

## Des dénominateurs enchaînés

Une cascade est une suite où le dénominateur de chaque étape est le numérateur de la précédente. 1 850 cas de protection deviennent 757 atteignant un service, et le nombre utile n'est pas les 41 % — c'est le maillon qui a le plus perdu.

---

Pierre Robentz Cassion

Leçon 2 sur 8

Santé publique et épidémiologie appliquée — Le dénominateur, c'est l'épidémiologie

## Ce que couvre cette leçon

- La forme
- Deux façons d'exprimer chaque étape
- Le premier dénominateur est le difficile
- La cascade vaccinale
- Surveillez le sens de l'abandon
- Dessinez-la, mais rappez le tableau
- La suite

## La forme — Exemple

people living with HIV

- > diagnosed

- > on treatment

- > virally suppressed

## Deux façons d'exprimer chaque étape — En Python (suite)

```
import pandas as pd

referrals = pd.read_csv("protection-referrals-2024.v1.csv")

steps = [
    ("cases", len(referrals)),
    ("consented", (referrals["consent_to_refer"] == True).sum()),
    ("referral made", ((referrals["consent_to_refer"] == True)
                       & (referrals["referral_made"] == True)).sum()),
    ("reached a service", ((referrals["consent_to_refer"] == True)
                           & (referrals["referral_made"] == True)
                           & (referrals["referral_accepted"] == True)).sum()),
]

first = steps[0][1]
previous = None
```

## Deux façons d'exprimer chaque étape — En Python (suite)

```
for name, count in steps:
    line = f"{name:20} {count:5} {count / first:6.1%} of all"
    if previous:
        line += f" {count / previous:6.1%} of previous"
    print(line)
    previous = count
```

## Deux façons d'exprimer chaque étape — En R

```
library(dplyr)

referrals |>
  summarise(
    cases = n(),
    consented = sum(consent_to_refer),
    referred = sum(consent_to_refer & referral_made),
    reached = sum(consent_to_refer & referral_made & referral_accepted)
  )
```

## Deux façons d'exprimer chaque étape

| Étape                            | n     | Sur le total | Sur la précédente |
|----------------------------------|-------|--------------|-------------------|
| Cas                              | 1 850 | 100 %        | —                 |
| Ont consenti au<br>référéncement | 1 638 | 88,5 %       | 88,5 %            |
| Référéncement émis               | 1 142 | 61,7 %       | <b>69,7 %</b>     |
| Ont atteint un service           | 757   | 40,9 %       | 66,3 %            |

## Deux façons d'exprimer chaque étape

- Les deux colonnes sont nécessaires et elles disent des choses différentes
- Classez par perte conditionnelle, agissez sur la plus grande

## Deux façons d'exprimer chaque étape — En Python

```
losses = pd.DataFrame(steps, columns=["step", "n"])
losses["lost"] = losses["n"].shift(1) - losses["n"]
losses["conditional_loss"] = losses["lost"] / losses["n"].shift(1)
print(losses.dropna().sort_values("conditional_loss", ascending=False))
```

## Deux façons d'exprimer chaque étape — En R

```
tibble::tibble(step = c("consented", "referred", "reached"),  
              n = c(1638, 1142, 757), previous = c(1850, 1638, 1142)) |>  
  mutate(conditional_loss = (previous - n) / previous) |>  
  arrange(desc(conditional_loss))
```

## Le premier dénominateur est le difficile

- **VIH** — les personnes vivant avec le VIH. Personne ne les compte ; le chiffre vient d'un modèle, et tout « %... »
- **Tuberculose** — les cas incidents estimés. De même, et l'écart entre estimés et notifiés est le constat phare de la... »
- **Protection** — les personnes ayant subi un incident. Inconnaissable, et la raison pour laquelle la cascade de... »

## Le premier dénominateur est le difficile — En Python

```
print("cascade base: cases known to the system, not people in need")
```

## Le premier dénominateur est le difficile — En R

*# The base of this cascade is cases the system saw. It is not incidence.*

## Le premier dénominateur est le difficile

- Dites quelle base vous avez employée — Une cascade partant d'un besoin estimé et une partant des cas déjà dans le...

## La cascade vaccinale — En Python

```
vax = pd.read_csv("vaccination-coverage-2024.v1.csv")
reported = vax[vax["report_submitted"] == True]
doses = reported.groupby("antigen")["doses_administered"].sum()

for first_dose, last_dose in [("penta1", "penta3"), ("mcv1", "mcv2")]:
    dropout = (doses[first_dose] - doses[last_dose]) / doses[first_dose]
    print(f"{first_dose} -> {last_dose}: {doses[first_dose]:,} -> "
          f"{doses[last_dose]:,}, dropout {dropout:.1%}")
```

## La cascade vaccinale — En R

```
doses <- vax |> filter(report_submitted) |>  
  summarise(doses = sum(doses_administered), .by = antigen)
```

- Penta1 vers penta3 : 13,8 % d'abandon. Première vers seconde dose de rougeole : 22,9 %
- C'est robuste au dénominateur — Numérateur et dénominateur viennent des mêmes formations aux mêmes mois, si bien qu'une. . .
- Cela mesure le mécanisme — La couverture monte si davantage d'enfants commencent ; l'abandon ne baisse que si davantage. . .

## Surveillez le sens de l'abandon — En Python

```
if doses["penta3"] > doses["penta1"]:
    print("negative dropout: check the denominator and the period")
```

## Surveillez le sens de l'abandon — En R

```
if (doses$doses[doses$antigen == "penta3"] > doses$doses[doses$antigen == "penta1"])  
  warning("negative dropout")
```

## Dessinez-la, mais rapportez le tableau — En Python

```
cascade = pd.DataFrame({
    "step": ["Cases", "Consented", "Referral made", "Reached service"],
    "n": [1850, 1638, 1142, 757],
})
cascade["of_all"] = cascade["n"] / cascade["n"].iloc[0]
cascade["of_previous"] = cascade["n"] / cascade["n"].shift(1)
cascade["base"] = "cases known to the system"
```

## Dessinez-la, mais rapportez le tableau — En R

```
tibble::tribble(  
  ~step,      ~n,  
  "Cases",    1850,  
  "Consented", 1638,  
  "Referral made", 1142,  
  "Reached",   757  
) |> mutate(of_all = n / first(n), of_previous = n / lag(n))
```

- Une cascade est un ensemble de décomptes déjà agrégés.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/public-health-epidemiology/  
epi-02-cascades](https://data-analysis.cassion.dev/fr/cours/public-health-epidemiology/epi-02-cascades)