

Statistiques avec



M2 Sciences du Langage

Remi.lafitte@univ-grenoble-alpes.fr

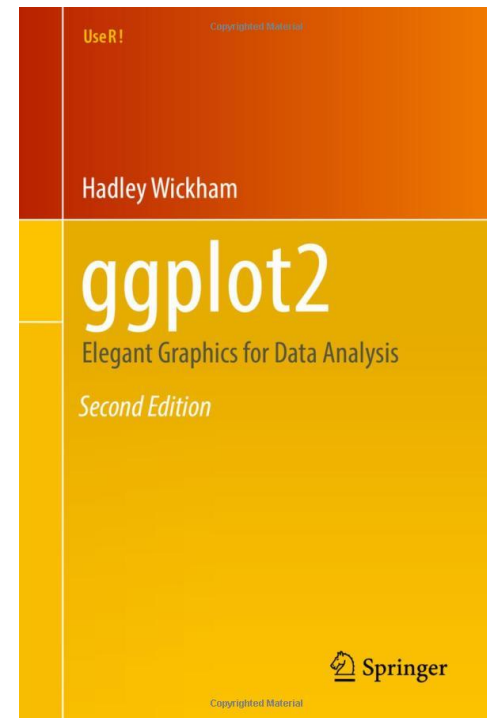
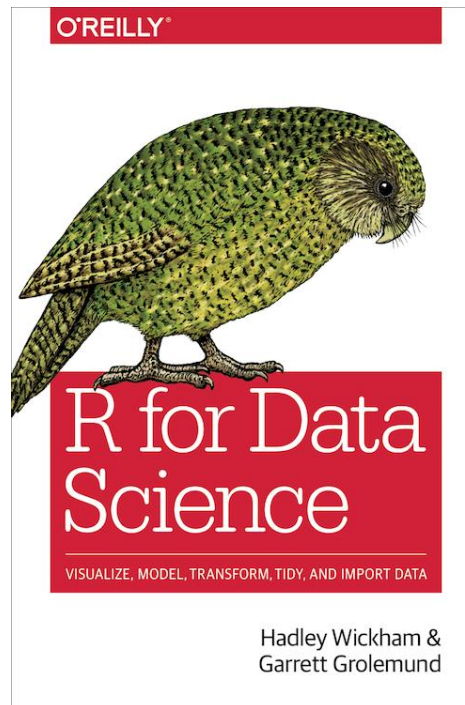
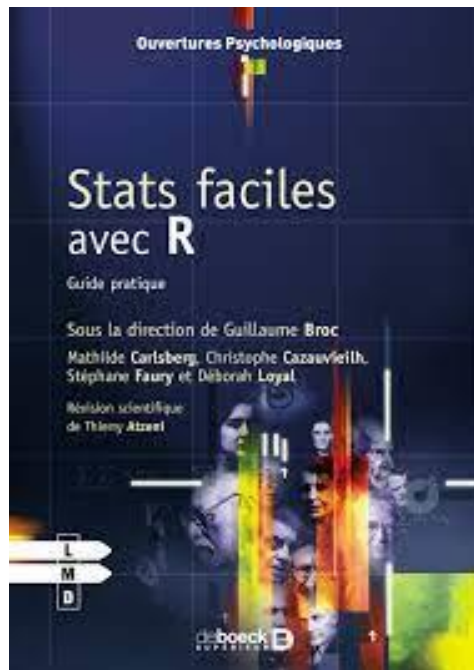
2023-2024

Ressources utiles

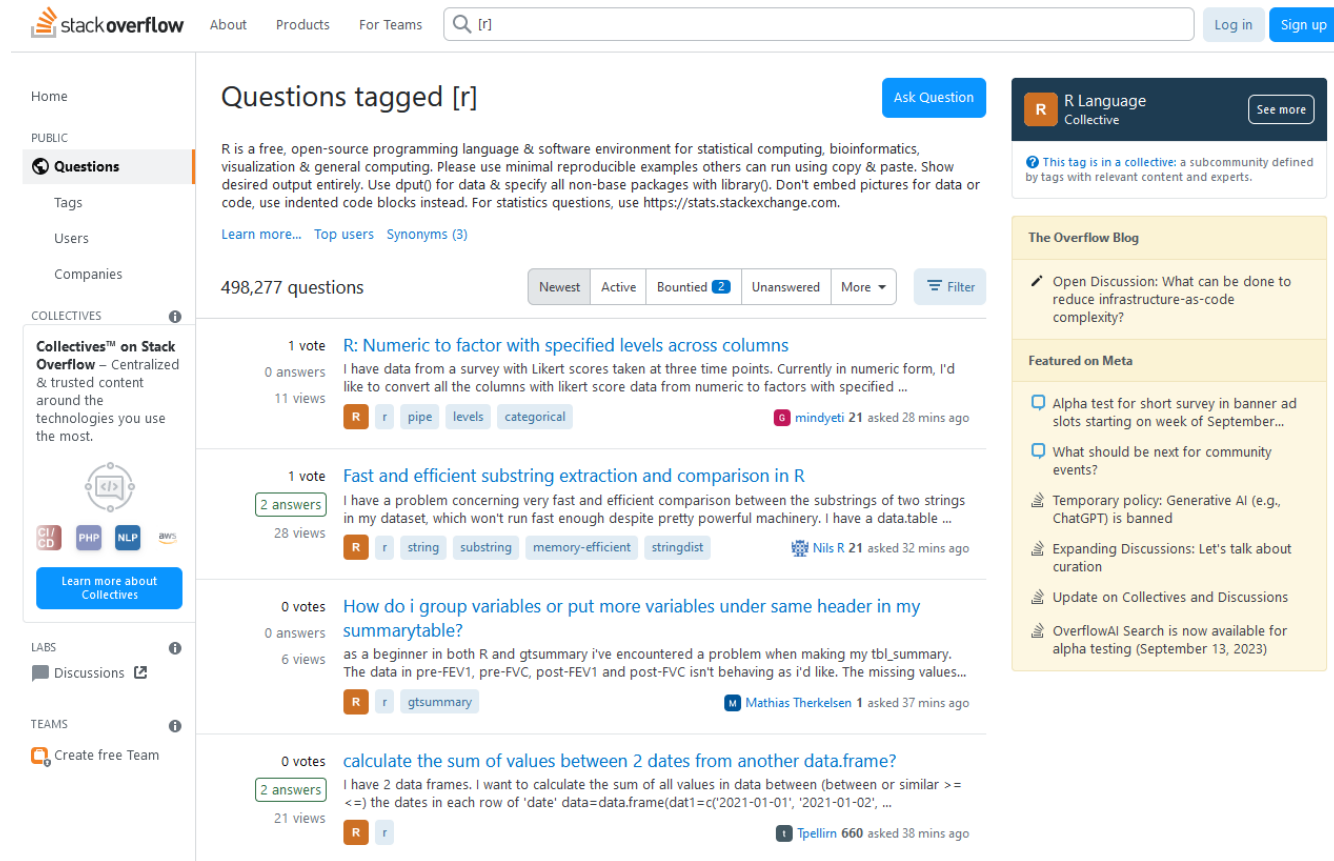


<https://r4ds.had.co.nz/>

<https://ggplot2-book.org/>



<https://stackoverflow.com/questions/tagged/r>



Stack Overflow About Products For Teams [Log in](#) [Sign up](#)

Questions tagged [r]

[Ask Question](#)

R is a free, open-source programming language & software environment for statistical computing, bioinformatics, visualization & general computing. Please use minimal reproducible examples others can run using copy & paste. Show desired output entirely. Use `dput()` for data & specify all non-base packages with `library()`. Don't embed pictures for data or code, use indented code blocks instead. For statistics questions, use <https://stats.stackexchange.com>.

[Learn more...](#) [Top users](#) [Synonyms \(3\)](#)

498,277 questions Newest Active Bountied 2 Unanswered More Filter

1 vote
0 answers
11 views

R: Numeric to factor with specified levels across columns

I have data from a survey with Likert scores taken at three time points. Currently in numeric form, I'd like to convert all the columns with likert score data from numeric to factors with specified ...

[R](#) [r](#) [pipe](#) [levels](#) [categorical](#)

[mindyeti](#) 21 asked 28 mins ago

1 vote
2 answers
28 views

Fast and efficient substring extraction and comparison in R

I have a problem concerning very fast and efficient comparison between the substrings of two strings in my dataset, which won't run fast enough despite pretty powerful machinery. I have a `data.table` ...

[R](#) [r](#) [string](#) [substring](#) [memory-efficient](#) [stringdist](#)

[Nils R](#) 21 asked 32 mins ago

0 votes
0 answers
6 views

How do I group variables or put more variables under same header in my summarytable?

as a beginner in both R and `gtsummary` I've encountered a problem when making my `tbl_summary`. The data in pre-FEV1, pre-FVC, post-FEV1 and post-FVC isn't behaving as I'd like. The missing values...

[R](#) [r](#) [gtsummary](#)

[Mathias Therkelsen](#) 1 asked 37 mins ago

0 votes
2 answers
21 views

calculate the sum of values between 2 dates from another data.frame?

I have 2 data frames. I want to calculate the sum of all values in data between (between or similar `>=` `<=`) the dates in each row of 'date' `data=data.frame(dat1=c('2021-01-01', '2021-01-02', ...`

[R](#) [r](#)

[Tpelir](#) 660 asked 38 mins ago

Collectives™ on Stack Overflow – Centralized & trusted content around the technologies you use the most.

[Learn more about Collectives](#)

LABS

[Discussions](#)

TEAMS

[Create free Team](#)

R Language Collective [See more](#)

This tag is in a collective: a subcommunity defined by tags with relevant content and experts.

The Overflow Blog

Open Discussion: What can be done to reduce infrastructure-as-code complexity?

Featured on Meta

Alpha test for short survey in banner ad slots starting on week of September...

What should be next for community events?

Temporary policy: Generative AI (e.g., ChatGPT) is banned

Expanding Discussions: Let's talk about curation

Update on Collectives and Discussions

OverflowAI Search is now available for alpha testing (September 13, 2023)

<https://posit.co/resources/cheatsheets/>
<https://stat545.com/>
<https://www.r-bloggers.com/blogs-list/>
<https://evalsp22.classes.andrewheiss.com/resource/r/>
https://stt4230.rbind.io/introduction/presentation_r/
<https://rstudio-education.github.io/hopr/>
<https://perso.ens-lyon.fr/lise.vaudor/grimoireStat/ book/intro.html>
<https://www.r-bloggers.com/>
<https://www.datanovia.com/en/>
<https://rcompanion.org/handbook/>
<https://r-graph-gallery.com/ggplot2-package.html>
<https://www.uvm.edu/~statdhtx/StatPages/R/ReadingData.html>
<https://fermin.perso.math.cnrs.fr/Files/IntroductionRStudio.html>

Blogs list

- - R
- - rstats
- --Jean Arreola--
- "R" you ready? - My advances in R - a learner's diary
- (R)very Day
- [citation needed] » R
- [R] tricks - Some tricks regarding [R]
- [R]appster
- [R]eliability
- @yaaang's blog » R
- #FunDataFriday - Little Miss Data
- #Stats - Emmanuel Qlámijüwõn | Digital Demographer | Health Researcher | Data Analyst
- %>% dreams
- 0xCAFEBABE
- A blog from Sydney
- A Blog On Data Analytics
- A HopStat and Jump Away » Rbloggers
- A Hugo website
- a Physicist in Wall Street
- A Pint of R
- A second megabyte of memory
- Aaron Schlegel's Notebook of Interesting Things - R
- Achim Zeileis
- Adventures in Analytics and Visualization
- Adventures in Statistical Computing
- AdventuresInData

Qui est R ?

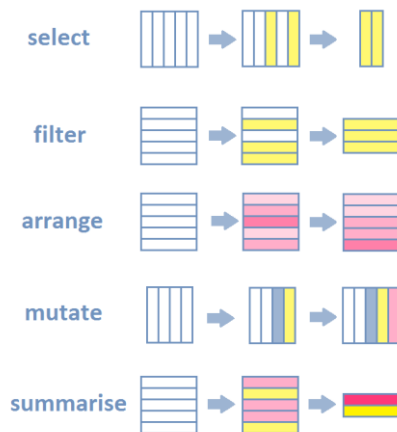
- logiciel d'analyse statistique et de graphiques
- créé en 1996 par Ross Ihaka et Robert Gentleman



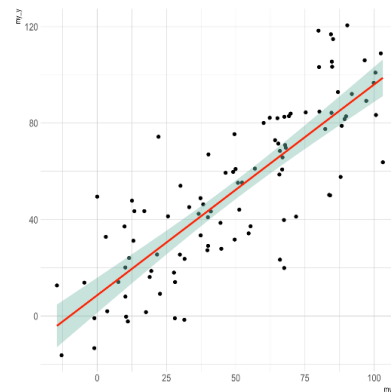
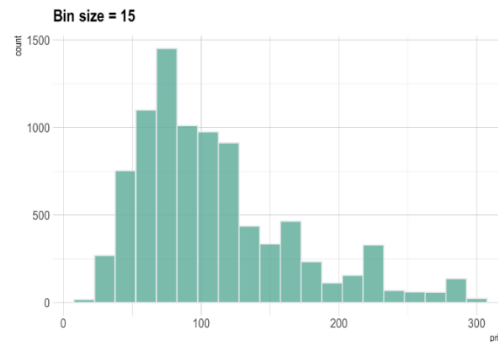
Pourquoi R ?

- Enorme souplesse !
- Simple à apprendre et gratuit

Manipulation des données



graphiques



statistiques

```
age
Min.   :18.0
1st Qu.:19.0
Median :24.1
Mean   :24.1
3rd Qu.:27.8
Max.   :34.0
```

```
> # Perform the Chi-Square test.
> print(chisq.test(car_data))
```

Pearson's Chi-squared test

```
data:  car_data
X-squared = 33.001, df = 10, p-value = 0.0002723
```

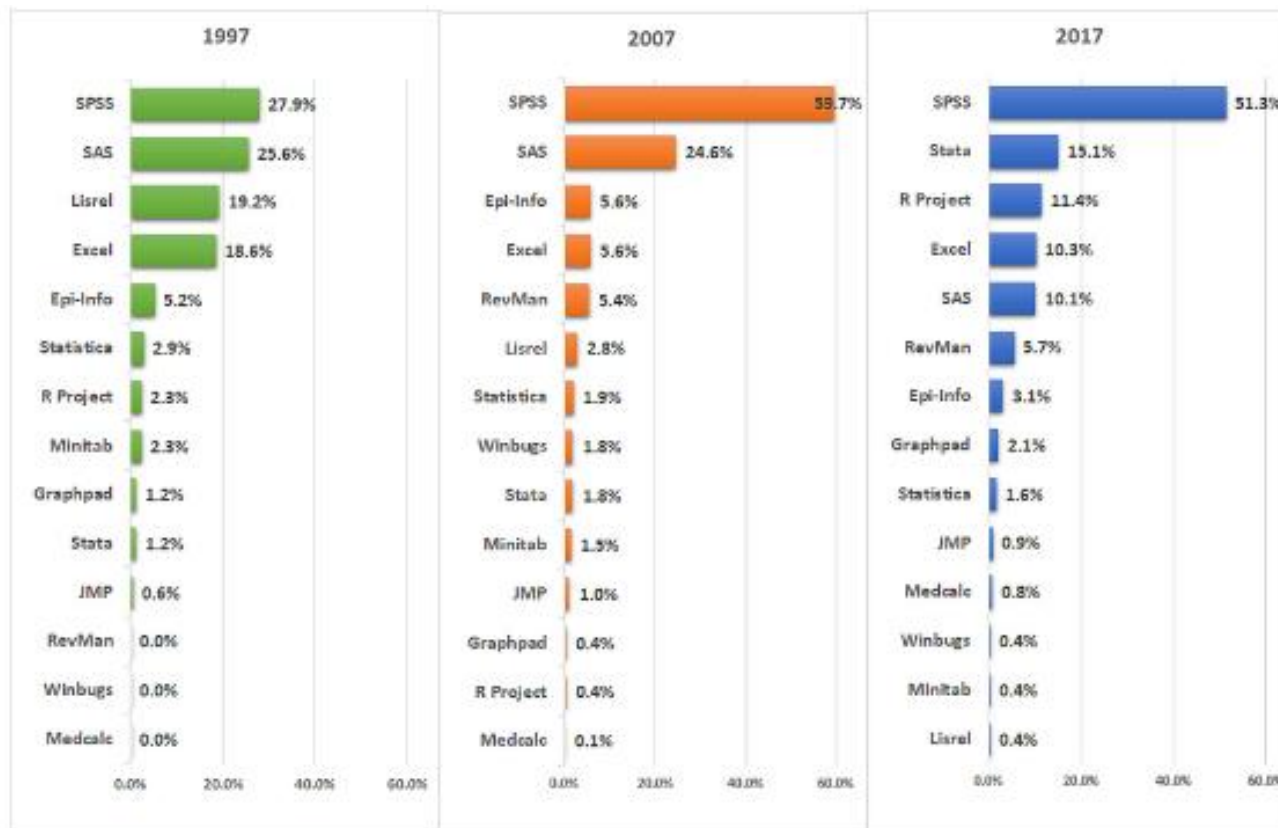
Pourquoi R ?



- De plus en plus populaire

Trends in the Usage of Statistical Software and Their Associated Study Designs in Health Sciences Research: A Bibliometric Analysis

Emad Masuadi¹, Mohamud Mohamud², Muhannad Almutairi³, Abdulaziz Alsunaidi³, Abdulmohsen K. Alswayed³, Omar F. Aldhafeeri³



Pourquoi R ?



- Permet une "reproductibilité" plus facile des analyses
 - “running the **same software** ...
 - on the **same input data** ...
 - and obtaining the **same results**”

Rougier, N. P., Hinsén, K., Alexandre, F., Arildsen, T., Barba, L. A., Benureau, F. C. Y., et al. (2017). *Sustainable Computational Science: The ReScience Initiative*. Available online at: <https://arxiv.org/abs/1707.04393>

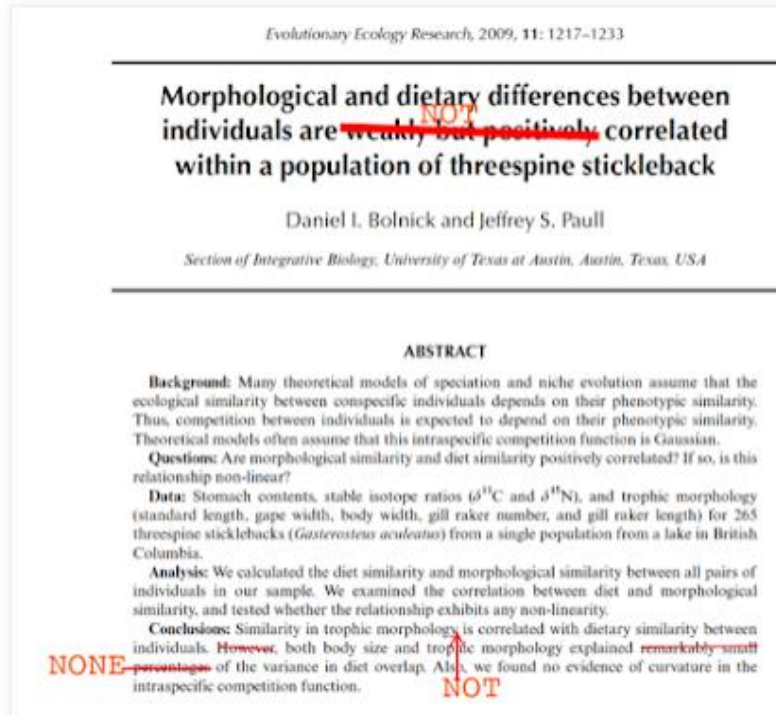


- Possibilité de « rapports automatisés » avec R Markdown (voir dernier TD)
- **Partage** du code avec des tiers <https://rmarkdown.rstudio.com/>
 - Code lisible/commenté
 - Code efficient (éviter les répétitions)
 - Code qui « roule » sur une autre machine

Pourquoi R ?



- Repérage et correction plus facile des erreurs



Drowning my R-sorrows in a glass of Hendry Zinfandel.

A more appropriate version of the first page of the newly retracted paper.

<http://ecoevoeco.blogspot.com/2016/12/wrong-lot.html>

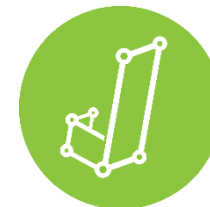
Pourquoi R ?



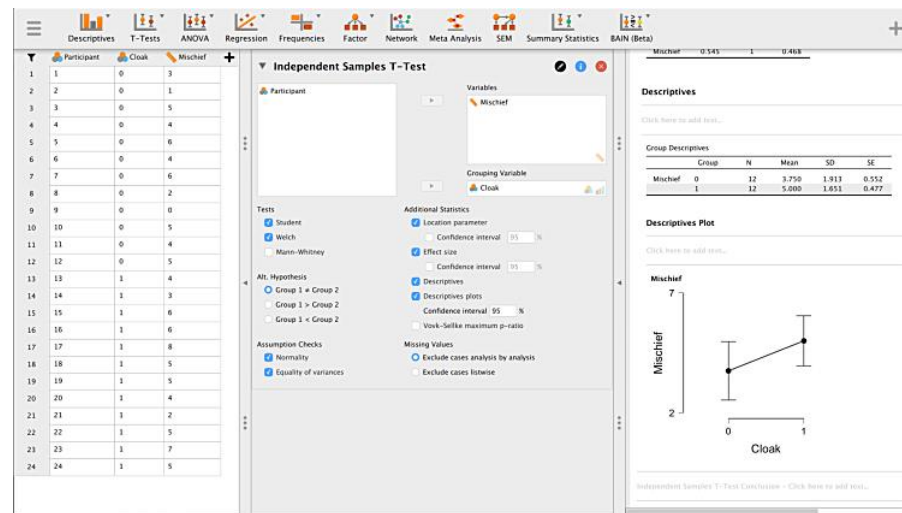
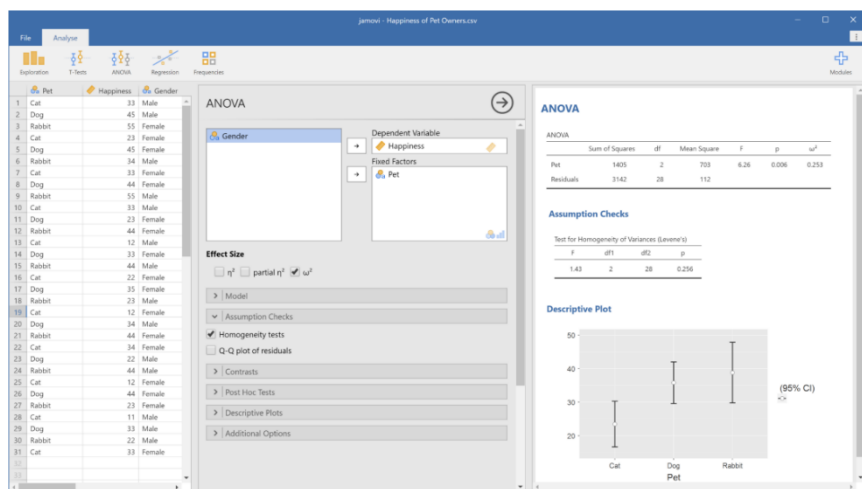
- Alternatives pour les moins R-enthousiastes



JAMOVI



JASP



Plan approximatif des TDs

- **TD1 27/10** **Les grandes bases de R**
- **TD2 17/11**
- **TD3 24/11** **khi-deux**
- **TD4 01/12** **corrélation**
- **TD5 08/12** **t-test**
- **TD6 15/12** **Graphiques avancés avec ggplot2**
- **TD6 15/12** **Rapports automatisés avec Rmarkdown**

Evaluation

Concepts et opérations basiques

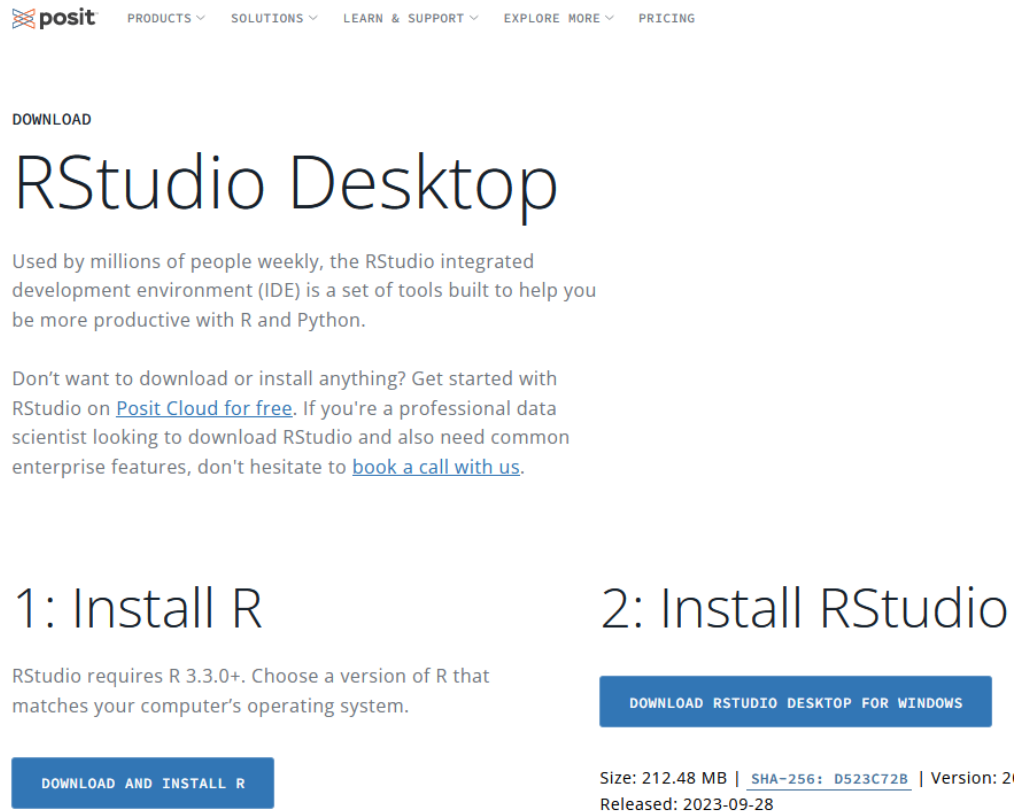


Concepts et opérations basiques

- Installer R et lancer une commande
- Fonctions
- Installer et charger un paquet
- Création et type d'objets
- Importation de données
- Objet et environnement
- Opérateurs, adressage et création de variables

Installation

- R et R studio
- <https://posit.co/download/rstudio-desktop/>



The screenshot shows the 'RStudio Desktop' download page on the Posit website. At the top, there's a navigation bar with the Posit logo and links for PRODUCTS, SOLUTIONS, LEARN & SUPPORT, EXPLORE MORE, and PRICING. Below the navigation bar, the word 'DOWNLOAD' is in small caps, followed by the main heading 'RStudio Desktop'. A paragraph describes RStudio as an integrated development environment (IDE) used by millions of people weekly. Below this, another paragraph offers alternatives like Posit Cloud for free or booking a call for enterprise features. The page is divided into two columns. The left column is titled '1: Install R' and contains a button labeled 'DOWNLOAD AND INSTALL R'. The right column is titled '2: Install RStudio' and contains a button labeled 'DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS'. At the bottom right, there is additional information: 'Size: 212.48 MB | SHA-256: D523C72B | Version: 2023.09.28 | Released: 2023-09-28'.

posit PRODUCTS SOLUTIONS LEARN & SUPPORT EXPLORE MORE PRICING

DOWNLOAD

RStudio Desktop

Used by millions of people weekly, the RStudio integrated development environment (IDE) is a set of tools built to help you be more productive with R and Python.

Don't want to download or install anything? Get started with RStudio on [Posit Cloud for free](#). If you're a professional data scientist looking to download RStudio and also need common enterprise features, don't hesitate to [book a call with us](#).

1: Install R

RStudio requires R 3.3.0+. Choose a version of R that matches your computer's operating system.

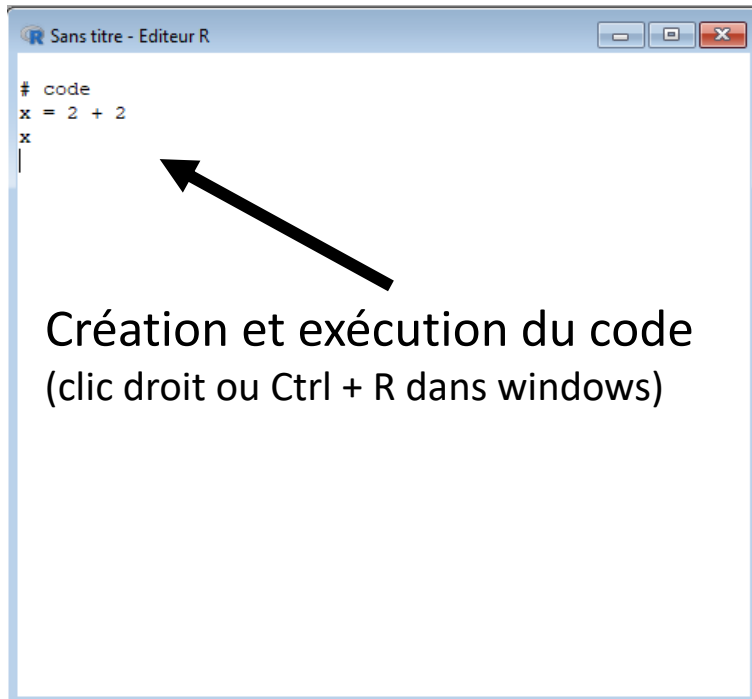
DOWNLOAD AND INSTALL R

2: Install RStudio

DOWNLOAD RSTUDIO DESKTOP FOR WINDOWS

Size: 212.48 MB | [SHA-256: D523C72B](#) | Version: 2023.09.28 | Released: 2023-09-28

Feuille de script

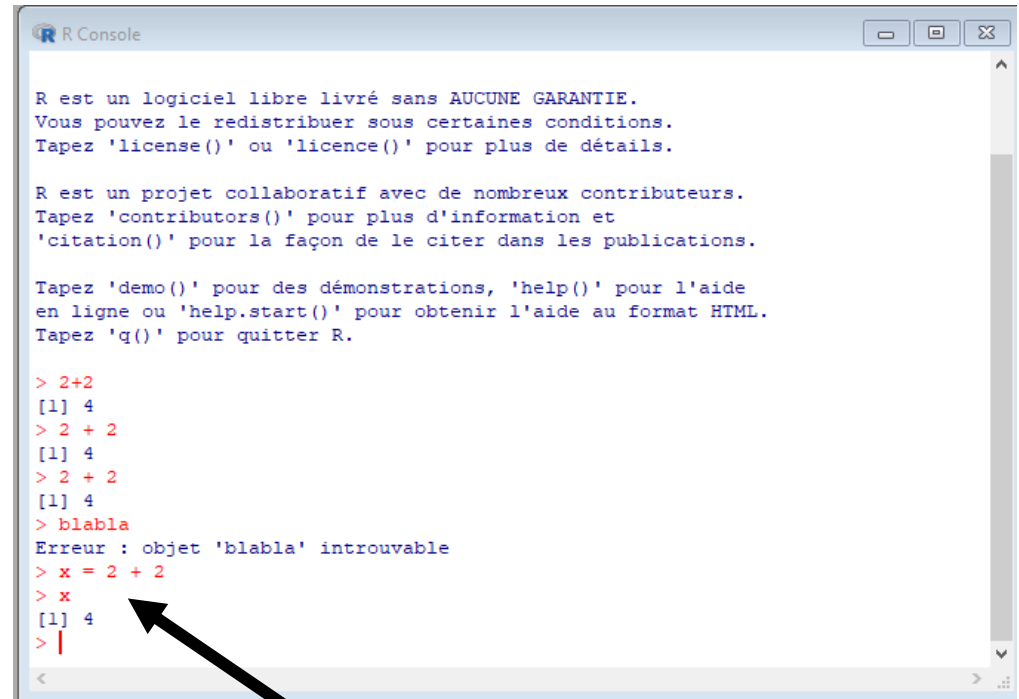


```
# code
x = 2 + 2
x
```

Création et exécution du code
(clic droit ou Ctrl + R dans windows)

→ Utilisation « austère » de R...

Console



```
R est un logiciel libre livré sans AUCUNE GARANTIE.
Vous pouvez le redistribuer sous certaines conditions.
Tapez 'license()' ou 'licence()' pour plus de détails.

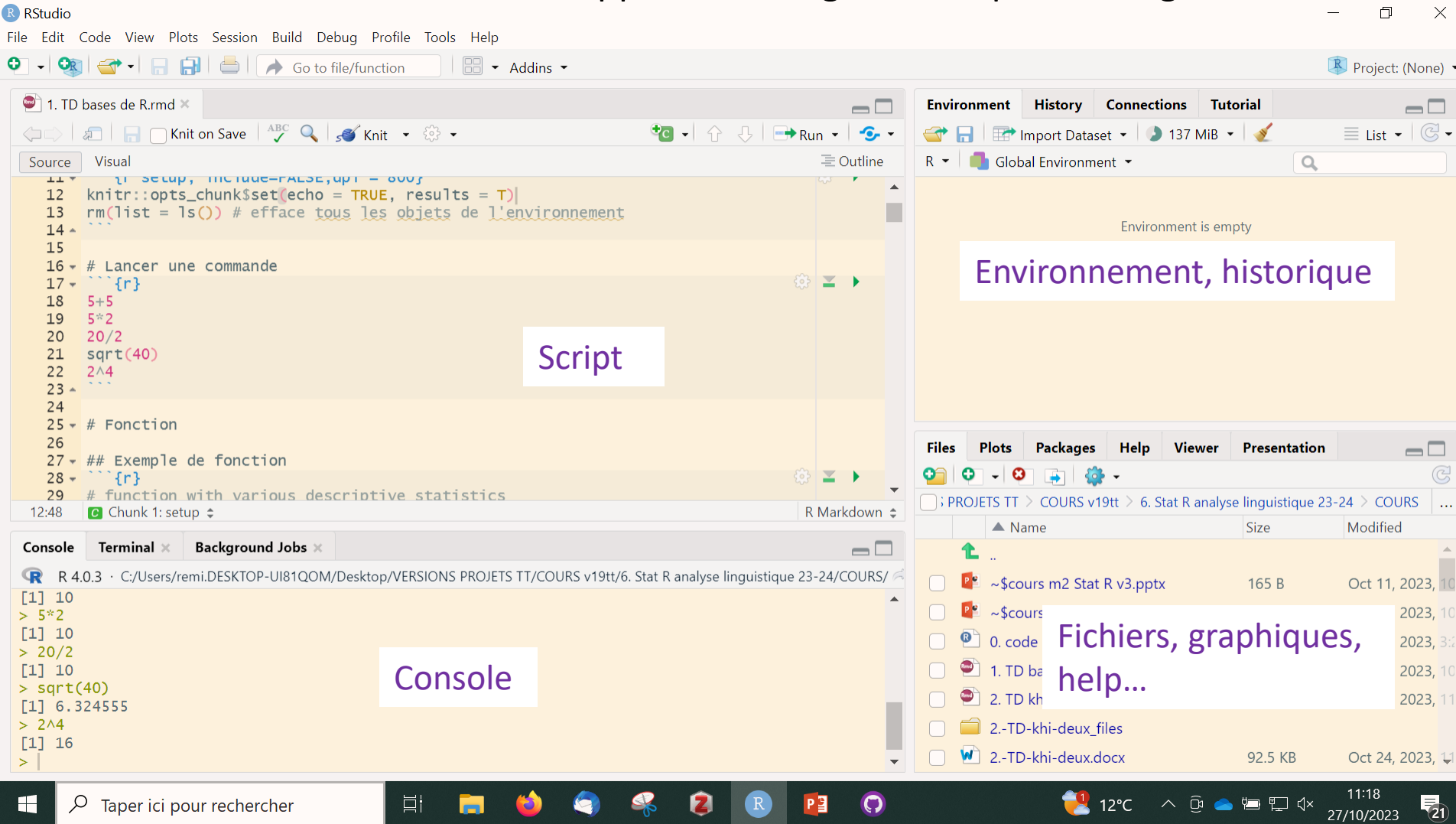
R est un projet collaboratif avec de nombreux contributeurs.
Tapez 'contributors()' pour plus d'information et
'citation()' pour la façon de le citer dans les publications.

Tapez 'demo()' pour des démonstrations, 'help()' pour l'aide
en ligne ou 'help.start()' pour obtenir l'aide au format HTML.
Tapez 'q()' pour quitter R.

> 2+2
[1] 4
> 2 + 2
[1] 4
> 2 + 2
[1] 4
> blabla
Erreur : objet 'blabla' introuvable
> x = 2 + 2
> x
[1] 4
> |
```

Affichage du résultat

- EDI : environnement de développement intégré → simplifie l'usage de R



The screenshot shows the RStudio IDE interface. The main editor window displays a script file named "1. TD bases de R.rmd". The script contains R code for setting up a chunk, clearing the environment, and calculating some values. The console window at the bottom shows the output of the code. The Environment pane on the right shows that the environment is empty. The Files pane at the bottom right shows a list of files in the current project.

Script

```
11 # knitr::opts_chunk$set(echo = TRUE, results = T)
12 knitr::opts_chunk$set(echo = TRUE, results = T)
13 rm(list = ls()) # efface tous les objets de l'environnement
14
15
16 # Lancer une commande
17 {r}
18 5+5
19 5*2
20 20/2
21 sqrt(40)
22 2^4
23
24
25 # Fonction
26
27 ## Exemple de fonction
28 {r}
29 # function with various descriptive statistics
```

Console

```
R 4.0.3 · C:/Users/remi.DESKTOP-UI81QOM/Desktop/VERSIONS PROJETS TT/COURS v19tt/6. Stat R analyse linguistique 23-24/COURS/
[1] 10
> 5*2
[1] 10
> 20/2
[1] 10
> sqrt(40)
[1] 6.324555
> 2^4
[1] 16
>
```

Environment, historique

Environment is empty

Fichiers, graphiques, help...

| Name | Size | Modified |
|----------------------------|---------|------------------|
| .. | | |
| ~\$cours m2 Stat R v3.pptx | 165 B | Oct 11, 2023, 10 |
| ~\$cours | | 2023, 10 |
| 0. code | | 2023, 3:2 |
| 1. TD bases de R | | 2023, 10 |
| 2. TD khi-deux | | 2023, 11 |
| 2.-TD-khi-deux_files | | |
| 2.-TD-khi-deux.docx | 92.5 KB | Oct 24, 2023, 11 |

Lancer une commande

- Exemples de commandes simples :

`5+5`

`5*2`

`20/2`

`sqrt(40)`

`2^4`

- placer le curseur sur la ligne
- Ctrl + Entrée dans windows
- Pour Mac, aucun idée : chercher le raccourci dans onglet « Code »



Fonctions

- Commandes qui exécutent des procédures



https://stt4230.rbind.io/programmation/fonctions_r/

- D'où viennent les fonctions ?
 - De librairies (*packages*) directement installées dans R : fonctions par défaut
 - De librairies créées par des tiers, qu'il nous faudra télécharger manuellement
 - De nous-mêmes : possibilité de créer ses propres fonctions

`ls("package:...nom du paquet... ")` affiche toutes les fonctions du paquet dans la console

```
> ls("package:graphics")
[1] "abline"      "arrows"      "assocplot"    "axis"         "Axis"         "axis.Date"    "axis.POSIXct"  "axTicks"
[9] "barplot"     "barplot.default" "box"          "boxplot"      "boxplot.default" "boxplot.matrix" "bxp"           "cdplot"
[17] "clip"        "close.screen" "co.intervals" "contour"       "contour.default" "coplot"        "curve"          "dotchart"
[25] "erase.screen" "filled.contour" "fourfoldplot" "frame"         "grconvertX"    "grconvertY"   "grid"           "hist"
[33] "hist.default" "identify"      "image"        "image.default" "layout"        "layout.show"  "lcm"            "legend"
[41] "lines"       "lines.default" "locator"      "matlines"     "matplot"       "matpoints"    "mosaicplot"     "mtext"
[49] "pairs"       "pairs.default" "panel.smooth" "par"          "persp"         "pie"          "plot"           "plot.default"
[57] "plot.design" "plot.function" "plot.new"     "plot.window"  "plot.xy"       "points"       "points.default" "polygon"
[65] "polypath"    "rasterImage"  "rect"        "rug"          "screen"        "segments"     "smoothScatter"  "spineplot"
[73] "split.screen" "stars"        "stem"        "strheight"    "stripchart"    "strwidth"     "sunflowerplot"  "symbols"
[81] "text"        "text.default" "title"       "xinch"        "xspline"       "xyinch"       "yinch"
```

`"graphics::"` affiche les fonctions dans l'ordre alphabétique dans une liste



function with various descriptive statistics

```
fxdescribe = function(x){  
  c(obs      = length(x),  
    missing  = sum(is.na(x),na.rm=T),  
    min      = min(x,na.rm=T),  
    max      = max(x,na.rm=T),  
    median   = median(x,na.rm=T),  
    q1       = quantile(x,na.rm=T,c(.25)),  
    q3       = quantile(x,na.rm=T,c(.75)),  
    mean     = mean(x,na.rm=T),  
    sd       = sd(x,na.rm=T),  
    `95lci`  = mean(x,na.rm=T)-(sd(x,na.rm = T)*1.96/sqrt(length(x))),  
    `95hci`  = mean(x,na.rm=T)+(sd(x,na.rm = T)*1.96/sqrt(length(x)))  
  )  
}  
# e.g. =  
# fxdescribe(c(NA,NA,2,5,6))
```

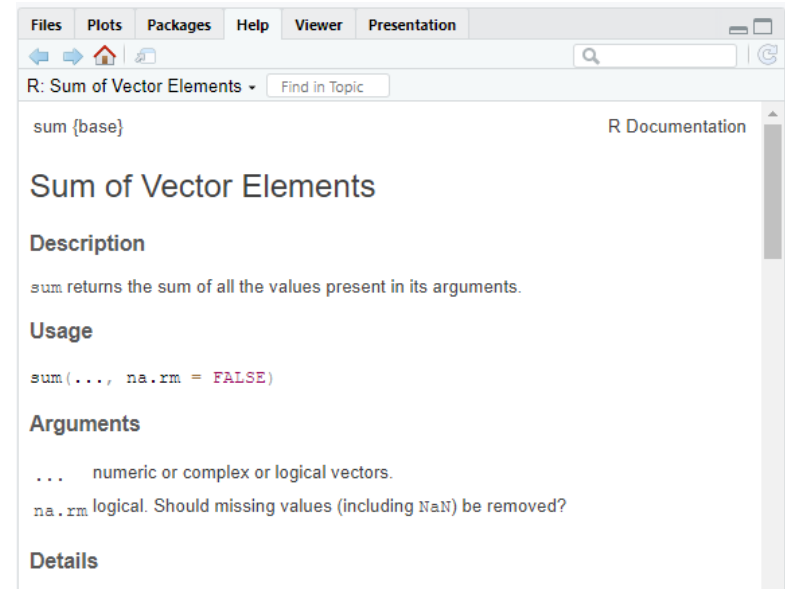
NB : penser au raccourci `Ctrl + c` pour # créer des commentaires

Exercice 1

- Exécutez et essayez de comprendre ce que font les fonctions suivantes :

```
seq(from = 0, to = 100, by = 20)
c(1:10)*2
rep(x = c(1,2,3), times = 3)
rep(x = c(1,2,3), times = 3, each = 3)
sum(1:3)
sum(is.na(c(NA,NA,1,NA)))
plot(x = c(1,2,3), y = c(4,5,6))
boxplot(1:100)
summary(c(1,8,9,7))
```

- Pour connaître les arguments d'une fonction, tapez par ex : **?sum** (le mieux) ou **args(sum)**

The screenshot shows the R documentation window for the `sum` function. The title bar includes tabs for Files, Plots, Packages, Help, Viewer, and Presentation. The main content area is titled "R: Sum of Vector Elements" and includes a search bar. The text "sum {base}" is at the top right. The "Description" section states: "sum returns the sum of all the values present in its arguments." The "Usage" section shows: `sum(..., na.rm = FALSE)`. The "Arguments" section lists: "... numeric or complex or logical vectors." and "na.rm logical. Should missing values (including NA) be removed?". The "Details" section is partially visible at the bottom.

Installer et charger un paquet

- Deux commandes à connaître par coeur :

```
install.packages("psych") # installe le paquet  
library("psych")          # charge le paquet
```

- **Exemples de paquets utiles :**

- **tidyverse** # pour manipuler des données
- **psych, DescTools, rstatix** # pour les statistiques
- **ggplot2** # pour les graphiques
- **readxl, writexl** # pour importer/exporter des fichiers excel



Installer et charger un paquet

Files

Plots

Packages

Help

Viewer

Presentation

Install

Update

| | Name | Description | Version | | |
|--------------------------|-------------|--|---------|-------------|-------------|
| User Library | | | | | |
| <input type="checkbox"/> | abe | Augmented Backward Elimination | 3.0.1 | <div></div> | <div></div> |
| <input type="checkbox"/> | abind | Combine Multidimensional Arrays | 1.4-5 | <div></div> | <div></div> |
| <input type="checkbox"/> | adaptMCMC | Implementation of a Generic Adaptive Monte Carlo Markov Chain Sampler | 1.4 | <div></div> | <div></div> |
| <input type="checkbox"/> | ade4 | Analysis of Ecological Data: Exploratory and Euclidean Methods in Environmental Sciences | 1.7-19 | <div></div> | <div></div> |
| <input type="checkbox"/> | adegraphics | An S4 Lattice-Based Package for the Representation of Multivariate Data | 1.0-18 | <div></div> | <div></div> |
| <input type="checkbox"/> | afex | Analysis of Factorial Experiments | 1.0-1 | <div></div> | <div></div> |
| <input type="checkbox"/> | AICcmodavg | Model Selection and Multimodel | 2.3-1 | <div></div> | <div></div> |

Création d'objets

- **Quelques règles :**

- Utilisation du "<-" (recommandé) ou du "=" (moins recommandé)

Voir https://stt4230.rbind.io/amelioration_code/bonnes_pratiques_r/

- R est sensible aux **majuscules/minuscules**
- Eviter les **accents**
- Un objet ne peut pas démarrer par un **chiffre**
- **Noms interdits** : *if, else, repeat, while, function, for, in, next, break, TRUE, FALSE, NULL, Inf, NaN, NA, NA_integer_, NA_real_, NA_complex_, NA_character_, ... et ..1, ..2, etc. »*

Voir https://stt4230.rbind.io/introduction/base_r/

```
objet <- "objet"
objet = 4
1objet <- 4 # erreur

## Error: <text>:3:2: symbole inattendu
```


Création d'objets

- Exemples d'objets :

```
vache    <- c("spassky", "karpov", "kasparov", "topalov")
couleur  <- c("noire", "noire", "marron", "blanche")
poids    <- c(900,600,700,650)
# "c()" signifie "concaténer"
```

- Possibilité d'appliquer des fonctions sur ces objets :

```
table(couleur)
```

```
## couleur
## blanche marron  noire
##          1      1      2
```

```
summary(poids)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    600.0   637.5   675.0   712.5   750.0   900.0
```

Type d'objets

- Objet simple
- Vecteur
- Facteur
- Matrice
- Dataframe (jeu de données)
- Liste

Objet simple

- On assigne quelque chose dans autre chose

```
objet1 <- 10
objet2 <- "Michel est dans le garage"

objet1
## [1] 10

objet2
## [1] "Michel est dans le garage"
```

Objet simple

- Plusieurs façons d'afficher un même objet

```
objet1 <- 10  
objet1
```

```
objet1 <- 10  
print(objet1)
```

```
objet1 <- 10 ; objet1
```

```
(objet1 <- 10)
```

Vecteur

- Objet qui rassemble des données (variable) **de même nature**

```
vache    <- c("spassky", "karpov", "kasparov", "topalov")
str(vache)
# characters (lettres)

poids    <- c(900, 600, 700, 650)
str(poids)
# numeric (chiffres)
```



str() peut être appliqué à tout type d'objet

Vecteur

- Les vecteurs de nature différente ne se mélangent pas très bien...

```
# Tentative de création d'un vecteur "hybride"
vache_poids <- c(vache, poids)
str(vache_poids)

## chr [1:8] "spassky" "karpov" "kasparov" "topalov" "900" "600" "700" "650"

# Echec
```

- Possible de changer la nature d'un vecteur

```
(poids <- as.character(poids))
## [1] "900" "600" "700" "650"

(poids <- as.numeric(poids))
## [1] 900 600 700 650
```

Facteur

- Objet qui organise les données d'un vecteur en **catégories ou niveaux**

```
facteur_vache <- factor(vache)
facteur_vache

## [1] spassky karpov kasparov topalov
## Levels: karpov kasparov spassky topalov

# NB : Les niveaux sont rangés automatiquement par ordre alphabétique

# possibilité de ré-ordonner (levels=) et de renommer (labels=) ces niveaux
facteur_vache <- factor(vache,
                        levels = c("spassky", "kasparov", "karpov", "topalov"),
                        labels = c("Spassky", "Kasparov", "Karpov", "Topalov"))

facteur_vache

## [1] Spassky Karpov Kasparov Topalov
## Levels: Spassky Kasparov Karpov Topalov
```

- Les "facteurs" seront très utiles quand nous aborderons le recodage de variables

Matrice

- Objet qui agence les données d'un vecteur en lignes et en colonnes
- Ensemble de vecteurs (variables) **de même nature**

```
vache    <- c("spassky", "karpov", "kasparov", "topalov")
couleur  <- c("noire", "noire", "marron", "blanche")
```

*# on combine les deux vecteurs avec la fonction **cbind()** qui veut dire :
column bind (combiner colonnes)*

```
MAT <- cbind(vache,couleur)
```

```
MAT
```

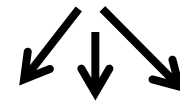
```
##      vache      couleur
## [1,] "spassky" "noire"
## [2,] "karpov"  "noire"
## [3,] "kasparov" "marron"
## [4,] "topalov" "blanche"
```


Dataframe (jeu de données)

- Matrice qui regroupe plusieurs vecteurs
- Ensemble de vecteurs (variables) **de nature différente**

```
DF <- data.frame(vache,couleur,poids)  
DF
```

colonne = variable



| | vache | couleur | poids |
|------|----------|---------|-------|
| ## 1 | spassky | noire | 900 |
| ## 2 | karpov | noire | 600 |
| ## 3 | kasparov | marron | 700 |
| ## 4 | topalov | blanche | 650 |

- En pratique, nous importons quasiment toujours un dataframe (DF) à partir d'un fichier xlsx, txt, ou csv

Liste

- Objet pouvant **contenir** des objets de natures différentes : vecteurs, matrices, dataframes ...

```
LS <- list(vache, MAT, DF)
LS
```

- Très pratique pour stocker temporairement des objets, notamment via des boucles

EXERCICE MAISON pour les plus téméraires : exécutez et essayer de comprendre cette simple boucle

```
LS <- list() # liste vide
for (VACHE in DF[, "vache"]) {
  LS[[VACHE]] <- DF[DF$vache == VACHE,]
}
LS # liste remplie !
```

```
## [[1]]
## [1] "spassky" "karpov"
## "kasparov" "topalov"
##
## [[2]]
##      vache      couleur
## [1,] "spassky"  "noire"
## [2,] "karpov"   "noire"
## [3,] "kasparov" "marron"
## [4,] "topalov"  "blanche"
##
## [[3]]
##      vache couleur poids
## 1  spassky  noire   900
## 2   karpov  noire   600
## 3 kasparov marron   700
## 4  topalov blanche  650
```

Exercice 2

- Créez un vecteur de type caractère contenant 4 noms d'animaux
- Créez un vecteur de type caractère contenant 4 pays
- Créez un vecteur de type numérique contenant 4 tailles
- Créez une matrice regroupant les vecteurs "animaux" et "pays"
- Transformez le vecteur "animaux" en facteur en changeant l'ordre des niveaux
- Créez une dataframe regroupant les vecteurs "pays" et "poids" et le facteur "animaux"
- Insérez le facteur "animaux" et le dataframe dans une même liste

Importation de données

Téléchargez tous les jeux de données (.csv et .xlsx) sur
<https://github.com/lafitter/M2-Science-du-Langage>

Le DF **metal_bands**

| | A | B | C | D |
|----|-------------|-------|------------|-----------|
| 1 | Territory | Bands | Population | Happiness |
| 2 | Afghanistan | 2 | 37466414 | 2.404 |
| 3 | Albania | 7 | 3088385 | 5.199 |
| 4 | Algeria | 16 | 43576691 | 5.122 |
| 5 | Andorra | 2 | 85645 | |
| 6 | Angola | 8 | 33642646 | |
| 7 | Argentina | 1907 | 45864941 | 5.967 |
| 8 | Armenia | 19 | 3011609 | 5.399 |
| 9 | Australia | 1545 | 25809973 | 7.162 |
| 10 | Austria | 664 | 8884864 | 7.163 |
| 11 | Azerbaijan | 9 | 10282283 | 5.173 |

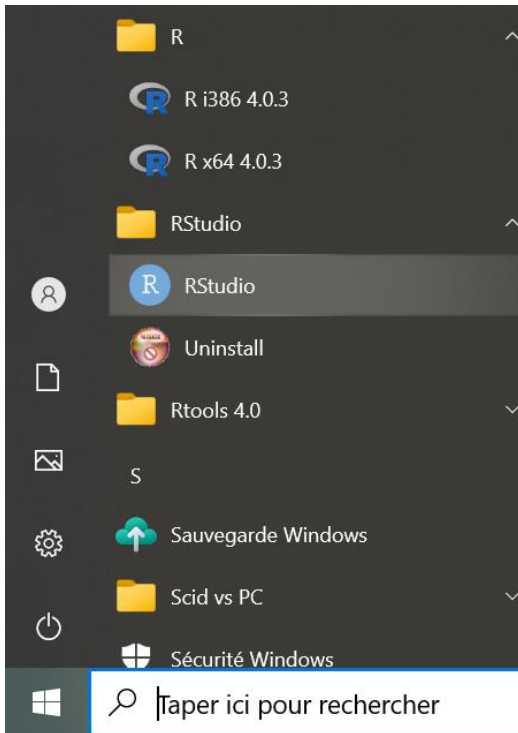
- Territory : pays
- Bands : nombre de groupes de métal
- Population : nombre d'habitant
- Happiness : échelle de bonheur
https://en.wikipedia.org/wiki/World_Happiness_Report



<https://www.brooklynvegan.com/study-says-coun/>
<https://p.migdal.pl/blog/2023/01/metal-bands-happiness-correlation/>

Adresse du dataframe





- Importer un DF nécessite de connaître son **répertoire de travail** ("l'adresse" du script R)
- **Cas 1** : R a été ouvert via le menu application (windows)



```
getwd() # get working directory  
"C:/Users/remi.../Documents"
```

Adresse du dataframe

- Nécessite de connaître son **répertoire de travail** ("l'adresse" du script R)
- **Cas 2** : R a été ouvert via un script R

| Nom | Modifié le | Type | Taille |
|--|------------------|-----------------------|----------|
|  cours m2 Stat R v3 | 11/10/2023 15:48 | Présentation Micr... | 4 749 Ko |
|  TD1_bases_R | 11/10/2023 15:40 | Fichier RMD | 4 Ko |
|  ReadingSkills | 11/10/2023 15:38 | Fichier CSV Micro... | 1 Ko |
|  ReadingSkills | 11/10/2023 15:36 | Feuille de calcul ... | 6 Ko |

```
getwd()
```

```
## [1] "C:/Users/remi.../Desktop/.../6. Stat R analyse linguistique 23-24  
/COURS"
```



Je conseille **le cas 2** car le DF et le script R ont la même adresse

Importation via le script

```
DF <- read.csv("metal_bands.csv", header = T, sep = ";", dec = ".")
```

↑
 fonction de base
pour lire les
fichiers .csv

↑
 nom du fichier
 ↓
 implicitement, l'adresse du fichier est :
 "C:/Users/.../6. Stat R analyse linguistique
 23-24/COURS/metal_bands.csv"

↑
 indique que la 1ère
ligne correspond au
nom des variables

↑
 séparateur
de colonne

↑
 type de
décimale



Ouvrez le fichier .csv avec bloc-notes pour identifier les bons séparateurs :

```
Territory;Bands;Population;Happiness
Afghanistan;2;37466414;2.404
Albania;7;3088385;5.199
Algeria;16;43576691;5.122
```

Importation via le script

- Vérifiez ensuite que le DF a été correctement importé avec **head()** et **str()**

```
head(DF, n = 4) # affiche les 4 premières lignes
```

```
##      Territory Bands Population Happiness
## 1 Afghanistan     2   37466414      2.404
## 2    Albania     7    3088385      5.199
## 3    Algeria    16   43576691      5.122
## 4    Andorra     2     85645      NA
```

```
str(DF) # TOUJOURS VERIFIER LE FORMAT DES VARIABLES !
```

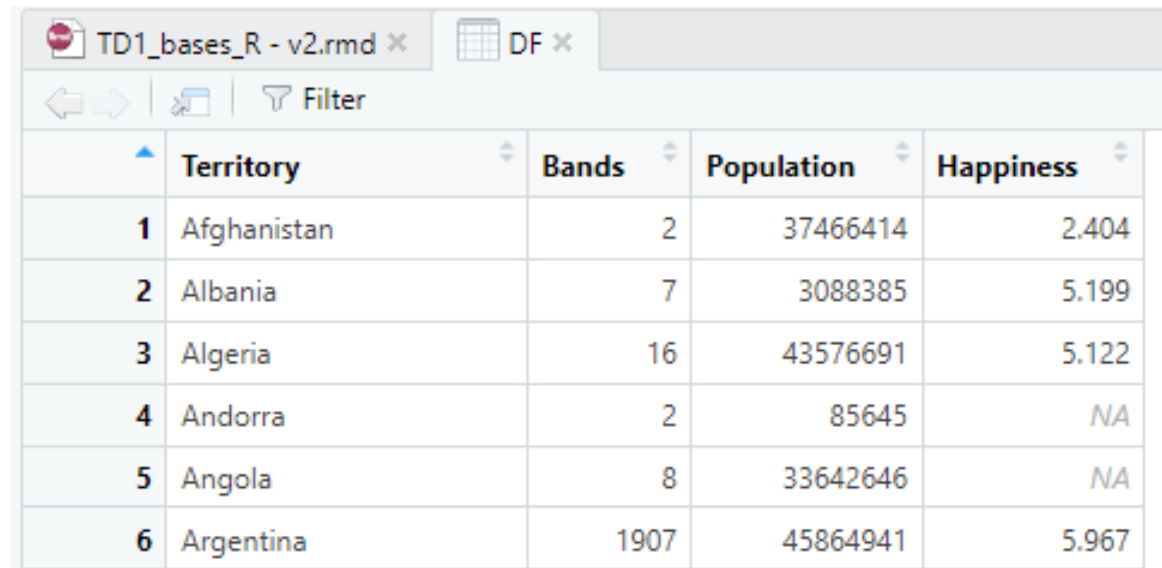
```
## 'data.frame':    174 obs. of  4 variables: # affiche 4 vecteurs
## $ Territory : chr  "Afghanistan" "Albania" "Algeria" "Andorra" ...
## $ Bands      : int   2 7 16 2 8 1907 19 1545 664 9 ...
## $ Population: int  37466414 3088385 43576691 85645 33642646 45864941 3011609
##              25809973 8884864 10282283 ...
## $ Happiness : num  2.4 5.2 5.12 NA NA ...
```

quand le parameter "dec = ..." est mal spécifié, on peut avoir des erreurs de formats ici

Importation via le script

- Utilisez **View()** pour visualiser l'ensemble du DF

View(DF)



| | Territory | Bands | Population | Happiness |
|---|-------------|-------|------------|-----------|
| 1 | Afghanistan | 2 | 37466414 | 2.404 |
| 2 | Albania | 7 | 3088385 | 5.199 |
| 3 | Algeria | 16 | 43576691 | 5.122 |
| 4 | Andorra | 2 | 85645 | NA |
| 5 | Angola | 8 | 33642646 | NA |
| 6 | Argentina | 1907 | 45864941 | 5.967 |

Importation via le script

- Utilisez **summary()** pour visualiser les caractéristiques de chaque variable, ainsi que le **nombre de données manquantes indiquées par "NA"**

summary(DF)

| ## Territory | Bands | Population | Happiness |
|---------------------|-----------------|-------------------|-----------------|
| ## Length:174 | Min. : 1.0 | Min. :5.321e+03 | Min. :2.404 |
| ## Class :character | 1st Qu.: 7.0 | 1st Qu.:2.712e+06 | 1st Qu.:4.889 |
| ## Mode :character | Median : 38.0 | Median :8.885e+06 | Median :5.569 |
| ## | Mean : 523.9 | Mean :4.681e+07 | Mean :5.554 |
| ## | 3rd Qu.: 285.0 | 3rd Qu.:3.364e+07 | 3rd Qu.:6.305 |
| ## | Max. :17557.0 | Max. :1.398e+09 | Max. :7.821 |
| ## | NA's :29 | NA's :29 | NA's :28 |

- NA = not applicable, not available, not assessed**

Importation via le script

- Si votre DF est de type .xlsx, plus pratique encore de l'importer avec la **library(readxl)**

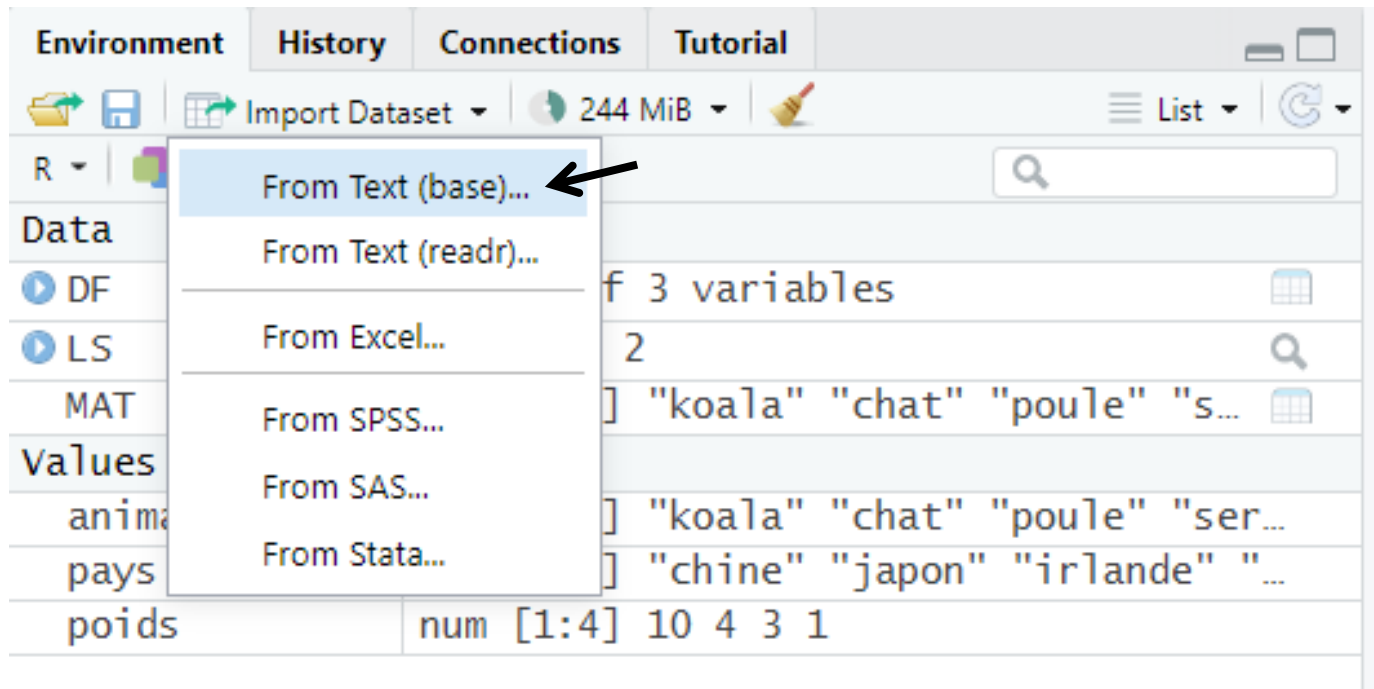
| | A | B | C | D |
|---|-------------|-------|------------|-----------|
| 1 | Territory | Bands | Population | Happiness |
| 2 | Afghanistan | 2 | 37466414 | 2,404 |
| 3 | Albania | 7 | 3088385 | 5,199 |
| 4 | Algeria | 16 | 43576691 | 5,122 |
| 5 | Andorra | 2 | 85645 | |
| 6 | Angola | 8 | 33642646 | |
| 7 | Argentina | 1907 | 45864941 | 5,967 |
| 8 | Armenia | 19 | 3011609 | 5,399 |

← NB : dans excel les décimales sont des virgules

```
# install.packages("readxl")  
library(readxl)  
DF <- readxl::read_xlsx("metal_bands.xlsx")  
head(DF)
```

↑ pas la peine de spécifier "sep = ..." et "dec = ..." ici

Importation clique-bouton



Importation clique-bouton

Import Dataset

Name:

Input File:

Encoding:

Heading: ☒ Yes ☐ No

Row names:

Separator:

Decimal:

Quote:

Comment:

na.strings:

☐ Strings as factors

Data Frame

| Territory | Bands | Population | Happiness |
|-------------|-------|------------|-----------|
| Afghanistan | 2 | 37466414 | 2.404 |
| Albania | 7 | 3088385 | 5.199 |
| Algeria | 16 | 43576691 | 5.122 |
| Andorra | 2 | 85645 | NA |
| Angola | 8 | 33642646 | NA |
| Argentina | 1907 | 45864941 | 5.967 |
| Armenia | 19 | 3011609 | 5.399 |
| Australia | 1545 | 25809973 | 7.162 |
| Austria | 664 | 8884864 | 7.163 |
| Azerbaijan | 9 | 10282283 | 5.173 |
| Bahrain | 6 | 1526929 | 6.647 |
| Bangladesh | 65 | 164098818 | 5.155 |
| Barbados | 3 | 301865 | NA |
| Belarus | 293 | 9441842 | 5.821 |
| Belgium | 666 | 11778842 | 6.805 |
| Belize | 1 | 405633 | NA |

Import Cancel

← prévisualisation

Exercice 3

- Importez et inspectez les fichiers "reading_skills" numérotés de 1 à 3, disponibles sur <https://github.com/lafitter/M2-Science-du-Langage>
- Attention aux séparateurs et décimales...

Importation via le script

- Essayons d'importer reading_skills4...
- Cas particulier mais fréquent : quand une variable contient des chiffres et des lettres

```
DF = read.csv("reading_skills4.csv", sep=";")
head(DF);str(DF)

## 'data.frame':    44 obs. of  5 variables:
## $ sujet      : int  1 2 3 4 5 6 7 8 9 10 ...
## $ age        : int  69 67 43 18 55 57 62 50 79 69 ...
## $ accuracy: num  0.884 0.765 0.915 0.984 0.884 ...
## $ dyslexia: chr  "no" "no" "no" "no" ...
## $ iq         : chr  "no data" "0.59" NA "1.144" ...
```

R pense logiquement que *iq* est un vecteur de type caractère à cause de cette valeur

en revanche R "sait" par défaut que "NA" doit être interprété comme "donnée manquante"

Importation via le script

- Cas particulier mais fréquent : quand une variable contient des chiffres et des lettres

```
DF = read.csv("reading_skills4.csv", sep=";", na.strings = c("NA", "no data"))  
head(DF); str(DF)
```

```
## 'data.frame':    44 obs. of  5 variables:  
## $ sujet      : int  1 2 3 4 5 6 7 8 9 10 ...  
## $ age        : int  69 67 43 18 55 57 62 50 79 69 ...  
## $ accuracy: num  0.884 0.765 0.915 0.984 0.884 ...  
## $ dyslexia: chr  "no" "no" "no" "no" ...  
## $ iq         : num  NA 0.59 NA 1.144 -0.676 ...
```

maintenant R sait que "no data" = "NA"

Objet et environnement

Environment

History

Connections

Tutorial

Import Dataset

253 MiB

R

Global Environment

List

Data

DF

44 obs. of 5 variables

LS

List of 4

MAT

chr [1:8] "spassky" "karpov" "kasparov" "topalov" "noire" "noire" "marro..."

Values

couleur

chr [1:4] "noire" "noire" "marron" "blanche"

facteur_vache

Factor w/ 4 levels "Spassky","Kasparov",...: 1 3 2 4

objet

4

objet1

10

objet2

"Michel est dans le garage"

poids

num [1:4] 900 600 700 650

vache

chr [1:4] "spassky" "karpov" "kasparov" "topalov"

VACHE

"topalov"

vache_poids

chr [1:8] "spassky" "karpov" "kasparov" "topalov" "900" "600" "700" "650"

Functions

fxdescribe

function (x)

Objet et environnement

```
rm(objet1)           # supprime un objet
ls()                 # liste des objets

## [1] "couleur"      "DF"           "facteur_vache" "fxdescribe"
## [5] "LS"           "MAT"          "objet2"        "poids"
## [9] "vache"        "VACHE"        "vache_poids"

ls(".GlobalEnv")     # liste des objets (idem)

## [1] "couleur"      "DF"           "facteur_vache" "fxdescribe"
## [5] "LS"           "MAT"          "objet2"        "poids"
## [9] "vache"        "VACHE"        "vache_poids"

search()             # montre environnement et paquets

## [1] ".GlobalEnv"    "package:readxl" "package:stats"
## [4] "package:graphics" "package:grDevices" "package:utils"
## [7] "package:datasets" "package:methods" "Autoloads"
## [10] "package:base"
```

Objet et environnement

- Question cruciale : comment accéder à un vecteur (variable) situé à l'*intérieur* d'un DF ?



```
DF <- read_xlsx("metal_bands.xlsx")
Bands          # ne fonctionne pas...
```

```
## Error in eval(expr, envir, enclos): objet 'Bands' introuvable
```

→ "Bands" **n'existe pas** dans l'environnement en tant *qu'objet*

```
ls()          # liste des objets
```

| | | | | | |
|----|-----|-----------|---------|-----------------|--------------|
| ## | [1] | "couleur" | "DF" | "facteur_vache" | "fxdescribe" |
| ## | [5] | "LS" | "MAT" | "objet2" | "poids" |
| ## | [9] | "vache" | "VACHE" | "vache_poids" | |

Objet et environnement

- Question cruciale : comment accéder à un vecteur (variable) situé à l'*intérieur* d'un DF ?



```
DF <- read_xlsx("metal_bands.xlsx")  
Bands          # ne fonctionne pas...  
  
## Error in eval(expr, envir, enclos): objet 'Bands' introuvable
```

→ "Bands" **n'existe pas** dans l'environnement en tant *qu'objet*

```
DF$Bands      # fonctionne !
```

- grâce au "\$", R reconnaît maintenant "Bands" comme un objet

Objet et environnement

- Il existe un 2^{ème} moyen d'accéder à "Bands", consistant à transformer le DF en environnement
- Combo **attach()/detach()**

```
attach(DF) # transforme le DF en environnement  
search()
```

```
## [1] ".GlobalEnv"      "DF"               "package:readxl"  
## [4] "package:stats"   "package:graphics" "package:grDevices"  
## [7] "package:utils"   "package:datasets" "package:methods"  
## [10] "Autoloads"       "package:base"
```

```
Bands
```

```
detach(DF) # enlève le DF de l'environnement
```

Objet et environnement

- Attention ! La solution attach/detach peut créer des conflits
 - Imaginons qu'un objet crée précédemment s'appelle déjà "Bands"
 - Dans ce cas attach() ne fonctionne plus

```
Bands = c("megadeath", "sepultura", "mylene farmer")
attach(DF)

Bands # pas vraiment le résultat attendu

## [1] "megadeath"      "sepultura"      "mylene farmer"

detach(DF)

rm(Bands) # maintenant ça devrait marcher
```



Sur ce sujet voir <https://www.uvm.edu/~statdhtx/StatPages/R/attaching.html>

Opérateurs

Opérateurs de commandes

| | |
|-------------------------|--------------------------|
| Séparateur de commandes | x = 12 ; x |
| Texte | "blabla" |
| Commentaire | # très beau code |
| Assignation | x <- 12 ou x = 12 |
| Extraction d'objet | DF\$Bands DF["Bands"] |

Opérateurs

Opérateurs arithmétiques

| | |
|--------------------------------------|---|
| addition/soustraction/multiplication | $(2+2-1)*4$ |
| division | $10/2$ |
| puissance | 2^2 |
| racine carée | <code>sqrt(4)</code> |
| exponentiel/logarithme | <code>exp(2)</code> ; <code>log(6)</code> |

Opérateurs

Opérateurs logiques et booléens

| | |
|---------------------------------|----------------------------|
| égal à | <code>==</code> |
| différent de | <code>!=</code> |
| strictement inférieur/supérieur | <code></></code> |
| inférieur/supérieur ou égal à | <code><= / >=</code> |
| ET en même temps | <code>&</code> |
| OU alors | <code> </code> |
| Non, sauf | <code>!</code> |

→ Seront particulièrement utiles pour **filtrer** les données

Adressage

- Adressage : **moyen de localiser les données**
- Si vecteur, coordonnées en 1 dimension :

```
vache[1]  
## [1] "spassky"
```

Adressage

- Adressage : **moyen de localiser les données**
- Si DF, coordonnées en 2 dimensions :

DF[**LIGNE**, **COLONNE**]

| ## | Territory | Bands | Population | Happiness |
|------|-------------|-------|------------|-----------|
| ## 1 | Afghanistan | 2 | 37466414 | 2.404 |
| ## 2 | Albania | 7 | 3088385 | 5.199 |
| ## 3 | Algeria | 16 | 43576691 | 5.122 |
| ## 4 | Andorra | 2 | 85645 | NA |

Adressage

- Adressage : **moyen de localiser les données**
- Si DF, coordonnées en 2 dimensions :

```
DF[2,] # extrait 2ème ligne
```

```
## Territory Bands Population Happiness  
## 2 Albania 7 3088385 5.199
```

```
DF[,2] # extrait 2ème colonne sous forme de vecteur
```

```
## [1] 2 7 16 2 8 1907 19 1545 664  
9 6 65 ...
```

```
DF[2] # extrait 2ème colonne sous forme de DF
```

```
## Bands  
## 1 2  
## 2 7  
## 3 16  
## 4 2
```

Adressage

- Pour l'adressage il est plus pratique d'utiliser des *noms* que des *chiffres*
- Quelques exemples de filtrages simples (sur la base d'une seule variable) :

```
DF[ , c("Bands", "Happiness") ]
DF[ DF$Territory == "France" , "Bands" ]
DF[ DF$Territory %in% c("France", "Uruguay"), ]
DF[ !DF$Territory %in% c("Togo", "Bulgaria"), ]
DF[ row.names(DF) %in% c(5, 87, 142) , ]
DF[ DF$Bands > 1 , ]
```

LIGNE

COLONNE



%in% : vérifie si valeurs du 1er argument (DF\$Territory) présentes dans le 2ème argument c("France", "Uruguay")

Adressage

- Filtrages complexes (sur la base de plusieurs variables) avec les opérateurs "&" et "|" :

```
DF2 <- DF[complete.cases(DF),] # garde uniquement les lignes sans NAA  
attach(DF2) # pour simplifier le code  
  
DF2[Bands < 5 & Population > 10000000, ]  
DF2[Territory %in% c("France", "Germany") | Bands > 2000, ]  
  
detach(DF2)
```

<https://www.mitchcraver.com/2021/06/15/subsetting-and-filtering-a-data-frame-in-r/>

Adressage

- Une fois que l'on a compris les bases de l'adressage, il est aisé de remplacer des données

```
DF2[DF2$Territory == "France", "Happiness"] <- 20
# Les Français passe de 6.7 à 20 points de bonheur

DF$Bands[is.na(DF$Bands)] <- 0
# remplace toutes les valeurs NA de Bands par un 0
```

- Possibilité de ré-ordonner les données en fonction d'une variable

```
head(DF[order(DF$Bands),]) # par ordre croissant de Bands
head(DF[order(-DF$Bands),]) # par ordre décroissant de Bands
```

Création de variables

- Comment modifier une variable du DF, ou rajouter une variable dans le DF ?

```
Happiness_rounded <- round(DF$Happiness,0)
# La fonction round() arrondit les chiffres, ici à 0 chiffres après la virgule

Happiness_rounded
# nous avons créé un vecteur, mais celui-ci est à l'extérieur du DF...

# nous pouvons résoudre ce problème avec cbind...
head(cbind(DF,Happiness_rounded))

# ...néanmoins il y a une solution beaucoup plus simple :
DF$Happiness_rounded <- round(DF$Happiness,0)
```



spécifie le DF dans
lequel on va insérer
la variable



spécifie le nom de
la nouvelle variable



la nouvelle variable
en question

Création de variables

- NB : la nouvelle variable doit avoir le même nombre de lignes que le DF. La seule exception est la possibilité de rajouter une *constante*

```
DF$Happiness_mean <- mean(DF$Happiness, na.rm = T)
head(DF)
```

| ## | Territory | Bands | Population | Happiness | Happiness_rounded | Happiness_mean |
|------|-------------|-------|------------|-----------|-------------------|----------------|
| ## 1 | Afghanistan | 2 | 37466414 | 2.404 | 2 | 5.553575 |
| ## 2 | Albania | 7 | 3088385 | 5.199 | 5 | 5.553575 |
| ## 3 | Algeria | 16 | 43576691 | 5.122 | 5 | 5.553575 |
| ## 4 | Andorra | 2 | 85645 | NA | NA | 5.553575 |
| ## 5 | Angola | 8 | 33642646 | NA | NA | 5.553575 |

mean(DF\$Happiness, na.rm = T)



NB : l'argument **na.rm = T** dans mean() permet de calculer la moyenne en ne prenant pas en compte les NA ; autrement la fonction mean() renverra un ... **NA**.

Création de variables

- La création de variables peut vite devenir "répétitive" en termes de code

Exemple de syntaxe répétitive

```
DF$Happiness_mean    <- mean(DF$Happiness, na.rm = T)
DF$Happiness_median  <- median(DF$Happiness, na.rm = T)
DF$Happiness_var     <- var(DF$Happiness, na.rm = T)
DF$Happiness_sd      <- sd(DF$Happiness, na.rm = T)
```

Création de variables

- Possible d'alléger la syntaxe avec la fonction **mutate()** du paquet "dplyr"

```
# install.packages("dplyr")
library(dplyr)

DF2 <- DF
DF2 <-  
  dplyr::mutate(.data = DF2,  
    Happiness_mean = mean(Happiness, na.rm = T),  
    Happiness_median = median(Happiness, na.rm = T),  
    Happiness_var = var(Happiness, na.rm = T),  
    Happiness_sd = sd(Happiness, na.rm = T),  
    Happiness_z = Happiness_mean/Happiness_sd  
  )
```

nul besoin de re-spécifier DF\$



possible de créer des variables à partir des variables
créées juste avant !

Aparté : une "dplyr-isation" de R croissante

- En pratique la syntaxe de "R de base" tend à être délaissée pour une syntaxe plus de type "dplyr"
- Possibilité de "switcher" progressivement vers la syntaxe dplyr une fois les bases acquises

```
DF[c("Bands", "Happiness")]
```

```
DF[DF$Bands < 5 &  
DF$Population > 10000000,]
```

```
DF[order(DF$Bands),]
```



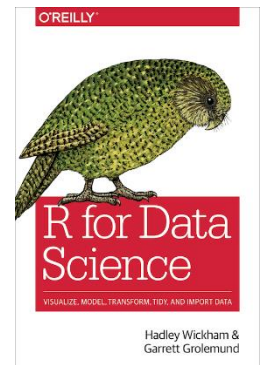
```
dplyr::select(DF, Bands, Happiness)
```

```
dplyr::filter(DF, Bands < 5 &  
Population > 10000000)
```

```
dplyr::arrange(DF, Bands)
```

<https://r4ds.had.co.nz/>

- livre numérique gratuit
- considéré comme un incontournable



Exercice 4

- Ré-importez et inspectez "reading_skills1.csv"
- Filtrez les observations des sujets 1 à 5 et sélectionnez uniquement les variables "sujet" et "dyslexia"
- Filtrez les valeurs de iq > 0 pour les sujets avec dyslexie
- Filtrez les valeurs de iq < 0 ou alors les sujets sans dyslexie
- Une valeur anormale de iq s'est glissée dans les données ; identifiez cette valeur et remplacez la par **-2**
- Créez une nouvelle variable correspondant à la moyenne de "accuracy"