Periods, and the aggregation operator nobody sets

One file, one indicator, three annual coverage figures — 6.5%, 60.8% and 77.5% — all from defensible-sounding sentences. The difference is how the denominator aggregates over time.

Pierre Robentz Cassion

Lesson 3 of 8

Routine Data and DHIS2 — Periods and completeness

What this lesson covers

- Period types, and the identifier that carries them
- Which period does a value belong to?
- The three annual figures
- Why they differ
- The aggregation operator
- Deadlines, expiry and locking
- Completeness is a separate record
- What comes next

Period types, and the identifier that carries them — Example

```
202408      monthly
2024W32     weekly
2024Q3      quarterly
2024        yearly
2024April   financial year starting April
```

Period types, and the identifier that carries them — In Python

```python
import pandas as pd

vax = pd.read_csv("vaccination-coverage-2024.v1.csv", parse_dates=["period"])
vax["period_id"] = vax["period"].dt.strftime("%Y%m")
print(sorted(vax["period_id"].unique())[:4])
```

Period types, and the identifier that carries them — In R

```
library(dplyr)
vax <- vax |> mutate(period_id = format(period, "%Y%m"))
```

Which period does a value belong to?

- **Right: the period the activity happened in.** A dose given on 28 August is August, however late the form arrives.
- **Wrong: the period the form was submitted in.** Common where data entry is batched, and it shifts activity forward by…
- **Wrong: the period the value was keyed in.** Same failure, one step further.

The three annual figures — In Python

```
penta3 = vax[vax["antigen"] == "penta3"]
reported = penta3[penta3["report_submitted"] == True]

doses = reported["doses_administered"].sum()

summed_denominator = reported["target_population"].sum()
annual_target = penta3.groupby("facility_id")["target_population"].first().sum()

print(f"doses: {doses:,}")
print(f"a) doses / summed monthly target : {doses / summed_denominator:.1%}")
print(f"b) doses / annual target, all 38 : {doses / annual_target:.1%}")
```

The three annual figures — In R

```
penta3 <- vax |> filter(antigen == "penta3")
reported <- penta3 |> filter(report_submitted)

doses <- sum(reported$doses_administered)
summed <- sum(reported$target_population)
annual <- penta3 |> summarise(t = first(target_population), .by = facility_id) |>
  summarise(sum(t)) |> pull()

c(a = doses / summed, b = doses / annual)
```

The three annual figures

Sentence	Figure
"Doses over target population, summed across the year"	**6.5%**
"Annual doses over the annual target population"	**60.8%**
"Annual doses over the annual target, among reporting facility-months"	**77.5%**

Why they differ

- 6.5% is a monthly figure wearing an annual label — `target_population` is an *annual* cohort — the surviving infants in...
- 60.8% counts silent facilities as having vaccinated nobody — The denominator is every facility's full annual cohort,...
- 77.5% is the defensible one — and it needs an explicit denominator adjustment:

Why they differ — In Python

```
months_reported = reported.groupby("facility_id").size()
targets = penta3.groupby("facility_id")["target_population"].first()
prorated = (targets * months_reported.reindex(targets.index).fillna(0) / 12).sum()

print(f"c) pro-rated denominator: {doses / prorated:.1%}")
```

Why they differ — In R

```r
months <- reported |> summarise(m = n(), .by = facility_id)
targets <- penta3 |> summarise(t = first(target_population), .by = facility_id)

prorated <- targets |>
  left_join(months, by = "facility_id") |>
  mutate(m = coalesce(m, 0)) |>
  summarise(sum(t * m / 12)) |> pull()

doses / prorated
```

The aggregation operator

Operator	Meaning	Right for
Sum	Add across periods	Counts of events: doses, admissions, consultations
Average	Mean across periods	Stock levels, staffing, anything that is a state not an event
Last value	Take the most recent	Populations, targets, register sizes

The aggregation operator

- Ask for the aggregation operator with the metadata — It is one field per data element, it is invisible in the export,...

Deadlines, expiry and locking

- **The deadline** is when the dataset is due. It drives the timeliness figure the DQA course computes.
- **Expiry days** lock the form a fixed number of days after the period ends. After that, entry needs an unlock.
- **A lock exception** unlocks one dataset, one org unit, one period. Every one is a decision, and a system with hundreds...

Completeness is a separate record

- Completeness in DHIS2 is a registration, not a computation — When a user clicks "complete" on a data entry form, the...

Completeness is a separate record — In Python

```
grid = (vax["facility_id"].nunique() * vax["period"].nunique()
        * vax["antigen"].nunique())
print(f"{grid} rows expected, {len(vax)} present, "
      f"{(vax['report_submitted'] == True).sum()} flagged reported")
```

Completeness is a separate record — In R

```
c(expected = n_distinct(vax$facility_id) * n_distinct(vax$period) * n_distinct(vax$antigen),
  present = nrow(vax),
  reported = sum(vax$report_submitted))
```

What comes next

- The reporting flag is the raw material of the completeness figure, and the next lesson turns it into a denominator.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/routine-data-and-dhis2/dhis2-03-periods-and-datasets