

(Re)découverte de R... en 1h45

Ecole de Bioinformatique IFB Inserm Inrae – Roscoff – Nov. 2025

Elise Jacquemet elise.jacquemet@pasteur.fr

Mathieu Genete mathieu.genete@univ-lille.fr

Elodie Darbo elodie.darbo@u-bordeaux.fr

Charlotte Berthelie charlotte.berthelie@sb-roscoff.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, nettoyer, exporter
2. faire des analyses statistiques plus ou moins complexes : description, exploration, modélisation ...
3. créer de jolies figures, rapports ...

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



Interfaces graphiques pour programmer en R :



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 2025 : R 4.5.2 - R est vivant 😎

Avantages et inconvénients

Avantages :

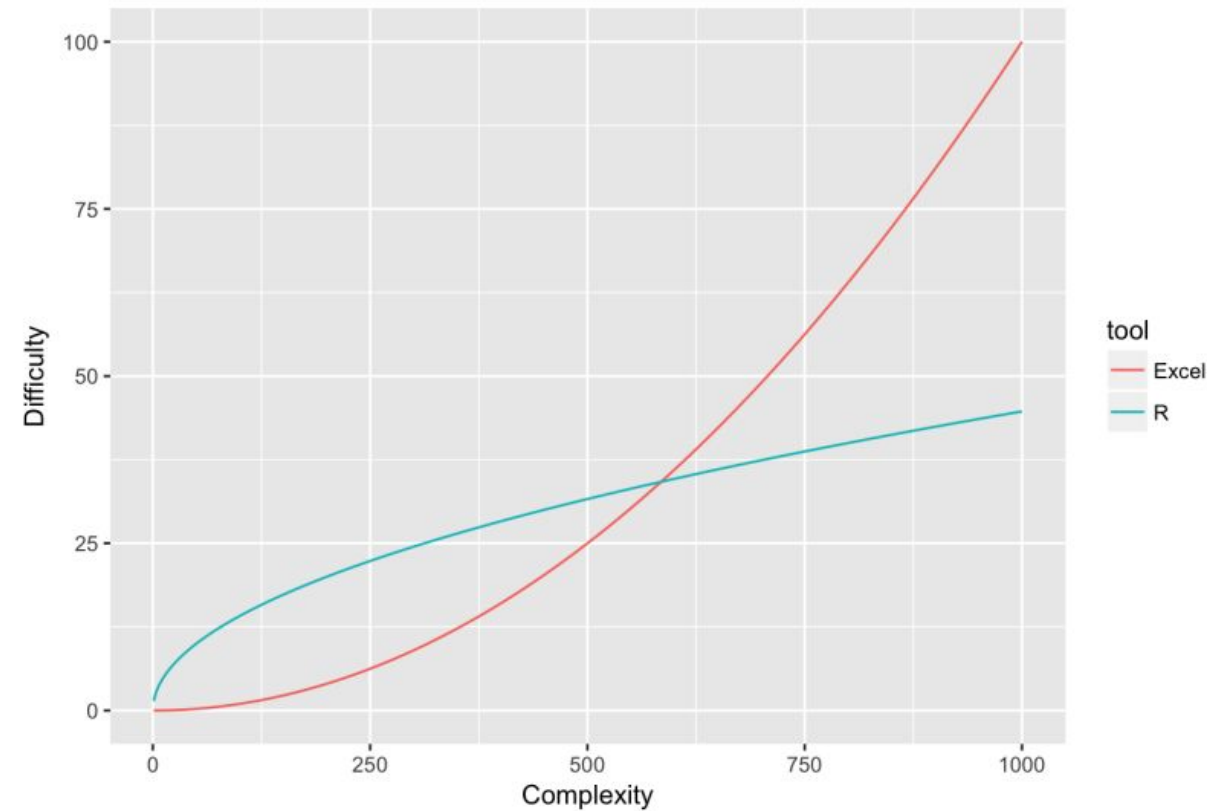
- Souplesse d'utilisation pour réaliser des analyses statistiques
- Libre et gratuit, même s'il existe maintenant des versions payantes de RStudio (shiny et/ou server)
- Traçabilité des analyses en écrivant les commandes R dans des notebooks (Rmd)
- Large communauté d'utilisateurs + aide en ligne
- Grand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



Source: R-bloggers

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.

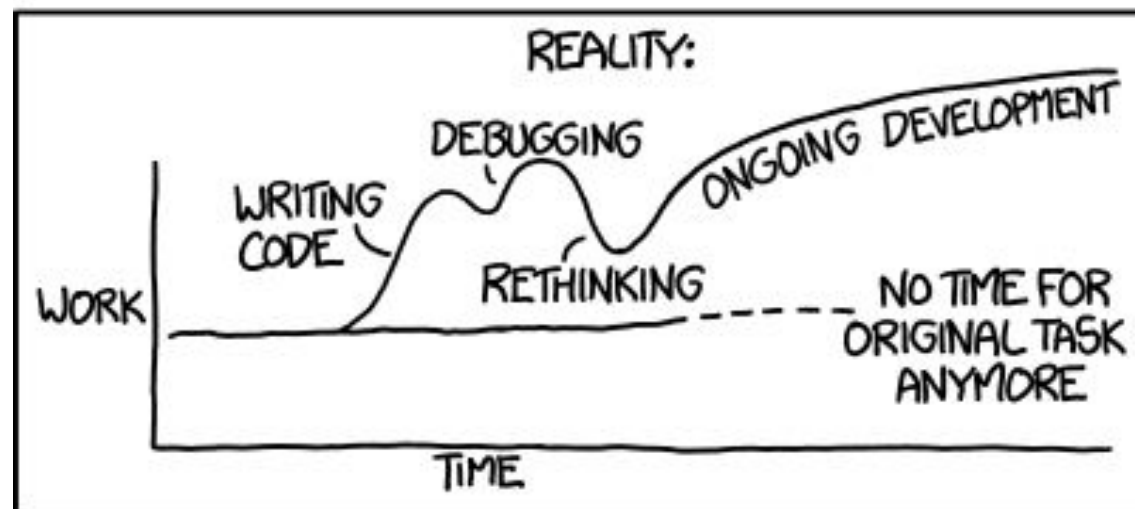
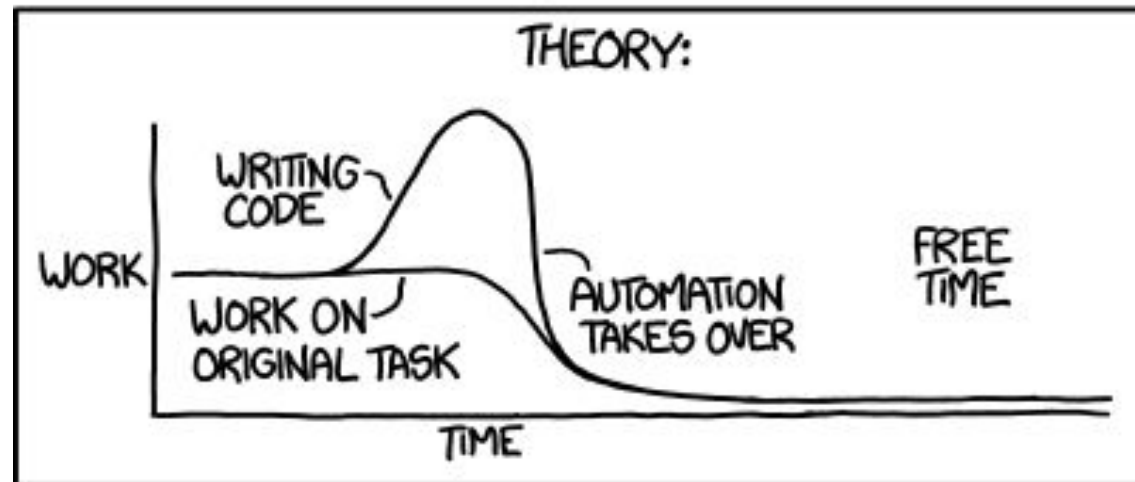
[Lire plus tard](#) [Europe](#) [Partager](#) [Commenter](#)



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

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

Read a table of data

```
read.table()
```

Merge two tables

```
merge()
```

Filter rows

```
data[data$x > 10]
```

Search for a string

Select columns

```
data[,c("x", "y")]
```

```
grep()
```

Calculate an average

Carry out a PCA

```
mean()
```

```
prcomp()
```

Add two matrices

```
mat1 + mat2
```

Export a table of data

```
write.table()
```

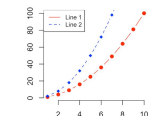
```
var()
```

Linear regression

Calculate a variance

```
lm()
```

```
plot()
```



Test a hypothesis

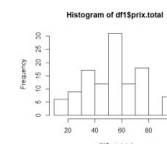
Draw a curve

```
t.test()
```

Draw a histogram

Convert data

```
hist()
```



```
as.matrix()
```

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

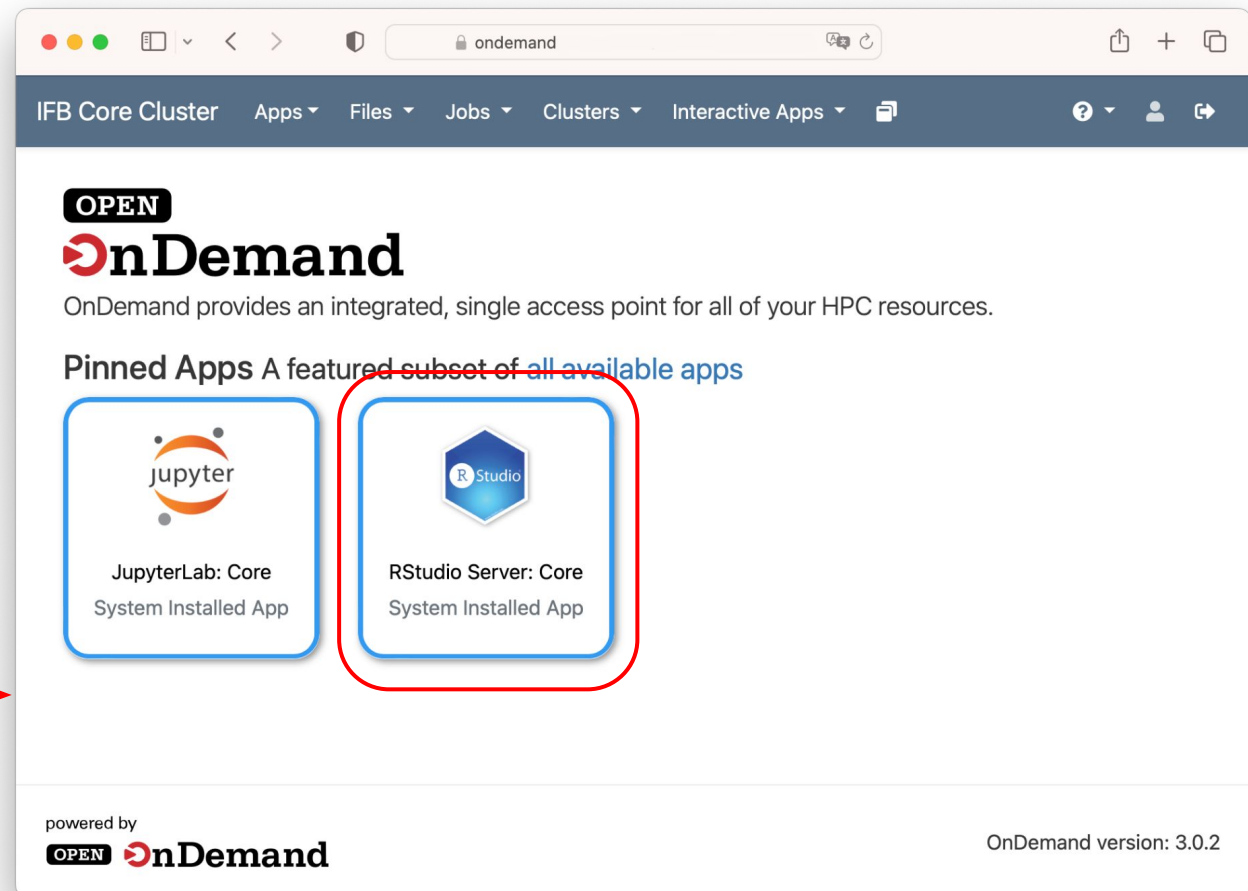
Username

Password

Login

Connexion avec les
identifiants IFB
Identifiant: *pnom*

Grant Access



Connexion à RStudio via OpenOnDemand

Remplir avec les ressources :

- Reservation: 2538_eb3i_n1_2025

Et tout le reste par défaut.

Puis:

Launch

Home / My Interactive Sessions / RStudio Server: Core

Interactive Apps

Desktops

Desktop: Core

R Shiny

RTrainer: Core

Servers

JupyterLab: Core

RStudio Server: Core

Visual Studio Code: Core

RStudio Server: Core

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

R version

4.4.1

Reservation

No reservation

Account

2538_eb3i_n1_2025

Partition

fast

Number of CPUs

4

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

Amount of memory

4G

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

8

☐ Clear my RStudio profile

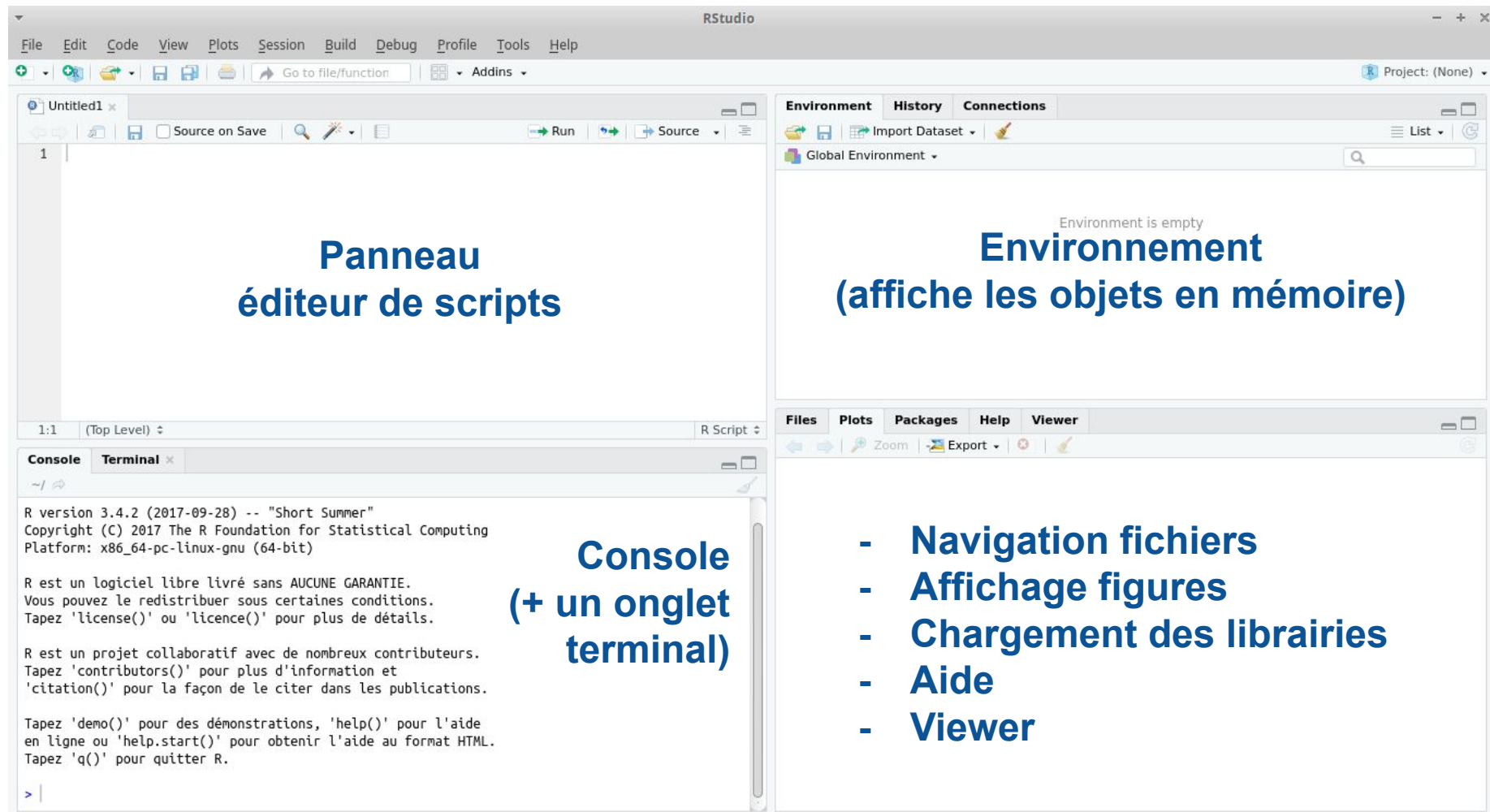
Check this box if RStudio keeps crashing at startup or you would like to reset your RStudio profile. This will remove RStudio sessions and cache from your home folder (.rstudio, .RData, etc.).

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 > Panel Layout



R sait tout faire : il compte

Tapez les commandes suivantes dans le panneau Console de RStudio

```
2 + 3
```

```
4 * 5
```

```
6 / 4
```

```
1, 2
```

```
1.2
```

Notion de variable/objet : valeur numérique

```
a <- 2  
print(a)  
a
```

Créer une variable nommée a et lui assigner une valeur
Afficher la valeur de la variable a
Même résultat: si on écrit le nom de variable, R l'affiche

```
a_bis = 2  
a_bis == a
```

Créer une variable nommée a_bis et lui assigner une valeur
Test d'égalité de deux variables

```
b <- 3  
a_plus_b <- a + b  
print(a_plus_b)
```

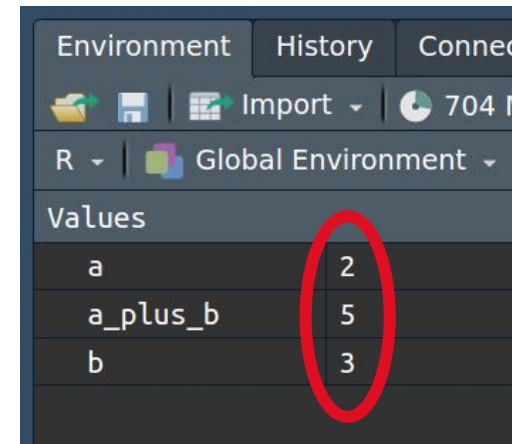
Assigner une valeur à une seconde variable
Effectuer un calcul avec 2 variables
Afficher le contenu de la variable a_plus_b

```
a <- 7  
print(a_plus_b)
```

Changer la valeur de a
Note: le contenu de a_plus_b n'est pas modifié

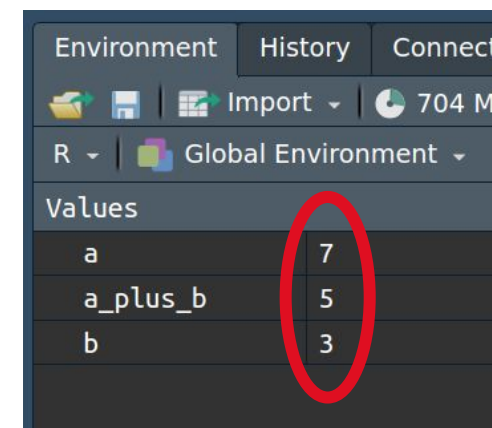
```
a_plus_b <- a + b  
print(a_plus_b)
```

On recalcule a_plus_b
La nouvelle valeur tient compte de la modification de a



The screenshot shows the R Studio Environment pane with the 'Global Environment' selected. The 'Values' table lists three variables: 'a' with value 2, 'a_plus_b' with value 5, and 'b' with value 3. The values 2, 5, and 3 are circled in red.

Variable	Value
a	2
a_plus_b	5
b	3



The screenshot shows the R Studio Environment pane after the variable 'a' has been updated to 7. The 'Values' table now shows 'a' with value 7, 'a_plus_b' with value 5, and 'b' with value 3. The values 7, 5, and 3 are circled in red.

Variable	Value
a	7
a_plus_b	5
b	3

Notion de variable/objet : vecteurs

```
mesures <- c(1,10)    ## Créer un vecteur
entiers <- 1:10        ## Créer un vecteur contenant une séquence d'entiers de 1 à 10
canards <- c("riri", "fifi", "loulou") ## Vecteur de chaînes de caractères
test <- c("riri", 1, 4:6) ## Un vecteur ne contient qu'une seule classe d'objets
```

canards	chr	[1:3]	"riri"	"fifi"	"loulou"
entiers	int	[1:10]	1	2	3 4 5 6 7 8 9 10
mesures	num	[1:2]	1	10	
test	chr	[1:5]	"riri"	"1"	"4" "5" "6"

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

```
entiers[2]     ## Accéder au 2e élément du vecteur
entiers[2] <- 12 ## Changer la valeur du 2e élément du vecteur
```



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

Notion de variable/objet : liste

```
empty <- list()           ## Créer une liste vide
courses <- list("banane"=2,"pomme"=5,"kiwi"=8)
vrac <- list("A"=1:4,"B"=c(0,1),"C"="hey")
## La liste peut contenir plusieurs classes d'objets
```

```
courses$banane           ## Accéder à la valeur de banane
courses[["banane"]]      ## équivalent
courses[[1]]             ## équivalent
```

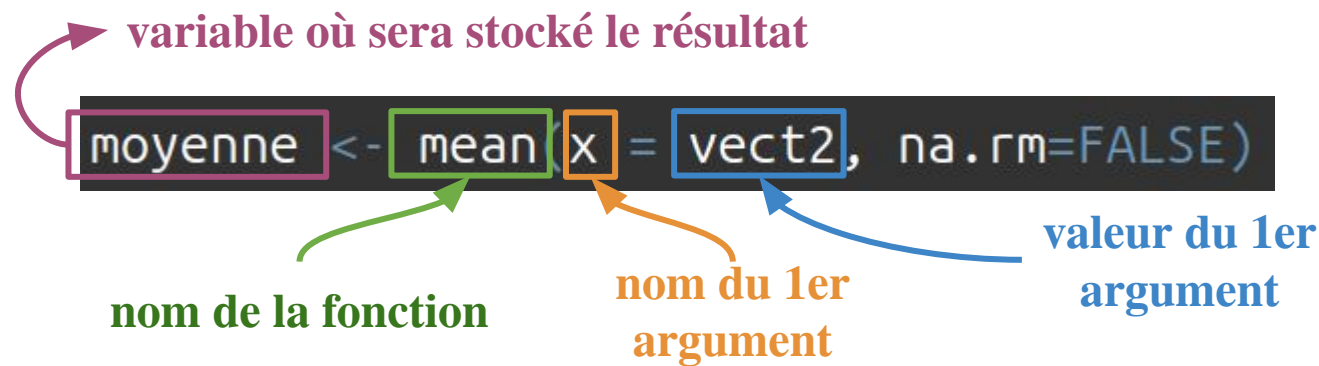
```
courses$banane <- 1      ## Changer une valeur dans la liste
```

```
empty[["clé"]] <- "valeur" ## Ajoute un élément à la liste vide
empty[4] <- 10             ## Le 4e élément de la liste prend la valeur 10
```

```
▼ courses      List of 3
  $ banane: num 1
  $ pomme : num 5
  $ kiwi  : num 8
empty          List of 0
▼ vrac         List of 3
  $ A: int [1:4] 1 2 3 4
  $ B: num [1:2] 0 1
  $ C: chr "hey"
```

```
▼ empty        List of 4
  $ clé: chr "valeur"
  $    : NULL
  $    : NULL
  $    : num 10
```


Notion de fonction



Variable vs argument



Les arguments sont soit :

- nommés → ils peuvent être écrits dans n'importe quel ordre
- non nommés → R prendra les valeurs dans l'ordre d'apparition

```
moy <- mean(vec2)
```

```
moy <- mean(x=vec2)
```

```
moy <- mean(x=vec2, na.rm=TRUE)
```

```
## calcule la moyenne des valeurs numériques du vecteur
```

```
## équivalent
```

```
## ignore les valeurs manquantes (défaut)
```

⇒ Possible de créer ses propres fonctions :
pour simplifier des tâches répétitives, reproduire à l'identique ...

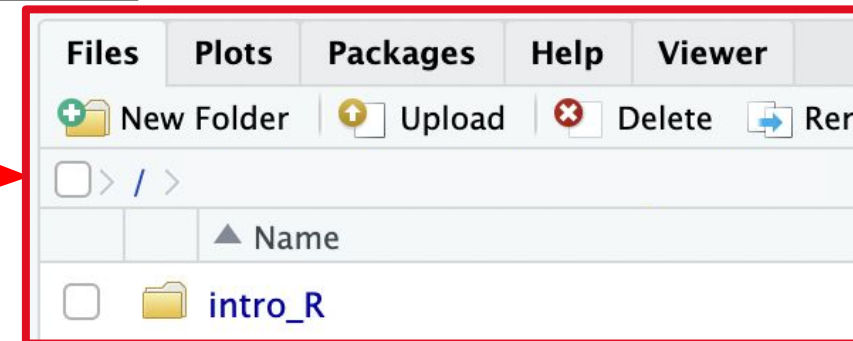
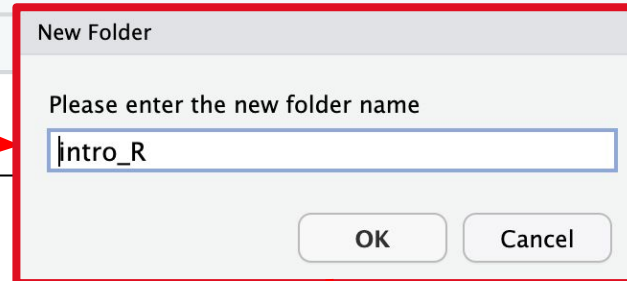
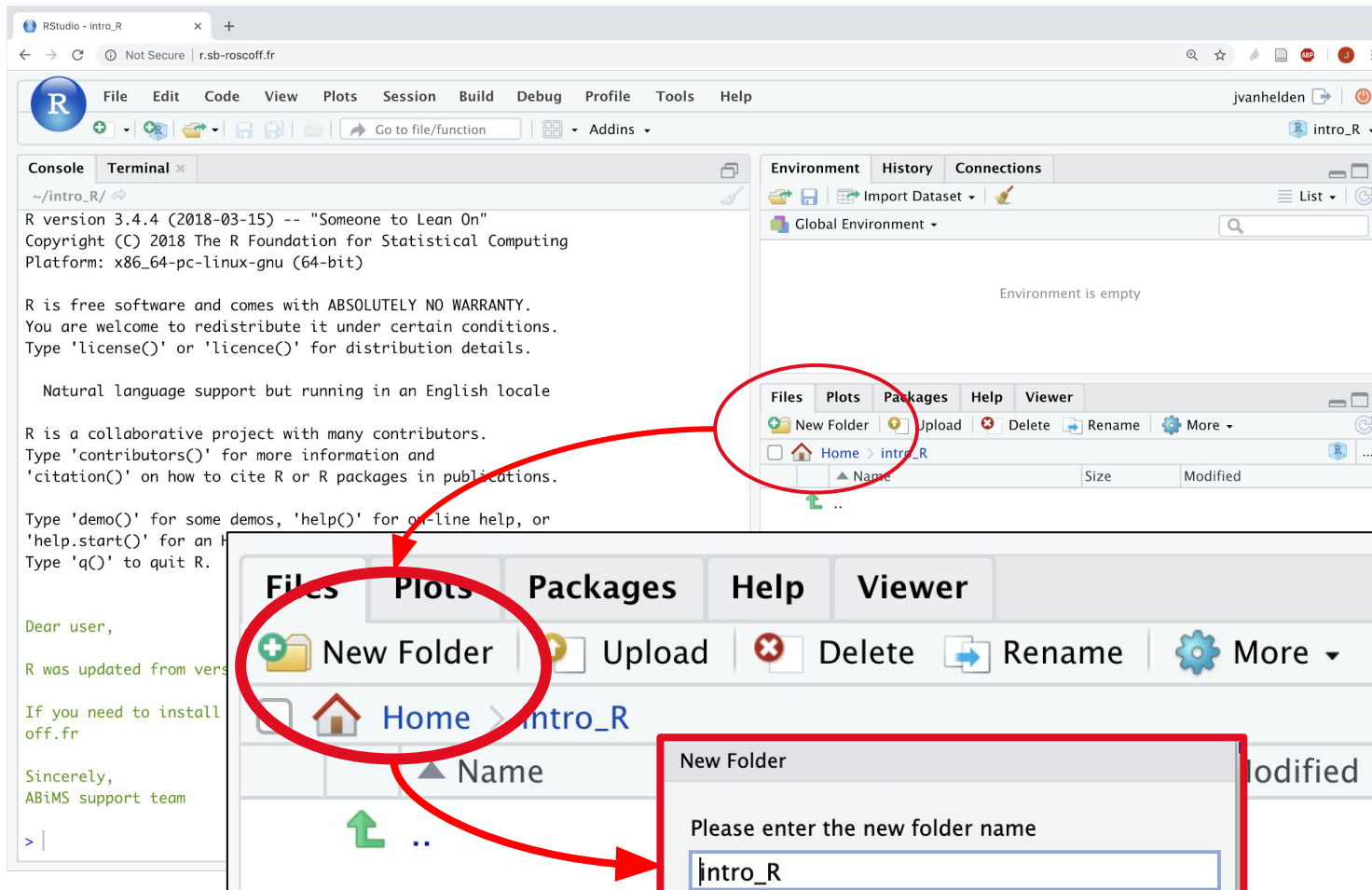
```
square <- function(x) {  
  # as many lines as you want here  
  result = x*x  
  return(result)  
}
```

Cas pratique : manipulation de matrices de données

Cas pratique

Chargement des données

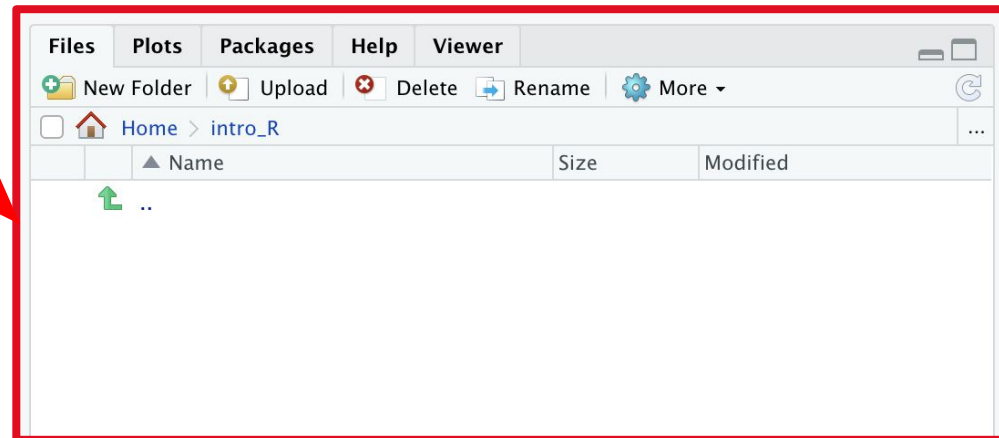
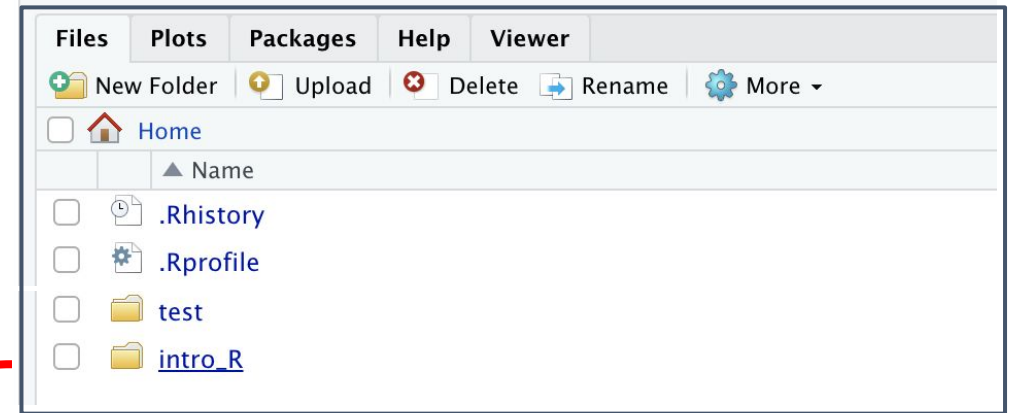
Créer un dossier `intro_R` pour vos résultats de ce TP



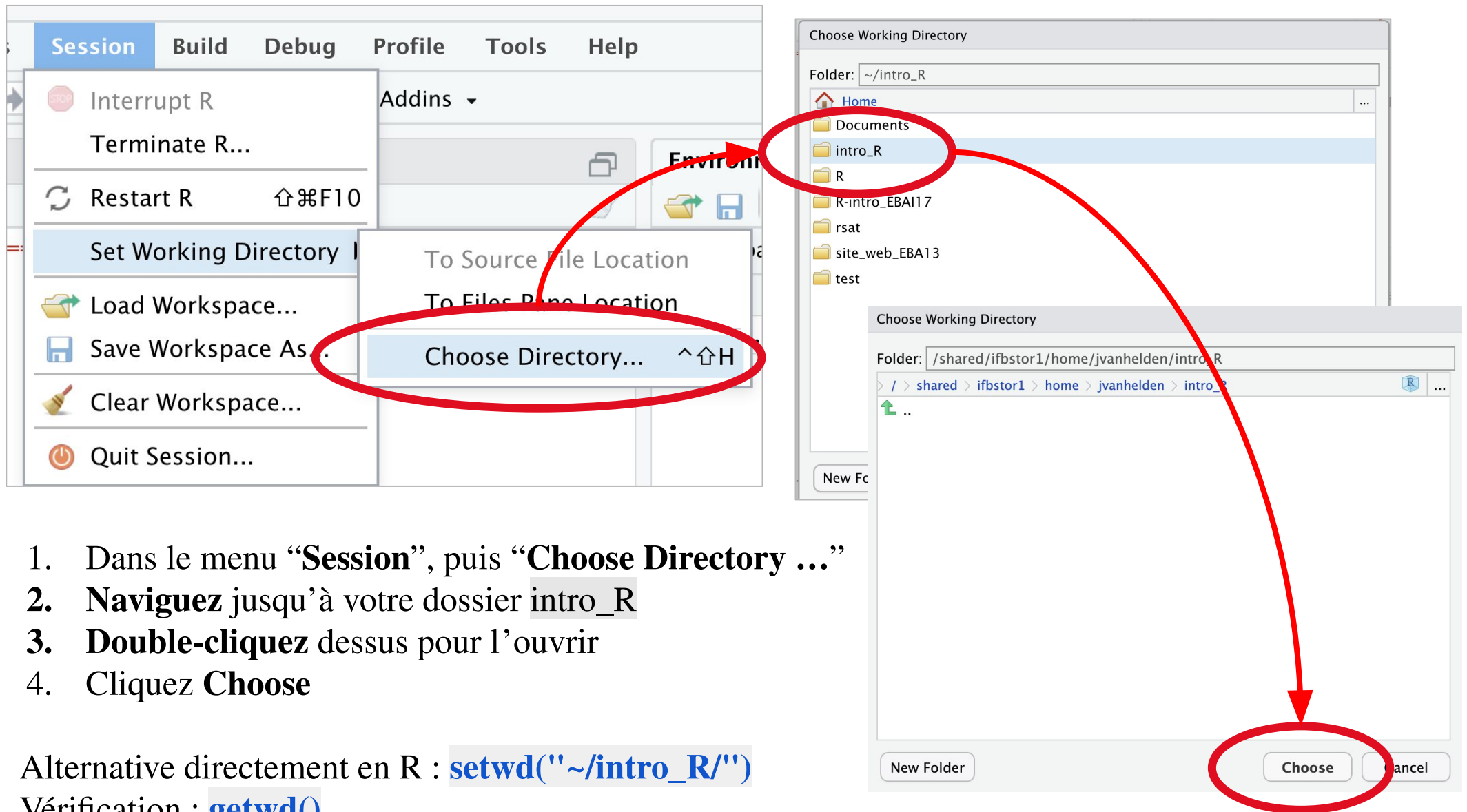
Se déplacer dans le dossier “intro_R”

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 illustrates the steps to set the working directory in RStudio. It shows the 'Session' menu with 'Choose Directory...' highlighted. Two 'Choose Working Directory' dialog boxes are shown: the first shows the 'intro_R' folder selected, and the second shows the full path '/shared/ifbstor1/home/jvanhelden/intro_R' and the 'Choose' button highlighted.

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

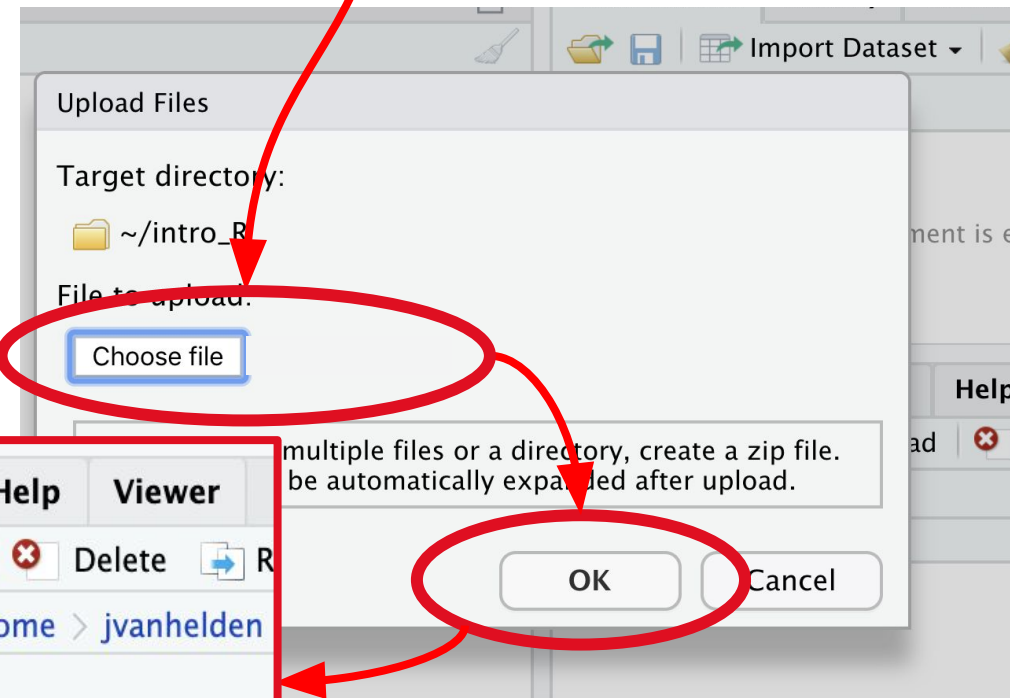
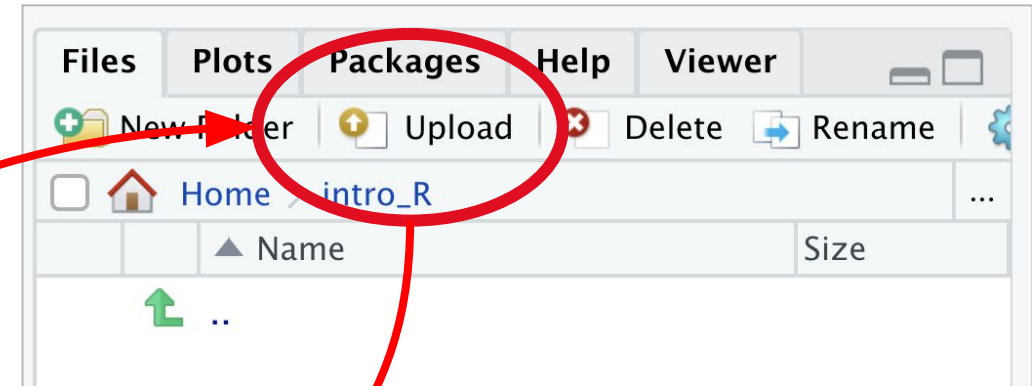
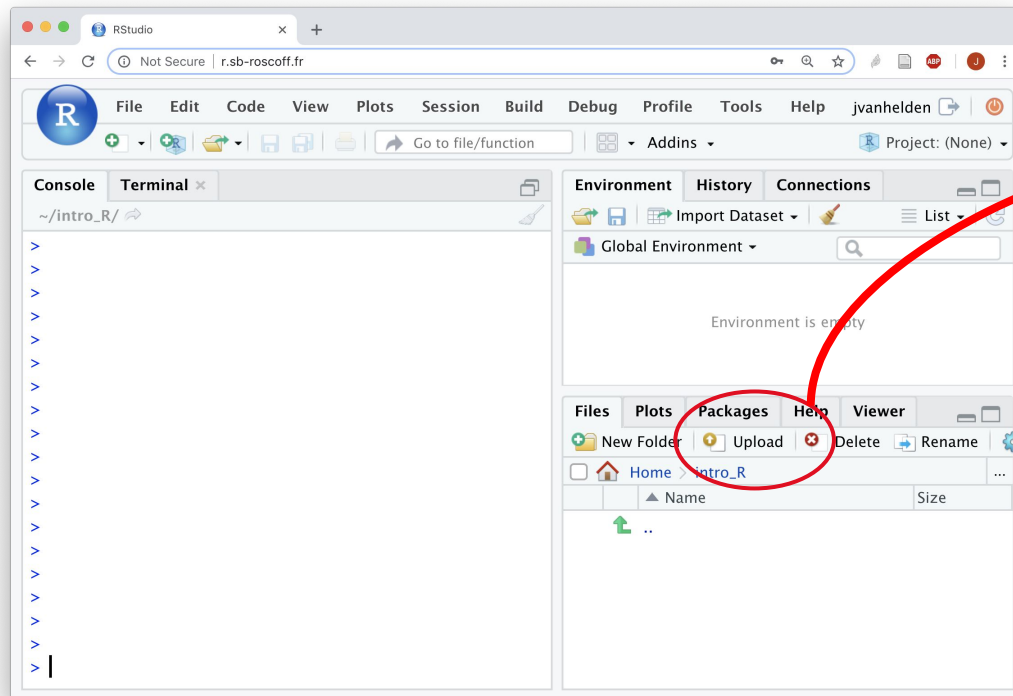
Fichiers disponibles sur le Moodle :

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

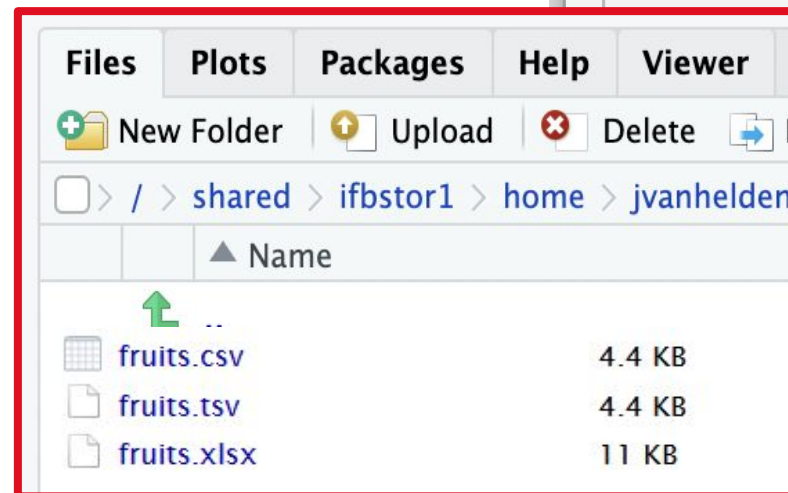
Attention: veuillez à sauvegarder les fichiers

- 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

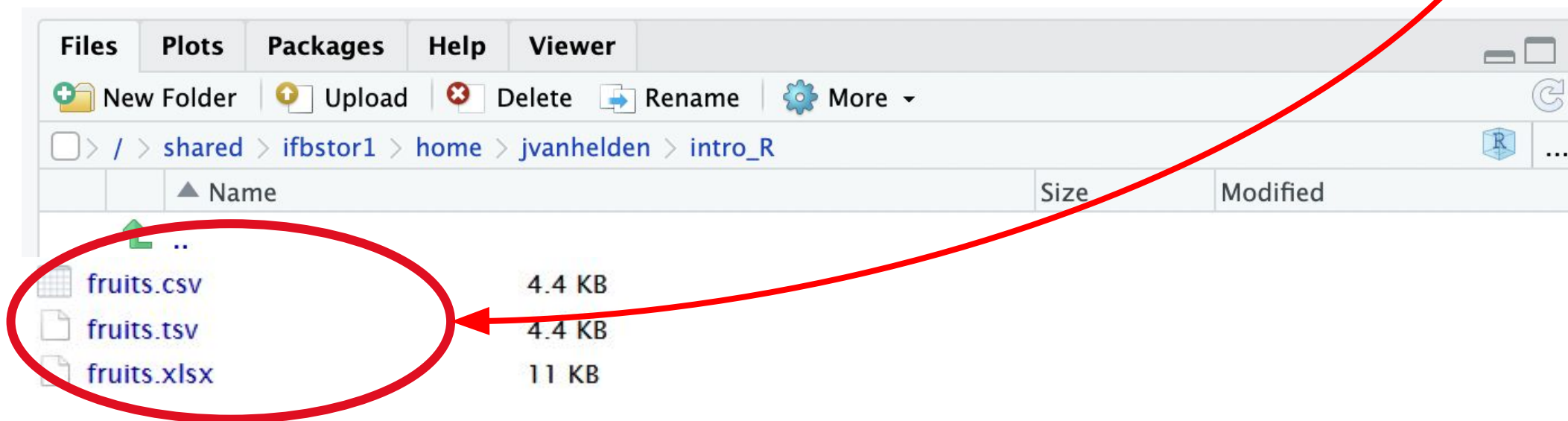
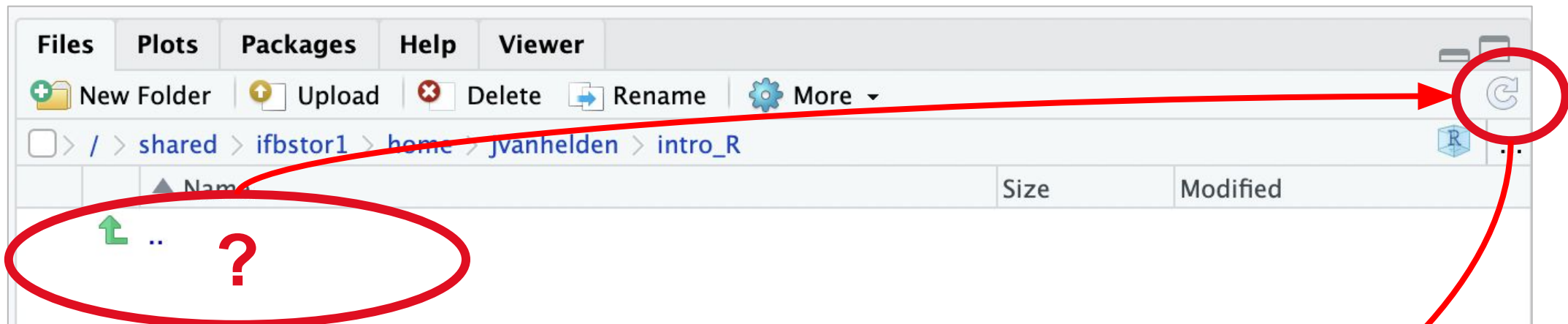


Au moyen du bouton **“Upload”**, téléversez les fichiers depuis votre ordinateur vers votre compte sur le serveur.



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 = ".")
```

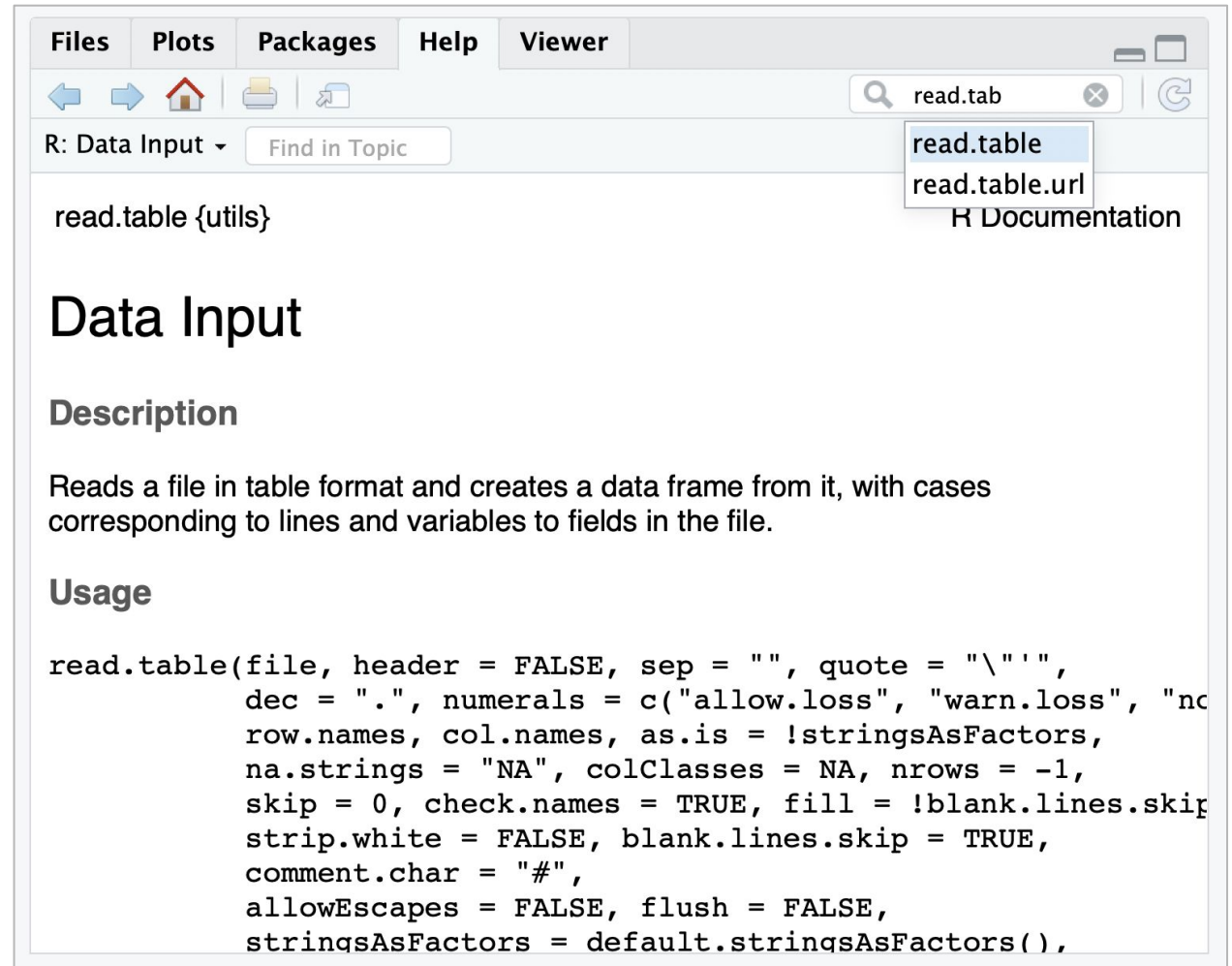
Accéder à l'aide d'une fonction :

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

Notation alternative :

```
?read.table
```

Ou encore, via le Help de RStudio :



The screenshot shows the RStudio Help viewer interface. The top menu bar includes 'Files', 'Plots', 'Packages', 'Help', and 'Viewer'. The 'Help' menu is open, showing a search bar with 'read.tab' and a dropdown list with 'read.table' and 'read.table.url'. The main content area displays the documentation for 'read.table {utils}'. The title 'Data Input' is prominently displayed. Below it, the 'Description' section states: '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.' The 'Usage' section shows the function signature: `read.table(file, header = FALSE, sep = "", quote = "\"\"", dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"), 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)`

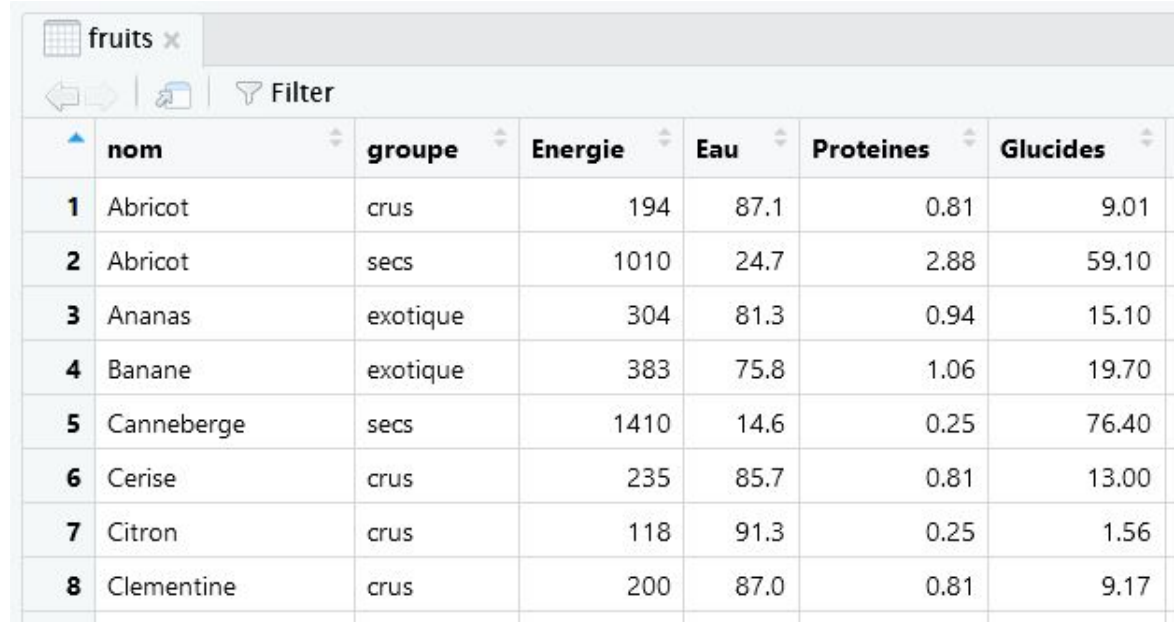
Un peu plus de lignes

`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**

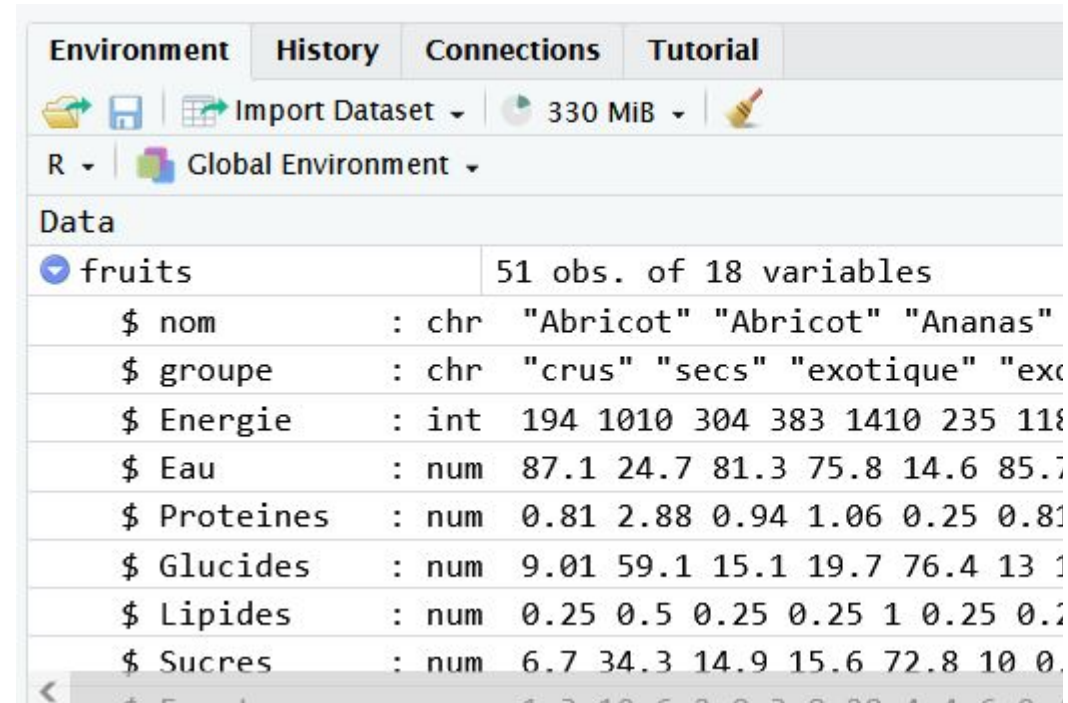
Noms des colonnes et des lignes :

`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' dataset loaded. The dataset has 51 observations and 18 variables. The variables are listed with their data types and sample values.

		51 obs. of 18 variables
\$ 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)]
```

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)
```

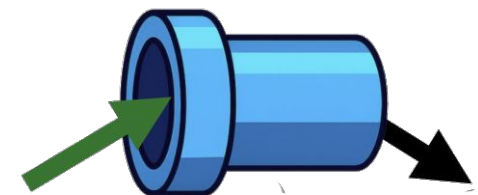
Pour aller plus loin ...

→ Packages pour faciliter le nettoyage des données :

- **mutate()** manipuler des variables d'un tableau
- **select()** choisir des variables selon leur nom
- **filter()** choisir des lignes selon leur contenu
- **summarise()** résumer l'information
- **arrange()** changer l'ordre des lignes
- **merge()** fusionner des tableaux
- ...



```
moy_par_diet <- ChickWeight %>%  
  filter(Time == 21) %>%  
  group_by(Diet) %>%  
  summarise(mean_weight = mean(weight))
```



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 Bulk)
- Pour aller plus loin : R Markdown
 - <https://bookdown.org/yihui/rmarkdown/>
 - <https://rmarkdown.rstudio.com/>



Take home messages

Symboles	Pourquoi ?	Exemples
= ou <- ou ->	affecter une valeur à une variable ou un argument	a <- 10 read.table(file = "file.txt")
()	délimiter les arguments d'une fonction	mean(x = c(10,6,8,2))
[]	accéder à un élément de vecteur	mesures[1] mesures[2:8] mesures[c(3,6,4)]
[[]] ou \$	accéder à un élément de liste	courses\$banane courses[["banane"]] courses[[1]]
[,] ou \$	accéder à des éléments de data frame	fruits\$nom fruits[1:5, "nom"]

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**

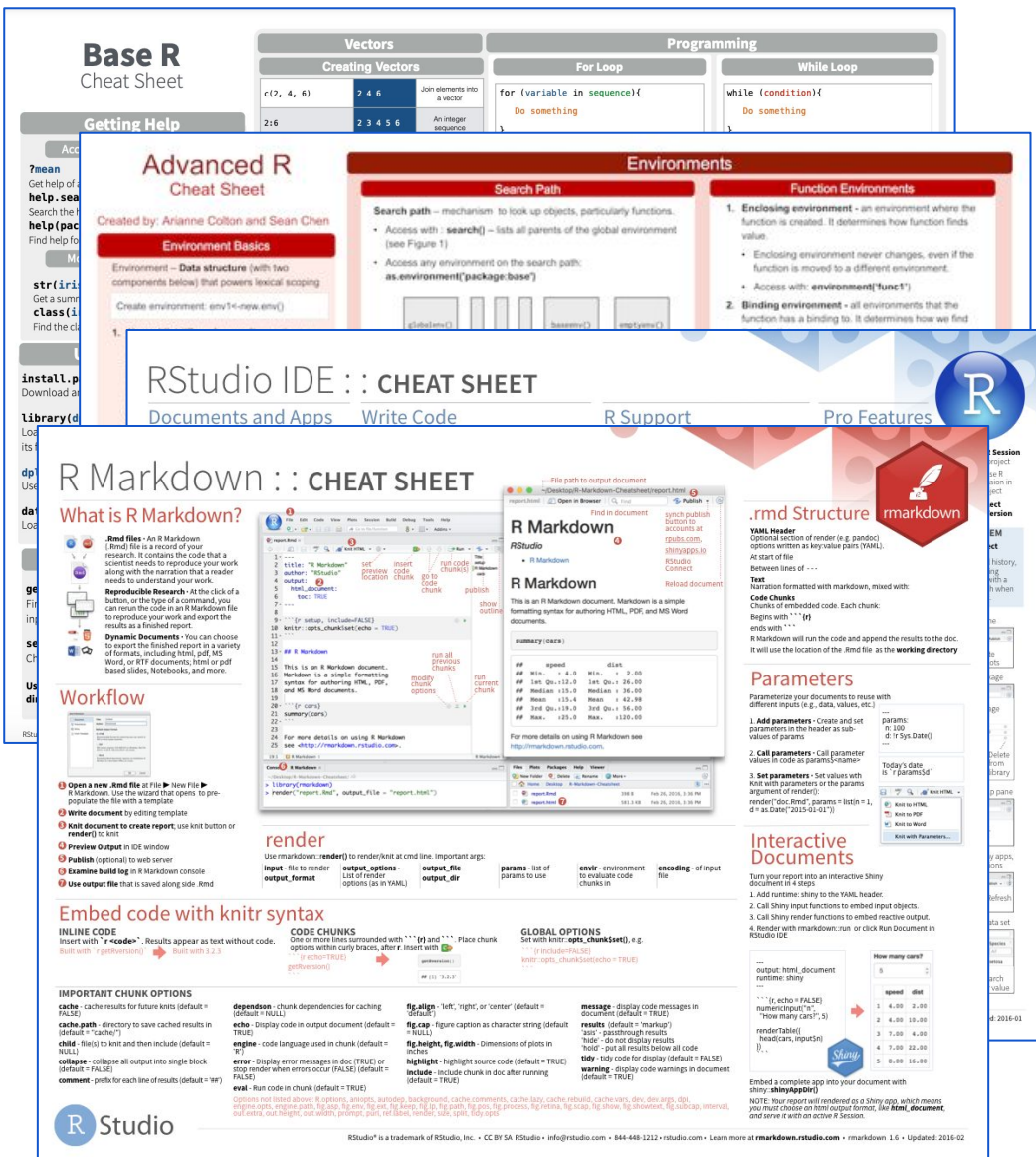


Une question ? Un besoin ? Un problème ? **Contactez la communauté IFB**

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



Ressources



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/>

Packages du tidyverse :

<https://tidyverse.org/>

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/>

