The design effect, and the one that came out below one

Clustering costs precision, stratification buys it back, and the design effect is the net. 1.60 for food insecurity and 0.91 for water access, on the same 996 interviews.

Pierre Robentz Cassion

Lesson 4 of 8

Survey Analysis, Sampling and Weighting — What the design buys

What this lesson covers

- What it is
- Why clustering costs you
- Why stratification buys some back
- The net effect, measured
- The one below one
- Where design effects come from when you have none
- Report it
- What comes next

What it is — Example

```
deff = variance under this design / variance under simple random sampling
```

What it is — Example

```
effective n = n / deff
```

Why clustering costs you — Example

```
deff_clustering = 1 + (m - 1) * ICC
```

Why clustering costs you — In Python

```
import pandas as pd

by_area = (
    survey.assign(insecure=survey["food_insecure"] == "true")
    .groupby("ea_id")["insecure"]
    .agg(["mean", "size"])
)

p = (survey["food_insecure"] == "true").mean()
m = by_area["size"].mean()
between = by_area["mean"].var(ddof=0)
icc = (between - p * (1 - p) / m) / (p * (1 - p))

print(f"mean cluster size {m:.1f}, ICC {icc:.3f}, "
      f"clustering deff {1 + (m - 1) * icc:.2f}")
```

Why clustering costs you — In R

```r
by_area <- survey |>
  summarise(rate = mean(food_insecure == "true"), n = n(), .by = ea_id)

p <- mean(survey$food_insecure == "true")
m <- mean(by_area$n)
icc <- (var(by_area$rate) * (nrow(by_area) - 1) / nrow(by_area) -
        p * (1 - p) / m) / (p * (1 - p))

c(m = m, icc = icc, deff = 1 + (m - 1) * icc)
```

Why clustering costs you

- **It is a property of the outcome, not of the survey.** Water source is highly clustered — a village has a borehole or...
- **Published values are the best guess you have.** DHS and SMART reports publish design effects, and using last round's...

Why stratification buys some back

- Stratification does the opposite.

The net effect, measured — In Python (cont.)

```
import numpy as np


def design_effect(df, outcome, weight="weight"):
    p = np.average(df[outcome] == "true", weights=df[weight])
    total_w = df[weight].sum()

    variance = 0.0
    for _, stratum in df.groupby("stratum"):
        areas = stratum.groupby("ea_id").apply(
            lambda g: (g[weight] * ((g[outcome] == "true") - p)).sum()
        )
        n = len(areas)
        if n < 2:
            continue
        variance += n / (n - 1) * ((areas - areas.mean()) ** 2).sum()
```

The net effect, measured — In Python (cont.)

```
    variance /= total_w**2

    srs = p * (1 - p) / len(df)
    return p, np.sqrt(variance), variance / srs


for outcome in ["food_insecure", "improved_water_source"]:
    p, se, deff = design_effect(survey, outcome)
    print(f"{outcome:24} p={p:.1%} se={se:.4f} deff={deff:.2f} "
          f"n_eff={len(survey) / deff:.0f}")
```

The net effect, measured — In R

```r
library(survey)

design <- svydesign(ids = ~ea_id, strata = ~stratum, weights = ~weight,
                    data = survey, nest = TRUE)

svymean(~I(food_insecure == "true"), design, deff = TRUE)
svymean(~I(improved_water_source == "true"), design, deff = TRUE)
```

The net effect, measured

Outcome	Estimate	Design effect	Effective n
Food insecure	29.1%	**1.60**	623
Improved water source	69.8%	**0.91**	1,098

The one below one

- So "design effect" is not a synonym for "penalty" — It is the net of two opposing forces, and which one wins depends on...

Where design effects come from when you have none

- **The previous round of the same survey**, same indicator, same area. Almost always available and almost always ignored.
- **A published survey report** for the same indicator in a comparable setting. DHS reports tabulate design effects;. . .
- **A convention.** SMART surveys commonly assume 1.5 for anthropometry; humanitarian assessments often use 2.0 for. . .
- **1.0** is never a defensible assumption for a cluster survey, and a protocol that uses it has under-sized the survey. . .

Report it — In Python

```
summary = pd.DataFrame({
    "outcome": ["Food insecure", "Improved water source"],
    "estimate": ["29.1%", "69.8%"],
    "ci_95": ["25.5-32.7%", "67.0-72.5%"],
    "deff": [1.60, 0.91],
    "n": [996, 996],
    "n_effective": [623, 1098],
})
```

Report it — In R

```
tibble::tribble(
  ~outcome,                ~estimate, ~deff, ~n_effective,
  "Food insecure",           "29.1%",  1.60,          623,
  "Improved water source",   "69.8%",  0.91,         1098
)
```

What comes next

- You have a design effect, which means you have a standard error, which means you can stop computing these things by hand.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/survey-analysis-sampling/survey-04-design-effect