# Three coverages that disagree

Administrative coverage, survey coverage and the dropout between doses. The first depends on a projection, the second on a sample, and only the third is immune to both.

---

Pierre Robentz Cassion

Lesson 5 of 8

Public Health and Applied Epidemiology — Comparing places

# What this lesson covers

- Three ways to measure the same thing
- Administrative coverage, and its two dependencies
- Survey coverage, and its two dependencies
- Why they disagree
- Dropout: the figure that survives all of it
- What the two dropouts say together
- Which to report
- What comes next

# Three ways to measure the same thing

| Method | Numerator | Denominator | Fails when |
|---|---|---|---|
| **Administrative** | Doses recorded by facilities | Population projection | The projection is wrong, or reporting is incomplete |
| **Survey** | Children with a card or recalled dose | Children sampled | Cards are lost; recall is poor; the sample is small |
| **Dropout** | First dose minus last dose | First dose | Rarely — and that is the point |

# Administrative coverage, and its two dependencies — In Python

```python
import pandas as pd

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

penta3 = reported[reported["antigen"] == "penta3"]
annual_target = (
    vax[vax["antigen"] == "penta3"].groupby("facility_id")["target_population"].first()
)
months = penta3.groupby("facility_id").size()
prorated = (annual_target * months.reindex(annual_target.index).fillna(0) / 12).sum()

print(f"doses {penta3['doses_administered'].sum():,}")
print(f"administrative coverage: "
      f"{penta3['doses_administered'].sum() / prorated:.1%}")
```

# Administrative coverage, and its two dependencies — In R

```r
library(dplyr)
# same shape: doses over a denominator pro-rated to months reported
```

# Administrative coverage, and its two dependencies

- 77.5% — among reporting facility-months
- The denominator is a projection — Surviving infants, from a census projection compounded forward at an assumed growth...
- The reporting rate was 76.5% — Facilities that did not report contribute no doses, so the figure is a lower bound...

## Survey coverage, and its two dependencies

- Card retention — Where cards are lost, coverage rests on caregiver recall, which overstates for some antigens and...
- Precision — Survey coverage arrives with a confidence interval, and the survey course showed that a cluster design...

# Survey coverage, and its two dependencies — In Python

```python
# From the survey course: a proportion with a design-adjusted interval
def coverage_interval(p, n, deff, t=1.99):
    se = (p * (1 - p) / n * deff) ** 0.5
    return p - t * se, p + t * se

print(coverage_interval(0.775, 900, 2.0))
```

# Survey coverage, and its two dependencies — In R

```r
coverage_interval <- function(p, n, deff, t = 1.99) {
  se <- sqrt(p * (1 - p) / n * deff)
  c(p - t * se, p + t * se)
}
```

# Why they disagree

- **The projection is wrong.** The commonest, and it moves administrative coverage only. A denominator too small pushes...
- **Reporting is incomplete.** Moves administrative down.
- **Catchment crossing.** Moves facility-level administrative figures in both directions and cancels at district level.
- **Recall error.** Moves survey coverage, usually up.
- **Different age cohorts.** Administrative coverage counts doses given this year; a survey counts children aged 12 to 23...
- The last one is the most common and the least suspected — Two figures for "penta3 coverage" that refer to different...

# Dropout: the figure that survives all of it — In Python

```
doses = reported.groupby("antigen")["doses_administered"].sum()

for start, end in [("penta1", "penta3"), ("mcv1", "mcv2")]:
    dropout = (doses[start] - doses[end]) / doses[start]
    print(f"{start} -> {end}: {dropout:.1%}")
```

# Dropout: the figure that survives all of it — In R

```
vax |> filter(report_submitted) |>
  summarise(doses = sum(doses_administered), .by = antigen)
```

# Dropout: the figure that survives all of it

- Penta1 to penta3: 13.8%. Measles first to second dose: 22.9%
- **no population denominator**, so no projection to be wrong;
- **the same reporting bias in both terms**, which largely cancels;
- **no cohort ambiguity**, because both doses are counted the same way.

# What the two dropouts say together — In Python

```
summary = pd.DataFrame({
    "series": ["Pentavalent 1->3", "Measles 1->2"],
    "dropout": [0.138, 0.229],
    "interval": ["weeks", "months"],
    "implication": ["retention within the early contact period",
                    "retention after the early contact period ends"],
})
```

# What the two dropouts say together — In R

```
tibble::tribble(
  ~series,              ~dropout, ~implication,
  "Pentavalent 1->3",      0.138, "retention within early contacts",
  "Measles 1->2",          0.229, "retention after early contacts end"
)
```

# Which to report — Example

| | | |
|---|---|---|
| Penta3 coverage, administrative | 77.5% | among reporting facility-months<br>(reporting rate 76.5%)<br>denominator: MoH projection from 2015 census |
| Penta1 to penta3 dropout | 13.8% | same source, both terms |
| Measles 1 to 2 dropout | 22.9% | |
| Survey coverage | not available this year | |

## What comes next

- Coverage compares a programme against a population.

## Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/public-health-epidemiology/epi-05-immunisation-coverage