

Panel regressions in Python

Programming & Data Management, Part II · Further topic F1
(optional, not examined)

Magnus Lodefalk
Örebro University

NA446A, Örebro University
Autumn 2026



Today's goals

By the end of today you can:

1. Estimate **pooled OLS, person fixed effects and two-way fixed effects** with `pyfixest`, the Python version of `reghdfe`
2. Choose **standard errors clustered by person**, and say why the default ones are too small
3. Put the three models in **one table** with `pf.etable`

Six terms for this module

panel key	person and year; together unique (<code>isid nr year</code>)
pooled OLS	one regression on all person-years, as if they were unrelated people
fixed effects	each person gets an own intercept; only changes <i>within</i> a person identify the slopes (<code>absorb(nr)</code>)
two-way FE	person and year intercepts: common shocks in a year are removed too (<code>absorb(nr year)</code>)
clustered SE	standard errors that allow a person's errors to be related over years
absorbed	a variable the fixed effects already contain: it cannot be estimated

- Package: `pyfixest`, the one the lab's papers use. Its formula `y ~ x | nr+year` reads like `reghdfe y x, absorb(nr year)`.

Add it once in your course folder: `uv add pyfixest`.

Three models

Step 1: load the panel and check its key

```
df = pd.read_csv(HERE / "data" / "wage_panel.csv")
print(df.shape, df["nr"].nunique())
print(df.groupby("nr").size().unique())
print(df.duplicated(["nr", "year"]).sum())
```

```
(4360, 12) 545
```

```
[8]
```

```
0
```

Vella and Verbeek's panel of 545 young American men, 1980–87, eight years each (balanced), saved in the folder.

- No xtset needed: pyfixest is told the person and year columns inside each formula.

Stata: import delimited using "data/wage_panel.csv", then isid nr year.

Step 2: pooled OLS

```
formula = ("lwage ~ educ + black + hisp"  
           " + exper + expersq + married + union")  
pooled_default = pf.feols(formula, data=df,  
                           vcov="iid")  
print(pooled_default.tidy().round(3))
```

	Estimate	Std. Error	t value
educ	0.099	0.005	21.248
union	0.180	0.017	10.518

- pyfixest adds the intercept itself, as Stata does. A year of schooling goes with 0.10 log points (about 10 per cent) higher wages, union membership with 0.18. Selected rows and columns.

Stata: `reghdfe lwage `x', noabsorb` (or `reg lwage `x'`): same numbers.

Step 3: three kinds of standard errors, same coefficients

```
pooled_robust = pf.feols(formula, data=df,  
                          vcov="hetero")  
pooled_cluster = pf.feols(formula, data=df,  
                           vcov={"CRV1": "nr"})
```

	coef	se_default	se_robust	se_cluster
educ	0.099	0.005	0.005	0.009
union	0.180	0.017	0.016	0.028

- Clustered: 1.2 to 2.1 times the default. 4,360 rows are not 4,360 independent people: a man's luck repeats every year.

Stata: `vce(robust)`, `vce(cluster nr)`: the same digits.

Check your understanding

Answer on your own first, then compare with a neighbour

Clustered standard errors here are up to 2.1 times the default ones. Predict: if the same 545 men were followed for 16 years instead of 8, would that ratio grow or shrink? Why?

Check your understanding

Answer on your own first, then compare with a neighbour

Clustered standard errors here are up to 2.1 times the default ones. Predict: if the same 545 men were followed for 16 years instead of 8, would that ratio grow or shrink? Why?

Answer: Grow. Each man's errors are related across his years, so every extra year of the same man adds less new information than a new man would. The default formula counts every row as independent; the more rows per cluster, the more it overstates the information in the data. What matters for precision is closer to the number of men (545) than to the number of rows.

Step 4: fixed effects for each person

```
fe_person = pf.feols(  
    "lwage ~ exper + expersq + married + union | nr",  
    data=df, vcov={"CRV1": "nr"})
```

largest within-person sd of educ: 0.0

	Estimate	Std. Error
exper	0.117	0.011
married	0.045	0.021
union	0.082	0.023

- Union: 0.18 pooled, 0.08 here, comparing the same man in and out of a union. Education never changes within a man: absorbed.

Stata: `reghdfe lwage exper expersq married union, absorb(nr) vce(cluster nr); xtreg ..., fe vce(cluster nr)` gives the same.

Step 5: two-way fixed effects, person and year

```
fe_twoway = pf.feols(  
    "lwage ~ expersq + married + union | nr+year",  
    data=df, vcov={"CRV1": "nr"})
```

	Estimate	Std. Error
expersq	-0.005	0.001
married	0.047	0.021
union	0.080	0.023

- Year effects remove what happened to everyone in a year (the business cycle). This is the lab's workhorse: AI Unboxed absorbs firm and industry-year effects.

Stata: `reghdfe lwage expersq married union, absorb(nr year) vce(cluster nr)`. Write `nr+year` without spaces (a pyfixest 0.60 label quirk).

Check your understanding

Answer on your own first, then compare with a neighbour

Step 5 left `exper` out. Predict: what happens if you put it back in (`"lwage ~ exper + expersq + married + union | nr+year"`), and why?

Check your understanding

Answer on your own first, then compare with a neighbour

Step 5 left `exper` out. Predict: what happens if you put it back in (`"lwage ~ exper + expersq + married + union | nr+year"`), and why?

Answer: It is absorbed. In these data every man's experience rises by exactly one each year, so within a man `exper` is the calendar year minus a constant: the person and year effects already contain it. `pyfixest` drops it with a warning ("dropped due to multicollinearity"), as `reghdfe` writes "omitted because of collinearity"; only `expersq` survives. Wage growth with experience and wage growth over calendar time cannot be told apart here.

Step 6: one table, three models

```
models = [pooled_cluster, fe_person, fe_twoway]
keep = ["educ", "exper", "married", "union"]
table = pf.etable(models, type="df", keep=keep,
    exact_match=True, signif_code=[0.01, 0.05, 0.1],
    coef_fmt="b:.3f* (se:.3f)")
```

		(1)	(2)	(3)
coef	exper	0.089*** (0.012)	0.117*** (0.011)	
	union	0.180*** (0.028)	0.082*** (0.023)	0.080*** (0.023)
fe	nr	-	x	x
	year	-	-	x

- Selected rows of `print(table.to_string()); type="tex"` gives the same table for a paper.

Stata: `esttab (SSC) or outreg2`. Stars at 10, 5, 1 per cent.

Summary, and where to go next

1. `pf.feols("y ~ x | nr+year", data=df, vcov={"CRV1": "nr"})` is `reghdfe y x, absorb(nr year) vce(cluster nr)`
2. Cluster by person (or firm): the default standard errors were only half to four fifths as large
3. Variables fixed within a person, or moving one-for-one with the year, are absorbed; `pf.etable` puts the models in one table

Read next (open, checked 30 Sep 2026):

- `pyfixest` (quickstart and examples in its README):
github.com/py-econometrics/pyfixest
- Turrell et al., *Coding for Economists*, "Regression" (`pyfixest`):
aeturrell.github.io/coding-for-economists/econmt-regression.html
- Next module: F4, difference-in-differences and event studies

Appendix: random effects and the Hausman test

```
panel = df.set_index(["nr", "year"])
re_model = RandomEffects.from_formula(
    "lwage ~ 1 + educ + black + hisp + exper"
    " + expersq + married + union", data=panel).fit()
```

```
union: RE 0.107 | FE 0.082
Hausman chi2(4) = 31.57 | p = 2e-06
```

pyfixest has no random effects, so this step uses `linearmodels` (optional: `uv add linearmodels`; the script skips it if missing). Write `1 +:` `linearmodels` adds no intercept. Hausman uses the *default* standard errors.

- The test rejects RE: use fixed effects. That is also the lab's default.

Stata: `xtreg `x', re`, then `hausman fe re`: 31.45 (the two programs estimate the variances slightly differently).