

The denominator that moves the cure rate nine points

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.

Pierre Robentz Cassion

Lesson 7 of 8

Nutrition Analysis: CMAM and SMART — Judging a programme

What this lesson covers

- Four outcomes and a standard
- The three candidate denominators
- Which one Sphere means
- Break it down by site
- The cause is usually distance, not compliance
- Weight gain, and the entries that cannot be right
- The performance table to publish
- What comes next

Four outcomes and a standard

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

Four outcomes and a standard

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

The three candidate denominators — In 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}")
```

The three candidate denominators — In R

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

The three candidate denominators

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	—

The three candidate denominators — In 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%}")
```

The three candidate denominators — In R

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

The three candidate denominators

- 73.6%, 78.7%, 82.3% — One numerator, three denominators, nine points

Which one Sphere means

- Children who reached a treatment outcome, excluding transfers — 984 here
- Children still in treatment are not an outcome — Seventy-one children were admitted late enough that they had not been. . .
- Transfers are somebody else's outcome — Forty-five children moved between programmes — outpatient to inpatient, or the. . .

Which one Sphere means — In 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)}")
```

Which one Sphere means — In R

```
performance <- cmam |> filter(!is.na(outcome), outcome != "transferred")  
  
performance |> count(outcome) |> mutate(rate = n / sum(n))
```

Which one Sphere means

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%	—	

Break it down by site — In 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))
```

Break it down by site — In R

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

Break it down by site

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%

Break it down by site

- The programme defaults at 13.7% and one site defaults at 20.5% — The district figure is inside the Sphere maximum and...

Break it down by site — In Python

```
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")
```

Break it down by site — In R

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

The cause is usually distance, not compliance — In Python

```
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))
```

The cause is usually distance, not compliance — In R

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

Weight gain, and the entries that cannot be right — In Python

```
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")
```

Weight gain, and the entries that cannot be right — In R

```
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))
```

The performance table to publish — Example

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.

What comes next

- The programme cures the children it admits.

Read the full lesson, with runnable code

[https://data-analysis.cassion.dev/courses/nutrition-analysis-cmam-smart/
nutrition-07-cmam-performance](https://data-analysis.cassion.dev/courses/nutrition-analysis-cmam-smart/nutrition-07-cmam-performance)