Quarto & R Markdown

Applied Data Science using R

Prof. Dr. Claudius Gräbner-Radkowitsch
Europa-University Flensburg, Department of Pluralist Economics
www.claudius-graebner.com | @ClaudiusGraebner | claudius@claudius-graebner.com

Europa-Universität
Flensburg

Europa-Universität
Flensburg
International Institute of Management
and Economic Education
Department of Pluralist Economics

Goals for today

I. Understand what **Quarto** is and how it relates to **R Markdown** and **knitr**

II. Write your first Quarto document

III. Render Quarto documents into **html** and **PDF** format

IV. Become aware of specific challenges for project management when using Quarto

What is Quarto?

First: What is R Markdown?

- R Markdown is a document format that allows you to write documents containing code of two languages:

  - R code to perform statistical analysis ➡ we know this (almost 😉)

  - Markdown code to create formatted text using a plain text editor

- Markdown is readable in its source and, if rendered, allows for formatting, such as **bold** or *italic* fonts, tables, headings…
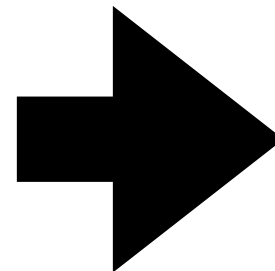
```
# Formatting in markdown
This is the source code. It is ugly,
but it is readable.

> **Note:** You can edit it on every editor,
and still can format your text.

You can have *italics*, you can have **bold**
texts, and much more!

## Tables

| Column 1 | Column 2  |
|:--------:|:---------:|
| Cell 1   | Cell 2    |
| Cell 3   | Cell 4    |
```

Formatting in markdown

This is the source code. It is ugly, but it is readable.

> **Note:** You can edit it on every editor, and still can format your text.

You can have *italics*, you can have **bold** texts, and much more!

Tables

| Column 1	Column 2
Cell 1	Cell 2
Cell 3	Cell 4

First: What is R Markdown?

- R Markdown is a document format that allows you to write documents containing code of two languages:

  - R code to perform statistical analysis → we know this (almost 😉)

  - Markdown code to create formatted text using a plain text editor

- Markdown is readable in its source and, if rendered, allows for formatting, such as **bold** or *italic* fonts, tables, headings…

- Together, R and Markdown allows you to write formatted texts and conducts statistical analysis within one file

  - Perfect to make research accessible and reproducible
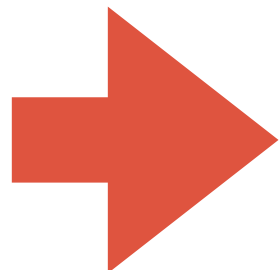
We will not cover the basics of markdown → quite boring in a group

Please do the **interactive Markdown tutorial** on the course webpage

Europa-Universität Flensburg

And what about Quarto?
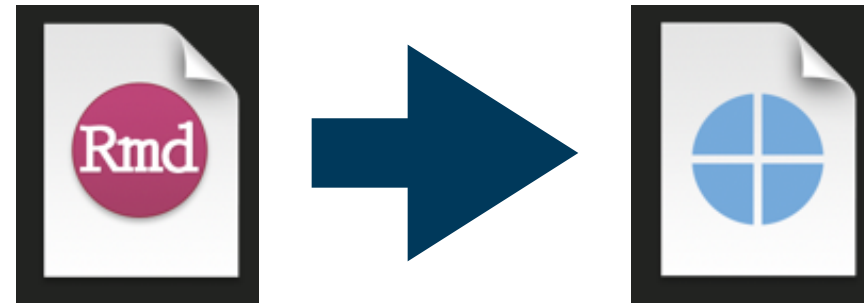
- Quarto is basically a next-generation version of R Markdown

- It allows you to do everything you can do in R Markdown, but…

  - …it works with more languages, including Python and Julia

  - …it has some additional features and capabilities

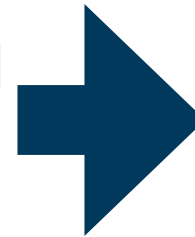- R Markdown will stay, but no need to learn it any more

What is Quarto?

The header - contains meta information

Markdown

Chunk options

R Chunks

```
1   ---
2   title: 'GDP and development'
3   author: "Claudius"
4   date: '2022-04-06'
5   format:
6     html:
7       theme: pulse
8       toc: true
9       toc-depth: 2
10      toc-location: body
11      number-sections: true
12  ---

14  # Packages used
15
16  ```{r}
17  library(DataScienceExercises)
18  library(ggplot2)
19  ```

21  # GDP and development indicators
22
23  While there are convincing critiques of GDP as a measure of well-being, there is also a clear
    relationship between GDP and socio-economic wellbeing indicators, such as life expectancy:
24
25  ```{r, include=FALSE}
26  gap_data <- DataScienceExercises::gaplifeexp2007
27  head(gdp_data, 3)
28
29
30  ```{r}
31  #| echo: false
32  #| warning: false
33  plot_preview <- ggplot2::ggplot(
34    data = gdp_data,
35    mapping = ggplot2::aes(
36      x = gdpPercap,
37      y = lifeExp,
38      size = pop,
39      fill = continent
40    )
41  ) +
42    ggplot2::geom_point(
43      shape=21, color="black", alpha=0.5) +
44    labs(
45      title = "Life expectancy and income per capita",
46      caption = "Note: size of bubbles represents population. Data: Gapminder",
47      x = "GDP per capita (int. Dollar)",
48      y = "Life expectancy in years"
49    ) +
50    ggplot2::scale_x_continuous(
51      labels = scales::number_format(scale = 0.001, suffix = "k")
52    ) +
53    ggplot2::scale_size_continuous(
54      guide = "none",
55      range = c(0.1, 24)
56    ) +
57    scale_fill_brewer(
58      palette = "Dark2"
59    ) +
60    ggplot2::theme_bw() +
61    theme(
62      legend.position = "bottom",
63      legend.title = ggplot2::element_blank(),
64      panel.border = ggplot2::element_blank(),
65      axis.line = ggplot2::element_line(colour = "grey"),
66      axis.ticks = ggplot2::element_line(colour = "grey")
67    )
68  plot_preview
69  ```
70
71  This relationship seems to be, however, heterogeneous across countries.
```

Europa-Universität Flensburg

What is Quarto?

- Quarto documents can be rendered to very different formats

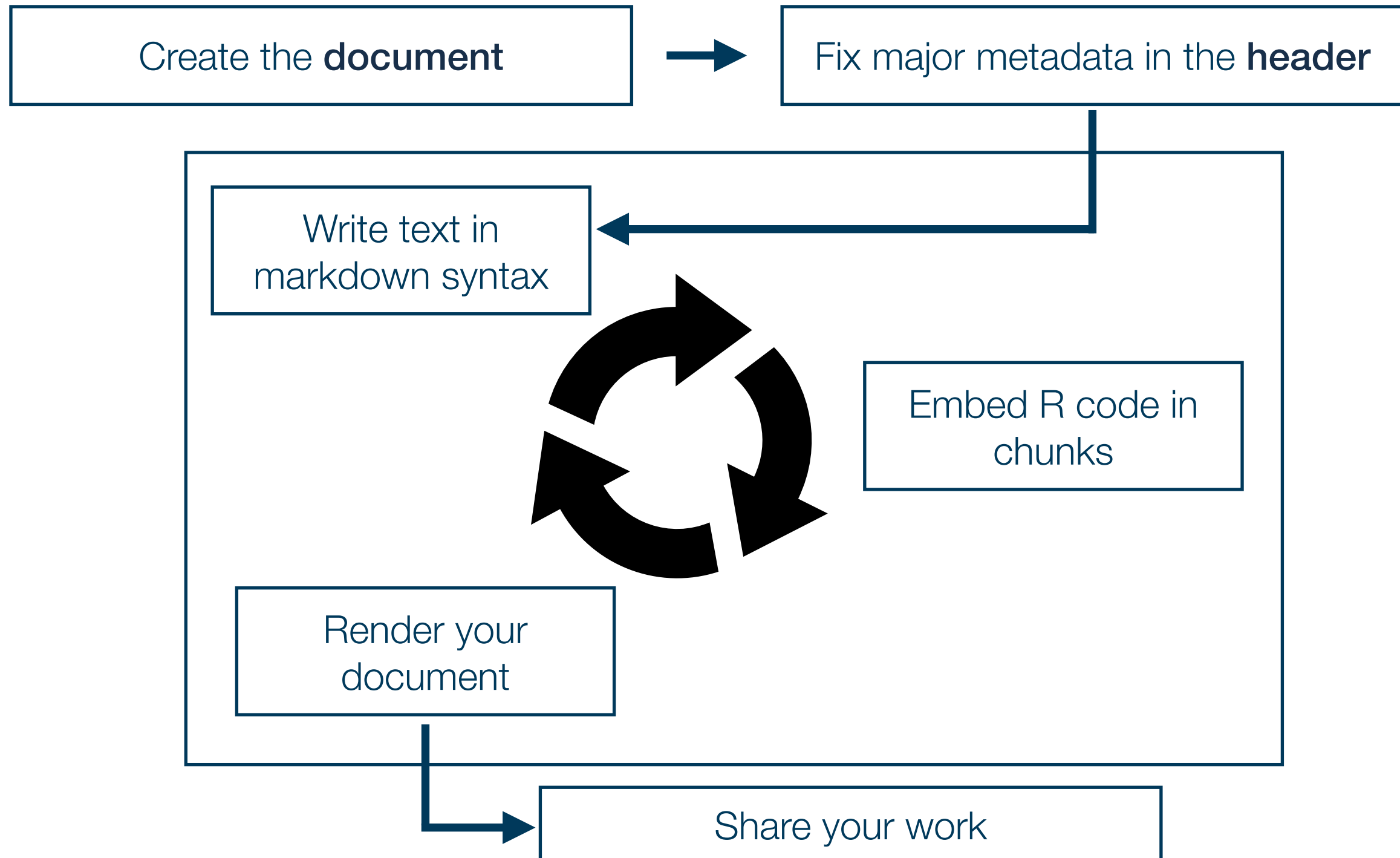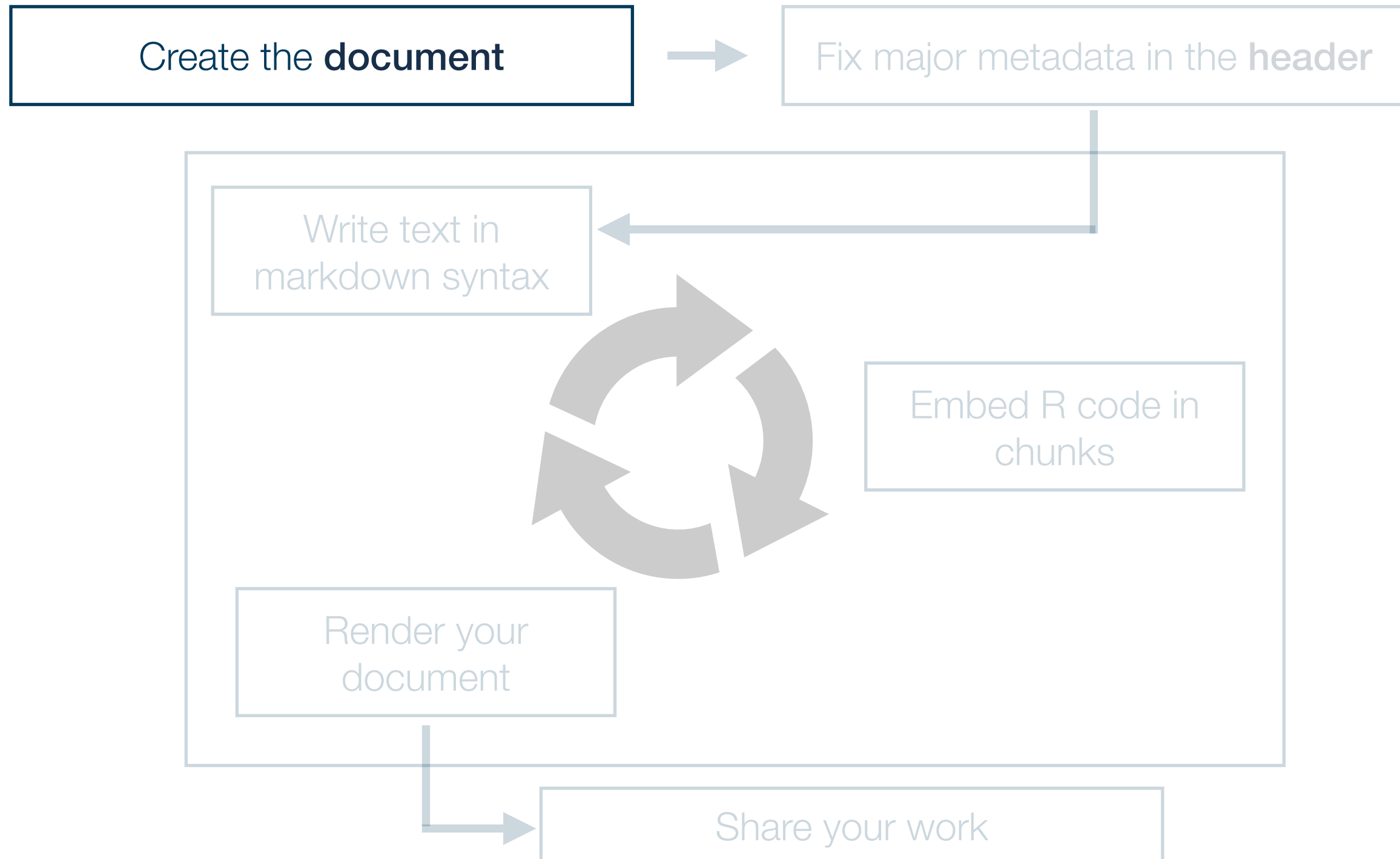| bookdown to write books	Many extensions	learnr to create exercises
blogdown to create websites		...and many more

- Basic **syntax the same** for all applications → this will be the focus here

  - Now go through the single steps required to get a Quarto document working

Europa-Universität Flensburg

Quarto step by step

Quarto step-by-step

Create the **document** → Fix major metadata in the **header**

Write text in markdown syntax

Embed R code in chunks

Render your document

Share your work

R Markdown step-by-step

Create the **document** → Fix major metadata in the **header**

Write text in markdown syntax
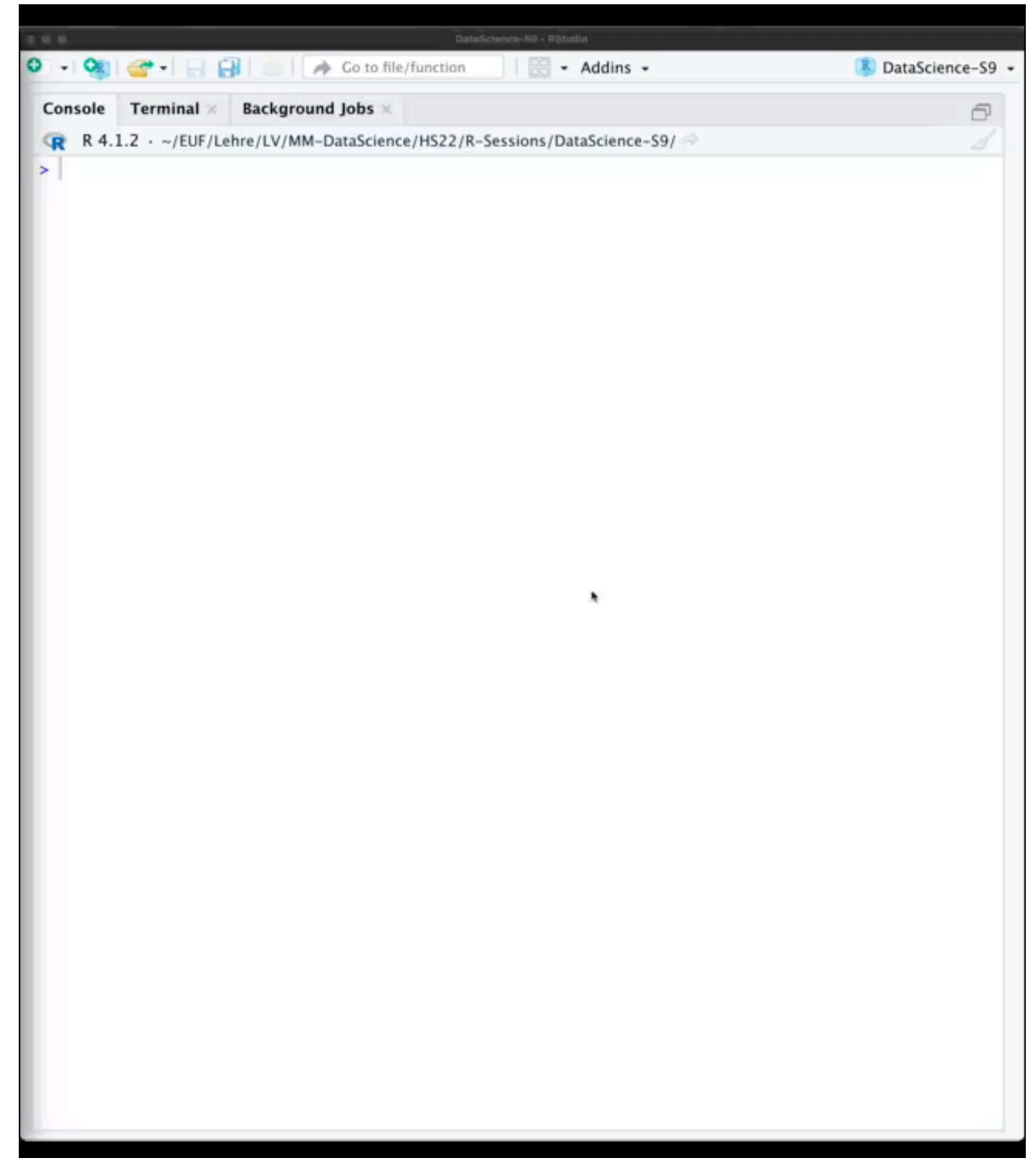
Embed R code in chunks
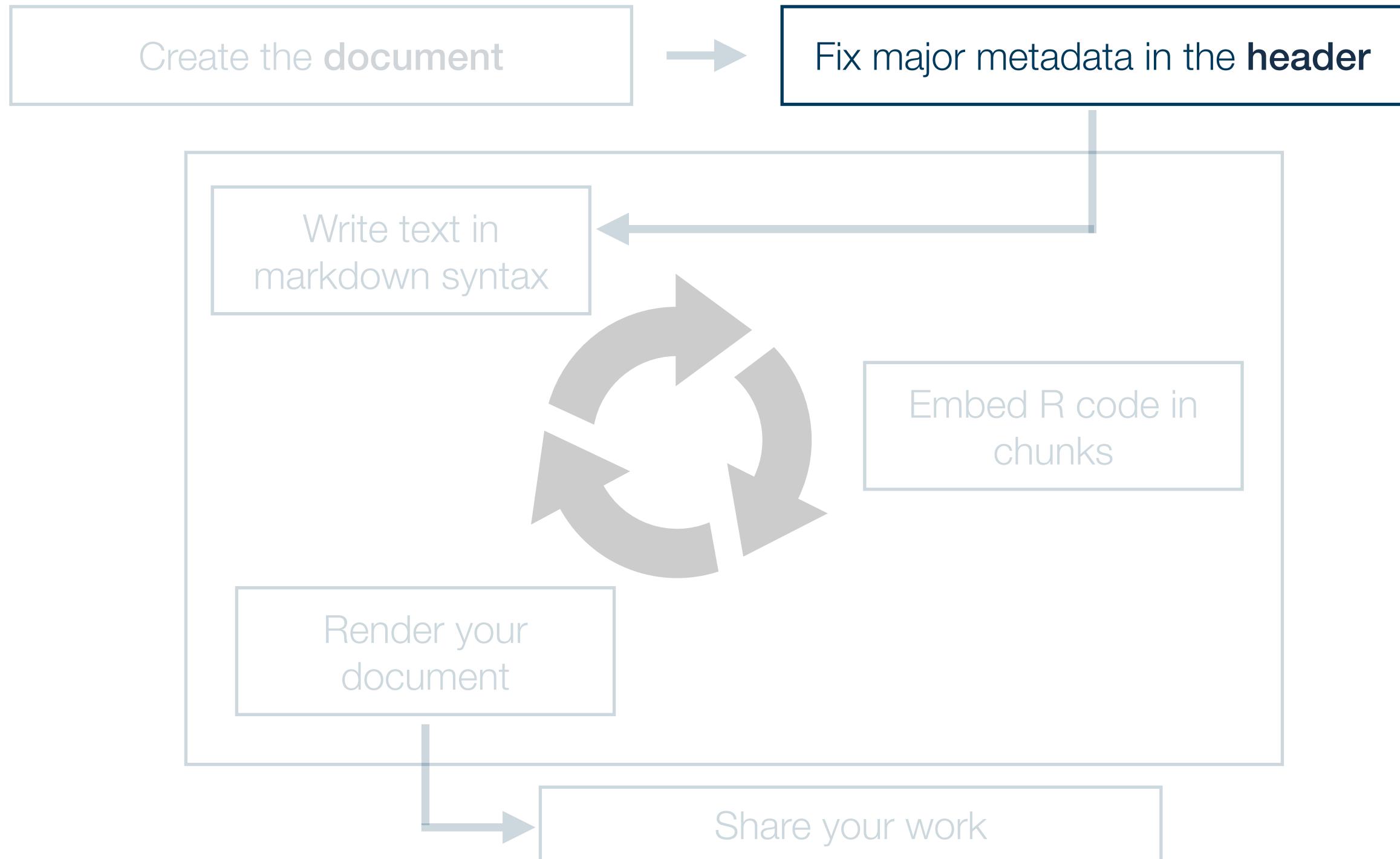
Render your document

Share your work

Quarto step-by-step
1. Creating the document

- First create a new Quarto document, and choose, if adequate, a template

  - `New ▸ Quarto document…`

- There are plenty of templates distributed as packages

  - For learning purposes its best to start with a blank document

- After creating the document its best to save it immediately

  - Either in the subdirectory `R`, or in a separate top-level directory `quarto`

Europa-Universität Flensburg

Quarto step-by-step

Create the **document** → Fix major metadata in the **header**

Write text in markdown syntax

Embed R code in chunks

Render your document

Share your work

Quarto step-by-step
2. Specify the header

```
 1  ---
 2  title: 'The title of my document'
 3  author: "Claudius"
 4  date: '2022-04-06'
 5  format:
 6    html:
 7      toc: true
 8      toc-location: body
 9      number-sections: true
10  ---
```

- The header contains meta data

  - It starts and ends with `---`

  - Usually you should set at least `title`, `author`, and `date`

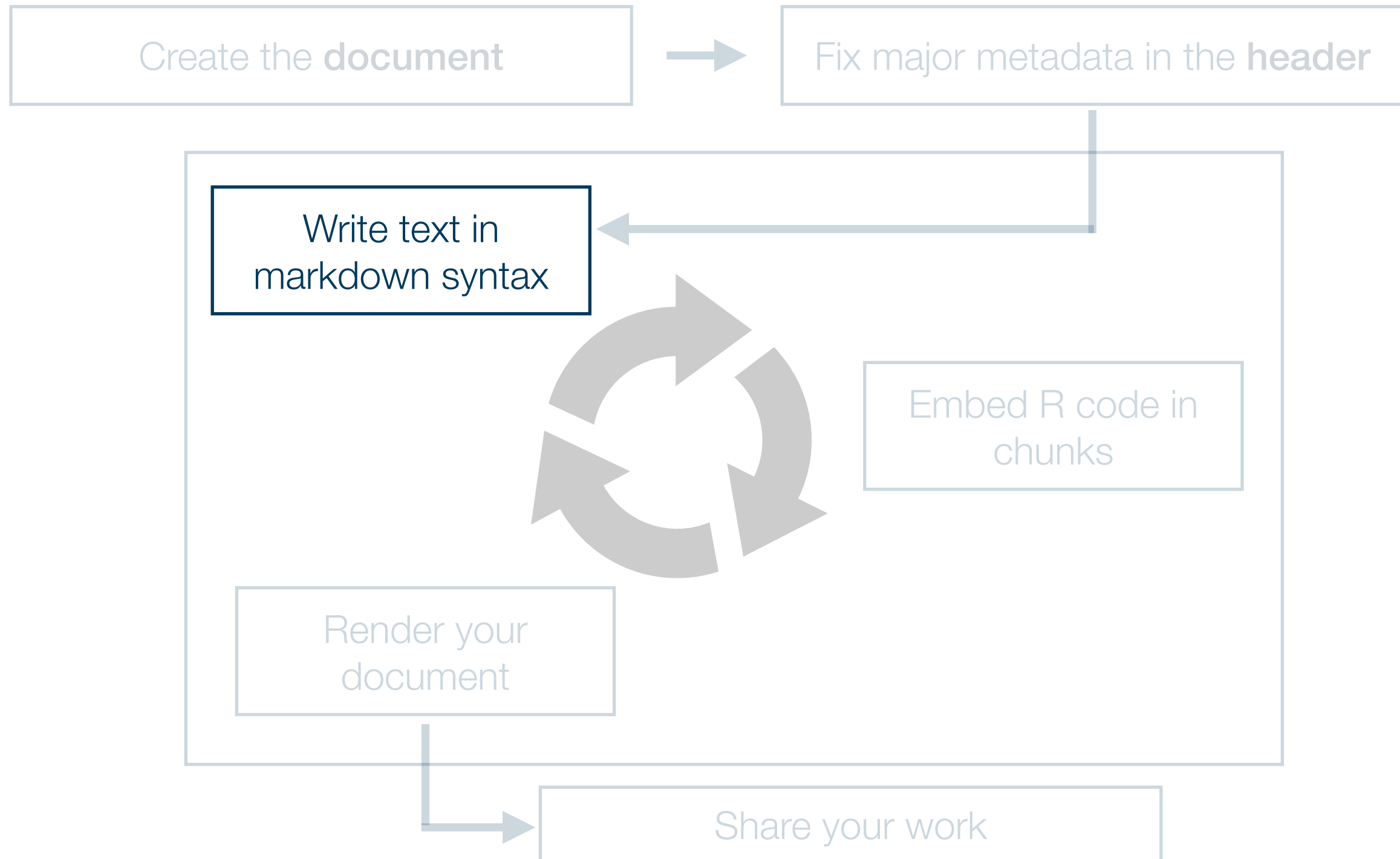- The output arguments are usually set later

  - Determines the argument `output_format` for the function `quarto::quarto_render()`, which is used to render output

  - The more specific comments translate into different arguments of `quarto::quarto_render()`, especially `pandoc_args`

- The headers is written in YAML

  - There is a nice overview over the major keywords in the Quarto docs (see further readings)

Quarto step-by-step

Create the **document** → Fix major metadata in the **header**

Write text in markdown syntax

Embed R code in chunks

Render your document

Share your work

Quarto step-by-step
3. Write the main text

- Just write the text as you would do in any normal text editor
  - To format the text, follow the Markdown syntax
  - Syntax best learned by example → interactive Markdown tutorial


```
example-doc.qmd ✕                                    ☐ ☐
◀ ▶  ⬀  💾  ☐ Render on Save    ABC✓ 🔍 ⮕ ⚙ ▾ +c

Source   Visual                              ☰ Outline

 1 ▾  ---
 2    title: 'The title of my document'
 3    author: "Claudius"
 4    date: '2022-04-06'
 5    format:
 6      html:
 7        toc: true
 8        toc-location: body
 9        number-sections: true
10 ▴  ---
11
12 ▾  # A level 1 heading
13
14    Just write some text.
15
16 ▾  ## A level 2 heading
17
18    You may...
19
20    * also
21    * have
22    * bullet
23    * lists
24
25    and much more!|
26

25:15   # A level 2 heading ⇕            Quarto ⇕
```
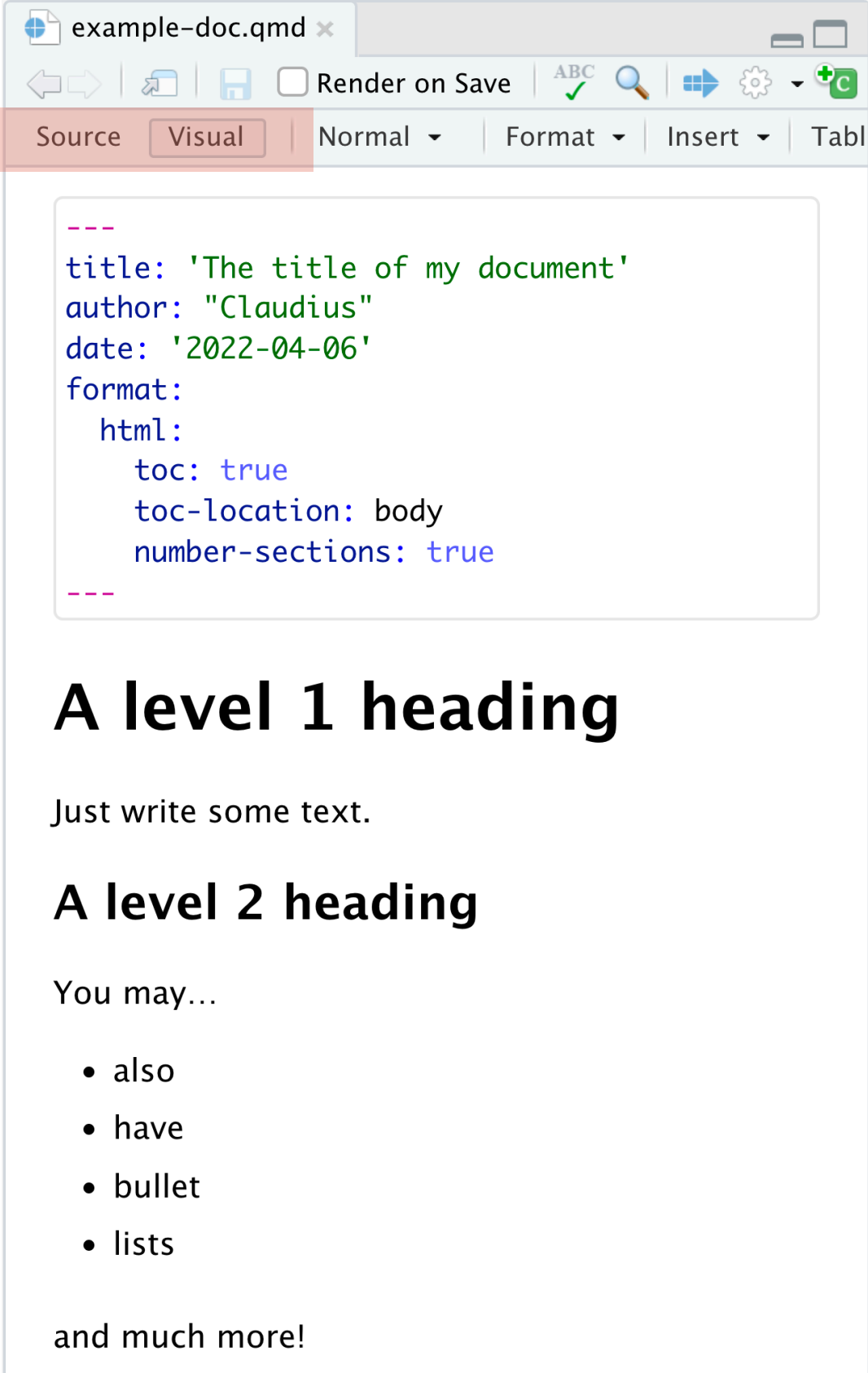
Quarto step-by-step
3. Write the main text

- Just write the text as you would do in any normal text editor

  - To format the text, follow the Markdown syntax

  - Syntax best learned by example → interactive Markdown tutorial

  - Check out the Visual Quarto editor!

example-doc.qmd

☐ Render on Save

Source | Visual | Normal ▾ | Format ▾ | Insert ▾ | Tabl

```
---
title: 'The title of my document'
author: "Claudius"
date: '2022-04-06'
format:
  html:
    toc: true
    toc-location: body
    number-sections: true
---
```

A level 1 heading

Just write some text.

A level 2 heading

You may…

- also
- have
- bullet
- lists

and much more!

01

Introduction

Each lesson introduces a single Markdown concept with an example. When you see a red pulsing circle in the example, select to examine it for details.
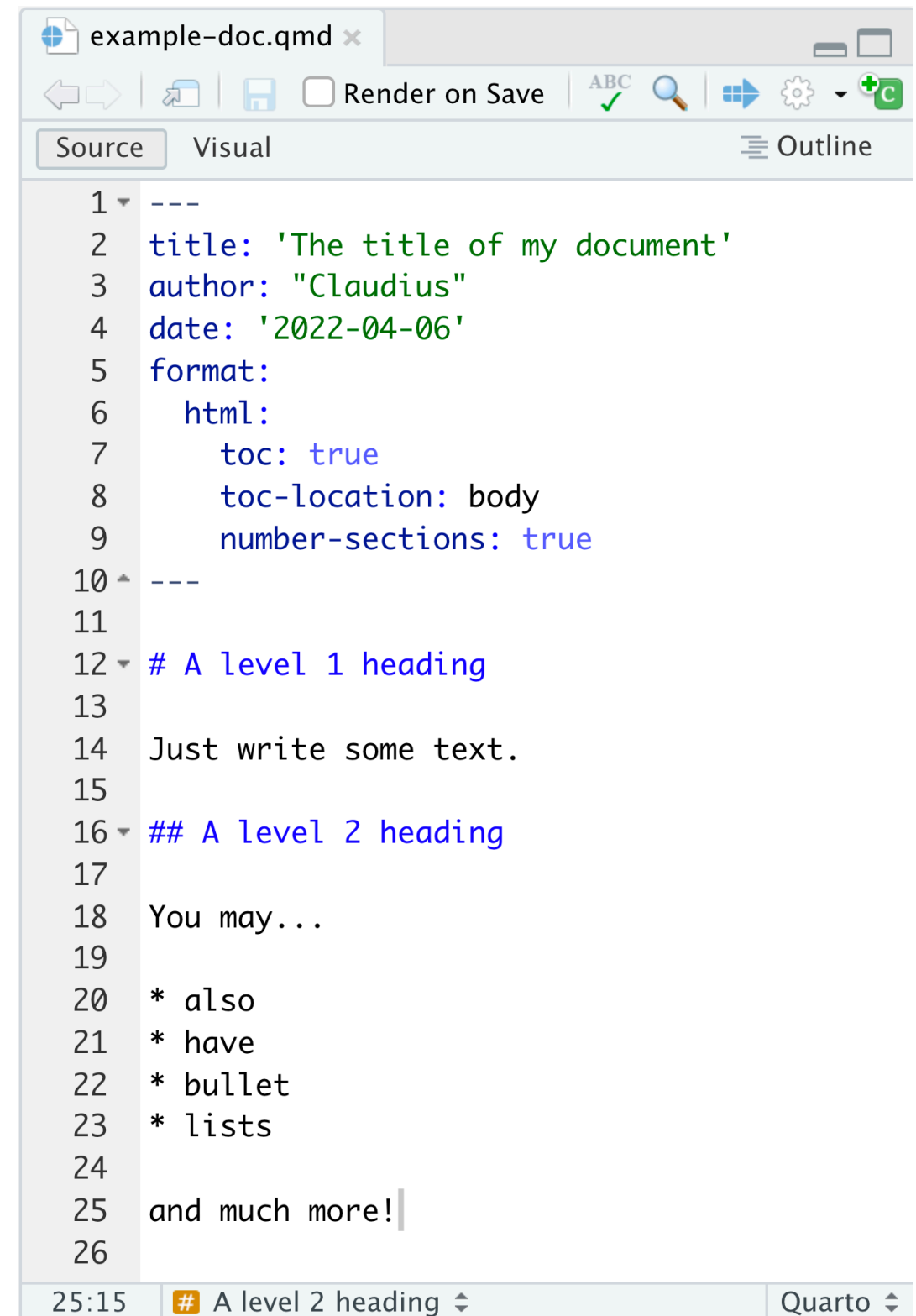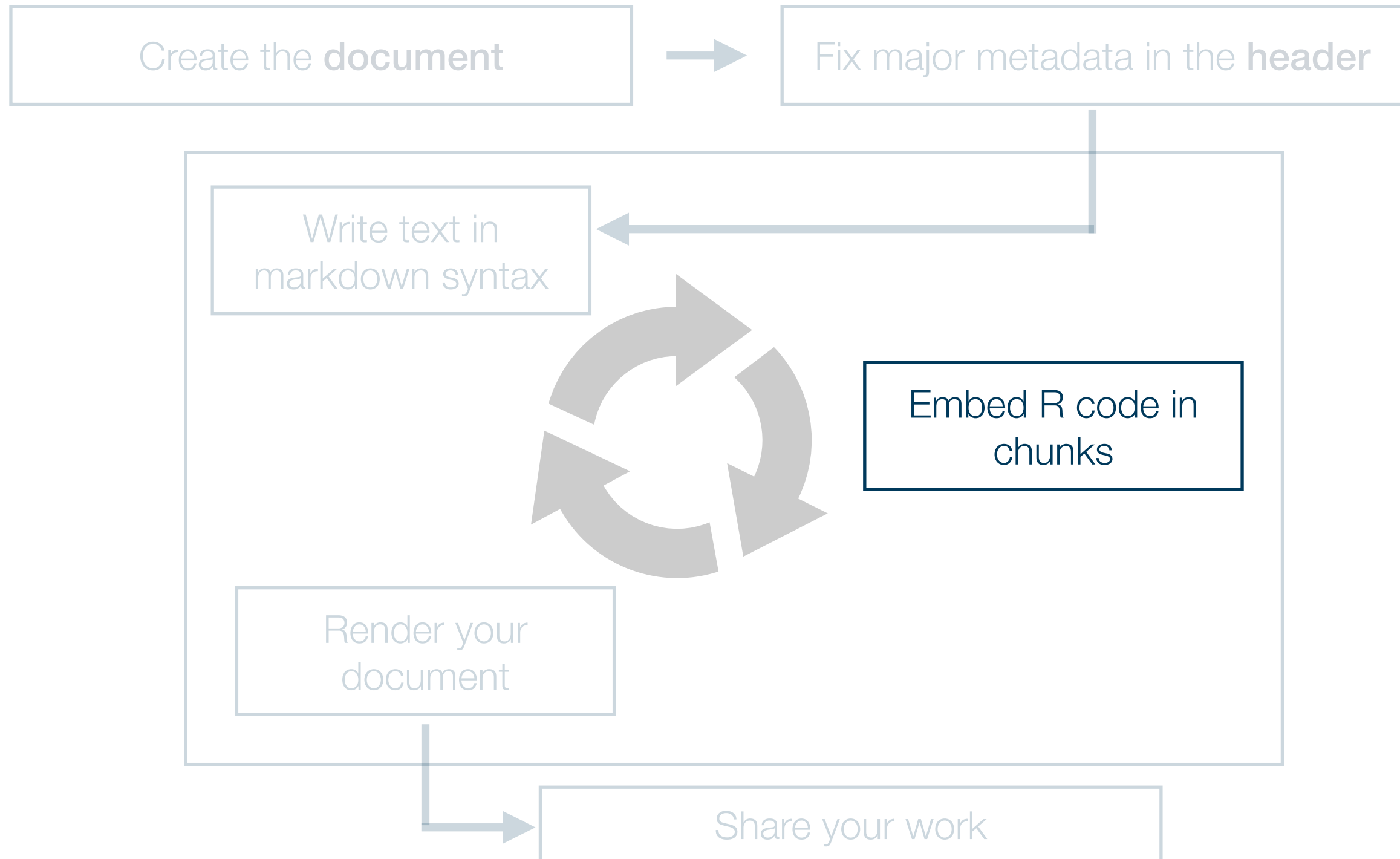
After studying the example, try a few practice exercises with your new knowledge. Skip to any lesson at any time via the navigation controls. Experiment and have fun!

This tutorial is open source – help us improve it!

BEGIN LESSON →

WHAT IS MARKDOWN?

Intro ▪
Emphasis ▪
Paragraphs ▪
Headings ▪
Blockquotes ▪
Lists ▪
Links ▪
Images ▪
Code ▪
Nested Lists ▪
The End ▪

Europa-Universität Flensburg

Quarto step-by-step

Create the **document** → Fix major metadata in the **header**

Write text in markdown syntax

Embed R code in chunks

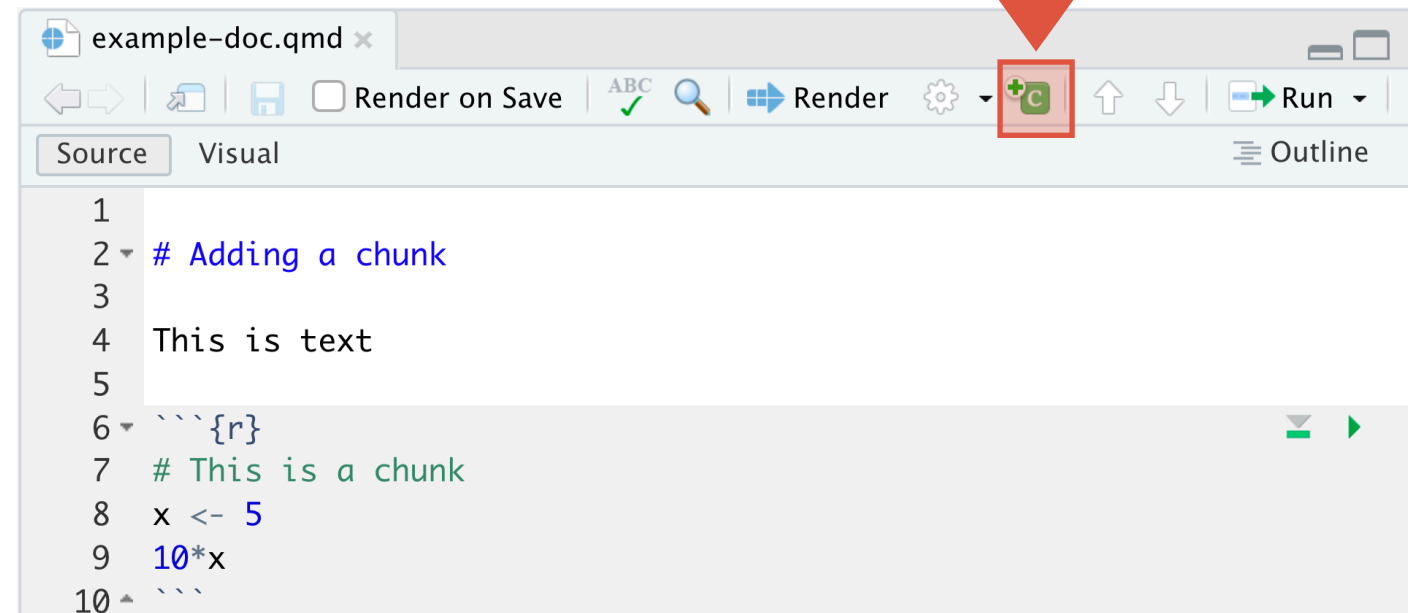Render your document

Share your work

Quarto step-by-step
4. Embed R code into your document

- R code is written within **chunks**

  - Shortcut on Mac: ⌘⌥ i

- Each chunk starts with a line ```` ```{r} ```` and ends with ```` ``` ````

- Within the chunk you write R code just as you already know it

- You can refer to variables defined in previous chunks

  - In principle you can refer to all objects defined previously in your session

  - But make sure that they were defined **in chunks above the one you work on** → doing otherwise would cause problems when rendering the file

  - To execute the chunk or all previous chunks you might use the buttons:
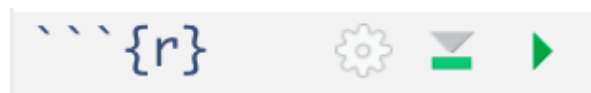
Quarto step-by-step
4. Embed R code into your document

- How the R code gets shown and executed in the final document is controlled via the chunk options

- Traditionally they were added to the first line of the chunk:

```
```{r name, echo=FALSE}
```
```

This is a chunk with `echo=TRUE` :

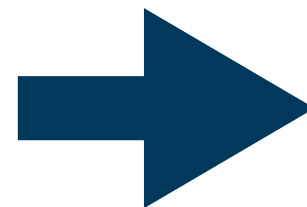```
2 + 2
```

```
## [1] 4
```

This is a chunk with `echo=FALSE` :

```
## [1] 4
```

- This modern variant is more consistent with Quarto syntax:

```
```{r, echo=FALSE}

```
```

```
```{r}
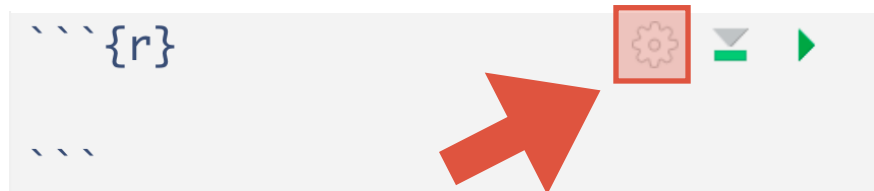#| echo: false

```
```

Quarto step-by-step
4. Embed R code into your document

- You get suggestions when pressing →ı

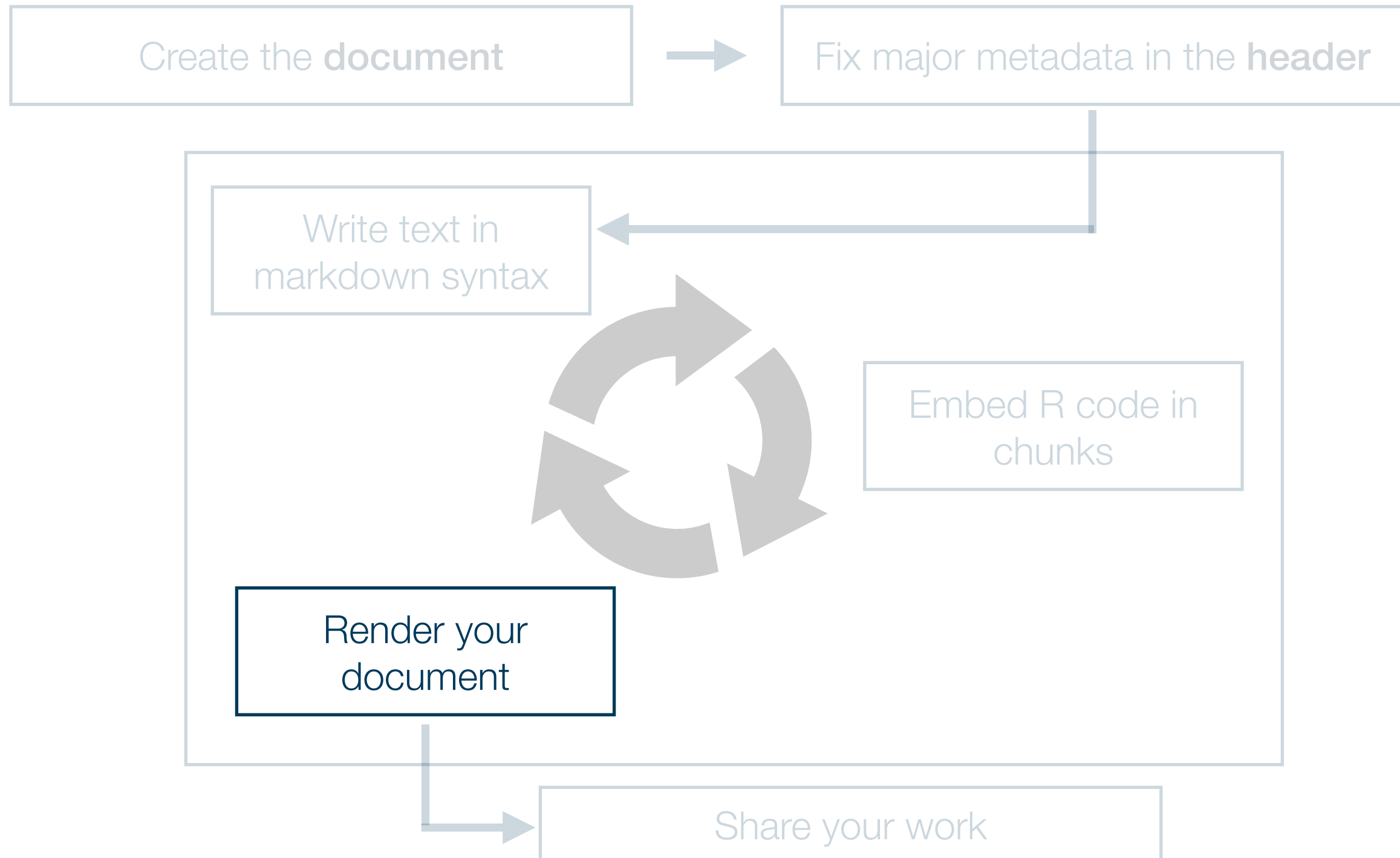- When creating a new chunk the point-and-click option menu is available:

- You can set default options for chunk options in the YAML header of the document under the keyword `execute`:

```
1 ---
2 title: 'The title'
3 author: "Claudius"
4 date: '2022-10-26'
5 execute:
6   warning: false
7   echo: false
```

- A full list of all chunk options can be found here:

https://quarto.org/docs/reference/cells/cells-knitr.html

Quarto step-by-step

Create the **document** → Fix major metadata in the **header**

Write text in markdown syntax

Embed R code in chunks

Render your document

Share your work

Quarto step-by-step
4. Render the documents

- This should in fact be done regularly during step 3
  - Otherwise its hard to identify the source of an error

- The format is determined in the header under `format`

Quarto step-by-step
4. Render the documents - examples for output options

The indented options below only apply to documents that are rendered into html format

General property sets (fonts, paragraphs, etc.)

Specification of the table of contents

Activate the numbering of sections

```
1  ---
2  title: 'GDP and development'
3  author: "Claudius"
4  date: '2022-10-26'
5  format:
6    html:
7      theme: pulse
8      toc: true
9      toc-depth: 2
10     toc-location: body
11   number-sections: true
12 ---
```

For a complete overview over output options and possible themes you should check the Quarto docs for the desired output format!

Europa-Universität Flensburg

Claudius Gräbner-Radkowitsch

24

Quarto step-by-step
4. Render the documents - implications of output options

```
1 ▾ ---
2   title: 'GDP and development'
3   author: "Claudius"
4   date: '2022-10-26'
5 ▴ ---
```

➡️

GDP and development

AUTHOR   PUBLISHED
Claudius   October 26, 2022

Packages used

```
library(DataScienceExercises)
library(ggplot2)
```

GDP and development indicators

While there are convincing critiques of GDP as a measure of well-being, there is also a clear relationship between GDP and socio-economic wellbeing indicators, such as life expectancy:
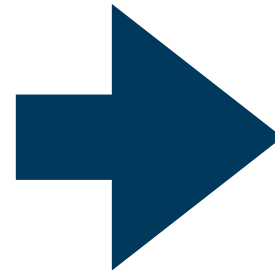
```
1 ▾ ---
2   title: 'GDP and development'
3   author: "Claudius"
4   date: '2022-10-26'
5   format:
6     html:
7       theme: pulse
8       toc: true
9       toc-depth: 2
10      toc-location: body
11      number-sections: true
12 ▴ ---
```

➡️

GDP and development

AUTHOR   PUBLISHED
Claudius   October 26, 2022

Table of contents

- 1 Packages used
- 2 GDP and development indicators
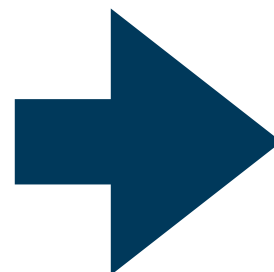- 3 Trends of divergence

1 Packages used

```
library(DataScienceExercises)
library(ggplot2)
```

2 GDP and development indicators

While there are convincing critiques of GDP as a measure of well-being, there is also a clear relationship between GDP and socio-economic wellbeing indicators, such as life expectancy:

Quarto step-by-step

Create the **document** → Fix major metadata in the **header**

Write text in markdown syntax

Embed R code in chunks

Render your document

Share your work

Quarto step-by-step
5. Share your work

- Many different possibilities → check the further readings

  - Key question is often about the format

- Here we consider a very straightforward solution for html-documents

  - Netlify Drop: https://app.netlify.com/drop

- Prerequisites:

  - You create an html document

  - All relevant output data is in one folder

  - You produce an output `index.html`
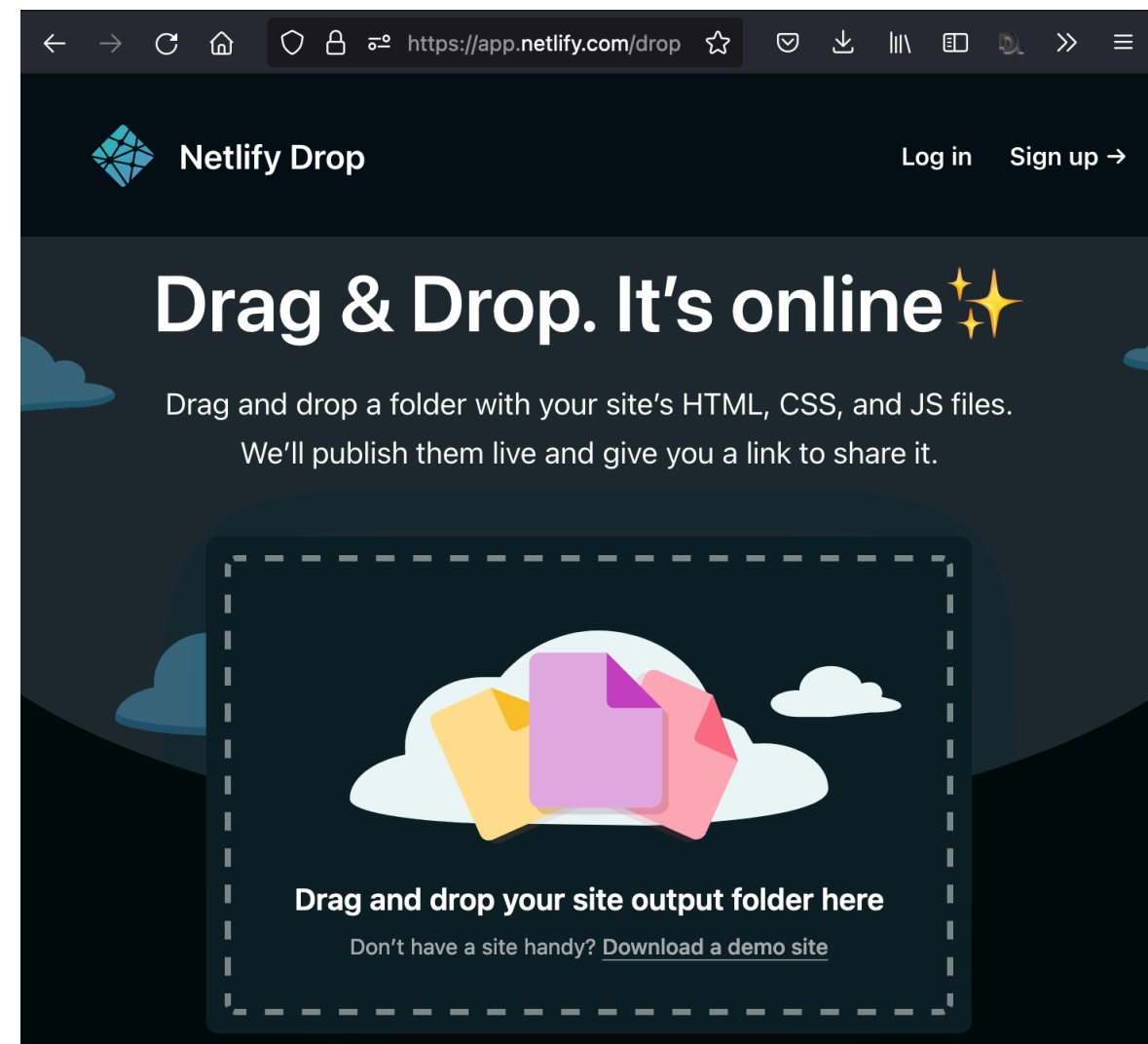
- Then its a nice way to distribute your document quickly
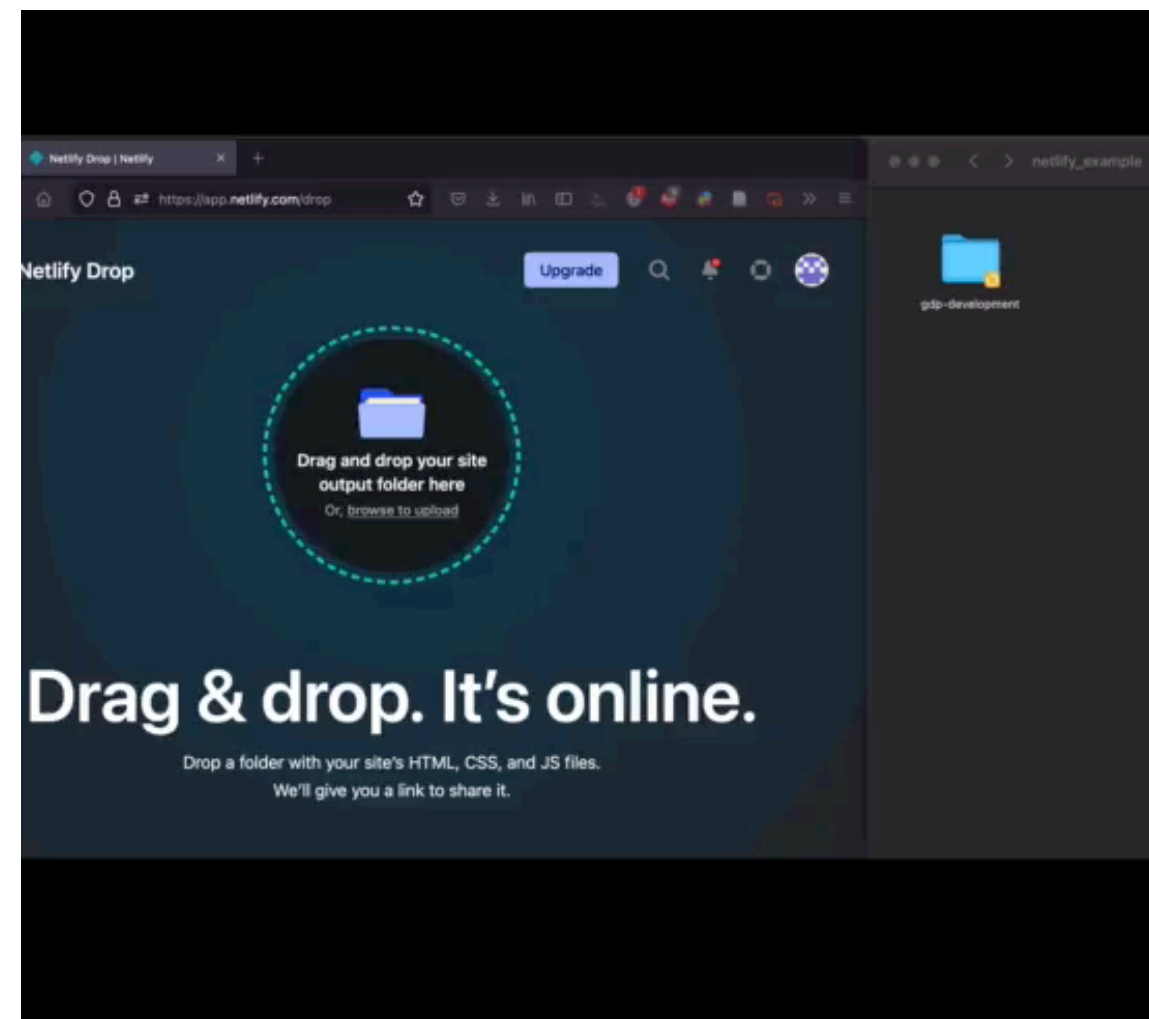
Quarto step-by-step
5. Share your work

- Many different possibilities → check the further readings

  - Key question is often about the format

- Here we consider a very straightforward solution for html-documents

  - Netlify Drop: https://app.netlify.com/drop

- Prerequisites:

  - You create an html document

  - All relevant output data is in one folder

  - You produce an output `index.html`

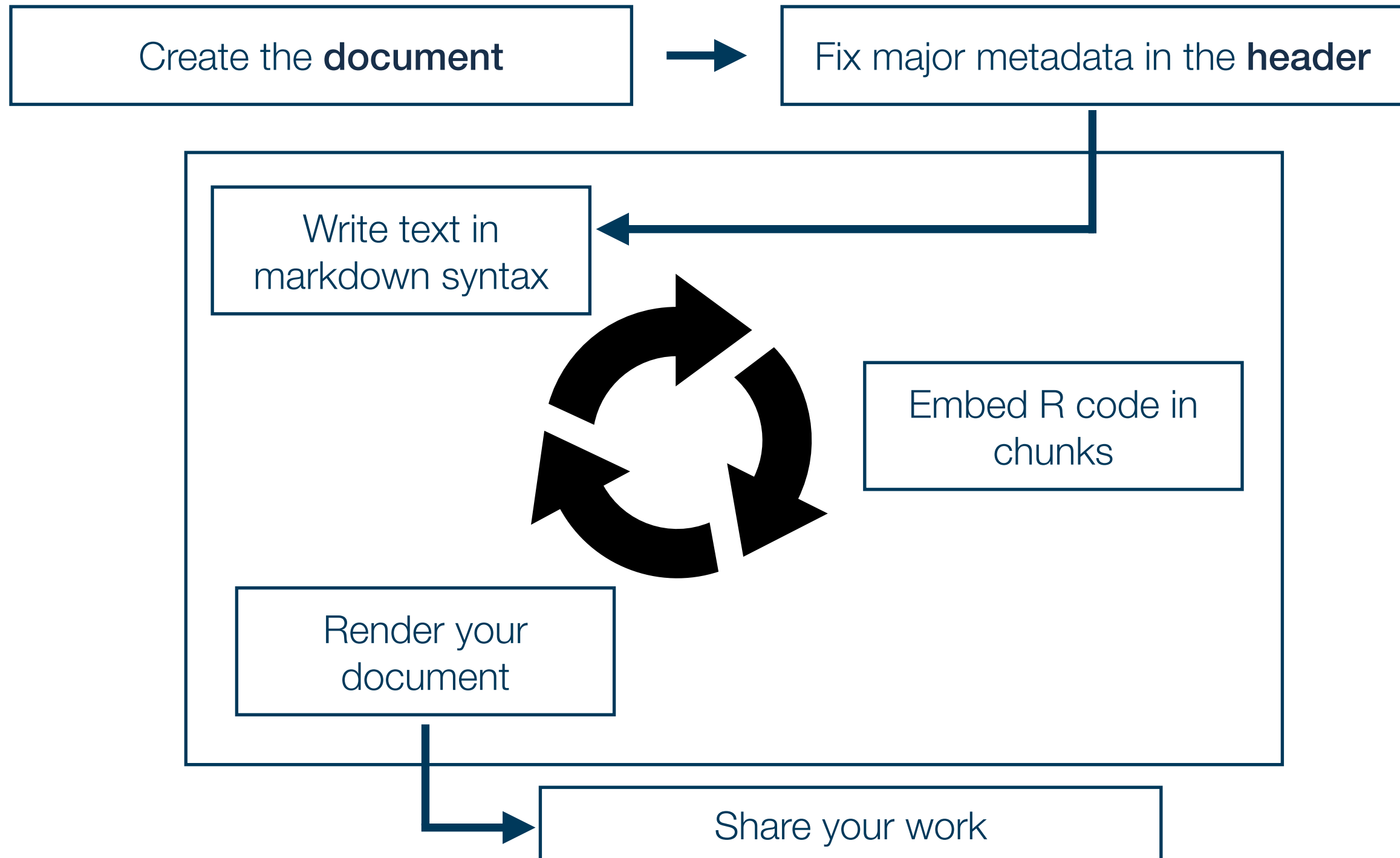- Then its a nice way to distribute your document quickly

  - If you are registered you can also choose a custom URL and much more

Quarto step-by-step

Create the **document** → Fix major metadata in the **header**

Write text in markdown syntax

Embed R code in chunks

Render your document

Share your work

Practice!

- Write and render a document!

- The text should include…

  - …a heading of level 1 and 2

  - …a text body

  - …a plot made with **ggplot2**

- Also add meta data on the **author**, **title**, and **date**

- Render the document into **html**

- Deploy the document via Netlify Drop and post the link on Moodle:

  - https://app.netlify.com/drop (Note: the .html file must be called **index.html**)

```
library(ggplot2)
ggplot(
  data = DataScienceExercises::aggGDPlifexp,
  mapping = aes(
    x = gdpPercap,
    y = lifeExp,
    color=continent)
  ) +
  geom_point() +
  theme_bw()
```

Some final remarks on project organisation

- The use of the `here` package is especially important when writing Quarto documents

  - When rendering an qmd document, your computer is **not** using your current working directory

  - Rather, the **working directory** is set to the **location of the qmd file**

  - This means that you cannot copy-paste code from R scripts that contains relative paths, except you use the `here`-package

- Its usually a good idea to put qmd-files either into the folder R or create a separate top level directory `quarto`

Avoid common mistakes

Avoiding common markdown mistakes

- There are some very common mistakes

  - These screw up you document considerable and make it painful to read…

  - …but are actually very easy to avoid

- Thus, after completing a markdown document, always look at the rendered version

  - Check whether any of the problems below still exists and eliminate them when necessary

Task:

Download `DesasterMarkdown.pdf` from the course homepage and collect the aspects that bother you the most!

The black list of markdown turpitudes

Problem	Solution
Overly long or wide tables	Only print what is necessary, check whether `str()` or `dplyr::glimpse()` are more adequate
Uninformative warnings or messages	Use the chunk options `warning` and `message` (maybe even set default to `false`)
Too many or too few code chunks visible	Make sure you used the chunk options `echo` and `include` correctly
Important output is missing	Make sure you used the chunk options `include` correctly
Figures in inappropriate sizes	Make sure to adjust `out-width` and `out-height` / `fig-width` and `fig-height`

- Upgrade: make tables pretty with `knitr::kable()` and `kableExtra`

- See the overview: https://quarto.org/docs/reference/cells/cells-knitr.html

Summary & outlook

Summary and outlook

- Quarto allows you to write documents that contain…

  - R code to perform statistical analysis

  - Markdown code to create formatted text using a plain text editor

- This allows you to do and describe your statistical analysis within one consistent document → makes research fully transparent and reproducible

- We covered the main steps of working with Quarto:

  - (1) create the documents, (2) set meta-data in the header, (3) write text, (4) embed R code, (5) render the document, and (6) share the final result

  - To learn markdown, just do the interactive tutorial

- There are a few mistakes that might easily screw up your document😡, but are just as easy to avoid😏

Summary and outlook

- We have now covered all the fundamentals of data preparation

- This was the most important part of the lecture since these are the tools you *always* need

- Now we will turn to some more advanced programming techniques and statistical applications: theory and modelling

---

Tasks until next week:
1. Fill in the **quick feedback survey** on Moodle
2. Read the **mandatory readings** posted on the course page
3. Complete the **interactive Markdown tutorial** linked on the course page
4. Do the **exercises** provided on the course page and **discuss problems** and difficulties via the Moodle forum

---