The coefficient is the difference

Fit attendance against a single feeding dummy. The intercept is 85.21% — the mean of schools with no programme. The coefficient is 4.94 points — the difference between the two means, to four decimal places. A regression with one dummy is a two-group comparison and nothing more.

Pierre Robentz Cassion

What this lesson covers

- The same comparison, written twice
- What every term means
- Where the two routes differ, and by how much
- Three predictors, three readings
- What the model does not say
- Report it whole
- What comes next

The same comparison, written twice — In Python (cont.)

```python
import pandas as pd
import numpy as np
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")          # two never-de-registered transfers

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"))
```

The same comparison, written twice — In Python (cont.)

```
model = smf.ols("rate ~ feeding_programme", data=per_student).fit()
print(model.summary().tables[1])
```

The same comparison, written twice — In R

```r
library(dplyr)

per_student |> lm(rate ~ feeding_programme, data = _) |> summary()
```

The same comparison, written twice

Term	Coefficient	SE	t
Intercept	**0.8521**	0.0070	121.30
`feeding_programme[T.milk]`	**0.0494**	0.0088	5.63

The same comparison, written twice — In Python

```
means = per_student.groupby("feeding_programme")["rate"].mean()
print(means)
print(f"difference: {means[True] - means[False]:.4f}")
```

The same comparison, written twice — In R

```
per_student |> summarise(rate = mean(rate), .by = feeding_programme)
```

The same comparison, written twice

- Mean without a programme: 0.8521. Mean with: 0.9015. Difference: 0.0494

What every term means

- The intercept is the fitted value when every predictor is zero — Here that is `feeding_programme = False`, so the...
- A coefficient is the change in the outcome for a one-unit change in that predictor, with the others held fixed — For a...
- A standard error is the same standard error the last course computed — The `t` column is coefficient over standard...

What every term means — In Python

```
coef = model.params["feeding_programme[T.True]"]
se = model.bse["feeding_programme[T.True]"]
print(f"{coef:+.4f}  95% CI [{coef - 1.96*se:+.4f}, {coef + 1.96*se:+.4f}]")
```

What every term means — In R

```
confint(model)
```

What every term means

- +0.0494, 95% CI +0.0322 to +0.0665 — the same interval, arrived at from the same data by a different route

Where the two routes differ, and by how much

- Ordinary least squares assumes one residual variance for both groups — It is the pooled-variance t-test written in...

Where the two routes differ, and by how much — In Python

```
from scipy import stats

fed = per_student[per_student["feeding_programme"]]["rate"]
unfed = per_student[~per_student["feeding_programme"]]["rate"]
print(f"variances: {fed.var():.5f} and {unfed.var():.5f}")
print("pooled t:", stats.ttest_ind(fed, unfed).statistic.round(2))
print("Welch  t:", stats.ttest_ind(fed, unfed, equal_var=False).statistic.round(2))
```

Where the two routes differ, and by how much — In R

```
t.test(rate ~ feeding_programme, data = per_student, var.equal = TRUE)
t.test(rate ~ feeding_programme, data = per_student)
```

Where the two routes differ, and by how much

- Regression reproduces the pooled t exactly — When the group variances are close the two barely differ; here they are. . .

Three predictors, three readings

Predictor	Coefficient reads as
Binary (`feeding_programme`)	The difference between the two groups
Continuous (`age_years`)	The change per one year
Categorical with k levels	k−1 coefficients, each a difference against the omitted level

Three predictors, three readings — In Python

```
enrolment = pd.read_csv("school-enrolment-2024.v1.csv")
current = enrolment[enrolment["school_year"] == 2024]
joined = per_student.merge(
    current[["student_id", "sex", "age_years", "admin2"]], on="student_id")

print(smf.ols("rate ~ admin2", data=joined).fit().params.round(4))
print(joined.groupby("admin2")["rate"].mean().round(4))
```

Three predictors, three readings — In R

```
lm(rate ~ admin2, data = joined) |> coef()
joined |> summarise(rate = mean(rate), .by = admin2)
```

Three predictors, three readings

- A categorical predictor produces one coefficient per level except one — and every coefficient is a comparison against...
- Say the reference level in the table caption — "Against Artibonite" is four words and it is the difference between a...

What the model does not say

- It is not an effect — The 4.9-point coefficient is the difference between two sets of schools that were not randomised...
- It is not a prediction worth making — Regression is taught elsewhere as a prediction machine, and for programme data...

Report it whole — Example

```
Attendance and school feeding, February to April 2024

  Linear model, one predictor, 1,200 students.

  Intercept (no programme)   85.21%
  Feeding programme          +4.94 points   95% CI +3.22 to +6.65

  The coefficient is the difference between the two group means and is
  reported as such. Schools were not randomised into the programme; the
  comparison is observational.

  The standard error assumes independent students. It is not, and lesson 6
  gives the corrected version.
```

Report it whole

- The last line is a promissory note the course pays off — and writing it is better than not knowing it is owed

What comes next

- One predictor reproduces a comparison you could have made without a model.

Read the full lesson, with runnable code

https://data-analysis.cassion.dev/courses/regression-for-programmes/reg-01-the-coefficient-is-the-difference