



How to do reproducible science?

The GitHub endeavor



Jérémie Carlot



CSIC
CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS



INSTITUTO
ESPAÑOL DE
OCEANOGRAFÍA



Class working plan



1) Theory and basics

- a) Markdown and Quarto documents
- b) Why is GitHub's fame increasing?
- c) How to link your GitHub account to statistical software and how to use it?
- d) Beyond data analysis

2) Practice

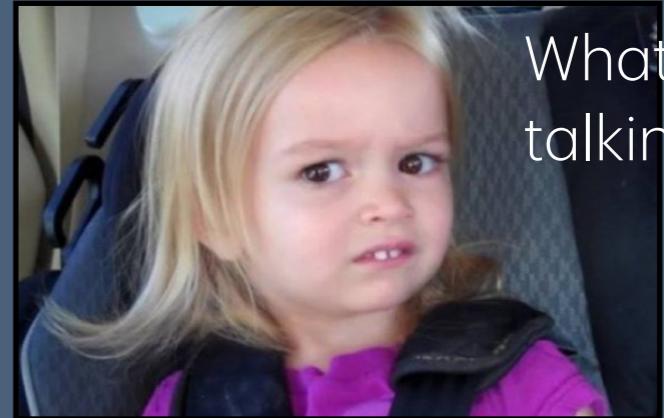
- a) Create a project to share
- b) Collecting data from previous peer-reviewed publication

Theory and basics

[>.....] 0%

Markdown and Quarto documents

Why people use Markdown instead
of a WYSIWYG editor?



What is he
talking about?

Theory and basics

[>.....] 0%

Markdown and Quarto documents

Why people use Markdown instead
of a WYSIWYG editor?



What is he
talking about?



Theory and basics

[>.....] 0%

Markdown and Quarto documents

Why people use Markdown instead of a WYSIWYG editor?



What is he talking about?



Portability (several apps)



Theory and basics

[>.....] 0%

Markdown and Quarto documents

Why people use Markdown instead of a WYSIWYG editor?



What is he talking about?



Portability (several apps)

Independence across systems



Theory and basics

[>.....] 0%

Markdown and Quarto documents

Why people use Markdown instead of a WYSIWYG editor?



What is he talking about?



Portability (several apps)



Independence across systems



Used beyond text editing



Theory and basics

[>.....] 0%

Markdown and Quarto documents

Why people use Markdown instead of a WYSIWYG editor?



What is he talking about?



Portability (several apps)



Independence across systems



Used beyond text editing



Free



Theory and basics

[>.....] 0%

Markdown and Quarto documents

Ok, so how does it work?

A markdown document is a document
with either .md or .markdown extension
(For quarto it would be .qmd)



Theory and basics

[>.....] 0%

Markdown and Quarto documents

Ok, so how does it work?

A markdown document is a document
with either .md or .markdown extension
(For quarto it would be .qmd)



Theory and basics

[>.....] 0%

Markdown and Quarto documents

Ok, so how does it work?

A markdown document is a document
with either .md or .markdown extension
(For quarto it would be .qmd)



The cool thing is that Markdown/Quarto are already
included for IDE softwares such as RStudio, Posit, Obsidian,
Visual Code...

Theory and basics

[>.....] 0%

Markdown and Quarto documents

But the real cool thing is that you can share dynamic and interactive documents (👉 Example)

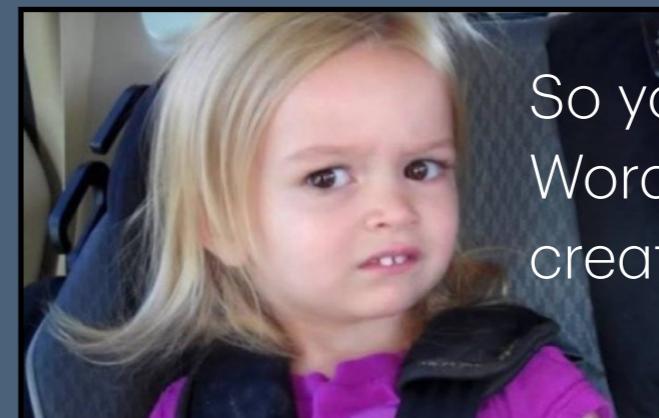
Theory and basics

[>.....] 0%

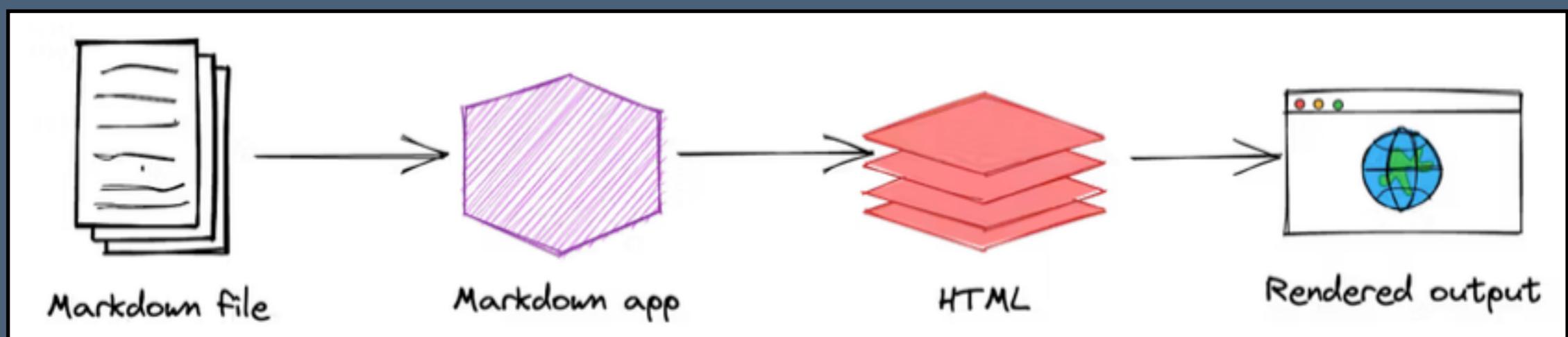
Markdown and Quarto documents

But the real cool thing is that you can share dynamic and interactive documents (👉 Example)

You can either export your document as a .pdf, a .html or .docx to share with others



So you're telling me Word is bad, but I can create word docs?



Theory and basics

[>.....] 0%

Markdown and Quarto documents

You can go even beyond classical documents: you can build your own slides using the `[xaringan]` package in R, or you can build your own website using `[blogdown]` or `[bookdown]`. Another option is to include shiny app within your .html document using the `[Shiny]` package. The resources are really vast!



Theory and basics

[>.....] 0%

Markdown and Quarto documents

OK. Concretely. How can I start my first Markdown doc?



File > New File > RMarkdown

The screenshot shows the RStudio interface. On the left, the 'New R Markdown' dialog box is open, showing fields for 'Title' (Document Title) and 'Author' (Your Name). Under 'Default Output Format', 'HTML' is selected, with a note that it's the recommended format. Below that are options for 'PDF' and 'Word'. On the right, the RStudio editor window displays the generated R Markdown code:

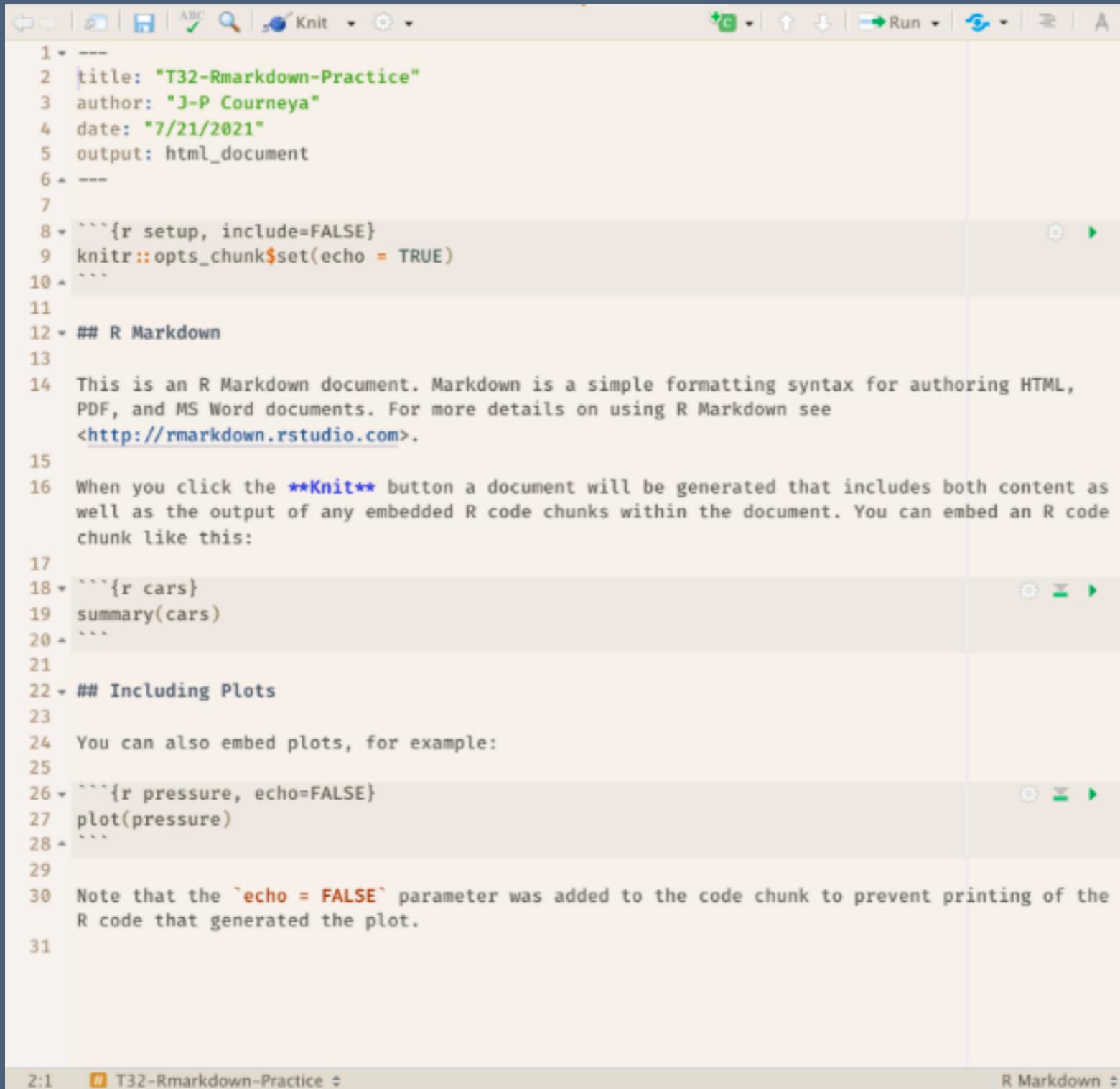
```
1 ---  
2 title: "T32-Rmarkdown-Practice"  
3 author: "J-P Courneya"  
4 date: "7/21/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see  
http://rmarkdown.rstudio.com.  
15  
16 When you click the Knit button a document will be generated that includes both content as  
well as the output of any embedded R code chunks within the document. You can embed an R code  
chunk like this:  
17  
18 ```{r cars}  
19 summary(cars)  
20 ```  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r pressure, echo=FALSE}  
27 plot(pressure)  
28 ```  
29  
30 Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the  
R code that generated the plot.  
31
```

At the bottom of the editor, the status bar shows '2:1 T32-Rmarkdown-Practice' and 'R Markdown'.

Theory and basics

[>.....] 0%

Markdown and Quarto documents



A screenshot of the RStudio interface showing an R Markdown document. The document contains code and text, with some sections highlighted in yellow. The top bar shows icons for file operations, search, and knit. The main area displays the following content:

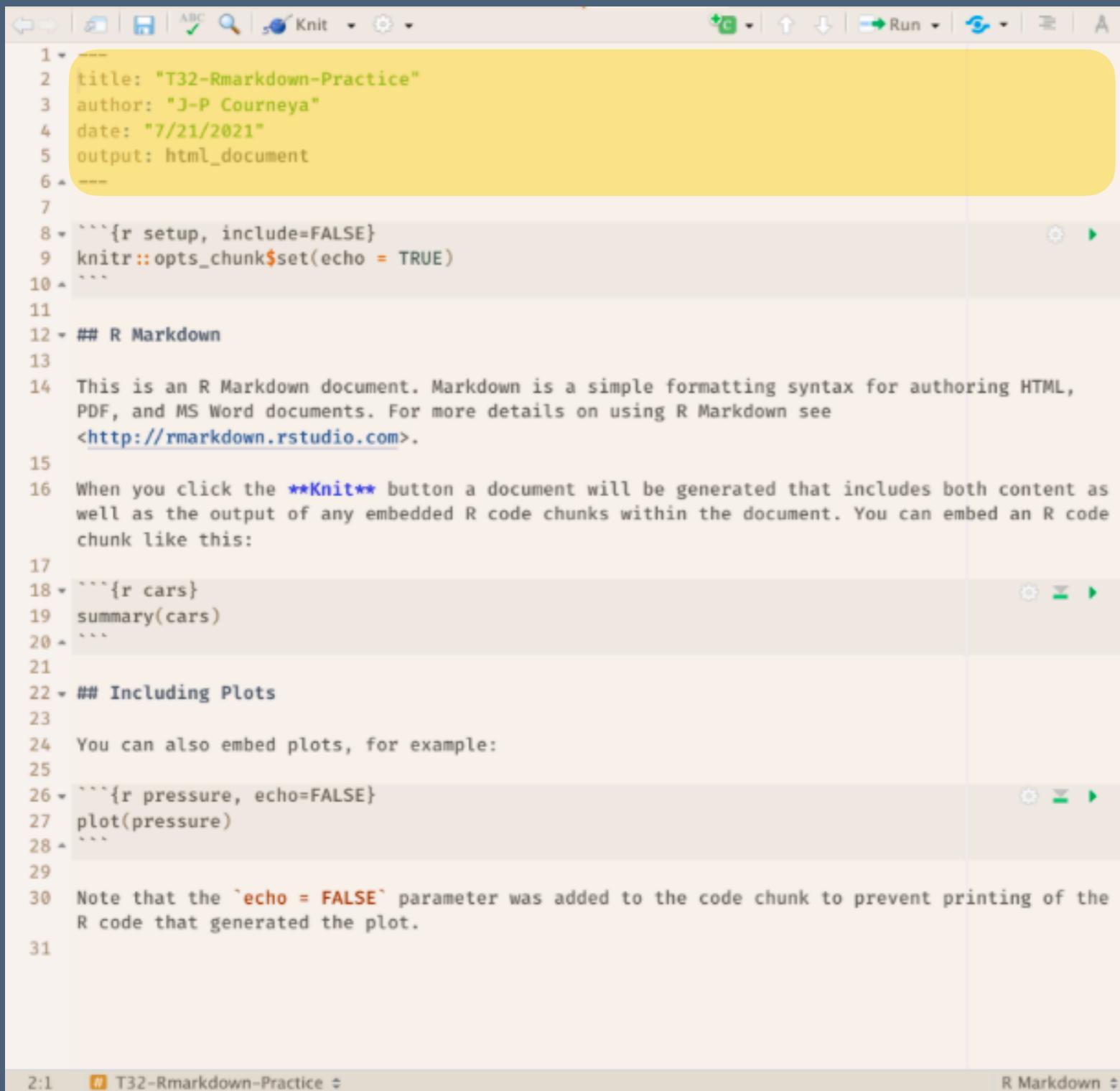
```
1 ---  
2 title: "T32-Rmarkdown-Practice"  
3 author: "J-P Courneya"  
4 date: "7/21/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML,  
PDF, and MS Word documents. For more details on using R Markdown see  
http://rmarkdown.rstudio.com.  
15  
16 When you click the Knit button a document will be generated that includes both content as  
well as the output of any embedded R code chunks within the document. You can embed an R code  
chunk like this:  
17  
18 ```{r cars}  
19 summary(cars)  
20 ```  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r pressure, echo=FALSE}  
27 plot(pressure)  
28 ```  
29  
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the  
R code that generated the plot.  
31
```

The status bar at the bottom left shows "2:1 T32-Rmarkdown-Practice" and the bottom right shows "R Markdown".

Theory and basics

[>.....] 0%

Markdown and Quarto documents



A screenshot of the RStudio interface showing an R Markdown document. The code editor displays the following content:

```
1 ---  
2 title: "T32-Rmarkdown-Practice"  
3 author: "J-P Courneya"  
4 date: "7/21/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML,  
PDF, and MS Word documents. For more details on using R Markdown see  
http://rmarkdown.rstudio.com.  
15  
16 When you click the Knit button a document will be generated that includes both content as  
well as the output of any embedded R code chunks within the document. You can embed an R code  
chunk like this:  
17  
18 ```{r cars}  
19 summary(cars)  
20```  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r pressure, echo=FALSE}  
27 plot(pressure)  
28```  
29  
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the  
R code that generated the plot.  
31
```

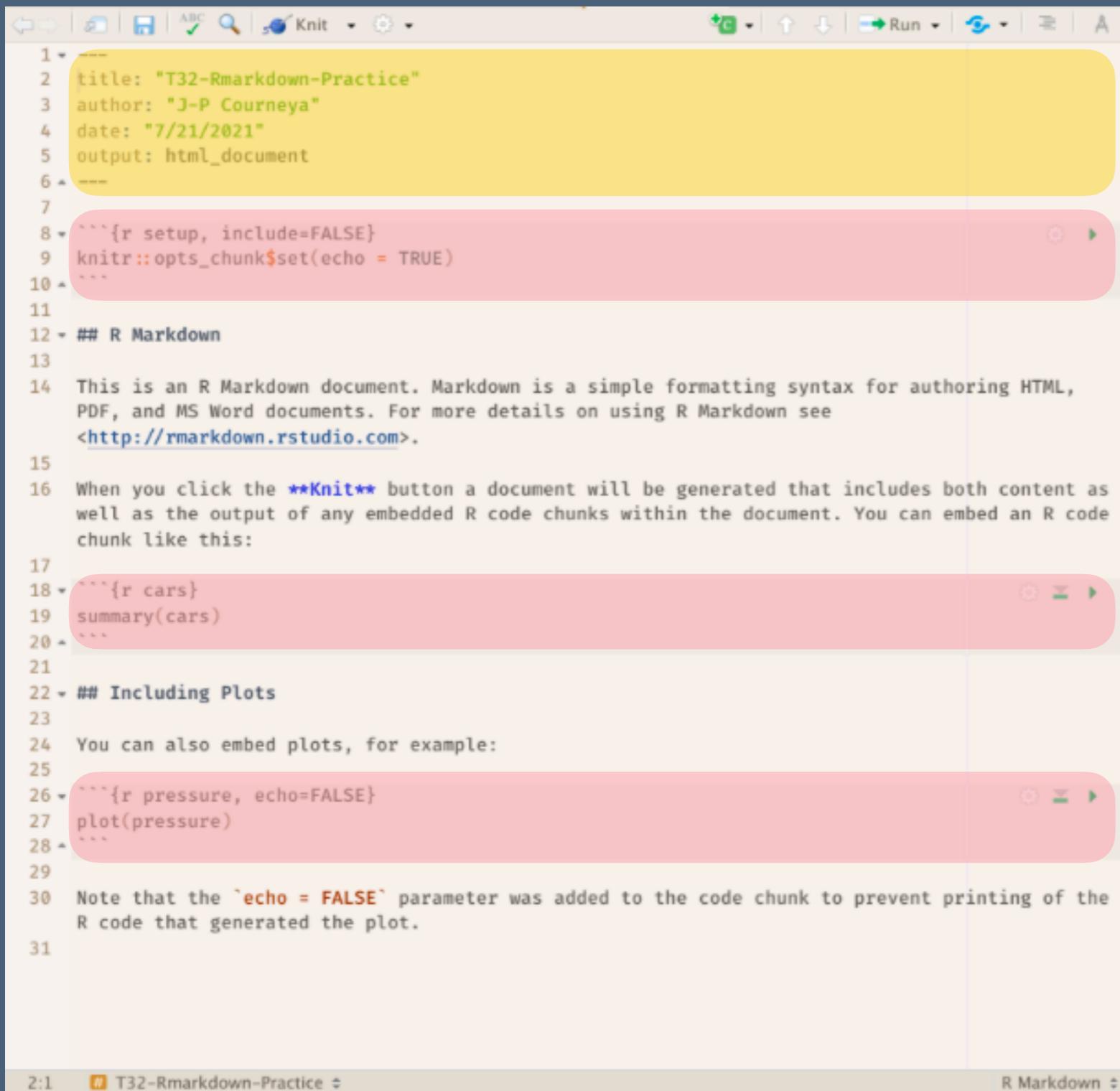
The YAML header (lines 1-6) is highlighted with a yellow background. The R code (lines 8-28) and the note at the bottom (line 30) are highlighted with a light gray background.

A YAML header

Theory and basics

[>.....] 0%

Markdown and Quarto documents



A screenshot of the RStudio interface showing an R Markdown document. The document contains the following content:

```
1 ---  
2 title: "T32-Rmarkdown-Practice"  
3 author: "J-P Courneya"  
4 date: "7/21/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10 ```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML,  
PDF, and MS Word documents. For more details on using R Markdown see  
http://rmarkdown.rstudio.com.  
15  
16 When you click the Knit button a document will be generated that includes both content as  
well as the output of any embedded R code chunks within the document. You can embed an R code  
chunk like this:  
17  
18 ```{r cars}  
19 summary(cars)  
20 ```  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r pressure, echo=FALSE}  
27 plot(pressure)  
28 ```  
29  
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the  
R code that generated the plot.  
31
```

The code is color-coded: green for YAML, blue for R code, and black for text. The RStudio toolbar at the top includes icons for back, forward, search, and knit.

A YAML header

Some chunk sections

Theory and basics

[>.....] 0%

Markdown and Quarto documents

A screenshot of a Quarto document editor interface. The document content is as follows:

```
1 ---  
2 title: "T32-Rmarkdown-Practice"  
3 author: "J-P Courneya"  
4 date: "7/21/2021"  
5 output: html_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10```  
11  
12 ## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:  
17  
18 ```{r cars}  
19 summary(cars)  
20```  
21  
22 ## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26 ```{r pressure, echo=FALSE}  
27 plot(pressure)  
28```  
29  
30 Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.  
31
```

The document is titled "T32-Rmarkdown-Practice" and has an output type of "html_document". It contains a YAML header with fields for title, author, date, and output. There are several code chunks, some of which are run in R (e.g., `summary(cars)`). The document also includes text sections and a note about including plots.

A YAML header

Some chunk sections

Some text sections

Theory and basics

[>.....] 0%

Markdown and Quarto documents



For the purpose of GitHub, we won't focus on chunk sections, but there is plenty of good literature about it



Some general code

If you want italics: _text_

If you want bold: ****text****

If you want a hyperlink text: [text](url)

Note that you can use emojis to be more friendly.

if you want , you can use codes such as :file_folder:

Theory and basics

[>.....] 0%

Markdown and Quarto documents

R Markdown :: CHEAT SHEET

What is R Markdown?

.Rmd files - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.

Reproducible Research - At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

Dynamic Documents - You can choose to export the finished report in a variety of formats, including html, pdf, MS Word, or RTF documents; html or pdf based slides, Notebooks, and more.

Workflow



- Open a new .Rmd file at File ▶ New File ▶ R Markdown. Use the wizard that opens to pre-populate the file with a template
- Write document by editing template
- Knit document to create report; use knit button or render() to knit
- Preview Output in IDE window
- Publish (optional) to web server
- Examine build log in R Markdown console
- Use output file that is saved along side .Rmd

This section shows the RStudio interface for creating and rendering an R Markdown document. It includes a screenshot of the RStudio menu bar, the R Markdown file editor with code and chunk controls, the R Markdown preview pane, and the RStudio file browser showing the generated HTML file.

render

Use rmarkdown::render() to render/knit at cmd line. Important args:

Input	- file to render	output_options	- List of render options (as in YAML)	output_file	params	- list of params to use	envir	- environment to evaluate code chunks in	encoding	- of input file

Embed code with knitr syntax

INLINE CODE
Insert with `r <code>`. Results appear as text without code.
Built with `r getRVersion()` → Built with 3.2.3

CODE CHUNKS
One or more lines surrounded with `{{r}}` and `{{ }}`. Place chunk options within curly braces, after r. Insert with {{r}}
`{{r echo=TRUE}}` → `{{r echo=TRUE}}`
`{{r getRVersion()}}` → `{{r getRVersion()}}

GLOBAL OPTIONS
Set with knitr::opts_chunk\$set(), e.g.

`{{r include=FALSE}}`
`knitr::opts_chunk\$set(echo = TRUE)`

IMPORTANT CHUNK OPTIONS

cache - cache results for future knits (default = FALSE)
cache.path - directory to save cached results in (default = "cache/")
child - file(s) to knit and then include (default = NULL)
collapse - collapse all output into single block (default = FALSE)
comment - prefix for each line of results (default = "#")
eval - Run code in chunk (default = TRUE)

dependson - chunk dependencies for caching (default = NULL)
echo - Display code in output document (default = TRUE)
engine - code language used in chunk (default = 'R')
error - Display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = FALSE)
include - Include chunk in doc after running (default = TRUE)
fig.align - 'left', 'right', or 'center' (default = 'default')
fig.cap - figure caption as character string (default = NULL)
fig.height, fig.width - Dimensions of plots in inches
highlight - highlight source code (default = TRUE)
out.extra, **out.height**, **out.width**, **prompt**, **purl**, **ref.label**, **render**, **size**, **split**, **tidy.opts**

Options not listed above: R.options, aniopts, autodep, background, cache.comments, cache.lazy, cache.rebuild, cache.vars, dev, dev.args, dpi, engine.opts, engine.path, fig.asp, fig.env, fig.ext, fig.keep, fig.lp, fig.path, fig.pos, fig.process, fig.retina, fig.scap, fig.show, fig.showtext, fig.subcap, interval, out.extra, out.height, out.width, prompt, purl, ref.label, render, size, split, tidy.opts

.rmd Structure

YAML Header
Optional section of render (e.g. pandoc) options written as key:value pairs (YAML).

At start of file

Between lines of ---

Text
Narration formatted with markdown, mixed with:

Code Chunks
Chunks of embedded code. Each chunk:

Begins with `{{r}}`
ends with `{{ }}

R Markdown will run the code and append the results to the doc. It will use the location of the .Rmd file as the **working directory**

Parameters

Parameterize your documents to reuse with different inputs (e.g., data, values, etc.)

1. Add parameters - Create and set parameters in the header as sub-values of params



2. Call parameters - Call parameter values in code as params\$<name>

3. Set parameters - Set values with Knit with parameters or the params argument of render():

render("doc.Rmd", params = list(n = 1, d = as.Date("2015-01-01")))

This section shows the RStudio interface for defining parameters in the YAML header of an R Markdown file. It includes a screenshot of the RStudio menu bar, the R Markdown file editor with parameter definitions, and the RStudio file browser showing the generated HTML file.

Interactive Documents

Turn your report into an Interactive Shiny document in 4 steps

- Add runtime: shiny to the YAML header.
- Call Shiny input functions to embed input objects.
- Call Shiny render functions to embed reactive output.
- Render with rmarkdown::run or click Run Document in RStudio IDE

This section shows the RStudio interface for creating an interactive Shiny application within an R Markdown document. It includes a screenshot of the RStudio menu bar, the R Markdown file editor with Shiny code, and the RStudio file browser showing the generated HTML file.

Embed a complete app into your document with shiny::shinyAppDir()
NOTE: Your report will be rendered as a Shiny app, which means you must choose an html output format, like html_document, and serve it with an active R Session.



Theory and basics

[>.....] 0%

Markdown and Quarto documents

R Markdown :: CHEAT SHEET

What is R Markdown?

.Rmd files - An R Markdown (.Rmd) file is a record of your research. It contains the code that a scientist needs to reproduce your work along with the narration that a reader needs to understand your work.

Reproducible Research - At the click of a button, or the type of a command, you can rerun the code in an R Markdown file to reproduce your work and export the results as a finished report.

Dynamic Documents - You can choose to export the finished report in a variety of formats, including html, pdf, MS Word, or RTF documents; html or pdf based slides, Notebooks, and more.

Workflow



- Open a new .Rmd file at File ▶ New File ▶ R Markdown. Use the wizard that opens to pre-populate the file with a template
- Write document by editing template
- Knit document to create report; use knit button or render() to knit
- Preview Output in IDE window
- Publish (optional) to web server
- Examine build log in R Markdown console
- Use output file that is saved along side .Rmd

Embed code with knitr syntax

INLINE CODE
Insert with `r <code>`. Results appear as text without code.
Built with `r getRVersion()` → Built with 3.2.3

CODE CHUNKS
One or more lines surrounded with `{{r}}` and `{{ }}`. Place chunk options within curly braces, after r. Insert with {{r echo=TRUE} getRVersion()} {{ }}

GLOBAL OPTIONS
Set with knitr::opts_chunk\$set(), e.g.

```{r include=FALSE}  
knitr::opts\_chunk\$set(echo = TRUE)  
```

IMPORTANT CHUNK OPTIONS

cache - cache results for future knits (default = FALSE)
cache.path - directory to save cached results in (default = "cache/")
child - file(s) to knit and then include (default = NULL)
collapse - collapse all output into single block (default = FALSE)
comment - prefix for each line of results (default = "#")
eval - Run code in chunk (default = TRUE)

dependson - chunk dependencies for caching (default = NULL)
echo - Display code in output document (default = TRUE)
engine - code language used in chunk (default = 'R')
error - Display error messages in doc (TRUE) or stop render when errors occur (FALSE) (default = FALSE)
Options not listed above: R.options, aniopts, autodep, background, cache.comments, cache.lazy, cache.rebuild, cache.vars, dev, dev.args, dpi, engine.opts, engine.path, fig.asp, fig.env, fig.ext, fig.keep, fig.lp, fig.path, fig.pos, fig.process, fig.retina, fig.scap, fig.show, fig.showtext, fig.subcap, interval, out.extra, out.height, out.width, prompt, purl, ref.label, render, size, split, tidy.opts

fig.align - 'left', 'right', or 'center' (default = 'default')
fig.cap - figure caption as character string (default = NULL)
fig.height, fig.width - Dimensions of plots in inches
highlight - highlight source code (default = TRUE)
include - Include chunk in doc after running (default = TRUE)
message - display code messages in document (default = TRUE)
results (default = 'markup')
'asis' - passthrough results
'hide' - do not display results
'hold' - put all results below all code
tidy - tidy code for display (default = FALSE)
warning - display code warnings in document (default = TRUE)

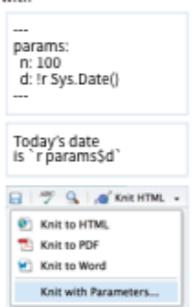
.rmd Structure

YAML Header
Optional section of render (e.g. pandoc) options written as key:value pairs (YAML).
At start of file
Between lines of ---
Text
Narration formatted with markdown, mixed with:
Code Chunks
Chunks of embedded code. Each chunk:
Begins with `{{r}}`
ends with `{{ }}`
R Markdown will run the code and append the results to the doc.
It will use the location of the .Rmd file as the **working directory**

Parameters

Parameterize your documents to reuse with different inputs (e.g., data, values, etc.)

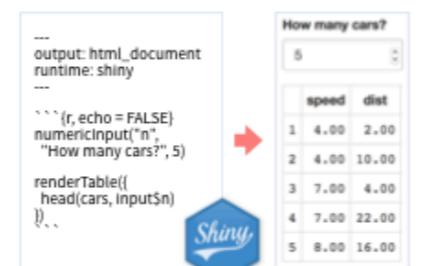
- Add parameters** - Create and set parameters in the header as sub-values of params
- Call parameters** - Call parameter values in code as params\$<name>
- Set parameters** - Set values with Knit with parameters or the params argument of render():
render("doc.Rmd", params = list(n = 1, d = as.Date("2015-01-01")))



Interactive Documents

Turn your report into an Interactive Shiny document in 4 steps

- Add runtime: shiny to the YAML header.
- Call Shiny input functions to embed input objects.
- Call Shiny render functions to embed reactive output.
- Render with rmarkdown::run or click Run Document in RStudio IDE



Embed a complete app into your document with shiny::shinyAppDir()

NOTE: Your report will be rendered as a Shiny app, which means you must choose an html output format, like **html_document**, and serve it with an active R Session.



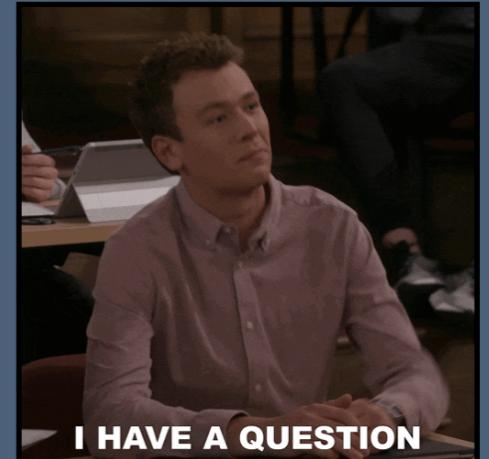
oh my

Theory and basics

[>.....] 0%

Markdown and Quarto documents

And so, why Markdown is important
for my understanding in GitHub?

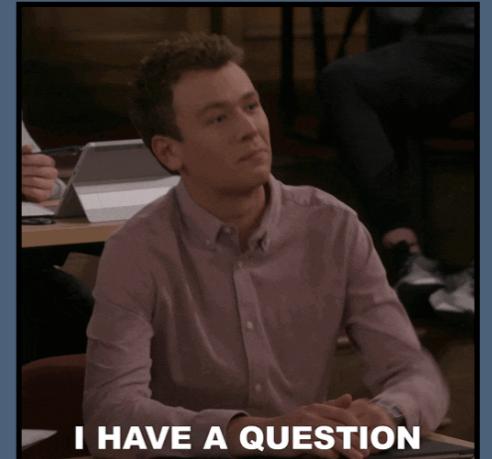


Theory and basics

[>.....] 0%

Markdown and Quarto documents

And so, why Markdown is important
for my understanding in GitHub?

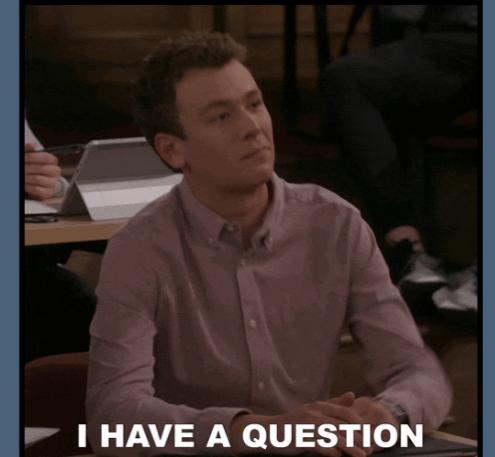


Theory and basics

[>.....] 0%

Markdown and Quarto documents

And so, why Markdown is important for my understanding in GitHub?



For each project you will need a README.md

A screenshot of a GitHub repository page for "GitHub_Tutorial". The repository is public and has 1 branch and 0 tags. The main branch is "main". There are 76 commits from "JayCrit" updating the README.md file. The repository has 13 stars, 1 watching, and 1 fork. The README file is linked and shows its history. The repository is described as presenting a global overview of GitHub and Git+R for reproducibility. It includes tags for "github", "git", "tutorial", "r", "reproducibility", and "good-practices". The footer mentions the GitHub tutorial by Jérémie Carlot, the Laboratoire d'Océanographie de Villefranche sur mer (LOV), and the IMEV. It also features icons for R, GitHub, and other projects.

GitHub Tutorial Public

Unpin Unwatch 1 Fork 1 Star 13

main 1 Branch 0 Tags Go to file Add file Code

JayCrit Update README.md 45277a7 · 5 months ago 76 Commits

Courses up-to-date 2 years ago

R bad_colors example git history 2 years ago

.gitignore change 2 years ago

README.md Update README.md 5 months ago

README

GitHub tutorial

This tutorial is presented by [Jérémie Carlot](#), as a result of his post-doctoral fellowship with the [Laboratoire d'Océanographie de Villefranche sur mer](#) at [IMEV](#) in France and supervised by [Nuria Teixido](#), [Steeve Comeau](#) and [Jean-Pierre Gattuso](#).

LOV R GitHub

About

This repository presents a global overview of why GitHub, and especially Git+R should be used by everyone, helping for reproducibility. This tutorial is provided with a class of ~ 1h30.

github git tutorial r
reproducibility good-practices

Readme Activity 13 stars 1 watching 1 fork

Releases No releases published Create a new release

Theory and basics

[=>.....] 15%

GitHub's fame

What should I found into my introductory README?

Theory and basics

[=>.....] 15%

GitHub's fame

What should I found into my introductory README?

- 1) A short introduction about this repository. It can be an abstract or it can be a short statement.

Theory and basics

[=>.....] 15%

GitHub's fame

What should I found into my introductory README?

- 1) A short introduction about this repository. It can be an abstract or it can be a short statement.
- 2) You will need to be organized with **ALWAYS** at least three folders



Data



R / Matlab / Python Scripts



Outputs

Theory and basics

[=>.....] 15%

GitHub's fame

What should I found into my introductory README?

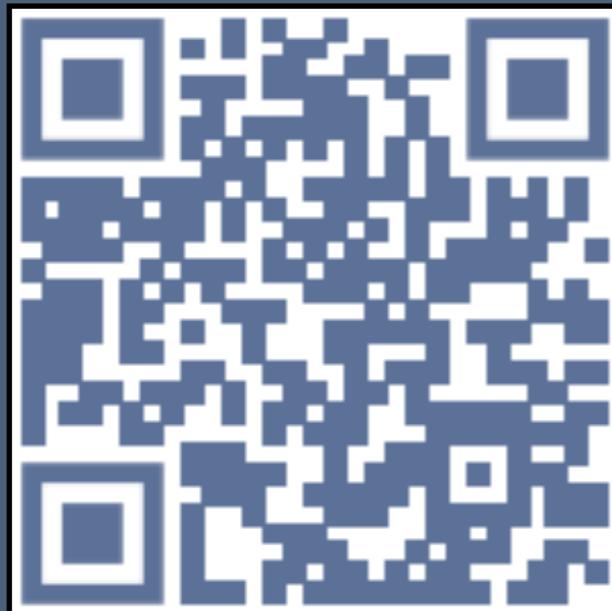
- 1) A short introduction about this repository. It can be an abstract or it can be a short statement.
- 2) You will need to be organized with **ALWAYS** at least three folders
 -  Data
 -  R / Matlab / Python Scripts
 -  Outputs
- 3) Finally you will need to state which packages you used. In R, you can run *sessioninfo::session_info()* or *base::session_info()*

Theory and basics

[=>.....] 15%

GitHub's fame

Some examples



CCA study



MME study



GCRMN

Theory and basics

[=>.....] 15%

GitHub's fame

OK, you can store documents as Google Drive. Why is it better?



Theory and basics

[=>.....] 15%

GitHub's fame

OK, you can store documents as Google Drive. Why is it better?



Theory and basics

[==>.....] 15%

GitHub's fame

OK, you can store documents as Google Drive. Why is it better?



1) you can pull an entire project

Once your project is on GitHub, it becomes reproducible—meaning that ANYONE can read and run your files on their own machine.



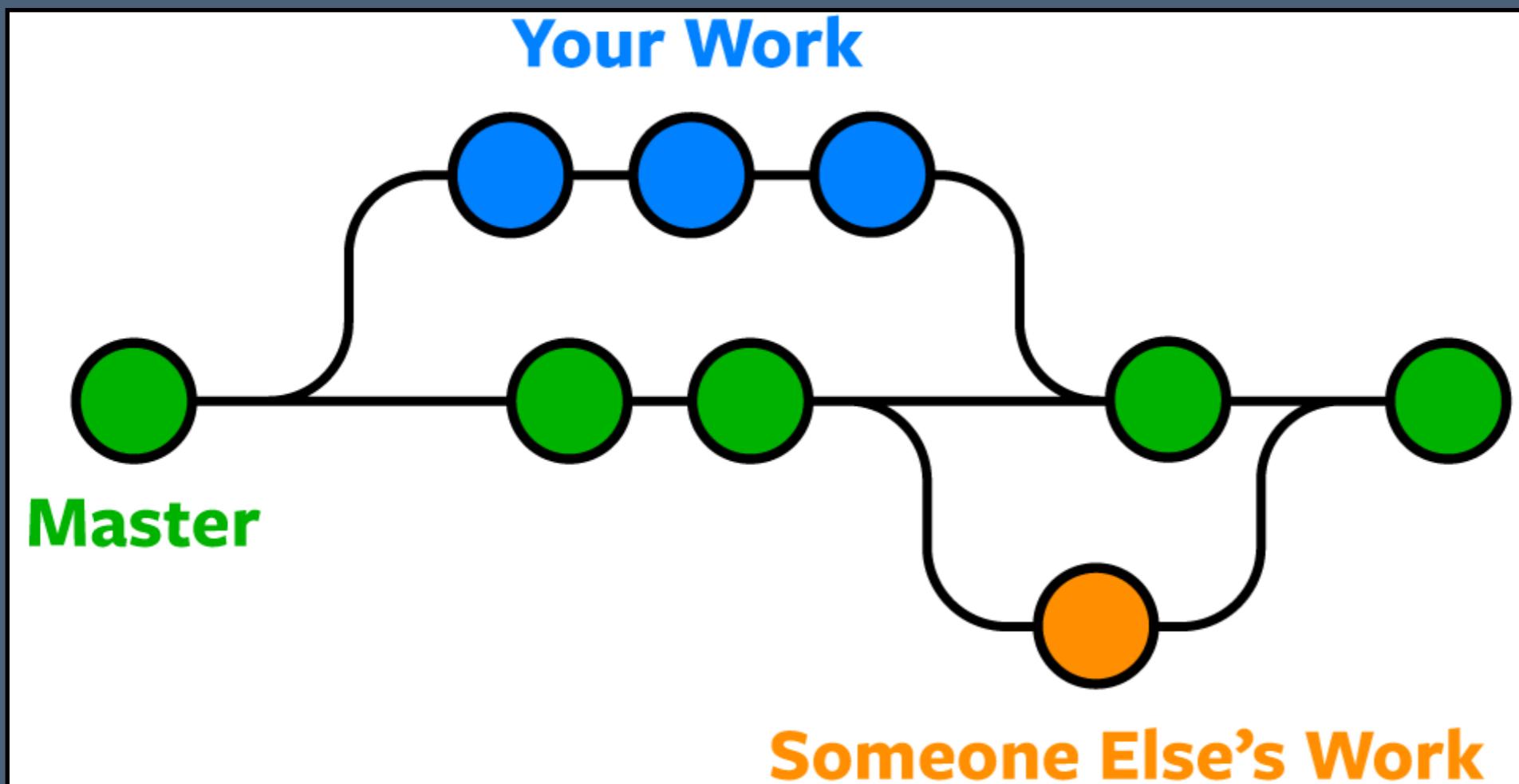
➡ See practice

Theory and basics

[==>.....] 15%

GitHub's fame

2) If you work in group, you can edit scripts in the meantime!

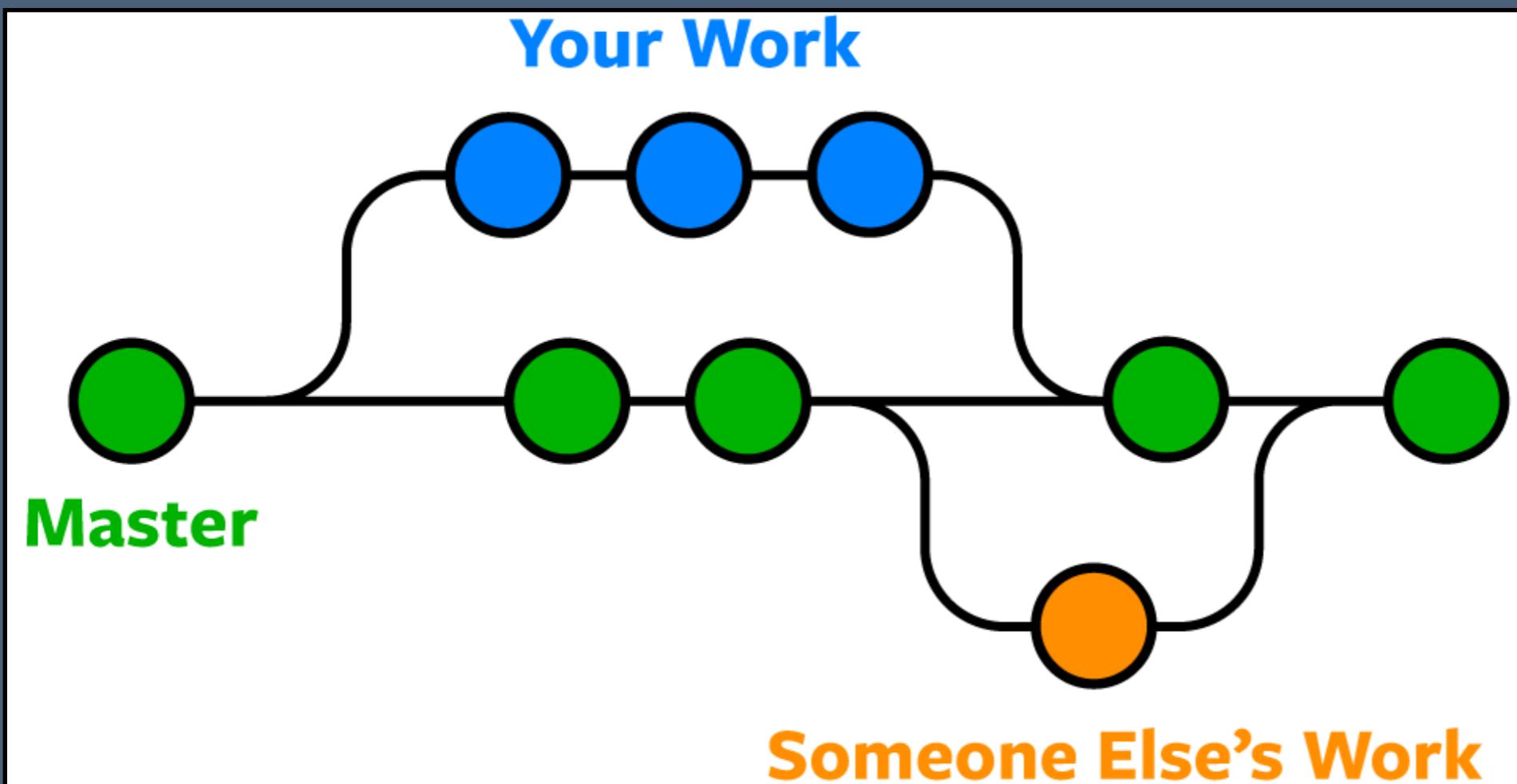


Theory and basics

[==>.....] 15%

GitHub's fame

2) If you work in group, you can edit scripts in the meantime!



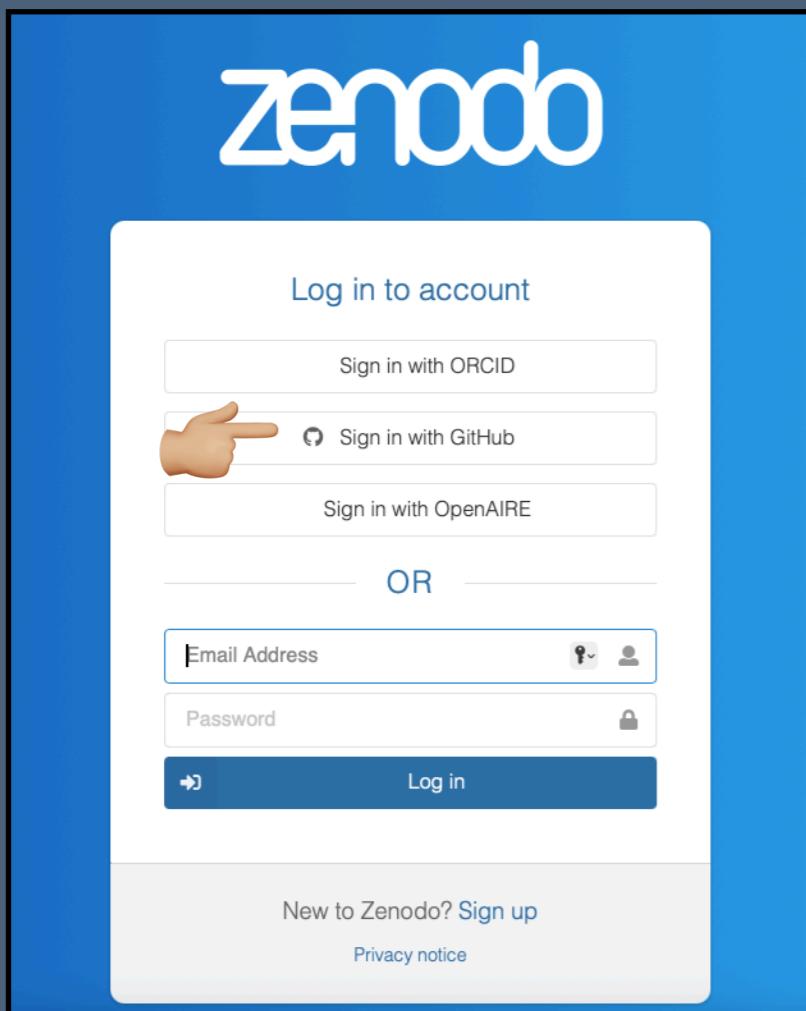
➡ See practice

Theory and basics

[=>.....] 15%

GitHub's fame

3) you can get a DOI which can help to get visibility for others and yourself!



Linked to Zenodo. You can share codes, data or both

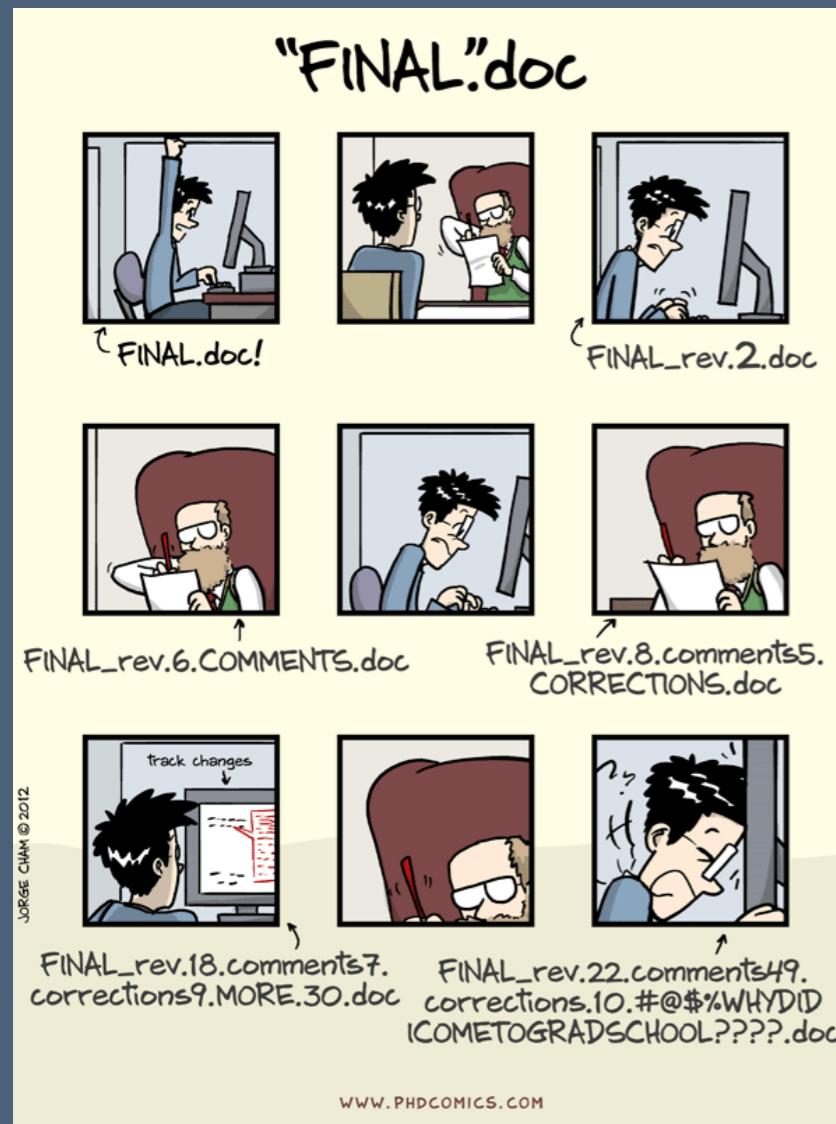
A screenshot of a Zenodo data record page. The title is 'JayCrIlt/MMEs_Mortality: Vulnerability of benthic trait diversity across the Mediterranean Sea following Mass Mortality Events – Data up-to-date'. It was published on February 3, 2025, and is version v1.1. The record is categorized as 'Software' and is 'Open'. On the right, there are statistics: 134 views and 37 downloads. Below these are sections for 'Versions' (listing v1.2, v1.1, and v1.0), 'Cite all versions?' (with a note about the DOI), and 'View all 3 versions'. The central part of the page lists numerous researchers and their affiliations, each with a small green circular icon. At the bottom, it says 'Data and code of Carlot et al. (2024) Vulnerability of benthic trait diversity across the Mediterranean Sea following Mass Mortality Events – LATEST DATA –'.

Theory and basics

[=>.....] 15%

GitHub's fame

4) You get a version control and an history track of changes!



Everything is stored and available whenever you need it and you keep only one document with a unique name

Theory and basics

[=>.....] 15%

GitHub's fame

Why GitHub it better than Google or others?

Theory and basics

[=>.....] 15%

GitHub's fame

Why GitHub it better than Google or others?

- 1) Easy download of everything

Theory and basics

[=>.....] 15%

GitHub's fame

Why GitHub it better than Google or others?

- 1) Easy download of everything
- 2) Easy teamwork efforts

Theory and basics

[=>.....] 15%

GitHub's fame

Why GitHub it better than Google or others?

- 1) Easy download of everything
- 2) Easy teamwork efforts
- 3) Publishable data and code

Theory and basics

[=>.....] 15%

GitHub's fame

Why GitHub it better than Google or others?

- 1) Easy download of everything
- 2) Easy teamwork efforts
- 3) Publishable data and code
- 4) Track of changes

Theory and basics

[==>.....] 15%

GitHub's fame

Why GitHub it better than Google or others?

- 1) Easy download of everything
- 2) Easy teamwork efforts
- 3) Publishable data and code
- 4) Track of changes
- 5) Reproducibility



Theory and basics

[=====>.....] 30%

How to use it?



You need one GitHub account (free)

Theory and basics

[=====>.....] 30%

How to use it?



You need one GitHub account (free)



You need one IDE (RStudio, VisualCode)

Theory and basics

[=====>.....] 30%

How to use it?



You need one GitHub account (free)



You need one IDE (RStudio, VisualCode)



You need to remember at least once
your account name and your password



Theory and basics

[=====>.....] 30%

How to use it?



First you'll need to link Github to your IDE via Git

1) Install Git



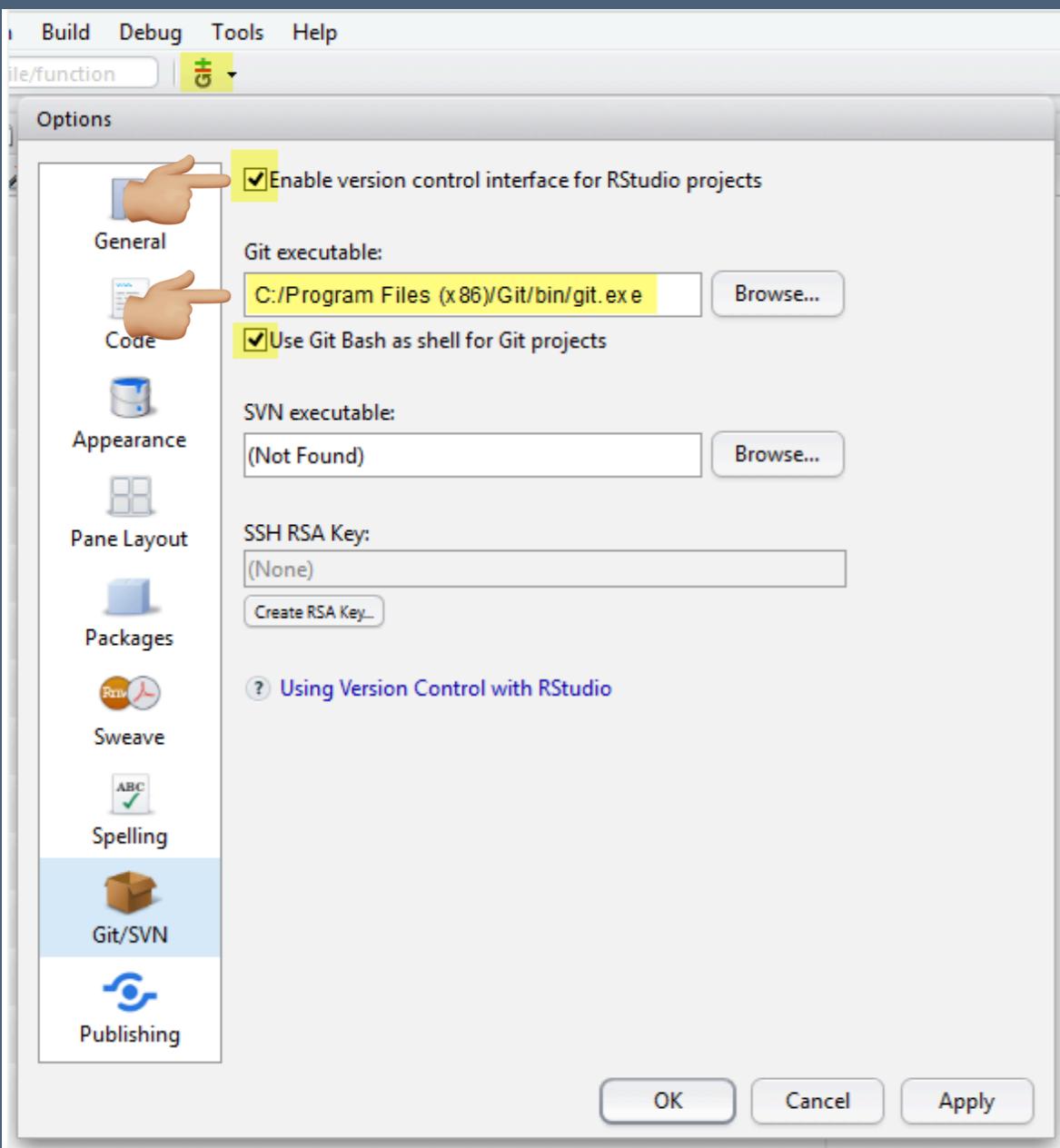
2) Setup Git in RStudio: Tell RStudio where to find the Git installation.
Tools > Global Options... click on Git/SVN

Theory and basics

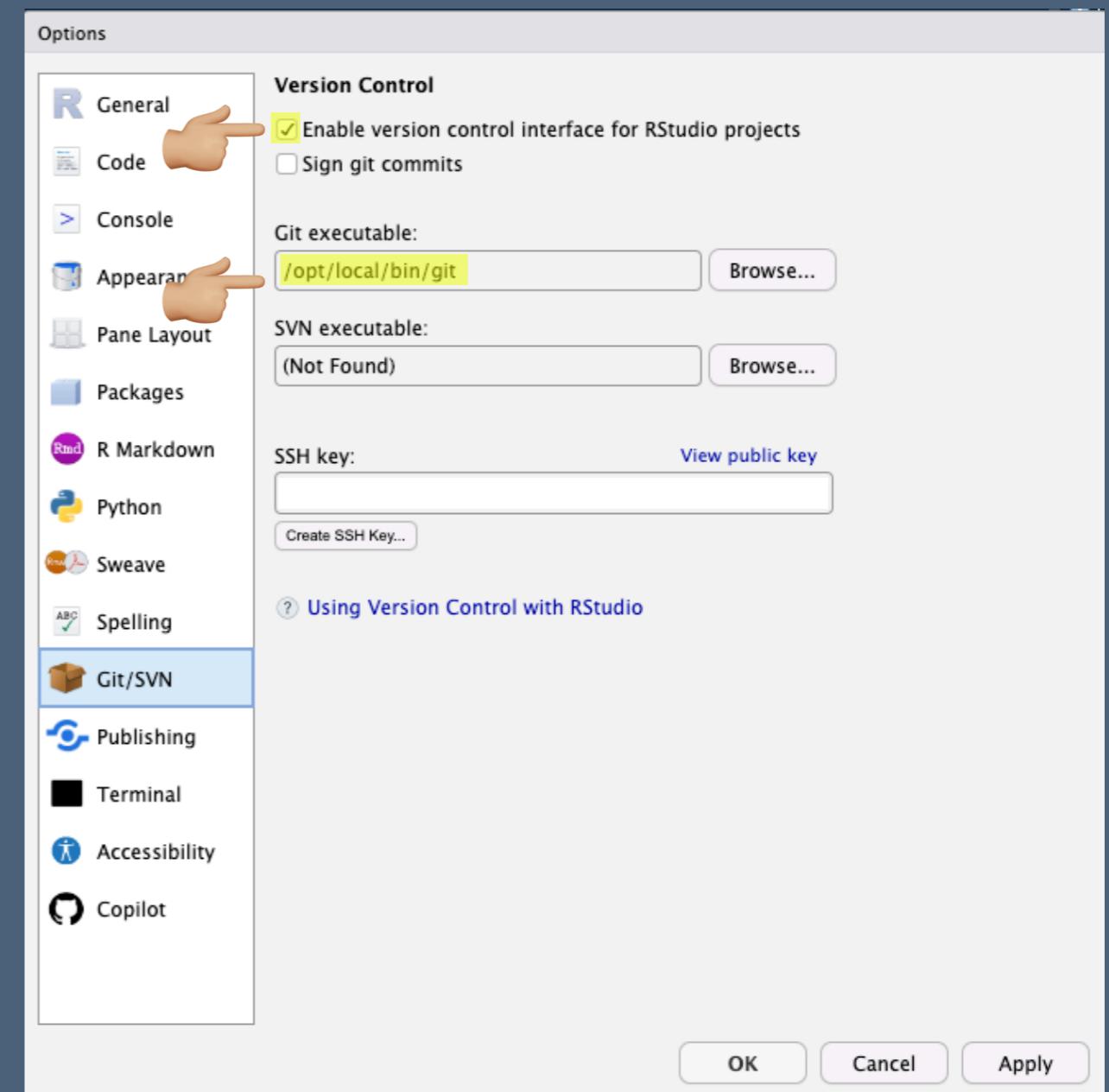
[=====>.....] 30%

How to use it?

Windows



Linux/OSX



Theory and basics

[=====>.....] 30%

How to use it?

- 3) Then you need to restart your IDE
- 4) And you'll need to set up Git

In the terminal, run:

```
git config --global user.name 'yourGitHubUsername'  
git config --global user.email 'name@provider.com'
```

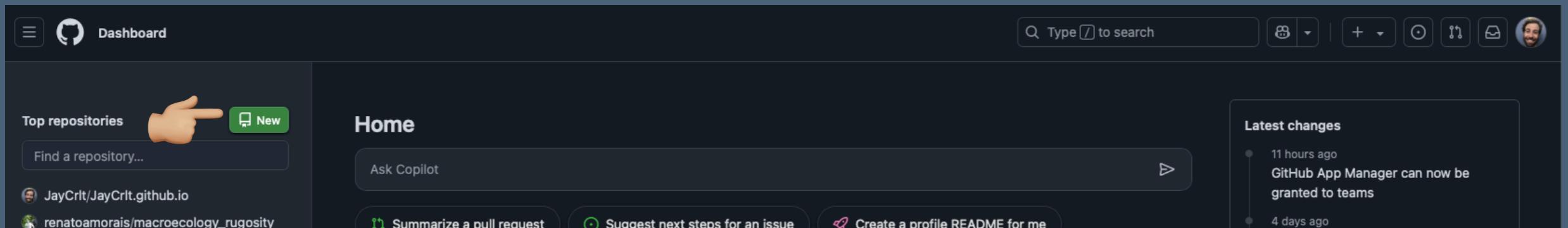
Theory and basics

[=====>.....] 30%

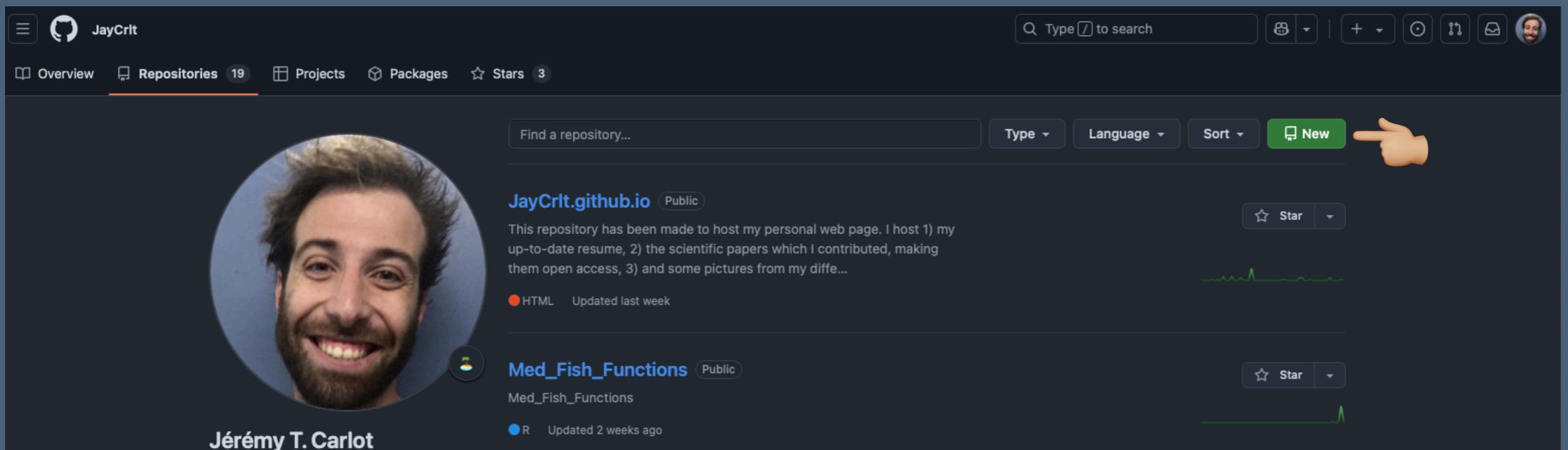
How to use it?



Now you can create a repository in GitHub.



The screenshot shows the GitHub dashboard. On the left, there's a sidebar with 'Top repositories' and a search bar. A hand cursor points at the green 'New' button in the top right corner of the sidebar. The main area is titled 'Home' and includes sections for 'Ask Copilot', 'Summarize a pull request', 'Suggest next steps for an issue', and 'Create a profile README for me'. To the right, there's a 'Latest changes' feed with two items: one from 11 hours ago about GitHub App Manager and another from 4 days ago. At the bottom, there's a user profile for 'JayCrlt' with 19 repositories, followed by a detailed view of the 'JayCrlt.github.io' repository, which is public and hosts a personal website.



This screenshot shows the GitHub repository page for 'JayCrlt.github.io'. It features a large circular profile picture of a smiling man with a beard. Below the profile picture is the name 'Jérémy T. Carlot'. The repository is described as public and used to host a personal website. It contains three files: an HTML file updated last week and two R files, one updated 2 weeks ago. The 'New' button in the top right corner of the repository header is highlighted with a hand cursor. The repository has 3 stars.

Theory and basics

[=====>.....] 30%

How to use it?

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 ***

 JayCrit /

Great repository names are short and memorable. Need inspiration? How about [crispy-octo-waffle](#) ?

Description (optional)



 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](#).

Add .gitignore


.gitignore template: [None](#)

Choose which files not to track from a list of templates. [Learn more about ignoring files](#).

Choose a license


License: [None](#)

A license tells others what they can and can't do with your code. [Learn more about licenses](#).

This will set  [main](#) as the default branch. Change the default name in your [settings](#).

 ⚠ You are creating a public repository in your personal account.

Create repository

You need a nickname (e.g., B-useful, Vadapesll...)

You can set either your repository in public or private

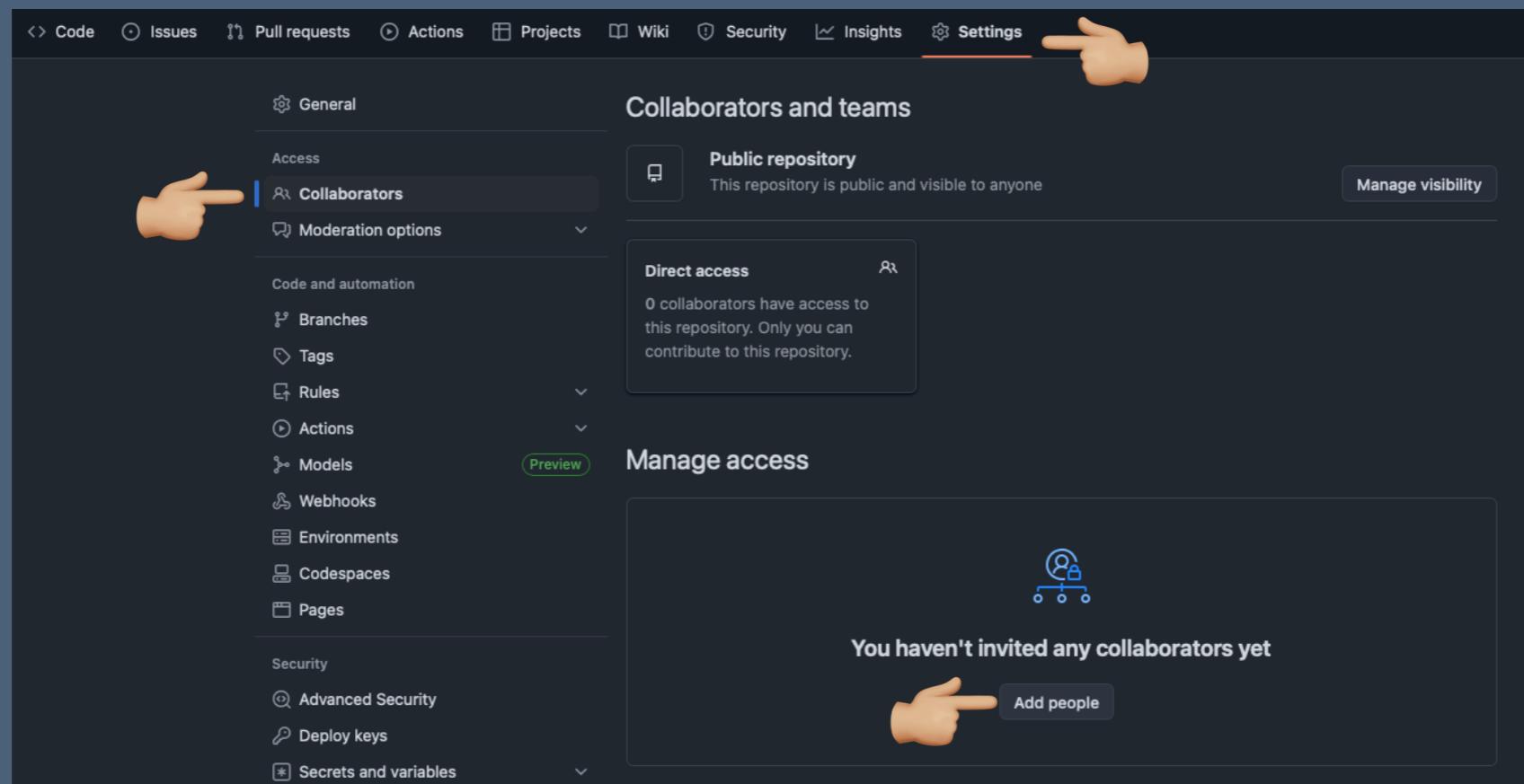
Add a README.md file 😊

Theory and basics

[=====>.....] 30%

How to use it?

At the beginning it's always good to set your repository as private. But it means that you are the only one who can access it. Unless you add collaborators



Theory and basics

[=====>.....] 30%

How to use it?

Now you have your repository, you want to link it to your project.

You MUST work with R Project.

The image consists of three side-by-side screenshots of the 'New Project Wizard' interface:

- Create Project**: Shows three options: 'New Directory' (Start a project in a brand new working directory), 'Existing Directory' (Associate a project with an existing working directory), and 'Version Control' (Checkout a project from a version control repository). The 'Version Control' option is highlighted.
- Create Project from Version Control**: Shows two options: 'Git' (Clone a project from a Git repository) and 'SVN' (Checkout a project from a Subversion repository). The 'Git' option is selected.
- Clone Git Repository**: A detailed view of the 'Git' selection. It includes fields for 'Repository URL:' (with a placeholder 'http://'), 'Project directory name:' (with a placeholder 'myproject'), and 'Create project as subdirectory of:' (with a placeholder '~/Desktop'). There is also a 'Browse...' button and a checkbox for 'Open in new session'.

Copy paste the URL of the repository

Theory and basics

[=====>.....] 30%

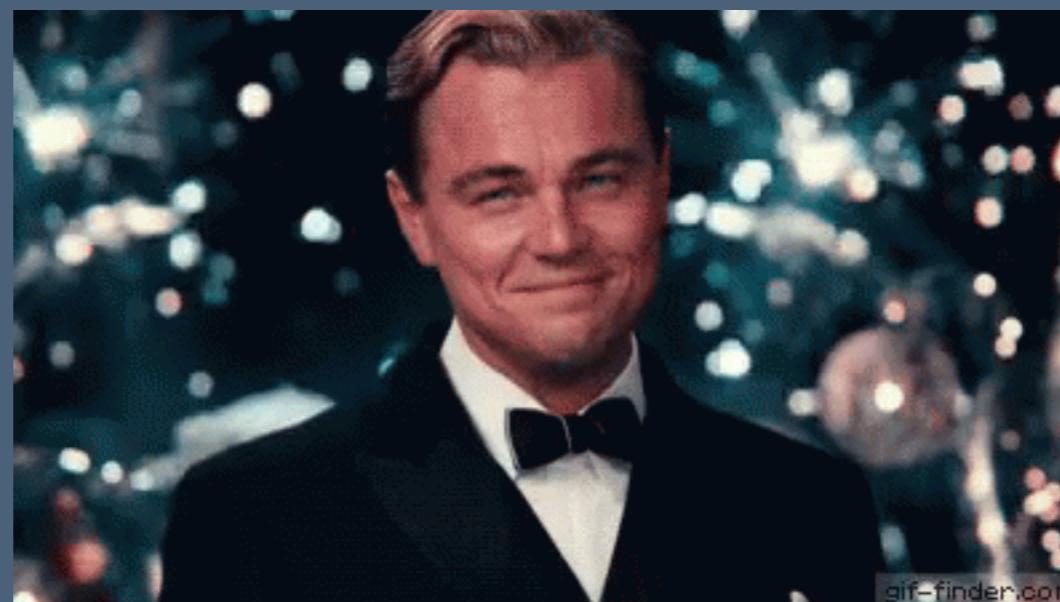
How to use it?

Remember!

You will need at least three folders

- ❑ Data
- ❑ R / Matlab / Python Scripts
- ❑ Outputs

So you can create these three folders in your R Project



Theory and basics

[=====>.....] 30%

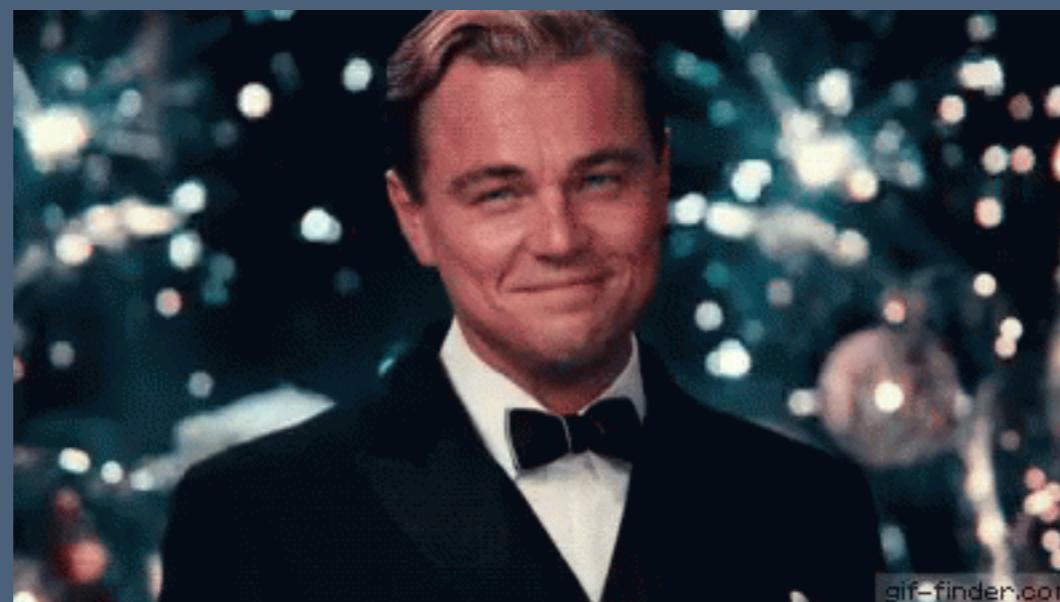
How to use it?

Remember!

You will need at least three folders

- ❑ Data
- ❑ R / Matlab / Python Scripts
- ❑ Outputs

So you can create these three folders in your R Project

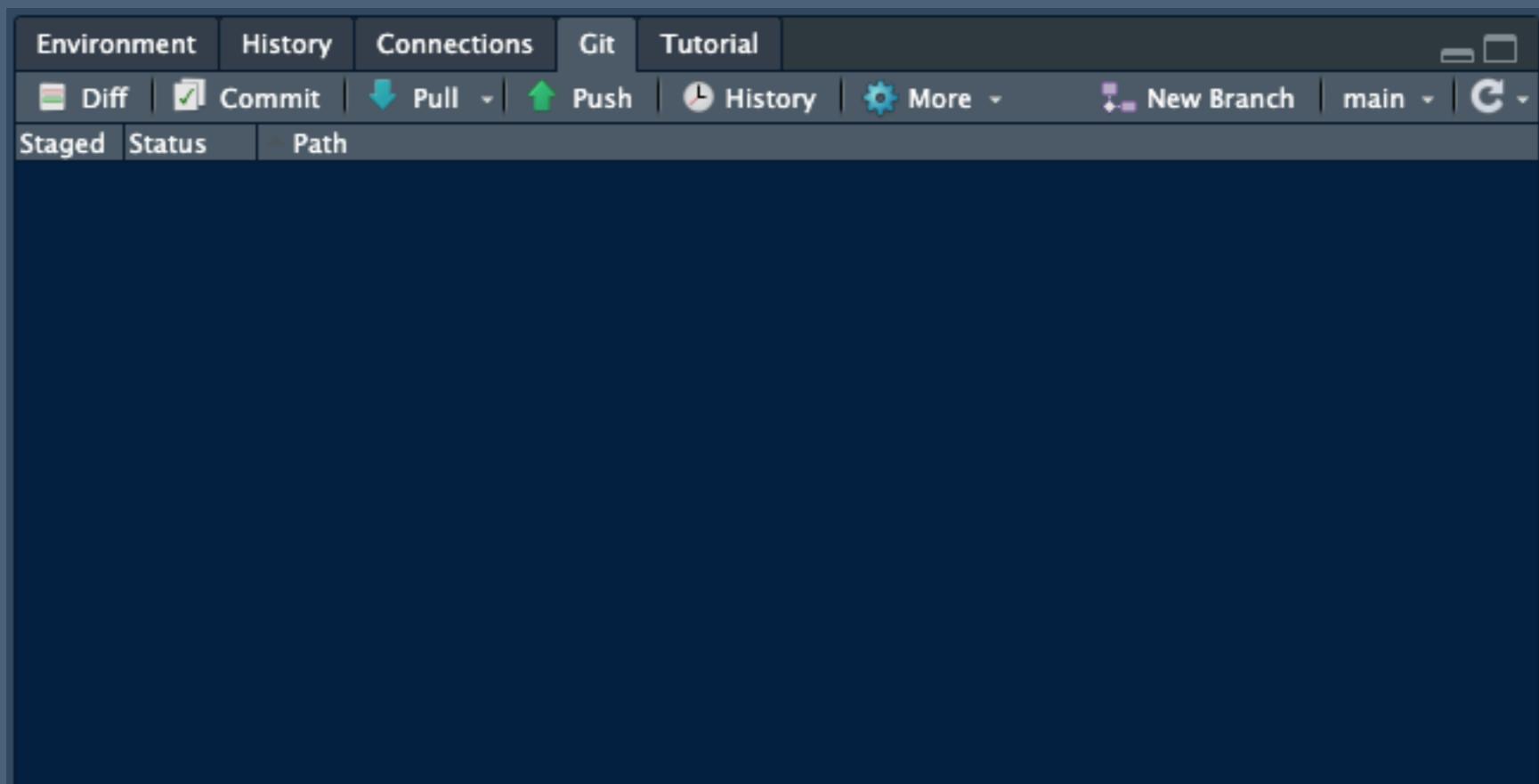


Theory and basics

[=====>.....] 30%

How to use it?

If everything has worked, you should have a new tag with git. And nothing is inside yet

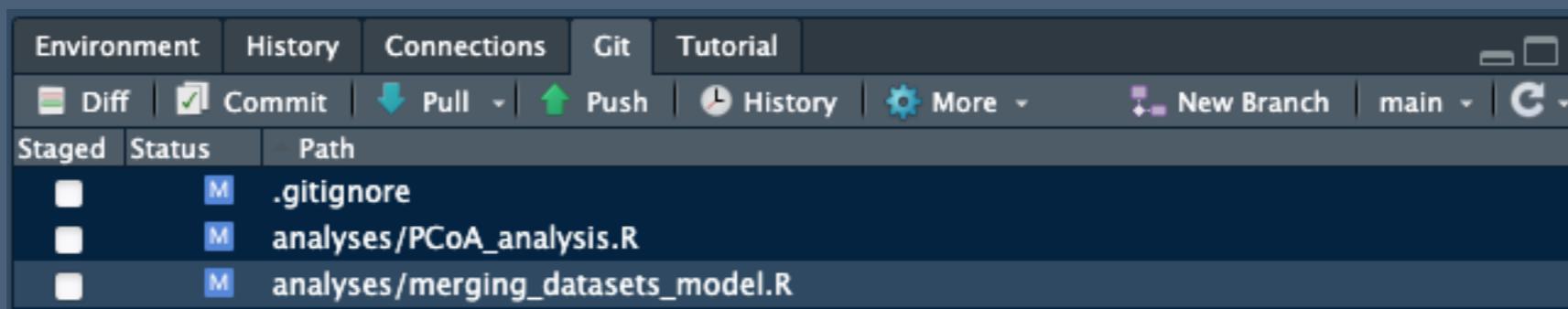


Theory and basics

[=====>.....] 30%

How to use it?

If you store documents (manually by dropping a picture in one of the three previous folders mentioned or automatically with save in your IDE), they will appear in the tab, ready to be uploaded

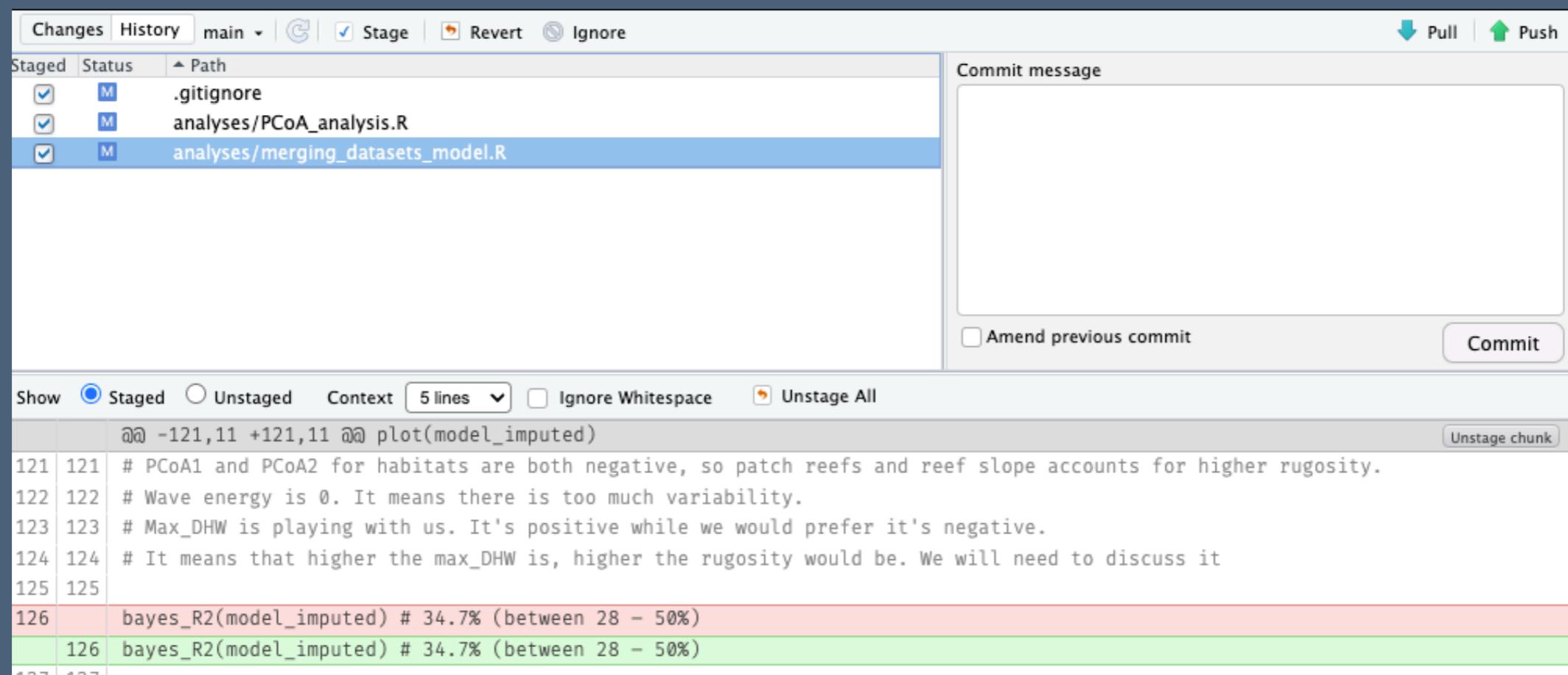


Theory and basics

[=====>.....] 30%

How to use it?

From now on, you'll have to get habits to commit and push/pull. Thick all the docs you have in your tab, and click on commit

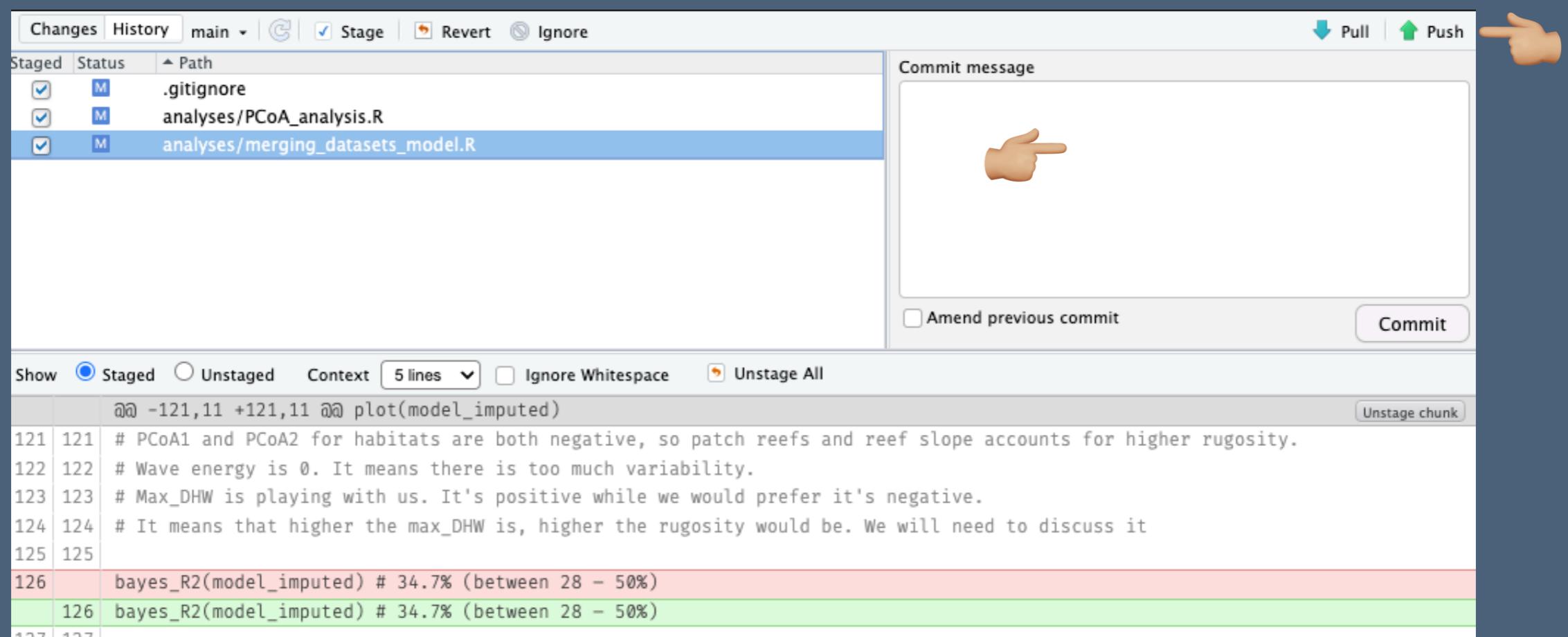


Theory and basics

[=====>.....] 30%

How to use it?

You must write a comment in the commit message. It is necessary for the tracking of changes and you can press PUSH



Theory and basics

[=====>.....] 30%

How to use it?

At this point, your IDE will ask you to fill the password. In reality, it will ask you to fill EVERY TIME the password. Which is somehow annoying.

What you can do then is to go to Settings in GitHub, scrolling down to Developer Settings from the right tab, then Personal Access Tokens and Tokens (Classic)

Theory and basics

[=====>.....] 30%

How to use it?

Edit personal access token (classic)

If you've lost or forgotten this token, you can regenerate it, but be aware that any scripts or applications using this token will need to be updated. [Regenerate token](#)

Note

Token_030322

What's this token for?

Expiration

This token has no expiration date. To set a new expiration date, you must [regenerate the token](#).

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

Save this token at least somewhere safe

Theory and basics

[=====>.....] 30%

How to use it?

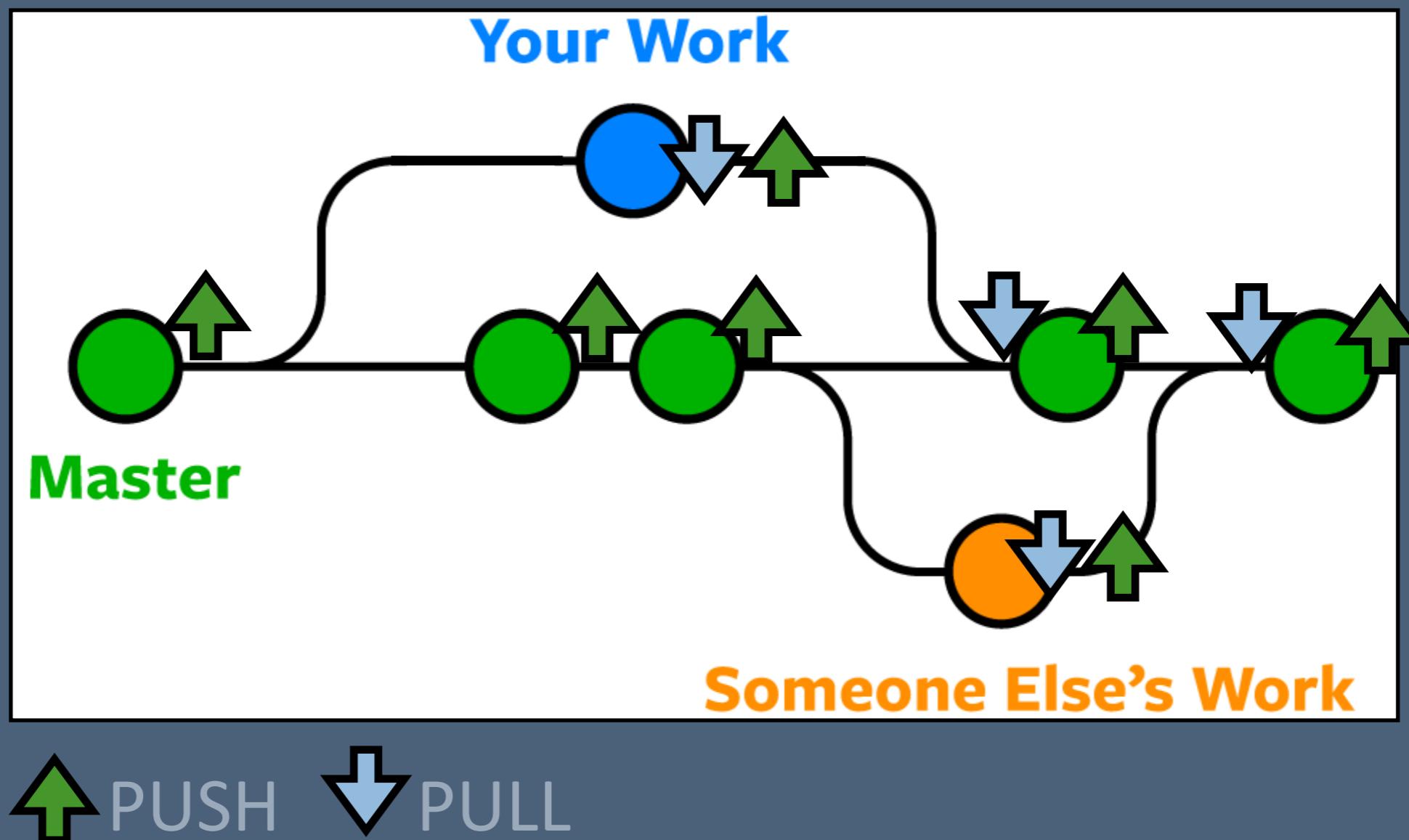
The pull option will be used if someone pushed from its machine to the repository. It allows you to sync what others did in your repository and it is mandatory. If you forget, you will not be able to push and you'll get an error just mentioning you need to pull before proceeding.

Theory and basics

[=====>.....] 30%

How to use it?

If three people work together for a project, it looks like this

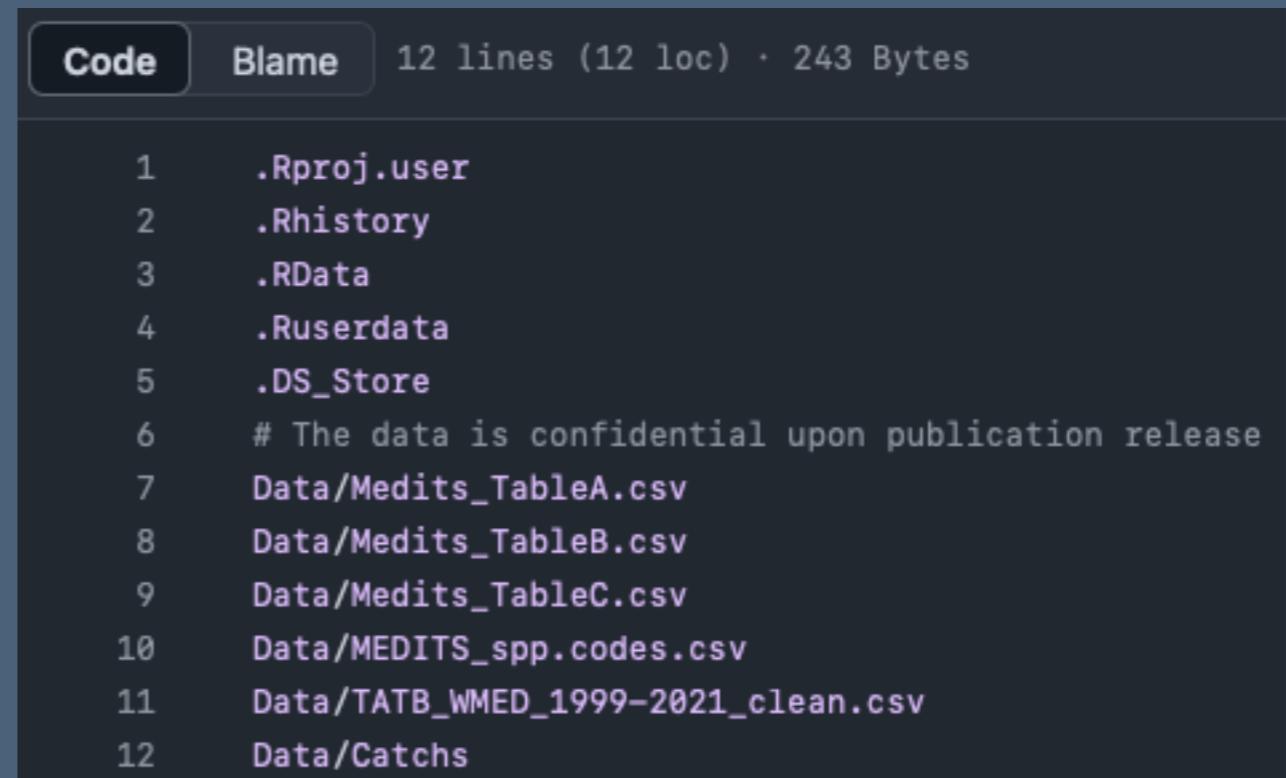


Theory and basics

[=====>.....] 30%

How to use it?

If for some reason your collaborators do not have GitHub but you want them to get access to your repository you can use `.gitignore` and set the repository as public but remove documents you do not want to share yet (*e.g.*, data)



A screenshot of a GitHub code editor interface. At the top, there are tabs for "Code" and "Blame", with "Code" being active. To the right of the tabs, it says "12 lines (12 loc) · 243 Bytes". The main area displays a list of 12 items, each preceded by a line number from 1 to 12. The items listed are:

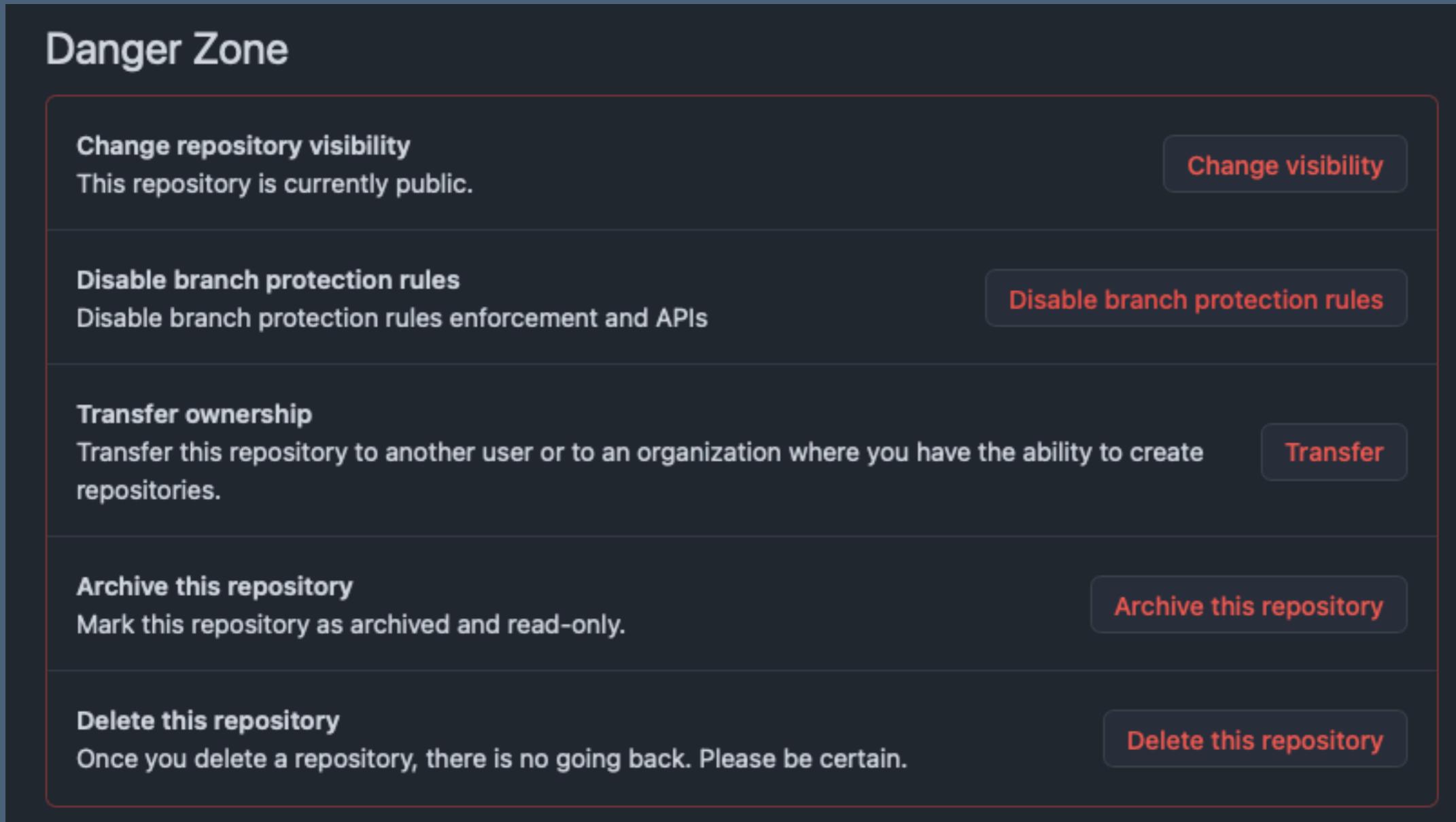
```
1 .Rproj.user
2 .Rhistory
3 .RData
4 .Ruserdata
5 .DS_Store
6 # The data is confidential upon publication release
7 Data/Meditis_TableA.csv
8 Data/Meditis_TableB.csv
9 Data/Meditis_TableC.csv
10 Data/MEDITIS_spp.codes.csv
11 Data/TATB_WMED_1999-2021_clean.csv
12 Data/Catchs
```

Theory and basics

[=====>.....] 30%

How to use it?

Finally, scrolling down the settings of your repository, you'll find the Danger Zone

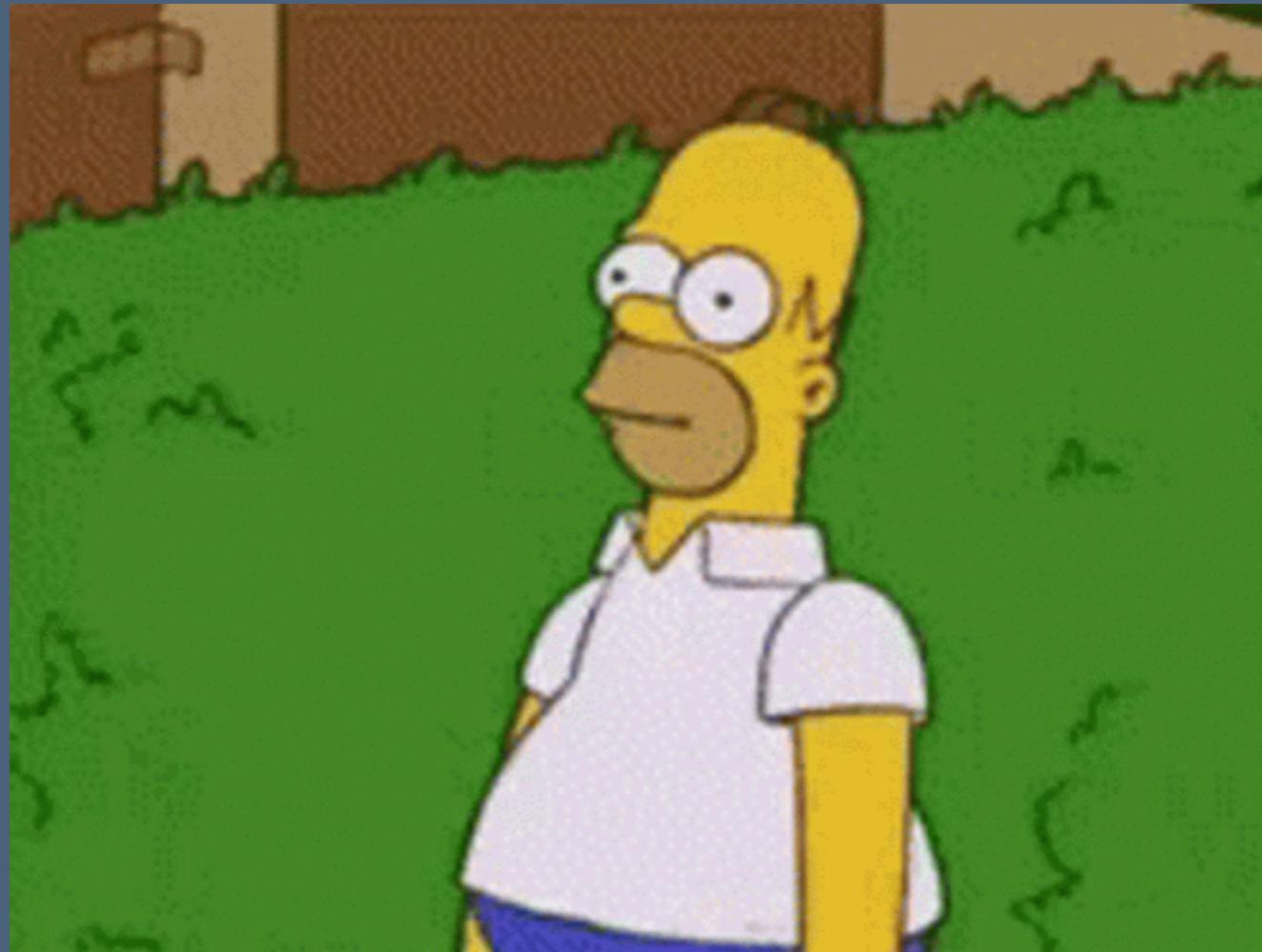


Practice

[=====>.....] 70%

Create a Project with GitHub

Now it's time for you to practice!

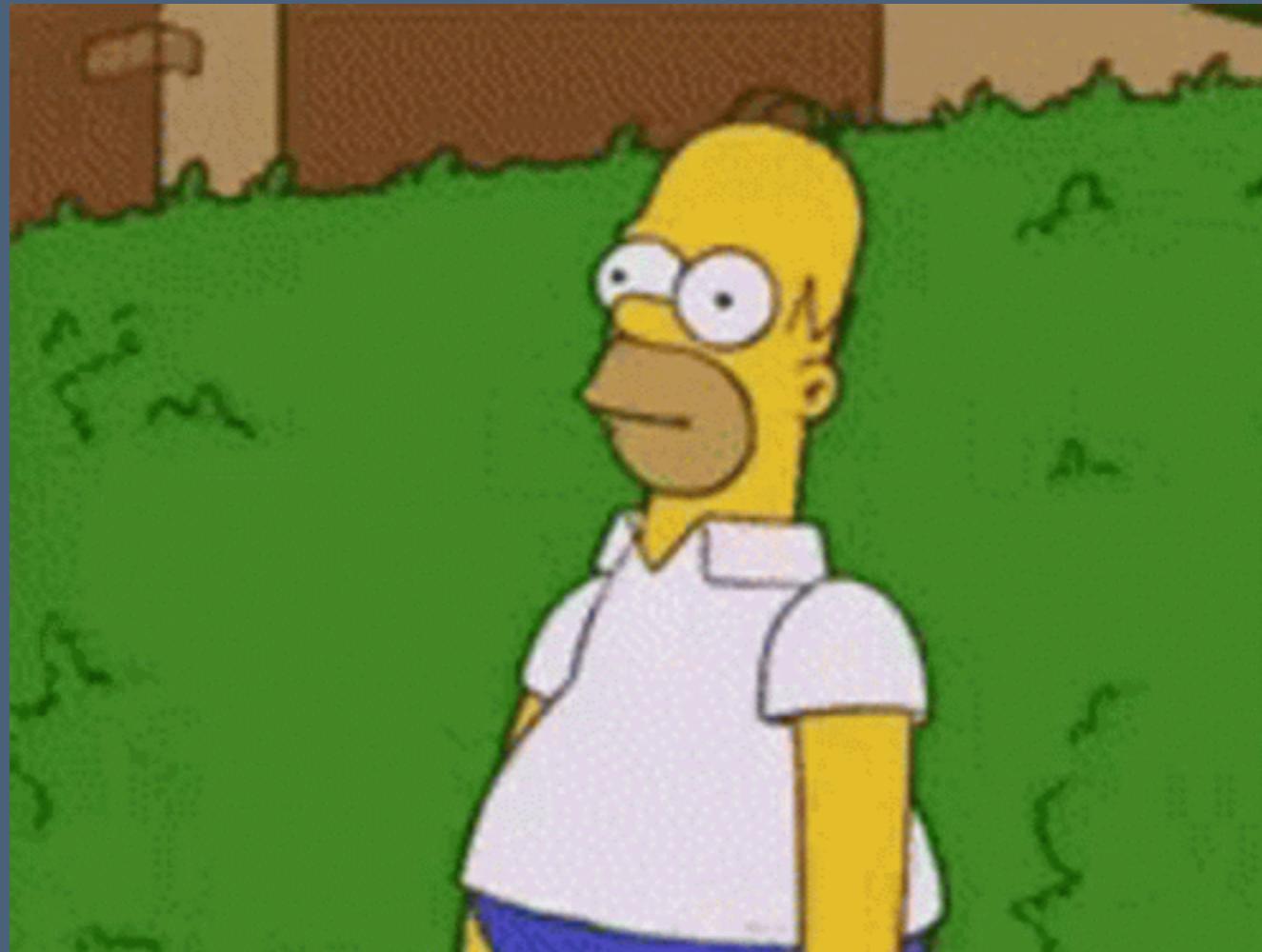


Practice

[=====>.....] 70%

Create a Project with GitHub

Now it's time for you to practice!



Practice

[=====>.....] 70%

Create a Project with GitHub

I want you to select a teammate and you create your project. It means that one needs to create the folder in GitHub and share the invitation, and the other needs to accept the invitation and download the project

You'll need to create a R silly script to share like:

```
library(ggplot2)
my_amazing_data <- iris
ggplot(my_amazing_data) +
  geom_point(aes(x = Sepal.Length, y = Sepal.Width, fill = Species), shape = 21) +
  scale_fill_manual(values = c("red", "blue", "green")) +
  theme_classic()
```

Practice

[=====>.....] 70%

Create a Project with GitHub

Once done, I want that the second one add something to the previous script and push it. The first one will have to pull once done to see it.

Practice

[=====>.....] 70%

Create a Project with GitHub

Once done, I want that the second one add something to the previous script and push it. The first one will have to pull once done to see it.

For the second exercise, I want you to download this class on your laptop. It will give you some hints if you need to refresh what we've seen today.



Theory and basics

[=====>...] 90%

Beyond data analysis

You can create one free GitHub account per mail. But you can generate as many free mails you want.



Theory and basics

[=====>...] 90%

Beyond data analysis

You can create one free GitHub account per mail. But you can generate as many free mails you want.



Theory and basics

[=====>...] 90%

Beyond data analysis

This is interesting because for each GitHub account, you can store FOR FREE one website. You can either store your personal website OR project website.

Theory and basics

[=====>...] 90%

Beyond data analysis

This is interesting because for each GitHub account, you can store FOR FREE one website. You can either store your personal website OR project website.

For that GitHub as one single restriction, the name of your website will be the name of your account

Theory and basics

[=====>...] 90%

Beyond data analysis

When you create a repository, you'll need to name it the name of your account followed by 'github.io'

The screenshot shows a web browser displaying a GitHub Pages site at jaycrlt.github.io. The page has a dark header with a navigation bar containing 'ABOUT ME', 'MY PUBLICATIONS', 'MY PICTURES', and 'CONTACT'. Below the header is a large image of two people on a boat, one taking a selfie. The main content area is titled 'ABOUT ME' and contains the following text:

I am a post-doctoral fellow with the Spanish Institute of Oceanography – Balearic Oceanographic Center at CSIC in Spain. I am mostly interested in measuring the physiological traits of marine organisms (e.g., calcification or net photosynthesis for benthic organisms, nutrients uptake for fish) to quantify how the ecosystem functioning is changing in the face of global change. My works involve a combination of fieldwork, *in situ* and ex-situ experiments, and advanced statistical modeling across local and large spatial scales.

You can download my full resume here: [Download](#)

last update: June 2025

Theory and basics

[=====>...] 90%

Beyond data analysis

Remember!

You can also set up really easy webpages using markdown, or you can use additional platforms to later store it via GitHub



MarkDown



Netlify

Reproducible science?

[=====>] 100%

The GitHub endeavor



Jérémie Carlot



CSIC
CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS



INSTITUTO
ESPAÑOL DE
OCEANOGRAFÍA



Reproducible science?

[=====>] 100%

The GitHub endeavor



Jérémie Carlot



CSIC
CONSEJO SUPERIOR DE INVESTIGACIONES CIENTÍFICAS



INSTITUTO
ESPAÑOL DE
OCEANOGRAFÍA

