

Aide-mémoire de programmation R et d'expressions régulières

Opérateurs (en ordre décroissant de priorité)

\$	extraction d'une liste
[[[indilage
^	puissance
-	changement de signe
:	génération de suites
%% %/% %0%	produit matriciel, modulo, division entière, produit extérieur
* /	multiplication, division
+ -	addition, soustraction
< <= == >= > !=	plus petit, plus petit ou égal, égal, plus grand ou égal, plus grand, différent de
!	négation logique
& &&	« et » logique
	« ou » logique
-> ->>	affectation
<- <-<	affectation

Indilage

x[⟨entiers positifs⟩] éléments aux indices ⟨entiers positifs⟩

x[⟨entiers négatifs⟩] tous les éléments sauf ceux aux indices ⟨entiers négatifs⟩

x[⟨condition⟩] éléments satisfaisant la ⟨condition⟩ (vecteur booléen)

x[⟨chaines⟩] éléments étiquetés ⟨chaines⟩

x[i,] lignes d'une matrice

x[, j] colonnes d'une matrice

x[i, ,] tranches horizontales d'un tableau

x[, j,] tranches verticales d'un tableau

x[, , k] tranches transversales d'un tableau

x[i, j,] carottes horizontales d'un tableau

x[i, , k] carottes transversales d'un tableau

x[, j, k] carottes verticales d'un tableau

x[[i]] élément i (unique) d'une liste

x\$nom élément étiqueté nom (unique) d'une liste

Extraction

head(x, n) n premiers éléments de x (ou x sans les n derniers éléments si n < 0)

tail(x, n) n derniers éléments de x (ou x sans les n premiers éléments si n < 0)

Tests logiques

is.na(x) TRUE pour les éléments NA

all(x) TRUE si tous les éléments sont TRUE

any(x) TRUE si au moins un élément est TRUE

Arrondi

round(x) arrondi

floor(x) arrondi à l'entier inférieur

ceiling(x) arrondi à l'entier supérieur

Sommaires pour les vecteurs

sum(x), prod(x) somme et produit des éléments de x

mean(x), var(x) moyenne et variance des éléments de x

min(x), max(x) minimum et maximum de x

diff(x) différences x[2] - x[1], x[3] - x[2], etc.

cumsum(x), cumprod(x) somme et produit cumulatif

cummin(x), cummax(x) minimum et maximum cumulatif

pmin(x, y), pmax(x, y) minimum et maximum élément par élément entre x et y

Sommaires pour les matrices

nrow(x), ncol(x) nombre de lignes et de colonnes

rowSums(x), colSums(x) somme par ligne et par colonne

rowMeans(x), colMeans(x) moyenne par ligne et par colonne

Manipulation de chaines de caractères

paste(..., sep) chaines de caractères concaténées, séparées par défaut par des espaces

paste0(...) chaines de caractères concaténées, sans séparateur

nchar nombre de caractères dans une chaîne

substr(x, start, stop) sous-chaines allant du caractère en position start à celui en position stop

strsplit(x, split) chaîne x séparée en liste de sous-chaines selon le caractère split

Suites et répétition

seq(from, to, by, length.out, along.with) suites générales

seq_along(along.with) suite de 1 à la longueur de along.with

seq_len(length.out) suite de 1 à length.out

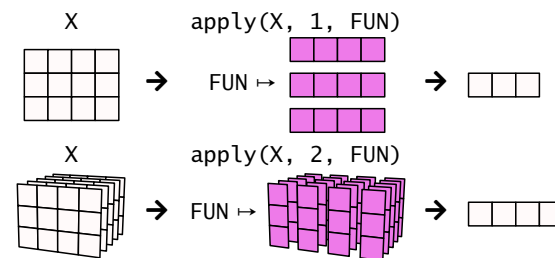
rep(x, times, length.out, each) répétition d'un vecteur

rep.int(x, times) répétition de x en entier times fois

rep_len(x, length.out) répétition de x jusqu'à un vecteur de longueur length.out

Application pour les matrices et les tableaux

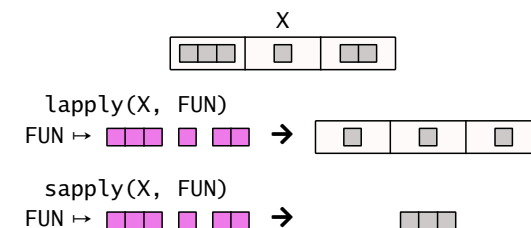
apply(X, MARGIN, FUN, ...) applique la fonction FUN sur les dimensions MARGIN de la matrice ou du tableau X



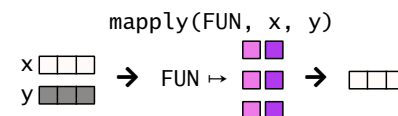
Application pour les listes et les vecteurs

lapply(X, FUN, ...) applique la fonction FUN à chaque élément de la liste ou du vecteur X et retourne le résultat sous forme de **liste**

sapply(X, FUN, ...) applique la fonction FUN à chaque élément de la liste ou du vecteur X et retourne le résultat sous forme de **vecteur**, si possible

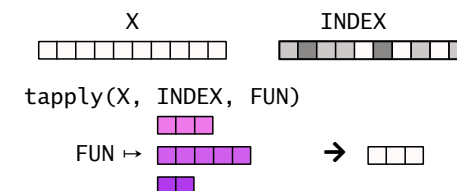


mapply(FUN, ...) applique FUN au vecteur des premiers éléments des arguments, au vecteur des deuxièmes éléments, etc.



Application pour les groupes de données

tapply(X, INDEX, FUN, ...) applique la fonction FUN à chacun des groupes du vecteur X définis par le facteur INDEX



Produit extérieur

outer(X, Y, FUN) FUN(X[i], Y[j]) pour toutes les combinaisons des éléments de X et Y

x %o% y raccourci de outer(x, y, "%")

Tri et recherche

sort(x) x trié en ordre croissant (ou décroissant avec decreasing = TRUE)

rank(x) rangs des éléments de x dans l'ordre croissant

order(x) ordre d'extraction des éléments pour trier x en ordre croissant (ou décroissant avec decreasing = TRUE)

rev(x) x en ordre inverse

unique(x) éléments uniques de x

match(x, table) indices des premières occurrences de x dans table

x %in% table TRUE si x se trouve dans table

which(x) indices des éléments TRUE

which.min(x) indice du minimum

which.max(x) indice du maximum

Structures de contrôle et boucles

if ((cond)) <expr> else <alt> évaluer <expr> si <cond> est TRUE et <alt> sinon

ifelse(test, yes, no) vecteur formé des éléments de yes ou no selon que l'élément de test est TRUE ou FALSE

return(value) retourner value et quitter la fonction

stop(...) signaler une erreur et quitter la fonction

warning(...) signaler un avertissement

for ((variable) in <suite>) <expr> boucle à dénombrement

while ((cond)) <expr> boucle à précondition

repeat <expr> boucle à condition d'arrêt

break force la sortie de la boucle

next force le passage à la prochaine itération

Recall(...) rappelle la fonction courante (récursion)

Bibliothèques et paquetages

library(package) charge package dans la session de travail

install.packages(pkgs) installe pkgs à partir de CRAN dans la bibliothèque par défaut

update.packages() met à jour les paquetages de toutes les bibliothèques

.libPaths() liste des bibliothèques

Tests unitaires

source(file) évaluer file en entier

stopifnot(...) arrêter si les expressions ne sont pas toutes TRUE

identical(x, y) TRUE si les objets sont identiques

all.equal(x, y) TRUE si x et y sont « presque égaux »

Importation et exportation

scan(file) données brutes du fichier file sous forme de vecteur simple

read.table(file) données en tableau du fichier file sous forme de tableau de données

read.csv(file) données en format CSV du fichier file sous forme de tableau de données

readRDS(file) données sérialisées du fichier file sous forme d'objet R

write(x, file, ncolumns) exporte le vecteur x sous forme de matrice de ncolumns colonnes

write.table(x, file) exporte le tableau de données x sous forme de tableau

write.csv(x, file) exporte le tableau de données x en format CSV

saveRDS(object, file) exporte l'objet object sous forme de données sérialisées

Analyse et contrôle de texte

grep(pattern, x) indices des éléments de x qui correspondent au motif d'expression régulière pattern

grepl(pattern, x) comme grep mais sous forme de vecteur booléen

sub(pattern, replacement, x) remplace dans chaque élément de x la première correspondance du motif d'expression régulière pattern par replacement

gsub(pattern, replacement, x) comme sub mais pour toutes les correspondances

Expressions régulières : opérateurs de base

	alternance (« ou »)
^	début de la ligne ; négation (à l'intérieur d'une classe)
\$	fin de la ligne
.	caractère de remplacement
*	zéro, une ou plusieurs occurrences
+	une ou plusieurs occurrences
?	zéro ou une occurrence
()	groupe de caractères
[]	classe de caractères
{ }	quantificateur du nombre d'occurrences
\	caractère d'échappement

Expressions régulières : outils

Dans les définitions de syntaxe ci-dessous, les éléments entre crochets [] sont optionnels.

grep [-Ecov] '<motif>' <fichiers>
affiche les lignes de <fichiers> qui correspondent à <motif>
-E motif est une expression régulière étendue
-c nombre de lignes qui correspondent au motif
-o portion de la ligne qui correspond au motif
-v lignes qui ne correspondent pas au motif

sed [-E] 's/<motif>/<remplacement>/[g]' <fichiers>
remplace dans chaque ligne de <fichiers> la première occurrence qui correspond à <motif> par <remplacement>
-E motif est une expression régulière étendue
g remplace toutes les occurrences sur une ligne

awk '<motif> { <action> } ...' <fichiers>
sépare chaque enregistrement de <fichiers> en champs puis exécute tour à tour <action> (print par défaut) quand un enregistrement correspond à <motif>

Variables automatiques

\$0, \$1, \$2, ... contenu de l'enregistrement complet, du premier champ, du deuxième champ, ...

NR numéro de l'enregistrement courant

NF nombre de champs dans l'enregistrement courant

FS séparateur de champs lors de la lecture

OFS séparateur de champs lors de l'écriture

Motifs

/<regex>/ correspond lorsque <regex> correspond
<expression> correspond lorsque <expression> est non nulle ou non vide

BEGIN correspond avant la lecture du premier enr.

END correspond après la lecture du dernier enr.

Opérateurs et fonctions

print <chaine> affiche <chaine> (\$0 par défaut)

<chaine> ~ /<regex>/ vrai lorsque <chaine> correspond à <regex>

<chaine> !~ /<regex>/ vrai lorsque <chaine> ne correspond pas à <regex>

<cond> ? <expr> : <alt> exécute <expr> si <cond> est vraie et <alt> sinon

Ligne de commande Unix

<commande 1> | <commande 2> (transfert de données ou tuyau) envoyer la sortie de <commande 1> en entrée à <commande 2>

<commande> > <fichier> (redirection) envoyer la sortie de <commande> dans <fichier>

<commande> < <fichier> (redirection) envoyer le contenu de <fichier> en entrée à <commande>