Three kinds of file, and only one of them is editable

Raw data is read-only, derived data is disposable, and code is the only thing anyone edits. Getting that boundary wrong is how an analysis becomes irreproducible in a way nobody notices for eleven months.

Pierre Robentz Cassion

Lesson 1 of 8

What this lesson covers

- The layout
- The test that decides whether you have it right
- Make the raw data unwriteable, literally
- Naming, which is provenance
- What goes in the repository
- The README that is actually read
- Report it whole
- What comes next

The layout — Example

```
project/
  data/
    raw/              never edited, never written to, ideally chmod -w
    reference/        lookup tables, thresholds, admin boundary codes
  R/  or  src/        the only files anyone edits
  outputs/
    derived/          cleaned data, disposable
    figures/          charts and the CSV of values behind each
    reports/          rendered documents
  tests/
  README.md
  environment lock    uv.lock or renv.lock
```

The layout

- Three kinds of file and the distinction is not tidiness
- Raw data is read-only — The export from CommCare, the CSV from DHIS2, the spreadsheet the ministry sent
- Derived data is disposable — Anything in `outputs/` can be deleted at any moment and reproduced by running the code
- Code is the only thing edited — Every correction, every recode, every exclusion is a line of code, which means it is...

The test that decides whether you have it right — Shell

```
rm -rf outputs/ && make all      # or: python run.py, Rscript run.R
git status --short               # should be empty
```

The test that decides whether you have it right

- Delete every output, rerun, and the repository should look untouched — That single command is the whole of. . .
- Three ways it usually fails — and each names something real:
- A file in `outputs/` that nothing produces — Someone made it by hand, once, and every run since has depended on it
- A step that only runs interactively — A cell in a notebook, a line pasted into a console, a manual download
- Something that must be run in a particular order and does not say so — The pipeline works on your machine because you. . .

Make the raw data unwriteable, literally — Shell

```
chmod -R a-w data/raw/
```

Make the raw data unwriteable, literally — In R

```r
# In R, the equivalent discipline is a function that only ever reads:
read_raw <- function(name) readr::read_csv(file.path("data/raw", name))
```

Make the raw data unwriteable, literally

- It costs one command and prevents the failure it names — An accidental `to_csv("data/raw/survey.csv")` in a notebook is...
- Where the raw file is large or restricted, commit its checksum instead

Make the raw data unwriteable, literally — In Python

```
import hashlib, pathlib

def fingerprint(path: pathlib.Path) -> str:
    return hashlib.sha256(path.read_bytes()).hexdigest()[:16]

print(fingerprint(pathlib.Path("data/raw/survey-2025.csv")))
```

Make the raw data unwriteable, literally — In R

```
tools::md5sum("data/raw/survey-2025.csv")
```

Make the raw data unwriteable, literally

- A checksum in the repository turns "is this the same export?" into a question with an answer — Ninety per cent of "the...

Naming, which is provenance

Name	What it tells you
`survey.csv`	Nothing
`final_survey_v2_FINAL.xlsx`	That there are several and this one won an argument
`household-survey-2025.v1.csv`	What, when, and which version

Naming, which is provenance

- Version by filename, never by overwriting — This platform's datasets do exactly that: a correction ships as `.v2.csv`...
- That is why the dataset files can be cached immutably — and why the course PDFs — which *are* regenerated in place —...

What goes in the repository

In	Out
Code, every script	Raw data with personal information
The environment lock file	Anything above about 50 MB
Small reference tables	Credentials, tokens, connection strings
Generated outputs the build cannot rebuild	`.Rhistory`, `__pycache__`, `.DS_Store`
A README	Outputs that regenerate in seconds

What goes in the repository

- The fourth row is a judgement rather than a rule — This platform commits its generated PDFs, decks and figures —. . .

The README that is actually read — Example

```
# Nutrition surveillance analysis, Artibonite

## What this produces
A quarterly GAM estimate by commune, and the figures in the cluster report.

## Run it
    uv sync
    uv run python run.py

## Where the data comes from
data/raw/muac-screening-*.csv -- monthly export from the screening database,
downloaded by the M&E officer on the 5th. Checksums in data/raw/CHECKSUMS.

## What you need to know
Commune names arrive spelled four ways; normalisation is in src/clean.py and
must run before anything else.
```

The README that is actually read

- The last section is the one that saves a successor a week — It is the knowledge that lives in your head, and the...

Report it whole — Example

```
Analysis reproducibility

  Raw exports are read-only under data/raw/ with SHA-256 checksums committed.
  All cleaning, exclusion and recoding is in version-controlled code; no raw
  file has been edited.

  Every output in outputs/ is reproducible by `uv run python run.py` from a
  clean checkout. Deleting outputs/ and rerunning leaves the repository
  unchanged.

  Dataset files are versioned by filename. The v1 file referenced by the
  March report has not been modified; the correction ships as v2.
```

Report it whole

- The second paragraph is the claim to make and the one to test before making — It is checkable in one command, and an...

What comes next

- Version control is what makes "the only thing edited is code" into a history you can read.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/reproducible-workflows/repro-01-three-kinds-of-file