Rate, ratio, proportion — and person-time

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.

Pierre Robentz Cassion

Lesson 1 of 8

What this lesson covers

- Three shapes, one word
- Prevalence and incidence
- What person-time adds
- Say the denominator and the multiplier in the name
- The at-risk population is not always the whole population
- The check to run on any rate
- What comes next

Three shapes, one word

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

Three shapes, one word

- Only the third is a rate — and the difference is the time in the denominator

Prevalence and incidence

- **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.

Prevalence and incidence — In 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")
```

Prevalence and incidence — In R

```r
library(dplyr)

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

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

What person-time adds — In Python

```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")
```

What person-time adds — In R

```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))
```

What person-time adds

- **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.

Say the denominator and the multiplier in the name — In Python

```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,
}
```

Say the denominator and the multiplier in the name — In R

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

Say the denominator and the multiplier in the name

- The multiplier — Per 100, per 1,000, per 100,000
- The denominator's population — "Per 1,000 population" and "per 1,000 children under five" are different indicators

The at-risk population is not always the whole population

- **Maternal mortality ratio** — deaths per 100,000 *live births*, not per population, because only pregnancies are at. . .
- **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. . .

The at-risk population is not always the whole population — In 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")
```

The at-risk population is not always the whole population — In R

```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))
```

The check to run on any rate

- **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?

The check to run on any rate — In Python

```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,
    }
```

The check to run on any rate — In R

```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)
}
```

What comes next

- One denominator is straightforward.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/public-health-epidemiology/
epi-01-rates-and-denominators