The two indicators nobody computes

Reporting completeness and timeliness, each with its own denominator. Eleven of thirty-eight facilities reported in August, and every coverage figure for that month is a statement about those eleven.

Pierre Robentz Cassion

Lesson 2 of 8

What this lesson covers

- Two indicators, and neither is in the report
- Reporting completeness
- Break it down three ways, always
- The denominator trap
- Timeliness needs a field you may not have
- Read every other figure through these two
- What comes next

Two indicators, and neither is in the report

- Open almost any routine health report from this sector and you will find coverage, caseload, cure rates and stock-outs.

Reporting completeness — In Python

```
import pandas as pd

vax = pd.read_csv("vaccination-coverage-2024.v1.csv", parse_dates=["period"])
vax["reported"] = vax["report_submitted"] == True

print(f"overall reporting rate: {vax['reported'].mean():.1%}")

by_month = vax.groupby(vax["period"].dt.strftime("%Y-%m"))["reported"].mean()
print((by_month * 100).round(0))
```

Reporting completeness — In R

```r
library(dplyr)
library(readr)

vax <- read_csv("vaccination-coverage-2024.v1.csv")

mean(vax$report_submitted)

vax |>
  mutate(month = format(period, "%Y-%m")) |>
  summarise(reporting_rate = mean(report_submitted), .by = month) |>
  arrange(month)
```

Reporting completeness

Month	Reporting rate
January	92%
February	82%
March	74%
April–July	82–87%
August	**29%**
September	**45%**
October	95%
November–December	76–87%

Break it down three ways, always — In Python

```
by_facility = vax.groupby("facility_id")["reported"].mean().sort_values()
print(by_facility.head(6))
print(f"{(by_facility == 1).sum()} facilities reported every month")

by_type = vax.groupby("facility_type")["reported"].mean()
print((by_type * 100).round(1))
```

Break it down three ways, always — In R

```
vax |> summarise(rate = mean(report_submitted), .by = facility_id) |> arrange(rate) |> head
    (6)

vax |> summarise(rate = mean(report_submitted), .by = facility_type)
```

Break it down three ways, always

Facility type	Reporting rate
Health centre	82.8%
District hospital	75.0%
Health post	71.7%

The denominator trap — In Python

```
# WRONG: the share of reports that were submitted, among reports that were submitted
wrong = vax[vax["reported"]]["reported"].mean()          # always 1.0
```

The denominator trap — In R

```r
# WRONG
vax |> filter(report_submitted) |> summarise(rate = mean(report_submitted))
```

The denominator trap — In Python

```
EXPECTED_FACILITIES = pd.read_csv("facility-master-list.csv")["facility_id"]

expected = len(EXPECTED_FACILITIES) * vax["period"].nunique()
submitted = vax.loc[vax["reported"], ["facility_id", "period"]].drop_duplicates().shape[0]
print(f"{submitted} of {expected} expected facility-months")
```

The denominator trap — In R

```r
expected  <- nrow(facility_master) * n_distinct(vax$period)
submitted <- vax |> filter(report_submitted) |> distinct(facility_id, period) |> nrow()
```

Timeliness needs a field you may not have — In Python

```
submissions["days_late"] = (
    submissions["submitted_on"] - submissions["deadline"]
).dt.days

print(f"on time: {(submissions['days_late'] <= 0).mean():.1%}")
print(f"median days late, of those late: "
      f"{submissions.loc[submissions['days_late'] > 0, 'days_late'].median():.0f}")
```

Timeliness needs a field you may not have — In R

```r
submissions <- submissions |>
  mutate(days_late = as.integer(submitted_on - deadline))

c(on_time = mean(submissions$days_late <= 0),
  median_late = median(submissions$days_late[submissions$days_late > 0]))
```

Timeliness needs a field you may not have

- This extract cannot support that calculation — and saying so is itself a DQA finding

Timeliness needs a field you may not have

Finding. Timeliness cannot be assessed. The monthly extract carries a submission flag but no submission date, although DHIS2 records one. **Corrective action.** Add the completeness date to the standard extract. **Owner.** HMIS focal point. **By.** Next quarterly extract.

Read every other figure through these two

- **Coverage, at 76.5% reporting.** A district total is a lower bound, not an estimate. Say "among reporting facilities"...
- **A month-on-month trend.** August's fall is a reporting fall until proven otherwise. Plot the reporting rate on the...
- **A facility ranking.** Facilities reporting seven months of twelve cannot be ranked against facilities reporting...

Read every other figure through these two — In Python

```
monthly = (
    vax[vax["reported"]]
    .groupby(vax["period"].dt.strftime("%Y-%m"))
    .apply(lambda g: g["doses_administered"].sum() / g["target_population"].sum())
    .rename("coverage")
    .to_frame()
)
monthly["reporting_rate"] = by_month
print(monthly.round(3))
```

Read every other figure through these two — In R

```
vax |>
  mutate(month = format(period, "%Y-%m")) |>
  summarise(
    coverage = sum(doses_administered[report_submitted]) /
               sum(target_population[report_submitted]),
    reporting_rate = mean(report_submitted),
    .by = month
  )
```

Read every other figure through these two

Two columns, always adjacent. A coverage figure without its reporting rate is an unlabelled number, and the label is the part that stops it being misread.

What comes next

- Completeness and timeliness are about whether a report arrived.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/data-quality-assessment/dqa-02-completeness-and-timeliness