

Où cesser de découper

Trois strates soutiennent une estimation. Strate par statut de déplacement donne neuf cellules dont la plus petite compte dix-huit ménages. Une règle fixée avant de regarder, et le tableau qui montre au lecteur où l'enquête s'est épuisée.

Pierre Robentz Cassion

Leçon 8 sur 8

Analyse d'enquêtes, échantillonnage et pondération — Ce que vous avez le droit de publier

Ce que couvre cette leçon

- La demande qui arrive toujours
- Ce que fait le découpage
- Comptez les grappes, pas seulement les ménages
- Fixez la règle avant de regarder
- Supprimez, mais montrez la cellule
- Estimation par domaine, et ce qui ressemble à un filtre
- Planifiez la désagrégation avant l'enquête
- Le tableau à publier
- Ce que ce cours vous laisse

- Le rapport d'enquête sort avec trois estimations par strate.

Ce que fait le découpage — En Python

```
for keys in [["stratum"],
             ["stratum", "displacement_status"],
             ["stratum", "main_livelihood"]]:
    cells = survey.groupby(keys).size()
    print(f"{' x '.join(keys):40} {len(cells):>3} cells, "
          f"smallest {cells.min():>3}, {(cells < 50).sum()} below 50")
```

Ce que fait le découpage — En R

```
survey |> count(stratum) |> summarise(cells = n(), smallest = min(n))  
survey |> count(stratum, displacement_status) |> summarise(cells = n(), smallest = min(n))  
survey |> count(stratum, main_livelihood) |> summarise(cells = n(), smallest = min(n))
```

Ce que fait le découpage

Désagrégation	Cellules	Plus petite	Sous 50
Strate	3	316	0
Strate $\times$ déplacement	9	18	5
Strate $\times$ moyen d'existence	18	6	9

Ce que fait le découpage

- Une seconde la fait tomber à 6.

Comptez les grappes, pas seulement les ménages — En Python

```
cells = survey.groupby(["stratum", "displacement_status"]).agg(  
    households=("household_id", "size"),  
    clusters=("ea_id", "nunique"),  
)  
print(cells.sort_values("clusters").head())
```

Comptez les grappes, pas seulement les ménages — En R

```
survey |>  
  summarise(households = n(), clusters = n_distinct(ea_id),  
            .by = c(stratum, displacement_status)) |>  
  arrange(clusters)
```

Comptez les grappes, pas seulement les ménages

- Une cellule issue de moins d'une dizaine de grappes ne peut pas soutenir une estimation de variance que vous voudriez. . .

Fixez la règle avant de regarder — En Python

```
MIN_HOUSEHOLDS = 50
MIN_CLUSTERS = 10
MAX_CI_WIDTH = 0.20      # 20 percentage points

def reportable(cell):
    return (cell["households"] >= MIN_HOUSEHOLDS
            and cell["clusters"] >= MIN_CLUSTERS
            and cell["ci_width"] <= MAX_CI_WIDTH)
```

Fixez la règle avant de regarder — En R

```
MIN_HOUSEHOLDS <- 50; MIN_CLUSTERS <- 10; MAX_CI_WIDTH <- 0.20

reportable <- function(d) {
  d$households >= MIN_HOUSEHOLDS & d$clusters >= MIN_CLUSTERS &
    d$ci_width <= MAX_CI_WIDTH
}
```

Supprimez, mais montrez la cellule — En Python

```
by_cell = svyby_equivalent(survey, ["stratum", "displacement_status"], "food_insecure")
by_cell["reported"] = by_cell.apply(reportable, axis=1)
by_cell.loc[~by_cell["reported"], ["estimate", "ci_low", "ci_high"]] = None
by_cell["note"] = by_cell["reported"].map(
    {False: "too few clusters to estimate", True: ""}
)
print(by_cell)
```

Supprimez, mais montrez la cellule — En R

```
by_cell <- svyby(~I(food_insecure == "true"), ~stratum + displacement_status,  
                design, svymean, vartype = "ci") |>  
  mutate(reported = reportable(cur_data()),  
         note = if_else(reported, "", "too few clusters to estimate"))
```

Supprimez, mais montrez la cellule

- Ne supprimez jamais la ligne — Une cellule masquée qui affiche encore son effectif de ménages et un motif dit au...

Estimation par domaine, et ce qui ressemble à un filtre — En R

WRONG: rebuilding the design from a filtered frame

```
displaced <- svydesign(ids = ~ea_id, strata = ~stratum, weights = ~weight,  
                      data = filter(survey, displacement_status == "displaced"),  
                      nest = TRUE)
```

RIGHT: a domain estimate keeps the full design in view

```
svyby(~I(food_insecure == "true"), ~displacement_status, design, svymean)
```

Estimation par domaine, et ce qui ressemble à un filtre — En Python

```
# The same rule: the variance calculation must still see every cluster,  
# including those with no displaced households in them.
```

Planifiez la désagrégation avant l'enquête

- **Suréchantillonner le sous-groupe**, ce à quoi sert la stratification. Cette enquête a suréchantillonné le rural. . .
- **Accepter un intervalle plus large** pour ce sous-groupe, en le disant à l'avance.
- **Abandonner l'exigence**, en écrivant dans le protocole que l'enquête n'en rendra pas compte.
- Ce qu'on ne peut pas faire, c'est décider après — Le jour où les données existent, les cellules sont ce qu'elles sont, . . .

```
final = by_cell[["stratum", "displacement_status", "households", "clusters",  
                "estimate", "ci_low", "ci_high", "note"]]  
final.to_csv("outputs/tables/food_insecurity_by_stratum_displacement.csv", index=False)
```

```
readr::write_csv(by_cell,  
  here::here("outputs", "tables", "food_insecurity_by_stratum_displacement.csv"))
```

Le tableau à publier

Strate	Statut	Ménages	Grappes	Estimation	IC 95 %	Note
Urbain	Résident	246	25	13,9 %	10,2-18,7	
Urbain	Déplacé	49	18	—	—	trop peu de ménages
Rural reculé	Résident	279	25	46,1 %	40,0-52,3	
Rural reculé	Retourné	24	12	—	—	trop peu de ménages

- Vous savez reconstituer des pondérations à partir d'une base et prouver qu'elles totalisent la population, dimensionner une enquête face à une exigence de précision et lire cette exigence à l'envers sur une enquête dont vous héritez, mesurer un effet de plan au lieu de le supposer, estimer avec le plan propagé, traiter explicitement la non-réponse et les remplacements, publier un intervalle de la bonne forme, et dire où l'échantillon cesse de soutenir un découpage.

Lire la leçon complète, avec le code exécutable

`https://data-analysis.cassion.dev/fr/cours/survey-analysis-sampling/
survey-08-how-far-to-disaggregate`