

Trois sortes de fichiers, et une seule est éditable

Les données brutes sont en lecture seule, les données dérivées sont jetables, et le code est la seule chose que quiconque édite. Se tromper sur cette frontière est la façon dont une analyse devient irreproductible sans que personne s'en aperçoive pendant onze mois.

Pierre Robentz Cassion

Leçon 1 sur 8

Chaînes d'analyse reproductibles — Le projet sur le disque

Ce que couvre cette leçon

- L'organisation
- Le test qui décide si c'est correct
- Rendez les données brutes non modifiables, littéralement
- Le nommage, qui est de la provenance
- Ce qui va dans le dépôt
- Le README qui est réellement lu
- Rapportez-le en entier
- La suite

L'organisation — Exemple

```
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
```

- Trois sortes de fichiers, et la distinction n'est pas une question de rangement
- Les données brutes sont en lecture seule — L'export de CommCare, le CSV de DHIS2, le tableur envoyé par le ministère
- Les données dérivées sont jetables — Tout ce qui est dans outputs/ peut être supprimé à tout moment et reproduit en...
- Le code est la seule chose éditée — Chaque correction, chaque recodage, chaque exclusion est une ligne de code, ce qui...

Le test qui décide si c'est correct — Shell

```
rm -rf outputs/ && make all  
git status --short
```

```
# or: python run.py, Rscript run.R  
# should be empty
```

Le test qui décide si c'est correct

- Supprimez chaque sortie, relancez, et le dépôt doit paraître intact — Cette seule commande est toute la...
- Trois façons dont cela échoue généralement — et chacune nomme quelque chose de réel :
- Un fichier dans outputs/ que rien ne produit — Quelqu'un l'a fait à la main, une fois, et chaque exécution depuis en...
- Une étape qui ne s'exécute qu'en interactif — Une cellule dans un carnet, une ligne collée dans une console, un...
- Quelque chose qui doit être exécuté dans un ordre précis et ne le dit pas — La chaîne fonctionne sur votre machine...

Rendez les données brutes non modifiables, littéralement — Shell

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

Rendez les données brutes non modifiables, littéralement — En 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))
```

Rendez les données brutes non modifiables, littéralement

- Cela coûte une commande et prévient l'échec qu'elle nomme — Un `to_csv("data/raw/survey.csv")` accidentel dans un...
- Là où le fichier brut est volumineux ou sensible, commitez plutôt sa somme de contrôle

Rendez les données brutes non modifiables, littéralement — En 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")))
```

Rendez les données brutes non modifiables, littéralement — En R

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

Rendez les données brutes non modifiables, littéralement

- Une somme de contrôle dans le dépôt transforme « est-ce le même export ? » en une question qui a une réponse —...

Le nommage, qui est de la provenance

Nom	Ce qu'il vous apprend
survey.csv	Rien
final_survey_v2_FINAL.xlsx	Qu'il y en a plusieurs et que celui-ci a gagné une dispute
household-survey-2025.v1.csv	Quoi, quand, et quelle version

Le nommage, qui est de la provenance

- Versionnez par le nom de fichier, jamais par écrasement — Les jeux de données de cette plateforme font exactement cela. . .
- C'est pourquoi les fichiers de données peuvent être mis en cache de façon immuable — et pourquoi les PDF de cours —. . .

Ce qui va dans le dépôt

Dedans	Dehors
Le code, chaque script Le fichier de verrouillage d'environnement Les petites tables de référence Les sorties générées que la compilation ne peut pas reconstruire Un README	Les données brutes à caractère personnel Tout ce qui dépasse une cinquantaine de Mo Identifiants, jetons, chaînes de connexion .Rhistory, __pycache__, .DS_Store Les sorties qui se régénèrent en quelques secondes

- La quatrième ligne est un jugement plutôt qu'une règle — Cette plateforme commite ses PDF, diaporamas et figures. . .

Le README qui est réellement lu — Exemple

```
# 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.
```

Le README qui est réellement lu

- La dernière section est celle qui épargne une semaine à un successeur — C'est la connaissance qui vit dans votre tête,...

Rapportez-le en entier — Exemple

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`.

- Le deuxième paragraphe est l'affirmation à faire et celle à tester avant de la faire
— Elle se vérifie en une commande,...

- La gestion de versions est ce qui transforme « la seule chose éditée est le code » en un historique lisible.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/reproducible-workflows/
repro-01-three-kinds-of-file](https://data-analysis.cassion.dev/fr/cours/reproducible-workflows/repro-01-three-kinds-of-file)