

# Introduction au Logiciel R

Nicolas PASQUIER  
Université Côte d'Azur  
Département Informatique  
Laboratoire I3S (UMR-7271 UCA/CNRS)  
<http://www.i3s.unice.fr/~pasquier>



## Présentation du Logiciel R

---

- R est un environnement intégré libre (gratuit) pour manipuler, analyser, générer des graphiques et des modèles à partir de données
- Deux modes d'utilisation
  - Mode interactif : commande saisie et exécutée immédiatement
  - Mode programmation : fonctions, structures de contrôle et itératives, etc.
- Forces du logiciel R
  - Langage très peu verbeux qui permet un développement très rapide
  - Documentations en ligne très complètes
  - Très grands nombre de librairies disponibles, le plus complet actuellement
- Langage interprété qui ne requiert aucun autre logiciel
  - Basé sur la notion de vecteur ce qui simplifie les calculs et limite l'utilisation de structures itératives (boucles)
  - Pas de typage ni de déclaration des variables nécessaires

## Session de Travail R

- Le répertoire de travail est déterminé au lancement de R

```
> setwd("C:/Documents/Exercices") # Set working directory
> getwd() # Affichage
[1] "C:/Documents/Exercices"
```

- Dans ce répertoire seront enregistrés

- Fichier .Rhistory : liste des commandes interactives exécutées durant la session
- Fichier .Rdata : environnement (variables, tableaux, etc.) créés durant la session
- Fichiers nomFichier.R : scripts R (suite de commandes R) enregistrés durant la session

- Quitter R

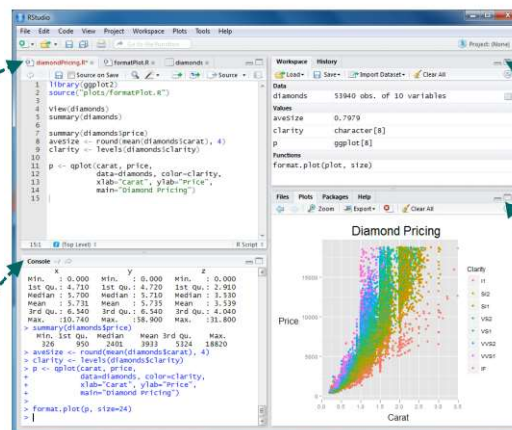
```
> q()
Sauver une image de la session ? [y/n/c] :
```

## L'Environnement de Développement Intégré R Studio

- R Studio permet de simplifier certaines manipulations par son interface graphique (e.g. gestion du répertoire de travail et chargement d'un fichier de données via un explorateur de fichier, installation de librairies, mises à jour, etc.)

Afficheur de code  
(scripts R) et  
d'objets textuels de  
l'environnement de  
travail (tableaux,  
vecteurs, etc.)

Console : saisie de  
commandes  
interactives et  
affichage immédiat  
du résultat



Gestionnaire de  
l'environnement de  
travail :  
informations sur les  
objets en mémoire,  
historiques des  
commandes, etc.

Gestionnaire de  
fichiers, affichage  
de l'aide, affichage  
des graphiques,  
gestion des  
librairies

- Logiciel libre, multi-plateformes, en versions installée ou portable (e.g. clé USB)

## Commandes R : Affectations et Expressions

- Toute commande R est soit une expression, soit une affectation

- Une expression est immédiatement évaluée

```
> pi                # Affichage, équivalent à print(pi) ou (pi)
[1] 3.141593         # Résultat
> 3 * 4             # Calcul
[1] 12               # Résultat
```

- Lors d'une affectation le résultat est stocké dans un objet

```
> x <- 5            # Affectation, création si l'objet n'existe pas
> x                 # Affichage, équivalent à print(x) ou (x)
[1] 5               # Résultat
```

- Les données sont stockées en mémoire sous forme de vecteurs

- Vecteur : liste d'éléments tous du même type (appelé mode en R)
- Variable : vecteur de taille 1 ; Liste de N éléments : vecteur de taille N ; Matrice de N x M éléments : N vecteurs de M éléments chacun

## Les Quatre Types de Variables de R Base

- character : une chaîne de caractères, délimitée par des guillemets « " »
  - Fonctions usuelles de manipulation : paste(), grep(), etc.
- logical : une valeur booléenne TRUE (en abrégée T) ou FALSE (en abrégé F)
  - Opérateurs logiques usuels : !, &, |, xor() (OU exclusif)
- numeric : un nombre qui peut être integer (entier) ou double (réel)
  - Opérateurs arithmétiques : +, -, \*, /, ^, %% (modulo), %/% (division entière)
- complex : nombre complexe sous forme algébrique (parties réelle et imaginaire) ou polaire (module et argument)
- R Base fournit des fonctions standards de test et conversion
  - Tester le type (mode) d'un vecteur : is.character(), is.logical(), etc.
    - Renvoie TRUE ou FALSE
  - Convertir un vecteur : as.character(), as.logical(), etc.
    - Renvoie un vecteur du type demandé, si la conversion est possible



# Principales Manipulations des Vecteurs

- La fonction `c()` concatène ses arguments pour constituer un vecteur

```
> v1 <- c(1, 3, 5, 7, 11)
> v1
[1] 1 3 5 7 11
```

- Si tous les éléments ne sont pas du même type, R force le type par coercion

```
> v2 <- c(1, 2.0, T, 2+2, "cinq")
> v2
[1] "1"      "2"      "TRUE"    "4"      "cinq"
> mode(v2)
[1] "character" # Type des valeurs stockées dans v2
```

- L'opérateur `[ ]` permet d'accéder aux valeurs d'un vecteur par leur indice

```
> v1[2] # 2e valeur du vecteur v1
[1] 3
> v1[2:4] # Valeurs d'indices 2 à 4
[1] 3 5 7
```

# Intérêt des Vecteurs : Simplification du Développement

- La traitement d'une seule valeur ou d'une liste de valeurs est identique

- Permet d'éviter le recours aux structures itératives (boucles)

```
> v1 <- c(1,2,3,4)
> for (i in 1:length(v1)) { # Itération sur les 4 éléments
+   print(sqrt(v1[i]))      # Affichage de la racine carrée
+ }
[1] 1
[1] 1.414214
[1] 1.732051
[1] 2
> sqrt(v1) # Application de sqrt() aux 4 éléments
[1] 1.000000 1.414214 1.732051 2.000000
```

- Réduit considérablement les temps de développement

- Autorise le prototypage et des tests rapides par les commandes interactives, et simplifie la manipulation du résultat

# Les Structures de Données de R Base

- R Base fournit des structures (appelées classes) d'éléments multiples

Classe	Objet	Modes	Multi-modes?
vector	Tableau uni-dimensionnel	num, char, comp ou log	Non
matrix	Tableau bi-dimensionnel	num, char, comp ou log	Non
array	Tableau n-dimensionnel	num, char, comp ou log	Non
data.frame	Tableau bi-dimensionnel	num, char, comp ou log	Oui
factor	Valeurs nominales (répertoire les valeurs possibles)	num, char	Non
ts	Série temporelle	num, char, comp ou log	Non
list	Liste d'objets	num, char, comp, log, vector, list, etc.	Oui

- R Base fournit des fonctions standards de test et conversion de classe
  - Tester la classe d'un objet : `is.vector(x)`, `is.factor(x)`, etc.
    - Renvoie TRUE ou FALSE
  - Convertir un objet vers une autre classe : `as.vector(x)`, `as.factor(x)`, etc.
    - Renvoie un objet de la classe demandée, si la conversion est possible

## Data Frame : Matrice de Données Hétérogènes

- Chaque colonne est un vecteur (attribut ou variable)
- Les colonnes peuvent être de modes (type de valeurs) différents

```
> v1 <- c("A", "B", "C")
> v2 <- c(3.2, 4.1, 2.7)
> d1 <- data.frame(v1, v2)      # Fonction constructeur
> d1
  v1    v2
1  A  3.2
2  B  4.1
3  C  2.7
> class(d1)                    # Affichage du type d'objet
[1] "data.frame"
```

- Type central en R pour la manipulation de fichiers de données

```
> mydata <- read.csv("C:/data.csv", header=TRUE, sep="\t", dec=",",
+ stringsAsFactors = TRUE)
```

## Accès aux Lignes et Colonnes des Data Frames

- L'opérateur de sélection [ , ] permet d'accéder aux colonnes par leur indice

```
> d1[,2]                # 2ème colonne  
[1] 3.2 4.1 2.7          # Vecteur résultat
```

- L'opérateur de sélection [ , ] permet d'accéder aux colonnes par leur nom

```
> d1[, "v2"]             # Colonne de nom « v2 »  
[1] 3.2 4.1 2.7
```

- L'opérateur \$ permet d'accéder à une colonne par son nom

```
> d1$v2                  # Colonne de nom « v2 »  
[1] 3.2 4.1 2.7
```

- L'opérateur de sélection [ , ] permet d'accéder aux lignes par leur indice

```
> d1[3,]                 # 3ème ligne  
  v1  v2                  # Data frame résultat  
3   C  2.7
```

- Listes de valeurs : d1[2:3], d1[,c("v1","v2")], d1[1:3,]

## Accès aux Valeurs dans un Data Frame

- L'opérateur de sélection [lignes, colonnes] permet d'accéder au contenu des cellules

```
> d1[3,2]                # Cellule de la 3ème ligne et 2ème colonne  
[1] 2.7
```

- Sélection de lignes : conditions dans le sélecteur de lignes

```
> d1[d1$v1 == "B", ]      # Ligne contenant "B" pour v1  
  v1  v2  
2   B  4.1
```

- Sélection de colonnes : conditions dans le sélecteur de colonnes

```
> d1[d1$v2 < 4, c("v1","v2")] # Colonnes « v1 » et « v2 »  
  v1  v2  
1   A  3.2  
3   C  2.7
```

## Modification des Valeurs dans un Data Frame

- L'opérateur \$ permet de créer une nouvelle colonne et la nommer

```
> d1$v3 <- c(T,F,T)      # Création d'une colonne « v3 »
> d1
  v1 v2 v3
1  A 3.2 TRUE
2  B 4.1 FALSE
3  C 2.7 TRUE
```

- L'opérateur [, ] permet de supprimer des colonnes et lignes

```
> d2 <- d1[, -3]          # Suppression de la 3ème colonne
> d2
  v1 v2
1  A 3.2
2  B 4.1
3  C 2.7
> d2 <- d2[-3,]           # Suppression de la 3ème ligne dans d2
```

## Représentation des Valeurs Manquantes

- Les valeurs manquantes doivent être représentées par la valeur NA (Not Available)

```
> d1$v4 <- c(10,NA,30)    # Création de v4 avec valeur manquante
> d1
  v1 v2 v3 v4
1  A 3.2 TRUE 10
2  B 4.1 FALSE NA
3  C 2.7 TRUE 30
> is.na(d1[,4])           # Teste si la valeur est manquante
[1] TRUE
```

- La valeur NA peut être manipulée comme les valeurs non manquantes

```
> sqrt(d1[,4])           # Racines carrées des valeurs de v4
  v4
1 3.162278
2    NA
3 5.477226
# Résultat si le calcul est impossible
```



## Les Librairies R

---

- Permettent d'étendre les possibilités offertes par R Base
- Téléchargement et installation d'une librairie (package)  

```
> install.packages("maps")           # Installation de la librairie
```
- Chargement d'une librairie pour son utilisation dans la session R courante  

```
> library(maps)                       # Activation de la librairie
```
- Obtenir la liste des librairies installées sur la machine  

```
> library()
```
- Format détaillé (chemin d'installation, numéro de version, etc.)  

```
> installed.packages()
```
- La fonction `RSiteSearch()` permet d'effectuer des recherches dans les listes de diffusion, archives, manuels et pages d'aide officiels de R  

```
> RSiteSearch("association rules")
```

## Ressources en Ligne pour le Développement en R

---

- La recherche d'informations dans les documentations, aides et forums (FAQ) sur Internet est un élément central du développement en R
  - CRAN – The R Manuals <https://cran.r-project.org/>
  - R Documentation <https://www.rdocumentation.org/>
  - Quick-R – Quickly access R <http://www.statmethods.net/>
  - Developpez.com – Rubrique R <http://r.developpez.com/>
  - Stack Overflow <http://stackoverflow.com/>
  - Forum francophone du logiciel R <http://forums.cirad.fr/logiciel-R/>
  - Programmation R <http://www.duclert.org/>
- Identifier la (ou les) librairie(s) R contenant la fonction adaptée au traitement à réaliser
- Identifier la fonction à utiliser et les valeurs de ses paramètres à fournir
- Identifier des exemples d'utilisations similaires ou proches



# Références et Bibliographie

---

- Sites Internet

- The R Project for Statistical Computing. Distributions Linux, Mac OS, Windows et Android de R. <https://www.r-project.org/>
- CRAN : The Comprehensive R Archive Network. Documentations officielles, librairies et ressources pour R. <https://cran.r-project.org/>
- R Studio : Environnement de Développement Intégré pour R. <https://www.rstudio.com/>

- Bibliographie

- [R pour les Débutants](#). Emmanuel Paradis. CRAN, 2005
- [Introduction à R](#). Julien Barnier. CRAN, 2013
- [Aide Mémoire R](#). Mayeul Kauffmann. CRAN, 2009
- [Introduction à la Programmation en R](#). Vincent Goulet. CRAN, 2016
- R in a Nutshell – A Desktop Quick Reference. Joseph Adler. O'Reilly, 2010. ISBN 978-1-449-31208-4