Baselines, targets, and what they make people do

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.

Pierre Robentz Cassion

Lesson 7 of 8

Indicator Design and the LogFrame — Judgement

What this lesson covers

- An indicator with a target is a different object
- What a baseline has to be
- The baseline you did not collect
- Four ways to set a target
- Milestones, not just an endpoint
- Every target changes behaviour
- Look for the boundary effect
- Revising a target
- What comes next

An indicator with a target is a different object

- On its own, an indicator is a measurement.

What a baseline has to be

- **Same indicator.** A baseline from a survey and an endline from routine data are not comparable. This is the commonest...
- **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.

What a baseline has to be — In Python

```
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.",
}
```

What a baseline has to be — In R

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

The baseline you did not collect

- **Reconstruct from routine data.** Usually possible and usually the best answer. Say which months and what the…
- **Use a comparable external estimate** — a DHS or MICS round, a previous SMART survey — and state the difference in…
- **Declare it unavailable and set a target on a trajectory instead**, so performance is judged on the direction rather…

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 programme
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

Milestones, not just an endpoint — In 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])
```

Milestones, not just an endpoint — In R

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

Milestones, not just an endpoint

- **Front-load nothing.** Programmes are slow to start. A trajectory flat for two quarters and steeper afterwards is more...
- **Respect seasonality.** In an agricultural or malaria context, comparing Q3 to Q2 is comparing two different worlds....

Every target changes behaviour

- **Cream-skimming.** A cure rate target encourages admitting the least severe cases, which raises the rate and reduces...
- **Boundary effects.** A target of 85% produces a suspicious number of reported values between 85% and 87%, and almost...
- **Effort reallocation.** Resources move from unmeasured activities to measured ones, including from things that matter...
- **Definition drift.** The definition quietly widens until the target is met. This is the one that looks like data...
- Pair every target with an indicator that would move in the opposite direction if it were being gamed — A cure rate...

Every target changes behaviour — In 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"],
})
```

Every target changes behaviour — In 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"
)
```

Look for the boundary effect — In 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")
```

Look for the boundary effect — In R

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

Revising a target — Example

```
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
```

What comes next

- You have an indicator, a baseline and a target.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/indicator-design-logframe/indicators-07-baselines-and-targets