Package: nonabsdid (via r-universe)

June 5, 2026

Type Package

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

Version 0.3.0

Description Runs several heterogeneity-robust difference-in-differences (DID) event-study estimators for non-absorbing (i.e., treatment can switch on and off) binary treatments through their own 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 'DIDmultiplegtDYN', 'PanelMatch', and 'fect', with an optional naive two-way fixed-effects reference series via 'fixest'. 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 tibble with a 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, PanelMatch, fect, fixest, knitr, rmarkdown, testthat (>= 3.0.0), vdiffr, withr

Config/testthat/edition 3

URL https://github.com/takuma1102/nonabsdid

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

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Repository https://takuma1102.r-universe.dev

Date/Publication 2026-06-05 21:48:30 UTC

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

RemoteRef HEAD

RemoteSha de82eee079057769a28c094fc4038eae342fcb15

Contents

`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

x	A supported estimator object.
method	Optional override for the ‘method‘ column. If ‘NULL‘, the default for that estimator is used.
outcome	Optional outcome name to record in the ‘outcome‘ column.
conf.level	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.
...	Method-specific arguments. See the individual method files for details (e.g. ‘pre_obj‘ for the ‘PanelEstimate‘ method).
pre_obj	A ‘placebo_test‘ result from ‘PanelMatch::placebo_test()‘, used to fill in the pre-treatment portion of the path.
add_reference	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()] no longer fits this absorbing parametrisation itself – it uses a distributed-lag design and performs the cumulation internally – 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")
tidy_fit

# With the DCDH estimator installed, coerce its native object directly.
## Not run:
  set.seed(1)
  panel <- expand.grid(id = 1:40, t = 1:10)
  panel$d <- rbinom(nrow(panel), 1, 0.3)
  panel$y <- 0.4 * 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")

## End(Not run)
```

nabs_event_plot *Plot one or more event-study tibbles 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

```
## Not run:
  # Default: color encodes pre/post
  nabs_event_plot(dcdh_tidy, panelmatch_tidy, ife_tidy,
                  reference = naive_twfe_tidy,
                  xlim = c(-6, 6), ylim = c(-2, 2),
                  ylab = "Effect on logged dollars")

  # Color + shape both encode the method (shared across pre/post); join points
  nabs_event_plot(dcdh_tidy, panelmatch_tidy, ife_tidy,
                  style = "method_shape", connect = TRUE,
                  reference = naive_twfe_tidy)

## End(Not run)
```

`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,
  ...
)
```

Arguments

`data`	A panel data frame.
`outcome`, `treatment`, `unit`, `time`	Character column names.

`method`	One of ‘"DCDH"‘, ‘"PanelMatch"‘, ‘"IFE"‘.
`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.
`...`	Extra arguments passed straight to the underlying estimator.

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

```
## Not run:
  set.seed(1)
  panel <- expand.grid(id = 1:40, t = 1:10)
  panel$d <- rbinom(nrow(panel), 1, 0.3)
  panel$y <- 0.4 * panel$d + rnorm(nrow(panel))
  res_dcdh <- nabs_event_study(panel, outcome = "y", treatment = "d",
                               unit = "id", time = "t",
                               method = "DCDH",
                               lags = 2, leads = 3)
  res_dcdh$tidy

## End(Not run)
```

`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", "PanelMatch", "IFE"),
  include_twfe = TRUE,
  lags = NULL,
  leads = NULL,
  controls = NULL,
  verbose = TRUE,
  ...
)
```

Arguments

`data`	A panel data frame.
`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", "PanelMatch", "IFE")‘ – the three classic heterogeneity-robust estimators.
`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 longest post-treatment span (capped at 8), and ‘lags‘ to roughly one quarter of the longest pre-treatment span (capped at 6). 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.
...	Forwarded to [nabs_event_plot()] (e.g. ‘xlim‘, ‘ylim‘, ‘palette‘, ‘ylab‘, ‘x_break_by‘).

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

```
## Not run:
  set.seed(1)
  panel <- expand.grid(id = 1:40, t = 1:10)
  panel$d <- rbinom(nrow(panel), 1, 0.3)
  panel$y <- 0.4 * panel$d + rnorm(nrow(panel))

  # Restrict to a single estimator for a fast, self-contained example.
  res <- nabs_event_study_simple(
    panel,
    outcome   = "y",
    treatment = "d",
    unit      = "id",
    time      = "t",
    methods   = "DCDH",
    lags = 2, leads = 3
  )
  res$plot
  res$tidy

## End(Not run)
```

`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 $-1, \ldots, -\text{lags}$) to report. Event time ‘-1‘ is the omitted reference.
`leads`	Non-negative integer: number of post-treatment periods (event times $0, \ldots, \text{leads}$) 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). Internally it uses the distributed-lag formulation of Schmidheiny and Siegloch

(2023): the design is built from treatment *changes* $\Delta D_{it} = D_{it} - D_{i,t-1}$, with the most distant lead and lag "binned" using the treatment *level*, and the reported event-study path is the cumulative sum of the distributed-lag coefficients. This recovers the usual event-study plot when treatment happens to be absorbing, but stays correct when it is not.

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]‘.

Standard errors for the cumulative event-study coefficients are obtained from the clustered variance-covariance matrix of the distributed-lag coefficients by the delta method (each event-study coefficient is a fixed linear combination of the distributed-lag coefficients).

Value

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

References

Schmidheiny, K., & Siegloch, S. (2023). On event studies and distributed-lags in two-way fixed effects models: Identification, equivalence, and generalization. *Journal of Applied Econometrics*, 38(5), 695-713.

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)
```

Index