

Les trente-sept ménages que le remplissage par zéro déclasserait

Compléter par des zéros une échelle de la faim incomplète déplace la prévalence de trois dixièmes de point et peut mal classer chacun des trente-sept ménages concernés. Lequel des deux compte dépend de si votre production est un pourcentage ou une liste.

Pierre Robentz Cassion

Leçon 2 sur 8

Analyse de la sécurité alimentaire et IPC — Calculez-le avant d'y croire

Ce que couvre cette leçon

- Deux instruments de plus, deux règles de plus
- L'échelle de la faim dans le ménage
- Ce que coûte réellement le remplissage par zéro
- L'indice réduit des stratégies de survie
- Trois instruments, trois échantillons analysables
- La suite

Deux instruments de plus, deux règles de plus

Instrument	Ce qu'il demande	La règle qui saute
ÉFM	Trois questions sur la privation de nourriture, cotées 0–2 chacune	Valide seulement si les trois sont renseignées
ICSR	Cinq comportements, jours sur les sept derniers, pondérés	Les pondérations ne valent pas toutes 1, et la cinquième vaut 3

L'échelle de la faim dans le ménage — En Python

```
import pandas as pd

survey = pd.read_csv("food-security-survey-2024.v1.csv")
HHS = ["hhs_no_food_in_house", "hhs_sleep_hungry",
       "hhs_day_and_night_without_eating"]

complete = survey[HHS].notna().all(axis=1)
score = survey.loc[complete, HHS].sum(axis=1)

bands = pd.cut(score, [-1, 1, 3, 6],
                labels=["little or none", "moderate", "severe"])
print(f"complete: {complete.sum()} of {len(survey)}")
print((bands.value_counts(normalize=True) * 100).round(1))
```

L'échelle de la faim dans le ménage — En R

```
library(dplyr)

hhs <- c("hhs_no_food_in_house", "hhs_sleep_hungry",
        "hhs_day_and_night_without_eating")

survey |>
  filter(if_all(all_of(hhs), ~ !is.na(.x))) |>
  mutate(score = rowSums(across(all_of(hhs))),
         band = cut(score, c(-1, 1, 3, 6),
                    labels = c("little or none", "moderate", "severe"))) |>
  count(band) |> mutate(share = n / sum(n))
```

L'échelle de la faim dans le ménage

Catégorie	Ménages	Part
Faim faible ou nulle (0–1)	1 186	57,2 %
Modérée (2–3)	564	27,2 %
Sévère (4–6)	325	15,7 %

L'échelle de la faim dans le ménage

- 2 075 ménages sur 2 112 ont répondu aux trois questions — Les trente-sept qui ne l'ont pas fait sont les intéressants

Ce que coûte réellement le remplissage par zéro — En Python

```
zero_filled = survey[HHS].fillna(0).sum(axis=1)
naive = pd.cut(zero_filled, [-1, 1, 3, 6],
               labels=["little or none", "moderate", "severe"])
print((naive.value_counts(normalize=True) * 100).round(1))
```

Ce que coûte réellement le remplissage par zéro — En R

```
survey |>  
  mutate(score = rowSums(across(all_of(hhs)), na.rm = TRUE)) |>  
  count(band = cut(score, c(-1, 1, 3, 6)))
```

Ce que coûte réellement le remplissage par zéro

	Cas complets	Complété par zéro
Faible ou nulle	57,2 %	57,5 %
Modérée	27,2 %	26,9 %
Sévère	15,7 %	15,5 %

Ce que coûte réellement le remplissage par zéro

- Trois dixièmes de point de pourcentage — Sur une prévalence, le remplissage par zéro est ici pratiquement inoffensif,...

Ce que coûte réellement le remplissage par zéro — En Python

```
partial = survey.loc[~complete, HHS]  
print(partial.sum(axis=1).value_counts().sort_index())
```

Ce que coûte réellement le remplissage par zéro — En R

```
survey |> filter(if_any(all_of(hhs), is.na)) |>  
  mutate(observed = rowSums(across(all_of(hhs)), na.rm = TRUE)) |> count(observed)
```

Ce que coûte réellement le remplissage par zéro

- Le coût du remplissage par zéro dépend donc entièrement de ce que produit l'analyse
- Si la production est une **prévalence**, le dommage vaut trois dixièmes de point et il faut dire que vous avez complété. . .
- Si la production est une **liste de ciblage**, vous venez de dire à trente-sept ménages qu'ils sont en sécurité. . .
- Nommez la production avant de choisir la règle — C'est la même décision que le dénominateur du taux de guérison PCIMA. . .

L'indice réduit des stratégies de survie — En Python

```
RCSI = {  
    "rcsi_less_preferred_food": 1,  
    "rcsi_borrowed_food": 2,  
    "rcsi_limit_portion_size": 1,  
    "rcsi_restrict_adult_consumption": 3,  
    "rcsi_reduce_meal_numbers": 1,  
}  
  
rcsi = sum(survey[column] * weight for column, weight in RCSI.items())  
print(f"median {rcsi.median():.0f}, mean {rcsi.mean():.1f}, max {rcsi.max():.0f}")  
print(f"rCSI 19 or above: {(rcsi >= 19).mean():.1%}")
```

L'indice réduit des stratégies de survie — En R

```
rcsi_weights <- c(rcsi_less_preferred_food = 1, rcsi_borrowed_food = 2,  
                  rcsi_limit_portion_size = 1,  
                  rcsi_restrict_adult_consumption = 3,  
                  rcsi_reduce_meal_numbers = 1)
```

L'indice réduit des stratégies de survie

- Médiane 19, moyenne 19,5, et 50,9 % à 19 ou plus — La pondération de 3 sur *restreindre la consommation des adultes. . .
- Il n'a pas de seuil universel — Contrairement au SCA, ses seuils sont propres au contexte et généralement fixés face à. . .
- Il monte avant que la consommation ne baisse — et il peut aussi *baisser* dans une crise, quand un ménage a épuisé les. . .

Trois instruments, trois échantillons analysables — En Python

```
denominators = pd.DataFrame({
    "instrument": ["Food Consumption Score", "Household Hunger Scale",
                  "reduced Coping Strategy Index"],
    "analysable": [1989, 2075, 2112],
    "excluded": [123, 37, 0],
    "rule": ["all eight groups answered", "all three questions answered",
            "complete in this file"],
})
print(denominators)
```

Trois instruments, trois échantillons analysables — En R

```
# Print this before the results, every time.
```

Trois instruments, trois échantillons analysables

- Trois nombres sur trois dénominateurs, et aucun ne vaut 2 112 — L'écart est assez petit pour être invisible dans un. . .

- La consommation et la faim disent ce qu'un ménage vit.

Lire la leçon complète, avec le code exécutable

`https://data-analysis.cassion.dev/fr/cours/food-security-ipc/
fs-02-hunger-and-coping`