Reproducible documents with Quarto

Scientific workflows: Tools and Tips 🛠️

2023-05-11

Selina Baldauf // Quarto

What is this lecture series?

Scientific workflows: Tools and Tips 🛠️

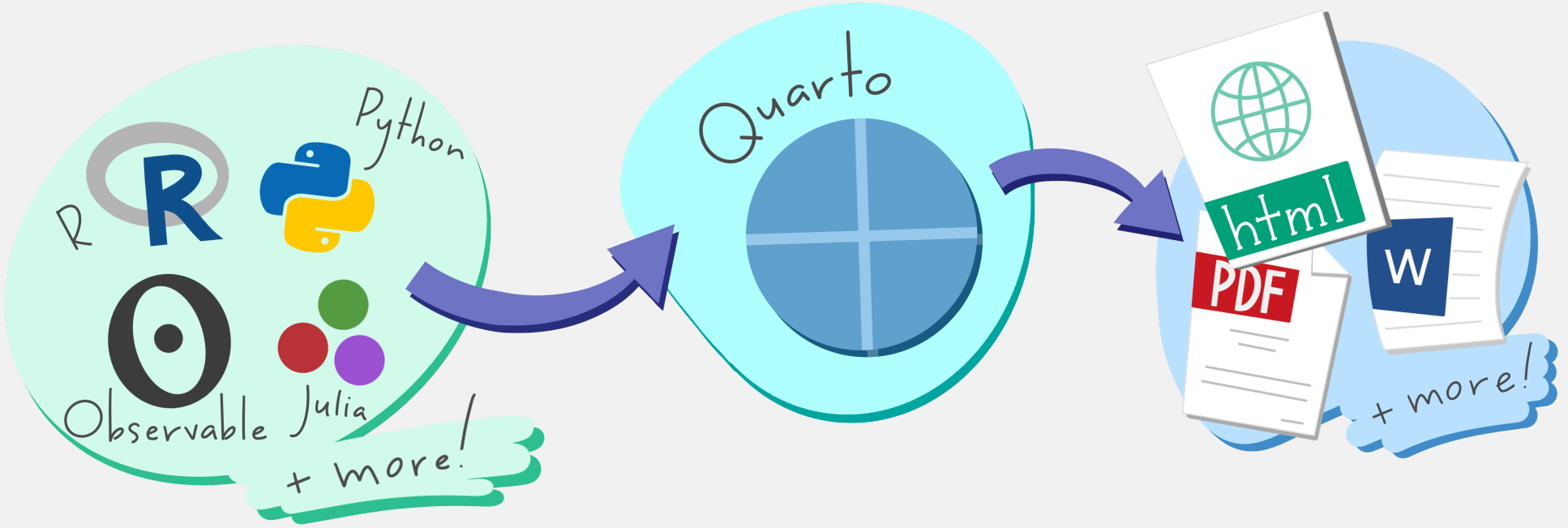
🗓️ Every 3rd Thursday 🕓 4-5 p.m. 📍 Webex

- One topic from the world of scientific workflows
- Material provided online
- If you don't want to miss a lecture
  - Subscribe to the mailing list
- For credit points: Send me a short message (Email or Webex)

What is Quarto?

> Quarto is an open-source scientific and technical publishing system

Basic idea: Create documents with dynamic content and text



Artwork from "Hello, Quarto" keynote by Julia Lowndes and Mine Çetinkaya-Rundel, presented at RStudio Conference 2022. Illustrated by Allison Horst

Document types with Quarto

Examples of document types that can be created with Quarto:

- **Documents**: HTML, PDF, Word
- **Presentations**: HTML, Powerpoint
- **Books**: HTML, ePub, PDF
- **Websites**
- ...

Today

Quarto is a huge topic and there are so many possibilities!

- Practical **introduction** and **overview**
- Main focus R and Positron/RStudio, but same workflow with other languages and other IDEs
- Download a quarto demo project from Github

How to get Quarto

Different options, depending on your workflow:

- **Integrated** in some IDEs (e.g. R Studio, Positron)
- Download the **CLI** for use with other IDEs and workflows
- There is also an **R package** to call quarto from (`install.packages("quarto")`)

Check out the Quarto website for download and tutorials for all options.

lo, Quarto” keynote by Julia Lowndes and Mine Çetinkaya-Rundel, presented at RStudio Conference 2022. Illustrated by Allison Horst.”

Artwork from “Hello, Quarto” keynote by Julia Lowndes and Mine Çetinkaya-Rundel, presented at RStudio Conference 2022. Illustrated by Allison Horst.”

Reproducible documents step by step

Table of contents
1 Introduction
2 Methods
3 Results
4 References

Other Formats
PDF
MS Word

form $y = ax + b + \epsilon$

3 Results

The mean weight of all penguin species is 4201.754386 g. *Gentoo* penguins have an average weight of 5076 g, *Adelie* penguins of 3701 g and *Chinstrap* penguins of 3733 g.

Figure 2 below shows that *Gentoo* penguins have the highest body mass.

▸ Code

Figure 2: Histogram of weight of the three penguin species.

There is a positive relationship between bill length and bill depth for all 3 species, as Figure 3 shows.

▸ Code

Figure 3: Scatter plot with regression lines showing the relationship between bill length and bill depth for the 3 penguin species

In general, it looks like the body characteristics differ between the sexes but also between the penguin species, as Table 1 below illustrates:

▸ Code

Table 1: A summary of the penguin data

species	sex	bill_length	bill_depth	flipper_length	body_mass
Adelie	female	37.25753	17.62192	187.7945	3368.836
Adelie	male	40.39041	19.07260	192.4110	4043.493
Chinstrap	female	46.57353	17.58824	191.7353	3527.206
Chinstrap	male	51.09412	19.25294	199.9118	3938.971
Gentoo	female	45.56379	14.23793	212.7069	4679.741
Gentoo	male	49.47377	15.71803	221.5410	5484.836

4 References

Allaire, JJ, Yihui Xie, Jonathan McPherson, Javier Luraschi, Kevin Ushey, Aron Atkins, Hadley

An HTML example

Reproducible documents step by step

1. *Create* a `.qmd` document
2. *Write* the document including:

- **text** e.g. introduction, methods, or discussion
- **code** (R, Python, Julia) that produces numbers, figures, tables, ...
- **metadata** that defines how the document should look like (e.g. which output format)

3. *Render* the document to a defined output format (e.g. PDF) using `Quarto`

References for all the elements

- Mardown syntax reference
- Code chunks:
  - R code
  - Python code
- YAML header options:
  - HTML
  - PDF
  - DOCX

The text body - Markdown

Markdown is a simple markup language to create formatted text, you can e.g.

- Make italic *text* with *text* or bold **text** with **text**
- Generate headers of different levels

```
# Header level 1
## Header level 2
### Header level 3
```

- Create bullet lists

```
A bullet point list

- item 1
- item 2
- item 3
```

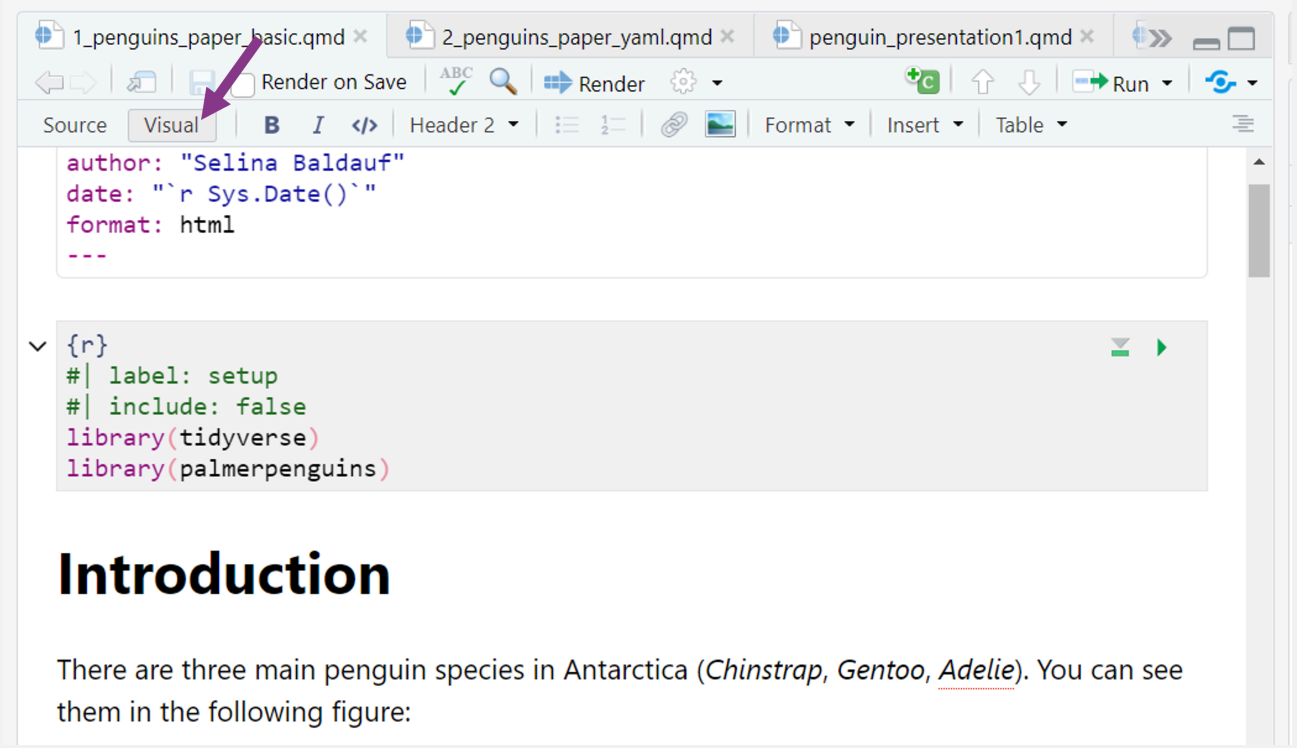
The text body - Markdown

You can also do more complex things like:

- Including images, links or footnotes
- Adding citations
- Latex style mathematical formulas

The text body - Markdown

- RStudio and Positron also have **visual editors**
- Convenient, word-like interface for formatting text and adding features.
  - E.g. Insert citations from Zotero library, DOI search, PubMed, …



The Code

Inline code starts and ends with 1 backtick

```
`{r} `
```

Example

```
The mean of the values 1, 2 and 3 is `{r} mean(1:3)`
```

Rendered output

The mean of the values 1, 2 and 3 is 2.

Same for Python:

```
The mean of the values 1, 2 and 3 is `{python} np.mean([1,2,3])`
```

The Code

Code chunks starts and ends with 3 backticks

````
```{r}
library(ggplot2)

ggplot(penguins, aes(flipper_len, body_mass)) +
  geom_point() +
  geom_smooth(method = "lm")
```
````

````
```{python}
import numpy as np
import matplotlib.pyplot as plt

r = np.arange(0, 2, 0.01)
theta = 2 * np.pi * r
fig, ax = plt.subplots(subplot_kw = {'projection': 'polar'})
ax.plot(theta, r)
ax.set_rticks([0.5, 1, 1.5, 2])
ax.grid(True)
plt.show()
```
````

The Code

Run code chunk

- Code chunks can be run inside the document
- Code chunks are run when document is rendered

The code

Code chunk have special comments that start with `#|` and that control the behaviour of the chunk.

````
```{r}
#| label: fig-penguins
#| fig-cap: Temperature and ozone level.
#| echo: false

library(ggplot2)

ggplot(penguins, aes(flipper_len, body_mass)) +
  geom_point() +
  geom_smooth(method = "lm")
```
````

- `label`: Figure and chunk label that can be referred to in text
- `fig-cap`: Figure caption
- `echo`: Include the output (i.e. the plot) in the document but don't show the code

YAML header

For Metadata

```yaml
---
title: "My first document"
subtitle: "Whatever subtitle makes sense"
author: "Selina Baldauf"
date: today
---
```

YAML header

For document output formats

```yaml
---
format: html
---
```

or other formats like `pdf`, `docx`, `revealjs`, `powerpoint`, ...

You can also specify multiple output formats

```yaml
---
title: "My first document"
author: "Selina Baldauf"
date: today
format:
  html: default
  pdf: default
  docx: default
---
```

YAML header

For document options

```yaml
---
title: "My first document"
author: "Selina Baldauf"
date: today
format:
  html:
    number-sections: true
    toc: true
    toc-location: left
---
```

- Some options are shared, some are specific to one format
- Be careful to get the indentation right!

YAML header

Execute options

```yaml
---
title: "My first document"
author: "Selina Baldauf"
date: today
format: html
execute:
  message: false
  warning: false
---
```

- Default options for code chunks
- Can be overwritten by local comments in code chunks

Render the document

Many different options:

- In RStudio/Positron/VS Code: Render button or keyboard shortcut (usually `Ctrl/Cmd + Shift + K`)
- In the terminal/console: `quarto render my_document.qmd`
- From R, using the `quarto` package:

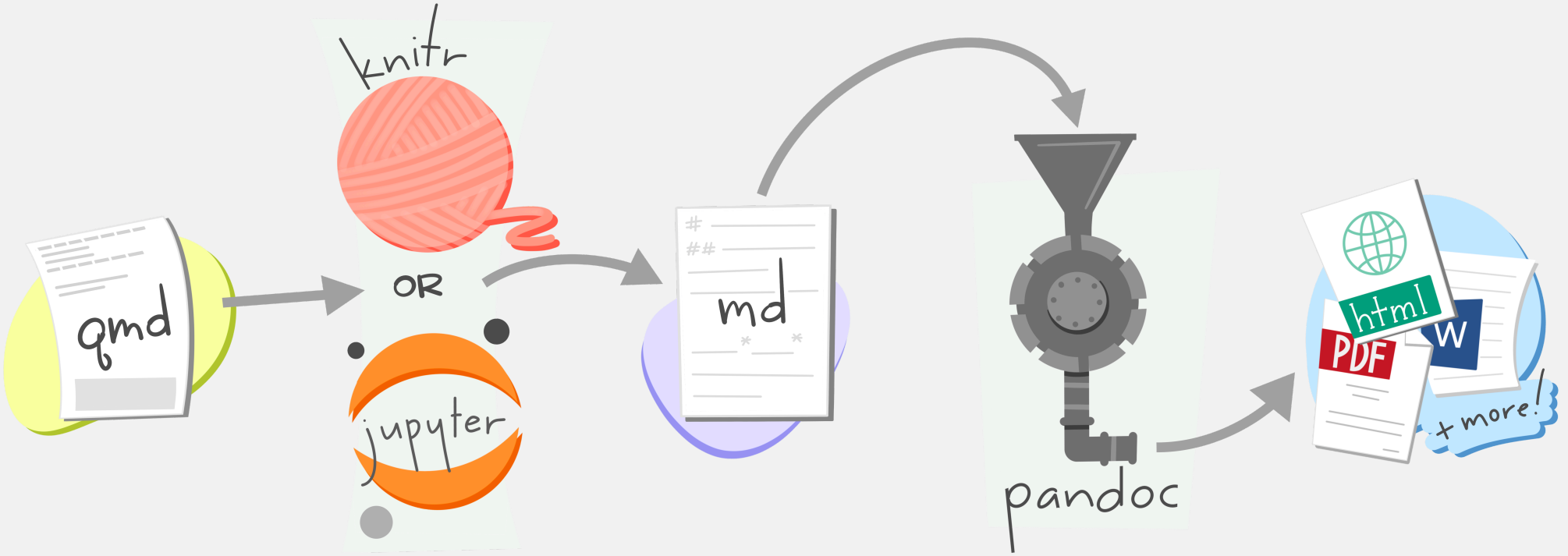
  ```
  quarto::quarto_render("my_document.qmd")
  ```

These commands can be customized with additional options, e.g.

```
quarto render my_document.qmd --to html
quarto render my_document.qmd --to docx
```

Render the document

What happens during rendering?



Artwork from "Hello, Quarto" keynote by Julia Lowndes and Mine Çetinkaya-Rundel,presented at RStudio Conference 2022. Illustrated by Allison Horst.

Parameterized reports

You can also define parameters to be used in your document

R (knitr engine): Add parameters to the YAML header

```yaml
---
title: "My first document"
format: html
params:
  species: "Adelie"
---
```

Use `params$species` to access.

Python (Jupyter engine): Add a special code chunk at beginning

````
```{python}
#| tags: [parameters]

species = 'Adelie'
```
````

Access via the parameter name `species`.

Parameterized reports

Render your document with different parameter inputs:

In R:

```r
quarto::quarto_render(
  input = "my_report.qmd",
  output_format = "pdf",
  output_file = "report_chinstrap.pdf",
  params = list(species = "Chinstrap")
)
```

In the console:

```
quarto render my_report.qmd --to pdf --output "report_chinstrap.pdf" -P species:Chinstrap
```

Summary

- Quarto combines formatted text and code in one document

Benefits

- **Reproducibility**: Code and text in one document
- **Flexibility**: Different output formats and programming languages
- **Version control friendly**: Text based
- **Parameterized reports**: Generate multiple reports from the same template
- ...

- Other **options and formats**: Presentations, Books, Websites
- Easily **publish** documents or websites
- Check out the **demo project** and the **resources** on the lecture website

"Artwork from "Hello, Quarto" keynote by Julia Lowndes and Mine Çetinkaya-Rundel RStudio Conference 2022. Illustrated by Allison Horst."

Next lecture

Topic tba

🗓️ 20th November 🕓 4-5 p.m. 📍 Webex

🔔 Subscribe to the mailing list

📧 For topic suggestions and/or feedback send me an email

The end :)

Questions?

"Artwork from"Hello, Quarto" keynote by Julia Lowndes and Mine Çetinkaya-Rundel, presented at RStudio Conference 2022. Illustrated by Allison Horst."

References

- Quarto website offers everything you need to get started
  - Download Quarto and starting guide for different IDEs
  - Guides for different output formats
  - Gallery with Examples
- Quarto introduction workshop on Youtube
- A curated collection of resources