

Comparer une formation sanitaire à son propre passé

Rapport à la médiane glissante, le problème des petits effectifs qui rend les plus grandes aberrations insignifiantes, et la série si stable qu'elle mérite un second regard.

Pierre Robentz Cassion

Leçon 5 sur 8

Évaluation de la qualité des données — Repérer les sites qui méritent une visite

Ce que couvre cette leçon

- La cohérence est une comparaison, et vous en choisissez le terme
- Rapport à la médiane
- Le problème des petits effectifs
- Adjoignez un plancher absolu au rapport
- Le sens n'est pas symétrique
- Des séries trop stables
- Classez les formations, ne signalez pas des lignes
- La suite

La cohérence est une comparaison, et vous en choisissez le terme

- À ses pairs — La formation A a rapporté 40 doses ; la médiane du district est de 25
- À son propre passé — La formation A a rapporté 40 doses ce mois-ci, contre sa propre médiane annuelle de 25

Rapport à la médiane — En Python

```
import pandas as pd

reported = vax[vax["reported"]].copy()

reported["own_median"] = reported.groupby(["facility_id", "antigen"])[
    "doses_administered"
].transform("median")
reported["ratio"] = reported["doses_administered"] / reported["own_median"]

print(reported.nlargest(5, "ratio")[
    ["facility_id", "antigen", "period", "doses_administered", "own_median", "ratio"]
])
```

Rapport à la médiane — En R

```
library(dplyr)

reported <- vax |>
  filter(report_submitted) |>
  mutate(own_median = median(doses_administered),
         ratio = doses_administered / own_median,
         .by = c(facility_id, antigen))

reported |> slice_max(ratio, n = 5)
```

- La médiane, pas la moyenne — La moyenne est tirée par l'aberration même que vous cherchez, si bien qu'une valeur peut. . .

```
reported = reported.sort_values(["facility_id", "antigen", "period"])
reported["rolling_median"] = (
    reported.groupby(["facility_id", "antigen"])["doses_administered"]
    .transform(lambda s: s.rolling(6, min_periods=3, center=True).median())
)
```

```
reported <- reported |>  
  arrange(facility_id, antigen, period) |>  
  mutate(rolling_median = zoo::rollapply(doses_administered, 6, median,  
                                          partial = TRUE, align = "center"),  
         .by = c(facility_id, antigen))
```

Le problème des petits effectifs

Formation	Antigène	Mois	Valeur	Médiane propre	Rapport
FAC013	mcv2	décembre	5	3	1,67
FAC037	opv3	septembre	13	8	1,62
FAC030	mcv2	octobre	8	5	1,60

Adjoignez un plancher absolu au rapport — En Python

```
RATIO_HIGH, RATIO_LOW, MIN_ABSOLUTE = 2.0, 0.5, 10

reported["difference"] = reported["doses_administered"] - reported["own_median"]
reported["flag"] = (
    ((reported["ratio"] > RATIO_HIGH) | (reported["ratio"] < RATIO_LOW))
    & (reported["difference"].abs() >= MIN_ABSOLUTE)
)
print(f"{reported['flag'].sum()} flags from {len(reported)} facility-antigen-months")
```

Adjoignez un plancher absolu au rapport — En R

```
reported <- reported |>
  mutate(
    difference = doses_administered - own_median,
    flag = (ratio > 2 | ratio < 0.5) & abs(difference) >= 10
  )
```

Adjoignez un plancher absolu au rapport

- Voilà le constat, et il tient en une ligne — Un contrôle qui produit un seul élément à investiguer sur 2 094. . .

Le sens n'est pas symétrique

- **Une baisse** est généralement réelle ou liée au rapportage. Rupture de stock, absence de personnel, grève, formation. . .
- **Un pic** est généralement une campagne, une sortie de proximité ou un rattrapage après rupture — toutes légitimes —. . .

Le sens n'est pas symétrique — En Python

```
reported["previous"] = reported.groupby(["facility_id", "antigen"])[
    "doses_administered"].shift(1)
reported["next"] = reported.groupby(["facility_id", "antigen"])[
    "doses_administered"].shift(-1)

spike_and_dip = reported[
    (reported["ratio"] > 1.5)
    & (reported["previous"] < reported["own_median"] * 0.7)
]
```

Le sens n'est pas symétrique — En R

```
reported <- reported |>  
  mutate(previous = lag(doses_administered),  
         next_value = lead(doses_administered),  
         .by = c(facility_id, antigen))
```

Des séries trop stables — En Python

```
stability = (  
    reported.groupby(["facility_id", "antigen"])["doses_administered"]  
    .agg(["mean", "std", "size"])  
)  
stability["cv"] = stability["std"] / stability["mean"]  
print(stability[stability["size"] >= 8].nsmallest(5, "cv"))
```

```
reported |>
  summarise(mean = mean(doses_administered),
            cv = sd(doses_administered) / mean(doses_administered),
            n = n(),
            .by = c(facility_id, antigen)) |>
  filter(n >= 8) |>
  slice_min(cv, n = 5)
```

Classez les formations, ne signalez pas des lignes — En Python

```
risk = (  
    reported.groupby("facility_id")  
    .agg(flags=("flag", "sum"), months=("flag", "size"))  
    .assign(flag_rate=lambda d: d["flags"] / d["months"])  
    .sort_values("flag_rate", ascending=False)  
)  
print(risk.head(6))
```

Classez les formations, ne signalez pas des lignes — En R

```
risk <- reported |>  
  summarise(flags = sum(flag), months = n(), .by = facility_id) |>  
  mutate(flag_rate = flags / months) |>  
  arrange(desc(flag_rate))
```

- Les contrôles de tendance comparent un nombre à son propre passé.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/data-quality-assessment/
dqa-05-trend-and-outlier-checks](https://data-analysis.cassion.dev/fr/cours/data-quality-assessment/dqa-05-trend-and-outlier-checks)