

Package: nonabsdid (via r-universe)

September 17, 2026

Type Package

Title Visualize Heterogeneity-Robust Event Studies for Non-Absorbing Treatments

Version 0.4.1.9000

Date 2026-06-23

Description Runs several heterogeneity-robust difference-in-differences (DID) event-study estimators for non-absorbing (i.e., treatment can switch on and off over time, allowing treatment reversal) binary treatments through their respective packages, harmonizes their output onto a common time axis and tidy data structure, and overlays them in a single 'ggplot2' panel for visual comparison. Supported estimators include those provided by 'DIDmultiplgtDYN', 'PanelMatch', and 'fect', with an optional naive two-way fixed-effects reference series via 'fixest'. The underlying methods are respectively described in Clement de Chaisemartin and Xavier D'Haultfoeuille. ``Difference-in-Differences Estimators of Intertemporal Treatment Effects." The Review of Economics and Statistics (2026) <doi:10.1162/rest\_a\_01414>, Kosuke Imai, In Song Kim, and Erik H. Wang. ``Matching methods for causal inference with time-series cross-sectional data." American Journal of Political Science 67.3 (2023) <doi:10.1111/ajps.12685>, Licheng Liu, Ye Wang, and Yiqing Xu. ``A practical guide to counterfactual estimators for causal inference with time-series cross-sectional data." American Journal of Political Science 68.1 (2024) <doi:10.1111/ajps.12723>, and Laurent R. Bergé, Kyle Butts, and Grant McDermott. ``Fast and user-friendly econometrics estimations: The R package 'fixest'." arXiv preprint (2026) <doi:10.48550/arXiv.2601.21749>. A single nabs_event_study() wrapper runs any supported estimator with a common interface; nabs_event_study_simple() provides a one-line front door for quick exploratory runs; the S3 generic as_nabs_event_study() coerces estimator output into a tidy stable schema; and nabs_event_plot() overlays multiple methods on a single

'ggplot2' panel, with optional naive two-way fixed effects
drawn in a neutral color as a reference.

License MIT + file LICENSE

Encoding UTF-8

Depends R (>= 4.1.0)

Imports cli, dplyr, ggplot2, rlang, stats, tibble

Suggests DIDmultiplegtDYN, polars, PanelMatch, fect, fixest, haven,
knitr, rmarkdown, testthat (>= 3.0.0), vdiffr, withr

Config/testthat/edition 3

URL <https://github.com/takuma1102/nonabsdid>,
<https://takuma1102.github.io/nonabsdid/>

BugReports <https://github.com/takuma1102/nonabsdid/issues>

VignetteBuilder knitr

Additional_repositories <https://community.r-multiverse.org>

Config/roxygen2/version 8.0.0

Repository <https://ropensci-staging.r-universe.dev>

Date/Publication 2026-07-07 21:23:51 UTC

RemoteUrl <https://github.com/takuma1102/nonabsdid>

RemoteRef HEAD

RemoteSha 92af671e0c99d6cc5bbe1eff4c0b813dac473707

Contents

aggregate_effects	3
as_nabs_effect_cells	3
as_nabs_event_study	6
nabs_effect_cells	10
nabs_event_plot	12
nabs_event_study	14
nabs_event_study_simple	17
nabs_read_dta	20
nabs_write_dta	21
naive_twfe	23
plot_effect_matrix	24

Index	27
--------------	-----------

aggregate_effects	<i>Collapse effect cells back onto an event-study path</i>
-------------------	--

Description

Aggregates a ‘nabs_effect_cell_tbl’ over cohorts to recover a one-dimensional path, returning a ‘nabs_event_study_tbl’ that plugs straight into [nabs_event_plot()]. This makes explicit that the event study is the cohort-collapsed view of the same cells.

Usage

```
aggregate_effects(cells, by = c("event_time", "calendar_time"))
```

Arguments

cells	A ‘nabs_effect_cell_tbl’.
by	Aggregation axis: “event_time” (default) or “calendar_time”.

Details

Point estimates are averaged across cohorts (weighted by ‘n’ when present). Re-aggregated standard errors are **not** computed here – collapsing SEs correctly needs the estimator’s replicate draws – so ‘std.error’ and the CI columns are returned as ‘NA’. Use this for a quick overlay, not for inference.

Value

A ‘nabs_event_study_tbl’ (with ‘NA’ standard errors).

Examples

```
raw <- expand.grid(cohort = 3:6, event_time = -2:4)
raw$estimate <- with(raw, ifelse(event_time < 0, 0, 0.2 * event_time))
cells <- as_nabs_effect_cells(raw, method = "FE")
aggregate_effects(cells)
```

as_nabs_effect_cells	<i>Coerce an estimator result to a tidy cohort-by-time effect-cell tibble</i>
----------------------	---

Description

‘as_nabs_effect_cells()’ is an S3 generic that converts the native output of a supported estimator into a *cohort x time* effect-cell schema – the input for [plot_effect_matrix()] heatmaps. It is the two-dimensional companion to [as_nabs_event_study()]: where the event-study schema collapses everything onto a single relative-time axis, this schema keeps the cohort (treatment onset period) as a second dimension so heterogeneity *across* cohorts stays visible.

Usage

```

as_nabs_effect_cells(
  x,
  method = NULL,
  outcome = NA_character_,
  conf.level = 0.95,
  ...
)

## S3 method for class 'data.frame'
as_nabs_effect_cells(
  x,
  method = NULL,
  outcome = NA_character_,
  conf.level = 0.95,
  ...
)

## S3 method for class 'did_multiplegt_dyn'
as_nabs_effect_cells(
  x,
  method = NULL,
  outcome = NA_character_,
  conf.level = 0.95,
  ...
)

## S3 method for class 'fect'
as_nabs_effect_cells(
  x,
  method = NULL,
  outcome = NA_character_,
  conf.level = 0.95,
  axis = c("event", "calendar"),
  weighted = TRUE,
  ...
)

```

Arguments

<code>x</code>	A supported estimator object, or a data frame with at least ‘cohort’, ‘event_time’, and ‘estimate’ columns.
<code>method</code>	Optional override for the ‘method’ column.
<code>outcome</code>	Optional outcome name recorded in the ‘outcome’ column.
<code>conf.level</code>	Confidence level used to derive ‘conf.low’ / ‘conf.high’ from ‘std.error’ when explicit bounds are not supplied. Default ‘0.95’.
<code>...</code>	Method-specific arguments (e.g. ‘axis’, ‘weighted’ for the ‘fect’ method).

axis	For the matrix axes only: <code>"event"</code> keeps <code>'event_time'</code> (default), <code>"calendar"</code> additionally fills <code>'calendar_time'</code> . Both columns are always present; this only affects which one <code>[plot_effect_matrix()]</code> defaults to.
weighted	Logical; weight the within-cell mean of <code>'eff'</code> by <code>'W.agg'</code> . Default <code>'TRUE'</code> .

Details

DCDH method

Expects a `'did_multiplegt_dyn'` object ***run with the 'by' option***, where the `'by'` variable is a unit-level onset cohort (e.g. each unit's first treated period). When `'by'` is set, the object is reshaped into one sublist per `'by'` level, each carrying its own event-study `'plot$data'` (`'Time'`, `'Estimate'`, `'LB.CI'`, `'UB.CI'`, and sometimes `'SE'`). This method walks those sublists and stacks them into the cohort-by-time schema, shifting the axis so onset sits at `'event_time = 0'` (the same `'-1'` shift the event-study tidier applies).

Building the cohort `'by'` variable and running DCDH for you is exactly what `[nabs_effect_cells()]` with `'method = "DCDH"'` does; call the generic directly only when you already have a `'by'`-run object in hand.

SEs are the estimator's own (`'se_method = "native"'`) when `'Time'`-level SEs are present in the plot data; otherwise they are recovered from the symmetric `'LB.CI'/'UB.CI'` bounds (`'se_method = "ci"'`), which is exact for DCDH's normal CIs, so `'show_se'` works either way.

fect method

Uses `'fect::imputed_outcomes()'` (fect $\geq 2.4.0$), the documented long-form accessor that returns one row per treated cell with columns `'id'`, `'time'`, `'event.time'`, `'cohort'`, `'eff'`, and `'W.agg'`. The cell estimate for each `'(cohort, event_time)'` group is the `'W.agg'`-weighted mean of the cell-level effects `'eff'` (set `'weighted = FALSE'` for an unweighted mean).

Standard errors come from the bootstrap surface: when the fit was produced with `'se = TRUE'` and `'keep.sims = TRUE'`, `'imputed_outcomes(replicates = TRUE)'` is re-aggregated within each replicate, and the cell SE is the standard deviation across replicates (with percentile CIs). Without stored sims the SE / CI columns are `'NA'` and `'se_method'` is `"none"`.

Value

A tibble of class `"nabs_effect_cell_tbl"`, one row per `'(cohort, event_time)'` cell, with columns documented in `[new_effect_cell_tbl()]`.

Status

This is an ***experimental*** feature line, separate from the stable event-study API. Only the `'fect'` family (`'IFE'` / `'FE'` / `'MC'`) and `'DCDH'` (`'DIDmultiplegtDYN'`) are supported; `'PanelMatch'` is deliberately omitted for now because a faithful cohort breakdown there needs the matched-set bootstrap to be re-aggregated by cohort, which is out of scope for this pass.

Cohort and event-time conventions

* `'cohort'` is the treatment ***onset calendar period*** (the first period a unit is treated). For repeated on/off treatment this is the **first** onset, so interpret later periods through the estimator's own carryover handling. * `'event_time'` is the relative period with `'0'` at onset, matching the

'nabs_event_study_tbl' convention. For 'fect' this is computed directly as 'calendar_time - cohort'; for 'DCDH' it is the native event-study axis shifted so onset sits at '0'. * The 'fect' surface only covers ****treated**** cells, so its matrix spans 'event_time >= 0'. 'DCDH' run with placebos additionally yields the pre-period ('event_time < 0') cells.

See Also

[plot_effect_matrix()] to draw the heatmap, [nabs_effect_cells()] to fit and tidy in one step, [aggregate_effects()] to collapse cells back onto an event-study path.

Examples

```
# The data.frame escape hatch needs no estimator packages.
raw <- expand.grid(cohort = 3:5, event_time = 0:3)
raw$estimate <- with(raw, 0.1 * event_time + 0.05 * (cohort - 4))
raw$std.error <- 0.08
cells <- as_nabs_effect_cells(raw, method = "DCDH", outcome = "y")
cells
```

as_nabs_event_study *Coerce an estimator result to a tidy event-study tibble*

Description

'as_nabs_event_study()' is an S3 generic that converts the native output object of a supported estimator into the unified 'nabs_event_study_tbl' schema used by [nabs_event_plot()]. Methods exist for objects of class "did_multiplegt_dyn" (from 'DIDmultiplegtDYN'), "PanelEstimate" (from 'PanelMatch'), "fect" (from 'fect'), and "fixest" (from 'fixest', used for the naive TWFE reference series).

Usage

```
as_nabs_event_study(
  x,
  method = NULL,
  outcome = NA_character_,
  conf.level = 0.95,
  ...
)

## S3 method for class 'fixest'
as_nabs_event_study(
  x,
  method = NULL,
  outcome = NA_character_,
  conf.level = 0.95,
  ...
)
```

```
## S3 method for class 'did_multiplegt_dyn'
as_nabs_event_study(
  x,
  method = NULL,
  outcome = NA_character_,
  conf.level = 0.95,
  ...
)

## S3 method for class 'fect'
as_nabs_event_study(
  x,
  method = NULL,
  outcome = NA_character_,
  conf.level = 0.95,
  ...
)

## S3 method for class 'list'
as_nabs_event_study(
  x,
  method = NULL,
  outcome = NA_character_,
  conf.level = 0.95,
  ...
)

## S3 method for class 'nabs_event_study_result'
as_nabs_event_study(
  x,
  method = NULL,
  outcome = NA_character_,
  conf.level = 0.95,
  ...
)

## S3 method for class 'nabs_event_study_simple'
as_nabs_event_study(
  x,
  method = NULL,
  outcome = NA_character_,
  conf.level = 0.95,
  ...
)

## S3 method for class 'PanelEstimate'
as_nabs_event_study(
```

```

x,
method = NULL,
outcome = NA_character_,
conf.level = 0.95,
pre_obj = NULL,
add_reference = TRUE,
...
)

```

Arguments

<code>x</code>	A supported estimator object.
<code>method</code>	Optional override for the ‘method’ column. If ‘NULL’, the default for that estimator is used.
<code>outcome</code>	Optional outcome name to record in the ‘outcome’ column.
<code>conf.level</code>	Confidence level for ‘conf.low’ / ‘conf.high’. Default ‘0.95’. When the underlying object stores its own CI bounds (e.g. ‘fect’), those are used as-is and ‘conf.level’ is recorded as metadata only.
<code>...</code>	Method-specific arguments. See the individual method files for details (e.g. ‘pre_obj’ for the ‘PanelEstimate’ method).
<code>pre_obj</code>	A ‘placebo_test’ result from ‘PanelMatch::placebo_test()’, used to fill in the pre-treatment portion of the path.
<code>add_reference</code>	Logical; if ‘TRUE’ (default) and ‘pre_obj’ is given, adds a ‘(time = -1, estimate = 0)’ row.

Details

A ‘data.frame’ method is also provided as an escape hatch: it accepts any frame that already contains ‘time’ and ‘estimate’ columns and fills in the rest of the schema if missing.

fixest method

Extracts coefficients on ‘time_to_event’ interactions of the form ‘time_to_event::<k>’ or ‘time_to_event::<k>:<interaction>’, the coefficient names produced by ‘fixest::i()’. These are treated as event-study **levels** (the classic absorbing-treatment parametrisation). Standard errors come from the model’s clustered VCOV; confidence intervals use the normal approximation and ‘conf.level’.

Note that [naive_twfe()] does not fit this absorbing parametrisation itself – it uses a distributed-lag design in treatment levels – but this method is retained so that models you fit yourself with ‘fixest::i()’ can still be tidied.

fect method

‘fect::fect()’ returns event-study coordinates in ‘\$time’ and ‘\$att’, with confidence-interval bounds in the two-column matrix ‘\$att.bound’. Standard errors are pulled from ‘\$est.att[, "S.E."]' when available; if the object was fit without ‘se = TRUE’, only the point estimates are returned and SE / CI columns are filled with ‘NA’.

The ‘method’ label is auto-detected from ‘x\$method’, the option that was passed to ‘fect::fect()’:

- “fe” -> “FE” (two-way fixed-effects imputation; Borusyak-style)

- `"ife"` -> `"IFE"` (interactive fixed effects; Bai 2009)
- `"mc"` -> `"MC"` (matrix completion; Athey et al. 2021)

Pass an explicit `'method'` argument to override this auto-detected label.

PanelMatch method

For `'PanelMatch::PanelEstimate()'` the post-treatment leads are stored as `'$estimate'` / `'$standard.error'` (singular). The pre-treatment placebo results from `'PanelMatch::placebo_test()'` use `'$estimates'` / `'$standard.errors'` (plural). To produce a single event-study path, pass the placebo object via `'pre_obj'`:

```
pm <- PanelMatch::PanelMatch(...)
pe <- PanelMatch::PanelEstimate(pm, panel.data = pd)
pl <- PanelMatch::placebo_test(pm, panel.data = pd, plot = FALSE)
tidy <- as_nabs_event_study(pe, pre_obj = pl)
```

A `'time = -1'` reference point with `'estimate = 0'` is inserted so that the event-study path is anchored at $t = -1$, matching common practice and the `'did'` / `'fixest::iplot'` convention. Disable with `'add_reference = FALSE'`.

Value

A tibble of class `"nabs_event_study_tbl"` with one row per relative period and the columns documented in the package overview.

Examples

```
# The data.frame escape hatch needs no estimator packages: pass a frame
# that already has `time` and `estimate`; the remaining schema columns
# (including CIs derived from `std.error`) are filled in automatically.
raw <- data.frame(
  time      = -3:4,
  estimate  = c(-0.05, 0.01, 0.00, 0.02, 0.30, 0.42, 0.38, 0.50),
  std.error = 0.12
)
tidy_fit <- as_nabs_event_study(raw, method = "DCDH", outcome = "y")

# With the DCDH estimator installed, coerce its native object directly.
if (requireNamespace("DIDmultiplegtDYN", quietly = TRUE) &&
    requireNamespace("polars", quietly = TRUE)) {
  set.seed(1)
  library(polars)
  panel <- expand.grid(id = 1:60, t = 1:10)
  panel$d <- with(panel, as.integer(
    (id %% 4 == 1 & t %in% 4:7) |
    (id %% 4 == 2 & t %in% 5:8) |
    (id %% 4 == 3 & t %in% 6:9)
  ))
  panel$y <- 0.2 * panel$t + 0.5 * panel$d + rnorm(nrow(panel))

  fit <- DIDmultiplegtDYN::did_multiplegt_dyn(
```

```

    df = panel,
    outcome = "y",
    group = "id",
    time = "t",
    treatment = "d",
    effects = 3,
    placebo = 2
  )
  as_nabs_event_study(fit, outcome = "y")
}

```

nabs_effect_cells

Fit an estimator and return cohort-by-time effect cells

Description

‘nabs_effect_cells()’ is the cohort-matrix counterpart to [nabs_event_study()]: it fits one supported estimator and returns the result already tidied into the ‘nabs_effect_cell_tbl’ schema, ready for [plot_effect_matrix()]. It wires up the per-estimator machinery that a cohort breakdown needs – a unit-level onset cohort for ‘DCDH’, and ‘keep.sims = TRUE’ for ‘fect’ bootstrap cell SEs – so you do not have to.

Usage

```

nabs_effect_cells(
  data,
  outcome,
  treatment,
  unit,
  time,
  method = c("DCDH", "IFE", "FE", "MC"),
  lags = 6L,
  leads = 8L,
  controls = NULL,
  cluster = unit,
  conf.level = 0.95,
  axis = c("event", "calendar"),
  dcdh_strategy = c("loop", "by"),
  nboots = 200L,
  max_cohorts = 30L,
  ...
)

```

Arguments

data	A panel data frame, or a path to a Stata ‘.dta’ file (which is read via [nabs_read_dta()]) with default settings).
------	--

outcome, treatment, unit, time	Character column names.
method	One of "DCDH", "IFE", "FE", "MC".
lags, leads	Integer pre- and post-period lengths.
controls	Optional character vector of covariate names.
cluster	Character; cluster variable. Defaults to 'unit'.
conf.level	Confidence level for the tidied output. Default 0.95.
axis	Which axis [plot_effect_matrix()] should default to: "event" (relative time, default) or "calendar". Both columns are populated regardless.
dcdh_strategy	How to obtain cohort-specific DCDH estimates: * "loop" (default) re-estimates the event study separately for each onset cohort against the never-treated units ('only_never_switchers = TRUE'). Robust – it reuses the stable event-study tidier – and the control group (never-treated) is constant and easy to interpret. * "by" runs a single 'did_multiplegt_dyn(..., by = cohort)' call and parses its per-level sublists. One estimation, native DCDH controls, but it depends on the package's nested-output layout.
nboots	Bootstrap replicates for the 'fect' family (default 200). Bootstrap draws are retained ('keep.sims = TRUE') so cell SEs can be formed.
max_cohorts	Safety cap on the number of distinct onset cohorts before 'nabs_effect_cells()' refuses to run (default 30); raise it deliberately.
...	Extra arguments passed straight to the underlying estimator. Stata-style aliases are also accepted here and translated with an informative message: 'df' (for 'data'), 'group' (for 'unit'), 'placebo' (for 'lags'), and 'effects' (for 'leads'; note 'leads = effects - 1', because nonabsdid places treatment onset at relative time 0). See the "nonabsdid for Stata users" vignette.

Value

A list of class "nabs_effect_cells_result" with elements 'cells' (an 'nabs_effect_cell_tbl'), 'fit' (native object, or a list of them for the DCDH loop), and 'call'.

Status

Experimental, and intentionally limited to 'DCDH' and the 'fect' family ('IFE' / 'FE' / 'MC'). 'PanelMatch' is not supported here.

See Also

[plot_effect_matrix()], [as_nabs_effect_cells()].

Examples

```
if (requireNamespace("fect", quietly = TRUE)) {
  set.seed(1)
  panel <- expand.grid(id = 1:80, t = 1:12)
  onset <- c(`1` = 4, `2` = 6, `3` = 8)[as.character(panel$id %>% 4)]
  panel$d <- as.integer(!is.na(onset) & panel$t >= onset)
```

```

panel$y <- 0.2 * panel$t + 0.4 * panel$d + rnorm(nrow(panel))
res <- nabs_effect_cells(panel, outcome = "y", treatment = "d",
                        unit = "id", time = "t", method = "FE",
                        nboots = 50)

res$cells
}

```

nabs_event_plot

Plot one or more event-study tipples on a single panel

Description

Overlays event-study estimates from any combination of supported estimators on a single ggplot2 panel. Two visual encodings are available via ‘style’:

Usage

```

nabs_event_plot(
  ...,
  style = c("prepost_color", "method_shape"),
  connect = FALSE,
  connect_linewidth = 0.4,
  reference = NULL,
  reference_color = "grey20",
  palette = "default",
  shapes = NULL,
  xlim = NULL,
  ylim = NULL,
  dodge = 0.5,
  point_size = 2.5,
  errorbar_width = 0.1,
  x_break_by = 2,
  show_pre_post_legend = TRUE,
  xlab = "Relative time to treatment change",
  ylab = "Estimated effect",
  base_size = 11
)

```

Arguments

...	One or more ‘nabs_event_study_tbl’ objects. Bare arguments and a single list are both accepted.
style	Visual encoding. One of “prepost_color” (default; color differs by pre/post) or “method_shape” (color and marker shape both encode the method, shared across pre/post).
connect	Logical. If ‘TRUE’, point estimates within each series are joined by a thin line. Default ‘FALSE’. The line is split at the treatment boundary so pre- and post-treatment segments are not joined across the discontinuity.

connect_linewidth	Width of the connecting line when 'connect = TRUE'. Default '0.4'.
reference	Optional 'nabs_event_study_tbl' to draw as a neutral-color reference layer (typically a naive TWFE estimate). Drawn under the main series.
reference_color	Color for the reference series. Default "grey20".
palette	Either "default" (the package's built-in palette, patterned after the DCDH/PanelMatch/IFE conventions in the codebase this package was extracted from), "colorblind" (Okabe-Ito), or a named character vector of colors. For 'style = "prepost_color"' the names are keyed by "<method>_<window>", e.g. 'c("DCDH_pre" = "#DE2D26", "DCDH_post" = "#3182BD", ...)'. For 'style = "method_shape"' the names are keyed by "<method>", e.g. 'c("DCDH" = "#DE2D26", ...)'.
shapes	Optional named integer vector of plotting symbols keyed by "<method>", used only when 'style = "method_shape"'. Defaults to the package's built-in shape set.
xlim, ylim	Numeric length-2 vectors for axis limits. 'NULL' lets ggplot2 choose.
dodge	Width of the position-dodge applied to points, lines, and error bars. The 'reference' series shares this dodge with the main series, so all series (including the naive TWFE reference) get their own evenly-spaced horizontal slot and their CIs do not overlap. Default '0.5'.
point_size, errorbar_width	Aesthetic controls for the geom layers.
x_break_by	Spacing between x-axis ticks (default 2, giving ... -4, -2, 0, 2, 4, 6 ...). Event-study time is integer, so this avoids ggplot2's default half-integer breaks like 2.5.
show_pre_post_legend	Logical. Only relevant for 'style = "prepost_color"'. If 'TRUE', the legend keys are labeled "<method>; pre" / "<method>; post". If 'FALSE', only one key per method is shown. Default 'TRUE'.
xlab, ylab	Axis labels.
base_size	Base font size passed to 'theme_minimal()'.

Details

* "prepost_color" (default) – each method gets its own color, with separate shades for pre- and post-treatment periods, mirroring common conventions in DCDH-style plots. Points are drawn as circles throughout. * "method_shape" – each method gets a single color *and* a single marker shape. Pre and post periods share both the color and the shape; they are told apart only by their position relative to time 0. Because method is double-encoded (color + shape), this style stays legible in grayscale.

An optional 'reference' series – typically a naive TWFE fit from [naive_twfe()] – is drawn in a neutral color (default black) so the reader can see what the heterogeneity-robust estimators are correcting against.

Set 'connect = TRUE' to join each series' point estimates with a thin line, in addition to the points and error bars.

Value

A ‘ggplot’ object.

Examples

```
dcdh_tidy <- as_nabs_event_study(
  data.frame(
    time = -2:3,
    estimate = c(-0.06, -0.02, 0.10, 0.22, 0.28, 0.31),
    std.error = 0.08
  ),
  method = "DCDH",
  outcome = "y"
)

ife_tidy <- as_nabs_event_study(
  data.frame(
    time = -2:3,
    estimate = c(-0.04, 0.00, 0.08, 0.18, 0.25, 0.27),
    std.error = 0.10
  ),
  method = "IFE",
  outcome = "y"
)

nabs_event_plot(dcdh_tidy, ife_tidy, xlim = c(-2, 3))
nabs_event_plot(dcdh_tidy, ife_tidy, style = "method_shape", connect = TRUE)
```

nabs_event_study

Run an event-study estimator with a unified interface

Description

‘nabs_event_study()’ is a thin wrapper around the three supported estimators (DCDH, PanelMatch, IFE/fect) that takes a single, common argument set and dispatches to the correct underlying package. It is **not** intended to expose every option of every estimator; for that, call the underlying packages directly and tidy their output with [as_nabs_event_study()].

Usage

```
nabs_event_study(
  data,
  outcome,
  treatment,
  unit,
  time,
  method = c("DCDH", "PanelMatch", "IFE", "FE", "MC"),
  lags = 6L,
```

```

    leads = 8L,
    controls = NULL,
    cluster = unit,
    conf.level = 0.95,
    cv = NULL,
    nboots = NULL,
    r = NULL,
    k = NULL,
    nlambdas = NULL,
    vartype = NULL,
    se = NULL,
    parallel = FALSE,
    cores = NULL,
    number.iterations = NULL,
    se.method = NULL,
    run_placebo = NULL,
    num.cores = NULL,
    ...
  )

```

Arguments

<code>data</code>	A panel data frame, or a path to a Stata <code>.dta</code> file (which is read via <code>[nabs_read_dta()]</code> with default settings).
<code>outcome, treatment, unit, time</code>	Character column names.
<code>method</code>	One of <code>"DCDH"</code> , <code>"PanelMatch"</code> , <code>"IFE"</code> .
<code>lags, leads</code>	Integer pre- and post-period lengths.
<code>controls</code>	Optional character vector of covariate names.
<code>cluster</code>	Character; cluster variable. Defaults to <code>'unit'</code> .
<code>conf.level</code>	Confidence level for the tidied output. Default 0.95.
<code>cv, nboots, r, parallel, cores</code>	Tuning knobs for the <code>'fect'</code> family (<code>'IFE'</code> , <code>'FE'</code> , <code>'MC'</code>); ignored by other methods. <code>'cv'</code> toggles cross-validation (default: on for <code>'IFE'</code> / <code>'MC'</code> , off for <code>'FE'</code>); <code>'r'</code> caps / fixes the number of interactive-fixed-effect factors; <code>'nboots'</code> is the bootstrap count (default 200). <code>'parallel'</code> defaults to <code>'FALSE'</code> because, on large panels, copying the data to parallel workers tends to exhaust memory rather than help; set <code>'parallel = TRUE'</code> (optionally with <code>'cores'</code>) for big speedups on small panels. These are first-class arguments so that, e.g., <code>'cv = FALSE'</code> no longer collides with internal defaults.
<code>k, nlambdas, vartype, se</code>	Further <code>'fect'</code> -family speed knobs. <code>'k'</code> is the number of cross-validation rounds; <code>'fect'</code> 's own default is 20, which is slow on large panels, so the wrapper defaults it to 5 when CV is on. <code>'nlambdas'</code> caps the MC regularisation grid (wrapper default 5 vs <code>'fect'</code> 's 10). <code>'vartype'</code> selects the variance estimator (<code>"bootstrap"</code> , <code>"jackknife"</code> , or <code>"parametric"</code>); <code>"parametric"</code> is available for <code>'IFE'</code> and avoids refitting the factor model on every resample, but is not supported for <code>'MC'</code> . <code>'se =</code>

`FALSE` skips uncertainty entirely for a fast point-estimate-only pass. Advanced ‘fect’ knobs (`‘tol’`, `‘max.iteration’`, `‘em’`, `‘lambda’`) may also be passed through `‘...’`.

`number.iterations`, `se.method`, `run_placebo`, `num.cores`

Tuning knobs for ‘PanelMatch’; ignored by other methods. `‘number.iterations’` is the bootstrap count (default 1000); lower it (e.g. 200) for tractability. `‘se.method’` selects the SE type (`“bootstrap”`, `“conditional”`, `“unconditional”`); the analytic `“conditional”/“unconditional”` methods skip the bootstrap entirely and are by far the biggest speed-up. `‘run_placebo = FALSE’` skips the separate placebo-test bootstrap (a second full bootstrap pass). `‘parallel’/‘num.cores’` are forwarded to `‘PanelEstimate()’` to spread the bootstrap across cores.

`...`

Extra arguments passed straight to the underlying estimator. Stata-style aliases are also accepted here and translated with an informative message: `‘df’` (for `‘data’`), `‘group’` (for `‘unit’`), `‘placebo’` (for `‘lags’`), and `‘effects’` (for `‘leads’`; note `‘leads = effects - 1’`, because `nonabsdid` places treatment onset at relative time 0). See the “nonabsdid for Stata users” vignette.

Details

What it does cover:

- Variable names (outcome, treatment, unit, time),
- Pre/post window length (`‘lags’`, `‘leads’`),
- Optional covariates and clustering,
- Reasonable defaults that match the three packages’ typical use.

Value

A list of class `“nabs_event_study_result”` with elements:

‘tidy’ An `‘nabs_event_study_tbl’`.

‘fit’ The native estimator object (for diagnostics).

‘call’ The call that produced it.

Examples

```
if (requireNamespace("DIDmultiplegtDYN", quietly = TRUE) &&
    requireNamespace("polars", quietly = TRUE)) {
  set.seed(1)
  library(polars)
  panel <- expand.grid(id = 1:60, t = 1:10)
  panel$d <- with(panel, as.integer(
    (id %% 4 == 1 & t %in% 4:7) |
    (id %% 4 == 2 & t %in% 5:8) |
    (id %% 4 == 3 & t %in% 6:9)
  ))
  panel$y <- 0.2 * panel$t + 0.5 * panel$d + rnorm(nrow(panel))

  res_dcdh <- nabs_event_study(
```

```

    panel,
    outcome = "y",
    treatment = "d",
    unit = "id",
    time = "t",
    method = "DCDH",
    lags = 2,
    leads = 2
  )
  res_dcdh$tidy
}

```

nabs_event_study_simple

One-line exploratory front door for non-absorbing event studies

Description

`nabs_event_study_simple()` is a deliberately opinionated convenience wrapper for the **first 30 seconds** of an analysis. You give it your data and the four column names that identify outcome / treatment / unit / time, and it tries to give you a sensible event-study figure with as little typing as possible.

Usage

```

nabs_event_study_simple(
  data,
  outcome,
  treatment,
  unit,
  time,
  methods = c("DCDH", "FE"),
  include_twfe = TRUE,
  lags = NULL,
  leads = NULL,
  controls = NULL,
  verbose = TRUE,
  full = FALSE,
  max_units = 5000L,
  sample_seed = 1L,
  keep_fits = FALSE,
  ...
)

```

Arguments

data	A panel data frame, or a path to a Stata <code>.dta</code> file (which is read via <code>[nabs_read_dta()]</code> with default settings).
------	---

outcome, treatment, unit, time	Character column names. The treatment column should be a 0/1 indicator (it is allowed to switch back to 0, i.e. non-absorbing).
methods	Character vector of estimators to run. Any subset of 'c("DCDH", "PanelMatch", "IFE", "FE", "MC")'. Default 'c("DCDH", "FE")' – a cheap first look (DCDH plus two-way-FE imputation, no cross-validation). The heavier estimators ('PanelMatch's bootstrap and 'IFE'/'MC's cross-validation) are opt-in: add them explicitly once the cheap pass looks reasonable, or call [nabs_event_study()] to tune them.
include_twfe	Logical; if 'TRUE' (default), also fit a naive TWFE reference series via [naive_twfe()] and overlay it in a neutral color.
lags, leads	Integer pre- and post-period lengths. If 'NULL' (default), reasonable values are auto-chosen from the panel: 'leads' is set to roughly one third of the typical (median) post-treatment span across treated units (capped at 8), and 'lags' to roughly one quarter of the typical (median) pre-treatment span (capped at 6). The median is used rather than the maximum so that a single unit with an unusually long history does not inflate the window. Override either explicitly to be sure of the window.
controls	Optional character vector of covariate names; passed straight through to each estimator.
verbose	Logical; if 'TRUE' (default), print a brief progress message before each estimator runs.
full	Logical; if 'FALSE' (default) and the panel has more than 'max_units' units, a random sample of 'max_units' units is used so the first pass stays fast. Set 'full = TRUE' to use every unit.
max_units	Integer; the unit cap used when 'full = FALSE' (default 5000).
sample_seed	Integer seed for the first-pass subsample, so the quick look is reproducible. The caller's global RNG state is left untouched.
keep_fits	Logical; if 'FALSE' (default) the heavy native estimator objects are not retained in '\$fits' (they can be gigabytes for 'fect'). Set 'TRUE' if you need them for diagnostics.
...	Forwarded to [nabs_event_plot()] (e.g. 'xlim', 'ylim', 'palette', 'ylab', 'x_break_by'). Stata-style aliases are also accepted here and translated with an informative message: 'df' (for 'data'), 'group' (for 'unit'), 'placebo' (for 'lags'), and 'effects' (for 'leads'; note 'leads = effects - 1'). See the "nonabsdid for Stata users" vignette.

Details

By default it runs **all three** heterogeneity-robust estimators (DCDH, PanelMatch, IFE) plus a naive TWFE reference, and returns a single overlay plot along with the tidy tibbles and raw fits. Use it to *see the picture quickly*; for a careful, publication-ready result, switch to [nabs_event_study()] and tune options per estimator.

If a particular estimator's package is not installed, that estimator is silently skipped with a message and the rest are still attempted. This is intentional: the goal of '_simple()' is to give you *something* to look at even if your environment isn't fully provisioned.

Errors from a single estimator (for instance, PanelMatch failing because there are too few clean controls in the lag window) are caught, reported as a warning, and the remaining estimators continue.

Value

A list of class `"nabs_event_study_simple"` with elements:

- 'plot'** A 'ggplot' object; the overlay figure.
- 'tidy'** A single combined 'nabs_event_study_tbl' with all methods.
- 'per_method'** Named list of per-method tidy tibbles.
- 'fits'** Named list of native estimator objects.
- 'twfe'** The TWFE reference (or 'NULL').
- 'call'** The matched call.

Examples

```
if (requireNamespace("DIDmultiplegtDYN", quietly = TRUE) &&
    requireNamespace("polars", quietly = TRUE)) {
  set.seed(1)
  library(polars)
  panel <- expand.grid(id = 1:60, t = 1:10)
  panel$d <- with(panel, as.integer(
    (id %% 4 == 1 & t %in% 4:7) |
    (id %% 4 == 2 & t %in% 5:8) |
    (id %% 4 == 3 & t %in% 6:9)
  ))
  panel$y <- 0.2 * panel$t + 0.5 * panel$d + rnorm(nrow(panel))

  res <- nabs_event_study_simple(
    panel,
    outcome = "y",
    treatment = "d",
    unit = "id",
    time = "t",
    methods = "DCDH",
    include_twfe = FALSE,
    lags = 2,
    leads = 2,
    verbose = FALSE
  )
  res$tidy
}
```

nabs_read_dta	<i>Read a Stata .dta file into an analysis-ready data frame</i>
---------------	---

Description

`'nabs_read_dta()'` is a thin convenience layer over `[haven::read_dta()]` that smooths out the two places where freshly imported Stata data tends to trip up R estimation packages:

Usage

```
nabs_read_dta(
  path,
  labelled = c("factor", "numeric", "keep"),
  missings = c("na", "keep"),
  encoding = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

<code>path</code>	Path to a <code>'dta'</code> file.
<code>labelled</code>	How to handle <code>'haven_labelled'</code> columns. One of: <code>"factor"</code> (default) Convert labelled columns to factors via <code>[haven::as_factor()]</code> . Unlabelled values keep their code as the level name. <code>"numeric"</code> Strip value labels via <code>[haven::zap_labels()]</code> , keeping the underlying numeric codes. Use this when a labelled column is really a numeric variable (e.g. a 0/1 treatment dummy that happens to carry labels). <code>"keep"</code> Leave <code>'haven_labelled'</code> columns untouched. Note that the estimator packages may not accept them.
<code>missings</code>	How to handle Stata extended missing values (<code>'a'-'z'</code>). <code>"na"</code> (default) collapses them to regular <code>'NA'</code> via <code>[haven::zap_missing()]</code> ; <code>"keep"</code> preserves the tags.
<code>encoding</code>	Passed to <code>[haven::read_dta()]</code> . Only needed for files written by Stata 13 or older with a non-default encoding.
<code>verbose</code>	Logical; if <code>'TRUE'</code> (default), print a one-line summary of what was read and converted.
<code>...</code>	Additional arguments passed to <code>[haven::read_dta()]</code> (e.g. <code>'col_select'</code> , <code>'n_max'</code>).

Details

- ****Labelled columns.**** Stata value labels arrive in R as `'haven_labelled'` vectors, which many modeling functions (including the estimator packages wrapped by `nonabsdid`) do not understand. By default these are converted to factors; set `'labelled = "numeric"'` to drop the labels and keep the underlying codes instead.

- **Extended missing values.** Stata's `‘.a’–‘.z’` arrive as **tagged** `‘NA’`s, which compare and print like ordinary `‘NA’` but can survive into model matrices in surprising ways. By default all tagged `‘NA’`s are collapsed to regular `‘NA’`.

Variable labels (Stata's `‘label variable’`) are preserved as `“label”` attributes on each column; they are harmless to the estimators and often useful for plot labels.

You rarely need to call this function yourself: `[nabs_event_study()]` and `[nabs_event_study_simple()]` accept a path to a `‘.dta’` file as their `‘data’` argument and route it through `‘nabs_read_dta()’` automatically.

Value

A tibble.

See Also

`[nabs_write_dta()]` for the reverse direction, and the "nonabsdid for Stata users" vignette (`‘vignette("nonabsdid-for-stata-users")’`) for a full Stata-to-R walk-through.

Examples

```
if (requireNamespace("haven", quietly = TRUE)) {
  # Round-trip a small labelled panel through a temporary .dta file.
  tmp <- tempfile(fileext = ".dta")
  panel <- data.frame(id = rep(1:3, each = 2), t = rep(1:2, 3),
                     d = c(0, 1, 0, 0, 1, 1),
                     y = rnorm(6))
  haven::write_dta(panel, tmp)

  mydata <- nabs_read_dta(tmp)
  head(mydata)
}
```

nabs_write_dta	<i>Write event-study results to a Stata .dta file</i>
----------------	---

Description

`‘nabs_write_dta()’` exports an `‘nabs_event_study_tbl’` – or anything that `[as_nabs_event_study()]` can coerce into one, including the result objects returned by `[nabs_event_study()]` and `[nabs_event_study_simple()]` – to a Stata `‘.dta’` file via `[haven::write_dta()]`.

Usage

```
nabs_write_dta(x, path, version = 14, label = NULL, verbose = TRUE)
```

Arguments

<code>x</code>	An <code>'nabs_event_study_tbl'</code> , an <code>'nabs_event_study_result'</code> , an <code>'nabs_event_study_simple'</code> , a supported estimator object, or a plain data frame with at least <code>'time'</code> and <code>'estimate'</code> columns. Anything that is not already a data frame is routed through <code>[as_nabs_event_study()]</code> .
<code>path</code>	Path of the <code>'dta'</code> file to write.
<code>version</code>	Stata file format version, passed to <code>[haven::write_dta()]</code> . Default <code>'14'</code> (readable by Stata 14 and later).
<code>label</code>	Optional dataset label (Stata's <code>'label data'</code>), passed to <code>[haven::write_dta()]</code> .
<code>verbose</code>	Logical; if <code>'TRUE'</code> (default), print a one-line summary including any column renames.

Details

The tidy schema uses dots in some column names (`'std.error'`, `'conf.low'`, `'conf.high'`), which are not valid Stata variable names. These are renamed to underscore versions (`'std_error'`, `'conf_low'`, `'conf_high'`) on the way out; any other invalid characters are likewise replaced with `'_'`.

This makes the "estimate in R, post-process in Stata" workflow a one-liner: a Stata-using coauthor can rebuild the event-study figure with `'twoway rcap'/'scatter'`, or feed the estimates into their own tables.

Value

The path, invisibly.

See Also

`[nabs_read_dta()]` for the reverse direction.

Examples

```
if (requireNamespace("haven", quietly = TRUE)) {
  tidy <- as_nabs_event_study(
    data.frame(time = -2:3,
               estimate = c(0.02, -0.01, 0, 0.4, 0.5, 0.45),
               std.error = 0.1),
    method = "DCDH", outcome = "y"
  )
  tmp <- tempfile(fileext = ".dta")
  nabs_write_dta(tidy, tmp)

  haven::read_dta(tmp)
}
```

naive_twfe

*Estimate a naive two-way fixed-effects (TWFE) event study***Description**

Runs a basic event-study TWFE regression of ‘outcome’ on leads and lags of the treatment, with unit and time fixed effects, using ‘fixest::feols()’. The result is ****deliberately unsophisticated**** – the point of ‘nonabsdid’ is to contrast this naive benchmark against heterogeneity-robust estimators (DCDH, ‘fect’, PanelMatch).

Usage

```
naive_twfe(
  data,
  outcome,
  treatment,
  unit,
  time,
  lags = 12L,
  leads = 6L,
  controls = NULL,
  cluster = unit,
  conf.level = 0.95
)
```

Arguments

data	A data frame (panel) in long format.
outcome, treatment, unit, time	Character scalars naming the outcome, the 0/1 (or ‘FALSE’/‘TRUE’) treatment indicator, the unit id, and the time variable.
lags	Non-negative integer: number of pre-treatment periods (event times $-1, \dots, -\text{lags}$) to report. Event time ‘-1’ is the omitted reference.
leads	Non-negative integer: number of post-treatment periods (event times $0, \dots, \text{leads}$) to report.
controls	Optional character vector of additional control columns.
cluster	Character vector of column names to cluster standard errors on. Defaults to ‘unit’.
conf.level	Confidence level for the returned tibble. Default 0.95.

Details

Unlike a classic event study, ‘naive_twfe()’ does ****not**** assume the treatment is absorbing. It is built for binary treatments that can switch on ***and off*** over time (e.g. a policy that is repealed, a

subsidy that lapses). It fits a distributed-lag TWFE in the treatment *levels*,

$$y_{it} = \alpha_i + \gamma_t + \sum_k \beta_k D_{i,t+k} + \varepsilon_{it},$$

i.e. the outcome on the leads and lags of the treatment indicator with unit and time fixed effects. The coefficient on lag 'k' is reported at event time '+k' and the coefficient on lead 'k' at event time '-k', so the path is defined relative to a treatment *change* rather than to a single absorbing onset. Event time '-1' is the omitted reference. Each β_k is a partial correlation, not a heterogeneity-robust dynamic effect – that is the point of the benchmark.

The naming of 'lags'/'leads' follows the package convention used elsewhere (and in the README): 'lags' counts pre-periods, 'leads' counts post-periods, so 'lags = 6, leads = 8' yields event times on '[-6, 8]'.

Coefficients and standard errors are read directly from the fitted model (clustered as requested); the reference period '-1' is reported as exactly zero.

Missing treatment values are read as untreated ('0') when the leads and lags are constructed. For this naive benchmark that is usually innocuous, but if treatment missingness is itself informative it can bias the reference path; the heterogeneity-robust estimators handle missingness on their own terms.

Value

An 'nabs_event_study_tbl' with 'method = "TWFE"'. The fitted 'fixest' model is attached as the "fit" attribute.

Examples

```
df <- data.frame(
  id = rep(1:4, each = 8),
  yr = rep(1:8, times = 4),
  d = c(rep(0, 8),
        0, 0, 1, 1, 1, 0, 0, 0,
        0, 0, 0, 1, 1, 1, 1, 0,
        rep(0, 8)),
  y = rnorm(32)
)
naive_twfe(df, outcome = "y", treatment = "d",
           unit = "id", time = "yr", lags = 2, leads = 3)
```

plot_effect_matrix	<i>Plot a cohort-by-time effect matrix as a heatmap</i>
--------------------	---

Description

Draws one or more 'nabs_effect_cell_tbl' objects as cohort (rows) by relative or calendar time (columns) heatmaps, with fill encoding the point estimate on a diverging scale centred at zero.

Usage

```
plot_effect_matrix(
  ...,
  axis = c("event", "calendar"),
  facet = TRUE,
  show_estimates = FALSE,
  show_se = FALSE,
  digits = 2,
  text_size = 2.6,
  title = NULL,
  caption = NULL,
  low = "#3182BD",
  mid = "#F7F7F7",
  high = "#DE2D26",
  limits = NULL,
  na_color = "grey92",
  xlab = NULL,
  ylab = "Onset cohort",
  legend_title = "Effect",
  base_size = 11
)
```

Arguments

...	One or more 'nabs_effect_cell_tbl' objects (bare args or a single list), typically from [nabs_effect_cells()] or [as_nabs_effect_cells()].
axis	"event" (default) puts 'event_time' on the x axis; "calendar" uses 'calendar_time'.
facet	Logical; facet by 'method' when more than one is present. Default 'TRUE'.
show_estimates	Logical; print the rounded estimate in each tile. Default 'FALSE'.
show_se	Logical; print the standard error in parentheses beneath the estimate (implies showing the estimate). Cells with 'NA' SE show the estimate alone. Default 'FALSE'.
digits	Rounding for the in-tile estimate / SE labels. Default '2'.
text_size	Font size for the in-tile labels. Default '2.6'.
title	Plot title. 'NULL' (default) auto-titles a single-method plot with its display label (the fect family shows as "Fect FE" / "Fect IFE" / "Fect MC"; "DCDH" is unchanged); faceted plots are left untitled (the strips name the methods). Pass a string to override, or 'NA' to suppress.
caption	A short gloss of the axes printed under the plot. 'NULL' (default) auto-writes a one-line note (rows = onset cohort, columns = time since onset / calendar time), and appends a note that fect has no pre-onset cells whenever a fect-family panel is shown. Pass a string to override, or 'NA' to suppress.
low, mid, high	Diverging fill colours for negative / zero / positive estimates.
limits	Optional length-2 numeric fill limits; 'NULL' (default) makes the scale symmetric around zero from the data range.

`na_color` Fill for empty '(cohort, time)' cells. Default `"grey92"`.
`xlab, ylab, legend_title` Axis and legend labels.
`base_size` Base font size for `'theme_minimal()'`.

Details

The intended use is **one method per plot**: a single-method call gets the method as its title automatically. Passing several methods facets them with a shared scale, but that side-by-side view gets crowded quickly, so for careful comparison prefer separate per-method heatmaps.

Value

A 'ggplot' object.

Examples

```

raw <- expand.grid(cohort = 3:6, event_time = -2:4)
raw$estimate <- with(raw, ifelse(event_time < 0, 0,
                                0.15 * event_time + 0.05 * (cohort - 4)))

raw$std.error <- 0.07
cells <- as_nabs_effect_cells(raw, method = "FE", outcome = "y")
plot_effect_matrix(cells) # auto title "FE"
plot_effect_matrix(cells, show_estimates = TRUE, show_se = TRUE)
  
```

Index

`aggregate_effects`, [3](#)
`as_nabs_effect_cells`, [3](#)
`as_nabs_event_study`, [6](#)

`nabs_effect_cells`, [10](#)
`nabs_event_plot`, [12](#)
`nabs_event_study`, [14](#)
`nabs_event_study_simple`, [17](#)
`nabs_read_dta`, [20](#)
`nabs_write_dta`, [21](#)
`naive_twfe`, [23](#)

`plot_effect_matrix`, [24](#)