

# **DS4SD PHD COURSE - REPRODUCIBLE WORKFLOWS**

RMarkdown, git and Github

Stefan Daume

22. November 2024

# SRC 2024 PHD COURSE 'DATA SCIENCE FOR SUSTAINABLE DEVELOPMENT'

*Reproducible Workflows Using R Markdown, Git And GitHub*

@stefandaume  
<https://scitingly.net/>  
stefan.daume@su.se

Stefan Daume  
Stockholm Resilience Centre, Stockholm University  
& Beijer Institute of Ecological Economics

22. November 2024

# **WHY LEARN R MARKDOWN & GIT/GITHUB?**

# KEY MOTIVATION

**Document** your analysis and enable **reproducibility** to follow **Open Science** principles.

Avoid repetitive and error-prone tasks.

**(R) MARKDOWN**

# YOU SHOULD USE R MARKDOWN IF YOU WANT TO ...

- integrate and document your data analysis dynamically, not statically
- concentrate on content rather than formatting
- share one document in many different formats (Markdown, PDF, Word, HTML)
- ensure correct citations and bibliographies
- switch between different citation formats
- ... and much more

# MARKDOWN VS MARKUP

**Markdown** allows us to concentrate on document structure and content. We can then worry about styling and presentation later.

**Markdown** is a type of **markup language** (like HTML), but it is lightweight and more readable.

# SOME TEXT WITH SIMPLE FORMATTING

This is a list:

- with some **bold** and
- some *italic* text.

And a [hyperlink](#) for good measure.



# MARKUP SAMPLES

## HTML

```
<p>This is a list:</p>
<ul>
<li>with some <strong>bold</strong>
and</li>
<li>some <em>italic</em> text.</li>
</ul>
<p>And a <a
href="https://bookdown.org/yihui/rmarkd
for good measure.</p>
```

## LaTeX

```
This is a list:

\begin{itemize}
\tightlist
\item
  with some \textbf{bold} and
\item
  some \emph{italic} text.
\end{itemize}

And a
\href{https://bookdown.org/yihui/rmarkd
{hyperlink} for good
measure.
```

# THE SAME WITH MARKDOWN

## Basic Markdown

```
This is a list:
```

```
* with some bold and  
* some italic text.
```

```
And a [hyperlink] (https://bookdown.org/yihui/rmarkdown/) for good measure.
```

# TYPICAL WORKFLOW WITH MARKDOWN:

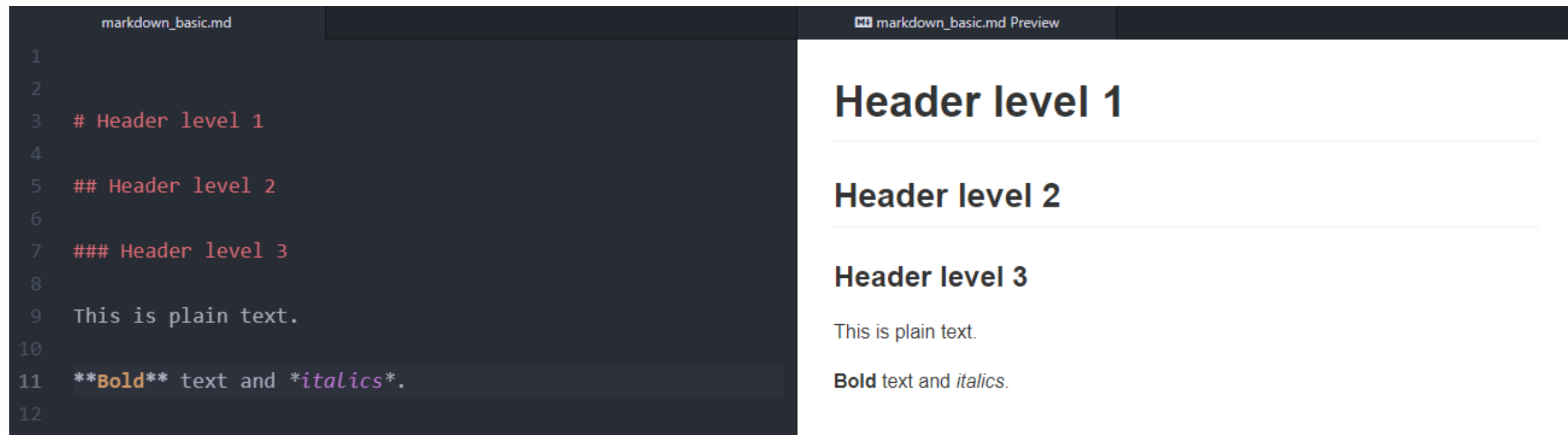
1. **write** content as a Markdown document,
2. **generate** the final document in a suitable output format (commonly HTML, PDF, Word)
3. **publish**

# ESSENTIAL MARKDOWN SYNTAX

# FILE STRUCTURE AND CONVENTIONS

- Markdown files are **simple text files** and can be created with any text editor.
- Markdown files typically end with the file extension **.md**

# BASIC FORMATTING AND STRUCTURING



The image shows a side-by-side comparison of a markdown file's source code and its rendered output. The left pane, titled 'markdown\_basic.md', contains the following source code:

```
1
2
3 # Header level 1
4
5 ## Header level 2
6
7 ### Header level 3
8
9 This is plain text.
10
11 Bold text and italics.
12
```

The right pane, titled 'markdown\_basic.md Preview', shows the rendered result of the source code:

- 'Header level 1' is rendered as a large heading with a horizontal line below it.
- 'Header level 2' is rendered as a medium heading with a horizontal line below it.
- 'Header level 3' is rendered as a small heading.
- 'This is plain text.' is rendered as a standard paragraph.
- '**Bold** text and *italics*.' is rendered with 'Bold' in bold font and 'italics' in italic font.

# LISTS AND LINKS

markdown_lists_links.md	markdown_lists_links.md Preview
<pre>1 Lists can be <b>**unordered**</b>: 2 3 * with each element started 4 * by an asterisk, potentially 5   * even as 6   * a nested list 7 8 or they can be <b>**ordered**</b>: 9 10 1. with each element started 11 2. by a number 12 1. ANY number, in fact 13 14 Hyperlinks with [custom labels](https://bookdown.org/yihui/   • rmarkdown/). 15 16 Precede with an exclamation mark for images: `[image   • title](path/to/image)` 17</pre>	<p>Lists can be <b>unordered</b>:</p> <ul style="list-style-type: none"><li>• with each element started</li><li>• by an asterisk, potentially<ul style="list-style-type: none"><li>◦ even as</li><li>◦ a nested list</li></ul></li></ul> <p>or they can be <b>ordered</b>:</p> <ol style="list-style-type: none"><li>1. with each element started</li><li>2. by a number</li><li>3. ANY number, in fact</li></ol> <p>Hyperlinks with <a href="#">custom labels</a>.</p> <p>Precede with an exclamation mark for images: <code>![image title](path/to/image)</code></p>

# EVEN TABLES

markdown_tables.md				markdown_tables.md Preview		
1						
2	Column 1	Column 2	Column 3			
3	-----	:-----:	-----:			
4	Left-aligned	Centered	Right-aligned			
5	is	Pipes are used	Colons determine			
6	the	to structure	the			
7	default.	the table.	alignment.			
8						

Column 1	Column 2	Column 3
Left-aligned	Centered	Right-aligned
is	Pipes are used	Colons determine
the	to structure	the
default.	the table.	alignment.

An overview of core markdown syntax can be found in [this R Markdown book chapter](#) and even more options in a condensed form as an [R Markdown cheat sheet](#).



# ‘R MARKDOWN’ VS ‘MARKDOWN’

- Purpose: dynamically weave together text, data and analysis workflows.
- This is accomplished with the `knitr` package, an R package conveniently integrated into the R Studio UI.

# DIFFERENCES TO BASIC MARKDOWN

- R Markdown files use the file extension `.Rmd` instead of `.md`.
- R Markdown files must start with a so-called **YAML header** section.
- R Markdown files are still text files but R Studio should be used to work with those files efficiently.

# YAML - YET ANOTHER MARKUP LANGUAGE?

The **YAML header** must be placed at the beginning of a document and is enclosed by three dashes ---.

```
---  
title: "Untitled"  
output: html_document  
date: '2024-11-22'  
---
```

Above is the default *YAML header* when creating a new R Markdown file in R Studio.

# YAML AIN'T MARKUP LANGUAGE!

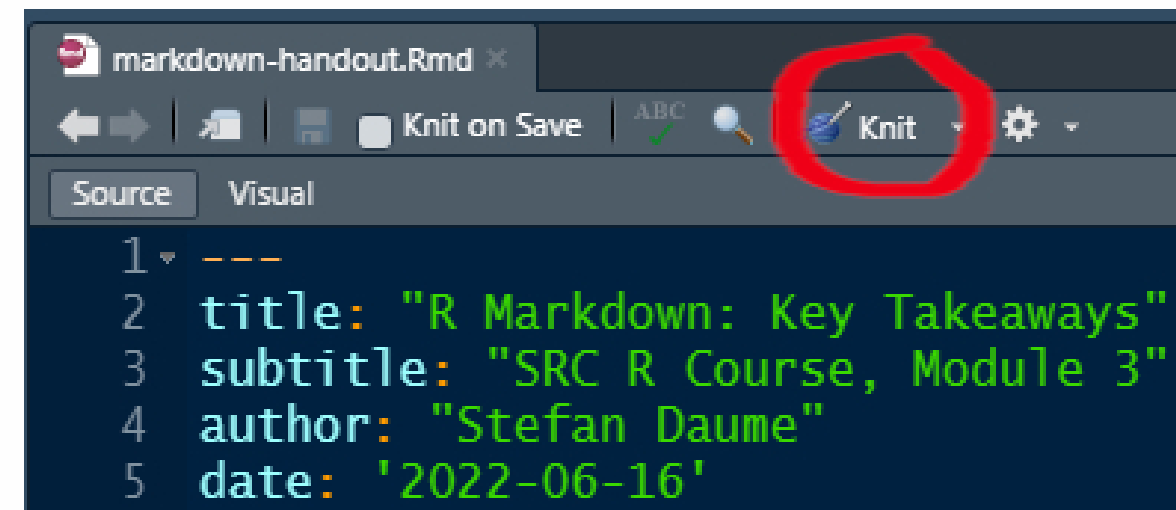
The **YAML header** contains meta-data (e.g. title, date, author(s) etc) as well as information about the output format and style.

A YAML header with more options might look like this:

```
---  
title: "R Course SRC"  
subtitle: "Module 3"  
date: "`r Sys.Date()`"  
author: 'Stefan Daume'  
output:  
  html_document:  
    toc: yes  
bibliography: references.bib  
link-citations: yes  
---
```

# EXERCISE

1. Create a default 'R Markdown' document in R Studio.
2. "knit" the document to HTML and view the result.
3. Use the **Knit** button to select different output formats and check the YAML header afterwards.



# R MARKDOWN: DATA-DRIVEN DOCUMENTS

- R Markdown allows to integrate your analysis as **R code** into the document
- The analysis (i.e. the R code) is executed and the results updated when you **knit** the document.
- Text and code are **interspersed**.
- Code sections are included in **code chunks** like this.

```
```${r some-explanatory-label, echo=FALSE}  
# here goes your R code  
```
```

# AN EXAMPLE FROM THE PREVIOUS SESSIONS

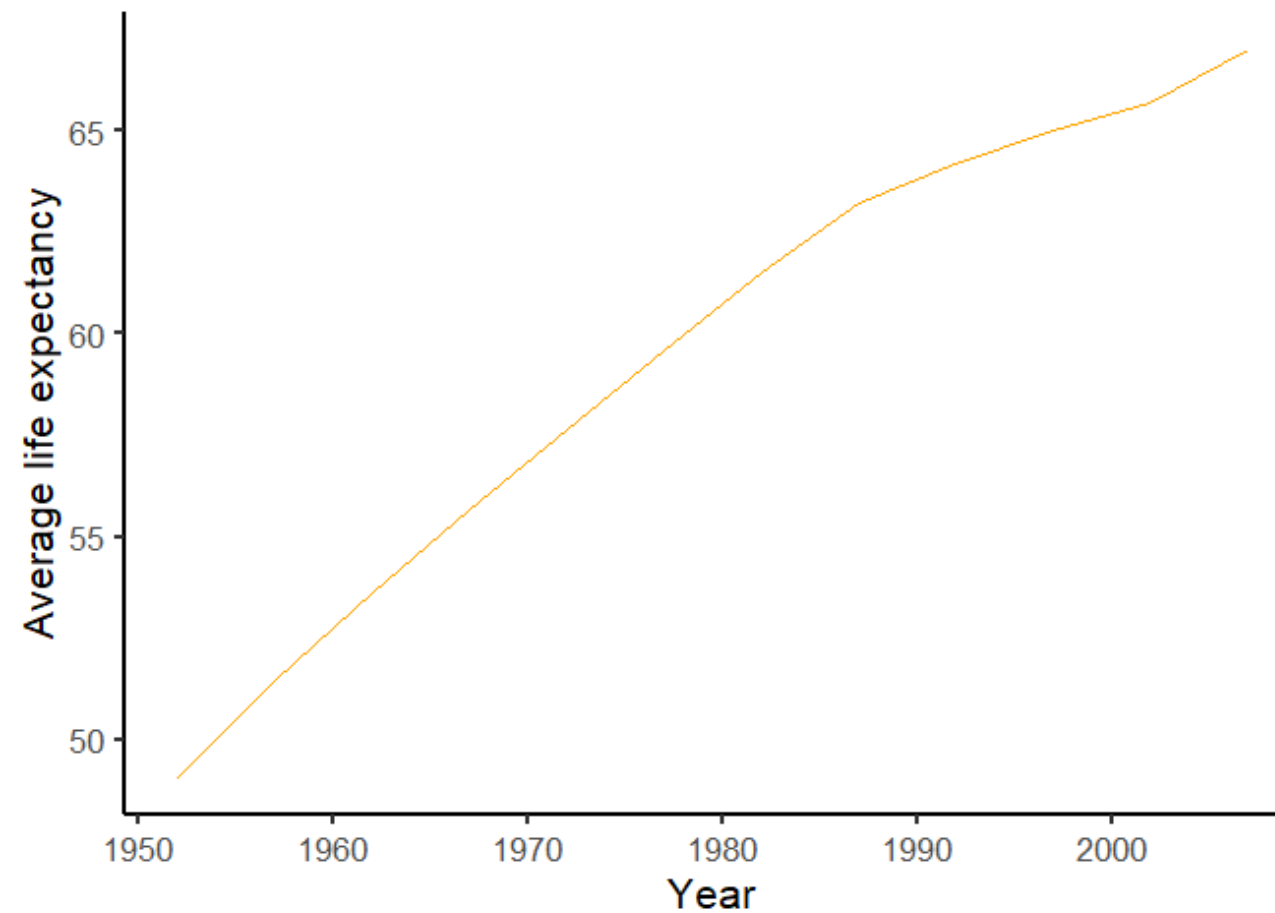
```
```{r life-expectancy, echo=FALSE, fig.cap="A figure caption."}
library(gapminder)

gapminder %>%
  group_by(year) %>%
  summarise(ae = mean(lifeExp)) %>%
  ggplot(aes(x = year, y = ae)) +
    geom_line(color = "orange") +
    labs(x = "Year",
         y = "Average life expectancy") +
    theme_classic(base_size = 16)
```
```

# PLOTS IN R MARKDOWN

```
```{r life-expectancy, echo=FALSE}
library(gapminder)

gapminder %>%
  group_by(year) %>%
  summarise(ale = mean(lifeExp)) %>%
  ggplot(aes(x = year, y = ale)) +
  geom_line(color = "orange") +
  labs(x = "Year",
        y = "Average life expectancy")
+
  theme_classic(base_size = 16)
```
```





# REMEMBER THE MARKDOWN TABLE FORMAT?

| markdown_tables.md |                                         | markdown_tables.md Preview |
|--------------------|-----------------------------------------|----------------------------|
| 1                  |                                         |                            |
| 2                  | Column 1   Column 2   Column 3          |                            |
| 3                  | -----  :-----:  -----:                  |                            |
| 4                  | Left-aligned   Centered   Right-aligned |                            |
| 5                  | is   Pipes are used   Colons determine  |                            |
| 6                  | the   to structure   the                |                            |
| 7                  | default.   the table.   alignment.      |                            |
| 8                  |                                         |                            |

| Column 1     | Column 2       | Column 3         |
|--------------|----------------|------------------|
| Left-aligned | Centered       | Right-aligned    |
| is           | Pipes are used | Colons determine |
| the          | to structure   | the              |
| default.     | the table.     | alignment.       |

# DYNAMIC TABLES WITH R MARKDOWN

This code ...

```
```{r}
# summarize gapminder data by
continent
gapminder_latest <- gapminder %>%
  filter(year == year_of_interest)
%>%
  group_by(continent) %>%
  summarise(avrg_le =
mean(lifeExp),
            avrg_gdp =
mean(gdpPercap))

# print the results as a table
gapminder_latest %>%
  knitr::kable()
```
```

... creates this table:

| continent | avrg_le  | avrg_gdp  |
|-----------|----------|-----------|
| Africa    | 54.80604 | 3089.033  |
| Americas  | 73.60812 | 11003.032 |
| Asia      | 70.72848 | 12473.027 |
| Europe    | 77.64860 | 25054.482 |
| Oceania   | 80.71950 | 29810.188 |

# CUSTOMIZING `kable` TABLES

This code ...

```
```{r}
# summarize gapminder data by
continent
gapminder_latest <- gapminder %>%
  filter(year == year_of_interest)
%>%
  group_by(continent) %>%
  summarise(avrg_le =
mean(lifeExp),
            avrg_gdp =
mean(gdpPercap))

# print the results as a table
gapminder_latest %>%
  knitr::kable(digits = c(0,1,2))
```
```

... creates this table:

| continent | avrg_le | avrg_gdp |
|-----------|---------|----------|
| Africa    | 54.8    | 3089.03  |
| Americas  | 73.6    | 11003.03 |
| Asia      | 70.7    | 12473.03 |
| Europe    | 77.6    | 25054.48 |
| Oceania   | 80.7    | 29810.19 |

# MORE EXPRESSIVE TABLES WITH `kableExtra` OR `gt`

The `kableExtra` and `gt` packages offer even more options:

- data-driven colouring
- interactive tables
- grouped headers
- tables with (interactive) sparklines
- and more ...

# kableExtra EXAMPLE

Table caption: Dynamic formatting with the the help of `kableExtra`. This example shows Gapminder data summarised by continent for the year 2007.

| Continent | Mean life expectancy | Mean GDP |
|-----------|----------------------|----------|
| Africa    | 54.8                 | 3089.03  |
| Americas  | 73.6                 | 11003.03 |
| Asia      | 70.7                 | 12473.03 |
| Europe    | 77.6                 | 25054.48 |
| Oceania   | 80.7                 | 29810.19 |

# gt EXAMPLE

Table caption: Dynamic formatting with the the help of the `gt` package. This example shows Gapminder data summarised by continent for the year 2007.

| Continent | Mean life expectancy | Mean GDP  |
|-----------|----------------------|-----------|
| Africa    | 54.8                 | 3,089.03  |
| Americas  | 73.6                 | 11,003.03 |
| Asia      | 70.7                 | 12,473.03 |
| Europe    | 77.6                 | 25,054.48 |
| Oceania   | 80.7                 | 29,810.19 |

# CENTRAL 'SETUP' CODE SECTION

```
```{r setup, include=FALSE}  
knitr::opts_chunk$set(echo = FALSE)  
  
library(readr)  
library(dplyr)  
library(ggplot2)  
library(gapminder)  
  
year_of_interest <- 2007  
```
```

Simplify library import and prepare datasets for reference in the whole document.

# HANDLING CITATIONS



# CITATIONS AND BIBLIOGRAPHIES

One of the most useful and powerful features for researchers using R Markdown.

# REQUIRES A BIBTEX DATABASE

A BibTeX database is simply a text file with the extension **.bib** and entries such as:

```
@misc{XieAllaire_et_2022,  
  author = {Xie, Yihui and Allaire, J. J. and Grolemond, Garrett},  
  title = {{R Markdown: The Definitive Guide}},  
  url = {https://bookdown.org/yihui/rmarkdown/},  
  urldate = {2022-06-07},  
  year = {2022}  
}
```

No need to write those. Export from your reference manager or journal pages.

# INCLUDE CITATIONS

Point to the `.bib` file in the  
YAML header.

```
---  
title: "R Course SRC"  
subtitle: "Module 3"  
date: "2024-11-20"  
author: 'Stefan Daume'  
output:  
  html_document:  
    toc: yes  
bibliography: references.bib  
link-citations: yes  
---
```

And then include citations in the  
text with the format

`[@CitationKey]`, which in  
the previously shown example  
was

`[@XieAllaire_et_2022]`,  
which is a reference to (Xie,  
Allaire, and Grolemund 2022).

# INCLUDE A BIBLIOGRAPHY

By default a bibliography is added to the end of the generated (i.e., `knitr`) document.

```
After presenting all results we have now reached the end of the document. Here  
should follow the bibliography.
```

```
# References
```

Add the header `# References` at the end of your document, `knit` and the complete bibliography is added to the output document.

# SWITCH CITATION AND BIBLIOGRAPHY STYLES DYNAMICALLY

Specify citation style in the YAML header.

```
---
title: "R Course SRC"
subtitle: "Module 3"
date: "2024-11-20"
author: 'Stefan Daume'
output:
  html_document:
    toc: yes
bibliography: references.bib
link-citations: yes
csl: ecology-and-society.csl
---
```

The [Citation Style Database](#) database contains thousands of journal [citation styles](#).

Download the relevant one, reference in the YAML header and the output document will have the required citation style.

# EASY SHARING AND ONLINE PUBLISHING

1. `knit` your R Markdown document to HTML
2. push the HTML to Github (next part of this module)
3. enable sharing of **Github Pages**

This is how this presentation works (and the others before).

**“CONTINUOUS ANALYSIS” AS THE ULTIMATE  
GOAL**

# KEY RESOURCES

- R Markdown
  - [R Markdown: The Definitive Guide](#) (Xie, Allaire, and Grolemund 2022)
  - [Cheatsheet: Dynamic documents with rmarkdown cheatsheet](#)



# **GIT & GITHUB**

# YOU NEED GIT AND GITHUB IF ... (NON-EXHAUSTIVE LIST)

- ... you have files like this, but realise that this is not efficient
  - my\_paper\_draft\_2021\_05\_16.docx
  - my\_paper\_draft\_2021\_05\_18.docx
  - my\_paper\_draft\_2021\_05\_19.docx
  - my\_paper\_draft\_2021\_05\_19\_v1.docx
  - my\_paper\_draft\_2021\_05\_19\_v2.docx
  - my\_paper\_draft\_2021\_05\_19\_v3\_with\_comments.docx
- ... you are not creating regular backups of your work
- ... you want to collaborate with others
- ... you want to maintain projects rather than a single file (Google Doc)
- ... you want to be able to easily revert back to previous versions of your work

# FOCUS OF THIS SESSION

git & GitHub are extremely versatile, feature-rich tools that enable collaboration on complex software projects.

# FOCUS OF THIS SESSION

We will only scratch the surface and focus on basic recipes and elements, namely:

- understanding the basic idea behind `git`
- use GitHub as a repository/backup for your work
- integrate git/GitHub into your workflow with R Studio
- share and collaborate with others

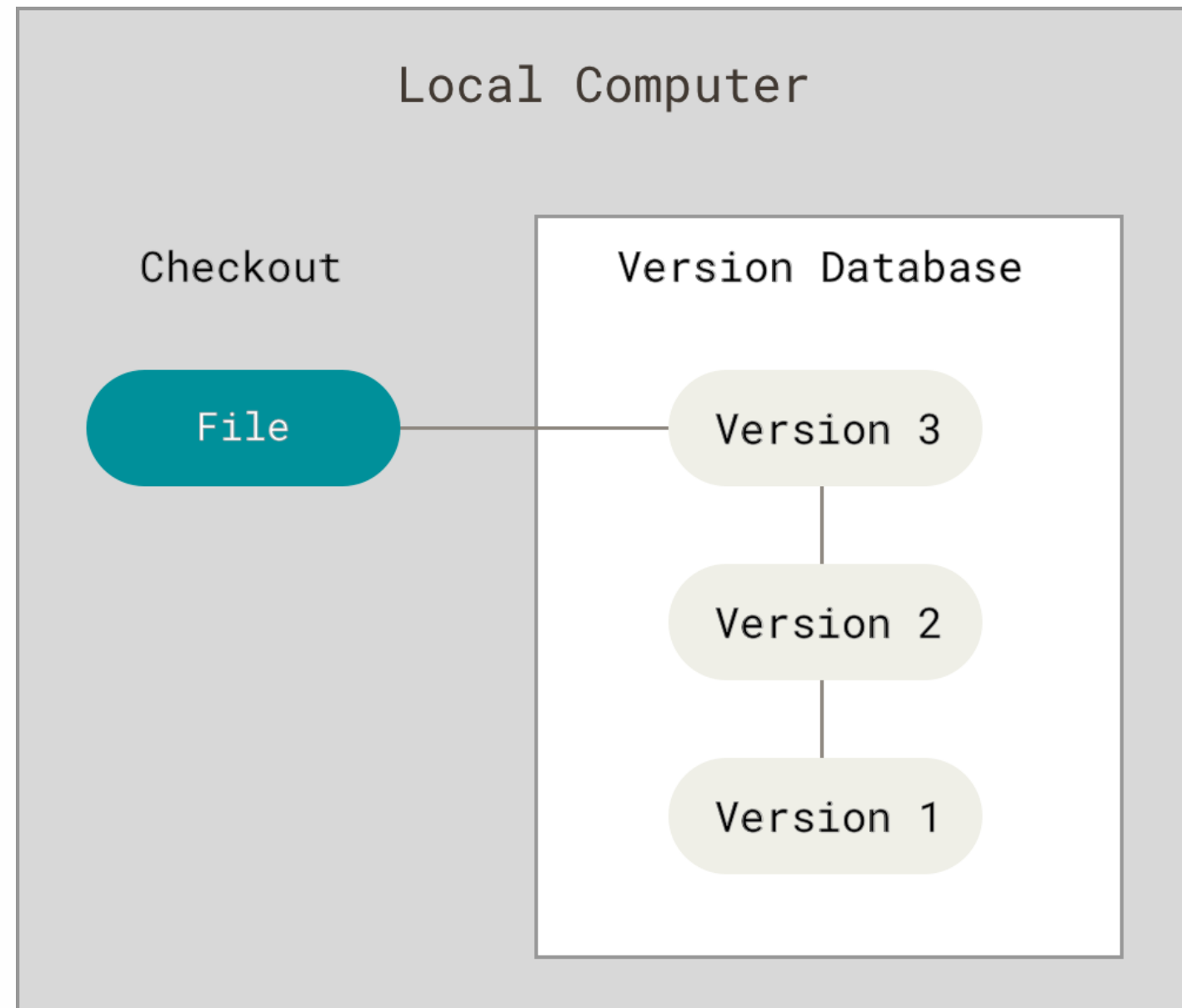
# KEY RESOURCES

- Git/Github:
  - [Happy Git and GitHub for the useR](#) (Bryan 2021)
  - “Excuse me, do you have a moment to talk about version control?” (Bryan 2017)
  - Advanced git use: [Pro Git](#) book (Chacon and Straub 2014)
  - [How to write a great commit message](#)

# GIT HISTORY

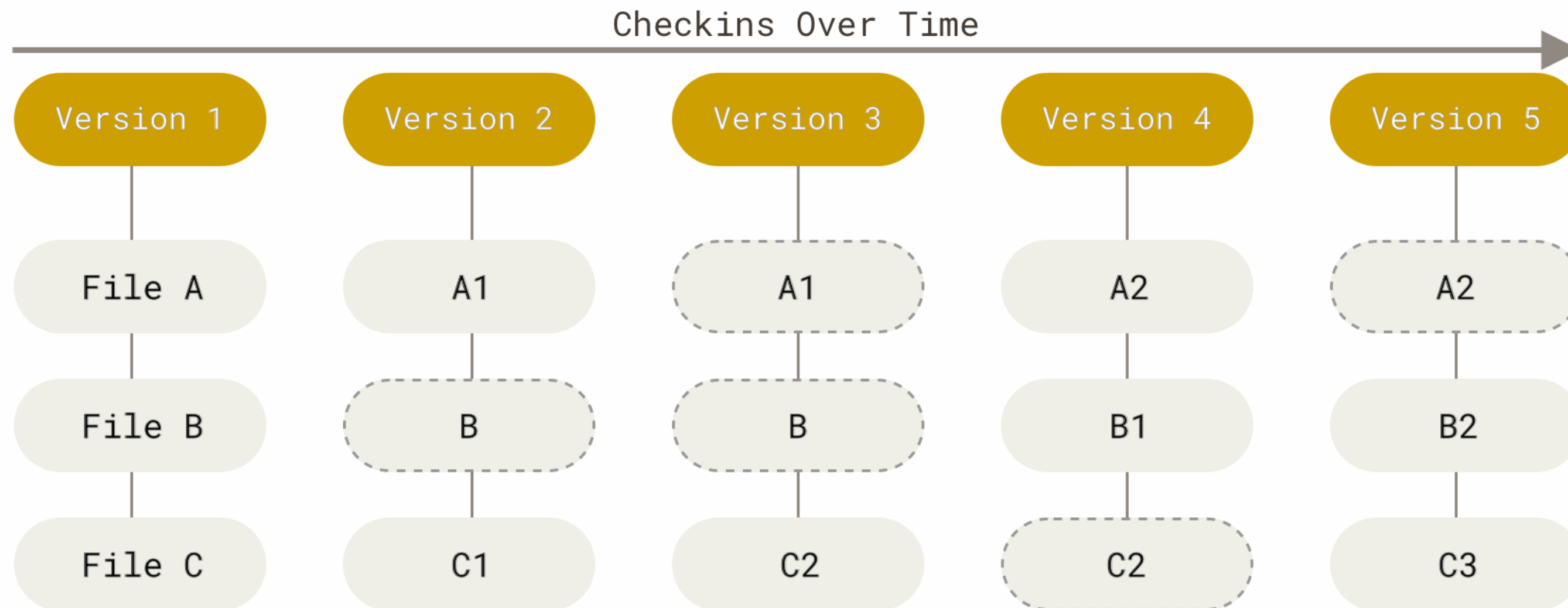
- Linux development, started 2005
- a version management system, i.e. tracks changes in project resources
- git takes snapshots of a managed project (image)
- distributed version control system (that means you always have a complete copy of your version history on your local computer)

# VERSION CONTROL



Local version control diagram, in [Pro Git](#) by Scott Chacon and Ben Straub, licensed under [CC BY-NC-SA 3.0](#)

# VERSIONS AS SNAPSHOTS



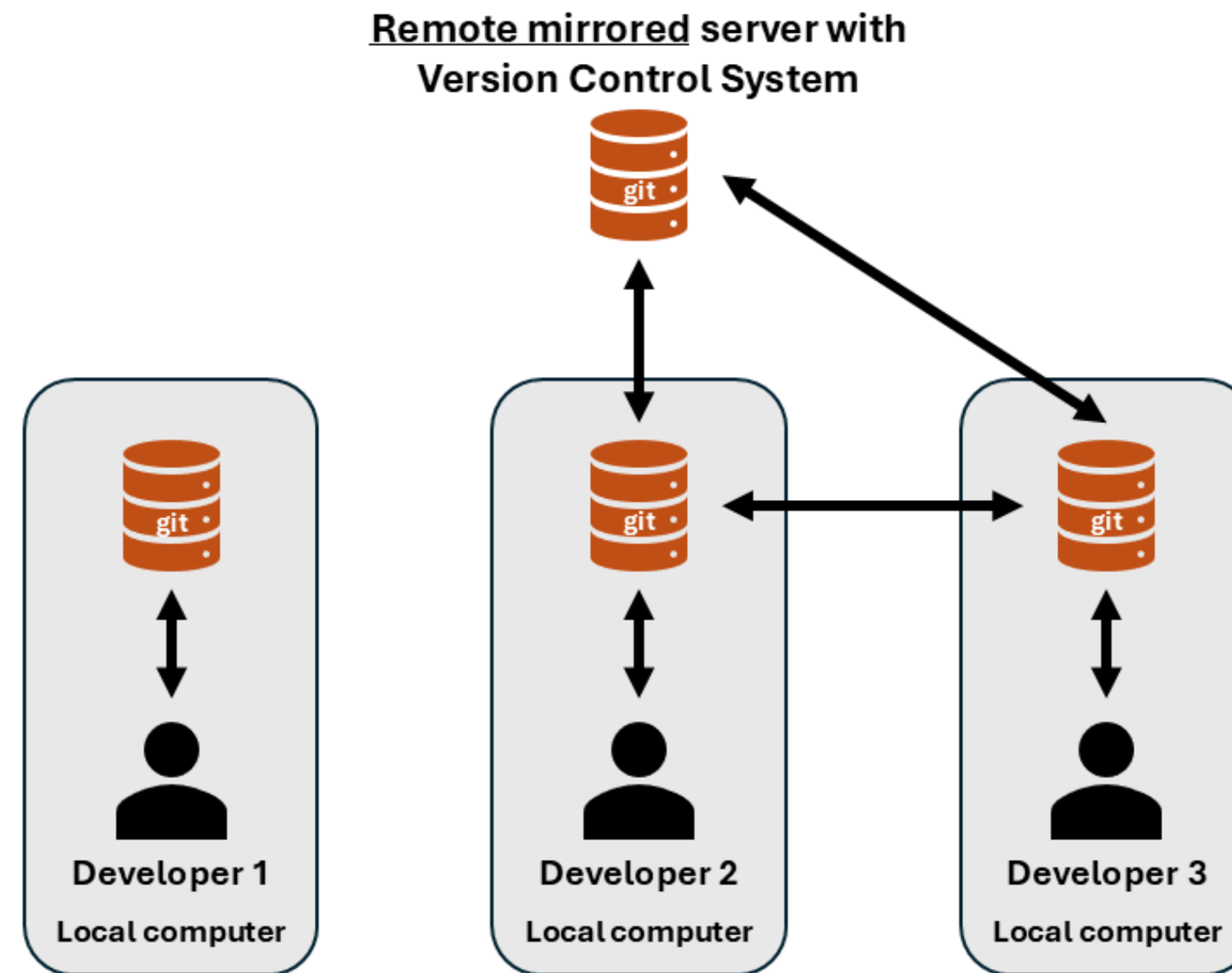
Versions as snapshots diagram, in [Pro Git](#) by Scott Chacon and Ben Straub, licensed under [CC BY-NC-SA 3.0](#)



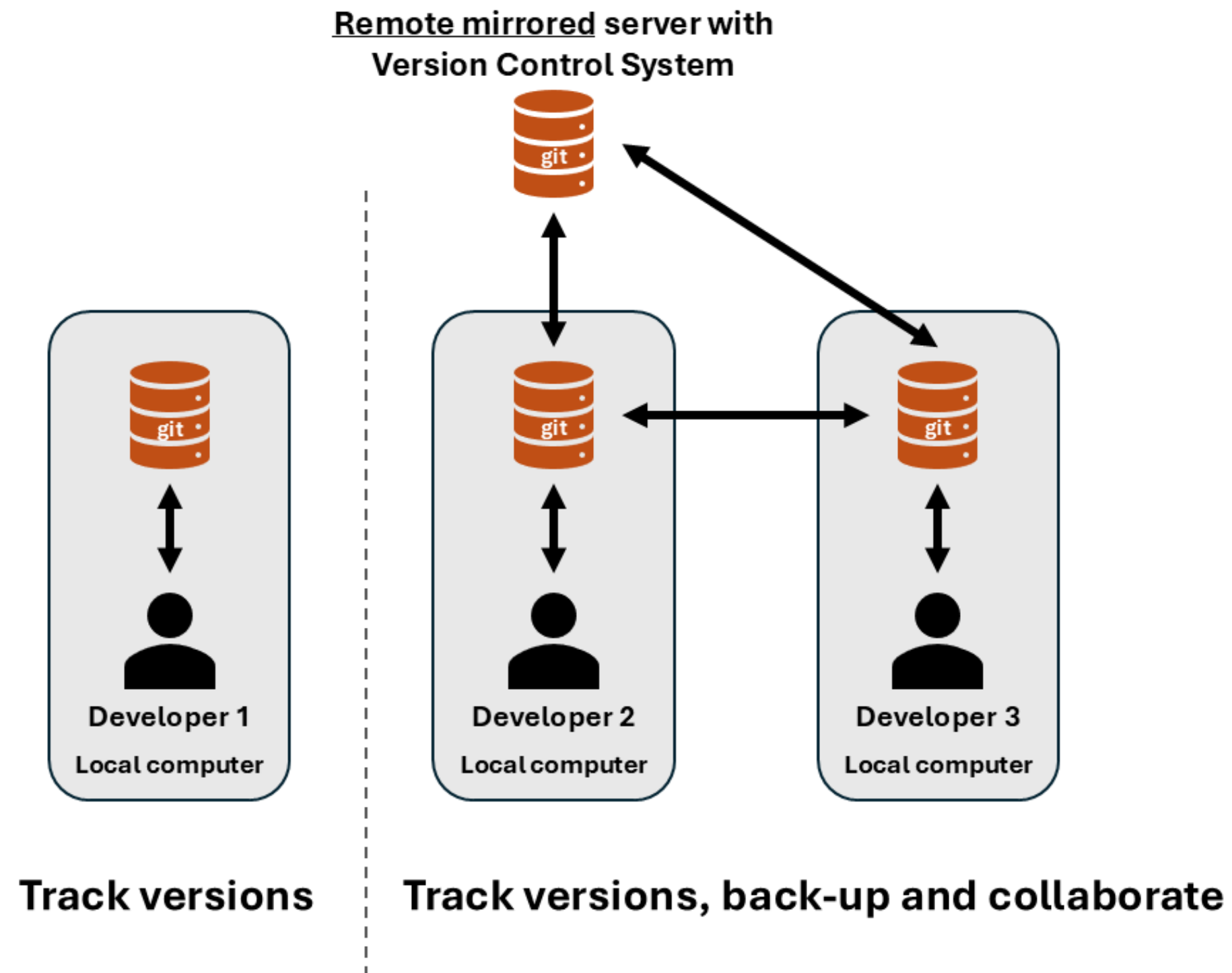
# CENTRALIZED VERSION CONTROL

Examples: CVS, Subversion

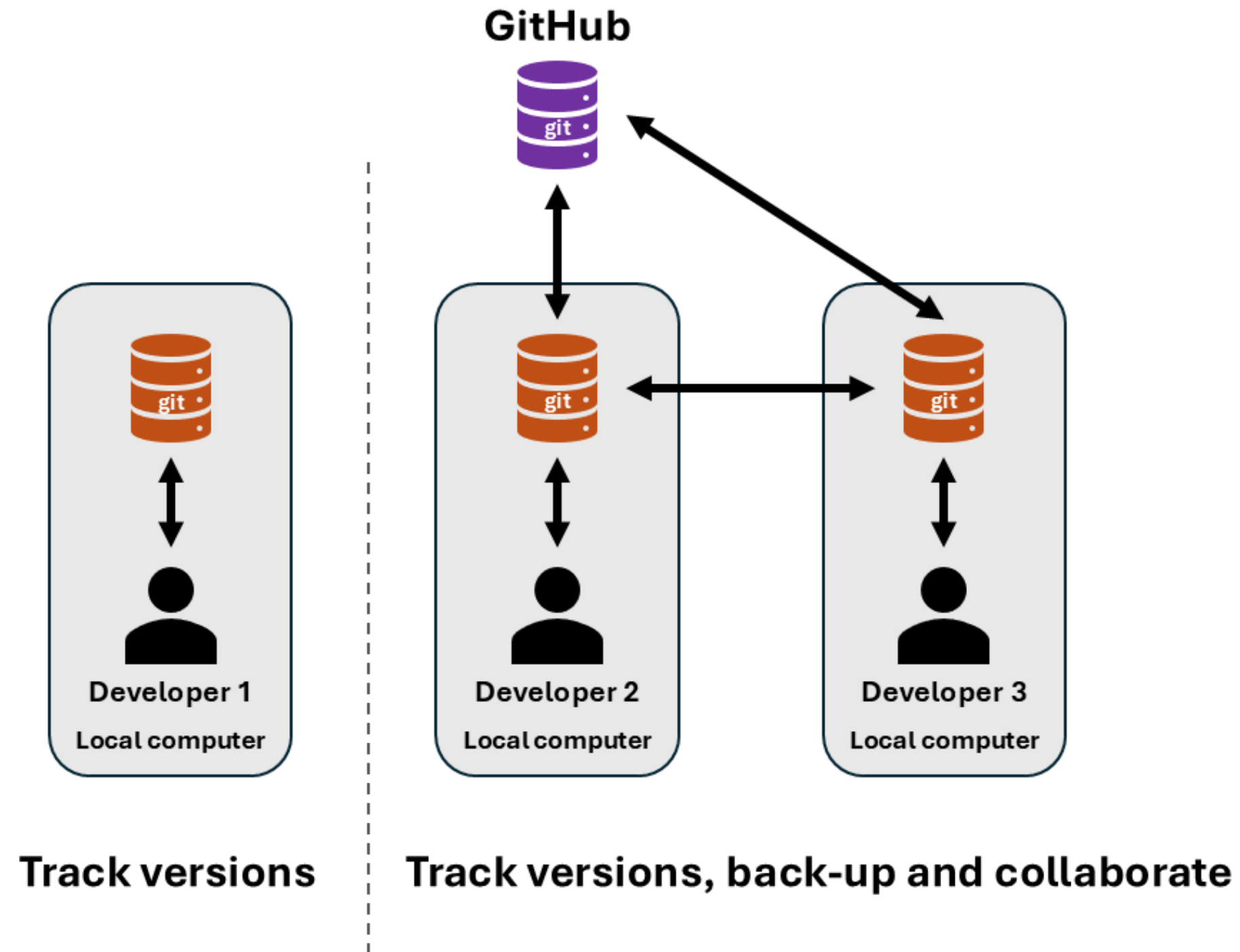
# DISTRIBUTED VERSION CONTROL



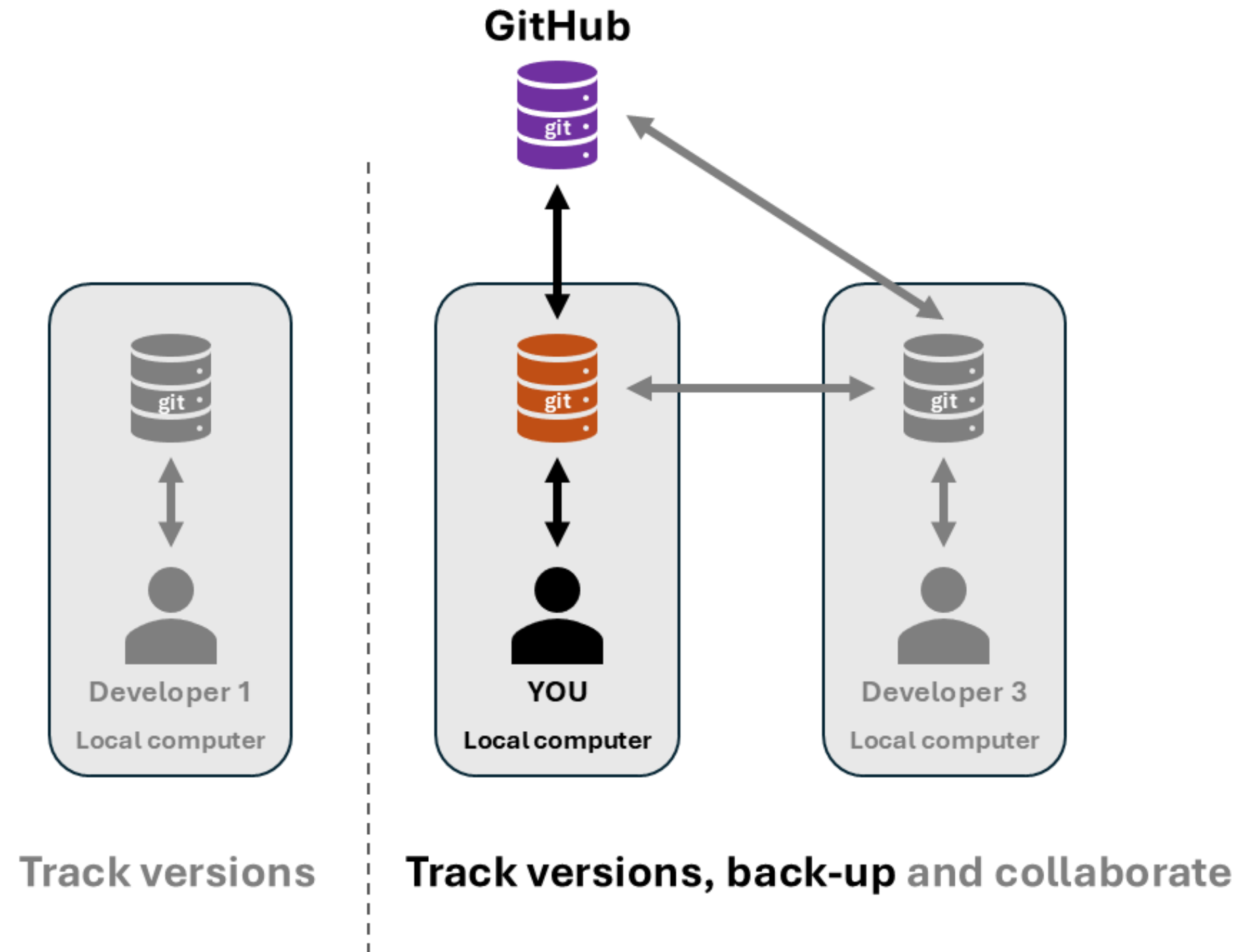
# DISTRIBUTED VERSION CONTROL



# GITHUB AS A HOSTED GIT REPO



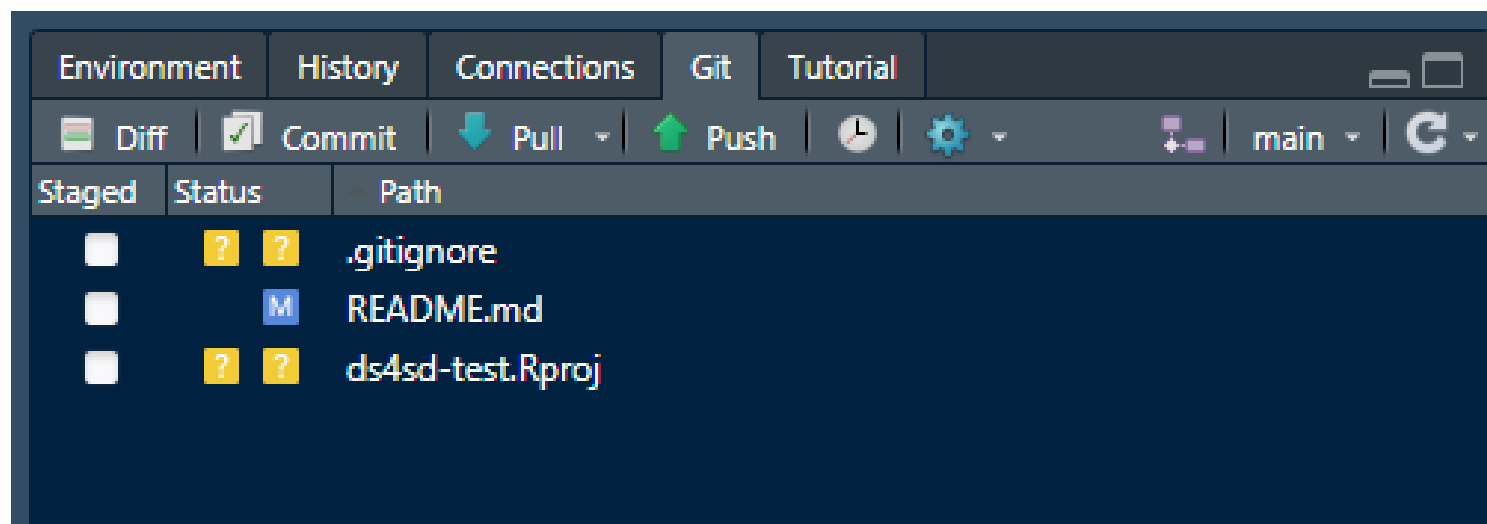
# FOCUS FOR TODAY



# A SIMPLE GIT/GITHUB USAGE SCENARIO

- create a project and **enable versioning** with `git`
- connect it with a **remote copy** (for sharing and backup)
- do work locally and track (**commit**) versions of your files
- **push** your changes (sync **to** the remote copy on GitHub)
- **pull** other's changes (sync **from** the remote copy on GitHub)

# INTEGRATED INTO R STUDIO



# KEY CONCEPTS

- repo
- cloning
- staging
- commit
- diff
- push
- pull
- branch (advanced)
- merge (advanced)
- remote origin



# SETTING GIT/GITHUB UP WITH R STUDIO

# DO THIS ONCE:

- sign up for a Github account
- install git locally (see (Bryan 2021))
- create a personal access token
  - either via Github (<https://github.com/settings/tokens>)
  - or via R with: `usethis::create_github_token()`
  - and then store it with `gitcreds::gitcreds_set()`

# INSTALLING AND CONFIGURING GIT

Select the installer for your OS: <https://git-scm.com/>

On the command line set:

```
$ git config --global user.name "JohnDoe"  
$ git config --global user.email johndoe@example.com
```

Use the same username/email you use for your GitHub account!

Check your settings:

```
$ git config --list
```

# CREATE A PAT (PERSONAL ACCESS TOKEN)

You can go to GitHub directly or trigger it from the command line:

```
uethis::create_github_token()
```

# CREATE A PAT (PERSONAL ACCESS TOKEN)

Configure and create PAT:

GitHub Apps

OAuth Apps

Personal access tokens

Fine-grained tokens

Tokens (classic)

Preview

## New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

**Note**

ds4sd test setup

What's this token for?

**Expiration \***

30 days

The token will expire on Sat, Dec 14 2024

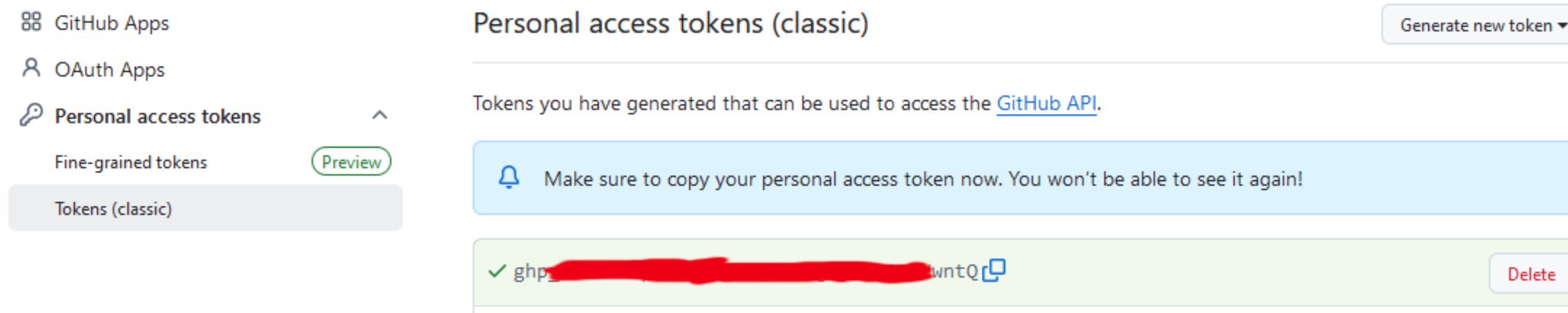
**Select scopes**

Scopes define the access for personal tokens. [Read more about OAuth scopes](#).

|                                                     |                                      |
|-----------------------------------------------------|--------------------------------------|
| <input checked="" type="checkbox"/> repo            | Full control of private repositories |
| <input checked="" type="checkbox"/> repo:status     | Access commit status                 |
| <input checked="" type="checkbox"/> repo_deployment | Access deployment status             |
| <input checked="" type="checkbox"/> public_repo     | Access public repositories           |
| <input checked="" type="checkbox"/> repo:invite     | Access repository invitations        |
| <input checked="" type="checkbox"/> security_events | Read and write security events       |
| <input checked="" type="checkbox"/> workflow        | Update GitHub Action workflows       |

# CREATE A PAT (PERSONAL ACCESS TOKEN)

Then copy it:



The screenshot shows the GitHub interface for managing Personal Access Tokens (classic). On the left sidebar, the navigation menu includes 'GitHub Apps', 'OAuth Apps', 'Personal access tokens' (which is expanded to show 'Fine-grained tokens' with a 'Preview' button and 'Tokens (classic)' which is selected), and 'Tokens (classic)'. The main content area is titled 'Personal access tokens (classic)' and includes a 'Generate new token' button. Below the title, a message states: 'Tokens you have generated that can be used to access the [GitHub API](#).' A blue notification box with a bell icon says: 'Make sure to copy your personal access token now. You won't be able to see it again!'. Below this, a list of generated tokens is shown. The first token is highlighted in green and has a checkmark icon. Its name is 'ghp' followed by a long string of redacted characters and ends with 'wntQ'. A copy icon is next to the token name. A 'Delete' button is located at the end of the token row.

Personal access tokens (classic) Generate new token ▼

Tokens you have generated that can be used to access the [GitHub API](#).

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp[redacted]wntQ Copy Delete

# STORE THE PAT FOR LOCAL USE

Set the credentials from the command line:

```
gitcreds::gitcreds_set()
```

Follow instructions and finally provide the PAT:

```
? Enter new password or token: ghp_XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
-> Adding new credentials...  
-> Removing credentials from cache...  
-> Done.
```

# ALTERNATIVE TO PATS

You can also configure **SSH keys** to connect to GitHub.

Consult [Set up keys for SSH](#) (Bryan 2021) to explore this option.



# DO THIS FOR EVERY NEW PROJECT:

- create a Github repo first (follow the [New project, Github first](#) workflow in (Bryan 2021))
  - Why? Its easiest! You have everything in place to create remote backups!
- say yes to creating a README
- copy the HTTPS link of your new repo
- then create an R Studio project with the option from “Version control > git”

# CREATE A NEW GITHUB REPO

- In your GitHub profile go to **Repositories**, and press “New”.
- Provide the repo information and press “Create Repository”.

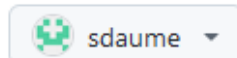
## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?

[Import a repository.](#)

*Required fields are marked with an asterisk (\*).*

Owner \*



Repository name \*

ds4sd-test

✓ ds4sd-test is available.

Great repository names are short and memorable. Need inspiration? How about [stunning-train](#) ?

Description (optional)

This is just a test repo for the ds4sd SRC course



**Public**

Anyone on the internet can see this repository. You choose who can commit.



**Private**

You choose who can see and commit to this repository.

Initialize this repository with:



**Add a README file**

This is where you can write a long description for your project. [Learn more about READMEs.](#)

# COPY THE REPO URL

- Go to **Repositories** and select the new repo.
- Copy the HTTPS repo URL.

The screenshot shows the GitHub interface for a repository named 'ds4sd-test' by user 'sdaume'. The repository is public and has 1 branch and 0 tags. The 'Code' dropdown menu is open, displaying the 'Clone' option with the HTTPS URL: `https://github.com/sdaume/ds4sd-test.git`. A button labeled 'Copy url to clipboard' is visible next to the URL. The repository description is 'This is just a test repo for the ds4sd SRC course'. The 'About' section on the right provides additional details: 0 stars, 1 watching, and 0 forks. The 'Releases' section indicates no releases are published and includes a link to 'Create a new release'.

Repository: **ds4sd-test** (Public)

Clone URL (HTTPS): `https://github.com/sdaume/ds4sd-test.git`

Repository Description: This is just a test repo for the ds4sd SRC course

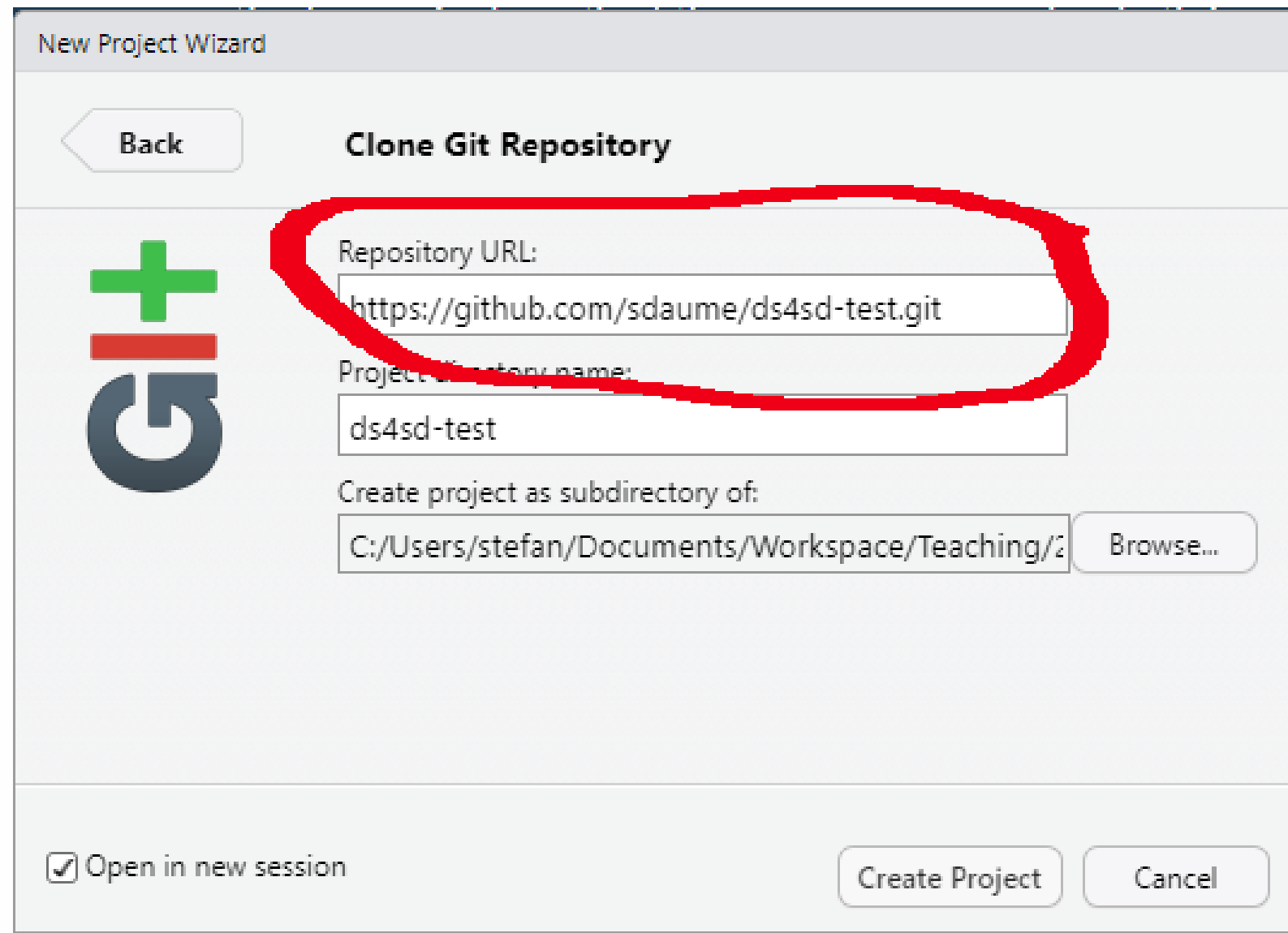
Repository Statistics:

- 0 stars
- 1 watching
- 0 forks

Releases: No releases published. [Create a new release](#)

# CREATE AN R STUDIO PROJECT WITH THE REPO

Create a new project via *File > New Project > Version Control > Git*



The screenshot shows the 'New Project Wizard' dialog box in R Studio, specifically the 'Clone Git Repository' step. The dialog has a 'Back' button on the left. On the left side of the main area is a large green plus sign above a red minus sign, which is above a large grey 'G' logo. The main area contains three text input fields: 'Repository URL:' with the value 'https://github.com/sdaume/ds4sd-test.git', 'Project directory name:' with the value 'ds4sd-test', and 'Create project as subdirectory of:' with the value 'C:/Users/stefan/Documents/Workspace/Teaching/2'. A red hand-drawn oval highlights the 'Repository URL' and 'Project directory name' fields. To the right of the 'Create project as subdirectory of:' field is a 'Browse...' button. At the bottom left is a checkbox labeled 'Open in new session' which is checked. At the bottom right are two buttons: 'Create Project' and 'Cancel'.

New Project Wizard

Back

Clone Git Repository

Repository URL:  
https://github.com/sdaume/ds4sd-test.git

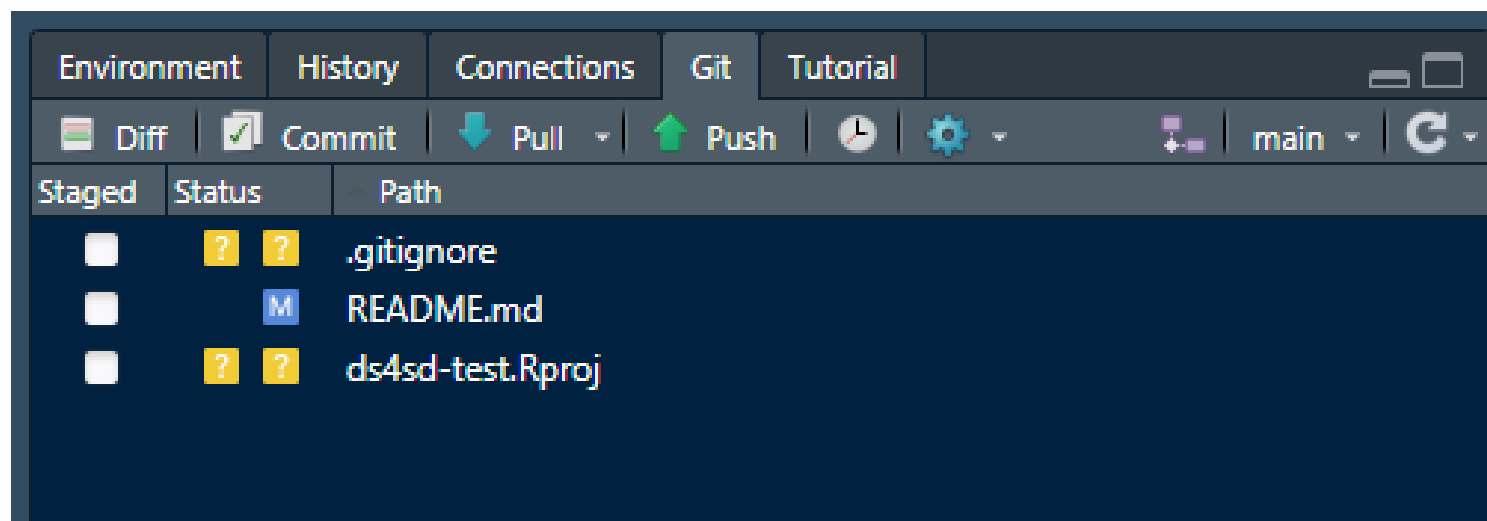
Project directory name:  
ds4sd-test

Create project as subdirectory of:  
C:/Users/stefan/Documents/Workspace/Teaching/2 Browse...

☒ Open in new session

Create Project Cancel

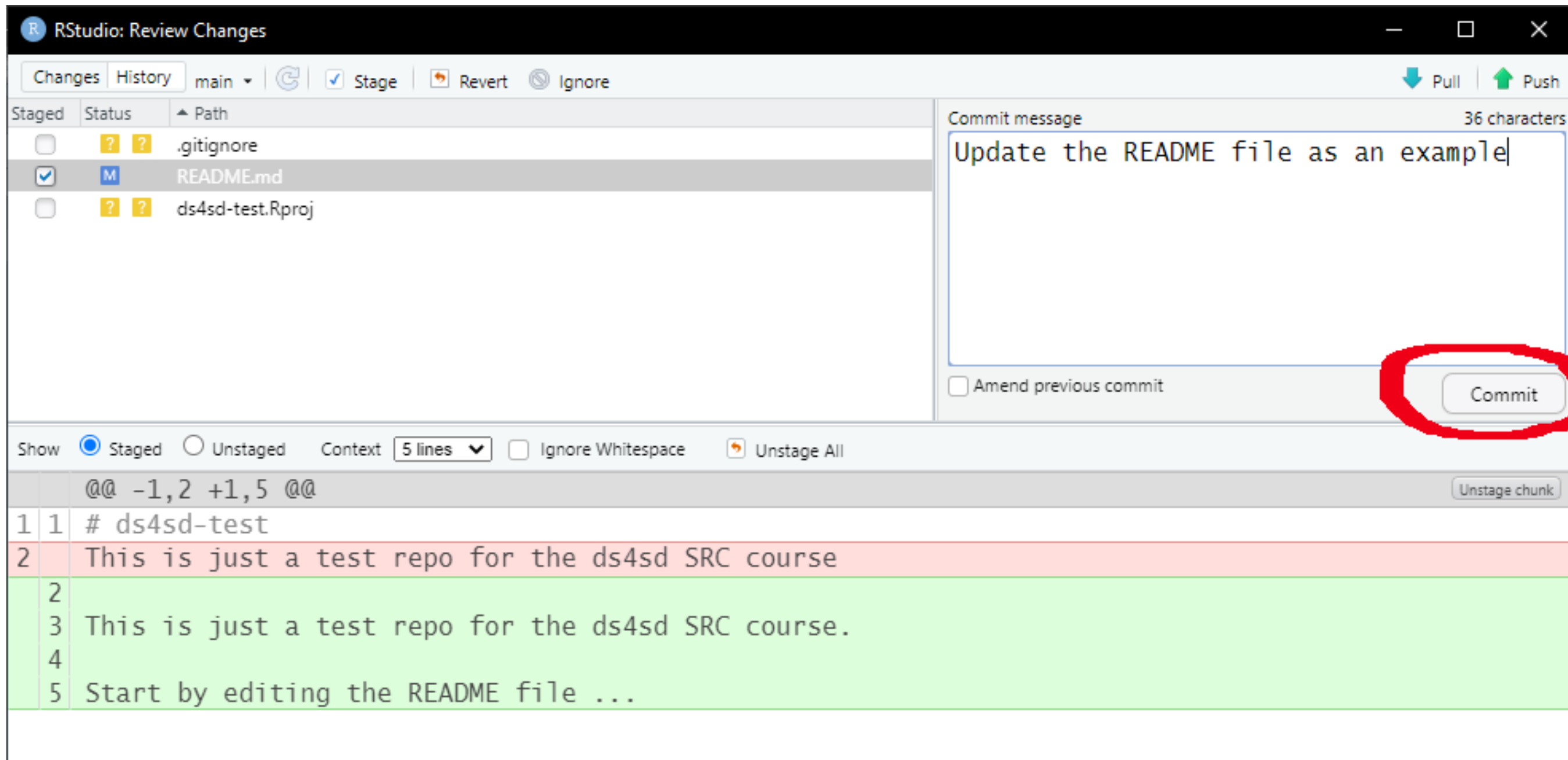
# NEW PROJECT TRACKED WITH GIT



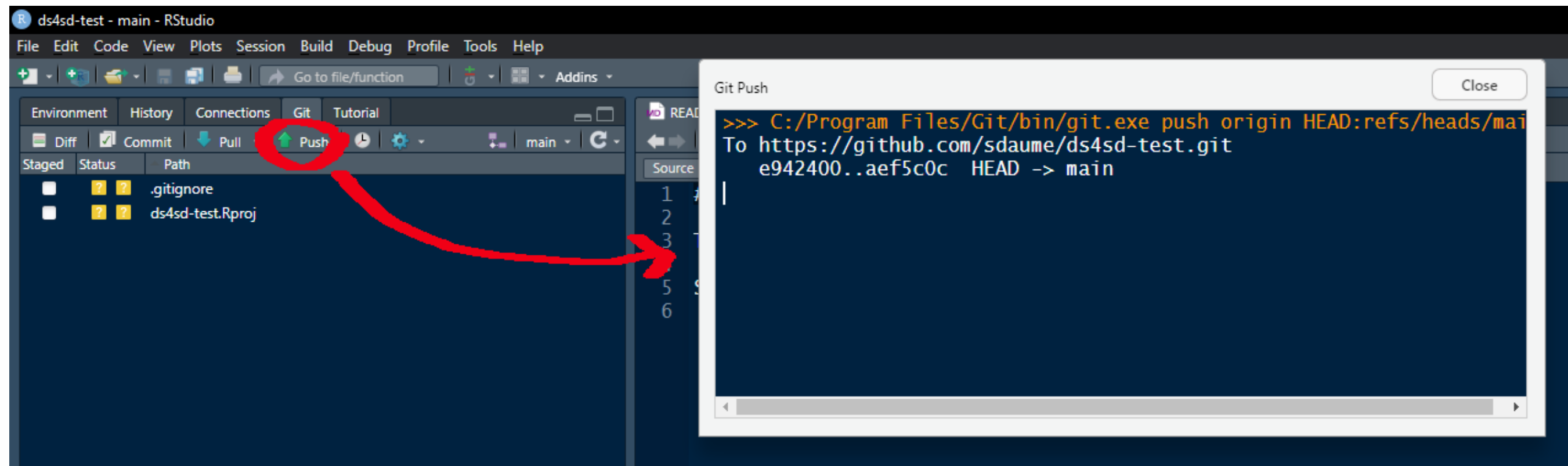
# WHEN YOUR NEW PROJECT IS SET UP

- make a change to the `README.md` (a useful project description)
- `commit` the changes of the `README` file
- and `push` to the remote Github repo
- check the Github repo

# COMMIT CHANGES



# PUSH CHANGES TO REMOTE GITHUB REPO



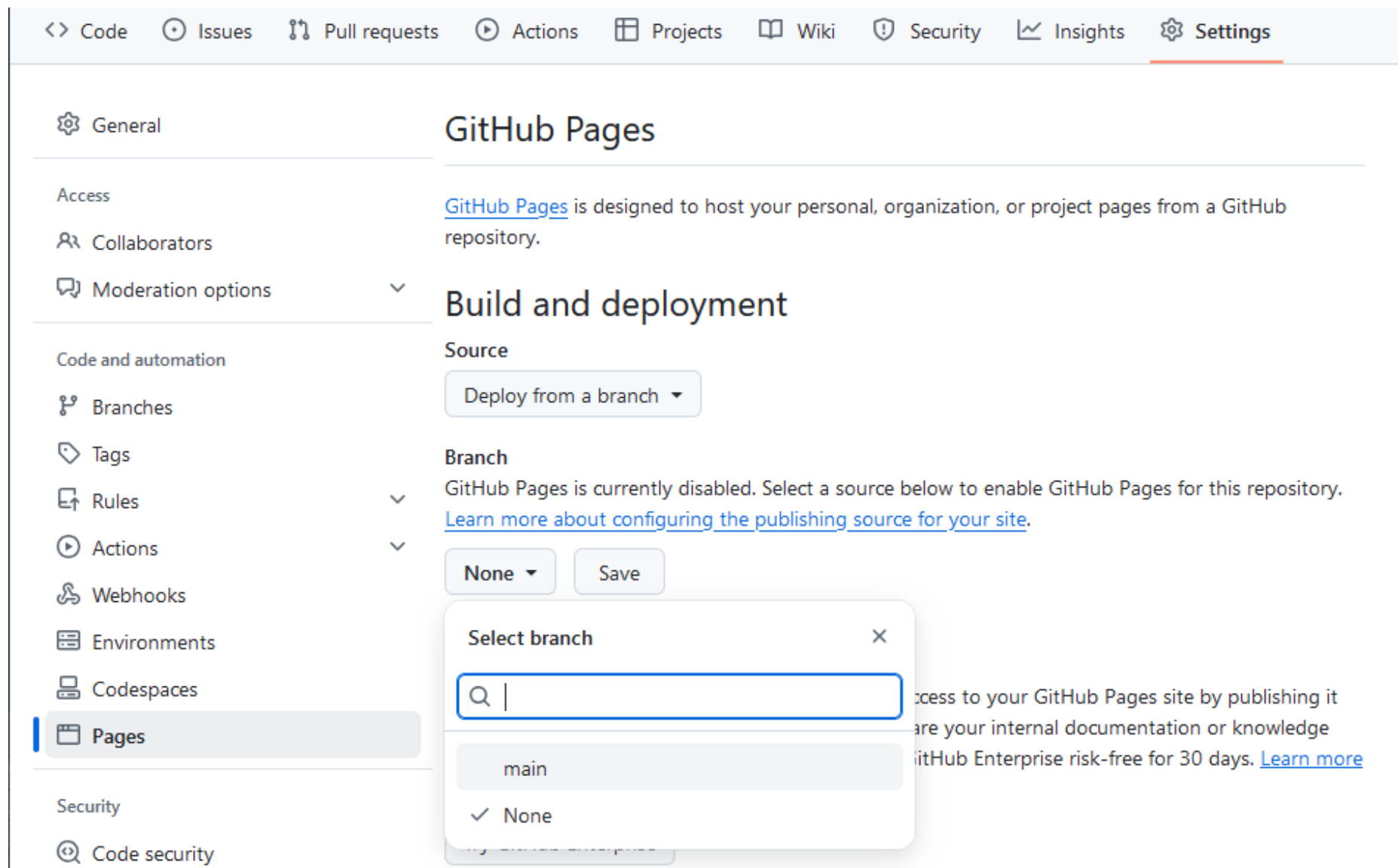


# PUBLISH YOUR CONTENT

Repo content can be hosted online via GitHub pages.

# ENABLE GITHUB PAGES FOR A REPO

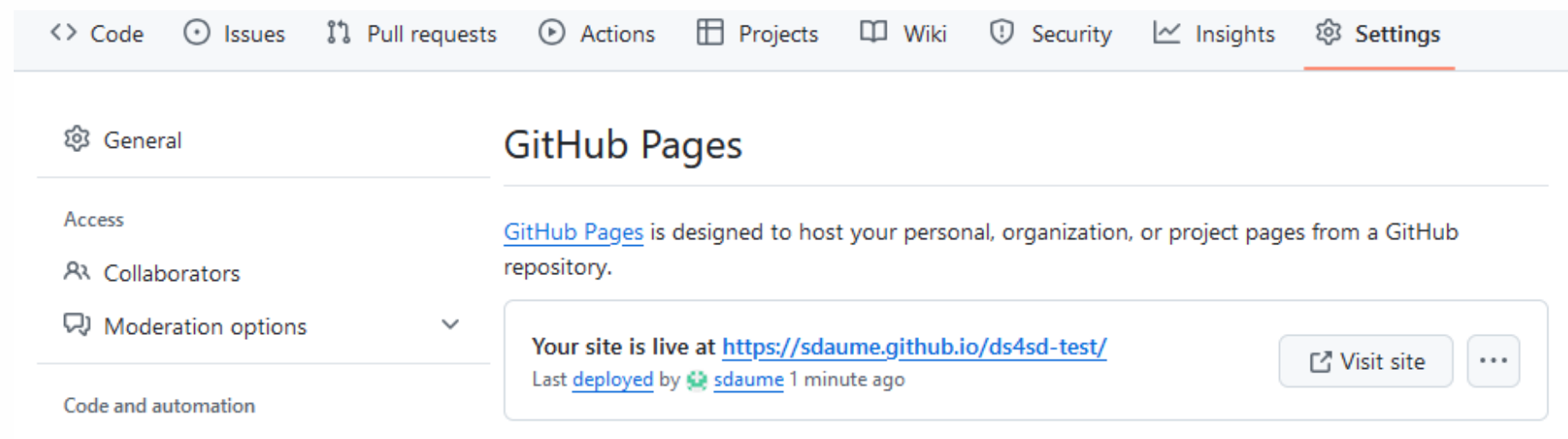
Go to *Settings* > *Pages* and select *Branch* > *main*



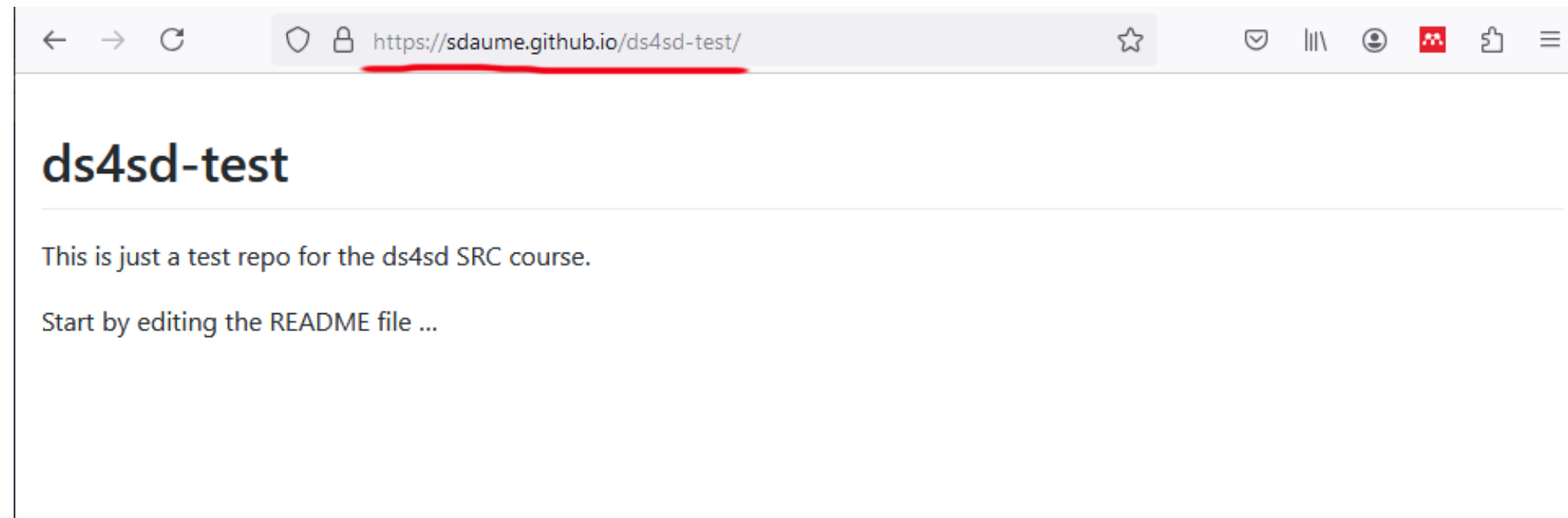
# ONE GITHUB PAGE PER REPO

Once enabled a site becomes available with the format:

`https://[GITHUB_USER].github.io/[REPO_NAME]/`



# DEPLOYED GITHUB PAGE



By default either the README is served or the content of a file called *index.html*, if it is available.

Alternatively, provide the filename in the URL, e.g.,  
<https://sdaume.github.io/ds4sd-test/default.html>

# **USEFUL TO KNOW FOR COMMITTS**

# NOT TRACKING RESOURCES

`.gitignore` allows to exclude resources from being tracked.

You may have sensitive files (e.g., pass keys, private data) that should not end up in a public repo.

# HOW TO WRITE A GREAT COMMIT COMMENT

Most important:

- Keep things atomic!

Document consistently:

- Keep the subject line short.
- Use the imperative mood in the subject line (Because a commit message should always complete the following line: “If applied, this commit will [YOUR\_SUBJECT\_LINE].”)
- Use the body to explain what and why vs. how (Because “the how” can be obtained from the *diff*. The commit message should provide the context for “the how”.)

# EXERCISES



# EXERCISE 1: SETUP GIT/GITHUB WITH R STUDIO

- Create a new repo on GitHub and
- Clone it as a new project in R Studio
- Edit the default README in your new R Studio project
- Commit the changes
- Push the changes to GitHub

# EXERCISE 2: CREATE AN R MARKDOWN DOCUMENT WITH DIFFERENT OUTPUT FORMATS

- In your new project create an R Markdown file
- Edit the file and insert
  - a simple plot with your own or Gapminder data
  - citation to references exported from your reference manager
- `knit` to the default output format (HTML)
- Try different output formats: PDF, Word

# EXERCISE 3: PUBLISH AN R MARKDOWN DOCUMENT VIA GITHUB

- Use your earlier R Markdown document
- `knit` to HTML, push to GitHub and **publish** the document
- Extra: Try to create a presentation as output

# REFERENCES

Bryan, Jennifer. 2017. “Excuse me, do you have a moment to talk about version control?” *PeerJ Preprints* 5:e3159v2 (August).  
<https://doi.org/10.7287/PEERJ.PREPRINTS.3159V2>.

———. 2021. “Happy Git and GitHub for the useR.”  
<https://happygitwithr.com/>.

Chacon, Scott, and Ben Straub. 2014. *Pro Git*. Apress.  
<https://doi.org/10.1007/978-1-4842-0076-6>.

Xie, Yihui, J. J. Allaire, and Garrett Grolemond. 2022. “R Markdown: The Definitive Guide.” <https://bookdown.org/yihui/rmarkdown/>.

# COLOPHON

SRC 2024 PhD course ‘Data Science for Sustainable Development’ — Reproducible Workflows using R Markdown and GitHub by *Stefan Daume*  
Presented on 22. November 2024.

## PRESENTATION DETAILS

**Author/Affiliation:** Stefan Daume, Stockholm Resilience Centre, Stockholm University

**Presentation URL:** <https://sdaume.github.io/ds4sd-2024-modules/workflows/slides/>

**Presentation Source:** <https://github.com/sdaume/ds4sd-2024-modules>

**Presentation PDF:** <https://github.com/sdaume/ds4sd-2024-modules/workflows/slides/2024-ds4sd-workflows.pdf>

## CREDITS & LICENSES

This presentation is delivered with the help of several free and open source tools and libraries. It utilises the [reveal.js](#) presentation framework and has been created using [RMarkdown](#), [knitr](#), [RStudio](#) and [Pandoc](#). [highlight.js](#) provides syntax highlighting for code sections. [MathJax](#) supports the rendering of mathematical notations. PDF and JPG copies of this presentation were generated with [DeckTape](#). Please note the respective licenses of these tools and libraries.

If not noted and attributed otherwise, the contents (text, charts, images) of this presentation are **Copyright © 2024 of the Author** and provided under a *CC BY 4.0* public domain license.