Package: bsynth (via r-universe)

October 29, 2026

Title Bayesian Synthetic Control

Version 1.0

Description Implements the Bayesian Synthetic Control method for causal inference in comparative case studies. This package provides tools for estimating treatment effects in settings with a single treated unit and multiple control units, allowing for uncertainty quantification and flexible modeling of time-varying effects. The methodology is based on the paper by Vives and Martinez (2022) <doi:10.48550/arXiv.2206.01779>.

License Apache License 2.0

URL https://github.com/google/bsynth, https://arxiv.org/abs/2206.01779

BugReports https://github.com/google/bsynth/issues

Encoding UTF-8

RoxygenNote 7.3.1

Biarch true

Depends R (>= 3.4.0)

Imports R6, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), cubelyr, dplyr, ggplot2, glue, magrittr, methods, purrr, rstan (>= 2.18.1), rstantools (>= 2.1.1), scales, tibble, tidyr, vizdraws (>= 1.1), stats, rlang

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

Suggests knitr, rmarkdown, roxygen2, gsynth, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation yes

Author Ignacio Martinez [aut, cre] (<https://orcid.org/0000-0002-3721-8172>), Jaume [aut]

Maintainer Ignacio Martinez <ignacio@martinez.fyi>

Config/pak/sysreqs make

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

Date/Publication 2024-06-26 02:42:48 UTC

RemoteUrl https://github.com/cran/bsynth

RemoteRef HEAD

RemoteSha 4e7203dc7099b50bed3dc719e87937b750cc773a

Contents

bsynth-package . . . 2
.get_par_long . . . 3
.get_plot_df . . . 3
.get_plot_df2 . . . 4
.get_synth_draws . . . 5
.get_synth_draws_predictor_match . . . 6
.get_synth_draws3d . . . 6
.makeWide . . . 7
.plot_tau . . . 8
bayesianFactor . . . 9
bayesianSynth . . . 12
time_tiles . . . 16

Index **17**

`bsynth-package` *The 'bsynth' package.*

Description

Provides causal inference with a Bayesian synthetic control method.

Author(s)

Maintainer: Ignacio Martinez <ignacio@martinez.fyi> (ORCID)

Authors:

- Jaume <vives@mit.edu>

References

Stan Development Team (2020). RStan: the R interface to Stan. R package version 2.21.2. https://mc-stan.org

See Also

Useful links:

- https://github.com/google/bsynth
- https://arxiv.org/abs/2206.01779
- Report bugs at https://github.com/google/bsynth/issues

`.get_par_long` *Get Parameter Estimates in Long Format*

Description

Helper function to get the long dataset of draws given a stan fit object.

Usage

```
.get_par_long(fit, par)
```

Arguments

`fit`	Stan object with the fitted model.
`par`	Variable to do the long table for.expand_more

Value

A tibble containing the parameter estimates in long format.

`.get_plot_df` *Returns Data Frame Ready for Plotting with Confidence Intervals*

Description

This function processes data frames containing synthetic and observed outcomes, calculates confidence intervals for the synthetic outcomes, and returns a combined data frame suitable for plotting the results.

Usage

```
.get_plot_df(y_synth_draws, pre_data, post_data, time, outcome, ci = 0.75)
```

Arguments

y_synth_draws	A data frame containing draws from the Stan fit object.
pre_data	A data frame with data before the intervention.
post_data	A data frame with data after the intervention.
time	The name of the time period variable (as a string).
outcome	The name of the outcome variable (as a string).
ci	The width of the credible confidence interval (default: 0.75).

Value

A data frame containing:

- time: The time period.
- outcome: The observed outcome.
- y_synth: The mean synthetic outcome.
- LB: The lower bound of the confidence interval for the synthetic outcome.
- UB: The upper bound of the confidence interval for the synthetic outcome.
- tau: The difference between the observed and synthetic outcomes.
- tau_LB: The lower bound of the confidence interval for tau.
- tau_UB: The upper bound of the confidence interval for tau.

.get_plot_df2 *Prepare Data Frame for Plotting with Multiple Treated Units*

Description

This function processes data for multiple treated units, calculating synthetic outcomes, confidence intervals, and treatment effects. It combines this information into a data frame suitable for plotting the results.

Usage

```
.get_plot_df2(y_synth_draws, data, treated_ids, id, time, outcome, ci = 0.75)
```

Arguments

y_synth_draws	A data frame containing synthetic outcome draws for each treated unit and time period.
data	A data frame with the original data, including outcomes for treated units.
treated_ids	A vector of identifiers for the treated units.
id	The name of the variable in data that identifies units (as a string).
time	The name of the time period variable (as a string).
outcome	The name of the outcome variable (as a string).
ci	The width of the credible confidence interval (default: 0.75).

Value

A data frame containing:

- `time`: The time period.
- `id`: The unit identifier (including "Average" for the average treatment effect).
- `outcome`: The observed outcome (for treated units).
- `y_synth`: The mean synthetic outcome (for treated units and the average).
- `LB`: The lower bound of the confidence interval for the synthetic outcome.
- `UB`: The upper bound of the confidence interval for the synthetic outcome.
- `tau`: The treatment effect (difference between observed and synthetic outcomes).
- `tau_LB`: The lower bound of the confidence interval for the treatment effect.
- `tau_UB`: The upper bound of the confidence interval for the treatment effect.

`.get_synth_draws` *Get Synthetic Draws in Tidy Format for Single Treated Unit*

Description

This internal helper function extracts synthetic draws from a Stan fit object, combines them with observed outcome data, and returns a tidy data frame suitable for further analysis or plotting. This function is specifically designed for scenarios with a single treated unit.

Usage

```
.get_synth_draws(fit, pre_data, post_data, time, outcome)
```

Arguments

`fit`	A Stan fit object containing the model results.
`pre_data`	A data frame with outcome data before the intervention.
`post_data`	A data frame with outcome data after the intervention.
`time`	The name of the time period variable (as a string).
`outcome`	The name of the outcome variable (as a string).

Value

A data frame containing:

- `draw`: The index of the synthetic draw.
- `time`: The time period.
- `y_synth`: The synthetic outcome for the given draw and time period.
- `outcome`: The observed outcome for the given time period.

.get_synth_draws_predictor_match

Get Synthetic Draws in Tidy Format for Single Treated Unit (Predictor Match Model)

Description

This internal helper function extracts synthetic draws from a Stan fit object generated by a predictor match model. It combines these draws with observed outcome data and returns a tidy data frame suitable for analysis or plotting. It specifically works with variable definitions from the predictor match model.

Usage

```
.get_synth_draws_predictor_match(fit, pre_data, post_data, time, outcome)
```

Arguments

`fit`	A Stan fit object containing the model results.
`pre_data`	A data frame with outcome data before the intervention.
`post_data`	A data frame with outcome data after the intervention.
`time`	The name of the time period variable (as a string).
`outcome`	The name of the outcome variable (as a string).

Value

A data frame containing:

- `draw`: The index of the synthetic draw.
- `time`: The time period.
- `y_synth`: The synthetic outcome for the given draw and time period.
- `outcome`: The observed outcome for the given time period.

.get_synth_draws3d

Get Synthetic Draws in Tidy Format for Multiple Treated Units (3D Array)

Description

This internal helper function extracts synthetic draws from a Stan fit object where the draws are stored in a 3D array. It handles multiple treated units and combines the draws with observed outcome data, returning a tidy data frame suitable for analysis or plotting.

Usage

```
.get_synth_draws3d(fit, data, id, treated_ids, time, outcome, intervention)
```

Arguments

fit	A Stan fit object containing the model results.
data	A data frame with the input data, including outcome, time, and unit identifier.
id	The name of the variable in data that identifies units (as a string).
treated_ids	A vector of identifiers for the treated units.
time	The name of the time period variable (as a string).
outcome	The name of the outcome variable (as a string).
intervention	The name of the variable in data that indicates the intervention time (as a string).

Value

A data frame containing:

- draw: The index of the synthetic draw.
- id: The identifier of the treated unit.
- time: The time period.
- y_hat: The synthetic outcome for the given draw, unit, and time period.

.makeWide *Convert Data to Wide Format*

Description

This internal helper function transforms data from a long format, where each row represents an observation for a specific unit and time, to a wide format, where each row represents a time period and each column represents a unit's outcome. It specifically focuses on separating treated and untreated units.

Usage

```
.makeWide(data, id, time, outcome, treatment)
```

Arguments

data	A data frame containing the input data.
id	The name of the variable in data that identifies units (as a string).
time	The name of the time period variable (as a string).
outcome	The name of the outcome variable (as a string).
treatment	The name of the variable in data that indicates treatment status (as a string).

Value

A data frame in wide format, where each row corresponds to a time period, and columns include the time variable, the treatment indicator, and the outcome values for each treated unit and all untreated units.

`.plot_tau` *Plot Treatment Effect Estimate*

Description

This internal helper function creates a plot to visualize the estimated treatment effect over time. It allows for faceting by a specified variable and optional subsetting of units to include in the plot.

Usage

```
.plot_tau(data, x, y, ymin, ymax, xintercept, facet, id, subset = NULL)
```

Arguments

`data`	A data frame containing the data to be plotted.
`x`	The name of the x-axis variable (typically the time period) (as a string).
`y`	The name of the y-axis variable (typically the treatment effect) (as a string).
`ymin`	The name of the variable containing the lower bound of the confidence interval (as a string).
`ymax`	The name of the variable containing the upper bound of the confidence interval (as a string).
`xintercept`	The time point of the intervention to be marked with a vertical dashed line.
`facet`	(Optional) The name of the variable to facet the plot by (as a string).
`id`	The name of the variable identifying the units (as a string).
`subset`	(Optional) A vector specifying a subset of units to include in the plot. If NULL, all units are included.

Value

A ggplot object displaying the treatment effect plot.

`bayesianFactor` *Create a Bayesian Synthetic Control Object Using Panel Data*

Description

A Bayesian Factor Model has raw data and draws from the posterior distribution. This is represented by an R6 Class.

Code and theory based on Pinkney 2021.

public methods:

- `initialize()` initializes the variables and model parameters
- `fit()` fits the stan model and returns a fit object
- `updateWidth` updates the width of the credible interval
- `placeboPlot` generates a counterfactual placebo plot
- `effectPlot` returns a plot of the treatment effect over time
- `summarizeLift`returns descriptive statistics of the lift estimate
- `biasDraws` returns a plot of the relative bias in a LFM
- `liftDraws` returns a plot of the posterior lift distribution
- `liftBias` returns a plot of the relative bias given a lift offset

Value

vizdraws object with the relative bias with offset.

Active bindings

`timeTiles` ggplot2 object that shows when the intervention happened.

`plotData` tibble with the observed outcome and the counterfactual data.

`interventionTime` returns the intervention time period.

`synthetic` ggplot2 object that shows the observed and counterfactual outcomes over time.

Methods

Public methods:

- `bayesianFactor$new()`
- `bayesianFactor$fit()`
- `bayesianFactor$updateWidth()`
- `bayesianFactor$summarizeLift()`
- `bayesianFactor$effectPlot()`
- `bayesianFactor$liftDraws()`
- `bayesianFactor$liftBias()`
- `bayesianFactor$biasDraws()`

- bayesianFactor$clone()

Method `new()`**:** Create a new bayesianFactor object.

Usage:

```
bayesianFactor$new(
  data,
  time,
  id,
  treated,
  outcome,
  ci_width = 0.75,
  covariates
)
```

Arguments:

`data` Long data.frame object with fields outcome, time, id, and treatment indicator.

`time` Name of the variable in the data frame that

`id` Name of the variable in the data frame that identifies the units (e.g. country, region etc).

`treated` Name of the variable in the data frame that contains the treatment assignment of the intervention.

`outcome` Name of the outcome variable.

`ci_width` Credible interval's width. This number is in the (0,1) interval.

`covariates` Dataframe with a column for id and the other columns Defaults to NULL if no covariates should be included in the model.

Details: params described in the data structure section of the documentation of the R6 class at the top of the file.

Returns: A new `bayesianFactor` object.

Method `fit()`**:** Fit Stan model.

Usage:

```
bayesianFactor$fit(L = 8, ...)
```

Arguments:

`L` Number of factors.

`...` other arguments passed to `rstan::sampling()`.

Method `updateWidth()`**:** Update the width of the credible interval.

Usage:

```
bayesianFactor$updateWidth(ci_width = 0.75)
```

Arguments:

`ci_width` New width for the credible interval. This number should be in the (0,1) interval.

Method `summarizeLift()`**:** summarizeLift returns descriptive statistics of the lift estimate.

Usage:

```
bayesianFactor$summarizeLift()
```

Method `effectPlot()`**:** effectPlot returns ggplot2 object that shows the effect of the intervention over time.

Usage:

```
bayesianFactor$effectPlot()
```

Method `liftDraws()`**:** Plots lift.

Usage:

```
bayesianFactor$liftDraws(from, to, ...)
```

Arguments:

`from` First period to consider when calculating lift. If infinite, set to the time of the intervention.

`to` Last period to consider when calculating lift. If infinite, set to the last period.

`...` other arguments passed to vizdraws::vizdraws().

Returns: vizdraws object with the posterior distribution of the lift.

Method `liftBias()`**:** Plot bias magnitude in terms of lift for period (firstT, lastT)

Usage:

```
bayesianFactor$liftBias(firstT, lastT, offset, ...)
```

Arguments:

`firstT` Start of the time period to compute relative bias over. Must be after the intervention.

`lastT` End of the time period to compute relative bias over. Must be after the intervention. over. They must be after the intervention.

`offset` Target lift %.

`...` other arguments passed to vizdraws::vizdraws().

Method `biasDraws()`**:** Plots relative upper bias / tau for a time period (firstT, lastT).

Usage:

```
bayesianFactor$biasDraws(small_bias = 0.3, firstT, lastT)
```

Arguments:

`small_bias` Threshold value for considering the bias "small".

`firstT`, `lastT` Time periods to compute relative bias over, they must after the intervention.

Returns: vizdraw object with the posterior distribution of relative bias. Bias is scaled by the time periods.

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
bayesianFactor$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

bayesianSynth *Create a Bayesian Synthetic Control Object Using Panel Data*

Description

A Bayesian Synthetic Control has raw data and draws from the posterior distribution. This is represented by an R6 Class.

public methods:

- `initialize()` initializes the variables and model parameters
- `fit()` fits the stan model and returns a fit object
- `updateWidth` updates the width of the credible interval
- `placeboPlot` generates a counterfactual placebo plot
- `effectPlot` returns a plot of the treatment effect over time
- `summarizeLift`returns descriptive statistics of the lift estimate
- `biasDraws` returns a plot of the relative bias in a LFM
- `liftDraws` returns a plot of the posterior lift distribution
- `liftBias` returns a plot of the relative bias given a lift offset Data structure:

Value

vizdraws object with the relative bias with offset.

Active bindings

`timeTiles` ggplot2 object that shows when the intervention happened.

`plotData` returns tibble with the observed outcome and the counterfactual data.

`interventionTime` returns intervention time period (e.g., year) in which the treatment occurred.

`synthetic` returns ggplot2 object that shows the observed and counterfactual outcomes over time.

`checks` returns MCMC checks.

`lift` draws from the posterior distribution of the lift.

Methods

Public methods:

- `bayesianSynth$new()`
- `bayesianSynth$fit()`
- `bayesianSynth$updateWidth()`
- `bayesianSynth$summarizeLift()`
- `bayesianSynth$effectPlot()`
- `bayesianSynth$placeboPlot()`
- `bayesianSynth$biasDraws()`

- bayesianSynth$liftDraws()
- bayesianSynth$liftBias()
- bayesianSynth$weightDraws()
- bayesianSynth$weightCorr()
- bayesianSynth$clone()

Method `new()`**:** Create a new bayesianSynth object.

Usage:

```
bayesianSynth$new(
  data,
  time,
  id,
  treated,
  outcome,
  ci_width = 0.75,
  gp = FALSE,
  covariates = NULL,
  predictor_match = FALSE,
  predictor_match_covariates0 = NULL,
  predictor_match_covariates1 = NULL,
  vs = NULL
)
```

Arguments:

`data` Long data.frame object with fields outcome, time, id, and treatment indicator.

`time` Name of the variable in the data frame that identifies the time period (e.g. year, month, week etc).

`id` Name of the variable in the data frame that identifies the units (e.g. country, region etc).

`treated` Name of the variable in the data frame that contains the treatment assignment of the intervention.

`outcome` Name of the outcome variable.

`ci_width` Credible interval's width. This number is in the (0,1) interval.

`gp` Logical that indicates whether or not to include a Gaussian Process as part of the model.

`covariates` Data.frame with time dependent covariates for for each unit and time field. Defaults to NULL if no covariates should be included in the model.

`predictor_match` Logical that indicates whether or not to run the matching version of the Bayesian Synthetic Control. This option can not be used with gp, covariates or multiple treated units.

`predictor_match_covariates0` data.frame with time independent covariates on each row and column indicating the control unit names (dim k x J+1).

`predictor_match_covariates1` Vector with time independent covariates for the treated unit (dim k x 1).

`vs` Vector of weights for the importance of the predictors used in creating the synthetic control. Defaults to equal weight for all predictors.

Returns: A new `bayesianSynth` object.

Method `fit()`**:** Fit Stan model.

Usage:

```
bayesianSynth$fit(...)
```

Arguments:

`...` other arguments passed to `rstan::sampling()`.

Method `updateWidth()`**:** Update the width of the credible interval.

Usage:

```
bayesianSynth$updateWidth(ci_width = 0.75)
```

Arguments:

`ci_width` New width for the credible interval. This number should be in the (0,1) interval.

Method `summarizeLift()`**:** returns descriptive statistics of the lift estimate.

Usage:

```
bayesianSynth$summarizeLift()
```

Method `effectPlot()`**:** effect ggplot2 object that shows the effect of the intervention over time.

Usage:

```
bayesianSynth$effectPlot(facet = TRUE, subset = NULL)
```

Arguments:

`facet` Boolean that is TRUE if we want to divide the plot for each unit.

`subset` Set of units to use in the effect plot.

Method `placeboPlot()`**:** Plot placebo intervention.

Usage:

```
bayesianSynth$placeboPlot(periods, ...)
```

Arguments:

`periods` Positive number of periods for the placebo intervention.

`...` other arguments passed to `rstan::sampling()`.

Returns: ggplot2 object for placebo treatment effect.

Method `biasDraws()`**:** Plots relative upper bias / tau for a time period (firstT, lastT).

Usage:

```
bayesianSynth$biasDraws(small_bias = 0.3, firstT, lastT)
```

Arguments:

`small_bias` Threshold value for considering the bias "small".

`firstT` Start of the time period to compute relative bias over. Must be after the intervention.

`lastT` End of the time period to compute relative bias over. Must be after the intervention.

Returns: vizdraw object with the posterior distribution of relative bias. Bias is scaled by the time periods.

Method `liftDraws()`**:** Plots lift.

Usage:

```
bayesianSynth$liftDraws(from, to, ...)
```

Arguments:

`from` First period to consider when calculating lift. If infinite, set to the time of the intervention.

`to` Last period to consider when calculating lift. If infinite, set to the last period.

`...` other arguments passed to vizdraws::vizdraws().

Returns: vizdraws object with the posterior distribution of the lift.

Method `liftBias()`**:** Plot Bias magnitude in terms of lift for period (firstT, lastT) pre_MADs / y0 relative to lift thresholds.

Usage:

```
bayesianSynth$liftBias(firstT, lastT, offset, ...)
```

Arguments:

`firstT` start of the time period to compute relative bias over. They must be after the intervention.

`lastT` end of the Time period to compute relative bias over. They must be after the intervention.

`offset` Target lift %.

`...` other arguments passed to vizdraws::vizdraws().

Method `weightDraws()`**:** Plot implicit weight distribution across draws.

Usage:

```
bayesianSynth$weightDraws()
```

Returns: ggplot object with weight distribution per unit.

Method `weightCorr()`**:** Plots correlations between weights across draws.

Usage:

```
bayesianSynth$weightCorr()
```

Returns: ggplot heatmap object with correlations.

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
bayesianSynth$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`time_tiles` *Time Tiles Plot of Intervention Impact*

Description

This function creates a time tiles plot visualizing when and which units are affected by an intervention. Each tile represents a unit at a specific time point, with the color indicating the treatment status.

Usage

```
time_tiles(data, time, id, status)
```

Arguments

`data`	A data frame containing the input data.
`time`	The name of the time period variable (as a string).
`id`	The name of the unit identifier variable (as a string).
`status`	The name of the variable that identifies the treatment status (as a string).

Value

A ggplot object displaying the time tiles plot.

Index

.get_par_long, 3
.get_plot_df, 3
.get_plot_df2, 4
.get_synth_draws, 5
.get_synth_draws3d, 6
.get_synth_draws_predictor_match, 6
.makeWide, 7
.plot_tau, 8

bayesianFactor, 9
bayesianSynth, 12
bsynth *(*bsynth-package*)*, 2
bsynth-package, 2

rstan::sampling(), *10*, *14*

time_tiles, 16