Introduction

Analytically reproducible documents

Hélène Langet

2025-02-04

Table of contents

1 Research = a dynamic process

- Research insights are typically disseminated through reports (e.g., scientific presentations, publications, etc), including a textual narrative detailing the research context, methods, and key findings, often supplemented with figures and tables to summarize results, and a final discussion, with findings serving as evidence to support conclusions and recommendations ;
- Research is an **iterative** and **dynamic** process, meaning there are no **final** or **definitive** results or reports ;
- In addition, we continuously build upon the work of others to generate new insights and discoveries.

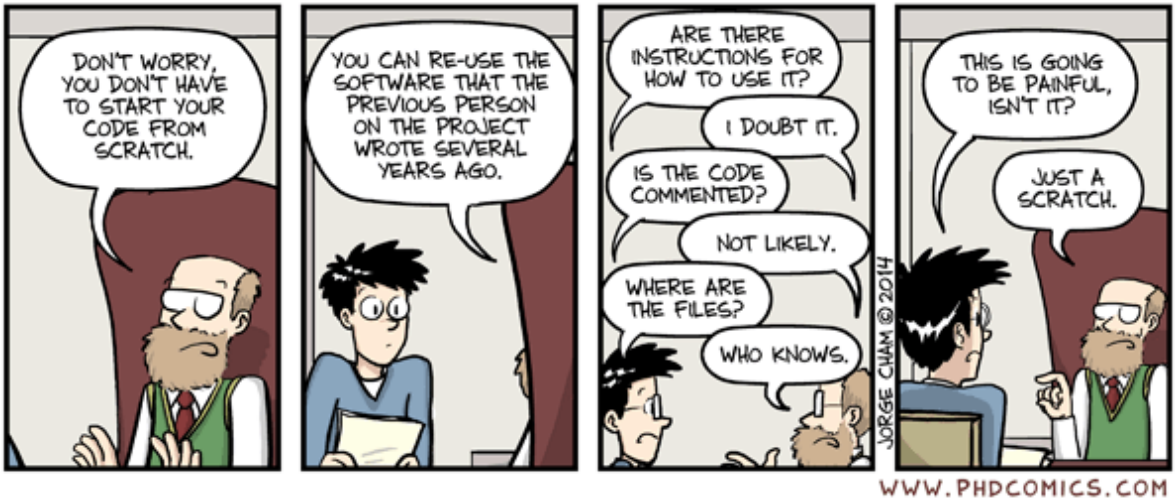


Figure 1: Image reproduced from Jorge Cham at PhDComics.

2 Reproducibility in research

Achieving reproducibility requires clear access to the underlying **data**, the **code** used for analysis, and the **results** produced. It also depends on documenting the **tools**, such as software and libraries, alongside the **computational environment**, including hardware configurations and operating systems (1).

3 Analytically reproducible documents

Analytically reproducible documents typically contain 3 main types of content, integrating code and natural language in a way that is called "*literate programming*" (2).

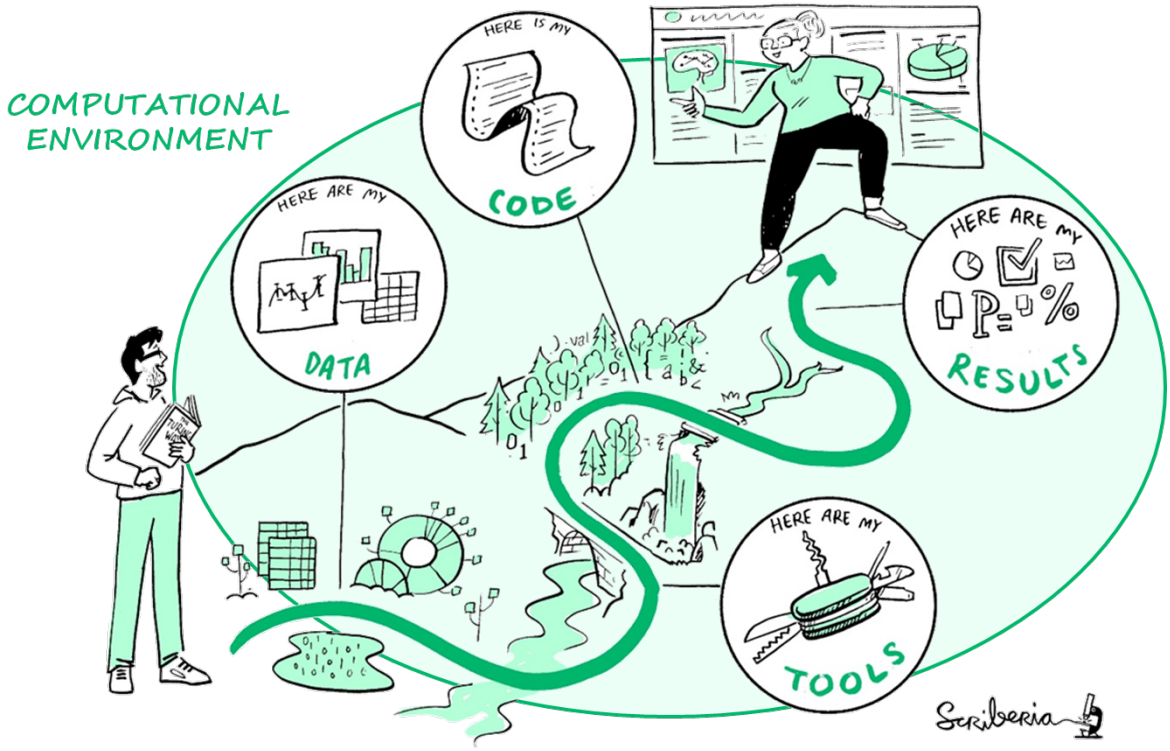


Figure 2: Illustration highlighting the key components of reproducibility in research, including data, code, results, tools, and the computational environment. This illustration is adapted from the one created by Scriberia with The Turing Way community, and used under a CC-BY 4.0 licence. DOI: 10.5281/zenodo.3332807.

3.1 Formatted text

These are languages that can be written using any **plain text editor**. They use *markup* elements to define how text should be displayed or printed.

3.1.1 HTML

HTML is used to structure content on the web.

```
<b>This text will be displayed in bold</b>
```

3.1.2 LaTeX

LaTeX is used for academic and technical documents.

```
\textbf{This text will be displayed in bold}
```

3.1.3 Markdown

Markdown is a lightweight markup language.

```
**This text will be displayed in bold**
```

3.2 Code commands

Different programming languages allow us to execute code to generate results or perform tasks.

3.2.1 R

````
```{r}
library(ggplot2)
data.frame(country=c("Nigeria","Kenya","India"), prevalence=c(14.5,9.2,3.5)) |>
  ggplot(aes(x=country, y=prevalence)) +
  geom_bar(stat="identity", fill="steelblue")
```
````

3.2.2 Python

````
```{r}
library(reticulate)
Sys.setenv(RETICULATE_PYTHON = "C:/ProgramData/anaconda3/python.exe")
```
````

````
```{python}
import matplotlib.pyplot as plt
plt.bar(['Nigeria', 'Kenya', 'India'], [14.5, 9.2, 3.5], color='steelblue')
plt.show()
```
````

3.2.3 Observable JS

````
```{ojs}
BarChart({x: ["Nigeria", "Kenya", "India"], y: [14.5, 9.2, 3.5], yLabel: "Prevalence (%)"}
```
````

3.3 Code outputs

The output from executing code often results in visualizations or printed results. Below are the corresponding outputs for each language:

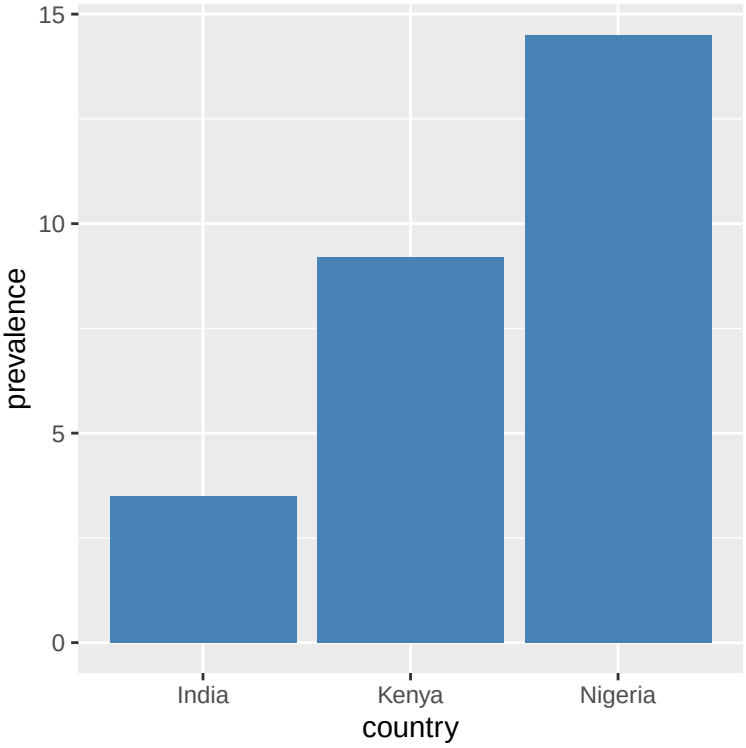
```
Call:  glm(formula = confirmed ~ age, family = binomial, data = df)

Coefficients:
(Intercept)          age
   1.312275     0.001292

Degrees of Freedom: 65668 Total (i.e. Null);  65667 Residual
Null Deviance:      66000
Residual Deviance: 66000    AIC: 66000
```

c1	c2
setosa	5.1
setosa	4.9
setosa	4.7

c1	c2
setosa	4.6
setosa	5.0



Documents can be rendered in different type of outputs e.g., MS Word, PDF, HTML, PowerPoint, etc.

Pandoc is a powerful open-source tool that enables seamless **conversion** between various document formats, making it an essential resource for working with markup languages like Markdown, HTML, and LaTeX. By using Pandoc, you can easily transform a document written in one markup language into a wide range of output formats, including MS Word, PDF, HTML, PowerPoint, and more, without needing to manually adjust formatting or structure.

This versatility makes Pandoc particularly valuable in workflows that involve literate programming. Pandoc also supports templates and extensions, allowing users to customize the output to meet specific stylistic or formatting requirements, simplifying the process of producing polished, professional documents.

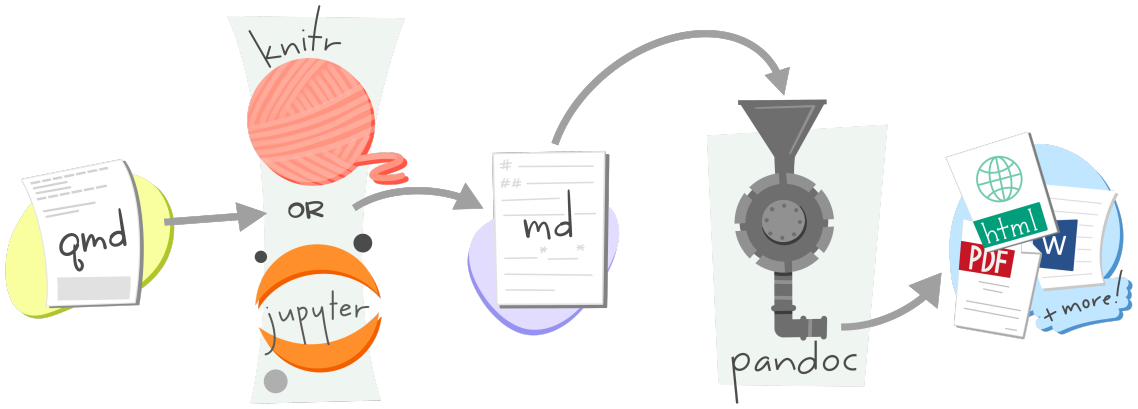


Figure 3: Process of transforming a Quarto document from its source format to the final rendered output. Artwork by Allison Horst.

Language	Commands / File types
Stata	**dyndoc command** putpdf command putdocx command putwrap command stmd command markstat command Limited capacity compared to Quarto
R	R Markdown (R + Markdown) R Sweave (R + LateX) R HTML (R + HTML) **Quarto (*.qmd)**
Python	Jupyter Notebook **Quarto (*.qmd)**

Figure 4: Existing tools for writing analytically reproducible documents

4 Existing tools for writing analytically reproducible documents

5 Quarto

- Quarto is the successor to R Markdown, but is not tied to the R language.
- Quarto files have a `.qmd` extension.

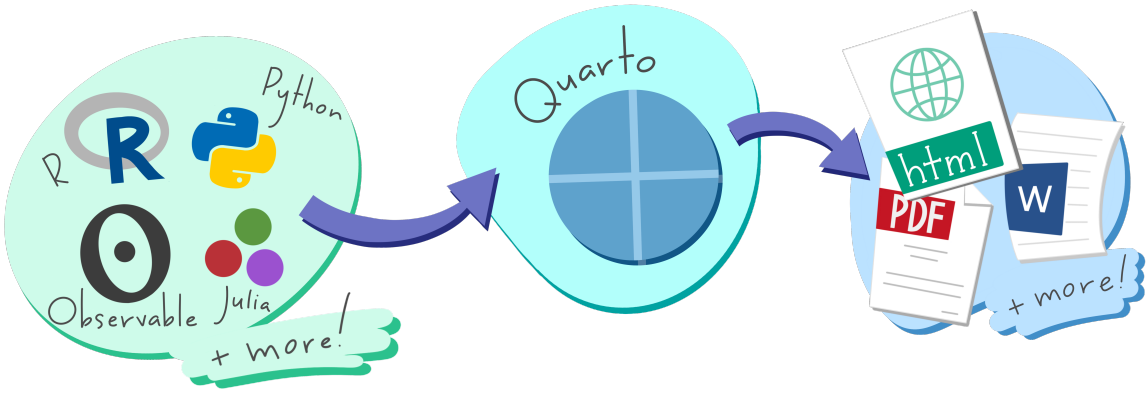
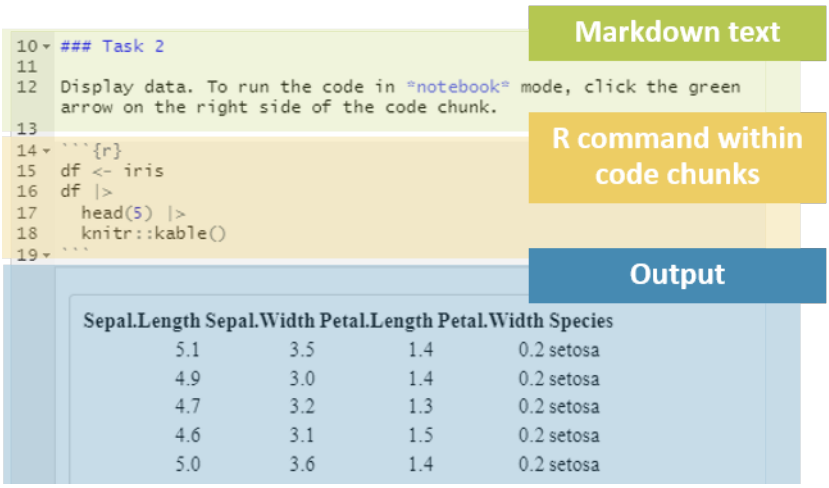
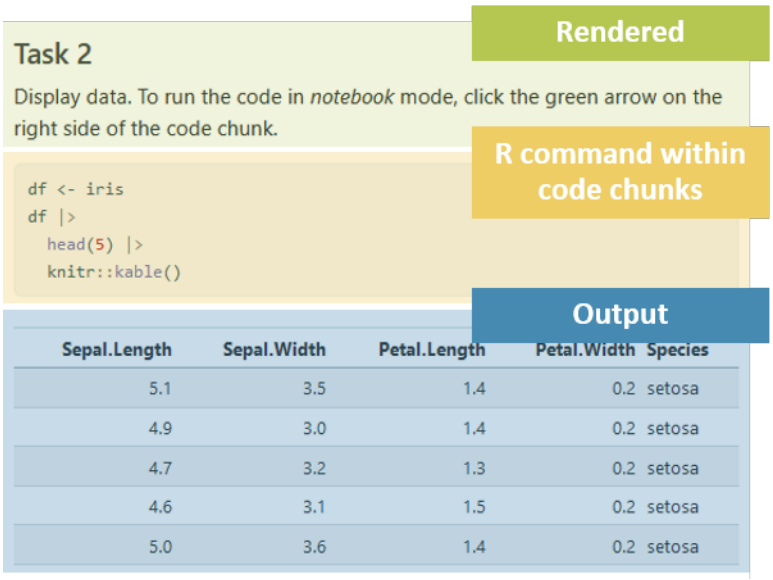


Figure 5: Artwork by Allison Horst.

5.1 Source document

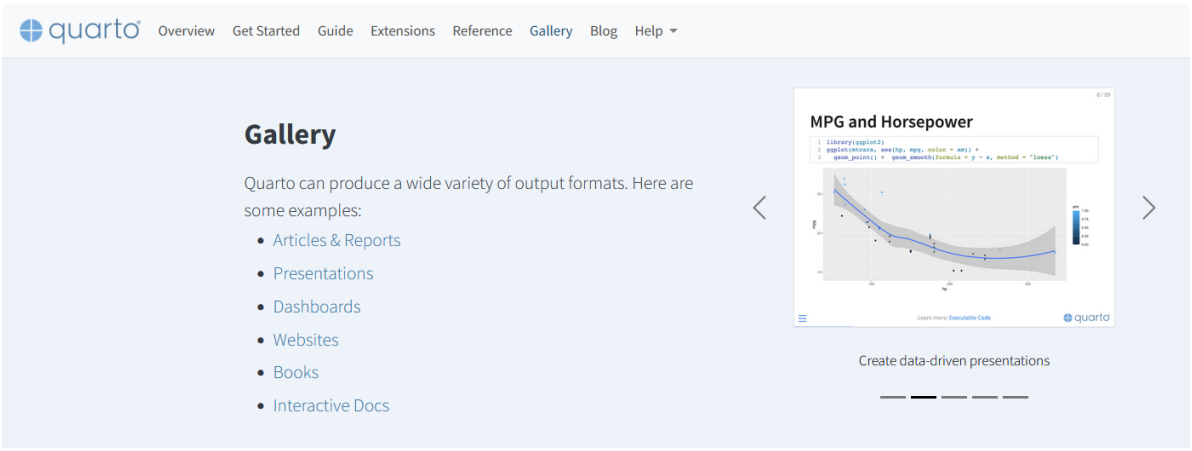


5.2 Rendered output



5.3 Quarto rendered outputs

- Quarto documents can be rendered into to many report formats including HTML, Word document and many more
- List of supported formats



5.4 Engines

An engine refers to the software or system that executes the embedded code within a document. The engine takes the code chunks written in a specific programming language (e.g., R, Python, or Julia), runs them, and returns the output, which is then incorporated into the rendered document.

Both knitr and Jupyter serve as engines to execute code embedded within a document, but they work in different programming environments.

5.4.1 knitr

This R package will read the code chunks, execute it, and 'knit' it back into the document. This is how tables and graphs are included alongside the text.

5.4.2 Juypter

Jupyter is a popular engine for running Python code interactively. It supports multiple programming languages, but Python is the most common.

6 To go further on reproducibility

- MOOC Reproducible research I - Methodological principles for transparent science (in French)
- MOOC Reproducible research II - Practices and tools for managing computations and data

1. The Turing Way Community. The Turing Way: A handbook for reproducible, ethical and collaborative research (1.0.2). https://book.the-turing-way.org/index.html; 2022.

2. Knuth DE. Literate Programming. The Computer Journal [Internet]. 1984 Jan;27(2):97–111. Available from: https://doi.org/10.1093/comjnl/27.2.97

3. Batra, Neale and Mousset, Mathilde and Spina, Alex and Florence, Isaac and Coyer, Liza and others. The Epidemiologist R Handbook. https://epirhandbook.com; 2021.

4. The Graph Courses Team. Websites and dashboards with Quarto. https://thegraphcourses.org/courses/websites-and-dashboards-with-r/; 2023.