Thirty-eight cases, not an odds ratio

One odds ratio of 0.39 produces a 22.7-point gap where completion is common and a 15.2-point gap where it is rare. The predicted probability is what a logistic model actually claims, and multiplied by the caseload it becomes 38 cases that did not reach a service.

Pierre Robentz Cassion

What this lesson covers

- Turn the model back into probabilities
- Why one odds ratio is several risk differences
- An interval on the marginal effect
- Multiply it by the caseload
- Four ways this goes wrong
- Report it whole
- What comes next

Turn the model back into probabilities — In 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(disp=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}")
```

Turn the model back into probabilities — In R

```r
library(marginaleffects)

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

Turn the model back into probabilities

- 26.11% against 45.52%, a difference of -19.41 points
- Compare it with the crude difference from the last lesson: -19.05 points — The adjusted odds ratio moved from 0.430 to...

Why one odds ratio is several risk differences — In 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))
```

Why one odds ratio is several risk differences — In R

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

Why one odds ratio is several risk differences

Service requested	No disability	Disability reported	Gap
Health	69.2%	46.5%	**−22.7 pts**
Psychosocial	64.5%	41.3%	−23.2 pts
Legal	41.0%	21.2%	−19.8 pts
Livelihood support	28.7%	13.5%	**−15.2 pts**

Why one odds ratio is several risk differences

- The odds ratio is 0.39 on every one of those rows — The model was fitted with a single disability coefficient, so by...
- A constant odds ratio is not a constant effect — Where an outcome is already common, the same odds ratio moves more...

An interval on the marginal effect — In Python

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

An interval on the marginal effect — In R

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

An interval on the marginal effect — In 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(disp=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))
```

An interval on the marginal effect — In R

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

An interval on the marginal effect

- -19.41 points, 95% CI -26.1 to -13.2 — over 400 bootstrap resamples
- Set the seed and say how many resamples — A bootstrap interval that cannot be reproduced is not an interval, and this...

Multiply it by the caseload — In 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}")
```

Multiply it by the caseload — In R

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

Multiply it by the caseload

- 197 cases reporting a disability. 90 would have completed at the comparator rate; 52 did. Thirty-eight cases short
- State the arithmetic, not just the result — A reader who can see $197 \times 45.5\% - 52$ can check it; a reader given only "38…

Four ways this goes wrong

- Predicting at the mean instead of averaging the predictions — Setting every covariate to its mean and predicting once…
- Reporting a marginal effect from a model with an interaction, without saying where — If the model lets the disability…
- Treating the shortfall as an effect — Thirty-eight cases is what the observed gap corresponds to, not what closing it…
- Extrapolating past the data — The model will happily predict a completion probability for an 80-year-old legal case in…

Report it whole — Example (cont.)

```
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.
```

Report it whole — Example (cont.)

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

Report it whole

- The second-to-last paragraph is the one that stops the average being over-applied — A single number is what gets...

What comes next

- Every covariate so far has been chosen because it looked relevant.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/regression-for-programmes/reg-04-report-it-as-a-probability