



Le ReJMiC présente :



Introduction à R

Journée d'initiation à la bio-informatique n°2
24 juin 2022

Noshine Mohammad



Qu'est-ce que le logiciel R ?



R est un système qui est communément appelé langage et logiciel. Il permet, entre autres, de réaliser des analyses statistiques. Plus particulièrement, il comporte des moyens qui rendent possibles la manipulation des données, les calculs et les représentations graphiques. **R** a aussi la possibilité d'exécuter des programmes stockés dans des fichiers textes.

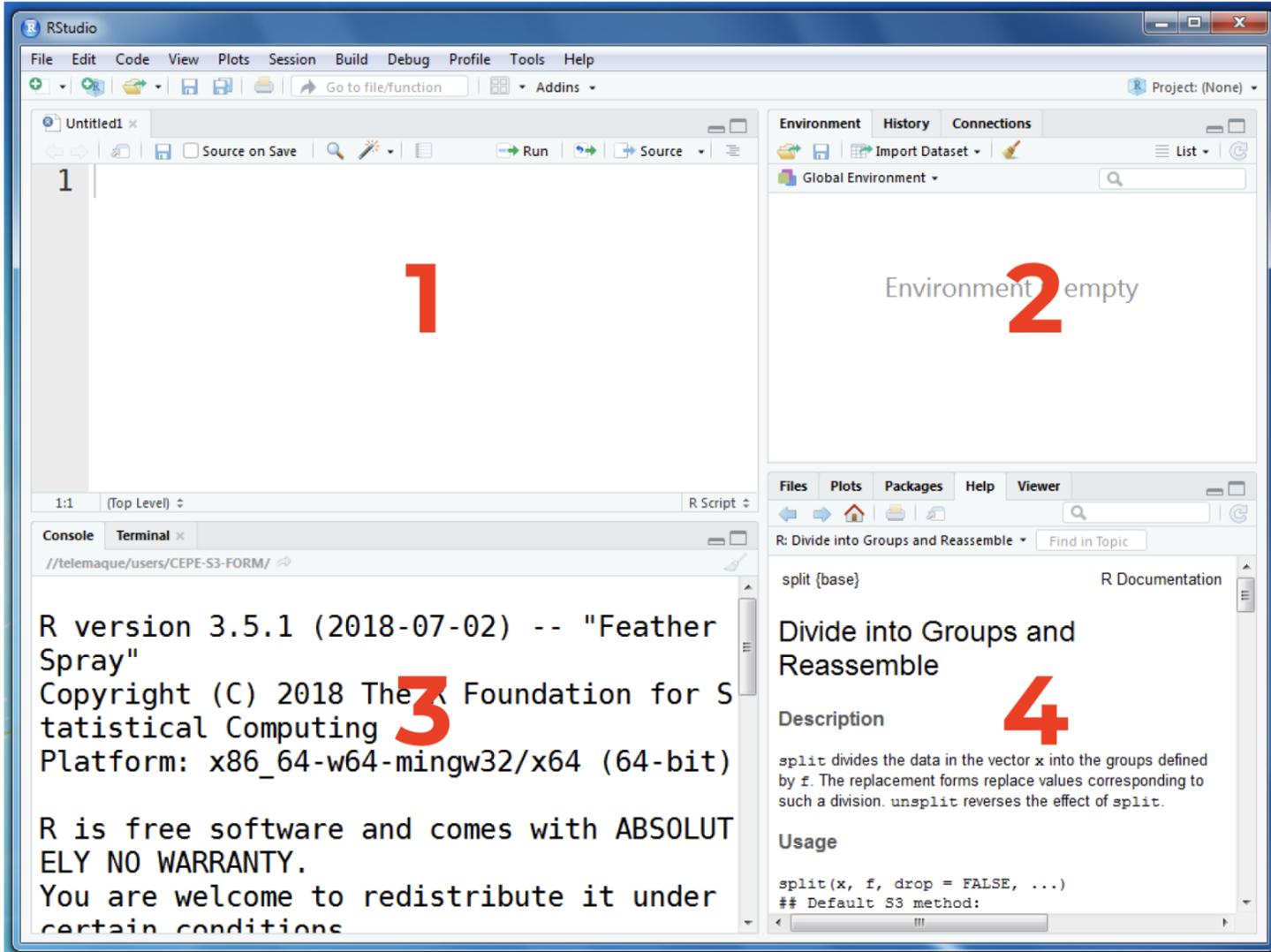
En effet **R** possède :

- un système efficace de manipulation et de stockage des données,
- différents opérateurs pour le calcul sur tableaux (et spécialement les matrices),
- un grand nombre d'outils pour l'analyse des données et les méthodes statistiques,
- des moyens graphiques pour visualiser les analyses,
- un langage de programmation simple et performant comportant : conditions, boucles, moyens d'entrées sorties, possibilité de définir des fonctions récursives.

L'utilisation de **R** est facilitée par l'installation d'un IDE (integrated development environment) ou "environnement de développement intégré".

RStudio est une application permettant de travailler en **R** dans un environnement de développement riche et complet que nous allons utiliser dans la suite de ce cours.

L'interface de RStudio

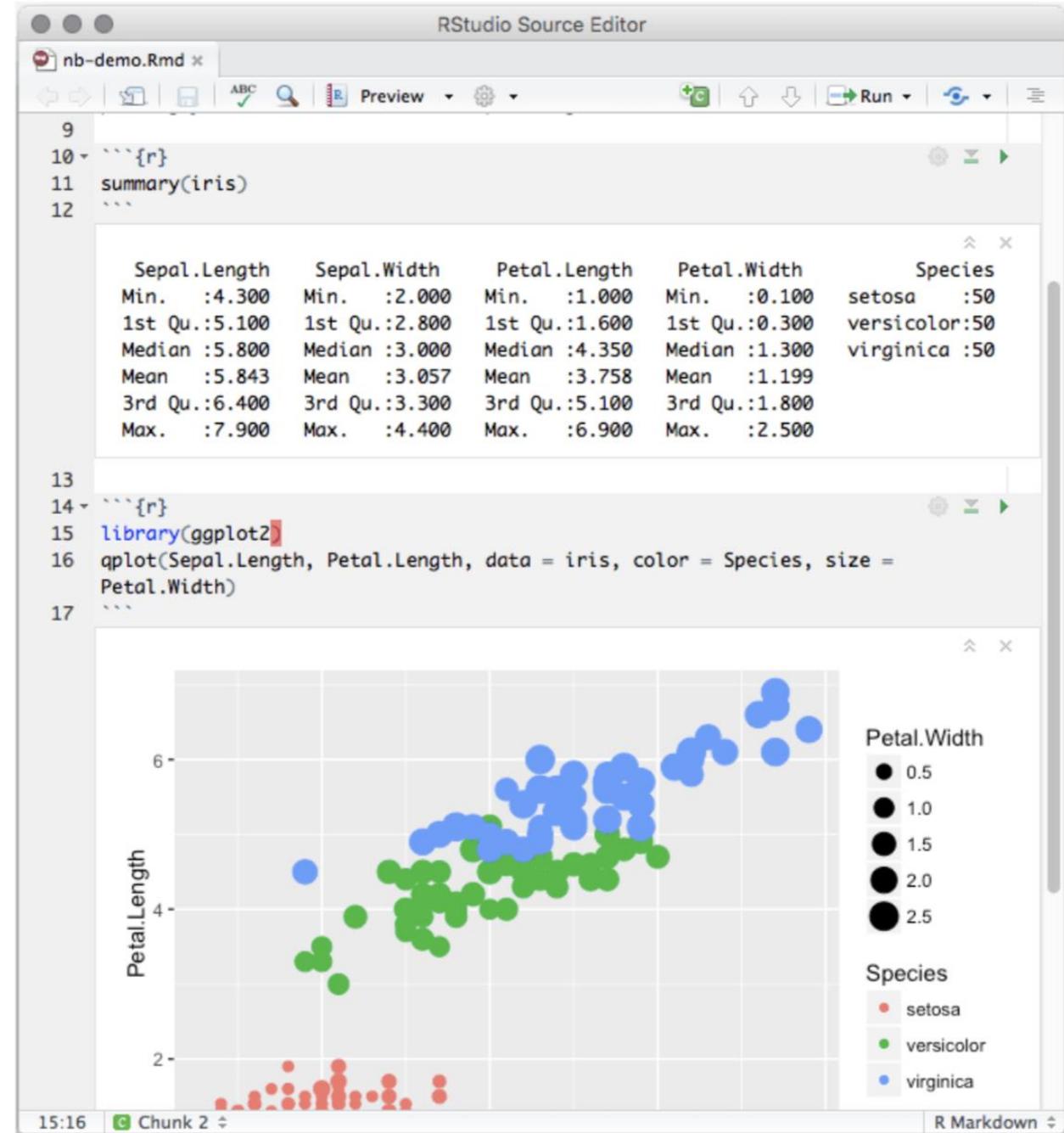


RStudio est divisé en 4 quadrants :

1. Éditeur de texte, de codes, etc.
2. Espace de travail, historique, importation, etc.
3. Console
4. Visualisations, aide

R Notebook

Un **R Notebook** est un document **R Markdown** contenant des morceaux de code/ bloc de code (**chunk**) qui peuvent être exécutés de manière indépendante et interactive, les résultats étant visibles immédiatement sous les entrées.

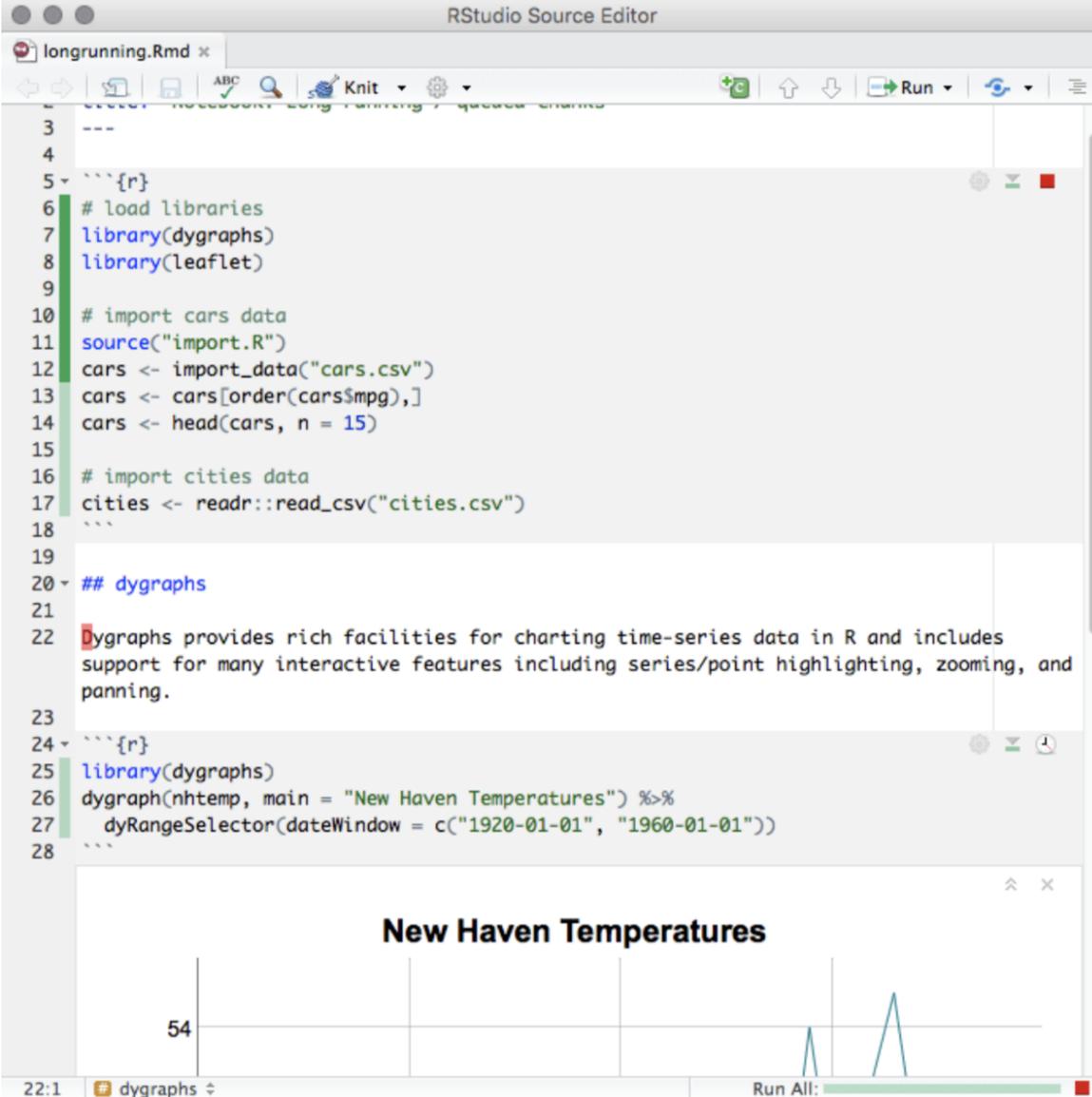


Les chunks

Les chunks du Notebook peuvent être insérés rapidement à l'aide du raccourci clavier Ctrl + Alt + I (macOS : Cmd + Option + I), ou via le menu Insertion de la barre d'outils de l'éditeur.

Le code dans le Notebook est exécuté avec les mêmes gestes que vous utiliseriez pour exécuter du code dans un document **R Markdown** :

- Utilisez le bouton triangulaire vert dans la barre d'outils d'un morceau de code dont l'info-bulle est "Exécuter le morceau actuel", ou Ctrl + Shift + Enter (macOS : Cmd + Shift + Enter) pour exécuter le morceau actuel.
- Appuyez sur Ctrl + Enter (macOS : Cmd + Enter) pour exécuter uniquement l'instruction en cours. L'exécution d'une seule instruction équivaut à l'exécution d'un chunk entier composé uniquement de cette instruction.
- Il existe d'autres façons d'exécuter un lot de chunks si vous cliquez sur le menu Exécuter dans la barre d'outils de l'éditeur, comme Exécuter tout, Exécuter tous les chunks ci-dessus et Exécuter tous les chunks ci-dessous.



The screenshot shows the RStudio Source Editor with a file named 'longrunning.Rmd'. The code is organized into several chunks, each with a green triangular run button in the right margin. The first chunk (lines 5-9) loads the 'dygraphs' and 'leaflet' libraries. The second chunk (lines 10-15) imports 'cars' data from 'import.R', orders it by mpg, and takes the first 15 rows. The third chunk (lines 16-18) imports 'cities' data from 'cities.csv'. The fourth chunk (lines 20-23) is a comment block for 'dygraphs'. The fifth chunk (lines 24-28) uses 'dygraphs' to create a plot titled 'New Haven Temperatures' with a date range selector from 1920-01-01 to 1960-01-01. Below the code, the plot is displayed, showing a line graph with a y-axis value of 54 and a date range selector.

```
3 ---
4
5 ```{r}
6 # load libraries
7 library(dygraphs)
8 library(leaflet)
9
10 # import cars data
11 source("import.R")
12 cars <- import_data("cars.csv")
13 cars <- cars[order(cars$mpg),]
14 cars <- head(cars, n = 15)
15
16 # import cities data
17 cities <- readr::read_csv("cities.csv")
18 ```
19
20 ## dygraphs
21
22 Dygraphs provides rich facilities for charting time-series data in R and includes
23 support for many interactive features including series/point highlighting, zooming, and
24 panning.
25
26 ```{r}
27 library(dygraphs)
28 dygraph(nhtemp, main = "New Haven Temperatures") %>%
29   dyRangeSelector(dateWindow = c("1920-01-01", "1960-01-01"))
30 ```
```

Les packages

Un *package* est un ensemble de fonctions développées par des utilisateurs de R, qui améliorent ou étendent les fonctionnalités de base de R.

Les *packages* sont au coeur de la logique collaborative de R, car ils permettent de bénéficier des contributions d'autres utilisateurs de R. Début 2021, il existe plus de 14 000 *packages* disponibles pour R, qui couvrent une multitude de tâches. Il est donc primordial de savoir où chercher des *packages* et comment les utiliser. Certains d'entre eux sont considérés comme **indispensables** (MASS, rpart, etc.) et sont fournis avec R. Les autres constituent des avancées récentes en statistiques et peuvent être téléchargés librement sur le réseau [CRAN](#).

Un *package* est constitué d'objets R (la plupart du temps des fonctions, parfois des données) et d'une documentation empaquetés en un seul fichier, dans le but d'être partagé facilement.

Pour installer un package, vous pouvez utiliser l'onglet `Package` de RStudio, ou la fonction `install.packages("nom_du_packa`

Afin d'utiliser un package installé, il faut ensuite le charger pour la session de travail avec la fonction `library(nom_du_package)`

Certains packages sont en **constante évolution**, avec de nouvelles versions régulièrement disponibles. Il est donc intéressant de les

Le mieux est de mettre à jour les packages à l'ouverture d'une session R, en utilisant la fonction `update.packages()` .