

# Notes de cours R

Thibaut Marmey

October 16, 2018

## Contents

<b>1</b>	<b>Programmation R</b>	<b>1</b>
1.1	Généralité . . . . .	1
1.2	Fonctions . . . . .	2
1.3	Manipulation de chaînes de caractères . . . . .	2
1.4	Gestion de données complexes . . . . .	2

## 1 Programmation R

### 1.1 Généralité

- *lien internet* : doc d'installation de R et Rstudio
- Utilisation de la documentation "aide" : *help()* ou *help("nom de la fonction")* ou *?log*
- Documentation plus détaillée sur internet : *help.start()*
- Interaction avec l'environnement R :
  - liste toutes les variables créées : *ls()*
  - liste des variables avec une succession de lettre particulière : *ls(pattern="mot")*
  - supprimer des variables : *rm(var)*
  - quitter le travail : *q()* ou *quit()*
- Ecrire des commentaires avec "#"
- Utiliser la fonction *print()* pour afficher les informations, textes sur la console
- Enregistrer les variables que l'on veut : *save(var1, var2, ..., file = "nameFile.extension")*.  
Enregistrement de ces fichiers dans le répertoires associés *Tools/Global Options*, ici c'est Documents/Notes-de-cours/Cours R/Test
-

## 1.2 Fonctions

- Tester le type d'une variable :
  - *is.character(var)*
  - *is.numeric(var)*
  - *is.logical(var)*
- Spécifier le typage de la variable :
  - *as.numeric(var)*
  - *as.logical(var)*
  - *as.character(var)*
- Renvoyer l'entier inférieur : *floor(nb)*
- Renvoyer l'entier supérieur : *ceiling(nb)*
- Arrondir à l'entier le plus proche : *round(nb)*
- fonction trigo : *cos(angleRad)*, *sin*, ...

## 1.3 Manipulation de chaînes de caractères

- Saisir des données depuis le clavier : *scan()*. Pour valider l'input appuyer sur *entrée*, et un nouvel input apparaitra. Pour arrêter la prise d'input appuyer directement sur *entrée*.
- Rentrer le nombre d'input avec l'appel de *scan(nmax=n)*
- La concaténation de texte et nombre avec *paste(texte, nombres, ...)*
- Combinaison de *scan()* et *paste()* : *paste('Bonjour j'ai ', scan(nmax=1), ' ans')*
- Permet de compter le nombre de lettres présentes dans la chaîne de caractères : *nchar("chaîne")*
- Mettre les caractères en majuscules ou minuscules : *toupper()*, *tolower()*
- Extraire une sous chaîne de caractères avec point de départ et d'arrivée inclus : *substr("chaîne", start, stop)*. La chaîne de caractères commence à 1.

## 1.4 Gestion de données complexes

- Le vecteur : élément de base du langage, c'est une liste d'éléments étant tous du même type.
  - Spécifier deux attributs : le type de ces éléments et la longueur du vecteur (nb d'éléments)
  - Les fonctions prennent toutes des vecteurs en paramètre. Elles renverront le résultat sous la forme d'un vecteur de longueur (généralement) égale à la longueur du vecteur d'entrée.

- Différentes méthodes pour créer un vecteur :
  - \* la fonction *vector(type, length)*  
*ex : vector("numeric", 10)* crée un vecteur de 10 éléments tous égaux à 0.  
 Les valeurs par défaut sont 0 (numeric), "" (character), FALSE (logical).
  - \* les fonctions *numeric(length)*, *character(length)*, *logical(length)*
- Générer des séries de nombres
  - \* Suite d'un nombre à un autre : *nb1:nb2*
  - \* Suite d'un nombre répété : *rep(element, length)*
  - \* Séquence de nombres : *seq(start, step, stop)*
- 
- Concaténer plusieurs vecteurs (ils doivent contenir les mêmes types de variables) :  
*c(vect1, vect2, ...)* (bonne méthode pour créer rapidement un vecteur avec des valeurs directement attribuées ex: *c(70, 50, 10, 0, 5)*)
- Nommer les éléments du vecteur :  
*names("nom1", "nom2", rep(NA,2), "nom3", ...)* (ici permet de nommer des éléments et d'en laisser 2 sans nom)
- Indexation numériques : travailler seulement sur une sous partie d'un vecteur. Les éléments des vecteurs sont indexés de 1 jusqu'au dernier.  
 Accéder à un élément du vecteur via son index : *vect1[index]*