

## Trente-huit cas, pas un rapport de cotes

Un seul rapport de cotes de 0,39 produit un écart de 22,7 points là où l'aboutissement est fréquent et de 15,2 points là où il est rare. La probabilité prédite est ce qu'un modèle logistique affirme réellement, et multipliée par la charge de cas elle devient 38 cas qui n'ont pas atteint un service.

---

Pierre Robentz Cassion

Leçon 4 sur 8

Régression appliquée aux données de programme — Des résultats en oui ou non

## Ce que couvre cette leçon

- Reconvertissez le modèle en probabilités
- Pourquoi un rapport de cotes fait plusieurs différences de risques
- Un intervalle sur l'effet marginal
- Multipliez-le par la charge de cas
- Quatre façons de se tromper
- Rapportez-le en entier
- La suite

## Reconvertissez le modèle en probabilités — En Python

```
import pandas as pd
import numpy as np
import statsmodels.formula.api as smf

model = smf.logit("completed ~ disability + case_category + age_band + sex"
                  " + service_requested + admin1", data=d).fit(dis=0)

everyone_with = d.assign(disability=1)
everyone_without = d.assign(disability=0)

p1 = model.predict(everyone_with).mean()
p0 = model.predict(everyone_without).mean()
print(f"{p1:.4f} vs {p0:.4f}    average marginal effect {p1 - p0:+.4f}")
```

## Reconvertissez le modèle en probabilités — En R

```
library(marginaleffects)  
  
avg_comparisons(model, variables = "disability")
```

## Reconvertissez le modèle en probabilités

- 26,11 % contre 45,52 %, une différence de  $-19,41$  points
- Comparez-la à l'écart brut de la leçon précédente :  $-19,05$  points — Le rapport de cotes ajusté est passé de 0,430 à . .

## Pourquoi un rapport de cotes fait plusieurs différences de risques — En Python

```
profile = dict(case_category="child-protection", age_band="0-11", sex="f",
               admin1="Artibonite")

rows = []
for service in ["health", "psychosocial", "legal", "livelihood-support"]:
    with_d = pd.DataFrame([**profile, "service_requested": service, "disability": 1])
    without = pd.DataFrame([**profile, "service_requested": service, "disability": 0])
    rows.append((service, model.predict(without)[0], model.predict(with_d)[0]))

print(pd.DataFrame(rows, columns=["service", "no disability", "disability"]).round(3))
```

## Pourquoi un rapport de cotes fait plusieurs différences de risques — En R

```
predictions(model, newdata = datagrid(service_requested = unique(d$service_requested),  
                                         disability = 0:1))
```

## Pourquoi un rapport de cotes fait plusieurs différences de risques

| Service demandé                   | Sans handicap | Handicap signalé | Écart     |
|-----------------------------------|---------------|------------------|-----------|
| Santé                             | 69,2 %        | 46,5 %           | –22,7 pts |
| Psychosocial                      | 64,5 %        | 41,3 %           | –23,2 pts |
| Juridique                         | 41,0 %        | 21,2 %           | –19,8 pts |
| Soutien aux moyens<br>d'existence | 28,7 %        | 13,5 %           | –15,2 pts |

## Pourquoi un rapport de cotes fait plusieurs différences de risques

- Le rapport de cotes vaut 0,39 sur chacune de ces lignes — Le modèle a été ajusté avec un seul coefficient de handicap, . . .
- Un rapport de cotes constant n'est pas un effet constant — Là où un résultat est déjà fréquent, le même rapport de . . .

# Un intervalle sur l'effet marginal — En Python

```
# statsmodels: delta method  
margins = model.get_margeff(at="overall")  
print(margins.summary())
```

## Un intervalle sur l'effet marginal — En R

```
avg_comparisons(model, variables = "disability")  # delta method, with CI
```

## Un intervalle sur l'effet marginal — En Python

```
# bootstrap, when the delta method is awkward or the estimator is custom
rng = np.random.default_rng(20260729)
draws = []
for _ in range(400):
    sample = d.sample(len(d), replace=True, random_state=int(rng.integers(1e9)))
    m = smf.logit(model.model.formula, data=sample).fit(dis=0)
    draws.append(m.predict(sample.assign(disability=1)).mean()
                  - m.predict(sample.assign(disability=0)).mean())
print(np.percentile(draws, [2.5, 97.5]).round(4))
```

## Un intervalle sur l'effet marginal — En R

*# 400 resamples is enough for a reportable interval; 2,000 for a published one.*

## Un intervalle sur l'effet marginal

- $-19,41$  points, IC à 95 %  $-26,1$  à  $-13,2$  — sur 400 rééchantillonnages bootstrap
- Fixez la graine et dites combien de rééchantillonnages — Un intervalle bootstrap qu'on ne peut pas reproduire n'est pas. . .

## Multipliez-le par la charge de cas — En Python

```
n_disability = int((d["disability"] == 1).sum())
comparator = d[d["disability"] == 0]["completed"].mean()
observed = d[d["disability"] == 1]["completed"].sum()
print(f"{n_disability} cases reporting a disability")
print(f"expected at the comparator rate: {comparator * n_disability:.0f}")
print(f"observed: {int(observed)}  shortfall: {comparator * n_disability - observed:.0f}")
```

## Multipliez-le par la charge de cas — En R

*# Three lines, and it is the only line of the analysis a manager will quote.*

## Multipliez-le par la charge de cas

- 197 cas signalant un handicap. 90 auraient abouti au taux de comparaison ; 52 l'ont fait. Trente-huit cas de moins
- Énoncez l'arithmétique, non le seul résultat — Un lecteur qui voit  $197 \times 45,5 \%$  — 52 peut vérifier ; un lecteur à qui. . .

## Quatre façons de se tromper

- Prédire à la moyenne au lieu de moyenner les prédictions — Fixer chaque covariable à sa moyenne et prédire une fois. . .
- Rapporter un effet marginal issu d'un modèle avec interaction sans dire où — Si le modèle laisse l'effet du handicap. . .
- Prendre le déficit pour un effet — Trente-huit cas est ce à quoi correspond l'écart observé, non ce que rapporterait sa. . .
- Extrapoler au-delà des données — Le modèle prédira volontiers une probabilité d'aboutissement pour un cas juridique de. . .

## Rapportez-le en entier — Exemple (suite)

Referral completion by disability status, adjusted

Logistic model, 1,581 consenting cases with complete covariates.

Adjusted for case category, age band, sex, service requested, department.

Average marginal effect    -19.4 points    95% CI -26.1 to -13.2  
(400 bootstrap resamples, seed 20260729)

Predicted completion        26.1% with a disability reported  
                                  45.5% without

Applied to the 197 cases reporting a disability, the gap is 38 cases that did not reach a service and would have at the comparator rate.

The gap is not constant: it is 22.7 points for health referrals, where completion is common, and 15.2 points for livelihood support, where it is not. The odds ratio (0.39) is the same on both.

## Rapportez-le en entier — Exemple (suite)

Observational. Cases were not randomised, and the model adjusts only for what the register records.

- L'avant-dernier paragraphe est celui qui empêche de surappliquer la moyenne —  
Un nombre unique est ce qui sera cité ; . . .

- Chaque covariable jusqu'ici a été choisie parce qu'elle paraissait pertinente.

Lire la leçon complète, avec le code exécutable

[https://data-analysis.cassion.dev/fr/cours/regression-for-programmes/  
reg-04-report-it-as-a-probability](https://data-analysis.cassion.dev/fr/cours/regression-for-programmes/reg-04-report-it-as-a-probability)