The covariate that improves every statistic and destroys the answer

Add whether a referral was made and AIC falls by 595, pseudo-R^2 quadruples, and 42.5% of the disability gap disappears. Every model-selection criterion says keep it. Every causal consideration says it must come out, and no fit statistic will ever tell you which.

Pierre Robentz Cassion

Lesson 5 of 8

Regression for Programme Data — Which covariates belong

What this lesson covers

- A covariate that looks obviously right
- Why it removes the effect
- Four kinds of covariate, and what each does
- Drawing it before fitting
- The collider, which is worse
- Report both, and say which is which
- What comes next

A covariate that looks obviously right — In Python

```
import statsmodels.formula.api as smf

BASE = ("completed ~ disability + case_category + age_band + sex"
        " + service_requested + admin1")

adjusted = smf.logit(BASE, data=d).fit(disp=0)
plus_gate = smf.logit(BASE + " + referral_made", data=d).fit(disp=0)

for name, m in [("adjusted", adjusted), ("+ referral_made", plus_gate)]:
    print(f"{name:16} AIC {m.aic:7.1f}   pseudo-R2 {m.prsquared:.3f}")
```

A covariate that looks obviously right — In R

```
adjusted  <- glm(completed ~ disability + case_category + age_band + sex +
                   service_requested + admin1, data = d, family = binomial())
plus_gate <- update(adjusted, . ~ . + referral_made)
AIC(adjusted, plus_gate)
```

A covariate that looks obviously right

Model	Odds ratio	Average marginal effect	AIC	Pseudo-R²
Crude	0.430	−19.05 pts	2138.5	0.012
Adjusted	0.388	**−19.41 pts**	1999.6	0.089
+ `referral_made`	0.494	**−11.15 pts**	**1404.2**	**0.365**

A covariate that looks obviously right

- AIC improves by 595 points. Pseudo-R^2 quadruples. And 42.5% of the effect vanishes

Why it removes the effect — In Python

```
print(d.groupby("disability")["referral_made"].agg(["mean", "size"]).round(4))
```

Why it removes the effect — In R

```
d |> summarise(made = mean(referral_made), n = n(), .by = disability)
```

Why it removes the effect

- A referral is made for 71.5% of cases with no disability reported and 53.8% of cases with one
- Conditioning on `referral_made` asks: among cases where a referral was made, is there still a gap? — The answer is yes,. . .
- A covariate on the causal path between exposure and outcome is a mediator, and adjusting for it removes exactly the part of the effect that runs through it

Four kinds of covariate, and what each does

Kind	Sits	Adjust?	If you get it wrong
Confounder	Causes both exposure and outcome	**Yes**	Biased estimate, either direction
Mediator	Between exposure and outcome	**No** (for a total effect)	Effect understated
Collider	Caused by both exposure and outcome	**Never**	Bias created where none existed
Competing cause	Causes only the outcome	Optional	Precision only

Four kinds of covariate, and what each does

- Only the first row is a reason to add a variable — The fourth is a reason you may add one, and it buys a smaller…
- A variable's kind is not visible in the data — Confounder, mediator and collider all produce a coefficient that moves…

Drawing it before fitting — Example

```
disability -------------------------------> completion
    |                                            ^
    +-------> referral made --------------------+

department --> disability        department --> completion
```

Drawing it before fitting

- Department is a confounder — where a case arises affects both who is registered with a disability and how well the...

Drawing it before fitting — In Python

```
TOTAL_EFFECT = ["disability", "case_category", "age_band", "sex",
                "service_requested", "admin1"]        # confounders only
MECHANISM = TOTAL_EFFECT + ["referral_made"]          # a different question
```

Drawing it before fitting — In R

```
# Two named model specifications, two questions, both legitimate.
```

Drawing it before fitting

- Name the models after the question rather than after the variables — A model called `MECHANISM` will not accidentally...

The collider, which is worse — In Python

```
closed = d[d["case_status"].isin(["closed-resolved", "closed-lost-contact"])]
print(f"closed cases: {len(closed)}")
print(closed.groupby("disability")["completed"].agg(["mean", "size"]).round(4))
```

The collider, which is worse — In R

```
d |> filter(case_status %in% c("closed-resolved", "closed-lost-contact")) |>
  summarise(rate = mean(completed), n = n(), .by = disability)
```

The collider, which is worse

Sample	**n**	**Gap (average marginal effect)**
All consenting cases	1,581	−19.4 points
Closed cases only	**860**	**−21.1 points**

The collider, which is worse

- Restricting to closed cases moves the gap to 21.1 points — and the movement is produced by the restriction rather than…
- Selecting rows is adjusting for a variable — Dropping open cases, keeping only completed forms, analysing only…

Report both, and say which is which — Example

```
Referral completion by disability status, decomposition

  Total gap                                        -19.4 points
    of which runs through referral-making          -8.3 points  (42.5%)
    remaining, among referrals made                -11.2 points

  Referrals are made for 53.8% of cases reporting a disability against 71.5%
  of others. Both gates contribute; neither explains the other away.

  The -11.2 figure is conditional on a referral having been made and must not
  be reported as the effect of disability on reaching a service.
```

Report both, and say which is which

- The last line is the one that keeps the decomposition honest — Both numbers are real, they answer different questions,. . .

What comes next

- Every model in this course so far has assumed each row is an independent observation.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/regression-for-programmes/
reg-05-the-covariate-that-hides-the-answer