

# R en quelques mots

---

Langage de programmation qui permet de :

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

Disponible sur Windows, Linux, Mac.

Historique :

- 1993 : début du projet R
- 2000 : sortie de R 1.0.0
- 2020 : R 4.0.2

# Avantages et inconvénients

---

## Avantages :

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

## Inconvénients :

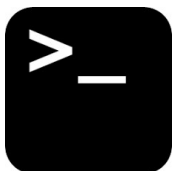
# R sait tout faire

---

Lire un tableau de données	<code>read.table()</code>
Fusionner deux tableaux	<code>merge()</code>
Filtrer des lignes	<code>data[data\$x &gt; 10]</code>
Sélectionner des colonnes	<code>data[,c("x","y")]</code>
Rechercher une chaîne de caractères	<code>grep()</code>
Calculer une moyenne	<code>mean()</code>
Réaliser une ACP	<code>prcomp()</code>
Additionner deux matrices	<code>mat1 + mat2</code>
Exporter un tableau de données	<code>write.table()</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>

# Différentes manières d'utiliser R

---



Localement via le terminal



**Localement via RStudio (utilisation classique) : logiciel qui ajoute une couche graphique à R**



Sur un serveur via le terminal et une connexion ssh



**Sur un serveur via un navigateur web pour accéder à RStudio server**

# Fichiers à récupérer sur votre machine

---

A partir d'un navigateur Web, téléchargez

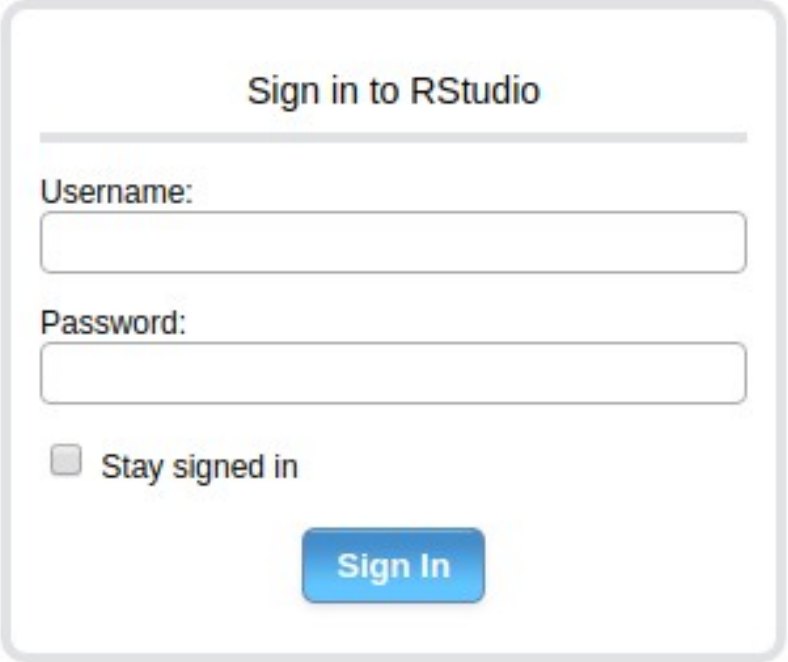
Le fichier `smp.csv` pour des tests.

# Ouverture ou connexion à RStudio

---

2 alternatives :

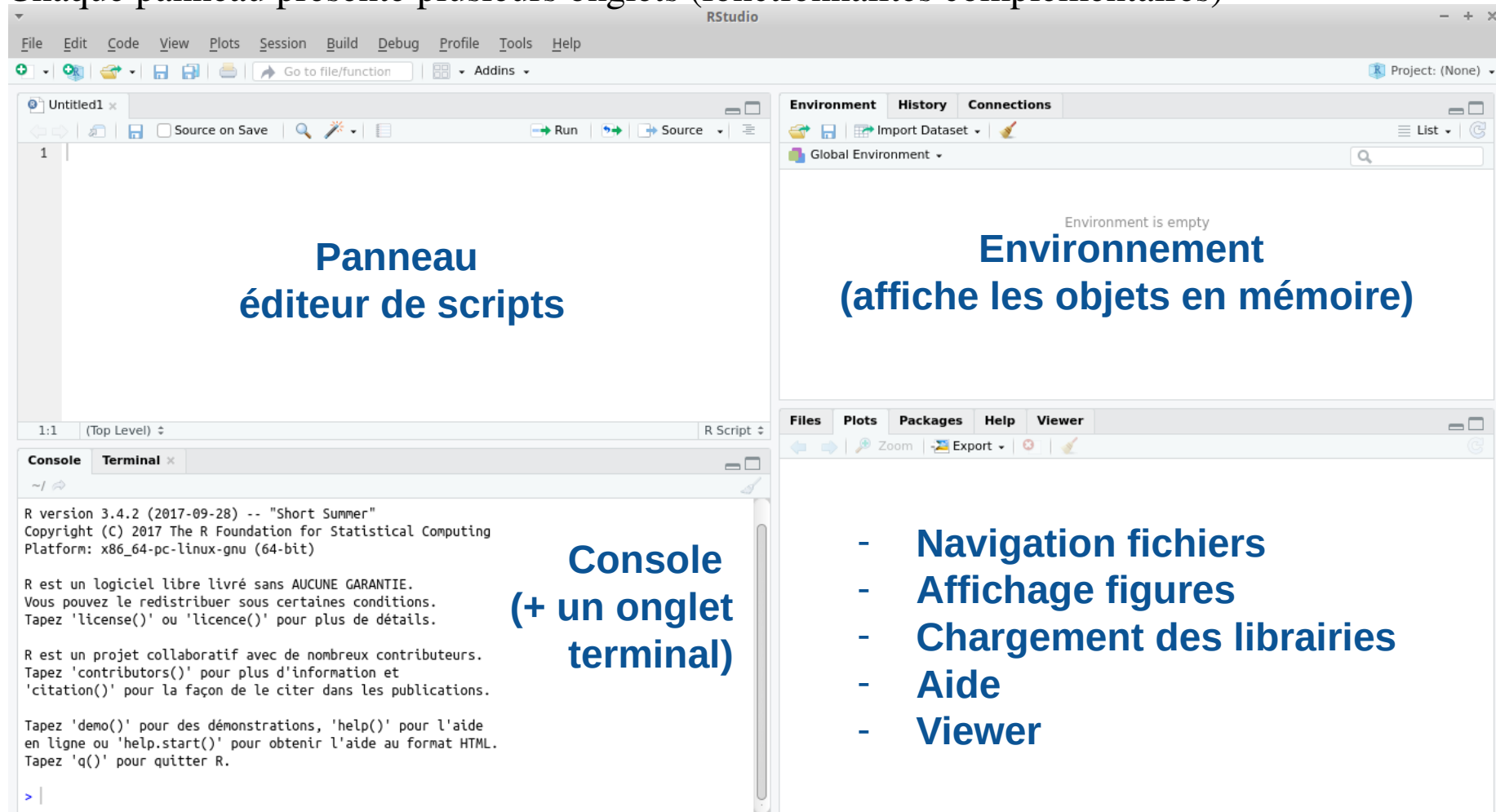
1. Ouvrir RStudio sur votre propre ordinateur (si installé)
- 2. Vous connecter au serveur Web R du cluster**



A screenshot of a web-based login form titled "Sign in to RStudio". The form is enclosed in a light gray rounded rectangle. It contains two input fields: "Username:" and "Password:", each with a corresponding text box. Below the password field is a checkbox labeled "Stay signed in". At the bottom center of the form is a blue button with the text "Sign In" in white.

# RStudio

- Disponible depuis 2011
- Ajoute un ecouche graphique au logiciel R
- Logiciel facilitant l'utilisation de R via 4 panneaux
- Chaque panneau présente plusieurs onglets (fonctionnalités complémentaires)



- Navigation fichiers
- Affichage figures
- Chargement des librairies
- Aide
- Viewer

# R sait tout faire : il compte

---

**2 + 3**

**4 \* 5**

**6 / 4**

**1:10**

**8:-9**

**1,2**

**1.2**



# Notion de variable/objet

```
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 évoque le nom de variable, R l'imprime
```

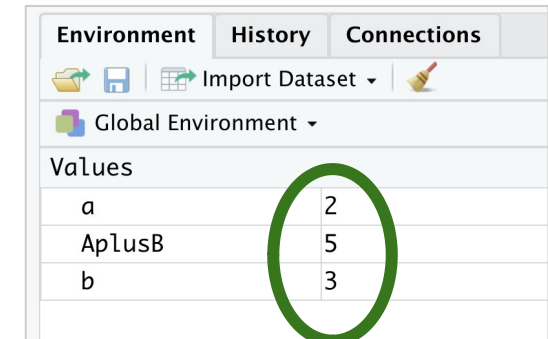
```
b <- 3      ## Assigner une valeur à une seconde variable
AplusB <- a + b  ## Effectuer un calcul avec 2 variables
print(AplusB)  ## Afficher le contenu de la variable AplusB
```

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

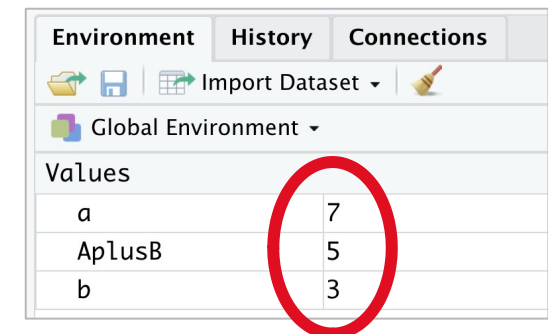
```
AplusB <- a + b  ## On recalcule AplusB
print(AplusB)    ## La nouvelle valeur tient compte de la modification de a
```

```
vec1 <- c(1,10)  ## Créer un vecteur
vec2 <- 1:10      ## Créer un vecteur contenant une séquence d'entiers de 1 à 10
vec2 + a          ## Somme d'un vecteur et d'un nombre
vec3 <- c("riri", "fifi", "loulou")  ## Vecteur de chaînes de caractères
vec3 / b          ## Diviser des chaînes de caractères par un nombre
```

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



Environment	History	Connections
Global Environment		
Values		
a	2	
AplusB	5	
b	3	



Environment	History	Connections
Global Environment		
Values		
a	7	
AplusB	5	
b	3	

# Notion de variable/objet

`a <- "bonjour"`      `##` variable nommée a et lui assigner une chaîne de caractère

`age <- c(18, 27, 34, 18, 24, NA, 30, 28, 19, 19)` `#`crée une variable appelée age (il n'est pas accents ou signes diacritiques) à laquelle on associe la liste des nombres

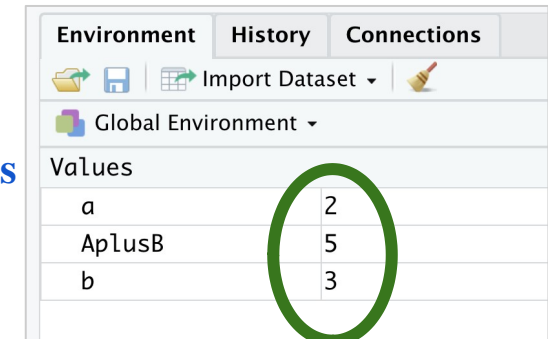
`donnee <- data.frame(age, sex=sexe)` `#`tableau de données hétérogènes  
`#`La commande `data.frame()` s'utilise de manière assez simple :  
on lui fournit la liste `#`des variables à inclure dans le tableau de données. Éventuellement, les variables peuvent être renommées directement lors de la construction du data frame, comme illustré dans l'exemple suivant.

Un dataframe peut être aussi créé par l'importation de données externes (excel, csv,...)

Pour connaître le type d'un objet, utiliser la fonction `mode`

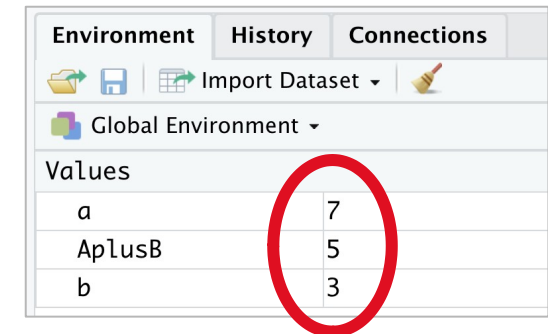
`mode(age)`

`mode(donnee)`



The screenshot shows the R Studio Environment pane. The 'Global Environment' is selected. The 'Values' section displays a data frame with three rows: 'a' with value 2, 'AplusB' with value 5, and 'b' with value 3. The values 2, 5, and 3 are circled in green.

Variable	Value
a	2
AplusB	5
b	3

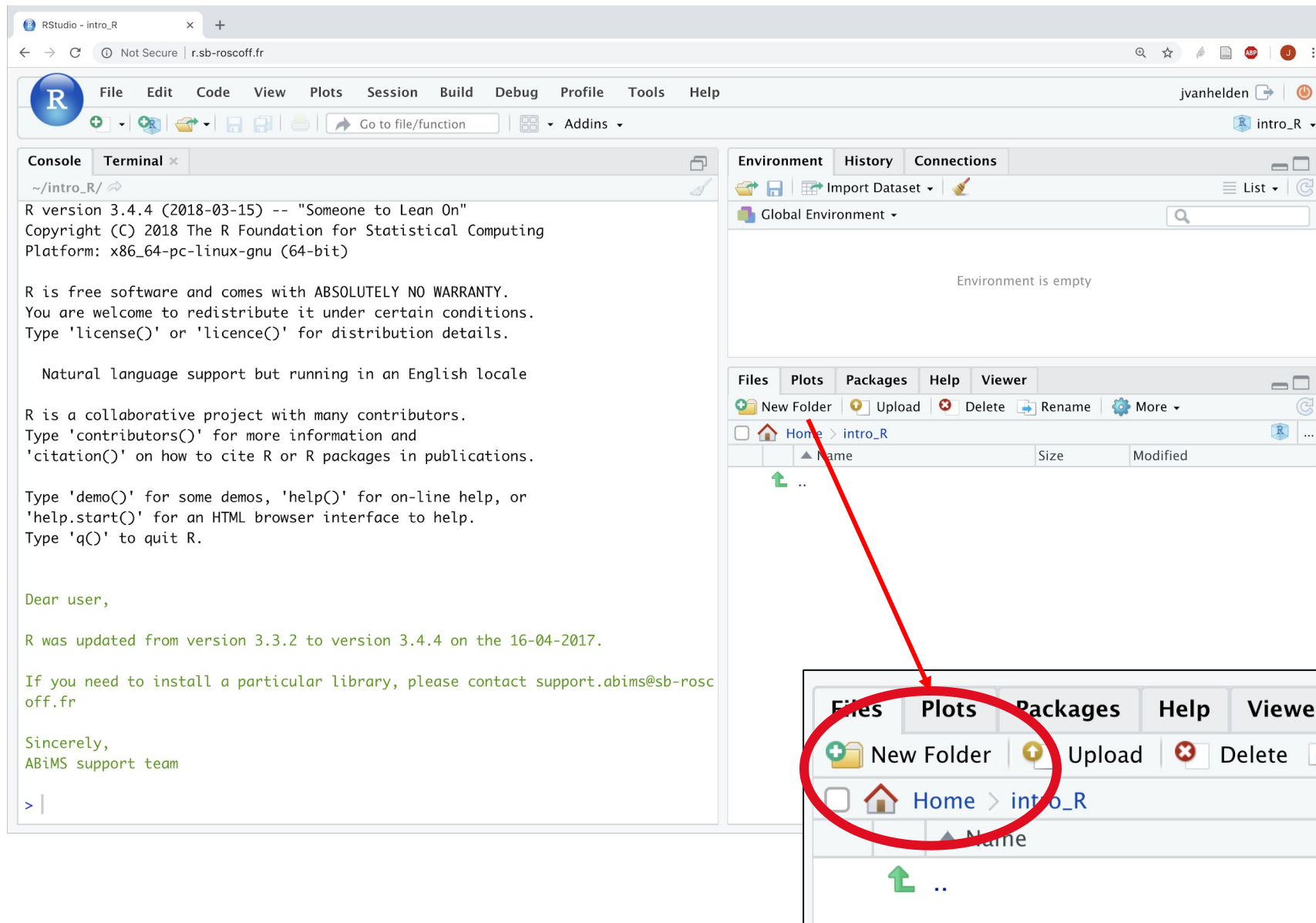


The screenshot shows the R Studio Environment pane. The 'Global Environment' is selected. The 'Values' section displays a data frame with three rows: 'a' with value 7, 'AplusB' with value 5, and 'b' with value 3. The values 7, 5, and 3 are circled in red.

Variable	Value
a	7
AplusB	5
b	3

# **Cas pratique : données issues d'une enquête sur la santé mentale en prison : smp.csv**

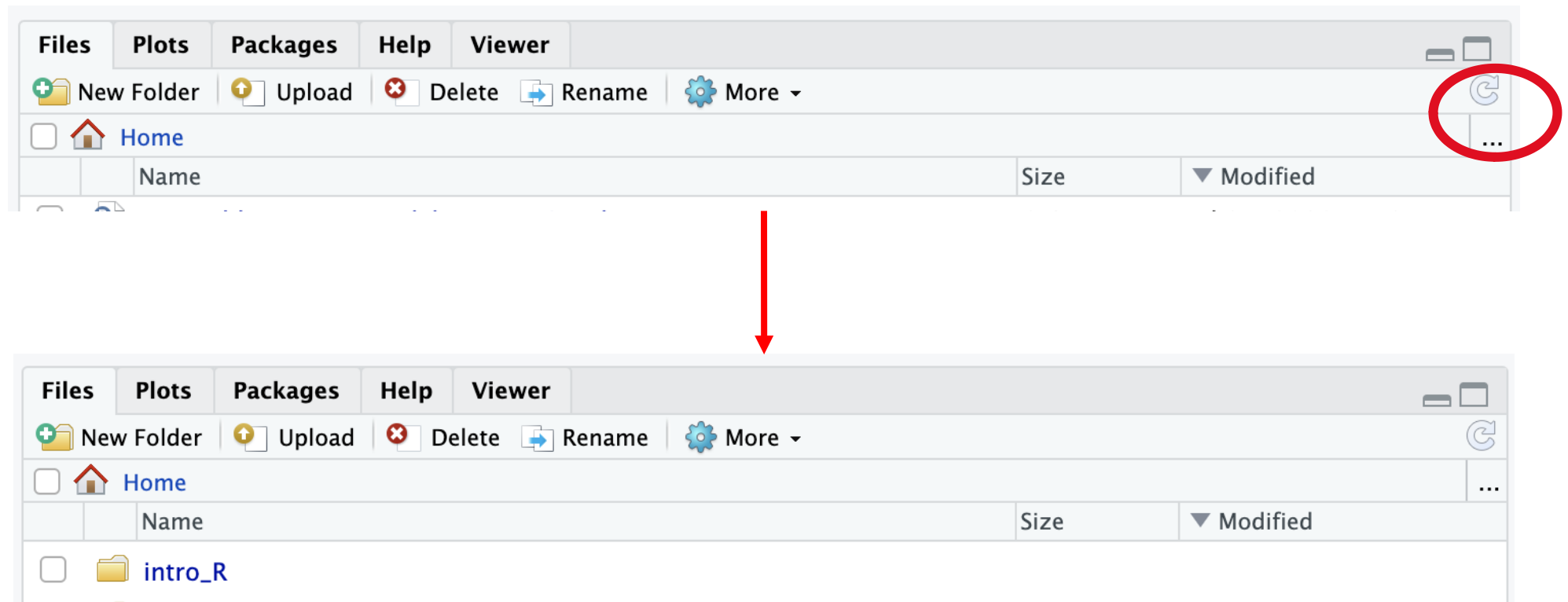
# Création d'un dossier "intro\_R"



# Actualisation du dossier

---

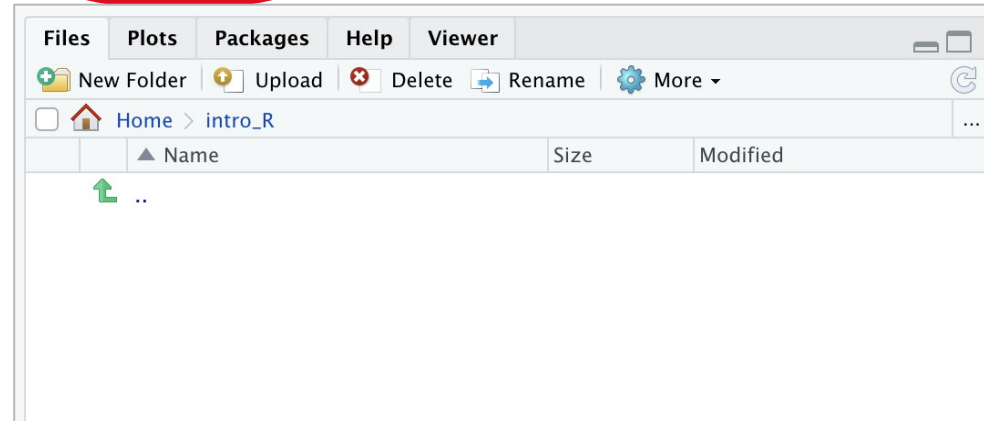
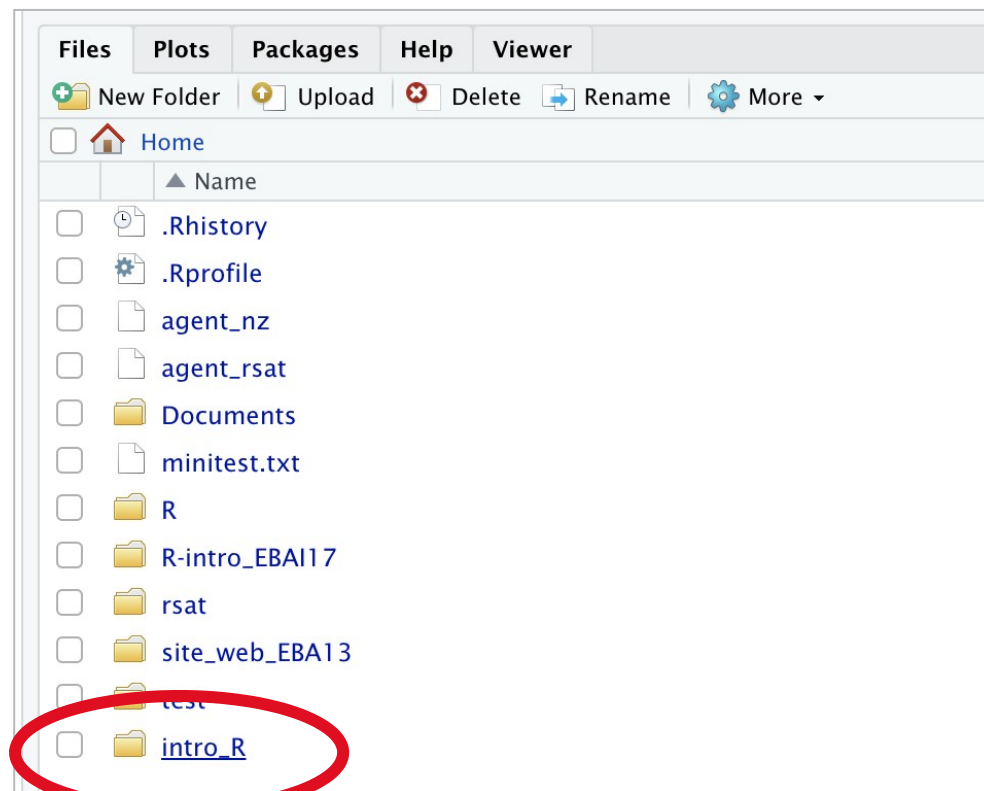
Dans certains cas, il faut actualiser le contenu du dossier pour pouvoir voir le nouveau sous-dossier. Vérifiez ensuite si `intro_R` apparaît bien dans le contenu de votre dossier principal.



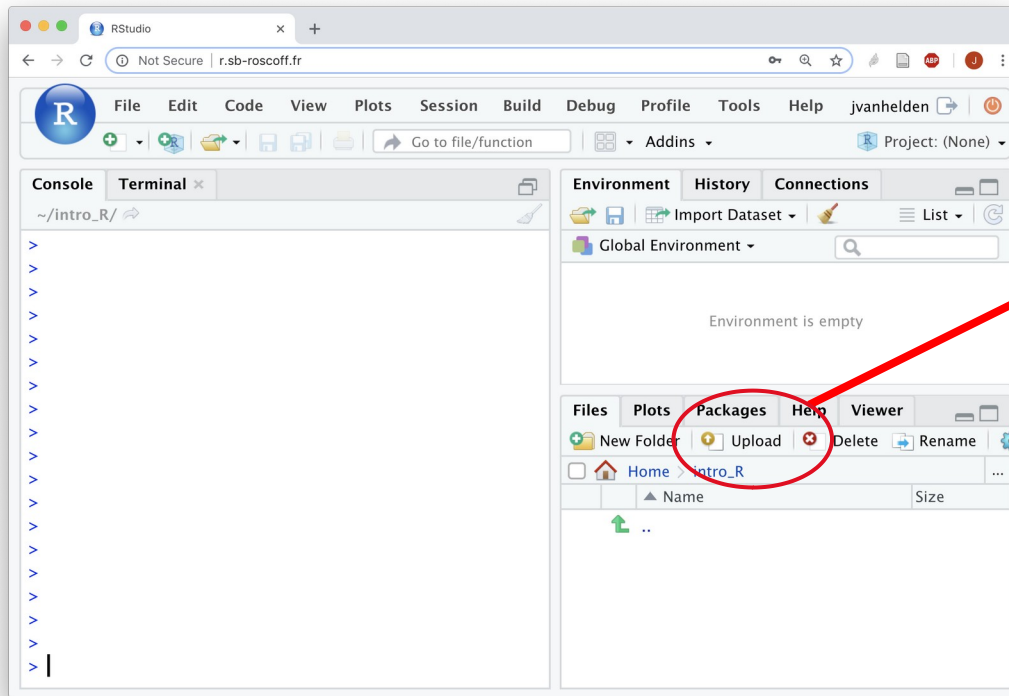
# Déplacement dans le dossier “intro\_R”

Double-cliquez sur le dossier “intro\_R”, pour vous y déplacer.

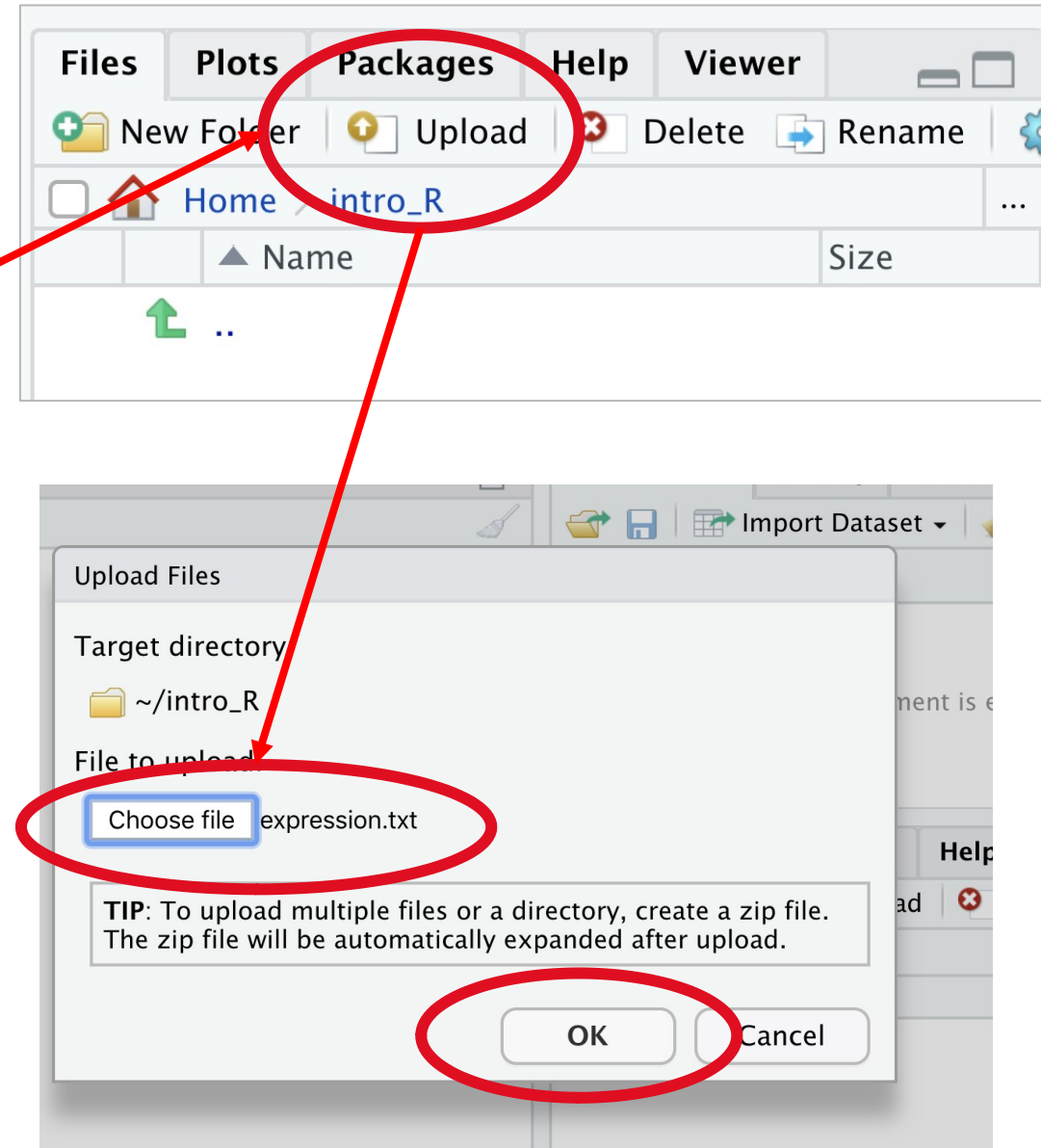
Puisque vous venez de créer le dossier il est vide (image du bas).



# Téléversement (“upload”) des données

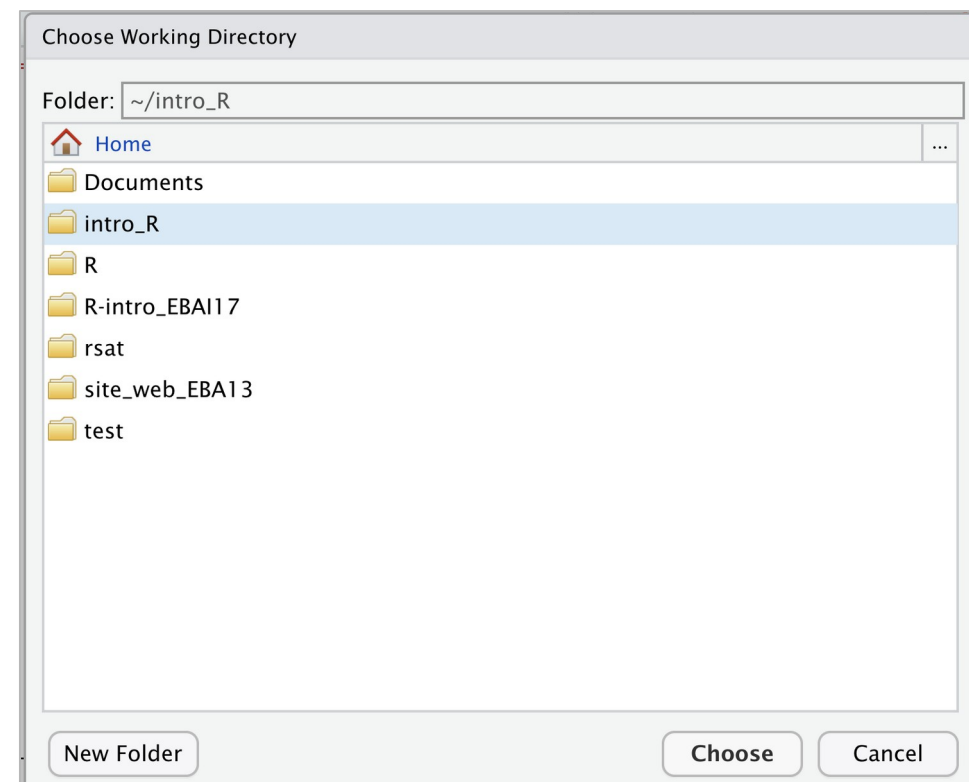
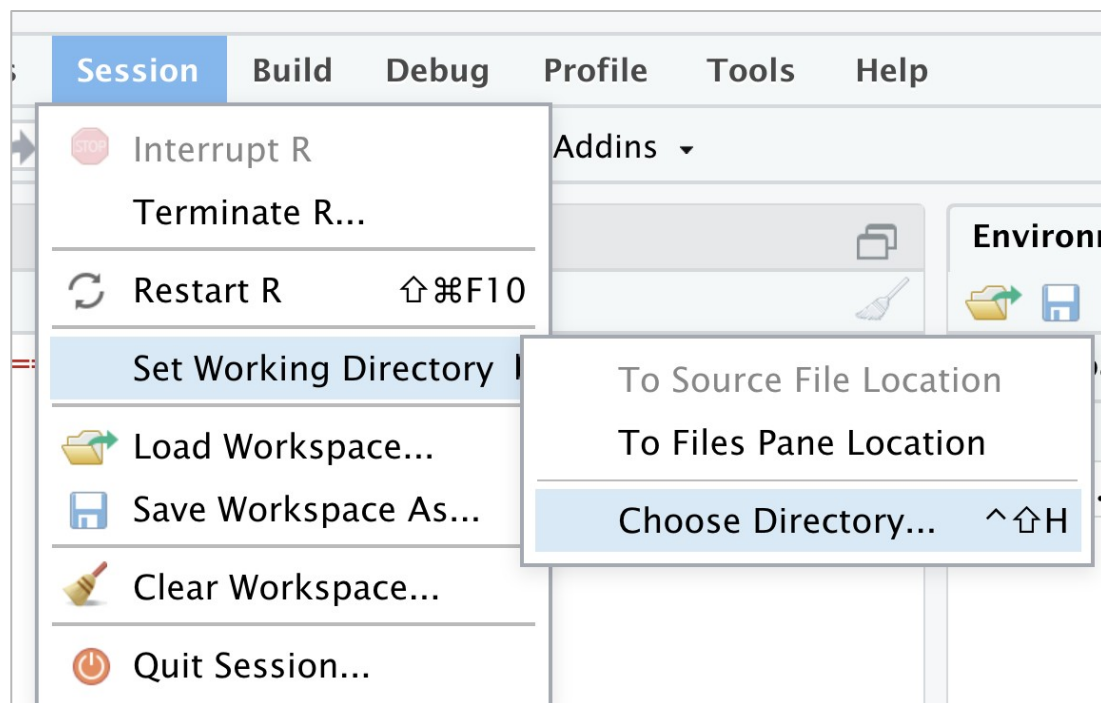


Au moyen du bouton “**Upload**”, téléversez les fichiers d’expression et d’annotation depuis votre ordinateur vers votre compte sur le serveur.



# Définissez votre dossier espace de travail (workdirectory)

---





# En ligne de commande votre home du cluster)

---

**Alternative:** dans le terminal du cluster, téléchargez et enregistrez dans votre home les fichiers de données:

- `smp.csv`
- `annotation.csv`: informations sur les gènes (id, name, chr, start, stop)

Ouvrez une connexion ssh

```
ssh [votre_login]@adresse_cluster
```

Où suis-je ?

```
pwd
```

Créez un répertoire “intro\_R”

```
mkdir -p ~/intro_R
```

Déplacez vous dans votre dossier

```
cd ~/intro_R
```

Où suis-je maintenant ?

```
pwd
```

Téléchargez les données

Qu’y a-t-il ici ?

```
ls -l
```

# Par des lignes de commandes directement depuis Rstudio

---

Définir une variable qui indique le chemin du dossier de travail (working directory).

```
work.dir <- "~/intro_R"
```

Note: R interprète le caractère “~” comme le “HOME” de Linux (cela marche aussi pour Windows!)

S’il n’existe pas encore, créer le dossier de travail.

(Commande Unix équivalente: "mkdir -p ~/intro\_R")

```
dir.create(work.dir, recursive = TRUE, showWarnings = FALSE)
```

Où suis-je ? (Commande Unix équivalente: "pwd")

```
getwd()
```

Aller dans ce dossier de travail (Commande Unix équivalente: "cd ~/intro\_R")

```
setwd(work.dir)
```

Et maintenant, où suis-je ?

```
getwd()
```

Qu'y a-t-il par ici ? (Commande Unix équivalente: "ls")

```
list.files()
```

```
dir()      ## Un autre nom pour la même commande
```

# Télécharger un fichier : the “geek” way (V3)

---

Nous avons montré ci-dessus comment télécharger des fichiers en utilisant l’interface graphique de RStudio.

Alternativement, on peut télécharger des fichiers au moyen de la commande R **download.file**.

Les deux commandes suivantes permettent de télécharger le fichier utilisé pour les exercices.

**download.file(url = "http://adresse/smp.csv", destfile = "smp.csv")**

Note : équivalent de la commande “wget” sous Unix.

# Chargement des données (dans la mémoire de R)

---

Charger le contenu du fichier "smp.csv" dans une variable nommée "data".

```
data<- read.csv2("smp.csv")
```

Accéder à l'aide d'une fonction

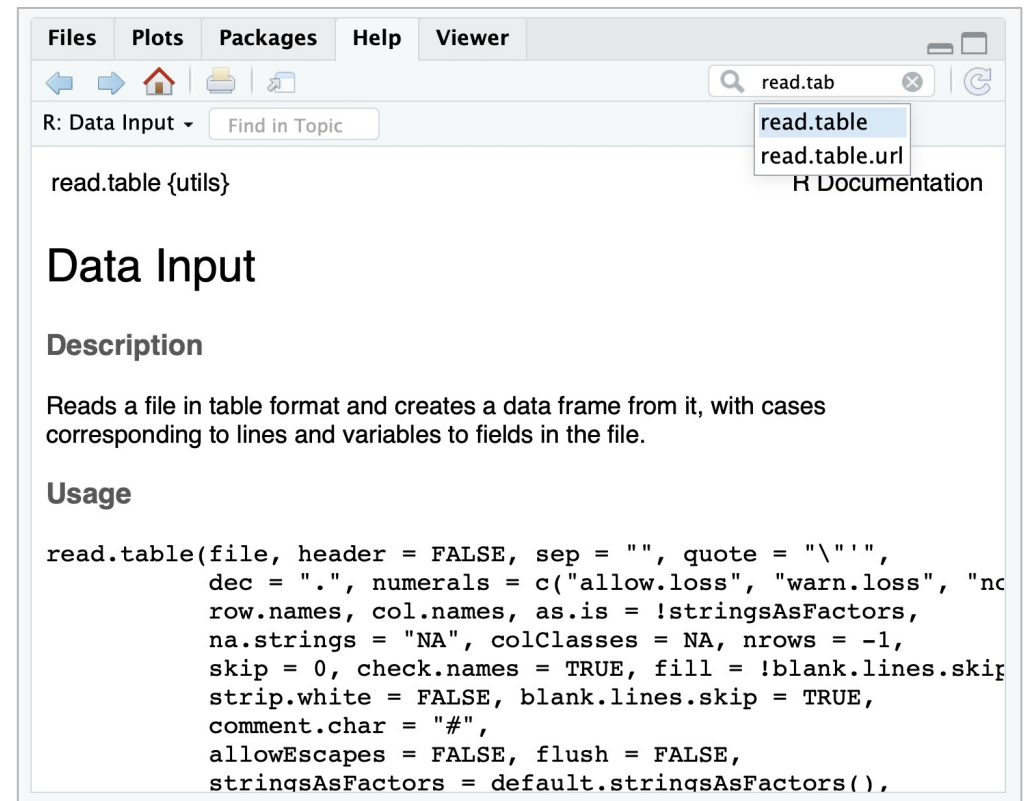
```
help(read.csv2)
```

Notation alternative

```
?read.csv2
```

## Recherche interactive sous RStudio

- Sélectionner l'onglet "Help" du panneau inférieur droit.
- Taper "read.table" dans la boîte de recherche.



# Affichage de l'objet “data”

---

Imprimer toutes les valeurs.

**print(data)**

Affichage des premières lignes de l'objet

**(data)**

La commande `str()` fournit un résumé de l'ensemble des variables avec leur type (ou mode de représentation)

– `int` pour les variables numériques, `factor` pour les variables catégorielles – et un aperçu des 10 premières Observations.

`str(data)`

# Caractéristiques d'un tableau de données

---

## Dimensions

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

## Noms des lignes et colonnes

`colnames(data)`  
`rownames(data)`

## Résumé rapide des données par colonne

`summary(data)`

`str(data)`

# Sélection de colonnes d'un tableau

---

Pour afficher l'ensemble des observations recueillies pour une variable donnée, on tapera le nom de cette variable préfixé du nom du data frame suivi du signe \$ . L'expression `data$age` désignera ainsi les valeurs prises par la variable age dans le data frame data . Plutôt que d'afficher l'ensemble des valeurs, on peut vouloir limiter l'affichage à certaines observations, que l'on désignera par leur numéro. On peut ainsi choisir d'afficher, la première, les deux premières, ou la première et la troisième observation à l'aide des instructions suivantes :

```
data$age[1]
```

```
## [1] 31
```

```
smp$age[c(1,2)]
```

```
smp$age[c(1,3)]
```

# Répresentation graphiques

---

Quelques représentations graphiques



# Histogrammes

---

```
hist(smp$age)
```

Pour colorer et changer le titre de l'axe des x :

```
hist(smp$age, col="grey", main="", xlab = "age")
```

# Boxplots

---

```
boxplot(smp$age, xlab="age")
```

Représenter l'âge en fonction d'un groupe :

```
boxplot(smp$age~smp$rs, ylab="age", xlab="recherche de sensation")
```

L'âge est supérieure quand on a faible niveau de sensation

# Nuage de points

---

```
plot(smp$age, smp$n.enfant)
```

Plus un détenu est âgé, plus il a un nombre moyen d'enfants plus grand

Bouger les points légèrement :

```
plot(jitter(smp$age), jitter(smp$n.enfant))
```

Evolution temporelle d'une variable quantitative, diagramme temporel ou diagramme de temperature :

# Resumé

---

- 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)
- Bonnes pratiques : <https://google.github.io/styleguide/Rguide.xml>

# Ressources

## Base R Cheat Sheet

### Getting Help

Acc

7mean

Get help of

help\_sea

Search the

help(pac

Find help fo

M

str(irl

Get a sum

class(i

Find the d

M

install.p

Download a

Library(d

Los

its

dp

Use

da

Los

Ge

fin

in

Ch

Us

di

R

Vectors		Programming	
Creating Vectors		For Loop	While Loop
c(2, 4, 6)	2 4 6	for (variable in sequence){	while (condition){
2:6	2 3 4 5 6	Do something	Do something
		}	}

## Advanced R Cheat Sheet

Created by: Arienne Colton and Sean Chen

### Environment Basics

Environment - Data structure (with two components below) that powers lexical scoping

Create environment: env1 <- new.env()

1.

Environments	
Search Path	Function Environments
<b>Search path</b> - mechanism to look up objects, particularly functions.	
• Access with <code>search()</code> - lists all parents of the global environment (see Figure 1)	
• Access any environment on the search path: <code>as.environment("package:base")</code>	
<b>Function Environments</b>	
1. <b>Enclosing environment</b> - an environment where the function is created. It determines how function finds value.	
• Enclosing environment never changes, even if the function is moved to a different environment.	
• Access with: <code>environment("func")</code>	
2. <b>Binding environment</b> - all environments that the function has a binding to. It determines how we find	

## RStudio IDE : : CHEAT SHEET

Documents and Apps

Write Code

R Support

Pro Features

## R Markdown : : CHEAT SHEET

### What is R Markdown?

**R Markdown** - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.

**Reproducible Research** - At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

**Dynamic Documents** - You can choose to export the finished report in a variety of formats, including HTML, pdf, MS Word, or RTF documents, HTML or pdf based slides, Notebooks, and more.

### Workflow

1. Open a new .Rmd file at File > New File > R Markdown. Use the wizard that opens to pre-populate the file with a template.
2. Write document by editing template.
3. Knit document to create report, use knit button or render() to knit.
4. Preview output in IDE window.
5. Publish (optional) to web server.
6. Examine build log in R Markdown console.
7. Use output file that is saved along side .Rmd.

### render

Use `markdown::render()` to render/knit at cmd line. Important args:

input	file to render
output_format	output format
output_dir	output directory
output_file	output file
output_options	List of render options (as in YAML)

### Embed code with knitr syntax

**INLINE CODE** - Insert with `<code>`. Results appear as text without code.

Built with `r gelteversion()` → Built with 3.2.3

**CODE CHUNKS** - One or more lines surrounded with ````` (r) and `````. Place chunk options within curly braces, after r. Insert with `<code>`.

Built with `r gelteversion()` → Built with 3.2.3

**GLOBAL OPTIONS** - Set with `knitr::opts_chunk$set()`, e.g.

```
knitr::opts_chunk$set(echo = TRUE)
```

### IMPORTANT CHUNK OPTIONS

<b>cache</b> - cache results for future knits (default = FALSE)	<b>dependencies</b> - chunk dependencies for caching (default = NULL)	<b>fig.align</b> - 'left', 'right', or 'center' (default = 'center')	<b>message</b> - display code messages in document (default = TRUE)
<b>cache.path</b> - directory to save cached results in (default = "cache")	<b>echo</b> - display code in output document (default = TRUE)	<b>fig.cap</b> - figure caption as character string (default = NULL)	<b>results</b> - 'asis' - passthrough results 'hide' - do not display results 'hold' - put all results below all code
<b>child</b> - files to knit and then include (default = NULL)	<b>engine</b> - code language used in chunk (default = 'R')	<b>fig.height</b> - figure height - Dimensions of plots in inches	<b>tidy</b> - tidy code for display (default = FALSE)
<b>collapse</b> - collapse all output into single block (default = FALSE)	<b>error</b> - Display error messages in doc (TRUE) or stop render when error occurs (FALSE) (default = FALSE)	<b>highlight</b> - highlight source code (default = TRUE)	<b>warning</b> - display code warnings in document (default = TRUE)
<b>comment</b> - prefix for each line of results (default = ">")	<b>eval</b> - Run code in chunk (default = TRUE)		

RStudio

RStudio is a trademark of RStudio, Inc. • CC BY SA RStudio • info@rstudio.com • 844-449-1212 • rstudio.com • Learn more at [rmarkdown.rstudio.com](http://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/>

# Merci pour votre attention

---

Sources :

- Ecole de Bioinformatique Aviesan-IFB : <https://tinyurl.com/r-intro-ebai20>
- Introduction à R, Christophe Lalanne & Bruno Falissard :  
<https://r.developpez.com/tutoriels/introduction-r/>

Ousmane Barra, [ousmane.barra@ujkz.bf](mailto:ousmane.barra@ujkz.bf), Whatsapp : 75 33 73 12

