

Testé sur un tiers, censuré sur un cinquième

E. coli a été testé sur 33,4 % des ménages, le chlore sur 40,0 %, et les deux sur 13,3 %. Dans le registre des points d'eau, 271 lectures de chlore sont la chaîne "<0.1" — et les supprimer relève la conformité apparente de seize points.

Pierre Robentz Cassion

Leçon 4 sur 8

Analyse EAH — Ce que Sphère demande et qu'une échelle ne dit pas

Ce que couvre cette leçon

- Le dénominateur change, et le rapport généralement non
- Les classes de risque du JMP
- Amélioré n'est pas sûr, et c'est ici que cela se démontre
- La valeur censurée, et pourquoi elle n'est pas manquante
- Que faire d'une valeur censurée
- L'incohérence entre champs qu'il faut nommer
- Rapportez-le comme un bloc à part
- La suite

Le dénominateur change, et le rapport généralement non — En Python

```
import pandas as pd

households = pd.read_csv("wash-household-survey-2024.v1.csv")

ecoli = households["ecoli_cfu_100ml"].notna()
chlorine = households["free_residual_chlorine_mgl"].notna()

print(f"E. coli tested:    {ecoli.sum():>5} = {ecoli.mean():.1%}")
print(f"chlorine tested:  {chlorine.sum():>5} = {chlorine.mean():.1%}")
print(f"both:             {(ecoli & chlorine).sum():>5} = {(ecoli & chlorine).mean():.1%}")
```

Le dénominateur change, et le rapport généralement non — En R

```
library(dplyr)

households |> summarise(
  ecoli = mean(!is.na(ecoli_cfu_100ml)),
  chlorine = mean(!is.na(free_residual_chlorine_mgl)),
  both = mean(!is.na(ecoli_cfu_100ml) & !is.na(free_residual_chlorine_mgl))
)
```

Le dénominateur change, et le rapport généralement non

- 802 ménages pour E. coli, 961 pour le chlore, et 320 pour les deux — Le dernier nombre est celui qui compte : toute. . .
- Imprimez le n analysable pour chaque chiffre de qualité — dans le tableau et non en note de bas de page

Les classes de risque du JMP — En Python

```
tested = households[ecoli]
classes = pd.cut(
    tested["ecoli_cfu_100ml"],
    [-1, 0, 10, 100, 10**9],
    labels=["safe <1", "low 1-10", "moderate 11-100", "high >100"],
)
print(classes.value_counts(normalize=True).round(3))
```

Les classes de risque du JMP — En R

```
households |>
  filter(!is.na(ecoli_cfu_100ml)) |>
  mutate(risk = cut(ecoli_cfu_100ml, c(-1, 0, 10, 100, Inf),
                    labels = c("safe", "low", "moderate", "high"))) |>
  count(risk) |> mutate(share = n / sum(n))
```

Les classes de risque du JMP

Classe de risque	Ménages	Part des testés
Sûre (<1 UFC/100 mL)	430	53,6 %
Faible (1–10)	173	21,6 %
Modérée (11–100)	139	17,3 %
Élevée (>100)	60	7,5 %

- Rapportez les classes, non une moyenne — Les dénombrements d'E

Amélioré n'est pas sûr, et c'est ici que cela se démontre — En Python

```
IMPROVED = {"piped-into-dwelling", "piped-into-yard", "public-tap",  
            "borehole", "protected-well", "protected-spring"}  
  
tested = tested.assign(improved=tested["water_source"].isin(IMPROVED))  
print(tested.groupby("improved")["ecoli_cfu_100ml"].agg(  
    n="size", safe=lambda s: (s < 1).mean(), high=lambda s: (s > 100).mean()  
).round(3))
```

Amélioré n'est pas sûr, et c'est ici que cela se démontre — En R

```
households |>  
  filter(!is.na(ecoli_cfu_100ml)) |>  
  mutate(improved = water_source %in% improved) |>  
  summarise(n = n(), safe = mean(ecoli_cfu_100ml < 1),  
            high = mean(ecoli_cfu_100ml > 100), .by = improved)
```

Amélioré n'est pas sûr, et c'est ici que cela se démontre

Source	Testés	Sûre	Risque élevé
Améliorée	625	61,1 %	3,8 %
Non améliorée ou de surface	177	27,1 %	20,3 %

La valeur censurée, et pourquoi elle n'est pas manquante — En Python

```
points = pd.read_csv("water-point-monitoring-2024.v1.csv")
readings = points["free_residual_chlorine_mgl"].dropna()

censored = readings.astype(str).str.startswith("<")
print(f"tested {len(readings)}, censored {censored.sum()}")
print(readings[censored].unique())
```

La valeur censurée, et pourquoi elle n'est pas manquante — En R

```
points |>
  filter(!is.na(free_residual_chlorine_mgl)) |>
  count(censored = startsWith(free_residual_chlorine_mgl, "<"))
```

La valeur censurée, et pourquoi elle n'est pas manquante — En Python

```
numeric = pd.to_numeric(readings, errors="coerce")    # <-- the silent one
print(f"after coercion: {numeric.isna().sum()} became NaN")

in_range = readings[~censored].astype(float).between(0.2, 0.5)
print(f"in target range, over all tested:  {in_range.sum() / len(readings):.1%}")
print(f"in target range, dropping censored: {in_range.mean():.1%}")
```

La valeur censurée, et pourquoi elle n'est pas manquante — En R

```
points |>
  filter(!is.na(free_residual_chlorine_mgl),
         !startsWith(free_residual_chlorine_mgl, "<")) |>
  summarise(in_range = mean(between(as.numeric(free_residual_chlorine_mgl), 0.2, 0.5)))
```

La valeur censurée, et pourquoi elle n'est pas manquante

- 30,9 % contre 46,5 % — Supprimer les lectures censurées relève la conformité apparente à la cible de chloration de près. . .

Que faire d'une valeur censurée

- Classer plutôt que moyenner — La cible est une plage, si bien qu'une lecture censurée répond parfaitement à la question. . .
- Substituer la limite, ou sa moitié — s'il vous faut vraiment une moyenne
- Rapporter la part sous la limite comme un nombre à part entière — 33,4 % de ces lectures sont sous le seuil de. . .

Que faire d'une valeur censurée — En Python

```
compliant = pd.to_numeric(readings.where(~censored), errors="coerce").between(0.2, 0.5)
result = pd.DataFrame({
    "readings": [len(readings)],
    "below detection": [censored.sum()],
    "in 0.2-0.5 range": [compliant.sum()],
    "compliance": [f"{compliant.sum() / len(readings):.1%}"],
})
print(result)
```

Que faire d'une valeur censurée — En R

The denominator is every reading, including the ones below the limit.

Que faire d'une valeur censurée

- Ne laissez jamais une limite de détection rétrécir un dénominateur — C'est la façon la plus courante dont un rapport de . .

L'incohérence entre champs qu'il faut nommer — En Python

```
treats = households["water_treated_at_home"]  
print(f"treat at home: {treats.sum()}")  
print(f"  of which no chlorine reading: {(treats & ~chlorine).sum()}")
```

L'incohérence entre champs qu'il faut nommer — En R

```
households |> filter(water_treated_at_home) |>  
  count(no_reading = is.na(free_residual_chlorine_mgl))
```

Rapportez-le comme un bloc à part — Exemple (suite)

Water quality

E. coli, point of collection	802 households tested (33.4% of survey)	
Safe <1 CFU/100 mL	53.6%	
Low 1-10	21.6%	
Moderate 11-100	17.3%	
High >100	7.5%	
Improved sources: 61.1% safe (n=625)		
Unimproved and surface: 27.1% safe (n=177)		
Free residual chlorine, water points	811 visits tested (30.8% of visits)	
Below detection limit (<0.1 mg/L)	33.4%	counted as non-compliant
Within 0.2-0.5 mg/L target	30.9%	

Quality was tested on a subsample and does not share the denominator of the access indicators above. 533 households reporting home treatment have no

Rapportez-le comme un bloc à part — Exemple (suite)

chlorine reading, so the tested subsample over-represents untreated water.

- Tout ce qui précède est une coupe transversale : ce qui était vrai le jour où quelqu'un est passé.

Lire la leçon complète, avec le code exécutable

https:

`//data-analysis.cassion.dev/fr/cours/wash-analysis/wash-04-water-quality`