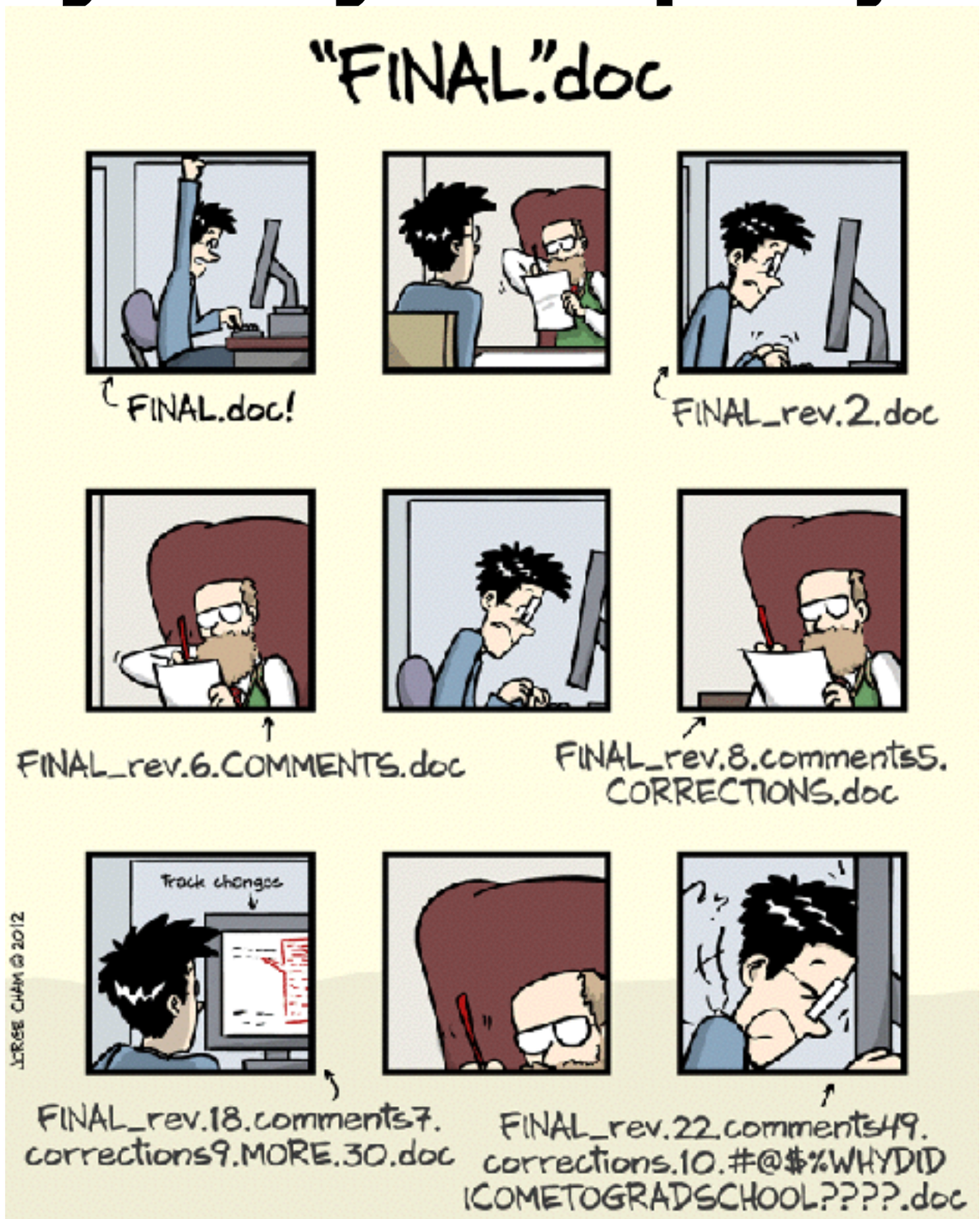
If you have not already done so, please download and install …

- **R 3.6.0**
  https://cran.r-project.org/
- **Rstudio Desktop**
  https://www.rstudio.com/products/rstudio/download/
- **git**
  https://git-scm.com/downloads

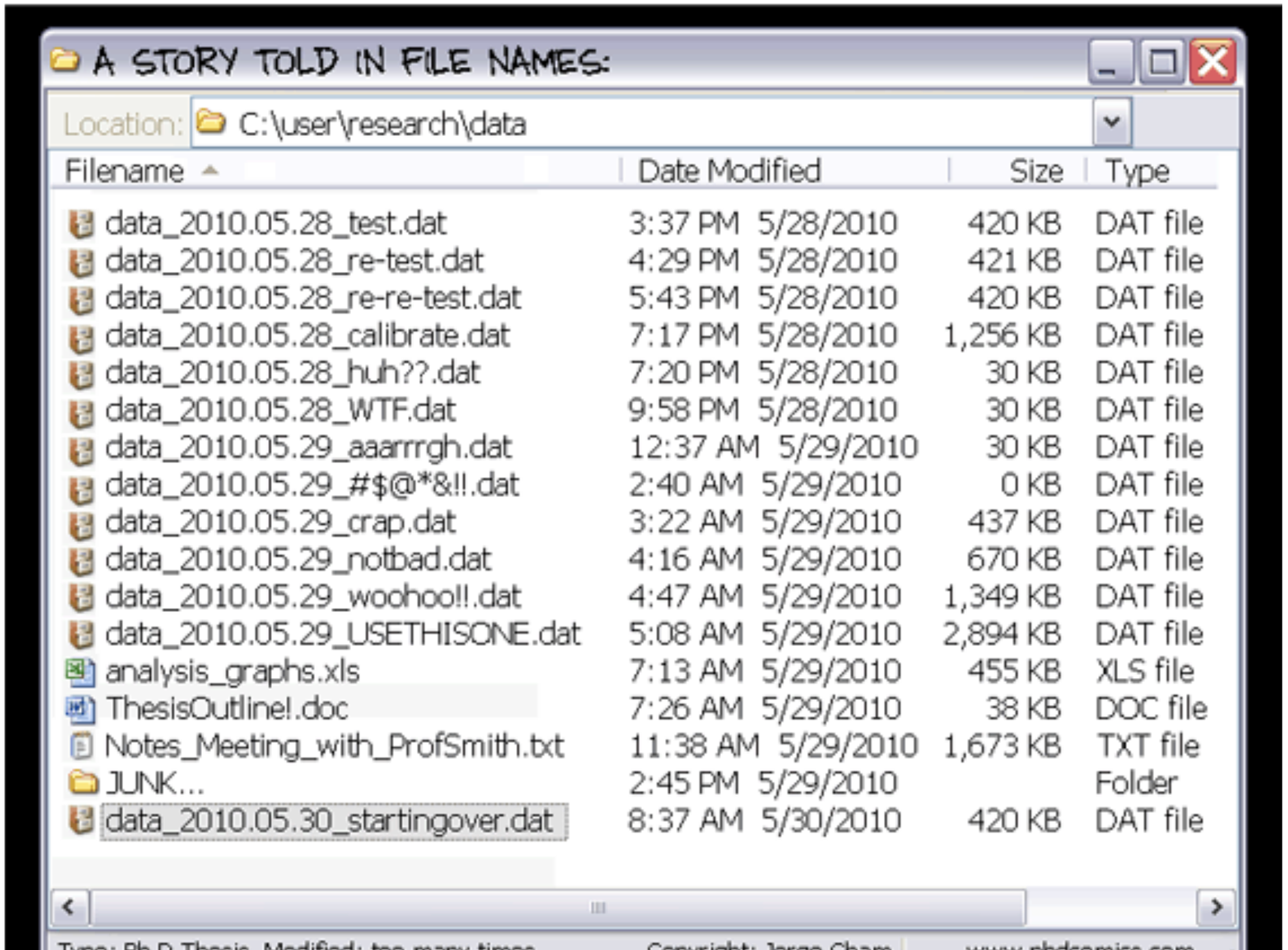
Tools for reproducible research: git/github, Rstudio, markdown

3D Morphometrics and Image Analysis Workshop
Aug 24, 2019

How do you record the history of your projects?



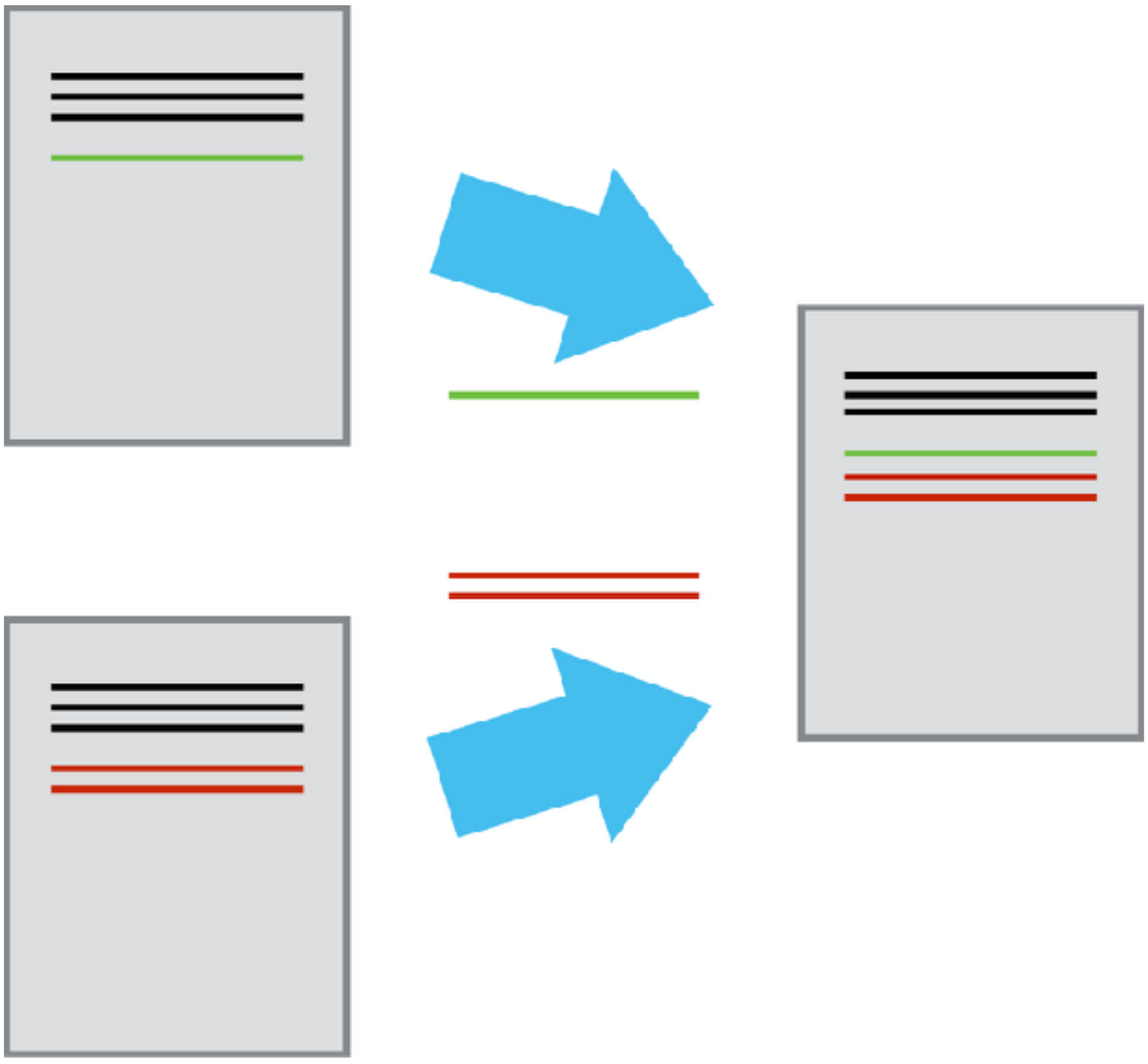
How do you record the history of your projects?



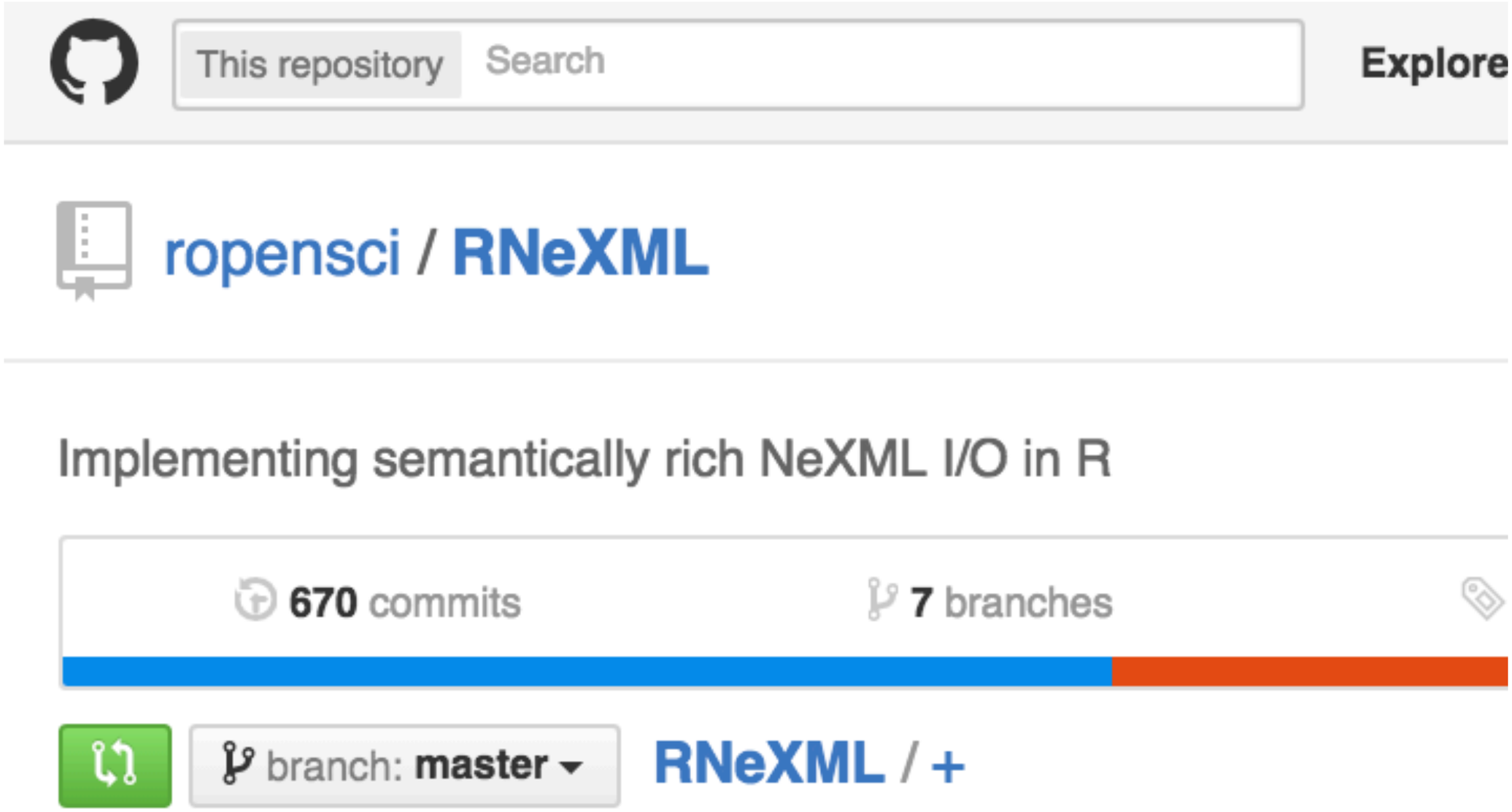
Best practice is version control

Version control starts with a base version of the document, and it saves just the changes you made at each step of the way.

Version control gives more flexibility



Version control with Git and Github



Git and Github in a larger community

Exposure: If someone needs to see your work or if you want them to try out your code, they can easily get it from GitHub. If they use Git, they can clone or fork your repository. If they don't use Git, they can still browse your project on GitHub like a normal website and even grab everything by downloading a zip archive.
Be a keener! If you care deeply about someone else's project, such as an R package you use heavily, you can track its development on GitHub. You can watch the repository to get notified of major activity. You can fork it to keep your own copy. You can modify your fork to add features or fix bugs and send them back to the owner as a proposed change.
Collaboration: If you need to collaborate on data analysis or code development, then everyone should use Git. Use GitHub as your clearinghouse: individuals work independently, then send work back to GitHub for reconciliation and transmission to the rest of the team. The advantage of Git/GitHub is highlighted by comparing these two ways of collaborating on a document:

- **Edit, save, attach.** In this workflow, everyone has one (or more!) copies of the document and they circulate via email attachment. Which one is "master"? Is it even possible to say? How do different versions relate to each other? How should versions be reconciled? If you want to see the current best version, how do you get it? All of this usually gets sorted out by social contract and a fairly manual process.
- **Google Doc.** In this workflow, there is only one copy of the document and it lives in the cloud. Anyone can access the most recent version on demand. Anyone can edit or comment or propose a change and this is immediately available to everyone else. Anyone can see who's been editing the document and, if disaster strikes, can revert to a previous version. A great deal of ambiguity and annoying reconciliation work has been designed away.

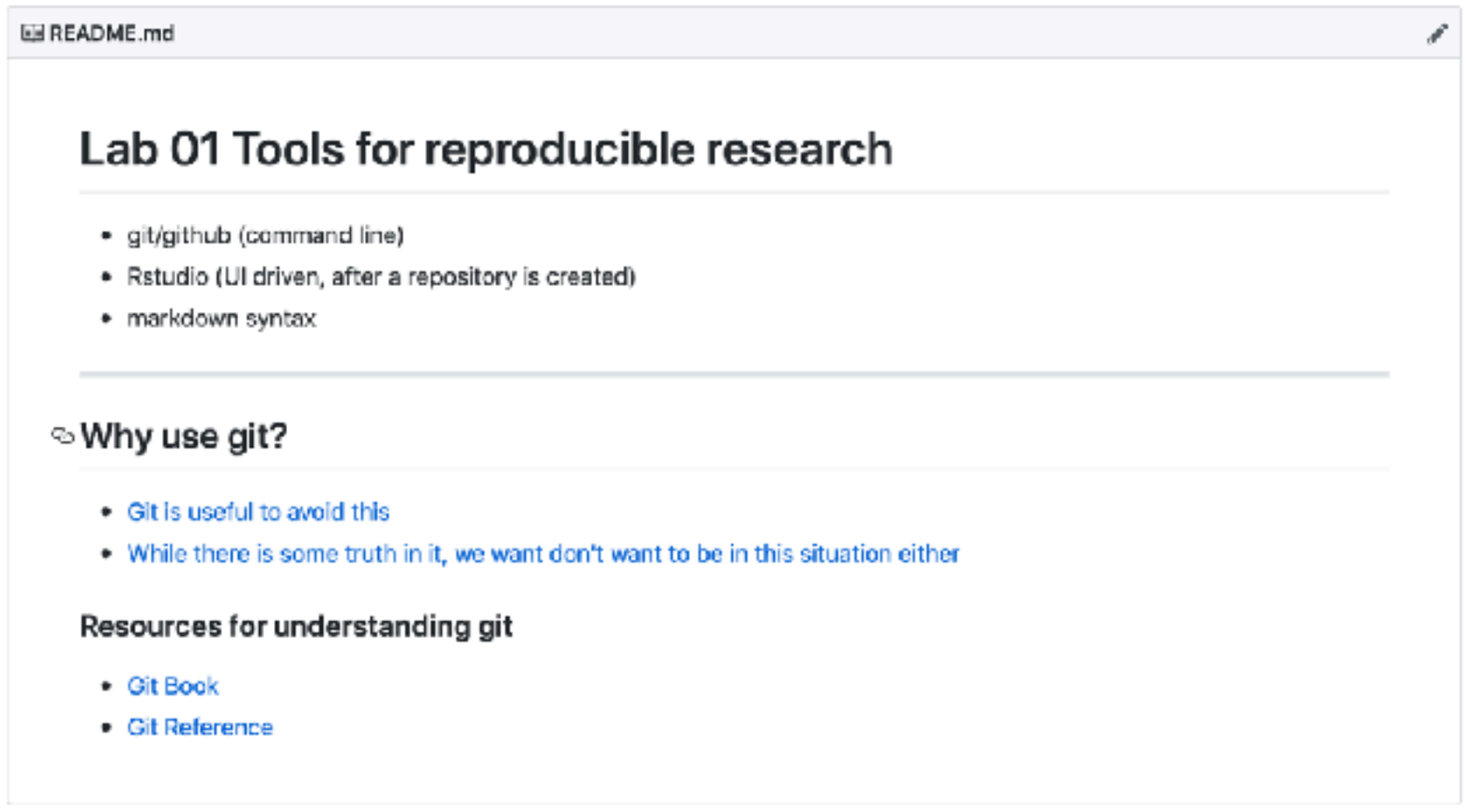
Managing a project via Git/GitHub is much more like the Google Doc scenario and enjoys many of the same advantages. It is definitely more complicated than collaborating on a Google Doc, but this puts you in the right mindset.

In this lab, we will

- Get git working in Rstudio
- Pull from SlicerMorphWorkshop
- Create your own repository and make a push
- Markdown

Markdown is a way to style text on web

Our webpage is written via Markdown



README.md

Lab 01 Tools for reproducible research

- git/github (command line)
- Rstudio (UI driven, after a repository is created)
- markdown syntax

Why use git?

- Git is useful to avoid this
- While there is some truth in it, we want don't want to be in this situation either

Resources for understanding git

- Git Book
- Git Reference

Mostly, Markdown is just a regular text with a few non-alphabetic characters thrown in like # or *.

Markdown examples

Text: ** bold, * italic, link

```
It's very easy to make some words **bold** and other words
*italic* with Markdown. You can even
[link to Google!](http://google.com)
```

It's very easy to make some words **bold** and other words *italic* with Markdown. You can even link to Google!

Markdown examples

List, bullet point, dashes

```
Sometimes you want numbered lists:

1. One
2. Two
3. Three

Sometimes you want bullet points:

* Start a line with a star
* Profit!

Alternatively,

- Dashes work just as well
- And if you have sub points, put two spaces before the dash or
star:
  - Like this
  - And this
```

Sometimes you want numbered lists:

1. One
2. Two
3. Three

Sometimes you want bullet points:

- Start a line with a star
- Profit!

Alternatively,

- Dashes work just as well
- And if you have sub points, put two spaces before t
  - Like this
  - And this

Markdown examples

Images:

```
If you want to embed images, this is how you do it:

![Image of Yaktocat]
(https://octodex.github.com/images/yaktocat.png)
```

If you want to embed images, this is how you do it:

Markdown examples

Headers ## and quotes >:

```
# Structured documents

Sometimes it's useful to have different levels of headings to
structure your documents. Start lines with a `#` to create
headings. Multiple `##` in a row denote smaller heading sizes.

### This is a third-tier heading

You can use one `#` all the way up to `######` six for
different heading sizes.

If you'd like to quote someone, use the > character before the
line:

> Coffee. The finest organic suspension ever devised... I beat
the Borg with it.
> - Captain Janeway
```

Markdown examples

Headers ## and quotes >:

Structured documents

Sometimes it's useful to have different levels of headings to structure your documents. Start lines with a `#` to create headings. Multiple `##` in a row denote smaller heading sizes.

This is a third-tier heading

You can use one `#` all the way up to `######` six for different heading sizes.

If you'd like to quote someone, use the > character before the line:

> Coffee. The finest organic suspension ever devised... I beat the Borg with it. - Captain Janeway

Markdown examples

Task list: [x] and emoji :EMOJICODE:

```
GitHub supports many extras in Markdown that help you reference
and link to people. If you ever want to direct a comment at
someone, you can prefix their name with an @ symbol: Hey
@kneath — love your sweater!

But I have to admit, tasks lists are my favorite:

- [x] This is a complete item
- [ ] This is an incomplete item

When you include a task list in the first comment of an Issue,
you will see a helpful progress bar in your list of issues. It
works in Pull Requests, too!

And, of course emoji!
```

Resources

https://guides.github.com/features/mastering-markdown/
https://git-scm.com/book/en/v2
https://git-scm.com/docs