

Un ménage est classé par sa pire stratégie

Dix stratégies, trois phases de sévérité, et la règle est le maximum plutôt que la somme. 18,2 % de ces ménages ont employé une stratégie d'urgence, et la façon dont un district a traité « sans objet » rend ses prévalences fausses tout en laissant sa classification intacte.

Pierre Robentz Cassion

Leçon 3 sur 8

Analyse de la sécurité alimentaire et IPC — L'adaptation est une séquence, non un score

Ce que couvre cette leçon

- Le module, et pourquoi ce n'est pas un indice de plus
- Sans objet n'est pas non
- Le district où la distinction a été perdue
- Le module partiel
- Pourquoi l'adaptation se mesure à part de la consommation
- La suite

Le module, et pourquoi ce n'est pas un indice de plus

Phase	Stratégies	Ce qu'elles ont en commun
Stress	Vente de biens du ménage, épargne épuisée, emprunt, vente d'animaux au-delà de l'habitude	Réduisent la capacité à absorber un choc futur ; réversibles
Crise	Vente d'actifs productifs, retrait des enfants de l'école, coupe des dépenses de santé	Réduisent la productivité future ; difficiles à inverser
Urgence	Vente de la maison ou de la terre, mendicité, vente des dernières femelles reproductrices	Réduisent la productivité future et sont quasi irréversibles

Le module, et pourquoi ce n'est pas un indice de plus

- La classification est un maximum, non une somme, et non un décompte — Un ménage qui a vendu une fois ses dernières. . .

Le module, et pourquoi ce n'est pas un indice de plus — En Python (suite)

```
import pandas as pd

coping = pd.read_csv("livelihood-coping-2024.v1.csv")

PHASE = {
    "sold_household_assets": "stress",
    "spent_savings": "stress",
    "borrowed_money": "stress",
    "sold_more_animals_than_usual": "stress",
    "sold_productive_assets": "crisis",
    "withdrew_children_from_school": "crisis",
    "reduced_health_expenditure": "crisis",
    "sold_house_or_land": "emergency",
    "begged": "emergency",
    "sold_last_female_animals": "emergency",
}
```

Le module, et pourquoi ce n'est pas un indice de plus — En Python (suite)

```
RANK = {"none": 0, "stress": 1, "crisis": 2, "emergency": 3}

used = coping[list(PHASE)].eq("yes")
severity = pd.Series("none", index=coping.index)
for strategy, phase in PHASE.items():
    higher = used[strategy] & (severity.map(RANK) < RANK[phase])
    severity[higher] = phase

print((severity.value_counts(normalize=True) * 100).round(1))
```

Le module, et pourquoi ce n'est pas un indice de plus — En R

```
library(dplyr)

phase <- c(sold_household_assets = "stress", spent_savings = "stress",
          borrowed_money = "stress", sold_more_animals_than_usual = "stress",
          sold_productive_assets = "crisis",
          withdrew_children_from_school = "crisis",
          reduced_health_expenditure = "crisis",
          sold_house_or_land = "emergency", begged = "emergency",
          sold_last_female_animals = "emergency")

rank <- c(none = 0, stress = 1, crisis = 2, emergency = 3)
```

Le module, et pourquoi ce n'est pas un indice de plus

Classification	Ménages	Part
Aucune	339	16,0 %
Stress	754	35,5 %
Crise	642	30,3 %
Urgence	386	18,2 %

Le module, et pourquoi ce n'est pas un indice de plus

- 48,5 % ont employé une stratégie de crise ou d'urgence — Ce seul chiffre est ce que le tableau de preuves IPC retient. . .

Sans objet n'est pas non — En Python

```
answers = coping["sold_last_female_animals"].value_counts()
print(answers)

applicable = coping["sold_last_female_animals"].isin(["yes", "no"])
print(f"prevalence among applicable: "
      f"{coping.loc[applicable, 'sold_last_female_animals'].eq('yes').mean():.1%}")
print(f"prevalence if n/a counted as no: "
      f"{coping['sold_last_female_animals'].eq('yes').mean():.1%}")
```

Sans objet n'est pas non — En R

```
coping |> count(sold_last_female_animals)
```

```
coping |>
```

```
  filter(sold_last_female_animals %in% c("yes", "no")) |>
```

```
  summarise(prevalence = mean(sold_last_female_animals == "yes"), n = n())
```

Sans objet n'est pas non

Stratégie	Parmi les ménages concernés	Si « sans objet » compté comme non
Vente d'animaux au-delà de l'habitude	37,4 %	27,5 %
Vente d'actifs productifs	19,8 %	17,3 %
Retrait des enfants de l'école	19,7 %	16,5 %
Vente des dernières femelles	8,6 %	5,8 %

- Dix points d'écart sur la première ligne — Le dénominateur d'une prévalence par stratégie est l'ensemble des ménages. . .
- La classification au niveau du ménage y survit et la prévalence par stratégie non — car un maximum sur des valeurs. . .

Le district où la distinction a été perdue — En Python

```
by_district = coping.groupby("district")["sold_last_female_animals"].agg(  
    yes=lambda s: (s == "yes").sum(),  
    applicable=lambda s: s.isin(["yes", "no"]).sum(),  
    not_applicable=lambda s: (s == "not-applicable").sum(),  
)  
by_district["correct"] = by_district["yes"] / by_district["applicable"]  
by_district["naive"] = by_district["yes"] / coping.groupby("district").size()  
print((by_district[["correct", "naive"]] * 100).round(1))
```

Le district où la distinction a été perdue — En R

```
coping |>
  summarise(yes = sum(sold_last_female_animals == "yes"),
            applicable = sum(sold_last_female_animals %in% c("yes", "no")),
            n = n(), .by = district) |>
  mutate(correct = yes / applicable, naive = yes / n)
```

Le district où la distinction a été perdue

District	Bon dénominateur	Tous les ménages
Nord-Ouest	12,3 %	6,8 %
Artibonite	9,7 %	5,3 %
Sud	7,5 %	4,5 %
Centre	6,5 %	6,4 %

Le district où la distinction a été perdue — En Python

```
print(coping.groupby("district")[list(PHASE)]  
      .apply(lambda block: (block == "not-applicable").sum().sum()))
```

Le district où la distinction a été perdue — En R

```
coping |> summarise(across(all_of(names(phase)),  
  ~ sum(.x == "not-applicable")), .by = district)
```

Le district où la distinction a été perdue

- Sur le bon dénominateur, Nord-Ouest vaut presque le double de Centre. Sur le mauvais, les deux sont identiques — Une. . .

```
emergency = [s for s, phase in PHASE.items() if phase == "emergency"]
partial = coping[emergency].eq("").all(axis=1) | coping[emergency].isna().all(axis=1)
print(f"{partial.sum()} households cannot be classified above crisis")
```

```
coping |> filter(if_all(c(sold_house_or_land, begged, sold_last_female_animals),  
                        ~ is.na(.x))) |> nrow()
```

Pourquoi l'adaptation se mesure à part de la consommation — En Python

```
survey = pd.read_csv("food-security-survey-2024.v1.csv")
merged = survey.merge(coping, on="household_id", how="left", indicator=True)
print(merged["_merge"].value_counts())
```

Pourquoi l'adaptation se mesure à part de la consommation — En R

```
survey |> left_join(coping, by = "household_id") |>  
  summarise(matched = sum(!is.na(district.y)), n = n())
```

Pourquoi l'adaptation se mesure à part de la consommation

- C'est l'argument pour mesurer les deux, et la leçon suivante est ce qui arrive quand on le fait

- Quatre instruments existent désormais sur ces ménages : consommation, faim, comportements de survie et stratégies d'adaptation.

Lire la leçon complète, avec le code exécutable

`https://data-analysis.cassion.dev/fr/cours/food-security-ipc/
fs-03-livelihood-coping-strategies`