

(Re)découverte de R... en 1h45

Ecole de Bioinformatique AVIESAN-IFB – Roscoff – Novembre 2024

Elodie Darbo elodie.darbo@u-bordeaux.fr

Elise Jacquemet elise.jacquemet@pasteur.fr

Olivier Kirsh olivier.kirsh@u-paris.fr

Audrey Onfroy audrey.onfroy@inserm.fr

D'après des slides d'Hugo Varet – hugo.varet@pasteur.fr



CNRS UPMC

Station Biologique
Roscoff

R en quelques mots



Langage de programmation qui permet de :

1. manipuler des données : importer, transformer, exporter
2. faire des analyses statistiques plus ou moins complexes : description, exploration, modélisation...
3. créer des (jolies) figures

Disponible sur [RCRAN](https://www.R-project.org/)



Il existe des interfaces graphiques pour programmer en R. Par exemple: RStudio (2011)



Point culture général

Mais pourquoi “R” ?

- R est né en 1993 en Nouvelle-Zélande, par **R**oss Ihaka et **R**obert Gentleman.
- R est une implémentation d’un autre langage de programmation: **S** (pour statistics).
- S est né en 1976 et son implémentation commerciale (S-PLUS) en 1988.
- Il y a des traces de S dans R :
 - la classe S3 (attribution d’une classe à des objets) et
 - la classe S4 (programmation orientée objet)

R est écrit en quoi ? En Fortran et en C, mais aussi en R 🤪

R, S, C... hein ?

Il existe *plein* de langages de programmation. (Presque) chaque lettre de l’alphabet désigne un langage !

Que s’est-il passé entre 1993 et aujourd’hui pour R ?

- 1997 : R fait partie du projet GNU, il est open-source, gratuit et collaboratif
- 2000 : sortie de R 1.0.0
- 31 octobre 2024 : R 4.4.2 - R est vivant 😎

Avantages et inconvénients

Avantages :

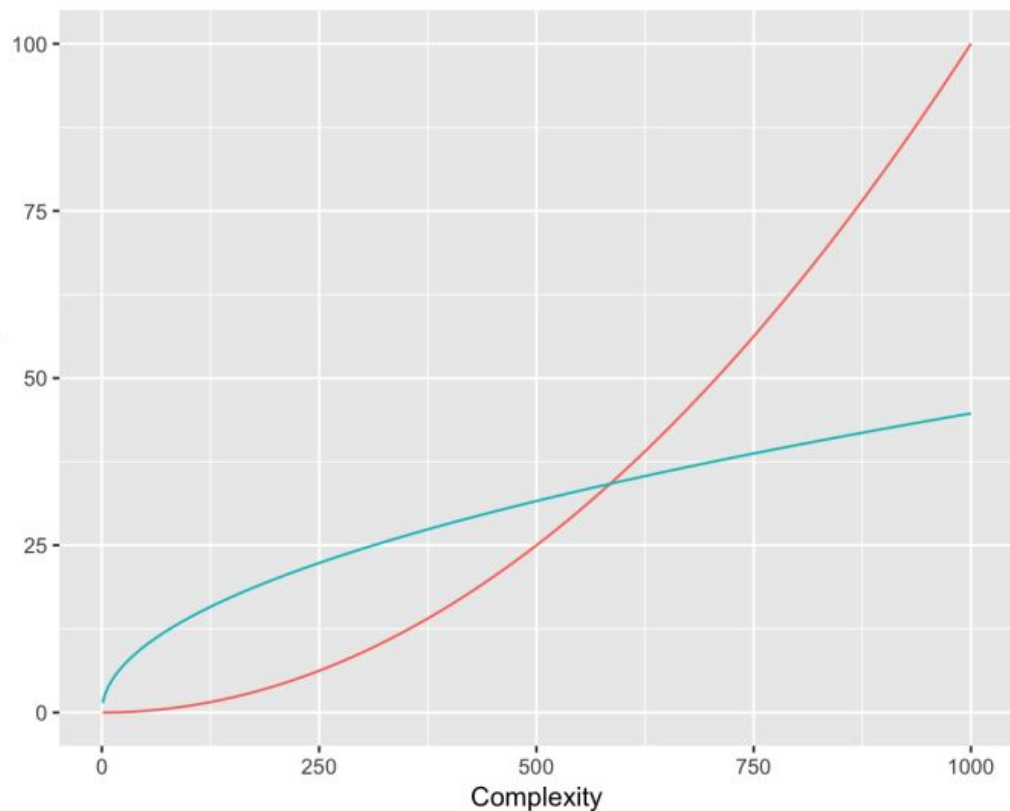
- Souplesse d'utilisation pour réaliser des analyses statistiques
- Libre et gratuit, même s'il existe maintenant des versions payantes de RStudio (shiny et/ou server)
- Traçabilité des analyses en écrivant les commandes R dans des notebooks (Rmd)
- Large communauté d'utilisateurs + aide en ligne
- Grand nombre de packages spécifiques

Inconvénients : langage de programmation, donc:

- Marche d'apprentissage à passer
- Savoir exprimer son besoin pour...
- ... savoir retrouver les informations utiles
- Avoir un regard critique sur les méthodes utilisées, pour appréhender le périmètre d'interprétation des résultats

R vs Excel (ou vs GraphPad Prism)

Difficulty vs. Complexity



tool
— Excel
— R

Covid : le Royaume-Uni passe à côté de milliers de cas à cause... d'un fichier Excel arrivé à saturation

Les autorités sanitaires britanniques ont reconnu que près de 16.000 cas de coronavirus en Angleterre sont passés sous le radar au cours de la semaine écoulée à cause d'un problème dans le chargement des données.

[Lire plus tard](#) [Europe](#) [Partager](#) [Commenter](#)

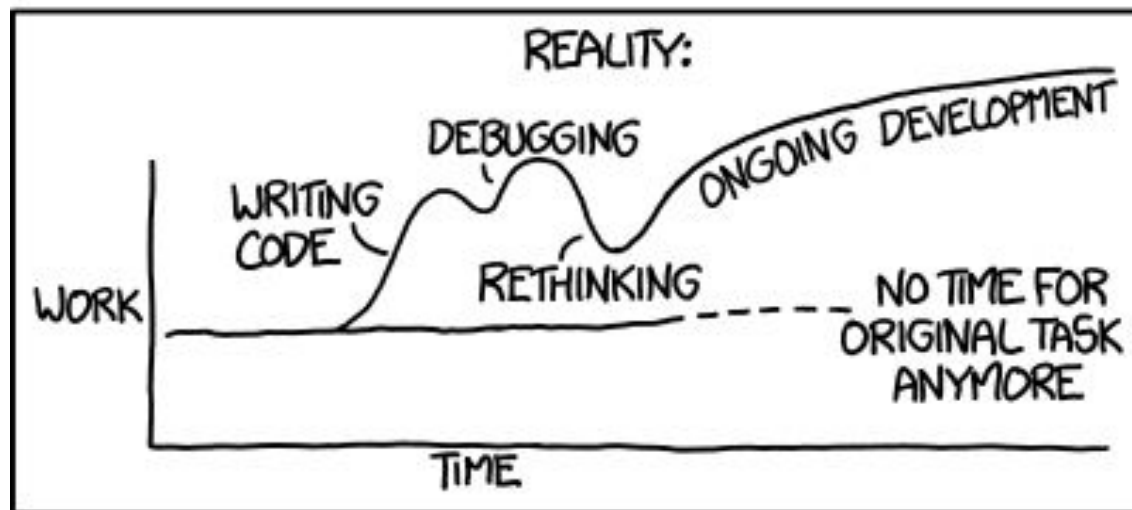
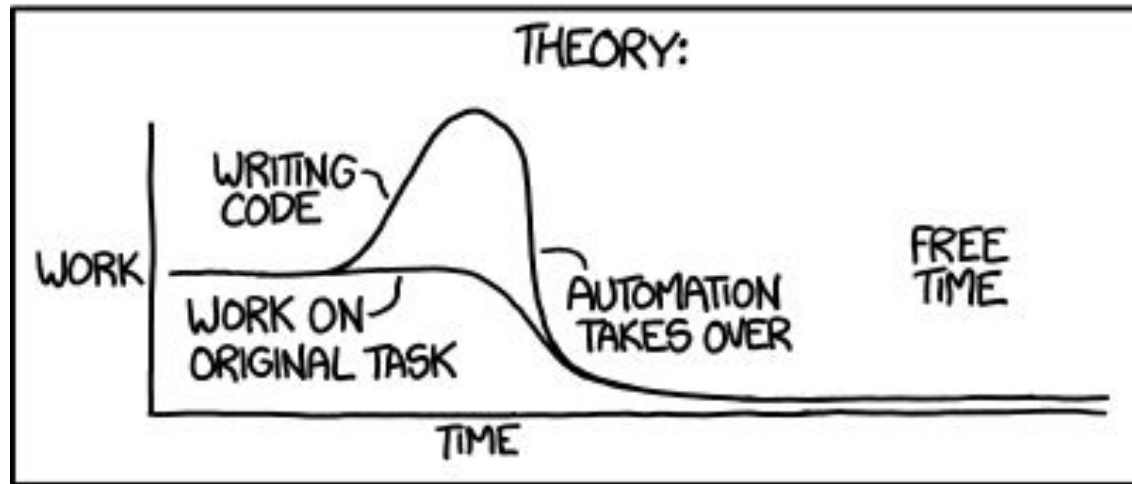


[Alexandre Counis, Les Echos, 5 oct. 2020](#)

Source: R-bloggers

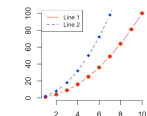
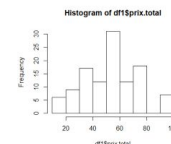
Automatiser des tâches répétitives pour libérer du temps (ou pas)

"I SPEND A LOT OF TIME ON THIS TASK.
I SHOULD WRITE A PROGRAM AUTOMATING IT!"



R sait tout faire : à chaque objectif, sa fonction

Lire un tableau de données		<code>read.table()</code>
Fusionner deux tableaux		<code>merge()</code>
Filtrer des lignes		<code>data[data\$x > 10]</code>
Sélectionner des colonnes		<code>data[,c("x", "y")]</code>
Rechercher une chaîne de caractères		<code>grep()</code>
Calculer une moyenne	Réaliser une ACP	<code>mean()</code>
Additionner deux matrices		<code>prcomp()</code>
Exporter un tableau de données		<code>mat1 + mat2</code>
Calculer une variance		<code>var()</code>
Régression linéaire		<code>lm()</code>
Tracer une courbe		<code>plot()</code>
Tester une hypothèse		<code>t.test()</code>
Dessiner un histogramme		<code>hist()</code>
Convertir des données		<code>as.matrix()</code>



Modes d'utilisation (liste non exhaustive)



Localement via le terminal



Localement via RStudio (utilisation classique)



Sur un serveur via le terminal et une connexion ssh



Sur un serveur, via un navigateur web pour accéder à RStudio Server

Connexion à RStudio via OpenOnDemand

Accéder au portail OpenOnDemand : <https://ondemand.cluster.france-bioinformatique.fr>

Log in to IFB Core Cluster

Username

Password

Login

Connexion avec les
identifiants IFB
Identifiant: *pnom*

Grant Access

IFB Core Cluster Apps Files Jobs Clusters Interactive Apps

OPEN
OnDemand

OnDemand provides an integrated, single access point for all of your HPC resources.

Pinned Apps A featured subset of all available apps

JupyterLab: Core System Installed App

RStudio Server: Core System Installed App

powered by
OPEN OnDemand

OnDemand version: 3.0.2

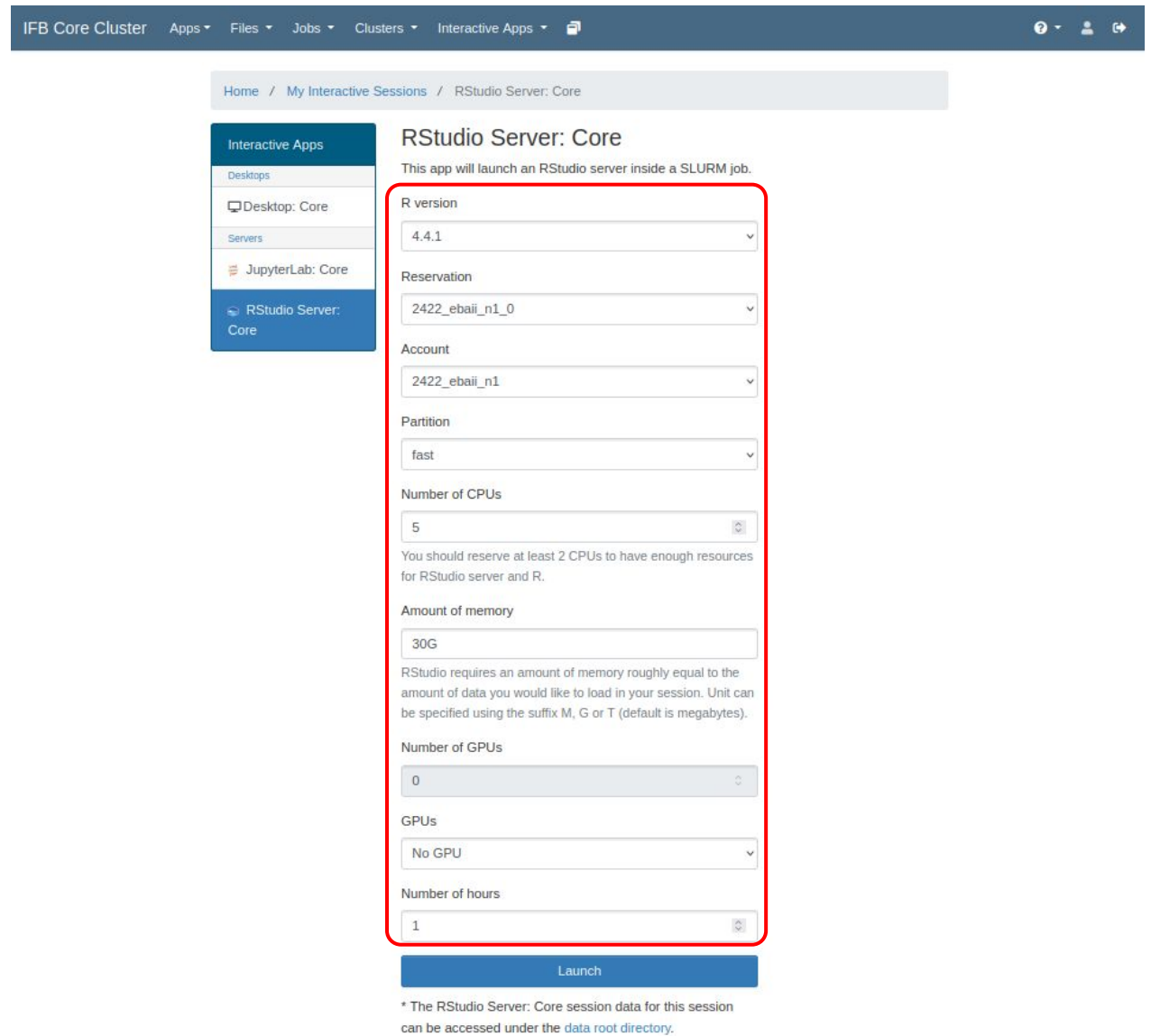
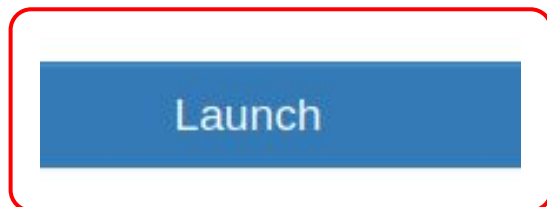
Connexion à RStudio via OpenOnDemand

Remplir avec les ressources :

- Reservation: 2422_ebail_n1_0

Et tout le reste par défaut.

Puis:



IFB Core Cluster Apps Files Jobs Clusters Interactive Apps

Home / My Interactive Sessions / RStudio Server: Core

Interactive Apps

- Desktops
- Desktop: Core
- Servers
- JupyterLab: Core
- RStudio Server: Core**

RStudio Server: Core

This app will launch an RStudio server inside a SLURM job.

R version
4.4.1

Reservation
2422_ebail_n1_0

Account
2422_ebail_n1

Partition
fast

Number of CPUs
5

You should reserve at least 2 CPUs to have enough resources for RStudio server and R.

Amount of memory
30G

RStudio requires an amount of memory roughly equal to the amount of data you would like to load in your session. Unit can be specified using the suffix M, G or T (default is megabytes).

Number of GPUs
0

GPUs
No GPU

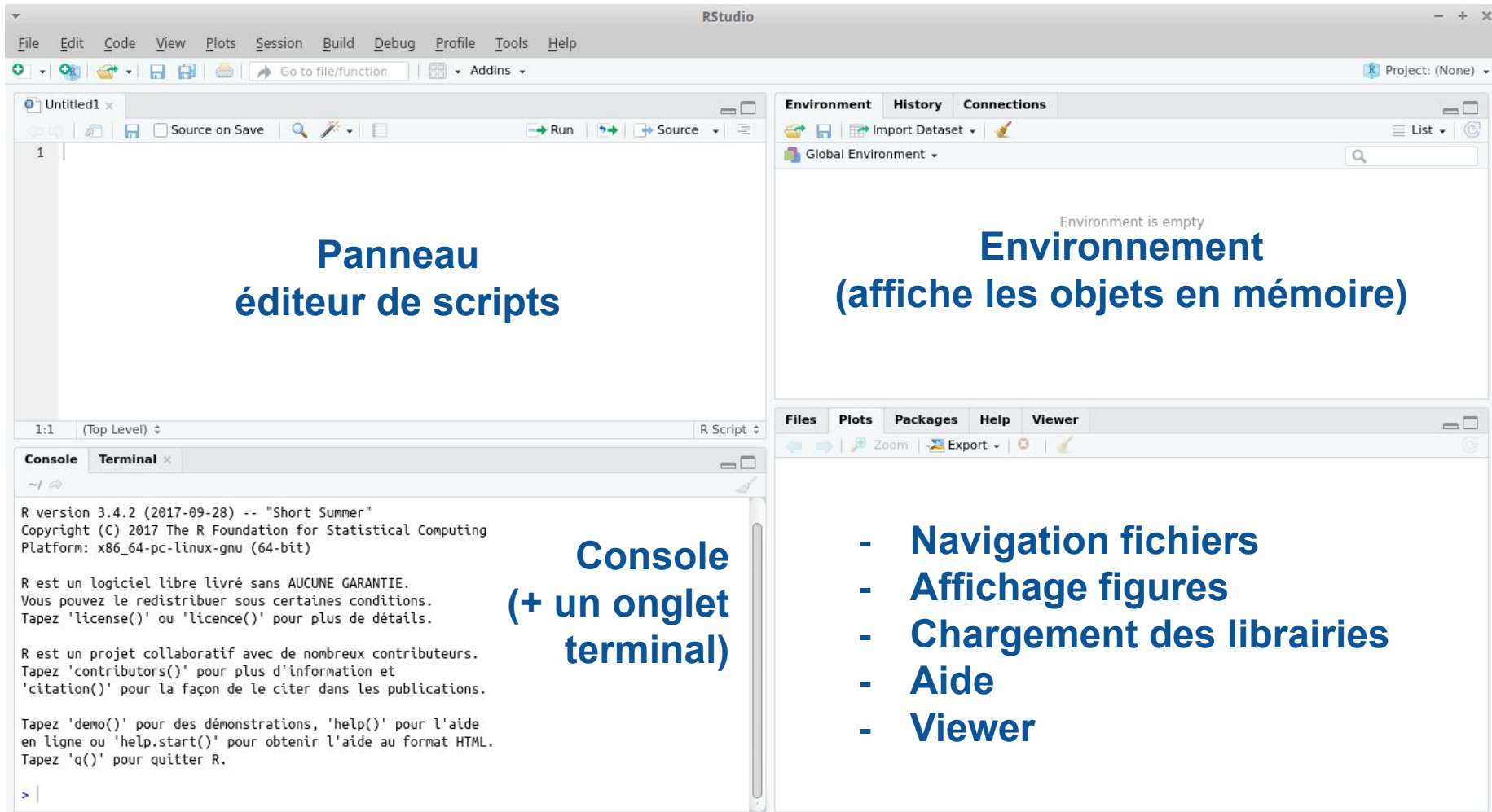
Number of hours
1

Launch

* The RStudio Server: Core session data for this session can be accessed under the [data root directory](#).

RStudio

- Logiciel facilitant l'utilisation de R, via 4 panneaux
- Pour changer l'agencement des panneaux: Tools > Global Options > Pane Layout



R sait tout faire : il compte

Tapez les commandes suivantes dans le panneau Console de RStudio

```
2 + 3
```

```
4 * 5
```

```
6 / 4
```

```
1:10
```

```
8:-9
```

```
1,2
```

```
1.2
```

Notion de variable/objet : valeur numérique

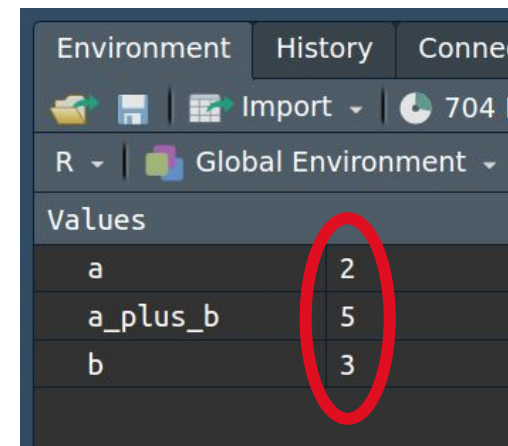
```
a <- 2      ## Créer une variable nommée a et lui assigner une valeur
print(a)   ## Afficher la valeur de la variable a
a          ## Même résultat: si on écrit le nom de variable, R l'imprime
```

```
a_bis = 2   ## Créer une variable nommée a_bis et lui assigner une valeur
a_bis == a  ## Test d'égalité de deux variables
```

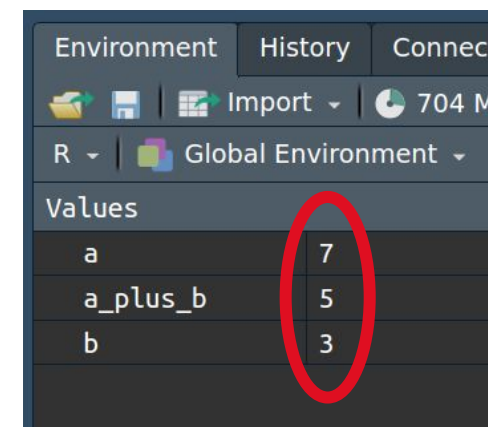
```
b <- 3      ## Assigner une valeur à une seconde variable
a_plus_b <- a + b ## Effectuer un calcul avec 2 variables
print(a_plus_b) ## Afficher le contenu de la variable a_plus_b
```

```
a <- 7      ## Changer la valeur de a
print(a_plus_b) ## Note: le contenu de a_plus_b n'est pas modifié
```

```
a_plus_b <- a + b ## On recalcule a_plus_b
print(a_plus_b)  ## La nouvelle valeur tient compte de la modification de a
```



Environment	History	Connect
R	Global Environment	704 M
Values		
a	2	
a_plus_b	5	
b	3	



Environment	History	Connect
R	Global Environment	704 M
Values		
a	7	
a_plus_b	5	
b	3	

Notion de variable/objet : vecteurs

```
vec1 <- c(1,10)    ## Créer un vecteur
vec2 <- 1:10       ## Créer un vecteur contenant une séquence d'entiers de 1 à 10
vec3 <- c("riri", "fifi", "loulou") ## Vecteur de chaînes de caractères
```

```
vec1      num [1:2] 1 10
vec2      int [1:10] 1 2 3 4 5 6 7 8 9 10
vec3      chr [1:3] "riri" "fifi" "loulou"
```

```
vec2 + a    ## Somme d'un vecteur et d'un nombre
vec2 / 2    ## Diviser un vecteur de nombres par un nombre
vec3 / 2    ## Diviser des chaînes de caractères par un nombre ?
```

Noms de variables **interdits**: TRUE, FALSE, T, F, c, t, pi, data, LETTERS, letters, ...

Cas pratique : manipulation de matrices de données

Cas pratique

Chargement des données

Créer un dossier `intro_R` pour vos résultats de ce TP

The screenshot shows the RStudio interface. The terminal window displays the R version 3.4.4 (2018-03-15) startup message, including the copyright notice and the message "Natural language support but running in an English locale". The Files pane shows the current directory structure, with the "New Folder" button circled in red. A red arrow points from the "New Folder" button to a dialog box.

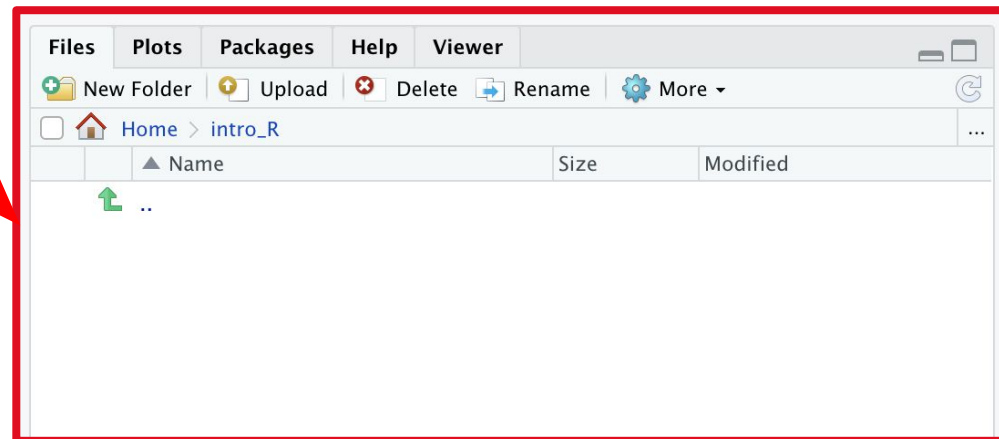
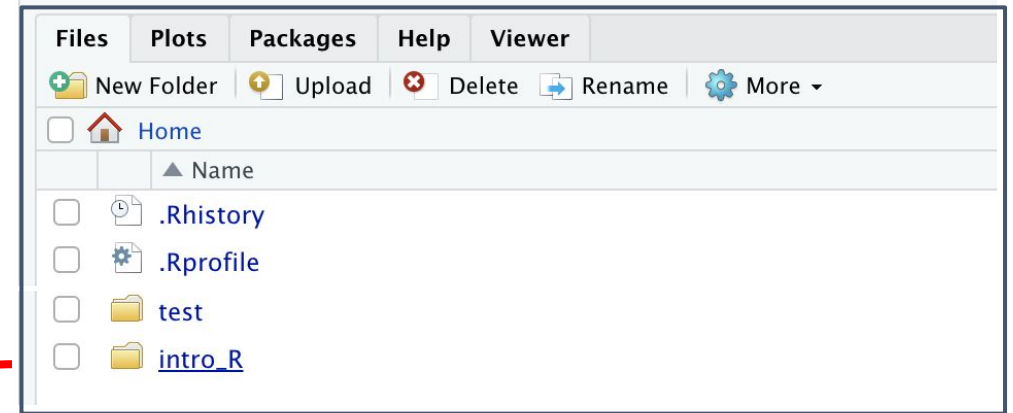
The "New Folder" dialog box is shown, with the text "Please enter the new folder name" and the input field containing "intro_R". The "OK" and "Cancel" buttons are visible at the bottom.

The screenshot shows the RStudio Files pane after the folder creation. The "New Folder" button is still circled in red. The "intro_R" folder is now visible in the file list below the navigation bar.

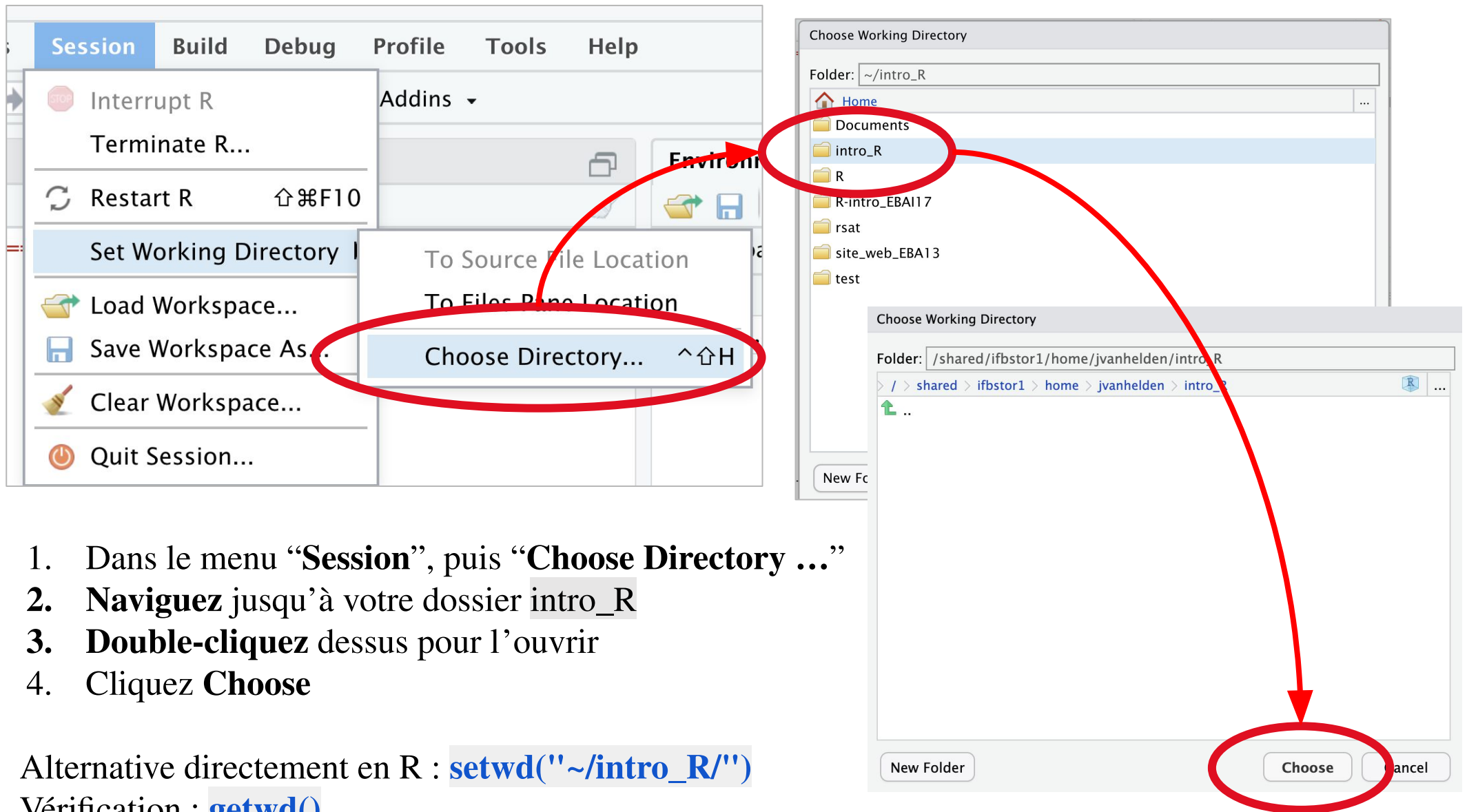
Se déplacer dans le dossier “intro_R”

Double-cliquez sur le dossier “intro_R”, pour vous y déplacer.

Puisque vous venez de créer le dossier il est vide (image du bas).



Définissez votre dossier espace de travail (working directory)



The image shows the RStudio interface with the 'Session' menu open. The 'Choose Directory...' option is highlighted with a red circle. A red arrow points from this option to the first 'Choose Working Directory' dialog box, where the 'intro_R' folder is selected and circled in red. A second red arrow points from the 'intro_R' folder to the 'Choose' button in the second 'Choose Working Directory' dialog box, which is also circled in red.

1. Dans le menu “**Session**”, puis “**Choose Directory ...**”
2. **Naviguez** jusqu’à votre dossier `intro_R`
3. **Double-cliquez** dessus pour l’ouvrir
4. Cliquez **Choose**

Alternative directement en R : `setwd("~/intro_R/")`
Vérification : `getwd()`

Téléchargez les données pour le TD

A partir d'un navigateur, téléchargez et enregistrez **sur votre ordi** les fichiers de données :

- `fruits.tsv` / `.csv` / `.xlsx` : tableau de données de fruits, disponibles en 3 différents formats

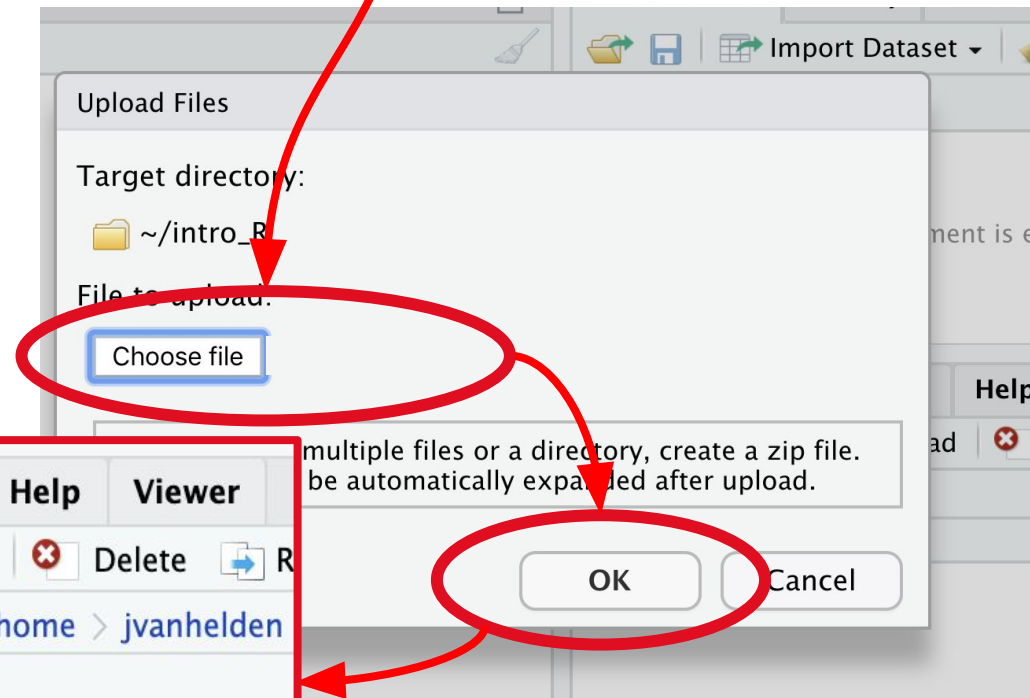
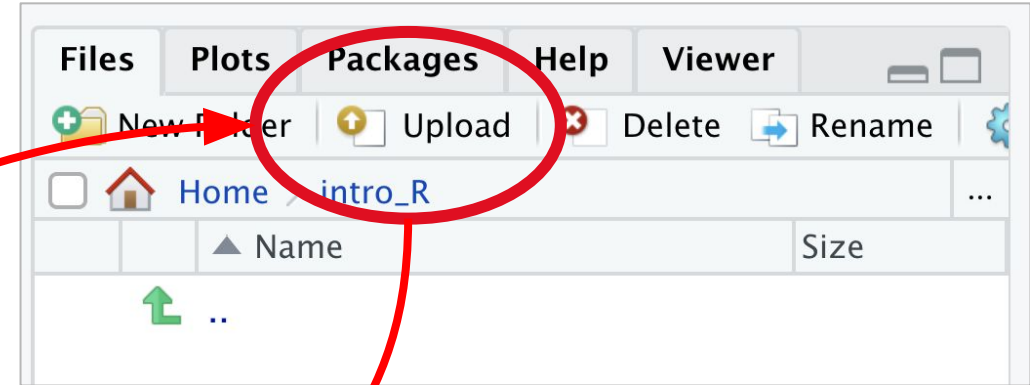
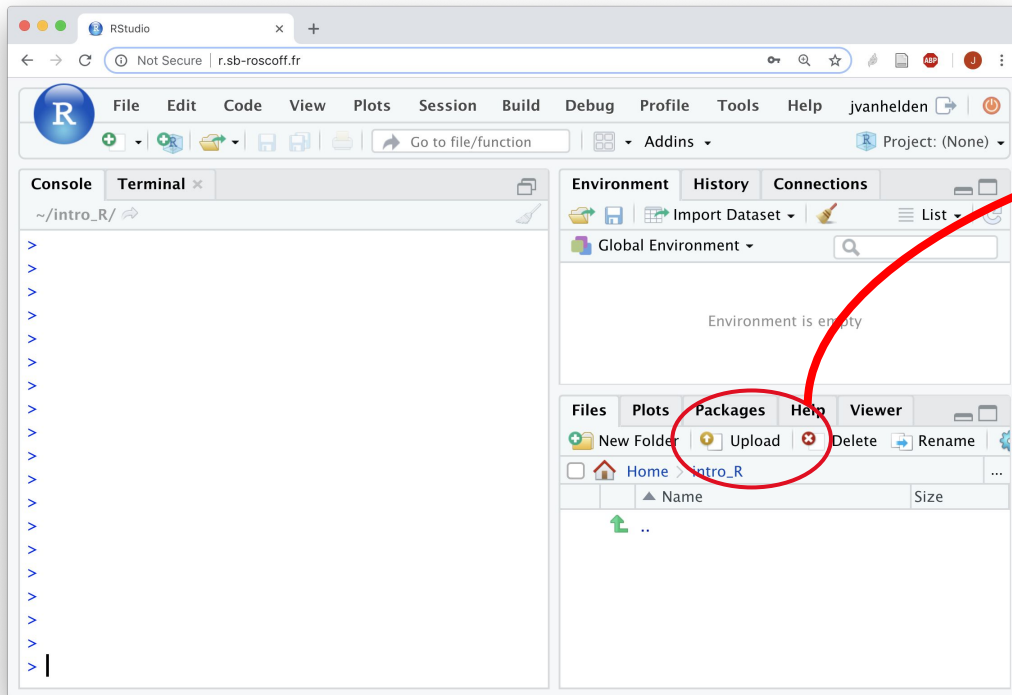
Fichiers disponibles sur le Moodle :

<https://moodle.france-bioinformatique.fr/course/view.php?id=28>

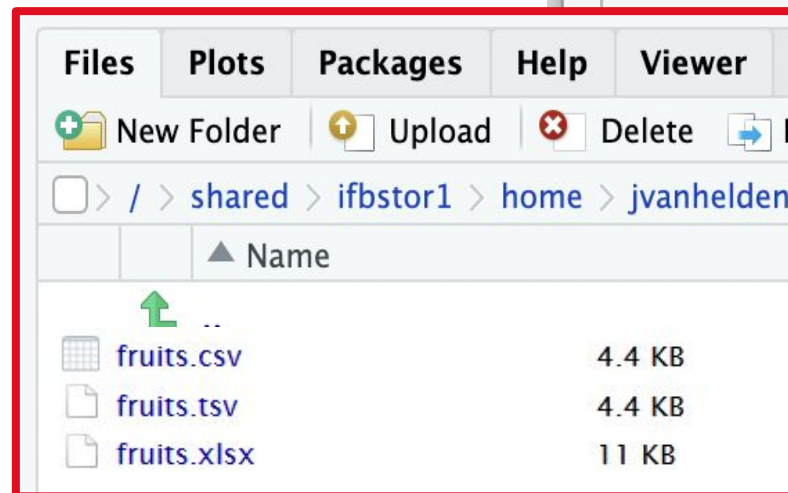
Attention: veuillez à sauvegarder les fichiers

- sous leur nom original,
- avec les extensions respectives (certains navigateurs omettent l'extension, ce qui poserait problème pour la suite du TP)

Téléversement (“upload”) des données

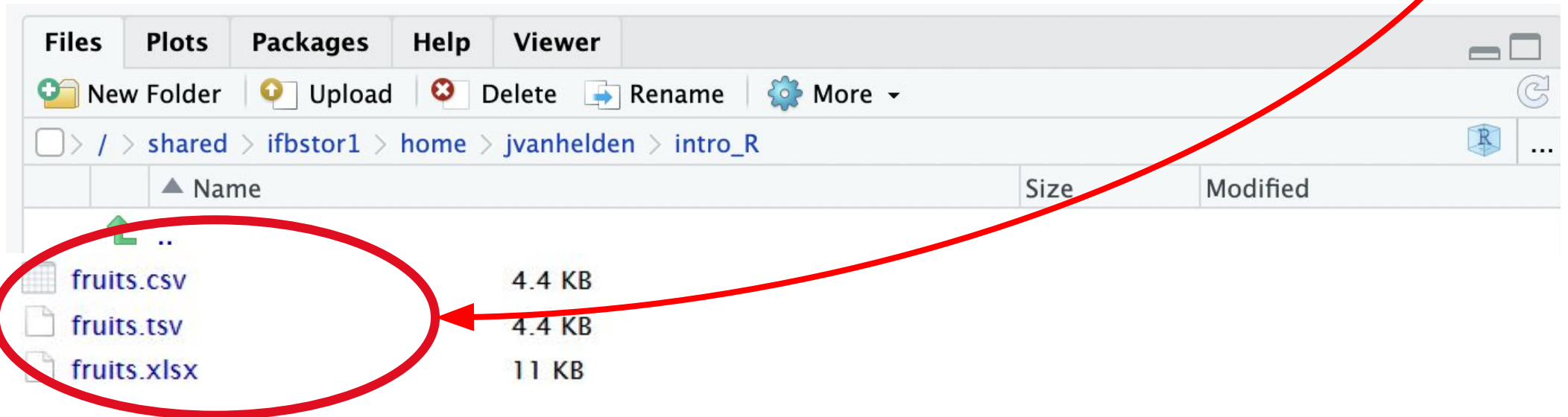
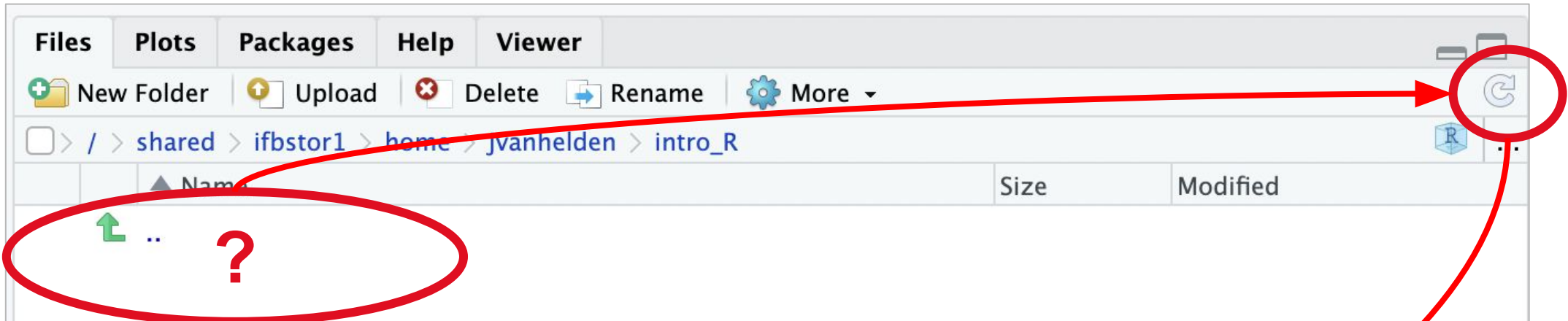


Au moyen du bouton **“Upload”**, téléversez les fichiers depuis votre ordinateur vers votre compte sur le serveur.



Actualisation du dossier

Dans certains cas, il faut actualiser le contenu du dossier pour pouvoir voir le nouveau sous-dossier. Vérifiez ensuite si `intro_R` apparaît bien dans le contenu de votre dossier principal.



Chargement des données (dans la mémoire de R)

Charger le contenu du fichier "expression.txt" dans une variable nommée "exprs".

```
fruits <- read.table(file = "fruits.tsv", header = TRUE, sep = "\t", dec = ".")
```

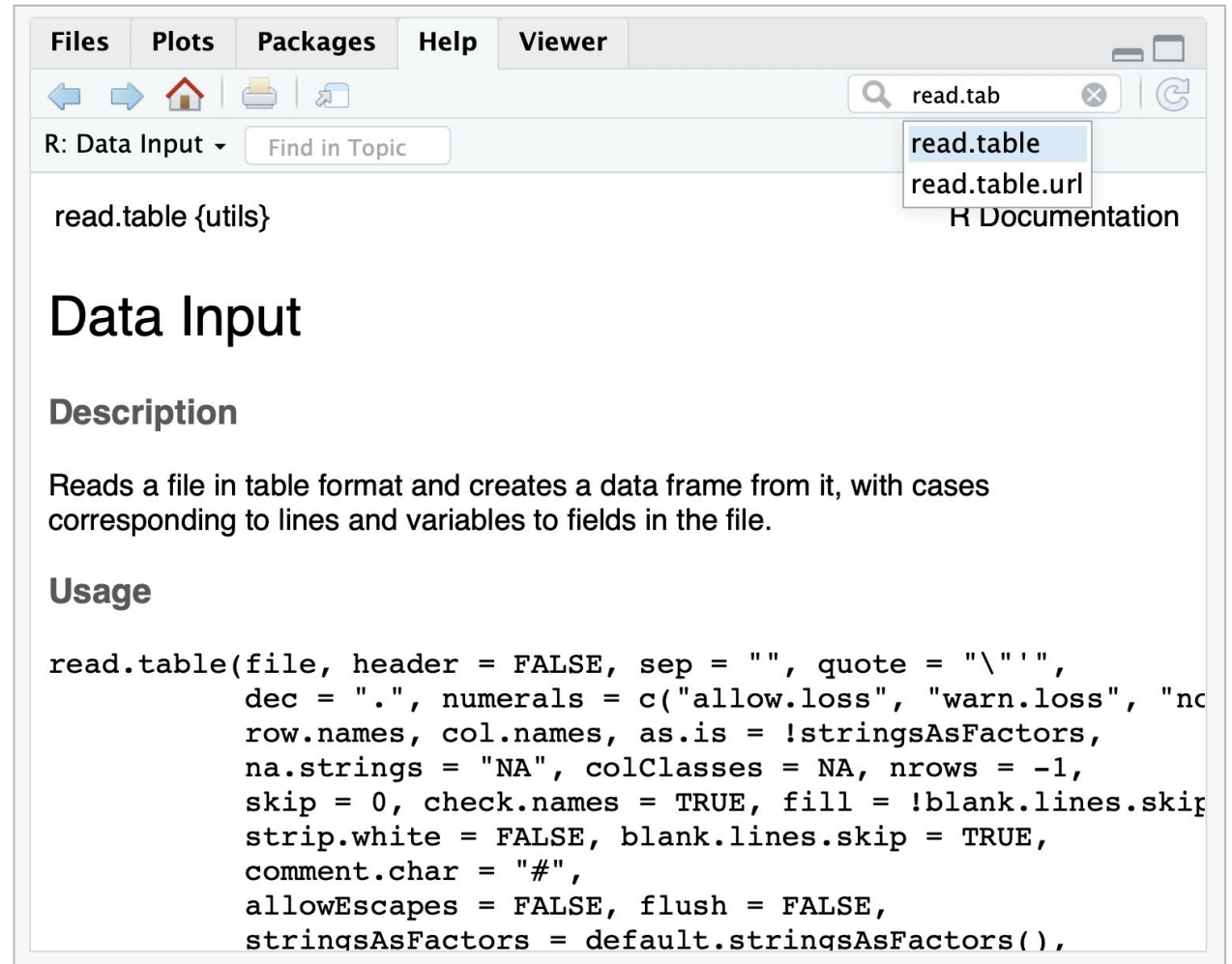
Accéder à l'aide d'une fonction :

```
help(read.table)
```

Notation alternative :

```
?read.table
```

Ou encore, via le Help de RStudio :



The screenshot shows the RStudio Help window for the `read.table` function. The window title is "read.tab" and the search bar contains "read.tab". The search results show "read.table" selected. The main content area displays the following information:

read.table {utils}

Data Input

Description

Reads a file in table format and creates a data frame from it, with cases corresponding to lines and variables to fields in the file.

Usage

```
read.table(file, header = FALSE, sep = "", quote = "\"\"",  
           dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),  
           row.names, col.names, as.is = !stringsAsFactors,  
           na.strings = "NA", colClasses = NA, nrows = -1,  
           skip = 0, check.names = TRUE, fill = !blank.lines.skip,  
           strip.white = FALSE, blank.lines.skip = TRUE,  
           comment.char = "#",  
           allowEscapes = FALSE, flush = FALSE,  
           stringsAsFactors = default.stringsAsFactors(),
```

Il y a beaucoup de fonctions “read.”

Il existe beaucoup de fonctions de base pour lire des **fichiers non R** :

`read.table`, `read.csv`, `read.csv2`, `read.tsv`, `read.delim`, ...

et aussi des packages spécifiques pour des fichiers particuliers :

- Lecture rapide de fichiers textes : `vroom`
- Excel: `readxl`, `openxlsx`, ...

Approche pratique :

1. demander à Google
“*Comment lire une table en R ?*”
2. adapter l'exemple

Pour des fichiers R :

- fichiers `.rds` : `toto = readRDS("path/to/file.rds")`
- fichiers `.rda` : `load("path/to/file.rda")`



Cas pratique

Manipulation d'un tableau de données

Affichage de l'objet "fruits"

Afficher toutes les valeurs

`print(fruits)`

Affichage des premières lignes de l'objet

`head(fruits)`

Affichage des dernières lignes de l'objet

`tail(fruits)`

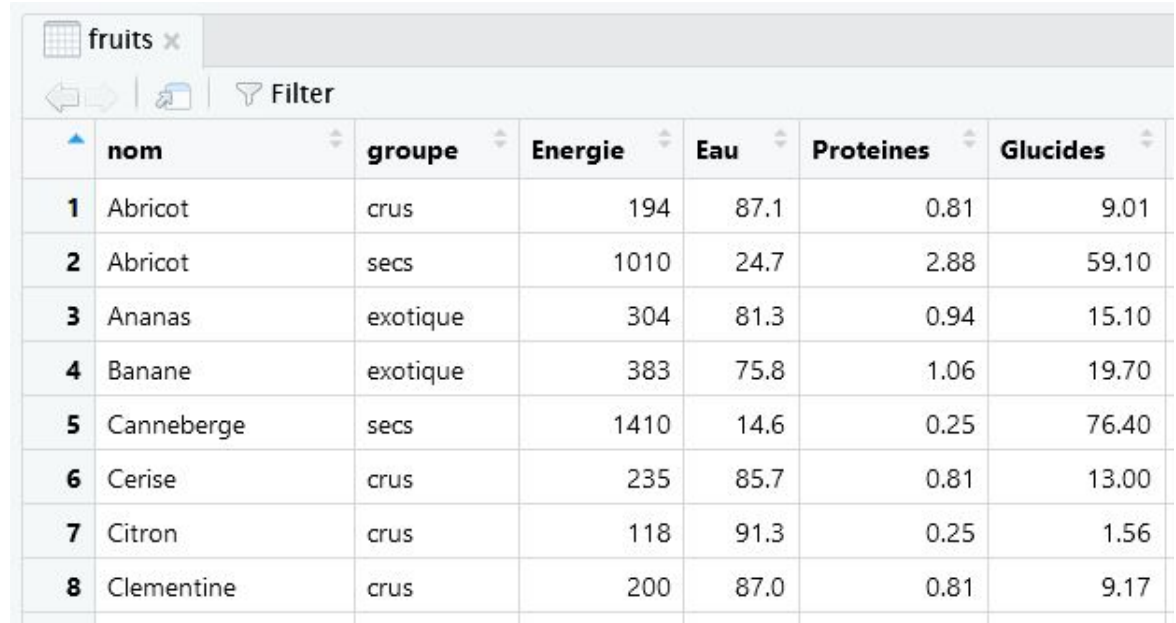
Un peu plus de lignes

`head(fruits, n = 20)`

Explorer le tableau dans un panneau de visualisation

`View(fruits)`

Note: vous pouvez cliquer sur une en-tête de colonne pour trier les données



	nom	groupe	Energie	Eau	Proteines	Glucides
1	Abricot	crus	194	87.1	0.81	9.01
2	Abricot	secs	1010	24.7	2.88	59.10
3	Ananas	exotique	304	81.3	0.94	15.10
4	Banane	exotique	383	75.8	1.06	19.70
5	Canneberge	secs	1410	14.6	0.25	76.40
6	Cerise	crus	235	85.7	0.81	13.00
7	Citron	crus	118	91.3	0.25	1.56
8	Clementine	crus	200	87.0	0.81	9.17

```
> head(fruits, n = 10)
```

```
      nom      groupe Energie  Eau Proteines
1  Abricot      crus   194 87.1     0.81
2  Abricot      secs  1010 24.7     2.88
3  Ananas  exotique   304 81.3     0.94
4  Banane  exotique   383 75.8     1.06
5  Canneberge secs  1410 14.6     0.25
6  Cerise    crus   235 85.7     0.81
7  Citron    crus   118 91.3     0.25
8  Clementine crus   200 87.0     0.81
9  CompoteMultiFruits compote 279 82.9     0.25
10 CompotePomme  compote 432 72.9     0.23
```

Caractéristiques d'un tableau de données

Dimensions :

`ncol(fruits)`

`nrow(fruits)`

`dim(fruits)`

Noms des colonnes et des lignes :

`colnames(fruits)`

`names(fruits)`

`rownames(fruits)`

Résumé rapide des données par colonne :

`summary(fruits)`

`str(fruits)`

À quoi correspondent les sorties de chaque commande ?

Caractéristiques d'un tableau de données

Dimensions :

`ncol(fruits)` ## Nombre de colonnes
`nrow(fruits)` ## Nombre de lignes
`dim(fruits)` ## Dimensions

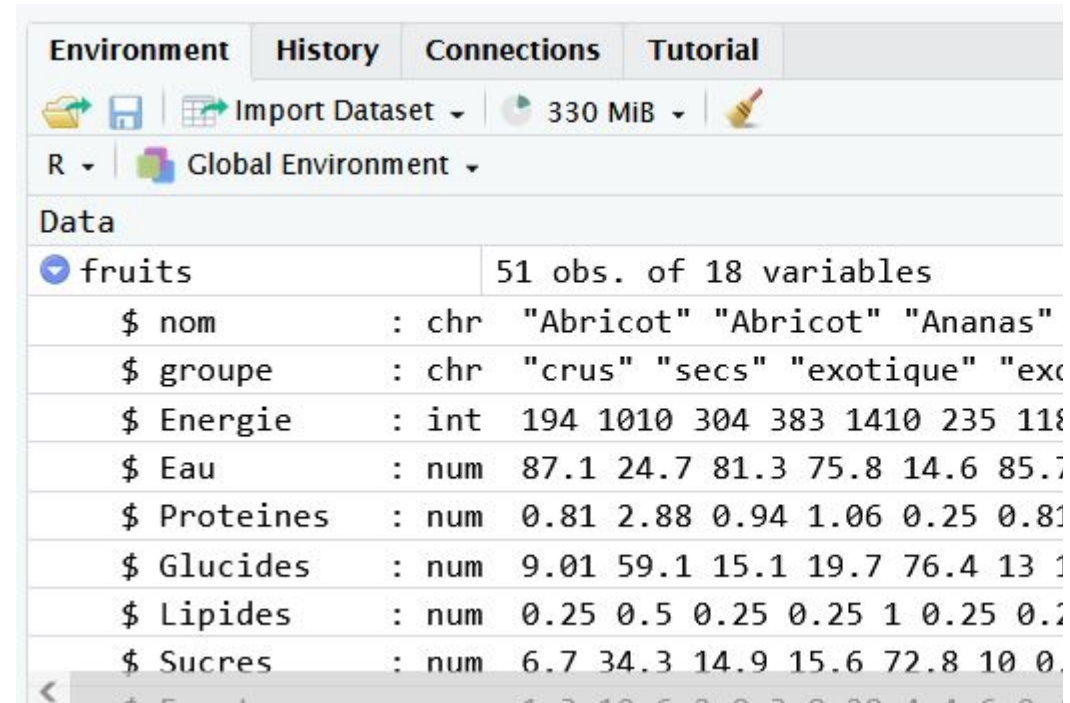
Noms des colonnes et des lignes :

`colnames(fruits)` ## Noms des colonnes
`names(fruits)` ## Noms des colonnes
`rownames(fruits)` ## Noms des lignes

Résumé rapide des données par colonne :

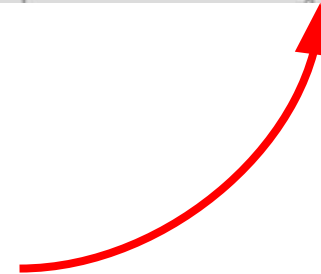
`summary(fruits)` ## Statistiques par colonne
`str(fruits)` ## Structure de la variable

-> Mêmes résultats que dans le panneau "Environment"



The screenshot shows the RStudio Environment pane with the 'fruits' data object selected. The data is summarized as follows:

Variable	Type	Sample Values
nom	chr	"Abricot" "Abricot" "Ananas"
groupe	chr	"crus" "secs" "exotique" "exotique"
Energie	int	194 1010 304 383 1410 235 118
Eau	num	87.1 24.7 81.3 75.8 14.6 85.7
Proteines	num	0.81 2.88 0.94 1.06 0.25 0.81
Glucides	num	9.01 59.1 15.1 19.7 76.4 13.1
Lipides	num	0.25 0.5 0.25 0.25 1 0.25 0.2
Sucres	num	6.7 34.3 14.9 15.6 72.8 10 0.



Affichage de colonnes d'un tableau

Valeurs stockées dans la colonne nommée "nom"

```
fruits$nom
```

Notation alternative

```
fruits[, "nom"]
```

Comment sélectionner plusieurs colonnes ? Par ex, *nom* et *groupe*

Sélection de colonnes par leur indice

```
fruits[, 2]
```

Comment sélectionner les colonnes 2 et 3 ?

et les 5 premières colonnes ?

Affichage de colonnes d'un tableau

Valeurs stockées dans la colonne nommée "nom"

```
fruits$nom
```

Notation alternative

```
fruits[, "nom"]
```

Comment sélectionner plusieurs colonnes ? Par ex, *nom* et *groupe*

```
fruits[, c("nom", "groupe")]
```

Sélection de colonnes par leur indice

```
fruits[, 2]
```

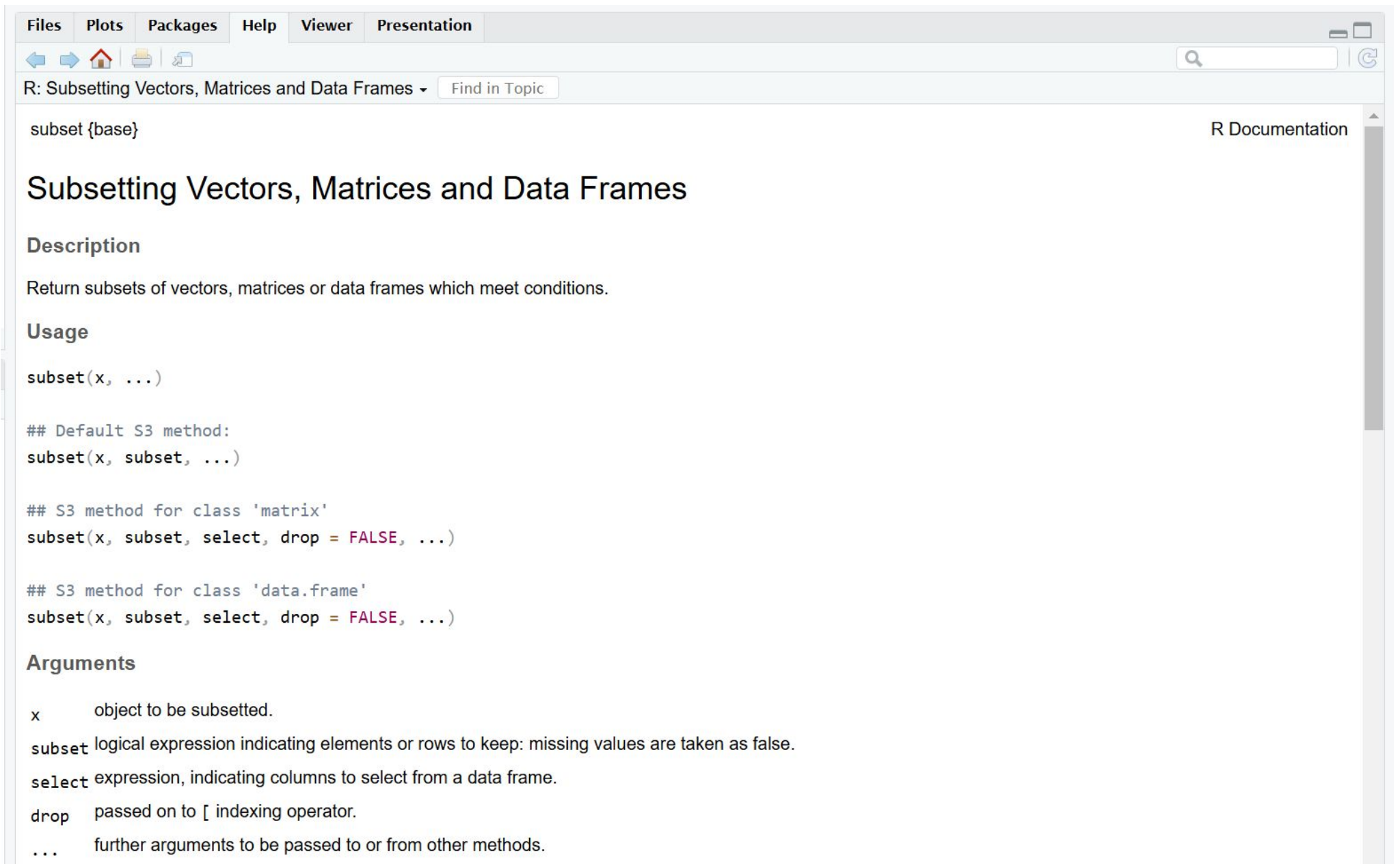
Comment sélectionner les colonnes 2 et 3 ?

```
fruits[, c(3, 2)]
```

et les 5 premières colonnes ?

```
fruits[, c(1:5)]
```

Sélection ou suppression de colonnes d'un tableau -> `subset()`



The image shows a screenshot of the R Documentation website for the `subset()` function. The browser window has tabs for 'Files', 'Plots', 'Packages', 'Help', 'Viewer', and 'Presentation'. The address bar shows the URL 'R: Subsetting Vectors, Matrices and Data Frames'. The main content area is titled 'Subsetting Vectors, Matrices and Data Frames' and includes a 'Description' section stating 'Return subsets of vectors, matrices or data frames which meet conditions.' and a 'Usage' section showing the function signature `subset(x, ...)` and several S3 methods for different classes like 'matrix' and 'data.frame'. The 'Arguments' section lists parameters: `x` (object to be subsetted), `subset` (logical expression), `select` (expression for columns), `drop` (passed on to indexing operator), and `...` (further arguments).

```
subset {base}
```

Subsetting Vectors, Matrices and Data Frames

Description

Return subsets of vectors, matrices or data frames which meet conditions.

Usage

```
subset(x, ...)
```

```
## Default S3 method:  
subset(x, subset, ...)
```

```
## S3 method for class 'matrix'  
subset(x, subset, select, drop = FALSE, ...)
```

```
## S3 method for class 'data.frame'  
subset(x, subset, select, drop = FALSE, ...)
```

Arguments

`x` object to be subsetted.

`subset` logical expression indicating elements or rows to keep: missing values are taken as false.

`select` expression, indicating columns to select from a data frame.

`drop` passed on to [indexing operator.

`...` further arguments to be passed to or from other methods.

Sélection ou suppression de colonnes d'un tableau

Sélectionner la première colonne de fruits :

```
fruitsbis <- subset(fruits, select = nom)
```

```
fruitsbis <- subset(fruits, select = 1)
```

Supprimer une ou plusieurs colonnes de fruits avec leur noms :

```
fruitsbis <- subset(fruits, select = -nom)
```

```
fruitsbis <- subset(fruits, select = -c(nom, Glucides, Sucres))
```

Supprimer la première colonne de fruits avec son index :

```
fruitsbis <- subset(fruits, select = -1)
```

Supprimer plusieurs colonnes de fruits avec leur index :

```
fruitsbis <- subset(fruits, select = -c(2, 3))
```

Il existe aussi des packages spécifiques pour faire cela: **tidyr**, **dplyr**, ...



Cas pratique

Réalisation de figures

Figures en R

Chargement des données (poids de poussins suivant 4 régimes différents au cours du temps):

```
data("ChickWeight")  
summary(ChickWeight)
```

Sélection des données pour T=21 :

```
dta <- ChickWeight[ChickWeight$Time == 21, ]
```

Affichage du design :

```
table(dta$Diet)
```

Visualisation des données avec **boxplot()** (du package de base, **graphics**):

```
boxplot(weight ~ Diet, data = dta)  
stripchart(weight ~ Diet, data = dta, add = TRUE, vertical = TRUE)
```

Visualisation avec **ggplot()** (du package **ggplot2**):

```
ggplot(data = dta, mapping = aes(x = Diet, y = weight, fill = Diet)) +  
  geom_boxplot() +  
  geom_jitter(width = 0.1, height = 0)
```

Take home messages

- Tout est faisable avec R
- **Définir et comprendre l'opération mathématique/statistique** avant de chercher la fonction R correspondante
- R est un langage :
 - plusieurs types et structures de données (out of scope)
 - énormément de commandes à découvrir (out of scope)
 - Google est votre ami
- Une infinité de :
 - ressources en ligne
 - tutoriels pour des analyses spécifiques (e.g. DESeq2 pour le RNA-Seq)
- Pour aller plus loin : R Markdown
 - <https://bookdown.org/yihui/rmarkdown/>
 - <https://rmarkdown.rstudio.com/>



Diverses ressources

Bonnes pratiques en R: propre, ordonné, cohérent et lisible

Voir les bonnes pratiques de syntaxe : <https://style.tidyverse.org/syntax.html>

Quelques exemples :

	Bien	Pas bien
nettoyer l'environnement	<code>rm(list = ls())</code>	confondre des objets
donner du sens au nom des variables	<code>data_wt, data_ko</code>	<code>data1, data2,</code> <code>data2final,</code> <code>data2finalvrai</code>
écrire des commentaires partout	avec <code>#</code> ou <code>##</code> , etc	0 commentaire
aérer le code	avec des espaces et des sauts de ligne	un gros bloc de 3km

Portail Open OnDemand : <https://ondemand.cluster.france-bioinformatique.fr>

- **RStudio**



- **Jupyter lab**



Une question ? Un besoin ? Un problème ? **Contactez la communauté IFB**

<https://community.france-bioinformatique.fr/>



Ressources

Base R Cheat Sheet

Advanced R Cheat Sheet

RStudio IDE :: CHEAT SHEET

R Markdown :: CHEAT SHEET

What is R Markdown?

Workflow

render

Embed code with knitr syntax

Interactive Documents

GLOBAL OPTIONS

IMPORTANT CHUNK OPTIONS

Session

Environment

EM

ct

History

with a

sh

age

Delete

from

Library

pane

/apps

post

refresh

data set

help

check

value

of 2016-01

R Studio

RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at rmarkdown.rstudio.com • rmarkdown 1.6 • Updated: 2016-02

R

<https://www.r-project.org/>

RStudio

<https://rstudio.com/>

R-bloggers

<https://www.r-bloggers.com/>

THINKR

<https://thinkr.fr/>

Rstudio Cheatsheets (un tas de thèmes):

<https://rstudio.com/resources/cheatsheets/>

Un peu de publicité pour R Markdown

Avec R Markdown :

- meilleure traçabilité
- meilleure répétabilité
- possibilité de visualiser:
 - code
 - résultats textuels
 - résultats graphiques



Ressources :

<https://bookdown.org/yihui/rmarkdown/>

<https://rmarkdown.rstudio.com/>

The screenshot displays the RStudio interface. On the left, the 'chunks.Rmd' editor shows R code chunks. The first chunk is titled 'R Code Chunks' and contains text explaining that R code chunks can include plots. The second chunk is an R code block that sets options for a plot, loads the 'ggplot2' library, summarizes the 'cars' dataset, and creates a scatter plot with a smoothed trend line. On the right, the 'RStudio: Preview HTML' window shows the rendered output. It includes the title 'R Code Chunks', the explanatory text, the R code from the second chunk, a summary table of the 'cars' dataset, and the resulting scatter plot with a smoothed trend line.

```
1 R Code Chunks
2 =====
3
4 With R Markdown, you can insert R code
5 chunks including plots:
6
7 ```{r qplot, fig.width=4, fig.height=3,
8 message=FALSE}
9 # quick summary and plot
10 library(ggplot2)
11 summary(cars)
12 qplot(speed, dist, data=cars) +
13   geom_smooth()
```

##	speed	dist
##	Min. : 4.0	Min. : 2
##	1st Qu.: 12.0	1st Qu.: 26
##	Median : 15.0	Median : 36
##	Mean : 15.4	Mean : 43
##	3rd Qu.: 19.0	3rd Qu.: 56
##	Max. : 25.0	Max. : 120

qplot(speed, dist, data = cars) + geom_smooth()