

An annex of only successful analyses is the one nobody believes

The annex exists so a reviewer can check the report without rerunning it. That means it holds the tests that found nothing, the analyses that were run and dropped, and the count of comparisons — because a reviewer who cannot see the denominator cannot weigh the findings.

Pierre Robentz Cassion

Lesson 6 of 8

Dashboards and Reporting — The report around it

What this lesson covers

- What the annex is for
- The test table
- Exclusions, with counts
- Deviations from the plan
- The exploratory section, labelled
- What an annex should not contain
- The reproduction note
- Report it whole
- What comes next

What the annex is for

The annex holds	So a reviewer can
Every test run, including the null ones	Weigh the findings against the count
The definitions, or a link to them	Check the indicator is what they think
Deviations from the analysis plan	See that changes were decided, not discovered
The exclusions, with counts and reasons	Reconstruct the denominator
The code, or where it is	Rerun it

What the annex is for

- Everything else belongs in the report or nowhere

The test table — Example

Comparison	Test	Statistic	p	Effect
Completion by disability	two-sample z	$z = -5.21$	<0.001	-19.4 pts
Over-age by sex	two-sample z	$z = 1.84$	0.066	+5.5 pts
Attendance by sex	two-sample z	$z = 2.28$	0.023	+0.56 pts
Attendance by feeding	Welch t	$t = 3.11$	0.008	+5.2 pts
Over-age by sex, 24 schools	two-sample z x24	2 of 24 at $p < 0.05$		0 survive Bonferroni

Comparisons run in total: 29. Findings reported in the body: 2.

The test table

- Three of those five rows are in the annex precisely because they found nothing — A non-significant gap, a significant. . .
- The last line is the one that changes how the report is read — Two findings from twenty-nine comparisons is a. . .
- An annex containing only successful tests tells a reviewer the count was managed — and there is no way to un-signal that

Exclusions, with counts — In Python

```
audit = []
audit.append(("all rows", len(survey)))
survey = survey[survey["age_months"] <= 60]
audit.append(("age above 60 months", len(survey)))
survey = survey.dropna(subset=["muac_mm"])
audit.append(("no MUAC recorded", len(survey)))

for label, remaining in audit:
    print(f"{label:28} {remaining:>6,}")
```

Exclusions, with counts — In R

Build the audit trail as you filter, not from memory afterwards.

Exclusions, with counts

- Print the count after each step, in order — A reviewer who sees 4,218 in the report and 4,206 in the annex knows. . .
- The commonest silent exclusion is a join — An inner join that drops a third of the data looks exactly like one that. . .

Deviations from the plan — Example

Deviations from the analysis plan

Planned: primary analysis on all enrolled students.

Done: restricted to the 585 with both assessment rounds.

Why: 156 students have no endline. Attrition analysis added; those who left scored 39.3% at baseline against 56.8% for those who stayed, so the panel is not representative and this is a stated limitation.

Deviations from the plan

- A listed deviation is a strength and an unlisted one found by a reviewer is the end of the report's credibility — and...

The exploratory section, labelled

- Two headings, and the label is the whole mechanism

The exploratory section, labelled — Example

B.1 Pre-specified analyses

1 model, reported in the body

B.2 Exploratory analyses

3 models, generating hypotheses only

The exploratory section, labelled

- A subgroup finding is exploratory unless the subgroup was named in advance —
The statistics course found two. . .

What an annex should not contain

- Raw data — Especially not here
- Screenshots of code — The code, or the repository, or nothing
- Every chart that did not make the report — An annex is not a graveyard; a chart that carried no finding carries none in. . .
- Output nobody can read — A pasted regression summary with forty rows of software formatting is not evidence, it is. . .

The reproduction note — Example

Reproduction

Code: `github.com/<org>/nutrition-surveillance`, commit `a3f9c21`

Data: `muac-screening-artibonite-2024.v1.csv`, SHA-256 `4f2a9c1b8e...`

Environment: Python 3.12, dependencies pinned in `uv.lock`

```
uv sync && uv run python run.py
```

Running this on a clean checkout reproduces every number in this report and this annex. Verified 2026-03-14.

The reproduction note

- "Verified" with a date is the claim — The workflows course's whole argument lands here: an analysis nobody can rerun is. . .

Report it whole — Example

Annex contents

- A. Indicator definitions (linked; numerator, denominator, level, standard, decision informed, for nine indicators)
- B.1 Pre-specified analyses (1)
- B.2 Exploratory analyses (3, labelled, not presented as findings)
- C. All comparisons run: 29, with statistic, p and effect size. Two are reported as findings in the body.
- D. Exclusions: 4,218 screenings, 12 excluded for an implausible age, with the count after each step.
- E. Deviations from the analysis plan: 1, listed with its reason.
- F. Reproduction: repository, commit, data checksum, environment, and the two commands. Verified 2026-03-14.

Report it whole

- Six sections and none of them is long — An annex is judged on whether a reviewer can find the thing they doubt, not on...

What comes next

- The report is written and the annex holds the working.

Read the full lesson, with runnable code

[https://data-analysis.cassion.dev/courses/dashboards-and-reporting/
dash-06-the-annex-that-shows-the-working](https://data-analysis.cassion.dev/courses/dashboards-and-reporting/dash-06-the-annex-that-shows-the-working)