Level up the reproducibility of your data and code!

A 2-hour, hands-on workshop

April Clyburne-Sherin
Instructor & Open Research Consultant
Website: aprilcs.com
Email: aprilcs@pm.me

Purpose

To introduce methods and tools in organization, documentation, automation, and dissemination of research that nudge it further along the reproducibility spectrum.

Outcome

Participants feel more confident applying reproducibility methods and tools to their own research projects.

Process

Participants practice new methods and tools with code and data during the workshop to explore what they do and how they might work in a research workflow. Participants can compare benefits of new practices and ask questions to help clarify which would provide them the most value to adopt.

Reproducibility as a spectrum

Remember that reproducibility is a spectrum - it is not all or nothing! Integrating one new best practice into your research nudges it up the spectrum. Let's see where we land on the spectrum today with a reproducibility check-up!

Reproducibility check-up:

- ❏ My project workspace is structured and organized, with a file naming convention.
- ❏ I have an updated project level README file that includes a file inventory.
- ❏ I have documented my data, such as a data dictionary or codebook.
- ❏ I pre-register my study plan, when appropriate.
- ❏ I automate my analysis when I can, such as with a master script or using make.
- ❏ I have documented my software and computational environment with versions.
- ❏ I track the versions of my materials manually or using version control.
- ❏ All my research materials are backed-up in three locations.
- ❏ I share my code, when appropriate.
- ❏ I share my data in a data repository, when appropriate.
- ❏ I share my study methods or study protocol.
- ❏ I report all my study results publicly, no matter their effect size or direction.

12 checks	Gold standard :D
11 checks	Level 11 Superior
10 checks	Level 10 Superior
9 checks	Level 9 Advanced
8 checks	Level 8 Advanced
7 checks	Level 7 Advanced
6 checks	Level 6 Intermediate
5 checks	Level 5 Intermediate
4 checks	Level 4 Intermediate
3 checks	Level 3 Novice
2 checks	Level 2 Novice
1 check	Level 1 Novice
No checks	Not reproducible :(

When considering which practice to bring into your workflow, adopt the practice that will benefit yourself first - as your future self is the most frequent reuser of your research.

Workshop materials

Links to recommended resources

Good enough practices in scientific computing. Wilson G, Bryan J, Cranston K, Kitzes J, Nederbragt L, et al. (2017) Good enough practices in scientific computing. PLOS Computational Biology 13(6): e1005510. https://doi.org/10.1371/journal.pcbi.1005510

The Practice of Reproducible Research: Case Studies and Lessons from the Data-Intensive Sciences. Kitzes, J., Turek, D., & Deniz, F. (Eds.). (2018). Oakland, CA: University of California Press.
https://www.gitbook.com/book/bids/the-practice-of-reproducible-research/details

Jupyter Notebooks and reproducible data science. Woodbridge M, Sanz D, Mietchen D, & Mounce R (2017).
https://markwoodbridge.com/2017/03/05/jupyter-reproducible-science.html.

Lessons learned: testing reproducibility

Following Open Data Day 2017, Mark Woodbridge, Daniel Sanz, Daniel Mietchen, & Ross Mounce published a blog post called "Jupyter Notebooks and reproducible data science". Their informal experiment was:

- Search PMC for papers that include research notebooks called Jupyter Notebooks -> 107 papers
- Attempt to rerun the notebook to reproduce the findings of the published paper.
- Only able to reproduce one research notebook.
- "My initial thought was that analysing the validity of the notebooks would simply involve searching the text of each article for a notebook reference, then downloading and executing it … It turned out that this was hopelessly naive…"
- Sharing alone does not guarantee reproducibility.
- The lessons they outline in the blog post are the framework for our workshop.

Organization

"It takes some effort to organize your research to be reproducible… the principal beneficiary is generally the author herself." - Jon Claerbout
Source: http://sepwww.stanford.edu/oldsep/matt/join/redoc/web/iris.html

Woodbridge et al. identified organizational barriers to reproducibility.

1. Files, data, dependencies needed to reproduce the findings were often missing.
2. Difficulty locating and identifying necessary materials.

We can organize for reproducibility:

```
.
|-- CITATION
|-- README
|-- LICENSE
|-- requirements.txt
|-- data
|   -- birds_count_table.csv
|-- doc
|   -- notebook.md
|   -- manuscript.md
|   -- changelog.txt
|-- results
|   -- summarized_results.csv
|-- src
|   -- sightings_analysis.py
|   -- runall.py
```

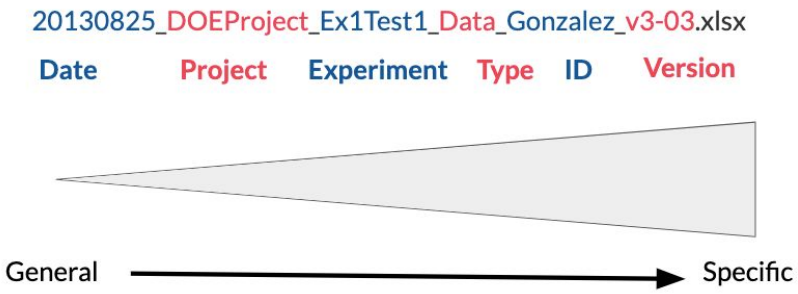
❏ Create a structured and centralized workspace for your project. Source: Karl Broman: http://kbroman.org/steps2rr/pages/organize.html

❏ Create subdirectories for like materials: separate data, code, and results.

❏ Always keep raw data.

❏ Always backup materials in three unique locations.

❏ Adopt an informative naming convention for your files. Source: http://guides.lib.purdue.edu/c.php?g=353013&p=2378292



❏ Adopt an electronic lab notebook. Source: Harvard Medical School Library eLN Features Matrix: https://docs.google.com/spreadsheets/d/1ar8fgwagOh30E31EAPL-Gorwn_g6XNf81g3VDQnQ_I8/edit?usp=sharing

Features	Specifications															
	Benchling	Biovia	Confluence	Docollab	ECL	ELOG	Evernote	Exemplar	Findings	Hivebench	IDBS	LabArchives	LabCollector	LabWare	LabVantage	LabW
Interactivity																
Intuitive Interface Design	✓	No response received	–	–	No response received	–	No response received	No response received	–	–	–	✓	No response received	–	–	
Auto Metadata Harvest	–	No response received	✗	✓	No response received	✗	No response received	No response received	✗	✓	–	✓	No response received	✗	✓	✓
Search functions can search across file formats and beyond typos	–	–	–	–	No response received	–	–	–	–	–	–	–	No response received	–	–	
Ability to manipulate files and images	–	No response received	–	–	No response received	–	No response received	–	–	–	–	–	No response received	–	–	
Support for multiple open windows	✓	–	✓	✓	No response received	✓	✓	–	–	✓	✓	✓	No response received	✓	–	
Ability to link out	✗	No response received	–	✓	✓	✓	✓	✓	✓	✓	✓	✓	No response received	✓	✓	✓
Support for Researcher Documentation																
Hyperlink support	✓	No response received	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	No response received	✓	✓	✓
Metadata Creation Prompts	✓	No response received	✗	✓	No response received	✓	✗	No response received	✗	✓	✓	✓	No response received	✗	✓	✓
Rights Management (licensing)	–	No response received	–	✓	No response received	–	No response received	No response received	✓	✓	–	✓	✓	✓	✓	✓
Protocol Integration	✓	–	✓	✓	No response received	✓	–	–	✓	✓	✓	✓	–	✓	–	✓
Adaptability to Lab workflows																
Accounts/Permissions Levels	✓	No response received	–	✓	✓	✓	✓	–	–	✓	✓	✓	✓	✓	–	✗
Internal Data Sharing	✓	–	–	✓	No response received	✓	✓	No response received	✓	✓	✓	✓	✓	✓	✓	
Adaptable to a Variety of Workflows	–	No response received	–	–	No response received	–	No response received	–	–	–	–	–	–	–	–	
Compatibility with authoring tools	✓	No response received	–	✓	No response received	✗	No response received	No response received	✗	–	–	✓	No response received	✗	–	✗
Windows Compatible	✓	No response received	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
Macintosh Compatible	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Linux Compatible	✓	✗	✓	✓	No response received	✓	No response received	✓	✗	✓	✓	✓	No response received	✓	✗	
Android Compatible	✓	✓	✓	✓	No response received	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✗
iOS Compatible	✓	✓	✓	✓	No response received	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗
Storage																
Cloud Storage	✓	No response received	✗	✓	No response received	✓	No response received	No response received	✓	✓	✓	✓	✓	✓	✓	✗
Local Storage	✗	No response received	✓	✗	No response received	✓	No response received	No response received	✓	✗	✓	✓	No response received	✓	✓	✓
Hybrid (cloud/local) Storage	✗	No response received	✗	✗	No response received	✗	No response received	No response received	✓	✓	✗	✓	No response received	✗	✗	✗
Versioning	–	–	–	–	No response received	–	No response received	–	–	–	–	–	–	–	–	
File Redundancy	–	No response received	–	–	No response received	–	No response received	No response received	–	–	–	–	No response received	–	–	
Creates stable URLs or persistent identifiers for entries	✓	No response received	✓	✓	No response received	✓	No response received	No response received	–	✓	✓	✓	No response received	✓	✓	✓
Can unregistered users access the data found at persistent links?	✓	No response received	✓	✗	No response received	✗	No response received	No response received	–	–	✗	✗	No response received	✗	✗	✗
Storage Capacity - Users	–	No response received	–	–	No response received	–	–	No response received	–	–	–	–	No response received	–	✓	
Storage Capacity - Max File Size	–	No response received	–	–	No response received	–	No response received	No response received	–	–	–	–	No response received	–	✓	

Organization

Let's create a structured, centralized workspace for a research project. It will hold our data, code, research notebooks, documentation, etc.

Exercise 1: Create a workspace for your research.
What tool we will use: Open Science Framework, www.osf.io. OSF is free, open source, and discipline agnostic. It allows private and public research projects and is integrated with commonly used organization and research tools such as Github, Google Drive, and Box.

1. Place a pink sticky note on your laptop.
2. Create an account at www.osf.io.
3. Click "Create New Project".
4. Type a title for your project. If you are practicing with your own research, choose a title based on your project. If you want to follow with an example, type "Candy selection happiness increases with more candy trade".
5. Storage location should default to select United States.
6. Click "Create" and then "Go to project".
7. Place a green sticky note on your laptop when you have a project created.

Exercise 2: Centralize your research materials into your workspace.
1. Place a pink sticky note on your laptop.
2. Go to https://github.com/aprilcs/sips-workshop.
3. Click "Clone or download" and select "Download ZIP".
4. Once they have downloaded, unzip your materials.
5. Now navigate back to your OSF project.
6. Under "Files", select OSF Storage (United States).
7. Click "Upload" and select the files you downloaded.
8. Place a green sticky note on your laptop when you have uploaded your materials.

Exercise 3: Create subdirectories for like materials.
1. Place a pink sticky note on your laptop.
2. Click "Add Components" and create a component named "Code" and "Data".
3. From the file tree, drag and drop the data into Data and code into Code.
4. Place a green sticky note on your laptop when you have moved the materials.

Documentation

Woodbridge et al. identified missing documentation as a barrier to reproducibility

1. Dependencies were often not documented.
2. Documentation on how to reproduce the results was missing or inadequate.

We can create documentation that improves reproducibility:

❏ Create a project level README file.

i. Resources on making a great README file from Cornell University (includes a template): data.research.cornell.edu/content/readme
ii. An example of a reproducible README in an AJPS Replication Package: dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/EZSJ1S
iii. Resource on using markdown to create your README: github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

❏ Document each element or variable in your dataset with a data dictionary. Resources on making a great data dictionary:

i. DataONE: https://www.dataone.org/best-practices/create-data-dictionary
ii. McGill Codebook Cookbook: http://www.medicine.mcgill.ca/epidemiology/joseph/pbelisle/CodebookCookbook.html
iii. UPenn: https://guides.library.upenn.edu/datamgmt/documentation
iv. Karl Broman: http://kbroman.org/dataorg/pages/dictionary.html

❏ Pre-register the study plan with the OSF (all studies), clinicaltrials.gov (clinical research), or PROSPERO (systematic reviews).

❏ Track versions of materials manually, using version control systems like git, or by choosing a tool that has built in versioning such Google Docs or the OSF.

❏ Document analyses, software, computational environment, packages, and other dependencies with versions.

i. Specify your packages and dependencies with versions. Resource on documenting dependencies: https://mybinder.readthedocs.io/en/latest/config_files.html

Python: pip freeze > ../code/requirements.txt
conda list --export > ../code/requirements.txt

R: install.R and runtime.txt

❏ Consider using container technology. Source: Boettiger, Carl. An introduction to Docker for reproducible research. https://doi.org/10.1145/2723872.2723882. Containers solve:

i. Dependency Hell - Provides other researchers with a binary image in which all the software has already been installed, configured, and tested.
ii. Imprecise documentation - Provides a human readable summary of the necessary software dependencies needed to execute the code.

❏ Consider using literate programming to document the analysis narrative with the code in an executable document.

i. Shows what you did and why you did it in one document.
ii. Easily shared.
iii. Best start: Jupyter Notebooks or RMarkdown in RStudio.

Exercise 4: Update your README file.

1. Place a pink sticky note on your laptop.
2. Click on the README.md file in the file tree of your OSF project.
3. Click "Edit" and update the "Contact" section with your details.
4. Click "Save".
5. Place a green sticky note on your laptop.

Exercise 5: Track changes using version control.

1. Place a pink sticky note on your laptop.
2. While in your README file, click "Revisions".
3. View and download the previous version of your README.
4. Place a green sticky note on your laptop.

Exercise 6: Explore literate programming.

5. Place a pink sticky note on your laptop.
6. While in your Code component, click "candy-trade.Rmd".
7. View and download the RMarkdown file and identify an R code chunk.
8. Place a green sticky note on your laptop.

Exercise 7: Explore documentation of computational environments.

9. Place a pink sticky note on your laptop.
10. While in your file tree, click on "install.R" and then "runtime.txt"
11. Brainstorm what you think the runtime.txt file means.
12. Place a green sticky note on your laptop.

Automation

What Woodbridge et al. found:

1. Manual manipulation or setup was needed to reproduce results, often without documentation of how the results were produced.

We can automate the execution of our analyses to improve reproducibility:

❏ Reproduce results automatically as a function of the data & the code.

❏ Create a master script to execute all analyses in order. Resource on automation using a master script: practicereproducibleresearch.org/core-chapters/3-basic.html

- ❏ Create a master script that executes your analysis scripts in order.
  - ❏ In R, use a run.r or main.r master script
    - ❏ Use source() to run your scripts
    - ❏ Run your install.r script
    - ❏ In your master script, use rmarkdown::render to render your RMarkdown document into your results directory.
  - ❏ In Python, use a main.py or run.sh master script
    - ❏ In your run.sh script, use nbconvert to execute your notebook into the results directory.
    - ❏ In your master script, use nbconvert to execute your notebook into your results directory.

❏ Use relative file paths rather than absolute file paths. Resource explaining paths by Karl Broman: http://kbroman.org/steps2rr/pages/organize.html.

Exercise 8: Explore a master script.

- Place a pink sticky note on your laptop.
- While in your code component, click "run.r"
- Brainstorm what you think each line of the master script does.
- Place a green sticky note on your laptop.

Exercise 9: Create a reproducible repository with Binder.

- Place a pink sticky note on your laptop.
- Go to https://github.com/aprilcs/sips-workshop and click "Launch Binder".
- Go to https://mybinder.org/ and type "https://github.com/aprilcs/sips-workshop" into the "GitHub repository name or URL" field.
- Place a green sticky note on your laptop.

Dissemination

What Woodbridge et al. found:

1. There is no standardized way of attaching materials to published articles.
2. Therefore it is difficult to discover and retrieve necessary materials.

We can share using repositories to make our research FAIR (findable, accessible, interoperable, and reusable):

- ❏ Share a link to a trusted repository that contains all our materials.
    - ❏ Obtain a DOI for your repository and use this link throughout your article.
        - ❏ Example: Github -> Binder -> Zenodo -> DOI linked in article.
    - ❏ Cross link repository with published article in metadata of each.
    - ❏ Mint a persistent identifier for your repository such as a DOI. This is unique and citable and persistent.
    - ❏ Archive the exact versions of the materials used for published findings.
    - ❏ Find & compare repositories through Repository of Research Data Repositories: https://www.re3data.org/.
- ❏ Specify a license for your data and your code and materials.
    - ❏ Creative Commons licenses are appropriate for data and text. CC-0 or CC-BY are recommended if you want reuse. Resource: https://creativecommons.org/.
    - ❏ Resource on choosing a data licence is the Digital Curation Center: http://www.dcc.ac.uk/resources/how-guides/license-research-data.
    - ❏ For software, permissive open source licenses such as the MIT, BSD, or Apache licenses promote reuse. Resource on choosing a code licence by Karl Broman: http://kbroman.org/steps2rr/pages/licenses.html.
        - ❏ Open Source Initiative: https://opensource.org/licenses
        - ❏ License picker: https://choosealicense.com/
- ❏ Consider new publishing models that counter publication bias in the research literature.
    - ❏ Preprints - Before you submit your paper for publication, post it for free on a preprint server such as arXiv: https://arxiv.org/. Peers can review your work, provide feedback, and you can ensure access to your work.
    - ❏ Registered Reports - Before you start your study, a journal peer-reviews your pre-registration. Feedback to improve your study is provided before your begin. Journal accepts the study in principle based on theory, design, analyses. The study is published when done.

- ❏ Publish your methods and protocol. Source: https://www.aje.com/en/arc/how-to-write-an-easily-reproducible-protocol/
    - ❏ Think of protocol as brief, modular, self-contained scientific publication. Include 3-4 sentence abstract that puts methodology in context, include as much detail as possible.
    - ❏ Resources: A protocol sharing tool like www.protocols.io allows publish sharing, versioning, and commenting on scientific protocols.

Exercise 10: Explore a reproducible repository with Binder.

- Place a pink sticky note on your laptop.
- Explore the Binder instances that you launched. If the one from Github is launched, navigate to the Code subdirectory and select "run.r".
- Highlight the code in "run.r" and click "Run".
- Place a green sticky note on your laptop when you have started running the master script, "run.r".

Exercise 11: Learn to mint a DOI for your repository.

- Place a pink sticky note on your laptop.
- Return to your OSF project.
- Select "Make Public" if you wish to see how to mint a DOI. Do not do this if you uploaded your own data.
- Now you will see a "Create DOI" option for your project.
- Place a green sticky note on your laptop when you see the "Create DOI" option.

Exercise 12: Pick a licence for your data and code and materials.

- Place a pink sticky note on your laptop.
- Return to your OSF project.
- Select "Add a licence" for your top level project as well as your Data and Code components, one by one.
- From the licence picker, select the licenses you wish.
- Place a green sticky note on your laptop when you have chosen your licenses.