

74,4 %, 33,9 %, 83,3 %

Trois taux de fonctionnalité issus d'un seul registre, tous exacts. Ils diffèrent parce qu'ils comptent des visites, des points et des personnes, et un seul répond à la question qu'on pose réellement à un programme d'eau.

Pierre Robentz Cassion

Leçon 5 sur 8

Analyse EAH — Le service est une année, non une journée

Ce que couvre cette leçon

- Ce que rend possible un registre à visites répétées
- Taux un : la part des visites ayant trouvé un point en service
- Taux deux : la part des points en service à chaque fois
- Taux trois : la part des personnes disposant d'un point en service
- Lequel rapporter
- Les deux points comptés deux fois
- La suite

Ce que rend possible un registre à visites répétées — En Python

```
import pandas as pd

points = pd.read_csv("water-point-monitoring-2024.v1.csv", parse_dates=["visit_date"])

print(f"visits: {len(points):,}")
print(f"water points: {points['water_point_id'].nunique()}")
print(points["functional_status"].value_counts())
```

Ce que rend possible un registre à visites répétées — En R

```
library(dplyr)
```

```
points |> summarise(visits = n(), water_points = n_distinct(water_point_id))
```

```
points |> count(functional_status)
```

Taux un : la part des visites ayant trouvé un point en service — En Python

```
WORKING = {"functional", "partially-functional"}  
working = points["functional_status"].isin(WORKING)  
  
print(f"visit-level functionality: {working.mean():.1%} of {len(points):,} visits")
```

Taux un : la part des visites ayant trouvé un point en service — En R

```
points |> summarise(functionality = mean(functional_status %in%  
  c("functional", "partially-functional")), n = n())
```

Taux un : la part des visites ayant trouvé un point en service

- 74,4 % — C'est ce que publie presque tout rapport sur les points d'eau, et cela répond à *« si je visite un point au. . .
- Partiellement fonctionnel compte comme en service — Un point qui tourne à débit réduit fournit de l'eau, et le classer. . .

Taux deux : la part des points en service à chaque fois — En Python

```
by_point = points.assign(ok=working).groupby("water_point_id")["ok"]
always = by_point.all()

print(f"point-level functionality: {always.mean():.1%} of {len(always)} points")
print(f"points that failed at least once: {(~always).sum()}")
```

Taux deux : la part des points en service à chaque fois — En R

```
points |>
  summarise(always = all(functional_status %in%
    c("functional", "partially-functional")), .by = water_point_id) |>
  summarise(share = mean(always), n = n())
```

Taux deux : la part des points en service à chaque fois

- 33,9 % — 82 points sur 242
- Les deux taux ne se contredisent pas — 74,4 % des visites et 33,9 % des points sont vrais tous les deux, du même. . .

Taux trois : la part des personnes disposant d'un point en service — En Python

```
served = points.dropna(subset=["users_estimated"])
population_weighted = (
    served.loc[served["functional_status"].isin(WORKING), "users_estimated"].sum()
    / served["users_estimated"].sum()
)
print(f"population-weighted functionality: {population_weighted:.1%}")
```

Taux trois : la part des personnes disposant d'un point en service — En R

```
points |>
  filter(!is.na(users_estimated)) |>
  summarise(weighted = sum(users_estimated[functional_status %in%
    c("functional", "partially-functional")]) / sum(users_estimated))
```

Taux trois : la part des personnes disposant d'un point en service

- 83,3 % — près de neuf points au-dessus du chiffre par visite

Taux trois : la part des personnes disposant d'un point en service — En Python

```
print(points.groupby("source_type").agg(  
    points=("water_point_id", "nunique"),  
    median_users=("users_estimated", "median"),  
    functionality=("functional_status", lambda s: s.isin(WORKING).mean()),  
) .round(3))
```

Taux trois : la part des personnes disposant d'un point en service — En R

```
points |>
  summarise(n = n_distinct(water_point_id),
            median_users = median(users_estimated, na.rm = TRUE),
            functionality = mean(functional_status %in%
                                c("functional", "partially-functional")), .by = source_type)
```

Taux trois : la part des personnes disposant d'un point en service

Type de source	Fonctionnalité	Usagers typiques
Borne de réseau	97,2 %	environ 1 290
Forage à pompe manuelle	75,5 %	environ 280
Source protégée	66,6 %	environ 200
Puits protégé	58,1 %	environ 185

Taux trois : la part des personnes disposant d'un point en service

- Les points qui desservent le plus de monde tombent le moins en panne — Pondérer par la population fait donc monter le...

Lequel rapporter — Exemple

Water point functionality, 2024

Visit-level	74.4%	1,957 of 2,629 monitoring visits
Point-level	33.9%	82 of 242 points working at every visit
Population-weighted	83.3%	of an estimated 96,700 users

Partially functional counted as working (134 visits).

Two points were re-registered after a handover and are counted twice;
removing them moves the visit-level figure to 74.7%.

Lequel rapporter

- S'il faut n'en donner qu'un, donnez le taux pondéré par la population et dites-le — car le standard porte sur des. . .

Les deux points comptés deux fois — En Python

```
duplicates = points[points["water_point_id"].str.startswith("WP09")]
print(duplicates.groupby("water_point_id")["community"].agg(["nunique", "first"]))
```

Les deux points comptés deux fois — En R

```
points |> filter(startsWith(water_point_id, "WP09")) |> count(water_point_id)
```

- Deux tiers de ces points ont défailli au moins une fois.

Lire la leçon complète, avec le code exécutable

```
https://data-analysis.cassion.dev/fr/cours/wash-analysis/  
wash-05-three-functionality-rates
```