R : Introduction à R — Analyse R | Cours 5

Ményssa Cherifa-Luron

2023-09-15

Contents

Introduction

Organisation

Contenu du Cours : - Démonstrations - Exercices

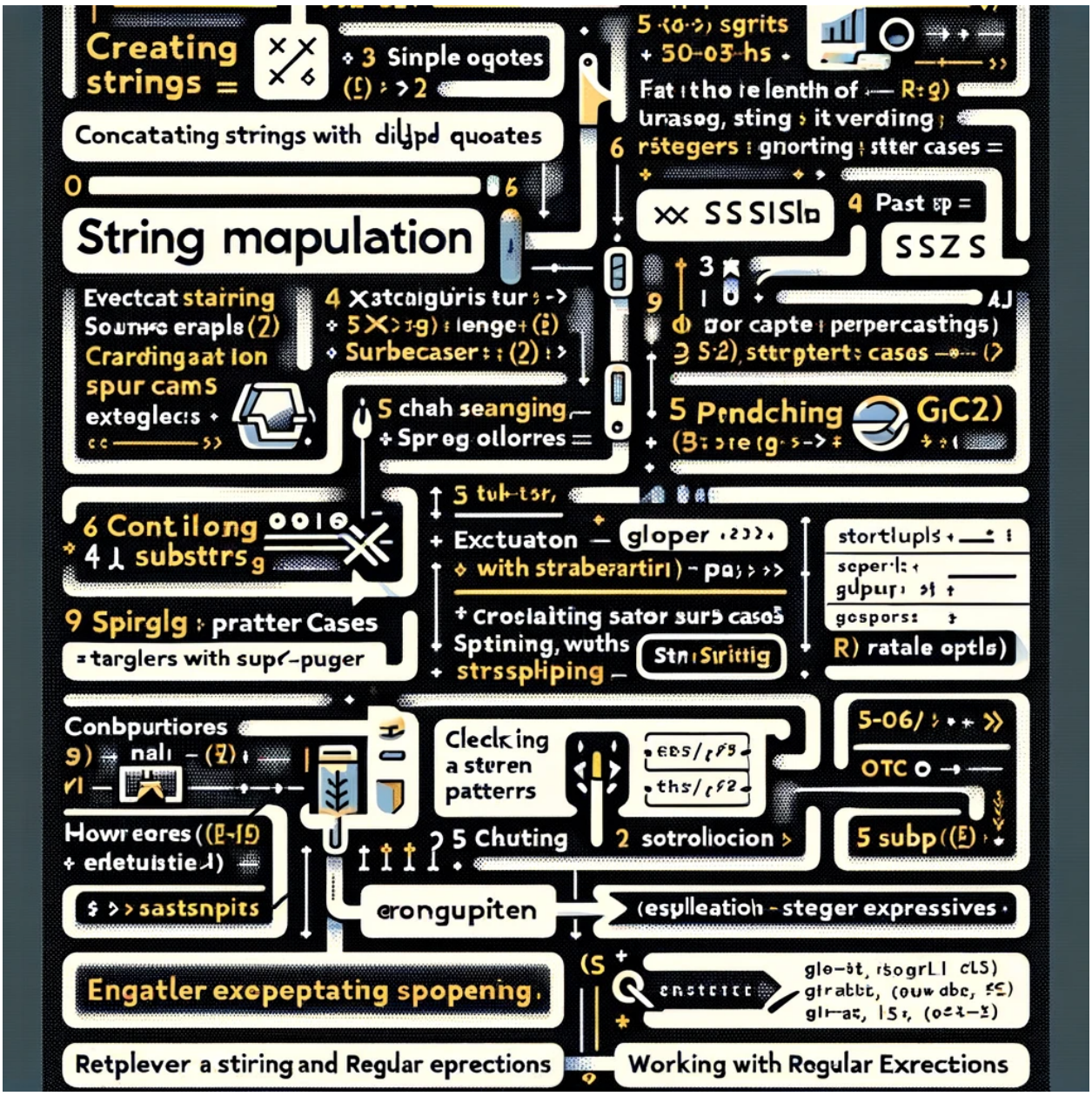

Figure 1: Illustration du langage R par DALL-E

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

Importation de données externes

Voici une explication détaillée des scripts R que vous avez fournis, qui traitent de l'importation de données externes et de l'aperçu des données dans le contexte du jeu de données `starwars`.

Importation de Données CSV avec `read.csv`

```r
# Importation d'un fichier CSV
bike <- read.csv("../data/Animation.csv", header = TRUE, sep = ";")
```

- `read.csv` est une fonction de base en R pour lire les fichiers CSV.
- `header = TRUE` indique que la première ligne du fichier contient les noms des colonnes.
- `sep = ";"` spécifie que le séparateur de colonnes dans le fichier CSV est un point-virgule.

Importation de Données Excel avec `readxl`

```r
# Importation d'un fichier Excel
library(readxl)

bike2 <- read_excel("../data/BikeSalesAnalysis.xlsx", sheet = "BikeSales")
```

- `read_excel` est une fonction du package `readxl`, utilisée pour lire des fichiers Excel.
- `sheet = "BikeSales"` spécifie le nom de la feuille de calcul à importer.

```r
# Récupération des noms des feuilles dans un fichier Excel
liste_noms <- excel_sheets("../data/BikeSalesAnalysis.xlsx")
```

- `excel_sheets` est utilisée pour lister toutes les feuilles disponibles dans un fichier Excel.

Aperçu des Données : Starwars

Visualisation des Données avec `View`

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
# Ouvre un aperçu interactif de la dataframe
View(starwars)
```

- `View` ouvre une fenêtre interactive pour visualiser la dataframe `starwars`. C'est utile pour explorer les données de manière graphique.

```r
# Aperçu structuré des données
glimpse(starwars)
```

Exploration des Données avec `glimpse`

```
## Rows: 87
## Columns: 14
## $ name       <chr> "Luke Skywalker", "C-3PO", "R2-D2", "Darth Vader", "Leia Or~
## $ height     <int> 172, 167, 96, 202, 150, 178, 165, 97, 183, 182, 188, 180, 2~
## $ mass       <dbl> 77.0, 75.0, 32.0, 136.0, 49.0, 120.0, 75.0, 32.0, 84.0, 77.~
## $ hair_color <chr> "blond", NA, NA, "none", "brown", "brown, grey", "brown", N~
## $ skin_color <chr> "fair", "gold", "white, blue", "white", "light", "light", "~
## $ eye_color  <chr> "blue", "yellow", "red", "yellow", "brown", "blue", "blue",~
## $ birth_year <dbl> 19.0, 112.0, 33.0, 41.9, 19.0, 52.0, 47.0, NA, 24.0, 57.0, ~
## $ sex        <chr> "male", "none", "none", "male", "female", "male", "female",~
## $ gender     <chr> "masculine", "masculine", "masculine", "masculine", "femini~
## $ homeworld  <chr> "Tatooine", "Tatooine", "Naboo", "Tatooine", "Alderaan", "T~
## $ species    <chr> "Human", "Droid", "Droid", "Human", "Human", "Human", "Huma~
## $ films      <list> <"The Empire Strikes Back", "Revenge of the Sith", "Return~
## $ vehicles   <list> <"Snowspeeder", "Imperial Speeder Bike">, <>, <>, <>, "Imp~
## $ starships  <list> <"X-wing", "Imperial shuttle">, <>, <>, "TIE Advanced x1",~
```

- `glimpse` est une fonction du package `dplyr` qui fournit un aperçu rapide de la structure d'une dataframe, y compris les noms des colonnes, les types de données et les premières valeurs de chaque colonne.

Ces scripts fournissent une base solide pour commencer à travailler avec des données externes en R, que ce soit en format CSV ou Excel, et offrent des méthodes pratiques pour inspecter rapidement les jeux de données.

Manipulation de données avec dplyr

Voici le contenu du cours sur la programmation avec l'opérateur `%>%` (pipe) en R, en utilisant le package `dplyr` et d'autres fonctions de manipulation de données. Ce cours aborde différentes techniques et fonctions pour manipuler les données de manière fluide et lisible.

L'opérateur : "%>%"

La programmation avec le "pipe" en R, symbolisée par l'opérateur `%>%`, est une technique populaire pour chaîner des opérations de manière lisible.

Cet opérateur, popularisé par le package `magrittr` et largement utilisé avec `dplyr`, permet de transférer le résultat d'une expression vers une autre, rendant le code plus lisible et évitant la création de multiples variables temporaires.

Les fonctions avancées

Les fonctions avancées utilisées avec l'opérateur `%>%` dans R, en particulier avec le package `dplyr`, offrent des capacités étendues pour manipuler et analyser des ensembles de données de manière efficace et intuitive.

Voici un aperçu de quelques-unes de ces fonctions :

Sélection et Filtrage Avancés

- `select()` : Permet de choisir des colonnes spécifiques dans un data frame. Peut être utilisée avec des fonctions comme `contains()`, `starts_with()`, `ends_with()`, et `matches()` pour sélectionner des colonnes basées sur des motifs dans leurs noms.

$<^2$ - `filter()` : Utilisée pour extraire des lignes qui satisfont certaines conditions. Supporte les opérateurs logiques (`&`, `|`) et peut être utilisée pour des comparaisons complexes.

Manipulation de Chaînes

- `separate()` : Divise une colonne en plusieurs colonnes en fonction d'un séparateur spécifié, ce qui est utile pour séparer les noms complets en prénoms et noms de famille, par exemple.

Groupement et Résumé

- `group_by()` : Permet de regrouper les données par une ou plusieurs colonnes. Très utile pour les analyses ultérieures qui nécessitent des calculs par groupe.
- `summarise()` : Utilisée pour créer des résumés statistiques des groupes de données. Elle peut être utilisée pour calculer des moyennes, des sommes, des médianes, etc., sur des groupes spécifiés par `group_by()`.

Tri et Organisation

- `arrange()` : Trie les données en fonction d'une ou plusieurs colonnes. Peut être utilisée avec `desc()` pour un tri décroissant.

Jointures

- `inner_join()`, `left_join()`, `right_join()`, `full_join()` : Permettent de combiner des data frames en fonction de colonnes clés, similairement aux jointures dans SQL.

Création et Transformation

- `mutate()` : Ajoute de nouvelles colonnes ou modifie des colonnes existantes. Par exemple, vous pouvez l'utiliser pour calculer de nouvelles valeurs à partir des données existantes.
- `transmute()` : Semblable à `mutate()`, mais ne garde que les nouvelles colonnes créées.

Aggrégation Conditionnelle et Opérations Plus Complexes

- `summarise_at()`, `mutate_at()`, `if_else()` : Permettent de réaliser des opérations sur un ensemble de colonnes spécifiées ou d'appliquer des conditions complexes.

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 **exercices5.txt**. Ce fichier contient les premiers exercices que vous devez pratiquer.
3. **Lisez et Essayez de Résoudre les Exercices :** Ouvrez le fichier **exercices4.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_exercices5.R** pour voir les solutions. Analysez les solutions pour comprendre les méthodes et logiques utilisées.