Attack rates and the 1% nobody meets

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.

Pierre Robentz Cassion

Lesson 4 of 8

Public Health and Applied Epidemiology — An outbreak, one case at a time

What this lesson covers

- Attack rate
- Age-specific attack rates
- Case fatality
- The threshold
- The explanation, and why it is unavailable
- The reporting block
- What comes next

Attack rate — In Python

```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))
```

Attack rate — In R

```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)
```

Attack rate

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

Age-specific attack rates — In 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))
```

Age-specific attack rates — In 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-specific attack rates

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

Age-specific attack rates

- Under-fives are hit three to four times as hard as adults of working age — in every district

Case fatality — In 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%}")
```

Case fatality — In R

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

Case fatality

- 3.90% on 974 cases with a recorded outcome

The threshold — In 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))
```

The threshold — In R

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

The threshold

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

The explanation, and why it is unavailable — In Python

```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))
```

The explanation, and why it is unavailable — In R

```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)
```

The explanation, and why it is unavailable

- The substantive one: admission itself is the protection — Cases never admitted carry most of the mortality, and the...
- The one that is not: Nord's delay is fabricated by its register — The previous lesson found 80% of its cases recorded...

The explanation, and why it is unavailable — In Python

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

The explanation, and why it is unavailable — In R

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

The explanation, and why it is unavailable

- Write that up as a limitation, not as a result — "Case fatality is highest in Nord; the onset-to-admission delay that…

The reporting block — Example

```
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.
```

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.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/public-health-epidemiology/
epi-04-attack-rates-and-cfr