

Les deux indicateurs que personne ne calcule

Complétude et promptitude du rapportage, chacune avec son dénominateur.
Onze formations sanitaires sur trente-huit ont rapporté en août, et tout taux de couverture de ce mois est une affirmation sur ces onze-là.

Pierre Robentz Cassion

Leçon 2 sur 8

Évaluation de la qualité des données — Ce que qualité veut dire ici

Ce que couvre cette leçon

- Deux indicateurs, et aucun n'est dans le rapport
- La complétude du rapportage
- Décomposez toujours de trois façons
- Le piège du dénominateur
- La promptitude exige un champ que vous n'avez peut-être pas
- Lisez tous les autres chiffres à travers ces deux-là
- La suite

Deux indicateurs, et aucun n'est dans le rapport

- Ouvrez presque n'importe quel rapport sanitaire de routine de ce secteur et vous y trouverez la couverture, la charge de cas, les taux de guérison et les ruptures de stock.

La complétude du rapportage — En Python

```
import pandas as pd

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

print(f"overall reporting rate: {vax['reported'].mean():.1%}")

by_month = vax.groupby(vax["period"].dt.strftime("%Y-%m"))["reported"].mean()
print((by_month * 100).round(0))
```

La complétude du rapportage — En R

```
library(dplyr)
library(readr)

vax <- read_csv("vaccination-coverage-2024.v1.csv")

mean(vax$report_submitted)

vax |>
  mutate(month = format(period, "%Y-%m")) |>
  summarise(reporting_rate = mean(report_submitted), .by = month) |>
  arrange(month)
```

La complétude du rapportage

Mois	Taux de rapportage
Janvier	92 %
Février	82 %
Mars	74 %
Avril à juillet	82 à 87 %
Août	29 %
Septembre	45 %
Octobre	95 %
Novembre et décembre	76 à 87 %

Décomposez toujours de trois façons — En Python

```
by_facility = vax.groupby("facility_id")["reported"].mean().sort_values()  
print(by_facility.head(6))  
print(f"{{(by_facility == 1).sum()}} facilities reported every month")
```

```
by_type = vax.groupby("facility_type")["reported"].mean()  
print((by_type * 100).round(1))
```

Décomposez toujours de trois façons — En R

```
vax |> summarise(rate = mean(report_submitted), .by = facility_id) |> arrange(rate) |> head  
      (6)
```

```
vax |> summarise(rate = mean(report_submitted), .by = facility_type)
```

Décomposez toujours de trois façons

Type de formation	Taux de rapportage
Centre de santé	82,8 %
Hôpital de district	75,0 %
Poste de santé	71,7 %

Le piège du dénominateur — En Python

```
# WRONG: the share of reports that were submitted, among reports that were submitted  
wrong = vax[vax["reported"]]["reported"].mean()           # always 1.0
```

Le piège du dénominateur — En R

WRONG

```
vax |> filter(report_submitted) |> summarise(rate = mean(report_submitted))
```

Le piège du dénominateur — En Python

```
EXPECTED_FACILITIES = pd.read_csv("facility-master-list.csv")["facility_id"]

expected = len(EXPECTED_FACILITIES) * vax["period"].nunique()
submitted = vax.loc[vax["reported"], ["facility_id", "period"]].drop_duplicates().shape[0]
print(f"{submitted} of {expected} expected facility-months")
```

Le piège du dénominateur — En R

```
expected <- nrow(facility_master) * n_distinct(vax$period)
submitted <- vax |> filter(report_submitted) |> distinct(facility_id, period) |> nrow()
```

La promptitude exige un champ que vous n'avez peut-être pas — En Python

```
submissions["days_late"] = (  
    submissions["submitted_on"] - submissions["deadline"]  
) .dt.days  
  
print(f"on time: {(submissions['days_late'] <= 0).mean():.1%}")  
print(f"median days late, of those late: "  
      f"{submissions.loc[submissions['days_late'] > 0, 'days_late'].median():.0f}")
```

La promptitude exige un champ que vous n'avez peut-être pas — En R

```
submissions <- submissions |>
  mutate(days_late = as.integer(submitted_on - deadline))

c(on_time = mean(submissions$days_late <= 0),
  median_late = median(submissions$days_late[submissions$days_late > 0]))
```

La promptitude exige un champ que vous n'avez peut-être pas

- Cette extraction ne permet pas ce calcul — et le dire est en soi un constat d'évaluation

La promptitude exige un champ que vous n'avez peut-être pas

Constat. La promptitude ne peut pas être évaluée. L'extraction mensuelle porte un indicateur de transmission mais aucune date de transmission, alors que DHIS2 en enregistre une. **Action corrective.** Ajouter la date de complétude à l'extraction standard. **Responsable.** Point focal SNIS. **Échéance.** Prochaine extraction trimestrielle.

Lisez tous les autres chiffres à travers ces deux-là

- **La couverture, à 76,5 % de rapportage.** Un total de district est une borne inférieure, pas une estimation. Écrivez «... »
- **Une tendance mensuelle.** La baisse d'août est une baisse de rapportage jusqu'à preuve du contraire. Tracez le taux. . .
- **Un classement de formations.** Des formations ayant rapporté sept mois sur douze ne peuvent pas être classées face à. . .

Lisez tous les autres chiffres à travers ces deux-là — En Python

```
monthly = (  
    vax[vax["reported"]]  
    .groupby(vax["period"].dt.strftime("%Y-%m"))  
    .apply(lambda g: g["doses_administered"].sum() / g["target_population"].sum())  
    .rename("coverage")  
    .to_frame()  
)  
monthly["reporting_rate"] = by_month  
print(monthly.round(3))
```

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

Deux colonnes, toujours adjacentes. Un taux de couverture sans son taux de rapportage est un nombre sans étiquette, et l'étiquette est ce qui l'empêche d'être mal lu.

- Complétude et promptitude disent si un rapport est arrivé.

Lire la leçon complète, avec le code exécutable

`https://data-analysis.cassion.dev/fr/cours/data-quality-assessment/
dqa-02-completeness-and-timeliness`