Package ‘sccic’

May 9, 2026

Type Package

Title Synthetic Control Changes-in-Changes Estimator

Version 0.1.1

Description Implements the Changes-in-Changes (CIC) estimator of Athey and Imbens (2006) <doi:10.1111/j.1468-0262.2006.00668.x> combined with synthetic control methods. Provides both the continuous CIC estimator (Theorem 3.1) and the discrete CIC estimator (Theorem 4.1) for integer-valued outcomes, with analytic and bootstrap inference. Also provides nonparametric estimation of the entire counterfactual distribution of outcomes for a treated group, allowing evaluation of average, quantile, and distributional treatment effects. Synthetic control weights are constructed via elastic net regularization to handle settings with many potential control units.

License GPL (>= 3)

Encoding UTF-8

Depends R (>= 3.5.0)

Imports glmnet, stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown, ggplot2, wooldridge, qte, Synth

Config/testthat/edition 3

VignetteBuilder knitr

RoxygenNote 7.2.3

URL `https://github.com/neilhwang/sccic`

BugReports `https://github.com/neilhwang/sccic/issues`

NeedsCompilation no

Author Neil Hwang [aut, cre]

Maintainer Neil Hwang <`neil.hwang@bcc.cuny.edu`>

Repository CRAN

Date/Publication 2026-04-10 11:00:08 UTC

Contents

check_data 2
check_support 3
cic 3
loo_donors 5
plot.sc_cic 6
plot_distributions 6
plot_qq 7
plot_qte 7
quantile_te 8
sc_cic 8
sc_weights 10
sensitivity_alpha 11
simulate_sccic 12
summarize_simulation 13

Index **14**

`check_data` *Validate CIC Input Data*

Description

Checks for common data issues and issues informative warnings. Called internally by `cic` when input is potentially problematic. Also available for manual use.

Usage

```
check_data(y_00, y_01, y_10, y_11)
```

Arguments

`y_00`, `y_01`, `y_10`, `y_11`	Numeric vectors of outcomes.

Value

Invisible `TRUE`. Produces warnings for potential issues.

`check_support` *Check Support Condition*

Description

Warns if the treated unit's pre-treatment outcomes fall outside the range of the synthetic control's pre-treatment outcomes, which can cause the CIC distributional transport to extrapolate.

Usage

```
check_support(x)
```

Arguments

`x`	An object of class `"sc_cic"`.

Value

Logical. `TRUE` if support condition is satisfied.

`cic` *Changes-in-Changes Estimator*

Description

Implements the Changes-in-Changes (CIC) estimator of Athey and Imbens (2006) for the average treatment effect on the treated in a two-group, two-period difference-in-differences setting.

Usage

```
cic(
  y_00,
  y_01,
  y_10,
  y_11,
  se = TRUE,
  boot = FALSE,
  boot_iters = 500L,
  seed = NULL,
  discrete = FALSE
)
```

Arguments

`y_00`	Numeric vector. Outcomes for the control group in the pre-treatment period.
`y_01`	Numeric vector. Outcomes for the control group in the post-treatment period.
`y_10`	Numeric vector. Outcomes for the treated group in the pre-treatment period.
`y_11`	Numeric vector. Outcomes for the treated group in the post-treatment period.
`se`	Logical. If `TRUE` (default), compute analytic standard errors using the asymptotic variance from Theorem 5.1 of Athey and Imbens (2006). Ignored (with a message) when `discrete = TRUE`, because Theorem 5.1 is derived under the continuous-distribution assumption.
`boot`	Logical. If `TRUE`, also compute bootstrap standard errors. Default is `FALSE`.
`boot_iters`	Integer. Number of bootstrap iterations. Default 500.
`seed`	Integer or `NULL`. Random seed for bootstrap.
`discrete`	Logical. If `FALSE` (default), use the continuous CIC estimator (Theorem~3.1 of Athey and Imbens 2006), which applies $F_{01}^{-1}(F_{00}(y_{10,i}))$ to each pre-treatment treated observation. If `TRUE`, use the discrete CIC estimator (Theorem~4.1), which integrates the counterfactual over the quantile band $[F_{00}(y^-), F_{00}(y)]$ to handle mass points in the outcome distribution. Use `discrete = TRUE` when the outcome takes a small number of distinct values (e.g., integer counts).

Details

The CIC estimator constructs a counterfactual distribution for the treated group in the post-treatment period by applying the transformation:

$$Y_{11}^{N,CIC} = F_{Y,01}^{-1}(F_{Y,00}(Y_{10}))$$

The average treatment effect is then:

$$\hat{\tau}^{CIC} = \frac{1}{N_{11}} \sum Y_{11,i} - \frac{1}{N_{10}} \sum F_{Y,01}^{-1}(F_{Y,00}(Y_{10,i}))$$

The analytic variance follows Theorem 5.1 of Athey and Imbens (2006):

$$Var(\sqrt{N}\hat{\tau}^{CIC}) = V^p/\alpha_{00} + V^q/\alpha_{01} + V^r/\alpha_{10} + V^s/\alpha_{11}$$

Value

An object of class `"cic"` containing:

`tau`	The CIC average treatment effect estimate.
`se`	Analytic standard error (if `se = TRUE`).
`z`	z-statistic.
`pval`	Two-sided p-value.
`counterfactual_mean`	Mean of the counterfactual distribution.
`tau_did`	The standard DID estimate for comparison.

`N`	Total sample size.
`n`	Named vector of group sample sizes.
`boot_se`	Bootstrap standard error (if `boot = TRUE`).
`ecdfs`	List of empirical CDF objects for each group.

References

Athey, S. and Imbens, G. W. (2006). Identification and Inference in Nonlinear Difference-in-Differences Models. *Econometrica*, 74(2), 431–497. doi:10.1111/j.14680262.2006.00668.x

Examples

```
# Workers' compensation example (Meyer, Viscusi, and Durbin 1995)
if (requireNamespace("wooldridge", quietly = TRUE)) {
  data("injury", package = "wooldridge")
  result <- cic(
    y_00 = injury$ldurat[injury$highearn == 0 & injury$afchnge == 0],
    y_01 = injury$ldurat[injury$highearn == 0 & injury$afchnge == 1],
    y_10 = injury$ldurat[injury$highearn == 1 & injury$afchnge == 0],
    y_11 = injury$ldurat[injury$highearn == 1 & injury$afchnge == 1]
  )
  print(result)
}
```

`loo_donors` *Leave-One-Out Donor Analysis*

Description

Re-estimates SC-CIC dropping one donor at a time to assess sensitivity to individual donors.

Usage

```
loo_donors(y_treated, y_donors, treatment_period, alpha = 1, seed = 42)
```

Arguments

`y_treated`	Numeric vector. Treated unit outcomes.
`y_donors`	Numeric matrix. Donor unit outcomes.
`treatment_period`	Integer. First treatment period index.
`alpha`	Elastic net mixing parameter.
`seed`	Integer or `NULL`. Random seed.

Value

A data frame with one row per donor, showing the SC-CIC estimate when that donor is excluded.

`plot.sc_cic` *Plot Pre-treatment Fit for SC-CIC*

Description

Plots the treated unit against the synthetic control over time, with a vertical line at the treatment period.

Usage

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

Arguments

`x`	An object of class `"sc_cic"`.
`...`	Additional arguments passed to `plot`.

Details

The plot shows the treated unit (solid line) and synthetic control (dashed line) over all time periods, with a vertical dashed line marking the start of treatment. Good pre-treatment fit is a necessary (but not sufficient) condition for valid SC-CIC inference.

Value

Invisible. Called for its side effect of producing a plot.

`plot_distributions` *Plot Counterfactual Distribution*

Description

Plots the empirical CDFs of the four group-period cells used in the CIC estimator, illustrating the distributional transport.

Usage

```
plot_distributions(x, ...)
```

Arguments

`x`	An object of class `"cic"` or `"sc_cic"`.
`...`	Additional arguments (currently unused).

Value

Invisible. Called for its side effect of producing a plot.

`plot_qq` *Q-Q Plot of Treated vs Synthetic Control (Pre-treatment)*

Description

Produces a quantile-quantile plot comparing the pre-treatment distributions of the treated unit and the synthetic control. This assesses whether the synthetic control tracks the treated unit's *distributional* dynamics, not just its mean—a necessary condition for CIC validity. Points on the 45-degree line indicate identical distributions.

Usage

```
plot_qq(x, ...)
```

Arguments

`x`	An object of class `"sc_cic"` or `"cic"`.
`...`	Additional arguments passed to `plot`.

Value

Invisible. Called for its side effect of producing a plot.

`plot_qte` *Plot Quantile Treatment Effects*

Description

Plot Quantile Treatment Effects

Usage

```
plot_qte(x, probs = seq(0.05, 0.95, 0.05), ...)
```

Arguments

`x`	An object of class `"cic"` or `"sc_cic"`.
`probs`	Numeric vector of quantiles.
`...`	Additional arguments passed to `plot`.

Value

Invisible. Called for its side effect of producing a plot.

`quantile_te` *Compute Quantile Treatment Effects*

Description

Estimates quantile treatment effects from a CIC fit by comparing quantiles of the actual post-treatment treated distribution with quantiles of the counterfactual distribution.

Usage

```
quantile_te(x, probs = seq(0.05, 0.95, 0.05))
```

Arguments

`x`	An object of class `"cic"` or `"sc_cic"`.
`probs`	Numeric vector of quantiles at which to compute effects. Default is `seq(0.05, 0.95, 0.05)`.

Details

The quantile treatment effect at quantile q is:

$$\hat{\tau}_q = \hat{F}^{-1}_{Y^I,11}(q) - \hat{F}^{-1}_{Y^N,11}(q)$$

where $\hat{F}^{-1}_{Y^N,11}$ is the CIC counterfactual distribution.

Value

A data frame with columns `quantile`, `actual`, `counterfactual`, and `qte` (quantile treatment effect).

`sc_cic` *Synthetic Control Changes-in-Changes Estimator*

Description

Combines synthetic control methods with the Changes-in-Changes estimator. First constructs a synthetic control unit from donor units using elastic net regularization, then applies the CIC estimator using the synthetic control as the comparison group.

Usage

```
sc_cic(
  y_treated,
  y_donors,
  treatment_period,
  alpha = 1,
  boot = TRUE,
  boot_iters = 500L,
  seed = NULL
)
```

Arguments

`y_treated`	Numeric vector. Outcome for the treated unit across all time periods (pre and post).
`y_donors`	Numeric matrix. Outcomes for donor units, with rows as time periods (matching `y_treated`) and columns as donor units.
`treatment_period`	Integer. The index (row number) of the first treatment period. Periods 1 to `treatment_period - 1` are pre-treatment.
`alpha`	Elastic net mixing parameter. `alpha = 1` (default) is lasso; `alpha = 0` is ridge.
`boot`	Logical. Compute bootstrap standard errors. Default `TRUE`. The bootstrap re-estimates the elastic net weights in every iteration to account for first-stage estimation uncertainty.
`boot_iters`	Integer. Number of bootstrap iterations. Default 500.
`seed`	Integer or `NULL`. Random seed for reproducibility.

Details

The procedure works in two steps:

Step 1: Synthetic Control Construction. In the pre-treatment period, the treated unit's outcome is regressed on the donor units' outcomes using elastic net (via `cv.glmnet`). This yields a sparse set of weights that construct a synthetic control unit as a weighted combination of donors.

Step 2: CIC Estimation. The CIC estimator is applied with the synthetic control as the "control group" and the treated unit as the "treatment group."

Inference. Because the synthetic control is an estimated object, the analytic asymptotic variance of Athey and Imbens (2006) does not directly apply. Instead, `sc_cic` provides bootstrap standard errors that re-estimate the elastic net weights in each bootstrap iteration, thereby accounting for first-stage estimation uncertainty. The bootstrap resamples time periods (with replacement) within the pre-treatment and post-treatment windows separately, preserving the panel structure.

Value

An object of class `"sc_cic"` inheriting from `"cic"`, with components:

`tau`	The SC-CIC average treatment effect estimate.

`se`	Bootstrap standard error (if `boot = TRUE`). Note: analytic standard errors from Athey and Imbens (2006) Theorem 5.1 are *not* provided for `sc_cic`, because they do not account for first-stage synthetic control estimation uncertainty. Use `cic` directly if analytic SEs are needed for a pre-specified control group.
`z`	z-statistic (bootstrap-based).
`pval`	Two-sided p-value (bootstrap-based).
`boot_se`	Same as `se` (for compatibility with `cic`).
`tau_did`	The SC-DID estimate for comparison.
`sc_weights`	Named vector of synthetic control weights (including intercept).
`sc_fitted`	Synthetic control outcome across all time periods.
`donors_selected`	Names of donor units with nonzero weights.
`pre_fit_rmse`	Root mean squared error of pre-treatment fit.

References

Athey, S. and Imbens, G. W. (2006). Identification and Inference in Nonlinear Difference-in-Differences Models. *Econometrica*, 74(2), 431–497.

Abadie, A., Diamond, A., and Hainmueller, J. (2010). Synthetic Control Methods for Comparative Case Studies. *Journal of the American Statistical Association*, 105(490), 493–505.

Examples

```
# Basque Country example
if (requireNamespace("Synth", quietly = TRUE)) {
  data("basque", package = "Synth")
  gdp <- reshape(basque[, c("regionno", "year", "gdpcap")],
                 idvar = "year", timevar = "regionno", direction = "wide")
  y_treated <- gdp[, "gdpcap.17"]
  donors <- as.matrix(gdp[, grep("gdpcap\\.", names(gdp))])
  donors <- donors[, !colnames(donors) %in% c("gdpcap.17", "gdpcap.1")]
  valid <- complete.cases(y_treated, donors)
  result <- sc_cic(y_treated[valid], donors[valid, ],
                   treatment_period = 16, seed = 42)
  print(result)
}
```

`sc_weights` *Extract Synthetic Control Weights*

Description

Returns a data frame of donor weights from an SC-CIC fit, sorted by absolute weight. Useful for inspecting which donors contribute to the synthetic control.

Usage

```
sc_weights(x, nonzero_only = TRUE)
```

Arguments

`x`	An object of class `"sc_cic"`.
`nonzero_only`	Logical. If `TRUE` (default), only return donors with nonzero weights.

Value

A data frame with columns `donor` and `weight`.

`sensitivity_alpha` *Sensitivity Analysis for SC-CIC*

Description

Re-estimates the SC-CIC treatment effect over a grid of elastic net penalty parameters, showing sensitivity to the regularization choice.

Usage

```
sensitivity_alpha(
  y_treated,
  y_donors,
  treatment_period,
  alphas = seq(0, 1, 0.2),
  seed = 42
)
```

Arguments

`y_treated`	Numeric vector. Treated unit outcomes.
`y_donors`	Numeric matrix. Donor unit outcomes.
`treatment_period`	Integer. First treatment period index.
`alphas`	Numeric vector. Grid of alpha values to evaluate. Default is `seq(0, 1, 0.2)`.
`seed`	Integer or `NULL`. Random seed.

Value

A data frame with columns `alpha`, `tau_cic`, `tau_did`, `n_donors`, and `pre_rmse`.

`simulate_sccic` *Simulation Study for SC-CIC*

Description

Generates data under controlled DGPs and evaluates SC-CIC performance.

Usage

```
simulate_sccic(
  n_sims = 500,
  T_pre = 25,
  T_post = 15,
  J = 15,
  tau_true = 1,
  dgp = c("linear", "nonlinear", "sc_good", "sc_bad"),
  alpha = 1,
  boot_iters = 200,
  seed = 42,
  verbose = TRUE
)
```

Arguments

`n_sims`	Integer. Number of simulation replications.
`T_pre`	Integer. Number of pre-treatment periods.
`T_post`	Integer. Number of post-treatment periods.
`J`	Integer. Number of donor units.
`tau_true`	Numeric. True average treatment effect.
`dgp`	Character. Data generating process. See Details.
`alpha`	Elastic net mixing parameter for SC construction.
`boot_iters`	Integer. Bootstrap iterations per simulation.
`seed`	Integer. Random seed.
`verbose`	Logical. Print progress.

Details

Four DGPs are available, designed to test different aspects of SC-CIC:

DGP 1: `"linear"` **— Baseline.** Outcomes are linear in a common factor and unit-specific loadings. DID is correctly specified. CIC matches DID. SC fits well. Purpose: verify the method works in the easy case.

DGP 2: `"nonlinear"` **— CIC advantage.** Cross-sectional DGP (not SC). N observations per cell. Control and treated have different distributions of unobservables. The production function is nonlinear and changes over time. DID is biased due to the nonlinear distributional shift; CIC is correct. Purpose: demonstrate the advantage of CIC over DID. Note: this tests `cic()`, not `sc_cic()`.

DGP 3: `"sc_good"` **— SC with good distributional fit.** The treated unit is a true sparse combination of donors plus noise. SC recovers the weights well; the distributional dynamics are similar. Purpose: show SC-CIC works when SC fit is good.

DGP 4: `"sc_bad"` **— SC with mean-only fit.** The SC matches the treated mean, but donors have much lower variance than the treated unit. The distributional transport is wrong. Purpose: show SC-CIC fails when distributional assumptions are violated.

Value

A data frame with simulation results.

Examples

```
# Quick example (runs in seconds)
r <- simulate_sccic(n_sims = 2, dgp = "nonlinear", tau_true = 1, boot_iters = 5, verbose = FALSE)
summarize_simulation(r, tau_true = 1)


# Full simulation
r <- simulate_sccic(n_sims = 200, dgp = "nonlinear", tau_true = 1)
summarize_simulation(r, tau_true = 1)
```

`summarize_simulation` *Summarize simulation results*

Description

Summarize simulation results

Usage

```
summarize_simulation(results, tau_true)
```

Arguments

`results`	Data frame from `simulate_sccic`.
`tau_true`	True treatment effect.

Value

Prints summary statistics and returns them invisibly.

Index

check_data, 2
check_support, 3
cic, *2*, 3, *10*
cv.glmnet, *9*

loo_donors, 5

plot, *6, 7*
plot.sc_cic, 6
plot_distributions, 6
plot_qq, 7
plot_qte, 7

quantile_te, 8

sc_cic, 8
sc_weights, 10
sensitivity_alpha, 11
simulate_sccic, 12
summarize_simulation, 13