Discussion of 'Teaching Reproducibility and Responsible Workflows'

Chris Paciorek

UC Berkeley Statistics

Reproducibility at JASA

- Initial requirement to provide code/data/workflow for Applications and Case Studies in 2016.
- Extended to Theory and Methods in 2021.
- We review for availability and quality of materials.
  - Insufficient resources to actually run code / attempt reproducibility.
- Resources we provide at jasa-acs.github.io/repro-guide:
  - standardized template to collect reproducibility information,
  - suggested repository structure, and
  - examples of excellent reproducibility work, including winners of (new) JASA Reproducibility Award

Goals / emphases in teaching

What do we want our students to learn and practice?

- Methods
- Data analysis / applied statistics workflow
- Technical writing
- Software development
- Tools for all of the above

Best practices

- Best practices in software development (and arguably application of statistical methods) are well-developed.

Best practices

- Best practices in software development (and arguably application of statistical methods) are well-developed.
- The 'scientific method' is taught in science classes starting early.

Best practices

- Best practices in software development (and arguably application of statistical methods) are well-developed.
- The 'scientific method' is taught in science classes starting early.
- To what extent do we have best practices for computational work?

Best practices

- Best practices in software development (and arguably application of statistical methods) are well-developed.
- The 'scientific method' is taught in science classes starting early.
- To what extent do we have best practices for computational work?
  - organizing code and documenting analysis choices (forking paths) during the tangled early stages of a project

Best practices

- Best practices in software development (and arguably application of statistical methods) are well-developed.
- The 'scientific method' is taught in science classes starting early.
- To what extent do we have best practices for computational work?
  - organizing code and documenting analysis choices (forking paths) during the tangled early stages of a project

    "lab" notebooks? formal tools?

Best practices

- Best practices in software development (and arguably application of statistical methods) are well-developed.
- The 'scientific method' is taught in science classes starting early.
- To what extent do we have best practices for computational work?
  - organizing code and documenting analysis choices (forking paths) during the tangled early stages of a project
    "lab" notebooks? formal tools?
    "auditable"

Best practices

- Best practices in software development (and arguably application of statistical methods) are well-developed.
- The 'scientific method' is taught in science classes starting early.
- To what extent do we have best practices for computational work?
  - organizing code and documenting analysis choices (forking paths) during the tangled early stages of a project
    "lab" notebooks? formal tools?
    "auditable"
  - organizing groups for data analysis projects

Best practices

- Best practices in software development (and arguably application of statistical methods) are well-developed.
- The 'scientific method' is taught in science classes starting early.
- To what extent do we have best practices for computational work?
  - organizing code and documenting analysis choices (forking paths) during the tangled early stages of a project
    "lab" notebooks? formal tools?
    "auditable"
  - organizing groups for data analysis projects
    particularly leader-less groups

Best practices

- Best practices in software development (and arguably application of statistical methods) are well-developed.
- The 'scientific method' is taught in science classes starting early.
- To what extent do we have best practices for computational work?
  - organizing code and documenting analysis choices (forking paths) during the tangled early stages of a project
    "lab" notebooks? formal tools?
    "auditable"
  - organizing groups for data analysis projects
    particularly leader-less groups
  - collaborative writing

Open questions for teaching these areas

- Degree of emphasis on methods/tools vs. writing/project management/collaboration?
- How guide/assess collaboration/project work?
  - How much to trust that students will learn simply from the project experience?
  - Can we hope to guide a group from the outside?
  - What works in terms of peer evaluation?
  - How much automated feedback/guidance/assessment is possible?
  - Best practices for scaffolding for projects / group work / collaboration?
- Can we better share teaching materials and best practices on these topics? ("Reproducible teaching"?)
- What is the role of academia vs. learning on the job?

When does history (versions) matter?

Version control is clearly a critical tool for software development.

How important is versioning/history in computational work?

Context	Internal use	External validity
Software	critical	no
Data analysis	yes	no
Statistical validity	no	yes
Writing	somewhat	no

Are version histories in Google Docs/Microsoft Word/Overleaf sufficient for writing, compared to full version control?

Where do we go from here in teaching and research?

- Efforts like this session to improve teaching of these ideas/tools.
- More emphasis in the discipline on replicability?
  - Encourage others to explore robustness of analyses?
  - How choose what work to investigate in more detail?
  - Reproducible materials a prerequisite for more focus on replicability/robustness.
- More standardized requirements/procedures for ensuring reproducibility?