

## **L'écart s'ouvre avant que le référencement ne soit fait**

Les cas avec handicap signalé aboutissent à 26,7 % contre 46,2 %. L'écart n'est pas à la porte du service — le consentement est identique à 89 % et 88,5 % — il s'ouvre à la porte où un travailleur social décide de faire ou non le référencement.

---

Pierre Robentz Cassion

Leçon 4 sur 8

Données de protection et VBG — Le circuit

## Ce que couvre cette leçon

- Trouvez la porte, non le total
- Quel service, et quelle zone
- L'écart qui compte
- Avant de le publier
- Rapportez les portes, non le total
- La suite

## Trouvez la porte, non le total — En Python (suite)

```
import pandas as pd

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

def cascade(frame):
    consented = frame[frame["consent_to_refer"]]
    made = consented[consented["referral_made"]]
    accepted = made[made["referral_accepted"]]
    reached = accepted[accepted["days_to_first_service"].notna()]
    return pd.Series({
        "n": len(frame), "consented": len(consented), "made": len(made),
        "accepted": len(accepted), "reached": len(reached),
    })

totals = cascade(referrals)
print(totals)
```

## Trouvez la porte, non le total — En Python (suite)

```
print("\nloss at each gate:")
print(f"  consent:  {1 - totals['consented'] / totals['n']:.1%}")
print(f"  referral: {1 - totals['made'] / totals['consented']:.1%}")
print(f"  accepted:  {1 - totals['accepted'] / totals['made']:.1%}")
print(f"  service:   {1 - totals['reached'] / totals['accepted']:.1%}")
```

## Trouvez la porte, non le total — En R

```
library(dplyr)

referrals |> summarise(
  n = n(),
  consented = sum(consent_to_refer),
  made = sum(consent_to_refer & referral_made),
  accepted = sum(referral_made & referral_accepted),
  reached = sum(referral_accepted & !is.na(days_to_first_service))
)
```

# Trouvez la porte, non le total

| Porte                         | Perte à ce point                     |
|-------------------------------|--------------------------------------|
| Consentement                  | 11,5 % — une décision, non une perte |
| <b>Référencement effectué</b> | <b>30,3 %</b>                        |
| <b>Référencement accepté</b>  | <b>33,7 %</b>                        |
| Service atteint               | 5,3 %                                |

## Trouvez la porte, non le total

- Les deux portes du milieu perdent presque tout — Une fois le référencement accepté, 94,7 % des cas atteignent le...

## Quel service, et quelle zone — En Python

```
by_service = referrals.groupby("service_requested").apply(cascade,  
                                                         include_groups=False)  
by_service["completion"] = by_service["reached"] / by_service["consented"]  
print((by_service["completion"] * 100).round(1).sort_values())
```

## Quel service, et quelle zone — En R

```
referrals |>
  summarise(consented = sum(consent_to_refer),
            reached = sum(referral_accepted & !is.na(days_to_first_service)),
            .by = service_requested) |>
  mutate(completion = reached / consented) |> arrange(completion)
```

## Quel service, et quelle zone

| Service                                | Aboutissement | Référencement fait | Accepté |
|----------------------------------------|---------------|--------------------|---------|
| <b>Appui aux moyens de subsistance</b> | <b>20,7 %</b> | 49 %               | 47 %    |
| Juridique                              | 30,0 %        | 61 %               | 50 %    |
| Sécurité et protection physique        | 33,8 %        | 61 %               | 58 %    |
| Psychosocial                           | 53,2 %        | 77 %               | 72 %    |
| <b>Santé</b>                           | <b>57,6 %</b> | 81 %               | 76 %    |

- L'appui aux moyens de subsistance aboutit au tiers du taux de la santé — et il échoue également aux deux portes du...

## L'écart qui compte — En Python

```
disability = referrals["disability_reported"].isin([True, "true", "Yes"])
no_disability = referrals["disability_reported"].isin([False, "false", "No"])

for label, mask in [("reported", disability), ("not reported", no_disability)]:
    row = cascade(referrals[mask])
    print(f"{label:14} n={row['n']:>5}  "
          f"consent {row['consented'] / row['n']:.1%}  "
          f"made {row['made'] / row['consented']:.1%}  "
          f"accepted {row['accepted'] / row['made']:.1%}  "
          f"→ {row['reached'] / row['consented']:.1%}")
```

```
referrals |>
  mutate(disability = disability_reported %in% c("true", "Yes")) |>
  summarise(n = n(), consented = sum(consent_to_refer),
            made = sum(consent_to_refer & referral_made),
            accepted = sum(referral_made & referral_accepted),
            reached = sum(referral_accepted & !is.na(days_to_first_service)),
            .by = disability)
```

## L'écart qui compte

|                     | n     | Consentement | Référencement<br>fait | Accepté       | Aboutissement |
|---------------------|-------|--------------|-----------------------|---------------|---------------|
| Handicap<br>signalé | 227   | 89,0 %       | <b>54,5 %</b>         | <b>56,4 %</b> | <b>26,7 %</b> |
| Non signalé         | 1 623 | 88,5 %       | 71,9 %                | 67,3 %        | 46,2 %        |

- Le consentement est identique — 89,0 % contre 88,5 % — Les personnes en situation de handicap sont exactement aussi. . .
- Localiser l'écart à une porte précise transforme une observation en action — « Les résultats sont plus mauvais pour les. . .

- La colonne handicap a quatre valeurs — Une zone a saisi Yes et No au lieu de true et false, si bien qu'un...

```
print(referrals["disability_reported"].value_counts(dropna=False))
```

```
referrals |> count(disability_reported)
```

- Le handicap est auto-déclaré et sous-déclaré — 227 sur 1 850 font 12,3 %, en dessous de la plupart des estimations de. . .
- Vérifiez la taille des cellules avant de désagréger davantage — Handicap par zone par service est exactement le tableau. . .

# Rapportez les portes, non le total — Exemple

Referral pathway, 1,638 consenting cases

|                              |       |
|------------------------------|-------|
| Referral made                | 69.7% |
| Accepted, of those made      | 66.3% |
| Reached service, of accepted | 94.7% |
| End to end, of consenting    | 43.8% |

The pathway fails upstream. Once a referral is accepted 94.7% of cases reach a service; the losses are at referral-making and acceptance.

Cases reporting a disability: 26.7% complete against 46.2%. Consent is identical (89.0% and 88.5%); the gap is entirely at referral-making (54.5% against 71.9%) and acceptance (56.4% against 67.3%).

Disability is self-reported by 12.3% of cases. Non-disclosure places some people in the comparison group, so the gap is a lower bound.

- Le circuit s'arrête quand un cas atteint un service.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/protection-and-gbv-data/  
prot-04-where-the-pathway-leaks](https://data-analysis.cassion.dev/fr/cours/protection-and-gbv-data/prot-04-where-the-pathway-leaks)