Package 'PStrata'

May 15, 2026

Type Package

Title Principal Stratification Analysis in R

Version 1.0.1

Date 2026-05-15

Encoding UTF-8

Author Bo Liu [aut, cre],
Fan Li [ctb]

Maintainer Bo Liu <bo.liu1997@gmail.com>

Description Estimating causal effects in the presence of post-treatment confounding using principal stratification. 'PStrata' allows for customized monotonicity assumptions and exclusion restriction assumptions, with automatic full Bayesian inference supported by 'Stan'. The main workflow is PStrataModel() to specify the model, fit() to run MCMC sampling, estimate() to extract potential outcomes, and contrast() to compute causal effects. Visualization tools are provided for diagnosis and interpretation. See Liu and Li (2023) <doi:10.48550/arXiv.2304.02740> for details.

URL https://github.com/LauBok/PStrata

BugReports https://github.com/LauBok/PStrata/issues

Depends R (>= 3.5.0)

Collate PStrata-package.R utils.R survival.R prior.R make_stancode.R sim_data_normal.R sim_data_Cox.R PStrataModel.R fit.R estimate.R contrast.R

License GPL (>= 2)

Suggests R.rsp

Imports ggplot2, rstan, lme4, reformulas, purrr, stringr, stats

LazyData true

RoxygenNote 7.3.3

NeedsCompilation no

Repository CRAN

Date/Publication 2026-05-15 15:50:20 UTC

Contents

contrast . . . 2
diagnostics . . . 3
estimate . . . 4
fit . . . 5
make_stancode . . . 6
plot.PStrataFit . . . 7
prior . . . 8
PStrataModel . . . 9
PStrata_PACKAGE . . . 10
sim_data_Cox . . . 11
sim_data_normal . . . 12
stancode.PStrataFit . . . 13
survival . . . 14

Index **15**

`contrast` *Contrast Potential Outcomes*

Description

Compute pairwise differences of posterior potential outcomes along strata, treatment arms, or time points.

Usage

```
contrast(object, ...)

## S3 method for class 'PSEstimate'
contrast(
  object,
  S = NULL,
  Z = NULL,
  T = NULL,
  type = c("all", "sequential", "cycle"),
  ...
)

## S3 method for class 'PStrataFit'
contrast(
  object,
  S = NULL,
  Z = NULL,
  T = NULL,
  type = c("all", "sequential", "cycle"),
  ...
)
```

Arguments

object	A PSEstimate, PSContrast, or PStrataFit object. If PStrataFit, estimate() is called automatically.
...	Additional arguments passed to estimate() when object is a PStrataFit.
S	Strata to contrast. TRUE for all pairwise, or a numeric/logical index vector. NULL (default) = no contrast.
Z	Treatment arms to contrast. TRUE for all pairwise.
T	Time points to contrast (survival only).
type	"all" (all pairwise), "sequential" (consecutive pairs), or "cycle" (consecutive + wrap-around).

Value

A PSContrast object (inherits from PSEstimate).

Examples

```
data(sim_data_normal)
model <- PStrataModel(
  S.formula = Z + D ~ 1,
  Y.formula = Y ~ 1,
  Y.family  = gaussian(),
  strata    = c(n = "00", c = "01", a = "11"),
  ER        = c("n", "a")
)
ps_fit <- fit(model, data = sim_data_normal, chains = 2, iter = 500)

# Treatment effect (Z contrast) for each stratum
ctr_z <- contrast(ps_fit, Z = TRUE)
summary(ctr_z)
plot(ctr_z)

# Stratum contrasts for each treatment arm
ctr_s <- contrast(ps_fit, S = TRUE)
summary(ctr_s)
```

---

diagnostics *MCMC Convergence Diagnostics*

Description

MCMC Convergence Diagnostics

Usage

```
diagnostics(object, ...)

## S3 method for class 'PStrataFit'
diagnostics(object, pars = NULL, ...)
```

Arguments

object	A PStrataFit object.
...	Currently unused.
pars	Character vector of parameter names. If NULL, key parameters are selected automatically.

Value

Invisibly returns the rstan summary matrix.

---

estimate *Extract Posterior Potential Outcomes*

Description

Maps posterior samples of outcome-group means back to the (stratum x treatment) grid via the SZDG table.

Usage

```
estimate(object, ...)

## S3 method for class 'PStrataFit'
estimate(object, type = c("probability", "RACE"), ...)
```

Arguments

object	A PStrataFit object.
...	Additional arguments (unused).
type	For survival models: "probability" (survival probability) or "RACE" (restricted average causal effect).

Value

A PSEstimate object containing

outcome_array	[S, Z, iter] array for non-survival, or [S, Z, T, iter] for survival.
is_survival	Logical.
time_points	Numeric vector (survival only).
model_info	List of strata names, treatment info, family, ER flags, outcome groups.

Examples

```
data(sim_data_normal)
model <- PStrataModel(
  S.formula = Z + D ~ 1,
  Y.formula = Y ~ 1,
  Y.family  = gaussian(),
  strata    = c(n = "00", c = "01", a = "11"),
  ER        = c("n", "a")
)
ps_fit <- fit(model, data = sim_data_normal, chains = 2, iter = 500)
est <- estimate(ps_fit)
summary(est)
plot(est)
```

`fit` *Fit a Principal Stratification Model*

Description

Estimate a principal stratification model by running MCMC via Stan. The model specification (a `PStrataModel`) is combined with observed data to produce posterior samples.

Usage

```
fit(model, ...)

## S3 method for class 'PStrataModel'
fit(
  model,
  data,
  chains = 4,
  iter = 2000,
  warmup = floor(iter/2),
  cores = 1,
  seed = NULL,
  .debug = FALSE,
  ...
)
```

Arguments

`model`	A `PStrataModel` object.
`...`	Additional arguments passed to `stan`.
`data`	A data frame containing all variables referenced in the model.
`chains`, `iter`, `warmup`, `cores`, `seed`	MCMC settings passed to `stan`.

.debug	Logical. If TRUE, read Stan helper files from inst/ rather than installed package data. Useful during development.

Value

An object of class PStrataFit (or PStrataFitSurvival).

Examples

```
data(sim_data_normal)
model <- PStrataModel(
  S.formula = Z + D ~ 1,
  Y.formula = Y ~ 1,
  Y.family  = gaussian(),
  strata    = c(n = "00", c = "01", a = "11"),
  ER        = c("n", "a")
)
ps_fit <- fit(model, data = sim_data_normal, chains = 2, iter = 500)
summary(ps_fit)
diagnostics(ps_fit)
plot(ps_fit)
cat(ps_fit$stancode)
```

---

make_stancode *Stan Code for PStrata Models*

Description

Generate the Stan code corresponding to the model. This is called internally by fit; use stancode.PStrataFit to retrieve the code from a fitted model.

Usage

```
make_stancode(object, filename = NULL, debug = FALSE)
```

Arguments

object	An internal model specification object.
filename	(optional) string. If not NULL, the Stan file is saved to this path.
debug	only for development/testing use.

Value

A string containing the Stan program.

Examples

```
data(sim_data_normal)
model <- PStrataModel(
  S.formula = Z + D ~ 1,
  Y.formula = Y ~ 1,
  Y.family  = gaussian(),
  strata    = c(n = "00", c = "01", a = "11"),
  ER        = c("n", "a")
)
ps_fit <- fit(model, data = sim_data_normal, chains = 2, iter = 500)
cat(ps_fit$stancode)
```

`plot.PStrataFit` *Plot PStrataFit Objects*

Description

By default, shows violin plots of posterior strata proportions and potential outcomes (non-survival). Use `what = "trace"` or `what = "dens"` for MCMC diagnostics.

Usage

```
## S3 method for class 'PStrataFit'
plot(x, what = c("default", "trace", "dens"), pars = "strata_prob", ...)
```

Arguments

`x`	A `PStrataFit` object.
`what`	`"default"` for strata proportions + potential outcomes, `"trace"` for traceplots, `"dens"` for density plots.
`pars`	Parameter names for trace/dens plots (ignored for default).
`...`	Additional arguments passed to plotting functions.

Value

A ggplot object.

prior *Prior Distribution Functions*

Description

Define prior functions used in PStrata.

Usage

```
prior_flat()

prior_normal(mu = 0, sigma = 1)

prior_t(mu = 0, sigma = 1, df = 1)

prior_cauchy(mu = 0, sigma = 1)

prior_lasso(mu = 0, sigma = 1)

prior_logistic(mu = 0, sigma = 1)

prior_chisq(df = 1)

prior_inv_chisq(df = 1)

prior_exponential(beta = 1)

prior_gamma(alpha = 1, beta = 1)

prior_inv_gamma(alpha = 1, beta = 1)

prior_weibull(alpha = 1, sigma = 1)
```

Arguments

mu, sigma, df, alpha, beta
: parameters for the prior distribution

Value

A list, including the following items.

name name of the distribution

type type of the distribution, one character string of "real" or "positive"

args a named list of all the input parameters

call a function call object of the prior distribution on the parameters

Examples

```
prior_flat()
prior_normal(0, 10)
prior_t(0, 2.5, df = 3)
prior_cauchy(0, 5)
prior_inv_gamma(2, 1)
```

PStrataModel *Define a Principal Stratification Model*

Description

Creates a model specification for principal stratification analysis. No data is required at this stage – the specification is purely symbolic. Use `fit` to estimate the model with data.

Usage

```
PStrataModel(
  S.formula,
  Y.formula,
  Y.family,
  strata,
  ER = NULL,
  prior_intercept = prior_flat(),
  prior_coefficient = prior_normal(),
  prior_sigma = prior_inv_gamma(),
  prior_alpha = prior_inv_gamma(),
  prior_lambda = prior_inv_gamma(),
  prior_theta = prior_normal(),
  survival.time.points = 50
)
```

Arguments

`S.formula`	formula for the stratum model (e.g., `Z + D ~ X1 + X2`).
`Y.formula`	formula for the outcome model (e.g., `Y ~ X1 + X2`).
`Y.family`	a `family` object (e.g., `gaussian()`, `survival("Cox")`).
`strata`	strata definition: character vector, list of vectors, or list of lists.
`ER`	exclusion restriction: character vector of strata names or logical vector.
`prior_intercept, prior_coefficient, prior_sigma, prior_alpha, prior_lambda, prior_theta`	prior distributions for model parameters.
`survival.time.points`	number of time points for survival outcomes.

Value

An object of class `PStrataModel`.

Examples

```
# Non-compliance with three strata (never-taker, complier, always-taker)
model <- PStrataModel(
  S.formula = Z + D ~ 1,
  Y.formula = Y ~ 1,
  Y.family  = gaussian(),
  strata    = c(n = "00", c = "01", a = "11"),
  ER        = c("n", "a")
)
print(model)
summary(model)


# Fit the model (requires rstan and C++ compiler)
data(sim_data_normal)
ps_fit <- fit(model, data = sim_data_normal, chains = 2, iter = 500)
summary(ps_fit)
plot(ps_fit)

# Extract potential outcomes and contrasts
est <- estimate(ps_fit)
summary(est)
plot(est)

ctr <- contrast(ps_fit)
summary(ctr)
plot(ctr)
```

`PStrata_PACKAGE` *PStrata: Principal STRATification Analysis for Data with Post-Randomization Confounding*

Description

The **PStrata** package is designed for estimating causal effects in the presence of post-treatment confounding using principal stratification. It provides an interface to fit the Bayesian principal stratification model, which is a complex mixture model, using **Stan**, a C++ package for obtaining full Bayesian inference. The formula syntax is an extended version of the syntax applied in many regression functions and packages, such as `lm`, `glm` and **lme4-package**, to provide a simple interface. A wide variety of distributions and link functions are supported, allowing users to fit linear, binary or count data, and survival models with principal stratification. Further modeling options include multiple post-treatment confounding variables and cluster random effects. The monotonicity and exclusion restriction assumptions can be easily applied, and prior specifications are flexible and encourage users to reflect their prior belief. In addition, all parameters can be inferred from the posterior distribution, which enables further analysis other than provided by the package.

Details

The Bayesian principal stratification analysis relies on two models, the principal stratum model and the outcome model. The main workflow is:

1. `PStrataModel`: specify the model (formulas, strata, priors).
2. `fit`: compile and run MCMC sampling via Stan.
3. `estimate`: extract posterior potential outcomes.
4. `contrast`: compute causal effect contrasts.

Based on the supplied formulas, data and additional information, it automatically generates the Stan code via `make_stancode` and fits the model using **Stan**.

The estimated probability for each principal stratum and the estimated mean response are calculated with **Stan** as it is faster and more space-efficient. Post-processing methods `summary` and `plot` provide overviews and visualization.

Because **PStrata** heavily relies on **Stan** for posterior sampling, a C++ compiler is required. The program **Rtools** (available on https://cran.r-project.org/bin/windows/Rtools/) comes with a C++ compiler for Windows. On Mac, Xcode is suggested. For further instructions on how to get the compilers running, please refer to the prerequisites section at the RStan-Getting-Started page.

References

The Stan Development Team. Stan Modeling Language User's Guide and Reference Manual. https://mc-stan.org/users/documentation/

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

See Also

`PStrataModel`, `fit`, `estimate`, `contrast`

---

sim_data_Cox *Simulated Dataset for Survival Outcome (Cox Model)*

Description

A dataset generated for illustration of the principal stratification analysis. This dataset represents the common case of non-compliance.

Usage

```
sim_data_Cox
```

Format

'sim_data_Cox' A data frame with 1,000 rows and 7 columns:

S Principal Strata: "never taker", "complier" or "always taker"

Z Randomized treatment arm: 0 = control, 1 = treatment

D Actual treatment arm: 0 = control, 1 = treatment

T True outcome: event time

C Censor time

delta Event indicator. 1 means true outcome is observed; 0 means otherwise

Y The observed event time or censor time

Details

The dataset represents the scenario where actual treatment might not be in compliance with the randomized (assigned) treatment. Defiers and always-takers are ruled out, leaving two strata, "never-taker" and "complier" randomly sampled with probability 0.3, 0.7 respectively. The assigned treatment Z is randomized with 0.5 probability for either arm. The true event time T is given by the following Weibull-Cox distribution

never-taker $Y \sim Weibull - Cox(theta = 1, mu = 0.3)$

complier $Y \sim Weibull - Cox(theta = 1, mu = 2 - 0.6 * Z)$

and the censor time C is uniformly drawn between 0.5 and 2.

The exclusion restriction assumption holds for never-takers in this generated dataset.

Examples

```
data(sim_data_Cox)
head(sim_data_Cox)
table(sim_data_Cox$D, sim_data_Cox$Z)
```

---

sim_data_normal *Simulated Dataset for Normal Outcome*

Description

A dataset generated for illustration of the principal stratification analysis. This dataset represents the common case of non-compliance.

Usage

```
sim_data_normal
```

Format

'sim_data_normal' A data frame with 1,000 rows and 4 columns:

S Principal Strata: "never taker", "complier" or "always taker"

Z Randomized treatment arm: 0 = control, 1 = treatment

D Actual treatment arm: 0 = control, 1 = treatment

Y Outcome

Details

The dataset represents the scenario where actual treatment might not be in compliance with the randomized (assigned) treatment. Defiers are ruled out, leaving three strata, "never taker", "complier" and "always taker" randomly sampled with probability 0.3, 0.2 and 0.5 respectively. The assigned treatment Z is randomized with 0.5 probability for either arm. The outcome Y is given by the following.

never taker $Y \sim N(3, 1)$

complier $Y \sim N(-1 - Z, 0.5)$

always taker $Y \sim N(1, 2)$

The exclusion restriction assumption holds for never takers and always takers in this generated dataset.

Examples

```
data(sim_data_normal)
head(sim_data_normal)
table(sim_data_normal$D, sim_data_normal$Z)
```

`stancode.PStrataFit` *Extract the Generated Stan Code*

Description

Extract the Generated Stan Code

Usage

```
stancode.PStrataFit(object, ...)
```

Arguments

`object`	A `PStrataFit` object.
`...`	Currently unused.

Value

The Stan code as a character string.

`survival` *Family Function for Survival Data*

Description

Construct a family object for survival data

Usage

```
survival(method = c("Cox", "AFT"), link = "identity")
```

Arguments

`method`	the parametric method used for survival data. Can be Cox or AFT.
`link`	a link function, currently only identity is implemented and used

Value

A `family` object

Examples

```
survival("Cox")
survival("AFT")
```

Index

∗ **datasets**
sim_data_Cox, 11
sim_data_normal, 12

contrast, 2, *11*

diagnostics, 3

estimate, 4, *11*

fit, 5, *6*, *9*, *11*

glm, *10*

lm, *10*
lme4-package, *10*

make_stancode, 6, *11*

plot, *11*
plot.PStrataFit, 7
prior, 8
prior_cauchy *(*prior*)*, 8
prior_chisq *(*prior*)*, 8
prior_exponential *(*prior*)*, 8
prior_flat *(*prior*)*, 8
prior_gamma *(*prior*)*, 8
prior_inv_chisq *(*prior*)*, 8
prior_inv_gamma *(*prior*)*, 8
prior_lasso *(*prior*)*, 8
prior_logistic *(*prior*)*, 8
prior_normal *(*prior*)*, 8
prior_t *(*prior*)*, 8
prior_weibull *(*prior*)*, 8
PStrata_PACKAGE, 10
PStrataModel, *5*, 9, *11*

sim_data_Cox, 11
sim_data_normal, 12
stan, *5*
stancode.PStrataFit, *6*, 13
summary, *11*
survival, 14