R : Introduction à R — Analyse R | Cours 2

Ményssa Cherifa-Luron

2023-09-15

Contents

Introduction

Les structures de données en R servent à organiser et stocker les données de manière à faciliter leur manipulation, analyse et visualisation. En fonction de la nature et des exigences des données, différentes structures sont utilisées pour optimiser les performances, la facilité d'utilisation et l'efficacité du traitement des données.

Chaque structure a ses caractéristiques et est choisie en fonction des besoins spécifiques de l'analyse ou de la manipulation de données en question. En somme, ces structures sont essentielles pour une gestion efficace et une analyse rigoureuse des données dans R.

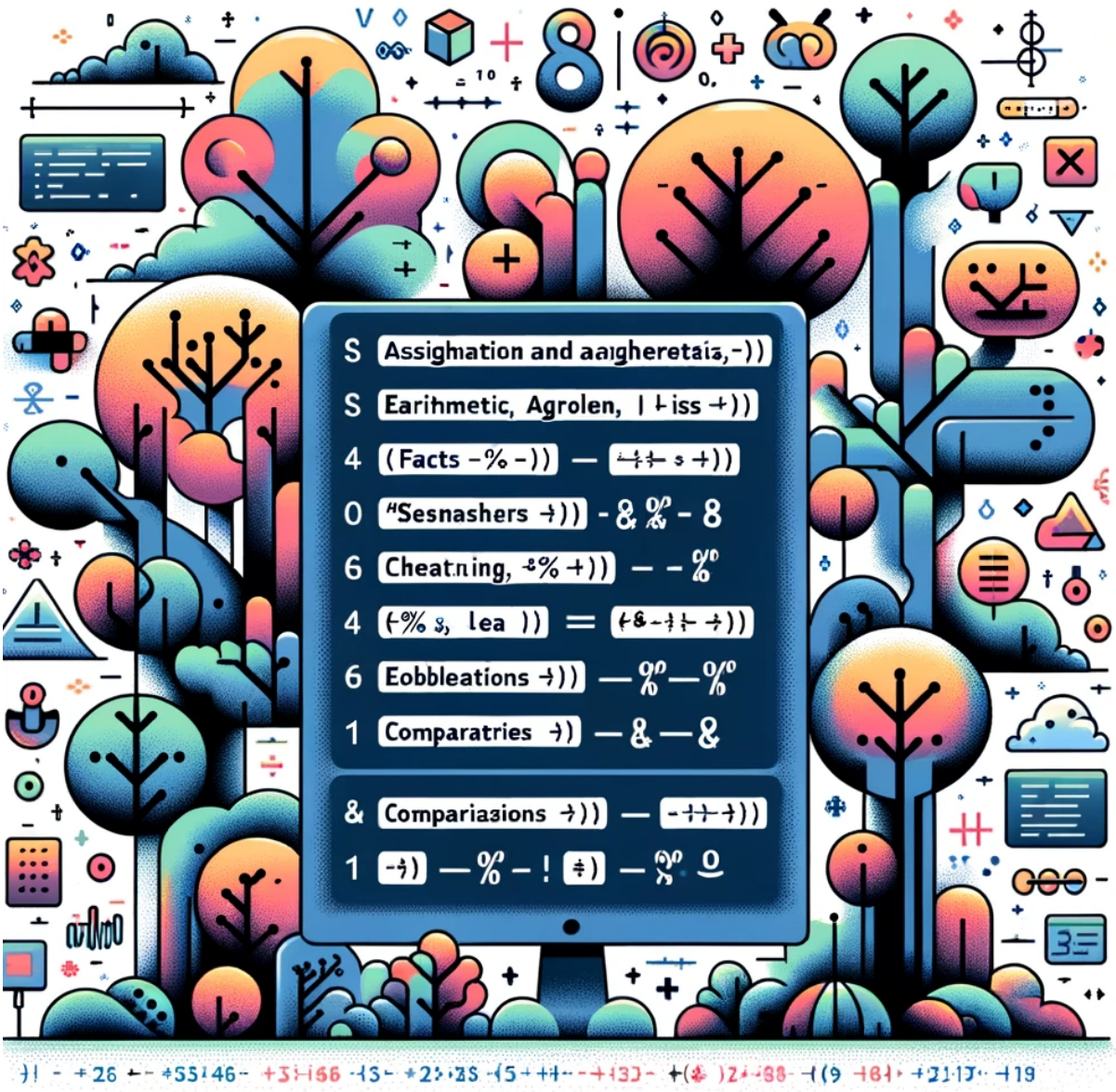

Figure 1: Illustration du langage R par DALL-E

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

Assignation et Opérateurs Arithmétiques, Booléens et Logiques

Opérateurs du Langage R

Tableau des principaux opérateurs en R :

Opérateur	Rôle
$	Extraction d'une liste ou d'une colonne
^	Puissance
-	Changement de signe
:	Génération de suites
%*%, %%, %/%	Produit matriciel, modulo, division entière
* /	Multiplication, division
+ -	Addition, soustraction
<, <=, ==, >=, >, !=	Comparaisons
!	Négation logique
&, &&	Opérateur ET logique
|, ||	Opérateur OU logique
->, -»	Assignation (droite)
<- , «-	Assignation (gauche)

Opérateurs d'Assignation

```
# Exemple d'assignation
x <- 10  # Assignation de 10 à x
```

Opérateurs Arithmétiques

```
# Exemple d'opération arithmétique
5 + 3 # Addition, renvoie 8
```

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

Opérateurs Booléens (de Comparaison)

```r
# Exemple d'opérateur booléen
2 < 5 # Renvoie TRUE car 2 est inférieur à 5
```

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

Opérateurs Logiques

```r
# Exemple d'opérateur logique
a <- 5
b <- 10

(a > 5) && (b < 10) # Renvoie TRUE si a > 5 ET b < 10
```

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

Autres Opérateurs et Fonctions Utiles

```r
# Utilisation de %in%
x <- c("A", "B", "C", "D")
x %in% c("D", "B") # Renvoie un vecteur de valeurs logiques
```

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

Les structures de données

En R, plusieurs types de structures de données sont utilisés pour stocker et manipuler les données. Chacune de ces structures a des caractéristiques et des usages spécifiques. Voici un aperçu des principales structures de données en R :

1. Vecteurs

- **Définition :** Les vecteurs sont des séquences d'éléments du même type (numérique, caractère, logique, etc.).
- **Création :** Utilisation de `c()` (combine) pour créer des vecteurs. Exemple : `vecteur <- c(1, 2, 3)`.
- **Usage :** Adaptés pour des opérations sur des séries d'éléments homogènes.

```r
# Création d'un vecteur
v <- c(1, 2, 3, 4)
```

2. Facteurs

- **Définition :** Les facteurs sont utilisés pour stocker des données catégorielles.
- **Création :** Utilisation de `factor()` pour créer des facteurs. Exemple : `facteur <- factor(c("oui", "non", "oui"))`.
- **Usage :** Importants pour l'analyse statistique, en particulier pour la modélisation.

```r
# Création d'un facteur
genre <- factor(c("F", "M", "NB", "F", "M", "NB", "M"))
```

3. Listes

- **Définition :** Les listes sont des collections d'éléments qui peuvent être de types différents.
- **Création :** Utilisation de `list()` pour créer des listes. Exemple : `liste <- list(nombre = 1, chaine = "R")`.
- **Usage :** Utiles pour stocker des données hétérogènes et complexes.

```r
# Création d'une liste
l <- list(1, "a", c(1, 2, 3))
```

4. Matrices

- **Définition :** Les matrices sont des collections d'éléments de même type organisés en un tableau à deux dimensions.
- **Création :** Utilisation de `matrix()` pour créer des matrices. Exemple : `matrice <- matrix(1:9, nrow = 3)`.
- **Usage :** Convient aux opérations mathématiques et statistiques impliquant des données en deux dimensions.

```r
# Création d'une matrice
m <- matrix(1:9, nrow=3)
```

5. Data Frames

- **Définition :** Les data frames sont similaires aux matrices mais peuvent contenir différents types de données dans chaque colonne.
- **Création :** Utilisation de `data.frame()` pour créer des data frames. Exemple : `df <- data.frame(Nom = c("Alice", "Bob"), Age = c(23, 25))`.
- **Usage :** Très utilisés en analyse de données pour représenter et manipuler des ensembles de données de manière tabulaire.

```r
# Création d'un data frame
df <- data.frame(name=c("Alice", "Bob"), age=c(25, 30))
```

Exercices

Pour commencer à pratiquer, suivez ces étapes :

1. **Accédez au Dépôt GitHub :** Visitez l'URL fournie : https://github.com/universdesdonnees/Introduction-a-R pour accéder au dépôt GitHub contenant les matériaux du cours.

2. **Trouvez le Fichier des Exercices :** Dans le dépôt, localisez le fichier nommé **exercices2.txt**. Ce fichier contient les premiers exercices que vous devez pratiquer.

3. **Lisez et Essayez de Résoudre les Exercices :** Ouvrez le fichier **exercices1.txt** et lisez attentivement les exercices. Essayez de les résoudre par vous-même dans votre environnement R (comme RStudio). Il est important de pratiquer par vous-même avant de regarder les solutions pour mieux apprendre.

4. **Consultez la Correction :** Une fois que vous avez tenté de résoudre les exercices, ou si vous rencontrez des difficultés, consultez le fichier **correction_exercices2.R** pour voir les solutions. Analysez les solutions pour comprendre les méthodes et logiques utilisées.