Three honest answers and one that is out on its own

Naive OLS, cluster-robust standard errors, a random intercept and school-level aggregation give standard errors of 0.88, 1.46, 1.48 and 1.53 points. Three of those agree and the fourth is the one every default fit produces.

Pierre Robentz Cassion

Lesson 6 of 8

What this lesson covers

- The same coefficient, four standard errors
- What a random intercept adds
- Choosing among the three
- Predictors at two levels
- Three levels, and when to stop
- Report it whole
- What comes next

The same coefficient, four standard errors — In Python

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

naive = smf.ols("rate ~ feeding_programme", data=per_student).fit()
robust = naive.get_robustcov_results(cov_type="cluster",
                                     groups=per_student["school_id"])

mixed = smf.mixedlm("rate ~ feeding_programme", data=per_student,
                    groups=per_student["school_id"]).fit()

by_school = (per_student.groupby(["school_id", "feeding_programme"])["rate"]
             .mean().reset_index())
aggregated = smf.ols("rate ~ feeding_programme", data=by_school).fit()
```

The same coefficient, four standard errors — In R

```
library(lme4); library(sandwich); library(lmtest)

naive      <- lm(rate ~ feeding_programme, data = per_student)
robust     <- coeftest(naive, vcov = vcovCL, cluster = ~school_id)
mixed      <- lmer(rate ~ feeding_programme + (1 | school_id), data = per_student)
aggregated <- lm(rate ~ feeding_programme, data = by_school)
```

The same coefficient, four standard errors

Approach	Coefficient	SE	t	n
Naive OLS	+4.93 pts	**0.88**	5.63	1,200
Cluster-robust SE	+4.93 pts	1.46	3.38	1,200
Random intercept	+5.10 pts	1.48	3.44	1,200
School-level OLS	+5.21 pts	1.53	3.41	24

The same coefficient, four standard errors

- Three of the four agree and one does not — The coefficients span 0.3 points; the three honest standard errors span...
- That is the shape to expect — Clustering rarely changes an estimate much and routinely changes its precision a lot, and...

What a random intercept adds — In Python

```
print(mixed.summary())
print(f"between-school variance: {float(mixed.cov_re.iloc[0, 0]):.5f}")
print(f"residual variance:       {mixed.scale:.5f}")
```

What a random intercept adds — In R

```
VarCorr(mixed)
```

What a random intercept adds

Model	Between-school variance	Residual variance	ICC
Intercept only	0.00142	0.02042	**0.065**
+ **feeding programme**	**0.00081**	0.02042	0.038

What a random intercept adds

- Feeding explains 42.8% of the between-school variance and none of the within-school variance — which is exactly what a...

Choosing among the three

Use	When	Cost
Aggregate to the cluster	Few clusters; the exposure is cluster-level	A big school counts the same as a small one
Cluster-robust SE	Many clusters (30+); you want the individual-level model	Unreliable below about 30 clusters
Random intercept	You want the variance components, or predictors at both levels	Assumes the random effect is uncorrelated with the predictors

Choosing among the three

- With 24 clusters, aggregate or fit the random intercept and say which — The cluster-robust sandwich is the standard...
- When the exposure varies only between clusters, all three converge — which the table above shows

Predictors at two levels — In Python

```
two_level = smf.mixedlm(
    "rate ~ feeding_programme + age_years + disability_reported",
    data=d, groups=d["school_id"]).fit()
print(two_level.summary().tables[1])
```

Predictors at two levels — In R

```
lmer(rate ~ feeding_programme + age_years + disability_reported +
       (1 | school_id), data = d)
```

Predictors at two levels

- `feeding_programme` varies only between schools. `age_years` and `disability_reported` vary within them — A single...
- The failure mode is putting a cluster-level exposure in a model with no cluster term — That is the naive row of the...

Three levels, and when to stop — In Python

```
# Two grouping levels: households nested within enumeration areas.
smf.mixedlm("outcome ~ x", data=survey,
            groups=survey["ea_id"], re_formula="1").fit()
```

Three levels, and when to stop — In R

```
lmer(outcome ~ x + (1 | ea_id / household_id), data = survey)
```

Three levels, and when to stop

- Add a level when something is assigned or measured at that level and you have enough units of it — Three levels with...
- Do not add a level for tidiness — A level with four groups adds a parameter that cannot be estimated and a false sense...

Report it whole — Example (cont.)

```
Attendance and school feeding, multilevel analysis

  Linear mixed model, 1,200 students in 24 schools.
  rate ~ feeding_programme + (1 | school_id)

  Feeding programme    +5.10 points   95% CI +2.20 to +8.00

  Between-school SD    0.028      Residual SD  0.143
  ICC 0.038 with the programme term; 0.065 without it, so the programme
  accounts for 43% of the between-school variance.

  The naive single-level model gives +4.93 points with a standard error of
  0.88 rather than 1.48. It is not reported: it treats 50 children in one
  school as 50 independent observations.

  Cluster-robust standard errors on the single-level model (+4.93, SE 1.46)
```

Report it whole — Example (cont.)

```
and school-level aggregation (+5.21, SE 1.53) give the same answer. With
24 clusters the robust sandwich is at the edge of its assumptions and is
reported as a check rather than as the headline.

Observational. Schools were not randomised into the programme.
```

Report it whole

- Reporting all three is four lines and it forecloses the obvious challenge — that the result depends on the method

What comes next

- This lesson matched the model to how the *programme* was assigned.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/regression-for-programmes/reg-06-a-term-for-the-school