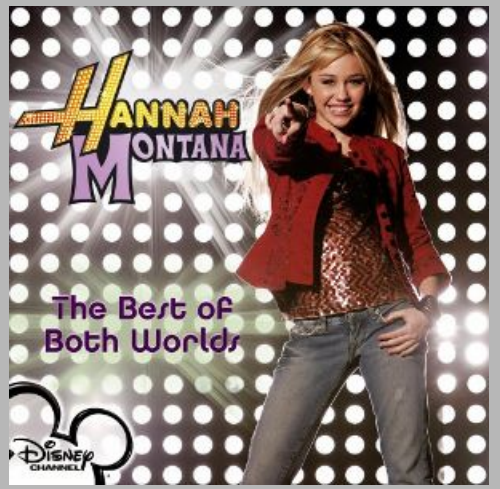
Collaborative writing workflows: building blocks towards reproducibility

Sara Stoudt
@sastoudt
sas072@bucknell.edu

Let’s dream for a minute...

What would an ideal tool for collaboratively writing a reproducible paper look like?

Really, we want Google Docs but for RMarkdown. (and that exists, sort of)

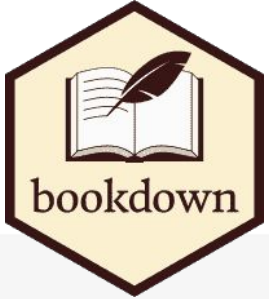


We also need to acknowledge typical student workflows.

https://fineartamerica.com/featured/i-know-how-to-divide-but-want-to-learn-how-to-divide-aaron-bacall.html?product=canvas-print

Version Control... but avoid burn-it-down moments.

- Break up the final project report into stand-alone pieces that are stored in separate RMarkdown files.
- Then different members of a team can work on different sections at the same time.



report.Rmd

```
---
title: My Report
output:
  pdf_document:
    toc: yes
---

```{r child = 'chapter1.Rmd'}
```

```{r child = 'chapter2.Rmd'}
```
```

chapter1.Rmd

```
# Chapter 1

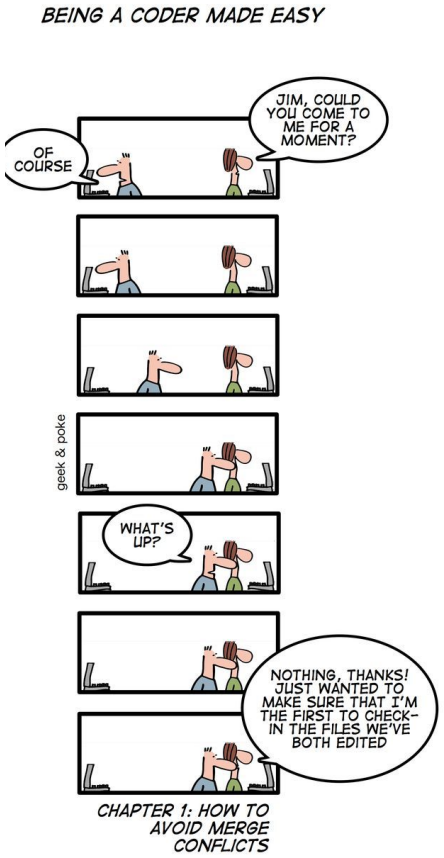
This is chapter 1.

```{r}
1
```
```

chapter2.Rmd

```
# Chapter 2

This is chapter 2.

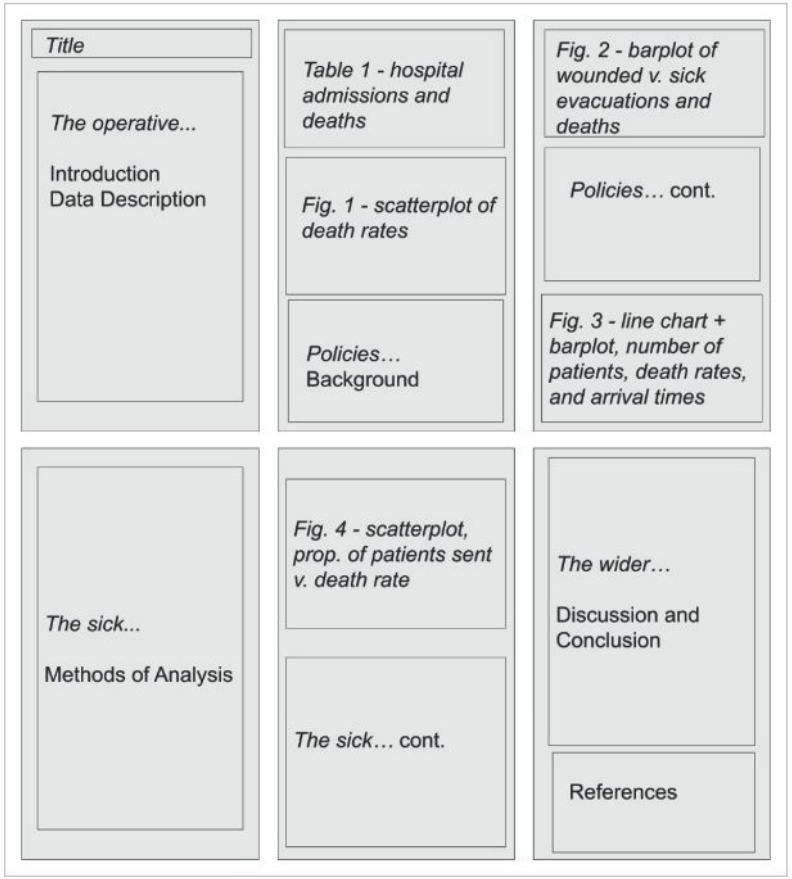
```{r}
2
```
```

https://stackoverflow.com/questions/25824795/how-to-combine-two-rmarkdown-rmd-files-into-a-single-output



https://geekandpoke.typepad.com/geekandpoke/2010/10/being-a-code-made-easy-chapter-1.html

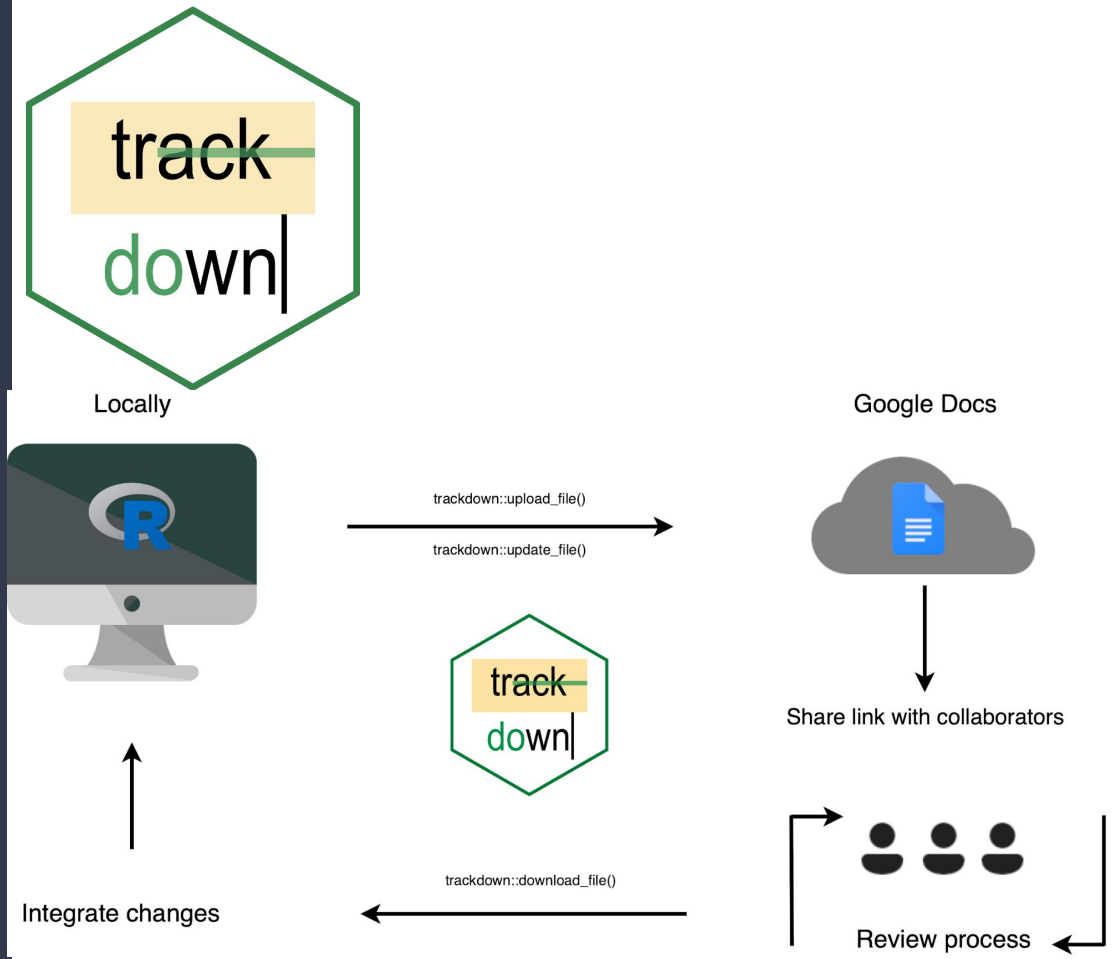
Added Benefit - an opportunity to talk about report structure



https://rss.onlinelibrary.wiley.com/doi/full/10.1111/1740-9713.01469

Version-Control-Free Compromise

- One person codes (perhaps in a group synchronously with one person driving).
- “Pushes” draft and preliminary output to Google Docs.
- Others write collaboratively.
- One person “pulls” text back to RStudio to render complete draft.



https://claudiozandonella.github.io/trackdown/articles/trackdown-workflow.html

Case Study - trackdown

Senior “Culminating Experience” Course

- 7 students who I already had before in another class where we learned some R basics
- bi-weekly R labs in pairs/groups of 3 (topic: statistical designs for scientific studies)

Initial Win

I followed the set-up instructions, and they worked perfectly.

Initial Pain Point

Only one person can “push” and “pull”, so effectively only one person could do the coding.

Detour: What do these approaches have in common?

Meta-communication is necessary.

We need to communicate in order to communicate.

Streamlined Version Control

- Who is working on which sections at what time?
- Establish checkpoints where they may be reviewing one another's work or switching responsibilities.
- Track via GitHub issues and pull requests.

Version-Control Free

- "Check-in" and "check-out" document: when code is ready to be written about, when text is ready to be re-incorporated into the R Markdown document
- Who is "driving" and how is code getting into the initial document?
- Create a new document (likely Google Doc) to track this information.

Case Study Redux

How I envisioned the workflow:

- Initial in-class work time (50 minutes)
  - Gloss the code for each question together with one person “driving”
  - Leave the interpretation/writing for later
  - “Push” code and output to Google Drive at the end of class
- Asynchronous divide and conquer outside of class on writing

Or

- Synchronous meet up to continue working on coding part

==

- “Driver” knits final document and passes it around for final review

How my students ended up working:

- Immediately divide-and-conquer questions
- E-mail/Slack “driver” the code for each piece
- use some of the writing features of Google Docs

Pain Points:

- Some teams completely gave up on trackdown
- Some teams submitted things that weren’t “up to date” by accident due to miscommunication

Smoothing Steps - Scaffold Revisions

- Peer review sections not in charge of.
- Revise to ensure consistency in style and tone of both writing, code, and formatting of output (including graphs and tables) across sections.
- Revise to strengthen transitions in the writing between sections
- Revise to remove redundancies across sections.
- Time must be left for these steps!

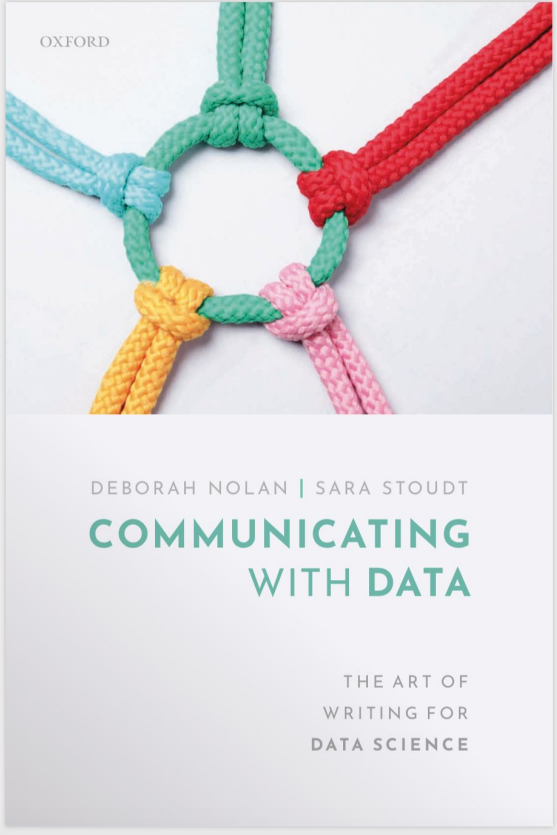
Zoom Out – A Reproducible Writing Process

- Everyone needs to agree on a standard data file, and if stored locally could get changed by individuals, causing problems.
- Streamlined-version-control compromise: one person's code changes in their section could affect another's section outputs (e.g. rename a column of the data)
- Version-control-free compromise: check-in/check-out miscommunications can cause frustration (e.g. writing about stale results)

Thank you! Any questions?

sas072@bucknell.edu

@sastoudt



https://datasciencebydesign.org/book-2
https://global.oup.com/academic/product/communicating-with-data-9780198862758
https://mathstatbites.org/

MathStat Bites