Stata Skill Portfolio

The Reference

# A Stata Reference for Empirical Economics

## Methods, Code, and Verified Results

*from data management to causal inference, one technique at a time*

Duc V. Le

Lê Vĩnh Đức

PhD Candidate, Department of Economics · Georgetown University

Office ICC-580 · 3700 O St NW, Washington, DC 20057, USA

dvl11@georgetown.edu

*Built and verified on Stata 19.5 SE · Updated June 2026*

# Contents

**Preface** **3**

**Roadmap: choosing a method** **4**

**I Setup and environment** **7**

**1 Getting started** **8**
1.1 Running the code . . . 8
1.2 Packages . . . 8

**II Empirical foundations** **9**

**2 Stata basics** **10**
2.1 The code . . . 10
2.2 The results . . . 11

**3 Data management** **13**
3.1 The code . . . 13
3.2 The results . . . 16

**4 Linear regression** **18**
4.1 The method . . . 18
4.2 The code . . . 18
4.3 The results . . . 19

**5 Panel data, DiD, and event studies** **22**
5.1 The method . . . 22
5.2 The code . . . 22
5.3 The results . . . 24

**6 Time series** **26**
6.1 The method . . . 26
6.2 The code . . . 26
6.3 The results . . . 27

**7 Importing public data** **29**
7.1 The code . . . 29
7.2 The results . . . 30

**III The causal-inference toolkit** **32**

**8 Instrumental variables (2SLS)** **33**
8.1 The method . . . 33
8.2 The code . . . 33
8.3 The results . . . 35

**9 Binary outcome models** **37**
9.1 The method . . . 37
9.2 The code . . . 37
9.3 The results . . . 39

**10 Staggered-adoption DiD** **40**
10.1 The method . . . 40
10.2 The code . . . 40
10.3 The results . . . 42

**11 Regression discontinuity** **44**
11.1 The method . . . 44
11.2 The code . . . 44
11.3 The results . . . 47

**12 Synthetic control** **49**
12.1 The method . . . 49
12.2 The code . . . 49
12.3 The results . . . 51

**13 Dynamic panel data (GMM)** **53**
13.1 The method . . . 53
13.2 The code . . . 53
13.3 The results . . . 55

**IV Appendices** **57**

**A Python/R → Stata cheat sheet** **58**

**B The methods, explained** **59**
B.1 Ordinary least squares . . . 59
B.2 Panel data, difference-in-differences, and event studies . . . 60
B.3 Time series . . . 61
B.4 Instrumental variables (2SLS) . . . 62
B.5 Binary outcome models . . . 63
B.6 Staggered-adoption difference-in-differences . . . 63
B.7 Regression discontinuity . . . 64
B.8 Synthetic control . . . 65
B.9 Dynamic panel data (GMM) . . . 66

**References** **68**

# Preface

This reference collects, in one place, the twelve scripts of the Stata skill portfolio together with the method behind each, the code, and the results it produces. The portfolio's individual files (`01_basics.do ... 11_synthetic_control.do`) and the package documentation (`packages/PACKAGES.md`) are excellent as references, but they are separate. This document binds them into a single, coherent textbook so the material reads as one body of knowledge.

**Code availability.** The twelve `.do` scripts behind every figure, table, and number in this reference — and this document itself — are openly available at `github.com/duc-v-le/stata-empirical-methods`.

A companion volume, *Stata Line-by-Line: An Empirical Economics Guide*, walks the same eleven scripts command by command for readers new to Stata's syntax; this book assumes that fluency and concentrates on the method, the code, and the results.

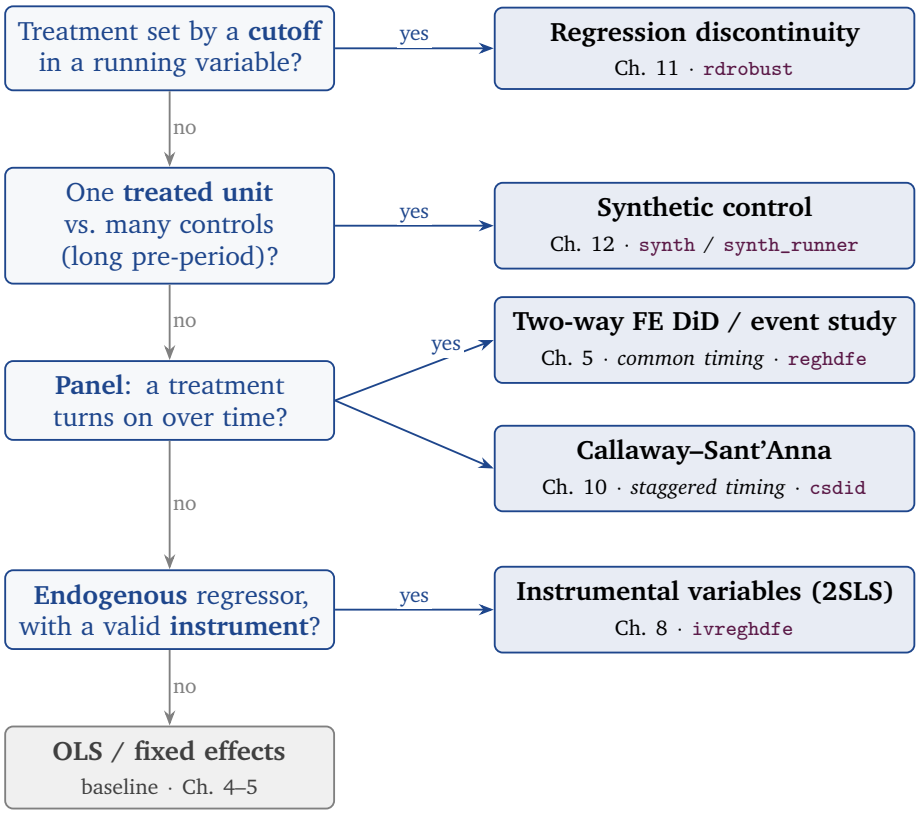
**What this is — and is not.** This is a *learning and demonstration* portfolio, not a research project. The data are either Stata's bundled example datasets, a real public series from FRED, or simulated with a fixed seed so that each estimator can be checked against a *known* true parameter. Every chapter ends by comparing the estimate to that truth.

**How the book is organised.** Part I sets up the environment and packages. Part II covers the empirical foundations — data management, regression, panel methods, time series, and importing public data. Part III is the modern causal-inference toolkit — instrumental variables, limited-dependent-variable models, staggered difference-in-differences, regression discontinuity, and synthetic control. The appendices give a Python/R→Stata translation table; a from-scratch explanation of every method (Appendix B, with — for each estimator — its idea, symbols, formula, *why* it works, the assumption and what breaks it, a worked example, and a tie-back to that chapter's own run showing it recovers the planted answer); and the references.

**Reproducing everything.** From the portfolio root, run `do 00_run_all.do` in Stata (after installing dependencies with `do packages/install_packages.do`). Each script writes its log to `logs/` and its figures and tables to `output/` — the same files embedded throughout this reference.

# Roadmap: choosing a method

The questions lead to the identification strategy that fits the data, then to that chapter. Outcome- and data-type choices (binary outcomes, a single time series, dynamic panels) are *orthogonal* to the design and are boxed separately at the bottom. For an unfamiliar method, Appendix B ("The methods, explained") builds each one up from scratch — intuition, formula, a worked example, and the result the chapter recovers.




*Orthogonal to the design — pick by the outcome / data type:*

| **Binary 0/1 outcome** | **Single time series** | **Dynamic panel** |
|---|---|---|
| Ch. 9 · logit / probit | Ch. 6 · AR + Newey–West | Ch. 13 · diff / sys GMM |
| `logit probit margins` | `tsset newey` | `xtabond xtdpdsys` |

**Always** clean and document the data first (Ch. 3); report robust or clustered standard errors.

## How to read the roadmap

The diagram is a *decision tree*, read from the top-left box straight down, one yes/no question at a time about the data and how the treatment came about. The first "yes" follows the arrow to the method on the right — the identification strategy — and the box names the chapter and the Stata command; a "no" drops to the next question. The questions run from the most specific design to the most general, so the *first* "yes" is the answer; if every answer is "no," the path ends at the plain-regression baseline at the bottom. The tree sorts by *identification strategy*, not by data shape: one panel dataset can land on regression discontinuity, difference-in-differences, instrumental variables, or the baseline, depending on where its causal leverage comes from.

**The five questions, in plain words.**

- **A cutoff in a running variable?** Did treatment switch on the instant some number crossed a threshold — a test score clearing a pass mark, an age passing 65, a vote share above 50%? Then units just below and just above the cutoff are otherwise alike, and the *jump* in the outcome at the threshold is the effect. ⇒ **regression discontinuity** (Ch. 11).
- **One treated unit, many controls?** Exactly one state (or firm, or country) was treated, with a long run of pre-treatment history? Build a weighted blend of the untreated units that mimics the treated one's path *before* treatment, and read the effect as the gap that opens *after*. ⇒ **synthetic control** (Ch. 12).
- **A panel where treatment turns on over time?** Many units followed over time, with a treatment that switches on. If everyone switches on the *same* date, two-way fixed-effects difference-in-differences (Ch. 5); if they switch at *different* dates (a staggered roll-out), naive two-way FE is biased, so use Callaway–Sant'Anna (Ch. 10).
- **An endogenous regressor with a valid instrument?** The key right-hand-side variable is correlated with the error — reverse causality, or an omitted confounder — but an *instrument* exists that shifts that regressor without otherwise touching the outcome. This question is about the *regressor*, not the data shape: it applies to a panel just as well as a cross-section — `ivreghdfe` instruments *while* absorbing fixed effects. (It sits on the "no" branch only because the prior box asks whether the design is a treatment-switches-on *difference-in-differences*, not whether the data happen to be a panel.) ⇒ **instrumental variables / 2SLS** (Ch. 8).
- **None of the above?** With no special design, fall back to OLS / fixed effects (Ch. 4–5) — the honest baseline. Report the result as a *descriptive association*, not a causal effect.

**The three boxes at the bottom answer a different question.** The tree chooses the *design* — how a correlation is argued to be causal. The bottom boxes are about the *shape of the outcome or data*, and they are *orthogonal*: one layers on top of whatever design the tree picked, not instead of it. A **binary 0/1 outcome** calls for logit/probit and `margins` (Ch. 9); a **single time series** for an autoregression with Newey–West HAC errors (Ch. 6); a **panel with a lagged dependent variable** (persistence) for dynamic-panel GMM (Ch. 13). They mix freely with the tree — a difference-in-differences *on a binary outcome*, say, uses a design box *and* the logit box.

**Two rules always apply.** Whatever the path: **clean and document the data first** (Ch. 3), and **report robust or clustered standard errors** — clustered at the level treatment varies (state, firm, individual).

**A worked walk-through.** Consider 50 states, a minimum-wage increase that different states enacted in different years, and the goal of estimating its effect on teen employment. Walking the tree: no cutoff (skip RD); more than one treated state (skip synthetic control); *yes* — it is a panel with treatment turning on over time, and the timing is *staggered*, so the arrow points to **Callaway–Sant'Anna** (Ch. 10). Employment is continuous, so no bottom box applies; the standard errors would be clustered by state. Four questions, one decision.

# Part I

# Setup and environment

# Chapter 1

# Getting started

## 1.1 Running the code

Each script is a self-contained `.do` file. Run one interactively with `do 01_basics.do`, or in batch from a shell:

```
/Applications/StataNow19SE/StataSE19.app/Contents/MacOS/stata-se -b do 00_run_all.do
```

The master script `00_run_all.do` runs all twelve in order. Outputs land in `output/` (figures, LaTeX/CSV tables, saved datasets) and logs in `logs/`.

## 1.2 Packages

Everything runs on base Stata except for a small set of community commands, all installed in one step with `do packages/install_packages.do`. Table 1.1 lists them and why each is needed; full provenance is in `packages/PACKAGES.md`. The two newest chapters add *no* dependencies: dynamic-panel GMM (Ch. 13) uses the built-in `xtabond`/`xtdpdsys`, and the efficient-GMM step (Ch. 8) the built-in `ivregress gmm` — Roodman's `xtabond2` is the popular community alternative for dynamic panels, but is not required here.

Table 1.1: Community (SSC/GitHub) packages used in the portfolio

| Package | Used in | Purpose |
|---|---|---|
| `reghdfe` | 04, 09 | Regression absorbing high-dimensional fixed effects |
| `ftools` | (dep) | Fast data operations; reghdfe dependency |
| `require` | (dep) | Version checker; reghdfe dependency |
| `estout` | 03, 04, 07, 08 | Publication tables (`esttab`) to LaTeX/CSV |
| `coefplot` | 04 | Coefficient / event-study plots |
| `ivreghdfe` | 07 | IV/2SLS with high-dimensional fixed effects |
| `ivreg2`, `ranktest` | (dep) | IV engine and weak-IV tests |
| `csdid` | 09 | Callaway–Sant'Anna staggered-adoption DiD |
| `drdid` | (dep) | Doubly-robust DiD; csdid dependency |
| `rdrobust` | 10 | Local-polynomial regression discontinuity |
| `rddensity`, `lpdensity` | 10 | RD manipulation / density test |
| `synth` | 11 | Abadie synthetic control |
| `synth_runner` | 11 | Placebo inference for synth (from GitHub) |
| `distinct` | (dep) | synth_runner dependency |

# Part II

# Empirical foundations

# Chapter 2

# Stata basics

The leap from Python or R is mostly syntactic: one dataset is held in memory at a time; `generate`/`replace` create and overwrite variables; `egen` handles group and window operations; and variable labels keep output self-documenting. The script below loads the bundled `auto` data, explores it, builds variables, and exports two figures.

## 2.1 The code

*The workflow.* Load a bundled dataset, inspect and summarise it, build and label variables with `generate`/`egen`, subset with `if`, draw a quick graph, and save a `.dta` file.

```
*==============================================================================
* 01_basics.do  ---  Stata fundamentals for applied economists
*------------------------------------------------------------------------------
* Run:  from the repository root, in Stata:  do 01_basics.do
* Batch: stata-se -b do 01_basics.do
*==============================================================================
version 17                      // write code against a fixed language version
clear all
set more off                    // don't pause for --more-- in long output
set linesize 90

capture log close
capture mkdir logs   // create the log folder if absent (gitignored)
log using "logs/01_basics.log", replace text

*------------------------------------------------------------------------------
* 1. Load data. sysuse loads datasets shipped with Stata (works offline).
*------------------------------------------------------------------------------
sysuse auto, clear              // 1978 automobile data (74 cars)
describe                        // structure: variables, types, labels  (~ R str())
codebook mpg foreign, compact   // quick audit of a few variables
list make price mpg in 1/5      // peek at rows  (~ head())

*------------------------------------------------------------------------------
* 2. Summarize / explore.
*------------------------------------------------------------------------------
summarize                       // means, sd, min, max for all numeric vars
summarize price, detail         // percentiles, skew/kurtosis
tabulate foreign                // one-way frequencies
tabulate foreign rep78, row     // two-way, with row percentages
bysort foreign: summarize mpg   // grouped summary (~ groupby().describe())

*------------------------------------------------------------------------------
* 3. Create & transform variables.
*    generate = new var; replace = overwrite; egen = "extended" generate.
*------------------------------------------------------------------------------
generate gpm = 1/mpg                        // gallons per mile
```

```
label variable gpm "Gallons per mile"
generate price_k = price/1000
generate byte expensive = price > 6000     // boolean -> 0/1 indicator
egen mpg_z = std(mpg)                       // standardized mpg
egen price_mean_by_for = mean(price), by(foreign)   // group mean (window fn)

* Conditional logic: replace ... if
generate size_class = "small"
replace  size_class = "large" if weight > 3500

* Missing values: Stata stores numeric missing as "." (sorts as +infinity).
codebook rep78                              // note the 5 missing values
count if missing(rep78)

*------------------------------------------------------------------------------
* 4. Labels make output self-documenting (a Stata habit worth keeping).
*------------------------------------------------------------------------------
label define yesno 0 "No" 1 "Yes"
label values expensive yesno
tabulate expensive

*------------------------------------------------------------------------------
* 5. A couple of graphs, exported to disk (no display needed in batch).
*------------------------------------------------------------------------------
histogram price, frequency title("Distribution of price")
graph export "output/01_hist_price.png", replace width(1200)

twoway (scatter price weight) (lfit price weight), ///
    title("Price vs. weight with linear fit") legend(off)
graph export "output/01_scatter_price_weight.png", replace width(1200)

*------------------------------------------------------------------------------
* 6. Save a cleaned copy (Stata's native .dta format).
*------------------------------------------------------------------------------
save "output/auto_clean.dta", replace

log close
display "01_basics.do finished OK"
```

## 2.2 The results

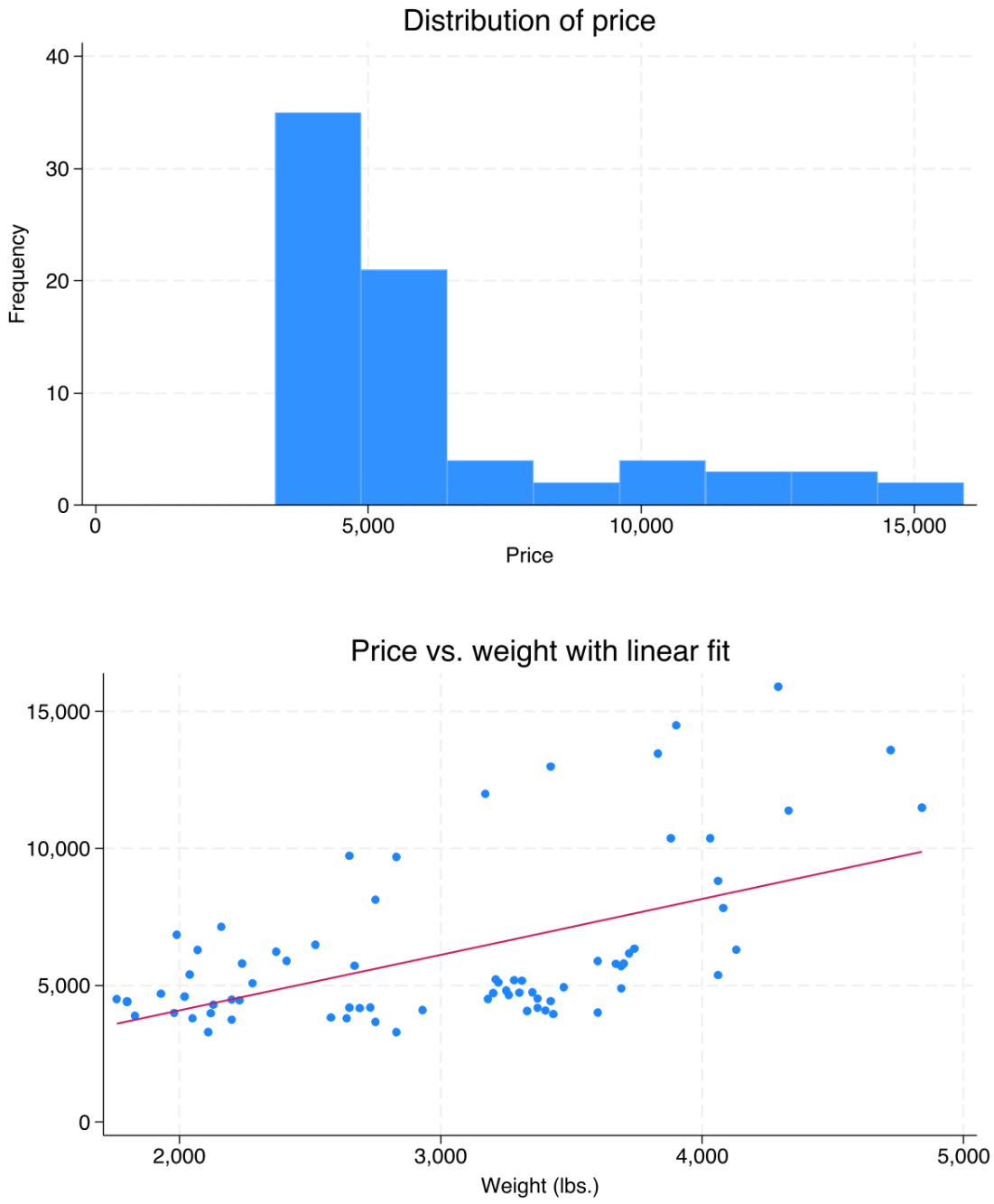



Figure 2.1: Distribution of price (top) and price vs. weight with a linear fit (bottom).

# Chapter 3

# Data management

The everyday plumbing of empirical work: joining datasets (`merge`), moving between wide and long layouts (`reshape`), aggregating (`collapse`), and handling dates, strings, and duplicates. These are the operations that consume most of the time in empirical work before any model runs.

The concept that is hardest to picture is `reshape`: the same data can sit in *long* form (one row per id–period) or *wide* form (one row per id, with the periods spread across columns). Figure 3.1 shows the two layouts and the commands that convert between them.

| id | year | income |
|---|---|---|
| 1 | 2010 | 100 |
| 1 | 2011 | 110 |
| 1 | 2012 | 130 |
| 2 | 2010 | 200 |
| 2 | 2011 | 220 |
| 2 | 2012 | 250 |

reshape wide →

← reshape long

| id | income2010 | income2011 | income2012 |
|---|---|---|---|
| 1 | 100 | 110 | 130 |
| 2 | 200 | 220 | 250 |

Figure 3.1: The same data in long form (left) and wide form (right); `reshape` converts between them.

**In practice.** Data cleaning is most of the work in any empirical project: real data arrive with inconsistent identifiers to harmonize, missing values to handle explicitly (Stata stores numeric missing as `.`, which sorts as $+\infty$), dates and strings to parse, and joins to verify — always check `_merge` after a `merge`.

## 3.1 The code

*The workflow.* Join two files with `merge`, reshape between wide and long with `reshape`, aggregate with `collapse`, and tidy dates, strings, and duplicate rows.

```
*==============================================================================
* 02_data_management.do  ---  merge, reshape, collapse, dates, strings, dups
*==============================================================================
version 17
clear all
set more off
set linesize 90
capture log close
capture mkdir logs   // create the log folder if absent (gitignored)
log using "logs/02_data_management.log", replace text

*------------------------------------------------------------------------------
* 1. MERGE  (joins). Split auto into two files, then merge 1:1 on a key.
```

```
*-------------------------------------------------------------------------------
sysuse auto, clear
keep make price mpg
save "output/_left.dta", replace

sysuse auto, clear
keep make weight foreign
save "output/_right.dta", replace

use "output/_left.dta", clear
merge 1:1 make using "output/_right.dta"
tabulate _merge                 // 3 = matched both; 1 = master only; 2 = using only
assert _merge==3                // sanity check the join is clean
drop _merge

* m:1 (many-to-one): a country-year panel picks up one GDP figure per country.
clear
input str3 country gdp
"USA" 21000
"CAN"  1700
"MEX"  1100
end
save "output/_gdp.dta", replace
clear
input str3 country int year
"USA" 2020
"USA" 2021
"CAN" 2020
"CAN" 2021
"MEX" 2020
"MEX" 2021
end
merge m:1 country using "output/_gdp.dta"   // each country's GDP fans out to its year rows
sort country year
list country year gdp _merge, sepby(country) noobs
drop _merge
erase "output/_gdp.dta"

* _merge values: an imperfect join shows 1 (master only), 2 (using only), 3 (matched).
clear
input str3 country pop
"USA" 331
"CAN"  38
"MEX" 126
end
save "output/_pop.dta", replace
clear
input str3 country gdp
"CAN"  1700
"MEX"  1100
"BRA"  1600
end
merge 1:1 country using "output/_pop.dta"   // master {CAN,MEX,BRA} vs using {USA,CAN,MEX}
tabulate _merge
sort _merge country
list country pop gdp _merge, sepby(_merge) noobs
drop _merge
erase "output/_pop.dta"

```

```
*------------------------------------------------------------------------------
* 2. RESHAPE  (wide <-> long), the panel-builder's workhorse.
*------------------------------------------------------------------------------
clear
input id year income
1 2010 100
1 2011 110
1 2012 130
2 2010 200
2 2011 220
2 2012 250
end
list, sepby(id)
reshape wide income, i(id) j(year)      // long -> wide  (income2010, income2011,...)
list
reshape long income, i(id) j(year)      // wide -> long  (back again)
list, sepby(id)

*------------------------------------------------------------------------------
* 3. COLLAPSE  (aggregate to a coarser level; ~ groupby().agg()).
*------------------------------------------------------------------------------
sysuse auto, clear
collapse (mean) mean_price=price mean_mpg=mpg ///
         (sd) sd_price=price (count) n=price, by(foreign)
list

* --- export the collapse summary as a table for the reference ---
capture file close cl
file open cl using "output/02_collapse_table.tex", write replace
file write cl "\begin{table}[htbp]\centering" _n
file write cl "\caption{\cmd{collapse}: the 74-car sample reduced to one row per origin}" _n
file write cl "\begin{tabular}{lrrrr}" _n "\toprule" _n
file write cl "Origin & Mean price & Mean mpg & SD price & \(N\) \\" _n "\midrule" _n
forvalues i = 1/`=_N' {
    local org = cond(foreign[`i']==0, "Domestic", "Foreign")
    file write cl "`org' & " %6.0f (mean_price[`i']) " & " %5.1f (mean_mpg[`i']) ///
        " & " %6.0f (sd_price[`i']) " & " %3.0f (n[`i']) " \\" _n
}
file write cl "\bottomrule" _n "\end{tabular}" _n "\end{table}" _n
file close cl

*------------------------------------------------------------------------------
* 4. DATES  (Stata stores dates as integers; display via formats).
*------------------------------------------------------------------------------
clear
input str10 raw
"2020-01-15"
"2020-06-30"
"2021-12-01"
end
generate edate = date(raw, "YMD")       // string -> numeric daily date
list raw edate, noobs                    // edate is a number: days since 1960-01-01
format edate %td
list raw edate, noobs                    // same column, now shown as a calendar date
generate year  = year(edate)
generate month = month(edate)
generate mdate = mofd(edate)            // monthly date
format mdate %tm
list
```

```
*------------------------------------------------------------------------------
* 5. STRING handling.
*------------------------------------------------------------------------------
sysuse auto, clear
generate brand = word(make, 1)          // first token of make
generate make_up = upper(make)
generate is_amc = strpos(make, "AMC")>0
generate spc = strpos(make, " ")        // position of the first space (0 if none)
list make brand make_up spc in 1/6, noobs
tabulate brand if is_amc

*------------------------------------------------------------------------------
* 6. DUPLICATES.
*------------------------------------------------------------------------------
clear
input id v
1 10
1 10
2 20
3 30
3 30
end
duplicates report                       // how many dup rows
duplicates drop                         // drop exact duplicates
list

* tidy up scratch files
erase "output/_left.dta"
erase "output/_right.dta"

log close
display "02_data_management.do finished OK"
```

## 3.2 The results

A `collapse` reduces the 74-car `auto` sample to one row per origin — foreign cars are pricier on average and noticeably more fuel-efficient in this 1978 sample:

Table 3.1: `collapse`: the 74-car sample reduced to one row per origin

| Origin | Mean price | Mean mpg | SD price | $N$ |
|---|---|---|---|---|
| Domestic | 6072 | 19.8 | 3097 | 52 |
| Foreign | 6385 | 24.8 | 2622 | 22 |

**Join quality.** After a `merge`, `_merge` labels every row — `1` master-only, `2` using-only, `3` matched — and an unmatched row receives missing (`.`) for the other file's variables. An imperfect join (master {CAN, MEX, BRA} against using {USA, CAN, MEX}):

```
+---------------------------------------+
| country   pop    gdp           _merge |
|---------------------------------------|
|     BRA     .   1600   Master only (1) |
```

```
|---------------------------------------|
|     USA   331      .    Using only (2) |
|---------------------------------------|
|     CAN    38   1700       Matched (3) |
|     MEX   126   1100       Matched (3) |
+---------------------------------------+
```

A clean join is all 3, which `assert _merge==3` enforces after every merge in the script.

**Dates and strings.** `date("2020-01-15","YMD")` returns the integer `21929` (days since 1 Jan 1960); `format %td` only changes the display to `15jan2020`. String helpers parse text in place — `word(make,1)` (first token), `upper(make)`, and `strpos(make," ")` (position of the first space):

```
+---------------------------------------------+
| make            brand         make_up   spc |
|---------------------------------------------|
| AMC Concord       AMC     AMC CONCORD     4 |
| AMC Pacer         AMC       AMC PACER     4 |
| AMC Spirit        AMC      AMC SPIRIT     4 |
| Buick Century   Buick   BUICK CENTURY     6 |
| Buick Electra   Buick   BUICK ELECTRA     6 |
| Buick LeSabre   Buick   BUICK LESABRE     6 |
+---------------------------------------------+
```

# Chapter 4

# Linear regression

↪ *the method, explained:* *Appendix B.1*

## 4.1 The method

The workhorse model is

$$y_i = \beta_0 + \beta_1 x_{1i} + \cdots + \beta_k x_{ki} + \varepsilon_i, \qquad \hat{\beta} = (X'X)^{-1}X'y,$$

with White (1980) heteroskedasticity-robust (HC1) standard errors,

$$\widehat{\mathrm{Var}}(\hat{\beta}) = (X'X)^{-1}\Big(\sum_i \hat{\varepsilon}_i^2\, x_i x_i'\Big)(X'X)^{-1}.$$

Factor-variable notation (`i.`, `c.`, `##`) builds dummies and interactions; `margins` reports the average marginal effect

$$\overline{\mathrm{ME}} = \frac{1}{n}\sum_i \frac{\partial\, \widehat{E}[y \mid x_i]}{\partial x},$$

the interpretable quantity once interactions are present. Diagnostics: a joint $F$ test, the Breusch–Pagan test for heteroskedasticity, and variance inflation factors.

**In practice.** OLS underlies hedonic price regressions (housing or car prices on characteristics), Mincer (1974) earnings equations (log wage on schooling and experience), and demand estimation — wherever the target is a conditional mean or a descriptive association.

## 4.2 The code

*The workflow.* Fit OLS with robust standard errors and factor-variable interactions, run post-estimation tests, and export a publication-ready table with `esttab`.

```
*==============================================================================
* 03_regression.do  ---  OLS, robust SE, factor variables, interactions,
*                        marginal effects, postestimation, publication tables
*==============================================================================
version 17
clear all
set more off
set linesize 90
capture log close
capture mkdir logs   // create the log folder if absent (gitignored)
log using "logs/03_regression.log", replace text

sysuse auto, clear
```

```
generate weight_t = weight/1000          // weight in 1000s of lbs
label variable weight_t "Weight (1000 lbs)"
label variable mpg "Mileage (mpg)"
label variable foreign "Foreign"

*-------------------------------------------------------------------------------
* 1. OLS with heteroskedasticity-robust standard errors.
*    i.foreign = factor (dummy); c.var = continuous; ## = full interaction.
*-------------------------------------------------------------------------------
eststo clear
eststo m1: regress price mpg weight_t, vce(robust)
eststo m2: regress price mpg weight_t i.foreign, vce(robust)
eststo m3: regress price c.mpg##c.weight_t i.foreign, vce(robust)

* Print a comparison table to the log.
esttab m1 m2 m3, se star(* 0.10 ** 0.05 *** 0.01) b(%9.2f) ///
    r2 label mtitles("Base" "+Foreign" "+Interaction") ///
    title("Determinants of car price")

* Export the same table to LaTeX and CSV (publication-ready deliverables).
esttab m1 m2 m3 using "output/03_regression_table.tex", replace ///
    se star(* 0.10 ** 0.05 *** 0.01) b(%9.2f) r2 label booktabs ///
    mtitles("Base" "+Foreign" "+Interaction") ///
    title("Determinants of car price\label{tab:price}")
esttab m1 m2 m3 using "output/03_regression_table.csv", replace ///
    se star(* 0.10 ** 0.05 *** 0.01) b(%9.3f) r2 label plain

*-------------------------------------------------------------------------------
* 2. Marginal effects (margins) --- interpret models on the outcome scale.
*-------------------------------------------------------------------------------
quietly regress price c.mpg##c.weight_t i.foreign, vce(robust)
margins, dydx(mpg)                       // average marginal effect of mpg
margins foreign                          // predicted price by foreign status
margins, dydx(mpg) at(weight_t=(2 3 4))  // how the mpg slope varies with weight

*-------------------------------------------------------------------------------
* 3. Postestimation diagnostics.
*-------------------------------------------------------------------------------
quietly regress price mpg weight_t i.foreign
test mpg weight_t                        // joint F-test (both = 0)
estat hettest                            // Breusch-Pagan heteroskedasticity test
estat vif                                // variance inflation (multicollinearity)
predict yhat                             // fitted values
predict ehat, residuals                  // residuals
rvfplot, yline(0) title("Residual vs. fitted")
graph export "output/03_rvfplot.png", replace width(1200)

log close
display "03_regression.do finished OK"
```

## 4.3 The results

The three columns build the specification up step by step. In Stata's factor-variable notation `i.foreign` is the indicator $\mathbf{1}[\texttt{foreign} = 1]$ and `c.mpg##c.weight_t` expands to *both* main effects

and their product, so the columns estimate

$$(1)\quad \texttt{price}_i = \beta_0 + \beta_1\,\texttt{mpg} + \beta_2\,\texttt{weight\_t} + \varepsilon_i,$$
$$(2)\quad \texttt{price}_i = \beta_0 + \beta_1\,\texttt{mpg} + \beta_2\,\texttt{weight\_t} + \beta_3\,\mathbf{1}[\texttt{foreign}] + \varepsilon_i,$$
$$(3)\quad \texttt{price}_i = \beta_0 + \beta_1\,\texttt{mpg} + \beta_2\,\texttt{weight\_t} + \beta_3\,(\texttt{mpg}\cdot\texttt{weight\_t}) + \beta_4\,\mathbf{1}[\texttt{foreign}] + \varepsilon_i,$$

each with heteroskedasticity-robust (HC1) standard errors (`vce(robust)`). In this 1978 sample, weight and foreign manufacture are strong positive predictors of price and the mileage–weight interaction is small. These are *descriptive associations* in a teaching dataset, not causal effects—features the model omits (trim, brand, engine quality) plausibly drive both a car's weight and its price, so a coefficient describes how price and that regressor co-move, not what would happen if it were changed.

Table 4.1: Determinants of car price

| | (1) Base | (2) +Foreign | (3) +Interaction |
|---|---|---|---|
| Mileage (mpg) | -49.51 | 21.85 | 292.83* |
| | (95.81) | (80.75) | (153.90) |
| Weight (1000 lbs) | 1746.56** | 3464.71*** | 5382.75*** |
| | (777.84) | (777.62) | (1084.04) |
| Domestic | | 0.00 | 0.00 |
| | | (.) | (.) |
| Foreign | | 3673.06*** | 3369.81*** |
| | | (664.94) | (757.93) |
| Mileage (mpg) × Weight (1000 lbs) | | | -118.91 |
| | | | (85.68) |
| Constant | 1946.07 | -5853.70 | -10105.04*** |
| | (4213.79) | (3873.72) | (3212.97) |
| Observations | 74 | 74 | 74 |
| $R^2$ | 0.293 | 0.500 | 0.524 |

Standard errors in parentheses

* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

The residual-versus-fitted plot fans out as the fitted value grows — visible heteroskedasticity, which is precisely why the standard errors are heteroskedasticity-robust.

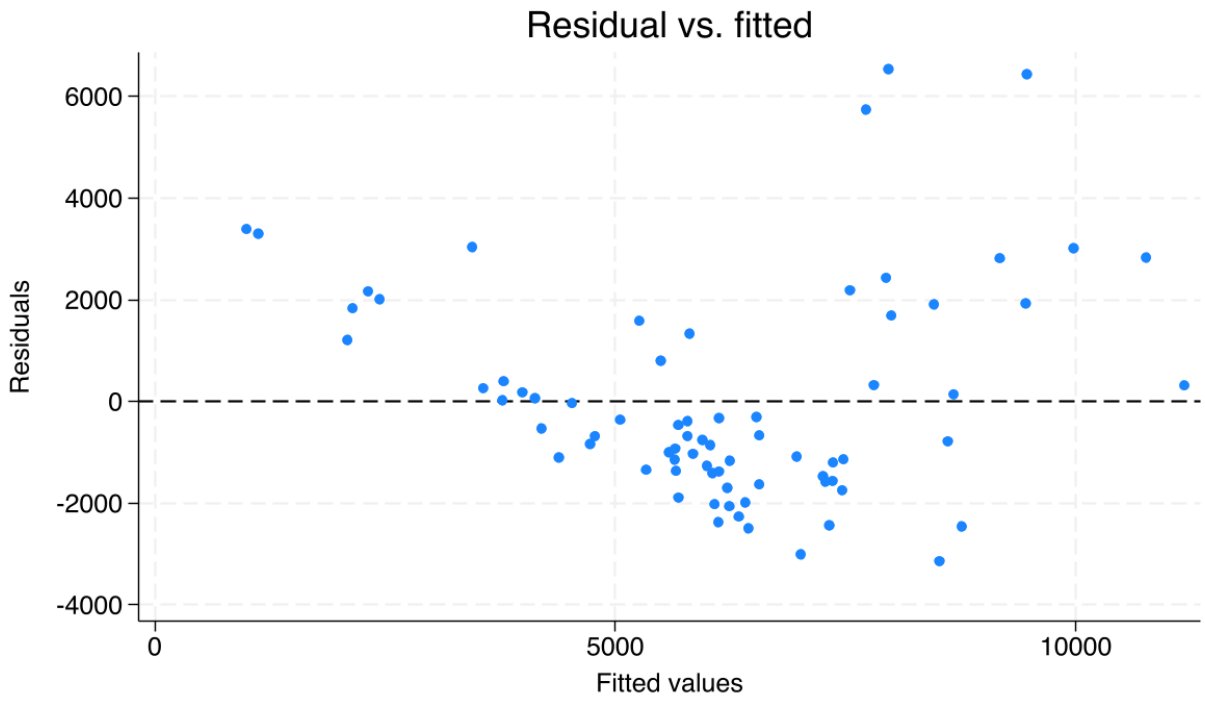



Figure 4.1: Residuals versus fitted values — the spread widens, indicating heteroskedasticity.

# Chapter 5

# Panel data, DiD, and event studies

↪ *the method, explained:* *Appendix B.2*

## 5.1 The method

With repeated observations on units over time, two-way fixed effects difference-in-differences is

$$y_{it} = \alpha_i + \gamma_t + \beta\,(\text{treat}_i \times \text{post}_t) + \varepsilon_{it},$$

where $\alpha_i$ absorbs unit level differences, $\gamma_t$ common shocks, and $\beta$ is the DiD estimand. The dynamic version traces the effect by event time $k$:

$$y_{it} = \alpha_i + \gamma_t + \sum_{k \neq -1} \beta_k\, \mathbf{1}[\text{treat}_i]\, \mathbf{1}[t - t_i^* = k] + \varepsilon_{it}.$$

Here $t_i^*$ is unit $i$'s treatment date, so $k = t - t_i^*$ is *event time* (periods since onset) and the product $\mathbf{1}[\text{treat}_i]\,\mathbf{1}[t - t_i^* = k]$ switches on only for a treated unit in the period $k$ steps from its own treatment; $\beta_k$ is the effect at that horizon. The sum skips $k = -1$ because one event-time coefficient is *not estimable* — the full set of lead/lag dummies is collinear with the fixed effects, so one period must serve as the dropped *baseline*. By convention that period is $k = -1$, the one *just before* treatment: dropping it fixes $\beta_{-1} \equiv 0$, and every other $\beta_k$ is then read *relative to* the eve of treatment. So the pre-treatment coefficients ($k < -1$) should sit near zero — a visual parallel-trends check — while the post-treatment ones ($k \geq 0$) trace out the dynamic effect. Keeping the leads in the sum — writing $k \neq -1$ rather than only $k \geq 0$ — is the whole point: with no treatment yet they ought to be zero, so they serve as a placebo test, the built-in pre-trend check an event study provides over a single DiD coefficient. Standard errors cluster by unit.

**In practice.** The two-by-two DiD is the template for policy evaluation — Card and Krueger's (1994) New Jersey–Pennsylvania minimum-wage study is the canonical example — and the event-study version traces the dynamics of a policy that switches on at a known date.

## 5.2 The code

*The workflow.* Declare the panel, estimate difference-in-differences by two-way fixed effects with `reghdfe`, and trace the dynamic event study with `coefplot`.

```
*===============================================================================
* 04_panel_did_eventstudy.do  ---  panel FE, difference-in-differences,
*                                  two-way FE with reghdfe, dynamic event study
*-------------------------------------------------------------------------------
* Builds a SYNTHETIC firm-by-year panel (fully reproducible, no external data),
* then runs the canonical causal-inference workflow of applied empirical work.
```

```
*==============================================================================
version 17
clear all
set more off
set linesize 90
set seed 90210
capture log close
capture mkdir logs   // create the log folder if absent (gitignored)
log using "logs/04_panel_did_eventstudy.log", replace text

*------------------------------------------------------------------------------
* 1. Simulate a panel: 150 firms x 12 years (2010-2021).
*    Treated firms (half) get a +1.5 effect starting in 2015 that grows over time.
*------------------------------------------------------------------------------
set obs 150
generate firm    = _n
generate fe_firm = rnormal(0, 2)          // firm fixed effect
generate byte treat = mod(firm, 2)==0     // half the firms are treated
expand 12
bysort firm: generate year = 2009 + _n    // 2010..2021
generate byte post = year >= 2015         // treatment turns on in 2015
generate rel = year - 2015                // event time relative to treatment

generate year_fe = 0.3*(year-2015) + 0.5*sin(year)   // common time shocks
generate tau = 0
replace  tau = 0.5*(rel+1) if treat==1 & year>=2015  // dynamic treatment effect
generate y = 3 + fe_firm + year_fe + tau + rnormal(0,1)
label variable y "Outcome"

xtset firm year                           // declare panel structure
xtdescribe

*------------------------------------------------------------------------------
* 2. Pooled OLS vs. fixed-effects DiD (the 2x2 interaction).
*------------------------------------------------------------------------------
eststo clear
eststo ols: regress y i.treat##i.post, vce(cluster firm)
eststo fe:  xtreg   y i.treat##i.post, fe vce(cluster firm)
* Under firm FE the time-invariant treat main effect drops; the DiD estimate
* is the 1.treat#1.post coefficient (true value = average post effect).

*------------------------------------------------------------------------------
* 3. Two-way fixed effects with reghdfe (absorb firm AND year FE).
*------------------------------------------------------------------------------
eststo twfe: reghdfe y 1.treat#1.post, absorb(firm year) vce(cluster firm)

esttab ols fe twfe, se star(* 0.10 ** 0.05 *** 0.01) b(%9.3f) ///
    mtitles("Pooled OLS" "Firm FE" "Two-way FE") ///
    keep(1.treat#1.post) label title("Difference-in-differences estimates")
esttab ols fe twfe using "output/04_did_table.tex", replace booktabs ///
    se star(* 0.10 ** 0.05 *** 0.01) b(%9.3f) keep(1.treat#1.post) label ///
    mtitles("Pooled OLS" "Firm FE" "Two-way FE") title("DiD estimates")

*------------------------------------------------------------------------------
* 4. Dynamic event study: treatment-effect path by event time.
*    The full set of event-time dummies sums to the (time-invariant) treated
*    indicator, which the firm FE absorbs, so one dummy must be dropped. We omit
*    t = -1 (the eve of treatment) as the single reference period, so every other
*    coefficient is read relative to it. Leads should sit near zero (no pre-trend);
```

```
*    lags trace the dynamic effect.
*-------------------------------------------------------------------------------
forvalues k = -5/6 {
    if `k'==-1 continue                    // omit t = -1: the single reference period
    local j = `k' + 6                      // 1..12 index -> a legal variable name
    generate evt`j' = (treat==1 & rel==`k')
    label variable evt`j' "t = `k'"
}
reghdfe y evt*, absorb(firm year) vce(cluster firm)

coefplot, keep(evt*) vertical yline(0) xline(4.5, lpattern(dash)) ///
    coeflabels(evt1="-5" evt2="-4" evt3="-3" evt4="-2" evt6="0" evt7="1" evt8="2" ///
               evt9="3" evt10="4" evt11="5" evt12="6") ///
    title("Event-study estimates (ref: t = -1)") ///
    xtitle("Event time (years relative to treatment)") ///
    ytitle("Effect on outcome")
graph export "output/04_event_study.png", replace width(1400)

log close
display "04_panel_did_eventstudy.do finished OK"
```

## 5.3 The results

The three columns apply the same DiD interaction `treat` × `post` under progressively richer fixed effects. In Stata's factor-variable notation `i.treat##i.post` expands to both main effects *and* their product, while `1.treat#1.post` is the product alone, so the columns estimate

$$
\begin{aligned}
(1) \quad & y_{it} = \beta_0 + \beta_1\,\texttt{treat}_i + \beta_2\,\texttt{post}_t + \beta\,(\texttt{treat}_i \times \texttt{post}_t) + \varepsilon_{it}, \\
(2) \quad & y_{it} = \alpha_i + \beta_2\,\texttt{post}_t + \beta\,(\texttt{treat}_i \times \texttt{post}_t) + \varepsilon_{it}, \\
(3) \quad & y_{it} = \alpha_i + \gamma_t + \beta\,(\texttt{treat}_i \times \texttt{post}_t) + \varepsilon_{it},
\end{aligned}
$$

for (1) pooled OLS, (2) firm fixed effects (`xtreg`, `fe`), and (3) two-way firm-and-year fixed effects (`reghdfe`). Each richer specification absorbs a main effect: the firm effect $\alpha_i$ soaks up the time-invariant $\texttt{treat}_i$, so column 2 drops it, and the year effect $\gamma_t$ soaks up $\texttt{post}_t$, so column 3 drops it too. That is why the code switches to the bare interaction `1.treat#1.post` in column 3 — the main effects are now collinear with the fixed effects and cannot be estimated separately. The DiD estimand $\beta$ survives throughout, and the simulated post-treatment effect (about 1.5) is recovered by all three columns (Table). The event study shows a flat pre-trend and a rising dynamic effect (Figure).

Table 5.1: DiD estimates

| | (1) Pooled OLS | (2) Firm FE | (3) Two-way FE |
|---|---|---|---|
| treat=1 × post=1 | 1.973*** | 1.973*** | 1.973*** |
| | (0.104) | (0.104) | (0.104) |
| Observations | 1800 | 1800 | 1800 |

Standard errors in parentheses

* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

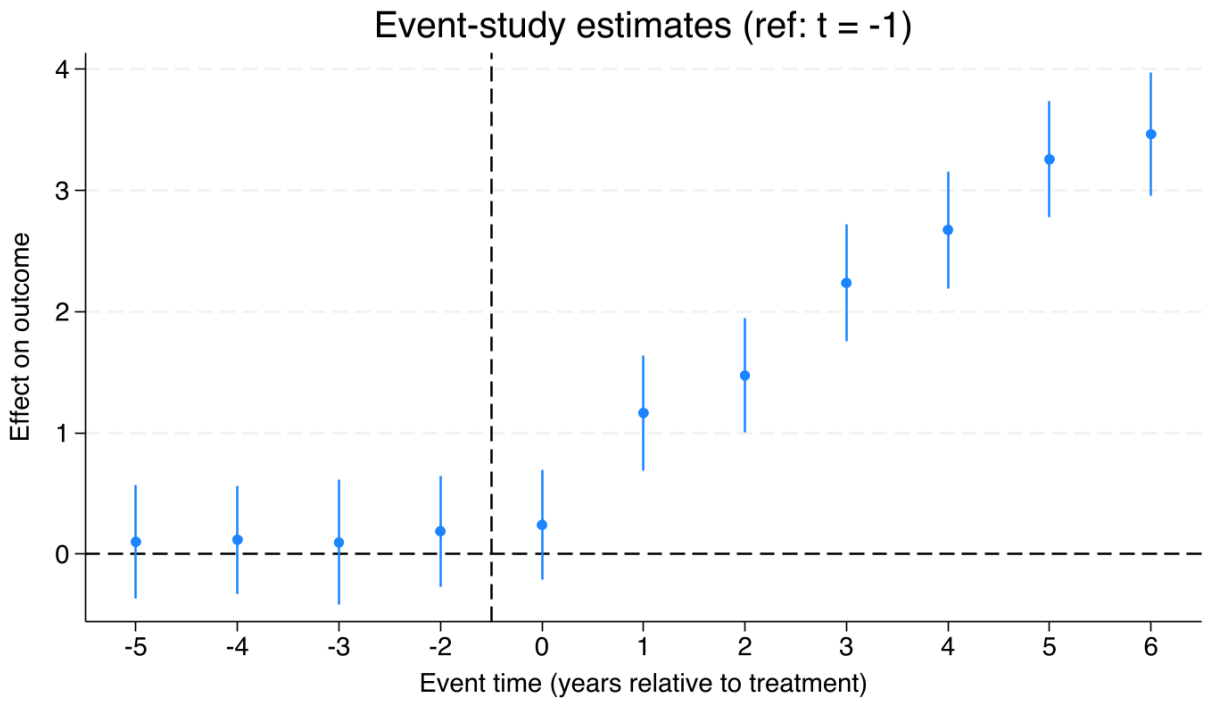



Figure 5.1: Event-study estimates: flat, insignificant leads; rising lags.

# Chapter 6

# Time series

↪ *the method, explained:* *Appendix B.3*

## 6.1 The method

A first-order autoregression with a level break,

$$y_t = c + \phi\, y_{t-1} + \delta\, \mathbf{1}[t \geq t^*] + \varepsilon_t,$$

declared with `tsset` so that the lag (`L.`), difference (`D.`), and lead (`F.`) operators work. Newey and West (1987) HAC standard errors are robust to serial correlation up to a chosen lag. The simulated series uses $\phi = 0.6$ with a $+3$ shift in 2015m1.

**In practice.** Persistence and HAC standard errors are everywhere in macro and finance — modelling inflation or output dynamics, volatility clustering in asset returns, and any regression whose residuals are serially correlated, where Newey–West is the default fix.

## 6.2 The code

*The workflow.* Declare the series with `tsset`, apply lag and difference operators, inspect the autocorrelation, and fit a lagged regression with Newey–West (HAC) standard errors.

```
*===============================================================================
* 05_timeseries.do  ---  tsset, lag/diff operators, autocorrelation,
*                        HAC (Newey-West) SE, and a simple event/break window
*-------------------------------------------------------------------------------
* Simulates a monthly AR(1) series with a level shift in 2015m1.
*===============================================================================
version 17
clear all
set more off
set linesize 90
set seed 4321
capture log close
capture mkdir logs    // create the log folder if absent (gitignored)
log using "logs/05_timeseries.log", replace text

*-------------------------------------------------------------------------------
* 1. Build a monthly time series and declare it with tsset.
*-------------------------------------------------------------------------------
set obs 120
generate t = _n
generate mdate = tm(2010m1) + t - 1      // Stata monthly date
format mdate %tm
tsset mdate                              // declare time-series structure
```

```
24
25 generate eps = rnormal(0,1)
26 generate y = eps in 1
27 replace  y = 0.6*L.y + eps in 2/L        // AR(1): recursion fills in order
28 replace  y = y + 3 if mdate >= tm(2015m1)  // structural level shift
29 label variable y "Simulated series"
30
31 *-------------------------------------------------------------------------------
32 * 2. Time-series operators: L. (lag), D. (difference), F. (lead).
33 *-------------------------------------------------------------------------------
34 generate dy   = D.y                       // first difference
35 generate y_l1 = L.y                       // first lag
36 summarize y dy
37
38 *-------------------------------------------------------------------------------
39 * 3. Plots: the series and its autocorrelation function.
40 *-------------------------------------------------------------------------------
41 tsline y, title("Monthly series with a 2015 level shift") ///
42     tline(2015m1, lpattern(dash))
43 graph export "output/05_tsline.png", replace width(1400)
44
45 ac y, title("Autocorrelation function")
46 graph export "output/05_acf.png", replace width(1200)
47
48 *-------------------------------------------------------------------------------
49 * 4. Regression with a lag + HAC (Newey-West) standard errors.
50 *-------------------------------------------------------------------------------
51 newey y L.y, lag(3)                       // robust to serial correlation up to 3 lags
52
53 *-------------------------------------------------------------------------------
54 * 5. Test the level break with an indicator (a mini event study).
55 *-------------------------------------------------------------------------------
56 generate byte post = mdate >= tm(2015m1)
57 regress y L.y i.post, vce(robust)        // coefficient on post = estimated jump
58
59 log close
60 display "05_timeseries.do finished OK"
```

## 6.3 The results

The two regressions map directly onto the time-series operators that `tsset` unlocks:

$$\begin{aligned} \texttt{newey y L.y, lag(3)}: &\quad y_t = c + \phi\, y_{t-1} + \varepsilon_t, \\ \texttt{regress y L.y i.post}: &\quad y_t = c + \phi\, y_{t-1} + \delta\, \mathbf{1}[t \geq \texttt{2015m1}] + \varepsilon_t, \end{aligned}$$

where the lag operator `L.y` is $y_{t-1}$ and `i.post` is the break indicator $\mathbf{1}[t \geq \texttt{2015m1}]$. The first fits the pure AR(1) with Newey–West HAC standard errors (robust to serial correlation up to three lags); the second adds the level-shift dummy, whose coefficient $\delta$ is the estimated jump. The simulated series is visibly persistent — its autocorrelation function decays slowly rather than dropping to zero — and the $+3$ level shift at 2015m1 is recovered by the break indicator ($\hat{\delta} \approx 3$).

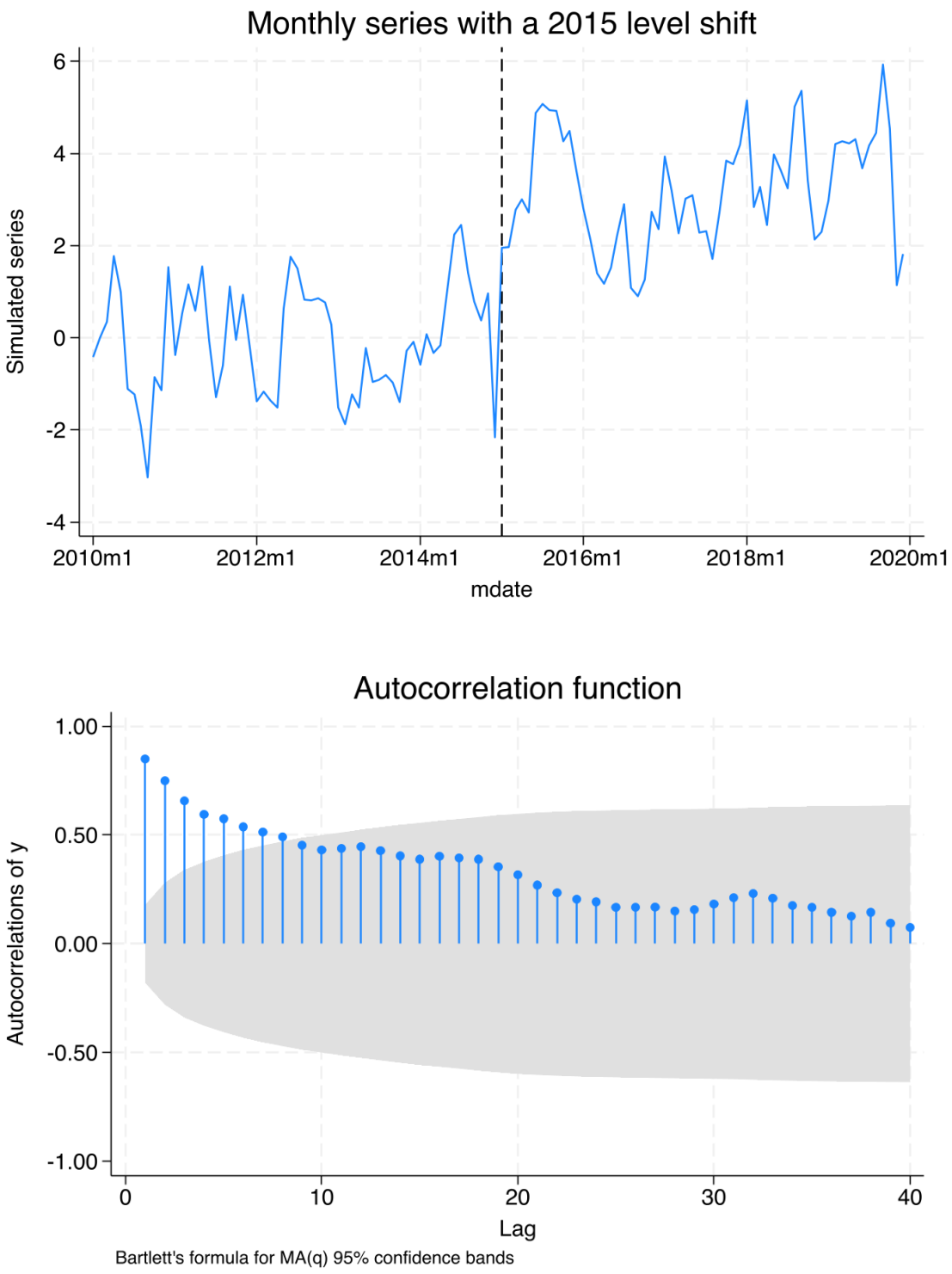



Figure 6.1: The simulated series with its 2015 level shift (top) and its autocorrelation function (bottom).

# Chapter 7

# Importing public data

Most empirical projects begin by getting public data into Stata. This script first round-trips a CSV — writing Stata's built-in `auto` data out and reading it straight back, to demonstrate the `export`/`import delimited` commands — then imports a *real* FRED series, the U.S. unemployment rate (`UNRATE`), *live* through Stata's native importer (`import fred`): it parses the date, declares a monthly time series, and plots it. The importer needs a free key, set once with `set fredkey` (which keeps it in the Stata config, not this script); a previously downloaded CSV is the offline fallback.

**Setting up the FRED key.** A one-time setup, so `import fred` works:

1. Register for a free API key at `fredaccount.stlouisfed.org/apikeys`.
2. In Stata, run `set fredkey YOUR_KEY, permanently` *once*. The key is stored in the Stata configuration — it persists across sessions and never appears in a do-file.
3. Thereafter `import fred UNRATE` (or any series ID, e.g. `GDPC1`, `CPIAUCSL`) pulls that series live; rerun step 2 to replace the key.

```
set fredkey YOUR_KEY, permanently  // once: free key at fredaccount.stlouisfed.org
import fred UNRATE, clear           // then pull any series live from FRED's API
```

**In practice.** The same pattern pulls in most public macro and finance data — FRED (Federal Reserve series), SEC EDGAR (filings), the World Bank and IMF, the BLS, and exchange or market feeds — which is where almost every empirical project begins.

## 7.1 The code

*The workflow.* Import and export delimited files, and pull a real public series (unemployment, from FRED) into Stata.

```
*==============================================================================
* 06_import_public_data.do  ---  getting public data INTO Stata
*------------------------------------------------------------------------------
* Empirical work often requires scraping publicly available data and
* cleaning it. This shows the three routes used in practice.
*==============================================================================
version 17
clear all
set more off
set linesize 90
capture log close
capture mkdir logs   // create the log folder if absent (gitignored)
log using "logs/06_import_public_data.log", replace text

```

```
*------------------------------------------------------------------------------
* (A) Round-trip a CSV to demonstrate import/export delimited (the CSV workhorses).
*     We write the built-in 'auto' data out and read it straight back --- auto is just
*     a throwaway here, the point is the file I/O, not the data. varnames(1)=header.
*------------------------------------------------------------------------------
sysuse auto, clear
keep make price mpg foreign
export delimited using "output/_cars.csv", replace
clear
import delimited "output/_cars.csv", varnames(1) clear
describe
list in 1/3
erase "output/_cars.csv"

*------------------------------------------------------------------------------
* (B) Import a REAL public series: the U.S. unemployment rate (FRED: UNRATE).
*     PRIMARY route --- Stata's native FRED importer, a live API call (the most
*     reproducible: no manual download step). It needs a free key, set ONCE with
*         set fredkey YOUR_KEY, permanently   (free key: fredaccount.stlouisfed.org)
*     The key then lives in the Stata config, never in this script. With no key
*     set we fall back to a CSV previously downloaded from FRED, so this always runs.
*------------------------------------------------------------------------------
capture import fred UNRATE, clear
if _rc == 0 {
    display "Live FRED API pull succeeded: " _N " observations."
    rename UNRATE unrate                    // the native importer keeps FRED's case
    generate mdate = mofd(daten)            // import fred already provides daten (%td)
}
else {
    display "No FRED key configured (rc = " _rc "); using the bundled CSV instead."
    import delimited "data/UNRATE.csv", varnames(1) clear
    rename observation_date datestr
    generate daten = date(datestr, "YMD")    // parse the YYYY-MM-DD string ourselves
    format daten %td
    generate mdate = mofd(daten)            // collapse to monthly
}
format mdate %tm
tsset mdate
summarize unrate
list datestr unrate in 1/3
list datestr unrate in -3/L                 // most recent observations

tsline unrate, title("U.S. unemployment rate (FRED: UNRATE)") ///
    ytitle("Percent") xtitle("")
graph export "output/06_unrate.png", replace width(1400)
save "output/fred_unrate.dta", replace
display "FRED UNRATE imported: " _N " monthly observations"

* To pull several series at once over a date window:
*   import fred UNRATE GDPC1, daterange(2000-01-01 .) aggregate(monthly)

log close
display "06_import_public_data.do finished OK"
```

## 7.2 The results

This is the one genuinely real series in the book — pulled *live* from FRED's API by `import fred`:

```
. import fred UNRATE, clear
Series ID                    Nobs    Date range                Frequency
UNRATE                       940     1948-01-01 to 2026-05-01  Monthly

Live FRED API pull succeeded: 941 observations.
```

The unemployment rate climbs in every postwar recession and spikes to nearly 15% in April 2020 at the onset of the COVID-19 pandemic, then falls back — a reminder that importing and cleaning data is the gateway to any empirical analysis.

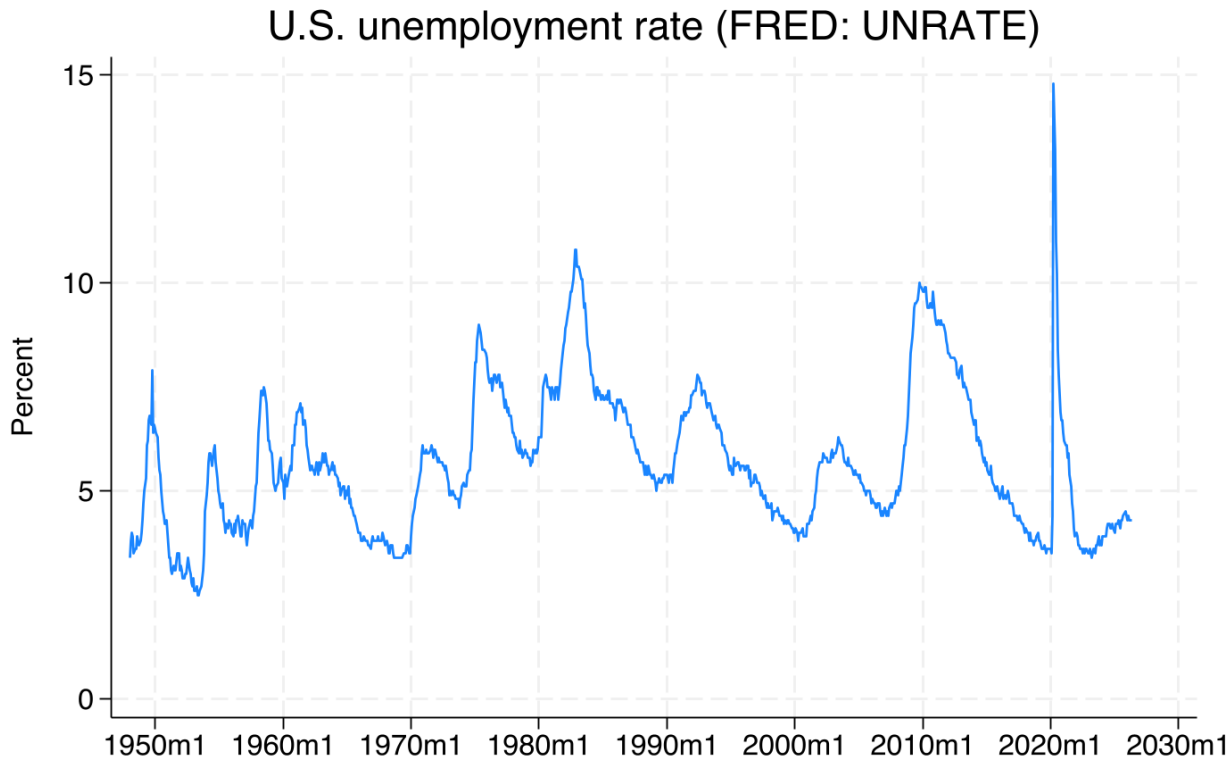



Figure 7.1: U.S. unemployment rate, 1948–2026 (FRED: UNRATE), imported and plotted in Stata.

# Part III

# The causal-inference toolkit

# Chapter 8

# Instrumental variables (2SLS)

↪ *the method, explained:* *Appendix B.4*

## 8.1 The method

When a regressor is correlated with the error, $\mathrm{Cov}(x_i, \varepsilon_i) \neq 0$, OLS is inconsistent. A valid instrument $z$ is relevant ($\mathrm{Cov}(z, x) \neq 0$) and exogenous ($\mathrm{Cov}(z, \varepsilon) = 0$) — think of a lottery that shifts who gets treated but is otherwise unrelated to the outcome. With one endogenous regressor and two instruments the model is

$$\underbrace{\mathtt{y} = \beta_0 + \beta_1\,\mathtt{x} + \varepsilon}_{\text{structural equation}}, \qquad \underbrace{\mathtt{x} = \pi_0 + \pi_1\,\mathtt{z1} + \pi_2\,\mathtt{z2} + v}_{\text{first stage}},$$

written `(x = z1 z2)` in Stata. Two-stage least squares is

$$\hat{\beta}_{2SLS} = (X'P_Z X)^{-1} X'P_Z y, \qquad P_Z = Z(Z'Z)^{-1}Z';$$

this is GMM with the weight matrix $(Z'Z)^{-1}$. Efficient two-step GMM (`ivregress gmm`) instead uses the optimal heteroskedasticity-robust weight matrix, gaining efficiency when the errors are heteroskedastic — and coinciding with 2SLS when they are not. `ivreghdfe` adds high-dimensional fixed effects. The standard checks are the first-stage $F$ (weak instruments; Staiger and Stock, 1997), the Durbin–Wu–Hausman test (endogeneity), and the Hansen $J$ (overidentification).

**In practice.** Famous instruments include quarter of birth for the returns to schooling (Angrist and Krueger 1991), the Vietnam draft lottery for veteran earnings (Angrist 1990), and cost shifters for demand estimation; modern work often uses judge- or examiner-leniency designs.

## 8.2 The code

*The workflow.* Contrast biased OLS with 2SLS, run the weak-instrument, endogeneity, and over-identification diagnostics, and refit with high-dimensional fixed effects via `ivreghdfe`.

```
*==============================================================================
* 07_iv_2sls.do  ---  instrumental variables / 2SLS, with diagnostics, and the
*                     high-dimensional-FE version (ivreghdfe)
*------------------------------------------------------------------------------
* Simulates an endogenous regressor x correlated with an unobserved confounder
* u; two valid instruments z1, z2 (independent of u). True slope on x is 2.0,
* so OLS should be biased and 2SLS should recover ~2.0.
*==============================================================================
version 17
clear all
set more off
```

```
set linesize 90
set seed 1234
capture log close
capture mkdir logs   // create the log folder if absent (gitignored)
log using "logs/07_iv_2sls.log", replace text

set obs 5000
generate id = _n
generate z1 = rnormal()                  // instrument 1
generate z2 = rnormal()                  // instrument 2
generate u  = rnormal()                  // unobserved confounder
generate x  = 0.6*z1 + 0.5*z2 + 0.6*u + rnormal()   // endogenous regressor
generate e  = 0.6*u + rnormal()          // structural error (corr with x via u)
generate grp = mod(id,50) + 1            // 50 groups, for the FE demo
bysort grp (id): generate gfe = rnormal() if _n==1
bysort grp (id): replace   gfe = gfe[1]
generate y  = 1 + 2*x + e + gfe          // TRUE slope on x = 2

*------------------------------------------------------------------------------
* 1. OLS (biased) vs. 2SLS (base Stata: ivregress).
*------------------------------------------------------------------------------
eststo clear
eststo ols:  regress y x, vce(robust)                       // biased upward
eststo tsls: ivregress 2sls y (x = z1 z2), vce(robust) first // first-stage shown

*------------------------------------------------------------------------------
* 2. IV diagnostics --- the standard validity checks for any IV design.
*------------------------------------------------------------------------------
estat firststage          // weak-instrument F (rule of thumb: > 10)
estat endogenous          // Durbin-Wu-Hausman: is x actually endogenous?
estat overid              // over-identification test (2 instruments, 1 endog var)

*------------------------------------------------------------------------------
* 2b. Efficient (two-step) GMM. 2SLS is GMM with a fixed weight matrix; `gmm' uses
*     the optimal heteroskedasticity-robust weight matrix, so it is more efficient
*     when errors are heteroskedastic. With homoskedastic errors (as here) it
*     essentially coincides with 2SLS; the over-id test is Hansen's J.
*------------------------------------------------------------------------------
ivregress gmm y (x = z1 z2), vce(robust)

*------------------------------------------------------------------------------
* 3. IV with high-dimensional fixed effects (ivreghdfe = ivreg2 + reghdfe).
*    Absorbs the group FE that base ivregress would need as dummies.
*------------------------------------------------------------------------------
eststo ivfe: ivreghdfe y (x = z1 z2), absorb(grp) vce(robust)

*------------------------------------------------------------------------------
* 4. Compare the slope on x across estimators.
*------------------------------------------------------------------------------
esttab ols tsls ivfe, b(%9.3f) se star(* 0.10 ** 0.05 *** 0.01) ///
    keep(x) mtitles("OLS (biased)" "2SLS" "IV + FE") ///
    title("True slope on x = 2.0: OLS is biased, IV recovers it")
esttab ols tsls ivfe using "output/07_iv_table.tex", replace booktabs ///
    b(%9.3f) se star(* 0.10 ** 0.05 *** 0.01) keep(x) ///
    mtitles("OLS" "2SLS" "IV+FE") title("IV vs OLS")

*------------------------------------------------------------------------------
* 5. Export an IV-diagnostics table for the reference (re-fit quietly first; the
*    last active estimates above are the IV+FE model, not the plain 2SLS).
```

```
*-------------------------------------------------------------------------------
quietly regress x z1 z2, vce(robust)
test z1 z2
local fsF = r(F)
quietly ivregress 2sls y (x = z1 z2), vce(robust)
quietly estat endogenous
local dwh  = r(r_score)
local dwhp = r(p_r_score)
quietly estat overid
local oj   = r(score)
local ojp  = r(p_score)
local dwhpt = cond(`dwhp'<0.001, "\(<\)0.001", string(`dwhp',"%5.3f"))
local ojpt  = string(`ojp',"%5.3f")

capture file close it
file open it using "output/07_iv_diag_table.tex", write replace
file write it "\begin{table}[htbp]\centering" _n
file write it "\caption{IV diagnostics (2SLS, heteroskedasticity-robust)}" _n
file write it "\begin{tabular}{lcc}" _n "\toprule" _n
file write it "Test & Statistic & \(p\) \\" _n "\midrule" _n
file write it "First-stage \(F\) (weak instruments) & " %6.1f (`fsF') " & --- \\" _n
file write it "Endogeneity: Durbin--Wu--Hausman (\(\chi^2_1\)) & " %6.1f (`dwh') " & `dwhpt' \\"
    _n
file write it "Overidentification: robust score (\(\chi^2_1\)) & " %5.2f (`oj') " & `ojpt' \\" _n
file write it "\bottomrule" _n "\end{tabular}" _n "\end{table}" _n
file close it

log close
display "07_iv_2sls.do finished OK"
```

## 8.3 The results

The true slope is $2.0$. The three columns are three estimators of the *same* structural equation, differing only in how the endogenous `x` enters:

$$\begin{aligned}
\text{(1) OLS:} \quad & y_i = \beta_0 + \beta_1\, \mathtt{x}_i + \varepsilon_i && (\text{raw } \mathtt{x} \to \text{biased}),\\
\text{(2) 2SLS:} \quad & y_i = \beta_0 + \beta_1\, \widehat{\mathtt{x}}_i + \varepsilon_i && (\widehat{\mathtt{x}} \text{ from the first stage}),\\
\text{(3) 2SLS+FE:} \quad & y_i = \alpha_{g(i)} + \beta_1\, \widehat{\mathtt{x}}_i + \varepsilon_i && (\alpha_g \text{ absorbed}),
\end{aligned}$$

fit by `regress y x`, `ivregress 2sls y (x = z1 z2)`, and `ivreghdfe` with `absorb(grp)`. The two IV estimators replace the contaminated `x` with the fitted value $\widehat{\mathtt{x}}$ from the first-stage regression of `x` on the instruments,

$$\widehat{\mathtt{x}}_i = \hat{\pi}_0 + \hat{\pi}_1\, \mathtt{z1}_i + \hat{\pi}_2\, \mathtt{z2}_i.$$

OLS is biased upward because $\mathrm{Cov}(\mathtt{x}, \varepsilon) > 0$ through the unobserved confounder; both IV estimators are consistent and recover the truth. Efficient GMM (`ivregress gmm`) returns $\hat{\beta}_1 \approx 1.96$, indistinguishable from 2SLS here because the simulated errors are homoskedastic — the optimal weight matrix buys nothing when 2SLS is already efficient.

The diagnostics confirm the design: the first-stage $F$ is far above the weak-instrument rule-of-thumb of 10; the Durbin–Wu–Hausman test strongly rejects exogeneity, so instrumenting is warranted; and the over-identification test does *not* reject, consistent with both instruments being valid.

Table 8.1: IV vs OLS

| | (1) OLS | (2) 2SLS | (3) IV+FE |
|---|---|---|---|
| x | 2.173*** | 1.960*** | 1.994*** |
| | (0.015) | (0.027) | (0.020) |
| $N$ | 5000 | 5000 | 5000 |

Standard errors in parentheses

* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

Table 8.2: IV diagnostics (2SLS, heteroskedasticity-robust)

| Test | Statistic | $p$ |
|---|---|---|
| First-stage $F$ (weak instruments) | 1243.7 | — |
| Endogeneity: Durbin–Wu–Hausman ($\chi^2_1$) | 95.6 | $<$0.001 |
| Overidentification: robust score ($\chi^2_1$) | 1.48 | 0.224 |

# Chapter 9

# Binary outcome models

↪ *the method, explained:* *Appendix B.5*

## 9.1 The method

For a 0/1 outcome,

$$P(y_i = 1 \mid x_i) = G(x_i'\beta),$$

with $G = \Lambda$ (logit) or $G = \Phi$ (probit); the linear probability model is OLS on the indicator. Raw coefficients are not comparable across these — the comparable quantity is the average marginal effect

$$\mathrm{AME}_j = \frac{1}{n}\sum_i g(x_i'\beta)\,\beta_j,$$

which is nearly identical for logit and probit. Logit odds ratios are $e^{\beta_j}$. Fit is summarised by a classification table and the area under the ROC curve.

**In practice.** Binary-outcome models fit any yes/no decision — loan default and credit approval, labour-force participation, mortgage acceptance, or vote choice — reporting the average marginal effects rather than the raw coefficients.

## 9.2 The code

*The workflow.* Fit the linear-probability, logit, and probit models, read the odds ratios, compute average marginal effects with `margins`, and plot the predicted probabilities.

```
*===============================================================================
* 08_logit_probit.do  ---  binary-outcome models and how to interpret them
*-------------------------------------------------------------------------------
* Models the probability a car is foreign as a function of mpg, weight, price.
* The key lesson: logit/probit coefficients are NOT directly interpretable ---
* use margins for average marginal effects and predicted probabilities.
*===============================================================================
version 17
clear all
set more off
set linesize 90
capture log close
capture mkdir logs   // create the log folder if absent (gitignored)
log using "logs/08_logit_probit.log", replace text

sysuse auto, clear
generate weight_t = weight/1000
generate price_k  = price/1000
label variable weight_t "Weight (1000 lbs)"
```

```
label variable price_k  "Price ($1000s)"

*-------------------------------------------------------------------------------
* 1. Three ways to model a 0/1 outcome: LPM, logit, probit.
*-------------------------------------------------------------------------------
eststo clear
eststo lpm:    regress foreign mpg weight_t price_k, vce(robust)   // linear prob.
eststo logit:  logit   foreign mpg weight_t price_k, vce(robust)
eststo probit: probit  foreign mpg weight_t price_k, vce(robust)
* NOTE: raw coefficients are on different scales --- do NOT compare directly.
esttab lpm logit probit, b(%9.3f) se star(* 0.10 ** 0.05 *** 0.01) ///
    mtitles("LPM" "Logit" "Probit") label ///
    title("Coefficients differ by scale --- compare marginal effects, not these")

*-------------------------------------------------------------------------------
* 2. Odds ratios (logit) --- a common reporting convention.
*-------------------------------------------------------------------------------
logit foreign mpg weight_t price_k, or

*-------------------------------------------------------------------------------
* 3. Average marginal effects --- the comparable, interpretable quantities.
*    Logit and probit AMEs are typically very close.
*-------------------------------------------------------------------------------
quietly logit foreign mpg weight_t price_k
margins, dydx(*) post
estimates store ame_logit
quietly probit foreign mpg weight_t price_k
margins, dydx(*) post
estimates store ame_probit
esttab ame_logit ame_probit, b(%9.4f) se ///
    mtitles("Logit AME" "Probit AME") title("Average marginal effects (compare these)")

*-------------------------------------------------------------------------------
* 4. Predicted probabilities + a plot across the range of mpg.
*-------------------------------------------------------------------------------
quietly logit foreign mpg weight_t price_k
margins, at(mpg=(12(4)40))
marginsplot, title("Predicted P(foreign) across mpg") ///
    ytitle("Predicted probability")
graph export "output/08_logit_pr.png", replace width(1300)

*-------------------------------------------------------------------------------
* 5. In-sample fit: classification table and ROC area.
*-------------------------------------------------------------------------------
quietly logit foreign mpg weight_t price_k
estat classification                    // sensitivity/specificity at p=0.5
lroc, nograph                           // area under ROC curve

log close
display "08_logit_probit.do finished OK"
```

## 9.3 The results

The three columns fit the same linear index $x_i'\beta = \beta_0 + \beta_1\,\texttt{mpg} + \beta_2\,\texttt{weight\_t} + \beta_3\,\texttt{price\_k}$ through three different link functions:

$$\begin{aligned}
\text{(1) LPM:} \quad & P(\texttt{foreign}_i = 1 \mid x_i) = x_i'\beta, \\
\text{(2) logit:} \quad & P(\texttt{foreign}_i = 1 \mid x_i) = \Lambda(x_i'\beta), \\
\text{(3) probit:} \quad & P(\texttt{foreign}_i = 1 \mid x_i) = \Phi(x_i'\beta),
\end{aligned}$$

fit by `regress`, `logit`, and `probit` respectively, with $\Lambda$ the logistic CDF and $\Phi$ the standard-normal CDF. Because the link functions differ, the raw coefficients are not comparable across columns — the comparable quantity is the average marginal effect (`margins, dydx(*)`). The logit and probit average marginal effects coincide to the third decimal — as expected, the two link functions imply nearly the same marginal effects — and in-sample fit is high (area under the ROC curve $\approx 0.96$). The plot shows the predicted probability falling smoothly with mileage.

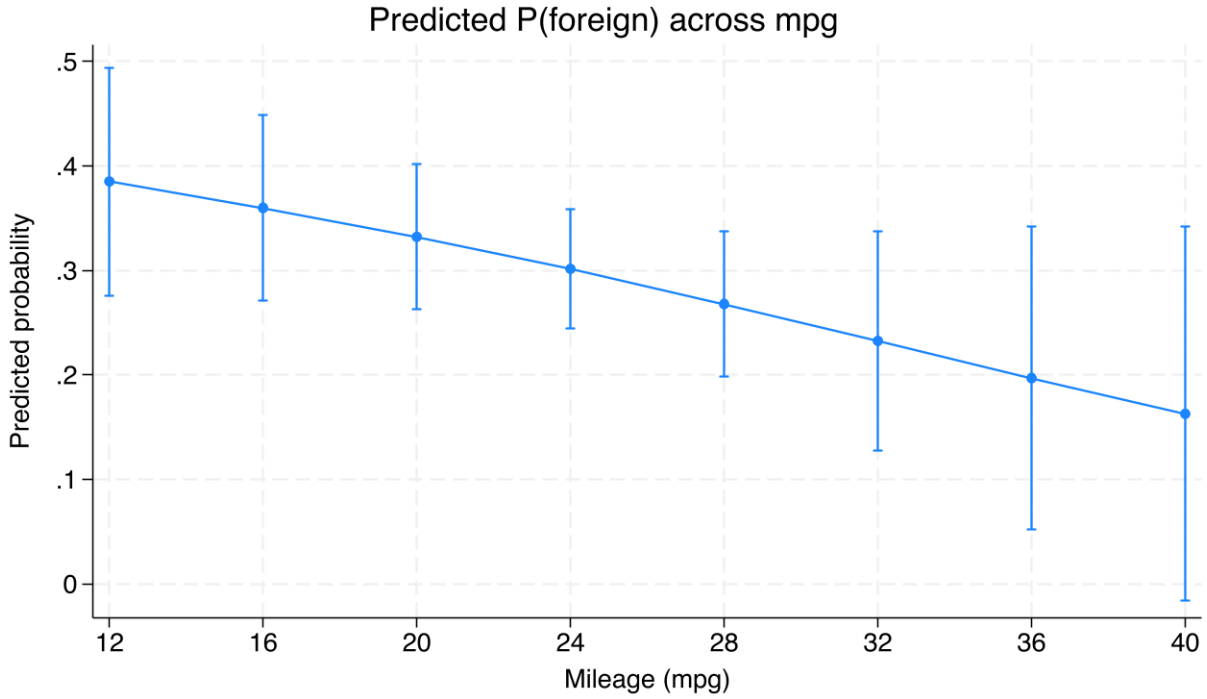



Figure 9.1: Predicted probability that a car is foreign, across mileage.

# Chapter 10

# Staggered-adoption DiD

↪ *the method, explained:* *Appendix B.6*

## 10.1 The method

When units are treated at different times (say, states adopting a policy in different years) and effects are dynamic, the two-way FE coefficient is a non-convex weighted average of effects and can be badly biased (Goodman-Bacon, 2021). Callaway and Sant'Anna (2021) instead estimate clean group-time average treatment effects,

$$ATT(g,t) = E[\,Y_t - Y_{g-1} \mid G = g\,] - E[\,Y_t - Y_{g-1} \mid C\,],$$

using not-yet- or never-treated units $C$ as comparisons, via a doubly-robust estimator. These are then aggregated (overall, by event time, by cohort, by calendar period).

**In practice.** This is the tool for staggered policy adoption — minimum-wage changes, Medicaid expansion, or marijuana legalization rolling out across states in different years — the very setting where naive two-way FE goes wrong.

## 10.2 The code

*The workflow.* Build a staggered-adoption panel, estimate clean group-time effects with `csdid`, aggregate them, and contrast with the biased naive two-way fixed-effects estimate.

```
*==============================================================================
* 09_staggered_did_csdid.do  ---  staggered-adoption DiD (Callaway & Sant'Anna)
*------------------------------------------------------------------------------
* When units are treated at DIFFERENT times, two-way FE DiD can be biased
* (the "forbidden comparison" / negative-weighting problem). Callaway &
* Sant'Anna (2021) estimate clean group-time ATTs and aggregate them.
*
* Simulates 3 cohorts: first-treated in 2014, in 2017, and never-treated.
* True effect is 0 pre-treatment and grows after each cohort's own start.
*==============================================================================
version 17
clear all
set more off
set linesize 90
set seed 555
capture log close
capture mkdir logs   // create the log folder if absent (gitignored)
log using "logs/09_staggered_did_csdid.log", replace text

*------------------------------------------------------------------------------
* 1. Build a staggered panel: 300 units x 12 years (2010-2021).
```

```
*    gvar = year of first treatment; 0 = never treated (csdid convention).
*-------------------------------------------------------------------------------
set obs 300
generate id = _n
generate g = 0                              // never-treated by default
replace  g = 2014 if id <= 100              // early cohort
replace  g = 2017 if id > 100 & id <= 200 // late cohort
generate u_id = rnormal()*1.5               // unit fixed effect
expand 12
bysort id: generate year = 2009 + _n      // 2010..2021

generate att = 0
replace  att = 0.4*(year - g + 1) if g>0 & year>=g   // dynamic, grows post-onset
generate yr_fe = 0.2*(year-2015)            // common year trend
generate y = 3 + u_id + yr_fe + att + rnormal()
label variable y "Outcome"

tabulate g, missing                         // cohort sizes (in unit-years)

*-------------------------------------------------------------------------------
* 2. Callaway & Sant'Anna estimator (doubly robust). No covariates here.
*-------------------------------------------------------------------------------
csdid y, ivar(id) time(year) gvar(g) method(dripw)

*-------------------------------------------------------------------------------
* 3. Aggregations of the group-time ATTs.
*-------------------------------------------------------------------------------
estat simple                                // single overall ATT
matrix _Tsimple = r(table)                  // capture the aggregate ATT now, before
                                            //   estat group/event overwrite r(table)
estat group                                 // ATT by treatment cohort
estat event                                 // dynamic / event-study path (LAST, so
                                            //   csdid_plot below plots THIS)

*-------------------------------------------------------------------------------
* 4. Event-study plot of the dynamic aggregation.
*    csdid_plot graphs the most recent estat aggregation -> run estat event last.
*-------------------------------------------------------------------------------
csdid_plot, title("Callaway-Sant'Anna event study") ///
    ytitle("ATT") xtitle("Event time (years relative to treatment)")
graph export "output/09_csdid_event.png", replace width(1400)

*-------------------------------------------------------------------------------
* 5. For contrast: the naive two-way FE estimate on the same data.
*    (Can be biased under staggered timing + dynamic effects --- that's the point.)
*-------------------------------------------------------------------------------
generate byte treated_now = (g>0 & year>=g)
reghdfe y treated_now, absorb(id year) vce(cluster id)

*-------------------------------------------------------------------------------
* 6. Export a table contrasting the Callaway-Sant'Anna ATT with the naive TWFE
*    estimate (the bias this method removes).
*-------------------------------------------------------------------------------
local twfe  = _b[treated_now]
local twse  = _se[treated_now]
local csatt = _Tsimple[1,1]
local csse  = _Tsimple[2,1]

capture file close ct
```

```
file open ct using "output/09_csdid_table.tex", write replace
file write ct "\begin{table}[htbp]\centering" _n
file write ct "\caption{Overall ATT: Callaway--Sant'Anna vs.\ naive two-way FE}" _n
file write ct "\begin{tabular}{lcc}" _n "\toprule" _n
file write ct "Estimator & ATT & Std.\ err. \\" _n "\midrule" _n
file write ct "Callaway--Sant'Anna (\pkg{csdid}) & " %5.3f (`csatt') " & " %5.3f (`csse') " \\"
    _n
file write ct "Naive two-way FE (\pkg{reghdfe}) & " %5.3f (`twfe') " & " %5.3f (`twse') " \\" _n
file write ct "\bottomrule" _n "\end{tabular}" _n "\end{table}" _n
file close ct

log close
display "09_staggered_did_csdid.do finished OK"
```

## 10.3 The results

The two rows of the table are two estimators on the *same* staggered panel. `csdid` — with `gvar(g)` passing each unit's treatment cohort ($g$ = year of first treatment, $0$ = never treated) — aggregates the clean group-time effects of §The method, whereas the naive two-way FE regresses the outcome on a single post-onset indicator:

$$\begin{aligned} \text{Callaway–Sant'Anna:} \quad & \widehat{ATT} \text{ aggregated from } ATT(g,t), \\ \text{naive two-way FE:} \quad & y_{it} = \alpha_i + \gamma_t + \beta^{\text{TWFE}}\, \texttt{treated\_now}_{it} + \varepsilon_{it}. \end{aligned}$$

The single coefficient $\beta^{\text{TWFE}}$ is the non-convex weighted average that misleads under staggered timing. The estimator returns a flat pre-trend (no anticipation) and a rising post-treatment path that tracks the planted dynamic effect. The overall ATT recovers the simulated average; the naive two-way FE estimate on the *same* data is substantially smaller — the negative-weighting bias this method is designed to remove.

Table 10.1: Overall ATT: Callaway–Sant'Anna vs. naive two-way FE

| Estimator | ATT | Std. err. |
|---|---|---|
| Callaway–Sant'Anna (`csdid`) | 1.495 | 0.112 |
| Naive two-way FE (`reghdfe`) | 1.009 | 0.073 |

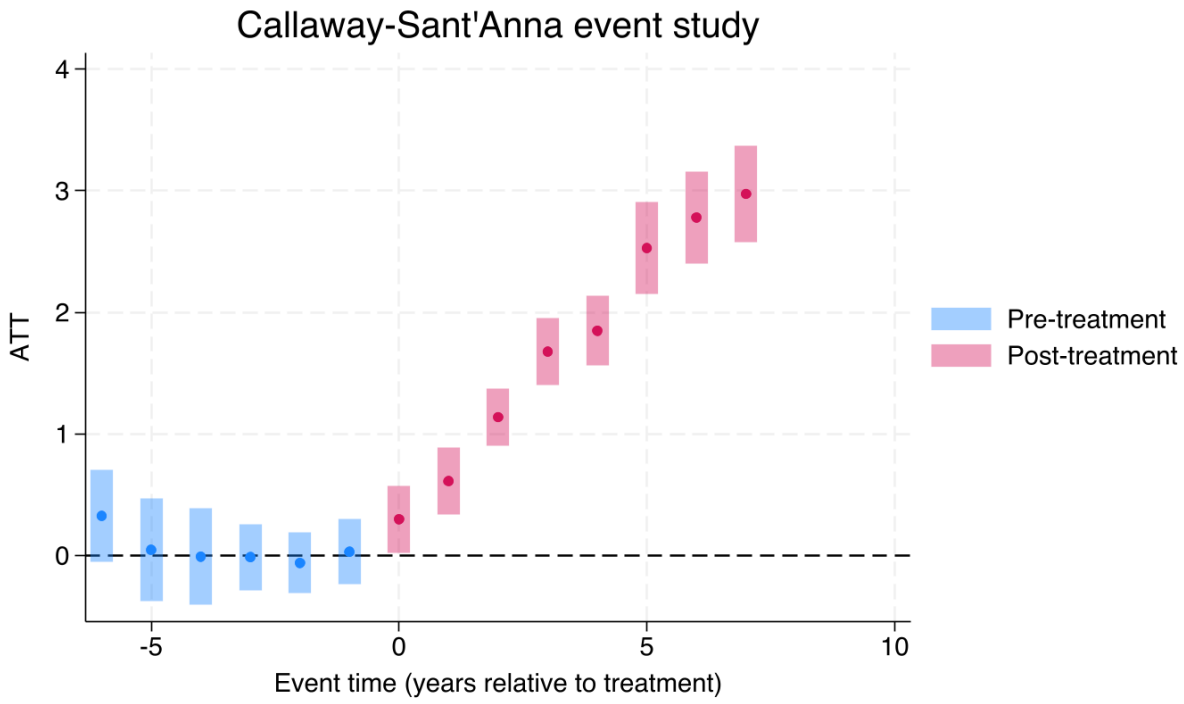



Figure 10.1: Callaway–Sant'Anna event study: pre-treatment (covering zero) vs. post-treatment.

# Chapter 11

# Regression discontinuity

↪ *the method, explained:* *Appendix B.7*

## 11.1 The method

When treatment switches at a cutoff of a running variable (say, a scholarship awarded once a test score clears a threshold), $d_i = \mathbf{1}[x_i \geq c]$, the sharp RD estimand is the jump in the conditional mean,

$$\tau = \underbrace{\lim_{x \downarrow c} E[Y \mid X = x]}_{\text{just ABOVE the cutoff}} - \underbrace{\lim_{x \uparrow c} E[Y \mid X = x]}_{\text{just BELOW the cutoff}},$$

estimated by local-linear regression each side of $c$ within an MSE-optimal bandwidth, with Calonico, Cattaneo and Titiunik (2014) bias-corrected robust confidence intervals. A manipulation test checks that the density of $X$ is continuous at $c$ (Cattaneo, Jansson and Ma, 2020).

**In practice.** Sharp RD exploits threshold rules: scholarship and test-score cutoffs (its original use, Thistlethwaite and Campbell 1960), class-size caps (the "Maimonides rule" of Angrist and Lavy 1999), vote-share thresholds for incumbency in close elections (Lee 2008), and program-eligibility cutoffs.

## 11.2 The code

*The workflow.* Simulate a sharp-RD dataset, estimate the jump with `rdrobust`, select the bandwidth `rdbwselect`, draw the annotated plot `rdplot`, test for manipulation with `rddensity`, and export the results table.

```
*===============================================================================
* 10_rdd_regression_discontinuity.do  ---  sharp regression discontinuity (RD)
*-------------------------------------------------------------------------------
* Units with running variable x >= 0 get treated. The outcome is a smooth
* function of x with a JUMP of 2.0 at the cutoff = the treatment effect.
* Uses Calonico-Cattaneo-Titiunik local-polynomial estimation (rdrobust).
*===============================================================================
version 17
clear all
set more off
set linesize 90
set seed 1010
capture log close
capture mkdir logs    // create the log folder if absent (gitignored)
log using "logs/10_rdd_regression_discontinuity.log", replace text

set obs 3000
```

```
generate x = runiform(-1,1)              // running variable; cutoff at 0
generate byte d = (x >= 0)               // sharp assignment
generate y = 3 + 1.5*x + 0.5*x^2 + 2.0*d + rnormal(0,0.5)  // TRUE jump = 2.0
label variable x "Running variable"
label variable y "Outcome"

*-------------------------------------------------------------------------------
* 1. RD point estimate: local-linear fit each side of the cutoff, MSE-optimal
*    bandwidth, bias-corrected robust confidence interval.
*-------------------------------------------------------------------------------
rdrobust y x, c(0)

*-------------------------------------------------------------------------------
* 2. Data-driven bandwidth selection (what rdrobust chose, and alternatives).
*-------------------------------------------------------------------------------
rdbwselect y x, c(0) all

*-------------------------------------------------------------------------------
* 3. The annotated RD picture. Binned means (rdplot genvars) + a local fit each
*    side, then three annotations that make the estimate legible:
*      - open dots at the two one-sided limits at the cutoff;
*      - a grey double arrow, offset just right of the cutoff, spanning the
*        estimated jump and labelled with rdrobust's tau-hat (so the picture
*        reports the SAME number as the results table); short dashed leaders
*        link the two limit dots to the arrow's ends;
*      - the control fit extrapolated across c as the "no-treatment"
*        counterfactual (dashed) -- what the treated units would have done.
*-------------------------------------------------------------------------------
rdplot y x, c(0) genvars hide
egen _bin = tag(rdplot_id)

* local fit each side (quadratic, matching the smooth DGP) + the two limits at c
quietly regress y c.x##c.x if x<0
predict double _fit_l if x<0                      // control fit
predict double _cf    if inrange(x,0,0.45)        // control fit extended past c
local yL = _b[_cons]                              // control limit  lim_{x up c}
quietly regress y c.x##c.x if x>=0
predict double _fit_r if x>=0                      // treated fit
local yR = _b[_cons]                              // treated limit  lim_{x down c}

* headline estimate (rdrobust local-linear) drives the arrow + its label
quietly rdrobust y x, c(0)
local tau  = e(tau_cl)
local yT   = `yL' + `tau'                          // treated limit implied by tau-hat
local taus : display %4.3f `tau'
local ymid = (`yL' + `yT')/2

twoway ///
    (scatter rdplot_mean_y rdplot_mean_x if rdplot_mean_x<0  & _bin, mcolor(midblue%35)   msize(
    small)) ///
    (scatter rdplot_mean_y rdplot_mean_x if rdplot_mean_x>=0 & _bin, mcolor(cranberry%35) msize(
    small)) ///
    (line _fit_l x if x<0,  sort lcolor(midblue)   lwidth(medthick)) ///
    (line _fit_r x if x>=0, sort lcolor(cranberry) lwidth(medthick)) ///
    (line _cf    x if inrange(x,0,0.45), sort lcolor(midblue%55) lpattern(dash) lwidth(medium))
    ///
    (pci `yT' 0 `yT' 0.24, lcolor(gs9) lpattern(dash) lwidth(thin)) ///
    (pci `yL' 0 `yL' 0.24, lcolor(gs9) lpattern(dash) lwidth(thin)) ///
    (pcarrowi `yL' 0.24 `yT' 0.24, lcolor(gs9) lwidth(medthick) mcolor(gs9) barbsize(2.2)) ///
```

```
    (pcarrowi `yT' 0.24 `yL' 0.24, lcolor(gs9) lwidth(medthick) mcolor(gs9) barbsize(2.2)) ///
    (scatteri `yL' 0 `yT' 0, mcolor(white) mlcolor(black) mlwidth(thin) msymbol(O) msize(medium))
     ///
    , ///
    xline(0, lpattern(dash) lcolor(gs10)) ///
    text(`ymid' 0.27 "{&tau} = `taus'", color(gs9) place(e) size(medium)) ///
    legend(order(3 "Control fit (x < c)" 4 "Treated fit (x {&ge} c)" ///
                 5 "Counterfactual (no treatment)") ///
           cols(1) position(11) ring(0) size(small) region(lstyle(none) fcolor(none))) ///
    title("Sharp RD: the outcome jumps at the cutoff") ///
    subtitle("binned means + local fit each side; grey arrow = rdrobust estimate") ///
    xtitle("Running variable  X   (sharp cutoff at c)") ytitle("Outcome  Y") ///
    xlabel(-1 -.5 0 "c = 0" .5 1)
graph export "output/10_rdplot.png", replace width(1600)
drop _fit_l _fit_r _cf _bin

*-------------------------------------------------------------------------------
* 4. Manipulation test: is the density of x continuous at the cutoff?
*    (A jump would suggest units sorted around the threshold --- an RD red flag.)
*    Here x is uniform, so the test should NOT reject.
*-------------------------------------------------------------------------------
rddensity x, c(0)

*-------------------------------------------------------------------------------
* 5. Export a compact results table for the reference. rdplot's genvars call
*    overwrote e(), so re-fit quietly to recover the rdrobust/rddensity scalars.
*-------------------------------------------------------------------------------
quietly rdrobust y x, c(0)
local tau = e(tau_cl)
local cil = e(ci_l_rb)
local cir = e(ci_r_rb)
local pv  = e(pv_rb)
quietly rddensity x, c(0)
local mpv = e(pv_q)
local ptxt = cond(`pv'<0.001, "\(<\)0.001", string(`pv',"%5.3f"))

capture file close rt
file open rt using "output/10_rd_table.tex", write replace
file write rt "\begin{table}[htbp]\centering" _n
file write rt "\caption{Sharp RD estimate vs.\ the true jump (\(\tau=2.0\))}" _n
file write rt "\begin{tabular}{lc}" _n "\toprule" _n
file write rt " & Estimate \\" _n "\midrule" _n
file write rt "RD effect (\(\hat\tau\)) & " %5.3f (`tau') " \\" _n
file write rt "Robust 95\% CI & [" %5.3f (`cil') ", " %5.3f (`cir') "] \\" _n
file write rt "Robust \(p\)-value & `ptxt' \\" _n
file write rt "\addlinespace" _n
file write rt "Manipulation test \(p\) (\pkg{rddensity}) &" %5.3f (`mpv') " \\" _n
file write rt "\bottomrule" _n "\end{tabular}" _n "\end{table}" _n
file close rt

log close
display "10_rdd_regression_discontinuity.do finished OK"
```

## 11.3 The results

`rdrobust y x, c(0)` fits a separate local-linear regression on each side of the cutoff $c = 0$, inside an MSE-optimal bandwidth $h$:

$$\begin{aligned} \text{below } (\mathtt{x} < 0): &\quad E[\mathtt{y} \mid \mathtt{x}] = \alpha_- + \beta_- \mathtt{x}, \\ \text{above } (\mathtt{x} \geq 0): &\quad E[\mathtt{y} \mid \mathtt{x}] = \alpha_+ + \beta_+ \mathtt{x}, \end{aligned}$$

and the sharp-RD estimate is the gap at the cutoff, $\hat{\tau} = \alpha_+ - \alpha_-$ — the $\lim_{x \downarrow c} - \lim_{x \uparrow c}$ of §The method. The estimate recovers the true jump of $2.0$ within a tight robust confidence interval, so the local-linear RD is approximately unbiased here. The manipulation test does not reject, so the running variable shows no sign of sorting at the cutoff — the design's key validity check passes.

Table 11.1: Sharp RD estimate vs. the true jump ($\tau = 2.0$)

| | Estimate |
|---|---|
| RD effect ($\hat{\tau}$) | 1.938 |
| Robust 95% CI | [1.770, 2.082] |
| Robust $p$-value | $<$0.001 |
| Manipulation test $p$ (`rddensity`) | 0.953 |

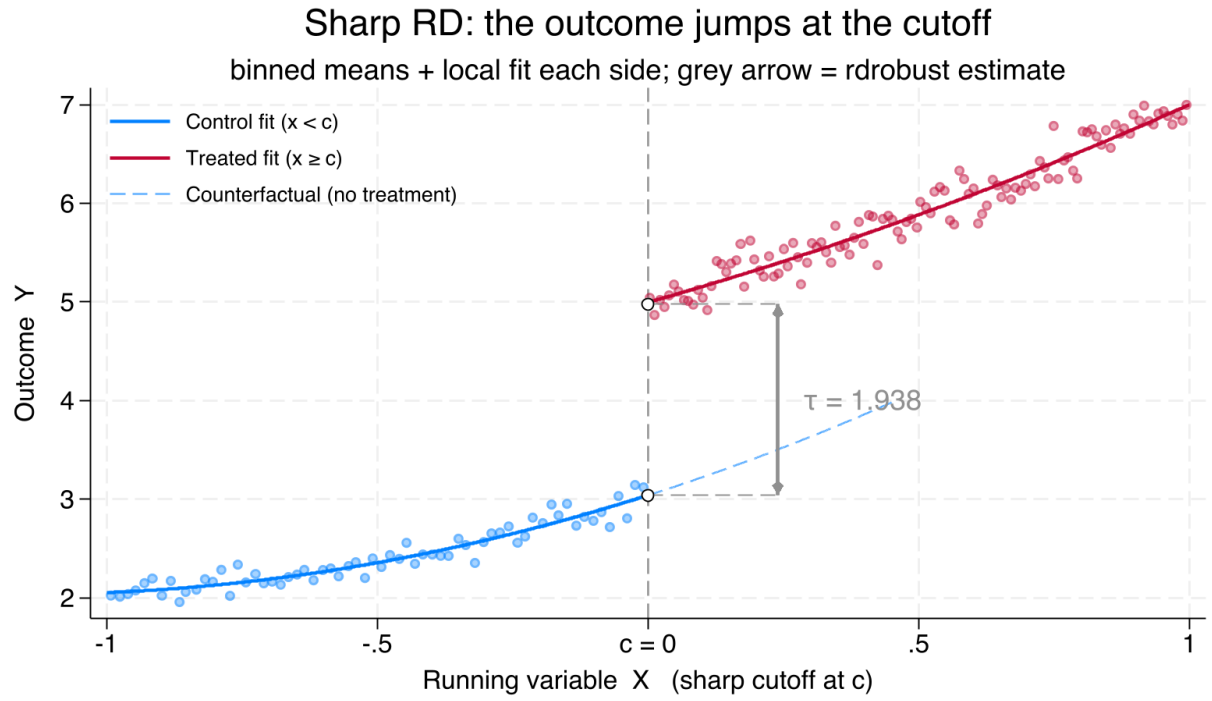



Figure 11.1: Sharp RD: binned means and a local fit each side of the cutoff. The grey double arrow (just right of the cutoff) marks the estimated jump (`rdrobust` $\hat{\tau} = 1.938$, the same figure as Table 11.1), with open dots at the two one-sided limits; the dashed line extrapolates the control fit across $c$ as the no-treatment counterfactual.

# Chapter 12

# Synthetic control

↪ *the method, explained:* *Appendix B.8*

## 12.1 The method

The synthetic control method (Abadie, Diamond and Hainmueller, 2010) takes one treated unit and many donors (say, one state that adopts a policy, set against a weighted blend of states that did not) and chooses non-negative weights $w_j$ summing to one that minimise the pre-treatment distance

$$(X_1 - X_0 W)' V (X_1 - X_0 W).$$

The synthetic control is $\sum_j w_j Y_{jt}$ and the estimated effect is the post-treatment gap $Y_{1t} - \sum_j w_j Y_{jt}$. Inference is by placebo: re-estimate treating each donor as treated and rank the treated unit's gap against the placebo distribution.

**In practice.** Synthetic control suits single-unit comparative cases: the effect of California's Proposition 99 tobacco tax (Abadie, Diamond and Hainmueller), terrorism and Basque GDP (Abadie and Gardeazabal 2003), and German reunification.

## 12.2 The code

*The workflow.* Build a donor panel, fit the `synth` weighted counterfactual for the treated unit, plot treated-vs-synthetic paths, and run placebo inference with `synth_runner`.

```
*==============================================================================
* 11_synthetic_control.do  ---  synthetic control method (Abadie et al.)
*------------------------------------------------------------------------------
* One treated unit, many controls. We build a "synthetic" version of the treated
* unit as a weighted average of controls that matches its PRE-treatment path,
* then read the treatment effect as the post-treatment gap between the two.
*
* Simulates 20 units x 30 periods; unit 1 is treated from t = 21 with a true
* effect of +3. A factor structure makes a good synthetic control feasible.
*==============================================================================
version 17
clear all
set more off
set linesize 90
set seed 2020
capture log close
capture mkdir logs   // create the log folder if absent (gitignored)
log using "logs/11_synthetic_control.log", replace text

*------------------------------------------------------------------------------
* 1. Build the panel (factor model: shared time factors x unit-specific loadings).
```

```
*-------------------------------------------------------------------------------
set obs 20
generate id = _n
generate u1 = runiform()                  // loading on factor 1 (fixed within unit)
generate u2 = runiform()                  // loading on factor 2
expand 30
bysort id: generate t = _n                // 1..30
generate f1 = sin(t/3)                    // common factors (functions of time)
generate f2 = t/30
generate y = 5 + 3*u1*f1 + 4*u2*f2 + rnormal(0,0.3)
replace  y = y + 3 if id==1 & t>20        // TRUE post-treatment effect = +3 on unit 1
label variable y "Outcome"

xtset id t                                // declare the panel

*-------------------------------------------------------------------------------
* 2. Fit the synthetic control for unit 1 (treatment starts t = 21). We save the
*    treated/synthetic paths with keep(), then draw a COLORED version of them
*    (synth's built-in `fig` is monochrome).
*-------------------------------------------------------------------------------
synth y y(5) y(10) y(15) y(20) u1 u2, ///
    trunit(1) trperiod(21) keep("output/synth_results.dta", replace)

preserve
    use "output/synth_results.dta", clear
    keep if !missing(_time)
    twoway (line _Y_treated   _time, lcolor(navy)      lwidth(medthick)) ///
           (line _Y_synthetic _time, lcolor(cranberry) lwidth(medthick) lpattern(dash)), ///
        xline(20.5, lpattern(dash) lcolor(gs10)) ///
        legend(order(1 "Treated unit" 2 "Synthetic control") rows(1) position(6)) ///
        title("Synthetic control: treated unit vs. its synthetic") ///
        xtitle("t") ytitle("Outcome")
    graph export "output/11_synth_path.png", replace width(1400)
restore

*-------------------------------------------------------------------------------
* 3. Inference by placebo: re-assign treatment to each control unit in turn and
*    compare the treated unit's gap to the placebo distribution (synth_runner).
*-------------------------------------------------------------------------------
synth_runner y y(5) y(10) y(15) y(20) u1 u2, ///
    trunit(1) trperiod(21) gen_vars

* Colored placebo "gap" plot: each control gap in light gray, the treated gap bold.
* synth_runner's gen_vars stores each unit's gap in `effect'.
levelsof id if id!=1, local(donors)
local plots ""
foreach d of local donors {
    local plots `plots' (line effect t if id==`d', lcolor(gs13) lwidth(vthin))
}
twoway `plots' (line effect t if id==1, lcolor(cranberry) lwidth(thick)), ///
    yline(0, lcolor(gs9)) xline(20.5, lpattern(dash) lcolor(gs10)) legend(off) ///
    title("Placebo gaps: treated unit (bold) vs. control units") ///
    xtitle("t") ytitle("Gap (treated - synthetic)")
graph export "output/11_synth_placebo.png", replace width(1400)

log close
display "11_synthetic_control.do finished OK"
```

## 12.3 The results

The true post-treatment effect is $+3$. `synth y y(5) y(10) y(15) y(20) u1 u2, trunit(1) trperiod(21)` chooses non-negative donor weights $w_j$ (summing to one) so that treated unit 1's pre-period predictors — the listed outcome lags and covariates — best match the weighted donors. The synthetic control is then $\sum_j w_j \, \mathrm{y}_{jt}$, and the estimated effect is the post-period gap

$$\hat{\tau}_t = \mathrm{y}_{1t} - \sum_j w_j \, \mathrm{y}_{jt}, \qquad t \geq 21.$$

The synthetic control tracks the treated unit closely *before* treatment — a good pre-fit is what licenses the comparison — and diverges by about 3 *after*. In the placebo distribution the treated unit's gap exceeds every control's, so the permutation $p$-value is near zero: the effect is unlikely to be an artefact of chance.

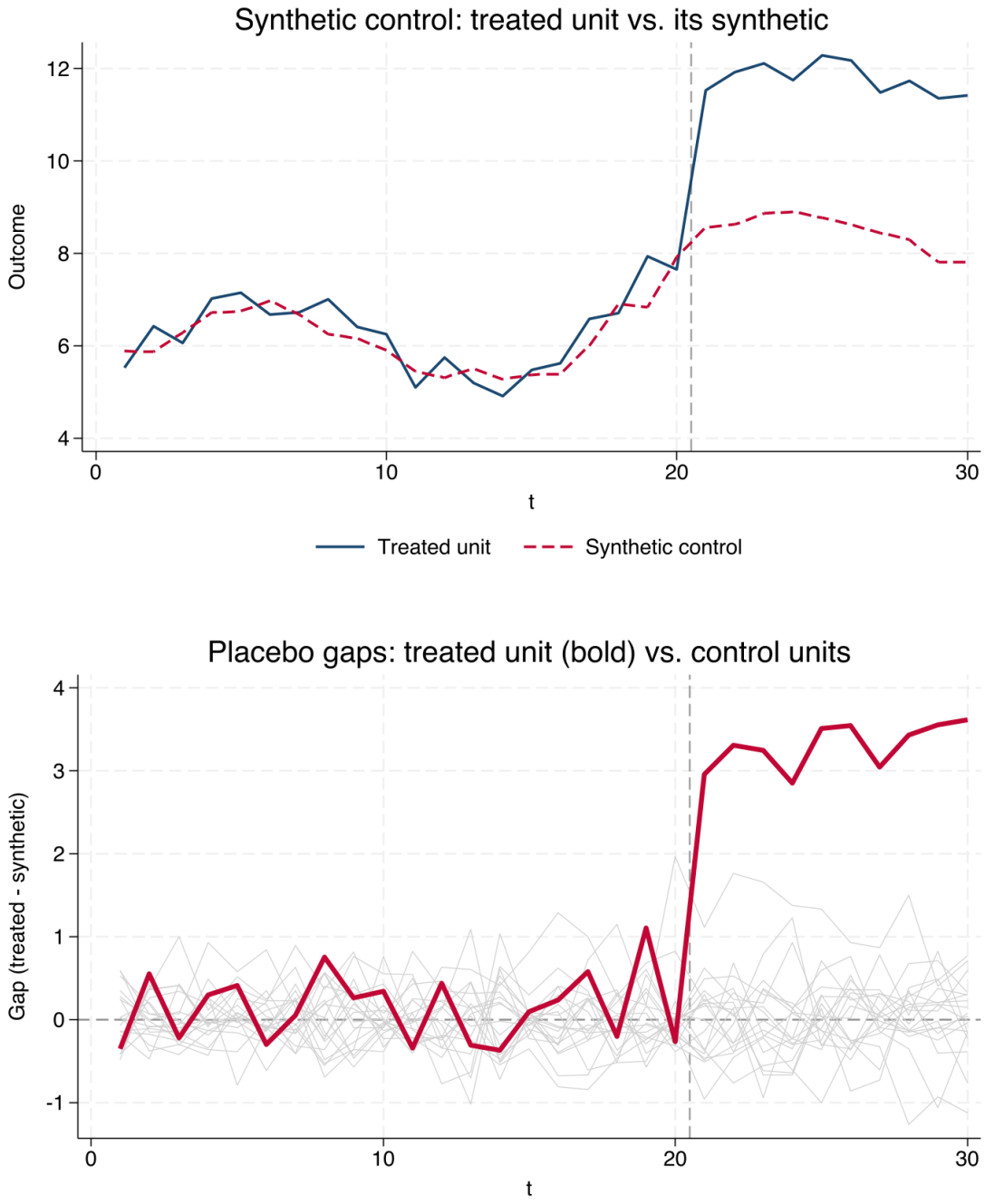



Figure 12.1: Treated vs. synthetic outcome path (top) and the treated gap against placebo gaps (bottom).

# Chapter 13

# Dynamic panel data (GMM)

↪ *the method, explained:* *Appendix B.9*

## 13.1 The method

A *dynamic* panel carries a lagged dependent variable,

$$y_{it} = \rho\, y_{i,t-1} + \beta\, \mathbf{x}_{it} + \alpha_i + \varepsilon_{it},$$

and that lag is the problem: $y_{i,t-1}$ contains $\alpha_i$, so it is correlated with the unit effect. Pooled OLS therefore *overstates* $\rho$, while the within (fixed-effects) transform correlates the demeaned lag with the demeaned error — the Nickell (1981) bias — so FE *understates* it, badly when $T$ is small; the truth is bracketed between the two. Arellano and Bond (1991) first-difference the equation to sweep out $\alpha_i$ and instrument the differenced lag $\Delta y_{i,t-1}$ with *deeper lagged levels* $y_{i,t-2}, y_{i,t-3}, \ldots$ (valid, since they are uncorrelated with $\Delta\varepsilon_{it}$) — this is *difference GMM*. Blundell and Bond (1998) add moment conditions in levels (lagged differences instrumenting the levels), giving the more efficient *system GMM*, far better behaved when $\rho$ is near one.

The key validity check is the Arellano–Bond test for serial correlation in the differenced residuals: AR(1) should reject (differencing mechanically induces it) but AR(2) should *not* — a significant AR(2) means the deeper lags are themselves endogenous and the design fails. Overidentification is the Sargan/Hansen $J$.

**In practice.** Dynamic panels are everywhere in corporate finance and macro — the persistence of firm leverage, of investment, of employment, or of national output — wherever a lagged dependent variable sits beside unit effects. Roodman's community `xtabond2` is the most popular implementation; the built-in `xtabond` and `xtdpdsys` used here need no install.

## 13.2 The code

*The workflow.* Simulate a dynamic panel with a known $\rho$, show that pooled OLS and within FE bracket the truth from above and below, then recover it with Arellano–Bond difference GMM (`xtabond`) and Blundell–Bond system GMM (`xtdpdsys`), checking the AR(2) moment condition.

```
*===============================================================================
* 12_dynamic_panel_gmm.do  ---  dynamic panel data: Arellano-Bond &
*                               Blundell-Bond (system) GMM
*-------------------------------------------------------------------------------
* A dynamic panel  y_it = rho*y_{i,t-1} + beta*x_it + alpha_i + e_it  has a
* built-in endogeneity: the lagged dependent variable is mechanically correlated
* with the unit effect alpha_i. Pooled OLS therefore OVERstates rho, and the
* within (FE) estimator UNDERstates it (Nickell bias, severe when T is small).
* Difference and system GMM instrument the lagged level with deeper lags and
```

```
* recover the truth. TRUE rho = 0.6, beta = 1.
*==============================================================================
version 17
clear all
set more off
set linesize 90
set seed 90210
capture log close
capture mkdir logs   // create the log folder if absent (gitignored)
log using "logs/12_dynamic_panel_gmm.log", replace text

*------------------------------------------------------------------------------
* 1. Simulate a dynamic panel: 500 firms; keep 12 periods after a burn-in so the
*    process has reached its stationary distribution.
*------------------------------------------------------------------------------
local N    = 500
local Tgen = 20
local Tkeep = 12
local rho  = 0.6
local beta = 1
set obs `N'
generate id = _n
generate alpha = rnormal()                  // unit (firm) fixed effect
expand `Tgen'
bysort id: generate t = _n
xtset id t
generate x   = 0.5*alpha + rnormal()        // regressor, correlated with alpha
generate eps = rnormal()
generate y = .
replace y = alpha + `beta'*x + eps if t==1            // starting value
forvalues s = 2/`Tgen' {                              // iterate the dynamics
    replace y = `rho'*L.y + `beta'*x + alpha + eps if t==`s'
}
drop if t <= `Tgen' - `Tkeep'               // burn-in: discard the early periods
bysort id (t): replace t = _n               // re-index time to 1..12
xtset id t

*------------------------------------------------------------------------------
* 2. The bias: pooled OLS (rho too high) vs. within/FE (rho too low).
*------------------------------------------------------------------------------
regress y L.y x, vce(cluster id)            // pooled OLS: rho biased UP
local rho_ols = _b[L.y]
local se_ols  = _se[L.y]
xtreg y L.y x, fe vce(cluster id)           // within: rho biased DOWN (Nickell)
local rho_fe = _b[L.y]
local se_fe  = _se[L.y]

*------------------------------------------------------------------------------
* 3. Arellano-Bond difference GMM and Blundell-Bond system GMM.
*    estat abond: AR(1) in the differenced errors SHOULD reject; AR(2) should NOT
*    -- the key check that the lagged-level moment conditions are valid.
*------------------------------------------------------------------------------
xtabond y x, lags(1) vce(robust)            // difference GMM (Arellano-Bond)
local rho_ab = _b[L.y]
local se_ab  = _se[L.y]
estat abond

xtdpdsys y x, lags(1) vce(robust)           // system GMM (Blundell-Bond)
local rho_bb = _b[L.y]
```

```
local se_bb  = _se[L.y]
estat abond

*-------------------------------------------------------------------------------
* 4. Persistence rho across estimators (true rho = 0.6).
*-------------------------------------------------------------------------------
display "rho:  OLS=" %5.3f `rho_ols' "  FE=" %5.3f `rho_fe' ///
        "  DiffGMM=" %5.3f `rho_ab' "  SysGMM=" %5.3f `rho_bb'

capture file close dp
file open dp using "output/12_dynpanel_table.tex", write replace
file write dp "\begin{table}[htbp]\centering" _n
file write dp "\caption{Dynamic panel: the persistence \(\rho\) (true \(\rho=0.6\))}" _n
file write dp "\begin{tabular}{lcccc}" _n "\toprule" _n
file write dp " & Pooled OLS & Within FE & Diff GMM & Sys GMM \\" _n "\midrule" _n
file write dp "\(\hat\rho\) (coef.\ on \(y_{t-1}\)) & " ///
    %5.3f (`rho_ols') " & " %5.3f (`rho_fe') " & " %5.3f (`rho_ab') " & " %5.3f (`rho_bb') " \\"
    _n
file write dp "(std.\ err.) & (" %5.3f (`se_ols') ") & (" %5.3f (`se_fe') ") & (" ///
    %5.3f (`se_ab') ") & (" %5.3f (`se_bb') ") \\" _n
file write dp "\bottomrule" _n "\end{tabular}" _n "\end{table}" _n
file close dp

log close
display "12_dynamic_panel_gmm.do finished OK"
```

## 13.3 The results

With a true persistence of $\rho = 0.6$, the four columns differ only in how each removes (or fails to remove) the unit effect $\alpha_i$:

$$
\begin{aligned}
&\text{(1) Pooled OLS:} && y_{it} = \rho\, y_{i,t-1} + \beta\, \mathrm{x}_{it} + (\alpha_i + \varepsilon_{it}) && (\rho \text{ biased up}),\\
&\text{(2) Within FE:} && \tilde{y}_{it} = \rho\, \tilde{y}_{i,t-1} + \beta\, \tilde{\mathrm{x}}_{it} + \tilde{\varepsilon}_{it} && (\text{Nickell: down}),\\
&\text{(3) Diff GMM:} && \Delta y_{it} = \rho\, \Delta y_{i,t-1} + \beta\, \Delta \mathrm{x}_{it} + \Delta \varepsilon_{it} && (\text{IV: } y_{i,t-2}, \dots),\\
&\text{(4) Sys GMM:} && \text{(3) plus the untransformed levels equation} && (\text{IV: } \Delta y_{i,t-1}),
\end{aligned}
$$

with $\tilde{z}_{it} = z_{it} - \bar{z}_i$ the within (demeaned) transform and $\Delta z_{it} = z_{it} - z_{i,t-1}$ the first difference. Pooled OLS leaves $\alpha_i$ in the error, where it correlates with the lag and biases $\rho$ *up*; within FE demeans it away but the demeaned lag correlates with the demeaned error (Nickell bias), biasing $\rho$ *down*; both GMM estimators first-difference $\alpha_i$ out and instrument the lag with the panel's own history, so only they recover the truth.

Table 13.1: Dynamic panel: the persistence $\rho$ (true $\rho = 0.6$)

| | Pooled OLS | Within FE | Diff GMM | Sys GMM |
|---|---|---|---|---|
| $\hat{\rho}$ (coef. on $y_{t-1}$) | 0.803 | 0.524 | 0.576 | 0.577 |
| (std. err.) | (0.004) | (0.009) | (0.017) | (0.016) |

The Arellano–Bond test confirms the design — AR(1) rejects while AR(2) does not, so the lagged-level instruments are valid:

```
. estat abond
Arellano-Bond test for zero autocorrelation in first-differenced errors
  Order      z       Prob > z
      1   -17.12      0.0000     <- AR(1): expected (differencing induces it)
      2    -0.94      0.3490     <- AR(2): does NOT reject -> instruments valid
```

A significant AR(2) would be the red flag; here $p \approx 0.35$, so the design passes.

# Part IV

# Appendices

# Appendix A

# Python/R → Stata cheat sheet

Table A.1: Common tasks across pandas / R / Stata

| Task | pandas / R | Stata |
|---|---|---|
| Load CSV | `pd.read_csv` / `read_csv` | `import delimited "f.csv", varnames(1) clear` |
| Inspect | `df.head()` / `head()` | `list in 1/5`, `describe`, `codebook` |
| New column | `df["x"]=...` / `mutate` | `generate x = ...` (`replace` to overwrite) |
| Group aggregate | `groupby().agg` | `collapse (mean)..., by(g)` / `egen ..., by(g)` |
| Reshape | `pivot/melt` | `reshape wide` / `reshape long` |
| Join | `merge` | `merge 1:1 key using "f.dta"` |
| OLS robust | `cov_type="HC1"` / `feols` | `regress y x, vce(robust)` |
| Fixed effects | `PanelOLS` / `feols(.\|fe)` | `xtreg ..., fe` / `reghdfe ..., absorb()` |
| Marginal effects | `get_margeff()` / `margins` | `margins, dydx(x)` |
| Lag | `.shift()` / `lag()` | `L.x` (after `tsset`) |

# Appendix B

# The methods, explained

This appendix is the explanatory companion to the body. The chapters state each method crisply assuming familiarity with the econometrics; here we build the nine estimators up from scratch. Each entry runs through the same beats: the *idea*; the *symbols*; what the defining *formula* means; *why it works* (why the estimator recovers the effect); *the catch* (the identifying assumption and what breaks it); a *worked example* to follow with pen and paper; and *in the chapter* — how that chapter's own run recovers the planted answer. (Stata basics, data management, and data import are tooling rather than estimators, so they are not repeated here.)

## B.1 Ordinary least squares

↪ *worked Stata chapter: Chapter 4*

**The idea.** OLS draws the straight line (or hyperplane) through a cloud of points that makes the total *squared* vertical distance to the points as small as possible. Each slope $\beta_j$ answers a *ceteris paribus* question: "if $x_j$ rises by one unit and the other regressors are held fixed, how much does the average of $y$ move?"

**Symbols.** $y$ the outcome (a car's price); $x_1, \ldots, x_k$ the regressors (mileage, weight); $\beta_0$ the intercept and $\beta_1, \ldots, \beta_k$ the slopes; $\varepsilon$ the error — everything else that moves $y$; $n$ the number of observations; $X$ the $n \times (k{+}1)$ matrix stacking a column of ones and the regressors.

**The formula.** Choose $\beta$ to minimise the sum of squared residuals $\sum_i (y_i - x_i'\beta)^2$. Differentiating and setting the gradient to zero gives the *normal equations* $X'(y - X\beta) = 0$, whose solution is

$$\hat{\beta} = (X'X)^{-1}X'y.$$

**Why it works.** Geometrically $\hat{\beta}$ *projects* $y$ onto the space spanned by the regressors: it makes the residual orthogonal to every $x_j$, so no linear function of the regressors can predict what is left over — that orthogonality is precisely the content of the normal equations. When the error variance is non-constant the point estimate is unchanged but its standard error must use the heteroskedasticity-robust "sandwich" $(X'X)^{-1}\big(\sum_i \hat{\varepsilon}_i^2 x_i x_i'\big)(X'X)^{-1}$.

**The catch.** $\hat{\beta}$ recovers the *causal* effect only if the regressors are *exogenous*, $E[\varepsilon \mid X] = 0$ — no omitted variable correlated with $x$, no reverse causality. The orthogonality OLS imposes *in sample* (residual $\perp x$) is mechanical; the assumption is that it also holds in the *population*. When it fails — unobserved ability in a wage–schooling regression — the slope is biased, which is exactly what instrumental variables repairs.

**Worked example.** Three points $(x, y) = (1, 2), (2, 2), (3, 5)$, so $\bar{x} = 2$ and $\bar{y} = 3$:

$$\text{slope} = \frac{\sum(x - \bar{x})(y - \bar{y})}{\sum(x - \bar{x})^2} = \frac{(-1)(-1) + 0 + (1)(2)}{1 + 0 + 1} = \frac{3}{2} = 1.5,$$

$$\text{intercept} = \bar{y} - 1.5\,\bar{x} = 3 - 3 = 0.$$

The fitted line is therefore $\hat{y} = 1.5\,x$.

**In the chapter.** Chapter 4 fits car price on mileage, weight, and origin in Stata's 74-car `auto` data with robust standard errors. Adding the foreign indicator and an interaction lifts the fit from $R^2 = 0.29$ to $0.52$, and weight enters strongly ($\approx 1{,}750$ per $1{,}000$ lbs). Because these are real data there is no planted coefficient to recover — the chapter reads the table mechanically, not as economics.

## B.2 Panel data, difference-in-differences, and event studies

↪ *worked Stata chapter: Chapter 5*

**The idea.** The goal is the effect of a treatment that some units receive at a known date. Comparing treated to untreated confounds pre-existing differences between them; comparing each group's before to after confounds whatever else changed over time. Difference-in-differences removes both at once: take each group's *change* across the date, then difference those two changes.

**Symbols.** $y_{it}$ outcome for unit $i$ in period $t$; $\alpha_i$ a unit fixed effect (time-invariant traits, differenced away); $\gamma_t$ a time fixed effect (shocks common to all units); $\text{treat}_i$ marks the eventually-treated group, $\text{post}_t$ the post period; $\beta$ the DiD effect.

**The formula.** The $2 \times 2$ DiD is the difference of two differences,

$$\beta = \big(\bar{y}_{\text{tr,post}} - \bar{y}_{\text{tr,pre}}\big) - \big(\bar{y}_{\text{co,post}} - \bar{y}_{\text{co,pre}}\big),$$

which is identical to the coefficient on the interaction $\text{treat}_i \times \text{post}_t$ in the two-way fixed-effects regression $y_{it} = \alpha_i + \gamma_t + \beta(\text{treat}_i \times \text{post}_t) + \varepsilon_{it}$.

**Why it works.** The control group's change estimates the common trend — what *would* have happened to the treated group anyway. Subtracting it strips that trend out of the treated group's change, leaving only the treatment. The unit fixed effects $\alpha_i$ absorb every fixed difference between units, so the comparison is made within unit, over time.

**The catch.** *Parallel trends*: absent treatment, the treated group's average would have moved in step with the control group's. It cannot be tested directly (the counterfactual is unobserved), but the *event study* — the path of effects by period relative to treatment — gives indirect evidence: if the pre-treatment "leads" sit flat near zero, the groups were moving together beforehand, which is reassuring.

**Reading the event study.** The dynamic version swaps the single $\beta$ for one coefficient $\beta_k$ per *event time* $k = t - t_i^*$ (periods relative to a unit's own treatment date $t_i^*$), fitted for every $k$ in the data *except* $k = -1$:

$$k: \underbrace{-5 \;\; -4 \;\; -3 \;\; -2}_{\text{leads : } \beta_k \approx 0} \;\Big|\; \underbrace{-1}_{\text{baseline}} \;\Big|\; \underbrace{0 \;\; 1 \;\; 2 \;\; 3 \;\; 4 \;\; 5 \;\; 6}_{\text{lags : the effects}}$$

One indicator must be dropped — the full lead/lag set is collinear with the fixed effects — and $k = -1$, the eve of treatment, is the conventional baseline ($\beta_{-1} \equiv 0$), so every $\beta_k$ is read relative to it. The *lags* ($k \geq 0$) are the effects of interest: the treated-minus-control gap $0, 1, 2, \ldots$ periods after onset. The *leads* ($k \leq -2$) are kept on purpose — they measure that same gap *before* treatment, where it ought to be zero, so they act as a placebo test. Flat, insignificant leads are the visual parallel-trends check; a large lead would mean the groups were already diverging, leaving the post-treatment estimates untrustworthy. Summing over $k \neq -1$ rather than only $k \geq 0$ is precisely what builds that test in — the advantage an event study has over a single DiD number.

**Worked example.** Treated mean $10 \to 16$ (change $+6$); control mean $10 \to 12$ (change $+2$). DiD $= 6 - 2 = 4$: of the treated group's six-point rise, two points are the common trend and four are the effect.

**In the chapter.** Chapter 5 simulates 150 firms over 2010–2021 with treatment switched on in 2015 whose effect grows $0.5$ per year — a true *average* post-treatment effect of $2.0$. Pooled OLS, firm FE, and two-way FE all return $1.973$ (SE $0.104$), and the event-study leads sit flat at zero while the lags climb: the estimator recovers the planted truth and shows no pre-trend.

## B.3 Time series

↪ *worked Stata chapter: Chapter 6*

**The idea.** In a series where today depends on yesterday, the observations are not independent draws — they carry information forward. Two consequences follow: we can *model* the persistence (an autoregression), and when we don't, we must *correct* ordinary standard errors, which otherwise treat correlated observations as independent and so overstate precision.

**Symbols.** $y_t$ the value at time $t$; $\phi$ the persistence (AR coefficient); $c$ an intercept; $\varepsilon_t$ the period-$t$ shock; $L$ the lag operator, $Ly_t = y_{t-1}$.

**The formula.** A first-order autoregression is $y_t = c + \phi\, y_{t-1} + \varepsilon_t$. Iterating it, a shock feeds into the next period at rate $\phi$, the one after at $\phi^2$, and so on, so its effect $k$ periods out is $\phi^k$. When residuals stay serially correlated, the Newey–West (HAC) variance reweights the autocovariances up to a chosen lag $L$:

$$\hat{\Omega} = \hat{\Gamma}_0 + \sum_{\ell=1}^{L} \left(1 - \tfrac{\ell}{L+1}\right)\left(\hat{\Gamma}_\ell + \hat{\Gamma}_\ell'\right).$$

**Why it works.** The geometric decay $\phi^k$ means distant shocks barely register, so the cumulative effect of a one-unit shock is the finite sum $1 + \phi + \phi^2 + \cdots = 1/(1-\phi)$. Newey–West keeps the usual point estimate but adds back the cross-period covariances that the independent-errors formula ignores, with the triangular weights $\left(1 - \ell/(L+1)\right)$ guaranteeing a positive variance.

**The catch.** *Stationarity* — for the AR(1), $|\phi| < 1$, so the series reverts rather than wandering off and its mean and variance are stable. A nearby trap is a *level break*: an unmodelled shift makes the series look more persistent than it is and biases $\hat{\phi}$ upward — control for the break and the bias disappears.

**Worked example.** With $\phi = 0.6$, a one-unit shock raises $y$ by $1$ now, $0.6$ next period, $0.36$ after that, …; the cumulative effect is $1/(1 - 0.6) = 2.5$.

**In the chapter.** Chapter 6 simulates a monthly AR(1) with $\phi = 0.6$ plus a $+3$ level shift in 2015. Regressing $y$ on its lag *without* the break gives $\hat{\phi} = 0.85$ — inflated, the level-break trap exactly; adding the break indicator recovers $\hat{\phi} = 0.57$ ($\approx 0.6$), and Newey–West (lag 3) widens the standard errors for the residual serial correlation.

## B.4 Instrumental variables (2SLS)

↪ *worked Stata chapter: Chapter 8*

**The idea.** If $x$ is correlated with the error — schooling and unobserved ability both raise wages — OLS cannot separate the effect of $x$ from the confounder, and its slope is biased. An *instrument* $z$ is a variable that moves $x$ but reaches $y$ *only through* $x$. Keeping only the variation in $x$ that $z$ explains discards the contaminated part and leaves a clean comparison.

**Symbols.** $y$ outcome; $x$ the endogenous regressor; $z$ the instrument; $\beta$ the causal slope; $\pi$ the first-stage slope; $\varepsilon$ the structural error.

**The formula.** *First stage*: regress $x$ on $z$, $x = \pi z + v$, and form the fitted $\hat{x}$ (the part of $x$ driven by $z$). *Second stage*: regress $y$ on $\hat{x}$. With a single instrument the two stages collapse to the Wald ratio

$$\hat{\beta}_{\text{IV}} = \frac{\widehat{\text{Cov}}(z, y)}{\widehat{\text{Cov}}(z, x)}.$$

**Why it works.** Because $z$ touches $y$ only through $x$, the numerator $\text{Cov}(z, y)$ equals $\beta$ times $\text{Cov}(z, x)$; dividing by the denominator returns $\beta$. The confounder, being unrelated to $z$, drops out — that is the whole trick. The price is precision: only the $z$-driven slice of $x$ is used, so IV standard errors exceed OLS's.

**The catch.** Two requirements. *Relevance*, $\text{Cov}(z, x) \neq 0$ — the instrument must actually move $x$, or the denominator is near zero and the estimate explodes (the "weak instrument" problem, watched with the first-stage $F$, rule of thumb $> 10$). *Exclusion*, $\text{Cov}(z, \varepsilon) = 0$ — no back-door from $z$ to $y$. Exclusion is not testable; it is argued from how $z$ arises.

**Worked example.** A lottery $z$ raises program take-up $x$ by $0.4$ and raises the outcome $y$ by $0.8$. Then $\hat{\beta}_{\text{IV}} = 0.8/0.4 = 2$: each unit of participation raises $y$ by 2.

**In the chapter.** Chapter 8 simulates an endogenous $x$ with a true slope of $2.0$ and a confounder that pushes OLS upward. OLS returns $2.173$ (biased high); 2SLS with two instruments returns $1.960$ and the high-dimensional-FE version $1.994$ — both land on $2.0$ once the confounded variation is purged, and the first-stage $F$ sits far above the weak-instrument bar.

## B.5 Binary outcome models

↪ *worked Stata chapter: Chapter 9*

**The idea.** When $y$ is $0/1$, fitting a straight line (the linear probability model) can predict probabilities below $0$ or above $1$ and forces a constant marginal effect. Logit and probit instead push a linear index $x'\beta$ through an S-shaped curve pinned to $[0, 1]$ that flattens at both ends.

**Symbols.** $y \in \{0, 1\}$; $x'\beta$ the linear index; $\Lambda$ the logistic CDF (logit); $\Phi$ the standard-normal CDF (probit); $g$ the matching density ($\Lambda'$ or $\phi$).

**The formula.** $P(y = 1 \mid x) = G(x'\beta)$ with $G = \Lambda$ or $\Phi$. Because $G$ is nonlinear, a coefficient $\beta_j$ is *not* the effect on the probability; the comparable summary is the average marginal effect, which averages the curve's slope over the sample,

$$\text{AME}_j = \frac{1}{n} \sum_i g(x_i'\beta)\, \beta_j.$$

For logit, $e^{\beta_j}$ is the odds ratio.

**Why it works.** The slope of the probability curve, $g(x'\beta)\beta_j$, is steepest in the middle and shallow at the extremes — a one-unit change in $x$ moves the probability a lot for a borderline case and little for an almost-certain one. Averaging that slope across the sample yields a single number on the same footing as an OLS coefficient, which is why the AME, not $\beta_j$, is the quantity to report.

**The catch.** The index is assumed linear and the link (logistic or normal) correctly specified. In practice the choice barely matters: logit and probit deliver nearly identical marginal effects, differing mainly in the scale of the raw coefficients, not the AMEs.

**Worked example.** A logit index of $0$ gives $P = \Lambda(0) = 0.5$; an index of $1$ gives $P = \Lambda(1) = e/(1 + e) \approx 0.73$. A coefficient of $0.7$ multiplies the *odds* by $e^{0.7} \approx 2$.

**In the chapter.** Chapter 9 models whether a car is foreign from its mileage, weight, and price. The logit coefficient on mileage is $-0.121$ — not a probability effect; the average marginal effect is $-0.009$, essentially identical to the probit AME ($-0.009$) and close to the LPM slope ($-0.015$): the textbook "logit ≈ probit" result. The odds ratio on mileage is $0.886$.

## B.6 Staggered-adoption difference-in-differences

↪ *worked Stata chapter: Chapter 10*

**The idea.** When units adopt treatment in *different* years, an ordinary two-way fixed-effects regression quietly uses already-treated units as if they were clean controls for later-treated ones. Because those "controls" are still responding to their own treatment, the comparison is contaminated — a "forbidden comparison" whose negative weights can shrink or even flip the estimate. Callaway–Sant'Anna rebuild the estimator from clean $2 \times 2$ blocks that only ever use not-yet-treated or never-treated units as controls, then average the blocks.

**Symbols.** $G = g$ the cohort (the first period a unit is treated); $t$ calendar time; $C$ a clean comparison group (never-treated or not-yet-treated); $ATT(g,t)$ the effect for cohort $g$ at time $t$; $e = t - g$ event time.

**The formula.** The building block compares a cohort's change since just before its own treatment to a clean control group's change over the same window,

$$ATT(g,t) = \underbrace{E[Y_t - Y_{g-1} \mid G = g]}_{\text{cohort } g\text{'s change}} - \underbrace{E[Y_t - Y_{g-1} \mid C]}_{\text{clean controls' change}},$$

and these $ATT(g,t)$ are then averaged — overall, by event time $e$, or by cohort.

**Why it works.** Each block is an honest DiD: a single cohort against controls that have *not* yet been treated, so no already-treated unit ever enters the control group. Keeping the cohorts separate until the final averaging step is what removes the negative weighting the pooled regression introduces.

**The catch.** Parallel trends must hold relative to the clean comparison group, and there must be *no anticipation* — units do not change behaviour before $g$ in expectation of treatment. When effects are dynamic and timing is staggered, it is the naive pooled estimate to distrust, not this one.

**Worked example.** A cohort treated in 2015 rises by $5$ from 2014 to 2016; the not-yet-treated units rise by $2$ over the same span. Then $ATT(2015, 2016) = 5 - 2 = 3$.

**In the chapter.** Chapter 10 simulates two cohorts (treated in 2014 and 2017) plus a never-treated group, with an effect that grows $0.4$ per year after each cohort's own onset (a true average near $1.5$). Callaway–Sant'Anna recovers $1.50$; the naive two-way FE on the *same* data returns only $1.01$ — the staggered-timing bias, made visible.

## B.7 Regression discontinuity

↪ *worked Stata chapter: Chapter 11*

**The idea.** Imagine a treatment that turns on the exact moment a continuous score hits a specific line. The individuals landing a fraction below and a fraction above that line are essentially twins; they share the same background, talent, and luck. The only difference is that one group got the treatment and the other missed it. By comparing these two identical groups right at the border, any sudden jump in their outcomes reveals the exact impact of the treatment. The threshold naturally creates a natural, cost-free randomized experiment for anyone nearby.

**Symbols.** $Y$ is the outcome (i.e. subsequent earnings, enrollment); $X$ is the running variable (the continuous score determining treatment, e.g., a test score); $c$ is the threshold or cutoff; $\tau$ (tau) is the treatment effect, measured by the vertical jump at the cutoff.

**The formula.** $E[Y \mid X = x]$ traces the average outcome as a function of the score; $\tau$ is its jump at $c$,

$$\tau = \underbrace{\lim_{x \downarrow c} E[Y \mid X = x]}_{\text{just above (treated)}} - \underbrace{\lim_{x \uparrow c} E[Y \mid X = x]}_{\text{just below (control)}}.$$

The arrows are one-sided limits: $x \downarrow c$ approaches $c$ from *above* (the treated side), $x \uparrow c$ from *below* (the control side).

**Why it works.** A student scoring $60$ (just qualifies) and one scoring $59.9$ (just misses) are alike in every way except the treatment, so nothing but the treatment can explain a jump in their outcomes *exactly* at $c$. A treated and an untreated unit cannot be observed at the *same* score, so the fit approaches $c$ from each side — a line on each side — and reads off the gap between them. Away from $c$ the curve is smooth; only at $c$ does treatment create a step.

**The catch.** *Continuity*: absent treatment, average outcomes would pass smoothly through $c$, so any jump is the treatment's doing. It fails if units can *manipulate* the score to land on the favourable side (a grader nudging a $59$ up to a $60$); that sorting puts a jump at $c$ that is not the treatment. A density/manipulation test checks whether units pile up just past the cutoff.

**Worked example.** Just below the cutoff the fitted line reaches $3.0$; just above it reaches $5.0$. Then $\tau = 5.0 - 3.0 = 2.0$.

**In the chapter.** Chapter 11 plants a jump of $2.0$ at the cutoff. `rdrobust` (local-linear, bias-corrected) estimates $\hat{\tau} = 1.938$ with a robust 95% CI of $[1.770,\ 2.082]$, and the `rddensity` manipulation test returns $p = 0.95$ — no bunching, because the $p$-value is high, we fail to reject the null hypothesis of a smooth density. This proves there is no manipulation of scores to "bunch" just above the threshold; confirming the design is clean, valid quasi-experiment, and the estimate lands on the truth.

## B.8 Synthetic control

↪ *worked Stata chapter: Chapter 12*

**The idea.** For a single treated unit — one state, one country — there is no obvious comparison group, so one is *built*: a weighted average of untreated "donor" units, with the weights chosen so the blend tracks the treated unit's path *before* treatment. Afterwards, the gap between the real unit and this synthetic twin is the estimated effect.

**Symbols.** Unit $1$ is treated, units $j \geq 2$ are donors; $w_j \geq 0$ are weights summing to one; $Y_{1t}$ is the treated outcome, $Y_{jt}$ the donor outcomes; $X_1, X_0$ are the pre-treatment characteristics of the treated unit and of the donors.

**The formula.** Choose the weights to match the pre-treatment period as closely as possible,

$$\min_{W} (X_1 - X_0 W)' V (X_1 - X_0 W), \qquad w_j \geq 0,\ \textstyle\sum_j w_j = 1,$$

then form the synthetic control $\sum_j w_j Y_{jt}$ and read the effect as the post-treatment gap $Y_{1t} - \sum_j w_j Y_{jt}$.

**Why it works.** If the weighted donors reproduce the treated unit's outcome for many pre-treatment periods, they plausibly share its underlying drivers (its exposure to the common shocks), so they stand in for what the treated unit *would* have done. The non-negativity and sum-to-one constraints keep the synthetic unit inside the donors' range — no extrapolation beyond the data.

**The catch.** A credible *pre-treatment fit* (a poor match means a poor counterfactual), *no interference* (the treatment must not spill onto the donors), and a comparable donor pool. With only one treated unit, inference is by *placebo*: pretend each donor was treated, compute its gap, and ask whether the real unit's gap is unusually large against that distribution.

**Worked example.** Two donors with weights $0.6$ and $0.4$ and post-treatment outcomes $8$ and $13$ give a synthetic value $0.6(8) + 0.4(13) = 10$. If the treated unit is $13$, the effect is $13 - 10 = 3$.

**In the chapter.** Chapter 12 plants a $+3$ effect on the treated unit from period $21$. The fitted synthetic control tracks it closely beforehand, and the post-treatment gaps average about $3.0$ (e.g. $2.96,\ 3.31,\ 3.25, \ldots$); every placebo $p$-value is essentially zero, so the real unit's gap stands out sharply from the donor placebos.

## B.9 Dynamic panel data (GMM)

↪ *worked Stata chapter: Chapter 13*

**The idea.** When today's outcome depends on yesterday's — leverage, investment, output all have momentum — the model needs a lagged dependent variable. But alongside unit fixed effects that lag is endogenous (it contains the fixed effect), and the standard estimators are biased in *opposite* directions. The fix is to instrument the lag with the panel's own deeper history.

**Symbols.** $y_{it}$ outcome; $\rho$ persistence (the coefficient on $y_{i,t-1}$); $\alpha_i$ the unit effect; $\Delta$ the first difference.

**The formula.** First-difference to kill $\alpha_i$,

$$\Delta y_{it} = \rho\, \Delta y_{i,t-1} + \beta\, \Delta x_{it} + \Delta \varepsilon_{it},$$

then instrument $\Delta y_{i,t-1}$ with lagged *levels* $y_{i,t-2}, y_{i,t-3}, \ldots$ (difference GMM). System GMM adds the untransformed levels equation, using lagged *differences* as its instruments.

**Why it works.** A level two or more periods back is correlated with $\Delta y_{i,t-1}$ but, after differencing, uncorrelated with $\Delta \varepsilon_{it}$ — exactly the relevance and exogeneity an instrument needs. GMM stacks the many such moment conditions and weights them efficiently.

**The catch.** The lagged-level instruments are valid only if $\varepsilon_{it}$ is itself serially uncorrelated — precisely what the AR(2) test checks. The instrument count also grows with $T^2$, so deep lags are often "collapsed"; and system GMM is preferred when $\rho$ is near one, where deep lags become weak instruments in differences.

**Worked example.** Pooled OLS reads $\hat{\rho} = 0.80$ and within FE reads $0.52$; the truth $(0.6)$ is bracketed between them. Instrumenting the lag with its own history, GMM returns $0.58$ — inside the bracket and close to the truth.

**In the chapter.** Chapter 13 simulates 500 firms over 12 periods with $\rho = 0.6$. Pooled OLS returns $0.803$, within FE $0.524$, and both difference and system GMM about $0.58$; the AR(2) test does not reject ($p \approx 0.35$), validating the lagged-level instruments.

# References


**Methods.**

- **White, Halbert**, 1980, A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity, *Econometrica* 48, 817–838. 10.2307/1912934
- **Newey, Whitney K., and Kenneth D. West**, 1987, A simple, positive semi-definite, heteroskedasticity and autocorrelation consistent covariance matrix, *Econometrica* 55, 703–708. 10.2307/1913610
- **Staiger, Douglas, and James H. Stock**, 1997, Instrumental variables regression with weak instruments, *Econometrica* 65, 557–586. 10.2307/2171753
- **Goodman-Bacon, Andrew**, 2021, Difference-in-differences with variation in treatment timing, *Journal of Econometrics* 225, 254–277. 10.1016/j.jeconom.2021.03.014
- **Callaway, Brantly, and Pedro H. C. Sant'Anna**, 2021, Difference-in-differences with multiple time periods, *Journal of Econometrics* 225, 200–230. 10.1016/j.jeconom.2020.12.001
- **Calonico, Sebastian, Matias D. Cattaneo, and Rocío Titiunik**, 2014, Robust nonparametric confidence intervals for regression-discontinuity designs, *Econometrica* 82, 2295–2326. 10.3982/ECTA11757
- **Cattaneo, Matias D., Michael Jansson, and Xinwei Ma**, 2020, Simple local polynomial density estimators, *Journal of the American Statistical Association* 115, 1449–1455. 10.1080/01621459.2019.1635480
- **Abadie, Alberto, Alexis Diamond, and Jens Hainmueller**, 2010, Synthetic control methods for comparative case studies, *Journal of the American Statistical Association* 105, 493–505. 10.1198/jasa.2009.ap08746
- **Nickell, Stephen**, 1981, Biases in dynamic models with fixed effects, *Econometrica* 49, 1417–1426. 10.2307/1911408
- **Arellano, Manuel, and Stephen Bond**, 1991, Some tests of specification for panel data: Monte Carlo evidence and an application to employment equations, *Review of Economic Studies* 58, 277–297. 10.2307/2297968
- **Blundell, Richard, and Stephen Bond**, 1998, Initial conditions and moment restrictions in dynamic panel data models, *Journal of Econometrics* 87, 115–143. 10.1016/S0304-4076(98)00009-8

**Canonical applications** (cited in the "In practice" notes).

- **Abadie, Alberto, and Javier Gardeazabal**, 2003, The economic costs of conflict: a case study of the Basque Country, *American Economic Review* 93, 113–132. 10.1257/000282803321455188

- **Angrist, Joshua D.**, 1990, Lifetime earnings and the Vietnam era draft lottery: evidence from social security administrative records, *American Economic Review* 80, 313–336.
- **Angrist, Joshua D., and Alan B. Krueger**, 1991, Does compulsory school attendance affect schooling and earnings?, *Quarterly Journal of Economics* 106, 979–1014. 10.2307/2937954
- **Angrist, Joshua D., and Victor Lavy**, 1999, Using Maimonides' rule to estimate the effect of class size on scholastic achievement, *Quarterly Journal of Economics* 114, 533–575. 10.1162/003355399556061
- **Card, David, and Alan B. Krueger**, 1994, Minimum wages and employment: a case study of the fast-food industry in New Jersey and Pennsylvania, *American Economic Review* 84, 772–793.
- **Lee, David S.**, 2008, Randomized experiments from non-random selection in U.S. House elections, *Journal of Econometrics* 142, 675–697. 10.1016/j.jeconom.2007.05.004
- **Mincer, Jacob**, 1974, Schooling, Experience, and Earnings (Columbia University Press for the NBER, New York).
- **Thistlethwaite, Donald L., and Donald T. Campbell**, 1960, Regression-discontinuity analysis: an alternative to the ex post facto experiment, *Journal of Educational Psychology* 51, 309–317. 10.1037/h0044319