

# Data management and analysis workshop

## Education Policy Research Practicum

Patrick Lavallee Delgado

University of Pennsylvania, Graduate School of Education

29 January 2025

# Outline

1. Data privacy

2. Data management

3. Data analysis

# Data privacy

The individuals in your data have a right to and expectation of privacy.

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 analysis

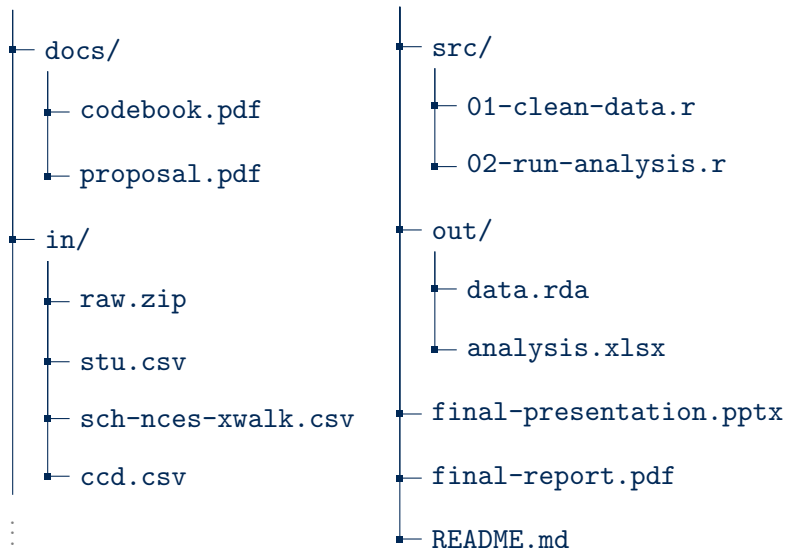
# Repository

The repository is the container for all project work. It documents and demonstrates – in words and in 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. This helps non-programmers follow along, a reviewer to verify the code does as you intend, and your future self to pick up from where you left off.

# Documentation

- **README.** This file gives an overview of the repository and how to use it. It also lists any additional requirements for replication, usually software.
- **Codebook.** This file describes each variable in the dataset assembled for the analysis, including labels, summary statistics, and notes about construction.

# 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/`.

# Sample codebook

race: Student race classification

Value	Label	Frequency
1	White	600
2	Black	250
3	Hispanic	250
4	Other	60
-9	Missing	10

Notes: “Other” includes students classified as Asian, American Indian or Alaskan Native, Native Hawaiian or Pacific Islander, and of two or more races.

# Sample codebook

`gpa`: Student grade point average

Mean	Min.	25%	50%	75%	Max.	Miss.
2.81	0.00	1.51	2.30	3.32	4.00	20

Notes: Passing marks where no letter grade was awarded are coded to 3 grade points. Students who were not enrolled or withdrew before the end of the semester are coded to missing.

# 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 analysis

# Example project

Imagine we are interested in exploring investments in students by the racial composition of school districts. Specifically, we have two research questions:

- RQ1. How does per pupil expenditure vary by school district composition?
- RQ2. Do these differences change with the size of the majority group?

In service of these research questions, we have publicly available data on the universe of public school districts in the US from the Common Core of Data.

# Data analysis pipeline

The data analysis pipeline is the series of processing steps that turn raw data into results. Work backwards from the anticipated deliverables to scope the pipeline:

4. What tables and figures would answer the questions?
3. What statistics would populate them?
2. What measures do the analyses require?
1. What data are available?

## Note

You may need to revise the research questions and empirical strategy to work within the constraints of the data you have.

# Table shell

Start with a mock-up of the table that would present the study findings. What would the data analysis pipeline need to be?

## Per-pupil expenditure by district composition

	Number of districts	Average per-pupil expenditure
Majority Asian		
Majority Black		
Majority Hispanic		
Majority White		
No majority		
All districts		

# Workshop

The notebooks linked below walk through common data analysis tasks for this example project:

- Dataset assembly
- Analysis
- Visualization