Computational Reproducibility

Daniel S. Katz
Jennifer Freeman Smith
Computational Reproducibility

- Depending on your field also known as: narrow replicability, pure replicability, analytical replicability, reproducibility

- If I took your original data and your original software and analysis code/scripts/pipeline, could I reproduce all the numbers, figures, tables, etc. in your report?
Computational Reproducibility

- Exactly what is being reproduced will vary across fields, e.g.
  - Data Science
    - An analysis that was done on an existing dataset
    - Do you get the same parameter estimates?
  - Computational Science
    - Simulations that were run to generate data/model/method
    - Do you get the same data/model/method?
    - Does running the model/method give the same results?
How hard can it be...

- **Quarterly Journal of Political Science**
  - 24 computational reproducibility checks 2012 - 2014
    - Only 4 perfect packages - no modifications required
    - 14 had results that differed between paper and authors code

- **American Journal of Political Science**
  - Mean number of resubmissions of package: 1.7
  - Average 8 hours per manuscript to reproduce and curate package
  - Median 53 days increase in publication workflow

- **ACM Transactions on Mathematical Software**
  - Too hard to try and reproduce everything right now
  - Badges for authors who put in extra work to make papers easy to reproduce
  - Additional volunteer reviewers for computational results
How hard can it be...

- **Quarterly Journal of Political Science**
  - 24 computational reproducibility checks 2012 - 2014
    - Only 4 perfect packages - no modifications required
    - 14 had results that differed between paper and authors code

- **American Journal of Political Science**
  - Mean number of resubmissions of package: 1.7
  - Average 8 hours per manuscript to reproduce and curate package
  - Median 53 days increase in publication workflow

- **ACM Transactions on Mathematical Software**
  - Too hard to try and reproduce everything right now
  - Badges for authors who put in extra work to make papers easy to reproduce
  - Additional volunteer reviewers for computational results

*Not that easy*
What are some barriers?
Activity: Analyze + Document

- Complete the following tasks and write instructions/documentation for your collaborator to reproduce your work starting with the original dataset (https://osf.io/qhz4y/).
  - Visualize (using whatever tools you like) life expectancy over time for Canada in 1950s and 1960s using a line plot
  - Something is clearly wrong with this plot! Turns out there’s a data error in the datafile: life expectancy for Canada in the year 1957 is coded as 9999999, it should actually be 69.96. Make this correction
  - Visualize life expectancy over time for Canada again, with corrected data
Activity: Swap + Discuss

- Swap instructions/documentation you used with your collaborator, and try to reproduce their work, first without talking to each other. If your collaborator does not have the software they need to reproduce your work, we encourage you to either help them install it, or walk them through it on your computer in a way that would emulate the experience (Remember, this could be part of the problem!)

- Then, talk to each other about the challenges you faced (or didn’t face) or why you were or weren’t able to reproduce their work
Discuss: What problems did you run into?
Barriers

● Lack of sharing of data/code/software
  ○ All are necessary to check computational reproducibility

● Lack of documentation
  ○ No re-executable code (e.g. description of what you did in excel)
  ○ Code without documentation
  ○ No information about what you need to run code (e.g. libraries, versions)
  ○ Software collapse
    ■ Software is built on operating system, compilers, libraries, which can change to the point where the software no longer can be built or no longer works
  ○ Data without code books/data dictionaries

● Proprietary formats
  ○ License fees or having to rewrite data/code completely into another language/format takes time and money and can lead to errors
Tools

- Many tools out there: Rstudio, Jupyter Notebook, ReproZip, OSF, etc.
  - And more being developed every day

- In general:
  - Want something that is free/open source
  - Helps us with documentation
  - Easily sharable

- Today: Jupyter Notebook, OSF
Jupyter Notebook

- Allows you to combine code, plain text, and output in a narrative notebook style

- Kind of like a lab/field notebook but for your analysis

- Allows for programming in python, but also R
  - R now also has it’s own notebook, R notebook
Why use a notebook?

- Could code directly in python, R, matlab, etc.
  - Would at least allow us to save scripts that we could share with others to help reproducibility

- Notebooks allow for us to combine code, input, output, and plain English descriptions in one document
  - Makes code easier to document and understand
  - Intermediate coding steps are saved in notebook style, so process is better documents
  - Output and code are intertwined so no possibility of copy paste errors
  - Notebooks easily publishable to web and sharable
Jupyter Notebook Demo

https://osf.io/sbnz7/
Virtual Machines and Containers

- Outcomes of code/software sometimes dependent on environment they’re run in
  - e.g. exactly which version of a library they use

- Virtual machines
  - Full encapsulation of running system (OS, hardware, processes, etc.)
  - Can be very large, slow to store/load

- Containers
  - Encapsulates just enough of the environment to run an application
  - Much smaller, lighter-weight
  - Allows us to recreate the running application and environment
  - Can includes code and build process
  - Includes environment variables
Docker

- Standard container today
- Can run locally or on the cloud
- Can run on HPC using Shifter/Singularity
Open Science Framework

http://osf.io
Recap

● Today
  ○ Defined computational reproducibility
  ○ Discussed current barriers
  ○ Introduced Jupyter Notebooks and OSF

● Tomorrow
  ○ Methods and Results Reproducibility