Week 2 Workshop: Setting up reproducible data science with R

Charlotte Hadley

1. Reproducible code

2. Reproducible referencing

3. Working with datasets in R reproducibly

4. Understanding the assessment template

Reproducible code

24/09/20

Reseach code is research software

If your research includes R code that reads and wrangles your data in such a way that you can:

- Create charts
- Test statistical hypotheses are compute statistical measures
- Calculate measures of your data

… then you're writing research software and it's an intrinsic part of your research.

You might even develop your R code into an R package that others can reuse for similar analyses.

However, that's definitely beyond the scope of this course.

Now you're writing software we need to equip you with some software development tools

24/09/20

Version control with git

Version control is extremely important in software development to ensure future changes to code do not break existing code.

This is particularly true when code is being worked on by multiple people.

Version control with git

Version control is extremely important in software development to ensure future changes to code don't break existing code.
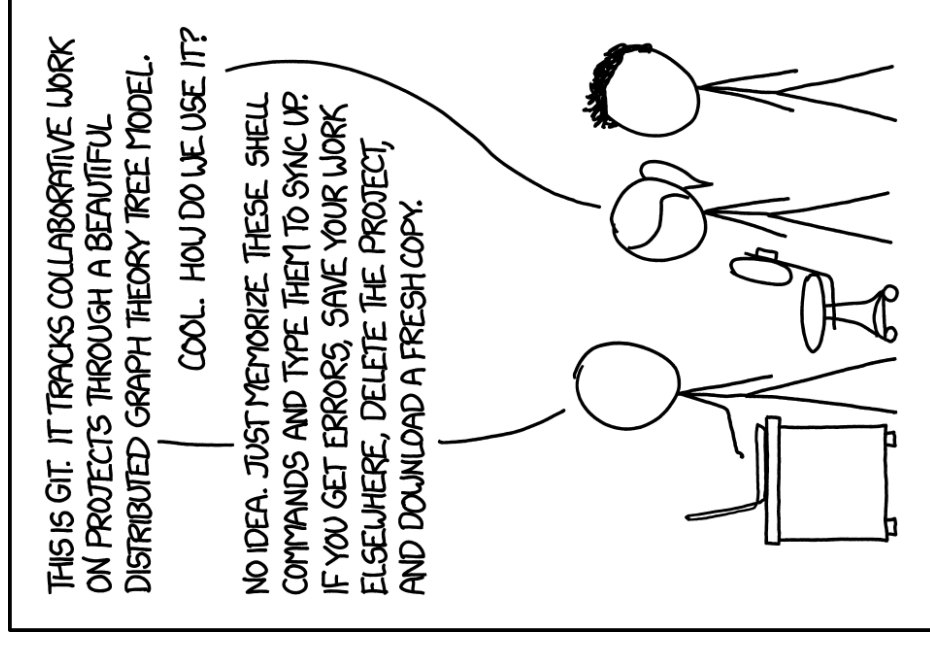
This is particularly true when code is being worked on by multiple people.

The most widely used version control tool used today is called g i t .

g i t is an extremely powerful, open-source tool.

We are going to ignore almost all of its features and use only what we need.

Just like most people who use it.

THIS IS GIT. IT TRACKS COLLABORATIVE WORK ON PROJECTS THROUGH A BEAUTIFUL DISTRIBUTED GRAPH THEORY TREE MODEL.

COOL. HOW DO WE USE IT?

NO IDEA. JUST MEMORIZE THESE SHELL COMMANDS AND TYPE THEM TO SYNC UP. IF YOU GET ERRORS, SAVE YOUR WORK ELSEWHERE, DELETE THE PROJECT, AND DOWNLOAD A FRESH COPY.

https://xkcd com/1597/

git repositories

Each individual project that uses git is called a repository.

If you'd like to use git to version control your assignment then that would be a repository.

24/09/20

git and the command line

git was originally designed to be used in the command line

- On Windows that usually means using cmd.exe

- On macOS and Linux that means the Terminal

This practically means that you'd be using a command line interface (CLI) to work with the git technology.

In the case of git it can be beneficial to use a GUI interface instead of the CLI to conceptualise what actions you're about to - and often more importantly - what actions were taken in the past.

GitHub

GitHub is one of many different tools for managing git repositories.

We're going to prefer GitHub because it has lots of really awesome features.

It's also owned by Microsoft which radically simplifies installing `git` onto your machine.

Let's first create an account.

1. Visit github.com and click sign-up (if you don't already have an account)

2. Don't worry about being creative with your username now - you can change it later without things exploding too much.

GitHub Desktop

When you install GitHub Desktop it installs both git and a GUI for working with repositories on GitHub.

1. Download the application from desktop.github.com/

2. Run through the installer

3. Sign-in with your GitHub account.

Organising your computer

From this point onwards we need to be careful about organising our code.

- RStudio projects should not contain projects themselves

- Git repositories should never contain another git repository[1]

It's best practice to have one folder on your machine call GitHub and for all repositories to be stored in that folder.

[1] This is getting into quite advanced territory, but if you needed a repo to depend on another you would use submodules.

24/09/20

Creating a new repository

We're going to make use of the {usethis} package to help us create a well-formed new repository.
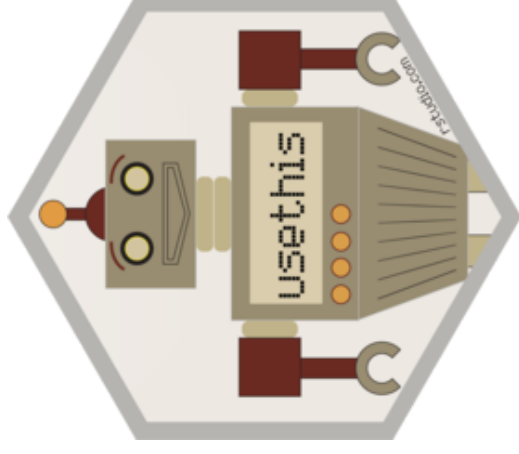
Please open RStudio and install the {usethis} package.
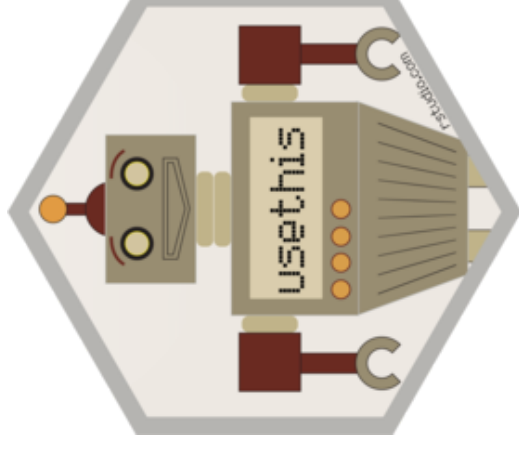
24/09/20

Creating a new repository

We're going to make us eof the {usethis} package to help us create a well-formed new repository.

I'm going to demons rate how to do this first afterwards I'll s bw the ins ructions and then create your own repos tory.

Creating a new repository

1. In RStudio use the "Project Menu" choose

- New Project...
- New directory
- New project

2. Set the name of the directory as e n g 7 2 1 8 - w e e k - 2 - w o r k s h o p

3. Create the project as a subdirectory of your GitHub folder

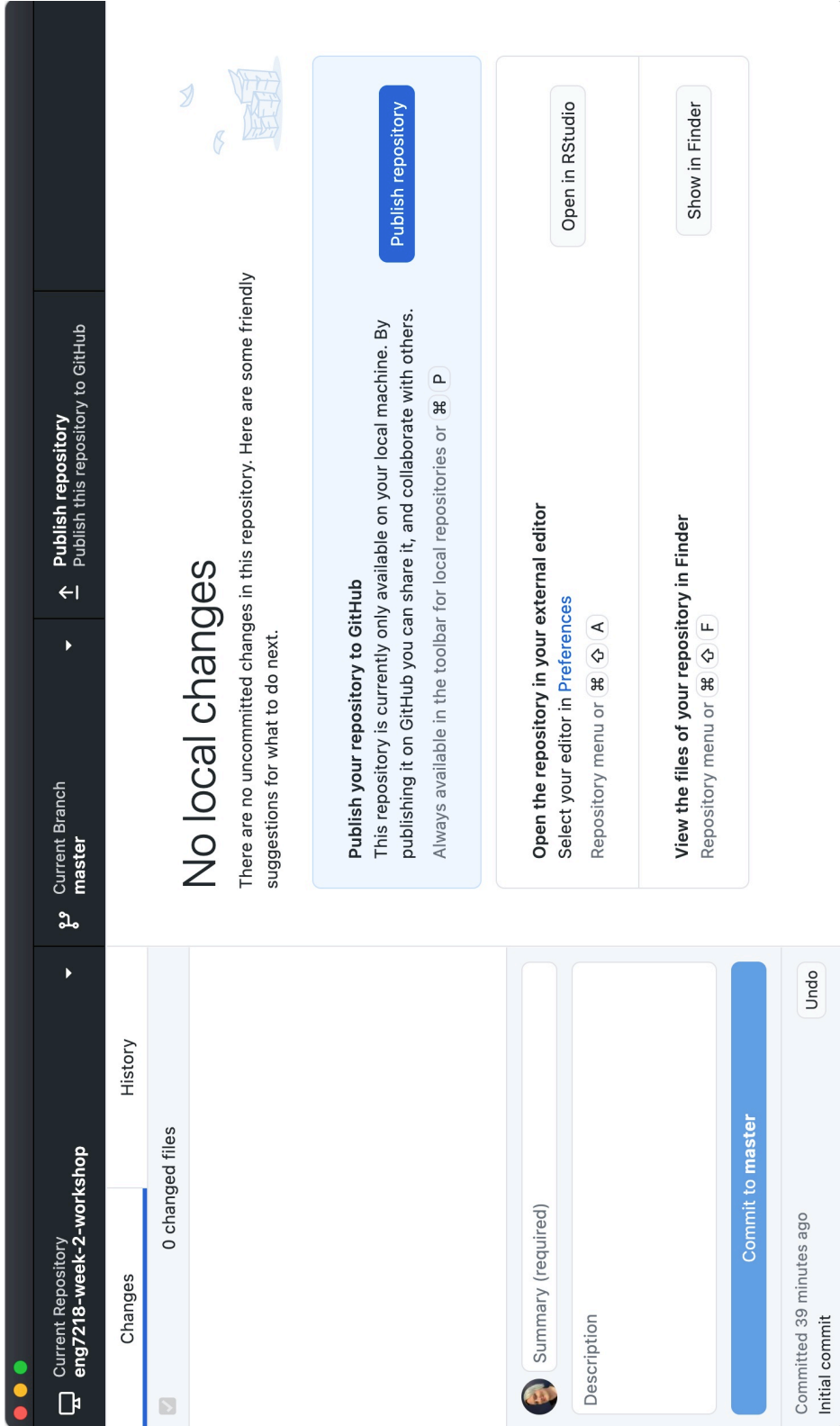4. Inside of this project run this in the R console

```
usethis::use_git()
```

5. Open GitHub Desktop and select

```
File > Add Local Repository
```

Publish to GitHub.com

If everything is good your GitHub Desktop will look like this - let's publish the repository!

Publish to GitHub.com

- We then get to name the repository on Github.com - use these as a name!

- We then get to choose if our code is private or not

  - If you're applying an emb ago to your d aa, you prob a want to do the same on GitHub!

  - Pub ic repositories can b seen by anyone and are us ful for your CV

Publish Repository

GitHub.com | GitHub Enterprise

Name
eng7218-week-2-workshop

Description

☑ Keep this code private

Organization
None

Cancel | Publish Repository

Repository menu or ⌘ ⇧ F

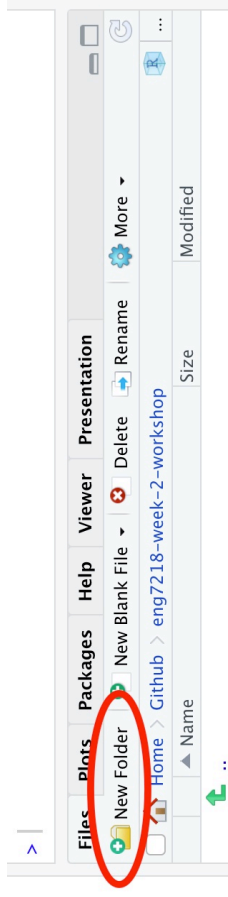# Let's add some sensitive data to the repo (I)

When adding data to your repository it's important to have two directories .

- d a t a - r a w contains your sensitive data and data anonymisation scripts

- d a t a contains a copy of your anonymous data

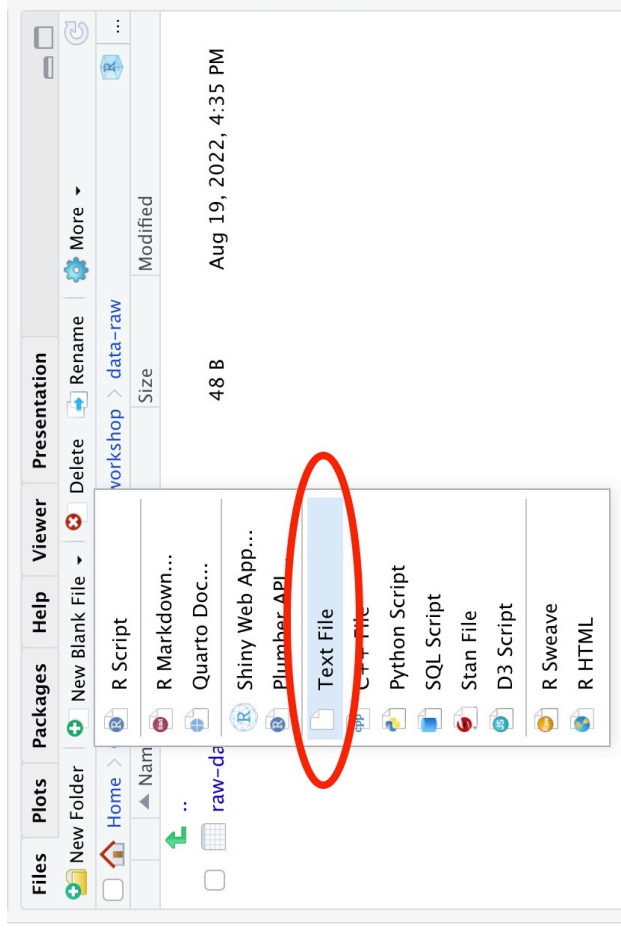We're going to prevent the d a t a - r a w folder from being sent to GitHub.com

24/09/20

Let's add some sensitive data to the repo (II)

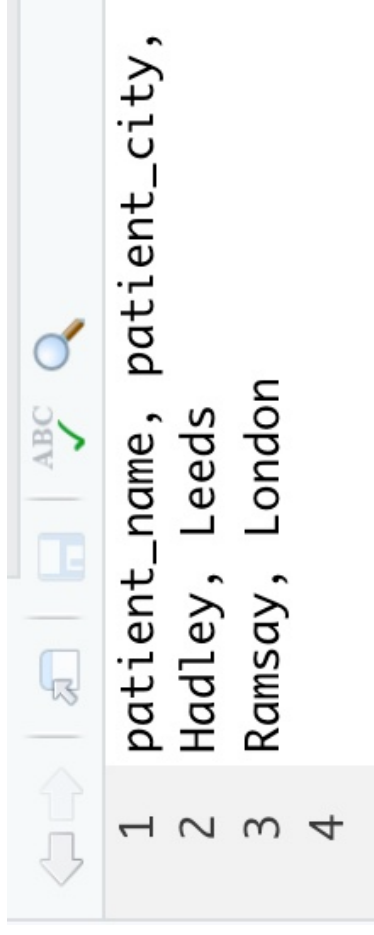1. Use the New Folder button to create a folder called d a t a - r a w

Files | Plots | Packages | Help | Viewer | Presentation

New Folder | New Blank File ▾ | Delete | Rename | More ▾

Home › Github › eng7218-week-2-workshop

▲ Name | Size | Modified

Let's add some sensitive data to the repo (II)

1. Use the "New Folder" button to create a folder called `data-raw`

2. Click inside the `data-raw` folder in the Files tab

3. Select "Text file" from the "New Blank File" menu

Let's add some sensitive data to the repo (II)

1. Use the "New Folder" button to create a folder called `data-raw`

2. Click inside the `data-raw` folder in the Files tab

3. Select "Text file" from the "New Blank File" menu

4. Add some faked data into the file

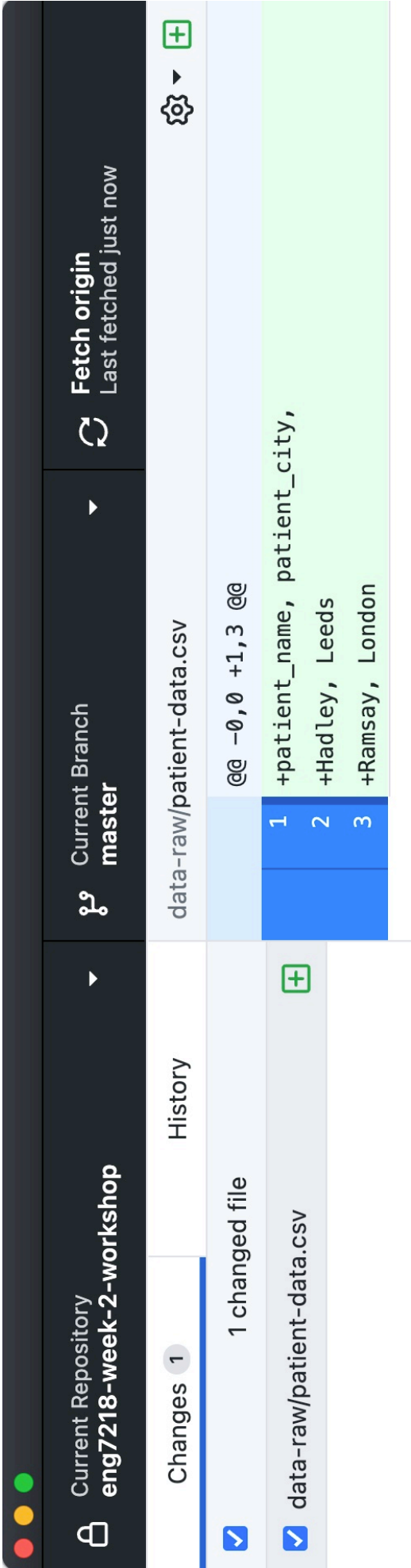5. Save this file into the `data-raw` folder and call it something like `patient-data.csv`

1	`patient_name, patient_city,`
2	`Hadley, Leeds`
3	`Ramsay, London`
4	

Check GitHub Desktop

When you switch to GitHub Desktop we see that git has tracked changes in the repo.

We want to prevent this entire directory from going to GitHub

- The repo on your machine is called "the local repository"
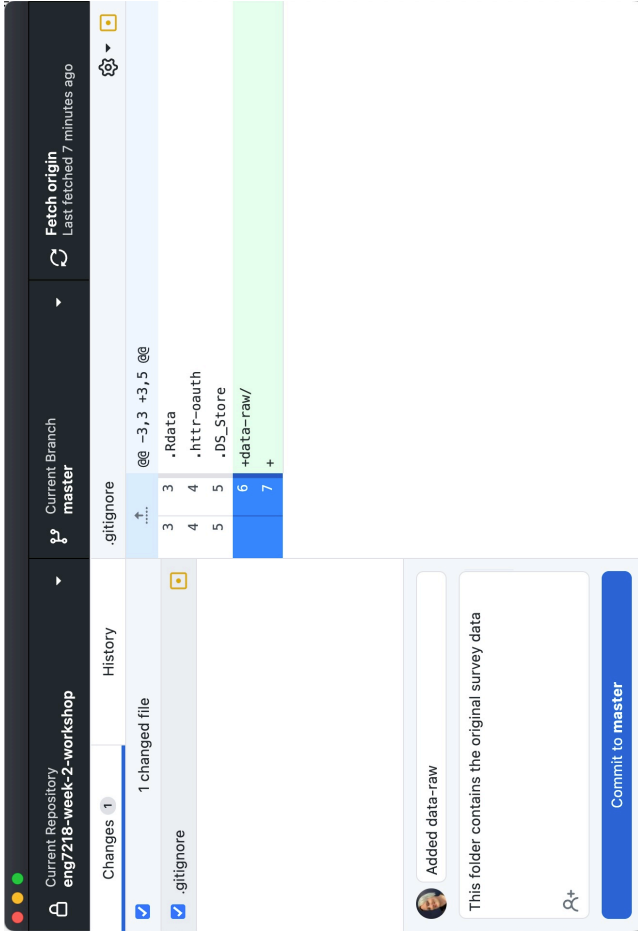
- The repo on Github is called "the remote repository"

Current Repository
eng7218-week-2-workshop

Current Branch
master

Fetch origin
Last fetched just now

Changes 1   History

1 changed file

data-raw/patient-data.csv

data-raw/patient-data.csv

```
@@ -0,0 +1,3 @@
1  +patient_name, patient_city,
2  +Hadley, Leeds
3  +Ramsay, London
```

Commit this change to the repository

1. Switch back to GitHub

2. Add a commit message

   These should be short and sweet, like "add data-raw"

3. Optionally add a summary

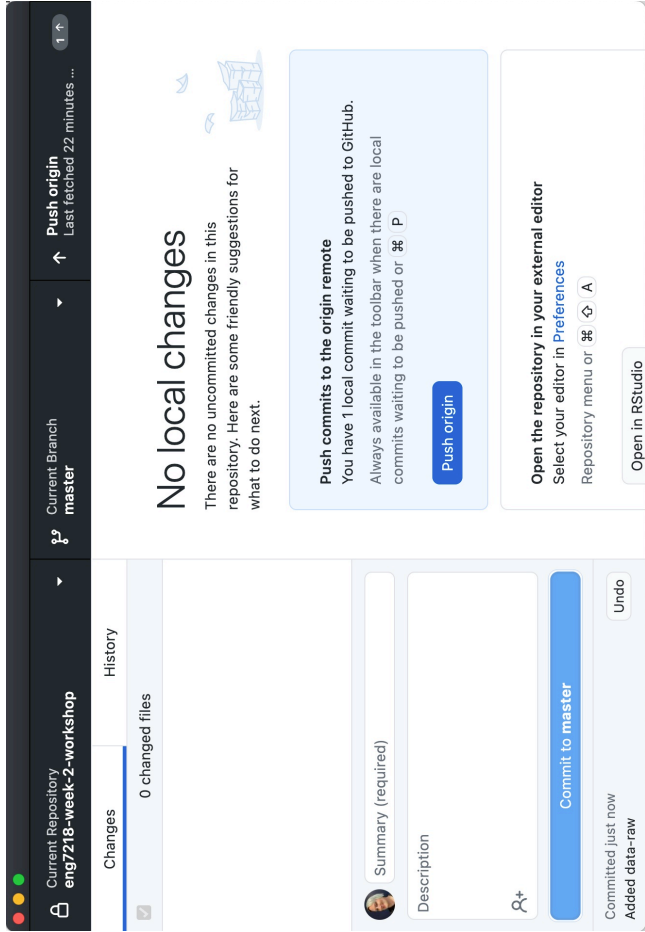4. Click "Commit to master"

In git, when we "commit" it's something we tell git to remember your change in the local repository.

Pushing our changes to the remote repository

After committing locally it's time to push the changes to GitHub.com

Click the "push origin" button and the changes will be sent to the remote!

Pulling is the opposite of pushing

Remember that git is designed for collaboration.

Imagine a collaborator has pushed a change to the remote repository but you don't have those changes in your machine.

If you refresh GitHub Desktop you'll be prompted to "pull origin" - which will bring the changes to your local repository.
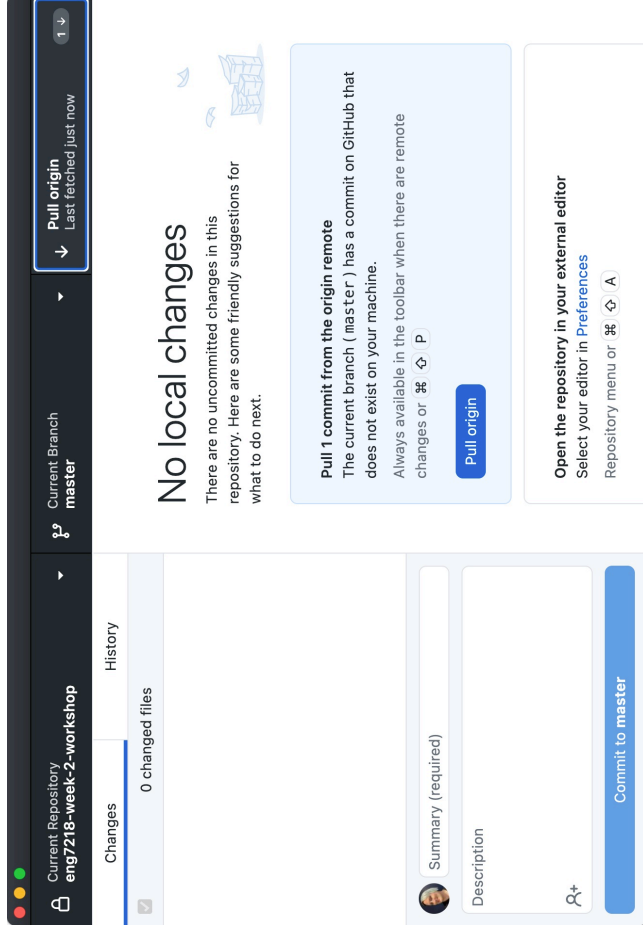
GitHub branches and pull requests

This is as far as we'll go with using git for version control.

In real-world projects you would use two additional features: branches and pull requests.

There is a really good simple to follow tutorial here: https://docs.github.com/en/get-started/quickstart/hello-world

There are also tutorials on LinkedIn Learning.

24/09/20

Modify the .gitignore file

1. Switch back to RStudio

2. Go to the project directory

3. Open the .gitignore file

4. Add data-raw/ to the file and save it

```
.gitignore*

1   .Rproj.user
2   .Rhistory
3   .Rdata
4   .httr-oauth
5   .DS_Store
6   data-raw/
7
```
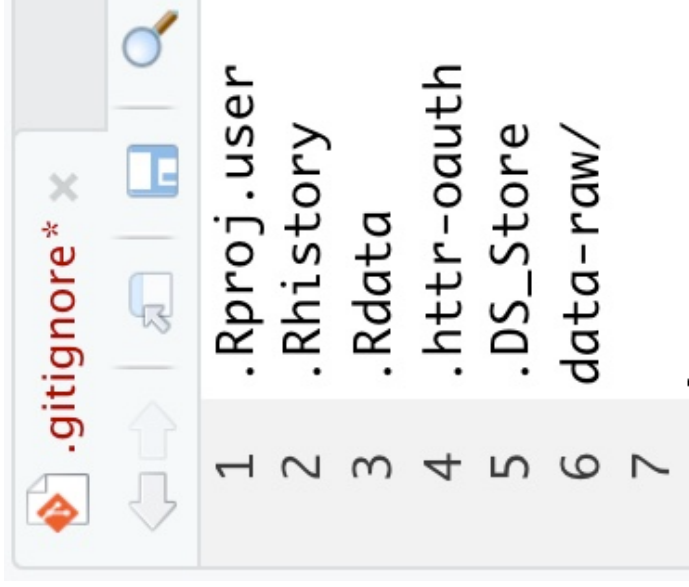
Connecting Figshare & Github

GitHub is an awesome place for code to live.

It's a bad choice for preserving code for the future.

Fortunately, it's simple to connect your Figshare account with GitHub which gives you a fairly reproducible coding workflow.

Reproducible referencing

https://engplaygroundworkshop/week-02workshop_reproducibility

Referencing is part of research (and the assessment)

Managing your references is an unavoidable part of doing research and is part of the assessment.

There are several reference management tools available.

- EndNote is widely taught however it's close source, expensive and [in my own personal opinion] rubbish

- Mendeley is much better but it's closed source has limited shared library functionality.

- Zotero is open source, has good shared library functionality and has lots of extensions.

I'm going to teach you how to use Zotero. But choose which software works for you!

Setting up Zotero

1. Create a free account at zotero.org/

2. Download the software from zotero.org/

3. Install the software and login to your Zotero account

24/09/20

Improving Zotero with Better Bibtex for Zotero

We're going to install the "Better Bibtex" addin which improves how Zotero generates bibtex files.

BibTeX is an extremely useful and very widely used tool for storing, managing and typsetting references
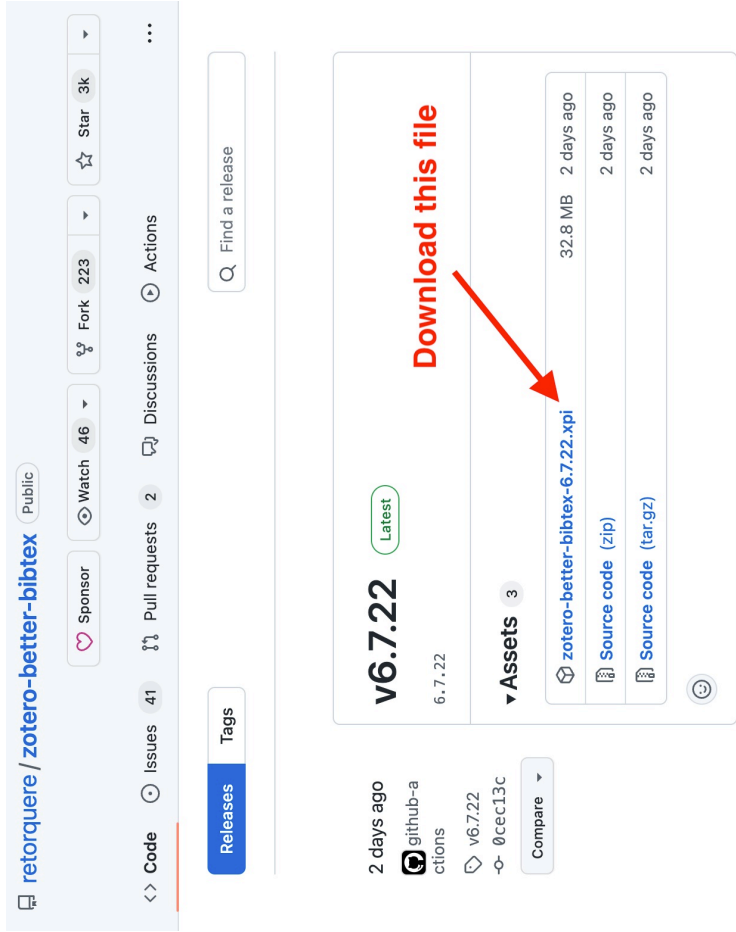
It was originally designed to work with LaTeX which we're purposefully avoiding.

24/09/20

Improving Zotero with Better Bibtex for Zotero

The instructions for installing Better Bibtex.

1. Download the tool from

https://github.com/retorquere/zotero-better-bibtex/releases

Download this file

Improving Zotero with Better Bibtex for Zotero

The instructions for installing Better Bibtex.

1. Download the tool from
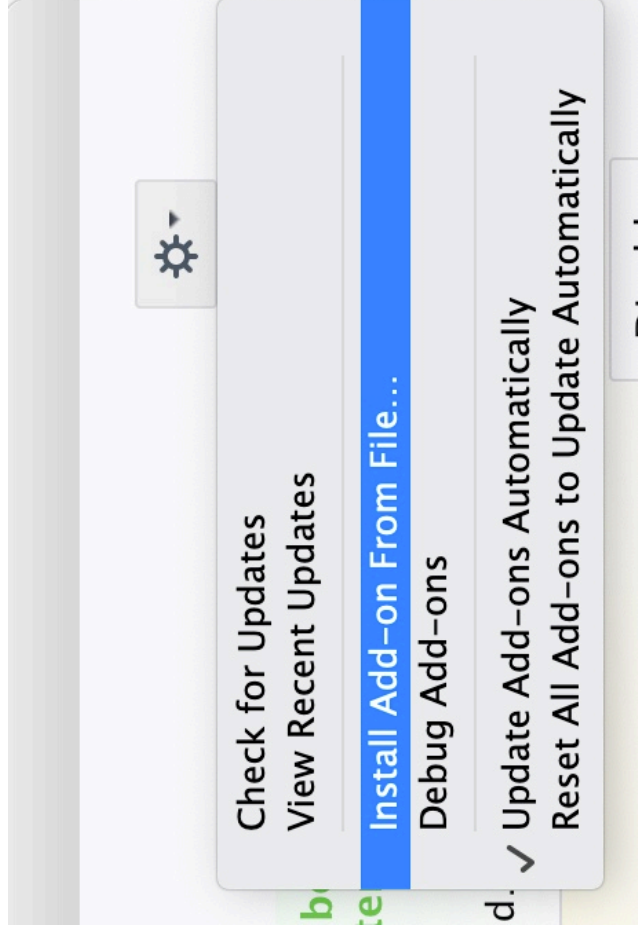https://github.com/retorquere/zotero-better-bibtex/releases

2. Open Zotero and select

   T o o l s > A d d - o n s

3. In the popup click on the ⚙ item and select "Install Add-on From File..."

4. Select the .xpi file you just download ed

5. Restart Zotero

Web browser plug-in

One of the best things about Zotero is how easy it is to add resources to your collection directly from the web browser

Please install the plugin for your browser of choice

zotero.org/download/connectors

Home > Connectors

Get Zotero

Install the Zotero Connector for Chrome

Zotero Connectors allow you to save to Zotero directly from your web browser.

Install

For Mac, Windows, and Linux

Firefox connector
Install

Edge connector
Install

Safari connector
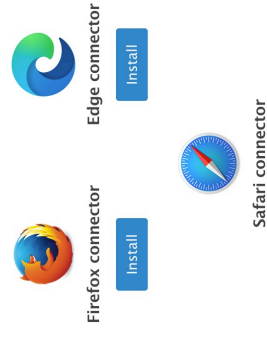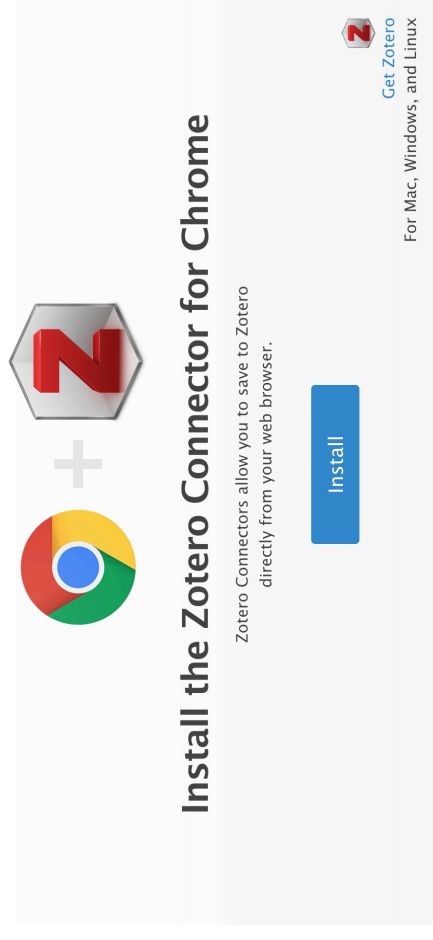
Adding a paper to Zotero

I'd like you to add the "Why Most Published Research Findings are False" paper to your Zotero collection.

You need your browser and Zotero open simultaneously.

1. Visit
   d oi.org/10.1371/journal.pmed.0020124

2. Click on the Zotero plugin icon and click "Done"

3. Go over to Zotero and you'll see the paper has been added!

Using Zotero

You can now use Zotero to organise your papers, web pages and reports. All of the things you need to cite and reference.

- Zotero includes a PDF viewer where you can annotate and highlight PDFs.

- You can organise things via folders.

- Tags are another useful tool for organising research
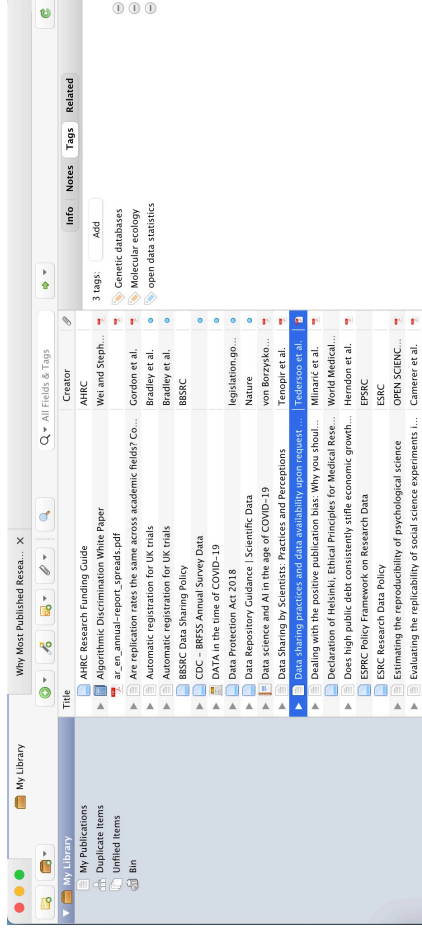
Connecting this to an RStudio project

Our goal is to set up a workflow where we can synchronise our Zotero library with an RStudio project so that we can write a report and add references as we grow our library.

We installed Better BibTeX because it allows us to automate this process.

You'll need to do this for each project you want to connect to your Zotero collection.

Sync Zotero with an RStudio Project

1. Add an .Rmd document to your RStudio project called something like `Report.Rmd`

- The YAML header is looking for a BibTeX file called `references` which we've not created yet

- We've add a "References" header before the references.

```
---
title: "Example report"
output:
  pagedown::html_paged:
    toc: false
knit: pagedown::chrome_print
bibliography: references.bib
---

# References

::: {#refs}
:::
```

Sync Zotero with an RStudio Project

1. Add an .Rmd document to your RStudio project called something like `Report.Rmd`

2. Open Zotero and go to `File > Export library`
   - Set the format as Better BibTeX
   - Select "Keep updated"

3. Save the file into your RStudio project and call it `references.bib`

Format: Better BibTeX

Translator Options

- ☐ Export Notes
- ☐ Export Files
- ☐ Include Annotations
- ☐ Use Journal Abbreviation
- ☑ Keep updated
- ☑ Background export

OK   Cancel

Sync Zotero with an RStudio Project

1. Add an .Rmd document to your RStudio project called something like `Report.Rmd`

2. Open Zotero and go to `File > Export library`

   o Set the format as Better BibTeX

   o Select "Keep updated"

3. Save the file into your RStudio project and call it `references.bib`

4. Open Zotero preferences and go to the Better BibTeX section

   • Choose the "Automatic Export" tab

   • Set "Automatic export" to "On Change"

Better BibTeX

| General | Sync | Search | Export | Cite | Advanced | Better BibTeX |

Citation keys | Export | Automatic export | Import | Miscellaneous

You can only review and remove exports here. To add an auto-export, perform an export as usual and check the 'Keep updated' option presented there.

Automatic export   On Change

My Library
My Library

Sync Zotero with an RStudio Project

1. Add an .Rmd document to your RStudio project called something like `Report.Rmd`

2. Open Zotero and go to `File > Export library`

3. Save the file into your RStudio project and call it `references.bib`

4. Save the file into your RStudio project and call it `references.bib`

5. Open Zotero preferences and go to the BetterBibTeX section

6. In your .Rmd file cite the paper we added using the citation key... and knit the document!

```
---
title: "Example report"
output:
  pagedown::html_paged:
    toc: false
knit: pagedown::chrome_print
bibliography: references.bib
---

Ah, yes the @ioannidis_why_2005

# References

::: {#refs}
:::
```
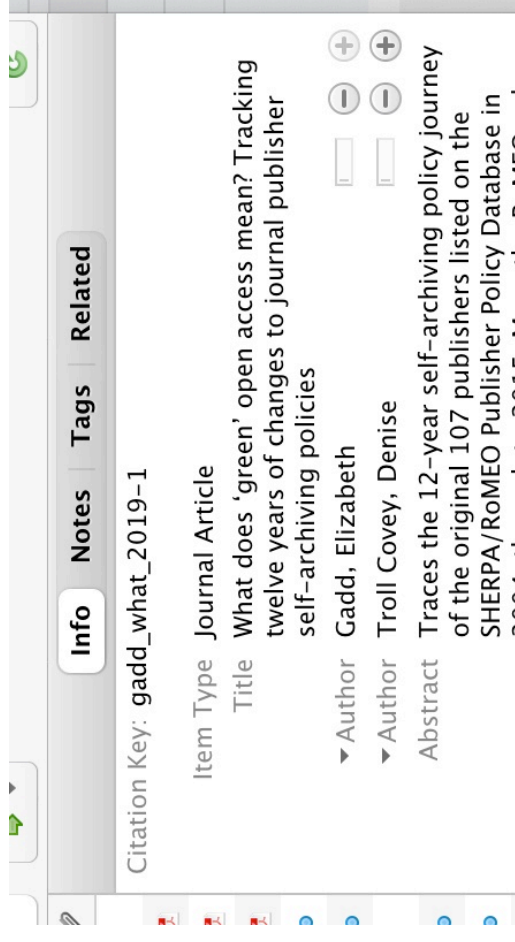
Zotero citation keys

Zotero automatically generate citation keys for you.

I strongly against editing these yourself - they're best as machine-readable data instead of human-read a ble aat

Citation Key: gadd_what_2019-1

| Info | Notes | Tags | Related |

Item Type   Journal Article

Title   What does 'green' open access mean? Tracking twelve years of changes to journal publisher self-archiving policies

▶ Author   Gadd, Elizabeth

▶ Author   Troll Covey, Denise

Abstract   Traces the 12-year self-archiving policy journey of the original 107 publishers listed on the SHERPA/RoMEO Publisher Policy Database in

Challenge

1. Find a paper from a journal

2. Add it to Zotero with the browser plug-in

3. Cite the new paper in your RMarkdown document

4. Knit the .Rmd file together and see the new item in your references

> You might want to look at the last few slides of Week 2's lecture slides which contain at least 10 references.

Zotero and MS Word

Good news, Zotero should automatically install a add-in for citing your collection within MS Word.

There are really good instructions on how to use this add-in on the Zotero web site.

1. Open MS Word and look for a Zotero tab. If it's not there continue with these instructions.

2. Open Zotero preferences and go to Cite > Word Processors

3. Click "Re-install Microsoft Word Add-in

4. Restart Zotero and MS Word

Adding things that aren't papers to Zotero

It's really easy to use Zotero with papers that have DOI.

But if you want to cite a website or white paper you might want to add this manually to your collection.

- Ensure that all items have a date.
  - ○ Often web pages reports don't have dates, but check the URL just in case
    `turing.ac.uk/.../2021-06/...pdf`
  - ○ If no date is available go for the current year
- If no authors are listed then use the name of the website or institution
- Add an "access date"

24/09/20

Week of Workshop 2

Shared Zotero collection

I've created a shared Zotero collection for this semester -
https://www.zotero.org/groups/4758381/eng7218-shared-2022/library.

Let's discuss how we might use this collection as a class.

What reproducibility skills have we learned?

- You need an ORCID! Get one and use it everywhere.

- Use Figshare to reserve DOI for research outputs before you start
  - It'll make it easier to write your research up!

- Use Figshare to store your talks, posters and anything else you want to make available for the future

- Use GitHub for version controlling your code
  - Keeps as tived aabff GitHub with the .gitignore file

- Connect your Figshare and GitHub repos to make your code citable and avaiable for the future

- Use Zotero for storing and managing all your references

- Use the Better BibTeX add-in to keep your .bib file up to date in an RStudio project.

24/09/20

Advanced topics (for the end)

To completely ensure the reproducibility of R code projects you would need to ensure that future users have access to the exact same packages as you used.

The best solution for this is to use the {renv} package.

I've personally avoided it for projects recently because I haven't worked on code that I need others to replicate into the future. If you're writing research software then please do consider using it. There's good documentation for it.