The denominator argument, and the cuts you promise

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.

Pierre Robentz Cassion

Lesson 4 of 8

Indicator Design and the LogFrame — The reference sheet

What this lesson covers

- Every argument about a number is an argument about the denominator
- Three denominators, one register
- The rule
- Say what it excludes, in the sheet
- Disaggregation is a promise about sample size
- Set a minimum cell size, in the sheet, in advance
- Which cuts actually change a decision
- What comes next

Every argument about a number is an argument about the denominator

- The numerator is usually agreed within a minute.

Three denominators, one register — In 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%}")
```

Three denominators, one register — In 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}"))
```

Three denominators, one register

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%

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.

The rule

- **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. . .
- **Referral completion.** Not all cases — cases that consented to be referred, which the protection dataset makes. . .
- **Attendance.** Not enrolled children times all school days — enrolled children times the days their school was open.

Say what it excludes, in the sheet — Example

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.

Disaggregation is a promise about sample size — In Python

```
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")
```

Disaggregation is a promise about sample size — In R

```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 is a promise about sample size

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

Disaggregation is a promise about sample size

- Adding one binary cut doubles the cells and halves their size — Going from two dimensions to three takes the smallest...

Set a minimum cell size, in the sheet, in advance — In 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: ""})
```

Set a minimum cell size, in the sheet, in advance — In R

```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", ""))
```

Which cuts actually change a decision

- **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.

What comes next

- You can now define one indicator precisely.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/indicator-design-logframe/indicators-04-denominators-and-disaggregation