Adjustment moves the estimate; clustering moves the standard error

Adding every child-level covariate in the register moves the feeding coefficient from 4.47 to 4.37 and leaves the standard error at 0.9. Adding a term for the school leaves the coefficient alone and moves the standard error to 1.5. These are two different problems and they are routinely confused.

Pierre Robentz Cassion

Lesson 2 of 8

Regression for Programme Data — A model is a comparison

What this lesson covers

- The one table this course is built around
- What "adjusting for" actually does
- Why the covariates barely moved anything
- Reading a coefficient table without being misled
- When adjustment cannot help
- Report it whole
- What comes next

The one table this course is built around — In Python (cont.)

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

attendance = pd.read_csv("school-attendance-2024.v1.csv")
roster = pd.read_csv("school-roster-2024.v1.csv").drop_duplicates(
    "student_id", keep="last")
MARKS = {"true": True, "Y": True, "false": False, "N": False}
marked = attendance[attendance["present"].isin(MARKS)].copy()
marked["attended"] = marked["present"].map(MARKS)
per_student = (marked.groupby("student_id")["attended"].mean()
               .rename("rate").reset_index().merge(roster, on="student_id"))

enrolment = pd.read_csv("school-enrolment-2024.v1.csv")
current = enrolment[enrolment["school_year"] == 2024]

d = per_student.merge(
```

The one table this course is built around — In Python (cont.)

```
    current[["student_id", "sex", "age_years", "disability_reported",
             "displacement_status", "admin2"]], on="student_id").dropna()
print(f"complete cases: {len(d)}")

m1 = smf.ols("rate ~ feeding_programme", data=d).fit()
m2 = smf.ols("rate ~ feeding_programme + sex + age_years"
             " + disability_reported + displacement_status", data=d).fit()
m3 = smf.ols("rate ~ feeding_programme + sex + age_years"
             " + disability_reported + displacement_status + admin2", data=d).fit()
m4 = m1.get_robustcov_results(cov_type="cluster", groups=d["school_id"])

for name, m in [("child covariates", m2), ("+ district", m3)]:
    print(f"{name:18} {m.params['feeding_programme[T.True]']:+.4f}")
```

The one table this course is built around — In R

```
library(dplyr)

m1 <- lm(rate ~ feeding_programme, data = d)
m2 <- lm(rate ~ feeding_programme + sex + age_years +
           disability_reported + displacement_status, data = d)
m3 <- update(m2, . ~ . + admin2)
```

The one table this course is built around

Model	Feeding coefficient	SE	t
M1 one predictor	**+0.0447**	0.0089	5.02
M2 + sex, age, disability, displacement	**+0.0437**	0.0090	4.85
M3 + district	**+0.0360**	0.0100	3.60
M4 = M1 with school-clustered SEs	**+0.0447**	**0.0154**	**2.90**

The one table this course is built around

- Adding covariates changed the coefficient and left the standard error alone — M1 to M3 moves the estimate from 4.5...
- Clustering changed the standard error and left the coefficient untouched — M4 is the *same fitted line* as M1 —...
- These are two different problems — One is about whether the comparison is between comparable groups; the other is about...

What "adjusting for" actually does

- A coefficient with covariates in the model is the comparison within levels of those covariates — pooled across them

What "adjusting for" actually does — In Python

```
within = (d.groupby("admin2")
          .apply(lambda g: g[g["feeding_programme"]]["rate"].mean()
                 - g[~g["feeding_programme"]]["rate"].mean()))
print(within.round(4))
print(f"unadjusted: {m1.params['feeding_programme[T.True]']:.4f}")
print(f"adjusted:   {m3.params['feeding_programme[T.True]']:.4f}")
```

What "adjusting for" actually does — In R

```
d |> summarise(gap = mean(rate[feeding_programme]) -
                     mean(rate[!feeding_programme]), .by = admin2)
```

What "adjusting for" actually does

District	Gap	Fed	Unfed
Artibonite	+3.8 pts	146	141
Centre	**−4.1 pts**	244	**42**
Nord-Ouest	+4.9 pts	251	45
Sud	+7.0 pts	104	178

What "adjusting for" actually does

- The adjusted coefficient is a weighted average of those four gaps — and seeing them separately is worth the two lines...
- A single coefficient averaged over a disagreement is still a number, and it is no longer a summary — Before reporting...

Why the covariates barely moved anything

- The feeding programme was assigned by school — Nothing about a *child* decided whether they got school meals, so child...
- District did move it — from 4.5 to 3.6, and for the same reason in reverse: schools with a feeding programme are not...
- The rule the design implies: confounders live at the level of assignment — For a school-level programme, look for...

Reading a coefficient table without being misled

- Report the sample size of the model, not of the file — Complete-case fitting dropped 49 students here — 1,200 rows...

Reading a coefficient table without being misled — In Python

```
print(f"file {len(per_student)}, model {int(m3.nobs)}")
```

Reading a coefficient table without being misled — In R

```
c(file = nrow(per_student), model = nobs(m3))
```

Reading a coefficient table without being misled

- Do not read the covariate coefficients as findings — They are adjusted for a set of variables chosen to answer a...
- Check whether an added covariate changed the sample — A covariate with missing values silently drops rows, so a...
- Report the unadjusted estimate too — The pair is what shows the reader how much work the adjustment did, and a report...

When adjustment cannot help

- The confounder is not in the file — Adjusting for what you measured says nothing about what you did not, and "adjusted"...
- The covariate is on the causal path — Lesson 5 shows a covariate that removes 42% of the effect by sitting between...
- The problem is the standard error, not the estimate — No covariate fixes clustering

Report it whole — Example (cont.)

```
Attendance and school feeding, adjusted analysis

  Linear model, 1,151 students with complete covariates (49 dropped).

  Unadjusted                      +4.47 points   95% CI +2.73 to +6.22
  Adjusted for district           +3.60 points   95% CI +1.64 to +5.56
  Child-level covariates (sex, age, disability, displacement) change the
  estimate by less than 0.1 points and are retained only to show that.

  The district gaps are +3.8, -4.1, +4.9 and +7.0 points. Centre runs the
  other way on 42 unfed students and is not averaged away silently.

  Standard errors above assume independent students and are too small; the
  school-clustered version of the unadjusted model gives +4.47 (95% CI
  +1.45 to +7.49).
```

Report it whole — Example (cont.)

```
Observational. Schools were not randomised into the programme, and
district adjustment does not make the remaining comparison causal.
```

Report it whole

- Both estimates, and the clustered interval, in six lines — A reader who sees only the adjusted number cannot tell...

What comes next

- Everything so far has had a continuous outcome.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/regression-for-programmes/reg-02-what-adjustment-moves