

# Débuter avec R et RStudio





langage de programmation



environnement de développement  
intégré

# **Un panorama de RStudio**

~/Dropbox/Teaching/Workshops/2020-gpl/01\_welcome-tidyverse - RStudio

Console Terminal x Jobs x

~/Dropbox/Teaching/Workshops/2020-gpl/01\_welcome-tidyverse/ ↵

```
R version 4.0.0 (2020-04-24) -- "Arbor Day"
Copyright (C) 2020 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.
```

> |

Environment History Connections Tutorial

Import Dataset ↵ List ↵ Global Environment ↵

Environment is empty

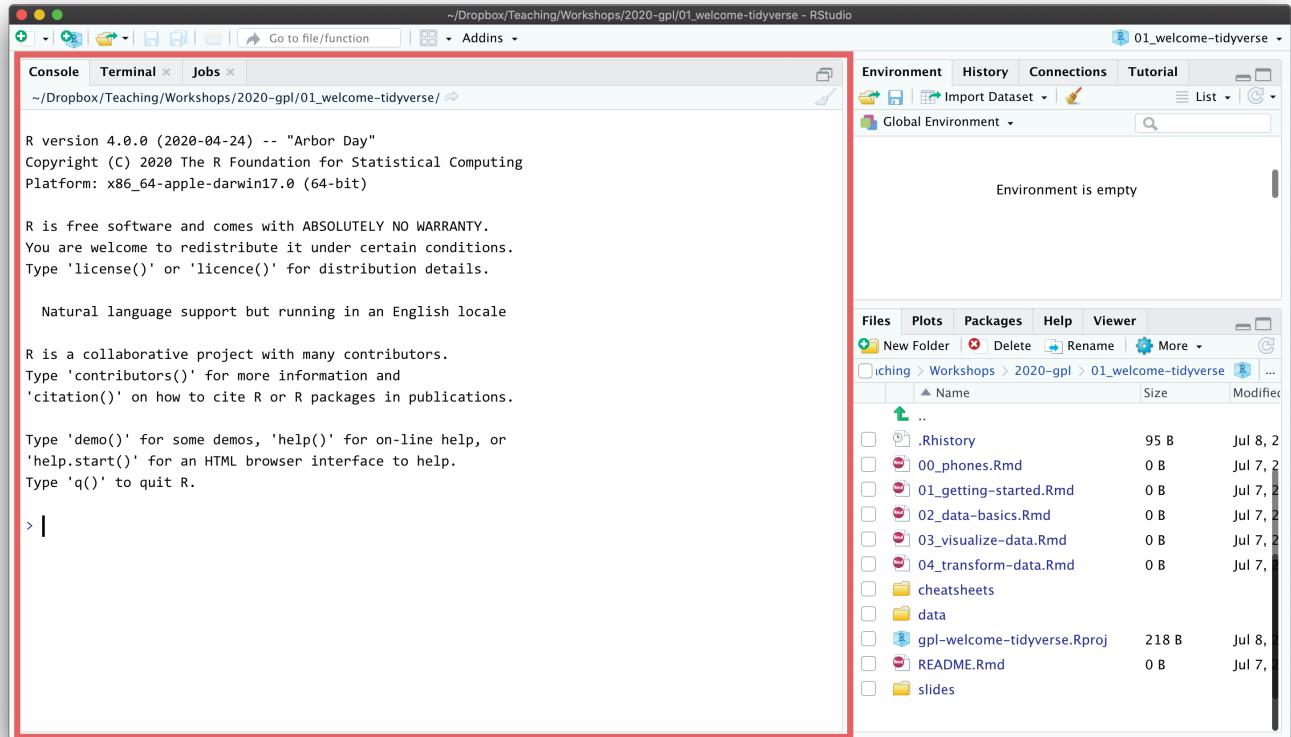
Files Plots Packages Help Viewer

New Folder Delete Rename More ...

... /iching > Workshops > 2020-gpl > 01\_welcome-tidyverse

	Name	Size	Modified
	..		
	.Rhistory	95 B	Jul 8, 2020
	00_phones.Rmd	0 B	Jul 7, 2020
	01_getting-started.Rmd	0 B	Jul 7, 2020
	02_data-basics.Rmd	0 B	Jul 7, 2020
	03_visualize-data.Rmd	0 B	Jul 7, 2020
	04_transform-data.Rmd	0 B	Jul 7, 2020
	cheatsheets		
	data		
	gpl-welcome-tidyverse.Rproj	218 B	Jul 8, 2020
	README.Rmd	0 B	Jul 7, 2020
	slides		

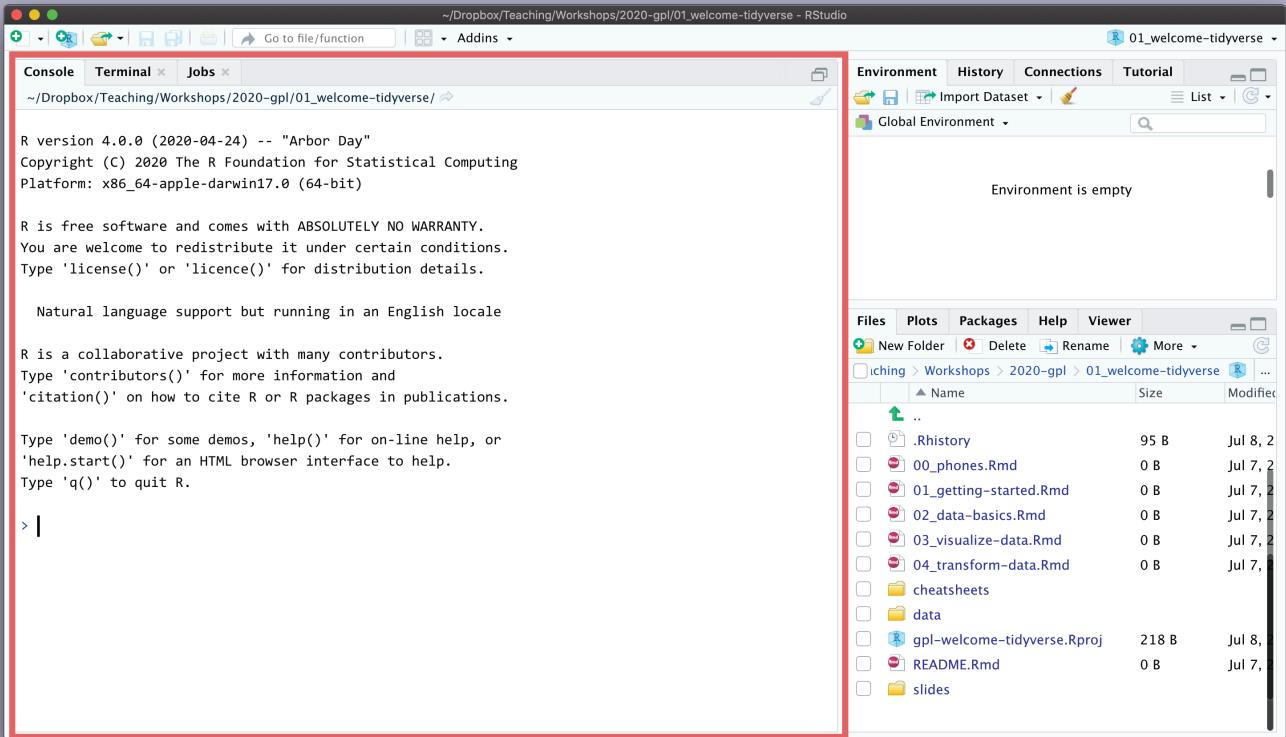
# Console



R est en attente de vos instructions

Tapez votre code, et appuyez sur Retour pour le compiler

# À votre tour



Tapez  $2 + 2$  dans la console

Appuyez sur Retour

2 + 2

```
## [1] 4
```

**Ce calcul est éphémère.  
Pour recompiler, il faudra retaper les instructions.**

**On va plutôt enregistrer le code dans un document R**

**On peut aussi assigner le résultat à une variable  
pour l'utiliser subséquemment**

# Arborescence et fichiers

**R pointe vers un répertoire local**

**Utiliser `getwd()` pour trouver l'emplacement**

**Travailler avec chemin relatif**

# Projets dans RStudio

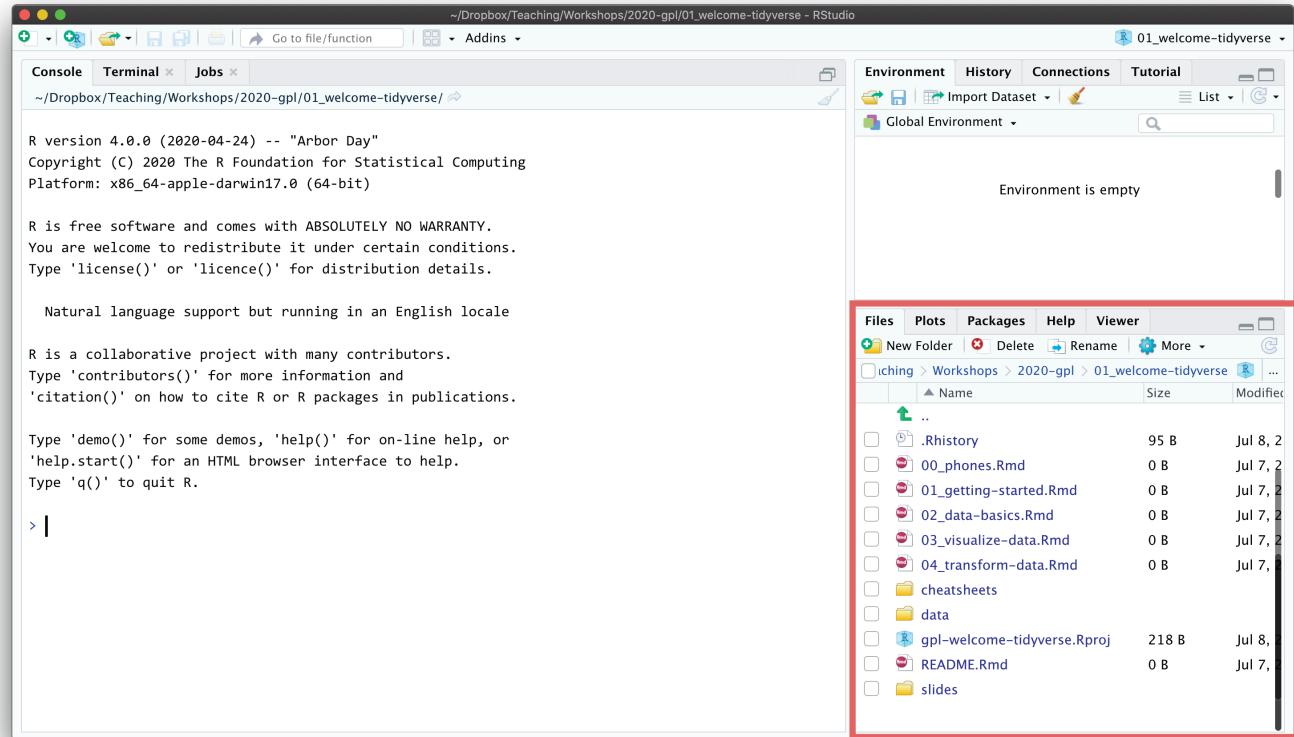
New Project Wizard

## Create Project

-  **New Directory**  
Start a project in a brand new working directory >
-  **Existing Directory**  
Associate a project with an existing working directory >
-  **Version Control**  
Checkout a project from a version control repository >

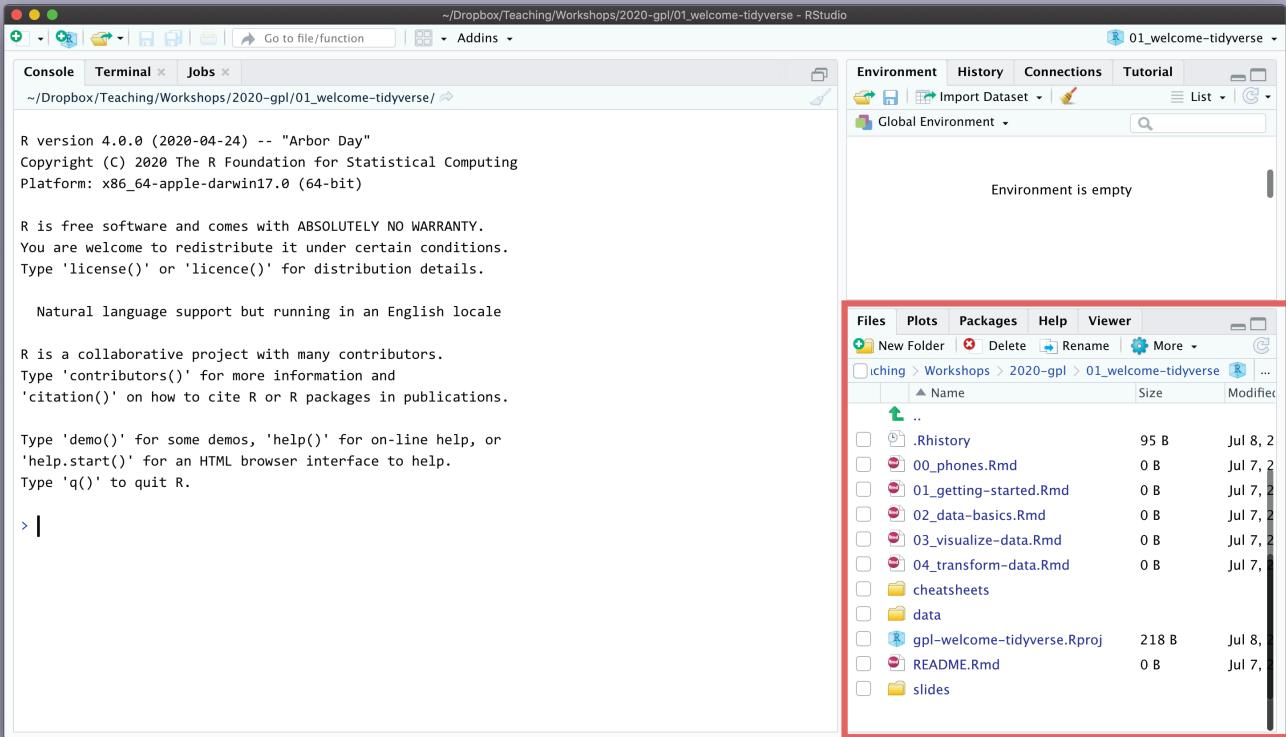
Cancel

# Panneau avec fichiers



Tous les fichiers dans  
votre répertoire de  
travail

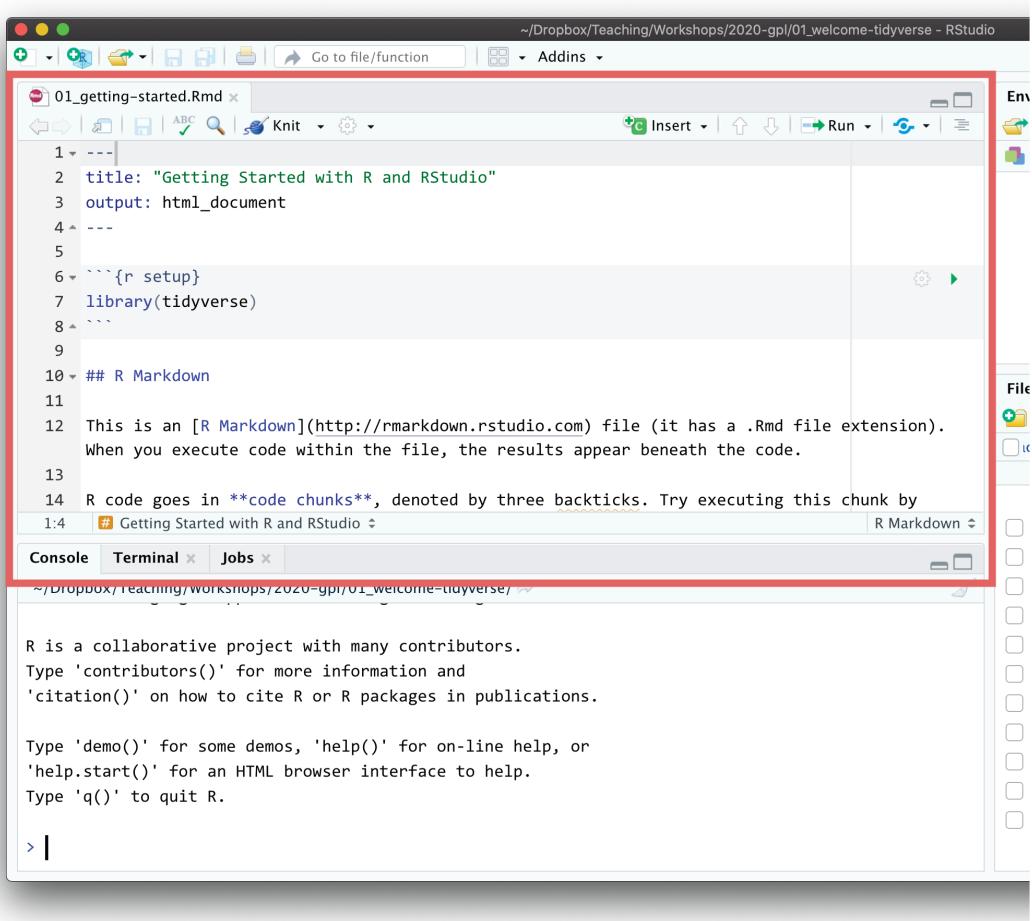
# À votre tour



Trouvez `a-intro.qmd`

Cliquez sur le nom pour ouvrir le fichier

# Panneau source

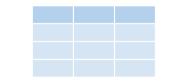
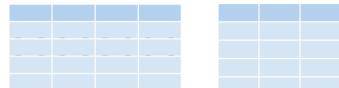
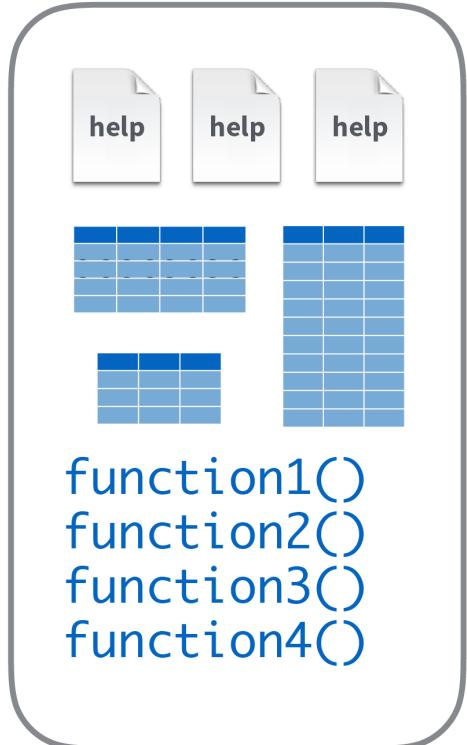


The screenshot shows the RStudio interface. On the left, the code editor displays the contents of a file named "01\_getting-started.Rmd". The code includes R Markdown syntax like backticks for code chunks and a library call to tidyverse. The right side of the interface shows the RStudio environment pane, which contains a sidebar titled "File" with several open tabs. Below the sidebar, there is a list of open files or documents. At the bottom of the interface, the R console window shows the standard R startup message and a prompt for further input.

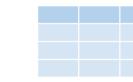
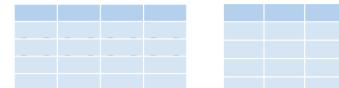
```
1: ---  
2: title: "Getting Started with R and RStudio"  
3: output: html_document  
4: ---  
5:  
6: `r setup`  
7: library(tidyverse)  
8:  
9:  
10: ## R Markdown  
11:  
12: This is an [R Markdown](http://rmarkdown.rstudio.com) file (it has a .Rmd file extension).  
When you execute code within the file, the results appear beneath the code.  
13:  
14: R code goes in **code chunks**, denoted by three backticks. Try executing this chunk by  
1:4: # Getting Started with R and RStudio R Markdown  
Console Terminal Jobs  
~/Dropbox/Teaching/Workshops/2020-gpl/01_welcome-tidyverse/ ~  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
> |
```

Les documents  
s'ouvrent ici

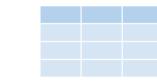
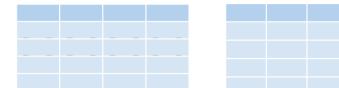
# Paquets



function5()  
function6()  
function7()  
function8()

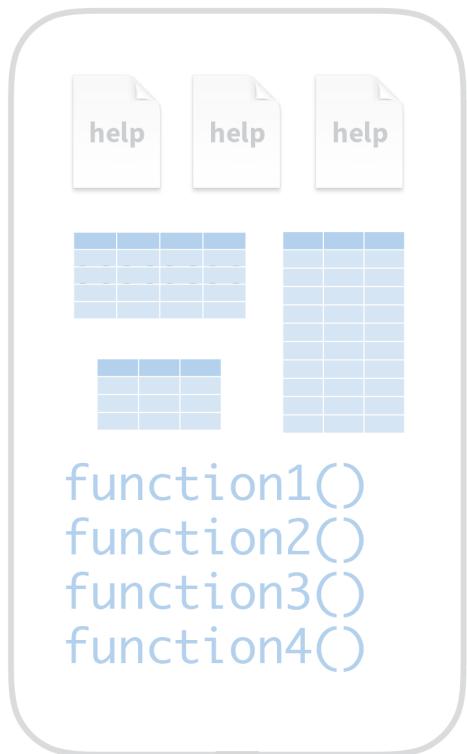


function9()  
functionA()  
functionB()  
functionC()

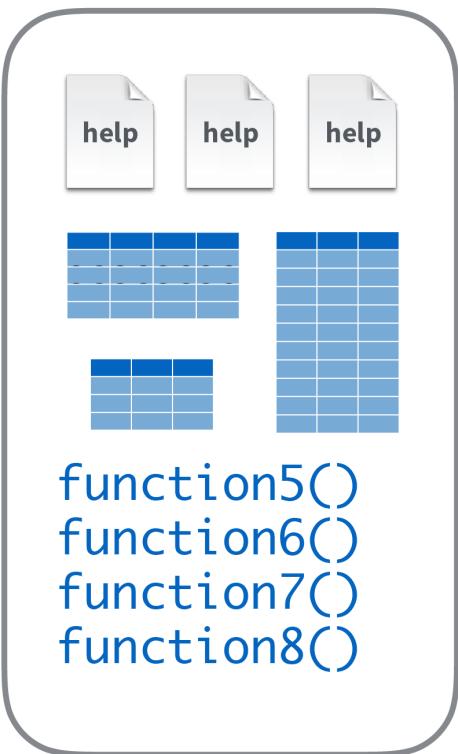


functionD()  
functionE()  
functionF()  
functionG()

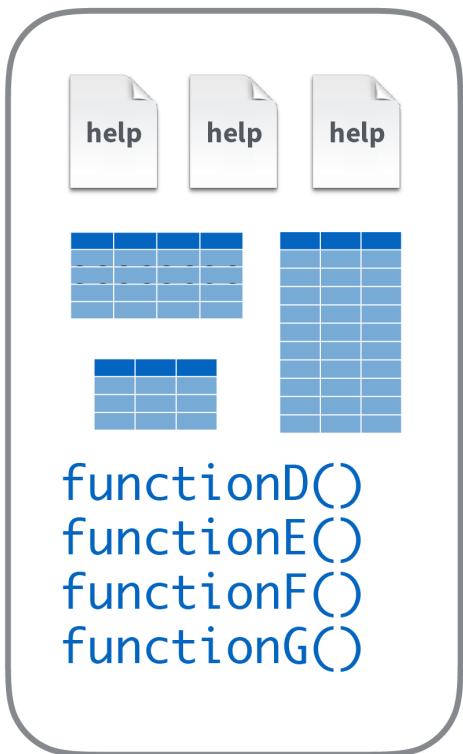
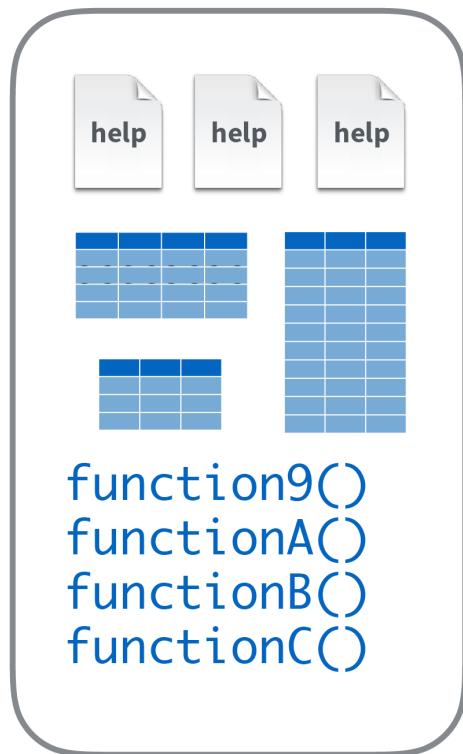
**Base R**



Base R



R Packages



# Utiliser des paquets

```
install.packages("nom")
```

```
library(nom)
```

Télécharge les fonctionnalités  
sur votre ordinateur.

Une fois par ordinateur.

Fichiers téléchargés via le CRAN.

Charge le paquet

Effectuer une fois par session

# À votre tour

Installez le paquet `remotes`

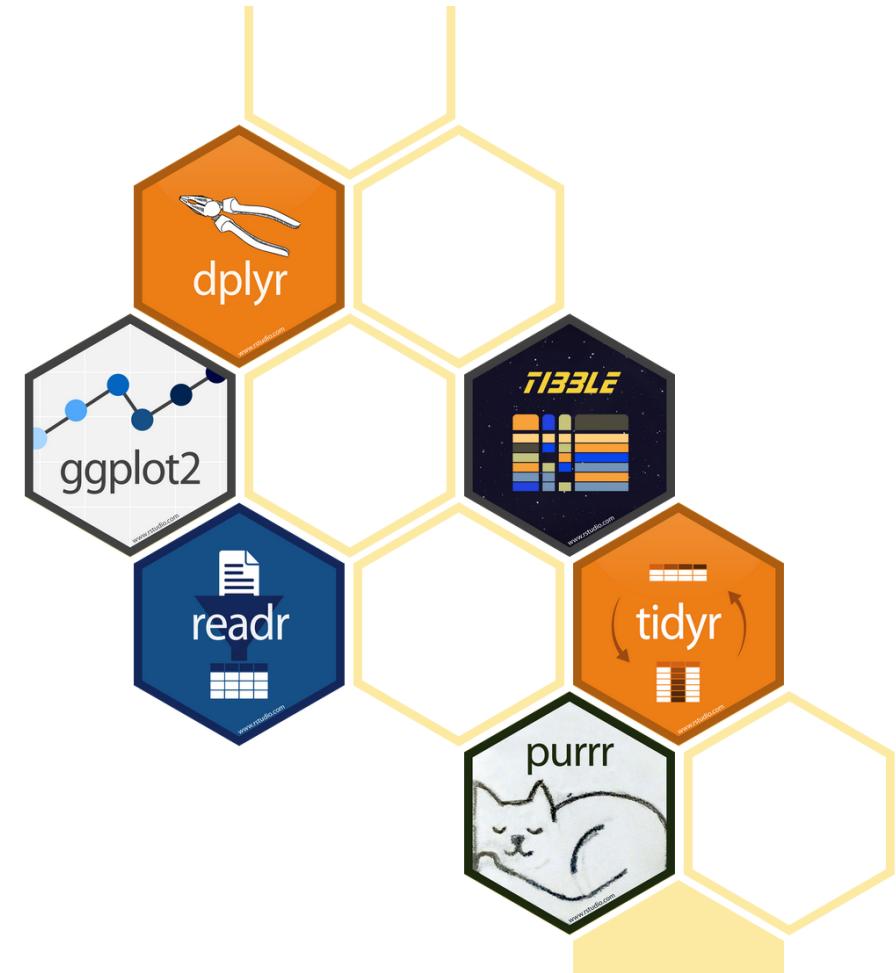
Utilisez les commandes suivantes pour installer `hecmulti`:

```
remotes::install_github("lbelzile/hecmulti")
```

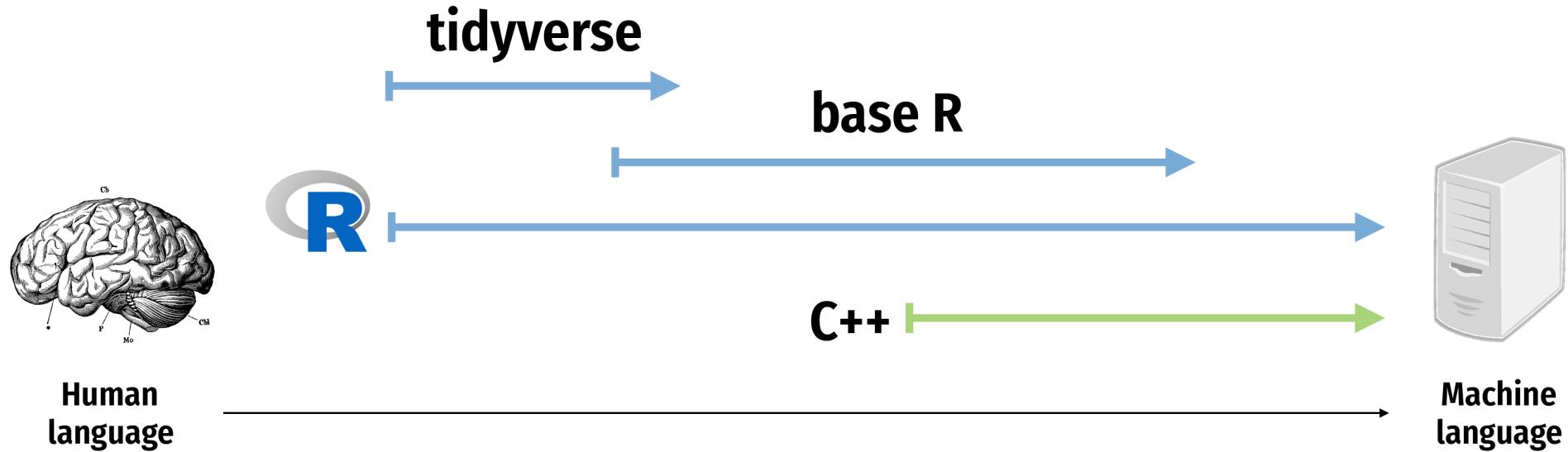
02:00

# tidyverse

« Le tidyverse est une collection de paquets R conçus par des programmeurs opiniâtre pour la science des données. Tous les paquets partagent la même philosophie de conception, et ont une grammaire et une structure commune. »

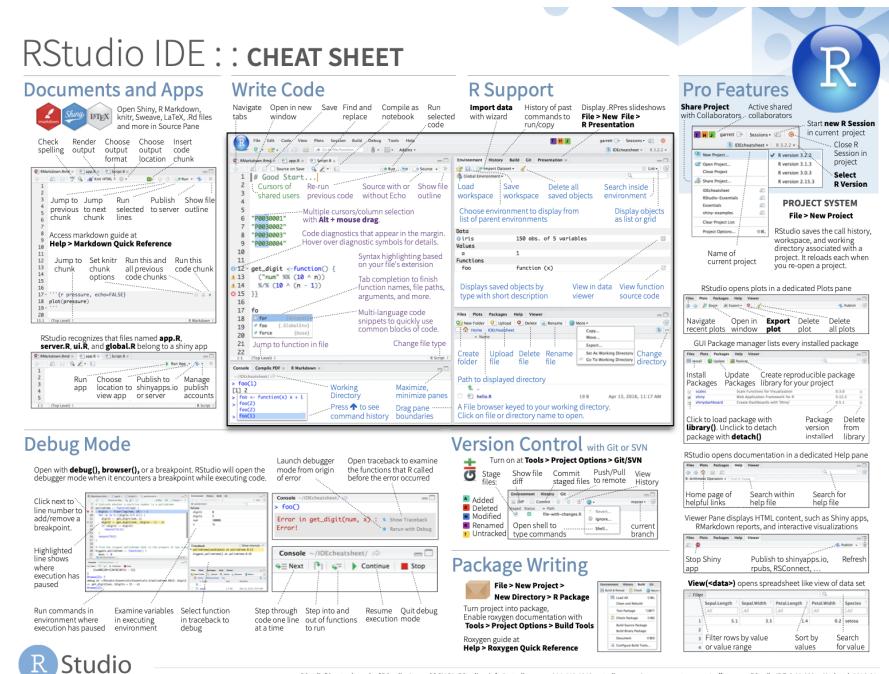


# tidyverse



# Aides-mémoires

Allez à Help > Cheatsheets pour trouver des guides de référence sur certains différents paquets



The image is a comprehensive RStudio IDE cheat sheet titled "RStudio IDE :: CHEAT SHEET". It is organized into several sections:

- Documents and UI Apps**: Shows keyboard shortcuts for navigating between documents and UI applications.
- Write Code**: A large section showing various keyboard shortcuts for code editing, including navigation, selection, and execution.
- R Support**: A screenshot of the RStudio interface showing the R console, workspace browser, and help viewer.
- Pro Features**: A screenshot of the RStudio interface showing the project system, package manager, and version control features.
- Debug Mode**: A screenshot of the RStudio interface showing the debugger mode with a call stack and variable inspection.
- Version Control**: A screenshot of the RStudio interface showing the Git or SVN integration.
- Package Writing**: A screenshot of the RStudio interface showing the roxygen documentation tool.

The bottom of the page includes the R Studio logo and a small footer with copyright information.

RStudio® is a trademark of RStudio, Inc. • CC BY SA RStudio - info@rstudio.com • 844-448-1212 • rstudio.com • Learn more at www.rstudio.com • RStudio IDE 0.99.832 • Updated: 2016-01

# Paquets et fonctionnalités

**Charger un paquet ajoute à l'environnement toutes ses fonctions et bases de données.**

**Peut masquer des fonctions homonymes!**

```
> library(tidyverse)
— Attaching packages ━━━━━━━━━━ tidyverse 1.3.2 ━━━━━━
✓ ggplot2 3.3.6      ✓ purrr    0.3.4
✓ tibble   3.1.7      ✓ stringr  1.4.0
✓ tidyr    1.2.0      ✓forcats  0.5.1
✓ readr    2.1.2
— Conflicts ━━━━━━━━━━ tidyverse_conflicts() ━━━━━━
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
```

# Fonctionnalités

## Charger une base de données

```
data(bd, package = "nomdupaquet")
```

## Utiliser directement une fonction sans charger un paquet

```
paquet::fonction(...)
```

# À votre tour

**Chargez les données renfe du paquet hecmulti**

```
data(renfe, package = 'hecmulti')  
renfe
```

01:00

# Quarto

The screenshot shows the Quarto IDE interface. At the top, there's a toolbar with various icons like file operations, render, and addins. Below it is a menu bar with Environment, History, Connections, Build, Git, and Tutorial. The main area has tabs for Files, Plots, Packages, Help, Viewer, and Presentation. A sidebar on the left contains a code editor with R code and a preview pane. The preview pane shows a document titled "Hello, Quarto" with sections like "Meet Quarto" and "Meet the penguins", each with its own sub-sections and images. At the bottom, there's a plot viewer showing a scatter plot of penguin flipper and bill lengths.

Format de document  
qui combine texte et code

Un livret  
pour votre analyse

# Quarto

## • ## Meet the penguins

The `penguins` data from the [\[\\*\\*palmerpenguins\\*\\*\]\(https://allisonhorst.github.io/palmerpenguins\)](https://allisonhorst.github.io/palmerpenguins) package contains size measurements for `r nrow(penguins)` penguins from three species observed on three islands in the Palmer Archipelago, Antarctica.

The plot below shows the relationship between flipper and bill lengths of these penguins.

## Texte

# Quarto

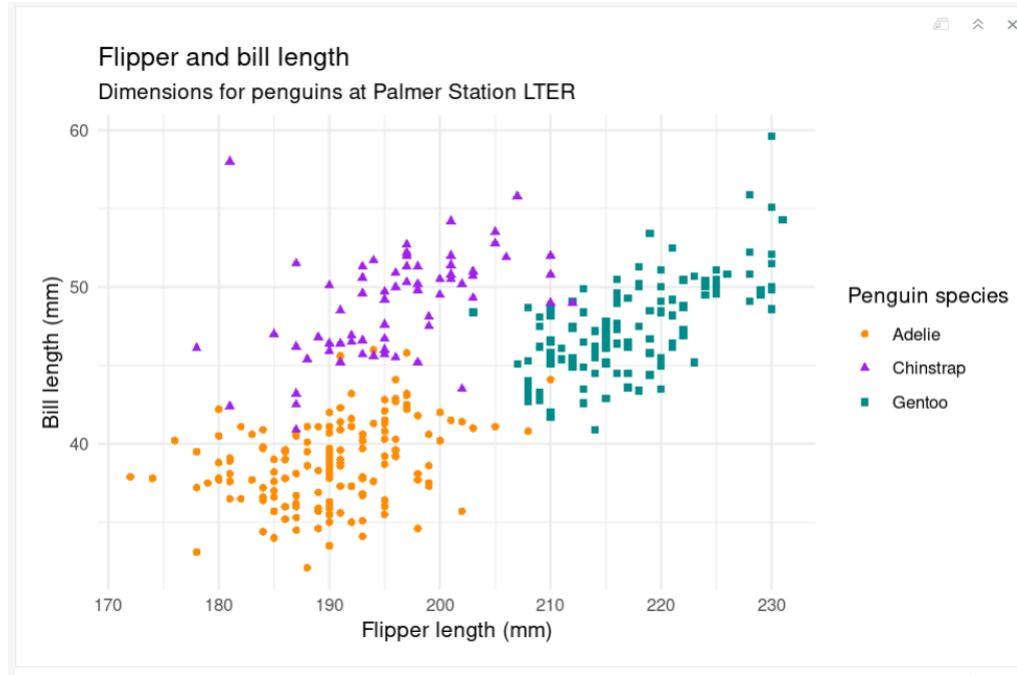
```
```{r}
#| label: plot-penguins
#| warning: false
#| echo: false

ggplot(penguins,
       aes(x = flipper_length_mm, y = bill_length_mm)) +
  geom_point(aes(color = species, shape = species)) +
  scale_color_manual(values = c("darkorange", "purple", "cyan4")) +
  labs(
    title = "Flipper and bill length",
    subtitle = "Dimensions for penguins at Palmer Station LTER",
    x = "Flipper length (mm)", y = "Bill length (mm)",
    color = "Penguin species", shape = "Penguin species"
  ) +
  theme_minimal()
````
```

Texte

Code

# Quarto



Texte

Code

Sortie

# À votre tour

```
```{r}
#| label: plot-penguins
#| warning: false
#| echo: false

ggplot(penguins,
       aes(x = flipper_length_mm, y = bill_length_mm)) +
  geom_point(aes(color = species, shape = species)) +
  scale_color_manual(values = c("darkorange", "purple", "cyan4")) +
  labs(
    title = "Flipper and bill length",
    subtitle = "Dimensions for penguins at Palmer Station LTER",
    x = "Flipper length (mm)", y = "Bill length (mm)",
    color = "Penguin species", shape = "Penguin species"
  ) +
  theme_minimal()
```
```

Lisez les instructions

Compilez le bloc de code  
avec le bouton démarrer

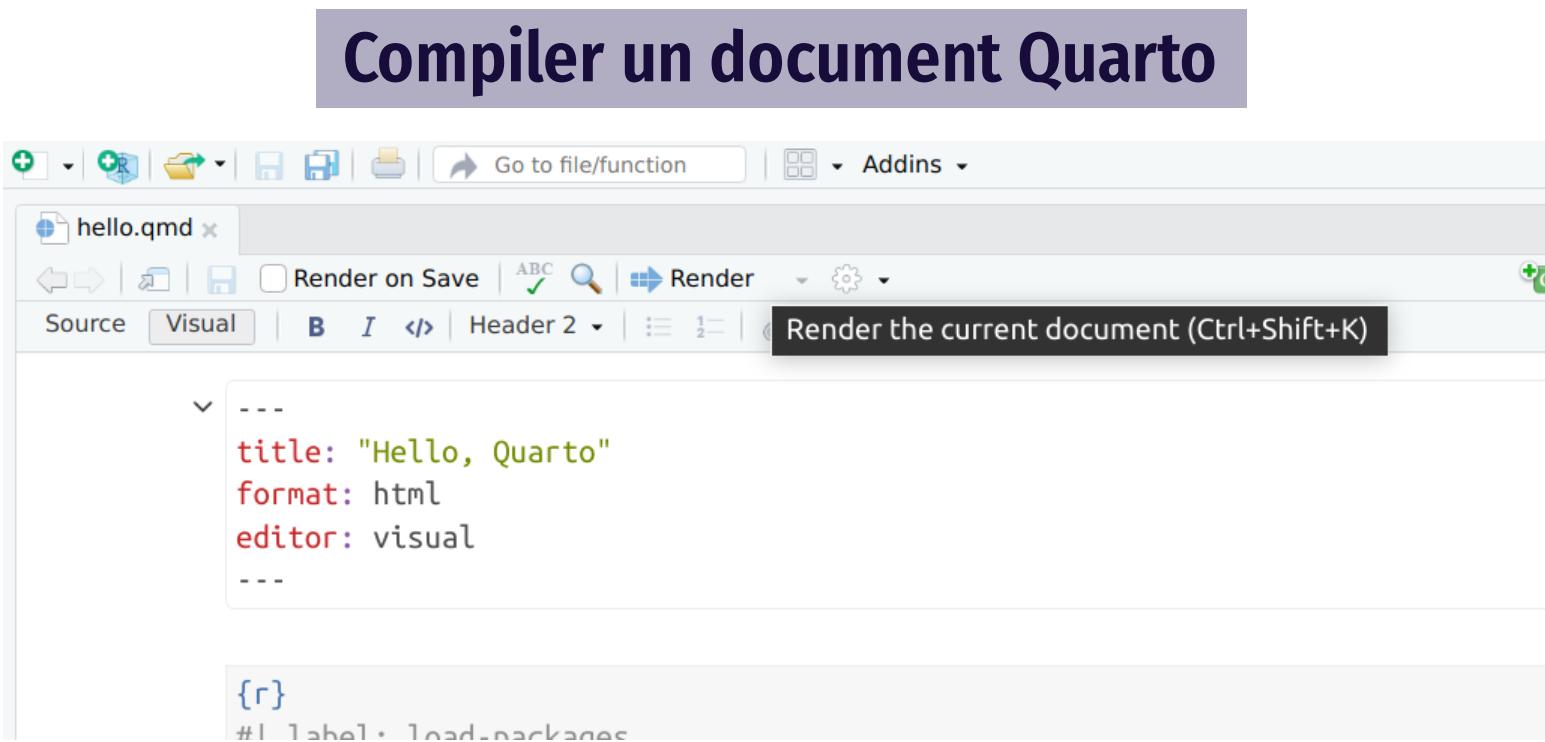
(icône avec triangle vert)

# À votre tour

Ajoutez un nouveau bloc

Inscrivez  $2 + 2$  et compilez

# Compilation



# À votre tour

**Quelle est la différence entre:**

```
filter(mtcars, cyl == 4)
```

```
four_cyls <- filter(mtcars, cyl == 4)
```

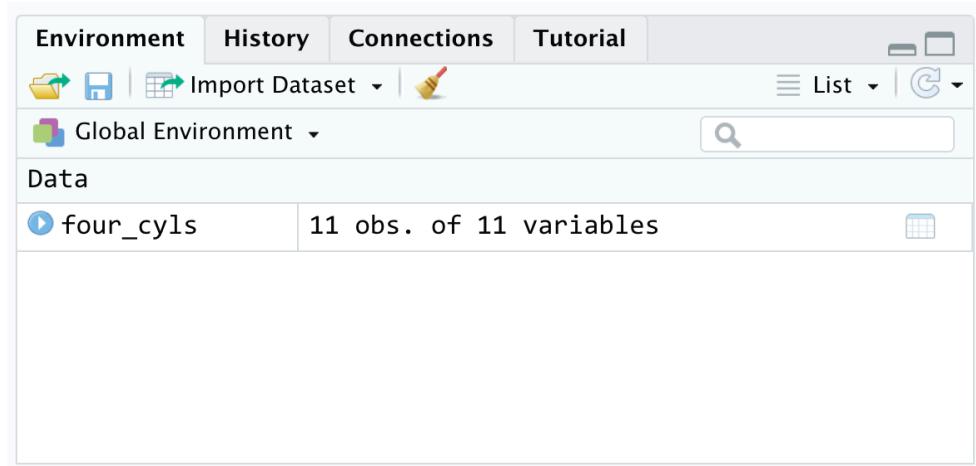
**Trouvez ces instructions dans votre carnet et compilez ce dernier.  
Quelle est la différence en sortie?**

# Assignation

**<- assigne la sortie à droite à la variable de gauche**

```
four_cyls <- filter(mtcars, cyl == 4)
```

# Panneau environnement



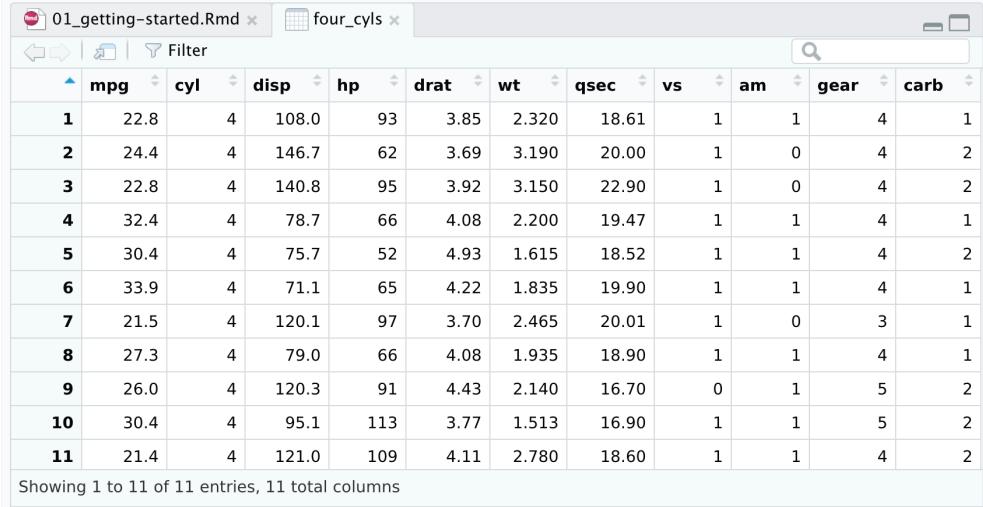
**Liste de toutes les variables existantes**

# À votre tour

**Trouvez `four_cyls` dans le panneau environnement.  
Cliquez sur le nom `four_cyls`**

**Qu'est-ce qui arrive?**

# Visualisation



|    | mpg  | cyl | disp  | hp  | drat | wt    | qsec  | vs | am | gear | carb |
|----|------|-----|-------|-----|------|-------|-------|----|----|------|------|
| 1  | 22.8 | 4   | 108.0 | 93  | 3.85 | 2.320 | 18.61 | 1  | 1  | 4    | 1    |
| 2  | 24.4 | 4   | 146.7 | 62  | 3.69 | 3.190 | 20.00 | 1  | 0  | 4    | 2    |
| 3  | 22.8 | 4   | 140.8 | 95  | 3.92 | 3.150 | 22.90 | 1  | 0  | 4    | 2    |
| 4  | 32.4 | 4   | 78.7  | 66  | 4.08 | 2.200 | 19.47 | 1  | 1  | 4    | 1    |
| 5  | 30.4 | 4   | 75.7  | 52  | 4.93 | 1.615 | 18.52 | 1  | 1  | 4    | 2    |
| 6  | 33.9 | 4   | 71.1  | 65  | 4.22 | 1.835 | 19.90 | 1  | 1  | 4    | 1    |
| 7  | 21.5 | 4   | 120.1 | 97  | 3.70 | 2.465 | 20.01 | 1  | 0  | 3    | 1    |
| 8  | 27.3 | 4   | 79.0  | 66  | 4.08 | 1.935 | 18.90 | 1  | 1  | 4    | 1    |
| 9  | 26.0 | 4   | 120.3 | 91  | 4.43 | 2.140 | 16.70 | 0  | 1  | 5    | 2    |
| 10 | 30.4 | 4   | 95.1  | 113 | 3.77 | 1.513 | 16.90 | 1  | 1  | 5    | 2    |
| 11 | 21.4 | 4   | 121.0 | 109 | 4.11 | 2.780 | 18.60 | 1  | 1  | 4    | 2    |

Showing 1 to 11 of 11 entries, 11 total columns

**Cliquer sur un objet dans le panneau environnement l'ouvre dans un onglet de visualisation interactive.**

# Objets, classes et types

# Vecteurs

Objet de base dans R

Un vecteur est une liste ordonnée de valeurs de même type  
(texte, nombres, etc.)

Concaténez avec `c()`:

```
c(1, 4, 2, 5, 7)
```

# Listes

**Une liste est une collection d'objets**

```
maliste <- list(elementun = 2:4, FALSE)  
maliste
```

```
## $elementun  
## [1] 2 3 4  
##  
## [[2]]  
## [1] FALSE
```

**Format de sortie fréquent pour les fonctions!**

# Éléments d'une liste

**Accéder à un élément  
par nom avec \$**

```
maliste$elementun
```

**Accéder à un élément  
par position avec [[ ]]**

```
maliste[[1]]
```

# Base de données (data.frame)

Une liste avec dimensions

Chaque colonne est un vecteur

Variables de types potentiellement différents

Mais de même longueur

**tibble (tbl): version tidyverse, plus sophistiquée**

# Types de données de base

|                              |               |   |
|------------------------------|---------------|---|
| <b>Entier (integer)</b>      | Nombre entier | <code>c(1L, 2L, 3L, 4L); 1:4</code>               |
| <b>Double</b>                | Nombres       | <code>c(1, 2.4, 3.14, 4)</code>                   |
| <b>Caractère (character)</b> | Texte         | <code>c("1", "bleu", "plaisir", "monstre")</code> |
| <b>Logique (logical)</b>     | Vrai ou faux  | <code>c(TRUE, FALSE, TRUE, FALSE)</code>          |

Utiliser la fonction `typeof()`  
pour déterminer le type d'un vecteur.

# Facteurs (factor, <fct>)

## Vecteurs de variables catégorielles

Encodées comme entiers avec étiquettes  
(étiquettes visible via `levels`)

## Traitement spécial dans les modèles

```
facteur <- factor(x = c(1,2,1,3),  
                   labels = c("petit", "moyen", "grand"))  
facteur #méthode print
```

```
## [1] petit moyen petit grand  
## Levels: petit moyen grand
```

# Valeurs manquantes

**NA, NaN, NA\_integer\_, etc.**

**is.na() pour détecter**

**summary() SUR data.frame  
pour décompte par variable**

# Fonctions

## Fonctions égal actions

Arguments en intrant, résultats en sortie

Spécifier les arguments avec `nomarg =`, sinon en ordre

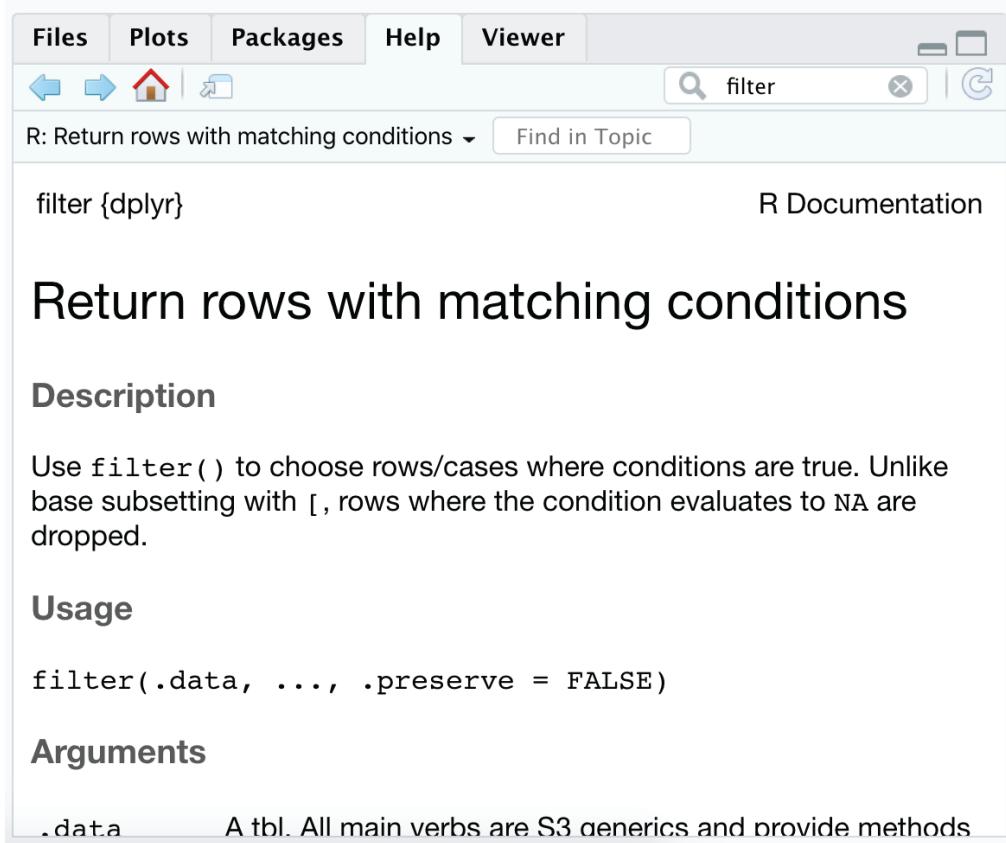
# Aide

**Pour voir la page d'aide sur une fonction R,  
tapez dans la console:**

```
?nom_fonction  
# Fonction pas dans l'environnement ou chaîne de caractères  
??"normal distribution"
```

**(ou utilisez un moteur de recherche!)**

# Panneau d'aide



The screenshot shows the R Help Viewer interface. The title bar includes 'Files', 'Plots', 'Packages', 'Help', and 'Viewer' tabs. A search bar at the top right contains the word 'filter'. Below the search bar, a dropdown menu says 'R: Return rows with matching conditions' and a 'Find in Topic' button. The main content area has a header 'filter {dplyr}' and 'R Documentation'. The text starts with 'Return rows with matching conditions'. It includes sections for 'Description' (explaining the function's purpose), 'Usage' (showing the command: `filter(.data, ..., .preserve = FALSE)`), and 'Arguments' (describing the `.data` argument). The text for 'Arguments' is partially cut off.

filter {dplyr} R Documentation

Return rows with matching conditions

**Description**

Use `filter()` to choose rows/cases where conditions are true. Unlike base subsetting with `[`, rows where the condition evaluates to NA are dropped.

**Usage**

```
filter(.data, ..., .preserve = FALSE)
```

**Arguments**

`.data` A `tbl`. All main verbs are S3 generics and provide methods

**Ces fichiers contiennent notamment des détails sur les arguments des fonctions**

**Il y a souvent des exemples à la fin**

# À votre tour

Lisez l'aide pour la fonction `seq`

Ajouter un bloc qui utilise `seq()` pour créer un vecteur de nombres de 5 à 30, espacés par sauts de 5  
(5, 10, 15, 20, ...)

02 : 00

```
seq(from = 5, to = 30, by = 5)
```

```
## [1] 5 10 15 20 25 30
```

# Problèmes fréquents de syntaxe #1

## Parenthèses ou accolades de fermeture manquante

```
mean(mtcars
```

```
"Oops problème
```

# Problèmes fréquents de syntaxe #2

Ajouter des accolades quand ce n'est pas adéquat (et vice versa)

```
mean("mtcars")
```

```
## Warning in mean.default("mtcars"): argument is not numeric or logical:  
## returning NA
```

```
## [1] NA
```

# Problèmes fréquents de syntaxe #3

## Mauvais nom de fonction (**R** est sensible à la casse)

```
MEAN(mtcars)
```

```
## Error in MEAN(mtcars): could not find function "MEAN"
```

# À votre tour

**Il y a trois blocs sous "Syntaxe fautive"**

**Compilez chacune, lisez le message d'erreur et essayez de corriger le code**

# Classes et méthodes

La classe d'un objet permet de définir des fonctions génériques (`methods`)

```
methods(class = "lm")
```

Autrement dit, le résultat dépend de la classe.

## Méthodes fréquentes

impression  
(`print`)

récapitulatif  
(`summary`)

graphique  
(`plot`)

# Importer des données

# Paquets pour l'importation de données



Données texte brutes

```
my_data <- read_csv("file.csv")
```



Chiffriers Excel

```
my_data <- read_excel("file.xlsx")
```



Données Stata, SPSS, et SAS

```
my_data <- read_stata("file.dta")
```