

Nineteen cells of one

Cross-tabulate area, category, age band and sex and this anonymous dataset produces 200 cells, 85 of them holding four cases or fewer and 19 holding exactly one. In a GBV dataset a cell of one is a person, and the table cannot leave the building.

Pierre Robentz Cassion

Lesson 2 of 8

Protection and GBV Data — What not to collect

What this lesson covers

- The dataset is anonymous and the table is not
- Why four and not one
- A rule, written as code
- What to do instead of publishing the cell
- The equity finding survives the rule
- The request that has to be refused
- What comes next

The dataset is anonymous and the table is not — In Python

```
import pandas as pd

referrals = pd.read_csv("protection-referrals-2024.v1.csv")

cells = referrals.groupby(
    ["admin2", "case_category", "age_band", "sex"], dropna=False
).size()

print(f"{len(cells)} cells")
print(f"  holding 1 case: {(cells == 1).sum()}")
print(f"  holding 4 or fewer: {(cells <= 4).sum()}")
```

The dataset is anonymous and the table is not — In R

```
library(dplyr)
```

```
referrals |>
```

```
  count(admin2, case_category, age_band, sex) |>
```

```
  summarise(cells = n(), of_one = sum(n == 1), of_four_or_fewer = sum(n <= 4))
```

The dataset is anonymous and the table is not

- 200 cells, 85 of them with four cases or fewer, 19 with exactly one

Why four and not one

- A cell of two identifies both people to each other — and to anyone who knows one of them
- Differencing recovers suppressed cells — If a row total is published and every cell in it but one is published, the. . .

Why four and not one — In Python

```
by_area = referrals.groupby(["admin2", "case_category"]).size().unstack()
print(by_area)
print("\nIf one cell is suppressed and the row total is printed,")
print("the suppressed cell is the total minus the others.")
```

Why four and not one — In R

```
referrals |> count(admin2, case_category) |>  
  tidyr::pivot_wider(names_from = case_category, values_from = n)
```

Why four and not one

- So a suppression rule needs a threshold and a secondary suppression — suppress small cells, then suppress enough. . .

A rule, written as code — In Python (cont.)

```
THRESHOLD = 5
```

```
def suppress(counts, threshold=THRESHOLD):  
    """Primary suppression, then a secondary pass so rows cannot be differenced."""  
    table = counts.copy()  
    small = table < threshold  
    table = table.mask(small, other=pd.NA)  
  
    # Secondary: if a row has exactly one suppressed cell, the row total gives  
    # it away. Suppress the next smallest cell in that row as well.  
    for index, row in table.iterrows():  
        if row.isna().sum() == 1:  
            remaining = counts.loc[index][~small.loc[index]]  
            if len(remaining):  
                table.loc[index, remaining.idxmin()] = pd.NA  
    return table
```

A rule, written as code — In Python (cont.)

```
print(suppress(by_area))
```

A rule, written as code — In R

```
suppress <- function(x, threshold = 5) ifelse(x < threshold, NA, x)  
# Plus the secondary pass: a row with exactly one NA is not suppressed at all.
```

A rule, written as code

- Write the rule as a function and apply it to every table — A rule applied by eye is applied inconsistently, and the one. . .

What to do instead of publishing the cell

- Aggregate up — Drop one dimension

What to do instead of publishing the cell — In Python

```
coarse = referrals.groupby(["admin2", "case_category"]).size()
print(f"cells under 5: {(coarse < 5).sum()} of {len(coarse)}")
```

What to do instead of publishing the cell — In R

```
referrals |> count(admin2, case_category) |> summarise(small = sum(n < 5))
```

What to do instead of publishing the cell

- Report the rate, not the count, on a denominator large enough to carry it —
"Completion is 31% for cases reporting a . . .
- Answer the question rather than supplying the table — A requester asking for area by age by sex usually wants to know. . .

The equity finding survives the rule — In Python

```
def completion(frame):
    consenting = frame[frame["consent_to_refer"]]
    reached = consenting[consenting["referral_accepted"] &
                        consenting["days_to_first_service"].notna()]
    return len(reached), len(consenting), len(reached) / len(consenting)

disability = referrals["disability_reported"].isin([True, "true", "Yes"])
print("reported:      %d/%d = %.1f%%" % completion(referrals[disability]))
print("not reported: %d/%d = %.1f%%" % completion(referrals[~disability]))
```

The equity finding survives the rule — In R

```
referrals |>
  filter(consent_to_refer) |>
  summarise(reached = sum(referral_accepted & !is.na(days_to_first_service)),
            n = n(), .by = disability_reported) |>
  mutate(completion = reached / n)
```

The equity finding survives the rule

- 26.7% against 46.2%, on denominators of 202 and 1,436 — The nineteen-point equity gap — the most important finding in...
- The analyses that fail the disclosure test are almost never the ones that matter — A four-way disaggregation of 1,850...

The request that has to be refused — Example (cont.)

Request: case counts by commune, month and incident category.

Response: declined.

Commune-by-month cells hold zero to three cases. At that resolution a count identifies individuals to anyone working in the area, and the dataset cannot be re-identified back but a person can be.

What I can provide instead:

- the same counts at district by quarter, no cell below five
- the completion rate and time to service for each commune, on annual denominators
- a written answer to the question the request is for, if you tell me what decision it feeds

If the requirement is genuinely commune-level operational planning, the right route is a data sharing agreement with the case management agency,

The request that has to be refused — Example (cont.)

not an extract.

The request that has to be refused

- Offer three alternatives and ask what decision the request feeds — A refusal with no alternative gets escalated and...

What comes next

- The rules are established.

Read the full lesson, with runnable code

```
https://data-analysis.cassion.dev/courses/protection-and-gbv-data/  
prot-02-disclosure-risk
```