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

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

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.

Version control with git

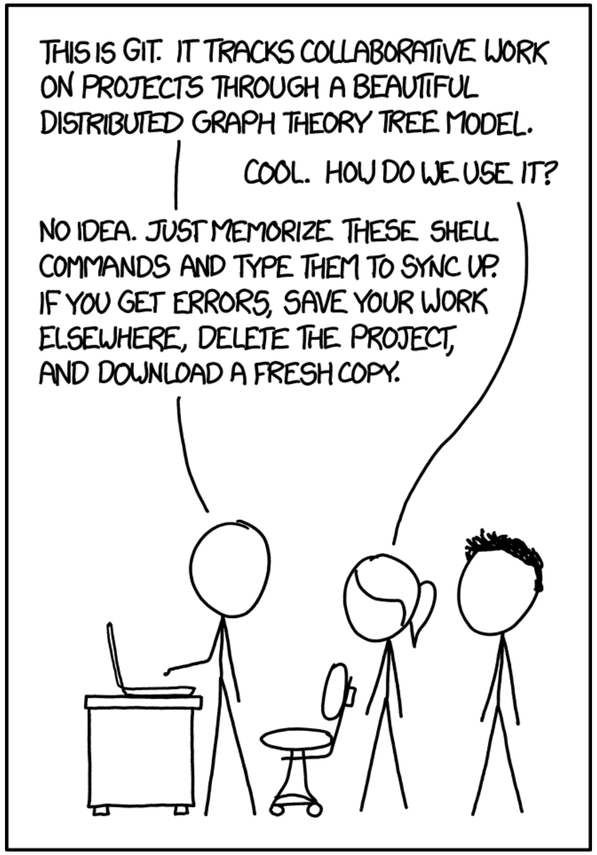
Version control is extremely important in software development to ensure future changes to code don't break existing code.

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

The most widely used version control tool used today is called `git`.

`git` 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 decided to use `git` to version control your assignment then that would be a repository.

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.

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.

Creating a new repository

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

I'm going to demonstrate how to do this first **afterwards** I'll show the instructions and then create your own repository.

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 `eng7218-week-2-workshop`

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

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

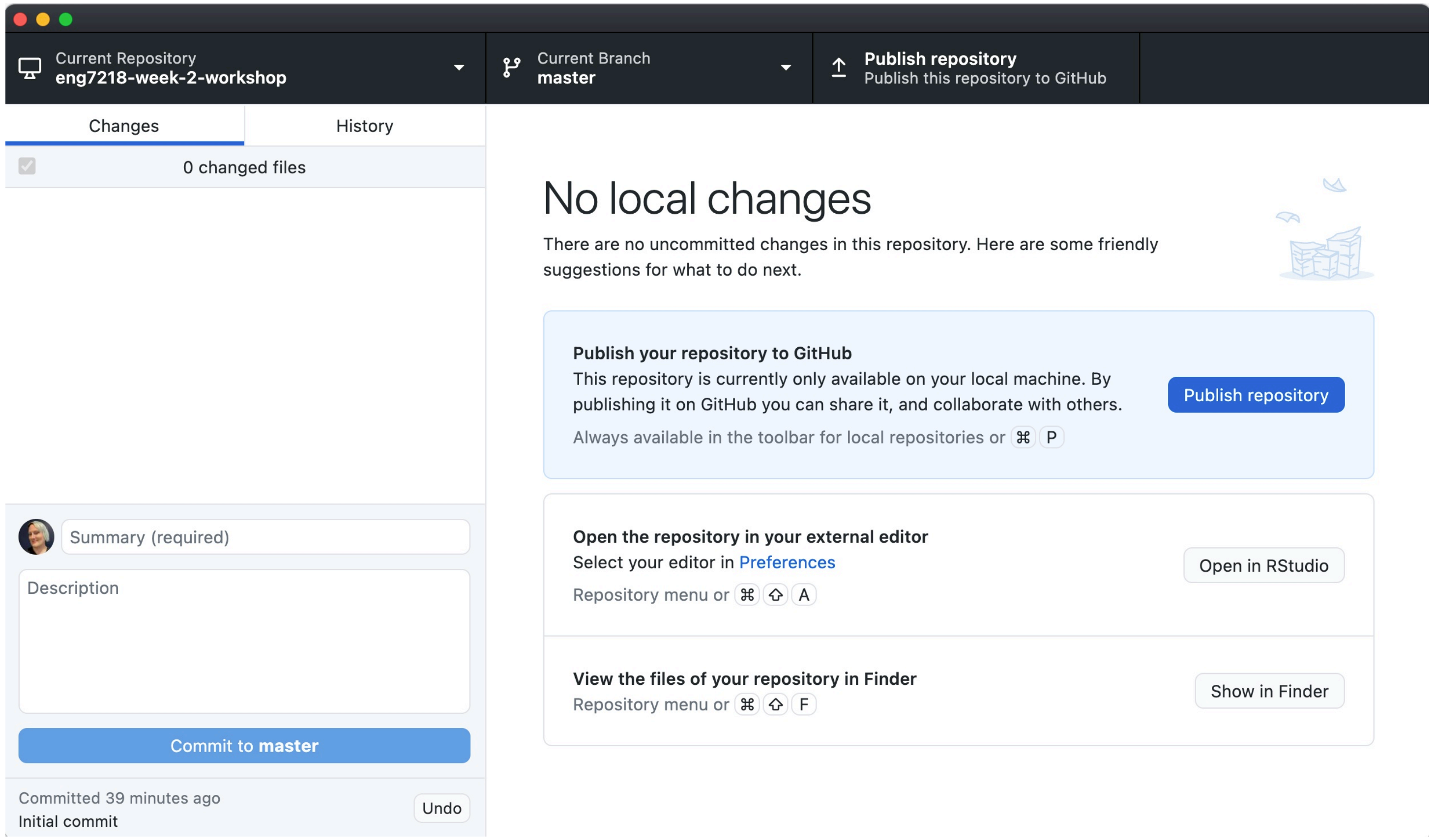
```
usethis::use_git()
```

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!



Current Repository
eng7218-week-2-workshop

Current Branch
master

Publish repository
Publish this repository to GitHub

Changes | History

0 changed files

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Publish your repository to GitHub
This repository is currently only available on your local machine. By publishing it on GitHub you can share it, and collaborate with others.
Always available in the toolbar for local repositories or ⌘ P

Publish repository

Open the repository in your external editor
Select your editor in Preferences
Repository menu or ⌘ ⇧ A

Open in RStudio

View the files of your repository in Finder
Repository menu or ⌘ ⇧ F

Show in Finder

Summary (required)

Description

Commit to **master**

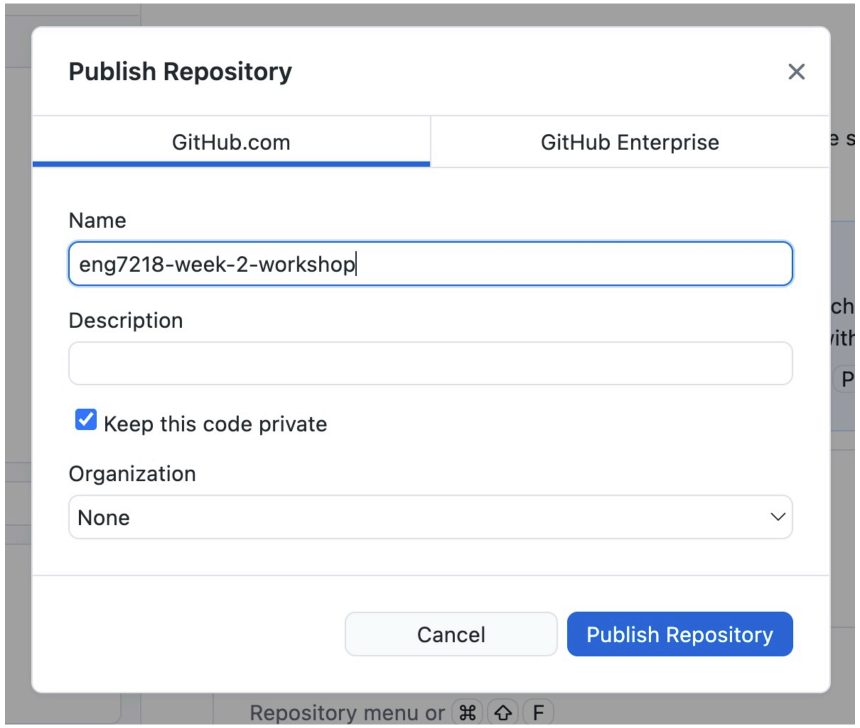
Committed 39 minutes ago
Initial commit

Undo

Workshop 15 / 48

Publish to GitHub.com

- We then get to name the repository on Github.com - use the same name!!
- We then get to choose if our code is private or not
    - If you're applying an embargo to your data, you probably want to do the same on GitHub!
    - Public repositories can be seen by anyone and are useful for your CV



Publish Repository

Name: eng7218-week-2-workshop

Description:

[✓] Keep this code private

Organization: None

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

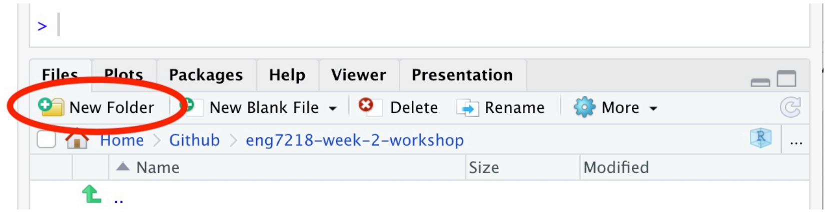
When adding data to your repository it's important to have **two** directories.

- `data-raw` contains your sensitive data and data anonymisation scripts
- `data` contains a copy of your anonymous data

> We're going to prevent the `data-raw` folder from being sent to GitHub.com

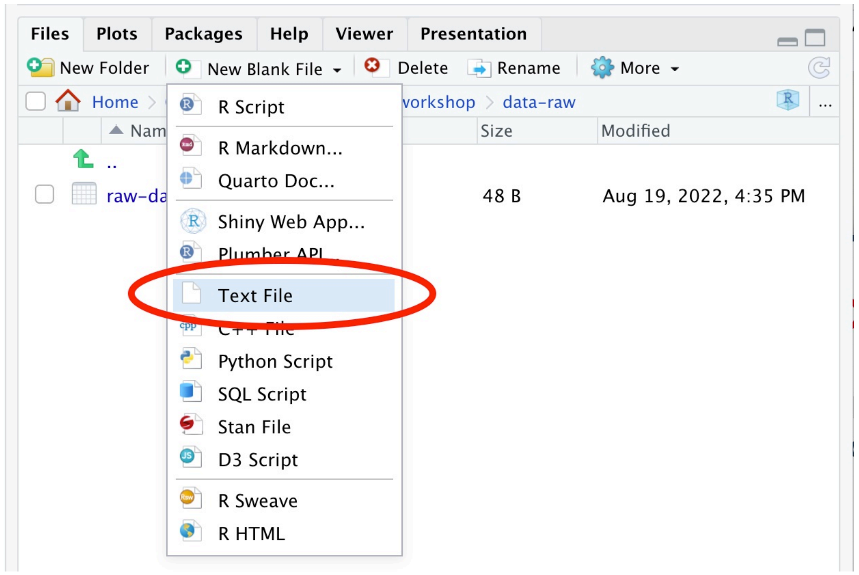
Let's add some sensitive data to the repo (II)

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



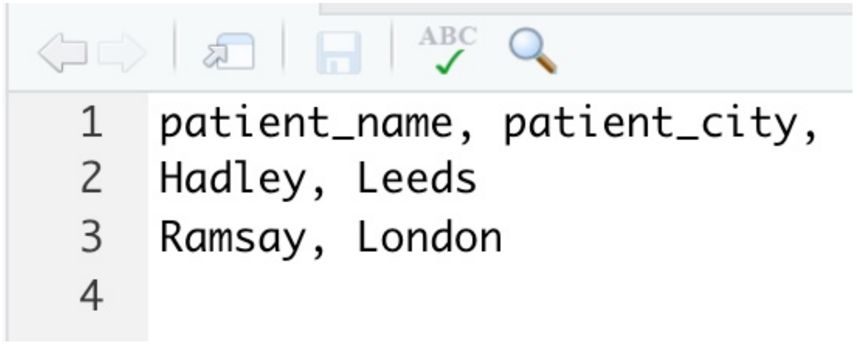
Let's add some sensitive data to the repo (II)

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



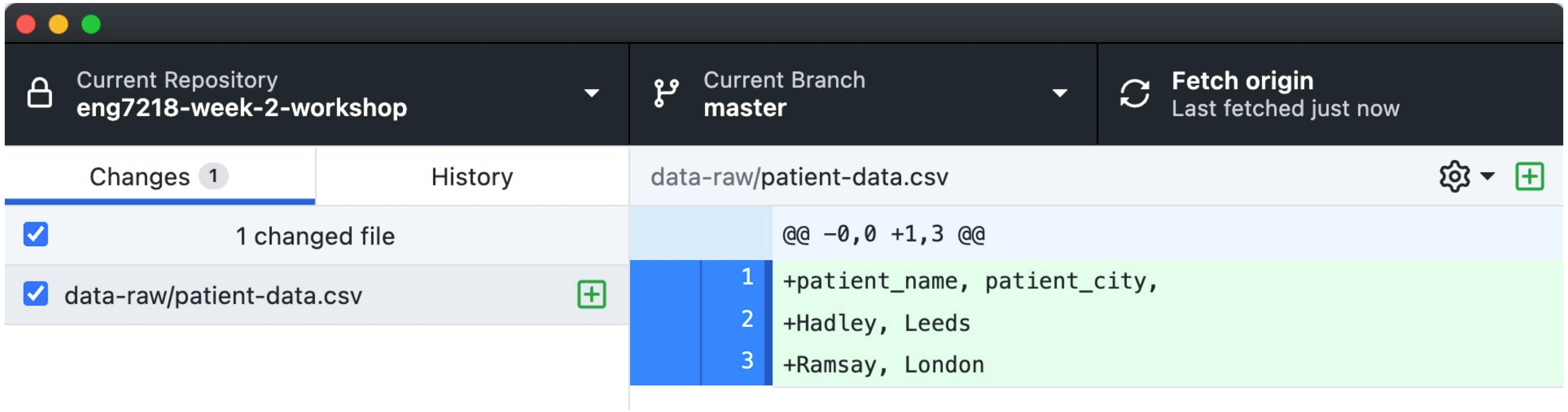
```
patient_name, patient_city,
Hadley, Leeds
Ramsay, London
```

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"



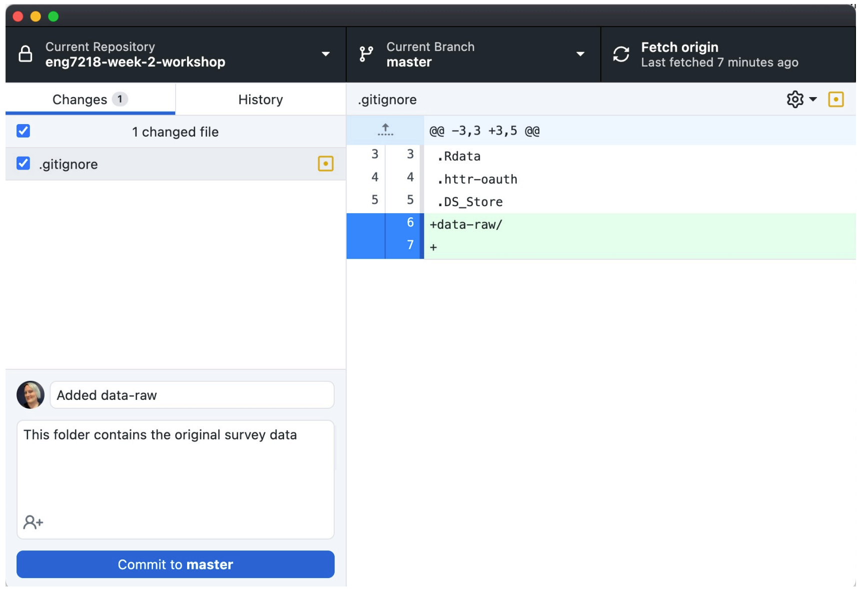
```
@@ -0,0 +1,3 @@
+patient_name, patient_city,
+Hadley, Leeds
+Ramsay, London
```

Commit this change to the repository

1. Switch back to GitHub
2. Add a commit message

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

3. **Optionally** add a summary
4. Click "Commit to master"



```
@@ -3,3 +3,5 @@
 .Rdata
 .httr-oauth
 .DS_Store
+data-raw/
+
```

Added data-raw

This folder contains the original survey data

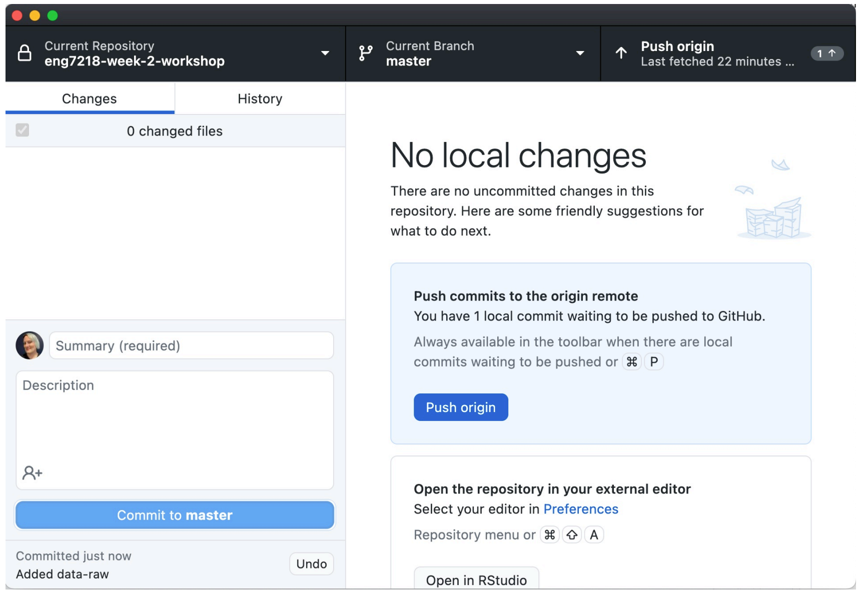
> In `git` when we "commit" something we tell `git` to remember our change in the local repository.

Workshop 22 / 48

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!



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Push commits to the origin remote
You have 1 local commit waiting to be pushed to GitHub.
Always available in the toolbar when there are local commits waiting to be pushed or ⌘ P

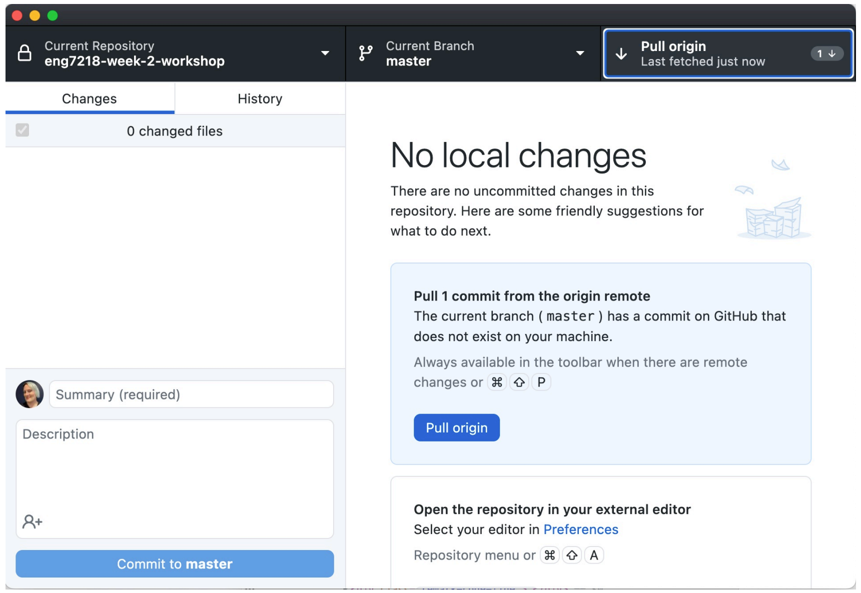
Open the repository in your external editor
Select your editor in Preferences
Repository menu or ⌘ ⇧ A

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.



No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Pull 1 commit from the origin remote
The current branch (`master`) has a commit on GitHub that does not exist on your machine.
Always available in the toolbar when there are remote changes or ⌘ ⇧ P

Open the repository in your external editor
Select your editor in Preferences
Repository menu or ⌘ ⇧ A

GitHub branches and pull requests

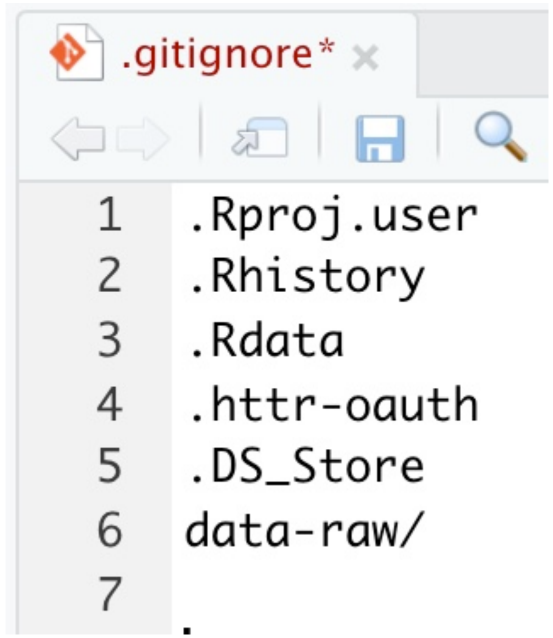
This is as far as we'll go with using `git` for version control.

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.

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**



```
.Rproj.user
.Rhistory
.Rdata
.httr-oauth
.DS_Store
data-raw/
```

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

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

Improving Zotero with Better Bibtex for Zotero

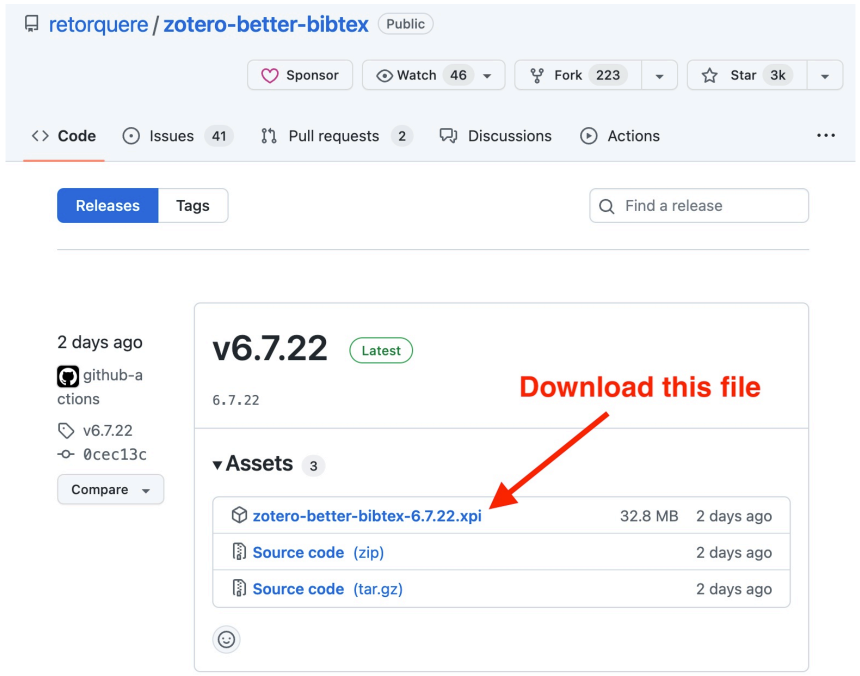
We're going to install the "Better Bibtex" addin which improves how Zotero generates bibtex files.

> 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.

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 days ago
github-actions
v6.7.22
0cec13c

v6.7.22

6.7.22

Assets 3

zotero-better-bibtex-6.7.22.xpi	32.8 MB	2 days ago
Source code (zip)		2 days ago
Source code (tar.gz)		2 days ago

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

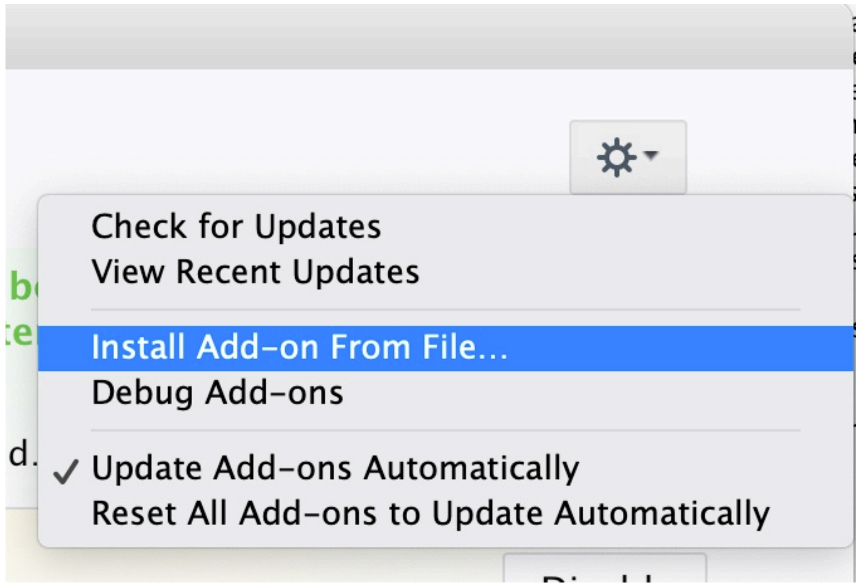
2. Open Zotero and select

   `Tools > Add-ons`

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

4. Select the `.xpi` file you just downloaded

5. Restart Zotero

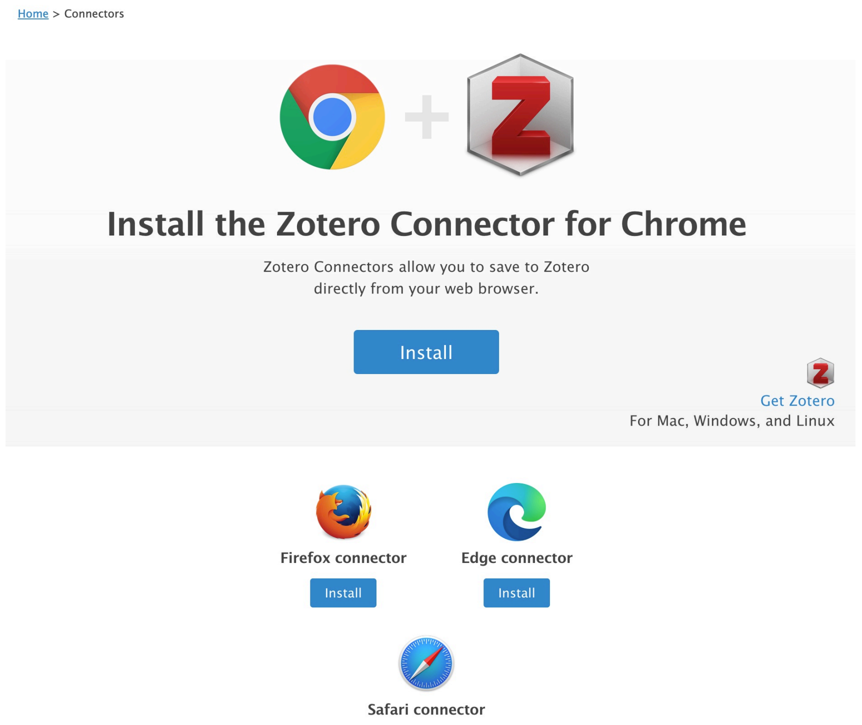

Check for Updates
View Recent Updates
Install Add-on From File...
Debug Add-ons
✓ Update Add-ons Automatically
Reset All Add-ons to Update Automatically

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

Install the Zotero Connector for Chrome

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

For Mac, Windows, and Linux

Adding a paper to Zotero

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

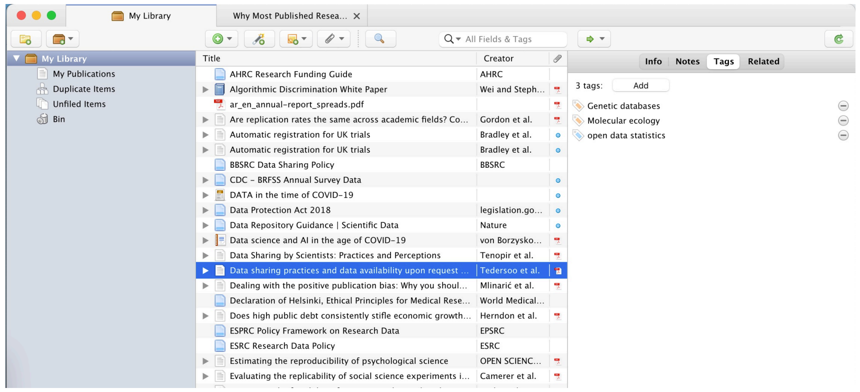
> You need your browser and Zotero open simultaneously.

1. Visit doi.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



Title	Creator
AHRC Research Funding Guide	AHRC
Algorithmic Discrimination White Paper	Wei and Steph...
ar_en_annual-report_spreads.pdf	
Are replication rates the same across academic fields? Co...	Gordon et al.
Automatic registration for UK trials	Bradley et al.
Automatic registration for UK trials	Bradley et al.
BBSRC Data Sharing Policy	BBSRC
CDC - BRFSS Annual Survey Data	
DATA in the time of COVID-19	
Data Protection Act 2018	legislation.go...
Data Repository Guidance | Scientific Data	Nature
Data science and AI in the age of COVID-19	von Borzysko...
Data Sharing by Scientists: Practices and Perceptions	Tenopir et al.
Data sharing practices and data availability upon request ...	Tedersoo et al.
Dealing with the positive publication bias: Why you shoul...	Mlinarić et al.
Declaration of Helsinki, Ethical Principles for Medical Rese...	World Medical...
Does high public debt consistently stifle economic growth...	Herndon et al.
EPSRC Policy Framework on Research Data	EPSRC
ESRC Research Data Policy	ESRC
Estimating the reproducibility of psychological science	OPEN SCIENC...
Evaluating the replicability of social science experiments i...	Camerer et al.

3 tags:

- Genetic databases
- Molecular ecology
- open data statistics

Connecting this to an RStudio project

Our goal is to setup 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

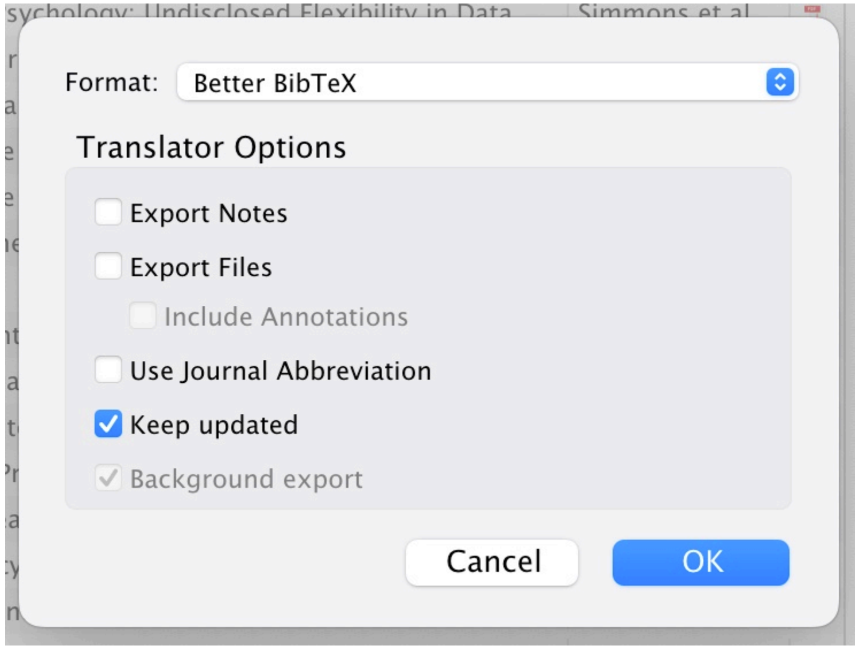
1. Add an .Rmd document to your RStudio project called something like `Report.Rmd`.

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

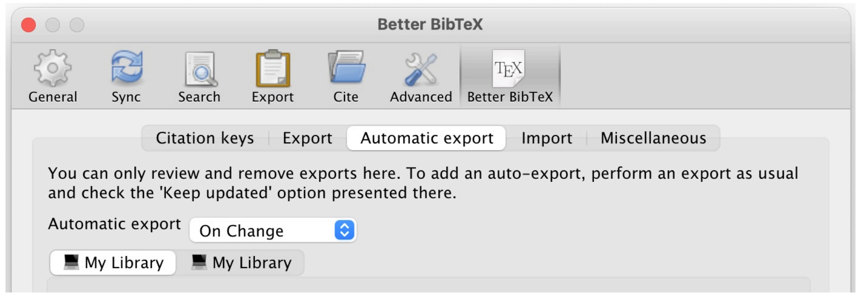
Translator Options

- [ ] Export Notes
- [ ] Export Files
  - [ ] Include Annotations
- [ ] Use Journal Abbreviation
- [✓] Keep updated
- [✓] Background export

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`
4. Open Zotero preferences and go to the BetterBibTeX section

- Choose the "Automatic Export" tab
- Set "Automatic export" to "On Change"



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

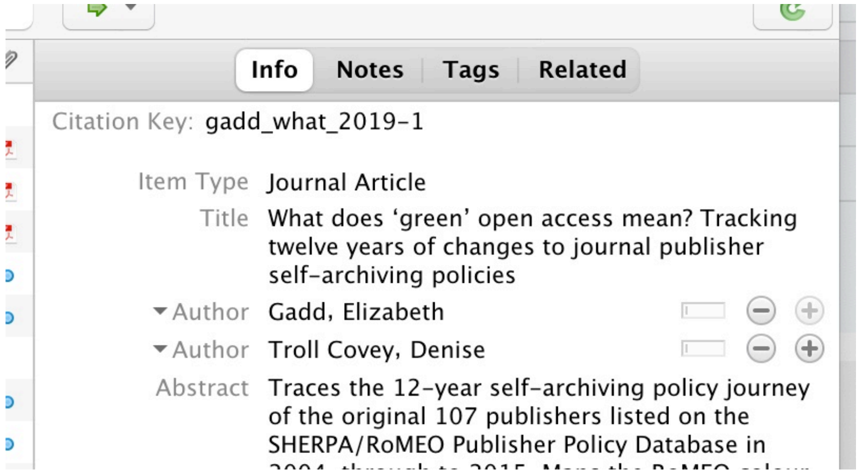
1. Add an .Rmd document to your RStudio project called something like `Report.Rmd`.
2. Open Zotero and go to `File > Export library`
3. Save the file into your RStudio project and call it `references.bib`
4. Save the file into your RStudio project and call it `references.bib`
5. Open Zotero preferences 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 paper.

# 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-readable data.



Citation Key: gadd_what_2019-1

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 40 references.

Zotero and MS Word

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

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

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

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 webpages/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 "accessed date"

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
  - Keep sensitive data off GitHub with the .gitignore file
- Connect your Figshare and GitHub repos to make your code citable and avaialble 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.

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.