CASSION · COURSE

Public Health and Applied Epidemiology

Incidence, prevalence, cascades and coverage for HIV, TB, malaria and immunisation programmes, plus the outbreak curve you may have to draw at short notice.

Instructor	Pierre Robentz Cassion
Level	Intermediate
Learner hours	22 h
Taught in	Python · R
Updated	July 28, 2026
Sectors	Public Health
Standards and methodologies	UNICEF indicator definitions · Sustainable Development Goals (SDG) · Sphere Standards

Read and run the course online at data-analysis.cassion.dev/courses/public-health-epidemiology

What you will be able to do

- Distinguish a rate, a ratio and a proportion, and say what person-time adds that a headcount denominator cannot
- Build a cascade as chained denominators and locate the step where the largest share is lost
- Draw an epidemic curve from a line list, and say what the missing onset dates do to its peak
- Compute attack rates and case fatality with their denominators stated, against the thresholds the response is judged on
- Age-standardise two districts and say how much of the apparent difference between them was structure
- Name the confounders in a routine-data comparison, and state plainly what an observational finding can and cannot claim

Prerequisites

None. This course assumes no prior programming experience.

Contents

About this course

Epidemiology is the discipline of denominators, and this course is where the platform's standing instruction — always state how an indicator is calculated — meets the measures that have names and published thresholds.

It runs on two registers. The **cholera line list** is 975 cases recorded one at a time over sixteen weeks, shipped with the population they came from, which is the only dataset here that supports an epidemic curve, an attack rate and case fatality at once. The **vaccination extract** has appeared in six earlier courses and this one finishes it: coverage three ways, and the dropout between doses that says whether children who start a schedule complete it.

Three results shape the course, all computed from those files.

Age structure is half the difference between the worst and best district. Crude cholera attack rates are 7.94 and 5.51 per 1,000. Standardised to a common population they are 7.25 and 5.95, so the gap narrows from 2.43 to 1.30. Half the apparent difference was who lives there, not what happened to them.

Case fatality is 3.90% against a 1% target, and 6.30% in one district against 2.00% in another. The predictor in this data is onset-to-admission delay, which is a service question rather than a clinical one.

The epidemic curve is not the outbreak. Forty-seven cases have no onset date, and one district's register is back-filled from the admission book, so its curve is really an admission curve shifted right.

Both Python and R throughout. You should have done module 3 — this course assumes you can defend a denominator, put an interval on a proportion, and read a routine figure through its reporting rate.

Part I

The denominator is the epidemiology

CHAPTER 1

Rate, ratio, proportion — and person-time

Lesson 1 of 8 · 135 min

Three shapes that get called "rate" and only one of them is, plus the denominator that makes incidence mean something when people enter and leave a population at different times.

1.1 Three shapes, one word

The indicator design course established four families of measure. Epidemiology uses three of them constantly and calls all three "rate", which is where most misreading starts.

Shape	Form	Range	Example
Proportion	Part over whole, both same units	0 to 1	Case fatality: deaths among cases
Ratio	Two quantities, not nested	any	Sex ratio of cases; odds ratio
Rate	Events over population-**time**	any positive	Incidence per 1,000 person-years

Only the third is a rate, and the difference is the time in the denominator. A proportion asks "what share"; a rate asks "how fast".

1.2 Prevalence and incidence

The pair this whole course turns on.

- **Prevalence** — how many people have the condition *now*. A proportion, from a survey, a snapshot.
- **Incidence** — how many *new* cases arise per unit of population-time. A rate, from surveillance, a flow.

They answer different questions and they move for different reasons. Prevalence rises if incidence rises **or if people survive longer with the condition**, which is why a successful treatment programme can raise HIV prevalence while lowering incidence — the most misread pair of numbers in this sector.

```python
import pandas as pd

cases = pd.read_csv("cholera-line-list-2024.v1.csv", parse_dates=["onset_date"])
population = pd.read_csv("district-population-2024.v1.csv")

total_population = population["population"].sum()
print(f"{len(cases)} cases in a population of {total_population:,}")
print(f"attack rate over the outbreak: {1000 * len(cases) / total_population:.2f} per
    1,000")
```

```r
library(dplyr)

cases <- readr::read_csv("cholera-line-list-2024.v1.csv")
population <- readr::read_csv("district-population-2024.v1.csv")
```

```r
c(cases = nrow(cases),
  population = sum(population$population),
  per_1000 = 1000 * nrow(cases) / sum(population$population))
```

975 cases in 145,000 people — **6.7 per 1,000 over the outbreak**. Strictly that is an *attack rate*, which is a proportion despite its name: cases over the population at risk, over a defined outbreak period. Lesson 4 takes it seriously.

1.3 What person-time adds

A headcount denominator assumes everyone was present and at risk for the whole period. In this sector they routinely were not — people arrive, leave, are born, die, or enter a programme mid-year.

Person-time counts each person for as long as they were actually at risk.

```python
# One person followed for 6 months contributes 0.5 person-years
follow_up = pd.DataFrame({
    "person_id": ["A", "B", "C"],
    "entered": pd.to_datetime(["2024-01-01", "2024-01-01", "2024-07-01"]),
    "exited": pd.to_datetime(["2024-12-31", "2024-06-30", "2024-12-31"]),
})
follow_up["person_years"] = (
    (follow_up["exited"] - follow_up["entered"]).dt.days / 365.25
)

events = 1
print(f"{events / follow_up['person_years'].sum():.3f} events per person-year")
print(f"naive headcount rate: {events / len(follow_up):.3f} per person")
```

```r
follow_up <- tibble::tibble(
  person_id = c("A", "B", "C"),
  entered = as.Date(c("2024-01-01", "2024-01-01", "2024-07-01")),
  exited  = as.Date(c("2024-12-31", "2024-06-30", "2024-12-31"))
) |>
  mutate(person_years = as.numeric(exited - entered) / 365.25)

c(per_person_year = 1 / sum(follow_up$person_years),
  naive = 1 / nrow(follow_up))
```

Three people, two of whom were present for half the year. The headcount denominator is 3; the person-time denominator is 2.0 person-years. **The two answers differ by a third**, and the second is the one that compares across periods and places with different amounts of follow-up.

Use person-time whenever:

- **Follow-up varies** — a treatment cohort where people enrol through the year.
- **The population changes** — displacement, camp arrivals, a catchment that grows.
- **You are comparing periods of different length** — a nine-month response against a twelve-month one.

Use a headcount denominator when everyone was present throughout and the period is fixed, which is what makes an outbreak attack rate legitimate.

1.4 Say the denominator and the multiplier in the name

```python
indicators = {
    "cholera_attack_rate_per_1000_outbreak": 1000 * len(cases) / total_population,
    "case_fatality_percent_of_cases_with_outcome": None,
    "incidence_per_1000_person_years": None,
}
```

```python
# The same discipline: the name carries the denominator and the multiplier
```

Two things belong in the name and are usually missing.

The multiplier. Per 100, per 1,000, per 100,000. Cholera attack rates are conventionally per 1,000; maternal mortality per 100,000; immunisation coverage per 100. Getting the convention wrong is a factor of a hundred and it happens.

The denominator's population. "Per 1,000 population" and "per 1,000 children under five" are different indicators. The indicator course made this general point; here the conventions are published and departing from one silently is worse than inventing one.

1.5 The at-risk population is not always the whole population

The denominator of a rate is the population **at risk of the event**.

- **Maternal mortality ratio** — deaths per 100,000 *live births*, not per population, because only pregnancies are at risk.
- **Neonatal mortality** — deaths in the first 28 days per 1,000 *live births*.
- **Cholera attack rate** — cases per 1,000 population, because everyone is at risk.
- **Measles attack rate** — arguably per 1,000 *susceptible* people, and the susceptible population is exactly what an immunisation programme changes.

```python
under_five = population.loc[population["age_band"] == "0-4", "population"].sum()
u5_cases = (cases["age_band"] == "0-4").sum()

print(f"under-five attack rate: {1000 * u5_cases / under_five:.2f} per 1,000")
print(f"all-age attack rate:    {1000 * len(cases) / total_population:.2f} per 1,000")
```

```r
under5 <- population |> filter(age_band == "0-4") |> summarise(sum(population)) |> pull()
u5_cases <- sum(cases$age_band == "0-4")

c(under_five = 1000 * u5_cases / under5,
  all_age = 1000 * nrow(cases) / sum(population$population))
```

Under-fives have roughly twice the all-age attack rate here. **That gap is the whole reason lesson 6 exists**: two districts with different age structures will differ on the all-age rate even if every age-specific rate is identical.

1.6 The check to run on any rate

Four questions, and a rate that cannot answer all four is not usable.

- **What is the numerator counting** — events, people, or episodes?
- **Who is in the denominator**, and were they all at risk?
- **Over what period**, and was everyone present for it?
- **What is the multiplier**, and is it the convention for this indicator?

```python
def describe_rate(numerator, denominator, period, multiplier, at_risk_note):
    return {
        "value": multiplier * numerator / denominator,
        "numerator": numerator, "denominator": denominator,
        "period": period, "multiplier": multiplier, "at_risk": at_risk_note,
    }
```

```r
describe_rate <- function(numerator, denominator, period, multiplier, at_risk) {
  list(value = multiplier * numerator / denominator, numerator = numerator,
       denominator = denominator, period = period, at_risk = at_risk)
}
```

That is the indicator reference sheet from module 3, compressed to the five fields an epidemiological measure cannot do without.

1.7 What comes next

One denominator is straightforward. The next lesson chains several together — the HIV and TB care cascades, where each step's denominator is the previous step's numerator, and the interesting number is which link loses the most.

CHAPTER 2

Denominators chained together

Lesson 2 of 8 · 135 min

A cascade is a sequence in which each step's denominator is the previous step's numerator. 1,850 protection cases become 757 reaching a service, and the useful number is not the 41% — it is which link lost the most.

2.1 The shape

The HIV treatment cascade is the best-known example and the structure is general:

```
people living with HIV
  -> diagnosed
    -> on treatment
      -> virally suppressed
```

Each arrow is a proportion, and **each step's denominator is the step before it.** That is what makes it a cascade rather than four separate indicators, and it is what makes the arithmetic worth doing carefully.

The TB care cascade has the same shape — estimated incident cases, notified, started on treatment, successfully treated — and so does every referral pathway, every immunisation schedule, and every programme where people can leave between steps.

2.2 Two ways to express each step

```
import pandas as pd

referrals = pd.read_csv("protection-referrals-2024.v1.csv")

steps = [
    ("cases", len(referrals)),
    ("consented", (referrals["consent_to_refer"] == True).sum()),
    ("referral made", ((referrals["consent_to_refer"] == True)
                       & (referrals["referral_made"] == True)).sum()),
    ("reached a service", ((referrals["consent_to_refer"] == True)
                           & (referrals["referral_made"] == True)
                           & (referrals["referral_accepted"] == True)).sum()),
]

first = steps[0][1]
previous = None
for name, count in steps:
    line = f"{name:20} {count:5}  {count / first:6.1%} of all"
    if previous:
        line += f"   {count / previous:6.1%} of previous"
    print(line)
    previous = count
```

```
library(dplyr)
```

```
referrals |>
  summarise(
    cases = n(),
    consented = sum(consent_to_refer),
    referred = sum(consent_to_refer & referral_made),
    reached = sum(consent_to_refer & referral_made & referral_accepted)
  )
```

Step	n	Of all	Of previous
Cases	1,850	100%	—
Consented to referral	1,638	88.5%	88.5%
Referral made	1,142	61.7%	**69.7%**
Reached a service	757	40.9%	66.3%

Both columns are needed and they say different things.

The *of all* column is the cascade as usually drawn — a staircase descending from 100% to 41%. It tells a donor how much of the caseload completes.

The *of previous* column is the one that identifies the problem. The largest single loss is at "referral made": 30.3% of people who consented never had a referral issued. That is a step inside the organisation's control, and it is invisible in the first column, where the drop from 88.5% to 61.7% looks similar to the drop from 61.7% to 40.9%.

Rank by conditional loss, act on the largest.

```
losses = pd.DataFrame(steps, columns=["step", "n"])
losses["lost"] = losses["n"].shift(1) - losses["n"]
losses["conditional_loss"] = losses["lost"] / losses["n"].shift(1)
print(losses.dropna().sort_values("conditional_loss", ascending=False))
```

```
tibble::tibble(step = c("consented", "referred", "reached"),
               n = c(1638, 1142, 757), previous = c(1850, 1638, 1142)) |>
  mutate(conditional_loss = (previous - n) / previous) |>
  arrange(desc(conditional_loss))
```

2.3 The first denominator is the hard one

Every cascade has one step that cannot be counted, only estimated, and it is always the first.

- **HIV**: people living with HIV. Nobody counts them; the figure comes from a model, and every "% diagnosed" inherits that model's uncertainty.
- **TB**: estimated incident cases. Same, and the gap between estimated and notified is the headline finding of most national TB reports.
- **Protection**: people who experienced an incident. Unknowable, and the reason the referral cascade here starts at *cases known to the system* rather than at need.

```
print("cascade base: cases known to the system, not people in need")
```

```
# The base of this cascade is cases the system saw. It is not incidence.
```

Say which base you used. A cascade starting from an estimated need and one starting from cases already in the system look identical and mean completely different things — the first measures whether the system reaches people, the second only whether it processes them.

2.4 The immunisation cascade

The vaccination extract carries one, and it needs no modelled denominator because every step is counted.

```python
vax = pd.read_csv("vaccination-coverage-2024.v1.csv")
reported = vax[vax["report_submitted"] == True]
doses = reported.groupby("antigen")["doses_administered"].sum()

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

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

Penta1 to penta3: 13.8% dropout. Measles first to second dose: 22.9%. Two things that makes possible and a coverage figure does not.

It is robust to the denominator. Numerator and denominator come from the same facilities in the same months, so a census projection nine years old cannot move it. The indicator design course made this argument; here it is the reason dropout is often the better indicator to report.

It measures the mechanism. Coverage rises if more children start; dropout falls only if more children who start also finish, which is what a defaulter-tracing or reminder intervention actually changes.

2.5 Watch the direction of the dropout

```python
if doses["penta3"] > doses["penta1"]:
    print("negative dropout: check the denominator and the period")
```

```r
if (doses$doses[doses$antigen == "penta3"] > doses$doses[doses$antigen == "penta1"])
  warning("negative dropout")
```

A negative dropout — more third doses than first — is arithmetically possible and always means something specific: a catch-up campaign, a period boundary problem, or children vaccinated elsewhere for the earlier dose. It is not an error to be clipped to zero; it is a finding to be explained.

2.6 Draw it, but report the table

A cascade chart is one of the few genuinely good default visualisations in this sector — the descending bars make the losses immediate. But the chart shows only the *of all* column, so publish the table beside it with both.

```python
cascade = pd.DataFrame({
    "step": ["Cases", "Consented", "Referral made", "Reached service"],
    "n": [1850, 1638, 1142, 757],
})
cascade["of_all"] = cascade["n"] / cascade["n"].iloc[0]
cascade["of_previous"] = cascade["n"] / cascade["n"].shift(1)
cascade["base"] = "cases known to the system"
```

```r
tibble::tribble(
  ~step,             ~n,
  "Cases",         1850,
  "Consented",     1638,
  "Referral made", 1142,
  "Reached",        757
) |> mutate(of_all = n / first(n), of_previous = n / lag(n))
```

Four columns, and the last one — the base — is what stops a reader comparing your cascade to somebody else's that started somewhere different.

2.7 What comes next

A cascade is a set of counts already aggregated. The next unit goes back to individual records: an outbreak line list, where the analysis starts by putting 975 cases in time order and asking what shape they make.

Part II

An outbreak, one case at a time

CHAPTER 3

The curve is not the outbreak

Lesson 3 of 8 · 135 min

975 cases, 47 without an onset date, and one district whose register is back-filled from the admission book — so its median onset-to-admission delay reads as zero while its case fatality is the highest of the three.

3.1 What the curve is for

An epidemic curve is cases by date of onset. It is the first thing drawn in any outbreak and it answers three questions no table answers as fast: **is it growing, where is the peak, and is the shape consistent with a point source or with person-to-person spread?**

It is also the analysis most easily broken by a missing field, which is what this lesson is really about.

3.2 Draw it

```python
import pandas as pd

cases = pd.read_csv("cholera-line-list-2024.v1.csv", parse_dates=["onset_date"])

with_onset = cases[cases["onset_date"].notna()]
print(f"{len(with_onset)} of {len(cases)} cases have an onset date")

curve = (
    with_onset.assign(week=with_onset["onset_date"].dt.to_period("W").dt.start_time)
    .groupby("week").size()
)
print(curve)
print(f"peak week {curve.idxmax().date()} with {curve.max()} cases")
```

```r
library(dplyr)

cases <- readr::read_csv("cholera-line-list-2024.v1.csv")

curve <- cases |>
  filter(!is.na(onset_date)) |>
  mutate(week = lubridate::floor_date(onset_date, "week")) |>
  count(week)
```

928 of 975 cases carry an onset date, and the curve peaks in the seventh week — the week ending 23 June — with 111 cases. **By onset, not by admission, not by date of entry** — the whole point of the curve is when people fell ill, and each of the other two dates is shifted by a different, unknown amount.

Use weekly bins for an outbreak of this length. Daily is too noisy to read; monthly hides the peak entirely.

3.3 What the missing dates do

Forty-seven cases have no onset date, and they are not scattered at random.

```python
missing = cases[cases["onset_date"].isna()]
print(missing.groupby("district").size())
print(f"{len(missing) / len(cases):.1%} of all cases")
```

```r
cases |> filter(is.na(onset_date)) |> count(district)
```

Three ways to handle them, and they give three different curves.

Exclude them. The default and usually right. The curve is then of 928 cases and the label must say so, because 975 will appear elsewhere in the same report.

Impute from admission date. Tempting and it shifts the whole curve later by the median delay, which flattens the rise and moves the apparent peak.

Plot them as a separate band. Honest and rarely done: a stacked bar with "onset unknown" in a distinct shade, placed at the admission week, so the reader sees both the curve and its uncertainty.

```python
imputed = cases.copy()
fallback = pd.to_datetime(imputed["admission_date"], errors="coerce")
imputed["onset_or_admission"] = imputed["onset_date"].fillna(fallback)

both = pd.DataFrame({
    "excluded": curve,
    "imputed": (imputed.assign(
        week=imputed["onset_or_admission"].dt.to_period("W").dt.start_time)
        .groupby("week").size()),
}).fillna(0).astype(int)
print(both)
```

```r
cases |>
  mutate(onset_or_admission = coalesce(onset_date, admission_date),
         week = lubridate::floor_date(onset_or_admission, "week")) |>
  count(week)
```

Whichever you choose, state the denominator on the chart. "n = 928 of 975 cases with a recorded onset date" is six words and it is the difference between a curve and a claim.

3.4 The district whose curve is not a curve

Now the finding this dataset was built for.

```python
delay = (
    pd.to_datetime(cases["admission_date"], errors="coerce") - cases["onset_date"]
).dt.days

by_district = cases.assign(delay=delay).dropna(subset=["delay"]).groupby("district")
print(by_district["delay"].median())
print((by_district["delay"].apply(lambda s: (s == 0).mean()) * 100).round(0))
```

```r
cases |>
  filter(!is.na(onset_date), !is.na(admission_date)) |>
```

```
    mutate(delay = as.integer(admission_date - onset_date)) |>
    summarise(median_delay = median(delay), zero_day = mean(delay == 0), .by = district)
```

District	Median delay	Same-day
Nord	**0 days**	**80%**
Centre	2 days	13%
Sud	2 days	20%

Nord's cases appear to reach treatment the same day they fall ill, four times as often as anywhere else. That is not a strong health system. It is a register back-filled from the admission book: somebody enters the admission date in both fields because the onset date was never asked.

Two things follow, and both are serious.

Nord's epidemic curve is an admission curve. It is shifted right by however long its cases actually took to arrive, so comparing peak timing across districts compares two different quantities.

Nord's delay statistic is unusable, and it is the statistic the next lesson uses to explain case fatality — where Nord is worst at 6.30% against 2.00% in Centre. The district that appears to have the fastest access has the highest mortality, and the resolution is that the appearance is an artefact.

```
print("Nord: median delay 0 days, case fatality 6.30%")
print("Centre: median delay 2 days, case fatality 2.00%")
```

```
# The contradiction is the finding: the fastest-looking district dies most.
```

A contradiction between two indicators from the same register is a data question before it is an epidemiological one. The cleaning course's rule about columns that should agree, arriving in a setting where getting it wrong misdirects an outbreak response.

3.5 Reading the shape

Once the curve is trustworthy, three readings are standard.

- **A single sharp peak** suggests a point source — one contaminated water point, one event — with cases appearing within one incubation period of each other.
- **A series of peaks at roughly one incubation period apart** suggests person-to-person propagation.
- **A long plateau** suggests continuing common-source exposure, which for cholera usually means the water supply has not been fixed.

This outbreak rises over six weeks and decays over ten, which is a propagated shape rather than a point source. **Say which reading you are making and why**, and be careful: the shape is also affected by when the response started, and an intervention that works truncates the curve in a way that looks like natural decay.

3.6 Plot cases, not rates, on the curve

A last practical point. The epidemic curve is a count over time, not a rate. Dividing by population makes districts comparable and destroys the thing the curve is for, which is the absolute burden arriving at treatment centres each week.

Plot counts. Put the rates in the table beside it, which is the next lesson.

3.7 What comes next

The curve says when. The next lesson says how much and how badly: attack rates with the population denominator this dataset ships, and case fatality against the 1% the response is judged on.

CHAPTER 4

Attack rates and the 1% nobody meets

Lesson 4 of 8 · 135 min

Attack rate needs the population; case fatality needs a denominator that excludes the case still admitted. 3.90% against a 1% target, 6.30% in one district — and the explanation is a delay statistic the previous lesson showed is broken.

4.1 Attack rate

The attack rate is cases over the population at risk, over the outbreak period. Despite the name it is a proportion, and it needs the population file the line list ships with.

```python
import pandas as pd

cases = pd.read_csv("cholera-line-list-2024.v1.csv")
population = pd.read_csv("district-population-2024.v1.csv")

by_district = (
    cases.groupby("district").size().rename("cases")
    .to_frame()
    .join(population.groupby("district")["population"].sum())
)
by_district["per_1000"] = 1000 * by_district["cases"] / by_district["population"]
print(by_district.round(2))
```

```r
library(dplyr)

cases |> count(district, name = "cases") |>
  left_join(summarise(population, population = sum(population), .by = district),
            by = "district") |>
  mutate(per_1000 = 1000 * cases / population)
```

District	Cases	Population	Per 1,000
Nord	381	48,000	**7.94**
Centre	401	62,000	6.47
Sud	193	35,000	**5.51**

Nord is the worst at 7.94 and Sud the best at 5.51, a gap of 2.43 per 1,000. Hold that number: lesson 6 shows that half of it is not what it appears to be.

Note also that the case counts do **not** rank the same way as the rates — Centre has the most cases and is not the worst-affected. **A count answers "where do we send supplies"; a rate answers "where is the risk highest".** Report both, and never let a bar chart of counts be read as a map of risk.

4.2 Age-specific attack rates

```python
by_band = (
    cases.groupby(["district", "age_band"]).size().rename("cases")
    .to_frame()
    .join(population.set_index(["district", "age_band"])["population"])
)
by_band["per_1000"] = 1000 * by_band["cases"] / by_band["population"]
print(by_band["per_1000"].unstack().round(2))
```

```r
cases |> count(district, age_band, name = "cases") |>
  left_join(population, by = c("district", "age_band")) |>
  mutate(per_1000 = 1000 * cases / population) |>
  tidyr::pivot_wider(id_cols = age_band, names_from = district, values_from = per_1000)
```

Age band	Nord	Centre	Sud
0-4	14.77	12.69	11.69
5-14	8.26	8.06	6.29
15-44	3.93	4.03	3.51
45+	6.41	4.75	5.71

Under-fives are hit three to four times as hard as adults of working age, in every district. That gradient is the substantive epidemiology, and it is also the reason the crude comparison above is misleading — Nord has twice the share of under-fives that Sud has.

4.3 Case fatality

Case fatality is deaths among cases. Two decisions in the denominator, and both have been made in this course before.

```python
with_outcome = cases[cases["outcome"].notna()]

print(f"cases: {len(cases)}, with an outcome: {len(with_outcome)}")
print(f"CFR: {(with_outcome['outcome'] == 'died').mean():.2%}")
```

```r
cases |> filter(!is.na(outcome)) |>
  summarise(n = n(), cfr = mean(outcome == "died"))
```

3.90% on 974 cases with a recorded outcome.

One case was still admitted at the cut-off and has no outcome. It is neither a death nor a recovery, and the same reasoning applies as to the seventy-one children still in CMAM treatment in the nutrition course: **a pending outcome is not an outcome**, and the denominator has to say which it used. Here it moves the figure by nothing; in an outbreak still running it moves it a great deal.

4.4 The threshold

Sphere and WHO treat **case fatality below 1%** as the mark of a well-managed cholera response. Untreated cholera kills a large share of severe cases; treated promptly with oral rehydration it kills almost nobody. The threshold is therefore a statement about access to treatment rather than about the pathogen.

```python
TARGET = 0.01
cfr = (with_outcome["outcome"] == "died").mean()
print(f"CFR {cfr:.2%} against a target of {TARGET:.0%}: "
      f"{'meets' if cfr < TARGET else 'does not meet'} the standard")

by_district_cfr = with_outcome.groupby("district")["outcome"].apply(
    lambda s: (s == "died").mean()
)
print((by_district_cfr * 100).round(2))
```

```r
cases |> filter(!is.na(outcome)) |>
  summarise(cfr = mean(outcome == "died"), n = n(), .by = district)
```

District	CFR	n
Nord	**6.30%**	381
Sud	3.11%	193
Centre	2.00%	400

Every district is above 1%, and Nord is more than six times it. **This is the finding of the outbreak** — not the attack rate, which is a fact about exposure, but the case fatality, which is a fact about the response.

4.5 The explanation, and why it is unavailable

The standard explanation for high cholera case fatality is delay to treatment, and the line list has the fields to test it.

```python
delay = (
    pd.to_datetime(cases["admission_date"], errors="coerce")
    - pd.to_datetime(cases["onset_date"], errors="coerce")
).dt.days

testable = cases.assign(delay=delay).dropna(subset=["delay", "outcome"])
banded = testable.assign(
    band=pd.cut(testable["delay"], [-1, 1, 3, 99], labels=["0-1", "2-3", "4+"])
)
print(banded.groupby("band")["outcome"].apply(lambda s: (s == "died").mean()).round(3))
```

```r
cases |>
  filter(!is.na(onset_date), !is.na(admission_date), !is.na(outcome)) |>
  mutate(delay = as.integer(admission_date - onset_date),
         band = cut(delay, c(-1, 1, 3, 99), labels = c("0-1", "2-3", "4+"))) |>
  summarise(cfr = mean(outcome == "died"), n = n(), .by = band)
```

Among admitted cases the gradient is there but modest — about 1.9% at nought to three days and 3.3% at four or more. It is smaller than the gap between districts, and there are two reasons why, one substantive and one not.

The substantive one: admission itself is the protection. Cases never admitted carry most of the mortality, and the delay variable exists only for those who were admitted. Comparing delay bands within admitted cases conditions on the thing that matters most.

The one that is not: Nord's delay is fabricated by its register. The previous lesson found 80% of its cases recorded with onset equal to admission. So the district with the

highest case fatality reports the shortest delay, and the comparison that would explain its mortality is precisely the comparison its data cannot support.

```
print("Nord CFR 6.30% with a median recorded delay of 0 days.")
print("The delay is a register artefact; the CFR is not.")
```

```
# One of these two numbers is real. Say which, in the report.
```

Write that up as a limitation, not as a result. "Case fatality is highest in Nord; the onset-to-admission delay that would test the usual explanation is not reliable in that district, because 80% of its cases record onset and admission on the same day" is the honest sentence, and it also generates the corrective action — fix the register — that a fabricated explanation would not.

4.6 The reporting block

```
Cholera outbreak, weeks 1-16

  Cases                      975      attack rate 6.7 per 1,000 (145,000 population)
  Attack rate by district    7.94 / 6.47 / 5.51 per 1,000 (Nord / Centre / Sud)
                             crude; see standardised rates before comparing

  Deaths                      38      case fatality 3.90%
  Denominator                974      one case still admitted at cut-off, excluded
  Against Sphere target       1%      not met in any district

  Highest CFR: Nord at 6.30% (n=381). Onset-to-admission delay is not usable
  in Nord (80% of cases record onset = admission), so the usual explanation
  cannot be tested there from this register.
```

Ten lines, and the last three are the ones that make it a finding rather than a table.

4.7 What comes next

Attack rates by district look like a comparison and are not one yet, because the districts do not have the same people in them. The next unit fixes that, after first settling a coverage question that has been open since module 2.

Part III

Comparing places

CHAPTER 5

Three coverages that disagree

Lesson 5 of 8 · 135 min

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.

5.1 Three ways to measure the same thing

Immunisation coverage is the most-reported health indicator in this sector and it is produced three different ways, which routinely disagree.

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

The DHIS2 course computed the first. The survey course gave you the machinery for the second. This lesson is why they differ and which to use.

5.2 Administrative coverage, and its two dependencies

```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%}")
```

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

77.5%, among reporting facility-months. That figure carries two dependencies and the DHIS2 course established both.

The denominator is a projection. Surviving infants, from a census projection compounded forward at an assumed growth rate. Nine years at 2.4% is a factor of 1.24, so a quarter of the denominator is an assumption.

The reporting rate was 76.5%. Facilities that did not report contribute no doses, so the figure is a lower bound unless their doses are imputed.

Administrative coverage above 100% is common and always means one of those two is wrong, plus a third possibility — children vaccinated outside their catchment, which inflates a facility numerator against a catchment denominator.

5.3 Survey coverage, and its two dependencies

A coverage survey samples children of a given age and asks whether they were vaccinated, verified against a card where one exists.

It removes the projection problem — the denominator is the sample — and introduces two others.

Card retention. Where cards are lost, coverage rests on caregiver recall, which overstates for some antigens and understates for others. Report card-verified and recall-based coverage separately; the gap between them is the measure of how much the estimate depends on memory.

Precision. Survey coverage arrives with a confidence interval, and the survey course showed that a cluster design widens it. A survey that estimates coverage to plus or minus six points cannot settle whether a district crossed an 80% target.

```
# 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))
```

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

5.4 Why they disagree

Five reasons, and knowing which applies changes what you do.

- **The projection is wrong.** The commonest, and it moves administrative coverage only. A denominator too small pushes administrative above survey.
- **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 months who were vaccinated at any time. These are not the same children.

The last one is the most common and the least suspected. Two figures for "penta3 coverage" that refer to different cohorts are not two estimates of one quantity; they are two quantities.

5.5 Dropout: the figure that survives all of it

```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%}")
```

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

Penta1 to penta3: 13.8%. Measles first to second dose: 22.9%.

Dropout is a ratio of two numerators from the same source, the same facilities and the same months. It therefore has:

- **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.

The cost is that it says nothing about level. A district could have 13.8% dropout and 30% coverage — good at retaining the children it starts, bad at starting them. **Report dropout beside coverage, never instead of it.**

5.6 What the two dropouts say together

Measles second-dose dropout is 22.9% against 13.8% for pentavalent, and the difference is informative rather than noise.

The pentavalent doses are given close together in the first months of life, when carers are already attending for other services. The measles second dose is given much later, often after the intensive contact period has ended. **A dropout that rises with the interval between doses is a reminder-and-outreach problem**, not a supply problem, and the corrective actions differ.

```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"],
})
```

```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"
)
```

5.7 Which to report

```
Penta3 coverage, administrative     77.5%   among reporting facility-months
                                            (reporting rate 76.5%)
                                            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
```

Four lines, and the last one is a finding. **Where no survey exists, say so** — because the reader's default assumption is that a coverage figure has been validated against one, and here it has not.

5.8 What comes next

Coverage compares a programme against a population. The next lesson compares two populations against each other, and finds that the comparison in lesson 4 was partly an artefact of who lives where.

CHAPTER 6

Half the difference was who lives there

Lesson 6 of 8 · 135 min

Nord's crude attack rate is 7.94 per 1,000 and Sud's is 5.51. Standardised to a common population they are 7.25 and 5.95, so the gap falls from 2.43 to 1.30 — and the missing half was age structure.

6.1 Two districts are not comparable as they stand

Lesson 4 put three crude attack rates side by side and the comparison looked straightforward. It is not, and the reason is in the population file.

```python
import pandas as pd

population = pd.read_csv("district-population-2024.v1.csv")

structure = (
    population.pivot(index="district", columns="age_band", values="population")
)
shares = structure.div(structure.sum(axis=1), axis=0)
print((shares * 100).round(1))
```

```r
library(dplyr)

population |>
  mutate(share = population / sum(population), .by = district) |>
  tidyr::pivot_wider(id_cols = district, names_from = age_band, values_from = share)
```

District	0-4	5-14	15-44	45+
Nord	**22%**	30%	35%	13%
Centre	15%	25%	42%	18%
Sud	**11%**	20%	44%	25%

Nord has twice the share of under-fives that Sud has, and lesson 4 established that under-fives have three to four times the attack rate of working-age adults.

So Nord would have a higher crude rate than Sud **even if every age-specific rate in the two districts were identical**. The crude comparison mixes two things: how risky each district is, and who lives in it.

6.2 Direct standardisation

The method is one line of arithmetic applied consistently: **compute each district's age-specific rates, then apply them to one common population.**

```python
cases = pd.read_csv("cholera-line-list-2024.v1.csv")

observed = (
```

```python
    cases.groupby(["district", "age_band"]).size().rename("cases").to_frame()
    .join(population.set_index(["district", "age_band"])["population"])
)
observed["rate"] = observed["cases"] / observed["population"]

standard = population.groupby("age_band")["population"].sum()
standard_share = standard / standard.sum()

standardised = (
    observed["rate"].unstack()          # district x age_band
    .mul(standard_share, axis=1).sum(axis=1)
)
crude = (
    cases.groupby("district").size()
    / population.groupby("district")["population"].sum()
)

comparison = pd.DataFrame({
    "crude_per_1000": 1000 * crude,
    "standardised_per_1000": 1000 * standardised,
})
comparison["difference"] = (
    comparison["standardised_per_1000"] - comparison["crude_per_1000"]
)
print(comparison.round(2))
```

```r
observed <- cases |> count(district, age_band, name = "cases") |>
  left_join(population, by = c("district", "age_band")) |>
  mutate(rate = cases / population)

standard <- population |> summarise(pop = sum(population), .by = age_band) |>
  mutate(share = pop / sum(pop))

observed |>
  left_join(standard, by = "age_band") |>
  summarise(standardised = sum(rate * share), .by = district)
```

District	Crude	Standardised	Change
Nord	7.94	**7.25**	−0.69
Centre	6.47	6.60	+0.13
Sud	5.51	**5.95**	+0.44

Nord falls, Sud rises, and the gap between them narrows from 2.43 to 1.30 per 1,000. About half the apparent difference between the worst and the best district was age structure rather than risk.

The remaining 1.30 is real. Nord is genuinely worse — its age-specific rates are higher in three of four bands — and standardisation is what lets you say that rather than assert it.

6.3 What the standard population is, and why it matters

The standard is whatever population you apply every district's rates to. Three common choices:

- **The combined study population**, as above. Simple, defensible, and internal — the standardised rates are comparable to each other and to nothing else.

- **A national population.** Lets you compare against other districts standardised the same way.
- **A published world standard** — the WHO or Segi world standard population. Lets you compare internationally, and produces numbers that look nothing like the crude rates.

State the standard. A standardised rate is only interpretable against others standardised to the same population, and two reports using different standards produce incomparable numbers that both look official.

```
print("Standard: combined population of the three districts, 145,000")


# Say it in the table caption, every time.
```

6.4 Indirect standardisation, and when you need it

Direct standardisation needs age-specific rates for every district, which needs enough cases in every cell. Where a district has four cases in a band, its age-specific rate is unstable and the direct method propagates that instability.

The indirect method inverts the problem: apply a **standard set of rates** to each district's own population, and compare observed cases to expected.

```
standard_rates = observed.groupby("age_band").apply(
    lambda g: g["cases"].sum() / g["population"].sum()
)

expected = (
    population.assign(rate=population["age_band"].map(standard_rates))
    .assign(expected=lambda d: d["population"] * d["rate"])
    .groupby("district")["expected"].sum()
)
smr = cases.groupby("district").size() / expected
print(smr.round(3))
```

```
standard_rates <- observed |>
  summarise(rate = sum(cases) / sum(population), .by = age_band)

population |>
  left_join(standard_rates, by = "age_band") |>
  summarise(expected = sum(population * rate), .by = district) |>
  left_join(count(cases, district, name = "observed"), by = "district") |>
  mutate(smr = observed / expected)
```

The result is a **standardised morbidity ratio**: observed over expected, where 1.0 means the district has exactly the cases its age structure predicts. Above 1 is worse than expected, below is better.

Use indirect standardisation when cells are small, and say which method you used — the two answer slightly different questions and their numbers are not interchangeable.

6.5 Age is not the only confounder

Standardisation removes the variable you standardise on and nothing else. Cholera attack rates also vary with water source, population density, distance to a treatment centre and

displacement status, and none of those is in the population file.

Standardising on age and then claiming the remaining difference is programme performance is the error this lesson creates the opportunity for, and the next lesson is entirely about it.

6.6 Report the pair

```
Cholera attack rate by district, weeks 1-16

  District   Crude    Age-standardised   Population
  Nord       7.94     7.25               48,000
  Centre     6.47     6.60               62,000
  Sud        5.51     5.95               35,000

  Standardised directly to the combined population of the three districts.
  About half the crude Nord-Sud difference is age structure: Nord has 22% of
  its population under five against Sud's 11%, and under-fives have three to
  four times the attack rate of adults aged 15-44.
```

Both columns, the standard named, and one sentence saying how much moved and why. A table with only the standardised column hides the fact that anything was adjusted; a table with only the crude column invites a comparison that is half demographic.

6.7 What comes next

Standardisation removed one alternative explanation. The next lesson lists the others — and asks what a difference between two districts is allowed to be attributed to when none of them can be removed.

Part IV

What the comparison can claim

CHAPTER 7

What else could explain it

Lesson 7 of 8 · 135 min

A confounder causes the outcome and differs between the groups. Four of them sit between the districts in this outbreak, one has been removed, one is unmeasured, and one is not a confounder at all — it is the mechanism.

7.1 The definition, and why it is worth being strict about

A **confounder** is a variable that satisfies three conditions at once:

- it is **associated with the exposure** — it differs between the groups being compared;
- it is a **cause of the outcome**, independently of the exposure;
- it is **not on the causal path** between the exposure and the outcome.

All three matter. A variable that differs between groups but does not cause the outcome is irrelevant. A variable that causes the outcome but is the same in both groups cannot explain a difference. And a variable on the causal path is not a confounder — it is the mechanism, and adjusting for it destroys the finding.

7.2 The four candidates in this outbreak

Nord has a standardised attack rate of 7.25 per 1,000 against Sud's 5.95, and case fatality of 6.30% against 3.11%. What else could explain those gaps?

Candidate	Differs between districts?	Causes the outcome?	On the path?
Age structure	Yes, sharply	Yes	No — a confounder, alrea
Water source and density	Almost certainly	Yes	No — a confounder, **unm**
Distance to treatment	Yes	Yes, for fatality	**Yes** — it is the mechanis
Register quality	Yes	No	No — not a confounder, a

Work down that table and each row needs a different response.

Age structure has been handled. Lesson 6 removed it and quantified what it was worth — about half the crude gap.

Water source, sanitation and crowding are the actual causes of cholera transmission, they certainly differ between a camp-like district and a rural one, and **they are not in this dataset**. That is the most important sentence in this lesson. An unmeasured confounder cannot be adjusted for, and its existence has to be stated rather than ignored.

```
print("Available for adjustment: age, sex, district")
print("Known to matter and unavailable: water source, sanitation, crowding, "
      "displacement status")
```

```
# The list of what you could not adjust for belongs in the limitations section.
```

Distance to treatment is the interesting one and it is not a confounder at all. The chain is: Nord is further from treatment centres → its cases arrive later → more of them die. Distance causes fatality *through* delay, so delay is on the causal path. **Adjusting for delay would remove the very effect you are trying to demonstrate**, which is the classic over-adjustment error.

Register quality is not a confounder either. Nord's back-filled onset dates do not cause deaths; they distort the measurement of the explanation. That is information bias, and it is fixed by fixing the register, not by adjusting.

7.3 Stratify to see it

The simplest adjustment, and the one that shows its working.

```python
import pandas as pd

cases = pd.read_csv("cholera-line-list-2024.v1.csv")
with_outcome = cases[cases["outcome"].notna()]

stratified = (
    with_outcome.assign(died=with_outcome["outcome"] == "died")
    .groupby(["district", "age_band"])
    .agg(cases=("died", "size"), deaths=("died", "sum"))
)
stratified["cfr"] = stratified["deaths"] / stratified["cases"]
print((stratified["cfr"].unstack() * 100).round(1))
```

```r
library(dplyr)

cases |>
  filter(!is.na(outcome)) |>
  summarise(cases = n(), cfr = mean(outcome == "died"), .by = c(district, age_band)) |>
  tidyr::pivot_wider(id_cols = age_band, names_from = district, values_from = cfr)
```

Two things to read off a stratified table, and only one of them is the adjusted estimate.

Does the difference persist within strata? If Nord's case fatality is higher in every age band, age is not the explanation. If it reverses in some bands, something more interesting is happening.

Are the strata consistent? A difference that is large in one band and absent in others is **effect modification** — the exposure genuinely acts differently in different groups — and it must be reported by stratum rather than averaged into a single adjusted figure.

Effect modification is a finding; confounding is a nuisance. Collapsing the first into a single number destroys the result; failing to remove the second manufactures one.

7.4 When stratification runs out

Two variables and the cells empty fast — the disaggregation lesson from module 3, arriving with a different consequence. Three districts by four age bands by two sexes is twenty-four cells on 974 outcomes, and cholera deaths are rare.

```python
cells = with_outcome.groupby(["district", "age_band", "sex"]).size()
print(f"{len(cells)} cells, smallest {cells.min()}, {(cells < 30).sum()} below 30")
```

```r
cases |> filter(!is.na(outcome)) |> count(district, age_band, sex) |>
```

```
    summarise(cells = n(), smallest = min(n), under_30 = sum(n < 30))
```

That is where regression takes over, and *Regression for Programme Data* later in the programme is about exactly this — adjusting for several variables at once without running out of cells. **Its purpose is the same as stratification's**, and a model that adjusts for a mechanism makes the same error as a stratification that does.

7.5 The three questions to ask of any comparison

Before attributing a difference to anything:

1. What differs between these groups other than the thing I am studying? List them. The list is the limitations section, and the ones you cannot measure belong in it too.

2. Which of them cause the outcome? Only those are confounders. A district having more schools does not confound a cholera comparison unless schools cause cholera.

3. Which of them are on the causal path? Do not adjust for those. Delay to treatment, in this outbreak, is how distance kills.

```
adjustment_plan = pd.DataFrame({
    "variable": ["age", "sex", "water source", "delay to treatment", "register quality"],
    "differs": [True, False, True, True, True],
    "causes_outcome": [True, True, True, True, False],
    "on_causal_path": [False, False, False, True, False],
    "action": ["adjust", "no need", "unmeasured; state it", "do not adjust",
               "fix the register"],
})
print(adjustment_plan)
```

```
tibble::tribble(
  ~variable,        ~action,
  "age",            "adjust",
  "water source",   "unmeasured; state it",
  "delay",          "do not adjust; it is the mechanism",
  "register",       "fix it; this is bias, not confounding"
)
```

Write that table before the analysis, not after. Deciding what to adjust for once you have seen which adjustment gives a nicer answer is the thing that makes observational epidemiology untrustworthy, and it is indistinguishable from honest work after the fact.

7.6 What comes next

You have named what else could explain a difference and removed what you could. The last lesson is what remains sayable — the sentence an observational comparison is entitled to, and the several it is not.

CHAPTER 8

The sentence this analysis is entitled to

Lesson 8 of 8 · 135 min

Nord's case fatality is 6.30% against Centre's 2.00%, and the delay gradient rests on one death among thirty cases. Four claims could be written from that, three of them are wrong, and the difference between them is the whole of this course.

8.1 Four sentences, one set of numbers

Everything this course computed points at one comparison, and there are four ways to write it up.

1. *"Nord's cholera response is failing: case fatality there is three times Centre's."*
2. *"Late presentation causes death — cases admitted four or more days after onset die at 3.3% against 1.9% for those admitted sooner."*
3. *"Case fatality in Nord is 6.30% (24 deaths among 381 cases with a recorded outcome) against 2.00% in Centre (8 of 400). Both exceed the 1% Sphere threshold. Onset-to-admission delay does not explain the gap in this data: Nord's onset dates are back-filled from the admission book, and the highest delay band carries one death among thirty cases."*
4. *"The data cannot support a comparison between districts."*

The third is the only defensible one, and it is also the only one long enough to be useful. Work out why each of the others fails and you have the content of this lesson.

8.2 Why the first three fail

"Nord's response is failing" attributes a difference to a cause the data does not distinguish. Nord differs from Centre in age structure, water source, distance to treatment, displacement status and register quality. Lesson 6 removed one of those, lesson 7 named the rest and found half of them unmeasured. **A comparison with one confounder removed is not a comparison with none.**

"Late presentation causes death" is a stronger claim than the design supports, and it is built on two cells that cannot carry it. Onset-to-admission delay is measured from a field one district back-fills — **80% of Nord's cases show a same-day delay**, which is the admission book talking, not a fast presentation. And the four-or-more-day band that produces the 3.3% contains **one death among thirty cases**: move that single case and the gradient disappears.

```
import pandas as pd

cases = pd.read_csv("cholera-line-list-2024.v1.csv", parse_dates=["onset_date"])
with_onset = cases[cases["onset_date"].notna()]

delayed = with_onset.assign(delay=lambda d: (
    pd.to_datetime(d["admission_date"]) - d["onset_date"]).dt.days)
```

```
print(delayed.groupby("district")["delay"]
      .agg(median="median", same_day=lambda s: (s == 0).mean()).round(3))

band = pd.cut(delayed["delay"], [-1, 1, 3, 99], labels=["0-1", "2-3", "4+"])
print(delayed.dropna(subset=["outcome"]).groupby(band).size())
```

```
library(dplyr)

cases |>
  filter(!is.na(onset_date)) |>
  mutate(delay = as.numeric(admission_date - onset_date)) |>
  summarise(median = median(delay), same_day = mean(delay == 0), .by = district)
```

Nord's median delay is nought days with 80% same-day; Centre's and Sud's are two days. And the three bands hold 372, 214 and 30 cases with an outcome. **Always print the cell counts under a gradient**, because a percentage prints identically whether it rests on three hundred cases or on one.

"The data cannot support a comparison" is the failure mode of the cautious analyst and it is the most expensive of the four. A response that stops reporting because its data is imperfect leaves the decision to be made on nothing, and there is no outbreak anywhere with clean data. **The job is to say what the data supports, with its limits attached, not to refuse.**

8.3 The four things a claim needs

Every one of them is missing from at least one of the first three sentences.

The numerator and the denominator, as counts. "6.30%" is unauditable; "24 deaths among 381 cases with a recorded outcome" can be checked and can be added to another district's. It also declares the exclusion — 974 of 975 cases have an outcome, and saying which denominator you used is how a reader knows whether the one still in treatment was counted.

The comparison it is against. A rate with no reference point is a number looking for an opinion. Sphere puts cholera case fatality below 1%; 3.90% is four times that and the sentence should say so.

The adjustment, or its absence. If you standardised, name the standard. If you did not, say the comparison is crude.

The limitation that could change the direction. Not a paragraph of hedging — the one or two things that, if they went the other way, would change what someone should do. Nord's back-filled dates are one. The unmeasured water source is the other.

```
claim = {
    "numerator": 24,
    "denominator": 381,
    "denominator_definition": "cases with a recorded outcome",
    "value": "6.30%",
    "benchmark": "Sphere: below 1%",
    "adjustment": "none; crude",
    "limitation": "onset dates back-filled in this district",
}
```

```
# The same fields, in the footnote of the table.
```

8.4 Strength of evidence, and where routine data sits

Not all observational findings are equally weak, and the differences have names.

Design	What it can claim	Where it appears here
Routine data, cross-section	Association, adjusted for what was measured	This whole course
Cohort	Association with the exposure preceding the outcome	The CMAM register, follow
Case-control	Association, efficiently for rare outcomes	Investigating the source of
Randomised trial	Causation	Almost never in programm

Programme data lives on the top row, and the top row can carry a great deal of weight if it is honest about which row it is on. What it cannot do is jump to the bottom one because the association is large or the mechanism is plausible.

Two of the strongest arguments available to observational work are still open to you, and this outbreak supplies one of each.

A **finding that survives adjustment** for everything you could measure is more credible than one that vanishes. Nord's excess narrowed under age standardisation and did not disappear, which is the stronger of the two results here.

A **dose–response gradient** — fatality rising steadily across delay bands rather than jumping at one cut-point — is harder to explain away than a single contrast. This is the argument that was **not** available: the top band held thirty cases, and a gradient resting on one death is not a gradient.

Neither, taken alone, makes a finding causal. Knowing which of them you have is what tells you how firmly to write.

8.5 Write the limitation as a decision, not an apology

The limitations paragraph most reports carry is a list of regrets. Make it a list of consequences instead.

```
Cholera case fatality, weeks 1-16

  Overall     3.90%   38 deaths / 974 cases with an outcome
  Nord        6.30%   24 / 381
  Sud         3.11%    6 / 193
  Centre      2.00%    8 / 400

  Against the Sphere threshold of 1%, every district is above standard.

  What this supports: prioritising treatment capacity in Nord.
  What it does not: attributing the gap to clinical management or to late
  presentation. Nord's register back-fills onset dates, so its delays are not
  comparable, and the 4+ day band holds one death among 30 cases. Water
  source and crowding, the main determinants of cholera severity, are not
  recorded at all.
  What would settle it: two weeks of prospectively recorded onset dates in
  Nord, and the WASH assessment already scheduled for week 18.
```

"What would settle it" is the line that makes the limitation useful. A limitation with no route out reads as an excuse; one with a named next step is a work plan, and it is what turns an imperfect analysis into the first half of a better one.

8.6 The habit this course was for

Three sentences, in this order, before writing anything up.

1. **What did I count, over what?** Numerator, denominator, and who was excluded.
2. **What else could explain it?** The table from lesson 7, written before the analysis rather than after.
3. **What should someone do differently because of this?** If nothing, the analysis was not worth the time, and if something, the claim has to be strong enough to bear it.

Every measure in this course — incidence and prevalence, the cascade, the epidemic curve, attack rates, case fatality, coverage three ways, the standardised rate — is an answer to the first. The last two are what stop the first from being misused, and they are the part no software computes for you.

8.7 What comes next

You can defend a rate, adjust a comparison and state what it is entitled to claim. The rest of module 4 applies the same discipline in other sectors — WASH service levels, education attendance, protection case management — and *Regression for Programme Data* comes back to lesson 7's problem with a tool that can hold several confounders at once.

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/public-health-epidemiology

Generated July 28, 2026.