13: Summary + Extras

bit.ly/2018rr

The most important tool is the mindset,
when starting, that the end product
will be reproducible.

– Keith Baggerly

Steps toward reproducible research

- Slow down
- Organize; document
- Everything with code
- Scripts → RMarkdown
- Code → functions → packages
- Version control with Git
- Automation with Make
- Choose a license
- Share your work with others

Organize your data

	A	B	C	D	E	F	G
1	1MIN						
2			Normal			Mutant	
3	B6	146.6	138.6	155.6	166	179.3	186.9
4	BTBR	245.7	240	243.1	177.8	171.6	188.1
5							
6	5MIN						
7			Normal			Mutant	
8	B6	333.6	353.6	408.8	450.6	474.4	423.8
9	BTBR	514.4	610.6	597.9	412.1	447.4	446.5

Organize your data

	A	B	C	D	E	F	G
1							
2	Date	11/3/14					
3	Days on diet	126					
4	Mouse #	43					
5	sex	f					
6	experiment		values			mean	SD
7	control		0.186	0.191	1.081	0.49	0.52
8	treatment A		7.414	1.468	2.254	3.71	3.23
9	treatment B		9.811	9.259	11.296	10.12	1.05
10							
11	fold change		values			mean	SD
12	treatment A		15.26	3.02	4.64	7.64	6.65
13	treatment B		20.19	19.05	23.24	20.83	2.17

Organize your data

	A	B	C	D	E	F
1	mouse_id	sex	week	date	glucose	weight
2	3005	M	4	3/30/2007	19.3	635
3	3005	M	6	4/11/2007	31	460.7
4	3005	M	8	4/27/2007	39.6	530.2
5	3017	M	4	10/6/2006	25.9	202.4
6	3017	M	6	10/19/2006	45.1	384.7
7	3017	M	8	11/3/2006	57.2	458.7
8	3434	F	4	11/22/2006	26.6	238.9
9	3434	F	6	12/6/2006	45.9	378
10	3434	F	8	12/22/2006	56.2	409.8
11	3449	M	4	1/5/2007	27.5	121
12	3449	M	6	1/19/2007	42.9	191.3
13	3449	M	8	2/2/2007	56.7	182.5
14	3499	F	4	1/5/2007	19.8	220.2
15	3499	F	6	1/19/2007	36.6	556.9
16	3499	F	8	2/2/2007	43.6	446

Organize your data

kbroman.org/dataorg

Broman & Woo (2018) Data organization in spreadsheets. Am Stat 78:2–10

doi:10.1080/00031305.2017.1375989

Challenges

- Daily maintenance
  - READMEs up to date?
  - Documentation matches code?
- Cleaning up the junk
  - Move defunct stuff into an `Old/` subdirectory?
- Start over from the beginning, nicely?

Sharing your work

- Why share?
  - Funding agency or journal requirement
  - Increased visibility
  - So that others can build on your work
- When?
  - Continuously and instantaneously
  - When you submit a paper
  - When your paper appears
- Risks?

bit.ly/rr_sharing_slides

I’m not worried about being scooped,
I’m worried about being ignored.

– Magnus Nordborg

- Share more
- Share sooner
- Share in a way that makes it easy for others to learn from and build upon

What to share?

- For sure
  - Primary dataset
  - Metadata
  - Data cleaning scripts
  - Analysis scripts
- It could help
  - Very-raw data
  - Processed/clean data
  - Intermediate results
- No
  - Confidential data (e.g. HIPAA data)
  - Passwords, private keys

Where to share?

- Domain-specific repository
  - Genbank, dbGaP, etc.
  - See `re3data.org`
- Figshare, Dryad, Zenodo
- Institutional repository
- GitHub, BitBucket

Resources

bit.ly/2018rr_resources

Some of the things we didn’t cover

Coding conventions

Why are they cool?

- They help you keep things consistent between team members
- They make code easier to read, and more likely to be used

Why didn't we cover them?

- Not enough time

Where would we point you?

- Hadley's recommendations `adv-r.had.co.nz/Style.html`
- Google's recommendations `google.github.io/styleguide/Rguide.xml`
- Tidyverse style guide `style.tidyverse.org`

Code review

Why is it cool?

- Helps to find bugs and clean up confusing bits
- Potentially a test of the reproducibility of your work

Why didn't we cover it?

- Not enough time

Where would we point you?

- Software Carpentry blog post, `bit.ly/swc_codereview`
- Titus Brown's blog post, `http://bit.ly/titus_codereview`

Software testing

Why is it cool?

- Explicit tests help you to avoid bugs, and to find bugs sooner

Why didn't we cover it?

- Not enough time

Where would we point you?

- testthat package, `github.com/hadley/testthat`
- Testing R Code book

Continuous integration (eg Travis)

Why is it cool?

- Automatically build and run tests when you push to GitHub
- Pull requests are automatically tested

Why didn't we cover it?

- Not enough time

Where would we point you?

- Julia Silge blog post, `juliasilge.com/blog/beginners-guide-to-travis`
- Hadley's R packages book, `r-pkgs.had.co.nz/check.html#travis`

Capturing dependencies

Why is it cool?

- Ensure that your carefully constructed reproducible project doesn't fail due to a change in one of the packages you use

Why didn't we cover it?

- Not enough time

Where would we point you?

- packrat package, `github.com/rstudio/packrat`
- checkpoint package, `github.com/RevolutionAnalytics/checkpoint`

Containers (e.g. docker)

Why are they cool?

- Capture your entire environment, so your project is *for sure* fully reproducible.

Why didn't we cover them?

- A bit technical

Where would we point you?

- Rocker: Docker for R
- R Docker tutorial

R Markdown templates

Why are they cool?

- More complete control over the appearance of your document

Why didn't we cover them?

- A bit technical

Where would we point you?

- R Markdown documentation

knitr Bootstrap

Why is it cool?

- Allows for generation of slicker reports

Why didn't we cover it?

- A bit technical

Where would we point you?

- `github.com/jimhester/knitrBootstrap`

GitHub pages

Why are they cool?

- Webpages built entirely in Markdown, providing nicer interfaces to your content

Why didn't we cover them?

- Tangential to *reproducible research*?

Where would we point you?

- `pages.github.com`
- `kbroman.org/simple_site`
- `bookdown.org/yihui/blogdown`

Bookdown

Why is it cool?

- Write a book (or book-like object) entirely in R Markdown

Why didn't we cover it?

- Not enough time

Where would we point you?

- `bookdown.org/yihui/bookdown`

workflowr

Why is it cool?

- R package to help generate a website with time-stamped, versioned reports of analyses.

Why didn't we cover it?

- Not enough time

Where would we point you?

- `jdblischak.github.io/workflowr`

Xaringan

Why is it cool?

- Use R Markdown to make slides for a talk

Why didn't we cover it?

- Not enough time

Where would we point you?

- `github.com/yihui/xaringan`

Shiny!

Why is it cool?

- Interactive pictures have pizzazz.

Why didn't we cover it?

- Tangential to *reproducible research*?

Where would we point you?

- `shiny.rstudio.com`
- `shiny.rstudio.com/tutorial`

Feedback we'd like from you (1)

What motivated us to teach this course?
What would we see as a positive outcome?

- Given this motivation, are we doing things right?
- What motivated you to take this course?
- Were there specific sessions you found really useful/really useless?
- Points you'd like us to expand on?
- Were there points you were hoping we'd cover that we didn't?

Feedback we'd like from you (2)

- Do you have examples/anecdotes you think we might be able to use that you'd be willing to share?
- Were there ways we could've used time more effectively?
- Can you see things you learned in this course changing how you do things day to day?
  - Why or why not?
  - Can we ask you again in 6 months?
  - Can we ask you again in a year?
- Could you write this down now? (anonymous is fine)