Packages R pour controler son travail et ainsi mieux optimiser, réutiliser et communiquer autours de ses analyses

Journées du PEPI IBIS 2023

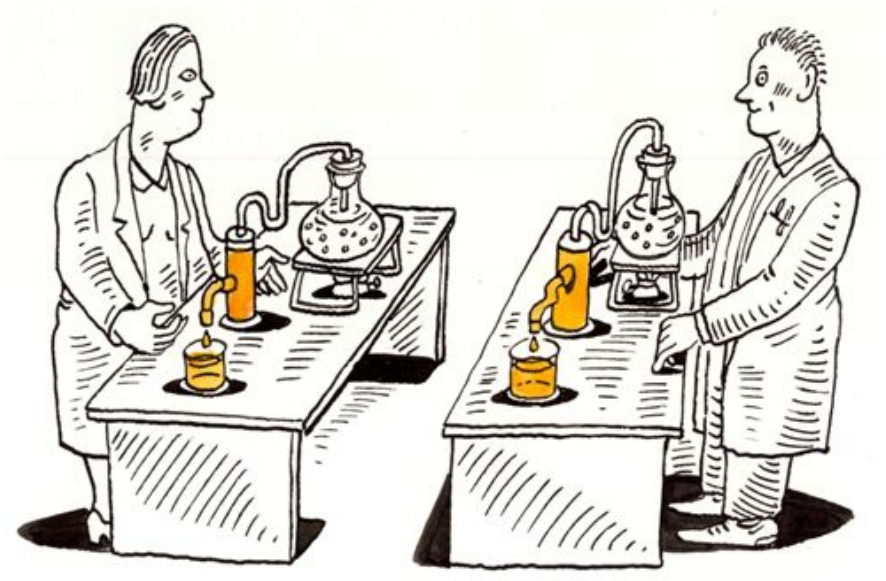
Cédric Midoux
PROSE & MaIAGE

Philippe Ruiz
MEDIS

September 15, 2023

Reproductibilité, pourquoi faire ?

Derrière la reproductibilité : la transparence dans la recherche



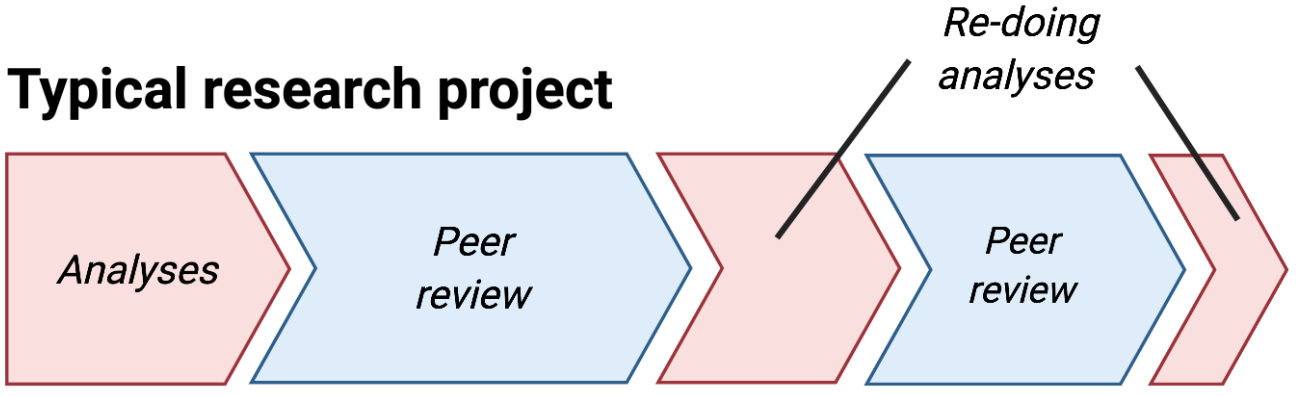
- Vous oblige à vérifier votre travail (partage des données + code)
- Votre futur vous-même vous remerciera
- Et vos collègues aussi
- En étant reproductible, vous renforcez votre crédibilité et votre réputation.
- La reproductibilité favorise la confiance dans le processus scientifique.

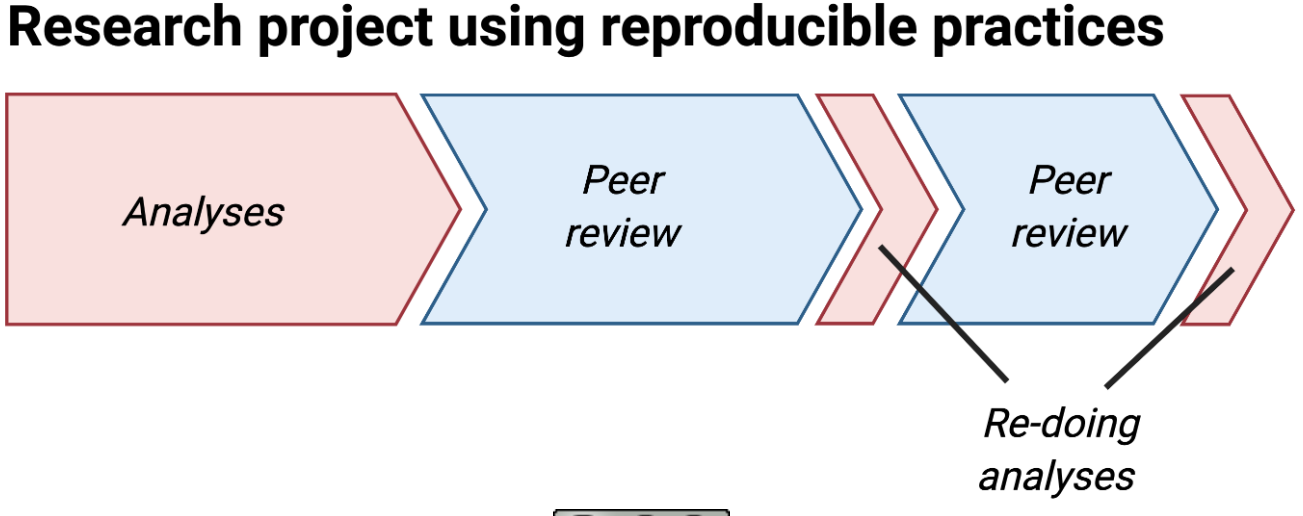
> Expliquer pour justifier et comprendre

> Refaire pour vérifier, corriger et réutiliser

Vous contribuez à l'accélération des progrès scientifiques

et vous ne perdez pas de temps …

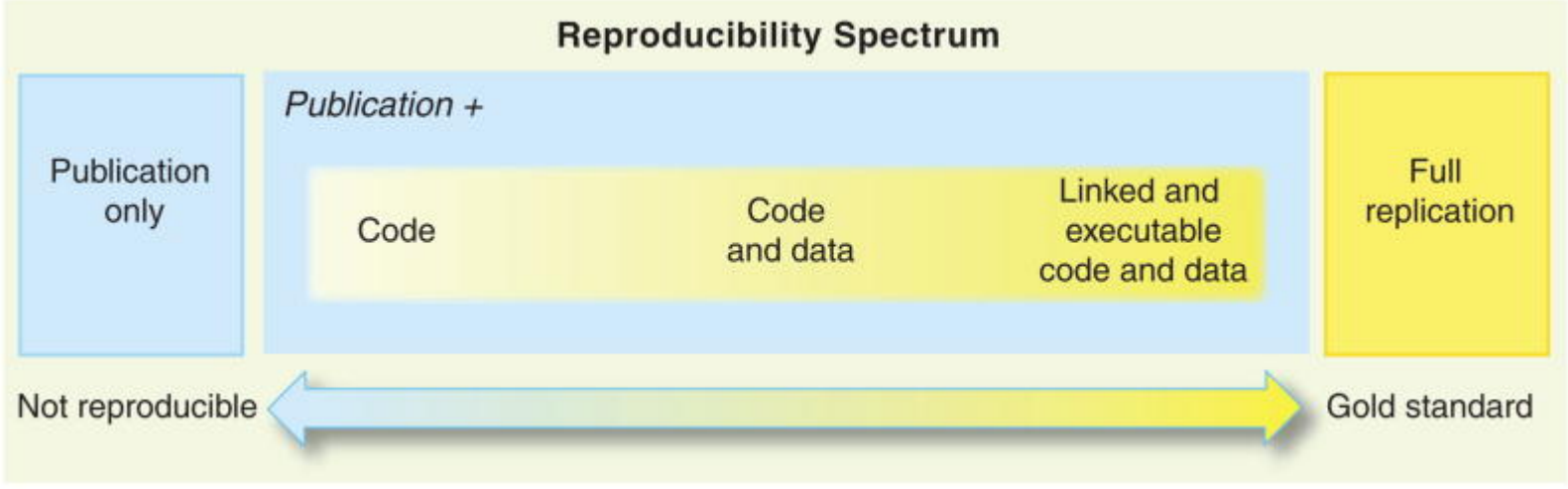




Quintana (2022)

un pièce des principes FAIR

Spectre de la reproctibilité

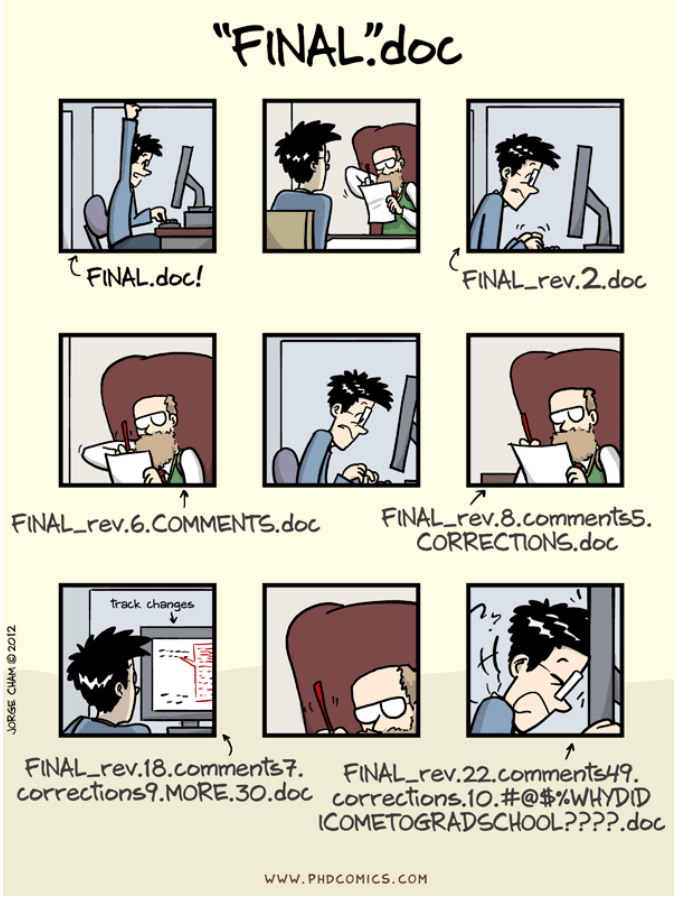


Peng (2011)

- Ne pas avoir peur d'avancer à petit pas, marche après marche
- Processus itératif et progressif

Controler ses sources (script & données)

Contrôle des versions (`git`)



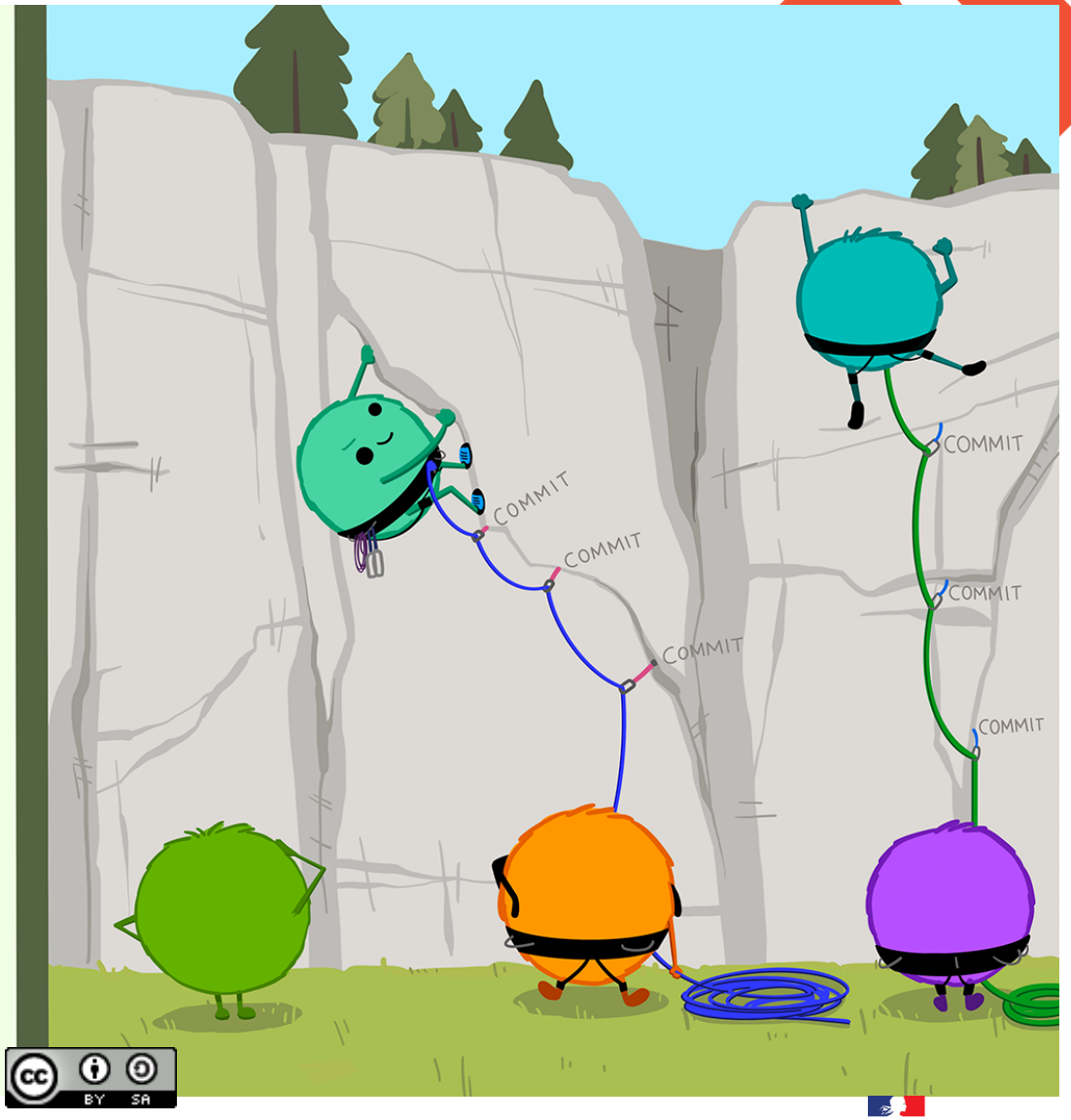
- Enregistrer les modifications apportées à un ensemble de fichiers
- Suivre l'historique et revoir toutes les modifications
- Revenir à des versions antérieures
- Travailler collaborativement sur des fonctionnalités parallèles

Ça marche avec des scripts et des codes, des protocoles & de la documentation, des rapports, n'importe quels documents!

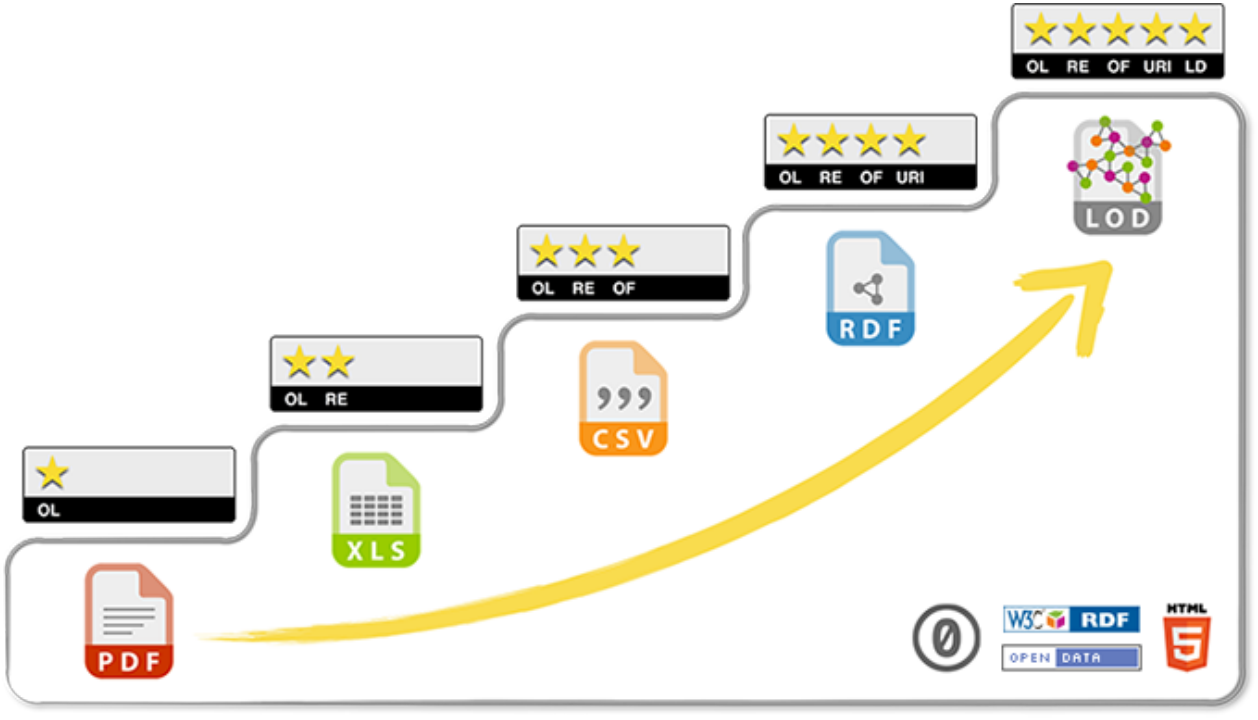
Comment ça marche ?

> Using a Git commit is like using anchors and other protection when climbing…**if you make a mistake, you can't fall past the previous commit**.
>
> Commits are also helpful to others, because **they show your journey, not just the destination**.
>
> – HADLEY WICKHAM & JENNY BRYAN

Wickham & Bryan, RPackages (https://r-packages.org/preface.html)



Open Data 5★



5 Star Data

- OL★ : Open License
- RE★ : machine REadable
- OF★ : Open Format
- URI★ : Uniform Resource Identifier
- LD★ : Linked Data

Comment on fait ?

- Rédiger un Plan de Gestion de Données
- Définir un espace collaboratif unique
  - nomenclature, sauvegarde, sécurité, …
- Décrire les métadonnées et un vocabulaire contrôlé
- Déposer les données dans un entrepôt international
- *Contrôler son environnement, son workflow, ses documents*

> Ca tombe bien, c'est la suite de l'exposé !

Travailler dans un environnement controlé

Constat

> While preparing a manuscript, to our surprise, attempts by team members to replicate these results produced different calculated NMR chemical shifts despite using the same Gaussian files and the same procedure outlined by Willoughby et al. […] these conclusions were based on chemical shifts that appeared to depend on the computer system on which step 15 of that protocol was performed.

Bhandari Neupane et al. (2019)

Table 2. Variability in Calculated Carbon Chemical Shifts of 1b

no.	LINUX (Ubuntu16)	Windows (ver. 10)	Mac (Mavericks)	Mac (Mojave)
1	172.4	173.2	173.2	172.7
2	36.0	37.7	37.7	39.3
3	68.3	68.4	68.4	69.0
4	70.6	70.5	70.5	71.2
5	79.0	78.4	78.4	79.0
6	15.5	13.3	13.3	13.3
7	162.4	162.5	162.5	161.8
8	110.4	109.8	109.8	110.3
9	155.0	156.5	156.5	155.5
10	116.2	115.5	115.5	116.0
11	131.6	131.6	131.6	131.7
12	127.1	126.6	126.6	127.0
13	126.7	125.6	125.6	126.3

Problématique

> Si un script ne fonctionne que sur l'ordi ou il a été développé, il mourra avec cet ordi

⚠ **Chez moi ça marche.**

- Quels packages utilisés ? avec quelle version ?
- Quelle version de R ?
- Et quel OS ?

renv

- `renv` permet une gestion fine des versions des packages au sein de chaque projet
- Pas d'effet de la mise à jour d'un package sur les autres projets
- Les versions des packages tracé dans un fichier `renv.lock`
- Particulièrement adapté à une utilisation avec git

renv - workflow

1. Initialisation d'un nouvel environnement local pour le projet avec une bibliothèque privée

```
1  renv::init()
```

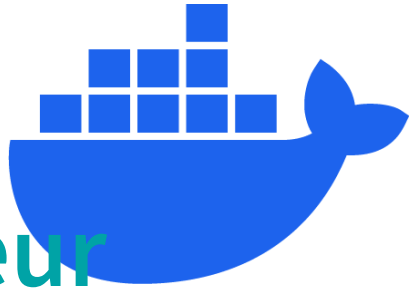
2. Installation des packages comme d'habitude

```
1  install.packages(...)
```

3. Sauvegarde de l'état de la bibliothèque privée du projet. Les packages utilisés et leur version sont détaillés dans le fichier `renv.lock`

```
1  renv::snapshot()
```

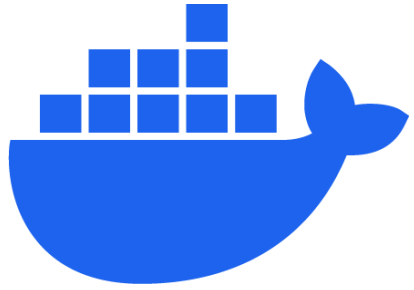
Docker



Aller plus loin, partagez votre ordinateur

- Docker est un outil de conteneurisation
- Un conteneur = un OS (+ du code)
- L'image créée est facilement partageable et exécutable sur un autre ordi, un autre OS
- Version allégée de la virtualisation

Docker - workflow



dockerfile

```
1  FROM rocker/verse:4.3.1
2  LABEL "AUTHOR" "cedric.midoux@inrae.fr"
3  LABEL "VERSION" "PEPI2023"
4
5  CMD echo "Hello, PEPI !!"
```

1. Créez l'image:

```
1  docker build -t hello:PEPI2023 .
```

2. Exécutez l'image

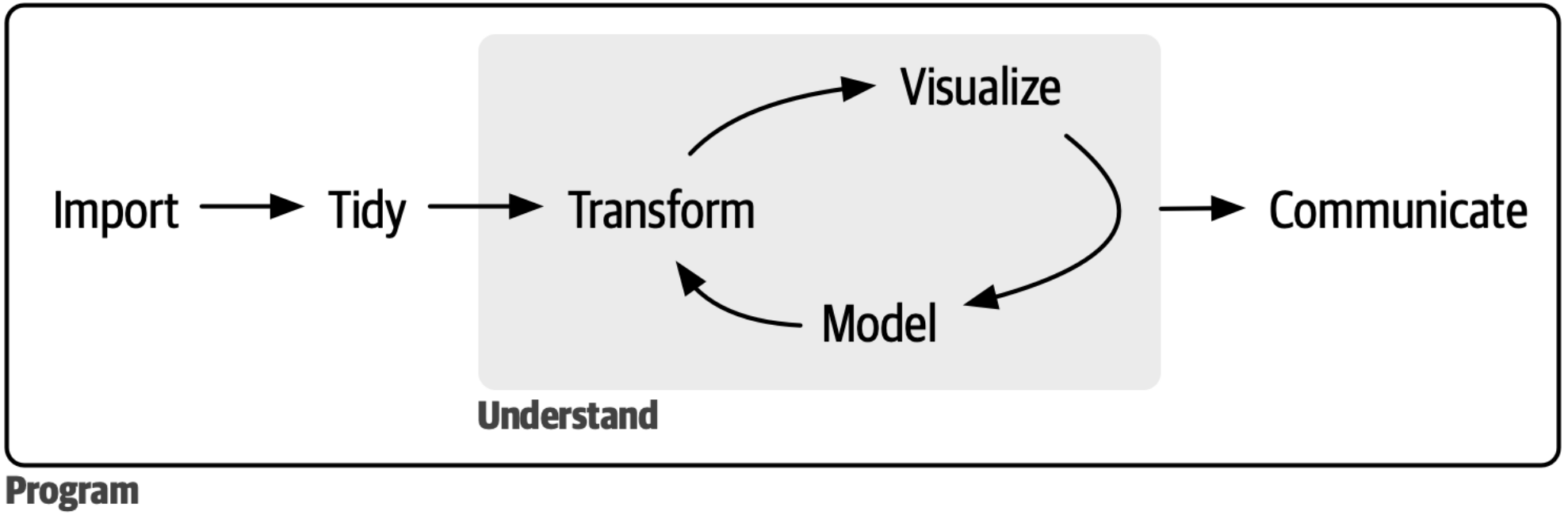
```
1  docker run --rm -d --name hello_container hello:PEPI2023
```

3. Publiez votre image

```
1  docker push hello:PEPI2023
```

Travailler dans flux d’analyses controlé

Data Science



Wickham, Çetinkaya-Rundel, and Grolemund (2023)

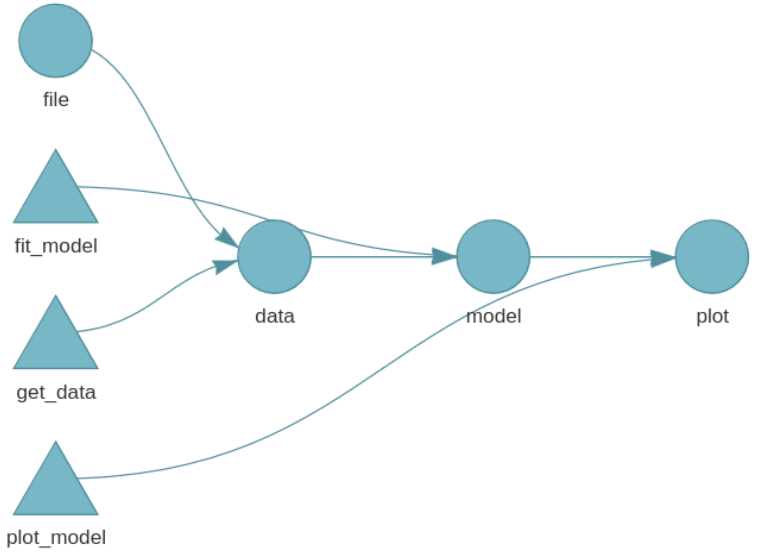
targets

Ce package permet de structurer un pipeline d'analyse sous une forme bien précise composé d'étapes écrit dans un schéma global (workflow). On pourrait le comparer à un petit `snakemake` ou `nextflow` sous `R`. Facilite la parallélisation.

Philosophie

écrire un pipeline d'analyse sous la forme d'un workflow dont chacune des étapes sont reliées et dépendantes les une des autres. Le buts est de structurer le workflow en étapes prédéfinies et toutes structurées de la même manière (une entrée, une fonction, une sortie) dont leur état est référencé lors de l'exécution du pipeline.

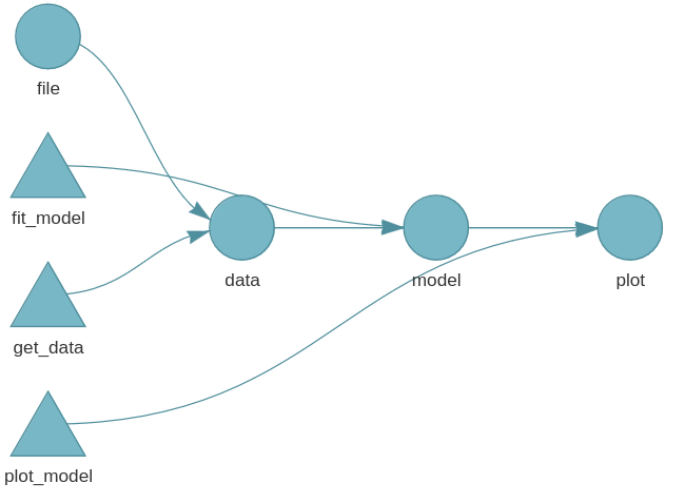
Schéma



```
├── _targets.R : le script d'execu
├── data/
│   ├── robject.RData : un objet R
│   ├── data.csv : les données
├── R/
│   ├── functionsMain.R : les fonc
│   ├── functionsPlots.R : les fon
│   ├── functionsTests.R : les fon
├── _output/
│   ├── output.csv : fichiers de s
├── _targets/
│   ├── meta/
│   ├── objects/
│   ├── user/
│   ├── workspaces/
```

_targets.R

```r
library(targets)
tar_source()
tar_option_set(packages = c("readr
list(
  tar_target(file, "data.csv", for
  tar_target(data, get_data(file))
  tar_target(model, fit_model(data
  tar_target(plot, plot_model(mode
)
```



Execution du pipeline

la fonction `tar_make()` exécute le pipeline dans son ensemble en respectant l'ordre des étapes écrites dans le fichier `_targets.R`.

```
> targets::tar_make()
▶ start target data
Warning: program compiled against libxml 210 using older 209
● built target data [11.07 seconds]
▶ start target dataFact
● built target dataFact [0.133 seconds]
▶ start target dataSub
● built target dataSub [0.015 seconds]
▶ start target betaTrans
converting counts to integer mode
gene-wise dispersion estimates
mean-dispersion relationship
final dispersion estimates
● built target betaTrans [1.379 seconds]
▶ start target prevalTab
● built target prevalTab [0.066 seconds]
▶ start target noDupname
● built target noDupname [0.022 seconds]
▶ start target dataSubTr
● built target dataSubTr [0.015 seconds]
▶ start target tabOut
```

```
'OTU_8151' for all replaced levels.
Consider editing this tax_table entry manually.
Row named: OTU_8163
contains no non-unknown values, returning:
'OTU_8163' for all replaced levels.
Consider editing this tax_table entry manually.
Row named: OTU_8193
contains no non-unknown values, returning:
'OTU_8193' for all replaced levels.
Consider editing this tax_table entry manually.
Registered S3 method overwritten by 'ggside':
  method from
  +.gg   ggplot2
● built target cca [0.378 seconds]
▶ start target deseqHeat
● built target deseqHeat [0.219 seconds]
▶ start target core
● built target core [0.175 seconds]
▶ start target betaTab
● built target betaTab [1.73 seconds]
▶ end pipeline [57.566 seconds]

>
```

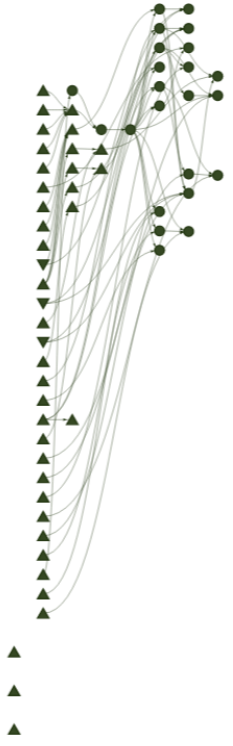
Ré-execution du pipeline

Lorsque l'on exécutera à nouveau le pipeline seules les étapes ayant été modifiées seront de nouveau exécutées.

```
✔ skip target dataFact
✔ skip target dataSub
✔ skip target betaTrans
✔ skip target prevalTab
✔ skip target noDupname
✔ skip target dataSubTr
✔ skip target tabOut
✔ skip target plsda
✔ skip target alphaTab
✔ skip target lefse
✔ skip target noDupname2
✔ skip target betaPlot
✔ skip target prevalPlot
✔ skip target deseqRes
✔ skip target dataAgg
✔ skip target compo
✔ skip target metaDataB
✔ skip target alphaPlot
✔ skip target cca
✔ skip target deseqHeat
✔ skip target core
✔ skip target betaTab
✔ skip pipeline [0.169 seconds]
```

Visualisation du pipeline

la fonction `tar_visnetwork()` affiche un DAG du pipeline au temps *t*, mettant en évidence l'état des étapes (ok, en retard, avec une erreur).

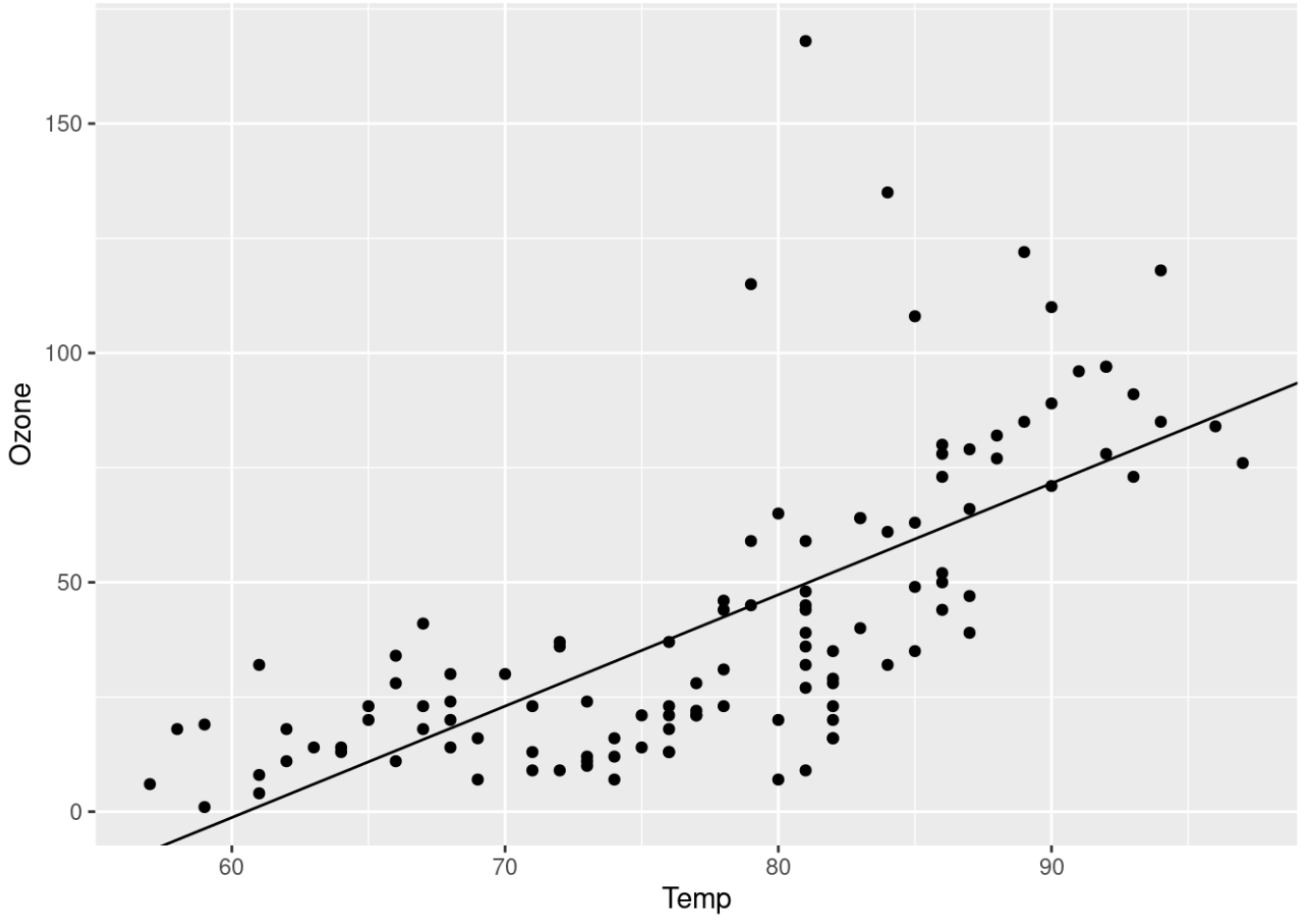


Un pipeline dans la vrai vie : workflow 16S

Visualisation des objets de sortie

Contrairement à une utilisation classique de R, les objets ne sont pas stockés dans l'environnent global mais dans les dossier `_targets` > `objects`. Il s'agit de fichiers compilés lisibles uniquement par targets via la commande `tar_read(object)`:

```
1  tar_read(plot)
```



Résultats

appeler les objets de sortie de targets dans les chuncks d'un fichier qmd:

```
1 summary(tar_read(model))
```

ou

```
1 tar_load(plot)
```

```
##
## Call:
## glm(formula = Aux ~ Causation + EPTrans + Country, family = binomial,
##     data = mydata)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.4837  -0.5344  -0.3428   0.3838   2.5340
##
## Coefficients:
##                    Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.8631     0.3771   4.941 7.79e-07 ***
## CausationInducive   -3.3725     0.3741  -9.015  < 2e-16 ***
## CausationPhysical    0.4661     0.6275   0.743 0.457575
## CausationVolitional -3.7373     0.4278  -8.735  < 2e-16 ***
## EPTransTr           -1.2952     0.3394  -3.816 0.000136 ***
## CountryBE            0.7085     0.2841   2.494 0.012633 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 609.05  on 454  degrees of freedom
## Residual deviance: 337.70  on 449  degrees of freedom
## AIC: 349.7
##
## Number of Fisher Scoring iterations: 5
```

Intégrer quarto au pipeline

Il est possible de générer un document `quarto` avec la fonction `quarto_render` du package `quarto`.

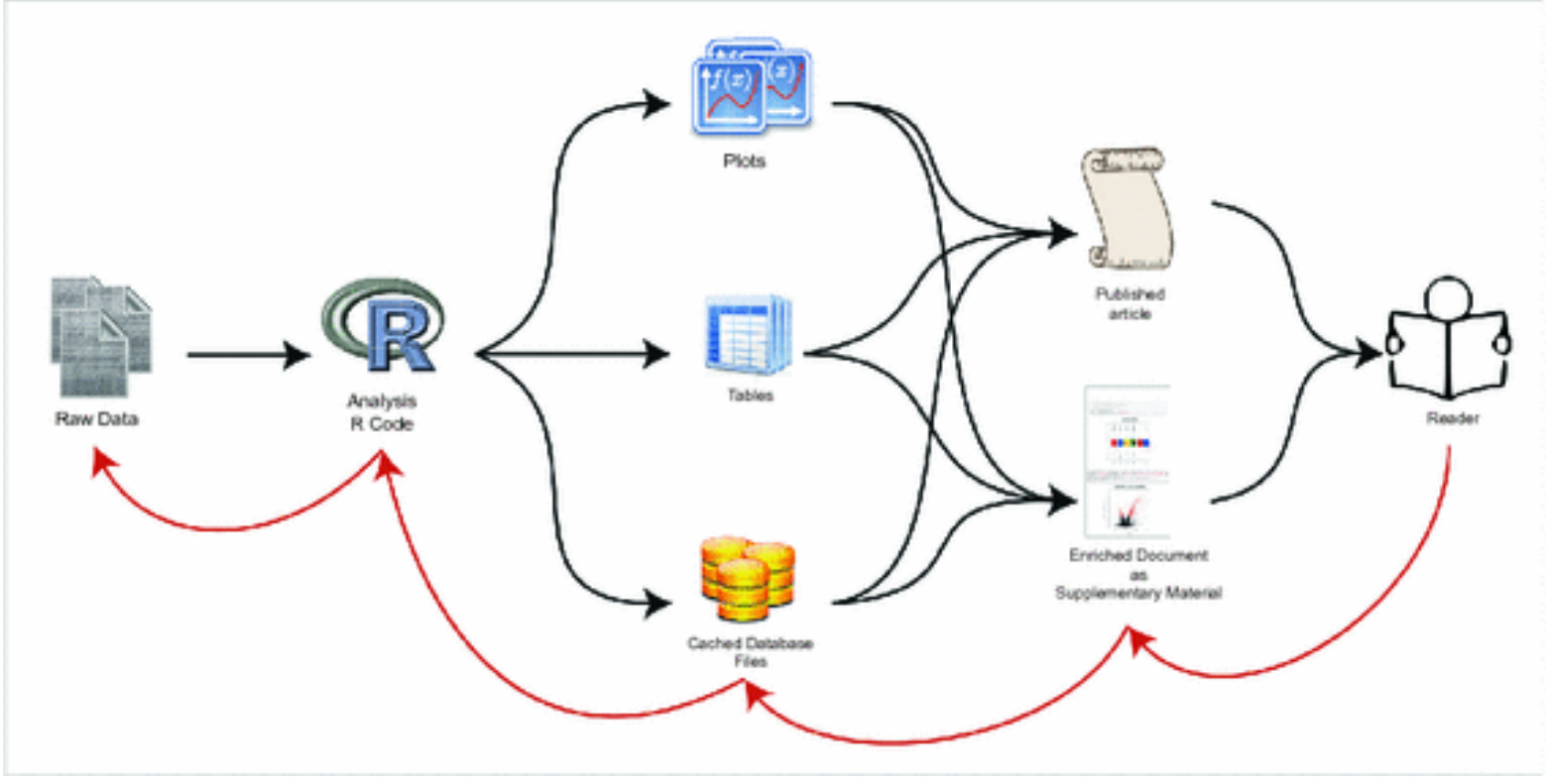
- générer un rapport lors d'une étape du pipeline avec :

```
1  tar_target(report, quarto_render("Report.qmd", output_format = "html"))
```

Controler ses supports de communication

RMarkdown

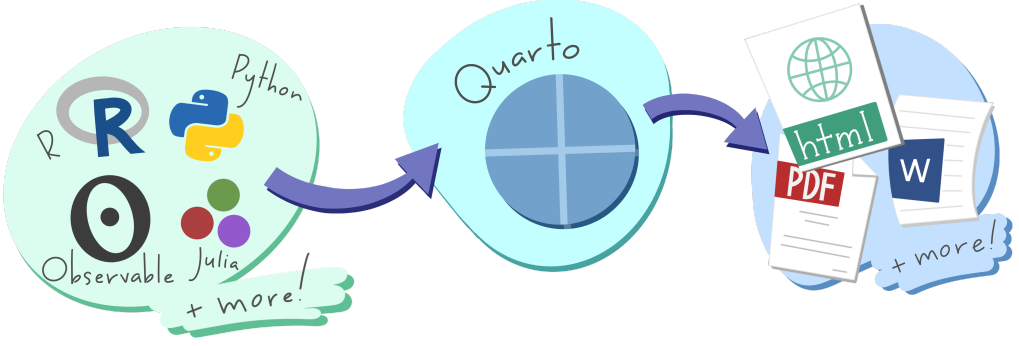
Unifier en unique document contexte, code, résultat, interprétation pour assure la cohérence des analyses …



Russo, Righelli, and Angelini (2016)

quarto

- Successeur de Rmarkdown
- Multi langages (R, Python, Julia, Observable)
- Documents de type rapports paginé, documents HTML, site web, livres, slides
- Interactivité
- Export en html, pdf, docx, ePub, …



An open-source scientific and technical publishing system

Quarto - exemples

```
---
title: "Quarto Computations"
---

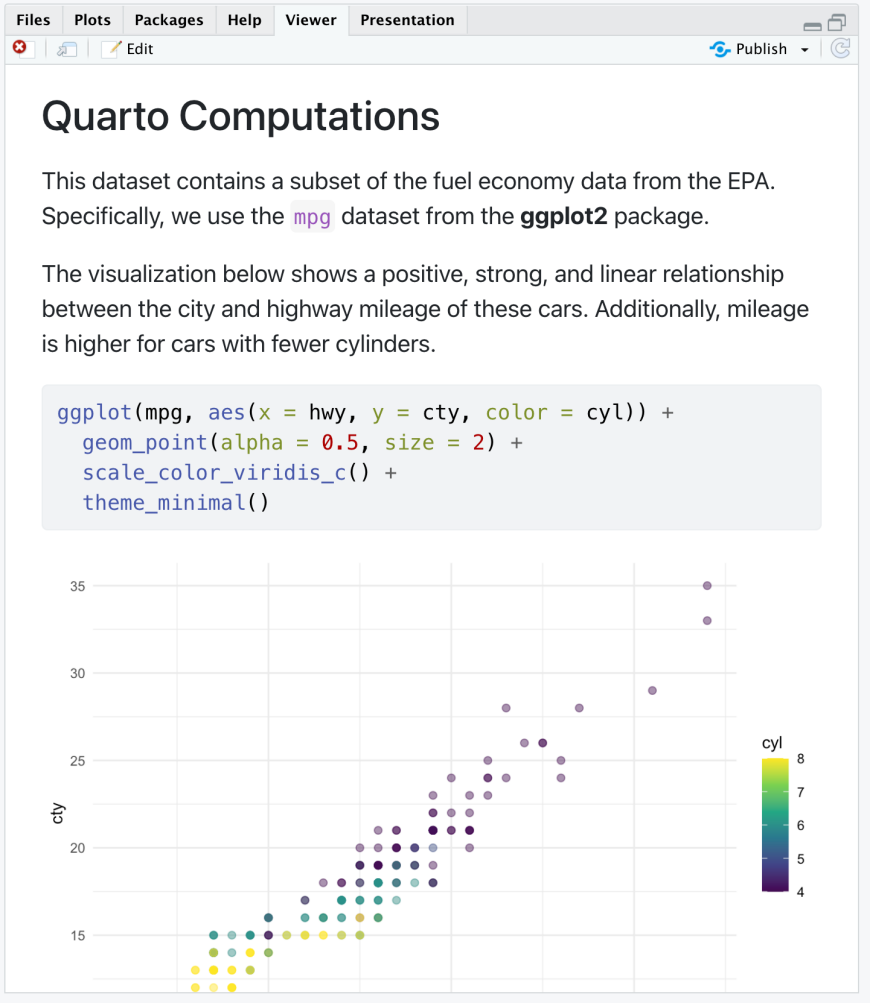
This dataset contains a subset of the
Specifically, we use the `mpg` dataset

```{r}
#| label: load-packages
#| echo: false

library(ggplot2)
```

The visualization below shows a positi
Additionally, mileage is higher for ca

```{r}
#| label: scatterplot
```


```

Quarto facilite le passage d'un format à l'autre

Document HTML / `targets`

lesson-1.qmd

```yaml
title: "Lesson 1"
format: html
```

Document PDF

lesson-1.qmd

```yaml
title: "Lesson 1"
format: pdf
```

Presentation

lesson-1.qmd

```yaml
title: "Lesson 1"
format: revealjs
```

Book

_quarto.yml

```yaml
project:
  type: book
  output-dir: _book
```

Website

_quarto.yml

```yaml
project:
  type: website
website:
  navbar:
    left:
      - lesson-1.qmd
```

And more in the Gallery …

Les nouveautés de quarto (HTML)

- YAML standardisé entre les formats
- Decouplé de RStudio
- Présentation plus cohérente entre les formats
- Tab Panels
- Code Highlighting
- Mise en cache des sorties (freezing)
- Mise en page précise

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

INRAE

RMarkdown

````
```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(DT)
```
````

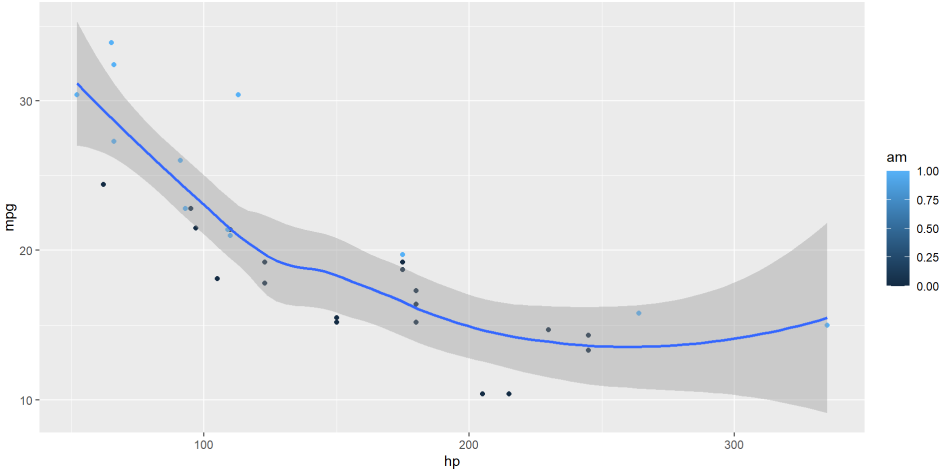
Quarto

````
```{r}
#| label: "setup"
#| include: false
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(DT)
```
````

Les options sont déplacées au sein du chunk avec #| (hash-pipe) pour chaque ligne

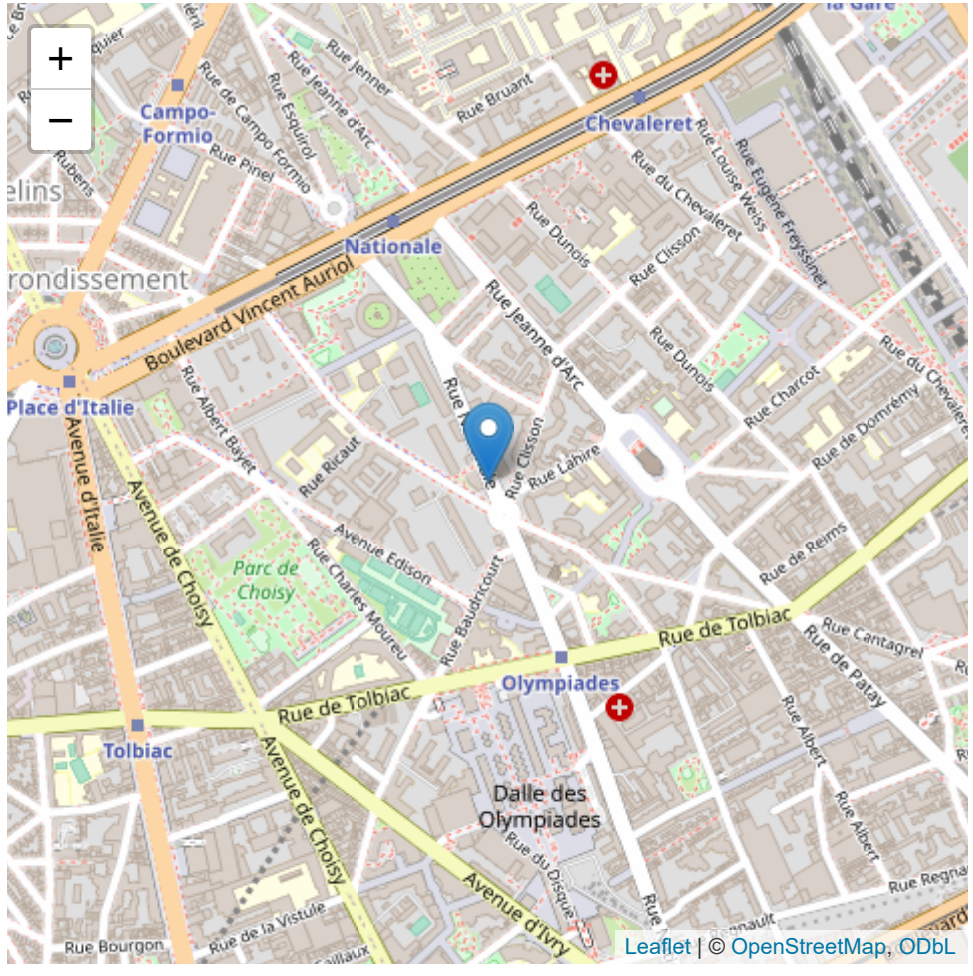
Code highlighting

```
1  ```{r}
2  #| label: my_plot
3  #| code-line-numbers: "|10"
4  #| output-location: column
5  library(ggplot2)
6  ggplot(
7    mtcars,
8    aes(hp, mpg)
9  ) +
10   geom_point(aes(color = am)) +
11   geom_smooth(formula = y ~ x, met
12 ```
```



Widgets

```{r}
#| output-location: column-fragmen
library(leaflet)
leaflet(width = "480px") %>%
  addTiles() %>%
  addMarkers(
    lat=48.829510,
    lng=2.364861,
    popup="Vous êtes ici !"
  )
```



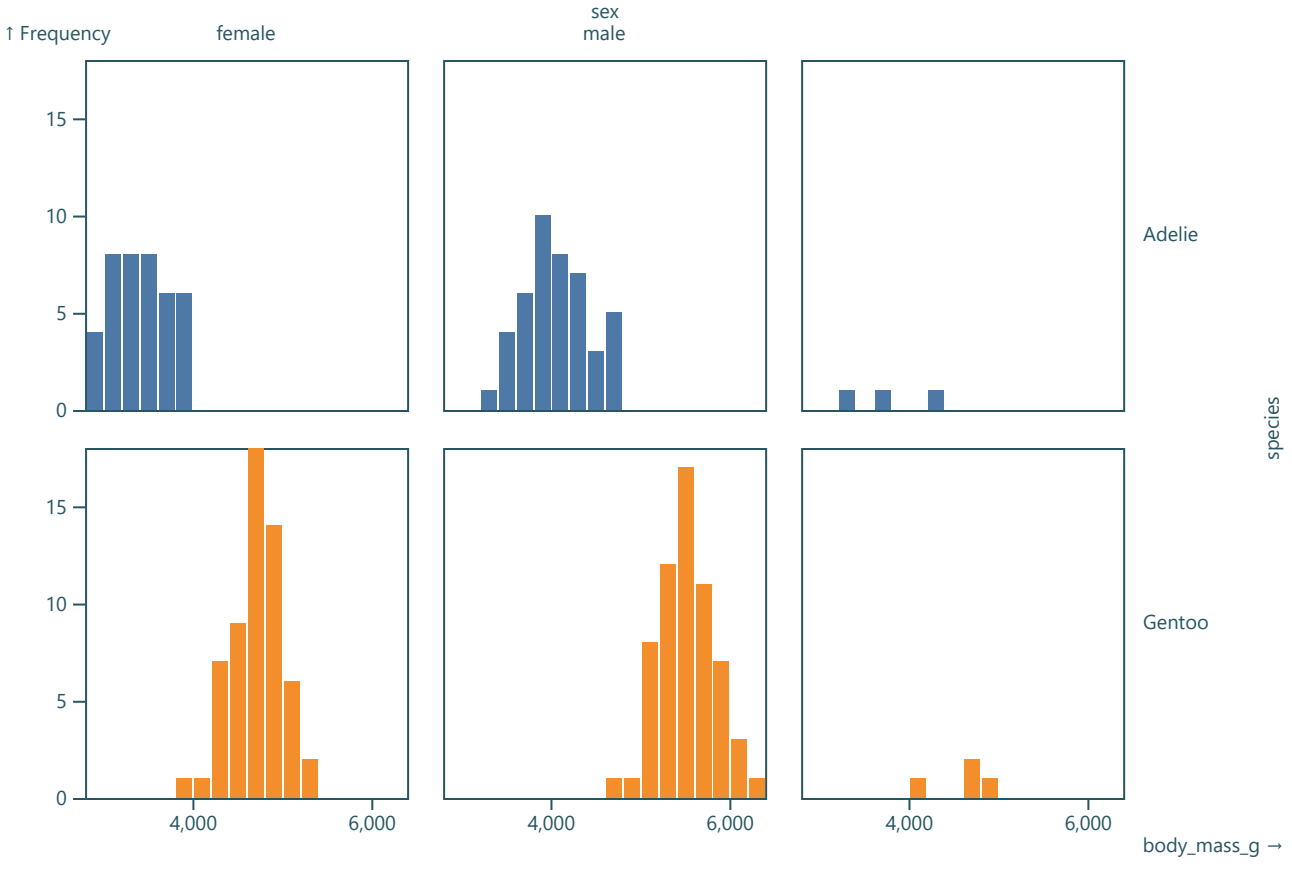
Interactivité

Bill length (min):

35

Islands:

- [x] Torgersen
- [x] Biscoe
- [ ] Dream



Diffusion en CI/CD : GitLab Pages

- Les fichiers de sortie du projet se trouvent dans un dossier nommé `public`
- `_quarto.yml`

```yaml
project:
  type: website
  output-dir: public
```

- `.gitlab-ci.yml`

```yaml
# The Docker image that will be used to bu
image: rocker/verse:4.2

# Functions that should be executed before
before_script:
  - quarto install extension davidcarayon/

pages:
  script:
    - quarto render
  artifacts:
    paths:
      # The folder that contains the files
      - public
  rules:
    # This ensures that only pushes to the
    a pages deploy
    - if: $CI_COMMIT_REF_NAME == $CI_DEFAU
```

Github Actions for Quarto

1. quarto-dev/quarto-actions/setup - Install Quarto
2. quarto-dev/quarto-actions/render - Render project
3. quarto-dev/quarto-actions/publish - Publish project

- Doc

Et si on mélange tout ça

- Mettez en place un PGD, une gestion des métadonnées et un vocabulaire contrôlé
- Déposez les données sur un entrepôt
- Versionnez le code avec `git` et partager le sur Git[Lab|Hub]
- Fixez l'OS et les dépendances avec `docker` et les packages avec `renv`
- Contrôlez le workflow avec `targets`
- Rédigez les document avec `quarto`
- Déportez les calculs grace à l'intégration continue
- Déployez les documents avec GitLab Pages

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

INRAE

Step by step

ToDo

- Mettez en place un PGD, une gestion des métadonnées et un vocabulaire contrôlé
- Déposez les données sur un entrepôt
- Versionnez le code avec `git` et partager le sur Git[Lab|Hub]
- Fixez l'OS et les dépendances avec `docker` et les packages avec `renv`
- Contrôlez le workflow avec `targets`
- Rédigez les document avec `quarto`
- Déportez les calculs grace à l'intégration continue
- Déployez les documents avec GitLab Pages

Step-by-step

- Essayez de nouvelles choses progressivement
- Testez sur un petit projet pour commencer
- Restez au courant des nouveautés
- Les techno évoluent et facilitent l'usage
- Échangez avec vos collègues
- Soyez pragmatique
- Adaptez l'usage à vos besoins

RÉPUBLIQUE FRANÇAISE
Liberté
Égalité
Fraternité

INRAE

Biblio

Bhandari Neupane, Jayanti, Ram P Neupane, Yuheng Luo, Wesley Y Yoshida, Rui Sun, and Philip G Williams. 2019. “Characterization of Leptazolines a–d, Polar Oxazolines from the Cyanobacterium Leptolyngbya Sp., Reveals a Glitch with the ‘Willoughby–Hoye’ Scripts for Calculating NMR Chemical Shifts.” *Organic Letters* 21 (20): 8449–53.

CIRAD-DGDRS-DIST-FRA, ed. 2017. “Le Cycle de Vie Des Données. Intégrer La Gestion de Données Scientifiques Aux Activités de Recherche.” CIRAD. https://agritrop.cirad.fr/594579/.

Peng, Roger D. 2011. “Reproducible Research in Computational Science.” *Science* 334 (6060): 1226–27. https://doi.org/10.1126/science.1213847.

Quintana, Daniel. 2022. “Five Things about Open and Reproducible Science That Every Early Career Researcher Should Know.” *Open Science Framework*, August. https://doi.org/10.17605/OSF.IO/DZTVQ.

Russo, Francesco, Dario Righelli, and Claudia Angelini. 2016. *Advantages and Limits in the Adoption of Reproducible Research and r-Tools for the Analysis of Omic Data*. Edited by Claudia Angelini, Paola MV Rancoita, and Stefano Rovetta. Cham: Springer International Publishing.

Sébire, Fanny. 2023. “Check-list de l’Institut Pasteur pour des bonnes pratiques de gestion des données de recherche.” https://hal.science/hal-04123336.

The Turing Way Community. 2022. “The Turing Way: A Handbook for Reproducible, Ethical and Collaborative Research.” Zenodo. https://doi.org/10.5281/ZENODO.3233853.

Wickham, H., M. Çetinkaya-Rundel, and G. Grolemund. 2023. *R for Data Science*. O’Reilly Media. https://r4ds.had.co.nz/.