

Construire les pondérations à partir de la base

Probabilité de sélection à chaque degré, la pondération de base qu'elle implique, l'ajustement pour non-réponse, et le contrôle qui prouve l'ensemble — des pondérations totalisant 56 428, exactement ce que la base déclare.

Pierre Robentz Cassion

Leçon 2 sur 8

Analyse d'enquêtes, échantillonnage et pondération — L'échantillon n'est pas la population

Ce que couvre cette leçon

- Une pondération est un nombre avec un seul sens
- Premier degré — probabilité proportionnelle à la taille
- Second degré — un tirage fixe
- Les deux probabilités se compensent
- L'ajustement pour non-réponse
- Le contrôle qui prouve toute la construction
- Pondérations normalisées, et quand s'en servir
- Pondérations par personne et de sous-échantillon
- Enregistrez les pondérations à côté des données
- La suite

Une pondération est un nombre avec un seul sens

- Une pondération de sondage, c'est le nombre d'unités de la population que représente l'unité tirée

Premier degré — probabilité proportionnelle à la taille — Exemple

$$P(\text{area } i \text{ selected}) = n_h * M_i / M_h$$

Premier degré — probabilité proportionnelle à la taille — En Python

```
import pandas as pd

frame = pd.read_csv("household-survey-frame-2025.v1.csv")

stratum_households = frame.groupby("stratum")["households"].sum()
selected = frame[frame["selected"] == True].copy()

selected["p_stage1"] = (
    25 * selected["households"] / selected["stratum"].map(stratum_households)
)
print(selected[["ea_id", "stratum", "households", "p_stage1"]].head())
```

Premier degré — probabilité proportionnelle à la taille — En R

```
library(dplyr)

stratum_households <- frame |> summarise(M_h = sum(households), .by = stratum)

selected <- frame |>
  filter(selected) |>
  left_join(stratum_households, by = "stratum") |>
  mutate(p_stage1 = 25 * households / M_h)
```

Second degré — un tirage fixe — Exemple

$$P(\text{household selected} \mid \text{area selected}) = m / M_i$$

Second degré — un tirage fixe — En Python

```
selected["p_stage2"] = 14 / selected["households"]  
selected["p_overall"] = selected["p_stage1"] * selected["p_stage2"]  
print(selected.groupby("stratum")["p_overall"].describe()[["min", "max"]])
```

Second degré — un tirage fixe — En R

```
selected <- selected |>
  mutate(p_stage2 = 14 / households,
         p_overall = p_stage1 * p_stage2)

selected |> summarise(min = min(p_overall), max = max(p_overall), .by = stratum)
```

Les deux probabilités se compensent — Exemple

$$P(\text{household}) = (n_h * M_i / M_h) * (m / M_i) = n_h * m / M_h$$

Les deux probabilités se compensent

- Deux conséquences à retenir — Une pondération qui varie à l'intérieur d'une strate dans un plan proportionnel à la...

Les deux probabilités se compensent — En Python

```
base_weight = stratum_households / (25 * 14)
print(base_weight.round(1))
```

Les deux probabilités se compensent — En R

```
stratum_households |> mutate(base_weight = M_h / (25 * 14))
```

Les deux probabilités se compensent

Strate	Ménages dans la base	Pondération de base
Urbain	21 270	60,8
Rural accessible	28 958	82,7
Rural reculé	6 200	17,7

L'ajustement pour non-réponse — En Python

```
survey = pd.read_csv("household-survey-2025.v1.csv")

interviewed = selected.set_index("ea_id")["households_interviewed"]

survey["base_weight"] = survey["stratum"].map(base_weight)
survey["nr_adjustment"] = 14 / survey["ea_id"].map(interviewed)
survey["weight"] = survey["base_weight"] * survey["nr_adjustment"]

print(survey["weight"].describe()[["min", "max"]].round(1))
```

L'ajustement pour non-réponse — En R

```
survey <- survey |>
  left_join(select(selected, ea_id, households_interviewed), by = "ea_id") |>
  left_join(mutate(stratum_households, base_weight = M_h / (25 * 14)), by = "stratum") |>
  mutate(weight = base_weight * 14 / households_interviewed)

range(survey$weight)
```

- Ajustez à l'intérieur de la zone, pas de la strate — Les ménages qui n'ont pas répondu dans une zone donnée ressemblent. . .

Le contrôle qui prouve toute la construction — En Python

```
total = survey["weight"].sum()
frame_total = frame["households"].sum()
print(f"weights sum to {total:,.0f}; frame holds {frame_total:,.0f}")
assert abs(total - frame_total) < 1
```

Le contrôle qui prouve toute la construction — En R

```
c(weights = sum(survey$weight), frame = sum(frame$households))
```

Le contrôle qui prouve toute la construction

- 56 428 et 56 428 — Les pondérations totalisent exactement le nombre de ménages que la base déclare, ce que « combien... »

Pondérations normalisées, et quand s'en servir — En Python

```
survey["weight_norm"] = survey["weight"] / survey["weight"].mean()
```

Pondérations normalisées, et quand s'en servir — En R

```
survey <- survey |> mutate(weight_norm = weight / mean(weight))
```

- Une pondération par personne — est la pondération du ménage multipliée par sa taille, car un ménage de huit représente. . .

Pondérations par personne et de sous-échantillon — En Python

```
people = survey[survey["household_size"].notna()].copy()
people["person_weight"] = people["weight"] * people["household_size"]
print(f"estimated population {people['person_weight'].sum():,.0f}")
```

```
people <- survey |>
  filter(!is.na(household_size)) |>
  mutate(person_weight = weight * household_size)

sum(people$person_weight)
```

- Une pondération de sous-échantillon — s'applique quand une unité est choisie parmi plusieurs

```
children = survey[survey["child_muac_mm"].notna()].copy()
children["child_weight"] = children["weight"] * children["children_under5"]
```

```
children <- survey |>  
  filter(!is.na(child_muac_mm)) |>  
  mutate(child_weight = weight * children_under5)
```

Enregistrez les pondérations à côté des données — En Python

```
survey[["household_id", "ea_id", "stratum", "base_weight",  
        "nr_adjustment", "weight"]].to_csv("outputs/weights.csv", index=False)
```

Enregistrez les pondérations à côté des données — En R

```
survey |>  
  select(household_id, ea_id, stratum, base_weight, households_interviewed, weight) |>  
  readr::write_csv(here::here("outputs", "weights.csv"))
```

- Vous savez désormais produire une estimation de population.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/survey-analysis-sampling/
survey-02-reconstructing-weights](https://data-analysis.cassion.dev/fr/cours/survey-analysis-sampling/survey-02-reconstructing-weights)