88% attendance, 62% of students

Average daily attendance is 88.4%. The share of students attending at least 90% of their marked days is 62.1%. Both come from the same 70,245 rows, and the twenty-six points between them are the children a mean cannot see.

Pierre Robentz Cassion

Lesson 3 of 8

Education Programme Analysis — Enrolled is not attending

What this lesson covers

- One register, two questions
- Average daily attendance
- The proportion regularly attending
- Which one a programme is judged on
- State the threshold, and where it came from
- The denominator decision nobody makes explicitly
- Report the pair
- What comes next

One register, two questions — In Python

```python
import pandas as pd

attendance = pd.read_csv("school-attendance-2024.v1.csv")
print(f"{len(attendance):,} rows, {attendance['student_id'].nunique():,} students")
print(attendance["present"].value_counts(dropna=False))
```

One register, two questions — In R

```
library(dplyr)
attendance |> count(present)
```

One register, two questions

- The `present` column has five values, not two — SCH09 recorded `Y` and `N` instead of `true` and `false` on 873 of its…

One register, two questions — In Python

```
MARKS = {"true": True, "Y": True, "false": False, "N": False}
marked = attendance[attendance["present"].isin(MARKS)].copy()
marked["attended"] = marked["present"].map(MARKS)
print(f"usable marks: {len(marked):,} of {len(attendance):,}")
```

One register, two questions — In R

```
attendance |>
  mutate(attended = case_when(present %in% c("true", "Y") ~ TRUE,
                              present %in% c("false", "N") ~ FALSE)) |>
  filter(!is.na(attended))
```

Average daily attendance — In Python

```
ada = marked["attended"].mean()
print(f"average daily attendance: {ada:.1%}")
```

Average daily attendance — In R

```
marked |> summarise(ada = mean(attended))
```

Average daily attendance

- 88.4% — Every mark counts once, so a student present 60 days out of 60 and one present 30 out of 60 contribute in...

The proportion regularly attending — In Python

```
by_student = marked.groupby("student_id")["attended"].agg(["mean", "size"])
eligible = by_student[by_student["size"] >= 20]

for threshold in (0.80, 0.85, 0.90):
    share = (eligible["mean"] >= threshold).mean()
    print(f"attending at least {threshold:.0%} of marked days: {share:.1%}")
```

The proportion regularly attending — In R

```
marked |>
  summarise(rate = mean(attended), days = n(), .by = student_id) |>
  filter(days >= 20) |>
  summarise(across(everything(), ~ mean(rate >= 0.9)))
```

The proportion regularly attending

Threshold	Students meeting it
At least 80% of days	87.3%
At least 85%	78.9%
At least 90%	**62.1%**

The proportion regularly attending

- 62.1% at the 90% threshold, against an average daily attendance of 88.4% — The two numbers are twenty-six points apart…
- A mean over marks describes the system; a proportion over students describes children — A school where every child…

Which one a programme is judged on

- Report both, and lead with the one that matches the decision

Which one a programme is judged on

The decision	The number
How many meals, desks, textbooks	Average daily attendance
Which children need a follow-up visit	Proportion below the threshold
Whether an intervention worked	Both, because they can move in opposite directions

Which one a programme is judged on — In Python

```
struggling = eligible[eligible["mean"] < 0.75]
print(f"students below 75%: {len(struggling)} "
      f"({len(struggling) / len(eligible):.1%})")
print(f"their marks as a share of all marks: "
      f"{struggling['size'].sum() / eligible['size'].sum():.1%}")
```

Which one a programme is judged on — In R

```
# The students furthest behind are a small share of the rows and the whole
# of the problem.
```

State the threshold, and where it came from — In Python

```
sensitivity = {f"{t:.0%}": f"{(eligible['mean'] >= t).mean():.1%}"
               for t in (0.75, 0.80, 0.85, 0.90, 0.95)}
print(sensitivity)
```

State the threshold, and where it came from — In R

```
# Print the curve, not the point.
```

The denominator decision nobody makes explicitly

- Without it — a student who enrolled in the last week of term and attended three days out of three appears as a 100%...
- With it — you have excluded exactly the late-arriving and early-leaving students, who are the ones an attendance...

The denominator decision nobody makes explicitly — In Python

```
print(f"students with any marks:      {len(by_student)}")
print(f"students with 20 or more:     {len(eligible)}")
```

The denominator decision nobody makes explicitly — In R

```
# Two denominators, both defensible, and the report says which.
```

Report the pair — Example

Attendance, February to April 2024

Average daily attendance	88.4%	69,974 usable marks
Students attending 90%+ of days	62.1%	1,200 students with 20+ marks
Students attending 85%+	78.9%	
Students below 75%	9.5%	114 students, the follow-up list

SCH09 recorded 873 marks as Y/N rather than true/false; recoded, not dropped. 271 days were never marked and are excluded from both figures.

What comes next

- Both numbers in this lesson divide by days that were marked.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/education-programme-analysis/edu-03-two-attendance-numbers