CASSION · COURSE

Indicator Design and the LogFrame

Write indicator definitions precise enough that two analysts computing them independently get the same number, and trace each one back to the change it is supposed to evidence.

Instructor	Pierre Robentz Cassion
Level	Intermediate
Learner hours	16 h
Taught in	Python · R
Updated	July 28, 2026
Sectors	Public Health · Emergency Response · Education
Standards and methodologies	Logical Framework Approach · Theory of Change · Results-Based Management (RBM) · OECD DAC evaluation criteria · UNICEF indicator definitions · Sustainable Development Goals (SDG)

Read and run the course online at data-analysis.cassion.dev/courses/indicator-design-logframe

What you will be able to do

- Trace an indicator back to the change it is meant to evidence, and say what the LogFrame lost on the way from the Theory of Change
- Write an indicator reference sheet complete enough that a second analyst reproduces your number without asking you a question
- Choose a denominator you can defend in a review, and state what your choice excludes
- Distinguish reach, coverage and cumulative counts, and stop a monthly series from counting the same person twelve times
- Adopt a standard indicator definition where one exists, and say precisely where yours departs from it
- Set a baseline and a target that survive a mid-term review, and name the behaviour each one will encourage

Prerequisites

None. This course assumes no prior programming experience.

Contents

About this course

This audience is judged on defending a number in a review, not on producing it. Module 2 got the data to a state where a number can be computed at all. This course is about which number, and why, and how to write it down so that the person who replaces you computes the same one.

The central artefact is the **indicator reference sheet**: a page per indicator carrying the numerator, the denominator, the disaggregation, the frequency, the source and — the field almost every template omits — the decision the number informs. An indicator that informs no decision is a reporting cost, and naming the decision is the fastest way to find out that you have several.

The recurring failure this course exists to prevent is the one the *Data Quality Assessment* course kept finding underneath everything else: two people computing different things under one indicator name. That is not a data quality problem and no amount of cleaning fixes it. It is a definition problem, and definitions are cheap to fix and permanent once fixed.

Both Python and R throughout, though this course has more prose and fewer pipelines than the rest of the module — the hard part of an indicator is rarely the arithmetic. Worked examples run against the DHIS2-shaped vaccination extract, where the penta1-to-penta3 dropout is 13.8% and every coverage figure rests on a denominator somebody projected from a 2015 census, and against the MUAC screening register, where 4,218 screenings are not 4,206 registrations and neither is the number of children.

You should have done *Data Analysis Foundations*. Module 2 is not a prerequisite, but every example assumes you already know why a missing report is not a zero.

Part I

From intention to indicator

CHAPTER 1

What breaks between the change and the number

Lesson 1 of 8 · 90 min

A Theory of Change says how the world is supposed to move; a LogFrame turns that into rows; an indicator turns a row into arithmetic. Something is lost at each translation, and knowing what lets you defend the number.

1.1 Three documents, two translations

Almost every programme in this sector carries three artefacts, produced by different people at different times, and the number you report comes out of the third.

- A **Theory of Change** — a narrative and usually a diagram, saying how the programme believes the world will move and under what assumptions.
- A **LogFrame** — a table of goal, outcomes, outputs and activities, each row carrying indicators, means of verification and assumptions.
- An **indicator definition** — arithmetic on a dataset.

Each arrow between them is a translation, and translations lose things. This lesson is about what specifically, because "the indicator does not measure what we care about" is a complaint that arrives too late to fix.

1.2 The first translation: change to row

A Theory of Change for a routine immunisation programme might say:

> If health workers are trained and vaccines are reliably in stock, then caregivers who bring a child for the first dose will complete the schedule, and fewer children will be susceptible to measles at school entry.

That sentence contains a mechanism, an assumption and a population. The LogFrame row it becomes usually says:

Level	Indicator
Output	Number of children receiving penta3

Three things went missing in one step.

The mechanism. The theory was about *completion* — children who start the schedule finishing it. A count of penta3 doses rises if more children start, if more complete, or if a campaign draws children in from a neighbouring district. The indicator cannot distinguish the thing the programme is actually claiming to do.

The assumption. "Vaccines reliably in stock" is a condition the theory depends on and no LogFrame row measures. When the indicator underperforms, the argument about why is unresolvable, because nobody was counting the thing the theory said mattered.

The population. "Children who are brought for a first dose" is a specific group. A dose count has no denominator at all.

1.3 Fix it at the LogFrame, not at the analysis

The completion mechanism is measurable, and it is a standard indicator:

```python
totals = (
    vax[vax["reported"]]
    .groupby("antigen")["doses_administered"].sum()
)
dropout = (totals["penta1"] - totals["penta3"]) / totals["penta1"]
print(f"penta1 {totals['penta1']:,}  penta3 {totals['penta3']:,}  dropout {dropout:.1%}")
```

```r
totals <- vax |>
  filter(report_submitted) |>
  summarise(doses = sum(doses_administered), .by = antigen)

p1 <- totals$doses[totals$antigen == "penta1"]
p3 <- totals$doses[totals$antigen == "penta3"]
c(penta1 = p1, penta3 = p3, dropout = (p1 - p3) / p1)
```

8,302 first doses, 7,160 third doses, **a dropout of 13.8%**. And the measles series is worse: 7,321 first doses against 5,647 second, a dropout of 22.9%.

That single number speaks directly to the theory in a way the dose count does not. It is also robust to the denominator argument — both numerator and denominator come from the same reporting facilities in the same months, so a census projection nine years old cannot move it.

When the mechanism is measurable, measure the mechanism. A dropout rate and a coverage figure answer different questions, and a LogFrame carrying only the second is describing the programme's output rather than its claim.

1.4 The second translation: row to arithmetic

The LogFrame says "number of children receiving penta3". The analyst has to turn that into code, and the row does not say:

- Children or **doses**? A dose count and a child count differ whenever a child is recorded twice, and the register does not distinguish them.
- Which **period** — by date of vaccination or date of report?
- Which **facilities** — those that reported, or all of them, with non-reporters counted as zero?
- **Catchment or residence?** A child from the next district vaccinated here.

Four questions, each with a defensible answer, and **two analysts answering them differently produce two different numbers from the same file**. Neither has made an error. This is the defect the whole course exists to prevent, and the fix is the reference sheet in unit 2.

1.5 Assumptions are the part everyone deletes

The right-hand column of a LogFrame is the one that gets emptied first, and it is the column that explains your results.

An assumption is a condition outside the programme's control that the causal chain depends on: vaccines arrive, the road is passable, the currency holds, the security situation permits movement. When a target is missed, the honest analysis is usually "the assumption failed", and that sentence is only available if somebody wrote the assumption down beforehand.

Give the important assumptions an indicator of their own. Stock-out days per quarter, road access days, staff vacancy rate. They are cheap to collect and they convert an argument into a fact.

1.6 What the levels mean, and where people cheat

Level	Question	Attribution
Activity	What did we do?	Fully ours
Output	What did that produce, directly?	Fully ours
Outcome	What changed for people?	Shared with everything else
Goal	What changed in the population?	Almost none of it ours

The universal temptation is to report an output and call it an outcome, because outputs are attributable and outcomes are not. "12,000 children vaccinated" is an output. "Measles susceptibility at school entry fell" is an outcome, it depends on four other actors, and attribution is exactly what the *Impact Evaluation Methods* course later spends twenty-four hours on.

Say which level you are at. A report that lists outputs under an outcome heading will be found out by any evaluator, and the finding is about credibility rather than about numbers.

1.7 Read the chain backwards as a test

The most useful five minutes in indicator design: start from your indicator and walk up.

> **Penta3 dropout rate** — measures whether children who start the schedule finish it — which the programme influences through reminders, outreach and stock — which the theory says reduces susceptibility — which is the change we want.

If the chain breaks, you have found something. It usually breaks in one of three places, and each has a different fix:

- **The indicator does not measure the mechanism.** Change the indicator.
- **The mechanism is not something the programme influences.** Change the programme's claim, or accept that you are monitoring context.
- **Nobody can name the change.** Stop and have that conversation before collecting anything.

> An indicator is a claim about causation compressed into arithmetic. The compression is lossy, and the reference sheet is where you record what was lost.

1.8 What comes next

You now know where an indicator comes from and what it inherits. The next lesson takes it apart — what an indicator is made of, why almost every one is a ratio, and the three

questions to ask before writing a single line of code against it.

CHAPTER 2

The anatomy of an indicator

Lesson 2 of 8 · 90 min

Four families, one unit of measure, a direction, and the three questions to answer before writing any code. Plus why 4,218 screenings, 4,206 registrations and the number of children are three different indicators.

2.1 What it is made of

Strip the framework language away and an indicator has five parts:

- **A population** — who or what is being counted.
- **A condition** — what has to be true of them to be in the numerator.
- **A denominator** — what they are counted against, unless it is a plain count.
- **A period** — over what stretch of time.
- **A unit of measure** — people, doses, episodes, percent, litres per person per day.

Miss any one and the indicator is ambiguous. Miss the last and it is unusable in a sentence, which is where it will end up.

2.2 Four families, and choosing between them

Family	Shape	Example	Watch
Count	A number of things	Children screened	No den
Proportion	Part over whole, both same units	Share of children with MUAC under 125 mm	Numera
Rate	Events over population-time	New admissions per 1,000 under-fives per month	The tim
Ratio / index	Two quantities not nested	Penta1-to-penta3 dropout; Food Consumption Score	Can exc

The commonest design error is reporting a **count** where the audience will read a **proportion**. "375 children referred" sounds like a lot or a little depending entirely on how many were screened, and the reader will supply their own denominator if you do not.

```python
screened = len(muac)
referred = muac["outcome"].str.startswith("referred").sum()

print(f"{referred} referred of {screened} screened = {referred / screened:.1%}")
```

```r
screened <- nrow(muac)
referred <- sum(startsWith(muac$outcome, "referred"))

sprintf("%d referred of %d screened = %.1f%%", referred, screened, 100 * referred /
    screened)
```

375 of 4,218, or 8.9%. **Report both.** The count is what a logistics officer orders supplies against; the proportion is what tells you whether this commune is worse than that one.

2.3 Three numbers that look like one

Here is the ambiguity that costs the most time in practice. The screening register supports three different "number of children" indicators.

```python
print("screening events:      ", len(muac))
print("distinct child_id:     ", muac["child_id"].nunique())
```

```r
c(events = nrow(muac), ids = dplyr::n_distinct(muac$child_id))
```

- **4,218 screening events.** What the community health workers did. The right numerator for workload, supply consumption and payment.
- **4,206 distinct identifiers.** Twelve forms were submitted twice on a poor connection, so twelve events are duplicates of an event that already exists.
- **About 4,200 children.** Six more children were re-registered under a new identifier — findable only by the record linkage the cleaning course taught, and never exactly knowable.

Three defensible numbers, one register, and the difference between the largest and the smallest is 0.4%. **The size of the gap is not the point.** The point is that "children screened" does not say which one, and the moment a donor compares your figure against a partner's, whichever of the three each of you chose determines whether you agree.

So the indicator name has to carry it: `screening_events`, `children_screened_deduplicated`. Never `children_screened` on its own.

2.4 The unit of measure belongs in the name

```python
indicators = {
    "children_screened_count": screened,
    "gam_prevalence_percent": round(100 * gam_cases / assessed, 1),
    "admissions_per_1000_under5_per_month": round(1000 * admissions / under5 / 12, 2),
}
```

```r
indicators <- list(
  children_screened_count              = screened,
  gam_prevalence_percent               = round(100 * gam_cases / assessed, 1),
  admissions_per_1000_under5_per_month = round(1000 * admissions / under5 / 12, 2)
)
```

Long names, and they are worth it. A column called `rate` in a spreadsheet emailed between four organisations will be multiplied by 100 by somebody, divided by 12 by somebody else, and compared against a figure with a different denominator by a third. The name is the cheapest defence available.

2.5 Direction: say which way is better

Every indicator needs a stated direction, because a surprising number are ambiguous.

- **Higher is better** — coverage, completion, attendance.
- **Lower is better** — dropout, defaulter rate, prevalence of acute malnutrition.

- **Neither, on its own** — referral counts. More referrals can mean better case finding or a deteriorating situation, and the indicator cannot tell you which.

That third category is larger than people expect and it is where dashboards go wrong: a red-amber-green traffic light applied to an indicator with no direction produces an alarm whose meaning nobody can state. **If you cannot say which way is better, the indicator needs a partner indicator, not a colour.**

2.6 Leading and lagging

A related distinction that decides whether an indicator is any use for management.

- **Lagging** — tells you what happened. GAM prevalence, cure rate, annual coverage. Accurate, essential for accountability, useless for steering, because by the time it moves the quarter is over.
- **Leading** — moves early and predicts. Stock-out days, defaulter rate in the first two weeks of treatment, consecutive-absence counts before a child drops out.

A monitoring system built entirely of lagging indicators reports faithfully on things nobody can now change. **Aim for at least one leading indicator per outcome**, and expect it to be noisier — that is the trade you are making.

2.7 Three questions before any code

Answer these in writing before opening an editor. They take five minutes and they prevent the rewrite.

1. What decision does this number inform? If the answer is "the donor asks for it", say so honestly and keep it cheap. If two different decisions come out, you have two indicators wearing one name.

2. What would make it move, other than the thing I care about? Coverage moves with the denominator, with reporting completeness, with population movement, and occasionally with vaccination. Listing the alternatives is how you know what to report alongside it.

3. Who else already computes this, and how? Almost always somebody does — UNICEF, WHO, the cluster, the ministry. Lesson 6 is entirely about that, and the answer changes your definition more often than not.

2.8 An indicator you cannot compute is not an indicator

The last check, and it kills more proposed indicators than any other.

Before it goes into a LogFrame, confirm the data exists: which system holds it, at what grain, how often, and who extracts it. An indicator whose means of verification is "programme records" is a promise nobody has checked.

```python
REQUIRED = {"child_id", "commune", "screening_date", "muac_mm", "oedema", "outcome"}
missing = REQUIRED - set(muac.columns)
assert not missing, f"indicator cannot be computed; missing {missing}"
```

```r
REQUIRED <- c("child_id", "commune", "screening_date", "muac_mm", "oedema", "outcome")
stopifnot(all(REQUIRED %in% names(muac)))
```

The *Data Quality Assessment* course found timeliness unmeasurable because a field was not in the export. That is this check, discovered eighteen months late.

> An indicator is a promise that a number can be produced, repeatedly, to the same definition. Everything else in this course is about keeping that promise.

2.9 What comes next

You have the anatomy. The next lesson turns it into the artefact this course is built around — the reference sheet, field by field, and the test that tells you whether yours is finished.

Part II

The reference sheet

CHAPTER 3

The reference sheet, field by field

Lesson 3 of 8 · 90 min

One page per indicator, thirteen fields, and a test that tells you when it is finished — hand it to a second analyst and see whether they get your number without asking a question.

3.1 The test the sheet has to pass

An indicator reference sheet is finished when **a competent analyst who has never met you can compute your number from it, from the raw data, without asking you a question.**

That is the only test worth applying, and it is brutal. Almost every reference sheet in circulation fails it at the same three points: the denominator source is named but not specified, the exclusions are not listed, and nobody wrote down what to do when a reporting unit is silent.

3.2 The fields

Field	What it holds
Name	Including the unit of measure
Definition	One sentence a non-analyst can read
Numerator	The exact condition, in words
Denominator	The exact population, and where it comes from
Exclusions	What is deliberately left out, and why
Unit of measure	Percent, count, per 1,000, litres per person per day
Direction	Higher is better, lower is better, or neither
Disaggregation	The cuts that will be reported, and the minimum cell size
Frequency	How often it is computed, and the reporting lag
Source	System, table, field. Not "programme records"
Method of computation	Enough that the arithmetic is unambiguous
Decision informed	What changes depending on the value
Known limitations	What the number cannot support

Thirteen fields, most of them a line. A sheet takes twenty minutes and it is never written again.

3.3 Worked: penta3 coverage

```
Name            penta3_coverage_percent_monthly
Definition      The share of the target infant population that received a third
                dose of pentavalent vaccine in the month.
Numerator       Third doses of pentavalent vaccine administered, as reported by
                facilities that submitted a monthly report for that month.
Denominator     Surviving infants in the catchment, from the MoH population
                estimates 2024, projected from the 2015 census at 2.4% annual
                growth, summed across facilities that reported.
Exclusions      Facility-months with report_submitted = false are excluded from
```

```
                  both numerator and denominator. Doses given to children outside
                  the catchment are not separable and remain in the numerator.
Unit              Percent
Direction         Higher is better
Disaggregation    Month, facility type, district. Minimum cell 30 in denominator.
Frequency         Monthly, reported 6 weeks after month end
Source            DHIS2 data element PENTA3_DOSES, org unit level 4, monthly
Computation       100 * sum(doses) / sum(target_population), over reporting
                  facility-months only. Not the mean of facility-level rates.
Decision          Whether a district is prioritised for outreach in the next
                  quarterly microplan.
Limitations       Reporting completeness was 76.5% in 2024 and fell to 29% in
                  August; the figure is a lower bound in any month where
                  completeness is below 90%. Catchments overlap, so facility-level
                  values are unreliable; use district level or above.
```

Read the exclusions and the computation rows together, because they are where the two-analyst test is usually failed.

"Over reporting facility-months only. Not the mean of facility-level rates." Those two sentences settle a difference of several points and neither is implied by the definition. A ratio of sums and a mean of ratios are different numbers, and both are reasonable readings of "coverage".

3.4 The field everyone omits

`Decision informed` is missing from most templates in use, and it is the field that does the most work.

Filling it in has three effects, and all three are uncomfortable in a useful way.

- **It exposes indicators that inform nothing.** Some of those are still required — a donor asks for them — and that is a legitimate answer. Write "donor reporting requirement, no operational decision" and stop spending analytical effort on it.
- **It exposes indicators that inform two decisions.** Coverage used both to prioritise outreach *and* to trigger a supply order needs two definitions, or one of the two decisions is being made on the wrong number.
- **It sets the precision you need.** An indicator that prioritises a district needs to be right about the ranking. One that triggers a supply order needs to be right about the level. Those are different requirements and they cost different amounts.

3.5 Keep it beside the code

A reference sheet in a Word file in somebody's mailbox is a reference sheet that drifts. Store it as data next to the script that computes the indicator.

```
import json
from pathlib import Path

sheet = json.loads(Path("indicators/penta3_coverage.json").read_text())

assert sheet["denominator_source"], "denominator source is required"
assert sheet["decision_informed"], "an indicator with no decision needs saying so"

numerator = vax.loc[vax["reported"] & (vax["antigen"] == "penta3"),
                    "doses_administered"].sum()
```

```python
denominator = vax.loc[vax["reported"] & (vax["antigen"] == "penta3"),
                      "target_population"].sum()

print(f"{sheet['name']}: {100 * numerator / denominator:.1f}")
```

```r
sheet <- jsonlite::read_json(here::here("indicators", "penta3_coverage.json"),
                             simplifyVector = TRUE)

stopifnot(nzchar(sheet$denominator_source), nzchar(sheet$decision_informed))

vax |>
  filter(report_submitted, antigen == "penta3") |>
  summarise(value = 100 * sum(doses_administered) / sum(target_population))
```

Two gains. The **assertions fail the build** when a sheet is incomplete, which is the same move the platform's own content schema makes. And the sheet ships with the result, so a table and its definitions travel together — the habit the foundations course introduced with a `definitions.csv` beside every output.

3.6 Write the exclusions as code, not as prose

The exclusions field is the one most likely to be true in the document and false in the script. Close the gap by generating one from the other.

```python
EXCLUSIONS = [
    ("non-reporting facility-months", lambda d: ~d["reported"]),
    ("other antigens", lambda d: d["antigen"] != "penta3"),
]

remaining = vax.copy()
for label, rule in EXCLUSIONS:
    dropped = rule(remaining).sum()
    remaining = remaining[~rule(remaining)]
    print(f"excluded {dropped:>5} rows: {label}")
print(f"{len(remaining)} rows in the indicator")
```

```r
EXCLUSIONS <- list(
  "non-reporting facility-months" = function(d) !d$report_submitted,
  "other antigens"                = function(d) d$antigen != "penta3"
)

remaining <- vax
for (label in names(EXCLUSIONS)) {
  rule <- EXCLUSIONS[[label]]
  cat(sprintf("excluded %5d rows: %s\n", sum(rule(remaining)), label))
  remaining <- remaining[!rule(remaining), ]
}
```

The printout is the exclusions field, with counts. Paste it into the sheet and the two cannot disagree.

3.7 Version it

An indicator definition changes. When it does, the series breaks, and the break has to be visible.

```
version   2.1
changed   2026-07-28
change    Denominator restricted to reporting facilities. Previously all
          facilities, with non-reporters counted at their target population,
          which understated coverage by about 23% in August 2024.
effect    Series revised from 2024-01. Values before v2.1 are not comparable.
```

Never silently improve a definition. A coverage figure that rises eight points because the definition changed, presented in the same chart as previous quarters, is the most convincing wrong finding an M&E system can produce.

3.8 The two-analyst test, in practice

Once a quarter, take an indicator, hand the sheet and the raw extract to a colleague who did not write it, and compare. It takes an hour and it finds things no review of the document does.

What it typically surfaces:

- The colleague used the calendar month; you used the reporting month.
- The colleague computed a mean of facility rates; you computed a ratio of sums.
- The colleague included the district hospital; your extract excluded it because of an org-unit level filter nobody documented.

Every one of those is a sheet that needed one more line. **The purpose of the exercise is to find the line, not to find out who was right.**

3.9 What comes next

The field that generates more disagreement than the other twelve combined is the denominator. The next lesson is entirely about choosing one, defending it, and saying honestly what it excludes.

CHAPTER 4

The denominator argument, and the cuts you promise

Lesson 4 of 8 · 90 min

Three denominators for one numerator, the rule that decides between them, and the disaggregation that turns twenty-four cells into forty-eight — three of which are too small to report.

4.1 Every argument about a number is an argument about the denominator

The numerator is usually agreed within a minute. Somebody counted the children with a MUAC below 125 mm and got 362, and nobody disputes it.

What follows is an hour about what to divide it by, and the reason the hour happens is that all the candidates are defensible.

4.2 Three denominators, one register

```python
measured = muac["muac_mm"].notna()
assessed = measured | muac["oedema"].notna()
cases = (muac["muac_mm"] < 125) | (muac["oedema"] == True)

for label, mask in [("all screened", pd.Series(True, index=muac.index)),
                    ("with a measurement", measured),
                    ("assessed either way", assessed)]:
    print(f"{label:22} n={mask.sum():5}  GAM={cases.sum() / mask.sum():.2%}")
```

```r
muac |>
  summarise(
    all_screened = n(),
    measured     = sum(!is.na(muac_mm)),
    assessed     = sum(!is.na(muac_mm) | !is.na(oedema)),
    cases        = sum(muac_mm < 125 | oedema, na.rm = TRUE)
  ) |>
  mutate(across(c(all_screened, measured, assessed), ~ cases / .x, .names = "gam_{.col}"))
```

Denominator	n	GAM
Everyone who came	4,218	8.58%
Children with a MUAC measurement	4,146	8.73%
Children assessed by measurement or oedema	4,216	8.59%

Three numbers, 0.15 points apart, and all three are in use in real reports.

The gap is small here, which is exactly why the lesson is worth learning on this file rather than on one where the answer is obvious. **The first denominator is wrong, and it is wrong in a way that does not show up as a large difference.** Dividing by everyone who came treats the 72 unmeasured children as though they had been measured and found well nourished. That is a claim about children nobody looked at.

4.3 The rule

> **The denominator is the population that was genuinely at risk of being in the numerator, over the same period, in the same place.**

Apply it and the answer here falls out. A child with no measurement and no oedema assessment could not have entered the numerator whatever their nutritional status, so they do not belong in the denominator. The second row is the defensible one, and the third is defensible if you count an oedema assessment as sufficient.

The same rule settles most of the arguments you will have:

- **Coverage.** Not everyone in the district — the target age group in the catchment, over the period.
- **Cure rate.** Not everyone admitted — those who reached an outcome. Children still in treatment at the cut-off are neither cured nor defaulted.
- **Referral completion.** Not all cases — cases that consented to be referred, which the protection dataset makes explicit and most systems do not.
- **Attendance.** Not enrolled children times all school days — enrolled children times the days their school was open.

4.4 Say what it excludes, in the sheet

Whichever you pick, the exclusion is a claim and it goes in writing.

```
Denominator   Children with a MUAC measurement recorded (n = 4,146 of 4,218).
Excludes      72 children (1.7%) screened but not measured, coded -99. Their
              nutritional status is unknown; if they were systematically the
              most distressed children, GAM is understated. Not testable from
              this register.
```

That last sentence is the one that makes the number defensible. You have named a direction of possible bias and said you cannot resolve it, which is a stronger position than any figure presented without it.

4.5 Disaggregation is a promise about sample size

A LogFrame that says "disaggregated by sex, age and district" has committed to producing cells, and cells have counts.

```
banded = muac[muac["age_months"].notna()].assign(
    band=lambda d: (d["age_months"] >= 24).map({True: "24-59", False: "6-23"})
)

by_two = banded.groupby(["commune", "band"]).size()
by_three = banded.groupby(["commune", "band", "sex"]).size()

print(f"commune x band:       {len(by_two)} cells, smallest {by_two.min()}")
print(f"commune x band x sex: {len(by_three)} cells, smallest {by_three.min()}, "
      f"{(by_three < 30).sum()} below 30")
```

```r
banded <- muac |>
  filter(!is.na(age_months)) |>
  mutate(band = if_else(age_months >= 24, "24-59", "6-23"))

banded |> count(commune, band) |> summarise(cells = n(), smallest = min(n))
banded |> count(commune, band, sex) |> summarise(cells = n(), smallest = min(n),
                                                 under_30 = sum(n < 30))
```

Disaggregation	Cells	Smallest cell	Cells under 30
Commune × age band	24	49	0
Commune × age band × sex	48	16	3

Adding one binary cut doubles the cells and halves their size. Going from two dimensions to three takes the smallest cell from 49 children to 16, and puts three cells below any reasonable reporting threshold.

A prevalence computed on 16 children moves by six percentage points when one child changes category. Publishing it in a table alongside a commune-level figure computed on 400 invites a reader to compare them as though they were the same kind of number.

4.6 Set a minimum cell size, in the sheet, in advance

```python
MIN_CELL = 30

table = (
    banded.assign(case=cases)
    .groupby(["commune", "band", "sex"])
    .agg(n=("case", "size"), cases=("case", "sum"))
)
table["rate"] = (table["cases"] / table["n"]).where(table["n"] >= MIN_CELL)
table["note"] = table["n"].lt(MIN_CELL).map({True: "suppressed: n < 30", False: ""})
```

```r
MIN_CELL <- 30

table <- banded |>
  summarise(n = n(), cases = sum(case), .by = c(commune, band, sex)) |>
  mutate(rate = if_else(n >= MIN_CELL, cases / n, NA_real_),
         note = if_else(n < MIN_CELL, "suppressed: n < 30", ""))
```

Two properties matter. The count `n` **stays visible** even where the rate is suppressed, so a reader can see the cell exists and why it is empty. And the threshold was set before the numbers were seen, which is the same discipline the DQA course applied to tolerance bands.

In protection and GBV data this stops being a statistical nicety and becomes a disclosure control, which the *Protection and GBV Data* course covers properly. The habit is the same and it is worth having before you need it.

4.7 Which cuts actually change a decision

The last question, and it is a budget question as much as an analytical one. Every disaggregation you promise has to be collected, cleaned, computed and checked, and most LogFrames promise more than anyone uses.

Ask of each cut: **would the programme do something different if this cut showed a gap?**

- **Sex** — yes, almost always. It changes targeting, staffing and messaging.
- **Age band** — yes for nutrition, where admission criteria differ by age.
- **District** — yes, it moves resources.
- **Facility type** — sometimes; it changes supervision, which is real.
- **Month** — usually a trend, not a disaggregation, and cheaper as a chart.

A cut that would change nothing is a reporting cost with a data quality risk attached, because every additional required field is another field that arrives empty.

4.8 What comes next

You can now define one indicator precisely. The next unit is about the family of indicators that count people, where the definitional problem is different: the same person appearing in twelve monthly reports, and what happens when somebody adds those reports up.

Part III

Counting people

CHAPTER 5

Reach, coverage, and the people counted twelve times

Lesson 5 of 8 · 90 min

Three indicators that all sound like "how many people did we help", and the monthly series that turns 4,200 children into 50,000 the moment somebody sums the column.

5.1 Three questions that sound like one

"How many people did the programme help?" has three answers, and a proposal, a donor report and a coverage figure each need a different one.

- **Reach** — how many *distinct people* received something, over a stated period.
- **Service volume** — how many *services* were delivered. Larger than reach whenever anyone comes twice.
- **Coverage** — what *share* of those who needed it got it. Requires a denominator and is the only one that says whether the programme is enough.

The names are used interchangeably in practice, which is how a programme reports 50,000 children reached in a district with 4,200 of them.

5.2 The cumulative sum that manufactures people

Here is the mechanism, and it is the single most common way this sector overstates its work.

```python
monthly = (
    muac.assign(month=muac["screening_date"].dt.strftime("%Y-%m"))
    .groupby("month")
    .agg(screenings=("child_id", "size"),
         distinct_children=("child_id", "nunique"))
)
monthly["cumulative_naive"] = monthly["screenings"].cumsum()

print(monthly)
print("sum of monthly counts:  ", monthly["screenings"].sum())
print("distinct children, year:", muac["child_id"].nunique())
```

```r
monthly <- muac |>
  mutate(month = format(screening_date, "%Y-%m")) |>
  summarise(screenings = n(), distinct_children = n_distinct(child_id), .by = month) |>
  arrange(month) |>
  mutate(cumulative_naive = cumsum(screenings))

c(sum_of_months = sum(monthly$screenings),
  distinct_year = n_distinct(muac$child_id))
```

On this register the two happen to be close, because it is a one-round campaign and few children were screened twice. **That is the exception.** In a monthly programme — outpatient therapeutic care, a cash transfer, a health facility — a beneficiary appears in every

month they receive something, and twelve monthly counts summed is twelve times a stable caseload.

The arithmetic that goes wrong:

	Correct	Wrong
Monthly reach, January	distinct people in January	—
Annual reach	distinct people across the year	sum of twelve monthly counts
Cumulative reach to date	distinct people since programme start	running total of monthly counts

A cumulative reach figure can only be computed from person-level data. If your reporting is monthly aggregates, you cannot deduplicate across months, and the honest annual figure is not available — you must either say so or report the maximum month rather than the sum.

```python
annual_reach = muac["child_id"].nunique()
service_volume = len(muac)
print(f"reach {annual_reach:,} children; {service_volume:,} screening events")
```

```r
c(reach = n_distinct(muac$child_id), volume = nrow(muac))
```

5.3 Deduplicating across months, when you can

```python
first_contact = (
    muac.sort_values("screening_date")
    .drop_duplicates(subset=["child_id"], keep="first")
    .assign(month=lambda d: d["screening_date"].dt.strftime("%Y-%m"))
)
new_by_month = first_contact.groupby("month").size()
print(new_by_month.cumsum())
```

```r
first_contact <- muac |>
  arrange(screening_date) |>
  distinct(child_id, .keep_all = TRUE) |>
  mutate(month = format(screening_date, "%Y-%m"))

first_contact |> count(month) |> arrange(month) |> mutate(cumulative = cumsum(n))
```

Counting each person in the month of their **first** contact gives a cumulative series that rises to the true reach and never exceeds it. It also produces a genuinely useful management number — *new* beneficiaries per month, which distinguishes a programme that is growing from one that is serving the same people repeatedly.

Report both, always: **new this month** and **total served this month**. They answer different questions and either alone is misleading.

5.4 Reach is not coverage, and the difference is the denominator

Reach counts who you served. Coverage says what fraction of those who needed the service got it, and it is the only one of the three that can say a programme is insufficient.

```python
reached = muac["child_id"].nunique()
under5_population = 21500      # from the administrative frame, projected

print(f"screening coverage: {reached / under5_population:.1%}")
```

```r
c(coverage = n_distinct(muac$child_id) / 21500)
```

Two failures to avoid, and both are common enough to have names.

Admissions over expected caseload is not coverage. Dividing the children admitted to treatment by the caseload you expected tells you how good your expectation was, not what share of malnourished children you reached. Real coverage needs the number of malnourished children, which needs a survey — which is why coverage surveys exist as a separate methodology.

A reach figure with a population denominator is not coverage either unless the population is the population *in need*. Screening 4,200 of 21,500 under-fives is screening coverage. It says nothing about what share of the malnourished children were found.

5.5 Cumulative counts in proposals

Multi-year proposals ask for "people reached over the life of the project", and this is where the double-counting becomes a compliance issue rather than a methods one.

Three rules that keep it honest:

- **State the deduplication level.** "Unique individuals, deduplicated within year, not across years" is a defensible and common compromise. Undocumented deduplication is not.
- **Never sum reach across sectors.** A household receiving water, food and a protection service is one household, and adding three sector reach figures triple-counts it. Report by sector, and give a deduplicated total only if you can actually compute one.
- **Say when you cannot.** "Aggregate figures are not deduplicated across partners; the true unique reach is lower" is a sentence donors read constantly and respect. The alternative is a number that collapses under audit.

5.6 Which one belongs in your LogFrame

If the row is about	Use
Programme effort and cost	Service volume
People served	Reach, with the deduplication level stated
Whether the response is sufficient	Coverage, with a needs-based denominator
Whether people who start finish	A completion or dropout rate

The last row is the one most often missing, and it is frequently the most informative — the *Theory of Change* lesson made that case with the 13.8% penta1 to penta3 dropout, which no reach or coverage figure would have revealed.

> Reach tells you what you did. Coverage tells you whether it was enough. A report with only the first is a description of activity, and every reader knows it.

5.7 What comes next

Almost every indicator in this lesson already has an official definition published by somebody — UNICEF, WHO, the cluster, the SDG framework. The next lesson is about finding it, adopting it, and saying precisely where yours departs from it.

CHAPTER 6

Somebody has already defined this

Lesson 6 of 8 · 90 min

Where the published definitions live, a rule for choosing between adopting one and writing your own, and how to document a departure so your number stays comparable.

6.1 The question to ask before writing a definition

Almost every indicator this sector reports has already been defined by somebody with more time, more consultation and more scrutiny than you will get. Finding that definition takes twenty minutes and it changes what you write.

The reason is not deference. It is that **a number nobody can compare to anything answers half the question.** A GAM prevalence that cannot be read against the IPC phase thresholds, or a coverage figure computed differently from the ministry's, has thrown away most of its value in exchange for local convenience.

6.2 Where the definitions live

Source	Covers	Note
WHO	Growth standards, immunisation, disease surveillance case definitions	The growth standa
UNICEF	Child protection, WASH, education, nutrition indicator definitions	Usually the most c
Sphere	Minimum standards and their indicators across humanitarian sectors	Gives thresholds, r
The SDG framework	231 global indicators with metadata sheets	The metadata she
IPC	Acute food insecurity and acute malnutrition classification	Defines the eviden
SMART	Nutrition survey methodology and plausibility criteria	Defines how the da
JMP	Water, sanitation and hygiene service ladders	The ladders are in
The cluster	Whatever the response is reporting on this year	Least stable, most

Two of those deserve emphasis for this audience. **The JMP service ladders are definitions, not categories** — "basic" water service is an improved source within a 30-minute round trip, and a programme classifying on source alone is not reporting the JMP indicator whatever it calls it. And **the SDG metadata sheets are the best free model of a reference sheet you will find**; read two of them before writing your own.

6.3 Adopt, adapt, or invent

A rule that resolves most cases:

Adopt when a standard definition exists and you can compute it. This is the default and it should cover most of your LogFrame. Adopting means adopting the whole thing — numerator, denominator, age bands, recall period — not the name.

Adapt when the standard definition exists but you genuinely cannot compute it, usually because a data element is missing. Adaptation is legitimate and it must be documented, because an adapted indicator carrying the standard name is the worst of both worlds.

Invent only when the thing you need to measure is specific to your programme and nobody else measures it. Rarer than people think. Before inventing, check whether the thing you want is really a *disaggregation* of a standard indicator, which it usually is.

6.4 Documenting a departure

The reference sheet gets two more fields when you depart from a standard.

```
Standard          WHO/UNICEF, penta3 coverage (WUENIC methodology)
Departure         Denominator uses MoH facility catchment estimates rather than
                  UN WPP surviving infants. Numerator restricted to facilities
                  reporting in the month, where the standard uses all facilities
                  with non-reporters imputed from the previous year.
Effect            Our figure is a lower bound relative to the standard. On 2024
                  data the difference is largest in August, where reporting
                  completeness was 29%.
Comparable to     Other districts using the same MoH denominators. Not directly
                  comparable to WUENIC national estimates.
```

Four lines, and the last one is the one a reader needs most. **Say what your number *can* be compared to.** Most misuse of an indicator is comparison against something incomparable, and it is almost always done in good faith by somebody who had no way to know.

6.5 Map to the standard where you can

Where you have person-level data, you can often compute both your definition and the standard one and publish the pair.

```python
DEFINITIONS = {
    "local_gam_muac": lambda d: (d["muac_mm"] < 125) | (d["oedema"] == True),
    "who_gam_muac_only": lambda d: d["muac_mm"] < 125,
}

for name, rule in DEFINITIONS.items():
    eligible = d = muac[muac["muac_mm"].notna()]
    print(f"{name:22} {rule(d).sum() / len(eligible):.2%}  (n={len(eligible)})")
```

```r
muac |>
  filter(!is.na(muac_mm)) |>
  summarise(
    local_gam_muac    = mean(muac_mm < 125 | oedema),
    who_gam_muac_only = mean(muac_mm < 125),
    n = n()
  )
```

Publishing both costs two lines of code and settles a question that would otherwise take an email thread. It also makes the effect of your departure a measured quantity rather than an assertion — which is the same move the cleaning log makes for every cleaning rule.

6.6 Comparability is the whole point

Three places where a locally-invented definition costs you concretely:

- **Against yourself, over time.** A definition changed between rounds breaks the series, and the break is invisible unless versioned.
- **Against your neighbours.** Cluster reporting aggregates partner figures. A partner using a different denominator makes the cluster total meaningless, and nobody finds out.

- **Against the threshold that triggers action.** IPC phases, Sphere minimum standards and the 15% GAM emergency threshold are all defined against specific indicator definitions. A GAM computed differently cannot be read against the threshold, and reading it anyway is how a response gets scaled on a number that does not mean what the threshold means.

That last one is not hypothetical. MUAC-based and weight-for-height-based GAM give different prevalences on the same children, and the IPC acute malnutrition thresholds are defined for specific measures. **Name the measure in the indicator name** — `gam_muac_percent` and `gam_whz_percent` are different indicators that both get called "GAM".

6.7 SDG tiers, and what they tell you

The SDG framework classifies its indicators into tiers, and the classification is useful well beyond the SDGs.

- **Tier 1** — definition agreed, data regularly produced by most countries.
- **Tier 2** — definition agreed, data not regularly produced.
- **Tier 3** — no agreed definition yet. *(Formally retired as a tier, but the situation it described is everywhere.)*

If your indicator is tier 3 in substance — no agreed definition — you are inventing, and you should expect to spend time defending it in every review. That is sometimes the right call. It is never the cheap one, and knowing which tier you are in tells you how much documentation the indicator will need for the rest of its life.

6.8 When the standard is wrong for you

Occasionally it genuinely is, and adopting it uncritically is its own failure.

- The **age band** does not match your programme. A standard defined for under-fives in a programme serving 6 to 23 months needs the standard computed on the standard's band and yours computed on yours, both published.
- The **recall period** does not fit the context. A seven-day food consumption recall in a context with a fortnightly distribution cycle will oscillate.
- The **denominator is unavailable**, and the standard assumes a census that does not exist for your area.

In each case, adapt and document. What you must not do is keep the standard's name, change its guts, and let a reader assume comparability that is not there.

> Adopting a standard definition is not a loss of analytical independence. It is what makes your number legible to everyone who has to act on it.

6.9 What comes next

You have a definition, a denominator and a standard to compare it against. The last unit is about the two numbers placed beside it — the baseline it started from and the target it is supposed to reach — and what each of them does to the behaviour of the people being measured.

Part IV

Judgement

CHAPTER 7

Baselines, targets, and what they make people do

Lesson 7 of 8 · 90 min

Two numbers that turn an indicator into a commitment. Four ways to set a target, the baseline measured six months after the programme started, and the fact that every target changes the behaviour of the people being measured.

7.1 An indicator with a target is a different object

On its own, an indicator is a measurement. Put a baseline and a target beside it and it becomes a commitment, a performance judgement and — this is the part people underestimate — an instruction to everyone whose work it measures.

This lesson is about setting both honestly, and about the behaviour each one creates.

7.2 What a baseline has to be

A baseline is the value of **the same indicator, computed the same way, before the intervention**. Every word in that sentence is load-bearing.

- **Same indicator.** A baseline from a survey and an endline from routine data are not comparable. This is the commonest baseline failure and it is usually discovered at the evaluation.
- **Same way.** Same denominator, same age bands, same recall period. Version the reference sheet at baseline and keep it.
- **Before.** A baseline collected in month six measures a population the programme has already been working with.

```
baseline = {
    "indicator": "penta3_coverage_percent_monthly",
    "value": 61.4,
    "source": "DHIS2, 2023 monthly mean over reporting facilities",
    "collected": "2024-01",
    "reference_sheet_version": "2.1",
    "note": "Computed on the same denominator as the target. 2023 reporting "
            "completeness was 71%, so the baseline is a lower bound.",
}
```

```
baseline <- list(
  indicator = "penta3_coverage_percent_monthly",
  value = 61.4,
  source = "DHIS2, 2023 monthly mean over reporting facilities",
  reference_sheet_version = "2.1"
)
```

The `reference_sheet_version` field is what makes a baseline survive three years. Without it, the endline analyst has a number and no way to know what it meant.

7.3 The baseline you did not collect

Frequently there isn't one, because the programme started before anyone thought about measurement. Three honest options, in order of preference:

- **Reconstruct from routine data.** Usually possible and usually the best answer. Say which months and what the reporting completeness was.
- **Use a comparable external estimate** — a DHS or MICS round, a previous SMART survey — and state the difference in method explicitly.
- **Declare it unavailable and set a target on a trajectory instead**, so performance is judged on the direction rather than on a distance from a number nobody has.

What you must not do is retrofit a baseline from the first period of programme data and present it as a pre-intervention value. An evaluator will find it, and the finding will be about honesty rather than about method.

7.4 Four ways to set a target

Method	Basis	Best when
Normative	A standard says so — Sphere 15 L/person/day, 90% coverage	The standard is the point of the prog
Historical	Best recent performance, or best comparable district	Routine data exists and is trusted
Capacity	What the resources can deliver, costed	Constraints are the binding factor
Negotiated	What the donor and government agreed	Honest to name this as what it is

Most real targets are the fourth, presented as the first. That is not necessarily wrong — a negotiated target is a legitimate commitment — but writing "negotiated" in the sheet changes the review conversation from "you failed" to "we agreed a number that assumed X, and X did not happen".

Whichever method, **write down the assumption the target rests on**. A coverage target of 85% assumes the outreach budget, the vehicle and the vaccine supply. All three are assumptions from lesson 1, and when the target is missed they are the analysis.

7.5 Milestones, not just an endpoint

A single target three years out gives you nothing to manage against for two and a half of them.

```python
import numpy as np

baseline_value, target_value, quarters = 61.4, 85.0, 12
trajectory = np.linspace(baseline_value, target_value, quarters + 1)[1:]
print([round(v, 1) for v in trajectory])
```

```r
baseline_value <- 61.4; target_value <- 85.0; quarters <- 12
round(seq(baseline_value, target_value, length.out = quarters + 1)[-1], 1)
```

A straight line is a placeholder, not a plan. Two adjustments make it realistic:

- **Front-load nothing.** Programmes are slow to start. A trajectory flat for two quarters and steeper afterwards is more honest and less punishing at the first review.

- **Respect seasonality.** In an agricultural or malaria context, comparing Q3 to Q2 is comparing two different worlds. Set milestones against the same quarter of the previous year.

7.6 Every target changes behaviour

This is the part that belongs in an M&E course rather than a management one, because you are the person who will see it in the data first.

When a number is used to judge, the people being judged optimise the number. Sometimes that is exactly what you wanted. Often it is not, and the ways it goes wrong are predictable:

- **Cream-skimming.** A cure rate target encourages admitting the least severe cases, which raises the rate and reduces the programme's value.
- **Boundary effects.** A target of 85% produces a suspicious number of reported values between 85% and 87%, and almost none at 84%.
- **Effort reallocation.** Resources move from unmeasured activities to measured ones, including from things that matter more.
- **Definition drift.** The definition quietly widens until the target is met. This is the one that looks like data quality and is not.

Pair every target with an indicator that would move in the opposite direction if it were being gamed. A cure rate target beside a mean admission severity; a coverage target beside the reporting rate; a caseload target beside the verification factor.

```python
watch = pd.DataFrame({
    "target_indicator": ["cure_rate_percent", "penta3_coverage_percent"],
    "gaming_risk": ["admit less severe cases", "widen the catchment definition"],
    "partner_indicator": ["mean_muac_at_admission_mm", "reporting_rate_percent"],
})
```

```r
watch <- tibble::tribble(
  ~target_indicator,         ~gaming_risk,                     ~partner_indicator,
  "cure_rate_percent",       "admit less severe cases",        "mean_muac_at_admission_mm",
  "penta3_coverage_percent", "widen the catchment definition", "reporting_rate_percent"
)
```

Write that table when the target is set, not when the number looks too good. It takes ten minutes and it is the difference between noticing in March and noticing at the evaluation.

7.7 Look for the boundary effect

Once a target exists, it is worth checking whether the distribution knows about it.

```python
reported_rates = performance["cure_rate_percent"]
near = reported_rates.between(85, 87).sum()
just_below = reported_rates.between(82, 84).sum()
print(f"{near} sites just above target, {just_below} just below")
```

```r
c(just_above = sum(dplyr::between(performance$cure_rate_percent, 85, 87)),
  just_below = sum(dplyr::between(performance$cure_rate_percent, 82, 84)))
```

A pile-up immediately above a threshold with a hole immediately below it is one of the most reliable signals in performance data. It is the same logic as the digit preference check in the DQA course, and it deserves the same care in wording: it is a pattern that warrants a question, not a conclusion about anyone.

7.8 Revising a target

Targets get revised. Do it in the open.

```
version    1.0 -> 1.1
changed    2026-07-28
from       85% by Q12
to         78% by Q12
reason     Outreach budget cut by 40% in Q3; the original target assumed the
           full outreach schedule. Trajectory recomputed from Q5 actuals.
approved   Programme manager and donor focal point, 2026-07-25
```

A revised target with a reason and an approval is a management decision. A target quietly edited in a spreadsheet is the finding that ends a partnership.

7.9 What comes next

You have an indicator, a baseline and a target. The final lesson is what happens when somebody external applies the OECD DAC criteria to all three — what an evaluator asks, which of your numbers survive the question, and how to have built the evidence before they arrive.

CHAPTER 8

What an evaluator does to your numbers

Lesson 8 of 8 · 75 min

Six OECD DAC criteria, the question each one asks of your indicator set, and the gaps you can only close before the evaluation rather than during it.

8.1 The review is the test your indicators were built for

At some point somebody external — a mid-term review, an end-of-project evaluation, a donor's monitoring adviser — takes your LogFrame, your reference sheets and your reported figures, and asks whether they support what the programme claims.

Most of what determines the answer was decided eighteen months earlier, in choices this course has already covered. This lesson is the checklist read backwards: what will be asked, and what you need to have already done.

8.2 The six criteria, and the question each asks of your data

The OECD DAC criteria are the vocabulary almost every evaluation in this sector uses.

Criterion	The question	What it needs from you
Relevance	Was this the right thing to do?	Needs assessment data, and indicators that measure the ne
Coherence	Did it fit with everything else?	Comparable definitions — the standards lesson, cashed in
Effectiveness	Did it achieve its objectives?	Baseline, target, and an outcome indicator that is not an o
Efficiency	Was it a reasonable use of resources?	Reach or volume with a denominator, and cost data joined
Impact	What difference did it make overall?	A counterfactual, or an honest statement that you do not k
Sustainability	Will the benefits last?	Something measured after handover, which almost nobody

Read down the right-hand column and you can predict which criteria your programme will score badly on, before anyone visits.

8.3 Effectiveness: where the reference sheet pays for itself

This is the criterion evaluations spend the most time on, and it is answered almost entirely from the artefacts of unit 2.

An evaluator will ask, in this order:

1. **What was the target, and where did it come from?** The four methods from the previous lesson. "Negotiated" is an acceptable answer; a blank is not.
2. **What was the baseline, measured how, and when?** Same indicator, same method, before.
3. **Is the reported value computed the same way as the baseline?** Version the sheet or you cannot answer this.
4. **What else could explain the change?** The alternatives you listed when answering question two of lesson 2.

```python
evidence = pd.DataFrame({
    "indicator": ["penta3_coverage_percent_monthly"],
    "baseline": [61.4],
    "baseline_period": ["2023 mean"],
    "target": [85.0],
    "target_basis": ["negotiated, assumed full outreach budget"],
    "latest": [72.8],
    "sheet_version": ["2.1"],
    "same_method": [True],
    "confounders": ["reporting completeness rose from 71% to 77% over the period"],
})
```

```r
evidence <- tibble::tribble(
  ~indicator,                        ~baseline, ~target, ~latest, ~sheet_version, ~same_
    method,
  "penta3_coverage_percent_monthly",      61.4,    85.0,    72.8,          "2.1",
    TRUE
)
```

The `confounders` field is the one that turns a defensive conversation into a professional one. Reporting completeness rising from 71% to 77% raises measured coverage without a single additional child being vaccinated. **Saying that yourself, first, is worth more than any number in the row.**

8.4 Impact: say what you cannot claim

The impact criterion asks what difference the programme made, which is a counterfactual question. Routine monitoring data cannot answer it, and pretending otherwise is the fastest way to lose an evaluator's confidence in everything else you wrote.

What you can honestly offer:

- **A before-and-after change**, labelled as such, with the alternative explanations named.
- **A comparison against a non-programme area**, with an explicit statement of why the two are or are not comparable.
- **Consistency with the theory of change** — the intermediate steps moved in the order the theory predicted, which is evidence and not proof.

What you cannot offer without a design that supports it is attribution. *Impact Evaluation Methods*, later in the programme, is entirely about the designs that do — randomisation, difference-in-differences, matching, regression discontinuity — and the honest position until then is the one stated plainly.

8.5 Efficiency needs a denominator you probably do not have

Efficiency questions are usually cost per unit of result, and they fail on the join rather than on the arithmetic: financial data is by budget line and month, programme data is by activity and facility, and nothing connects them.

If efficiency is going to be assessed, decide the unit at design time and make the two systems share a key. Retrofitting it at evaluation produces a number nobody believes, including the person who computed it.

8.6 The four gaps you can only close beforehand

By the time the evaluator arrives, these are fixed. Every one of them is cheap eighteen months earlier and impossible on the day.

- **No baseline on the same definition.** Fixed by versioning the sheet at the start.
- **No data on the assumptions.** Fixed by giving the two or three critical assumptions their own indicators.
- **No outcome indicator, only outputs.** Fixed at LogFrame design, and lesson 1 is about spotting it.
- **No post-handover measurement.** Fixed by budgeting one follow-up round, which almost no proposal does and every sustainability section needs.

8.7 Build the evidence pack as you go

The artefact that makes a review straightforward is a folder, maintained continuously, not assembled in the fortnight before the visit.

```
evidence/
  indicators/               reference sheets, versioned
  baseline/                 values, method notes, extract date
  targets/                  values, basis, revision history
  series/                   every period, every indicator, one row each
  dqa/                      assessment rounds and follow-ups
  assumptions/              stock-outs, access days, vacancy rates
  limitations.md            what the data cannot support
```

Six of those seven come out of work this module already required. The reference sheets are unit 2, the DQA rounds are the previous course, the series is the analysis table from the joining course. **The evidence pack is mostly a naming convention over things you already have.**

8.8 The one-page summary an evaluator actually wants

```python
summary = (
    evidence[["indicator", "baseline", "target", "latest", "same_method"]]
    .assign(progress=lambda d: (d["latest"] - d["baseline"])
                               / (d["target"] - d["baseline"]))
)
print(summary.round(2))
```

```r
evidence |>
  mutate(progress = (latest - baseline) / (target - baseline)) |>
  select(indicator, baseline, target, latest, progress, same_method)
```

Progress against trajectory, per indicator, with a flag saying whether the method was constant. Anything more elaborate gets rebuilt by the evaluator anyway; anything less gets rebuilt by the evaluator from your raw files, with their assumptions rather than yours.

> An evaluation is not an examination of your programme. It is an examination of whether your evidence supports what your programme said. Those come apart more often than anyone expects, and the reference sheet is where they are held together.

8.9 Where this course leaves you

You can trace an indicator from the change it is meant to evidence down to the arithmetic, write a reference sheet a stranger can compute from, defend a denominator, distinguish reach from coverage, adopt a standard definition and document your departure from it, set a baseline and a target that survive scrutiny, and assemble the evidence a review will ask for.

The rest of module 3 takes two specific measurement problems seriously. *Survey Analysis, Sampling and Weighting* is next and is the heaviest course in the programme so far — weights, strata, design effect and confidence intervals, which is what turns an estimate into an estimate with a margin. After it, *Routine Data and DHIS2* goes back to the aggregate systems this course kept borrowing from, and answers the question that always follows a coverage figure: what exactly was counted?

About this edition

This PDF is generated from the same source as the web version of the course, so the two cannot drift. The web edition carries the runnable code, the downloadable datasets and any corrections issued since this file was produced.

> Every dataset referenced in this course is synthetic or fully de-identified. No record traces to a real person.

This course is published in English and in French. The French edition is a professional adaptation using the terminology UNICEF, WHO, WFP, OCHA and national ministries publish in — not a literal translation.

Read and run the course online at data-analysis.cassion.dev/courses/indicator-design-logframe

Generated July 28, 2026.