

Package: synthdid (via r-universe)

September 22, 2024

Title Synthetic Difference-in-Difference Estimation

Version 0.0.9

Author Dmitry Arkhangelsky [aut] Susan Athey [aut] David A. Hirshberg [aut, cre] Guido W. Imbens [aut] Stefan Wager [aut]

Maintainer David A. Hirshberg <david.a.hirshberg@gmail.com>

Description Estimate average treatment effects in panel data.
Currently provides methods only for the case that all treated units adopt treatment at the same time.

Depends R (>= 3.4)

Imports mvtnorm

License GPL (>= 2) | BSD_3_clause + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.1

Suggests testthat, ggplot2, CVXR

URL <https://github.com/synth-inference/synthdid>

BugReports <https://github.com/synth-inference/synthdid/issues>

Repository <https://skranz.r-universe.dev>

RemoteUrl <https://github.com/skranz/synthdid>

RemoteRef master

RemoteSha a05029b97d7118bd698c4b5aeea8516c46b4901e

Contents

ar2_correlation_matrix	2
california_prop99	3
CPS	4
decompose_Y	4

did_estimate	5
estimate_dgp	5
fit_ar2	6
format.synthdid_estimate	6
lindsey_density_estimate	7
panel.matrices	7
PENN	8
plot.synthdid_estimate	9
print.synthdid_estimate	9
randomize_treatment	10
sc_estimate	10
simulate_dgp	11
sparsify_function	12
summary.synthdid_estimate	12
synthdid_controls	13
synthdid_effect_curve	13
synthdid_estimate	14
synthdid_placebo	15
synthdid_placebo_plot	16
synthdid_plot	16
synthdid_rmse_plot	19
synthdid_se	19
synthdid_units_plot	20
timesteps	20
vcov.synthdid_estimate	21

Index 23

ar2_correlation_matrix

compute the correlation matrix of a time series generated by an ar2 model

Description

compute the correlation matrix of a time series generated by an ar2 model

Usage

```
ar2_correlation_matrix(ar_coef, T)
```

Arguments

ar_coef	the coefficients of the ar2 model: c(lag-1-coefficient, lag-2-coefficient)
T	the length of the time series

Value

the correlation matrix

california_prop99 *California proposition 99*

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 treated 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

```
# Load tobacco sales in long panel format.
data("california_prop99")
# Transform to N*T matrix format required for synthdid,
# where N is the number of units and T the time periods.
setup <- panel.matrices(california_prop99)
```

CPS

*CPS***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

decompose_Y

*Decompose Y into components F, M, and E as described in Section 3.1.1 also computes a set of 'unit factors': the first **rank** left singular vectors from SVD(Y)*

Description

Decompose Y into components F, M, and E as described in Section 3.1.1 also computes a set of 'unit factors': the first **rank** left singular vectors from SVD(Y)

Usage

decompose_Y(Y, rank)

Arguments

Y the outcomes

rank the assumed rank of the signal component $L=F+M$ **Value**

a list with elements F, M, E, and unit_factors

did_estimate	<i>synthdid_estimate for diff-in-diff estimates. Takes all the same parameters, but by default, passes options to use the diff-in-diff estimator</i>
--------------	--

Description

synthdid_estimate for diff-in-diff estimates. Takes all the same parameters, but by default, passes options to use the diff-in-diff estimator

Usage

```
did_estimate(Y, N0, T0, ...)
```

Arguments

Y	the observation matrix.
N0	the number of control units. Rows 1-N0 of Y correspond to the control units.
T0	the number of pre-treatment time steps. Columns 1-T0 of Y correspond to pre-treatment time steps.
...	additional options for synthdid_estimate

Value

an object like that returned by synthdid_estimate

estimate_dgp	<i>Estimates the DGP parameters used in the placebo studies in Sections 3 and 5 of the synthetic difference in differences paper. Described there in Section 3.1.1.</i>
--------------	---

Description

Estimates the DGP parameters used in the placebo studies in Sections 3 and 5 of the synthetic difference in differences paper. Described there in Section 3.1.1.

Usage

```
estimate_dgp(Y, assignment_vector, rank)
```

Arguments

Y	an NxT matrix of outcomes
assignment_vector	an Nx1 vector of treatment assignments
rank	the rank of the estimated signal component L

Value

a list with elements F, M, Sigma, pi as described in Section 3.1.1 and an element ar_coef with the AR(2) model coefficients underlying the covariance Sigma

fit_ar2	<i>Estimate ar2 coefficients from iid time series</i>
---------	---

Description

Estimate ar2 coefficients from iid time series

Usage

```
fit_ar2(E)
```

Arguments

E a matrix with those time series as rows

Value

a vector of ar2 coefficients: c(lag-1-coefficient, lag-2-coefficient)

format.synthdid_estimate	<i>Format a synthdid object</i>
--------------------------	---------------------------------

Description

Format a synthdid object

Usage

```
## S3 method for class 'synthdid_estimate'
format(x, ...)
```

Arguments

x The object to format
 ... Additional arguments (currently ignored).

`lindsey_density_estimate`

Computes a density estimator by smoothing a histogram using Poisson regression. Implementation of "Lindsey's method", as described in Chapter 10 of "Computer age statistical inference: algorithms, evidence, and data science" by Bradley Efron and Trevor Hastie (2016).

Description

Computes a density estimator by smoothing a histogram using Poisson regression. Implementation of "Lindsey's method", as described in Chapter 10 of "Computer age statistical inference: algorithms, evidence, and data science" by Bradley Efron and Trevor Hastie (2016).

Usage

```
lindsey_density_estimate(x, K, deg)
```

Arguments

- | | |
|------------------|---|
| <code>x</code> | • one-dimensional vector of data; |
| <code>K</code> | • number of bins in the histogram; |
| <code>deg</code> | • degree of natural splines used in Poisson regression; |

Value

a list with 2 fields, centers and density, which are K-dimensional vectors containing the bin centers and estimated density within each bin respectively.

`panel.matrices`

Convert a long (balanced) panel to a wide matrix

Description

Converts a data set in panel form to matrix format required by synthdid estimators. A typical long panel data set looks like [unit, time, outcome, treatment]. Synthdid requires a balanced panel with simultaneous adoption of treatment: each unit must be observed at all times, and all treated units must begin treatment simultaneously. This function creates num.units x num.time.periods matrices Y and W of outcomes and treatment indicators. In these matrices, columns are sorted by time, and by default (when treated.last=TRUE), rows for control units appear before those of treated units.

Usage

```
panel.matrices(
  panel,
  unit = 1,
  time = 2,
  outcome = 3,
  treatment = 4,
  treated.last = TRUE
)
```

Arguments

panel	A data.frame with columns consisting of units, time, outcome, and treatment indicator.
unit	The column number/name corresponding to the unit identifier. Default is 1.
time	The column number/name corresponding to the time identifier. Default is 2.
outcome	The column number/name corresponding to the outcome identifier. Default is 3.
treatment	The column number/name corresponding to the treatment status. Default is 4.
treated.last	Should we sort the rows of Y and W so treated units are last. If FALSE, sort by unit number/name. Default is TRUE.

Value

A list with entries Y: the data matrix, N0: the number of control units, T0: the number of time periods before treatment, W: the matrix of treatment indicators.

Examples

```
# Load tobacco sales in long panel format.
data("california_prop99")
# Transform to N*T matrix format required for synthdid,
# where N is the number of units and T the time periods.
setup <- panel.matrices(california_prop99, unit = 1, time = 2, outcome = 3, treatment = 4)

# Compute synthdid estimate
synthdid_estimate(setup$Y, setup$N0, setup$T0)
```

PENN

PENN

Description

PENN

Usage

```
data(PENN)
```

Format

A data frame with 3219 rows and 5 variables.

country country

year year

log_gdp log_gdp

dem dem

educ educ

plot.synthdid_estimate

Plot a synthdid object

Description

Plot a synthdid object

Usage

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

Arguments

x The object to plot
... Additional arguments (currently ignored).

print.synthdid_estimate

Print a synthdid object

Description

Print a synthdid object

Usage

```
## S3 method for class 'synthdid_estimate'  
print(x, ...)
```

Arguments

x The object to print
 ... Additional arguments (currently ignored).

randomize_treatment *randomize treatment to n units with probability pi then if the number of treated units is zero, assign treatment to one unit uniformly at random and if the number of treated units exceeds a cap, remove treatment uniformly at random so it is exactly that cap*

Description

randomize treatment to n units with probability pi then if the number of treated units is zero, assign treatment to one unit uniformly at random and if the number of treated units exceeds a cap, remove treatment uniformly at random so it is exactly that cap

Usage

```
randomize_treatment(pi, N, N1)
```

Arguments

pi the randomization probabilities
 N the number of units
 N1 the cap on the number of treated units

Value

a binary vector of length N, with ones indicating assignment to treatment

sc_estimate *synthdid_estimate for synthetic control estimates. Takes all the same parameters, but by default, passes options to use the synthetic control estimator By default, this uses only 'infinitesimal' ridge regularization when estimating the weights.*

Description

synthdid_estimate for synthetic control estimates. Takes all the same parameters, but by default, passes options to use the synthetic control estimator By default, this uses only 'infinitesimal' ridge regularization when estimating the weights.

Usage

```
sc_estimate(Y, N0, T0, eta.omega = 1e-06, ...)
```

Arguments

Y	the observation matrix.
N0	the number of control units. Rows 1-N0 of Y correspond to the control units.
T0	the number of pre-treatment time steps. Columns 1-T0 of Y correspond to pre-treatment time steps.
eta.omega	determines the level of ridge regularization, $\text{zeta.omega} = \text{eta.omega} * \text{noise.level}$, as in <code>synthdid_estimate</code> .
...	additional options for <code>synthdid_estimate</code>

Value

an object like that returned by `synthdid_estimate`

<code>simulate_dgp</code>	<i>Simulates data from DGPs used in the placebo studies in Sections 3 and 5 of the synthetic difference in differences paper. Described there in Section 3.1.1.</i>
---------------------------	---

Description

Simulates data from DGPs used in the placebo studies in Sections 3 and 5 of the synthetic difference in differences paper. Described there in Section 3.1.1.

Usage

```
simulate_dgp(parameters, N1, T1)
```

Arguments

parameters	a list of dgp parameters (F,M,Sigma,pi) as output by <code>estimate.dgp</code>
N1	a cap on the number of treated units,
T1	the number of treated periods.

Value

a list with 3 elements: the outcome matrix Y, the number of control units N0, and the number of control periods T0. The first N0 rows of Y are for units assigned to control, the remaining rows are for units assigned to treatment.

sparsify_function	<i>A function mapping a numeric vector to a (presumably sparser) numeric vector of the same shape to be passed onto synthdid_estimate.</i>
-------------------	--

Description

A function mapping a numeric vector to a (presumably sparser) numeric vector of the same shape to be passed onto synthdid_estimate.

Usage

```
sparsify_function(v)
```

Arguments

v	a vector
---	----------

summary.synthdid_estimate	<i>Summarize a synthdid object</i>
---------------------------	------------------------------------

Description

Summarize a synthdid object

Usage

```
## S3 method for class 'synthdid_estimate'
summary(object, weight.digits = 3, fast = FALSE, ...)
```

Arguments

object	The object to summarize
weight.digits	The number of digits to use when displaying weights (omega, lambda)
fast	Be fast but less accurate, e.g. jackknife instead of bootstrap se estimate
...	Additional arguments (currently ignored).

synthdid_controls	<i>Outputs a table of important synthetic controls and their corresponding weights, sorted by weight. The table is truncated to exclude synthetic controls that do not matter for any estimate — for each estimate, the truncated controls may have total weight no larger than 1-mass.</i>
-------------------	---

Description

Outputs a table of important synthetic controls and their corresponding weights, sorted by weight. The table is truncated to exclude synthetic controls that do not matter for any estimate — for each estimate, the truncated controls may have total weight no larger than 1-mass.

Usage

```
synthdid_controls(estimates, sort.by = 1, mass = 0.9, weight.type = "omega")
```

Arguments

estimates	a list of estimates output by synthdid_estimate. Or a single estimate.
sort.by	the index of the estimate to sort by. Defaults to 1.
mass	which controls the length of the table. Defaults to 0.9.
weight.type	'omega' for units, 'lambda' for time periods

synthdid_effect_curve	<i>Outputs the effect curve that was averaged to produce our estimate</i>
-----------------------	---

Description

Outputs the effect curve that was averaged to produce our estimate

Usage

```
synthdid_effect_curve(estimate)
```

Arguments

estimate	as output by synthdid_estimate
----------	--------------------------------

synthdid_estimate	<i>Computes the synthetic diff-in-diff estimate for an average treatment effect on a treated block.</i>
-------------------	---

Description

See 'Synthetic Difference in Differences' by Arkhangelsky et al. This implements Algorithm 1.

Usage

```
synthdid_estimate(
  Y,
  N0,
  T0,
  X = array(dim = c(dim(Y), 0)),
  noise.level = sd(apply(Y[1:N0, 1:T0], 1, diff)),
  eta.omega = ((nrow(Y) - N0) * (ncol(Y) - T0))^(1/4),
  eta.lambda = 1e-06,
  zeta.omega = eta.omega * noise.level,
  zeta.lambda = eta.lambda * noise.level,
  omega.intercept = TRUE,
  lambda.intercept = TRUE,
  weights = list(omega = NULL, lambda = NULL),
  update.omega = is.null(weights$omega),
  update.lambda = is.null(weights$lambda),
  min.decrease = 1e-05 * noise.level,
  max.iter = 10000,
  sparsify = sparsify_function,
  max.iter.pre.sparsify = 100
)
```

Arguments

Y	the observation matrix.
N0	the number of control units (N_{co} in the paper). Rows 1-N0 of Y correspond to the control units.
T0	the number of pre-treatment time steps (T_{pre} in the paper). Columns 1-T0 of Y correspond to pre-treatment time steps.
X	an optional 3-D array of time-varying covariates. Shape should be $N \times T \times X \times C$ for C covariates.
noise.level	an estimate of the noise standard deviation sigma. Defaults to the standard deviation of first differences of Y.
eta.omega	determines the tuning parameter $zeta.omega = eta.omega * noise.level$. Defaults to the value $(N_{tr} T_{post})^{1/4}$.
eta.lambda	analogous for lambda. Defaults to an 'infinitesimal' value 1e-6.

<code>zeta.omega</code>	if passed, overrides the default <code>zeta.omega = eta.omega * noise.level</code> . Deprecated.
<code>zeta.lambda</code>	analogous for lambda.
<code>omega.intercept</code>	Binary. Use an intercept when estimating omega.
<code>lambda.intercept</code>	Binary. Use an intercept when estimating lambda.
<code>weights</code>	a list with fields <code>lambda</code> and <code>omega</code> . If non-null <code>weights\$lambda</code> is passed, we use them instead of estimating lambda weights. Same for <code>weights\$omega</code> .
<code>update.omega</code>	If true, solve for omega using the passed value of <code>weights\$omega</code> only as an initialization. If false, use it exactly as passed. Defaults to false if a non-null value of <code>weights\$omega</code> is passed.
<code>update.lambda</code>	Analogous.
<code>min.decrease</code>	Tunes a stopping criterion for our weight estimator. Stop after an iteration results in a decrease in penalized MSE smaller than <code>min.decrease^2</code> .
<code>max.iter</code>	A fallback stopping criterion for our weight estimator. Stop after this number of iterations.
<code>sparsify</code>	A function mapping a numeric vector to a (presumably sparser) numeric vector of the same shape, which must sum to one. If not null, we try to estimate sparse weights via a second round of Frank-Wolfe optimization initialized at <code>sparsify</code> (the solution to the first round).
<code>max.iter.pre.sparsify</code>	Analogous to <code>max.iter</code> , but for the pre-sparsification first-round of optimization. Not used if <code>sparsify=NULL</code> .

Value

An average treatment effect estimate with `'weights'` and `'setup'` attached as attributes. `'weights'` contains the estimated weights lambda and omega and corresponding intercepts, as well as regression coefficients beta if X is passed. `'setup'` is a list describing the problem passed in: Y, N0, T0, X.

<code>synthdid_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

```
synthdid_placebo(estimate, treated.fraction = NULL)
```

Arguments

estimate as output by synthdid_estimate

treated.fraction the fraction of pre-treatment data to use as a placebo treatment period Defaults to NULL, which indicates that it should be the fraction of post-treatment to pre-treatment data

synthdid_placebo_plot *For our estimator and a placebo, plots treated and synthetic control trajectories and overlays a 2x2 diff-in-diff diagram. Requires ggplot2*

Description

For our estimator and a placebo, plots treated and synthetic control trajectories and overlays a 2x2 diff-in-diff diagram. Requires ggplot2

Usage

```
synthdid_placebo_plot(estimate, overlay = FALSE, treated.fraction = NULL)
```

Arguments

estimate as output by synthdid_estimate.

overlay binary, indicates whether plots should be overlaid or shown in different facets. Defaults to FALSE.

treated.fraction as in synthdid_placebo

synthdid_plot *Plots treated and synthetic control trajectories and overlays a 2x2 diff-in-diff diagram of our estimator. In this overlay, the treatment effect is indicated by an arrow. The weights lambda defining our synthetic pre-treatment time period are plotted below. If a list of estimates is passed, plots all of them. By default, does this in different facets. To overlay estimates in the same facet, indicate a facet for each estimator in the argument 'facet'.*

Description

For SC estimates, i.e., if lambda is a vector of zeros, plots the trajectories and SC estimate of the effect, but no diagram.

Usage

```

synthdid_plot(
  estimates,
  treated.name = "treated",
  control.name = "synthetic control",
  spaghetti.units = c(),
  spaghetti.matrices = NULL,
  facet = NULL,
  facet.vertical = TRUE,
  lambda.comparable = !is.null(facet),
  overlay = 0,
  lambda.plot.scale = 3,
  trajectory.linetype = 1,
  effect.curvature = 0.3,
  line.width = 0.5,
  guide.linetype = 2,
  point.size = 1,
  trajectory.alpha = 0.5,
  diagram.alpha = 0.95,
  effect.alpha = 0.95,
  onset.alpha = 0.3,
  ci.alpha = 0.3,
  spaghetti.line.width = 0.2,
  spaghetti.label.size = 2,
  spaghetti.line.alpha = 0.3,
  spaghetti.label.alpha = 0.5,
  se.method = "jackknife",
  alpha.multiplier = NULL
)

```

Arguments

<code>estimates</code>	a list of estimates output by <code>synthdid_estimate</code> . Or a single estimate.
<code>treated.name</code>	the name of the treated curve that appears in the legend. Defaults to 'treated'
<code>control.name</code>	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. <code>spaghetti.unit %in% rownames(Y)</code> must work. Defaults to the empty list.
<code>spaghetti.matrices</code>	a list of matrices — one for each element of <code>estimates</code> — of trajectories to plot individually. The rows of these matrices should be the same length as the trajectories in <code>Y</code> and they must be named — <code>set rownames(spaghetti.trajectories[[i]])</code> — so trajectories can be labeled in the plot.
<code>facet</code>	a list of the same length as <code>estimates</code> indicating the facet in which to plot each estimate. The values of the elements of the list are used to label the facets. If <code>NULL</code> , plot each estimate in a different facet. Defaults to <code>NULL</code> .

facet.vertical	TRUE if facets should be stacked vertically. Defaults to FALSE (horizontal).
lambda.comparable	TRUE if the weights lambda should be plotted in such a way that the ribbons have the same mass from plot to plot, assuming the treated curve is the same. Useful for side-by-side or overlaid plots. Defaults to FALSE if facet is not passed, TRUE if passed.
overlay	a number in [0,1] defaulting to 0, can be used to shift the control trajectory toward the treated trajectory. When a nonzero value is passed, we plot the control after subtracting that fraction of the diff-in-diff style adjustment for the difference between lambda-weighted pre-treatment averages of the treated and control. With intercept of almost one, this makes it easier to assess parallel-ness by making trajectories closer. With intercept of one, this essentially overlays the curves, and plotting a diagram is suppressed as in the case of a SC estimate. To use different values for different plots, pass these values as an attribute 'overlay' of each estimate. If a vector is passed, plots at different intercept levels indicated by the 'frame' aesthetic. ggplotly will interpret this as an animation.
lambda.plot.scale	determines the scale of the plot of the weights lambda.
trajectory.linetype	the linetype of the treated and synthetic control trajectories
effect.curvature	the curvature of the arrows indicating the treatment effect. Defaults to zero. Nonzero values help avoid overplotting when plotting multiple estimates in one facet.
line.width	the line width.
guide.linetype	determines the (ggplot) linetype of the vertical segments of the parallelogram
point.size	determines the size of the points of the parallelogram
trajectory.alpha	determines transparency of trajectories
diagram.alpha	determines transparency of diff-in-diff diagram
effect.alpha	determines transparency of effect arrows
onset.alpha	determines transparency of vertical lines indicating onset of treatment
ci.alpha	determines transparency of the arrows illustrating upper and lower bounds of a 95% confidence interval for the effect
spaghetti.line.width	determines the width of spaghetti trajectories
spaghetti.label.size	determines the size of spaghetti labels
spaghetti.line.alpha	determines transparency of spaghetti trajectories
spaghetti.label.alpha	determines transparency of spaghetti trajectory labels
se.method	determines the method used to calculate the standard error used for this confidence interval. if 'none', don't show the interval

`alpha.multiplier`

a vector of the same length as estimates, is useful for comparing multiple estimates in one facet but highlighting one or several. All plot elements associated with the estimate are displayed with alpha multiplied by the corresponding element of `alpha.multiplier`. Defaults to a vector of ones.

Details

Requires `ggplot2` Due to differences between `ggplot` and `ggplotly`, this will warn about an unknown aesthetic frame.

<code>synthdid_rmse_plot</code>	<i>A diagnostic plot for <code>sc.weight.fw.covariates</code>. Plots the objective function, regularized RMSE, as a function of the number of Frank-Wolfe / Gradient steps taken. Requires <code>ggplot2</code></i>
---------------------------------	---

Description

A diagnostic plot for `sc.weight.fw.covariates`. Plots the objective function, regularized RMSE, as a function of the number of Frank-Wolfe / Gradient steps taken. Requires `ggplot2`

Usage

```
synthdid_rmse_plot(estimates)
```

Arguments

`estimates` a list of estimates output by `synthdid_estimate`. Or a single estimate.

<code>synthdid_se</code>	<i>Calculate the standard error of a synthetic diff in diff estimate. Deprecated. Use <code>vcov.synthdid_estimate</code>.</i>
--------------------------	--

Description

Calculate the standard error of a synthetic diff in diff estimate. Deprecated. Use `vcov.synthdid_estimate`.

Usage

```
synthdid_se(...)
```

Arguments

`...` Any valid arguments for `vcov.synthdid_estimate`

`synthdid_units_plot` *Plots unit by unit difference-in-differences. Dot size indicates the weights ω_i used in the average that yields our treatment effect estimate. This estimate and endpoints of a 95% CI are plotted as horizontal lines. Requires `ggplot2`*

Description

Plots unit by unit difference-in-differences. Dot size indicates the weights ω_i used in the average that yields our treatment effect estimate. This estimate and endpoints of a 95% CI are plotted as horizontal lines. Requires `ggplot2`

Usage

```
synthdid_units_plot(
  estimates,
  negligible.threshold = 0.001,
  negligible.alpha = 0.3,
  se.method = "jackknife",
  units = NULL
)
```

Arguments

`estimates` as output by `synthdid_estimate`. Can be a single one or a list of them.

`negligible.threshold` Unit weight threshold below which units are plotted as small, transparent `xs` instead of circles. Defaults to `.001`.

`negligible.alpha` Determines transparency of those `xs`.

`se.method` the method used to calculate standard errors for the CI. See `vcov.synthdid_estimate`. Defaults to `'jackknife'` for speed. If `'none'`, don't plot a CI.

`units` a list of control units — elements of `rownames(Y)` — to plot differences for. Defaults to `NULL`, meaning all of them.

`timesteps` *Get timesteps from panel matrix Y*

Description

`timesteps` are stored as `colnames(Y)`, but column names cannot be Date objects. Instead, we use strings. If they are strings convertible to dates, return that

Usage

```
timesteps(Y)
```

Arguments

Y a matrix

Value

its column names interpreted as Dates if possible

```
vcov.synthdid_estimate
```

Calculate Variance-Covariance Matrix for a Fitted Model Object

Description

Provides variance estimates based on the following three options

- The bootstrap, Algorithm 2 in Arkhangelsky et al.
- The jackknife, Algorithm 3 in Arkhangelsky et al.
- Placebo, Algorithm 4 in Arkhangelsky et al.

Usage

```
## S3 method for class 'synthdid_estimate'
vcov(
  object,
  method = c("bootstrap", "jackknife", "placebo"),
  replications = 200,
  ...
)
```

Arguments

object A synthdid model

method the CI method. The default is bootstrap (warning: this may be slow on large data sets, the jackknife option is the fastest, with the caveat that it is not recommended for SC).

replications the number of bootstrap replications

... Additional arguments (currently ignored).

Details

The jackknife is not recommended for SC, see section 5 in Arkhangelsky et al. "placebo" is the only option that works for only one treated unit.

References

Dmitry Arkhangelsky, Susan Athey, David A. Hirshberg, Guido W. Imbens, and Stefan Wager. "Synthetic Difference in Differences". arXiv preprint arXiv:1812.09970, 2019.

Index

ar2_correlation_matrix, 2

california_prop99, 3
CPS, 4

decompose_Y, 4
did_estimate, 5

estimate_dgp, 5

fit_ar2, 6
format.synthdid_estimate, 6

lindsey_density_estimate, 7

panel.matrices, 7
PENN, 8
plot.synthdid_estimate, 9
print.synthdid_estimate, 9

randomize_treatment, 10
rank, 4

sc_estimate, 10
simulate_dgp, 11
sparsify_function, 12
summary.synthdid_estimate, 12
synthdid_controls, 13
synthdid_effect_curve, 13
synthdid_estimate, 14
synthdid_placebo, 15
synthdid_placebo_plot, 16
synthdid_plot, 16
synthdid_rmse_plot, 19
synthdid_se, 19
synthdid_units_plot, 20

timesteps, 20

vcov.synthdid_estimate, 21