The round nobody drove

2,629 visits were made of 2,904 due. Nord-Ouest reads best of the three districts at 76.4% and its answer is uncertain by thirteen points, against two and a half for the district that visited nearly everything.

Pierre Robentz Cassion

Lesson 7 of 8

What this lesson covers

- The denominator that was never questioned
- Coverage is not spread evenly, which is the whole problem
- Bound the answer
- Is the missingness random?
- What not to do
- Report coverage beside every rate
- What comes next

The denominator that was never questioned — In Python

```
import pandas as pd

points = pd.read_csv("water-point-monitoring-2024.v1.csv", parse_dates=["visit_date"])

due = points["water_point_id"].nunique() * 12
made = len(points)
print(f"due {due:,}, made {made:,}, coverage {made / due:.1%}")
```

The denominator that was never questioned — In R

```r
library(dplyr)

points |> summarise(
  due = n_distinct(water_point_id) * 12,
  made = n(),
  coverage = n() / (n_distinct(water_point_id) * 12)
)
```

The denominator that was never questioned

- 2,629 of 2,904 — 90.5% coverage — Two hundred and seventy-five rounds were never driven, and the register does not. . .

Coverage is not spread evenly, which is the whole problem — In Python

```
coverage = points.groupby("district").agg(
    points=("water_point_id", "nunique"),
    visits=("water_point_id", "size"),
)
coverage["due"] = coverage["points"] * 12
coverage["coverage"] = coverage["visits"] / coverage["due"]
print(coverage.round(3))
```

Coverage is not spread evenly, which is the whole problem — In R

```
points |>
  summarise(points = n_distinct(water_point_id), visits = n(), .by = district) |>
  mutate(due = points * 12, coverage = visits / due)
```

Coverage is not spread evenly, which is the whole problem

District	Points	Visits made	Due	Coverage
Sud-Est	79	913	948	**96.3%**
Centre	79	882	948	93.0%
Nord-Ouest	84	834	1,008	**82.7%**

Coverage is not spread evenly, which is the whole problem — In Python

```
monthly = points.pivot_table(
    index=points["visit_date"].dt.month, columns="district",
    values="water_point_id", aggfunc="size",
)
print(monthly)
```

Coverage is not spread evenly, which is the whole problem — In R

```
points |> count(district, month = lubridate::month(visit_date)) |>
  tidyr::pivot_wider(names_from = district, values_from = n)
```

Bound the answer

- **Upper bound**: every missed visit would have found a working point. That is the observed rate, which already assumes. . .
- **Lower bound**: every missed visit would have found a broken point.

Bound the answer — In Python

```
WORKING = {"functional", "partially-functional"}
ok = points["functional_status"].isin(WORKING)

bounds = points.assign(ok=ok).groupby("district").agg(
    ok=("ok", "sum"), made=("ok", "size"), points=("water_point_id", "nunique"),
)
bounds["due"] = bounds["points"] * 12
bounds["observed"] = bounds["ok"] / bounds["made"]
bounds["lower"] = bounds["ok"] / bounds["due"]
bounds["band"] = bounds["observed"] - bounds["lower"]
print((bounds[["observed", "lower", "band"]] * 100).round(1))
```

Bound the answer — In R

```
points |>
  mutate(ok = functional_status %in% c("functional", "partially-functional")) |>
  summarise(ok = sum(ok), made = n(), pts = n_distinct(water_point_id),
            .by = district) |>
  mutate(due = pts * 12, observed = ok / made, lower = ok / due)
```

Bound the answer

District	Observed	Lower bound	Band
Nord-Ouest	**76.4%**	63.2%	**13.2**
Centre	75.7%	70.5%	5.3
Sud-Est	71.4%	68.8%	2.6

Bound the answer

- Nord-Ouest is the best district on the observed figure and the only one whose ranking is not safe — Its band overlaps...
- A number computed on 96% of its denominator and a number computed on 83% are not comparable, and nothing in the table says so unless you put it there

Is the missingness random? — In Python

```
observed = points.pivot_table(
    index="water_point_id", columns="round", values="functional_status",
    aggfunc="first",
)
followed_by_gap = []
for point, row in observed.iterrows():
    for r in range(1, 12):
        if pd.isna(row.get(r)):
            continue
        broken = row[r] not in WORKING
        followed_by_gap.append((pd.isna(row.get(r + 1)), broken))

check = pd.DataFrame(followed_by_gap, columns=["gap_next", "broken"])
print(check.groupby("gap_next")["broken"].agg(["mean", "size"]).round(3))
```

Is the missingness random? — In R

```
# Same shape: was the point broken at the visit immediately before a missed round?
```

Is the missingness random?

- 26.7% of visits that were followed by a missed round found a broken point, against 24.7% of visits that were followed...

What not to do

- Carry the last observation forward — Fill each missing round with the point's previous status
- Impute from the district mean — Assumes the missed points are like the visited ones in the same district, which is the...
- Drop the district — Removes the worst-covered district from the report, which is nearly always also the hardest to...

What not to do — In Python

```
print("Reported: 76.4% (observed), 63.2% to 76.4% (bounds), coverage 82.7%")
print("Not reported: a single imputed number that hides which of these it is")
```

What not to do — In R

```
# The band and the coverage are the finding. Neither is a caveat.
```

Report coverage beside every rate — Example

Water point functionality by district, 2024

District	Functionality	Coverage	Range if unvisited were broken
Sud-Est	71.4%	96.3%	68.8 - 71.4
Centre	75.7%	93.0%	70.5 - 75.7
Nord-Ouest	76.4%	82.7%	63.2 - 76.4

275 of 2,904 scheduled visits were not made, concentrated in Nord-Ouest during June to September when roads are impassable. Districts are not ranked here: Nord-Ouest's range overlaps both others.

What comes next

- You now have every WASH number this course can produce and a set of caveats attached to each.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/wash-analysis/

wash-07-the-round-nobody-drove