Package: PSPI (via r-universe)

November 1, 2026

Type Package

Title Propensity Score Predictive Inference for Generalizability and Transportability

Version 1.4

Date 2026-09-30

Author Jungang Zou [aut, cre], Qixuan Chen [aut], Joseph Schwartz [aut], Nathalie Moise [aut], Roderick Little [aut], Robert McCulloch [ctb], Rodney Sparapani [ctb], Charles Spanbauer [ctb], Robert Gramacy [ctb], Jean-Sebastien Roy [ctb]

Maintainer Jungang Zou <jungang.zou@gmail.com>

Description Provides a suite of Propensity Score Predictive Inference (PSPI) methods to generalize treatment effects in trials to target populations. The package includes an existing model Bayesian Causal Forest (BCF) and four PSPI models (BCF-PS, FullBART, SplineBART, DSplineBART). These methods leverage Bayesian Additive Regression Trees (BART) to adjust for high-dimensional covariates and nonlinear associations, while SplineBART and DSplineBART further use propensity score based splines to address covariate shift between trial data and target population.

License GPL-2

Encoding UTF-8

Depends R (>= 4.1.0)

Imports Rcpp, arm, dplyr, mvtnorm, stringr, stats, nnet, methods

LinkingTo Rcpp, RcppArmadillo, RcppDist, RcppProgress, pg

RoxygenNote 7.3.3

SystemRequirements GNU make

Suggests knitr, rmarkdown, mitml, ggplot2

NeedsCompilation yes

Config/pak/sysreqs make

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

Date/Publication 2026-10-01 23:58:11 UTC

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

RemoteRef HEAD

RemoteSha 3e701a4b1794b7e796fc40618f6c7e92c7014dc4

Contents

PSPI-package 2
in_trial 3
invgumbel 3
PSPI_fit 4
PSPI_generalizability 6
PSPI_predict 9
sim_generalizability 11
sim_trans 12
summary.PSPI_generalizability 14

Index **16**

`PSPI-package` *Propensity Score Predictive Inference for Generalizability and Transportability*

Description

Provides a suite of Propensity Score Predictive Inference (PSPI) methods to generalize treatment effects in trials to target populations. The package includes an existing model Bayesian Causal Forest (BCF) and four PSPI models (BCF-PS, FullBART, SplineBART, DSplineBART). These methods leverage Bayesian Additive Regression Trees (BART) to adjust for high-dimensional covariates and nonlinear associations, while SplineBART and DSplineBART further use propensity score based splines to address covariate shift between trial data and target population.

Details

PSPI provides Bayesian methods for generalizing treatment effects from clinical trials to target populations. It implements five models-`BCF`, `BCF_P`, `FullBART`, `SplineBART`, and `DSplineBART`-built on Bayesian Additive Regression Trees (BART). Spline-based variants (`SplineBART` and `DSplineBART`) use propensity score transformations and spline terms to handle covariate shift between datasets. Core computations rely on efficient MCMC routines implemented in C++.

This package modifies and extends C++ code originally derived from the BART3 package, developed by Rodney Sparapani, which is licensed under the GNU General Public License version 2 (GPL-2).

The modified code is redistributed in accordance with the GPL-2 license. For more details on the modifications, see the package's documentation.

References

BART3 package: https://github.com/rsparapa/bnptools/tree/master, originally developed by Rodney Sparapani.

`in_trial` *Identify which population rows are included in a trial subset*

Description

Returns a logical vector indicating, for each row in `population`, whether that row is included in `trial`, where `trial` is assumed to be a (row) subset of `population`.

Usage

```
in_trial(population, trial)
```

Arguments

`population`	A data frame or a matrix representing the covariates of population
`trial`	A data frame or a matrix representing the covariates of trial dataset, assumed to be a subset of `population` by rows.

Value

A logical vector of length `nrow(population)`. The `i`-th element is `TRUE` if the `i`-th row of `population` appears in `trial`, and `FALSE` otherwise.

Examples

```
pop <- data.frame(id = 1:5, x = c(0.1, 0.2, 0.3, 0.4, 0.5))
tri <- pop[c(2, 4), ]
in_trial(pop, tri)
```

`invgumbel` *Gumbel quantile transform*

Description

Computes the inverse cumulative distribution function (quantile function) of the standard Gumbel distribution via the probability integral transform:

$$Q(p) = -\log\{-\log(p)\}, \quad 0 < p < 1.$$

Usage

```
invgumbel(x)
```

Arguments

x	Numeric vector or matrix of probabilities in $(0, 1)$.

Value

A numeric vector or matrix of the same dimensions as x, containing $-\log\{-\log(x)\}$.

PSPI_fit *Fit a PSPI Model (Without Population Data)*

Description

Fits a PSPI model using trial data only, without requiring population-level covariates. The returned fit object can be saved with `saveRDS()` and transferred to a remote server where population data resides, then used with `PSPI_predict` to obtain population-level treatment effect estimates.

This two-stage workflow supports federated analysis where trial data and population data cannot be co-located.

Usage

```
PSPI_fit(
  X,
  Y,
  A,
  pi,
  model,
  transformation = "InvGumbel",
  nburn = 4000,
  npost = 4000,
  n_knots = NULL,
  order = 3,
  ntrees_s = 200,
  sparse = FALSE,
  augment = FALSE,
  verbose = FALSE,
  seed = NULL
)
```

Arguments

X	Matrix of covariates for the trial data.
Y	Numeric vector or logical vector of observed outcomes in the trial.
A	Integer vector of treatment assignments (0, 1, ..., J-1).
pi	Numeric matrix (n x J) of propensity scores for trial participants.
model	Character string specifying which PSPI model to use. Options: "BCF", "BCF_P", "FullBART", "SplineBART", "DSplineBART" (alias "MSplineBART").
transformation	Character string for propensity score transformation. Options: "Identity", "Logit", "Cloglog", "InvGumbel" (default).
nburn	Number of burn-in iterations (default = 4000).
npost	Number of posterior iterations saved (default = 4000).
n_knots	Integer vector of spline knots per arm. If NULL, chosen automatically.
order	Order of spline basis functions (default = 3).
ntrees_s	Number of trees for each BART component (default = 200).
sparse	Logical; sparse Dirichlet prior (default = FALSE).
augment	Logical; data augmentation for variable selection (default = FALSE).
verbose	Logical; print progress (default = FALSE).
seed	Optional random seed.

Value

An object of class "PSPI_fit" containing serialized BART tree structures, spline parameters, and metadata. Can be saved with saveRDS() and used with PSPI_predict.

See Also

PSPI_predict, PSPI_generalizability

Examples

```
sim <- sim_generalizability(scenario = "linear", n_trial = 60)
ps_trial <- sim$population$ps[sim$population$selected]

fit <- PSPI_fit(
  X = as.matrix(sim$trials[, paste0("X", 1:10)]),
  Y = sim$trials$Y,
  A = sim$trials$A,
  pi = cbind(ps_trial, ps_trial),
  model = "SplineBART",
  nburn = 1, npost = 1
)
```

`PSPI_generalizability` *Propensity Scores Predictive Inference for Generalizability and Transportability*

Description

This is the main function of the **PSPI** package. It runs Bayesian models that generalize findings from a clinical trial to a target population, estimating the average treatment effects and potential outcomes. Propensity scores of trial participation play the central role for generalizability analysis. When covariate shift is an issue, we recommend PSPI-SplineBART and PSPI-DSplineBART, which leveraging Bayesian Additive Regression Trees (BART) to model high-dimensional covariates, and propensity scores based splines to extrapolate smoothly.

Users provide trial data (covariates, outcomes, treatment, and propensity scores) along with population-level covariates and propensity scores. Propensity scores can be the true values or estimated from some models. The function then performs Monte Carlo Markov chain (MCMC) for the posterior inference.

Usage

```
PSPI_generalizability(
  X,
  Y,
  A,
  pi,
  X_pop,
  A_pop = NULL,
  pi_pop,
  model,
  transformation = "InvGumbel",
  restrict_covariates = NULL,
  nburn = 4000,
  npost = 4000,
  n_knots = NULL,
  order = 3,
  ntrees_s = 200,
  sparse = FALSE,
  augment = FALSE,
  verbose = FALSE,
  seed = NULL,
  pate_bootstrap = TRUE
)
```

Arguments

`X`	Matrix of covariates for the trial data.
`Y`	Numeric vector or logical vector of observed outcomes in the trial. Numeric vector is for continuous variable, while logical vector is for binary outcome.

A	Integer vector of treatment assignments. Values should be 0, 1, ..., J-1 where 0 is the reference (control) arm and J is the total number of treatment arms.
pi	Numeric matrix ($n \times J$) of propensity scores for trial participants. Column j contains the propensity score associated with arm j. In a standard analysis with a single participation propensity score $\hat{\pi} = \Pr(S = 1 \mid X)$, the same score can be supplied in all columns. The number of columns must equal length(unique(A)).
X_pop	Matrix of covariates for the target population data.
A_pop	Integer vector of treatment assignments for population data (default = NULL).
pi_pop	Numeric matrix ($N \times J$) of propensity scores for the target population. Must have the same number of columns as pi.
model	Character string specifying which PSPI model to use (see Details).
transformation	Character string indicating the transformation applied to the propensity scores. Options are "Identity", "Logit", "Cloglog", or "InvGumbel" (default).
restrict_covariates	Restrict population data within support of trial samples. This parameter specifies the names of covariates to restrict (default = NULL).
nburn	Number of burn-in iterations (default = 4000).
npost	Number of posterior iterations saved after burn-in (default = 4000).
n_knots	Integer vector specifying the number of spline knots per arm. Length $J - 1$ for "SplineBART" or J for "DSplineBART". If NULL, defaults are chosen automatically.
order	Order of spline basis functions (default = 3).
ntrees_s	Number of trees used for the BART component (default = 200).
sparse	Whether to perform variable selection based on a sparse Dirichlet prior rather than simply uniform; see Linero 2016.
augment	Whether data augmentation is to be performed in sparse variable selection; see Linero 2016.
verbose	Logical; if TRUE, prints progress messages.
seed	Optional random seed for reproducibility.
pate_bootstrap	Logical (default TRUE). If TRUE, after MCMC the function attaches a [npost, J] matrix post_outcome_PATE containing Bayesian-bootstrap (Dirichlet re-weighting) draws of the target-population mean $E[Y(a)]$ for each treatment arm. This propagates the uncertainty of the empirical covariate distribution $\widehat{F}_X$ into the posterior, giving population (PATE) intervals in addition to the standard mixed-ATE (MATE) intervals obtained by averaging post_outcome[, , j] uniformly across the target sample (Li, Ding & Mealli, 2022). Set FALSE to skip and save memory.

Details

Model choices

The model argument selects the type of PSPI model to be fitted:

- "BCF" – Bayesian Causal Forests (Hahn et al., 2020).
- "BCF_P" – BCF with the propensity score as an additional predictor.
- "FullBART" – Uses three BARTs to estimate treatment effects.
- "SplineBART" – Incorporates a natural cubic spline for heterogeneous treatment effects.
- "DSplineBART" (alias "MSplineBART") – Adds another natural cubic spline for the prognostic score.

Multi-arm treatment support

All models support $J \geq 2$ treatment arms. The model structure is:

$$Y_i = h(X_i, \hat{\pi}_{i,0}) + \sum_{j=1}^{J-1} I(A_i = j) \cdot s_j(X_i, \hat{\pi}_{i,j}) + \epsilon_i$$

where arm 0 is the reference (control) group, $h(\cdot)$ is the prognostic function, and each $s_j(\cdot)$ captures the heterogeneous treatment effect for arm j relative to the reference.

Propensity score transformations

Since splines are sensitive to scales of predictor, robust transformation is needed. The propensity scores (pi for trial, pi_pop for population) can be optionally transformed before modeling using one of the following:

- "Identity" – uses the raw propensity scores directly (no transformation).
- "Logit" – applies the logit transform: $g(p) = \log(p/(1-p))$.
- "Cloglog" – complementary log–log transform: $g(p) = \log(-\log(1-p))$.
- "InvGumbel" – inverse Gumbel transform: $g(p) = -\log(-\log(p))$. Default choice.

Users can experiment with different transformations to assess model sensitivity.

Spline settings

Spline-based models ("SplineBART" and "DSplineBART") allow flexible extrapolation to address covariate shift. The number and order of spline basis functions can be customized through the following parameters:

- n_knots_inter, order_inter: number and order of spline knots for treatment-interaction effects. Available for both SplineBART and DSplineBART.
- n_knots_main, order_main: number and order of spline knots for main effects. Available only for DSplineBART.

If any of these are left as NULL, default values are chosen automatically based on the cube root of the sample size (ensuring a reasonable smoothness level).

Value

A list containing posterior samples and model summaries:

post_outcome A three-dimensional array of dimension [npost, N, J]. Entry [m, i, j] is the m-th posterior draw of the potential outcome for population unit i under treatment arm j. Population-level potential outcome means are obtained by averaging over units: rowMeans(post_outcome[,,j]).

`post_interaction` A three-dimensional array of dimension `[npost, N, J-1]`. Entry `[m, i, j]` is the m-th posterior draw of the treatment effect for unit i comparing arm j to the reference arm (arm 0). The population average treatment effect for arm j is `rowMeans(post_interaction[,,j])`.

`post_outcome_train` Matrix `[npost, n]` of fitted values on the trial sample.

`post_sigma` Vector of length `npost` containing posterior draws of the residual standard deviation.

`post_beta` (SplineBART and DSplineBART only) List of matrices containing posterior draws of the natural cubic spline coefficients, one matrix per spline component.

`post_gamma2` (SplineBART and DSplineBART only) Matrix of posterior draws of the spline shrinkage parameters.

Note

This function utilizes modified C++ code originally derived from the BART3 package (Bayesian Additive Regression Trees). The original package was developed by Rodney Sparapani and is licensed under GPL-2. Modifications were made by Jungang Zou, 2024. For more information about the original BART3 package, see: https://github.com/rsparapa/bnptools/tree/master/BART3

Examples

```
# Example with simulated data
sim <- sim_trans(scenario = "linear", n_trial = 60)

fit <- PSPI_generalizability(
  X = as.matrix(sim$trials[, paste0("X", 1:10)]),
  Y = sim$trials$Y,
  A = sim$trials$A,
  pi = cbind(sim$trials$ps_trial, sim$trials$ps_trial),
  X_pop = as.matrix(sim$population[, paste0("X", 1:10)]),
  pi_pop = cbind(sim$population$ps_target, sim$population$ps_target),
  model = "SplineBART",
  transformation = "InvGumbel",
  #restrict_covariates = c("X1", "X2"),
  verbose = FALSE,
  nburn = 1, npost = 1
)

str(fit)
```

PSPI_predict *Predict from a Fitted PSPI Model on Population Data*

Description

Generates posterior predictions of potential outcomes and treatment effects for a target population, using a fitted PSPI model object produced by `PSPI_fit`. This function only requires population-level covariates and propensity scores — no trial data is needed.

Usage

```
PSPI_predict(
  fit,
  X_pop,
  pi_pop,
  restrict_covariates = NULL,
  X_trial = NULL,
  verbose = FALSE,
  pate_bootstrap = TRUE
)
```

Arguments

fit	An object of class "PSPI_fit" returned by PSPI_fit, or loaded via readRDS().
X_pop	Numeric matrix of covariates for the target population (N x p). Must have the same columns (same variables, same order) as the trial X used in PSPI_fit.
pi_pop	Numeric matrix (N x J) of propensity scores for the target population. Must have the same number of columns as the pi used in PSPI_fit.
restrict_covariates	Character vector of covariate names to restrict population data to the support of trial samples (default = NULL). Requires column names on X_pop.
X_trial	Optional trial covariate matrix, needed only when restrict_covariates is not NULL, to determine the support.
verbose	Logical; print progress (default = FALSE).
pate_bootstrap	Logical (default TRUE). If TRUE, attaches a [npost, J] matrix post_outcome_PATE containing Bayesian-bootstrap (Dirichlet re-weighting) draws of the target-population mean $E[Y(a)]$ for each arm, propagating the uncertainty of the empirical covariate distribution $\widehat{F}_X$ into the posterior (population-ATE estimand of Li, Ding & Mealli, 2022). Set FALSE to skip.

Value

A list containing:

post_outcome Array [npost, N, J] of posterior draws of potential outcomes.

post_interaction Array [npost, N, J-1] of posterior draws of treatment effects relative to arm 0.

See Also

PSPI_fit, PSPI_generalizability

Examples

```
## Not run:
fit <- readRDS("pspi_fit.rds")
result <- PSPI_predict(fit, X_pop, pi_pop = cbind(ps_pop, ps_pop))
```

```
# Population ATE for arm 1 vs arm 0
ate <- rowMeans(result$post_interaction[,,1])
quantile(ate, c(0.025, 0.5, 0.975))

## End(Not run)
```

`sim_generalizability` *Simulate a superpopulation and a randomized trial under PSPI generalizability scenarios*

Description

Generates a superpopulation of 100,000 individuals (treated as the target population) with seven continuous and three binary covariates, constructs potential outcomes `Y1` and `Y0` according to the chosen scenario, and returns: (i) a target sample of size `n_sample` drawn as a simple random sample from the target population; (ii) a randomized trial of size `n_trial` drawn as a selective sample from the target sample via a logistic participation model; and (iii) the true estimands at both the *mixed-ATE (MATE)* level (averaged over the target sample) and the *population-ATE (PATE)* level (averaged over the 100K superpopulation), following the estimand taxonomy in Li, Ding & Mealli (2022).

Treatment assignment A is drawn independently at the superpopulation level with probability `prop`, representing marginal randomization. Optional diagnostic plots visualize covariate and outcome overlap between the target sample and the trial.

Usage

```
sim_generalizability(
  n_trial = 200,
  n_sample = 1000,
  scenario = "linear",
  seed = NULL,
  prop = 0.5,
  plot = FALSE
)
```

Arguments

`n_trial`	Integer. Trial sample size drawn by weighted sampling from the target sample. Must satisfy `n_trial <= n_sample`.
`n_sample`	Integer. Target sample size drawn as a simple random sample from the target population. Defaults to `1000`.
`scenario`	Character. One of `"linear"`, `"linear+covariate shift"`, `"nonlinear"`, `"nonlinear+covariate shift"`. Controls both the outcome-generating process and the participation model.
`seed`	Optional integer seed for reproducibility. If `NULL`, the current RNG state is used.

prop — Numeric in [0,1]. Randomization probability $\Pr(A = 1)$ within the superpopulation.

plot — Logical. If TRUE, constructs and prints two diagnostic figures comparing covariate and outcome distributions between the target sample and the trial. The plot objects are also returned inside the output list under $plots. Defaults to FALSE.

Value

A named list with three (or four, when plot = TRUE) elements:

population Data frame of size n_sample (the target sample). Columns: X1:X10 (covariates), A (treatment indicator), Y1, Y0 (potential outcomes), ps (oracle trial participation propensity score), and selected (logical; TRUE if the row is also in the trial).

trials Data frame of size n_trial (the selective trial sample drawn from population). Columns: X1:X10, A, ps, and the observed outcome Y = A*Y1 + (1-A)*Y0.

true_values Named list of six true estimands: true_ATE_MATE, true_Y1_MATE, true_Y0_MATE (averaged over the *target sample*, size n_sample); true_ATE_PATE, true_Y1_PATE, true_Y0_PATE (averaged over the *target population*, here the superpopulation).

plots (Only when plot = TRUE.) Named list with covariates and outcomes ggplot objects.

Examples

```
set.seed(2025)
sim <- sim_generalizability(n_trial = 200, n_sample = 1000,
                            scenario = "nonlinear+covariate shift", prop = 0.5,
                            plot = TRUE)
str(sim$population)
table(sim$trials$A)               # trial treatment allocation
mean(sim$population$selected)      # fraction of target sample in trial
sim$true_values                    # true target-population ATE, E[Y1], E[Y0]

# Smaller trial, linear scenario with covariate shift
sim2 <- sim_generalizability(n_trial = 60, n_sample = 1000,
                             scenario = "linear+covariate shift",
                             seed = 1, prop = 0.6)
nrow(sim2$trials)                  # 60
```

---

sim_trans *Simulate a superpopulation, a randomized trial, and a target sample under PSPI transportability scenarios*

Description

Generates a superpopulation of 100,000 individuals with seven continuous and three binary covariates, constructs potential outcomes Y1 and Y0 according to the chosen scenario, and returns a randomized trial, a target population, a target sample, and the true estimands at both the *mixed-ATE*

(MATE) level (averaged over the target sample) and the *population-ATE (PATE)* level (averaged over the target population), following the estimand taxonomy in Li, Ding & Mealli (2022). Optional diagnostic plots visualize covariate and outcome overlap between the superpopulation, the trial, and the target sample.

Usage

```
sim_trans(
  n_trial = 200,
  n_target_pop = 5000,
  n_sample = 1000,
  scenario = "linear",
  seed = NULL,
  prop = 0.5,
  target_shift_coef = -0.3,
  plot = FALSE
)
```

Arguments

`n_trial`	Integer. Trial sample size drawn from the superpopulation.
`n_target_pop`	Integer. Target population size drawn from the non-randomized portion of the superpopulation. Defaults to `5000`.
`n_sample`	Integer. Target sample size drawn as a simple random sample from the target population. Must satisfy `n_sample <= n_target_pop`. Defaults to `1000`.
`scenario`	Character. One of `"linear"`, `"linear+covariate shift"`, `"nonlinear"`, `"nonlinear+covariate shift"`.
`seed`	Optional integer seed for reproducibility.
`prop`	Numeric in `[0,1]`. Randomization probability $\Pr(A = 1)$ within the superpopulation.
`target_shift_coef`	Numeric. Coefficient on `X1` in the target population selection model $\Pr(\text{target} \mid X) = \text{invlogit}(c \cdot X_1)$. Defaults to `-0.3`, which produces a mild shift (target slightly over-samples low X_1, opposite but gentle relative to the trial participation model). Set to `0` for (approximately) simple random sampling from the non-trial superpopulation; increase in magnitude for stronger covariate shift.
`plot`	Logical. If `TRUE`, constructs and prints two diagnostic figures comparing covariate and outcome distributions across the superpopulation, the trial, and the target sample. The plot objects are also returned inside the output list under `$plots`. Defaults to `FALSE`.

Value

A named `list` with three (or four, when `plot = TRUE`) elements:

`population` Data frame of size `n_sample` (the target sample drawn as a SRS from the target population). Columns: `X1:X10` (covariates), `A` (treatment indicator, unused for target inference), `Y1`, `Y0` (potential outcomes, provided for verification), `ps_trial` (oracle trial participation PS), `ps_target` (oracle target enrollment PS).

`trials` Data frame of size `n_trial` (the randomized trial). Columns: `X1:X10`, `A`, observed outcome `Y = A*Y1 + (1-A)*Y0`, `ps_trial`, `ps_target`.

`true_values` Named list of six true estimands: `true_ATE_MATE`, `true_Y1_MATE`, `true_Y0_MATE` (averaged over the *target sample*, size `n_sample`); `true_ATE_PATE`, `true_Y1_PATE`, `true_Y0_PATE` (averaged over the *target population*, size `n_target_pop`, corresponding to $E[Y^a \mid S = 0]$ and $E[Y^1 - Y^0 \mid S = 0]$).

`plots` (Only when `plot = TRUE`.) Named list with `covariates` and `outcomes` ggplot objects.

Examples

```
set.seed(2025)
sim <- sim_trans(n_trial = 200, n_target_pop = 5000, n_sample = 1000,
                 scenario = "nonlinear", prop = 0.5, plot = TRUE)
str(sim$population)
table(sim$trials$A)
sim$true_values

sim2 <- sim_trans(n_trial = 60, scenario = "linear+covariate shift",
                  seed = 1, prop = 0.6)
nrow(sim2$trials)
```

---

summary.PSPI_generalizability

Summarize a fitted PSPI object

Description

S3 summary method for objects returned by `PSPI_generalizability` or `PSPI_predict`. Produces posterior point estimates and credible intervals for both the *mixed average treatment effect (MATE)* and the *population average treatment effect (PATE)*, on each potential outcome $E[Y(a)]$ and on each treatment contrast $E[Y(a) - Y(0)]$.

MATE averages the conditional treatment effect uniformly over the observed target sample (the de-facto default of most Bayesian causal inference; Li, Ding & Mealli 2022, eq.~(3.2)). **PATE** uses Bayesian-bootstrap (Dirichlet) weights to integrate the conditional effect against the posterior of the covariate distribution, giving the strictly larger and more honest population-level uncertainty (Li, Ding & Mealli 2022, eq.~(2.2)).

Usage

```
## S3 method for class 'PSPI_generalizability'
summary(object, level = 0.95, ...)

## S3 method for class 'PSPI_predict'
summary(object, level = 0.95, ...)
```

Arguments

object	A fitted object of class "PSPI_generalizability" or "PSPI_predict", typically holding the array post_outcome (npost x N x J) and, if pate_bootstrap = TRUE was used during fitting, the matrix post_outcome_PATE (npost x J).
level	Coverage level for the credible intervals (default 0.95).
...	Ignored (S3 conformity).

Value

A data frame of class "summary.PSPI" with one row per (estimand, target) combination and columns estimand ("E[Y(a)]" or "E[Y(a)-Y(0)]"), target ("MATE" or "PATE"), mean, lower, upper, width, sd.

See Also

PSPI_generalizability, PSPI_predict

Examples

```
## Not run:
sim <- sim_generalizability(n_trial = 200, n_sample = 1000,
                            scenario = "linear", seed = 1)
fit <- PSPI_generalizability(
  X = as.matrix(sim$trials[, paste0("X", 1:10)]),
  Y = sim$trials$Y, A = sim$trials$A,
  pi = cbind(sim$trials$ps, sim$trials$ps),
  X_pop = as.matrix(sim$population[, paste0("X", 1:10)]),
  pi_pop = cbind(sim$population$ps, sim$population$ps),
  model = "DSplineBART", nburn = 500, npost = 500)
summary(fit)

## End(Not run)
```

Index

in_trial, 3
invgumbel, 3

PSPI *(*PSPI-package*)*, 2
PSPI-package, 2
PSPI_fit, 4, *9, 10*
PSPI_generalizability, *5*, 6, *10*, *14, 15*
PSPI_predict, *4, 5*, 9, *14, 15*

sim_generalizability, 11
sim_trans, 12
summary.PSPI_generalizability, 14
summary.PSPI_predict
        *(*summary.PSPI_generalizability*)*,
        14