

Difference-in-differences and event studies in Python

Programming & Data Management, Part II • Further topic F4
(optional, not examined)

Magnus Lodefalk
Örebro University

NA446A, Örebro University
Autumn 2026



Today's goals

By the end of today you can:

1. Compute a **difference-in-differences** by hand from four means, and get the same number from a **two-way fixed-effects regression**
2. Estimate and plot an **event study**, and read its pre-trends
3. Use **Poisson with fixed effects** for a count outcome, as the lab does for job ads

Six terms for this module

treated, control	firms the policy (or shock) reaches, and firms it does not
pre, post	years before and from the start (here 2020)
DiD	treated firms' change minus control firms' change
parallel trends	the assumption: without treatment, both groups would have moved alike
event study	one treated-control gap per year, against a reference year (2019); the years before are the <i>leads</i>
Poisson FE	a regression for counts (0, 1, 2, ...) with fixed effects; keeps the zeros

- Same toolkit as F1: `pf.feols` with `| firm+year`, clustered by firm; plus `pf.fepois` for counts.

Needs `pyfixest` only: `uv add pyfixest`.

The data are made up, so we know the answer

```
panel["ln_emp"] = (3.0 + panel["firm_effect"]  
                  + panel["year_effect"]  
                  + TRUE_EFFECT * panel["treat_post"]  
                  + noise)
```

	firm	year	treat	post	treat_post	ln_emp	hires
0	1	2015	1	0	0	3.207630	9

f4_make_data.py: 500 firms, 2015–24; 216 treated from 2020. Treated firms are larger (level), all share one year path (parallel trends), the rest is noise. The **true effect** is **−0.10** on log employment and on hires.

- Like a register extract: one row per firm and year. Seeded: same file every run.

Stata: `set seed, expand, rnormal(), rpoisson()`; the script's comments give each line.

Difference-in-differences

Step 1: the 2×2 DiD by hand, from four means

```
means = df.groupby(["treat", "post"])["ln_emp"].mean()
change_treated = means.loc[(1, 1)] - means.loc[(1, 0)]
change_control = means.loc[(0, 1)] - means.loc[(0, 0)]
did_by_hand = change_treated - change_control
```

```
change, treated: -0.0306
```

```
change, control: 0.0797
```

```
DiD by hand:      -0.1103
```

- Control firms grew by 0.08, so that is what treated firms would have done too. They fell by 0.03 instead: the effect is $-0.03 - 0.08 = -0.11$.

```
Stata: table treat post, statistic(mean ln_emp).
```

Step 2: the same number as a regression

```
twfe = pf.feols("ln_emp ~ treat_post | firm+year",  
                data=df, vcov={"CRV1": "firm"})  
print(twfe.tidy().round(4))
```

	Estimate	Std. Error	2.5%	97.5%
treat_post	-0.1103	0.0111	-0.132	-0.0885
true effect	-0.1 inside the 95% CI: True			

- The same -0.1103 as by hand (balanced panel, one start year). The interval contains the true -0.10 : **the estimator finds the answer.**

Stata: `reghdfe ln_emp treat_post, absorb(firm year) vce(cluster firm)`: -0.1103 (0.0111).

Check your understanding

Answer on your own first, then compare with a neighbour

Predict: drop the year effects and keep only firm effects
(`"ln_emp ~ treat_post | firm"`). What estimate do you get,
and why is it wrong?

Check your understanding

Answer on your own first, then compare with a neighbour

Predict: drop the year effects and keep only firm effects (`"ln_emp ~ treat_post | firm"`). What estimate do you get, and why is it wrong?

Answer: -0.031 , exactly the treated firms' own before-after change from Step 1. Without year effects nothing tells the regression what would have happened anyway, so the common growth of $+0.08$ is mixed into the estimate and hides most of the true -0.10 . The control group is what makes it a difference-in-differences.

Step 3: an event study, one gap per year

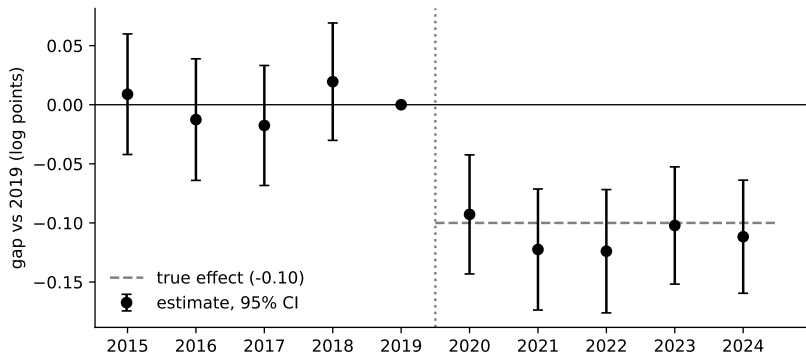
```
event = pf.feols(  
  "ln_emp ~ i(year, treat, ref=2019) | firm+year",  
  data=df, vcov={"CRV1": "firm"})  
es_table = event.tidy()
```

	year	Estimate	Std. Error
year::2018:treat	2018	0.0195	0.0253
mean of the leads (pre-2020):		-0.0004	
mean of the lags (2020 on):		-0.1106	

- A dummy for treated $\times$ each year except 2019. Leads near 0, lags near -0.10. Selected rows and columns.

Stata: `ev_2015...ev_2024 (no 2019) in a forvalues loop, then reghdfe ln_emp ev_*, absorb(firm year)`
`vce(cluster firm).`

Step 4: the event-study figure



- Before 2020 the gaps scatter around zero: no pre-trend. From 2020 they sit on the true effect. This is the figure referees look at first.

`ax.errorbar(...)` in matplotlib; the script adds the 2019 zero as a row. Stata: `coefplot (SSC)`.

Check your understanding

Answer on your own first, then compare with a neighbour

The leads are all small and insignificant (2018: 0.020, standard error 0.025). Does that prove the parallel-trends assumption?

Check your understanding

Answer on your own first, then compare with a neighbour

The leads are all small and insignificant (2018: 0.020, standard error 0.025). Does that prove the parallel-trends assumption?

Answer: No. It shows no *detectable* difference before 2020, and each lead's confidence interval still allows a gap of about ± 0.05 , half the size of the effect. Parallel trends is about what would have happened *after* 2020 without treatment, which no data can show. Honest practice asks how much of a violation would overturn the result (next slide).

Step 5: Poisson with fixed effects for hires

```
poisson = pf.fepois("hires ~ treat_post | firm+year",  
                   data=df, vcov={"CRV1": "firm"})  
b_poisson = poisson.coef()["treat_post"]  
pct = 100 * (np.exp(b_poisson) - 1)
```

```
rows with zero hires: 30  
                Estimate  Std. Error  
treat_post      -0.099      0.0182  
per cent change in hires: -9.4  
true per cent change:      -9.5
```

- Counts have zeros, and $\log(0)$ is undefined. Poisson keeps them and is read on the log scale. The lab's canaries paper estimates job ads this way.

Stata: `ppmlhdfc hires treat_post, absorb(firm year) vce(cluster firm)`: -0.0990 (0.0182).

Step 6: one table with the three models

```
models = [twfe, firm_only, poisson]
table = pf.etable(models, type="df",
    signif_code=[0.01, 0.05, 0.1],
    coef_fmt="b:.3f* (se:.3f)")
```

		ln_emp				hires	
		(1)		(2)		(3)	
coef	treat_post	-0.110***	(0.011)	-0.031***	(0.008)	-0.099***	(0.018)
fe	firm		x		x		x
	year		x		-		x

- Column (2) is the concept check: highly “significant” and wrong. Stars do not tell you whether the design is right.

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

Where to go next

Pre-trends, honestly. Rambachan and Roth (2023), “A More Credible Approach to Parallel Trends”, *Review of Economic Studies* 90(5): 2555–2591: how large a violation would overturn the result? HonestDiD (R; Stata version linked):

github.com/asheshrambachan/HonestDiD

Staggered adoption. When firms start in different years, two-way FE also compares late with early starters and can mislead. Callaway and Sant’Anna (2021), “Difference-in-Differences with Multiple Time Periods”, *Journal of Econometrics* 225(2): 200–230; open version arxiv.org/abs/1803.09015; R package did: github.com/bcallaway11/did; Stata: `csdid`

In Python. `pyfixest` also has staggered-adoption estimators (did2s, Sun–Abraham event studies): github.com/py-econometrics/pyfixest

Links checked 30 Sep 2026.

Summary

1. DiD is treated change minus control change; with one start year and a balanced panel, `pf.feols("y ~ treat_post | firm+year")` gives exactly the same number
2. `i(year, treat, ref=2019)` gives the event study; plot it, and read the leads as a check, not a proof
3. `pf.fepois` for counts; cluster by the unit that is treated (here the firm)
 - On made-up data with a true effect of -0.10 , all three estimators recover it: -0.110 (0.011), lags averaging -0.111 , and Poisson -0.099 (0.018).