

Package: sdid (via r-universe)

October 19, 2026

Title Updated Synthetic Difference-in-Differences Estimation and Inference

Version 0.1.0

Description Synthetic difference-in-differences, synthetic control and difference-in-differences estimates of the average treatment effect on the treated in panel data, following Arkhangelsky, Athey, Hirshberg, Imbens and Wager (2021) <[doi:10.1257/aer.20190159](https://doi.org/10.1257/aer.20190159)> and the Stata implementation of Clarke, Pailanir, Athey and Imbens (2024) <[doi:10.1177/1536867X241297914](https://doi.org/10.1177/1536867X241297914)>. Supports simultaneous and staggered treatment adoption, bootstrap, jackknife and placebo inference, covariate adjustment, and parallel replications in compiled code. Derived from the synthdid package by the authors of the 2021 paper.

Depends R (>= 4.0.0)

Imports Rcpp, stats, utils, ggplot2 (>= 3.4.0)

LinkingTo Rcpp, RcppArmadillo

License BSD_3_clause + file LICENSE

URL <https://github.com/alcocer-jj/sdid>

BugReports <https://github.com/alcocer-jj/sdid/issues>

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Suggests generics, knitr, rmarkdown, ggrepel, mvtnorm, tibble, data.table, testthat, withr, scales, CVXR

Config/roxygen2/version 8.0.0

VignetteBuilder knitr

RoxygenNote 7.3.3

Repository <https://alcocer-jj.r-universe.dev>

Date/Publication 2026-10-19 09:33:42 UTC

RemoteUrl <https://github.com/alcocer-jj/sdid>
RemoteRef HEAD
RemoteSha bd71467bd94d3ae2faa8a99789324a514723a83d

Contents

california_prop99	2
CPS	3
PENN	3
sdid	4
sdid-methods	7
sdid-tidiers	8
sdid_placebo	9
sdid_placebo_plot	10
sdid_plot	11
sdid_units_plot	13
weights.sdids	14
Index	15

california_prop99	<i>California proposition 99</i>
-------------------	----------------------------------

Description

A dataset containing per-capita cigarette consumption (in packs). In year 1989 California imposed a Tobacco tax. The column treated is 1 from then on for California.

Usage

```
data(california_prop99)
```

Format

A data frame with 1209 rows and 4 variables:
State US state name, character string
Year Year, integer
PacksPerCapita per-capita cigarette consumption, numeric
treated the treatment indicator 0: control, 1: treated, numeric

Source

Abadie, Alberto, Alexis Diamond, and Jens Hainmueller. "Synthetic control methods for comparative case studies: Estimating the effect of California's tobacco control program." Journal of the American statistical Association 105, no. 490 (2010): 493-505.

Examples

```
data(california_prop99)
sdid(california_prop99, "PacksPerCapita", "State", "Year", "treated",
     vce = "placebo", seed = 1213, graph = FALSE)
```

CPS	<i>CPS</i>
-----	------------

Description

CPS

Usage

```
data(CPS)
```

Format

A data frame with 2000 rows and 8 variables.

- state** state
- year** year
- log_wage** log_wage
- hours** hours
- urate** urate
- min_wage** min_wage
- open_carry** open_carry
- abort_ban** abort_ban

PENN	<i>PENN</i>
------	-------------

Description

PENN

Usage

```
data(PENN)
```

Format

A data frame with 5328 rows and 5 variables.

```
country country
year year
log_gdp log_gdp
dem dem
educ educ
```

sdid

Synthetic difference-in-differences estimation and inference

Description

Estimates average effects on the treated (ATT) in panel data using synthetic difference-in-differences. As in the Stata package `sdid` (Clarke et al. 2024), it accepts a balanced panel in long format, detects whether a staggered treatment approach is needed, and estimates the effect with bootstrap, jackknife, or placebo inference. Under staggered adoption, `sdid()` estimates adoption-period effects, and the overall effect is a weighted average of these with weights proportional to each cohort's number of treated units times its number of post-treatment periods, following Algorithm 1 of Clarke et al. (2024).

Usage

```
sdid(
  data,
  outcome,
  unit,
  time,
  treatment,
  vce,
  covariates = NULL,
  method = c("sdid", "sc", "did"),
  reps = 50,
  seed = NULL,
  level = 95,
  alternative = c("two.sided", "greater", "less"),
  covariate.method = c("optimized", "projected"),
  standardize = TRUE,
  projection.sample = c("controls", "untreated"),
  num.threads = NULL,
  graph = TRUE,
  quietly = FALSE,
  ...
)
```

Arguments

<code>data</code>	A balanced panel in long format, with one row per unit and period. A data frame, tibble or data.table.
<code>outcome, unit, time, treatment</code>	Names of the columns holding the outcome, the unit identifier, the period and the treatment indicator. The indicator is 1 for a treated unit from its first treated period onward and 0 otherwise. Units first treated in the same period form a cohort, and if not every treated unit is first treated in the same period, the design is staggered.
<code>vce</code>	The method of inference, which must be given. "bootstrap" resamples units, and "placebo" gives units never treated the treatment pattern of the treated units, both over reps replications. "jackknife" leaves out one unit at a time with the weights held fixed. "noinference" skips inference. The bootstrap and the jackknife need more than one treated unit, and for a staggered design the jackknife needs more than one in every cohort. The placebo needs more units never treated than treated.
<code>covariates</code>	Optional names of numeric columns of time-varying covariates to adjust for. Defaults to NULL, adjusting for none.
<code>method</code>	The estimator. "sdid" (the default) for synthetic difference-in-differences, "sc" for synthetic control or "did" for difference-in-differences.
<code>reps</code>	The number of bootstrap or placebo replications. Defaults to 50. More give a more precise standard error.
<code>seed</code>	An optional seed for the replications, so that results can be reproduced. It applies within this call only, and the state of the random number generator is restored afterward. Defaults to NULL.
<code>level</code>	The confidence level of the intervals, in percent. Defaults to 95.
<code>alternative</code>	The alternative hypothesis of the test of the effect against zero. "two.sided" (the default) tests whether the effect differs from zero, "greater" whether it is positive and "less" whether it is negative. A one-sided test gives one-sided p-values and intervals open in its direction, so that <code>level = 95</code> corresponds to a one-sided test at 5%. Choose the direction before seeing the estimate. The estimate and its standard error are the same under every alternative.
<code>covariate.method</code>	How the covariates are adjusted for, ignored without covariates. "optimized" (the default) estimates their coefficients jointly with the weights. "projected" first regresses the outcome on the covariates and fixed effects, then estimates on the adjusted outcome, as Kranz (2022) proposes.
<code>standardize</code>	For the optimized method, TRUE (the default) to scale each covariate to mean zero and standard deviation one while solving, which keeps covariates on very different scales from stalling the solver. Coefficients are reported in the covariates' own units either way.
<code>projection.sample</code>	For the projected method, the observations its regression uses. "controls" (the default) uses the units never treated in every period. "untreated" adds the treated units' periods before treatment, a larger sample that assumes the treatment had no effect before it began.

<code>num.threads</code>	The number of threads for the replications. Defaults to <code>NULL</code> , which uses the option <code>sdid.num.threads</code> if it is set and every available thread otherwise. Results do not depend on it.
<code>graph</code>	<code>TRUE</code> (the default) to draw the trends plot of the estimate, as <code>sdid_plot(fit)</code> would.
<code>quietly</code>	<code>TRUE</code> to print nothing, for example in a loop. By default, <code>sdid()</code> prints a header, a progress bar and the results table, even when the result is assigned. Doesn't affect <code>graph</code> .
<code>...</code>	Further estimator settings: <code>zeta.omega</code> and <code>zeta.lambda</code> (regularization of the unit and time weights), <code>max.iter</code> (solver iterations), and <code>sparsify</code> (sets small unit weights to zero).

Details

The result can be passed to the plot functions `sdid_plot`, `sdid_units_plot` and `sdid_placebo_plot`, and to `sdid_placebo` for an in-time placebo test. For a staggered design, the plots show one panel per cohort.

Value

An object of class "sdid", returned invisibly. Use the methods in [sdid-methods](#) and `weights.sdid()` to extract results rather than reading the list directly. The main components are:

- `estimate` The ATT. In a staggered design, the weighted average of the cohort effects.
- `se, z, p.value, conf.int` Standard error, z statistic, p-value and confidence interval. All are `NA` with `vce = "noinference"`, and the jackknife SE is `NA` when a cohort's weight rests on a single control. One-sided intervals have an infinite bound.
- `cohorts` A data frame with one row per adoption period: `period`, `treated` (number of treated units), `post` (number of post-treatment periods), `weight` (share of the overall effect), and the cohort's `estimate`, `se`, `z`, `p.value`, `conf.low` and `conf.high`. A block design has a single row.
- `beta` Covariate coefficients, or `NULL` without covariates. With the optimized method and staggered adoption, a matrix with one column per cohort.
- `level, alternative, vce, reps, method, design, covariate.method, variables, call` The settings used for the fit. `design` is "block" or "staggered".
- `setup, estimates` The panel as matrices and the per-cohort estimates used by the plot functions. These are internal and may change.

References

- Damian Clarke, Daniel Paila  ir, Susan Athey, and Guido Imbens. "On Synthetic Difference-in-Differences and Related Estimation Methods in Stata". *The Stata Journal* 24(4), 557-598, 2024.
- Dmitry Arkhangelsky, Susan Athey, David A. Hirshberg, Guido W. Imbens, and Stefan Wager. "Synthetic Difference in Differences". *American Economic Review* 111(12), 4088-4118, 2021.

Examples

```
data(california_prop99)
fit = sdid(california_prop99, "PacksPerCapita", "State", "Year",
           "treated", vce = "placebo", seed = 1213, graph = FALSE)
confint(fit, level = 0.9)
```

sdid-methods

*Methods for sdid results***Description**

Standard generics applied to the result of `sdid()`. `weights.sdid()` extracts the unit and time weights.

Usage

```
## S3 method for class 'sdid'
print(x, digits = 5, ...)

## S3 method for class 'sdid'
summary(object, ...)

## S3 method for class 'sdid'
coef(object, ...)

## S3 method for class 'sdid'
vcov(object, ...)

## S3 method for class 'sdid'
confint(object, parm, level = object$level/100, ...)

## S3 method for class 'sdid'
nobs(object, ...)

## S3 method for class 'sdid'
plot(x, ...)
```

Arguments

<code>x, object</code>	The result of <code>sdid()</code> .
<code>digits</code>	The number of decimals printed for estimates, standard errors and confidence bounds. Defaults to 5.
<code>...</code>	Passed to <code>sdid_plot()</code> by <code>plot()</code> , and otherwise not used.
<code>parm</code>	Not used, as the result has a single parameter, the effect, named by the treatment column.
<code>level</code>	The confidence level of the interval, as a proportion, such as 0.9. Defaults to the level of the fit.

Value

`print()` Prints the header and results tables that `sdid()` prints, and returns `x` invisibly.

`summary()` Prints the same tables, then for each cohort its number of controls and pre-treatment periods with positive weight and its five largest unit weights, and returns object invisibly.

`coef()` The estimated effect, a number named by the treatment column.

`vcov()` Its variance, the square of the standard error, as a 1 by 1 matrix whose row and column are named by the treatment column. It is NA without inference.

`confint()` A 1 by 2 matrix holding the lower and upper bounds of the normal confidence interval at `level`, with the row named by the treatment column and the columns by the bounds' percentiles. For a one-sided alternative the interval is one-sided, open in the direction of the alternative.

`nobs()` The number of observations, the number of units times the number of periods.

`plot()` The trends plot of `sdid_plot()`, a ggplot object.

sdid-tidiers

Tidy and glance methods for sdid results

Description

Summaries of the result of `sdid()` in the form the broom package defines, which `modelsummary` and other packages for tables of estimates use. `tidy()` gives a row per estimate and `glance()` one row describing the data. The methods become available once the generics package is loaded, as it is by `modelsummary` and `broom`, so `modelsummary(fit)` works without further setup. Arguments given to `modelsummary` that it does not use itself, such as `cohorts` and `labels`, are passed on to `tidy()`.

Usage

```
## S3 method for class 'sdid'
tidy(
  x,
  conf.int = TRUE,
  conf.level = x$level/100,
  cohorts = FALSE,
  labels = NULL,
  ...
)

## S3 method for class 'sdid'
glance(x, ...)
```

Arguments

<code>x</code>	The result of <code>sdid()</code> .
<code>conf.int</code>	TRUE (the default) to include the bounds of the confidence interval.
<code>conf.level</code>	The confidence level of the interval, as a proportion. Defaults to the level of the fit. <code>modelsummary</code> sets it from its <code>conf_level</code> argument.
<code>cohorts</code>	TRUE to add a row for each adoption period of a staggered design, after the overall effect. Defaults to FALSE. A design with a single adoption period gets no further rows.
<code>labels</code>	Optional names for all the rows, in order, the overall effect first and then, with <code>cohorts = TRUE</code> , each adoption period. By default the overall effect is named by the treatment column and each cohort by its adoption period.
<code>...</code>	Not used.

Value

`tidy()` returns a data frame with one row per estimate and columns `term`, `estimate`, `std.error`, `statistic(z)`, `p.value`, and, if `conf.int = TRUE`, `conf.low` and `conf.high`. These are NA without inference.

`glance()` returns one row with `nobs`, `n.units`, `n.periods` and `covariates` (comma-separated names, or NA).

Examples

```
data(california_prop99)
fit = sdid(california_prop99, "PacksPerCapita", "State", "Year",
           "treated", vce = "placebo", seed = 1213, graph = FALSE,
           quietly = TRUE)
if (requireNamespace("generics", quietly = TRUE)) {
  generics::tidy(fit)
  generics::glance(fit)
}
```

<code>sdid_placebo</code>	<i>Computes a placebo variant of our estimator using pre-treatment data only</i>
---------------------------	--

Description

Computes a placebo variant of our estimator using pre-treatment data only

Usage

```
sdid_placebo(estimate, treated.fraction = NULL, placebo.time = "auto")
```

Arguments

<code>estimate</code>	The result of <code>sdid()</code> with a single adoption period.
<code>treated.fraction</code>	The share of the periods before treatment that the placebo treats, counted back from the last of them. Defaults to <code>NULL</code> , which uses the share of all periods that are treated. Give either this or <code>placebo.time</code> , not both.
<code>placebo.time</code>	The period at which the placebo treatment begins, as it appears in the time column, such as 1981, or a date for a panel indexed by dates. It must come before the real treatment, with at least one period before it. Defaults to "auto", which sets it from <code>treated.fraction</code> .

Value

The placebo estimate. Use `c()` to get it as a number, print it for a one-line summary, or plot it with `sdid_placebo_plot()`.

<code>sdid_placebo_plot</code>	<i>Plot an estimate with its in-time placebo</i>
--------------------------------	--

Description

Plots the treated and synthetic control trajectories of an `sdid()` result, with an in-time placebo stacked below. Requires `ggplot2`.

Usage

```
sdid_placebo_plot(
  estimate,
  treated.fraction = NULL,
  style = c("minimal", "effects"),
  placebo.time = "auto",
  x.breaks = NULL,
  y.breaks = NULL
)
```

Arguments

<code>estimate</code>	The result of <code>sdid()</code> with a single adoption period.
<code>treated.fraction</code>	The share of the periods before treatment that the placebo treats, as in <code>sdid_placebo()</code> . Defaults to <code>NULL</code> .
<code>style</code>	The plot's style, "minimal" (the default) or "effects", as in <code>sdid_plot()</code> .
<code>placebo.time</code>	The period at which the placebo treatment begins, as in <code>sdid_placebo()</code> . Defaults to "auto".
<code>x.breaks, y.breaks</code>	Optional ticks for the time and outcome axes, as in <code>sdid_plot()</code> . The outcome ticks apply to both panels.

Value

A ggplot object, which can be extended with +.

sdid_plot	<i>Plot treated and synthetic control trajectories</i>
-----------	--

Description

Plots the treated and synthetic control trajectories of an `sdid()` result, along with the time weights, which show which pre-treatment periods were most informative for predicting post-treatment outcomes among the controls.

Usage

```
sdid_plot(
  estimates,
  treated.name = "treated",
  control.name = "synthetic control",
  spaghetti.units = c(),
  spaghetti.matrices = NULL,
  lambda.comparable = FALSE,
  overlay = 0,
  lambda.plot.scale = 3,
  effect.curvature = 0.3,
  line.width = 0.5,
  point.size = 1,
  onset.alpha = 0.8,
  spaghetti.line.width = 0.2,
  spaghetti.label.size = 2,
  spaghetti.line.alpha = 0.3,
  spaghetti.label.alpha = 0.5,
  alpha.multiplier = NULL,
  style = c("minimal", "effects"),
  lambda.axis = TRUE,
  x.breaks = NULL,
  y.breaks = NULL
)
```

Arguments

<code>estimates</code>	The result of an <code>sdid</code> object or a list of <code>sdid</code> estimates. For a staggered <code>sdid</code> result, each adoption period is plotted in a stacked panel.
<code>treated.name</code>	A string conveying the name of the treated curve that appears in the legend. Defaults to "treated".
<code>control.name</code>	A string conveying the name of the control curve that appears in the legend. Defaults to 'synthetic control'.

<code>spaghetti.units</code>	A list of units to plot individually alongside the treatment and control curves. Defaults to the empty list.
<code>spaghetti.matrices</code>	A list of matrices, one per estimate, of extra trajectories to plot. Each row is a trajectory, one column per period, and row names are used as labels.
<code>lambda.comparable</code>	Logical argument where if TRUE, the lambda weight axis is scaled to range from 0 to 1. Useful for when comparing across stacked panels. Defaults to FALSE.
<code>overlay</code>	Between 0 and 1. Shifts the synthetic control toward the treated line by this fraction of their pre-treatment gap, which makes it easier to see whether they move in parallel. Defaults to 0 (no shift). A vector of values can be animated with <code>plotly</code> .
<code>lambda.plot.scale</code>	Height of the time-weight band: the range of the trajectories divided by this number. Defaults to 3.
<code>effect.curvature</code>	Curvature of the effect arrow in the effects style.
<code>line.width</code>	Width of the trajectory lines.
<code>point.size</code>	Size of the points in the effects style.
<code>onset.alpha</code>	Opacity of the red dashed line that marks treatment.
<code>spaghetti.line.width</code> , <code>spaghetti.label.size</code> , <code>spaghetti.line.alpha</code> , <code>spaghetti.label.alpha</code>	Width, label size and opacity of the spaghetti trajectories and their labels.
<code>alpha.multiplier</code>	One number per estimate that scales its opacity, to emphasize some estimates over others.
<code>style</code>	The plot's style, "minimal" (the default) or "effects". See Details.
<code>lambda.axis</code>	Logical argument where if TRUE, each panel with a time weights' band shows the band's scale on a right-hand axis. Defaults to TRUE.
<code>x.breaks</code> , <code>y.breaks</code>	Optional tick positions for the time and outcome axes (numbers, or dates for a date axis). The axes are extended to the outermost ticks if needed. With stacked panels, <code>y.breaks</code> applies to every panel.

Details

Two plotting styles are available: "minimal" and "effects". The default, `style = "minimal"`, plots both trajectories in black, with the treated line solid and the synthetic control dashed. The time weights are plotted as a gray band with their scale on a right-hand axis, and a red dashed line marks treatment. `style = "effects"` adds a diff-in-diff diagram of the treatment effect and fades the trajectory lines.

A staggered `sdid()` result is drawn as one panel per cohort, stacked in a column and labeled by adoption period. A list of results is drawn the same way, one panel each. To change the axis ticks,

use `x.breaks` and `y.breaks` rather than adding your own scale, which would drop the time-weight axis.

Requires `ggplot2`.

Value

A `ggplot` object, which can be extended with `+`.

<code>sdid_units_plot</code>	<i>Plots each control unit's weight and its own estimate</i>
------------------------------	--

Description

Synthetic difference-in-differences estimates the effect as a weighted average, over the control units, of the difference-in-differences comparison of the treated units with each control alone, using the time weights. This plot shows those comparisons with their unit weights, for an estimate or each cohort of a staggered design.

Usage

```
sdid_units_plot(
  estimates,
  type = c("scatter", "bars"),
  cohort = NULL,
  label.units = NULL,
  weighted.only = FALSE
)
```

Arguments

<code>estimates</code>	The result of <code>sdid()</code> , or a named list of results with a single adoption period each, drawn in panels labeled by their names. A staggered result is drawn with a panel for each cohort, unless <code>cohort</code> chooses one.
<code>type</code>	The kind of plot. "scatter" (the default) plots each control's weight against its own estimate. "bars" draws a bar from the effect to each weighted control's own estimate, ordered by weight. See Details.
<code>cohort</code>	For a staggered result, the adoption period of the one cohort to plot, as it appears in the result's cohorts. Defaults to <code>NULL</code> , plotting every cohort in its own panel.
<code>label.units</code>	Which controls to label: "all", "weighted" (positive weight only), or a number <code>n</code> for the <code>n</code> largest weights. Defaults to "all" for one panel and 10 for several. The bars only show labeled controls.
<code>weighted.only</code>	TRUE to leave controls with no weight out of the scatter. Defaults to FALSE, drawing them as open circles.

Details

type = "scatter" (the default) plots each control's weight against its own estimate, with the overall effect as a red line. Controls with zero weight are drawn as open circles. type = "bars" draws a bar from the effect to each weighted control's estimate, largest weight first; bars to the right pull the estimate up, bars to the left pull it down.

Requires ggplot2. If ggrepel is installed, it is used to keep labels from overlapping.

Value

A ggplot object, which can be extended with +.

weights.sdid	<i>Unit and time weights of an sdid result</i>
--------------	--

Description

Unit and time weights of an sdid result

Usage

```
## S3 method for class 'sdid'
weights(object, ...)
```

Arguments

object	The result of sdid().
...	Not used.

Value

A list of two data frames. omega has a row per unit never treated and lambda a row per period, each with a column of weights for each adoption period. A period after a cohort's adoption has no time weight for that cohort and is NA.

Index

california_prop99, [2](#)
coef.sdidi(sdidi-methods), [7](#)
confint.sdidi(sdidi-methods), [7](#)
CPS, [3](#)

glance.sdidi(sdidi-tidiers), [8](#)

nobs.sdidi(sdidi-methods), [7](#)

PENN, [3](#)
plot.sdidi(sdidi-methods), [7](#)
print.sdidi(sdidi-methods), [7](#)

sdidi, [4](#)
sdidi(), [7](#), [8](#)
sdidi-methods, [6](#), [7](#)
sdidi-tidiers, [8](#)
sdidi_placebo, [9](#)
sdidi_placebo_plot, [10](#)
sdidi_placebo_plot(), [10](#)
sdidi_plot, [11](#)
sdidi_plot(), [8](#)
sdidi_units_plot, [13](#)
summary.sdidi(sdidi-methods), [7](#)

tidy.sdidi(sdidi-tidiers), [8](#)

vcov.sdidi(sdidi-methods), [7](#)

weights.sdidi, [14](#)
weights.sdidi(), [6](#), [7](#)