Week 5:
Panel Regression Recap

Dr Christian Engels
ce50@st-andrews.ac.uk

FI5699 Dissertation Module

Department of Finance
University of St Andrews Business School

Introduction

What We Will Cover Today

This session recaps the econometric foundations behind most empirical finance dissertations.

1. **Why Panel Data?** — the structure behind your dataset
2. **The Pooled OLS Problem** — what goes wrong without fixed effects
3. **Fixed Effects** — what they absorb and what they cannot
4. **Clustered Standard Errors** — getting inference right
5. **Running Regressions in Python** — `pyfixest` in practice
6. **Presenting Results** — publication-quality tables

By the end of today, you will be able to estimate, interpret, and present fixed effects regressions with clustered standard errors in Python.

Table of Contents

Recap: What You Built in Week 4

Concept	What you learned
Polars DataFrames	Creating, inspecting, and querying tables
Expressions	`select`, `with_columns`, `filter`, `group_by`
Stock returns	Downloading prices, computing `pct_change()`, `.over()`
Joins & reshaping	Combining tables, pivot/unpivot
Visualisation	`plotnine` — line charts, histograms, cumulative returns

Today we move from data manipulation to **statistical analysis** — the regressions that will form the core of your dissertation.

The Central Question for Your Dissertation

Most empirical finance dissertations ask a question of the form:

*"Does **X** affect **Y**, after controlling for things we can't directly observe?"*

Example	X	Y
Corporate finance	Employment protection laws	Innovation output
Asset pricing	ESG score	Stock returns
Banking	Capital requirements	Lending volume

The workhorse tool for answering these questions is the fixed effects panel regression with clustered standard errors.

Our Running Example: Bena, Ortiz-Molina & Simintzi (2022)

"Shielding Firm Value: Employment Protection and Process Innovation"
Journal of Financial Economics, 146(2), 637–664.

Feature	Value
Observations	45,263 firm-years
Firms	4,447
Period	1975–1997
Treatment	Employment protection laws
Outcome	Process innovation

Variable	Mean	SD
Process innov.	1.14	1.82
Product innov.	1.88	2.19
Patent stock	1.71	1.61
State-yr pat.	1.67	0.68

We will use this dataset throughout: first learning the theory, then running it in Python, then seeing real output.

Why Panel Data?

Panel Data Tracks Units Over Time

A panel dataset has two dimensions:

- **Cross-section** — many units (firms, countries, individuals)

firm	year	Y	X
AAPL	2020	3.2	0.8
AAPL	2021	4.1	1.2
AAPL	2022	2.9	0.6
MSFT	2020	5.7	1.5
MSFT	2021	6.3	1.8

within

This structure is what makes fixed effects possible — and what makes standard errors tricky.

Panel Data Tracks Units Over Time

A panel dataset has two dimensions:

- **Cross-section** — many units (firms, countries, individuals)
- **Time series** — each unit observed repeatedly over time

firm	year	Y	X
AAPL	2020	3.2	0.8
AAPL	2021	4.1	1.2
AAPL	2022	2.9	0.6
MSFT	2020	5.7	1.5
MSFT	2021	6.3	1.8

within

This structure is what makes fixed effects possible — and what makes standard errors tricky.

Panel Data Tracks Units Over Time

A panel dataset has two dimensions:

- **Cross-section** — many units (firms, countries, individuals)
- **Time series** — each unit observed repeatedly over time
- Each row is a **unit–time** pair (e.g. firm–year)

This structure is what makes fixed effects possible — and what makes standard errors tricky.

firm	year	Y	X
AAPL	2020	3.2	0.8
AAPL	2021	4.1	1.2
AAPL	2022	2.9	0.6
MSFT	2020	5.7	1.5
MSFT	2021	6.3	1.8

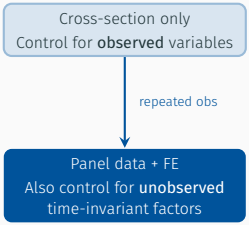
within

Panel Data Lets You Control for the Unobservable

With **cross-sectional** data alone, you can only control for what you observe.

With **panel data**, you can also control for things you *cannot* directly measure:

- Firm culture, management quality



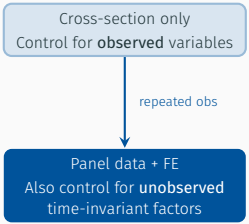
This is the power of fixed effects.

Panel Data Lets You Control for the Unobservable

With **cross-sectional** data alone, you can only control for what you observe.

With **panel data**, you can also control for things you *cannot* directly measure:

- Firm culture, management quality
- Country-level institutions



This is the power of fixed effects.

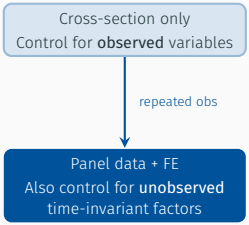
Panel Data Lets You Control for the Unobservable

With **cross-sectional** data alone, you can only control for what you observe.

With **panel data**, you can also control for things you *cannot* directly measure:

- Firm culture, management quality
- Country-level institutions
- Macroeconomic shocks hitting all firms in a given year

This is the power of fixed effects.



The Pooled OLS Problem

Pooled OLS Ignores Unobserved Heterogeneity

Consider estimating:

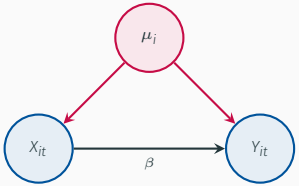
$$Y_{it} = \alpha + \beta X_{it} + \varepsilon_{it}$$

The problem: ε_{it} contains **everything else** that affects Y — including time-invariant characteristics of each firm (μ_i) that also correlate with X.

Example:

- Y = innovation output

If R&D culture correlates with both Y and X, your $\hat{\beta}$ is biased.



The red arrows create omitted variable bias.

Pooled OLS Ignores Unobserved Heterogeneity

Consider estimating:

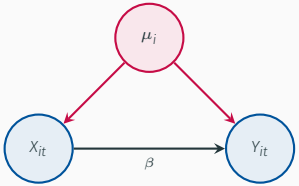
$$Y_{it} = \alpha + \beta X_{it} + \varepsilon_{it}$$

The problem: ε_{it} contains **everything else** that affects Y — including time-invariant characteristics of each firm (μ_i) that also correlate with X.

Example:

- Y = innovation output
- X = employment protection law

If R&D culture correlates with both Y and X, your $\hat{\beta}$ is biased.



The red arrows create omitted variable bias.

Pooled OLS Ignores Unobserved Heterogeneity

Consider estimating:

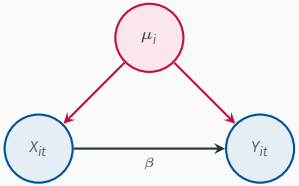
$$Y_{it} = \alpha + \beta X_{it} + \varepsilon_{it}$$

The problem: ε_{it} contains **everything else** that affects Y — including time-invariant characteristics of each firm (μ_i) that also correlate with X.

Example:

- Y = innovation output
- X = employment protection law
- μ_i = firm R&D culture (unobserved)

If R&D culture correlates with both Y and X, your $\hat{\beta}$ is biased.



The red arrows create omitted variable bias.

Fixed Effects

Fixed Effects Remove Time-Invariant Confounders

The fixed effects model adds a unit-specific intercept α_i:

$$Y_{it} = \alpha_i + \beta X_{it} + \varepsilon_{it}$$

This is equivalent to **demeaning** — subtracting each unit's time average:

$$(Y_{it} - \bar{Y}_i) = \beta(X_{it} - \bar{X}_i) + (\varepsilon_{it} - \bar{\varepsilon}_i)$$

Fixed effects exploit only **within-unit variation over time**. Anything constant within a unit (culture, geography, founding year) is absorbed by α_i and cannot bias $\hat{\beta}$.

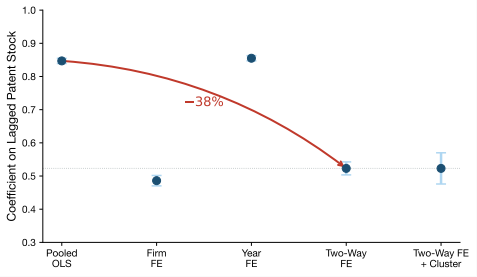
Common Fixed Effects in Finance

Fixed effect	Absorbs	Example
Firm (α_i)	Time-invariant firm traits	Management quality, founding conditions
Year (δ_t)	Common time shocks	Financial crises, regulatory changes
Firm + Year	Both simultaneously	The standard "two-way FE"
Industry $\times$ Year	Industry-specific trends	Tech boom affects tech firms differently
State $\times$ Year	Region-specific shocks	State-level policy changes

More fixed effects = more unobservables absorbed, but also **less remaining variation** to estimate β from. There is always a trade-off.

Firm FE cuts the patent stock coefficient nearly in half

	(1) OLS	(2) Firm	(3) Year
Patent stock	0.847***	0.486***	0.855***
	(0.004)	(0.008)	(0.004)
State-yr pat.	−0.019*	0.312***	−0.069***
	(0.008)	(0.019)	(0.008)
Firm FE		✓	
Year FE			✓
N	45,263	44,898	45,263
R^2	0.555	0.745	0.569



Once we compare each firm *to itself* over time, the estimate drops from 0.847 to 0.486 — much of the pooled estimate reflected cross-firm differences.

Fixed Effects Are Not a Magic Bullet

Imai & Kim (2019) and Mummolo & Peterson (2018) identify critical limitations:

FE assumes no dynamics

- Past outcomes do not cause current treatment

FE uses only within variation

If treatment and outcome feed back on each other over time, FE alone is insufficient.

Fixed Effects Are Not a Magic Bullet

Imai & Kim (2019) and Mummolo & Peterson (2018) identify critical limitations:

FE assumes no dynamics

- Past outcomes do not cause current treatment
- Past treatment does not affect current outcome

If treatment and outcome feed back on each other over time, FE alone is insufficient.

FE uses only within variation

Fixed Effects Are Not a Magic Bullet

Imai & Kim (2019) and Mummolo & Peterson (2018) identify critical limitations:

FE assumes no dynamics

- Past outcomes do not cause current treatment
- Past treatment does not affect current outcome
- Violations $\rightarrow$ inconsistent $\hat{\beta}$

If treatment and outcome feed back on each other over time, FE alone is insufficient.

FE uses only within variation

Fixed Effects Are Not a Magic Bullet

Imai & Kim (2019) and Mummolo & Peterson (2018) identify critical limitations:

FE assumes no dynamics

- Past outcomes do not cause current treatment
- Past treatment does not affect current outcome
- Violations $\rightarrow$ inconsistent $\hat{\beta}$

If treatment and outcome feed back on each other over time, FE alone is insufficient.

FE uses only within variation

- $\hat{\beta}$ comes from *within-unit changes*, not cross-sectional differences

Fixed Effects Are Not a Magic Bullet

Imai & Kim (2019) and Mummolo & Peterson (2018) identify critical limitations:

FE assumes no dynamics

- Past outcomes do not cause current treatment
- Past treatment does not affect current outcome
- Violations $\rightarrow$ inconsistent $\hat{\beta}$

If treatment and outcome feed back on each other over time, FE alone is insufficient.

FE uses only within variation

- $\hat{\beta}$ comes from *within-unit changes*, not cross-sectional differences
- Counterfactuals should reflect *plausible within-unit shifts*

Fixed Effects Are Not a Magic Bullet

Imai & Kim (2019) and Mummolo & Peterson (2018) identify critical limitations:

FE assumes no dynamics

- Past outcomes do not cause current treatment
- Past treatment does not affect current outcome
- Violations $\rightarrow$ inconsistent $\hat{\beta}$

If treatment and outcome feed back on each other over time, FE alone is insufficient.

FE uses only within variation

- $\hat{\beta}$ comes from *within-unit changes*, not cross-sectional differences
- Counterfactuals should reflect *plausible within-unit shifts*
- Many published papers overstate effect sizes by using overall (not within) variation

Interpreting FE Results Honestly

Mummolo & Peterson (2018) surveyed 54 studies using FE and found that most discussed counterfactuals far larger than any within-unit variation observed.

Their checklist for your dissertation:

1. **Residualize** X with respect to your fixed effects — this reveals the actual variation driving $\hat{\beta}$
2. **Report the within-unit standard deviation** of X, not the overall SD
3. **Use plausible counterfactuals** — a one-SD *within-unit* shift, not a min-to-max shift
4. **Report the share of units with zero within-unit variation** — these contribute nothing to estimation

If your counterfactual is 5× the within-unit SD, your effect size claim is implausible.

Clustered Standard Errors

OLS Standard Errors Assume Independence

OLS standard errors assume that ε_{it} and ε_{js} are uncorrelated for any two different observations. In panel data, this assumption almost always fails:

Firm effect: Residuals for the same firm are correlated across years.

Apple's unobserved shocks in 2020 are correlated with its shocks in 2021.

$\Rightarrow$ OLS SEs too small $\Rightarrow$ false significance

Time effect: Residuals for different firms are correlated within the same year.

A recession hits all firms simultaneously.

$\Rightarrow$ OLS SEs too small $\Rightarrow$ false significance

Petersen (2009): with moderate within-cluster correlation, **15% of tests reject at 1%** when the true rejection rate should be 1%.

Clustered SEs Allow for Within-Group Correlation

Clustered standard errors relax the independence assumption *within* clusters:

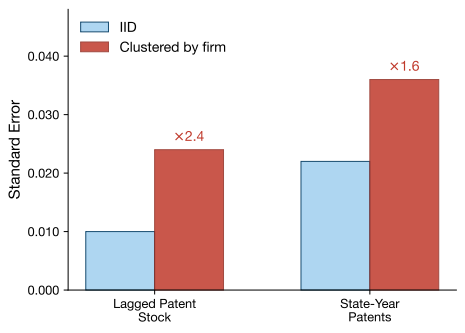
$$\hat{V}_{\text{cluster}} = (X'X)^{-1} \left[\sum_{g=1}^{G} X_g' \hat{\varepsilon}_g \hat{\varepsilon}_g' X_g \right] (X'X)^{-1}$$

Cluster by	Allows correlation	When to use
Firm	Within firm, across time	Firm-level shocks persist
Year	Within year, across firms	Common macro shocks
Firm + Year	Both dimensions	Both effects present

Rule of thumb: You need at least 30–50 clusters for reliable inference. Clustering by year with only 10 years of data is problematic.

Clustering doubles standard errors but does not change point estimates

	(4) Two-Way FE (IID SE)	(5) Two-Way FE (Clustered SE)
Patent stock	0.523***	0.523***
	(0.010)	(0.024)
State-yr pat.	0.114***	0.114**
	(0.022)	(0.036)
Firm + Year FE	✓	✓
S.E. type	IID	Clustered
N	44,898	44,898
R^2	0.749	0.749



IID SEs assume independence. Petersen (2009, *RFS*) shows they are biased downward in panels.

Rule: Cluster at minimum by firm, or at the treatment level.

Petersen (2009): Diagnose Your Data's Dependence Structure

Compare standard errors across methods to identify which effect dominates:

Pattern you observe	Diagnosis
$SE_{firm} \gg SE_{White}$	Firm effect dominates $\rightarrow$ cluster by firm
$SE_{year} \gg SE_{White}$	Time effect dominates $\rightarrow$ cluster by time
Both are large	Both effects $\rightarrow$ two-way cluster

Petersen's finding from real data:

- **Asset pricing** (firm–month): time effect dominates $\rightarrow$ Fama–MacBeth or cluster by time

Petersen (2009): Diagnose Your Data's Dependence Structure

Compare standard errors across methods to identify which effect dominates:

Pattern you observe	Diagnosis
$SE_{firm} \gg SE_{White}$	Firm effect dominates → cluster by firm
$SE_{year} \gg SE_{White}$	Time effect dominates → cluster by time
Both are large	Both effects → two-way cluster

Petersen's finding from real data:

- **Asset pricing** (firm–month): time effect dominates → Fama–MacBeth or cluster by time
- **Corporate finance** (firm–year): firm effect dominates → cluster by firm

Two-Way Clustering Handles Both Dimensions

When both firm and time effects are present, use two-way clustering (Thompson 2005):

$$\hat{V}_{\text{two-way}} = \hat{V}_{\text{firm}} + \hat{V}_{\text{time}} - \hat{V}_{\text{White}}$$

- Combines the firm-clustered and time-clustered variance matrices

In `pyfixest`, two-way clustering is a single argument: `vcov={"CRV1": "firm + year"}`.

Two-Way Clustering Handles Both Dimensions

When both firm and time effects are present, use two-way clustering (Thompson 2005):

$$\hat{V}_{\text{two-way}} = \hat{V}_{\text{firm}} + \hat{V}_{\text{time}} - \hat{V}_{\text{White}}$$

- Combines the firm-clustered and time-clustered variance matrices
- Subtracts the White (diagonal) matrix to avoid double-counting

In `pyfixest`, two-way clustering is a single argument: `vcov={"CRV1": "firm + year"}`.

Two-Way Clustering Handles Both Dimensions

When both firm and time effects are present, use two-way clustering (Thompson 2005):

$$\hat{V}_{\text{two-way}} = \hat{V}_{\text{firm}} + \hat{V}_{\text{time}} - \hat{V}_{\text{White}}$$

- Combines the firm-clustered and time-clustered variance matrices
- Subtracts the White (diagonal) matrix to avoid double-counting
- Produces correctly sized confidence intervals when both effects are present

In `pyfixest`, two-way clustering is a single argument: `vcov={"CRV1": "firm + year"}`.

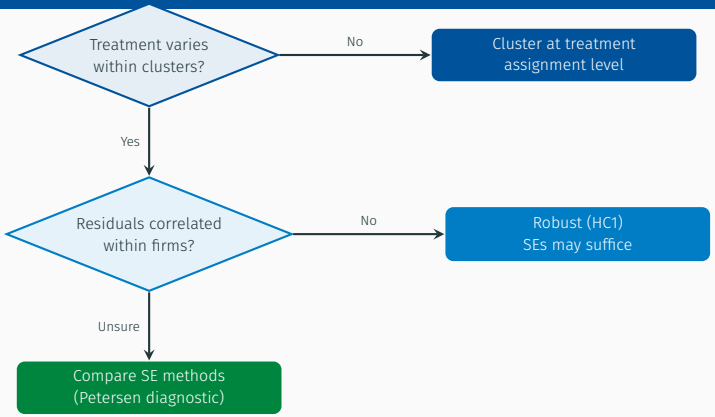
Abadie et al. (2023): Clustering Is Not Always Necessary

A common misconception: "always cluster to be safe." Abadie, Athey, Imbens & Wooldridge (2023) show this can be severely over-conservative.

Scenario	Correct approach
Treatment at cluster level (e.g. state law)	Cluster by that level
Treatment varies within clusters, all sampled	Robust (HC1) SEs may suffice
Clusters sampled from larger population	Cluster for sampling uncertainty

In their Census application, conventional cluster SEs were **19**× the correct SEs. Unnecessary clustering destroys power.

A Decision Framework for Your Dissertation



For most corporate finance dissertations: **two-way FE** (firm + year) with **SEs clustered by firm** is a sensible default. But always check.

From Stata to Python

Stata to Python: The Same Regression, Two Languages

Most published finance papers use Stata's `reghdfe`. Your dissertation will use Python's `pyfixest` — which has the **same syntax**.

Stata

```
reghdfe y x1 x2,
    absorb(firm year)
    cluster(firm)
```

Python (pyfixest)

```python
import pyfixest as pf

pf.feols(
    "y ~ x1 + x2 | firm + year",
    vcov={"CRV1": "firm"},
    data=df,
)
```

The formula syntax: `depvar ~ covariates | fixed_effects`

The `|` separates what you estimate from what you absorb.

A Real Example: Bena, Ortiz-Molina & Simintzi

Stata (original)

```
reghdfe cl_pcs gf ic pp
    L1log1pspat L1log1psrd
    L1logsales L1logmtbr
    L1pgdp L1pctdem,
    absorb(gvkey fyear)
    cluster(statecd)
```

Python (pyfixest)

```
fit = pf.feols(
    "cl_pcs ~ gf + ic + pp"
    " + L1log1pspat + L1log1psrd"
    " + L1logsales + L1logmtbr"
    " + L1pgdp + L1pctdem"
    " | gvkey + fyear",
    vcov={"CRV1": "statecd"},
    data=df,
)
```

Same regression, same results. The translation is almost mechanical.

Running Regressions in Python

Getting Started with `pyfixest`

Install it in your project:

```
$ uv add pyfixest
```

Then import and load the Bena et al. (2022) panel data:

```python
import pandas as pd
import pyfixest as pf

df = pd.read_csv("data/bena_ortiz_molina_simintzi_2022.csv")
print(f"Observations: {len(df):,}")
print(f"Firms:        {df['gvkey'].nunique():,}")
print(f"Years:        {df['fyear'].min()} - {df['fyear'].max()}")
```

```
Observations: 45,263
Firms:        4,447
Years:        1975 - 1997
```

Step 1: Pooled OLS (No Fixed Effects)

```
m1 = pf.feols("cl_pcs ~ L1log1pspat + logstyrpat", data=df)
m1.summary()
```

```
Dep. var.: cl_pcs, Fixed effects: 0
Observations:  45263

| Coefficient   |   Estimate |  Std. Error |  t value |
|:--------------|----------: |----------:  |--------: |
| L1log1pspat   |      0.847 |       0.004 |  235.513 |
| logstyrpat    |     -0.019 |       0.008 |   -2.239 |
---
RMSE: 1.217  R2: 0.555
```

Argument	What it does
`"cl_pcs ~ L1log1pspat + logstyrpat"`	Formula: outcome ~ covariates
`data=df`	The DataFrame containing your variables

Step 2: Add Firm Fixed Effects

```
m2 = pf.feols("cl_pcs ~ L1log1pspat + logstyrpat | gvkey", data=df)
```

```
Dep. var.: cl_pcs, Fixed effects: gvkey
Observations:  44898

| Coefficient   |   Estimate |  Std. Error |  t value |
|:--------------|----------: |----------:  |--------: |
| L1log1pspat   |      0.486 |       0.008 |   58.615 |
| logstyrpat    |      0.312 |       0.019 |   16.139 |
---
RMSE: 0.921  R2: 0.745
```

- The `| gvkey` tells `pyfixest` to absorb firm fixed effects

Step 2: Add Firm Fixed Effects

```python
m2 = pf.feols("cl_pcs ~ L1log1pspat + logstyrpat | gvkey", data=df)
```

```
Dep. var.: cl_pcs, Fixed effects: gvkey
Observations:  44898

| Coefficient   |   Estimate |  Std. Error |  t value |
|:--------------|----------: |----------:  |--------: |
| L1log1pspat   |      0.486 |       0.008 |   58.615 |
| logstyrpat    |      0.312 |       0.019 |   16.139 |
---
RMSE: 0.921  R2: 0.745
```

- The `| gvkey` tells `pyfixest` to absorb firm fixed effects
- Patent stock coefficient drops from 0.847 to 0.486 — within-firm variation only

Step 2: Add Firm Fixed Effects

```
m2 = pf.feols("cl_pcs ~ L1log1pspat + logstyrpat | gvkey", data=df)
```

```
Dep. var.: cl_pcs, Fixed effects: gvkey
Observations:  44898

| Coefficient   |   Estimate |  Std. Error |  t value |
|:--------------|----------: |----------:  |--------: |
| L1log1pspat   |      0.486 |       0.008 |   58.615 |
| logstyrpat    |      0.312 |       0.019 |   16.139 |
---
RMSE: 0.921  R2: 0.745
```

- The `| gvkey` tells `pyfixest` to absorb firm fixed effects
- Patent stock coefficient drops from 0.847 to 0.486 — within-firm variation only
- 365 singleton firms (one observation) are dropped

Step 3: Two-Way Fixed Effects

```python
m4 = pf.feols(
    "cl_pcs ~ L1log1pspat + logstyrpat | gvkey + fyear", data=df
)
```

```
Dep. var.: cl_pcs, Fixed effects: gvkey+fyear
Observations:  44898

| Coefficient   |   Estimate |  Std. Error |  t value |
|:--------------|----------: |----------:  |--------: |
| L1log1pspat   |      0.523 |       0.010 |   51.572 |
| logstyrpat    |      0.114 |       0.022 |    5.122 |
---
RMSE: 0.914  R2: 0.749
```

- `gvkey + fyear` absorbs firm and year fixed effects simultaneously

Step 3: Two-Way Fixed Effects

```
m4 = pf.feols(
    "cl_pcs ~ L1log1pspat + logstyrpat | gvkey + fyear", data=df
)
```

```
Dep. var.: cl_pcs, Fixed effects: gvkey+fyear
Observations:  44898

| Coefficient   |   Estimate |  Std. Error |  t value |
|:--------------|----------: |----------:  |--------: |
| L1log1pspat   |      0.523 |       0.010 |   51.572 |
| logstyrpat    |      0.114 |       0.022 |    5.122 |
---
RMSE: 0.914  R2: 0.749
```

- `gvkey + fyear` absorbs firm and year fixed effects simultaneously
- Coefficient stabilises at **0.523** — the preferred two-way FE estimate

Step 3: Two-Way Fixed Effects

```
m4 = pf.feols(
    "cl_pcs ~ L1log1pspat + logstyrpat | gvkey + fyear", data=df
)
```

```
Dep. var.: cl_pcs, Fixed effects: gvkey+fyear
Observations:  44898

| Coefficient   |   Estimate |  Std. Error |  t value |
|:--------------|----------: |----------:  |--------: |
| L1log1pspat   |      0.523 |       0.010 |   51.572 |
| logstyrpat    |      0.114 |       0.022 |    5.122 |
---
RMSE: 0.914  R2: 0.749
```

- `gvkey + fyear` absorbs firm and year fixed effects simultaneously
- Coefficient stabilises at **0.523** — the preferred two-way FE estimate
- Equivalent to Stata's `absorb(gvkey fyear)`

Step 4: Add Clustered Standard Errors

```python
# One-way clustering by firm
m5 = pf.feols("cl_pcs ~ L1log1pspat + logstyrpat | gvkey + fyear",
              vcov={"CRV1": "gvkey"}, data=df)
```

```
Inference:  CRV1
| Coefficient   |   Estimate |  Std. Error |  t value |
|:--------------|----------: |----------:  |--------: |
| L1log1pspat   |      0.523 |       0.024 |   22.256 |
| logstyrpat    |      0.114 |       0.036 |    3.149 |
```

vcov argument	What it does
`"iid"`	Homoskedastic (classical) SEs
`"HC1"`	Heteroskedasticity-robust
`{"CRV1": "gvkey"}`	Cluster by firm
`{"CRV1": "gvkey + fyear"}`	Two-way cluster by firm and year

You Can Change Standard Errors After Estimation

A powerful feature: re-compute SEs without re-running the regression.

```
# Estimate once
fit = pf.feols("cl_pcs ~ L1log1pspat + logstyrpat | gvkey + fyear",
               data=df)

# Compare different SE methods
fit.vcov("iid").summary()       # SE = 0.010
fit.vcov("HC1").summary()       # SE = 0.012
fit.vcov({"CRV1": "gvkey"}).summary()  # SE = 0.024
```

This is Petersen's diagnostic in practice: run the regression once, then compare how SEs change across clustering methods to identify the dependence structure.

Extracting What You Need

```
fit = pf.feols("cl_pcs ~ L1log1pspat + logstyrpat | gvkey + fyear",
               vcov={"CRV1": "gvkey"}, data=df)

fit.coef()       # coefficient estimates
fit.se()         # standard errors
fit.tstat()      # t-statistics
fit.pvalue()     # p-values
fit.confint()    # confidence intervals
fit.tidy()       # everything in one DataFrame
```

`fit.tidy()` returns a clean DataFrame with coefficients, SEs, t-stats, p-values, and confidence intervals — ready for further analysis or export.

Running Multiple Specifications at Once

`pyfixest` can estimate multiple models in a single call:

```
# csw0(): cumulative stepwise, including baseline
fits = pf.feols("cl_pcs ~ csw0(L1log1pspat, logstyrpat) | gvkey + fyear",
                vcov={"CRV1": "gvkey"}, data=df)
```

This estimates **three** models:

Model	Formula
(1)	`cl_pcs ~ 1 | gvkey + fyear`
(2)	`cl_pcs ~ L1log1pspat | gvkey + fyear`
(3)	`cl_pcs ~ L1log1pspat + logstyrpat | gvkey + fyear`

This mirrors a “kitchen sink” progression — adding controls one at a time.

Exercise 1: Run Your First Regressions

Try It Yourself (10 minutes)

1. Install `pyfixest`: `uv add pyfixest`
2. Load the Bena et al. data from `week-5-code/data/`
3. Run a pooled OLS: `pf.feols("cl_pcs ~ L1log1pspat + logstyrpat", data=df)`
4. Add firm FE: `pf.feols("cl_pcs ~ ... | gvkey", data=df)`
5. Add two-way FE: `pf.feols("cl_pcs ~ ... | gvkey + fyear", data=df)`
6. Compare: how does the coefficient on `L1log1pspat` change?
7. **Bonus:** Use `.vcov()` to compare IID, robust, and firm-clustered SEs

Presenting Results

Publication-Quality Tables with `etable()`

```
pf.etable(
    [m1, m2, m3, m4, m5],
    labels={"cl_pcs": "Process Innovation",
            "L1log1pspat": "Lagged Patent Stock",
            "logstyrpat": "State-Year Patents"},
)
```

This is the actual output from our five models:

	(1) Pooled	(2) Firm FE	(3) Year FE	(4) Two-Way	(5) Clustered
Lagged Patent Stock	0.847***	0.486***	0.855***	0.523***	0.523***
	(0.004)	(0.008)	(0.004)	(0.010)	(0.024)
State-Year Patents	−0.019*	0.312***	−0.069***	0.114***	0.114**
	(0.008)	(0.019)	(0.008)	(0.022)	(0.036)
Firm FE		✓		✓	✓
Year FE			✓	✓	✓
S.E. type	IID	IID	IID	IID	Clustered
N	45,263	44,898	45,263	44,898	44,898
R^2	0.555	0.745	0.569	0.749	0.749

Customising the Table

```
pf.etable(
    [m5, m6],
    labels={"cl_pcs": "Process Innovation",
            "cl_pdt": "Product Innovation",
            "L1log1pspat": "Lagged Patent Stock",
            "logstyrpat": "State-Year Patents"},
    coef_fmt="b \n (se)",     # coefficient over SE
    signif_code=[0.01, 0.05, 0.1],
    type="gt",                # interactive table
)
```

Option	What it does
`labels`	Replace variable names with readable labels
`coef_fmt`	Control display format (`b (t)` for t-stats)
`type="tex"`	Output as LaTeX for your dissertation
`keep=["L1log1pspat"]`	Show only selected coefficients

Exporting Tables for Your Dissertation

```python
# Export as LaTeX
pf.etable(
    [m1, m2, m4, m5],
    labels={"L1log1pspat": "Lagged Patent Stock",
            "logstyrpat": "State-Year Patents"},
    type="tex",
    file_name="results/table_1.tex",
)
```

This writes a complete LaTeX table using **booktabs** and **threeparttable** that you can include directly in your dissertation:

```latex
\input{results/table_1.tex}
```

Workflow: Run your Python script → export table to **.tex** → include in your LaTeX document. Change one number in Python and the table updates everywhere.

Coefficient Plots: Visualising Your Results

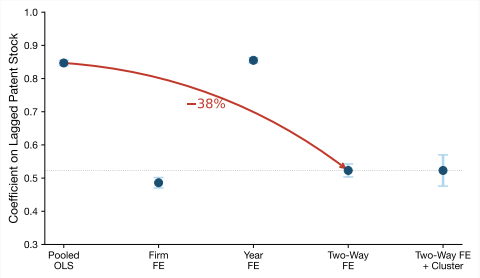
Sometimes a chart communicates better than a table:

```
pf.coefplot([m1, m2, m4, m5])
```

This plots each coefficient with its confidence interval. Useful for:

- Comparing coefficients across specifications

You can also use **plotnine** with `fit.tidy()` for full control over the plot aesthetics.



Actual output: the coefficient on patent stock drops by 38% from pooled OLS to two-way FE.

Coefficient Plots: Visualising Your Results

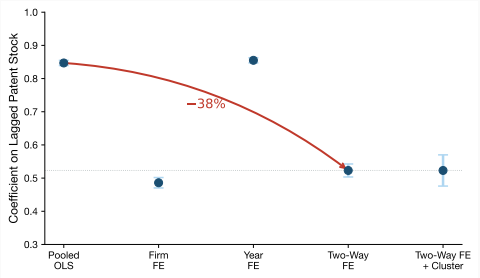
Sometimes a chart communicates better than a table:

```
pf.coefplot([m1, m2, m4, m5])
```

This plots each coefficient with its confidence interval. Useful for:

- Comparing coefficients across specifications
- Showing which variables are significant

You can also use **plotnine** with `fit.tidy()` for full control over the plot aesthetics.



Actual output: the coefficient on patent stock drops by 38% from pooled OLS to two-way FE.

Coefficient Plots: Visualising Your Results

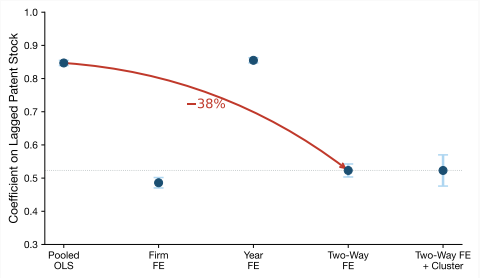
Sometimes a chart communicates better than a table:

```
pf.coefplot([m1, m2, m4, m5])
```

This plots each coefficient with its confidence interval. Useful for:

- Comparing coefficients across specifications
- Showing which variables are significant
- Event study plots (leads and lags)

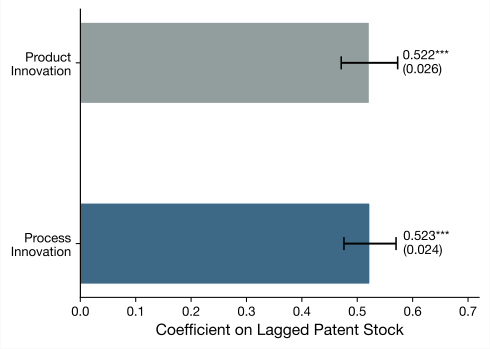
You can also use **plotnine** with `fit.tidy()` for full control over the plot aesthetics.



Actual output: the coefficient on patent stock drops by 38% from pooled OLS to two-way FE.

Patent stock predicts both process and product innovation equally

	Process	Product
Patent stock	0.523***	0.522***
	(0.024)	(0.026)
State-yr pat.	0.114**	0.136**
	(0.036)	(0.042)
Firm FE	✓	✓
Year FE	✓	✓
Clustered SE	✓	✓
N	44,898	44,898
R^2	0.749	0.718



The coefficient on patent stock is nearly identical for both outcomes. The regional innovation environment matters slightly more for product innovation.

Testing Hypotheses About Your Coefficients

The `marginaleffects` package works with `pyfixest` for hypothesis testing:

```
$ uv add marginaleffects
```

```python
from marginaleffects import hypotheses

fit = pf.feols("cl_pcs ~ L1log1pspat + logstyrpat | gvkey + fyear",
               vcov={"CRV1": "gvkey"}, data=df)

# Test: is the patent stock effect = state-year effect?
hypotheses(fit, "L1log1pspat - logstyrpat = 0")

# Non-linear hypothesis (delta method)
hypotheses(fit, "(L1log1pspat / logstyrpat) = 1")
```

The delta method computes correct standard errors for non-linear transformations of coefficients — essential for ratios, percentage effects, and elasticities.

Exercise 2: Build a Regression Table

Try It Yourself (10 minutes)

1. Re-run the five models from `01_fixed_effects_regressions.py`
2. Use `pf.etable([m1, m2, m3, m4, m5])` to display them side by side
3. Add labels with the `labels=` argument
4. Try `pf.coefplot([m1, m4, m5])`
5. **Bonus:** Export the table as LaTeX with `type="tex"`
6. **Bonus:** Swap the dependent variable to `cl_pdt` (product innovation) and compare

Summary

What You Learned Today

Econometric foundations

- Panel data: unit × time structure
- Pooled OLS and omitted variable bias
- Fixed effects absorb time-invariant confounders
- FE limitations: no dynamics, within variation only
- Mummolo & Peterson: honest counterfactuals

Inference and Python

- Why OLS SEs are biased in panels
- Clustered SEs: one-way, two-way
- Petersen's diagnostic for choosing SE method
- Abadie et al.: clustering is not always needed
- `pyfixest`: `feols()`, `etable()`, `coefplot()`
- LaTeX export for your dissertation

Four Papers You Should Know

Paper	Key insight for your dissertation
Petersen (2009)	Compare SE methods to diagnose dependence structure; firm effect → cluster by firm; time effect → Fama–MacBeth or cluster by time
Imai & Kim (2019)	FE assumes no dynamic feedback between treatment and outcome; if treatment persists or outcomes feed back, FE is inconsistent
Mummolo & Peterson (2018)	Use within-unit SD for counterfactuals; many published effect sizes are overstated
Abadie et al. (2023)	Cluster at the level of treatment assignment; unnecessary clustering wastes statistical power

Quick Self-Check

Can you answer these in your own words?

1. What does a firm fixed effect absorb, and what can it *not* absorb?

Quick Self-Check

Can you answer these in your own words?

1. What does a firm fixed effect absorb, and what can it *not* absorb?
2. Why did the patent stock coefficient drop from 0.847 to 0.486 when we added firm FE?

Quick Self-Check

Can you answer these in your own words?

1. What does a firm fixed effect absorb, and what can it *not* absorb?
2. Why did the patent stock coefficient drop from 0.847 to 0.486 when we added firm FE?
3. When should you cluster by firm vs. by time vs. two-way?

Quick Self-Check

Can you answer these in your own words?

1. What does a firm fixed effect absorb, and what can it *not* absorb?
2. Why did the patent stock coefficient drop from 0.847 to 0.486 when we added firm FE?
3. When should you cluster by firm vs. by time vs. two-way?
4. In `pyfixest`, what does the `|` in the formula do?

Quick Self-Check

Can you answer these in your own words?

1. What does a firm fixed effect absorb, and what can it *not* absorb?
2. Why did the patent stock coefficient drop from 0.847 to 0.486 when we added firm FE?
3. When should you cluster by firm vs. by time vs. two-way?
4. In `pyfixest`, what does the `|` in the formula do?
5. What is the difference between `vcov="HC1"` and `vcov={"CRV1": "gvkey"}`?

Quick Self-Check

Can you answer these in your own words?

1. What does a firm fixed effect absorb, and what can it *not* absorb?
2. Why did the patent stock coefficient drop from 0.847 to 0.486 when we added firm FE?
3. When should you cluster by firm vs. by time vs. two-way?
4. In `pyfixest`, what does the `|` in the formula do?
5. What is the difference between `vcov="HC1"` and `vcov={"CRV1": "gvkey"}`?
6. Why does Mummolo & Peterson (2018) say you should residualize X before interpreting effect sizes?

What's Next

- After Spring Vacation (Week 6): Event Studies and Instrumental Variable Estimations

If something breaks: read the error message, check your formula syntax, try again.
Still stuck? Email me at ce50@st-andrews.ac.uk.

What's Next

- After Spring Vacation (Week 6): Event Studies and Instrumental Variable Estimations
- **Homework:**

If something breaks: read the error message, check your formula syntax, try again.
Still stuck? Email me at ce50@st-andrews.ac.uk.

What's Next

- After Spring Vacation (Week 6): Event Studies and Instrumental Variable Estimations
- **Homework:**
  - Run `week-5-code/01_fixed_effects_regressions.py` and verify you get the same results

If something breaks: read the error message, check your formula syntax, try again.
Still stuck? Email me at ce50@st-andrews.ac.uk.

What's Next

- After Spring Vacation (Week 6): Event Studies and Instrumental Variable Estimations
- **Homework:**
  - Run `week-5-code/01_fixed_effects_regressions.py` and verify you get the same results
  - Replicate one regression table from a published paper in your area using `pyfixest`

If something breaks: read the error message, check your formula syntax, try again.
Still stuck? Email me at ce50@st-andrews.ac.uk.

What's Next

- After Spring Vacation (Week 6): Event Studies and Instrumental Variable Estimations
- **Homework:**
  - Run `week-5-code/01_fixed_effects_regressions.py` and verify you get the same results
  - Replicate one regression table from a published paper in your area using `pyfixest`
  - Read Petersen (2009) Sections I–III

If something breaks: read the error message, check your formula syntax, try again.
Still stuck? Email me at ce50@st-andrews.ac.uk.

What's Next

- After Spring Vacation (Week 6): Event Studies and Instrumental Variable Estimations
- **Homework:**
  - Run `week-5-code/01_fixed_effects_regressions.py` and verify you get the same results
  - Replicate one regression table from a published paper in your area using `pyfixest`
  - Read Petersen (2009) Sections I–III
- Resources:

If something breaks: read the error message, check your formula syntax, try again.
Still stuck? Email me at ce50@st-andrews.ac.uk.

What's Next

- After Spring Vacation (Week 6): Event Studies and Instrumental Variable Estimations
- **Homework:**
  - Run `week-5-code/01_fixed_effects_regressions.py` and verify you get the same results
  - Replicate one regression table from a published paper in your area using `pyfixest`
  - Read Petersen (2009) Sections I–III
- Resources:
  - pyfixest docs: https://www.pyfixest.org/quickstart.html

If something breaks: read the error message, check your formula syntax, try again.
Still stuck? Email me at ce50@st-andrews.ac.uk.

What's Next

- After Spring Vacation (Week 6): Event Studies and Instrumental Variable Estimations
- **Homework:**
  - Run `week-5-code/01_fixed_effects_regressions.py` and verify you get the same results
  - Replicate one regression table from a published paper in your area using `pyfixest`
  - Read Petersen (2009) Sections I–III
- Resources:
  - pyfixest docs: https://www.pyfixest.org/quickstart.html
  - Tidy Finance (Python): https://www.tidy-finance.org/python/fixed-effects-and-clustered-standard-errors.html

If something breaks: read the error message, check your formula syntax, try again.
Still stuck? Email me at ce50@st-andrews.ac.uk.

What's Next

- After Spring Vacation (Week 6): Event Studies and Instrumental Variable Estimations
- **Homework:**
  - Run `week-5-code/01_fixed_effects_regressions.py` and verify you get the same results
  - Replicate one regression table from a published paper in your area using `pyfixest`
  - Read Petersen (2009) Sections I–III
- Resources:
  - pyfixest docs: https://www.pyfixest.org/quickstart.html
  - Tidy Finance (Python): https://www.tidy-finance.org/python/fixed-effects-and-clustered-standard-errors.html
  - Petersen SE website: https://www.kellogg.northwestern.edu/faculty/petersen/htm/papers/se/se_programming.htm

If something breaks: read the error message, check your formula syntax, try again.
Still stuck? Email me at ce50@st-andrews.ac.uk.