

# Formation au traitement de données avec R

---

# À propos de R

---

R est un langage orienté vers le traitement et l'analyse quantitative de données, dérivé du langage S. Il est développé depuis les années 90 par un groupe de volontaires de différents pays et par une large communauté d'utilisateurs et utilisatrices. C'est un logiciel libre, publié sous [licence GNU GPL](#).

L'utilisation de R présente plusieurs avantages :

- c'est un logiciel multiplateforme, qui fonctionne aussi bien sur des systèmes Linux, Mac OS X ou Windows.
- c'est un logiciel libre, développé par ses utilisateurs et utilisatrices, diffusable et modifiable par tout un chacun.
- c'est un logiciel gratuit.
- c'est un logiciel puissant, dont les fonctionnalités de base peuvent être étendues à l'aide d'extensions développées par la communauté. Il en existe plusieurs milliers.
- c'est un logiciel avec d'excellentes capacités graphiques.

# À propos de R

Comme rien n'est parfait, on peut également trouver quelques inconvénients :

- le logiciel, la documentation de référence et les principales ressources sont en anglais. Il est toutefois parfaitement possible d'utiliser R sans spécialement maîtriser cette langue et il existe de plus en plus de ressources francophones.
- R n'est pas un logiciel au sens classique du terme, mais plutôt un langage de programmation. Il fonctionne à l'aide de scripts (des petits programmes) édités et exécutés au fur et à mesure de l'analyse.
- en tant que langage de programmation, R a la réputation d'être difficile d'accès, notamment pour ceux n'ayant jamais programmé auparavant.

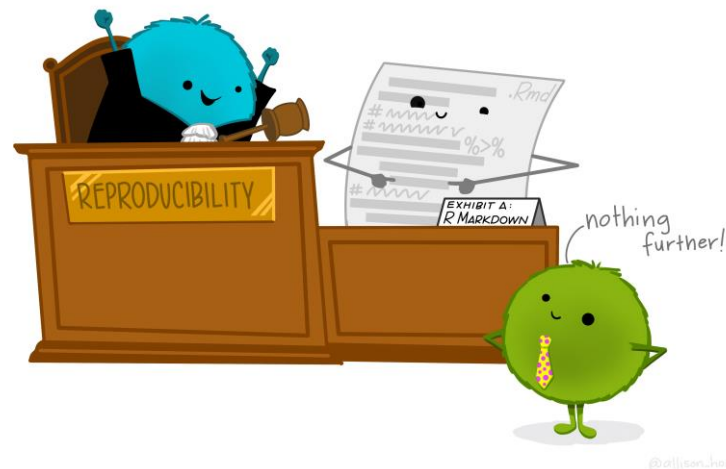




# À propos de R

Le fait de structurer ses analyses sous forme de scripts (suite d'instructions effectuant les différentes opérations d'une analyse) présente de nombreux avantages :

- le script conserve l'ensemble des étapes d'une analyse, de l'importation des données à leur analyse en passant par les manipulations et les recodages.
- on peut à tout moment revenir en arrière et corriger ou modifier ce qui a été fait.
- il est très rapide de réexécuter une suite d'opérations complexes.
- on peut très facilement mettre à jour les résultats en cas de modification des données sources.
- le script garantit, sous certaines conditions, la reproductibilité des résultats obtenus.



@allison\_harel

# À propos de RStudio

---

RStudio n'est pas à proprement parler une interface graphique pour R, il s'agit plutôt d'un *environnement de développement intégré*, qui propose des outils facilitant l'écriture de scripts et l'usage de R au quotidien.

C'est une interface bien supérieure à celles fournies par défaut lorsqu'on installe R sous Windows ou sous Mac.

RStudio est également un logiciel libre et gratuit. Une version payante existe, mais elle ne propose pas de fonctionnalités indispensables.



# Prérequis

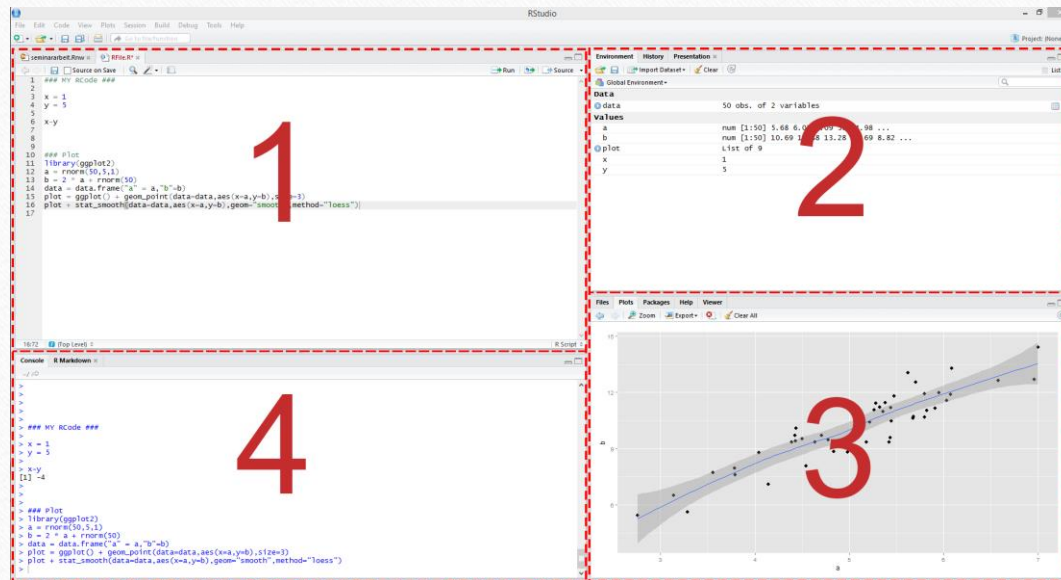
---

Pour installer R, il suffit de se rendre sur une des pages suivantes :

- [Installer R sous Windows](#)
- [Installer R sous Mac](#)

Pour installer RStudio, rendez-vous sur [la page de téléchargement du logiciel](#) et installez la version adaptée à votre système.

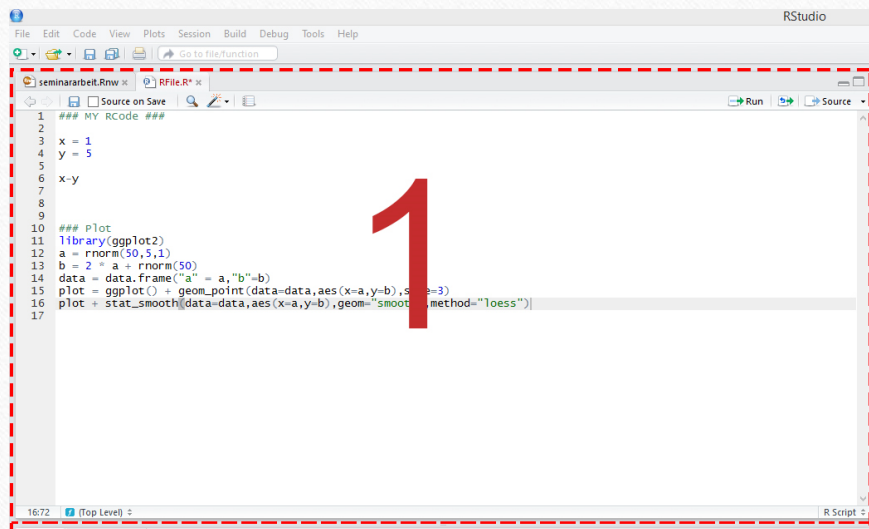
# L'interface Rstudio





# L'interface Rstudio

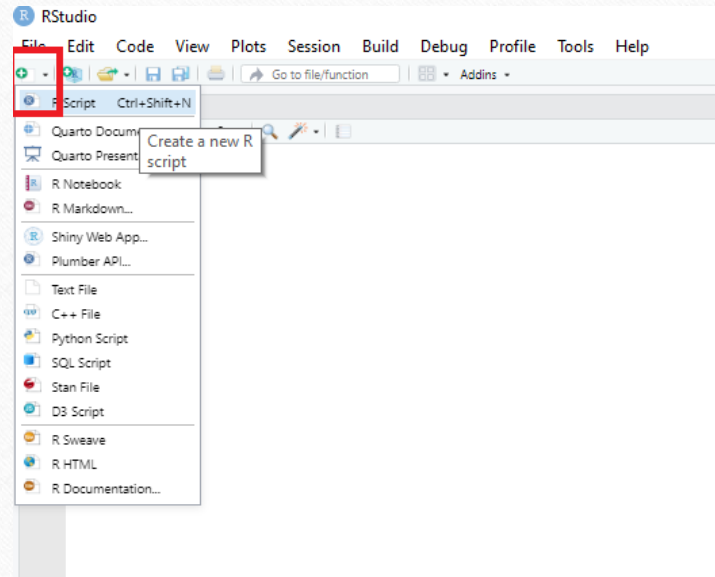
La zone d'éditeur de textes :  
c'est ici que vous écrivez  
vos instructions (les  
« scripts ») de traitements de  
données.





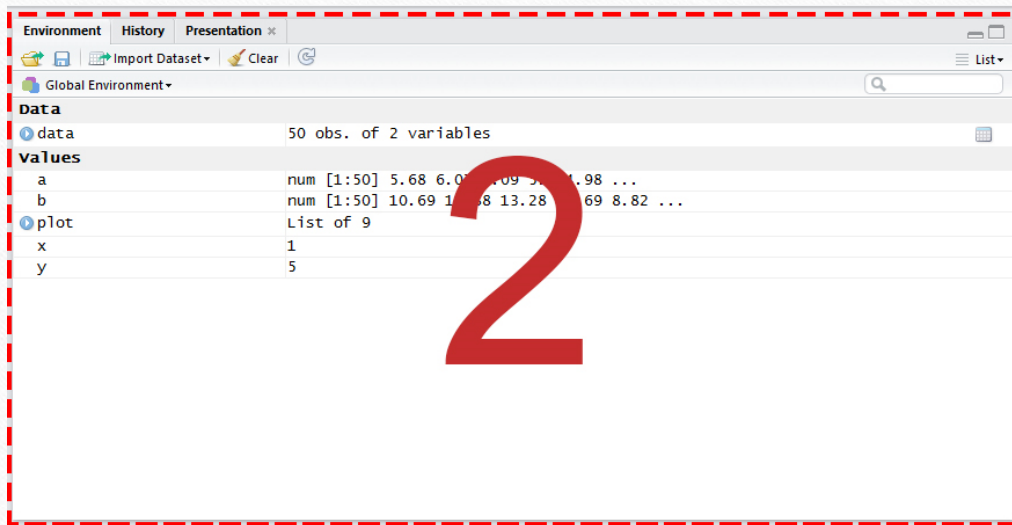
# L'interface Rstudio

- Pour créer un nouveau script, il suffit de cliquer sur « New file » et de choisir « R script ».



# L'interface Rstudio

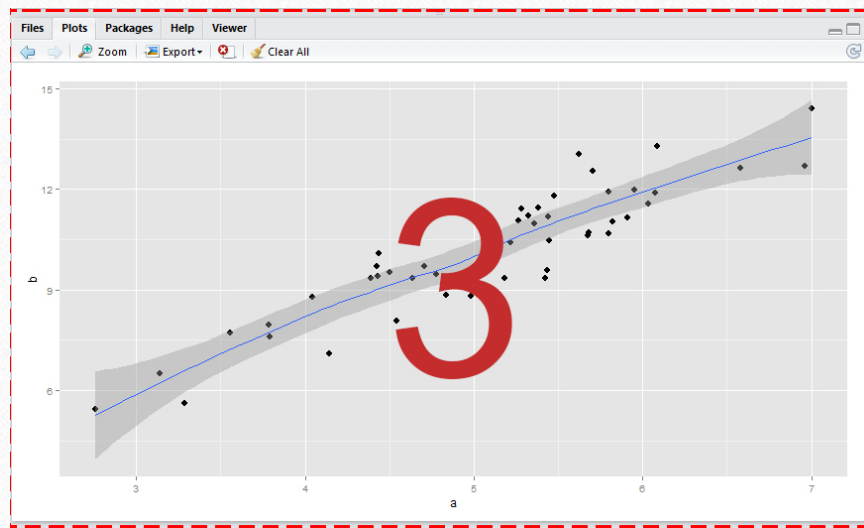
- L'environnement de travail : c'est ici que vous verrez les objets créés lors de votre session.
- Ceux-ci consomment de la mémoire vive, donc faites attention à ne pas trop le remplir !



# L'interface Rstudio

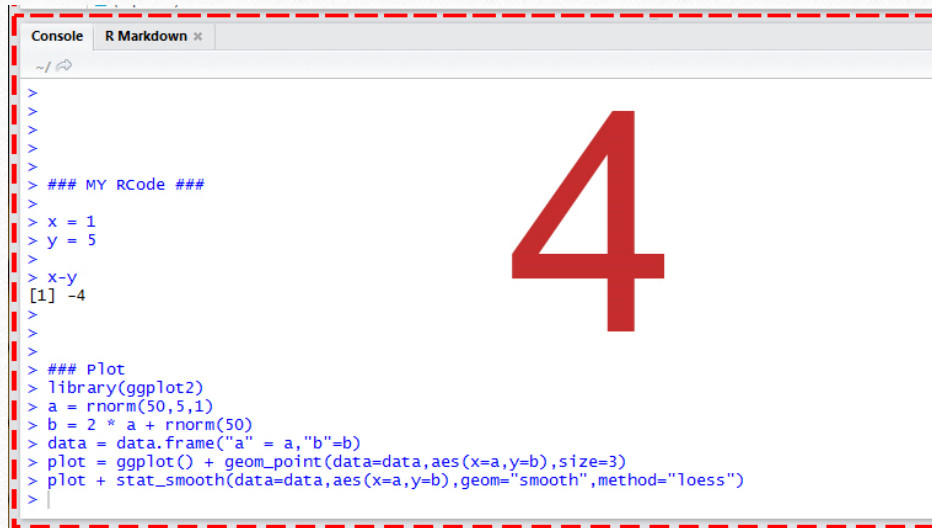
Les autres fenêtres au choix :

- Visualisation des fichiers du répertoire de travail (« **Files** »)
- Visualisation des graphiques produits (« **Plots** »)
- Liste des packages installés (« **Packages** »)
- Aide en ligne (« **Help** »)



# L'interface Rstudio

- La console : c'est ici que vous voyez l'exécution des instructions, vos résultats, et les éventuels messages d'erreurs si il y en a.



```
Console R Markdown x
~ / ↻
>
>
>
>
> ### MY RCode ###
> x = 1
> y = 5
> x-y
[1] -4
>
>
> ### Plot
> library(ggplot2)
> a = rnorm(50,5,1)
> b = 2 * a + rnorm(50)
> data = data.frame("a" = a,"b"=b)
> plot = ggplot() + geom_point(data=data,aes(x=a,y=b),size=3)
> plot + stat_smooth(data=data,aes(x=a,y=b),geom="smooth",method="loess")
>
```



# Les objets

[Formation-R/scripts/cours 1 prise en main.R](#)

---

- Sur R, chaque résultat est stocké dans la mémoire vive de l'ordinateur sous forme d'objets qui ont chacun un nom.
- Un objet représente un concept, une idée. Il se matérialise par une entité qui possède sa propre identité. Dans celle-ci, l'on compte deux aspects majeurs: la structure interne et le comportement.
- L'utilisateur agit sur les objets avec des opérateurs (arithmétiques, logiques, comparaison) et des fonctions (qui sont elles-mêmes des objets).

# Les objets

Formation-R/scripts/cours 1 prise en main.R)

---

On assigne un objet avec `<-` ou `=`

Dans R on distingue différents types d'objets :

- caractères (strings en anglais);
- nombres (entiers ou réels);
- dates;
- valeurs logiques qui ne prennent que deux valeurs: TRUE (vrai) ou FALSE (faux);
- facteurs qui sont un format spécial dans R prévu pour les variables catégorielles.

**Ce qui se ressemble s'assemble : on ne peut pas mélanger des objets de différents types !**

# Les vecteurs

Formation-R/scripts/cours 1 prise en main.R)

---

Stocker plusieurs valeurs d'un même type dans un objet fait de cet objet **un vecteur**.

Pour créer un vecteur on peut additionner plusieurs objets simples, ou stocker les valeurs à l'intérieur d'un `c()`.

Exemple : `c(3,12,7)` est un vecteur qui contient les valeurs 3, 12 et 17

Si une opération effectuée sur un vecteur, celle-ci sera appliquée à l'ensemble des valeurs de ce dernier.

R dispose d'un ensemble fonctions de bases pour les calculs sur vecteur

# Les objets de type numérique

---

Les valeurs continues  
(« double » dans R) et  
discrètes (« entier »  
dans R)





# Les objets de type caractère

Les valeurs caractères peuvent être désordonnées, ordonnées, ou binaires.



# Les objets complexes

---

- Les vecteurs sont une forme d'objet complexe
- Les matrices : collection de vecteurs organisée de façon rectangulaire, ne peuvent former une matrice que des éléments de même nature.
- Les tibble : format d'organisation de données en forme rectangulaire (tableau), permet de combiner des variables numériques et caractères.
- Les lists : objets permettant de contenir des données non structurées de la même façon.

# Travaux pratiques : analyses univariées et bivariées

---

- Comment analyser des objets complexes ? Connaitre leur structure et leur distribution ? Découverte des fonctions de bases d'analyses univariées :  
[Formation-R/scripts/cours 2 analyses univaries.R at main · IACPouembout/Formation-R \(github.com\)](#)
- Comment croiser des objets complexes et faire apparaitre des corrélations entre ces derniers ? Découverte des fonctions de bases d'analyse bivariées :  
[Formation-R/scripts/cours 3 analyse bivaries.R at main · IACPouembout/Formation-R \(github.com\)](#)



Exploration, manipulation de  
données, jointures et « tidy data »

---



# Les étapes d'une exploitation de données

- La plupart du temps, les données brutes ne sont pas utilisables telles quel.
- Il faut d'abord préparer les données à leur exploitation :
  - 1) Découverte des données
  - 2) Structuration des données
  - 3) Nettoyage des données, suppression des outliers et corrections.
  - 4) Enrichissement des données, ajout de données additionnelles
  - 5) Validation des données
  - 6) Publication



# Découverte des données

---

- L'exploration de données est une boucle pouvant se résumer aux étapes suivantes :
  1. Générer des questions à propos de vos données.
  2. Chercher des réponses en visualisant, transformant et modélisant vos données.
  3. Utilisez ce que vous avez appris pour redéfinir vos questions ou en générer de nouvelles.

Durant cette phase initiale d'exploration, sentez vous libre de vous posez toutes les questions possibles !

*“Far better an approximate answer to the right question, which is often vague, than an exact answer to the wrong question, which can always be made precise.” — John Tukey*

# Découverte des données

La visualisation comme outil d'exploration des données

---

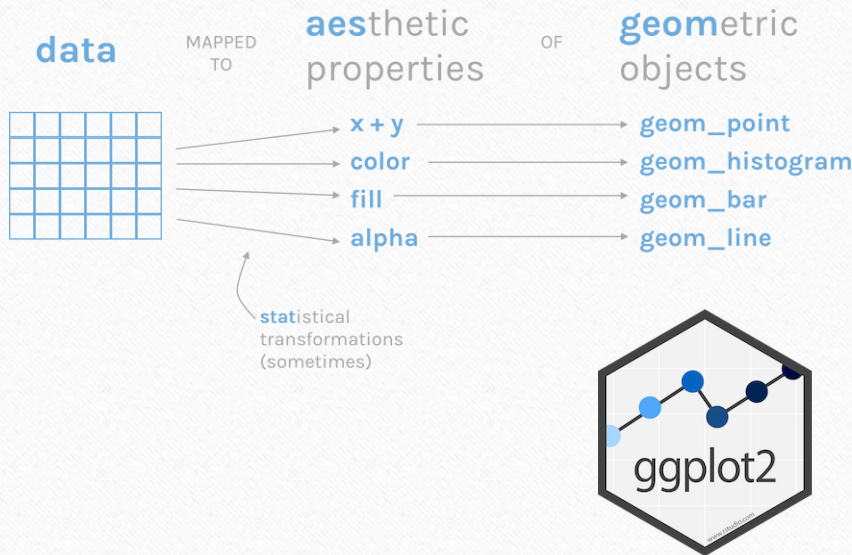
- “*The simple graph has brought more information to the data analyst’s mind than any other device.*” — John Tukey
- R possède un puissant moteur graphique interne, qui permet de "dessiner" dans un graphique en y rajoutant des segments, des points, du texte, ou toutes sortes d'autres symboles. Celles-ci peuvent cependant être difficiles à manier pour des graphiques plus aboutis.

# Découverte des données

La visualisation comme outil d'exploration des données

[Formation-R/scripts/cours\\_4\\_visualisation\\_donnees.R](#) at main · IACPouembout/Formation-R ([github.com](#))

- L'extension [ggplot2](#) développée par Hadley Wickham et mettant en œuvre la "grammaire graphique" théorisée par Leland Wilkinson, devient vite indispensable lorsque l'on souhaite réaliser des graphiques plus complexes.
- Le concept de grammaire graphique part du principe qu'un graphique associe les données aux attributs esthétiques (couleur, forme, taille) d'objets géométriques (points, lignes, barres), tout en présentant les possibles transformations statistiques (échelles).





# Découverte des données

Recodages et modifications conditionnelles

[Formation-R/scripts/cours 5 recodage et fonctions conditionnelles.R at main · IACPouembout/Formation-R \(github.com\)](#)

---

- Les données ne sont parfois pas prêtes à être directement visualisées, il ne faut pas s'interdire des étapes de recodage ou de filtrage durant la phase d'exploration !
- Les fonctionnalités des packages du tidyverse facilitent ici grandement cette tâche : [dplyr](#) pour manipuler des données, et [forcats](#) pour la gestion des facteurs
- Le package [questionr](#) est également très utile pour le recodage interactif



# Manipuler des données

Filtrer et produire des indicateurs

[Formation-R/scripts/cours\\_6\\_manipulation\\_donnees.R at main · IACPouembout/Formation-R \(github.com\)](https://github.com/IACPouembout/Formation-R/blob/main/scripts/cours_6_manipulation_donnees.R)

---

- Le package dplyr fournit un ensemble de fonctions vous permettant de manipuler vos données, de créer des nouvelles variables et les filtrer de façon conditionnelle.



# Manipuler des données

Enrichir ses données, données relationnelles et jointures

[Formation-R/scripts/cours\\_6\\_manipulation\\_donnees.R at main · IACPouembout/Formation-R \(github.com\)](https://github.com/IACPouembout/Formation-R/blob/main/scripts/cours_6_manipulation_donnees.R)

---

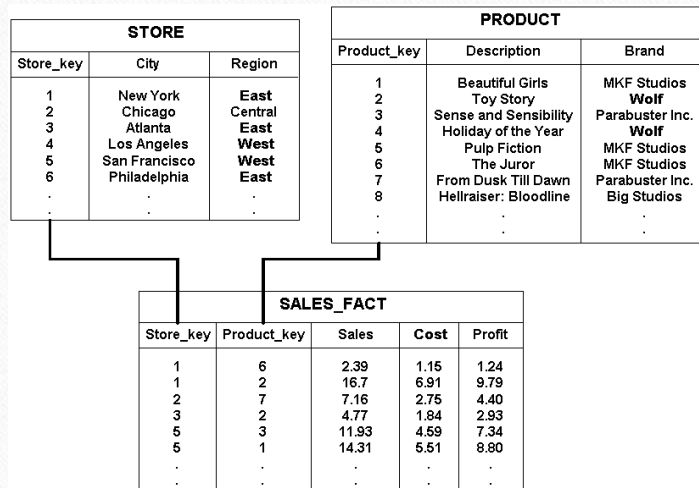
- Il est rare de n'utiliser qu'une seule table de données pour une enquête. En général, celles-ci sont séparées en plusieurs tables en fonction de l'entité observée.
- Ces données sont appelées «données relationnelles » et sont liées les unes aux autres par une clé identifiant.

# Manipuler des données

## Enrichir ses données, données relationnelles et jointures

[Formation-R/scripts/cours\\_6\\_manipulation\\_donnees.R](https://github.com/IACPouembout/Formation-R) at main · IACPouembout/Formation-R (github.com)

- Exemple d'une base de données relationnelle



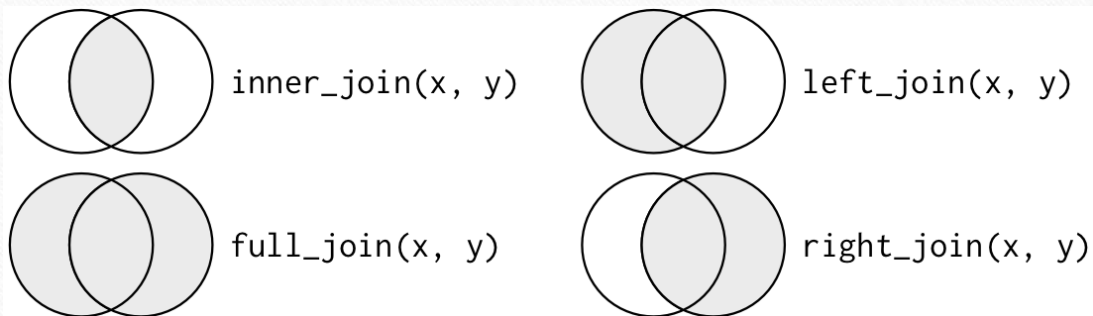


# Manipuler des données

Enrichir ses données, données relationnelles et jointures

[Formation-R/scripts/cours\\_6\\_manipulation\\_donnees.R at main · IACPouembout/Formation-R \(github.com\)](https://github.com/IACPouembout/Formation-R/blob/main/scripts/cours_6_manipulation_donnees.R)

- Pour récupérer des informations disponibles dans deux tables différentes, il faut les joindre par leur clé identifiant, c'est le principe de la jointure de tables.



# Structuration des données

Les « tidy data » : optimiser ses données

[Formation-R/scripts/cours\\_7\\_tidy\\_data\\_pivotage.R at main · IACPouembout/Formation-R \(github.com\)](#)

Le tidy data (« données propres ») est un concept développé par H. Wickham, afin de faciliter et standardiser le traitement de données.

“**TIDY DATA** is a standard way of mapping the meaning of a dataset to its structure.”

—HADLEY WICKHAM

In tidy data:

- each variable forms a column
- each observation forms a row
- each cell is a single measurement

each column a variable

id	name	color
1	floof	gray
2	max	black
3	cat	orange
4	donut	gray
5	merlin	black
6	panda	calico

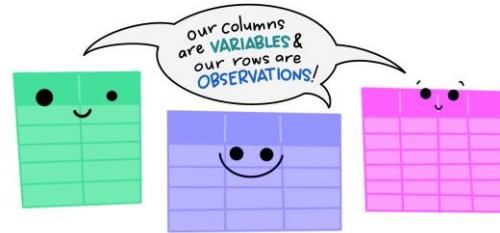
each row an observation

# Structuration des données

## Les « tidy data » : optimiser ses données

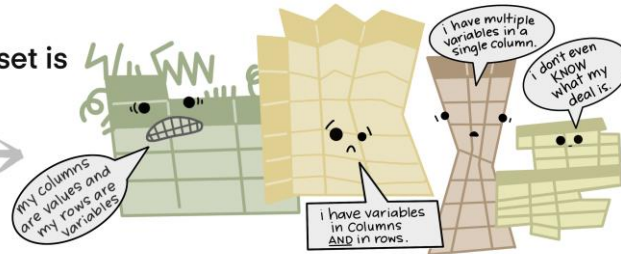
[Formation-R/scripts/cours\\_7\\_tidy\\_data\\_pivotage.R at main · IACPouembout/Formation-R \(github.com\)](#)

The standard structure of tidy data means that  
"tidy datasets are all alike..."



"...but every messy dataset is  
messy in its own way."

—HADLEY WICKHAM



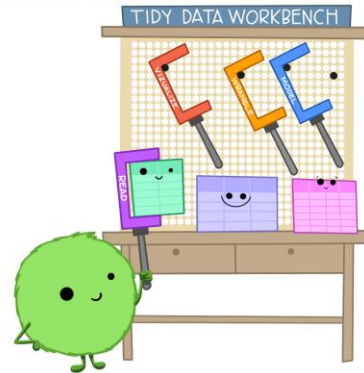


# Structuration des données

## Les « tidy data » : optimiser ses données

[Formation-R/scripts/cours\\_7\\_tidy\\_data\\_pivotage.R at main · IACPouembout/Formation-R \(github.com\)](#)

When working with tidy data,  
we can use the **same tools** in  
**similar ways** for different datasets...



...but working with untidy data often means  
reinventing the wheel with **one-time**  
**approaches** that are **hard to iterate or reuse**.





# Structuration des données

Les « tidy data » : optimiser ses données

[Formation-R/scripts/cours\\_7\\_tidy\\_data\\_pivotage.R at main · IACPouembout/Formation-R \(github.com\)](#)

Exemple de conversion de données en version « tidy »

country	year	cases
Afghanistan	1999	745
Afghanistan	2000	2666
Brazil	1999	37737
Brazil	2000	80488
China	1999	212258
China	2000	213766

country	1999	2000
Afghanistan	745	2666
Brazil	37737	80488
China	212258	213766

table4

Sur R, le package `tidyr` permet de pivoter les données facilement.



# Nettoyage de données

- 80 % du temps d'une exploitation de données consiste en du « data cleaning » !
- Il faut d'une part, corriger les erreurs structurelles : mauvais noms de variables, modalités inconstantes, doublons...
- D'autre part, repérer les valeurs aberrantes et/ou manquantes, les corriger, les imputer ou les supprimer.



# Publication

## Produire ses rapports avec Rmarkdown

- L'extension rmarkdown permet de générer des documents de manière dynamique en mélangeant texte mis en forme et résultats produits par du code R. Les documents générés peuvent être au format HTML, PDF, Word, et bien d'autres
- Rmarkdown ne fait pas partie du tidyverse, mais elle est installée et chargée par défaut par RStudio



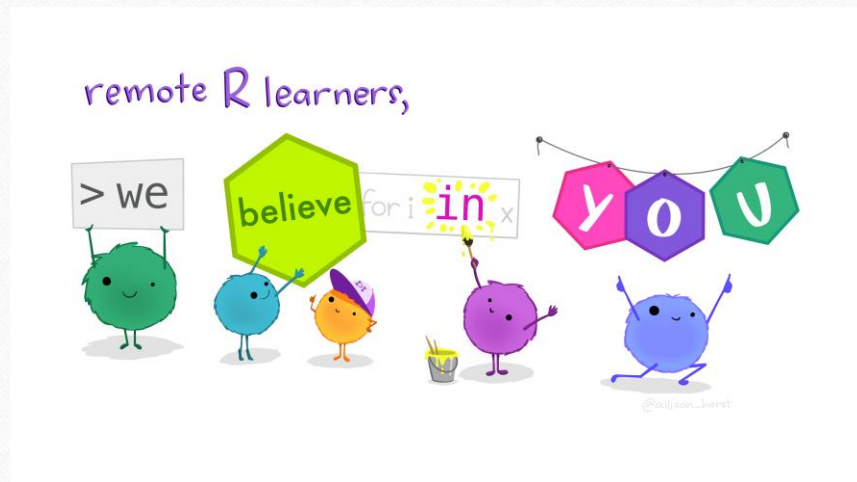


# Merci de votre attention

Le plan de ce document est en partie basée sur celui du document « Introduction au tidyverse » écrite par Julien Barnier, disponible sur <https://juba.github.io/tidyverse>

Les illustrations sont le fruit du travail de Allison Horst <https://github.com/allisonhorst/stats-illustrations>

Ces derniers sont mis à disposition selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 4.0 International](#).





# Autres ressources

---

- [R for data science](#)
- [ggplot2: Elegant Graphics for Data Analysis](#)
- [R Graphics Cookbook](#)
- [Contes et stats R](#)
- [Analyse-R](#)