Coding Reproducibility

Lessons Learned

Julia Clark

BITSS Transparency and Reproducibility Workshop
New Delhi
16 March 2017

Overview

1. About PDEL data transparency project

Overview

1. About PDEL data transparency project
2. What makes code reproducible?

Overview

1. About PDEL data transparency project
2. What makes code reproducible?
3. Lessons Learned:

Overview

1. About PDEL data transparency project
2. What makes code reproducible?
3. **Lessons Learned**:
    - Complete

Overview

1. About PDEL data transparency project
2. What makes code reproducible?
3. **Lessons Learned**:
    - Complete
    - Runs and reproduces

Overview

1. About PDEL data transparency project
2. What makes code reproducible?
3. **Lessons Learned**:
    - Complete
    - Runs and reproduces
    - Readable

Overview

1. About PDEL data transparency project
2. What makes code reproducible?
3. **Lessons Learned**:
    - Complete
    - Runs and reproduces
    - Readable
    - Protects PII

About PDEL

UCSD's Policy Design and Evaluation Lab (pdel.ucsd.edu)

Rigorous social science methodology + information technology $\rightarrow$ policies and programs that alleviate poverty, promote health, welfare, and security, and enhance accountability.

UCSD's Policy Design and Evaluation Lab (pdel.ucsd.edu)

Rigorous social science methodology + information technology $\rightarrow$ policies and programs that alleviate poverty, promote health, welfare, and security, and enhance accountability.

- Interdisciplinary collaboration

UCSD's Policy Design and Evaluation Lab (pdel.ucsd.edu)

Rigorous social science methodology + information technology $\rightarrow$ policies and programs that alleviate poverty, promote health, welfare, and security, and enhance accountability.

- Interdisciplinary collaboration
- Project and fund management support

UCSD's Policy Design and Evaluation Lab (pdel.ucsd.edu)

Rigorous social science methodology + information technology $\rightarrow$ policies and programs that alleviate poverty, promote health, welfare, and security, and enhance accountability.

- Interdisciplinary collaboration
- Project and fund management support
- Data transparency services—helping researchers replicate studies and prepare files for public dissemination

What Makes Code Reproducible?

Replication files that are …

1. Complete but parsimonious

Replication files that are …

1. Complete but parsimonious

2. Run and reproduce results with one click

Replication files that are …

1. Complete but parsimonious
2. Run and reproduce results with one click
3. Readable and interpretable by humans

Replication files that are …

1. Complete but parsimonious
2. Run and reproduce results with one click
3. Readable and interpretable by humans
4. Protect personal information

Why do we care?

- **Unselfish reasons**—part of the scientific process and a public good

Why do we care?

- **Unselfish reasons**—part of the scientific process and a public good
- **Selfish reasons**—make code more usable for yourself, catch potentially embarrassing errors before they become public, boost your transparency credibility

Lessons Learned

1. Complete and Parsimonious

Necessary: All materials needed to generate and decipher results are included in the replication files, including …

1. Complete and Parsimonious

Necessary: All materials needed to generate and decipher results are included in the replication files, including …

- Code—for analysis AND cleaning/merging data files

1. Complete and Parsimonious

Necessary: All materials needed to generate and decipher results are included in the replication files, including …

- Code—for analysis AND cleaning/merging data files
- Data—raw and manipulated

1. Complete and Parsimonious

Necessary: All materials needed to generate and decipher results are included in the replication files, including …

- Code—for analysis AND cleaning/merging data files
- Data—raw and manipulated
- Supplementary files (codebooks, readme files, etc.)

1. Complete and Parsimonious

Necessary: All materials needed to generate and decipher results are included in the replication files, including …

- Code—for analysis AND cleaning/merging data files
- Data—raw and manipulated
- Supplementary files (codebooks, readme files, etc.)

1. Complete and Parsimonious

Necessary: All materials needed to generate and decipher results are included in the replication files, including …

- Code—for analysis AND cleaning/merging data files
- Data—raw and manipulated
- Supplementary files (codebooks, readme files, etc.)

Sufficient: Unnecessary files (e.g., old versions of figures, tables, data not used in analysis) should NOT be included—AKA, don't just share your project directory as-is!

2. Runs & Reproduces

Code and data should **reproduce** the paper's results without error.

2. Runs & Reproduces

Code and data should **reproduce** the paper's results without error.

- This includes ALL tables, graphs, etc. in paper

2. Runs & Reproduces

Code and data should **reproduce** the paper's results without error.

- This includes ALL tables, graphs, etc. in paper
- Ideally code can be executed with a **single click**

2. Runs & Reproduces

Code and data should **reproduce** the paper's results without error.

- This includes ALL tables, graphs, etc. in paper
- Ideally code can be executed with a **single click**
- Great if it runs on your machine, but always good to test on other computer/OS/software version to debug

3. Readable and interpretable (by humans!)

Code should be streamlined and legible, with intuitively organized files.

3. Readable and interpretable (by humans!)

Code should be streamlined and legible, with intuitively organized files.

- Clearly labeled files within a logical folder structure

3. Readable and interpretable (by humans!)

Code should be streamlined and legible, with intuitively organized files.

- Clearly labeled files within a logical folder structure
- Separate code for data analysis/merging/cleaning, ideally with master script to run all

3. Readable and interpretable (by humans!)

Code should be streamlined and legible, with intuitively organized files.

- Clearly labeled files within a logical folder structure
- Separate code for data analysis/merging/cleaning, ideally with master script to run all
- Comment to help reader navigate/interpret

3. Readable and interpretable (by humans!)

Code should be streamlined and legible, with intuitively organized files.

- Clearly labeled files within a logical folder structure
- Separate code for data analysis/merging/cleaning, ideally with master script to run all
- Comment to help reader navigate/interpret
- Declutter syntax (ample use of spaces, indentation, headers)

3. Readable and interpretable (by humans!)

Code should be streamlined and legible, with intuitively organized files.

- Clearly labeled files within a logical folder structure
- Separate code for data analysis/merging/cleaning, ideally with master script to run all
- Comment to help reader navigate/interpret
- Declutter syntax (ample use of spaces, indentation, headers)
- Code for main analysis should be prominent & clearly labeled

4. Protects PII

Personally identifiable information (PII) includes direct identifiers (e.g., name, address, etc.) and indirect identifiers—personal characteristics that, when combined, can identify a person.

4. Protects PII

Personally identifiable information (PII) includes direct identifiers (e.g., name, address, etc.) and indirect identifiers—personal characteristics that, when combined, can identify a person.

- This info should be removed/altered in public data files, with original files stored securely

4. Protects PII

Personally identifiable information (PII) includes direct identifiers (e.g., name, address, etc.) and indirect identifiers—personal characteristics that, when combined, can identify a person.

- This info should be removed/altered in public data files, with original files stored securely
- When possible, anonymize ***before*** merging/cleaning so that data and code for these processes can be shared publicly