CASSION · COURSE

Nutrition Analysis: CMAM and SMART

Anthropometry, admission and outcome indicators, and a SMART survey analysed end to end against the WHO growth standards and the Sphere performance thresholds.

Instructor	Pierre Robentz Cassion
Level	Intermediate
Learner hours	24 h
Taught in	Python · R
Updated	July 28, 2026
Sectors	Nutrition
Standards and methodologies	WHO Child Growth Standards · SMART survey · Sphere Standards · Integrated Food Security Phase Classification (IPC) · UNICEF indicator definitions

Read and run the course online at data-analysis.cassion.dev/courses/nutrition-analysis-cmam-smart

What you will be able to do

- Compute weight-for-height, height-for-age and weight-for-age z-scores against the WHO 2006 standards, from raw measurements
- Apply the SAM and MAM case definitions correctly, including the oedema rule that overrides anthropometry
- Say where MUAC and weight-for-height disagree, by how much, and which children each criterion finds
- Produce a SMART plausibility report and decide whether a survey should be accepted
- Estimate GAM and SAM prevalence with a design-adjusted interval and read it against the IPC acute malnutrition phases
- Compute CMAM performance on the denominator Sphere actually specifies, and find the site the district figure is hiding
- Explain why programme admissions over expected caseload is not coverage, and what estimating coverage requires

Prerequisites

None. This course assumes no prior programming experience.

Contents

About this course

This is the first course in the programme where the sector content is the hard part. Everything in modules 2 and 3 was method that transferred across sectors. This does not: the WHO growth standards, the SMART plausibility criteria, the Sphere performance thresholds and the IPC phases are specific, published, consequential, and this audience is held to all four.

The course is built on three registers that between them cover the whole programme cycle. A **SMART survey** of 930 children, shipped as raw weight and height so the z-scores must be computed. A **screening register** of 4,218 children, which is where cases are found. And a **CMAM admission register** of 1,100 episodes, which is where they are treated and where performance is judged.

Three findings from that data shape the course, and each is a real number computed from the committed files rather than a claim.

The two admission criteria find different children. Of the children severe by either measure, 468 are severe by weight-for-height alone, 24 by MUAC alone and 142 by both — a concordance of 22.4%. The disagreement is age: the MUAC-only children average 14.3 months, the weight-for-height-only children 31.4.

The denominator decides the cure rate. Computed as Sphere specifies — over children who reached an outcome, excluding transfers — the programme cures 82.3%. Divide the same numerator by all 1,100 admissions and it reads 73.6%.

The district figure hides the site. Defaulting is 13.7% overall, inside the Sphere maximum of 15%, and 20.5% at one site, outside it.

Both Python and R throughout. You should have done module 3, or at least be able to defend a denominator and put an interval on a proportion — this course adds the clinical definitions to both.

Part I

Measuring a child

CHAPTER 1

The four measurements, and what each is for

Lesson 1 of 8 · 120 min

Weight, length or height, MUAC and oedema. Each answers a different question, each has a precision that decides what it can support, and the one that is not a measurement at all is the one that overrides the others.

1.1 Four things, measured on one child

Everything in this course rests on four observations, and an analyst who does not know how each is taken will misread all of them.

Measurement	Unit	Precision	What it is sensitive to
Weight	kg	0.1 kg	Acute deficit — it falls fast and recovers fast
Length / height	cm	0.1 cm	Chronic deficit — it accumulates and does not recover
MUAC	mm	1 mm	Acute deficit, and mortality risk more directly than weight
Oedema	present / absent	—	Not a measurement; a clinical sign, and it overrides

Two of those rows are worth dwelling on, because they explain most of what follows.

Weight moves and height does not. A child who was hungry for two months is lighter than a child of the same height who was not, and will regain the weight in weeks if fed. A child who was hungry for two years is *shorter*, and will not regain the height. That single fact is why the sector distinguishes wasting from stunting and why the two need different indicators and different programmes.

Oedema is not on a scale. It is a clinical sign — bilateral pitting oedema, checked by pressing both feet for three seconds — and a child who has it is severely acutely malnourished whatever their weight or arm circumference. Every case definition in the next unit carries an oedema clause, and every analysis that computes SAM as "count below the z-score cut-off" is wrong by however many oedematous children the register holds.

1.2 Length or height, and the 0.7 cm that is not a rounding error

Children under 24 months are measured **lying down** — that is length. Children 24 months and over are measured **standing** — that is height. The same child measures about 0.7 cm longer lying than standing, because the spine decompresses.

The WHO standards are published against both, and the rule is:

- Under 24 months, use the **length** standard, and if the child was measured standing, **add 0.7 cm** before looking up.
- 24 months and over, use the **height** standard, and if the child was measured lying, **subtract 0.7 cm**.

```
import pandas as pd

smart = pd.read_csv("smart-nutrition-survey-2024.v1.csv")
print(smart["measured_lying"].value_counts())
```

```r
library(dplyr)
smart |> count(measured_lying)
```

752 measured lying, 178 standing. **The register records the position, which is what makes the adjustment possible** — and a register that does not is a register whose z-scores carry an unquantifiable error for every child measured in the non-standard position.

Get the sign backwards and you bias the youngest half of the survey. The joining course's lab already made you do this; here is why it matters clinically.

1.3 MUAC, and why it is used at all

MUAC is a tape around the upper arm, read to the millimetre, and it looks crude next to a z-score computed from two calibrated instruments. It is used for three reasons that between them decide most community programmes.

- **It needs one measurement, not two.** A community health worker with a tape can screen a village; a weight-for-height screening needs scales, a height board and a reference table.
- **It predicts mortality at least as well.** For a child of a given age, a small arm is a strong predictor of dying, and in several studies a better one than weight-for-height.
- **It requires no age.** Which matters enormously in populations without birth records, where the age that a z-score needs is itself an estimate.

The cost is the subject of lesson 4: MUAC grows with age independently of nutritional status, so a fixed cut-off finds different children at different ages.

1.4 Precision decides what a measurement can support

```python
muac = pd.read_csv("muac-screening-artibonite-2024.v1.csv")
values = muac.loc[muac["muac_mm"].notna(), "muac_mm"]

near_cutoff = values.between(123, 127).sum()
print(f"{near_cutoff} of {len(values)} children within 2 mm of the 125 mm cut-off")
```

```r
muac |>
  filter(!is.na(muac_mm)) |>
  summarise(near = sum(between(muac_mm, 123, 127)), n = n())
```

MUAC is read to the millimetre and repeat measurements by two trained workers routinely differ by 2 to 3 mm. So every child within about 3 mm of a cut-off is a child whose classification depends on who held the tape.

That does not make the cut-off wrong — a threshold has to be somewhere — but it has two consequences an analyst must carry:

- **Never report a MUAC-based rate to two decimal places.** The measurement does not support it.
- **Expect a pile-up at the cut-off in programme data.** A worker who reads 114 and knows 115 is the admission threshold is making a clinical judgement, not falsifying a record, and the distribution will show it.

The same argument applies to weight at 0.1 kg and height at 0.1 cm, and lesson 5 turns it into a formal check.

1.5 Read the distribution before computing anything

```python
cmam = pd.read_csv("cmam-admissions-2024.v1.csv")

print(cmam[["muac_admission_mm", "weight_admission_kg", "height_cm"]].describe().round(1))
print(f"height missing: {cmam['height_cm'].isna().sum()} of {len(cmam)}")
```

```r
cmam <- readr::read_csv("cmam-admissions-2024.v1.csv")

summary(select(cmam, muac_admission_mm, weight_admission_kg, height_cm))
sum(is.na(cmam$height_cm))
```

Seventy-two of 1,100 admissions have no height. That is not a data quality nuisance to be cleaned away — it is the reason weight-for-height cannot be computed for those children while MUAC can, and therefore the reason any comparison of the two criteria has two different denominators. Lesson 4 is built on exactly that.

1.6 What a nutrition analyst checks first

Five checks, in order, before any indicator:

- **Position recorded?** Without `measured_lying`, the 0.7 cm adjustment is a guess.
- **Age present, and how was it obtained?** A z-score needs age; a heaped age distribution says it was estimated, which lesson 5 measures.
- **Oedema recorded as a separate field?** If it is folded into a general "complications" flag, the SAM count cannot be computed correctly.
- **Units.** MUAC in mm or cm, weight in kg, height in cm. The cleaning course found seven centimetre entries in the screening register; they are always there.
- **Plausible ranges.** MUAC 80–220 mm, weight 2–30 kg, height 45–125 cm for 6–59 months. Outside those is a recording error, not a finding.

```python
implausible = cmam[
    ~cmam["muac_admission_mm"].between(80, 220)
    | ~cmam["weight_admission_kg"].between(2, 30)
    | ~cmam["height_cm"].between(45, 125).fillna(True)
]
print(f"{len(implausible)} admissions outside plausible ranges")
```

```r
cmam |>
  filter(!between(muac_admission_mm, 80, 220) |
         !between(weight_admission_kg, 2, 30) |
         (!is.na(height_cm) & !between(height_cm, 45, 125))) |>
  nrow()
```

> Anthropometry is the only part of this sector's data where the measurement error is well characterised and published. Use that: it tells you what precision your indicator can carry, and it is the reason a two-point difference between two surveys is usually nothing.

1.7 What comes next

You have four measurements and know what each is for. The next lesson turns two of them into the quantity every prevalence in this course rests on — a z-score against the WHO 2006 standards, computed from the reference table rather than trusted from a column.

CHAPTER 2

From weight and height to a z-score

Lesson 2 of 8 · 150 min

The LMS method, the reference table lookup, the three z-scores and what each measures. Computed on 930 children it gives a mean of -0.67 and a standard deviation of 1.22 — and that second number is already a finding.

2.1 What a z-score says

A z-score answers one question: **how far is this child from the median of a healthy reference population, in standard deviations?**

A weight-for-height z-score of -2 means this child weighs what a child two standard deviations below the median weighs, for their height. It is not a percentage of anything, and it is not comparable to a percentile without conversion.

The reference is the **WHO 2006 Child Growth Standards**, built from a multi-country study of children raised in conditions that do not constrain growth — breastfed, non-smoking households, adequate healthcare. That construction is deliberate: the standards describe how children *should* grow, not how children in any particular place do.

2.2 Three z-scores, three questions

Score	Compares	Deficit it detects	Recovers?
Weight-for-height (WHZ)	Weight against height	Acute — wasting	Yes, in weeks
Height-for-age (HAZ)	Height against age	Chronic — stunting	No
Weight-for-age (WAZ)	Weight against age	Both together — underweight	Partly

Weight-for-age is the one to be careful with. It mixes the two deficits, so a low WAZ does not say whether the child is wasted, stunted or both, and the programme response differs. It survives in growth monitoring because it needs no height board, and it should not be used to classify acute malnutrition.

This course computes weight-for-height, because that is what the case definitions and the IPC thresholds are built on.

2.3 The LMS method

The standards are published as three parameters per sex, per measurement, per 0.1 cm of length or height:

- **L** — the Box-Cox power that normalises a skewed distribution
- **M** — the median
- **S** — the coefficient of variation

```
z = ((weight / M)^L - 1) / (L * S)
```

```python
import pandas as pd

reference = pd.read_csv("who-2006-weight-for-lenhei.csv")
print(reference.head(3))
print(f"{len(reference)} rows: sex x standard x lenhei in 0.1 cm steps")
```

```r
reference <- readr::read_csv("who-2006-weight-for-lenhei.csv")
nrow(reference)
```

2,404 rows. **Read the table; do not reimplement the interpolation from a textbook.** R has the official `anthro` package; Python has no maintained equivalent that installs cleanly, which is exactly why this table is committed here.

2.4 The lookup key has three parts

This is the part that goes wrong, and the joining course's lab was built on it.

```python
def lookup_key(row):
    lying = row["measured_lying"] == "true"
    standard = "L" if row["age_months"] < 24 else "H"

    lenhei = row["height_cm"]
    if standard == "L" and not lying:
        lenhei += 0.7
    elif standard == "H" and lying:
        lenhei -= 0.7

    return row["sex"], standard, round(lenhei, 1)
```

```r
lookup_key <- function(sex, age, height, lying) {
  standard <- if (age < 24) "L" else "H"
  lenhei <- height +
    if (standard == "L" && !lying) 0.7 else if (standard == "H" && lying) -0.7 else 0
  list(sex = sex, lorh = standard, lenhei = round(lenhei, 1))
}
```

Three things in that function are decisions, not mechanics.

The standard follows the age, not the position. A 30-month-old measured lying is still assessed against the height standard, with the adjustment applied.

The adjustment goes on the measurement, not the standard. You are converting the measurement into the one the standard expects.

The rounding is to one decimal, on both sides, in one place. A float computed from an adjustment is not the same float as one read from a CSV, which is the inexact-key join the joining course spent a lab on.

2.5 Computing it

```python
reference["key"] = list(zip(reference["sex"], reference["lorh"],
                            reference["lenhei"].round(1)))
lms = reference.set_index("key")[["l", "m", "s"]]

def whz(row):
    key = lookup_key(row)
```

```
    if key not in lms.index or pd.isna(row["weight_kg"]):
        return None
    l, m, s = lms.loc[key]
    return (((row["weight_kg"] / m) ** l) - 1) / (l * s)

smart["whz"] = smart.apply(whz, axis=1)
print(f"{smart['whz'].notna().sum()} of {len(smart)} computable")
```

```
# In R, use the official package rather than the table:
# anthro::anthro_zscores(sex = ..., age = ..., weight = ..., lenhei = ..., measure = ...)
```

864 of 930 computable. The 66 that are not split into three groups and the distinction matters:

- **Missing weight or age** — 32 children, and they are missing data.
- **Height outside the reference range** — the length standard runs 45.0 to 110.0 cm and the height standard 65.0 to 120.0 cm. A child outside it is genuinely unclassifiable.
- **A measurement in the wrong unit** — four heights recorded in metres, which the joining course's lab found by anti-join. These are recoverable.

Report the three separately. "66 children excluded" hides that some were data errors you could have fixed.

2.6 Flagging: two rules, two answers

Extreme z-scores are excluded before any prevalence, and there are two conventions.

- **WHO flags** — fixed bounds, exclude z outside -5 to +5. Absolute, comparable across surveys.
- **SMART flags** — relative, exclude observations more than 3 SD from the *survey's own* mean. Adapts to the survey, and shrinks it more when the survey is noisier.

```
who_flagged = smart["whz"].between(-5, 5)

mean, sd = smart["whz"].mean(), smart["whz"].std()
smart_flagged = smart["whz"].between(mean - 3 * sd, mean + 3 * sd)

print(f"WHO flags keep {who_flagged.sum()}, SMART flags keep {smart_flagged.sum()}")
```

```
smart <- smart |>
  mutate(who_ok = between(whz, -5, 5),
         smart_ok = between(whz, mean(whz, na.rm = TRUE) - 3 * sd(whz, na.rm = TRUE),
                            mean(whz, na.rm = TRUE) + 3 * sd(whz, na.rm = TRUE)))
```

They exclude different children and give slightly different rates. **State which you used**; a SMART plausibility report requires it, and two surveys using different rules are not directly comparable.

2.7 The standard deviation is already a finding

```python
analysable = smart.loc[who_flagged, "whz"]
print(f"n = {len(analysable)}, mean = {analysable.mean():.2f}, "
      f"sd = {analysable.std():.2f}")
```

```r
smart |> filter(who_ok) |> summarise(n = n(), mean = mean(whz), sd = sd(whz))
```

852 children, **mean -0.67, standard deviation 1.22**.

The mean is unremarkable — a population somewhat below the reference median, which is normal in this sector. The standard deviation is not. A well-measured survey produces a weight-for-height SD between about **0.8 and 1.2**, because the reference population's SD is 1 by construction and real populations are only slightly more variable.

1.22 sits at the edge of acceptable, and an SD above the range means one of three things: measurement error inflating the spread, a genuinely heterogeneous population, or a data entry problem. **It is the single most informative number in a plausibility report**, and lesson 5 is about reading it properly.

Note also what it does to the prevalence. A wider distribution puts more children past a fixed cut-off, so an inflated SD raises GAM without any child being more malnourished.

2.8 Save the z-scores with their inputs

```python
smart[["child_id", "cluster", "team", "age_months", "sex", "weight_kg",
       "height_cm", "measured_lying", "oedema", "whz"]].to_csv(
    "outputs/smart_with_zscores.csv", index=False)
```

```r
readr::write_csv(smart, here::here("outputs", "smart_with_zscores.csv"))
```

Keep the inputs beside the output. When somebody disagrees with a prevalence, the argument is almost never about the arithmetic — it is about the adjustment, the flagging rule or the excluded children, and all three are recoverable only if the inputs travelled with the result.

2.9 What comes next

You have a z-score for every analysable child. The next unit turns it into a classification — the case definitions for severe and moderate acute malnutrition, and the clinical sign that overrides both.

Part II

Who is a case

CHAPTER 3

The case definitions, and the clause everyone drops

Lesson 3 of 8 · 120 min

Severe, moderate and global acute malnutrition, the cut-offs the sector agreed on, and the oedema clause that makes every "count below the threshold" implementation wrong.

3.1 The definitions

Three terms, and they nest.

Term	Weight-for-height z	MUAC	Oedema
SAM — severe acute malnutrition	below -3	below 115 mm	present
MAM — moderate acute malnutrition	-3 to below -2	115 to below 125 mm	—
GAM — global acute malnutrition	below -2	below 125 mm	present

GAM is SAM plus MAM. It is not a separate condition; it is the total burden of acute malnutrition, and it is the figure the IPC phases and the 15% emergency threshold are defined against.

Two structural points before any code.

Each row is a complete definition on its own. A child is SAM by weight-for-height, *or* by MUAC, *or* by oedema — the criteria are alternatives, not requirements. Lesson 4 is entirely about how far the first two disagree.

The oedema column is an OR, not an AND. A child with bilateral pitting oedema is severe whatever their measurements say. That clause is one term in a boolean expression and it is the term most often left out.

3.2 The implementation that is wrong

```python
# WRONG: the definition everybody writes first
sam_wrong = (smart["whz"] < -3).sum()
```

```r
# WRONG
sum(smart$whz < -3, na.rm = TRUE)
```

It is wrong in two ways at once. It drops the oedematous children, and `na.rm` or a null-propagating comparison silently drops the children with no z-score — so both the numerator and the denominator are quietly different from what the label claims.

3.3 The implementation that is right

```python
import pandas as pd

analysable = smart[smart["whz"].between(-5, 5)].copy()
```

```python
oedema = analysable["oedema"] == True

analysable["sam"] = (analysable["whz"] < -3) | oedema
analysable["mam"] = (analysable["whz"].between(-3, -2, inclusive="left")) & ~oedema
analysable["gam"] = analysable["sam"] | analysable["mam"]

n = len(analysable)
for label in ["sam", "mam", "gam"]:
    print(f"{label.upper():4} {analysable[label].sum():>4} / {n}  "
          f"{analysable[label].mean():.1%}")
```

```r
analysable <- smart |> filter(between(whz, -5, 5))

analysable <- analysable |>
  mutate(sam = whz < -3 | oedema,
         mam = whz >= -3 & whz < -2 & !oedema,
         gam = sam | mam)

analysable |> summarise(across(c(sam, mam, gam), list(n = sum, rate = mean)), n = n())
```

852 analysable children: **GAM 14.9%, SAM 3.8%.**

Three details in that code are load-bearing.

MAM excludes oedema explicitly. An oedematous child whose z-score falls in the moderate band is severe, not moderate, and without the `& ~oedema` they would be counted in both — so SAM plus MAM would exceed GAM.

The bounds are half-open. `-3 <= z < -2`. A child at exactly -3.0 is severe. Getting this wrong moves a handful of children and is the kind of thing two analysts discover they disagree on at the worst moment.

The denominator is stated. 852, not 930 — the difference is children with no computable z-score and children outside the flagging bounds, and lesson 2 required you to report those three groups separately.

3.4 Check that the parts sum to the whole

```python
assert (analysable["sam"] & analysable["mam"]).sum() == 0, "a child is both"
assert analysable["gam"].sum() == analysable["sam"].sum() + analysable["mam"].sum()
```

```r
stopifnot(sum(analysable$sam & analysable$mam) == 0,
          sum(analysable$gam) == sum(analysable$sam) + sum(analysable$mam))
```

Two assertions, and they catch the oedema mistake, the bound mistake and any future edit that breaks the nesting. Run them every time.

3.5 The MUAC definitions, on the register

The screening register has MUAC and oedema and no weight or height, so it supports the MUAC-based definitions only.

```python
muac = pd.read_csv("muac-screening-artibonite-2024.v1.csv")
measured = muac[muac["muac_mm"].notna()].copy()
oed = measured["oedema"] == True
```

```python
measured["sam"] = (measured["muac_mm"] < 115) | oed
measured["gam"] = (measured["muac_mm"] < 125) | oed

print(f"n = {len(measured):,}  GAM {measured['gam'].mean():.1%}  "
      f"SAM {measured['sam'].mean():.1%}")
```

```r
muac |>
  filter(!is.na(muac_mm)) |>
  summarise(n = n(),
            gam = mean(muac_mm < 125 | oedema),
            sam = mean(muac_mm < 115 | oedema))
```

Name the measure in the indicator. `gam_muac_percent` and `gam_whz_percent` are different indicators that both get called "GAM", and the indicator design course's rule about putting the measure in the name is not pedantry here — the next lesson shows the two disagreeing on three-quarters of severe cases.

3.6 What a definition does not say

Three things the case definition deliberately leaves out, each of which someone will assume.

Age. The definitions apply to children 6 to 59 months. Below six months the standards and the case definitions are different; above 59 months they do not apply at all. Check the age range before classifying.

```python
out_of_scope = ~analysable["age_months"].between(6, 59)
print(f"{out_of_scope.sum()} children outside 6-59 months")
```

```r
sum(!between(analysable$age_months, 6, 59), na.rm = TRUE)
```

Admission. A case definition says who *is* malnourished. Whether they are admitted depends on the programme's protocol, which may use one criterion, both, or MUAC only for community screening and weight-for-height at the site. The register's `admission_criterion` records what the site entered, not everything the child met.

Severity within severe. SAM with complications needs inpatient care; SAM without needs outpatient. The definition does not distinguish them, and the appetite test and clinical signs that do are not in an anthropometric dataset.

> A case definition is a classification rule, not a treatment decision. Confusing the two produces a caseload figure that does not match any programme's admissions and an argument nobody can resolve from the data.

3.7 What comes next

You can classify a child by either measure. The next lesson puts the two measures on the same children and finds that they disagree on three-quarters of severe cases — and that the disagreement has a cause you can name in one number.

CHAPTER 4

Where the two criteria disagree

Lesson 4 of 8 · 150 min

468 children severe by weight-for-height alone, 24 by MUAC alone, 142 by both. A concordance of 22.4%, and the whole of the disagreement is age — 31.4 months against 14.3.

4.1 The question the register can answer

Both admission criteria are in the case definitions and both are used in the field. Whether they find the same children is an empirical question, and it needs a register with both measurements on the same child.

The CMAM admission register has exactly that: MUAC and weight-for-height at admission, for 1,100 episodes. This lesson is what it says.

4.2 Compute both on the same children

```python
import pandas as pd

cmam = pd.read_csv("cmam-admissions-2024.v1.csv")
cmam["whz"] = cmam.apply(whz, axis=1)          # from lesson 2

both = cmam[cmam["whz"].notna()].copy()
print(f"{len(both)} of {len(cmam)} admissions have both measurements")

both["sam_whz"] = both["whz"] < -3
both["sam_muac"] = both["muac_admission_mm"] < 115
```

```r
both <- cmam |> filter(!is.na(whz))

both <- both |>
  mutate(sam_whz = whz < -3,
         sam_muac = muac_admission_mm < 115)
```

1,028 of 1,100. The 72 without a height are the missing-height defect from the dataset's known issues, and they are the first finding: **the two criteria do not have the same denominator even on the same register**, because one needs two measurements and the other needs one.

4.3 The cross-tabulation

```python
table = pd.crosstab(both["sam_muac"], both["sam_whz"],
                    rownames=["SAM by MUAC"], colnames=["SAM by WHZ"])
print(table)
```

```r
both |> count(sam_muac, sam_whz)
```

	Not severe by WHZ	Severe by WHZ
Not severe by MUAC	394	**468**
Severe by MUAC	**24**	142

Read the two bold cells. **468 children are severe by weight-for-height and not by MUAC. 24 are severe by MUAC and not by weight-for-height.** 142 are severe by both.

```python
overlap = table.loc[True, True]
union = table.values.sum() - table.loc[False, False]
print(f"concordance {overlap / union:.1%} of {union} severe by either measure")
```

```r
both |>
  summarise(overlap = sum(sam_muac & sam_whz),
            union = sum(sam_muac | sam_whz)) |>
  mutate(concordance = overlap / union)
```

22.4%. Of the children this register would call severe by one measure or the other, fewer than a quarter are called severe by both.

That number is not an artefact of this dataset. Published comparisons find overlaps in the 10% to 40% range depending on the population, and a programme that has not measured its own is assuming one.

4.4 The disagreement is age

```python
mean_age = pd.Series({
    "MUAC only": both.loc[both["sam_muac"] & ~both["sam_whz"], "age_months"].mean(),
    "WHZ only": both.loc[both["sam_whz"] & ~both["sam_muac"], "age_months"].mean(),
    "Both": both.loc[both["sam_muac"] & both["sam_whz"], "age_months"].mean(),
})
print(mean_age.round(1))
```

```r
both |>
  mutate(group = case_when(sam_muac & sam_whz ~ "both",
                           sam_muac ~ "muac only",
                           sam_whz ~ "whz only",
                           TRUE ~ "neither")) |>
  summarise(mean_age = mean(age_months), n = n(), .by = group)
```

Group	Mean age
Severe by MUAC only	**14.3 months**
Severe by both	15.6 months
Severe by weight-for-height only	**31.4 months**

There it is. **The children MUAC finds and weight-for-height misses are half the age of the children weight-for-height finds and MUAC misses.**

The mechanism is simple and it is not a defect of either measure. MUAC grows with age: a healthy arm is about 135 mm at six months and about 155 mm at five years. A fixed 115 mm cut-off is therefore a much deeper deficit for a five-year-old than for an infant, so MUAC becomes progressively harder to fail as a child grows. Weight-for-height has no age term at all and applies equally across the range.

```python
by_age = both.assign(
    band=pd.cut(both["age_months"], [5, 12, 24, 36, 60],
                labels=["6-11", "12-23", "24-35", "36-59"])
).groupby("band")[["sam_muac", "sam_whz"]].mean()
print((by_age * 100).round(1))
```

```r
both |>
  mutate(band = cut(age_months, c(5, 12, 24, 36, 60),
                    labels = c("6-11", "12-23", "24-35", "36-59"))) |>
  summarise(across(c(sam_muac, sam_whz), mean), .by = band)
```

Run that and the two lines cross. MUAC identifies more of the youngest children; weight-for-height identifies more of the oldest. **A programme admitting on MUAC alone is running a younger caseload than one admitting on weight-for-height alone**, and that is a clinical fact about who gets treated, not a measurement detail.

4.5 Which one is right?

Neither, and the question is the wrong one. What each is *for* differs.

- **MUAC predicts mortality at least as well**, and better in several studies. If the purpose of admission is to treat the children most likely to die, that is a strong argument.
- **Weight-for-height is the survey standard**, and the IPC thresholds and international comparisons are built on it. If the purpose is to classify a population, that is decisive.
- **MUAC is operationally feasible** at community level. Weight-for-height is not.

The sector's answer is to use both, admit on either, and discharge on the one the child was admitted on. Which is sensible clinically and creates the analytical problem this lesson exists for.

4.6 What it means for your numbers

Four consequences, and each is a thing you will be asked about.

Caseload depends on the criterion. Admitting on either criterion gives a caseload of 634 severe children here; on MUAC alone, 166; on weight-for-height alone, 610. Those are not estimates of the same quantity.

Prevalence is not comparable across measures. A GAM by MUAC and a GAM by weight-for-height are different indicators. Lesson 6 shows what that does when a figure is read against the 15% threshold.

Discharge must use the admission criterion. A child admitted on MUAC and discharged on weight-for-height may be discharged before recovering, or held long after. The CMAM protocol is explicit and registers routinely are not.

A programme changing criterion breaks its own series. A caseload that rises 40% when the protocol changes is a definitional break, and the indicator design course's rule about versioning a definition applies exactly.

```python
for name, mask in [("either", both["sam_muac"] | both["sam_whz"]),
                   ("muac only", both["sam_muac"]),
                   ("whz only", both["sam_whz"])]:
    print(f"{name:10} caseload {mask.sum():>4}")
```

```
both |> summarise(either = sum(sam_muac | sam_whz),
                  muac = sum(sam_muac), whz = sum(sam_whz))
```

4.7 Report the pair, always

```
Severe acute malnutrition, admissions 2024, n = 1,028 with both measurements

  By MUAC (<115 mm)                166
  By weight-for-height (z < -3)    610
  By either                        634
  By both                          142    (22.4% of those severe by either)

  Children severe by MUAC alone average 14.3 months; by weight-for-height alone,
  31.4 months. The two criteria identify overlapping but substantially different
  caseloads, and the difference is age.
```

Six lines. They pre-empt the question, they name the mechanism, and they stop somebody comparing a MUAC-based caseload to a weight-for-height-based one as though the gap were a change in nutrition.

4.8 What comes next

You can classify children and you know what the classification depends on. The next unit turns to whether the survey that produced the measurements can be believed at all — the SMART plausibility report, which is where the standard deviation of 1.22 from lesson 2 finally gets read.

Part III

Judging a survey

CHAPTER 5

Can this survey be believed?

Lesson 5 of 8 · 150 min

The SMART plausibility report, on a survey that fails two of its checks. A standard deviation of 1.22, one team measuring 0.6 z-scores low, 69% of its heights on a half centimetre, and a quarter of ages on a whole year.

5.1 A survey is accepted or it is not

The SMART methodology does something unusual and valuable: it publishes a set of checks a survey must pass before its prevalence is used, and the checks are computable from the survey's own data.

That means a survey can be **rejected on its own evidence**, before any argument about what the number means. This lesson runs the checks on the platform's SMART survey, which passes some and fails others.

5.2 The checks

Check	What it detects	Acceptable
Flagged records	Impossible or extreme measurements	under about 2.5%
Standard deviation of WHZ	Measurement error inflating the spread	0.8 to 1.2
Digit preference	Rounding rather than reading	low, and even across teams
Age heaping	Age estimated rather than documented	low
Sex ratio	Selection bias in who was measured	near 1.0
Team bias	One team measuring differently	means close across teams

5.3 Flags

```python
computable = smart["whz"].notna()
flagged = computable & ~smart["whz"].between(-5, 5)

print(f"{computable.sum()} computable, {flagged.sum()} flagged "
      f"({flagged.sum() / computable.sum():.1%})")
```

```r
smart |>
  filter(!is.na(whz)) |>
  summarise(n = n(), flagged = sum(!between(whz, -5, 5)),
            rate = flagged / n)
```

12 of 864, about 1.4%. **Passes.** A flag rate above a few percent means measurement or entry problems severe enough to question everything else, so this check runs first and gates the rest.

5.4 The standard deviation

```python
analysable = smart.loc[smart["whz"].between(-5, 5), "whz"]
print(f"n = {len(analysable)}, mean = {analysable.mean():.2f}, "
      f"sd = {analysable.std():.2f}")
```

```r
smart |> filter(between(whz, -5, 5)) |> summarise(n = n(), sd = sd(whz))
```

1.22. Outside the acceptable band, at the top edge.

This is the most important number in a plausibility report and it is worth being precise about why. The WHO reference population has a standard deviation of 1 by construction. A real population is slightly more variable, so 1.0 to 1.2 is what a well-measured survey produces. Above that, the spread is being inflated by something, and there are three candidates:

- **Measurement error.** Random error in weight or height widens the distribution without changing its centre. The commonest cause.
- **A genuinely heterogeneous population.** Two very different sub-populations sampled together.
- **Entry error.** Transposed digits, wrong units, mixed-up records.

An inflated SD raises the prevalence. A wider distribution puts more children past a fixed cut-off, so GAM rises without a single child being more malnourished. That is why the check exists and why it is not optional.

5.5 Digit preference

A measurement read from an instrument has a near-uniform last digit. One rounded by eye does not.

```python
smart["terminal"] = (smart["height_cm"] * 10).round() % 10
by_team = (
    smart.assign(rounded=smart["terminal"].isin([0, 5]))
    .groupby("team")
    .agg(rounded=("rounded", "mean"), n=("rounded", "size"))
)
print((by_team["rounded"] * 100).round(0))
```

```r
smart |>
  mutate(rounded = round(height_cm * 10) %% 10 %in% c(0, 5)) |>
  summarise(share = mean(rounded), n = n(), .by = team)
```

Team	Heights ending .0 or .5
1	18%
2	**69%**
3	18%
4	23%

Two of ten digits is 20% under no preference. Teams 1, 3 and 4 are at it. **Team 2 is at 69%**, three and a half times the others.

Fails, and it needs no significance test — the DQA course's calibration discipline applies to marginal results, and this is not marginal. The mechanism is nameable: the team read the measuring board to the nearest half centimetre instead of the millimetre.

Note what that does. Rounding to 0.5 cm adds noise to height, which propagates into weight-for-height, which inflates the standard deviation — so the two failing checks are related, and one is partly causing the other.

5.6 Age heaping

```python
ages = smart["age_months"].dropna()
whole_years = (ages % 12 == 0).mean()

print(f"{whole_years:.0%} of ages on an exact whole year")
print(ages.value_counts().reindex([22, 23, 24, 25, 26]).to_dict())
```

```r
smart |>
  filter(!is.na(age_months)) |>
  summarise(whole_years = mean(age_months %% 12 == 0))
```

24% on a whole year, against about 9% expected. Sixty-six children at exactly 24 months, against 15 and 19 either side.

Fails, and the cause is almost never carelessness — it is that birth records are not universal and a carer asked a child's age answers in years. The corrective action is a local events calendar and probing against siblings, not a reprimand.

It matters here for a specific reason. The WHO standards switch reference table at 24 months and the SMART age groups are bounded at 12-month marks, so heaping deposits a clump of children exactly where the classification changes.

5.7 Sex ratio and team bias

```python
ratio = (smart["sex"] == "m").sum() / (smart["sex"] == "f").sum()
print(f"sex ratio m:f = {ratio:.2f}")

by_team = (
    smart[smart["whz"].between(-5, 5)]
    .groupby("team")
    .agg(mean_whz=("whz", "mean"), n=("whz", "size"))
)
print(by_team.round(2))
```

```r
smart |>
  filter(between(whz, -5, 5)) |>
  summarise(mean_whz = mean(whz), n = n(), .by = team)
```

Team	Mean WHZ	n
1	-0.69	204
2	-0.42	228
3	**-1.09**	224
4	-0.44	196

Sex ratio near 1.0 — **passes**.

Team means spread from -0.42 to -1.09. **Fails.** Clusters were assigned to teams independently of nutritional status, so a spread of 0.67 z-scores between teams is not a difference in the children; it is a difference in the measuring.

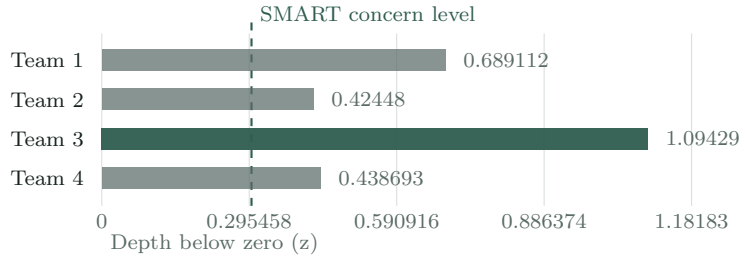

Figure 5.1: Mean weight-for-height z-score by measurement team, plotted as depth below zero. Clusters were assigned to teams independently of nutrition status, so a spread of this size between teams is a measurement fault rather than a real difference between populations.

What it costs is direct:

```python
gam_by_team = (
    smart[smart["whz"].between(-5, 5)]
    .assign(gam=lambda d: (d["whz"] < -2) | (d["oedema"] == True))
    .groupby("team")["gam"].mean()
)
print((gam_by_team * 100).round(1))
```

```r
smart |>
  filter(between(whz, -5, 5)) |>
  summarise(gam = mean(whz < -2 | oedema), .by = team)
```

Team 3's clusters give GAM of 22.3% against 9.2% to 16.2% for the others. **Team 3 contributes a quarter of the sample and pulls the survey-wide estimate up by roughly two points.**

5.8 The verdict

```python
plausibility = pd.DataFrame({
    "check": ["Flagged records", "SD of WHZ", "Digit preference",
              "Age heaping", "Sex ratio", "Team bias"],
    "value": ["1.4%", "1.22", "69% (team 2)", "24% whole years", "1.05",
              "-0.42 to -1.09"],
    "acceptable": ["<2.5%", "0.8–1.2", "even across teams", "low", "~1.0",
                   "close across teams"],
    "verdict": ["pass", "fail", "fail", "fail", "pass", "fail"],
})
print(plausibility)
```

```r
tibble::tribble(
  ~check,              ~value,            ~verdict,
  "Flagged records",   "1.4%",            "pass",
  "SD of WHZ",         "1.22",            "fail",
  "Digit preference",  "69% (team 2)",    "fail",
```

```
  "Age heaping",       "24%",             "fail",
  "Sex ratio",         "1.05",            "pass",
  "Team bias",         "-0.42 to -1.09",  "fail"
)
```

Four failures. **What follows is a judgement, not an arithmetic result**, and the honest options are three:

- **Reject the survey.** Defensible with four failures, and expensive — six weeks and a budget, gone.
- **Accept with stated limitations.** Report the prevalence with the plausibility table beside it and an explicit statement that the SD and team bias both push the estimate up. Most common, and it requires the table to travel with the figure.
- **Reanalyse excluding team 3.** Defensible only if you say so prominently, and it costs a quarter of the sample and therefore widens the interval.

What you must not do is publish the prevalence without the plausibility report. The number is not wrong; it is unqualified, and the qualification changes what it supports.

5.9 Write it up without accusing anyone

Do not write	Write
"Team 2's measurements are unreliable"	"Team 2's height readings end in .0 or .5 in 69% of cases against 18-23% fo
"Team 3 falsified measurements"	"Team 3's mean weight-for-height is 0.4 to 0.7 z-scores below the other tea
"Ages are inaccurate"	"24% of ages fall on an exact whole year against 9% expected, indicating a

The right-hand column names what was observed, its size and a mechanism. It is also what makes the corrective action obvious — recalibrate a scale, retrain on board reading, add an events calendar — where the left-hand column produces a defensive team and worse data next round.

5.10 What comes next

You know whether the survey can be believed and with what caveats. The next lesson turns it into the number it exists to produce — GAM and SAM prevalence, with an interval that accounts for the cluster design, read against the IPC phases.

CHAPTER 6

The interval that spans two IPC phases

Lesson 6 of 8 · 150 min

GAM is 14.9%, at the top of IPC Phase 3. The design-adjusted interval runs 11.1% to 18.7%, which spans Phase 3 and Phase 4 — and the survey cannot say which the population is in.

6.1 The number the survey exists to produce

Everything so far has been preparation. The survey was funded to answer one question — how much acute malnutrition is there — and this lesson produces the answer in the form it has to take: a prevalence, an interval, and a phase.

6.2 The estimate, with the design

A SMART survey is a cluster survey. The survey course established what that means; here it arrives with a threshold attached.

```python
import numpy as np
import pandas as pd

analysable = smart[smart["whz"].between(-5, 5)].copy()
analysable["gam"] = (analysable["whz"] < -2) | (analysable["oedema"] == True)
analysable["sam"] = (analysable["whz"] < -3) | (analysable["oedema"] == True)

def cluster_prevalence(df, column, cluster="cluster"):
    n = len(df)
    p = df[column].mean()

    totals = df.groupby(cluster)[column].agg(["sum", "size"])
    residual = totals["sum"] - p * totals["size"]
    m = len(totals)

    variance = m / (m - 1) * (residual**2).sum() / n**2
    se = np.sqrt(variance)
    srs = p * (1 - p) / n
    return {"p": p, "se": se, "deff": variance / srs, "clusters": m, "n": n}

for column in ["gam", "sam"]:
    r = cluster_prevalence(analysable, column)
    t = 2.045                                   # t(0.975, 29 df)
    print(f"{column.upper()}: {r['p']:.1%}  95% CI {r['p'] - t * r['se']:.1%} to "
          f"{r['p'] + t * r['se']:.1%}  deff {r['deff']:.2f}  n {r['n']}")
```

```r
library(survey)

design <- svydesign(ids = ~cluster, weights = NULL, data = analysable)

svyciprop(~I(whz < -2 | oedema), design, method = "logit")
svymean(~I(whz < -2 | oedema), design, deff = TRUE)
```

	Estimate	95% CI	Design effect	n
GAM	14.9%	**11.1 – 18.7%**	2.29	852
SAM	3.8%	2.3 – 5.2%	1.20	852

Two things to hold about that table before reading the phase.

The design effect is 2.29 for GAM and 1.20 for SAM. Two indicators, one survey, two design effects — because malnutrition clusters geographically and severe malnutrition, being rarer, clusters less detectably. The survey's 852 children are worth about 372 independent ones for GAM.

The naive interval would be 12.5 – 17.3%. Ignoring the clustering makes the interval 37% narrower, and the next section is what that narrowing would have cost.

6.3 The IPC phases

The IPC classifies acute malnutrition prevalence into five phases, and they are what turn a percentage into a decision.

Phase	GAM by weight-for-height
1 — Acceptable	under 5%
2 — Alert	5 to 9.9%
3 — Serious	10 to 14.9%
4 — Critical	**15 to 29.9%**
5 — Extremely critical	30% and above

```
def ipc_phase(gam):
    for threshold, phase in [(0.05, 1), (0.10, 2), (0.15, 3), (0.30, 4)]:
        if gam < threshold:
            return phase
    return 5

r = cluster_prevalence(analysable, "gam")
t = 2.045
low, high = r["p"] - t * r["se"], r["p"] + t * r["se"]

print(f"point estimate: phase {ipc_phase(r['p'])}")
print(f"interval spans: phase {ipc_phase(low)} to phase {ipc_phase(high)}")
```

```
c(point = 0.149, low = 0.111, high = 0.187)
```

The point estimate is 14.9%, which is Phase 3 — by one tenth of a point. The interval runs 11.1% to 18.7%, which spans Phase 3 and Phase 4.

6.4 What that means, said plainly

This is the moment the whole course has been building to, and the honest statement is uncomfortable.

> Global acute malnutrition is estimated at 14.9% (95% CI 11.1–18.7). The point estimate falls in IPC Phase 3 (Serious). The confidence interval spans Phase 3 and Phase 4 (Critical), so **this survey cannot determine which phase the population is in.**

Three things that sentence does.

It refuses to round to the threshold. 14.9% reported as "about 15%" has made a phase classification by rounding, and the difference between Phase 3 and Phase 4 is a difference in response.

It states the interval before the phase. A reader who sees "Phase 3" first will not revise it when they reach the interval.

It says the survey cannot decide. That is the three-branch verdict the survey course insisted on, and here the third branch is the correct one.

6.5 What would have happened without the design effect

```
srs_se = np.sqrt(r["p"] * (1 - r["p"]) / r["n"])
print(f"naive interval: {r['p'] - 1.96 * srs_se:.1%} to {r['p'] + 1.96 * srs_se:.1%}")
```

```
p <- 0.149; n <- 852
c(p - 1.96 * sqrt(p * (1 - p) / n), p + 1.96 * sqrt(p * (1 - p) / n))
```

12.5% to 17.3%. **Still spanning both phases** — so on this survey the design effect does not change the verdict, and saying so is worth as much as a finding.

But note how close it came. Had the estimate been 13.5%, the naive interval would have sat inside Phase 3 and the design-adjusted one would have crossed into Phase

1. **The clustering decides the classification whenever the estimate is within

   about two points of a threshold**, which in this sector is most of the time.

6.6 Read it with the plausibility report

The previous lesson found four failed checks, and two of them push in a known direction.

- **The inflated standard deviation (1.22) raises GAM.** A wider distribution puts more children past a fixed cut-off.
- **Team 3's low measurements raise GAM.** Its clusters give 22.3% against 9.2 to 16.2% for the others, and it contributes a quarter of the sample.

So the estimate is more likely to be too high than too low, which pushes the true value toward the lower half of the interval — toward Phase 3 rather than Phase 4.

```
without_team3 = analysable[analysable["team"] != 3]
r3 = cluster_prevalence(without_team3, "gam")
print(f"excluding team 3: {r3['p']:.1%} on n={r3['n']}, "
      f"{r3['clusters']} clusters")
```

```
analysable |> filter(team != 3) |> summarise(gam = mean(gam), n = n())
```

Report the sensitivity, do not substitute it. The headline stays the full-sample estimate; the exclusion goes beside it as evidence about direction. Silently dropping a quarter of a survey because it gives an inconvenient answer is the thing this whole module exists to prevent.

6.7 The full reporting block

```
Global acute malnutrition (weight-for-height z < -2 or oedema)
  14.9%   95% CI 11.1-18.7   n = 852   design effect 2.29   30 clusters
  IPC Phase 3 by point estimate; interval spans Phase 3 and Phase 4.

Severe acute malnutrition (z < -3 or oedema)
  3.8%    95% CI 2.3-5.2     n = 852   design effect 1.20

Plausibility: 4 of 6 SMART checks failed (SD 1.22, digit preference in one team,
age heaping 24%, team bias -0.42 to -1.09). Two of the four push the estimate
upward. Excluding the affected team gives 12.6% on 628 children.

Analysable denominator: 852 of 930 measured. 66 had no computable z-score
(32 missing weight or age, 30 outside the reference range, 4 heights in metres,
recoverable) and 12 were flagged by the WHO bounds.
```

Twelve lines. They carry the estimate, its interval, its design, its classification, its known biases and its denominator, and there is nothing left for a reviewer to ask that the block does not answer.

6.8 What comes next

The survey says what the situation is. The next unit asks what the programme did about it — CMAM performance against the Sphere standards, and the denominator that moves the cure rate by nine points.

Part IV

Judging a programme

CHAPTER 7

The denominator that moves the cure rate nine points

Lesson 7 of 8 · 150 min

Cured, defaulted, died, non-response — against the Sphere minimum standards, on the denominator Sphere actually specifies. 82.3% or 73.6%, depending on a choice nobody documents, and one site outside the standard the district figure hides.

7.1 Four outcomes and a standard

A CMAM programme is judged on what happened to the children it admitted, and the Sphere minimum standards give the thresholds.

Outcome	Sphere minimum, outpatient therapeutic care
Cured	over 75%
Died	under 10%
Defaulted	under 15%
Non-response	no fixed threshold; investigated if high

Supplementary feeding has its own set — cured over 75%, died under 3%, defaulted under 15%. The numbers differ; the structure does not.

Every one of those is a proportion, and the argument is always the denominator.

7.2 The three candidate denominators

```python
import pandas as pd

cmam = pd.read_csv("cmam-admissions-2024.v1.csv")

print(f"admissions:            {len(cmam):>5}")
print(f"with an outcome:       {cmam['outcome'].notna().sum():>5}")
print(f"excluding transfers:   "
      f"{(cmam['outcome'].notna() & (cmam['outcome'] != 'transferred')).sum():>5}")
```

```r
cmam |> summarise(
  admissions = n(),
  with_outcome = sum(!is.na(outcome)),
  excluding_transfers = sum(!is.na(outcome) & outcome != "transferred")
)
```

Denominator	n	What it treats the excluded as
All admissions	1,100	Children still in treatment counted as not cured
Reached an outcome	1,029	Transfers counted as an outcome
Reached an outcome, excluding transfers	**984**	—

```python
cured = (cmam["outcome"] == "cured").sum()
for label, n in [("all admissions", 1100), ("with an outcome", 1029),
                 ("excluding transfers", 984)]:
    print(f"cure rate on {label:22} {cured / n:.1%}")
```

```r
cured <- sum(cmam$outcome == "cured", na.rm = TRUE)
c(all = cured / 1100, outcome = cured / 1029, sphere = cured / 984)
```

73.6%, 78.7%, 82.3%. One numerator, three denominators, nine points.

And the middle one crosses a threshold: on all admissions the programme is *below* the Sphere minimum of 75% and on the Sphere denominator it is comfortably above. The difference is not performance. It is a denominator nobody wrote down.

7.3 Which one Sphere means

Children who reached a treatment outcome, excluding transfers. 984 here.

The two exclusions are for different reasons and both are right.

Children still in treatment are not an outcome. Seventy-one children were admitted late enough that they had not been discharged at the cut-off. They are neither cured nor defaulted; they are in treatment. Counting them as anything is a claim about a future that has not happened.

Transfers are somebody else's outcome. Forty-five children moved between programmes — outpatient to inpatient, or the reverse — and their treatment outcome will be recorded where they end up. Counting them here would count them twice across the two programmes, or credit this programme with a result it did not produce.

```python
performance = cmam[cmam["outcome"].notna() & (cmam["outcome"] != "transferred")]

rates = performance["outcome"].value_counts(normalize=True)
print((rates * 100).round(1))
print(f"n = {len(performance)}")
```

```r
performance <- cmam |> filter(!is.na(outcome), outcome != "transferred")

performance |> count(outcome) |> mutate(rate = n / sum(n))
```

Outcome	Rate	Sphere	
Cured	82.3%	over 75%	pass
Defaulted	13.7%	under 15%	pass
Died	1.0%	under 10%	pass
Non-response	2.9%	—	

Four rows, all inside the standard. **This is where most CMAM reports stop, and it is where the finding is hiding.**

7.4 Break it down by site

```python
by_site = (
    performance.assign(**{o: performance["outcome"] == o
                          for o in ["cured", "defaulted", "died", "non-response"]})
```

```
    .groupby("site_id")
    .agg(n=("outcome", "size"), cured=("cured", "mean"),
         defaulted=("defaulted", "mean"), died=("died", "mean"))
)
print((by_site[["cured", "defaulted", "died"]] * 100).round(1))
```

```
performance |>
  summarise(n = n(),
            cured = mean(outcome == "cured"),
            defaulted = mean(outcome == "defaulted"),
            died = mean(outcome == "died"),
            .by = site_id) |>
  arrange(desc(defaulted))
```

Site	n	Defaulted
SITE-03	156	**20.5%**
SITE-02	168	15.5%
SITE-06	125	13.6%
SITE-05	142	11.3%
SITE-01	188	11.2%
SITE-04	205	11.2%

The programme defaults at 13.7% and one site defaults at 20.5%. The district figure is inside the Sphere maximum and that site is not, by five points.

That is the whole reason the DQA course insisted on distributions over means, and it arrives here with a clinical consequence: a defaulting child is a child who stopped treatment before recovering.

```
flagged = by_site[(by_site["defaulted"] > 0.15) | (by_site["cured"] < 0.75)]
print(f"{len(flagged)} of {len(by_site)} sites outside a Sphere standard")
```

```
performance |>
  summarise(cured = mean(outcome == "cured"),
            defaulted = mean(outcome == "defaulted"), .by = site_id) |>
  filter(defaulted > 0.15 | cured < 0.75)
```

Two sites, and the second is marginal at 15.5%. **Flag against the standard, not against the district mean** — the standard is what the programme is held to.

7.5 The cause is usually distance, not compliance

Defaulting has a literature and it is consistent: the strongest predictor is travel time to the site. A carer who must lose a day of work every week to walk two hours will stop coming when the child looks better, which is rational and is not non-compliance.

```
length_of_stay = (
    pd.to_datetime(performance["discharge_date"])
    - pd.to_datetime(performance["admission_date"])
).dt.days

print(performance.assign(stay=length_of_stay)
      .groupby("outcome")["stay"].median().round(0))
```

```
performance |>
  mutate(stay = as.integer(as.Date(discharge_date) - as.Date(admission_date))) |>
  summarise(median_stay = median(stay), .by = outcome)
```

Defaulters leave far earlier than cured children — the median stay is about three weeks against about eight. **A defaulter is not a treatment failure; it is a child who left before the treatment finished**, and the corrective action is decentralisation or transport support, not counselling.

That is the root cause discipline from the DQA course, applied to a clinical indicator.

7.6 Weight gain, and the entries that cannot be right

```
gain = (
    (performance["weight_discharge_kg"] - performance["weight_admission_kg"])
    / performance["weight_admission_kg"]
    / length_of_stay * 1000
)
print(gain.describe().round(1))
print(f"{(gain < 0).sum()} discharges with weight loss")
```

```
performance |>
  mutate(gain = (weight_discharge_kg - weight_admission_kg) /
                weight_admission_kg / as.integer(as.Date(discharge_date) -
                as.Date(admission_date)) * 1000) |>
  summarise(median = median(gain, na.rm = TRUE), negative = sum(gain < 0, na.rm = TRUE))
```

Weight gain in grams per kilogram per day is the standard measure of treatment response, and nine discharges show weight loss. That is possible in a child who died or defaulted early, and it is also exactly what a transposed entry looks like. **The register cannot distinguish them**, which is a finding for the data quality section rather than a number to clean away.

7.7 The performance table to publish

```
CMAM performance, 2024

  Admissions                          1,100
  Still in treatment at cut-off          71    excluded: not an outcome
  Transferred                            45    excluded: outcome recorded elsewhere
  Performance denominator               984

  Cured            82.3%   Sphere >75%    pass
  Defaulted        13.7%   Sphere <15%    pass
  Died              1.0%   Sphere <10%    pass
  Non-response      2.9%

  One site (SITE-03, n=156) defaults at 20.5%, outside the Sphere maximum.
  Median stay for defaulters is 24 days against 58 for cured children,
  consistent with distance rather than non-response to treatment.
```

The excluded rows are shown, not deleted. A reader can reconstruct any of the three denominators from that block, which is the difference between a performance table and an assertion.

7.8 What comes next

The programme cures the children it admits. The last lesson asks the harder question — what share of the children who needed treatment it ever saw — and why the figure most programmes report as coverage is not coverage.

CHAPTER 8

Why admissions over expected caseload is not coverage

Lesson 8 of 8 · 120 min

The figure most programmes report as coverage is a ratio of two things, neither of which is the number of malnourished children. What coverage actually requires, and what to report when you cannot estimate it.

8.1 The figure everyone reports

Somewhere in almost every CMAM report is a line like this:

> Programme coverage: 68%

and underneath it, if you look, the calculation:

```
admissions during the period / expected caseload for the period
```

The expected caseload is itself computed, usually as:

```
population under five x GAM prevalence x incidence correction x coverage target
```

```
UNDER_FIVE = 21500
GAM_PREVALENCE = 0.149          # from the SMART survey
INCIDENCE_CORRECTION = 1.6      # new cases arising over the year
COVERAGE_TARGET = 0.50

expected = UNDER_FIVE * GAM_PREVALENCE * INCIDENCE_CORRECTION * COVERAGE_TARGET
admissions = 1100

print(f"expected caseload {expected:,.0f}; admissions {admissions:,}; "
      f"'coverage' {admissions / expected:.0%}")
```

```
expected <- 21500 * 0.149 * 1.6 * 0.50
c(expected = expected, ratio = 1100 / expected)
```

That number is not coverage. It is worth being precise about what it is, because it is not useless — it is just not what its label says.

8.2 What it actually measures

Coverage is a proportion: children treated, over children who needed treatment. The formula above has neither.

The numerator is admissions, not children. A child readmitted after relapse is two admissions. The reach lesson in the indicator course made this general point; here it inflates the numerator by whatever the relapse rate is.

The denominator is a forecast, not a count. Every term in it is an estimate: the population from a census projection, the prevalence from a survey with its own interval,

the incidence correction from a literature value, and the coverage target from a proposal. Multiply four estimates and the product carries all four uncertainties.

The coverage target is in the denominator. Look at it again. A programme that targeted 50% coverage divides by half the caseload, so hitting the target reads as 100%. The metric is constructed to reach 100% when the plan is met, which makes it a **plan achievement ratio** — a perfectly reasonable management indicator, wearing the wrong name.

```
without_target = admissions / (UNDER_FIVE * GAM_PREVALENCE * INCIDENCE_CORRECTION)
print(f"ratio without the coverage target in the denominator: {without_target:.0%}")
```

```
1100 / (21500 * 0.149 * 1.6)
```

Take the target out and the same programme reads at half the figure. Nothing about the programme changed.

8.3 What coverage requires

The definition is not in dispute. Coverage is:

```
children with SAM currently in the programme
--------------------------------------------
children with SAM in the population
```

and the denominator can only come from **finding malnourished children in the population who are not in the programme**. No register can supply it, because a register only contains children who came.

That is why coverage surveys exist as a separate methodology. The current standards — SQUEAC, SLEAC and their successors — do exactly this: active case-finding in the community, then classify each case found as in or out of the programme.

```
# What a coverage survey produces, and a register never can
cases_found = 84
cases_in_programme = 39
print(f"point coverage: {cases_in_programme / cases_found:.0%}")
```

```
c(point_coverage = 39 / 84)
```

Two coverage measures are in use and they answer different questions:

- **Point coverage** — cases currently in the programme, over cases found. Suits a programme with short treatment episodes.
- **Period coverage** — cases in the programme or recently discharged cured, over cases found. Suits longer episodes, and credits the programme for children it already cured.

State which. They differ by several points and both are called coverage.

8.4 The barriers question is the useful half

A coverage survey's estimate is a number with a wide interval. Its other output is often more actionable: for every case found and not in the programme, the carer is asked why.

The answers cluster, and they map to different fixes:

- **Did not know the child was malnourished** — screening and community awareness.
- **Did not know the programme existed** — mobilisation.
- **Distance, cost, time** — decentralisation, which is also the defaulting cause from the previous lesson.
- **Rejected previously** — an admission criterion problem, and lesson 4 explains how a child can be rejected on one measure while malnourished by another.
- **Stigma, or a previous bad experience** — quality of care.

```
barriers = pd.DataFrame({
    "barrier": ["Did not know child was malnourished", "Distance",
                "Did not know programme existed", "Previously rejected",
                "Carer unavailable"],
    "cases": [17, 12, 8, 5, 3],
})
barriers["share"] = barriers["cases"] / barriers["cases"].sum()
```

```
barriers <- tibble::tribble(
  ~barrier,                  ~cases,
  "unaware child malnourished",  17,
  "distance",                    12,
  "unaware of programme",         8,
  "previously rejected",          5,
  "carer unavailable",            3
)
```

Report the barriers even when the coverage estimate is too imprecise to use. Forty-five interviews cannot pin a coverage figure to five points and can perfectly well tell you that half the missed cases are a screening problem rather than an access problem, and those have different budgets.

8.5 What to report when you have no coverage survey

Which is most of the time, because a coverage survey costs money the programme usually does not have. Three honest options, and none is to relabel the ratio.

Report the plan achievement ratio, correctly named. "Admissions were 68% of the caseload planned for the period" is true, useful for supply planning, and makes no claim about children who never came.

Report admissions against expected incident cases, with the assumptions stated. Drop the coverage target from the denominator, list the four inputs and their sources, and label it an estimate of programme reach against expected need.

Use the routine-to-survey ratio. The DHIS2 course's last lesson decomposed the gap between a screening register and a survey; that gap is an indirect signal about who the programme is not seeing, and it is free.

```
report = pd.DataFrame({
    "indicator": ["Admissions", "Expected incident cases", "Plan achievement",
                  "Coverage"],
    "value": [1100, 5126, "21%", "not estimated"],
    "note": ["episodes, not children",
```

```
              "population x prevalence x incidence correction; three estimates",
              "admissions / expected incident cases",
              "requires active case-finding in the community"],
})
```

```
tibble::tribble(
  ~indicator,                ~value,          ~note,
  "Admissions",              "1,100",         "episodes, not children",
  "Expected incident cases", "5,126",         "three stacked estimates",
  "Plan achievement",        "21%",           "admissions / expected",
  "Coverage",                "not estimated", "requires a coverage survey"
)
```

The last row is the one that matters. **"Not estimated" is a finding**, and it is a far better line in a report than a number that will be quoted for three years as something it is not.

8.6 Where this course leaves you

You can measure a child, compute a z-score against the WHO standards, apply the case definitions including the clause everyone drops, say where the two admission criteria disagree and why, run a SMART plausibility report and decide whether a survey can be believed, produce a prevalence with an interval and read it against the IPC phases, compute CMAM performance on the denominator Sphere specifies, and say honestly what your programme does and does not know about coverage.

That is the nutrition analyst's job, and it is the first of module 4's six sector courses.

Next in the module is *Public Health and Applied Epidemiology* — incidence, prevalence, cascades and coverage for HIV, TB, malaria and immunisation, plus the outbreak curve you may have to draw at short notice. It runs on the vaccination extract this programme has been using since module 2, and it starts where this course's denominators leave off: with person-time, which is the denominator that makes incidence mean anything at all.

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/nutrition-analysis-cmam-smart

Generated July 28, 2026.