

Le taux de rapportage est un dénominateur, pas une note de bas de page

Le système calcule quatre mesures de complétude et elles divergent. Celle que cite votre rapport décide si un taux de couverture est une estimation ou une borne inférieure, et août vaut 29 % dans les deux cas.

Pierre Robentz Cassion

Leçon 4 sur 8

Données de routine et DHIS2 — Périodes et complétude

Ce que couvre cette leçon

- Quatre mesures, un seul mot
- Là où la réponse du système diffère de la vôtre
- Le taux de rapportage comme dénominateur de tout le reste
- Les deux façons de traiter une formation muette
- Rapportez la paire
- La suite

Quatre mesures, un seul mot

Mesure	Numérateur	Dénominateur
Taux de rapportage	Ensembles marqués complets	Ensembles attendus
Rapports effectifs	Ensembles marqués complets	— (un décompte)
Rapports attendus	—	Unités affectées × périodes
Taux de rapportage à temps	Complétés avant l'échéance	Ensembles attendus

Quatre mesures, un seul mot — En Python

```
import pandas as pd

vax = pd.read_csv("vaccination-coverage-2024.v1.csv", parse_dates=["period"])
vax["reported"] = vax["report_submitted"] == True

facility_months = vax.drop_duplicates(["facility_id", "period"])
print(f"expected: {len(facility_months)}")
print(f"actual:    {facility_months['reported'].sum()}")
print(f"rate:      {facility_months['reported'].mean():.1%}")
```

Quatre mesures, un seul mot — En R

```
library(dplyr)

vax |>
  distinct(facility_id, period, .keep_all = TRUE) |>
  summarise(expected = n(), actual = sum(report_submitted),
            rate = mean(report_submitted))
```

Là où la réponse du système diffère de la vôtre

- Les rapports attendus comptent l'affectation, non l'existence — Une formation fermée en mars reste au dénominateur le. . .
- La complétude est un clic, non un calcul — Une formation peut saisir toutes ses valeurs sans jamais marquer le. . .

Là où la réponse du système diffère de la vôtre — En Python

```
has_data = (  
    vax.groupby(["facility_id", "period"])["doses_administered"]  
    .sum().gt(0).rename("has_data")  
)  
flags = facility_months.set_index(["facility_id", "period"])["reported"]  
  
crosstab = pd.crosstab(flags, has_data.reindex(flags.index))  
print(crosstab)
```

Là où la réponse du système diffère de la vôtre — En R

```
vax |>
  summarise(reported = first(report_submitted),
            has_data = sum(doses_administered) > 0,
            .by = c(facility_id, period)) |>
  count(reported, has_data)
```

Le taux de rapportage comme dénominateur de tout le reste — En Python

```
monthly = (  
    vax[vax["antigen"] == "penta3"]  
    .groupby(vax["period"].dt.strftime("%Y-%m"))  
    .apply(lambda g: pd.Series({  
        "reporting_rate": g["reported"].mean(),  
        "coverage_reporting": (g.loc[g["reported"], "doses_administered"].sum()  
                               / g.loc[g["reported"], "target_population"].sum()),  
    })))  
)  
print((monthly * 100).round(1))
```

Le taux de rapportage comme dénominateur de tout le reste — En R

```
vax |>
  filter(antigen == "penta3") |>
  mutate(month = format(period, "%Y-%m")) |>
  summarise(
    reporting_rate = mean(report_submitted),
    coverage = sum(doses_administered[report_submitted]) /
      sum(target_population[report_submitted]),
    .by = month
  )
```

Le taux de rapportage comme dénominateur de tout le reste

Mois	Taux de rapportage	Couverture parmi les déclarants
Juillet	82 %	...
Août	29 %	...
Septembre	45 %	...
Octobre	95 %	...

Le taux de rapportage comme dénominateur de tout le reste

- Trois règles en découlent, et elles sont le livrable de cette leçon
- **En dessous de 90 % de rapportage, un taux de couverture est une borne inférieure**, non une estimation. Étiquetez-le. . .
- **Ne comparez pas deux périodes aux taux de rapportage sensiblement différents sans le dire.** Août face à juillet. . .
- **Une courbe de tendance exige le taux de rapportage sur le même graphique.** Tracée seule, une série de couverture. . .

Les deux façons de traiter une formation muette

- L'exclure du numérateur et du dénominateur — Couverture parmi les formations ayant rapporté
- Imputer ses doses — Remplir à partir des mois récents de la formation, en le disant

Les deux façons de traiter une formation muette — En Python

```
by_facility = (  
    vax[(vax["antigen"] == "penta3") & vax["reported"]]  
    .groupby("facility_id")["doses_administered"].median()  
)  
missing = (vax["antigen"] == "penta3") & ~vax["reported"]  
imputed_total = (vax.loc[missing, "facility_id"].map(by_facility).sum()  
    + vax.loc[(vax["antigen"] == "penta3") & vax["reported"],  
        "doses_administered"].sum())  
  
annual_target = (vax[vax["antigen"] == "penta3"]  
    .groupby("facility_id")["target_population"].first().sum())  
print(f"imputed annual coverage: {imputed_total / annual_target:.1%}")
```

Les deux façons de traiter une formation muette — En R

```
median_by_facility <- vax |>  
  filter(antigen == "penta3", report_submitted) |>  
  summarise(m = median(doses_administered), .by = facility_id)
```

Les deux façons de traiter une formation muette

- N'imputez jamais en silence — Un chiffre imputé et un chiffre rapporté dans la même colonne sans marquage sont le. . .

```
summary = pd.DataFrame({  
    "period": ["2024-08", "2024-09", "2024-10"],  
    "facilities_reporting": [11, 17, 36],  
    "facilities_expected": [38, 38, 38],  
    "reporting_rate": ["29%", "45%", "95%"],  
    "coverage_basis": ["among reporting facilities"] * 3,  
})
```

```
tibble::tribble(  
  ~period,    ~reporting, ~expected, ~basis,  
  "2024-08",      11,       38, "among reporting facilities",  
  "2024-09",      17,       38, "among reporting facilities",  
  "2024-10",      36,       38, "among reporting facilities"  
)
```

- Tout ce qui précède a tourné sur un CSV exporté à la main.

Lire la leçon complète, avec le code exécutable

```
https://data-analysis.cassion.dev/fr/cours/routine-data-and-dhis2/  
dhis2-04-reporting-rate-as-denominator
```