Vingt-quatre tests, deux constats, aucun effet

Le registre d'inscription tire le sexe indépendamment de tout le reste, si bien que l'écart de retard d'âge selon le sexe vaut exactement zéro par construction. Testez-le école par école et deux des vingt-quatre ressortent significatives, ce qui est à peu près ce que le hasard vous devait.

Pierre Robentz Cassion

Leçon 5 sur 8

Ce que couvre cette leçon

- Une comparaison dont on connaît la réponse
- Deux, c'est ce que le hasard vous devait
- Bonferroni : divisez le seuil par le nombre de tests
- Benjamini-Hochberg, quand vingt-quatre devient deux cents
- La correction ne détruit pas les constats réels
- Comptez les tests, y compris ceux que vous n'avez pas rapportés
- Rapportez-le en entier
- La suite

Une comparaison dont on connaît la réponse — En Python (suite)

```
import pandas as pd
import numpy as np
from statsmodels.stats.proportion import proportions_ztest

enrolment = pd.read_csv("school-enrolment-2024.v1.csv")
clean = enrolment[(enrolment["school_year"] == 2024)
                  & (enrolment["age_years"] <= 20)
                  & enrolment["grade"].between(1, 6)]
clean = clean.assign(over_age=clean["age_years"] > clean["grade"] + 5)

results = []
for school, group in clean.groupby("school_id"):
    counts = group.groupby("sex")["over_age"].agg(["sum", "size"])
    if counts["size"].min() < 5:
        continue
    stat, p = proportions_ztest(counts["sum"], counts["size"])
```

Une comparaison dont on connaît la réponse — En Python (suite)

```
    results.append({"school": school, "p": p,
                    "boys": counts.loc["m", "sum"] / counts.loc["m", "size"],
                    "girls": counts.loc["f", "sum"] / counts.loc["f", "size"]})

tests = pd.DataFrame(results).sort_values("p")
print(f"{len(tests)} schools tested, {(tests['p'] < 0.05).sum()} significant")
```

Une comparaison dont on connaît la réponse — En R

```r
library(dplyr)

clean |>
  summarise(k = sum(over_age), n = n(), .by = c(school_id, sex)) |>
  tidyr::pivot_wider(names_from = sex, values_from = c(k, n)) |>
  filter(n_m >= 5, n_f >= 5) |>
  rowwise() |>
  mutate(p = prop.test(c(k_m, k_f), c(n_m, n_f))$p.value)
```

Une comparaison dont on connaît la réponse

École	Garçons	Filles	p
SCH23	62,5 % (10/16)	25,9 % (7/27)	**0,018**
SCH06	39,1 % (9/23)	12,5 % (3/24)	**0,036**
SCH21	43,5 % (10/23)	25,8 % (8/31)	0,173
SCH03	50,0 % (11/22)	30,8 % (8/26)	0,175
SCH04	27,3 % (3/11)	55,6 % (5/9)	0,199

Une comparaison dont on connaît la réponse

- Vingt-quatre écoles testées, deux significatives à $p < 0{,}05$ — Les deux seraient écrites
- Aucune des deux n'est réelle — Le générateur n'a jamais laissé le sexe toucher quoi que ce soit

Deux, c'est ce que le hasard vous devait — En Python

```
n_tests = len(tests)
print(f"tests run: {n_tests}")
print(f"expected significant if nothing is real: {0.05 * n_tests:.1f}")
print(f"chance of at least one: {1 - 0.95 ** n_tests:.1%}")
```

Deux, c'est ce que le hasard vous devait — En R

```
# 1 - 0.95^24. The arithmetic is one line and it is the whole lesson.
```

Deux, c'est ce que le hasard vous devait

- Faux positifs attendus : 1,2. Observés : 2 — La probabilité d'obtenir *au moins un* résultat significatif à partir de…
- Un seuil de 0,05 est une promesse portant sur un test — Il dit que si rien ne se passe, vous serez trompé une fois sur…
- C'est pourquoi « nous avons regardé par école et deux écoles se détachent » n'est pas un constat — C'est une…

Bonferroni : divisez le seuil par le nombre de tests — En Python

```
BONFERRONI = 0.05 / n_tests
print(f"corrected threshold: {BONFERRONI:.5f}")
print(f"surviving: {(tests['p'] < BONFERRONI).sum()}")
```

Bonferroni : divisez le seuil par le nombre de tests — En R

```r
p.adjust(tests$p, method = "bonferroni") < 0.05
```

Bonferroni : divisez le seuil par le nombre de tests

- Seuil 0,05/24 = 0,00208. Survivants : aucun — Le plus petit p des vingt-quatre vaut 0,018, un ordre de grandeur plus...

Benjamini-Hochberg, quand vingt-quatre devient deux cents — En Python

```python
from statsmodels.stats.multitest import multipletests

reject, adjusted, _, _ = multipletests(tests["p"], alpha=0.05, method="fdr_bh")
print(f"BH survivors: {reject.sum()}")
```

Benjamini-Hochberg, quand vingt-quatre devient deux cents — En R

```r
p.adjust(tests$p, method = "BH") < 0.05
```

Benjamini-Hochberg, quand vingt-quatre devient deux cents

- Benjamini-Hochberg ne retient lui non plus aucune des vingt-quatre — ce qui est la bonne réponse ici — il n'y a rien à...
- Employez Bonferroni pour une petite famille confirmatoire ; BH pour un large dépistage exploratoire — Les deux tiennent...

La correction ne détruit pas les constats réels — En Python

```
previous = enrolment[enrolment["school_year"] == 2023].set_index("student_id")
clean = clean.assign(last_year=clean["student_id"].map(previous["end_of_year_status"]))
returning = clean[clean["last_year"].isin(["repeated", "promoted"])]

survivors = []
for school, group in returning.groupby("school_id"):
    counts = group.groupby("last_year")["over_age"].agg(["sum", "size"])
    if len(counts) < 2 or counts["size"].min() < 5:
        continue
    stat, p = proportions_ztest(counts["sum"], counts["size"])
    survivors.append(p)

survivors = np.array(survivors)
print(f"{len(survivors)} schools, {(survivors < 0.05).sum()} significant, "
      f"{(survivors < 0.05 / len(survivors)).sum()} survive Bonferroni")
```

La correction ne détruit pas les constats réels — En R

```r
# Same loop, different comparison. The generator made this one real.
```

La correction ne détruit pas les constats réels

Comparaison	Tests	Significatifs à 0,05	Survivent à Bonferroni
Retard d'âge selon le **sexe** (aucun effet réel)	24	2	**0**
Retard d'âge selon le **redoublement de 2023** (réel)	14	14	**9**

La correction ne détruit pas les constats réels

- L'écart de 94,5 % contre 30,6 % du cours d'éducation survit à la correction dans neuf des quatorze écoles où il peut...

Comptez les tests, y compris ceux que vous n'avez pas rapportés

- Les sous-groupes regardés puis abandonnés — Découper par district, ne rien voir, et découper par année ensuite fait…
- Les résultats essayés tour à tour — Tester le retard d'âge, puis la présence, puis l'achèvement contre le même…
- Les seuils déplacés — « Retard d'âge » à année + 2 plutôt qu'à année + 1 est un second test de la même idée
- L'analyse que quelqu'un d'autre a lancée sur les mêmes données — Rarement connue, et à demander quand un relecteur…

Comptez les tests, y compris ceux que vous n'avez pas rapportés — En Python

```
plan = {
    "outcomes": ["over_age"],
    "groupings": ["sex"],
    "subgroups": ["school_id"],
}
n_planned = 1 * 1 * 24
print(f"tests declared in the analysis plan: {n_planned}")
```

Comptez les tests, y compris ceux que vous n'avez pas rapportés — En R

```
# Write the number down before running anything. It cannot be recovered after.
```

Comptez les tests, y compris ceux que vous n'avez pas rapportés

- Déclarez la famille avant de la lancer — Après coup, le nombre de tests est affaire de mémoire et d'intérêt personnel,. . .

Rapportez-le en entier — Exemple

```
Over-age enrolment by sex, school-level analysis

  24 schools tested (both sexes with at least 5 students in grades 1-6).
  2 schools significant at p < 0.05: SCH23 (p = 0.018), SCH06 (p = 0.036).

  With 24 tests, 1.2 significant results are expected by chance alone and
  the probability of at least one is 71%. Neither school survives the
  Bonferroni threshold of 0.0021, and neither is reported as a finding.

  For comparison, over-age by 2023 repetition is significant in all 14
  schools testable and survives Bonferroni in 9 of them.
```

Rapportez-le en entier

- La ligne de comparaison est ce qui rend le bloc convaincant plutôt que défensif — Elle montre la même procédure. . .

La suite

- Tous les tests jusqu'ici ont supposé qu'une ligne était une observation indépendante.

Lire la leçon complète, avec le code exécutable

https://data-analysis.cassion.dev/fr/cours/applied-statistics-for-programmes/stat-05-twenty-four-tests