

R : Introduction à R — Analyse R | Cours 1

Ményssa Cherifa-Luron

2023-09-15

Contents

Introduction	1
Organisation	2
Ressources	2
Les bases de la programmation R	2
Qu'est-ce que R ?	2
Concepts de base de l'analyse des données.	2
Installation de R et RStudio sur différentes plateformes.	2
Premiers pas avec R	5
Commentaires	5
Installer et importer un package	5
Demander de l'aide	5
Écrire et exécuter son premier script R.	5
Créer un nouveau script	6
Démonstration	6
Création, modification, suppression d'un objet et structure dans R	6
Opérations arithmétiques	9
Unité élémentaire : le vecteur	10
Exercices	12

Introduction

Créé en 1993, R est un langage de programmation open source spécialement conçu pour l'analyse statistique et la visualisation des données. Facile à prendre en main et très efficace, il est un outil essentiel pour tous les acteurs.trices de la data science.

Le cours vise à vous fournir une solide base de programmation en R, en mettant l'accent sur la compréhension des concepts fondamentaux, la pratique régulière et l'évaluation continue de vos compétences. Les démonstrations, les exercices et les QCM sont conçus pour vous aider à développer votre maîtrise de R au fil du cours.

Organisation

Contenu du Cours : - Démonstrations - Exercices

Ressources

- Rbloggers : Blog Populaire sur le langage R
- Datacamp : Plateforme d'apprentissage de la data science interactive
- Big Book of R : Edition qui rassemble des livres open-source sur le langage R
- Introduction accélérée au langage R pour la data science : L'essentiel et plus de tout ce que nous verrons

Les bases de la programmation R

Qu'est-ce que R ?

R est un langage de programmation et un environnement logiciel libre dédié à l'analyse statistique et à la représentation graphique. Il est largement utilisé par les statisticiens, data scientist et les chercheurs pour diverses analyses et projets de recherche.

L'une des principales raisons pour lesquelles R est si populaire est sa flexibilité et sa capacité à s'intégrer avec d'autres langages de programmation comme Python, Java, et C++. De plus, R dispose d'une vaste bibliothèque de packages, ce qui le rend extrêmement extensible.

Concepts de base de l'analyse des données.

L'analyse des données est le processus d'inspection, de nettoyage, de transformation et de modélisation des données dans le but de découvrir des informations utiles et de soutenir la prise de décision.

Avec R, vous pouvez effectuer toutes ces étapes de manière efficace et précise. Dans ce cours, vous retrouverez les concepts de base de l'analyse des données avec R : types de données, manipulation des données, visualisation des données et l'interprétation des résultats d'analyse.

Installation de R et RStudio sur différentes plateformes.

Installation de R : Rendez-vous sur le site officiel de R et téléchargez la version appropriée pour votre système d'exploitation (Windows, Mac, Linux). Suivez les instructions d'installation.

Installation de RStudio : Après avoir installé R, allez sur le site officiel de RStudio et téléchargez la version gratuite de RStudio Desktop. Installez-le comme n'importe quel autre logiciel.

Lancement de RStudio : Une fois RStudio installé, ouvrez-le. Vous verrez quatre panneaux principaux :

- la 1) console
- 2) l'éditeur de script (source),
- 3) l'environnement et l'historique, et
- les 4) fichiers/plots/packages/aide (output).

RStudio est une interface de développement intégrée (IDE) pour R.

Elle offre une interface utilisateur conviviale pour R, ce qui facilite l'écriture de code, l'exécution de scripts et la visualisation des résultats.

! Toujours prendre la bonne habitude de définir un projet dans RStudio

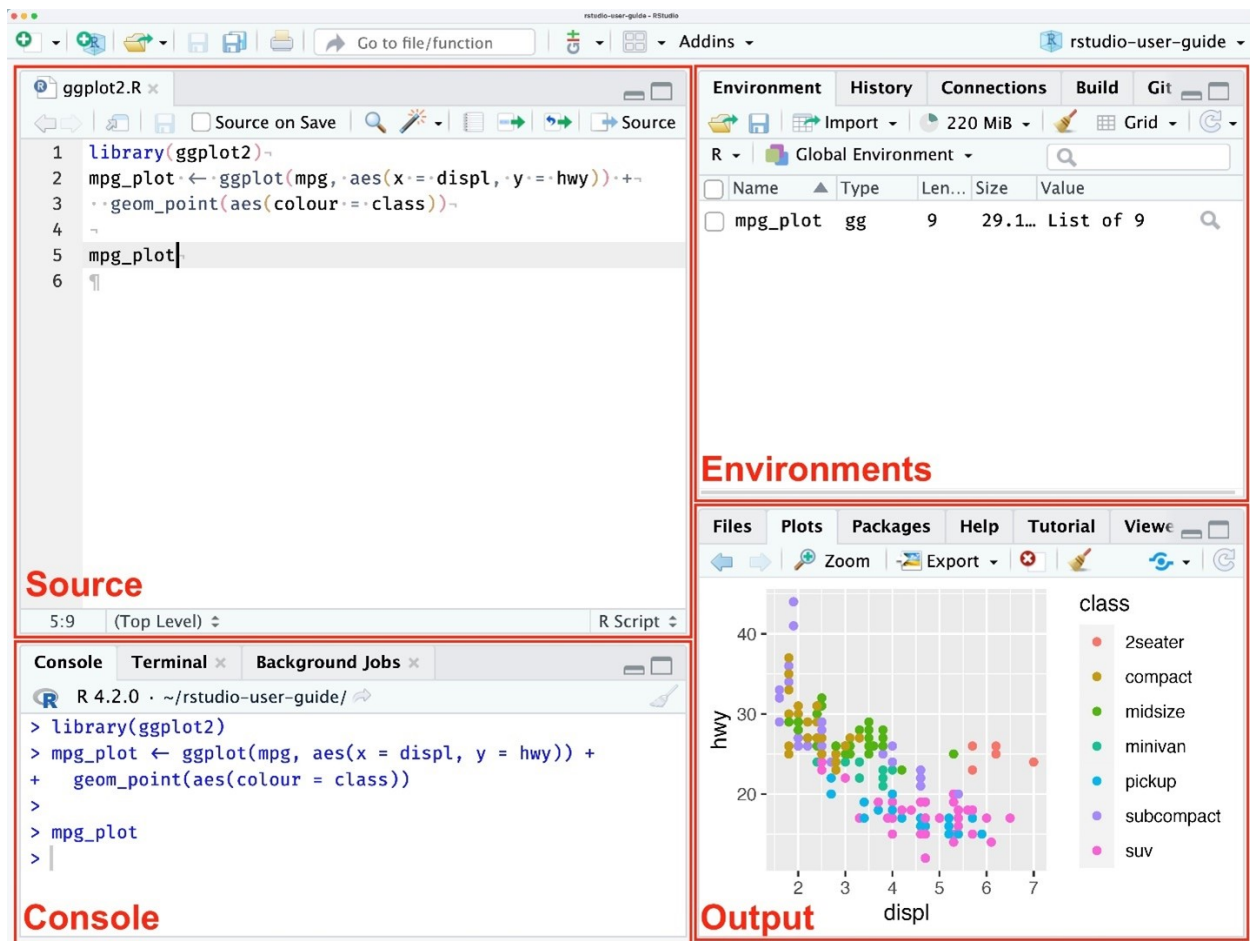
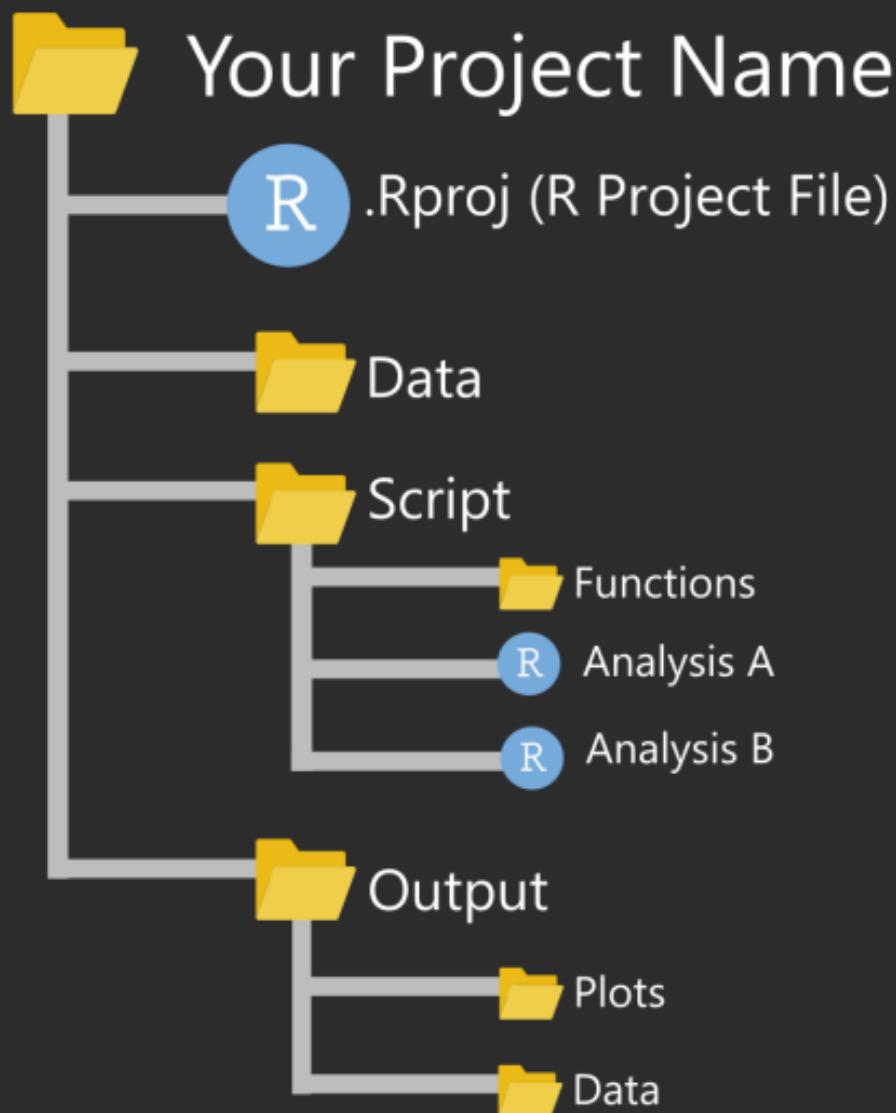


Figure 1: Figure 1. Interface de RStudio

A basic R project set up



<https://martinctc.github.io/>

Figure 2: Figure 2. Structure basique pour les projets. Source: <https://martinctc.github.io/>.

Premiers pas avec R

Pour commencer avec R et RStudio Dans la console, vous pouvez taper des commandes R et voir les résultats immédiatement. Essayez de taper `2 + 2` et appuyez sur Entrée. Vous devriez voir le résultat 4.

```
2 + 2
```

```
## [1] 4
```

Commentaires

Un commentaire en R est un morceau de texte dans votre script R qui n'est pas exécuté comme une partie du code. Les commentaires sont essentiels pour documenter ce que fait votre code, clarifier des parties complexes, ou laisser des notes pour vous-même ou pour d'autres personnes qui pourraient lire ou utiliser votre code plus tard.

En R, un commentaire est créé en utilisant le symbole `#`.

```
# Ceci est un commentaire simple
```

```
x <- 5 # Ceci est un commentaire suivant une instruction
```

Installer et importer un package

Les packages sont des collections de fonctions R et de données. Pour installer un package, utilisez la commande

```
# install.packages("nom_du_package")
```

Par exemple, pour installer le package `ggplot2`, tapez `install.packages("ggplot2")`. Pour utiliser le package, tapez `library(ggplot2)`.

```
# library(ggplot2)
```

Demander de l'aide

Si vous êtes perdu ou si vous avez besoin d'informations sur une fonction, tapez `?nom_de_la_fonction`. Par exemple, `?mean` vous communiquera des informations sur la fonction `mean`.

```
?mean
```

```
## starting httpd help server ... done
```

```
# help(mean)
```

Écrire et exécuter son premier script R.

Écrire et exécuter un script R est une étape fondamentale pour se familiariser avec le langage R. Un script R est simplement un fichier contenant une séquence de commandes que R peut exécuter. Voici comment vous pouvez écrire et exécuter votre premier script R.

Créer un nouveau script

Dans RStudio, allez dans File > New File > R Script. Cela ouvrira un nouvel onglet d'éditeur où vous pouvez commencer à écrire votre script.

Écrire le script Dans l'éditeur, commencez à taper vos commandes R. Par exemple, vous pouvez créer un vecteur, calculer sa moyenne, ou tracer un graphique.

Exécuter le script Pour exécuter votre script, vous pouvez soit sélectionner les lignes que vous souhaitez exécuter et cliquer sur le bouton Run, soit sauvegarder votre script et le source dans la console R en utilisant la commande `source("chemin_vers_votre_script.R")`.

Sauvegarder le script Il est toujours recommandé de sauvegarder votre script pour une utilisation future. Pour ce faire, allez dans File > Save ou appuyez sur Ctrl + S.

Démonstration

Travailler par projets dans RStudio. Création du premier script dans le projet enseignement R Tour d'horizon de RStudio : commandes d'exécution, cheatsheets, les packages... Concept de base : • Création, modification, suppression d'un objet et structure dans R • Opérations arithmétiques • Unité élémentaire : le vecteur

Création, modification, suppression d'un objet et structure dans R

```
# Répertoire de travail et changer de répertoire

# Afficher le répertoire courant : getwd()

# Changer de répertoire courant
# setwd(dir = "chemin_vers_nouveau_repertoire")

# Installation d'un package
# install.packages("ggplot2")

# Chargement d'un package
# library(dplyr)

# Afficher tous les jeux de données par défaut dans R : data()
data()

# Afficher la description d'un jeu de données
?mtcars

# Charger un jeu de données
data("mtcars")

# Création de la variable x : " <-"
x <- 2

# Afficher la variable x
print(x)
```

```
## [1] 2
```

```
# Afficher le type de la variable x  
typeof(x)
```

```
## [1] "double"
```

```
# Creation d'une chaine de caractere  
chaine <- "Hello, Word!"  
print(chaine)
```

```
## [1] "Hello, Word!"
```

```
typeof(chaine)
```

```
## [1] "character"
```

```
# Règle de nommage des variables (R sensible à la casse)  
# tout en minuscule : nomvariable ;  
# - séparation par un point : nom.variable ;  
# - séparation par un trait de soulignement : nom_variable ;  
# - lowerCamelCase, avec la première lettre en minuscule : nomVariable ;  
# - PascalCase, avec la première lettre en majuscule : NomVariable.
```

```
mavariabale <- "jour"  
ma.variable <- "semaine"  
print(mavariabale)
```

```
## [1] "jour"
```

```
print(ma.variable)
```

```
## [1] "semaine"
```

```
mavariabale123 <- "mois"  
print(mavariabale123)
```

```
## [1] "mois"
```

```
# Affichage de la variable x : print() / sprintf  
print(x)
```

```
## [1] 2
```

```
sprintf("%d",x)
```

```
## [1] "2"
```

```
sprintf("%s", chaine)
```

```
## [1] "Hello, Word!"
```

```
x <- 3  
sprintf("J'ai %d bananes", x)
```

```
## [1] "J'ai 3 bananes"
```

```
# Valeur arrondie : round()  
mon_nombre <- 3.2  
  
# Arrondir à la valeur entière  
round(mon_nombre, 0)
```

```
## [1] 3
```

```
# Vérification du type de la variable : class, is.character, is.numeric ...  
class(mon_nombre)
```

```
## [1] "numeric"
```

```
is.numeric(mon_nombre) # renvoie TRUE
```

```
## [1] TRUE
```

```
is.character(mon_nombre) # renvoie FALSE
```

```
## [1] FALSE
```

```
# Convertir des variables numériques <- char / char -> numérique  
age_character <- "16"  
is.character(age_character)
```

```
## [1] TRUE
```

```
age_numeric <- as.numeric(as.character(age_character))  
is.numeric(age_numeric)
```

```
## [1] TRUE
```

```
# Suppression d'un objet : rm  
rm(age)
```

```
## Warning in rm(age): objet 'age' introuvable
```



```
# Affiche les éléments dans l'enV  
ls()
```

```
## [1] "age_character" "age_numeric"   "chaine"         "ma.variable"  
## [5] "mavariable"     "mavariable123"  "mon_nombre"     "mtcars"  
## [9] "x"
```

```
# Supprime tous les éléments dans l'enV  
rm(list = ls())
```

Opérations arithmétiques

```
# + : addition,  
# - : soustraction,  
# * : multiplication,  
# / : division,  
# ^ ou **: puissance,  
# sqrt() : racine carré  
# %/% : division entière,  
# %% : modulo = reste de la division entière
```

```
x <- 3  
y <- 5  
x + y
```

```
## [1] 8
```

```
x - y
```

```
## [1] -2
```

```
x * y
```

```
## [1] 15
```

```
x / y
```

```
## [1] 0.6
```

```
# Puissance  
x ^ 2
```

```
## [1] 9
```

```
x**2
```

```
## [1] 9
```

```
# Racine carré  
sqrt(25)
```

```
## [1] 5
```

```
x %% y
```

```
## [1] 3
```

Unité élémentaire : le vecteur

```
# Création d'un vecteur : int, char  
mon_vecteur <- c(1, 9, 10, 35)  
print(mon_vecteur)
```

```
## [1] 1 9 10 35
```

```
vecteur_caractere <- c("Hello", "Word", "!")  
print(vecteur_caractere)
```

```
## [1] "Hello" "Word"  "!"
```

```
# Typage automatique de 2 en chaîne de caractère  
vecteur_multiple <- c("Hello", 2)  
print(vecteur_multiple)
```

```
## [1] "Hello" "2"
```

```
# Indicage  
vecteur_caractere[2] # affiche l'élément 2 du vecteur
```

```
## [1] "Word"
```

```
vecteur_caractere[3] # affiche l'élément 3 du vecteur
```

```
## [1] "!"
```

```
mon_vecteur[1:3] # affiche les 3 premières valeurs du vecteur
```

```
## [1] 1 9 10
```

```
mon_vecteur <- c(1, 9, 10, 35)  
addition <- mon_vecteur[1] + mon_vecteur[4]  
print(addition)
```

```
## [1] 36
```

```
# Opérations d'un vecteur : Longueur, somme, moyenne, ...
length(mon_vecteur)
```

```
## [1] 4
```

```
sum(mon_vecteur)
```

```
## [1] 55
```

```
mean(mon_vecteur)
```

```
## [1] 13.75
```

```
# Cas des Valeurs manquantes dans un vecteur
vecteur_valeur_manquante <- c(1, NA, 10, 24)
```

```
sum(vecteur_valeur_manquante) # renvoie NA
```

```
## [1] NA
```

```
sum(vecteur_valeur_manquante, na.rm = TRUE) # retire les valeurs manquantes
```

```
## [1] 35
```

```
mean(vecteur_valeur_manquante) # renvoie NA
```

```
## [1] NA
```

```
mean(vecteur_valeur_manquante, na.rm = TRUE) # retire les valeurs manquantes
```

```
## [1] 11.66667
```

```
# Combinaison de vecteur
mon_vecteur1 <- c(1, 9, 10, 35)
mon_vecteur2 <- c(27, 49)
```

```
nouveau_vecteur <- c(mon_vecteur1, mon_vecteur2)
print(nouveau_vecteur)
```

```
## [1] 1 9 10 35 27 49
```

```
# Combinaison de vecteur et "collage": paste, paste0 et sprintf (%s, %f ou %g)
nom <- "Dupont"
prenom <- "Jean"
age <- 15
```

```
identite <- paste(prenom, nom, sep = " ")
print(identite)
```

```
## [1] "Jean Dupont"
```

```
# Recyclage de identité avec ajout de l'age
identite <- paste(identite, age, sep = " ")

# Recyclage d'identité avec l'ajout de "ans"
identite <- paste0(identite, " ans")
print(identite)
```

```
## [1] "Jean Dupont 15 ans"
```

```
# Définition de l'identité avec paste0
identite <- paste0("Je m'appelle ", prenom, " ", nom, ",", " j'ai ", age, " ans")
```

```
# Recyclage d'un vecteur
x <- c(1,5)
y <- c(9, 76, 20, 30)
x + y
```

```
## [1] 10 81 21 35
```

Exercices

Il temps de passer aux premiers exercices contenues dans le fichier **exercices1.txt**. Vous trouverez la corrections des exercices dans le fichier **correction_xercices1.R**.