The anatomy of an indicator

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.

Pierre Robentz Cassion

Lesson 2 of 8

Indicator Design and the LogFrame — From intention to indicator

What this lesson covers

- What it is made of
- Four families, and choosing between them
- Three numbers that look like one
- The unit of measure belongs in the name
- Direction: say which way is better
- Leading and lagging
- Three questions before any code
- An indicator you cannot compute is not an indicator
- What comes next

What it is made of

- **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.

Four families, and choosing between them

Family	Shape	Example	Watch for
Count	A number of things	Children screened	No denominator; cannot be compared across places
Proportion	Part over whole, both same units	Share of children with MUAC under 125 mm	Numerator must be a subset of the denominator
Rate	Events over population-time	New admissions per 1,000 under-fives per month	The time unit must be in the label
Ratio / index	Two quantities not nested	Penta1-to-penta3 dropout; Food Consumption Score	Can exceed 1; direction is not obvious

Four families, and choosing between them — In Python

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

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

Four families, and choosing between them — In R

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

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

Three numbers that look like one — In Python

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

Three numbers that look like one — In R

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

Three numbers that look like one

- **4,218 screening events.** What the community health workers did. The right numerator for workload, supply consumption…
- **4,206 distinct identifiers.** Twelve forms were submitted twice on a poor connection, so twelve events are duplicates…
- **About 4,200 children.** Six more children were re-registered under a new identifier — findable only by the record…

The unit of measure belongs in the name — In 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),
}
```

The unit of measure belongs in the name — In R

```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)
)
```

Direction: say which way is better

- **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,...

Leading and lagging

- **Lagging** — tells you what happened. GAM prevalence, cure rate, annual coverage. Accurate, essential for...
- **Leading** — moves early and predicts. Stock-out days, defaulter rate in the first two weeks of treatment,...

Three questions before any code

- 1. What decision does this number inform? — If the answer is "the donor asks for it", say so honestly and keep it cheap
- 2. What would make it move, other than the thing I care about? — Coverage moves with the denominator, with reporting...
- 3. Who else already computes this, and how? — Almost always somebody does — UNICEF, WHO, the cluster, the ministry

An indicator you cannot compute is not an indicator — In 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}"
```

An indicator you cannot compute is not an indicator — In R

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

An indicator you cannot compute is not an indicator

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.

What comes next

- You have the anatomy.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/indicator-design-logframe/indicators-02-what-an-indicator-is