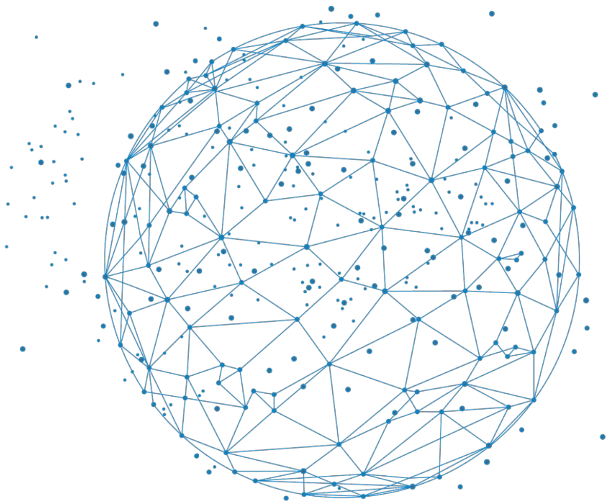
2025 June 5th



FAIR principles applied to bioinformatics

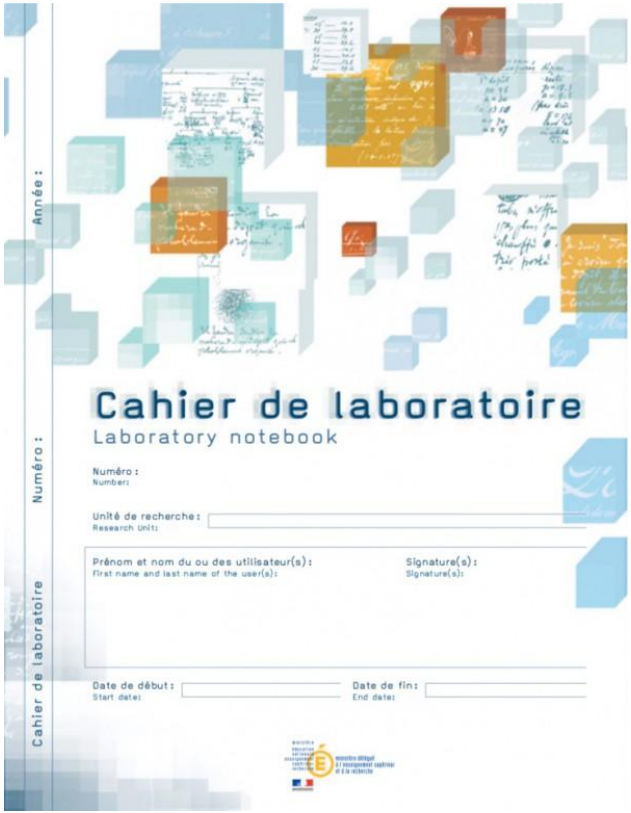
Thomas Denecker
CNRS - IFB
https://orcid.org/0000-0003-1421-7641

IFB INSTITUT FRANÇAIS DE BIOINFORMATIQUE

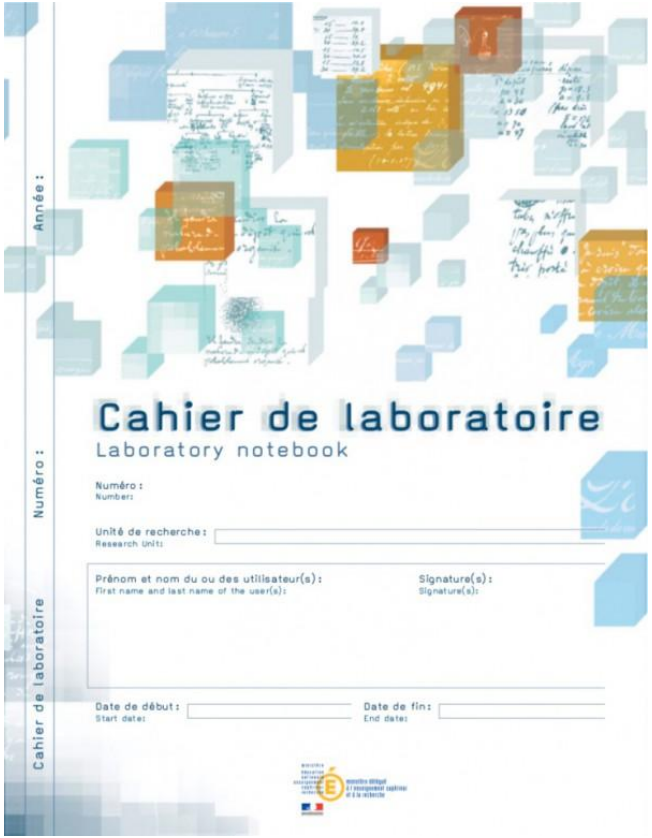
elixir FRANCE

cnrs INRAE Inserm cea

What is this?


Année :
Numéro :
Cahier de laboratoire
Cahier de laboratoire
Laboratory notebook
Numéro :
Number:
Unité de recherche :
Research Unit:
Prénom et nom du ou des utilisateur(s) :
First name and last name of the user(s):
Signature(s) :
Signature(s):
Date de début :
Start date:
Date de fin :
End date:

What is it for ?



The laboratory notebook allows :

- day-to-day recording of the details of the work
- to report on the progress and scientific experimentation, from the idea to the conclusion
- to keep knowledge in a lab

Also very useful for drafting a patent or for proving anteriority.

What is it for ?

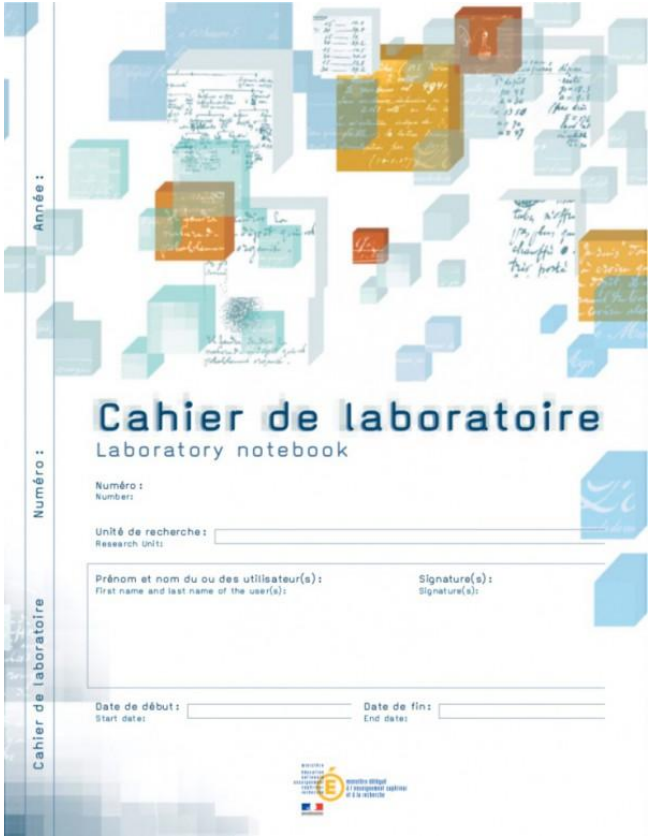
A legal tool :

- Each notebook and the pages are numbered.
- On the cover page, we find on each notebook the mentions of the owner of the results.
- Each page has two parts at the bottom intended to be dated and signed: two signatures = two people, user and third party (witness),
  - ideally a third party not involved in the research work but capable of understanding it

https://www.curie.asso.fr/-Cahier-de-laboratoire-national-.html

https://slideplayer.fr/slide/3817405/

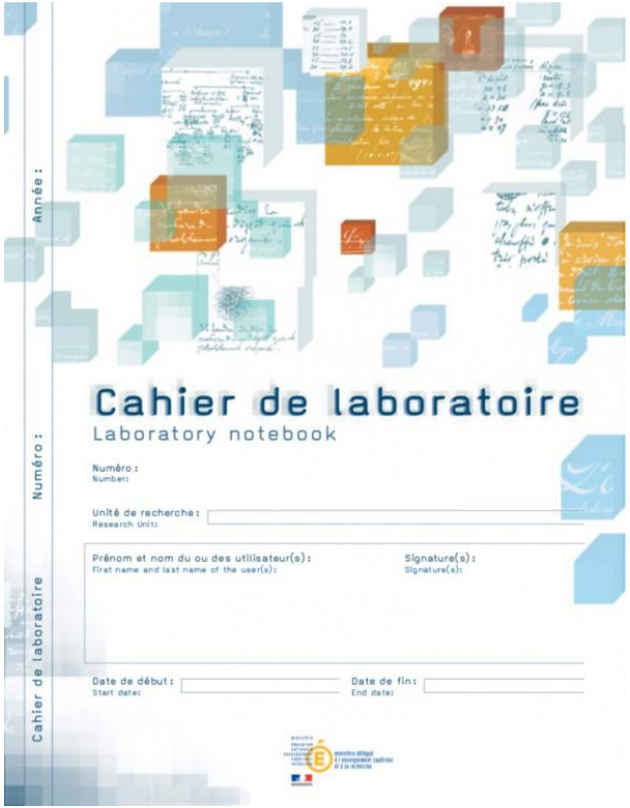
What is it for ?



For all those who carry out research work :

- researchers,
- engineers,
- technicians,
- doctoral students,
- trainees,
- etc.

Are you using it ?



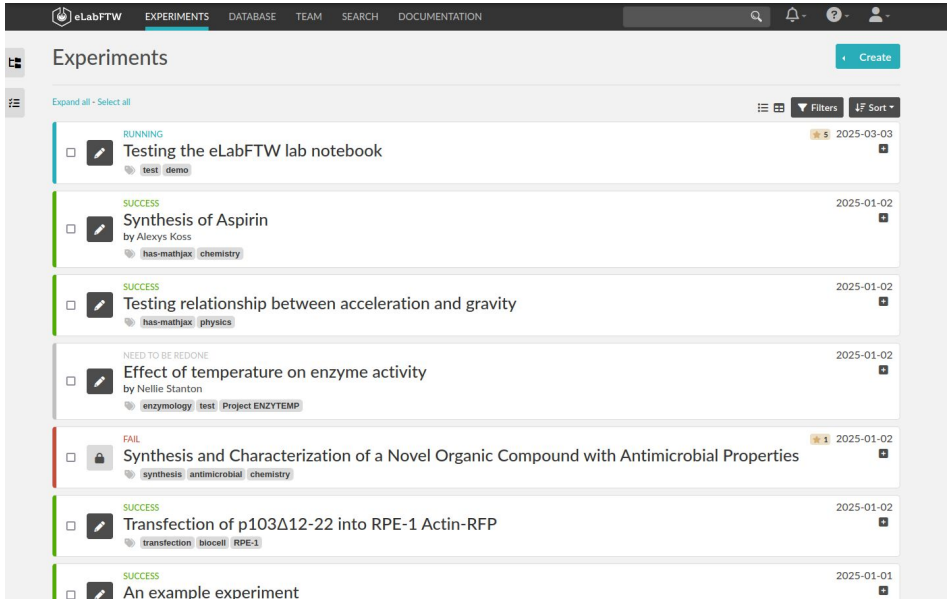
Electronic Laboratory Notebooks (ELN)

Modern LN since 2009 (C.U.R.I.E. Network)

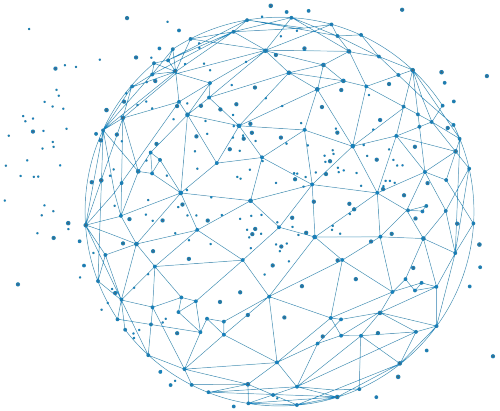
But less and less adapted to recent evolutions of our work

- Increased data quantity
- Change in the nature of data
- Dematerialization
- Security

We need an electronic tool for individual traceability.



What about programming?



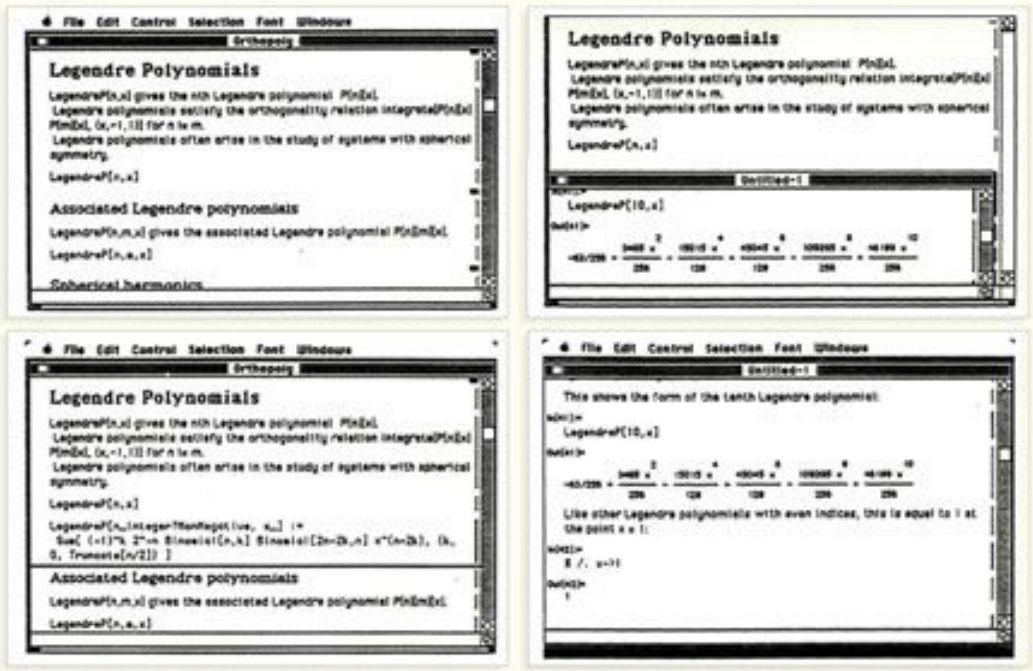
Instead of imagining that our main task is to instruct a computer what to do, let us concentrate rather on explaining to human beings what we want a computer to do.

— Donald E. Knuth, Literate Programming, 1984

A literate computing environment is one that allows users not only to execute commands interactively, but also to store in a literate document the results of these commands along with figures and free-form text.

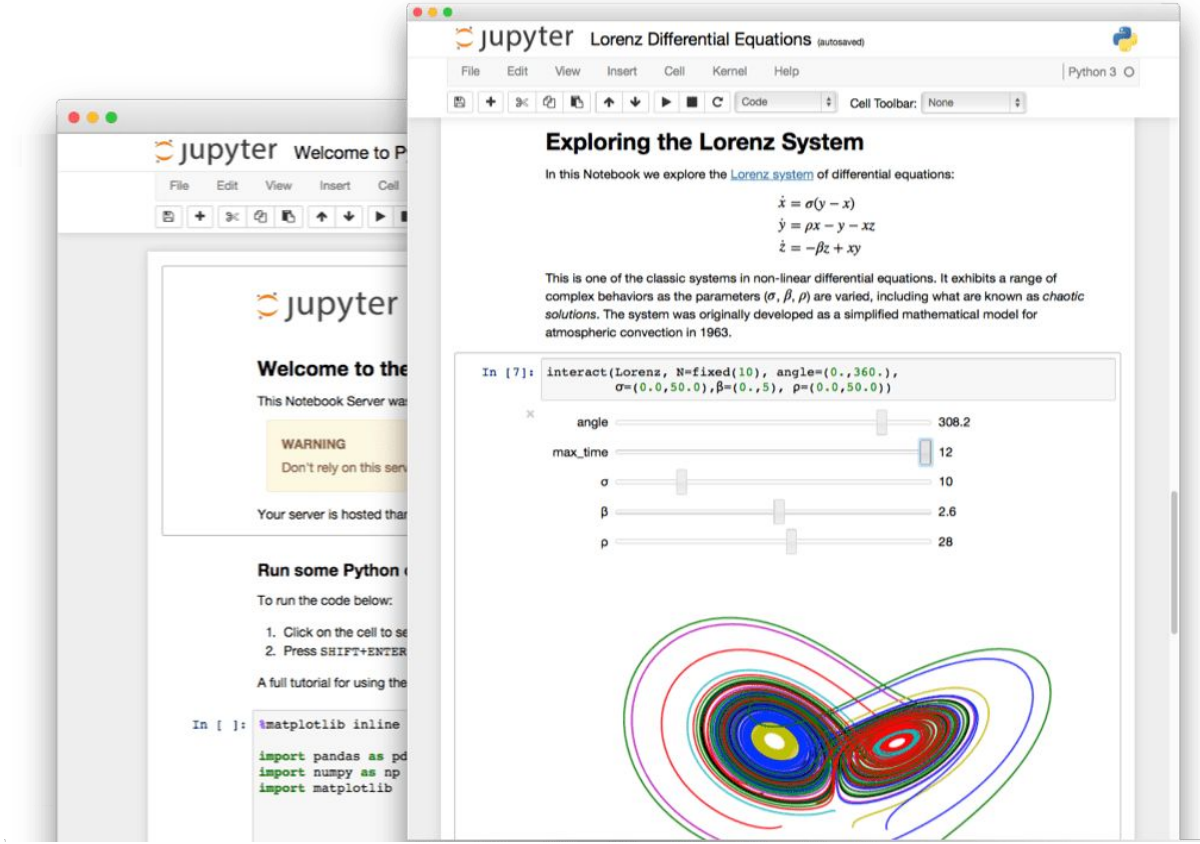
- Millman KJ and Perez F (2014)

And the concept was already in place in the 1980s



Wolfram Mathematica notebook (1987)

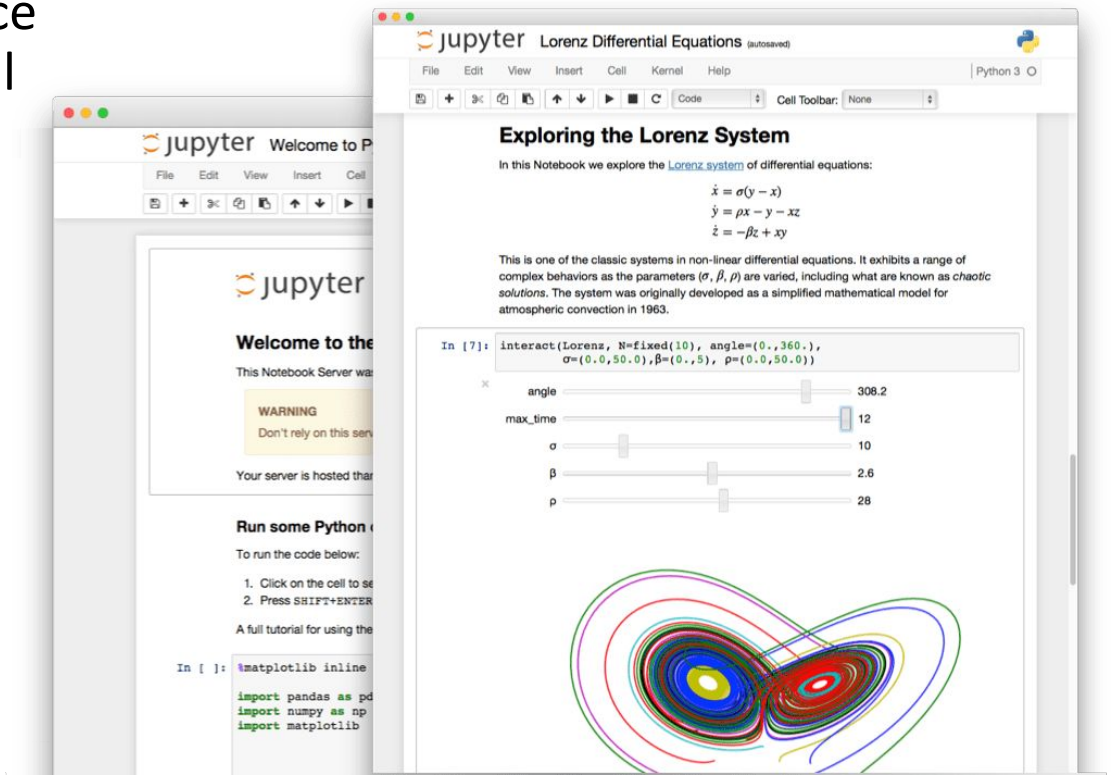
What does it look like now ?



Interactive programming interface allowing to combine both natural and computer languages.

In one file:

- Explanations
- Code
- Results
- Graphs and plots



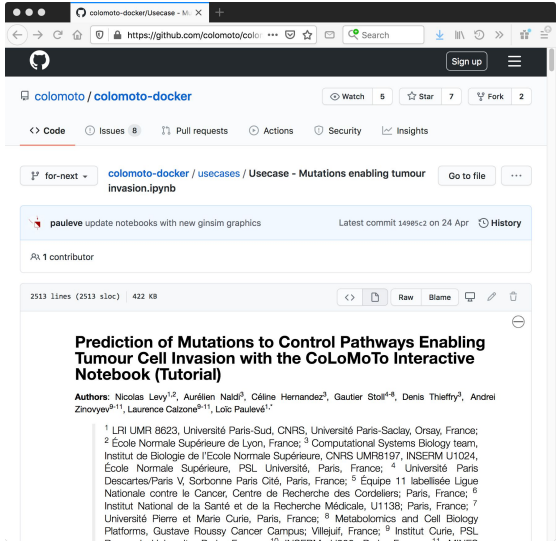
Why using literate programming frameworks ?

Use cases:

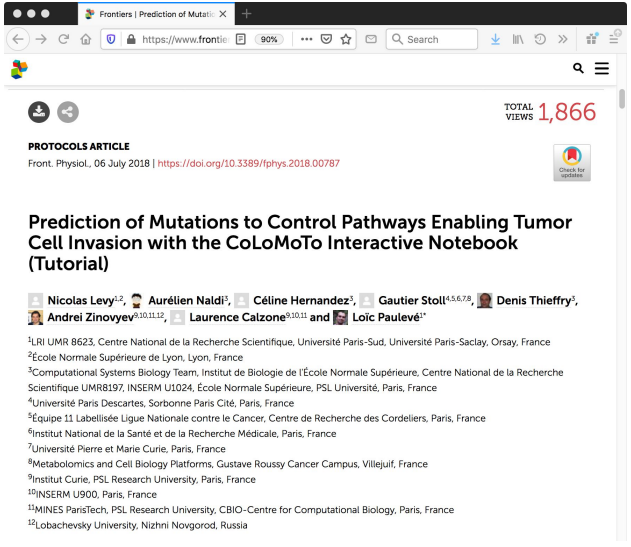
- Labbook
- Day to day analyses
- Analysis reports
- Writing scientific article

Example of an article entirely written using a notebook

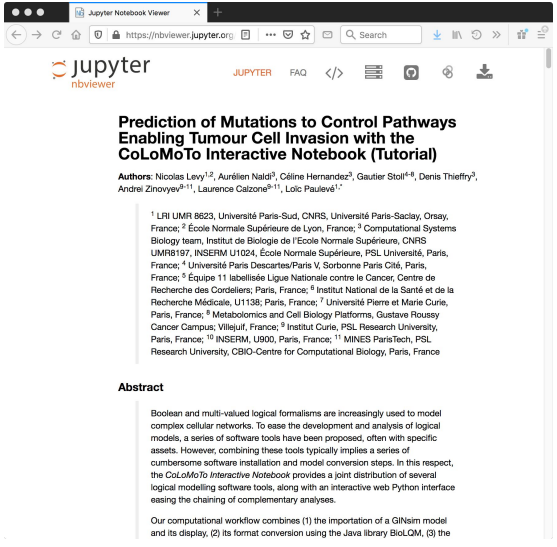
File (on a repository)



Published article



Executable file



DOI:10.3389/fphys.2018.00787

Towards the end of scientific articles?

omic analysis of elongated skulls
ensive female-biased immigration
arly Medieval Bavaria

Krishna R. Veeramah[a], Andreas Rott[b,1], Melanie Groß[c,1], Lucy van Dorp[d], Saioa López[e], Karola Kirsanow[f], Christian Sell[f], Jens Blöcher[f], Daniel Wegmann[f,g], Vivian Link[f,g], Zuzana Hofmanová[f,g], Joris Peters[h,i], Bernd Trautmann[h], Anja Gairhos[j], Jochen Haberstroh[j], Bernd Päffgen[k], Garrett Hellenthal[d], Brigitte Haas-Gebhard[l], Michaela Harbeck[b,2,3], and Joachim Burger[c,2,3]

[a]Department of Ecology and Evolution, Stony Brook University, Stony Brook, NY 11794-5245; [b]State Collection for Anthropology and Palaeoanatomy, Bavarian Natural History Collections, 80333 Munich, Germany; [c]Palaeogenetics Group, Institute of Organismic and Molecular Evolution, Johannes Gutenberg University Mainz, 55099 Mainz, Germany; [d]UCL Genetics Institute, Department of Genetics, Evolution and Environment, University College London, WC1E 6BT London, United Kingdom; [e]Cancer Institute, University College London, WC1E 6DD London, United Kingdom; [f]Department of Biology, University of Fribourg, 1700 Fribourg, Switzerland; [g]Swiss Institute of Bioinformatics, 1700 Fribourg, Switzerland; [h]ArchaeoBioCenter and Institute for Palaeoanatomy, Domestication Research and the History of Veterinary Medicine, Ludwig Maximilian University, 80539 Munich, Germany; [i]Bavarian State Archaeological Collection, 80538 Munich, Germany; [j]Bavarian State Department of Monuments and Sites, 80539 Munich, Germany; and [k]Institute of Prehistoric and Protohistoric Archaeology, Ludwig Maximilian University, 80799 Munich, Germany

Edited by Eske Willerslev, University of Copenhagen, Copenhagen, Denmark, and approved January 30, 2018 (received for review November 21, 2017)

Modern European genetic structure demonstrates strong correlations with geography, while genetic analysis of prehistoric humans has indicated at least two major waves of immigration from outside the continent during periods of cultural change. However, population-level genome data that could shed light on the demographic processes occurring during the intervening periods have been absent. Therefore, we generated genomic data from 41 individuals dating mostly to the late 5th/early 6th century AD from present-day Bavaria in southern Germany, including 11 whole genomes (mean depth 5.56×). In addition we developed a capture array to sequence neutral regions spanning a total of 5 Mb and 486 functional polymorphic sites to high depth (mean 72×) in all individuals. Our data indicate that while men generally had

to form in the 5th century AD, and that it emanated from a combination of the romanized local population of the border province of the former Roman Empire and immigrants from north of the Danube (2). While the Baiuvarii are less well known than some other contemporary groups, an interesting archaeological feature in Bavaria from this period is the presence of skeletons with artificially deformed or elongated skulls (Fig. 1A).

Artificial cranial deformation (ACD), which is only possible during early childhood, is a deliberate and permanent shaping of the head performed with great effort. In some societies reshaping the human skull has been seen as an ideal of beauty, while it

POPULATION BIOLOGY

Significance

PNAS / Richard Goerg / Getty / The Atlantic

SCIENCE

THE SCIENTIFIC PAPER IS OBSOLETE

Here's what's next.

By James Somers

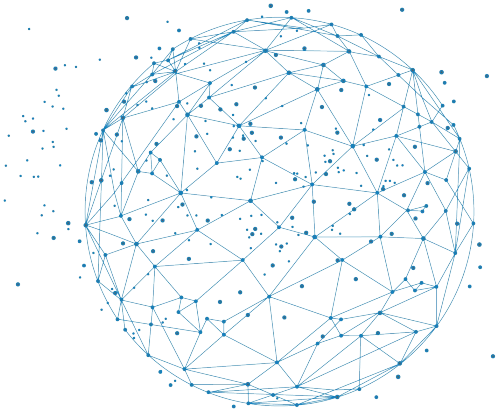
APRIL 5, 2018

SHARE SAVE

THE SCIENTIFIC paper—the actual form of it—was one of the enabling inventions of modernity. Before it was developed in the 1600s, results

We need to be careful

Markup / Markdown



Definition

A markup language uses tags to define elements within a document.

Three different types and usage

- Presentational (used by traditional word-processing systems)
  - Markup is invisible
- Procedural, provides instructions to process the text (e.g. TeX, PostScript)
  - Markup is visible and can be directly manipulated by the author.
- Descriptive, to label documents parts (e.g. LaTeX, HTML, XML...)
  - Emphasizes the document structure.

Example

Example in HTML

```
<h1>Heading</h1>
<h2>Sub-heading</h2>
<a href="www.webpage.com">Link</a>
<ul>
  <li>List-item1</li>
  <li>List-item2</li>
  <li>List-item3</li>
</ul>
```

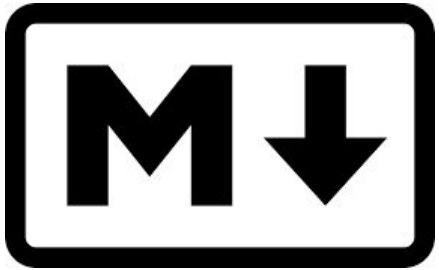
Heading

Sub-heading

Link

- List-item1
- List-item2
- List-item3

Markdown is a Lightweight markup language



Designed to be :

- easy to write using any generic text editor (plain-text-formatting syntax)
- easy to read in its raw form

From GitHub's help page
https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax

Example in markdown

```
# Heading

## Sub-heading

### Another deeper heading

A [link](http://example.com).

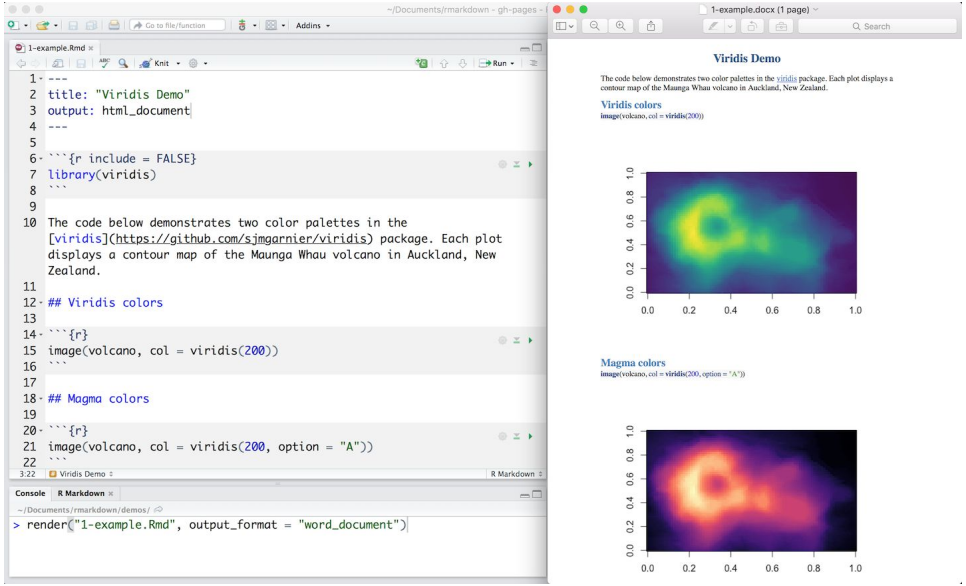
Text attributes _italic_, *italic*, **bold**, `monospace`.

Bullet list:

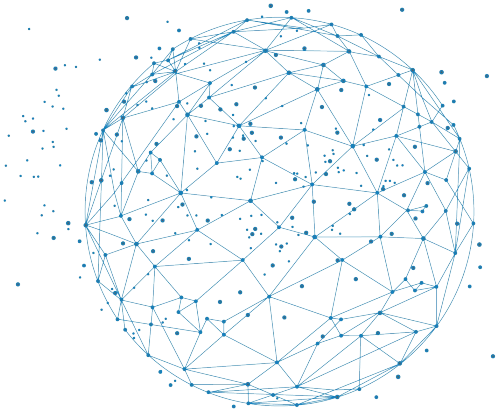
  * apples
  * oranges
  * pears
```

But how is this useful for literate programming?

When you want to alternate between code (to be interpreted) and formatting information, you precisely need a lightweight language for the formatting part.



Best practices



Best practices

Journal of Data Mining and Digital Humanities

EPI sciences overlay journals

Notebook and Open science : toward more FAIR play

Mariannig Le Béchec[1]*, Célya Gruson Daniel[2,3], Clémence Lascombes[3], Émilien Schultz[4]

1 CREM, Université de Lorraine

2 Inno[3]

3 COSTECH – Université de Technologie de Compiègne

4 CREST – GENES/IPP

*Corresponding author: Mariannig Le Béchec mariannig.le-bechec@lorraine.fr

Abstract

Notebooks are now commonly used in digital research practices. Despite their increasing ubiquity, the characteristics, roles, and uses associated with notebooks have seldom been studied from a from a social science perspective. In this article, we present an overview of the available empirical work on notebooks in order to describe existing practices, typologies crafted to grasp their diversity, and their limitations when used in data analysis workflows. Following this review, which highlights a focus of studies on interactive computational notebooks specifically within data science rather than research practices in academic contexts, we discuss the role of notebooks as a vector and lever for the FAIR (Findable, Accessible, Interoperable, Reusable) principles associated with open science.

keywords

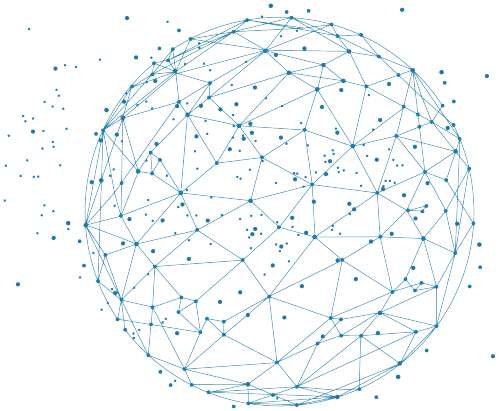
notebook, literate programming, jupyter, open science, FAIR

https://hal.science/hal-04549986/document

General categories	Best practices
Make your analysis traceable and reproducible	• Use a version control system to manage project dependency • Manage project dependencies • Provide applications without third-party dependencies • Put imports at the beginning of the file • Ensure that the entire code functions correctly, not just the modified part
Write quality code (i.e. code that can be easily shared and reused)	• Structure your code into modules (abstract the code into functions and place them in a dedicated module; place dependencies at the beginning of the notebook) • Test your code • Name your notebooks consistently • Respect standards • Use relative paths • Define requirements
Exploit the paradigm of literate programming	• Document your code for yourself and others • Use Markdown headings to structure your notebook
Keep your notebook clear and concise	• Keep your notebook clear • Keep your notebook concise
Differentiate between artefacts produced during development and production	• Differentiate between artefacts produced during development and production
Adopt open distribution	• Make your notebook available • Make your data available

Table 1: Catalogue of good practices specific to the use of notebooks extracted from the literature review

Limites



In this article, the authors highlight several limitations:

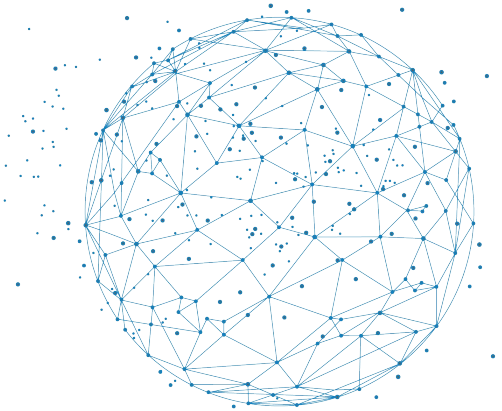
Reproducibility

- Lack of documentation
- Lack of consistency in cell execution
- Version tracking
- The quality of the code is often low

Interface dependent

Limited for large data sets

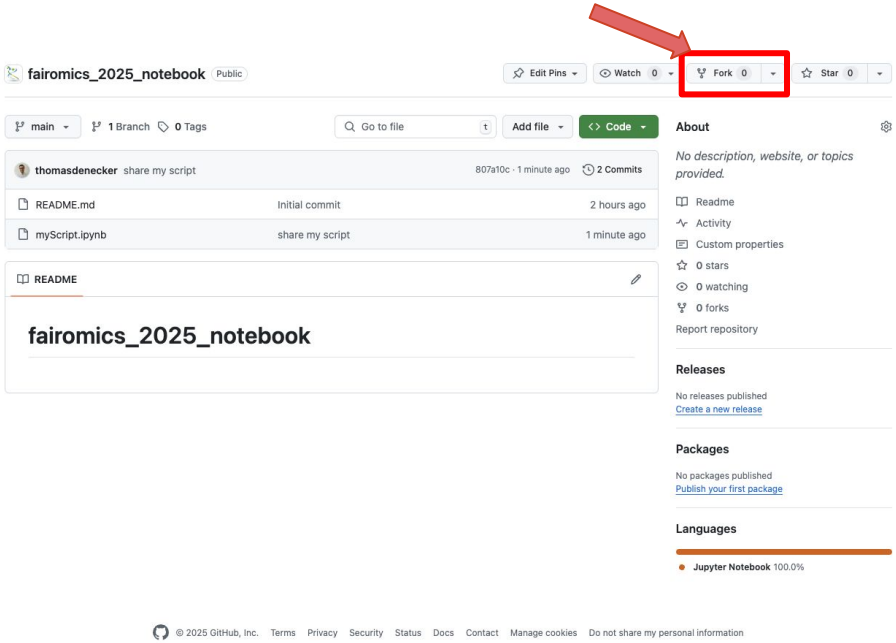
Let’s try



Step 1 - Fork the project

A fork is a personal copy of someone else's repository on your GitHub/Gitlab account.

It allows you to freely experiment with changes without affecting the original project.



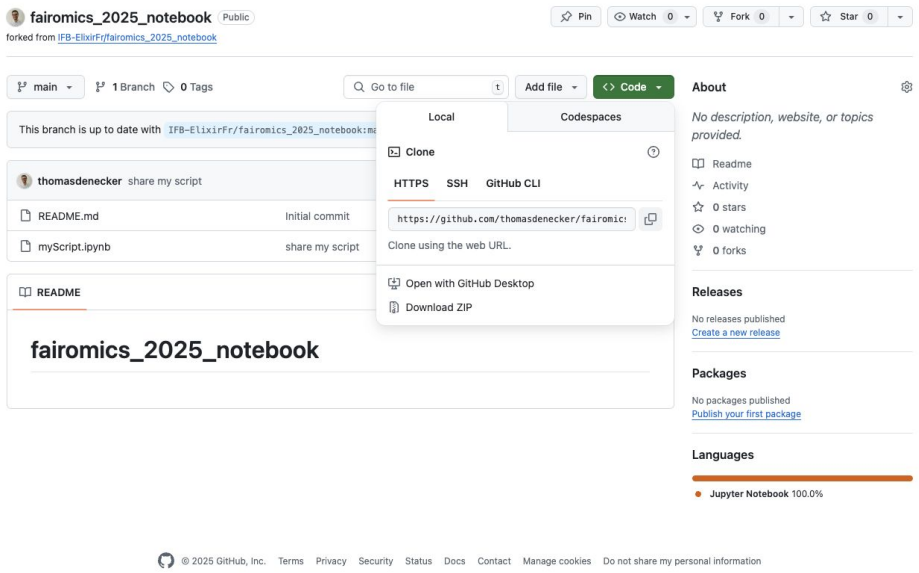
https://github.com/IFB-ElixirFr/fairomics_2025_notebook

Step 2 - Clone the project

`git clone` is a command used to download a remote Git repository to your local machine.

It:

- Copies the entire project history
- Creates a local working directory
- Sets up a link to the remote (usually called origin)



Let’s improve this notebook

myScript.ipynb

Code git Bash

```
[ ]: ORDERS=("order1" "order2" "order3")

mkdir -p fries burgers served

for order in "${ORDERS[@]}"; do
    echo "=== Processing $order ==="

    echo "Cutting potatoes for $order..."
    sleep 1
    echo "Potatoes cut for $order" > "fries/cut_$order.txt"

    echo "Frying fries for $order..."
    sleep 2
    echo "Fries cooked for $order" > "fries/fried_$order.txt"

    echo "Shaping burger patty for $order..."
    sleep 1
    echo "Raw patty shaped for $order" > "burgers/raw_patty_$order.txt"

    echo "Cooking patty for $order..."
    sleep 2
    echo "Patty cooked for $order" > "burgers/cooked_patty_$order.txt"

    echo "Assembling burger for $order..."
    sleep 1
    echo "Burger assembled for $order" > "burgers/burger_$order.txt"

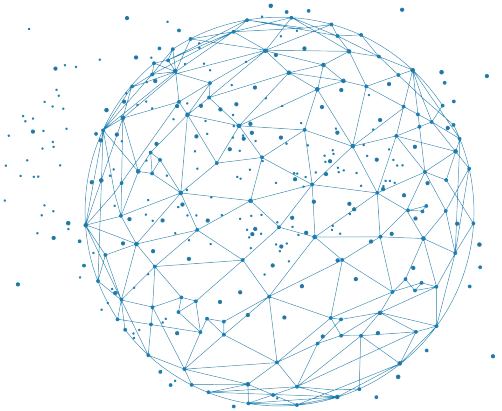
    echo "Serving burger and fries for $order..."
    sleep 1
    echo "Burger and fries served for $order" > "served/$order.txt"

    echo "=== $order is ready! ==="
    echo
done

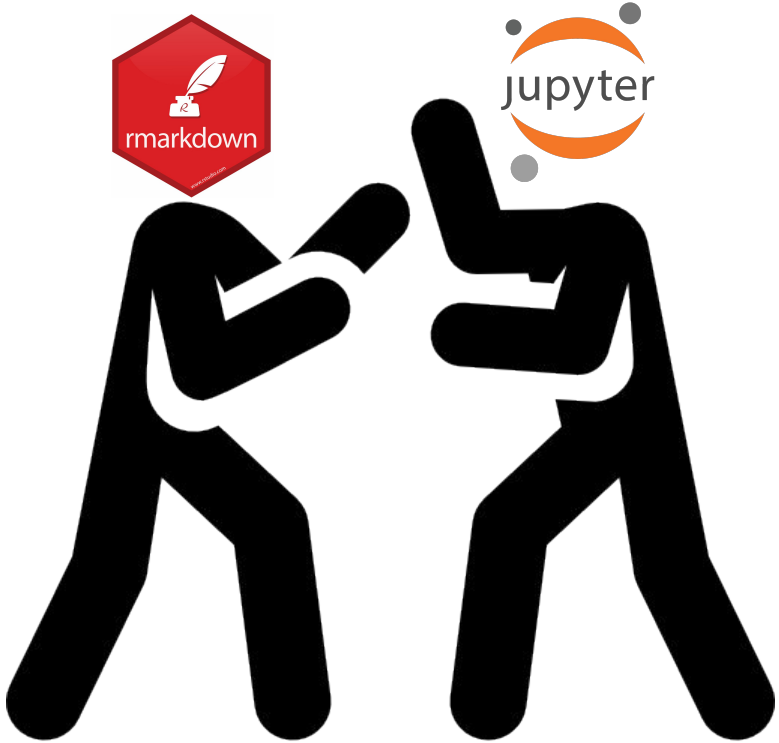
echo "All orders are complete."
```

Don’t forget to push your modifications ;)

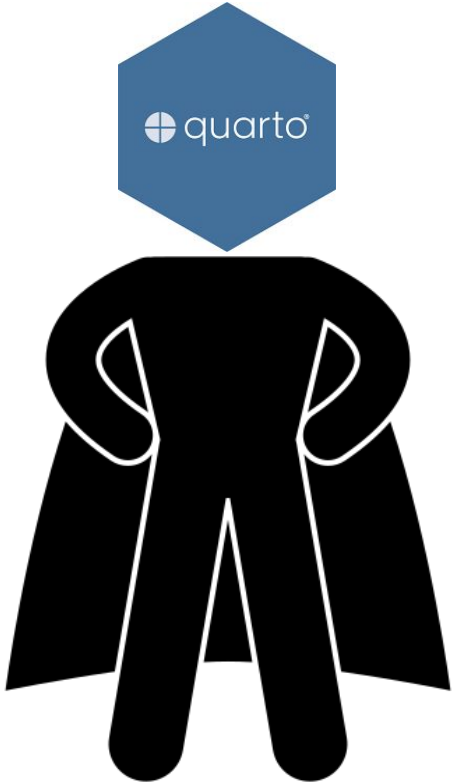
Alternative to Jupyter notebooks



Notebooks for bioinformatics



A replacement?



R Notebooks : Sweave (2002)

At the beginning, there was nothing. Then came Sweave.

Vol. 2/3, December 2002 28

Sweave, Part I: Mixing R and LaTeX

A short introduction to the Sweave file format and corresponding R functions

by Friedrich Leisch

This is the first article in a two part mini series on Sweave (Leisch, 2002), a tool that allows to embed the R code for complete data analyses in LaTeX documents. In this issue we will introduce the Sweave file format and R functions to process it, and demonstrate how to use Sweave as a reporting tool for literate statistical practice (Rossini, 2001). The companion article scheduled for the next issue of R News will concentrate on how to use files in Sweave format to write primers or manuals for R packages that can be automatically checked for syntax errors in the code or inconsistencies between examples and implementation.

The traditional way of writing a report as part of a statistical data analysis project uses two separate steps: First, the data are analyzed, and afterwards the results of the analysis (numbers, graphs, ...) are used as the basis for a written report. In larger projects the two steps may be repeated alternately, but the basic procedure remains the same. R supports this in a number of ways: graphs can be saved as PDF, EPS, or WMF which in turn can be included in LaTeX or Word documents. LaTeX tables can be created by specifying the columns and row separators in `write.table()` or using the package **xtable**. The basic paradigm is to write the report around the results of the analysis.

The purpose of Sweave is to create dynamic reports, which can be updated automatically if data or analysis change. Instead of inserting a prefabricated graph or table into the report, the master document contains the *R code* necessary to obtain it. When run through R, all data analysis output (tables, graphs, ...) is created on the fly and inserted into a final LaTeX document. The report can be automatically updated if data or analysis change, which allows for truly reproducible research.

A small example

Sweave source files are regular noweb files (Ramsey, 1998) with some additional syntax that allows control over the final output. Noweb is a simple literate programming tool which allows to combine program source code and the corresponding documentation into a single file. These consist of a sequence of code and documentation segments, called *chunks*. Different command line programs are used to extract the code ("*tangle*") or typeset documentation together with the code ("*weave*").

A small Sweave file is shown in Figure 1, which contains four code chunks embedded in a simple LaTeX document. '`<<...>>=`' at the beginning of a line marks the start of a code chunk, while a '`@`' at the beginning of a line marks the start of a documentation chunk. Sweave translates this into a regular LaTeX document, which in turn can be compiled by `latex` to Figure 2.

The code chunks

The main work of Sweave is done on the code chunks. All code chunks are evaluated by R in the order they appear in the document[1]. Within the double angle brackets we can specify options that control how the code and the corresponding output are rendered in the final document. The first code chunk (lines 5–8 in Figure 1) declares that neither the R code (`echo=false`) nor output (`results=hide`) shall be included. The purpose of this chunk is to initialize R by loading packages and data, we want to hide these technical details from the reader.

Let us skip the text in lines 10–19 for the moment and go directly to the next code chunk in lines 20–22. It uses the default settings for all options (nothing is specified within the double angle brackets): both input and output are shown to the user (see Figure 2), the chunk is rendered such that it emulates the R console when the code is typed at the prompt. All input and output are automatically encapsulated in verbatim-like environments.

The next code chunk can be found at lines 30–31. It uses the package **xtable** to pretty-print the coefficient matrix of the linear regression model. By specifying `results=tex` we tell Sweave that the output of this code chunk is regular TeX code and hence needs no protection by a verbatim environment.

The last code chunk in lines 36–38 is marked as a figure chunk (`fig=true`) such that Sweave creates EPS and PDF files corresponding to the plot created by the commands in the chunk. Furthermore, an `\includegraphics{}` statement is inserted into the LaTeX file. Options `width` and `height` are passed to R's graphics devices and determine the size of the figure in the EPS and PDF files.

In line 28 we use `\SweaveOpts{echo=false}` to modify the default for option `echo` to the value of `false` for all code chunks following, hence the code for the last two chunks is not shown in Figure 2. It has exactly the same effect as if we had included `echo=false` within the double angle brackets of the two chunks.

[1]There are ways to suppress evaluation or re-use chunks, which is beyond the scope of this article.


```
\documentclass[a4paper]{article}

\begin{document}

<<echo=false,results=hide>>=
library(lattice)
library(xtable)
data(cats, package="MASS")
@

\section*{The Cats Data}

Consider the \texttt{cats} regression example from Venables \& Ripley
(1997). The data frame contains measurements of heart and body weight
of \Sexpr{nrow(cats)} cats (\Sexpr{sum(cats$Sex=="F")} female,
\Sexpr{sum(cats$Sex=="M")} male).

A linear regression model of heart weight by sex and gender can be
fitted in R using the command
<<>>=
lm1 = lm(Hwt~Bwt*Sex, data=cats)
lm1
@
Tests for significance of the coefficients are shown in
Table~\ref{tab:coef}, a scatter plot including the regression lines is
shown in Figure~\ref{fig:cats}.

\SweaveOpts{echo=false}

<<results=tex>>=
xtable(lm1, caption="Linear regression model for cats data.", label="tab:coef")
@

\begin{figure}
  \centering
<<fig=true,width=12,height=6>>=
lset(col.whitebg())
print(xyplot(Hwt~Bwt|Sex, data=cats, type=c("p", "r")))
@
  \caption{The cats data from package MASS.}
  \label{fig:cats}
\end{figure}

\end{document}
```

Figure 1: A minimal Sweave file: `example.Snw`.

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	2.9813	1.8428	1.62	0.1080
Bwt	2.6364	0.7759	3.40	0.0009
SexM	−4.1654	2.0618	−2.02	0.0453
Bwt:SexM	1.6763	0.8373	2.00	0.0472

Table 1: Linear regression model for cats data.

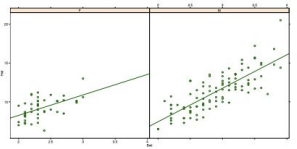


Figure 1: The cats data from package MASS.

The Cats Data

Consider the `cats` regression example from Venables & Ripley (1997). The data frame contains measurements of heart and body weight of 144 cats (47 female, 97 male).

A linear regression model of heart weight by sex and gender can be fitted in R using the command

```
> lm1 = lm(Hwt ~ Bwt * Sex, data = cats)
> lm1

Call:
lm(formula = Hwt ~ Bwt * Sex, data = cats)

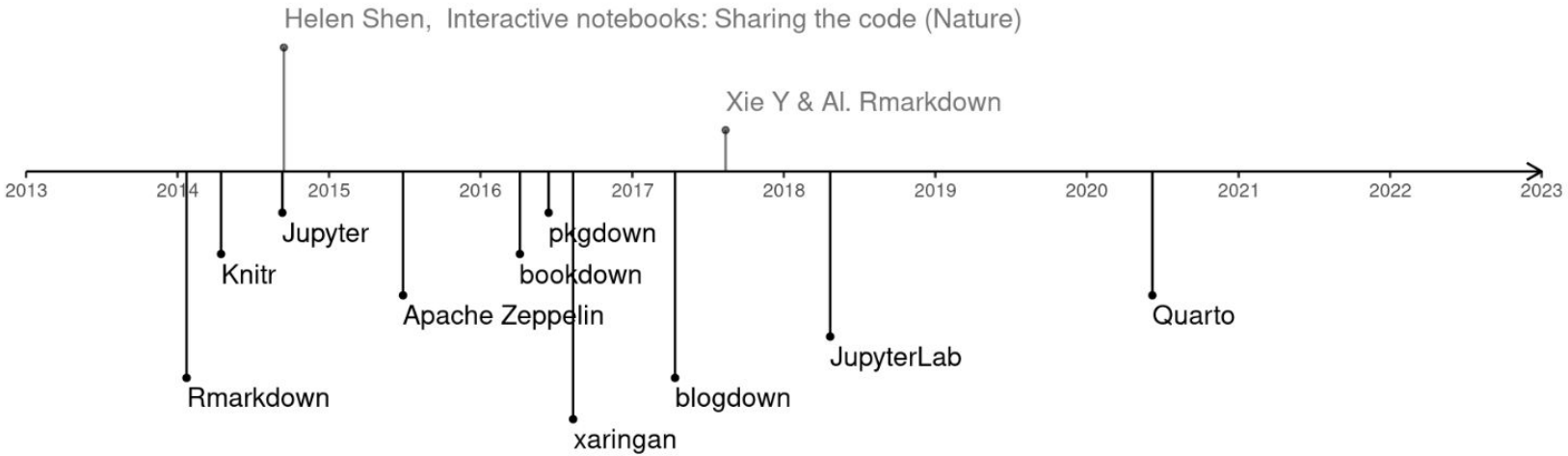
Coefficients:
(Intercept)          Bwt          SexM      Bwt:SexM
      2.981        2.636       -4.165        1.676
```

Tests for significance of the coefficients are shown in Table 1, a scatter plot including the regression lines is shown in Figure 1.

Figure 2: The final document is created by running `latex` on the intermediate file 'example.tex' created by `Sweave("example.Snw")`.

And people saw that the path would be long...

Appearance of packages allowing the creation of notebooks



https://camembr.quarto.pub/hello-quarto/#/les-packages

”The knitr package was designed to be a transparent engine for dynamic report generation with R, solve some long-standing problems in Sweave, and combine features in other add-on packages into one package”

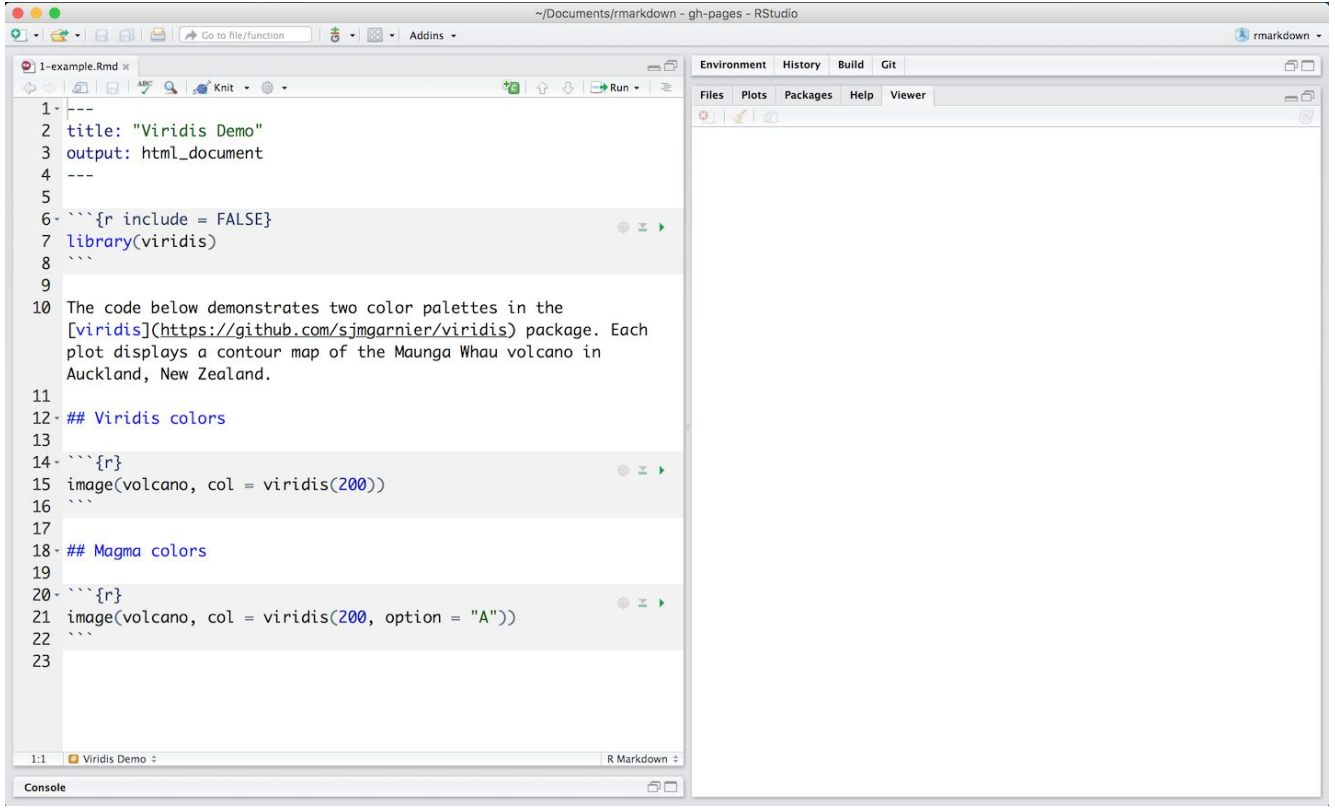
https://yihui.org/knitr/

”When you run render, R Markdown feeds the .Rmd file to knitr, which executes all of the code chunks and creates a new markdown (.md) document which includes the code and its output.

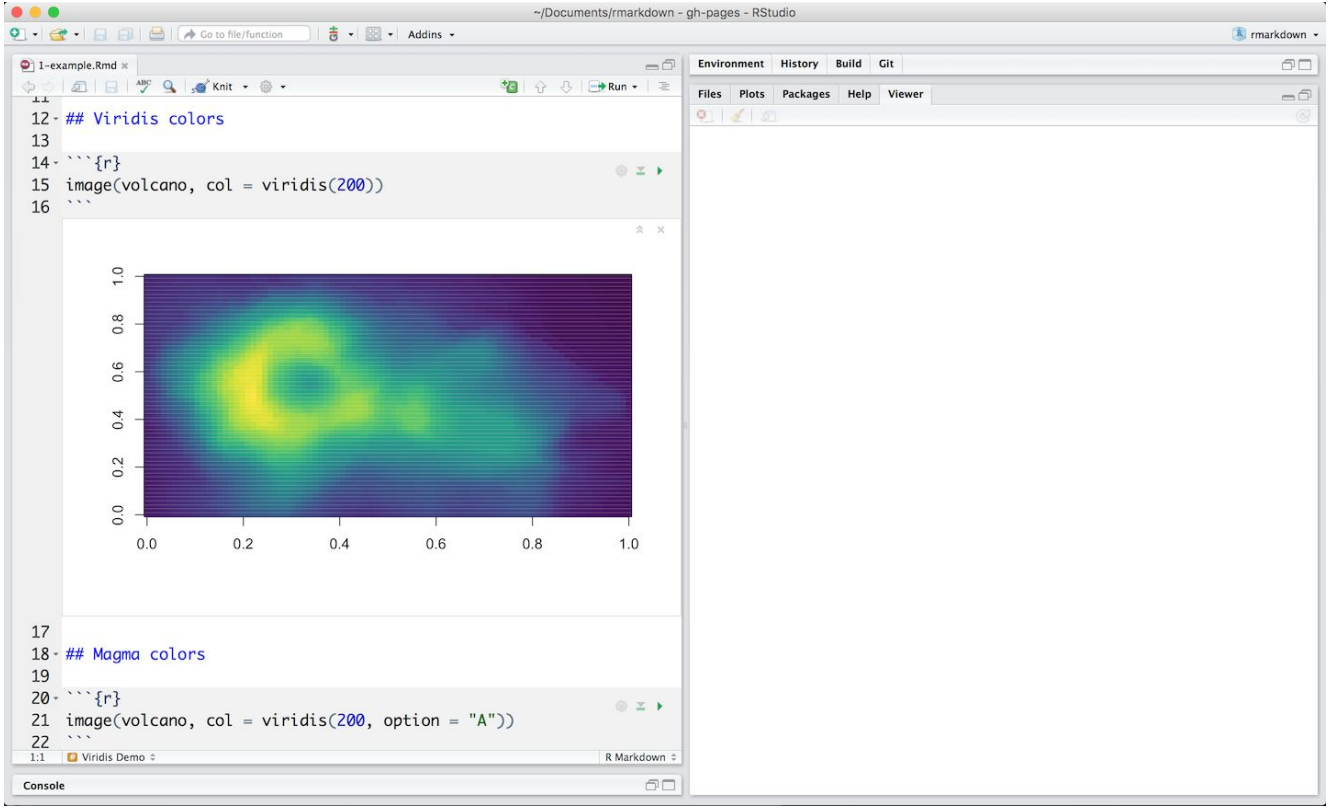
The markdown file generated by knitR is then processed by pandoc which is responsible for creating the finished format.”

https://rmarkdown.rstudio.com

R Notebooks : RMarkdown (2012)



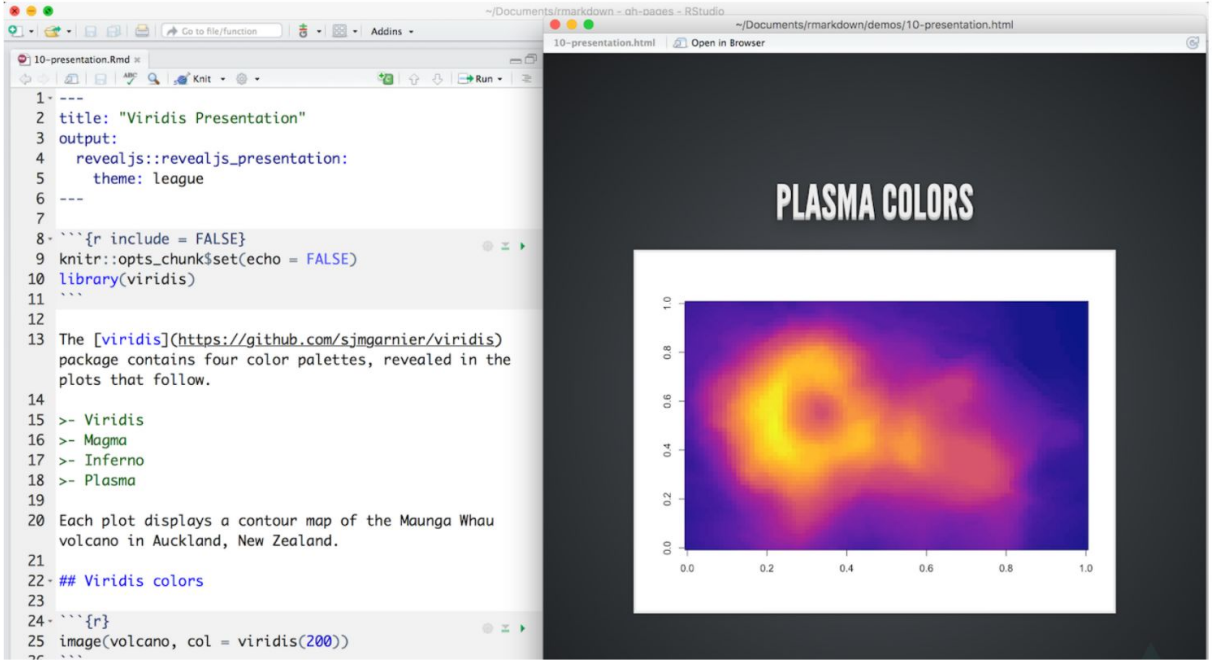
R Notebooks : RMarkdown (2012)



R Notebooks and more…

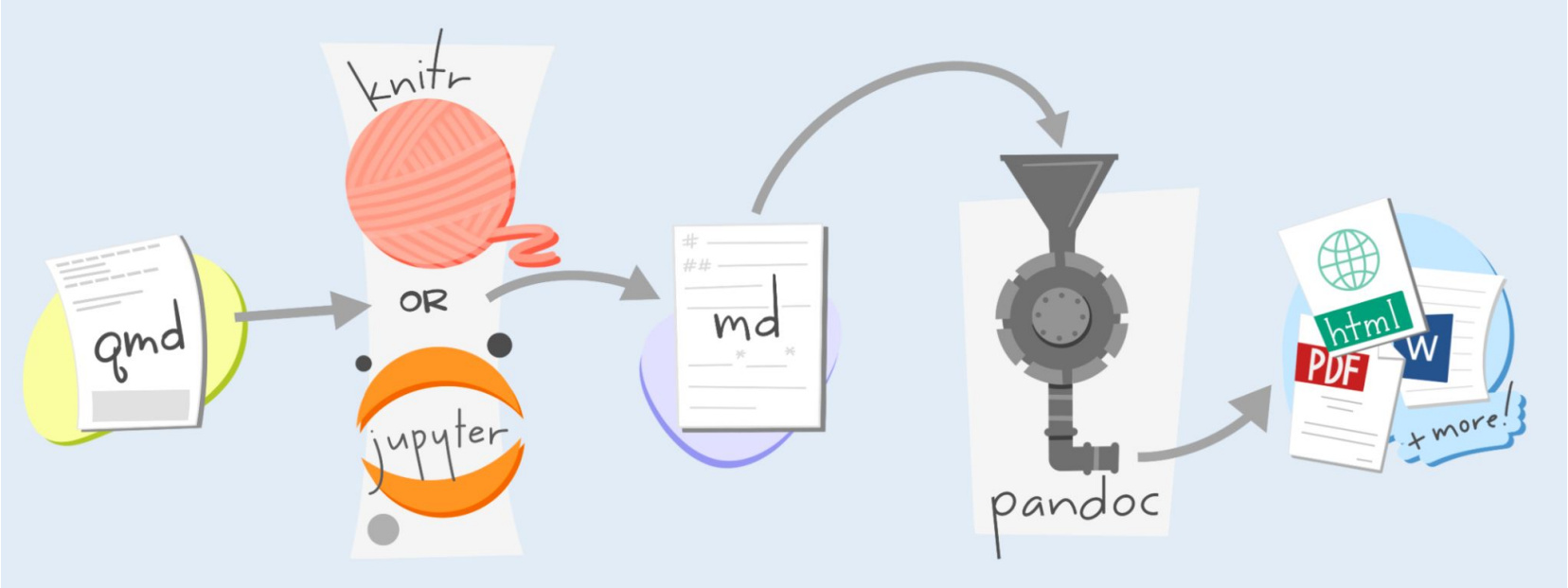
- Markdown Basics
- Output Formats
- Notebooks
- **Slide Presentations**
- Dashboards
- Websites
- Interactive Documents
- Cheatsheets

file below, which is available here on RStudio Cloud.



Quarto is an open-source scientific and technical publishing system where authors :

- Can use Jupyter notebooks or with plain text markdown in your favorite editor.
- Create dynamic content with Python, R, Julia, and Observable.
- Publish reproducible, production quality articles, presentations, websites, blogs, and books in HTML, PDF, MS Word, ePub, and more.
- Share results in a lot of publishing systems like GitHub.


knitr
qmd
OR
jupyter
md
pandoc
html
PDF
W
+ more!

With R

```
---
title: "ggplot2 demo"
author: "Norah Jones"
date: "5/22/2021"
format:
  html:
    fig-width: 8
    fig-height: 4
    code-fold: true
---

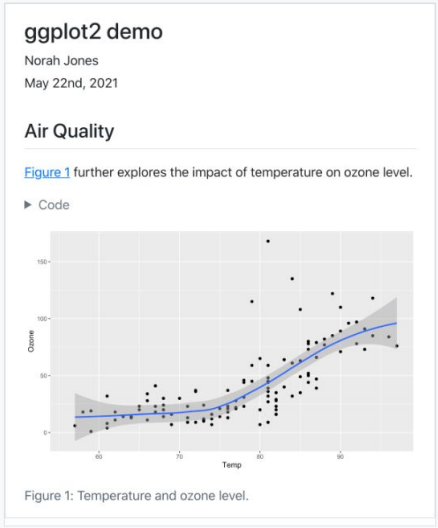
## Air Quality

@fig-airquality further explores the impact of
temperature on ozone level.

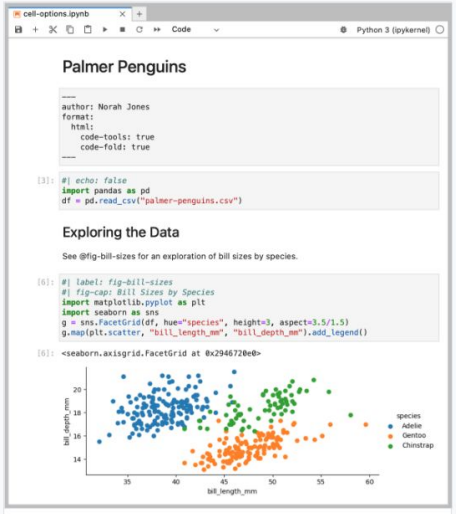
```{r}
#| label: fig-airquality
#| fig-cap: "Temperature and ozone level."
#| warning: false

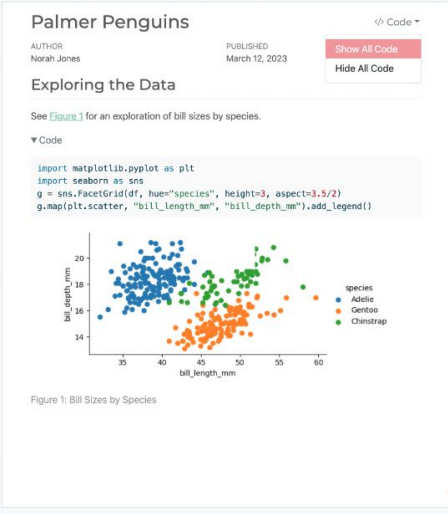
library(ggplot2)

ggplot(airquality, aes(Temp, Ozone)) +
  geom_point() +
  geom_smooth(method = "loess"
)
```
```



With Jupyter





Rmarkdown

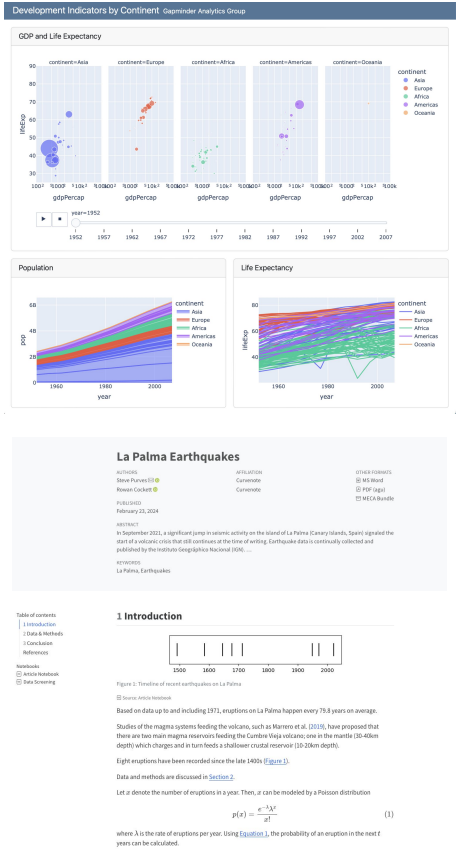
```
quarto render code/supplementary_material.Rmd --to html
quarto render code/supplementary_material.Rmd --to docx
```

Jupyter

```
quarto render code/supplementary_material.ipynb --to html
quarto render code/supplementary_material.ipynb --to docx
```

Quarto and more…

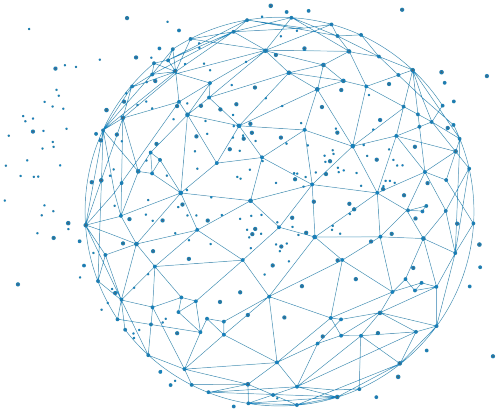
- Documents
  - HTML
  - PDF
  - MS Word
  - Typst
  - Markdown
  - All Formats
- Presentations
- Dashboards
- Websites
- Books
- Manuscripts
- Interactivity
- Publishing
- Projects
- Advanced



Comparaison

Method	Jupyter	Rmarkdown	Quarto
IDE	JupyterHub, JupyterLab	R, Rstudio	VS Code, Jupyter, Rstudio, Neovim, TextEditor
Code mixing	Limited	Yes	Yes
Format	ipynb	Rmd	Qmd, Rmd, ipynb
Output	Asciidoc, HTML, LaTeX, MD, PDF, RST, Slide (Reveal)	HTML, PDF, Docx, ODT, RTF, MD, Slides (Powerpoint, Reveal,...), Dashboard, ...	HTML, PDF, Docx, ODT, Epub, RTF, MD, Slides (Powerpoint, Reveal,...), Wiki (MediaWiki, ...), Book, and more !
Reproductibility	Easy	Easy	Yes (if done from the start)

Observable



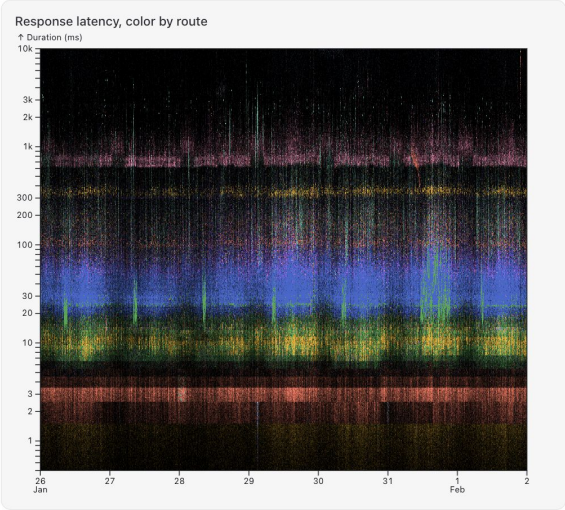
Observable Framework

Observable is an online platform that enables users to create, execute, and share interactive notebooks. These notebooks on Observable are based on JavaScript and allow users to combine code, visualizations, and text within a single interactive document.

Analyzing web logs

Web logs capture traffic metadata, such as the request time and route, how long the server took to respond, the response size, and so on. Analyzing web logs sheds light on both server performance and client behavior. Yet the common practice of summary statistics (*e.g.*, 95th-percentile latency) often hides interesting patterns! This is because performance varies wildly based on the nature of the request, and unusual clients such as bots can easily hide in a sea of "natural" traffic.

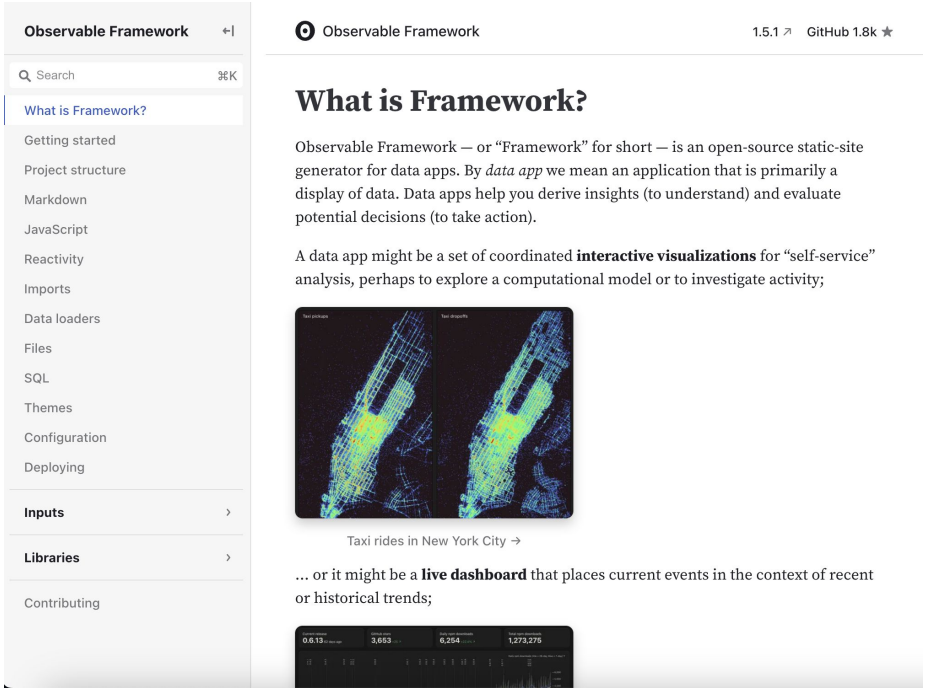
What if — instead of summarizing — we plotted *every* request as a dot with time along $x\rightarrow$ and latency (on a log scale) along $y\uparrow$?



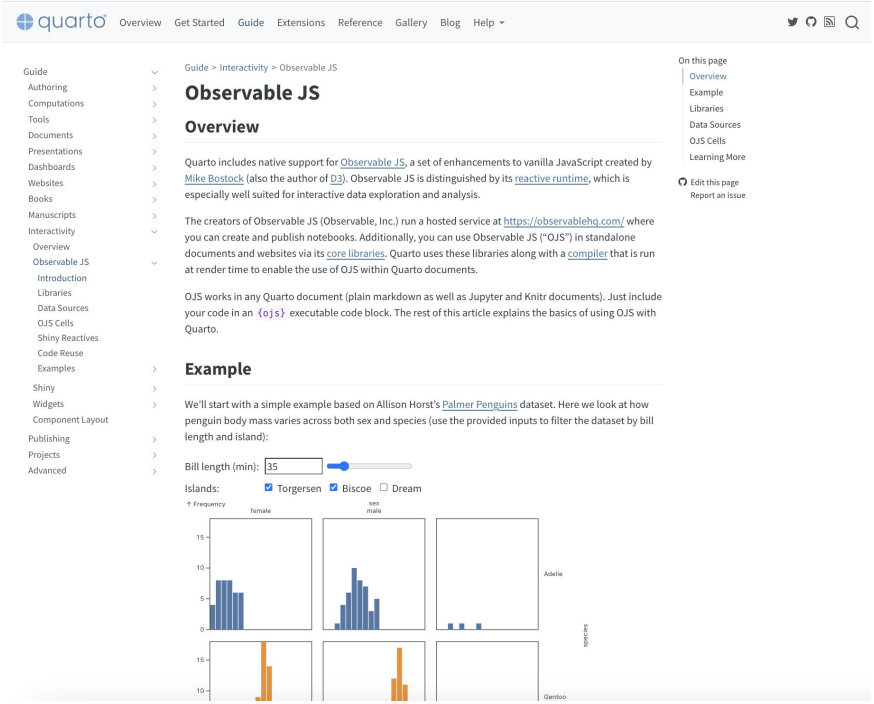
The plot above shows a sample of 7,633,176 requests to Observable servers over a 7-day period. Color encodes the associated route. Hover to see the route.

https://observablehq.com/framework/examples/api/

Official documentation



Via quarto



Thank you for your attention !

IFB
INSTITUT FRANÇAIS DE BIOINFORMATIQUE

elixir FRANCE

cnrs INRAE Inserm cea INVESTIR L'AVENIR