A survival rate you should not publish

Chain one year's promotion rates and 19.3% of children reach the final grade. Count repeaters as retained and it is 40.3%. Both come from the same file, and neither is a completion rate.

Pierre Robentz Cassion

Lesson 6 of 8

Education Programme Analysis — The cohort

What this lesson covers

- The calculation everyone reaches for
- Why it is wrong
- What to compute instead
- Who leaves
- The missing data is not missing at random
- Report it as a risk profile
- What comes next

The calculation everyone reaches for — In Python (cont.)

```
import pandas as pd

enrolment = pd.read_csv("school-enrolment-2024.v1.csv")
clean = enrolment[enrolment["age_years"] <= 20]
cohort = clean[clean["school_year"] == 2023]

survival, cumulative = {}, 1.0
for grade in range(1, 7):
    grade_rows = cohort[cohort["grade"] == grade]
    promoted = grade_rows["end_of_year_status"].isin(
        ["promoted", "completed-final-grade"]).mean()
    survival[grade] = cumulative
    cumulative *= promoted
    print(f"grade {grade}: promotion {promoted:.1%}, "
          f"survival to here {survival[grade]:.1%}")
```

The calculation everyone reaches for — In Python (cont.)

```
print(f"\nsurvival to grade 6 entry: {cumulative:.1%}")
```

The calculation everyone reaches for — In R

```r
library(dplyr)

cohort |>
  filter(between(grade, 1, 6)) |>
  summarise(promotion = mean(end_of_year_status %in%
              c("promoted", "completed-final-grade")), .by = grade) |>
  arrange(grade) |>
  mutate(survival = cumprod(lag(promotion, default = 1)))
```

The calculation everyone reaches for

Grade	Promotion	Cumulative survival
1	89.3%	100.0%
2	83.0%	89.3%
3	71.7%	74.1%
4	76.2%	53.1%
5	70.9%	40.5%
6	67.3%	28.7%
—	—	**19.3%**

The calculation everyone reaches for

- 19.3% of children entering grade 1 reach the end of primary school — It is a striking number, it is what the method...

Why it is wrong

- Repeaters are counted as failures — A child who repeats grade 2 is not out of school — they are in grade 2 again

Why it is wrong — In Python

```python
promotion_or_repeat = cohort[cohort["grade"].between(1, 6)].groupby("grade")[
    "end_of_year_status"].apply(
    lambda s: s.isin(["promoted", "completed-final-grade", "repeated"]).mean())

still_enrolled = 1.0
for grade, rate in promotion_or_repeat.items():
    still_enrolled *= rate
print(f"survival counting repeaters as retained: {still_enrolled:.1%}")
```

Why it is wrong — In R

```
# Retention, not promotion, is what a survival rate needs.
```

Why it is wrong

- Transfers are counted as failures too — 2.9% left for another school and the chain reads them as leaving education
- A single year is assumed to be six — The grade-6 promotion rate observed in 2023 belongs to children who entered in...
- The grade sizes are not a cohort — Grade 1 has 149 students and grade 6 has 110, and some of that is genuine attrition...

What to compute instead

- Retention, not promotion — if you must use a single year

What to compute instead — In Python

```python
def chained(statuses):
    rate, product = {}, 1.0
    for grade in range(1, 7):
        rows = cohort[cohort["grade"] == grade]["end_of_year_status"]
        product *= rows.isin(statuses).mean()
        rate[grade] = product
    return product

promotion_only = chained(["promoted", "completed-final-grade"])
retained = chained(["promoted", "completed-final-grade", "repeated"])
print(f"chained promotion: {promotion_only:.1%}")
print(f"chained retention: {retained:.1%}")
```

What to compute instead — In R

```
# Same chain, different numerator. The gap is repetition compounding.
```

What to compute instead

- 19.3% against 40.3% — More than twenty points of the apparent loss was repetition being read as exit, and the retention...
- Follow real students where you can — This register has two years and the same identifiers, so a one-year transition is...

What to compute instead — In Python

```
years = clean.pivot_table(index="student_id", columns="school_year",
                          values="grade", aggfunc="first")
both = years.dropna()
print(f"students observed in both years: {len(both)}")
print(f"  advanced a grade: {(both[2024] > both[2023]).mean():.1%}")
print(f"  same grade:       {(both[2024] == both[2023]).mean():.1%}")
```

What to compute instead

- 1,103 students appear in both years: 88.5% advanced a grade and 11.5% are in the same grade twice — That is a measured...

What to compute instead — In R

```r
enrolment |>
  select(student_id, school_year, grade) |>
  tidyr::pivot_wider(names_from = school_year, values_from = grade) |>
  filter(!is.na(`2023`), !is.na(`2024`))
```

What to compute instead

- A one-year transition observed on real students beats a six-year chain assembled from one year's rates — even though it…

Who leaves — In Python

```
year23 = cohort.assign(
    over_age=cohort["age_years"] > cohort["grade"] + 5,
    dropped=cohort["end_of_year_status"].eq("dropped-out"),
)
for column in ("over_age", "disability_reported", "displacement_status", "sex"):
    table = year23.groupby(column, dropna=False)["dropped"].agg(["mean", "size"])
    print((table * [100, 1]).round(1), "\n")
```

Who leaves — In R

```r
cohort |>
  mutate(dropped = end_of_year_status == "dropped-out") |>
  summarise(dropout = mean(dropped), n = n(), .by = displacement_status)
```

Who leaves

Cut	Dropout	n
Over-age	**19.5%**	478
In-age	4.8%	814
Returnee	19.7%	71
Disability reported	17.9%	112
Internally displaced	14.7%	163
Host community	11.6%	95
Resident	8.6%	940
Boys	10.4%	634

...

Who leaves

- Over-age is the strongest predictor and the largest group — Disability and displacement roughly double the rate on...
- Sex shows nothing — 10.4% against 10.0% on 634 and 658 students

The missing data is not missing at random — In Python

```
print(year23["displacement_status"].isna().sum(), "students with no status")
print(year23.loc[year23["displacement_status"].isna(), "dropped"].mean())
```

The missing data is not missing at random — In R

```
cohort |> filter(is.na(displacement_status)) |> nrow()
```

The missing data is not missing at random

- Be precise about what that does and does not bias — The blanks are removed roughly at random *within* those two...
- the **dropout rate** for internally displaced and returnee students is unbiased — 14.7% and 19.7% are the right numbers...
- their **counts and population share** are understated by about a tenth. 234 students are recorded as displaced or...

Report it as a risk profile — Example (cont.)

Dropout, 2023 cohort, 1,122 students

Overall	10.2%	
Over-age for grade	19.5%	478 students
In-age	4.8%	814
Returnee	19.7%	71 students
Disability reported	17.9%	112
Internally displaced	14.7%	163
Resident	8.6%	940

No meaningful difference by sex: 10.4% boys, 10.0% girls.

42 students have no displacement status, all drawn from displaced and returnee households. The rates above are unbiased for those groups; their

Report it as a risk profile — Example (cont.)

counts are understated by about a tenth.

Not reported: survival to the final grade. A reconstructed cohort from a single year gives 19.3% on promotion and 40.3% on retention, neither of which is a completion rate. A true cohort needs six years of the register or a household survey. The measured one-year transition is 88.5%.

Report it as a risk profile

- "Not reported, and why" is the most useful line in that block — Somebody will ask for a completion rate, and the answer...

What comes next

- Enrolment, attendance and retention describe whether children are in school.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/education-programme-analysis/edu-06-who-drops-out