Happy Git and Github for the useR
Session 03 - Early Github Wins

Boook club R-Ladies Bergen, R-Ladies Den Bosch, R-Ladies Amsterdam

Book by Jenny, presentation by Martine

Some house rules

- R-Ladies is dedicated to providing a harassment-free experience for everyone
- We do not tolerate harassment of participants in any form
- See the code of conduct

[Boook club R-Ladies Bergen, R-Ladies Den Bosch, R-Ladies Amsterdam]

Program for today

- Get started with GitHub
  - 15 New project, GitHub first
  - 16 Existing project, GitHub first
  - 17 Existing project, GitHub last
  - 18 Test drive R Markdown
  - 19 Render an R script

Get started with GitHub

- New project?
  - Easiest way: GitHub first
- Existing project?
  - GitHub first: pragmatic
  - GitHub last: more proper way to connect, especially if already a Git history

Chp 15 - New project, Github first

New project? Easiest way: GitHub first - Why?

- Also sets up local Git repo for immediate pulling and pushing
- Under the hood, we are doing `git clone`
- The remote GitHub repo is configured as the origin remote for your local repo and your local main branch is now tracking the main on GitHub

New project? Easiest way: GitHub first - Steps 1/n

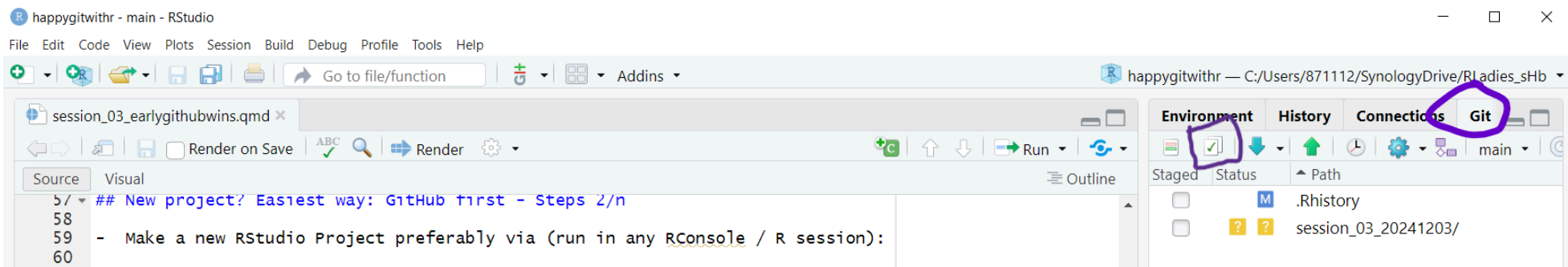
New project? Easiest way: GitHub first - Steps 2/n

- Make a new RStudio Project preferably via (run in *any* RConsole / R session):

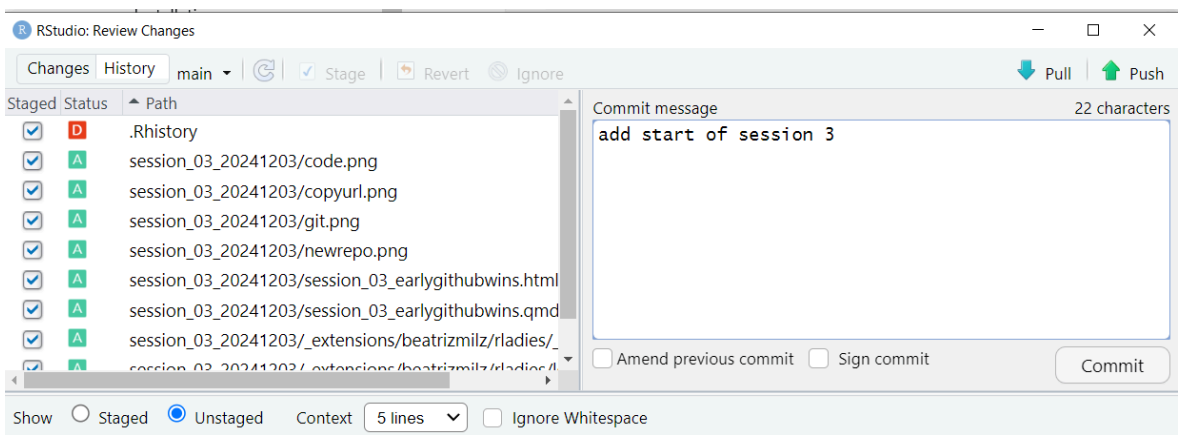
```
usethis::create_from_github(
  repo_spec = "https://github.com/YOU/YOUR_REPO.git",
  destdir = "~/path/to/where/you/want/the/local/repo/"
)
```

New project? Easiest way: GitHub first - work, save & com

- Work on project, then click on `Git` tab:



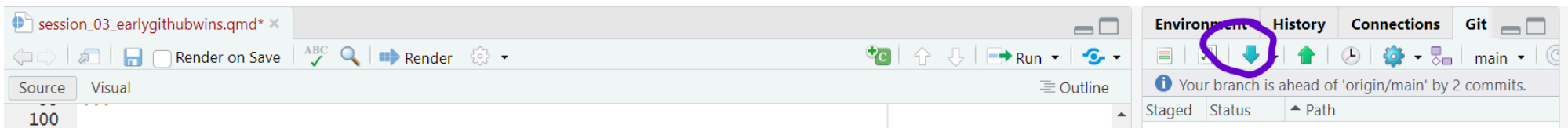
- Open `commit` screen by click on `Commit` button in git menu:



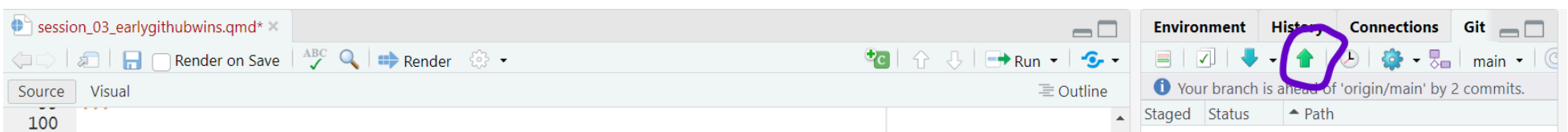
- check boxes of things you like to commit
- add `Commit` message
- hit `Commit` button at bottom

New project? Easiest way: GitHub first - push

- New work in your local Git repository, not online yet
- **Pull** from GitHub
  - Why? Establish this habit for the future! If you make changes to the repo in the browser or from another machine or (one day) a collaborator has pushed, you will be happier if you pull those changes in before you attempt to push.



- Probably nothing will happen, i.e. you'll get the message "Already up-to-date." This is just to establish a habit.
- Click the green "Push" button to send your local changes to GitHub:



New project? Easiest way: GitHub first - push

- But then this happened ☹

- I found this: https://forum.posit.co/t/issues-with-pushing-from-rstudi-to-git/126509
- I realised indeed my PAT had changed
- The webpage https://carpentries.github.io/sandpaper-docs/github-pat.html advised `usethis::git_sitrep()` and after running that I discovered:
  - Yes, my PAT could be the problem
  - An advise to read Chp 25 of the book we are reading https://happygitwithr.com/common-remote-setups.html 😄
- I ran `gitcreds::gitcreds_set()` and entered my PAT when asked
- Then I checked again `usethis::git_sitrep()` and saw it was oke
- Tried pushing again (after a pull ☺) and it worked 😊😊😊

[Boook club R-Ladies Bergen, R-Ladies Den Bosch, R-Ladies Amsterdam]

Chp 16 - Existing project, Github first

Existing project, GitHub first - Make repo

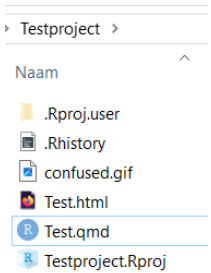
- slightly goofy way, in order to avoid using Git at the command line
- not for always, but nice for starters
- more options in the book, this seems to be the preferred

1. Have all your files for your R project in the same folder
2. Make a new repo on github
3. Now click the green "<> Code" button and copy the URL
4. In *any* R-session, run:

```
usethis::create_from_github(
  repo_spec = "https://github.com/YOU/YOUR_REPO.git",
  destdir = "~/path/to/the/folder/in/step/1/"
)
```

So far not so different from "New project, github first"

Existing project, Github first - demo 1/2



1. Made an R-project in a folder:
2. Made a test repo on github
3. and copied the url
4. Ran the code

```
usethis::create_from_github(
  repo_spec = "https://github.com/YOU/YOUR_REPO.git",
  destdir = "~/path/to/the/folder/in/step/1/")
```

Ran into:

```
i Defaulting to 'https' Git protocol
✓ Setting `fork = FALSE`
New project 'Test' is nested inside an existing project 'C[redacted]/Testproject/', which is rarely a good idea.
If this is unexpected, the here package has a function, `here::dr_here()` that reveals why '[redacted]Testproject/' is regarded as a project.
Do you want to create anyway?

1: Not now
2: Yup
3: No way

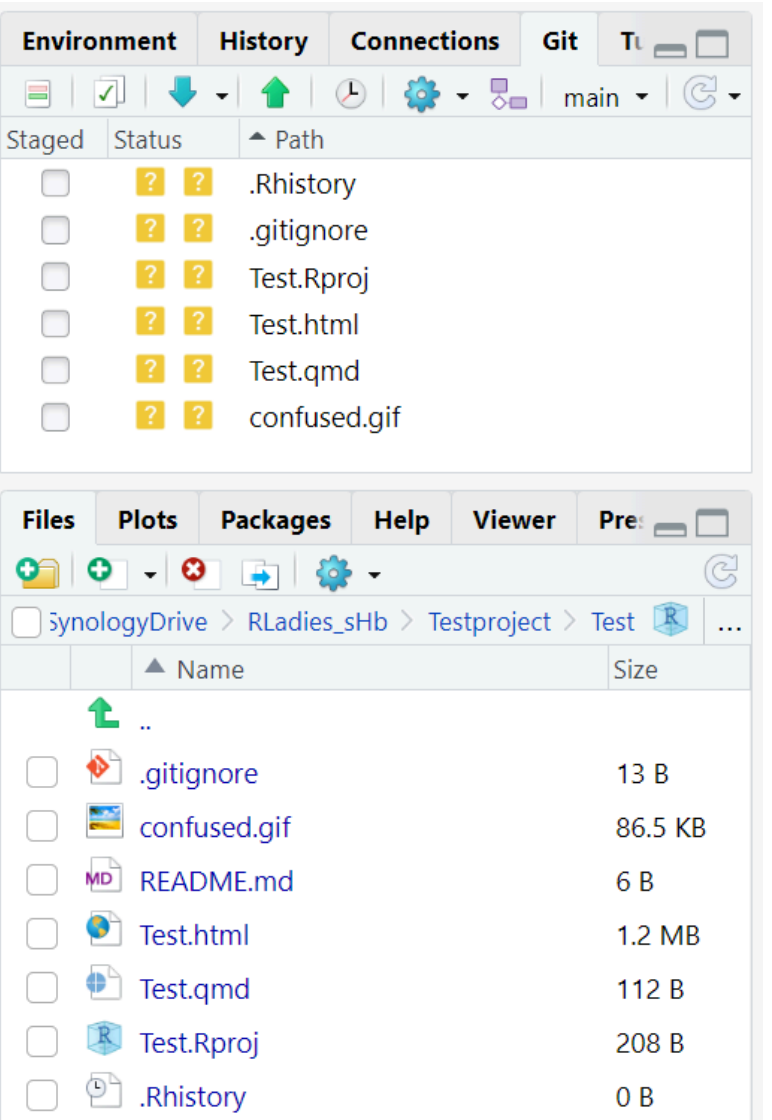
Selection:
```

Conclusion: JUST have the FILES in a folder, not already a project

Existing project, Github first - demo 2/2

Existing project, Github first - content in

- So after the local project is there, relocate the files to that folder
- Open the project in RStudio
- All the files should be here if your move/copy was successful
- Are they showing up in the Git pane with questions marks? They should be appearing as new untracked files.



Existing project, Github first - stage commit

Existing project, Github first - what to commit?

- .Rhistory probably not
- .RProj?
  - Posit says **yes** at https://forum.posit.co/t/should-rproj-files-be-added-to-gitignore/1269/2
- .gitignore?
  - People say **yes** at https://stackoverflow.com/questions/5765645/should-you-commit-gitignore-into-the-git-repos and https://stackoverflow.com/questions/767147/how-can-i-stop-gitignore-from-appearing-in-the-list-of-untracked-files

Existing project, Github first - push and check

[Boook club R-Ladies Bergen, R-Ladies Den Bosch, R-Ladies Amsterdam]

Chp 17 - Existing project, Github last

Existing project, Github last - Why? & How

- connecting existing local R project to GitHub,
  when for some reason cannott or don't want "GitHub first" workflow
  - example: existing project already git repo, with a history you care about

How?

EPGL - Prepare the local project (optional)

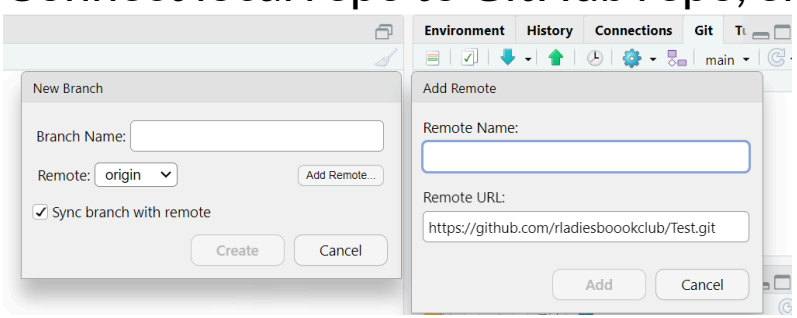
- Make the .RProj;
  - Within RStudio: `File > New Project > Existing Directory` **OR**
  - R console: `usethis::create_project("path/to/your/project")`
- Open the project
- Check or make it a git project:
  - `Git` tab available? Then it is oke
  - No `git` tab? , choose:
    1. R Console: `usethis::use_git()`
    2. RStudio: go to `Tools > Project Options … > Git/SVN`.
       Under "Version control system", select "Git". Confirm New Git Repository? Yes!
    3. In the shell, with working directory set to the project's directory, do `git init`.
  - Relaunch project automatically with options 1 and 2. Do it yourself for option 3

EPGL - Stage & Commit

- If your local project was already a Git repo and was up-to-date, move on.
- Otherwise:
  - Click the “Git” tab in upper right pane
  - Check “Staged” box for all files you want to commit.
    - Default: stage everything
    - *When to do otherwise: this will all go to GitHub. So consider if that is appropriate for each file. You can absolutely keep a file locally, without committing it to the Git repo and sending to GitHub. Just let it sit there in your Git pane, without being staged. No harm will be done. If this is a long-term situation, list the file in .gitignore.*
  - If you’re not already in the Git pop-up, click “Commit”
  - Type a message in “Commit message” & click “Commit” button

EPGL - Create and connect a GitHub repo

- With `usethis`: In your project, in the R Console; `usethis::use_github()`
  - Creates a new repo on GitHub & configures that as origin remote for local repo
  - Sets up your local default branch (e.g. main) to track same on origin and does an initial push
  - Opens the new repo in your browser
- Without `usethis`:
  - Create a new repo on GitHub, **DO NOT initialize this repository with anything**
  - Copy a clone URL
  - Connect local repo to GitHub repo, either with RStudio:



    - or command line:
      - `git remote add origin https://github.com/...../....git`
      - `git push --set-upstream origin main`

[Book club R-Ladies Bergen, R-Ladies Den Bosch, R-Ladies Amsterdam]

[Boook club R-Ladies Bergen, R-Ladies Den Bosch, R-Ladies Amsterdam]

Chp 18 - Test drive R Markdown

Make an Rmd file

- Launch RStudio in project that is a Git repo connected to GitHub repo
  - My example project was not on github, so I added it:

```
> usethis::use_github(organisation = "rladiesboookclub")
✓ Creating GitHub repository 'rladiesboookclub/Test'
✓ Setting remote 'origin' to 'https://github.com/rladiesboookclub/Test.git'
✓ Pushing 'main' branch to GitHub and setting 'origin/main' as upstream branch
✓ Opening URL 'https://github.com/rladiesboookclub/Test'
```

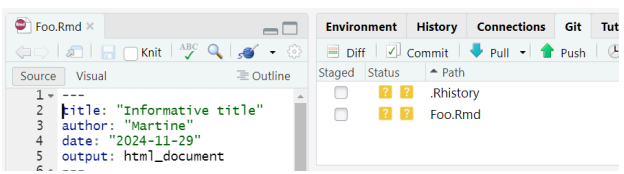
- Add new file (Rmd)

```
---
title: "Informative title"
author: "Martine"
date: "2024-11-29"
output: html_document
---

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

R Markdown
```

- After saving, you see



- Commit, Render to html, commit again and push
- Check files on github:
  - Rmd is readable, but no evaluated code
  - html, ah well

[Boook club R-Ladies Bergen, R-Ladies Den Bosch, R-Ladies Amsterdam]

Test / Foo.Rmd

MMJansen add foo

Code | Blame | 30 lines (21 loc) · 872 Bytes

```
---
title: "Informative title"
author: "Martine"
date: "2024-11-29"
output: html_document
---

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

R Markdown

This is an R Markdown document. Markdown is a simple formatti

When you click the **Knit** button a document will be generat

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

Including Plots

You can also embed plots, for example:

```{r pressure, echo=FALSE}
plot(pressure)
```
```

Test / Foo.html

MMJansen add the html

Code | Blame | 433 lines (350 loc) · 623 KB

```
<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />
<meta name="generator" content="pandoc" />
<meta http-equiv="X-UA-Compatible" content="IE=EDGE" />


<meta name="author" content="Martine" />

<meta name="date" content="2024-11-29" />

<title>Informative title</title>

<script>// Pandoc 2.9 adds attributes on both header and div.
// be compatible with the behavior of Pandoc < 2.8).
document.addEventListener('DOMContentLoaded', function(e) {
  var hs = document.querySelectorAll("div.section[class*='le
  var i, h, a;
  for (i = 0; i < hs.length; i++) {
    h = hs[i];
    if (!/^h[1-6]$/i.test(h.tagName)) continue;  // it shoul
    a = h.attributes;
    while (a.length > 0) h.removeAttribute(a[0].name);
  }
```

Output format

Advice

"Creating, committing, and pushing markdown (i.e., .md files) is a very functional, lighweight publishing strategy. Use output: github_document or, if output is html_document, add keep_md: true. In both cases, it is critical to also commit and push everything inside foo_files, i.e. any figures that have been created. Now people can visit and consume your work on GitHub, like any other webpage."

"… In this incremental manner, develop your report. Add code to this chunk. Refine it. Add new chunks. Go wild! But keep running the code "manually" to make sure it actually works. If the code doesn't work with you babysitting it, I can guarantee you it will fail, in a more spectacular and cryptic way, when run at arms-length via "Knit HTML" or rmarkdown::render().Clean out your workspace and restart R and re-run everything periodically, if things get weird."

[Boook club R-Ladies Bergen, R-Ladies Den Bosch, R-Ladies Amsterdam]

Chp 19 - Render an R script

Some remarks

- In R markdown, prose is top-level and code is tucked into chunks
- In R scripts, code is top-level and prose is tucked into comments
- R scripts can be rendered

Foo.Rmd | renderrscript.R | renderrscript.md

Source on Save | Run | So

```r
#' ---
#' title: "R scripts can be rendered!"
#' output: github_document
#' ---
#'
#' Here's some prose in a very **special** comment.
#' Let's summarize the built-in dataset `VADeaths`.
#  Here is a regular code comment, that will remain as such
min(VADeaths[,1])

#' Here's some more prose. I can use usual markdown syntax to make things
#' **bold** or *italics*. Let's use an example from the `dotchart()` help to
#' make a Cleveland dot plot from the `VADeaths` data. I even bother to name
#' this chunk, so the resulting PNG has a decent name.
#+ dotchart
dotchart(VADeaths, main = "Death Rates in Virginia - 1940")
```

- In de console

```
rmarkdown::render("renderrscr
```

Foo.Rmd | renderrscript.R | renderrscript.md

Preview on Save | Preview | Run

Source | Visual

```
R scripts can be rendered!
==========
871112
2024-11-29

Here's some prose in a very **special** comment. Let's summarize the
built-in dataset `VADeaths`.

```r
#  Here is a regular code comment, that will remain as such
min(VADeaths[,1])
```

    ## [1] 11.7

Here's some more prose. I can use usual markdown syntax to make things
bold or *italics*. Let's use an example from the `dotchart()` help
to make a Cleveland dot plot from the `VADeaths` data. I even bother to
name this chunk, so the resulting PNG has a decent name.

```r
dotchart(VADeaths, main = "Death Rates in Virginia - 1940")
```

<!-- -->
```

The end of session 3!

- **Meetup for the Chapters**
  - R-Ladies Amsterdam
  - R-Ladies Bergen
  - R-Ladies Den Bosch
- **We need YOU as a presenter!**
  - choose your own way of presenting: quarto-revealjs, ppt, …

[Boook club R-Ladies Bergen, R-Ladies Den Bosch, R-Ladies Amsterdam]