

The log that answers the question before it is asked

One row per rule applied — what it was, how many rows it touched, what changed, who approved it. Generated from the code rather than written afterwards, and shipped as the annex that ends the argument.

Pierre Robentz Cassion

Lesson 8 of 8

Data Cleaning and Validation — The record that survives you

What this lesson covers

- The question you will be asked
- A log is a table, not a narrative
- The seven columns, and why each earns its place
- Generate it from the code
- The before-and-after table is the deliverable
- What must not be in the log
- Where the log lives
- The annex the report needs
- Where this course leaves you

The question you will be asked

- A donor's monitoring visit: *"Your caseload fell 4% from the draft to the final report. What changed?"*
- The programme manager: *"Last quarter Gros-Morne was second worst. Now it is fourth. Did something improve?"*
- Your successor, in an email: *"There is a script here that drops rows. Do you know why?"*

A log is a table, not a narrative

- One row per rule applied. Every row carries a count

A log is a table, not a narrative

rule	severity	rows	action	effect on the indicator	decided by	date
muac-range	error	7	excluded	GAM 8.62% to 8.63%	R. Cassion	2026-07-14
exact-duplicate	error	12	one copy kept	caseload 366 to 360	R. Cassion	2026-07-14
re-registration	warning	6	merged after review	caseload 360 to 354	M. Joseph, supervisor	2026-07-16
oedema-yn-coding	warning	25	recoded to false	GAM unchanged	R. Cassion	2026-07-14
age-missing	note	226	kept, flagged	age-disaggregated table only	R. Cassion	2026-07-14
district-mapping	warning	75	mapped to HT07	Nord-Ouest 14.7% to 15.0%	A. Pierre, field office	2026-07-15

The seven columns, and why each earns its place

- **rule** — the identifier from the rule table in lesson 5, so the log and the code use the same name. If the log says. . .
- **severity** — what the check meant, so a reader can skim to the errors.
- **rows** — the count. Not "a few", not "some outliers". A number.
- **action** — excluded, corrected, recoded, merged, kept and flagged. Five verbs cover almost everything, and. . .
- **effect on the indicator** — the before and after. This is the column people skip and the only one a donor reads.
- **decided by** — a person. Not "the team". The rule with no name against it is the rule that turns out to be wrong.

The seven columns, and why each earns its place (cont.)

- **date** — when, so the log can be read against the version of the output it produced.

Generate it from the code — In Python (cont.)

```
LOG = []
```

```
def apply_step(df, rule, mask, action, note=""):
    before = indicator(df)
    if action == "exclude":
        out = df[~mask]
    elif action == "flag":
        out = df.assign(**{f"flag_{rule}": mask})
    else:
        raise ValueError(action)
    after = indicator(out)

    LOG.append({
        "rule": rule,
        "rows": int(mask.sum()),
```

Generate it from the code — In Python (cont.)

```
    "action": action,
    "before": before,
    "after": after,
    "effect": round(after - before, 4),
    "note": note,
})
return out
```

```
muac = apply_step(muac, "muac-range", ~muac["muac_mm"].between(80, 220), "exclude")
muac = apply_step(muac, "age-missing", muac["age_months"].isna(), "flag",
                  "kept; age-disaggregated tables only")
```

```
log = pd.DataFrame(LOG)
log.to_csv("outputs/cleaning-log.csv", index=False)
```

Generate it from the code — In R (cont.)

```
LOG <- list()

apply_step <- function(df, rule, mask, action, note = "") {
  before <- indicator(df)
  out <- switch(action,
    exclude = df[!mask, ],
    flag     = dplyr::mutate(df, "flag_{rule}" := mask),
    stop("unknown action: ", action)
  )
  after <- indicator(out)

  LOG[[length(LOG) + 1]] <- tibble::tibble(
    rule = rule, rows = sum(mask, na.rm = TRUE), action = action,
    before = before, after = after, effect = round(after - before, 4), note = note
  )
  out
}
```

Generate it from the code — In R (cont.)

```
}  
  
muac <- apply_step(muac, "muac-range", !dplyr::between(muac$muac_mm, 80, 220), "exclude")  
muac <- apply_step(muac, "age-missing", is.na(muac$age_months), "flag",  
                  "kept; age-disaggregated tables only")  
  
readr::write_csv(dplyr::bind_rows(LOG), here::here("outputs", "cleaning-log.csv"))
```

Generate it from the code

- A step cannot happen without being logged — because logging is what the function does
- The effect is measured, not estimated — `before` and `after` come from the same `indicator()` function, evaluated on...

The before-and-after table is the deliverable — In Python

```
print(f"Records received:  {len(raw):>6,}")  
print(f"Excluded:          {len(raw) - len(analysis):>6,}")  
print(f"Analysed:          {len(analysis):>6,}")  
print(f"GAM, raw:           {indicator(raw):>6.1%}")  
print(f"GAM, cleaned:       {indicator(analysis):>6.1%}")
```

The before-and-after table is the deliverable — In R

```
cat(sprintf("Records received: %6d\n", nrow(raw)))  
cat(sprintf("Excluded:           %6d\n", nrow(raw) - nrow(analysis)))  
cat(sprintf("Analysed:           %6d\n", nrow(analysis)))  
cat(sprintf("GAM, raw:           %6.1f%%\n", 100 * indicator(raw)))  
cat(sprintf("GAM, cleaned:        %6.1f%%\n", 100 * indicator(analysis)))
```

What must not be in the log

- **No direct identifiers.** Not names, not phone numbers, not GPS coordinates. A log row reading 'merged CH03315 into...
- **No small cells in a sensitive category.** "3 rows recoded, GBV case category rape, Marmelade" identifies people in...
- **No free-text detail copied from a case note.** The rule is what you applied, not what the record said.

Where the log lives — Example

outputs/

profiles/muac-2024-q4.json	arrival profile	(lesson 1)
validation/muac-2024-q4.json	validation findings	(lesson 7)
cleaning-log.csv	decisions	(this lesson)
review/duplicate-candidates.csv	reviewed and signed	(lesson 4)
tables/gam_by_commune.csv	the result	
tables/definitions.csv	numerator, denominator, source	

The annex the report needs

- Data source and completeness — Where the file came from, how many records, what share of expected reporting units it. . .
- Cleaning and exclusions — The log table, or a summary of it with the full table attached
- Limitations — What the cleaning could not fix

The annex the report needs

The annex is not a confession. It is the reason the number in the executive summary can be defended, and the professional difference between a figure and a claim.

Where this course leaves you

- You can take a raw programme export, describe honestly what arrived, find what is wrong with it in four different ways, decide what to do about each fault against a published standard, make those decisions run again next quarter without you, and hand the whole thing to an auditor with the working attached.

Read the full lesson, with runnable code

[https://data-analysis.cassion.dev/courses/data-cleaning-and-validation/
cleaning-08-the-cleaning-log](https://data-analysis.cassion.dev/courses/data-cleaning-and-validation/cleaning-08-the-cleaning-log)