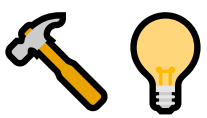
Reproducible documents with Quarto

Scientific workflows: Tools and Tips 

5/11/23

What is this lecture series?

Scientific workflows: Tools and Tips 🛠️

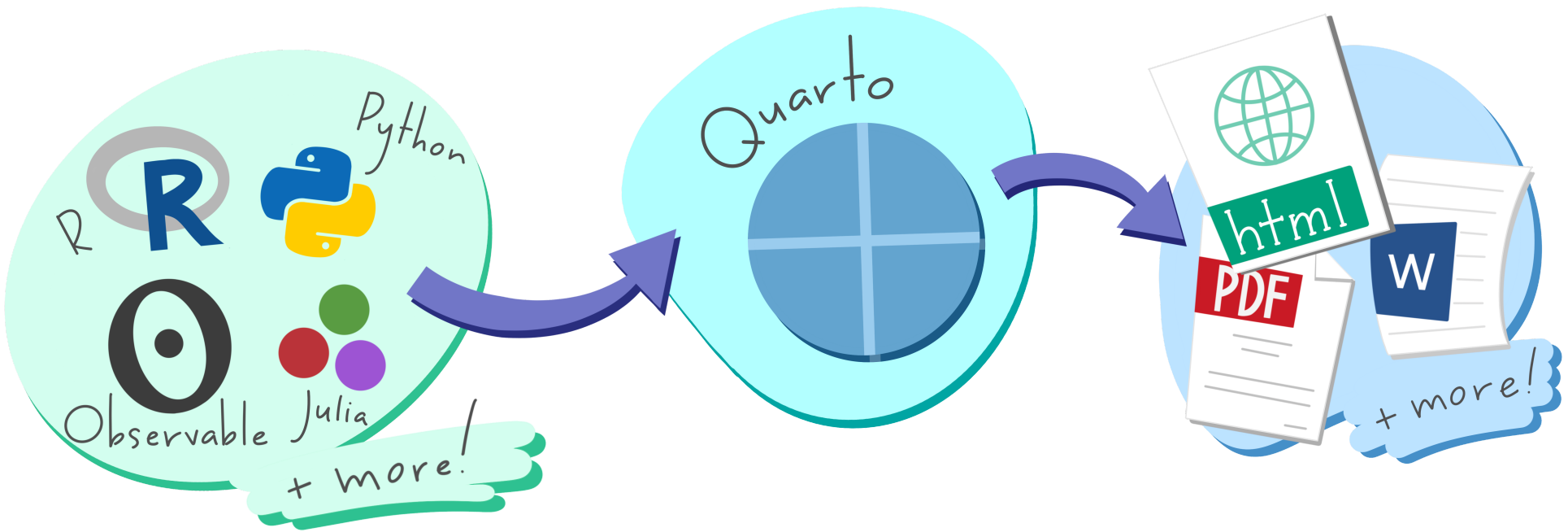
🗓️ Every 3rd Thursday 🕐 4-5 p.m. 📍 Webex

- One topic from the world of scientific workflows
- For topic suggestions send me an email
- If you don't want to miss a lecture
  - Check out the lecture website
  - Subscribe to the mailing list
- Slides provided on Github

Reproducible documents with 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

Reproducible documents with Quarto

Document types that can be created with Quarto (examples):

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

Before we start

Quarto is a huge topic and there are so many cool Quarto things!

Goal of today: Introduction to Quarto and an **overview** of different document types and their possibilities.

- Focus on R and R Studio
- Keep in mind: This also works with other languages and other IDEs, the principles are all the same.

How to get Quarto

Different options, depending on your workflow

- **Integrated in** new versions of **R Studio** (Update R Studio via `Help -> Check for Updates`)
- Download the **CLI** for use with other IDE (e.g. Visual Studio code)
- **R package** `quarto` that provides an interface to Quarto (preconditions are Quarto CLI or new version of R Studio)

```
install.packages("quarto")
```

Check out the Quarto website for download and more info.

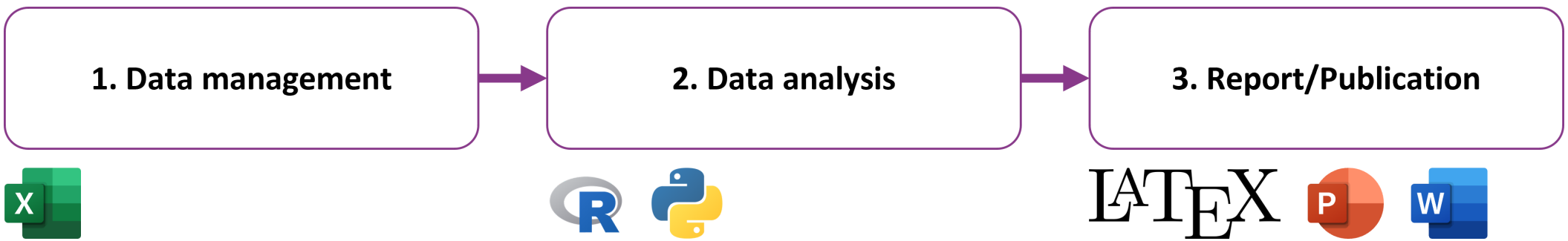
The classic use case

> Reproducible documents for data analysis

An example document

Download the demo project from Github

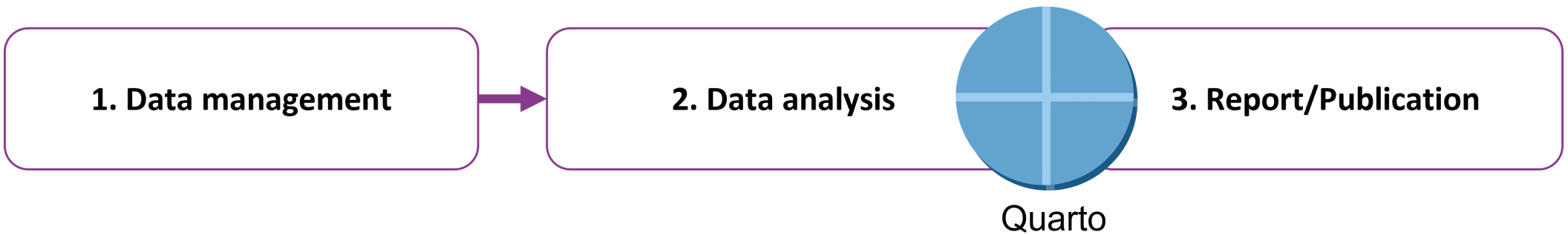
A standard workflow



Problem: Manual updates are **error prone** and **non-reproducible**

A Quarto workflow

Solution: Use a **Quarto workflow** → everything (code, text, metadata) in one place. Let `Quarto` do the magic



Advantages of this workflow:

- Easy to redo analysis
- No more copy pasting
- Reproducibility → workflow independent of the person that wrote it

The basic Quarto workflow

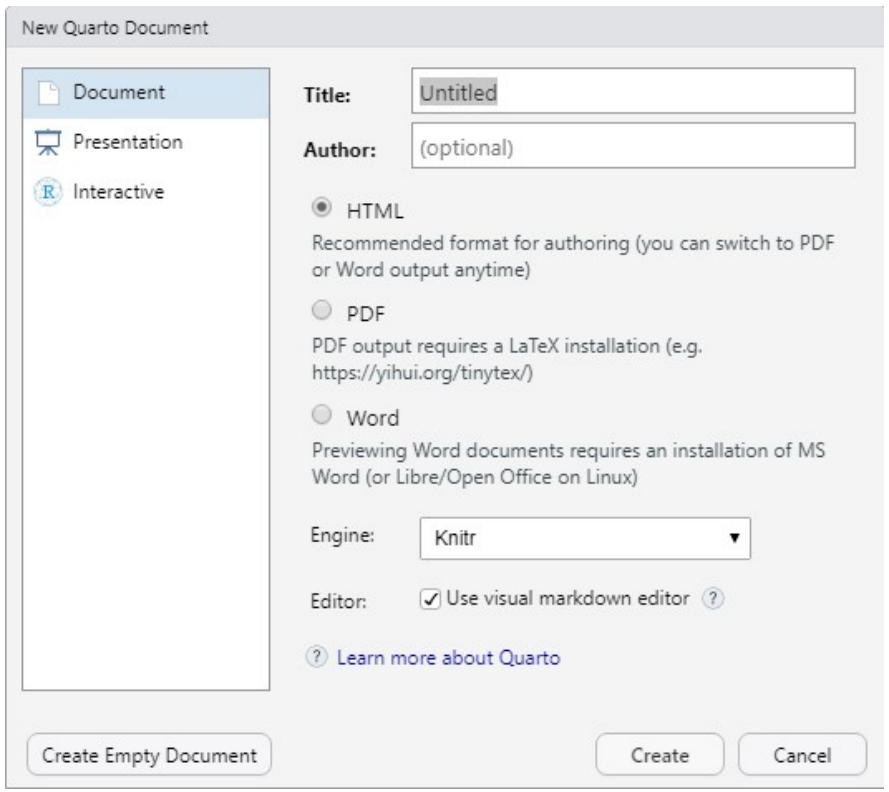
1. **Create** a `.qmd` document
2. **Write** the document

   - *text* that e.g. introduces you topic, interprets your results, ...
   - *code* (R, Python, Julia) that produces numbers, figures, tables, ...
   - *metadata* that defines how the result should look like (e.g. which output format)

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

Step 1 - Create the document

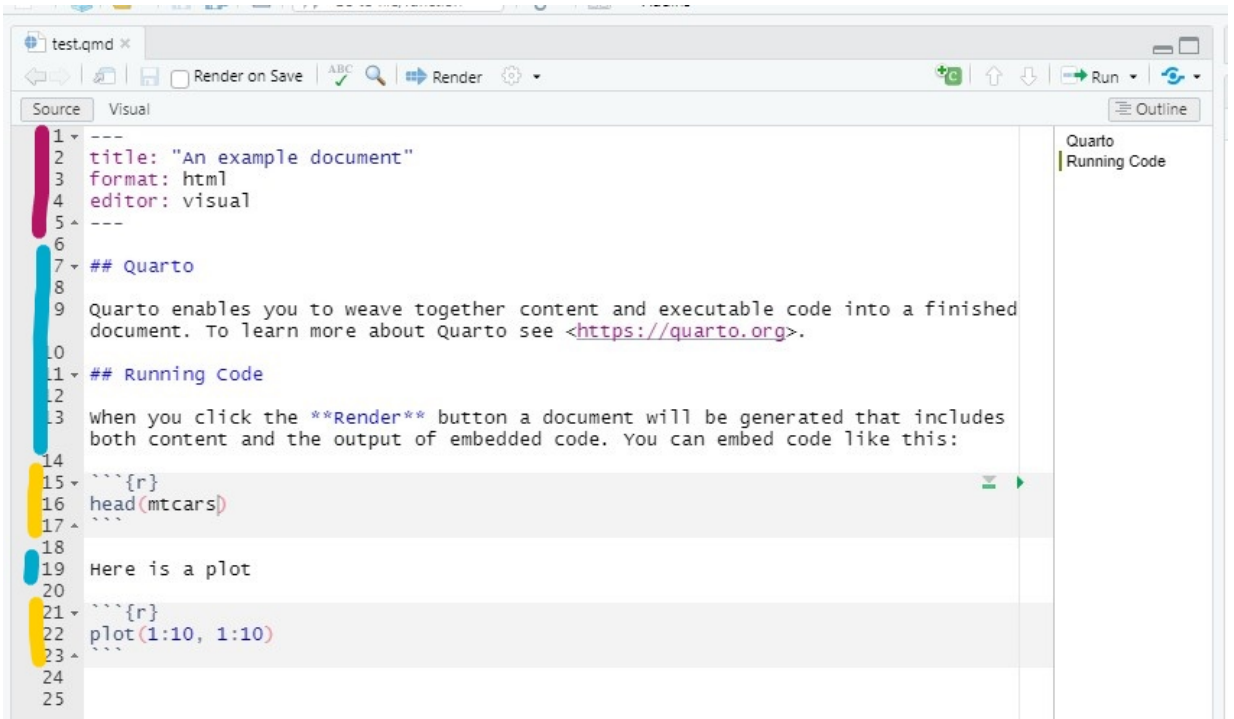
Open R Studio and go to `File -> New File -> Quarto Document`



Just click `Create` and the file will open in R Studio.

In other environments you can just create and empty file with `.qmd` ending

Step 2 - Write the document



You can edit:

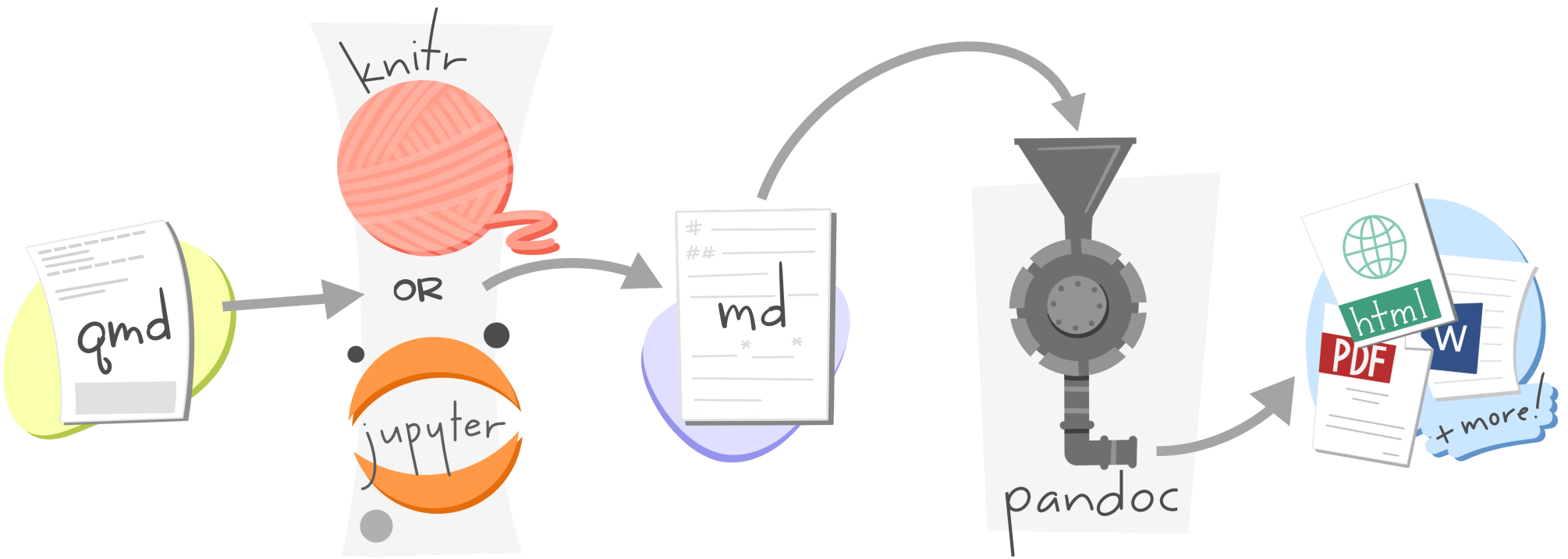
- YAML header: Metadata and control of the output format
- Markdown text: Formatted text body
- Code chunks: Python, R, Julia code

Step 3 - Render the document

- Click the `Render` button in R Studio
- Keyboard shortcut `Ctrl + Shift + K`
- Call the `quarto::quarto_render()` function
- In the terminal: `quarto render doc.qmd`

Step 3 - 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.

Basic Elements of a `.qmd` document

> Text body, Code, YAML header

The text body - Markdown

- Markdown is a simple markup language to create formatted text

Markdown allows you to do simple things like

- Making text bold or italic
- Creating headers of different levels
- Creating bullet lists

Or more complex things like:

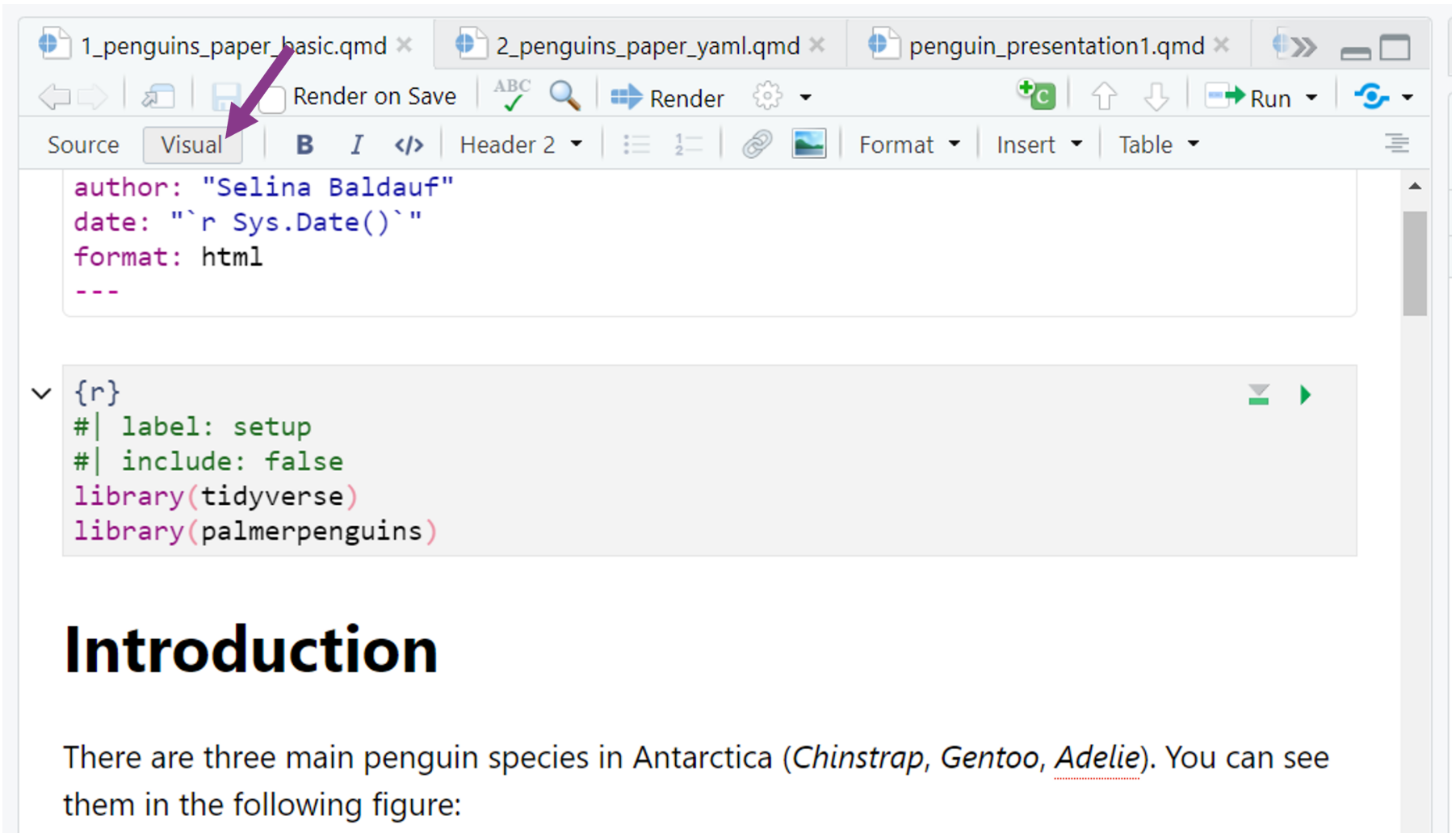
- Including images, links or footnotes
- Add citations and cross-references
- Latex style mathematical formulas

The text body - Markdown

If you don't want to use markdown, there is a really nice feature in R Studio: **The visual editor**.

The visual editor

Convenient, word-like interface for formatting text and adding features.



The visual editor

Using the visual editor, makes many things that would be painful in Markdown really easy.

My favorite feature in the visual editor:

- Add citations (`Insert -> Citation`) from Zotero library, DOI search, PubMed, ...

The Code

Code can be included in **code chunks** as **inline code**

- Code chunks can contain any type of R/Python/Julia code
- Code is (by default) executed and output is included in document
  - Text output
  - Figures
  - ...

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)`
```

looks like this:

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

The Code

Code chunks starts and ends with 3 backticks

````
```{r}
library(ggplot2)
ggplot(airquality, aes(Temp, Ozone)) +
  geom_point() +
  geom_smooth(method = "loess")
```
````

````
```{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

Insert a code chunk

- Insert a code chunk by going to `Code` -> `Insert chunk`
- Use the keyboard shortcut `Ctrl + Alt + I` / `Cmd + Option + I`
- Inline chunks have to be typed or use the `</>` symbol in visual mode

The Code

Run code chunk

- Code chunks are evaluated by `knitr` when rendering the document
- Code chunks can also be run like normal R code
- Run Code chunk by clicking on the green arrow next to the chunk

````
```{r cars}
summary(cars)
```
````

The code

Code chunk have special comments that start with `#|`, e.g.

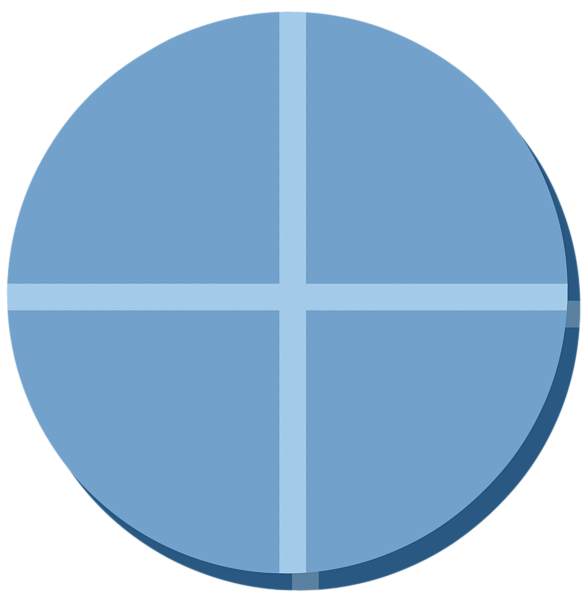
````
```{r}
#| label: fig-airquality
#| fig-cap: Temperature and ozone level.
#| include: false

library(ggplot2)

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

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

Include image via code chunk

Include an image from a code chunk via `knitr::include_graphics()`.

````
```{r}
#| echo: false
#| out.width: "20%"
#| fig-align: center
#| fig-cap: Quarto logo
#| fig-cap-location: margin
knitr::include_graphics("images/2023_05_11_quarto/quarto_straight.png")
```
````



Quarto logo

YAML header

For Metadata

```
---
title: "My first document"
subtitle: "Whatever subtitle makes sense"
author: "Selina Baldauf"
date: "`r Sys.Date()`"
---
```

- Inline R code can print the current date at knitting time

YAML header

For document output formats

```yaml
---
title: "My first document"
author: "Selina Baldauf"
date: "`r Sys.Date()`"
format: html
---
```

You can also specify multiple output formats

```yaml
---
title: "My first document"
author: "Selina Baldauf"
date: "`r Sys.Date()`"
format:
  html: default
  pdf: default
  docx: default
---
```

YAML header

For document options

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

```yaml
---
title: "My first document"
author: "Selina Baldauf"
date: "`r Sys.Date()`"
format:
  html:
    number-sections: true
    toc: true
    toc-location: left
    code-fold: true
    df-print: kable
  pdf:
    toc: true
    number-sections: true
    df-print: kable
  docx: default
---
```

YAML header

Execute options

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

```yaml
---
title: "My first document"
author: "Selina Baldauf"
date: "`r Sys.Date()`"
format: html
execute:
  echo: false
  warning: false
---
```

References for all the elements

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

Summary

- Quarto can combine formatted text and code to create reproducible documents

Use cases for scientists

- Write reports and documents about data analysis
- Use `qmd` instead of `.R` files to add text already to the code (e.g. for interpretation or methods)
- Great for teaching because you can add explanations to your code
- Many more ...

Outlook

Quarto offers many more things like:

- Presentations (Powerpoint or Revealjs/HTML)
- Websites
- Easily publish your documents or websites on quartopub or Github
  - See here for more information
- Control the rendering process via R scripts using `quarto::quarto_render()`
- Parameterized reports

Next lecture

Version control with Git

Git is an essential skill if you use any programming language. It allows you to keep track of changes over time, collaborate with others, and maintain a clear and organized file structure. This can save time, improve research efficiency, and makes it easy to publish your code.

 15th June  4-5 p.m.  Webex

 For topic suggestions and/or feedback send me an email

 Subscribe to the mailing list

Thank you for your attention :)

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