnw. C’est essentiellement un fichier  $\LaTeX$ entrecoupé de blocs de code R. Les figures générées sont enregistrées par défaut dans le sous répertoire figure. Le nom du fichier généré correspond au label du bloc. Le label de la figure est identique au label du bloc de code Le code  $\LaTeX$ associé aux tableaux peut être généré automatiquement via le paquet xtable Le label de la table est identique au label du bloc de code

**<< \* >>=**  
marque le début du bloc de code; \* est la liste des options contrôlant l’affichage et l’exécution du bloc de code. Le label du bloc de code est foo (optionnel). Certaines options du chunk sont propres aux figures. Le signe @ délimite fin du bloc de code.

**eval=**  
option indiquant si le bloc de code doit être évalué

**echo=**  
option indiquant si le bloc de code doit être affiché

**results=**  
option indiquant comment le résultat doit être affiché  
‘hide’ pas d’affichage du résultat  
‘verbatim’ affichage comme dans la console R (par défaut)  
‘asis’ affichage dans le document de sortie

**include=**  
option indiquant si le bloc doit être inclu dans le document

**fig.width=, fig.height=**  
largeur et hauteur de la figure générée

**out.width=, out.height=**  
largeur et hauteur de la figure dans le document

**fig.path=**  
indique le répertoire dans lequel sera sauvegardé la figure

**dev=**  
indique le device utilisé

**fig.align=**  
indique la justification de la figure dans le document

**fig.cap=**  
indique le titre de la figure dans le document

**\Sexpr()**  
insérer la valeur d’une expression R dans le texte

**xtable(monTableau, caption="titre")**  
génère le code  $\LaTeX$ associé à l’objet monTableau. Pour que le tableau soit inséré tel quel dans le document, il faut spécifier l’option results=‘asis’ pour le bloc de code associé.

```
\documentclass{article}
\begin{document}
Un bloc de codes R ou chunk
<<foo, echo=TRUE>>=
z <- 1+1
@
La valeur de z est \Sexpr{z}.
\end{document}
```