Reproducible research with R and friends

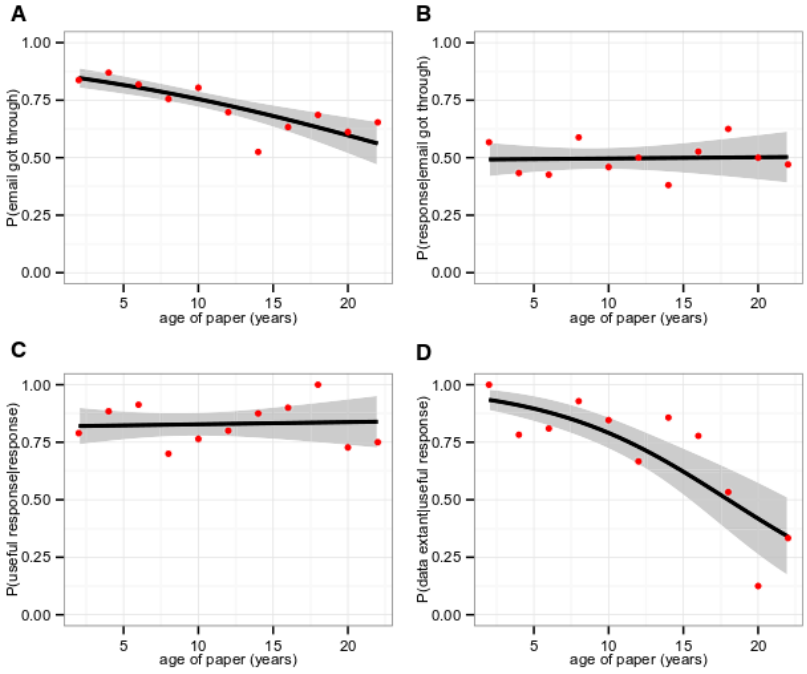
Ben Bolker

00:25 02 July 2015

The big picture

reproducibility: why now?

- the reproducibility crisis (?)
- growth of meta-science
- more complex analyses
- more collaboration
- more network/storage availability
- dark data and data rot



(Vines et al. 2014)

why you?

- ethical obligation/good science
- re-use = visibility
- personal sanity: past-you and future-you
- shiny toys

your workflow

metadata

- data describing your data: locations, species names, etc.
- not well supported in R
- conflict between simplicity/portability/convenience and metadata maintenance
- EML package?
- revision control systems for metadata about *changes*

workflow tips

- batch vs interactive processing
- DRY (don't repeat yourself) – functions should be in one place, re-usable/re-used
- organic process:
  - experiments in console window
  - rough code in main script
  - code → functions in main script
  - functions → separate file
  - functions → package
- batch runs

future-proofing

- package/R version:
  - print `sessionInfo()` at end of output
  - `checkpoint`, `packrat` packages

data handling

- process is most important, but tools are important too
- host/archive data online/centrally
- private repository while developing, open on publication
- changes to primary data should be versioned and stored

file formats

- data formats:
  - **AVOID EXCEL**: date-mangling, binary format, undiff-able
  - CSV: low-tech but most portable etc. (careful of quotation marks)

  - RData (`rda` or `rds`): fast, compact, flexible, but not human-readable or useful outside the R ecosystem (store as intermediate product)
- keep data in human-readable, convenient, *consistent* structure
  - e.g. consistent date/time formats, coding
- locales/encodings

directory structure

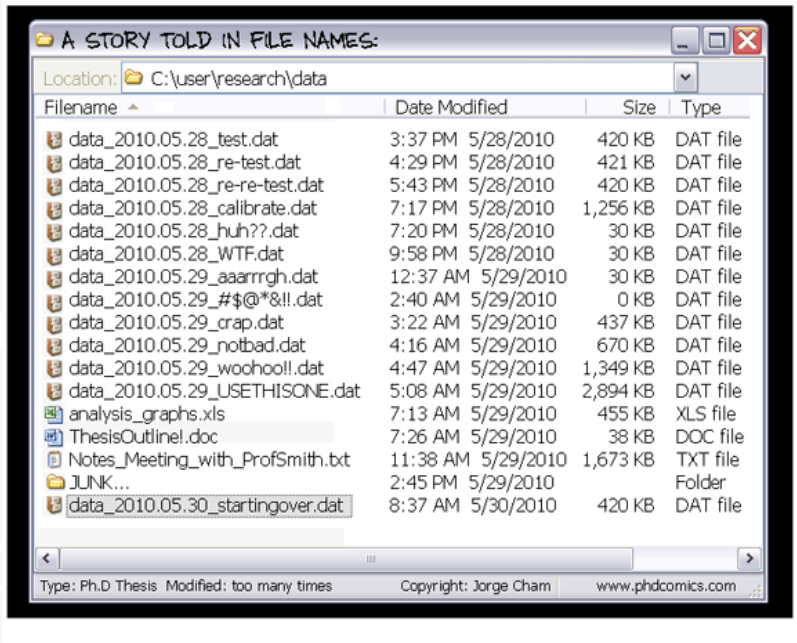
- your code should not contain any *absolute file paths* (e.g. `C:\Users\Joe` or `/home/users/joe`)
- other users should be able to mimic your setup exactly
- you can include a commented-out `setwd()` command in the code for your own reference
- use `Set working directory to source file location` in RStudio
- maybe subdirectories for data etc. or sub-projects, if complex

using the command line

- `Terminal` on MacOS, Start Menu/`cmd` on Windows (Cygwin for more features)
- `R CMD BATCH`
- change directory with `cd` (Unix), `chdir` (Windows)

Revision control systems

Revision control



PhD comics

- archive materials (code and data) securely
- track changes
- view changes
- roll back changes
- document changes

xkcd

Revising data

- make only "permanent" changes to data files (recorded on archives)
- temporary changes only within R script
  - recoding
  - subsetting

Git

- revision control system
- materials accessible both off- and online
- enables collaborative work
- can set up local server ...

Github and Bitbucket

- free, public storage
- private repositories available on BB, for students and academics on GH
- web front end
- development tools (issue tracker, wiki, etc etc)

Data repositories

- data publishing, with or without a journal article
- GH/BB much better than nothing, but *not* archival ...
- Git repositories can be copied to other servers (you always have a nearly up-to-date version locally)
- Dryad
- Ecological Archives
- `http://oad.simmons.edu/oadwiki/Data_repositories`

Github via RStudio

- setting up RStudio to work with GH
- starting a new GH repo/project

Rmarkdown

history

- literate programming (Knuth 1992)
- ancestor of RR
- similar tools, different scope

- software development: code as documentation
- CWEB → Sweave → knitr

Markdown basics

- formatting; italic, bold, bulleted lists, section headings
- math via included LaTeX, e.g. `$\sqrt{x^2 + y^2}$` = $\sqrt{x^2 + y^2}$
- tables
- bibliographic citations

RMarkdown basics

- embedded, highlighted code chunks:

  ```{r mychunk}

  and end with triple back-quote

- figures embedded automatically
- code chunk caching
- inline expressions: `` `r foo+bar` ``

*Rmarkdown: code chunk options*

- Set per chunk, e.g. `{r mychunk,echo=TRUE,eval=FALSE}` or globally via `opts_chunk$set(...)`
  - `eval`: evaluate?
  - `echo`: show code?
  - `warning/message/error`: show/stop?
  - `results`: `"markup"` is default, I sometimes use `"hide"`
  - `tidy`: reformat code?
  - `cache`: cache results?
- Use caching sparingly; big runs should be separated into a batch file

*Rmarkdown tips: figures*

- figure size adjustment
- graphics formats: bitmap (PNG) vs vector (PDF)
- `useRaster` for big images

*Output formats*

- PDF
- HTML
- docx
  ```

Collaboration

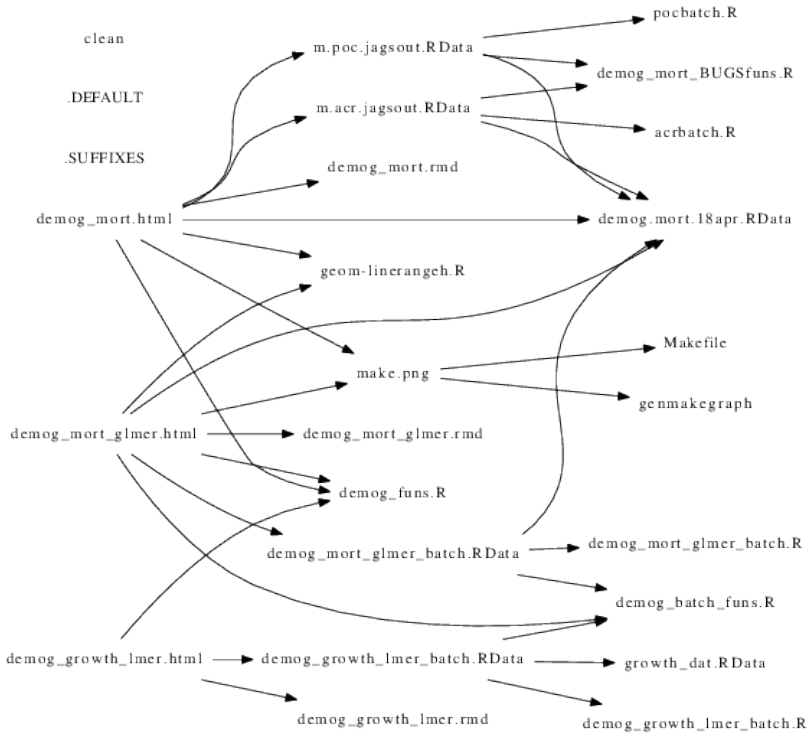
- it's just hard
- e-mailing back and forth
- platforms: Dropbox, Google drive/docs, Github/Bitbucket
- markup: Word, google doc, PDF/Acrobat, everything else

Workflow automation

- `make`

```
demog_mort_glmer.html: demog_mort_glmer.rmd demog_funs.R demog_batch_funs.R demog.mort.18apr.RData demog_m
    Rscript -e "library('rmarkdown'); render('demog_mort_glmer.rmd')"

demog_mort_glmer_batch.RData: demog_mort_glmer_batch.R demog_batch_funs.R demog.mort.18apr.RData
    R CMD BATCH --vanilla demog_mort_glmer_batch.R
```



- remake

Exercises

- set up an `Rmd` document

Resources

- Software Carpentry
- ROpenSci
- Rich Fitzjohn on RR
  - blog post
  - github
- notes from Jenny Bryan? e.g. `https://stat545-ubc.github.io/automation00_index.html`

References

Knuth, D. E. 1992. *Literate Programming*. Center for the Study of Language; Information.

Vines, Timothy H., Arianne Y.K. Albert, Rose L. Andrew, Florence Débarre, Dan G. Bock, Michelle T. Franklin, Kimberly J. Gilbert, Jean-Sébastien Moore, Sébastien Renaut, and Diana J. Rennison. 2014. "The Availability of Research Data Declines Rapidly with Article Age." *Current Biology* 24 (1): 94–97. doi:10.1016/j.cub.2013.11.014.