

Git does not forget, which is the point and the danger

A repository remembers every version of every file it has ever held, which is exactly what you want for code and exactly what you cannot allow for a beneficiary list. Deleting the file in a later commit does not remove it.

Pierre Robentz Cassion

Lesson 2 of 8

Reproducible Analysis Workflows — The project on disk

What this lesson covers

- The commit that cannot be taken back
- The `.gitignore` you write before the first commit
- What "personal data" means here, precisely
- The check that runs before the commit does
- What to do if it has already happened
- Commits on analysis work
- What a repository of programme analysis should contain
- Report it whole
- What comes next

The commit that cannot be taken back — Shell

```
git add data/raw/beneficiaries.csv
git commit -m "add survey data"
# ... realised three days later ...
git rm data/raw/beneficiaries.csv
git commit -m "remove data"
```

The commit that cannot be taken back

- The file is still in the repository — It is in the history, it is in every clone, it is on the remote, and it is in the...
- On a repository holding protection or health data, that is a disclosure — Not a mistake to fix later — a disclosure,...
- Which is why the rule is preventive rather than corrective — You cannot un-commit personal data; you can only stop it...

The .gitignore you write before the first commit — Example (cont.)

```
# Raw data. Nothing under here is ever committed.
data/raw/*
!data/raw/.gitkeep
!data/raw/CHECKSUMS

# Derived data, regenerated by the pipeline
outputs/

# Credentials, in every form they arrive in
.env
*.pem
*_secret*
config.local.*

# Editor and language noise
.Rhistory
```

The .gitignore you write before the first commit — Example (cont.)

```
.RData  
__pycache__/  
*.pyc  
.DS_Store  
.ipynb_checkpoints/
```

The `.gitignore` you write before the first commit

- Write it first, commit it first — A `.gitignore` added after the first commit is a `.gitignore` that arrived too late...
- The negations matter — `data/raw/*` excludes everything, then `!data/raw/CHECKSUMS` lets back exactly the file that...

What "personal data" means here, precisely

Looks anonymous	Is not, because
A case ID	It joins to a case management system that has the name
GPS coordinates of a household	It is an address
Date of birth plus commune plus sex	Three fields identify most people in a small commune
A photograph of a form	Every field on it
A free-text "notes" column	It contains whatever the caseworker typed

What "personal data" means here, precisely

- The protection course established the arithmetic — a four-way disaggregation of an anonymous dataset produced nineteen. . .
- So the test is not "does it have a name column" — It is whether any combination of what the file holds could pick one. . .

The check that runs before the commit does — Shell

```
# .git/hooks/pre-commit -- or the pre-commit framework, or a CI step
if git diff --cached --name-only | grep -qE '^data/raw/'; then
    echo "Refusing to commit anything under data/raw/" >&2
    exit 1
fi
```

The check that runs before the commit does — In R

The same check as an R function, run by whatever CI you have.

The check that runs before the commit does

- A hook that refuses is worth more than a policy that reminds — The failure mode is someone in a hurry at 6pm, and a...
- Add a scan for the shapes secrets take — because a token pasted into a script is the other common case

What to do if it has already happened

- One: report it — Your organisation has a data-incident procedure
- Two: rotate anything that was a credential — A committed token is compromised permanently; deleting it changes nothing
- Three: rewrite the history — with `git filter-repo` or BFG, and force-push
- Four: assume it was fetched — On a public repository, assume the file was retrieved and mirrored, and treat the...

What to do if it has already happened — Shell

```
git filter-repo --path data/raw/beneficiaries.csv --invert-paths
```

What to do if it has already happened

- Step three is the only one that looks like a fix and it is the least important

Commits on analysis work

- Commit the code and the output together when the output is committed — A figure and the script that made it belong in...
- Write the why in the message — `fix commune names` is what the diff already shows

Commits on analysis work — Shell

```
git commit -m "Group four spellings of Nord-Ouest before computing coverage"
```

Ungrouped, the worst district splits into fragments of 727, 33, 22 and 20 households and none of them looks alarming. The 20-household fragment shows 0% open defecation, which a district table would report as a solved problem."

Commits on analysis work

- Branch per analysis question, not per day — A branch that answers "does coverage differ by district" can be reviewed. . .

What a repository of programme analysis should contain — Example

project/	
.gitignore	committed first
README.md	what it produces, how to run it, what to know
uv.lock	the environment, pinned
src/	every transformation
tests/	the checks that fail loudly
data/	
raw/	.gitkeep and CHECKSUMS only
reference/	committed: thresholds, admin codes, lookup tables
outputs/	ignored

What a repository of programme analysis should contain

- Reference tables are committed and raw data is not — and the line between them is whether the file describes people

Report it whole — Example

Data handling in this analysis

No raw data is committed. `data/raw/` is excluded by `.gitignore`, with only a `CHECKSUMS` file tracked so the exports used can be identified.

A pre-commit hook refuses any staged file under `data/raw/` and a CI step scans for credential patterns.

Reference tables (WHO growth standards, IPC thresholds, admin codes) are committed; they describe no individual.

The repository is private and access is limited to the M&E team. Raw exports are held in [the organisation's storage], not here.

Report it whole

- The last line is the one that gets forgotten and it is the one an audit asks for —
Saying where the data *is* matters. . .

What comes next

- Code in version control still does not reproduce a number if the environment differs.

Read the full lesson, with runnable code

```
https://data-analysis.cassion.dev/courses/reproducible-workflows/  
repro-02-git-and-what-ever-goes-in
```