Data Management and Cleaning Workshop

Education Policy Research Practicum

Patrick Lavallee Delgado

University of Pennsylvania Graduate School of Education

7 February 2024

Penn GSE
GRADUATE SCHOOL OF EDUCATION
UNIVERSITY of PENNSYLVANIA

Outline

1. Data privacy
2. Data management
3. Data cleaning

Data privacy

Data is great and all, but...

Many laws and regulations govern access and disclosure of personal data, rules which extend to research. Major legislation in the US includes:

- National Research Act (IRB)
- FERPA (students)
- COPPA (children online)
- HIPAA (health)

Some data are subject to more stringent regulation by other jurisdictions, such as GDPR (EU) and CCPA (California).

Data privacy

Some data elements may be particularly sensitive and require special handling.

- Personally identifiable information (PII): information or groupings of information that can distinguish or trace the identity of an individual, e.g. name, social security number.
- Controlled unclassified information (CUI): information the government owns that requires safeguarding or dissemination controls as provided by law or regulation, e.g. trade secrets, budget and policy deliberations.

Note

Negligence can cause harm. Researchers have a legal and ethical responsibility to ensure data privacy and security.

Data use

A data use agreement (DUA) governs the exchange of data, establishing who is permitted to access the data, how they can use the data, and what can be disclosed of the data.

The DUA may require that the data be stored and processed in a secure location, e.g. remote server, cold room. It may also require that the data be destroyed upon project completion.

Note

We will store data on Box and invite you to the Box folder. *Do not move, copy, or share data out of the Box folder.*

Outline

1. Data privacy

2. Data management

3. Data cleaning

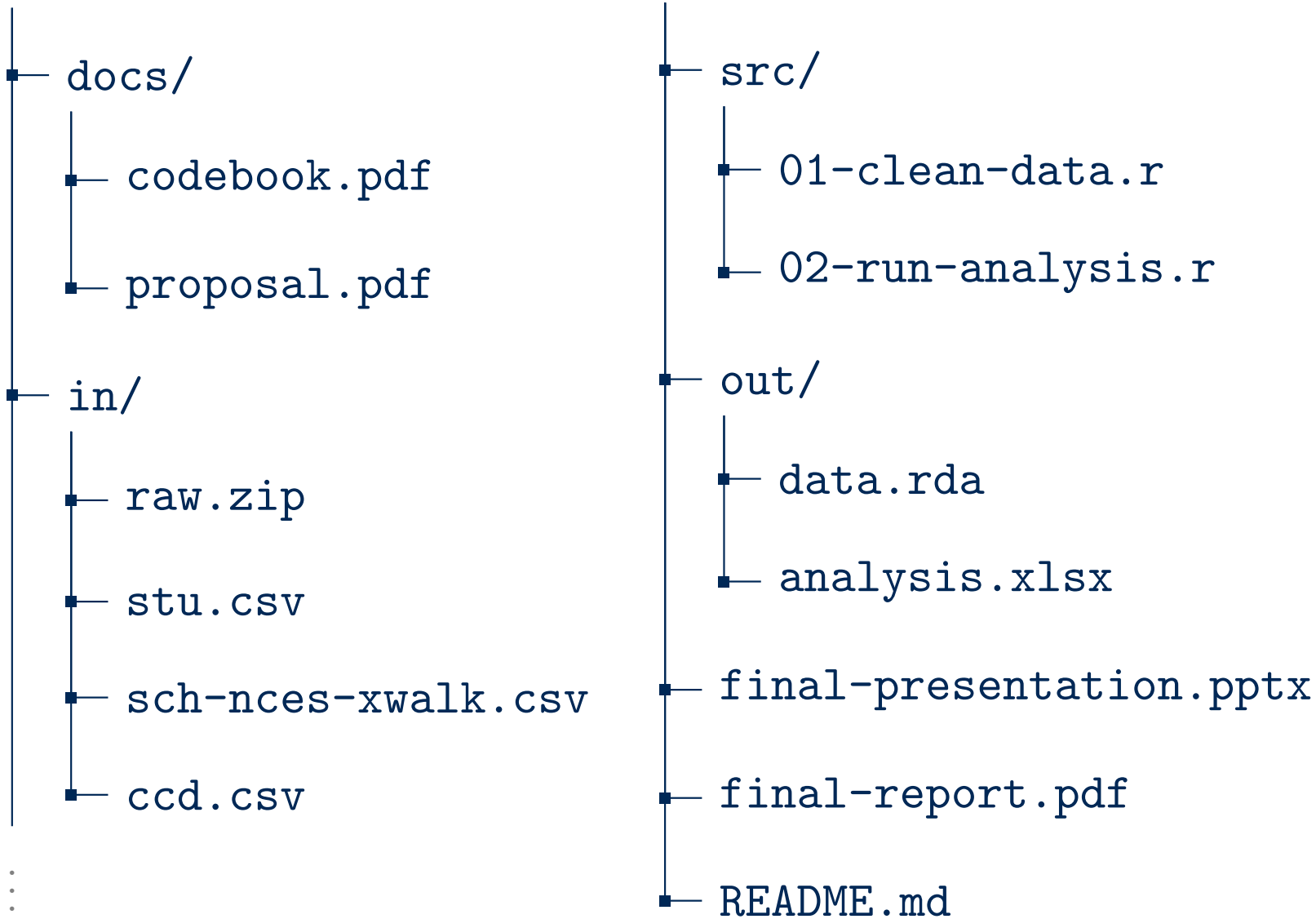
Repository

The repository is the container for all project work. It documents and demonstrates – in words and code – how the data answer the research questions, and allows the client to replicate the results.

Replication should be turnkey. Anyone who returns to your work should be able to:

- Run your code with minimal effort and without error.
- Get the same results in your report and presentation.
- Understand what you did, how you did it, and why.

Sample repository structure



Documentation

- **Describe and defend.** Demonstrate the thought, intention, and thoroughness of your work and help others understand what might not be obvious about the data and analysis.
- **Data description.** Discuss the data used in the analysis with the final report. List the data sources, data elements, summary statistics, and major decisions in cleaning and variable construction.
- **Comment your code.** Provide in-line notes that describe what the code is doing. They helps non-programmers follow along, a reviewer verify the code does as you intend, and your future self pick up from where you left off.
- **README.** This file gives an overview of the repository and how to use it. It also lists any additional requirements for replication, usually software.

Sample README

Project name
Project team
Date submitted

Brief description of the project. This repository provides for replication of our findings.

Contents:

- `in/`
  - `raw.zip` Data from client, **not included**.
  - `ccd.csv` School characteristics from Common Core of Data.
  - ...
- `src/`
  - `01-clean-data.r` Assembles analysis file from data.
  - `02-run-analysis.r` Creates tables and figures.
- ...

Sample README

Requirements:

- Data from client
- R version 4.2.3 or later
- tidyverse version 2.0.0 or later

How to replicate:

1. Satisfy data and software requirements above.
2. Open R.
3. Set the working directory to this repository.
4. Run each program in `src/` sequentially.
5. Find tables and figures in `out/`.

Quality assurance

- **Do not overwrite raw data.**
- **Programming.** The code should assemble the analysis file and produce all analyses, tables, and figures from the raw data. Nothing should require human intervention.
- **Diagnostics.** Report summary statistics in tables or figures. Do the data make sense? Check what needs to be cleaned and that cleaning happens correctly.
- **Version control.** Save snapshots of programs and output before implementing major changes in case something goes wrong. I recommend Git, but it has a steep learning curve.
- **Code review.** Walk through each program with the project team to check for agreement on cleaning, variable construction, and statistical analysis, as well as for errors.

Quality assurance

- **Defensive programming.** Document assumptions of the data and expectations of the code with assert statements that break the program when the assumptions fail.

```
# There are no missing values.
stopifnot(all(!is.na(df$x)))
```

- **Separate code from data.** Data are values and code is behavior. Instead of hardcoding values, load another dataset. This preserves data integrity and provides reusable code.

```
# This is prone to error.
# df$sch_nm[df$sch_id == "A"] <- "Abbot"

# Treat data like data.
sch <- read.csv("in/sch.csv")
df <- left_join(df, sch, by = "sch_id",
                relationship = "many-to-one")
```

Outline

1. Data privacy

2. Data management

3. Data cleaning

Workshop

Imagine we want to measure student achievement and growth in seventh grade. We have data that follow students from sixth to seventh grade and describe their demographic characteristics and exit exam scores in core subjects.

The notebooks linked below walk through common data cleaning and analysis tasks on an example dataset engineered to resemble real data.

- Cleaning example
- Analysis example