Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *doi: 10.18637/jss.v000.i00*

PStrata: An R Package for Principal Stratification

Bo Liu
Duke University

Fan Li
Duke University

Abstract

Post-treatment confounded variables are common in causal inference. Such variables occur after the treatment assignment but before the outcome. Special cases include noncompliance in randomized experiments, truncation by death, recruitment bias, etc. Principal stratification (Frangakis and Rubin 2002) is a general framework for handling post-treatment confounding in causal inference. However, principal stratification is not accessible to most applied researchers because its implementation requires complex inference tools and highly customized programming. We develop the R package **PStrata** to automatize statistical analysis of principal stratification for a wide range of common scenarios, including one or multiple, binary or categorical post-treatment variables, continuous, discrete or time-to-event outcomes with both unstructured and clustered data. **PStrata** adopts the Bayesian mixture modeling approach for estimation and inference. It implements an innovative computing architecture that combines R and Stan, a probabilistic programming language for Bayesian analysis. First, R provides the user-interface where the user specifies the setting and assumptions. Then, R generates corresponding Stan code and feed it into Stan to automatize posterior inference of the model parameters. Finally, R takes the posterior simulations output from Stan to calculate point and interval estimates of the causal estimands. We demonstrate **PStrata** via a real application and simulations.

Keywords: Causal inference, Instrumental variable, Intercurrent events, Post-treatment variables, Principal stratification, Noncompliance, Stan.

1. Introduction

Post-treatment confounded variables are common in causal inference. Such variables occur after the treatment assignment but before the outcome, and thus are often referred to as intermediate variables. They cannot be adjusted in the same fashion as pre-treatment variables in causal inference (Rosenbaum 1984). A prominent example is noncompliance to the assigned treatment in randomized trials, which includes treatment switching and discontinuation. An-other example is outcome truncated by a terminal or competing event like death. These cases are known as intercurrent events in clinical trials and pose challenges for evaluating the treatment effects. Principal stratification (Frangakis and Rubin 2002) is a general framework for handling post-treatment confounding in causal inference. It originates from the instrumental variable approach to noncompliance (Angrist, Imbens, and Rubin 1996) and has been expanded to a wide range of settings, e.g. truncation by death (e.g. Rubin 2006; Zhang, Rubin, and Mealli 2009), surrogate endpoints (e.g. Gilbert and Hudgens 2008), recruitment bias in cluster randomized trials (Papadogeorgou, Liu, Li, and Li 2023). Principal stratification has received an intense interest since the influential ICH E9 Addendum on statistical principles for clinical trials (ICH E9 (R1) 2020) listed it as a valid approach to intercurrent events. However, implementing principal stratification requires complex modeling and inference tools and highly customized programming. Therefore, despite the sizeable statistical literature, principal stratification has not been accessible to most applied researchers. And there is a lack of software and corresponding tutorials. Even public available computer code is sparse.

To fill this gap, we develop the R package **PStrata** to automatize statistical analysis of principal stratification. **PStrata** focuses on the general setting where the data on all principal strata are potentially observable, i.e. there is no stratum with structural missing data. **PStrata** adopts the Bayesian mixture modeling approach for estimation and statistical inference. Furthermore, **PStrata** implements an innovative computing architecture that combines R and Stan, a probabilistic programming language for Bayesian analysis. First, R provides the user-interface where the user specifies the setting and assumptions. Then, R generates corresponding Stan code and feeds it into Stan to automatize posterior inference of the model parameters. Finally, R takes the posterior simulations output from Stan to calculate point and interval estimates of the causal estimands. **PStrata** accommodates one or multiple, binary or categorical intermediate variables, continuous, discrete or time-to-event outcomes with both unstructured and clustered data.

This article introduces and illustrates the **PStrata** package. Section 2 reviews the principal stratification framework and the scope of **PStrata**. Section 3 describes the main functions in **PStrata**. Section 4 illustrates these main functions with a case study. Section 5 showcases the use of **PStrata** on several complex scenarios by simulations. **PStrata** is publicly available on CRAN.

2. Overview of principal stratification

2.1. Setup and estimands

We start with the simplest case of a binary treatment and a binary intermediate variable. Assume we have an *iid* sample from a population, with N units. For unit i ($i = 1, \ldots, N$), let Z_i be the treatment assigned to (1 for treatment and 0 for control), Y_i be the outcome, and D_i be a binary intermediate variable that occurs after Z_i and before Y_i. We also observe a set of pre-treatment variables X_i. We assume SUTVA (Rubin 1984) throughout the paper. Because D_i occurs post treatment, it has two potential values, $D_i(z)$ ($z = 0, 1$), of which only the one corresponding to the treatment is observed: $D_i = D_i(Z_i)$. The outcome Y_i also has two potential values, $Y_i(z, D_i(z))$, of which only the one corresponding to the treatment is observed: $Y_i = Y_i(Z_i, D_i(Z_i))$. In the literature, the potential outcome $Y_i(z, D_i(z))$ is often shorthanded to $Y_i(z)$, a convention we follow throughout the paper.

A principal stratification is the classification of units according to their joint potential values of the intermediate variable, $S_i = (D_i(0), D_i(1))$, which is called a principal stratum. With a binary Z and D, there are four strata: $S_i \in \{(0,0), (0,1), (1,0), (1,1)\}$. Principal strata S_i possess a key property: they remain unaffected by the treatment assignment, and thus can be viewed as a pre-treatment variable. Therefore, comparisons of $Y_i(1)$ and $Y_i(0)$ within a principal stratum s are causal effects, known as principal causal effects (PCEs):

$$\tau_s = \mathbb{E}\{Y_i(1) - Y_i(0) \mid S_i = s\} = \mathbb{E}\{Y_i(1) \mid S_i = s\} - \mathbb{E}\{Y_i(0) \mid S_i = s\},$$

for all s. The standard average treatment effect (ATE) is a weighted sum of the PCEs,

$$\tau^{\text{ATE}} = \mathbb{E}\{Y_i(1) - Y_i(0)\} = \sum_s \pi_s \tau_s,$$

where $\pi_s = \Pr(S_i = s)$ is the proportion of the stratum s. Principal stratification offers a refined picture of treatment effect heterogeneity between different subpopulations. Specific case-dependent PCEs are often of the primary interest in the analysis.

We now give two examples of the above general framework. The first example is noncompliance. Here Z is the (randomly) assigned treatment, D is the actual treatment, Y is an outcome. Noncompliance arises when $D_i \neq Z_i$ for some units. The four strata have specific terms (Angrist *et al.* 1996): $S_i = (0,0)$ never-takers, namely, units who would take control regardless of the initial assignment; $S_i = (0,1)$ compliers, namely, units who would take treatment if assigned to treatment and would take control if assigned to control; $S_i = (1,0)$ defiers, namely, units whose actual treatment status is the opposite to the assignment; $S_i = (1,1)$ always-takers, namely, units who would take treatment regardless of the initial assignment. The ATE is the standard intention-to-treat (ITT) effect, which measures the causal effect of the assignment, i.e. effectiveness. In contrast, under suitable assumptions, the causal effect of the compliers can be interpreted as the effect of the actual treatment, i.e. efficacy (Hirano, Imbens, Rubin, and Zhou 2000). This setting is applicable to a variety of situations, differing in the substantive contexts, e.g. randomized encouragement experiment. For simplicity, we will generically refer to this setting as "noncompliance" and use the associated nomenclature in this paper.

The second example is "truncation by death;" below we introduce it in its original context (Rubin 2006). Here Z is a treatment, D is the survival status of a patient sometime after the treatment, Y is the quality of life at the end of the study. Obviously, quality of life is not defined for deceased patients, in other words, the outcome is truncated (or censored) by death. This is different from the standard administrative censoring or truncation in survival analysis. We can define four principal strata based on a patient's joint potential survival status under treatment and control: $S_i = (0,0)$ patients who would die regardless of the initial assignment; $S_i = (0,1)$ patients who would die under control but survive under treatment; $S_i = (1,0)$ patients who would survive under control but die under treatment; $S_i = (1,1)$ patients who would survive regardless of the initial assignment, namely, always-survivors. Because the always-survivor is the only stratum where the outcome is well defined under both treatment and control, it is the only stratum where a causal comparison is meaningful. Therefore, the causal effect specific to this stratum, namely, the survivor average causal effect, is a primary causal estimand. Truncation-by-death is applicable to many different contexts where the

Table 1: Composition of principal strata in observed cells of assigned and actual treatment (Z, D)

	$D = 0$	$D = 1$
$Z = 0$	never-takers, compliers	always-takers, defiers
$Z = 1$	never-takers, defiers	always-takers, compliers

intermediate variable is not death but rather a competing or semi-competing event of the outcome (e.g. Zhang *et al.* 2009).

In terms of data structure, the main difference between the noncompliance and truncation-by-death settings is that conceptually outcomes are observed in all units in the former but are only observed on a subset of units in the latter. Therefore, causal inference in these two settings require entirely different assumptions and inferential procedures. The current version of **PStrata** focuses on the noncompliance setting.

2.2. Model-based estimation

The central challenge in statistical analysis within principal stratification is that, due to the fundamental problem of causal inference, individual principal stratum membership S_i is not observed. In fact, without any assumptions, each observed cell of (Z, D) is composed of a mixture of two strata, as listed in Table 1. Therefore, additional assumptions are required to identify and estimate the PCEs and the main task in analysis is to disentangle the latent mixtures from the observed data.

The first standard assumption is that the treatment assignment is unconfounded conditional on pre-treatment variables.

Assumption **1.** *(Unconfounded assignment).* $\{Y_i(0), Y_i(1), D_i(0), D_i(1)\} \perp\!\!\!\perp Z_i \mid X_i$.

With additional assumptions, the PCEs of some strata can be nonparametrically identified (Angrist *et al.* 1996) (see discussion of Assumption 2-3 later). However, such assumptions are not always plausible. Model-based approach is more flexible, efficient, and can incorporate covariates. In particular, a latent mixture model is natural given the data structure in principal stratification (Imbens and Rubin 1997). This is the approach **PStrata** adopts. Below we will briefly outline the model-based estimation strategy; for more details, see Li, Ding, and Mealli (2023) Section 7.2. Under Assumption 1, one needs to specify two models with unknown global parameter θ: (i) S-model, i.e., a principal strata model given covariates: $\Pr(S_i \mid X_i, \theta_s)$; (ii) Y-model, i.e., an outcome model given the stratum, covariates and treatment: $\Pr(Y_i \mid S_i, X_i, Z_i, \theta_y)$. With these models, the full likelihood is as follows

$$l(\theta) \propto \prod_{i=1}^{n} \sum_{s \in \mathcal{S}: D_i = D(s, Z_i)} \Pr(S_i = s \mid X_i, \theta) \Pr(Y_i \mid S_i = s, Z_i, X_i, \theta), \tag{2.1}$$

where $\mathcal{S}$ is the set of all possible principal strata and $D(s, z)$ denotes actual treatment D_i induced by principal stratum $S_i = s$ and assigned treatment $Z_i = z$. The likelihood function (2.1) resembles that of a mixture model with a latent S_i. Estimation of θ can be conducted through either the frequentist's approach such as the EM algorithm or the Bayesian approach.

PStrata adopts the Bayesian approach because it allows (i) automatic inference for not only the model parameters but also all derived quantities of the parameter such as τ_s, and (ii) straightforward extension to more complicated settings, as will be illustrated.

Apart from the S-model and Y-model, the Bayesian approach requires to specify the prior distribution of θ, $\pi(\theta)$, which, in conjunction with the observed data, leads to the posterior distribution of θ, $\Pr(\theta \mid Z, D, Y, X) \propto \pi(\theta) l(\theta)$. Conventionally, as in Imbens and Rubin (1997) and Hirano *et al.* (2000), the posterior distribution is simulated by a Gibbs sampler (Geman and Geman 1984), which alternatively draws from the posterior (predictive) distribution of the missing strata S_i and the parameter θ. This procedure requires non-trivial and case-specific derivation of conditional distributions and thus is challenging to generalized. To circumvent this challenge, the **PStrata** package utilizes the Stan programming language (Carpenter, Gelman, Hoffman, Lee, Goodrich, Betancourt, Brubaker, Guo, Li, and Riddell 2017; Stan Development Team 2022) to enable automated sampling through the Hamiltonian Monte Carlo (HMC) method (Neal 2011).

Note that the PCEs can be written as

$$\tau_s = \mathbb{E}\{\mathbb{E}\{Y_i \mid Z_i = 1, S_i = s, X_i\} \mid S_i = s\} - \mathbb{E}[\mathbb{E}\{Y_i \mid Z_i = 0, S_i = s, X_i\} \mid S_i = s]. \quad (2.2)$$

Let $g_{z,s}(x;\theta) = \mathbb{E}(Y_i \mid Z_i = z, S_i = s, X_i = x, \theta)$ and $p_s(x;\theta) = \Pr(S_i = s \mid X_i = x, \theta)$, both of which can be directly obtained from the S-model and Y-model. Upon obtaining the posterior samples of the parameter θ, the PCE can be computed as

$$\widehat{\tau}_s(\theta) = \frac{\sum_{i=1}^n g_{1,s}(X_i;\theta) p_s(X_i;\theta)}{\sum_{i=1}^n p_s(X_i;\theta)} - \frac{\sum_{i=1}^n g_{0,s}(X_i;\theta) p_s(X_i;\theta)}{\sum_{i=1}^n p_s(X_i;\theta)}. \quad (2.3)$$

Let $\theta_1, \ldots, \theta_K$ be samples from the posterior distribution of θ. We can plug them into (2.3) to obtain $\widehat{\tau}_s(\theta_1), \ldots, \widehat{\tau}_s(\theta_K)$, which approximates the posterior distribution of τ_s, from which we can calculate the point estimate and credible intervals.

PStrata assumes a multinomial logistic model for S_i and a generalized linear model for Y_i. Specifically, the S-model is given by

$$\log\left\{\frac{\Pr(S_i = s \mid X_i)}{\Pr(S_i = s_0 \mid X_i)}\right\} = \eta_s + X_i'\xi_s, \quad (2.4)$$

where s_0 is an arbitrary reference stratum. The Y-model is given by a generalized linear model where the outcome Y_i is assumed to be generated from a distribution in the exponential dispersion family, with the mean function given by

$$\mathbb{E}\{Y_i \mid S_i = s, Z_i = z, X_i\} = g^{-1}(\alpha_{s,z} + X_i'\beta_{s,z}) \quad (2.5)$$

with some link function g. The distributional form of Y_i and the choice of the link function is specified according to the type of outcome variable. In summary, the parameters include the intercepts $\{\eta_s : s \in \mathcal{S} \setminus \{s_0\}\}$, $\{\alpha_{s,z} : s \in \mathcal{S}, z \in \{0,1\}\}$, the coefficients $\{\xi_s : s \in \mathcal{S} \setminus \{s_0\}\}$, $\{\beta_{s,z} : s \in \mathcal{S}, z \in \{0,1\}\}$ and possibly additional distribution-specific parameters such as $\{\sigma^2_{s,z} : s \in \mathcal{S}, z \in \{0,1\}\}$.

Role of assumptions in model-based inference

Inference of the PCEs can be sharpened by (i) additional assumptions that reduce the number of unknown parameters (Imbens and Rubin 1997), and (ii) covariates X_i that are predictive of the principal strata S_i or/and the outcome Y_i (Mealli and Pacini 2013; Long and Hudgens 2013). Two common assumptions are monotonicity and exclusion restriction.

Assumption **2.** *(Monotonicity):* $D(1) \geq D(0)$.

Assumption **3.** *(Exclusion restriction): for stratum* s *in which* $D(0) = D(1)$, *i.e.,* $s = (0,0)$ *or* $(1,1)$, $Y_i(1) = Y_i(0)$.

In the context of noncompliance, monotonicity rules out defiers: $\Pr(S_i = d) = 0$. Exclusion restriction rules out any direct effect from the assigned treatment to the outcome not mediated through the actual treatment among never-takers and always-takers, so that $\tau_{(0,0)} = \tau_{(1,1)} = 0$. Under Assumptions 1-3, Angrist *et al.* (1996) showed that the PCE of the compliers is nonparametrically identifiable and is equivalent to the two-stage least square estimand as in the instrumental variable method.

More generally, monotonicity specifies restrictions on $D_i(0)$ and $D_i(1)$; various versions of monotonicity reduce the number of possible strata $\mathcal{S}$ and thus the number of parameters to be estimated in (2.1). Exclusion restriction specifies the relation between the outcome distributions for different strata. A generalized version of Assumption 3 is that for stratum s with $D_i(0) = D_i(1)$, then $\Pr\{Y_i(0)|X_i, s\} = \Pr\{Y_i(1)|X_i, s\}$. This forces the corresponding parameters in different mixture components in the Y-model to be the same and thus also reduce the number of free parameters.

PStrata supports user-specification of monotonicity and exclusion restriction. Despite the potential variations in the number of strata and parameters resulting from these assumptions, the posterior distribution of the parameters can be derived in a similar, straightforward manner, and the sampling and inference process remains nearly identical using the HMC sampling method.

2.3. Time-to-event outcome

Survival or more generally time-to-event outcomes are prevalent in real world applications. **PStrata** implements the Bayesian mixture model based method for survival analysis developed in Liu, Wruck, and Li (2023). Below we briefly describe the estimands, assumptions and models; more details can be found in Liu *et al.* (2023). Assume the primary outcome of interest is the time T_i to the first occurrence of an event, which is subject to right censoring at time C_i. Due to the censoring, T_i and C_i are not observed; instead, we observe the censored event time $Y_i = \min\{T_i, C_i\}$ and the event indicator $\delta_i = \mathbf{1}(T_i < C_i)$. Denote by $T_i(z)$ the potential survival time under treatment $z(= 0, 1)$. A common causal estimand with time-to-event outcome is the principal survival probability causal effect (SPCE)

$$\tau_s^{SPCE}(t) = \Pr(T_i(1) > t \mid S_i = s) - \Pr(T_i(0) > t \mid S_i = s). \tag{2.6}$$

A second estimand is the restricted average causal effect (RACE) in survival time,

$$\tau^{RACE}(t) = \mathbb{E}[T_i(1) \wedge t] - \mathbb{E}[T_i(0) \wedge t], \quad 0 \leq t \leq t_{\max}, \tag{2.7}$$

where $a \wedge b \equiv \min(a, b)$.

To identify the SPCE and RACE, we need to make an additional assumption on the censoring process.

Assumption **4.** *(Independent censoring):* $T_i \perp\!\!\!\perp C_i \mid \{X_i, S_i, Z_i\}$.

Under Assumptions 1 and 4, we only need to specify a Y-model $\Pr(T_i \mid S_i, Z_i, X_i)$ to estimate the SPCE. **PStrata** implements the Weibull-Cox proportional hazard model (Cox 1972; Abrams, Ashby, and Errington 1996)

$$h(t; S_i = s, Z_i = z, X_i) = t^{\exp(\theta_{s,z})-1} \exp(\alpha_{s,z} + X_i' \beta_{s,z}). \tag{2.8}$$

and the accelerated failure time (AFT) model (Wei 1992)

$$\log(T_i) = \alpha_{s,z} + X_i' \beta_{s,z} + \epsilon_i, \tag{2.9}$$

where ϵ_i follows a normal distribution $N(0, \sigma_{s,z}^2)$.

Liu *et al.* (2023) show that under parameter θ, the stratum SPCE can be estimated by

$$\widehat{\tau}_s^{SPCE}(t; \theta) = \frac{\sum_{i=1}^n G_{1,s}(t, X_i; \theta) p_s(X_i; \theta)}{\sum_{i=1}^n p_s(X_i; \theta)} - \frac{\sum_{i=1}^n G_{0,s}(t, X_i; \theta) p_s(X_i; \theta)}{\sum_{i=1}^n p_s(X_i; \theta)}, \tag{2.10}$$

where $p_s(X_i; \theta) = \Pr(S_i = s \mid X_i; \theta)$, which is assumed to follow the multinomial logistic S-model 2.4, and $G_{z,s}(t, X_i; \theta) = \Pr(T_i > t \mid S_i = s, Z_i = z, X_i; \theta)$, the analytical form of which can be derived for both the Cox and AFT models. Similarly, the RACE can be estimated as

$$\widehat{\tau}_s^{RACE}(t; \theta) = \frac{\sum_{i=1}^n H_{1,s}(t, X_i; \theta) p_s(X_i; \theta)}{\sum_{i=1}^n p_s(X_i; \theta)} - \frac{\sum_{i=1}^n H_{0,s}(t, X_i; \theta) p_s(X_i; \theta)}{\sum_{i=1}^n p_s(X_i; \theta)}, \tag{2.11}$$

where $H_{z,s}(t, X_i; \theta) = \mathbb{E}[T_i \wedge t \mid S_i = s, Z_i = z, X_i; \theta]$.

2.4. Extensions

PStrata can accommodate several additional settings, encompassing non-binary treatment, non-binary and multiple intermediate variables. Below we briefly introduce these settings, which, despite their increased complexity, retain similarities in terms of inference with the binary setting.

Non-binary treatment

The treatment Z_i can take multiple values, which we denote by $\mathcal{Z}$. In the binary setting, $\mathcal{Z} = \{0, 1\}$. In general, we denote $\mathcal{Z} = \{0, 1, \ldots, Z_{\max} - 1\}$. The principal strata will be similarly defined by the potential values of D_i, $S_i = (D_i(0), \ldots, D_i(Z_{\max} - 1))$.

Non-binary intermediate variable

The intermediate variable D_i can take multiple values from a set $\mathcal{D} = \{0, 1, \ldots, D_{\max} - 1\}$. One example is the non-compliance case where units can either switch treatment or discontinue from any treatment altogether. In this context, we can define the actual treatment as having three distinct levels: two for both treatment arms and an additional level to account for discontinuation (Liu *et al.* 2023). Without any restriction, there are $D_{\max}^2$ potential strata under binary treatment.

Multiple intermediate variables

There can be multiple intermediate variables. For example, in Frumento, Mealli, Pacini, and Rubin (2013), there is both noncompliance and truncation-by-death. To simplify the analysis, these events can be collectively considered as a single intermediate variable, which allows us to define the principal strata effectively. For instance, if there are two intermediate variables denoted as D_{1i} and D_{2i}, we can define a composite intermediate variable $D_i = (D_{1i}, D_{2i})$. Consequently, the principal strata can be defined as $S_i = (D_{1i}(0), D_{2i}(0), D_{1i}(1), D_{2i}(1))$.

In the above three settings, without any restriction, the number of potential principal strata increases exponentially. Usually various versions of monotonicity tailored to specific applications are imposed to reduce the number of principal strata.

Multilevel data

Multilevel data arise when the units are not independent samples from the population but instead belong to natural clusters, such as schools, hospitals, or regions. For instance, in a clinical trial conducted across multiple medical centers, patients from the same medical center may exhibit correlated outcomes.

In this context, we assume the presence of J pre-defined clusters, denoted by $j = 1, \ldots, J$, each associated with specific cluster-level covariates represented as V_j. Every subject, indexed as $i = 1, \ldots, n$, is pre-assigned to a fixed and known cluster, denoted as C_i. Let X_i represent the individual-level covariates for subject i, and we define W_i as the concatenated vector $W_i = (X_i, V_{C_i})$, which combines the individual and cluster-level covariates of subject i. To account for the dependence in outcomes within clusters, we include in both S-model and Y-model normally distributed cluster-specific random intercepts. The likelihood function, though complex in form, can be calculated with these model specification and the posterior samples can be drawn by **PStrata** using the Stan language.

PStrata accommodates nearly all scenarios of multilevel settings that can be formulated using the R package **lme4**. For more comprehensive information, refer to Bates, Mächler, Bolker, and Walker (2015). Across all the aforementioned settings, the fundamental inference framework remains consistent. The primary distinction arises from variations in the number of parameters, resulting from the different levels of S_i and Z_i. Notably, **PStrata** simplifies the process by enabling automatic Bayesian inference of the causal effect in all these settings. This package allows users to concentrate on selecting the principal stratification settings that align with their data while delegate the technical details to the back-end.

3. Overview of the package

As discussed in Section 2, the Bayesian inference consists of three main steps. First, specify the models and prior distribution of the model parameters, and derive the likelihood of the data. Second, draw samples from the posterior distribution of these parameters. Third, calculate estimands with these posterior samples. **PStrata** closely adheres to these steps, as depicted in Figure 1. First, the user specifies the data generating models, namely the S-model and Y-model, along with the prior distribution of the model parameters. Additionally, the user sets assumptions that are appropriate for the analysis. Next, **PStrata** utilizes the Stan programming language to perform the crucial step of drawing samples from the posterior distribution. To accomplish this, **PStrata** automatically generates the corresponding Stan code and data based on the user's input. This code and data are subsequently processed by Stan, which generates the samples from the posterior distribution. Finally, these samples are returned to **PStrata**, where they are used to estimate causal estimands and their associated uncertainties. **PStrata** also provides summary and visualization of the results.

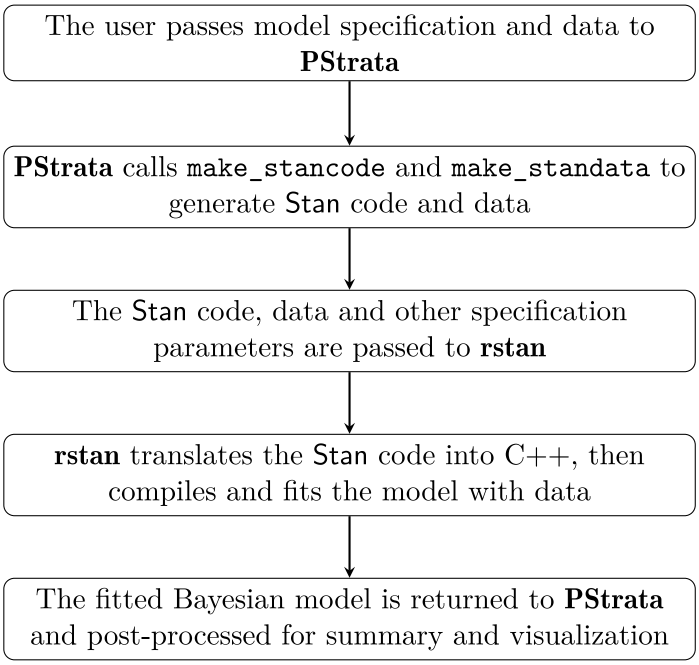
The user passes model specification and data to **PStrata**
PStrata calls `make_stancode` and `make_standata` to generate `Stan` code and data
The `Stan` code, data and other specification parameters are passed to **rstan**
rstan translates the `Stan` code into C++, then compiles and fits the model with data
The fitted Bayesian model is returned to **PStrata** and post-processed for summary and visualization

Figure 1: Overview of the model fitting process in **PStrata**

The central function of **PStrata**, named `PStrata`, is an integrated module that bridges between the front-end and back-end components. On the front-end, users input model specifications and data, and in return, receive causal estimates as output. Unlike the Gibbs sampler, the sampling process on the back-end is decoupled from the front-end, alleviating users from the complexities associated with sampling. The core submodules within **PStrata**, namely `PSObject`, `PSSample`, `PSOutcome`, and `PSContrast`, collectively fulfill this requirement. The `PSObject` submodule interprets the user input to generate a principal stratification object containing crucial information about model specifications, providing sufficient details to determine the posterior distribution of parameters. Subsequently, the `PSSample` submodule calls **Stan** to generate samples from the posterior distribution. These posterior samples are then returned to `PStrata` and further processed by `PSOutcome` and `PSContrast` to calculate potential outcomes and causal effects. The relationships between these modules are depicted in Figure 2.

The procedure is divided into two phases: the *design* phase and the *analysis* phase, based on whether they occur before or after invoking `Stan` to generate posterior samples.

3.1. Design phase

The core function for the design phase is the `PSObject` function where users specify the S-model and Y-model, define assumptions regarding monotonicity and exclusion restriction (ER), and set prior distributions for parameters within these models. The function is declared as follows:

```
PSObject(S.formula, Y.formula, Y.family, data, strata, ER,
```

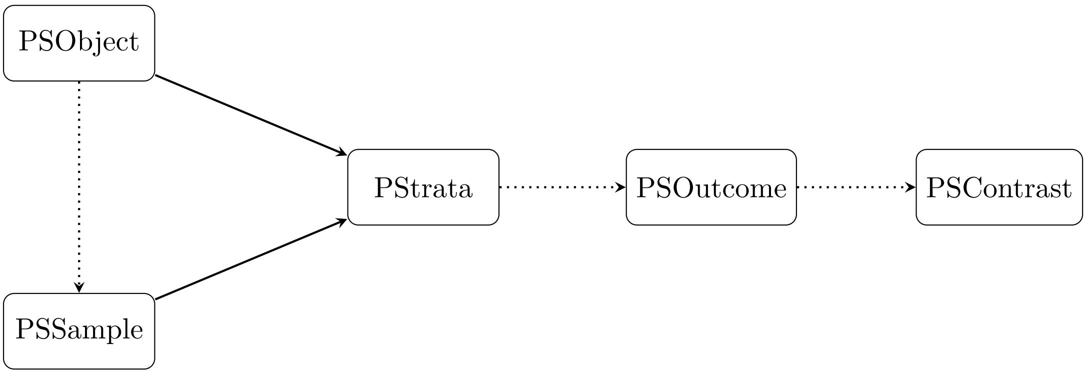

Figure 2: Core modules designed in **PStrata**. Solid arrows indicate that the start-point module is called by the end-point module. Dotted arrows indicate that the outcome of the start-point module is passed to the end-point module as an argument.

```
    prior_intercept, prior_coefficient, prior_sigma,
    prior_alpha, prior_lambda, prior_theta, survival.time.points)
```

Below we will provide more details of each of these arguments.

S-model and Y-model

Central to our model specification are the S-model (2.4) and the Y-model (2.5). The user needs to select the linear predictors X_i and choose the appropriate distribution for the generalized linear model.

In **PStrata**, we define the linear predictors in the S-model and Y-model with a `formula` object in R, which aligns with the established convention in the `lm()` and `glm()` functions. Specifically, the argument `S.formula` has the syntax like `Z + D ~ X1 + X2`. The left hand side consists of two variables `Z` and `D` concatenated by the plus (`+`) sign, representing the treatment and intermediate variable, respectively. The argument `Y.formula` follows the similar syntax, such as `Y ~ X1 + X2`, where the variable `Y` on the left hand side denotes the outcome variable. In both `S.formula` and `Y.formula`, the right hand side consists of the linear predictors. Each predictor can be a raw covariate (*e.g.* `X1`), a transformed covariate (*e.g.* `log(X1)`) or the interaction between multiple covariates (*e.g.* `X1:X2`).

The generalized linear model for the outcome is specified by the `Y.family` argument, which represents both the choice of probability distribution family and the associated link function, following the convention of `glm()`. **PStrata** supports nearly every family and link function in the `glm()` function. Table 2 provides a comprehensive list.

Assumptions

Recall that the principal stratum $S_i = (D_i(0), D_i(1))$. In **PStrata**, each stratum is denoted by a `character` object accordingly, with the following representations: “11” for always-takers, “00” for never-takers, “01” for compliers, and “10” for defiers.

The monotonicity assumption determines the set of all possible strata. This can be specified by listing all strata in the `strata` argument. For example in the context of binary noncompliance, we may specify `strata = c("00", "01", "10", "11")` if we do not assume monotonicity.

Family	Link Functions	Density Function
`gaussian`	`identity`, `log`, `inverse`	$\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{-(y-\mu)^2}{2\sigma^2}\right)$
`binomial`	`logit`, `probit`, `cauchit`, `log`, `cloglog`	$\mu^y(1-\mu)^{1-y}$
`Gamma`	`identity`, `log`, `inverse`	$\frac{(\alpha/\mu)^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \exp(-\frac{\alpha y}{\mu})$
`poisson`	`identity`, `log`, `sqrt`	$\mu^y e^{-\mu}/(y!)$
`inverse.gaussian`	`1/mu^2`, `inverse`, `identity`, `log`	$\sqrt{\frac{\lambda}{2\pi y^3}} \exp\left(-\frac{\lambda(y-\mu)^2}{2y\mu^2}\right)$

Table 2: Supported families, link functions and corresponding density/mass functions expressed by the mean μ and additional parameters in **PStrata**.

With monotonicity in a two-sided noncompliance setting, we may specify `strata = c("00", "01", "11")`. Similarly, with monotonicity in a one-sided noncompliance setting, we may specify `strata = c("00", "01")`, or optionally for better clarify with labels `strata = c(nt = "00", co = "01")`, where `nt` and `co` stand for never-takers and compliers, respectively.

In **PStrata**, the exclusion restriction assumption is encapsulated by the `ER` argument, which accepts a boolean vector of the same length as `strata`. This vector indicates whether the exclusion restriction is assumed for each principal stratum. For simplicity, the monotonicity and exclusion restriction assumptions can be jointly specified by appending an asterisk after each stratum where the exclusion restriction is assumed. For instance, `strata = c(nt = "00*", co = "01", at = "11*")` denotes the combination of monotonicity with the exclusion restriction assumption for both always-takers and never-takers.

Priors

The S-model (2.4) and the Y-model (2.5) introduce a range of parameters, including intercepts (η and $\alpha_{s,z}$), coefficients (ξ_s and $\beta_{s,z}$), and additional parameters tailored to specific distribution families, such as the variance parameter $\sigma^2_{s,z}$ when the outcome follows a Gaussian distribution. Depending on the type of the parameter, the prior distribution can be designated through `prior_intercept`, `prior_coefficient`, and, when applicable, `prior_sigma`, `prior_alpha`, or `prior_lambda` for Gaussian, Gamma, and inverse Gaussian families, respectively. While many of these parameters take values over the entire real line, some, like $\sigma^2_{s,z}$, are constrained to positive values. **PStrata** offers flexibility in specifying prior distributions for these parameters. Table 3 provides a comprehensive list of supported prior distributions classified by the support.

By default, the prior distribution for the intercept is the flat (improper) distribution over the entire real line, and the prior distribution of the coefficients is the standard Gaussian distribution. The default prior distribution for other parameters is the inverse-gamma distribution $\mathcal{IG}(1,1)$.

Time-to-event outcome

PStrata provides full support for inference involving right-censored outcomes, with slightly adjusted syntax. Specifically, the Y-model now incorporates an additional variable: the event indicator, denoting whether the event occurred before censoring. Consequently, the `Y.formula` requires an adaptation to `Y + delta ~ X1 + X2`, where both the censored event

Support	Prior distribution	Specification
$(-\infty, \infty)$	Uniform (improper)	`prior_flat()`
	Normal	`prior_normal(mean = 0, sigma = 1)`
	Student t	`prior_t(mean = 0, sigma = 1, df = 1)`
	Cauchy	`prior_cauchy(mean = 0, sigma = 1)`
	Double exponential	`prior_lasso(mean = 0, sigma = 1)`
	Logistic	`prior_logistic(mean = 0, sigma = 1)`
$(0, \infty)$	Chi squared	`prior_chisq(df = 1)`
	Inverse Chi squared	`prior_inv_chisq(df = 1)`
	Exponential	`prior_exponential(beta = 1)`
	Gamma	`prior_gamma(alpha = 1, beta = 1)`
	Inverse Gamma	`prior_inv_gamma(alpha = 1, beta = 1)`
	Weibull	`prior_weibull(alpha = 1, sigma = 1)`

Table 3: List of prior distributions.

time `Y` and the event indicator `delta` are included on the left-hand side. Moreover, when dealing with right-censored outcomes, the `Y.family` parameter should be set to `survival`. Depending on the desired model, the user can further specify `method = "Cox"` for the Weibull-Cox model or `method = "AFT"` for the AFT model. Finally, the user must provide a vector of time points to the `survival.time.points` argument, which denotes the specific time points at which survival probabilities will be computed. The estimand is by default `estimand = "probability"`, which calculates the SACE, but can be customized to `estimand = "RACE"` to calculate the RACE.

Multilevel data

PStrata can include random effects in presence of clustered data. This is achieved by specifying these effects within the `S.formula` and/or `Y.formula` using the same syntax as the widely-used **lme4** package. For instance, consider a scenario where an additive random effect for the outcome is included, with `C` designating the cluster variable. In this case, the Y-model is defined as `Y.formula = Y ~ X1 + X2 + (1 | C)`, where `(1 | C)` implies the inclusion of a random intercept for each level of `C`. For more comprehensive guidance on incorporating random effects, please refer to the documentation of **lme4** (Bates *et al.* 2015).

Non-binary treatment or intermediate variables

The syntax adapts to scenarios involving non-binary treatment or intermediate variables. Even as the number of strata increases in such cases, the `strata` parameter can be specified in a consistent manner. Each stratum is represented by a character string that denotes the potential values of the intermediate variable. For instance, with a three-arm treatment, each stratum can be characterized by a string of length three, representing the values of $D(0)$, $D(1)$, and $D(2)$, respectively.

In scenarios involving multiple intermediate variables, one needs to include all these variables on the left-hand side of `S.formula`. For instance, we can specify it as `Z + D1 + D2 ~ X1`

`+ X2`. In this context, each stratum is still represented by a character string. To do this, consider each intermediate variable separately to get a character string representation and then concatenate these strings together, separated by the `|` symbol. Specifically, the stratum defined by $(D_1(0), D_1(1), D_2(0), D_2(1)) = (0, 1, 1, 1)$ can be represented by `"01|11"`.

3.2. Analysis phase

PStrata calls **rstan**, which provides an R interface to the Stan language to draw samples from the posterior distribution of the parameters. In the analysis phase, **PStrata** use these posterior samples to calculate causal effects and provide summary tables and visualization.

The main function `PStrata` calls module `PSObject` and `PSSample` to do the analysis, and return a `PStrata` object containing the model information and posterior samples of the parameters, which can be used to calculate the stratum probability. Furthermore, modules `PSOutcome` and `PSContrast` can be used to estimate the stratum-specific mean outcome and causal effects.

The typical `PStrata` snippet looks like

```
PStrata(PSobject = NULL, S.formula, Y.formula, Y.family, data = NULL,
    strata = NULL, 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, filename = NULL, ...)
```

The `summary.PStrata()` function syntheses information from the `PStrata` object and returns the estimated proportions of each principal stratum. Furthermore, the `PSOutcome()` function estimates the expected value of outcomes of each principal stratum, and the`PSContrast()` function calculates the contrast of outcomes between treatment groups or strata. When the contrast is taken between treatment groups, it represents the principal causal effect.

4. Case study of a real application

We demonstrate the **PStrata** package using a case study that aims to estimate the effect of hospitalization of influenza vaccine. The data was originally collected and analyzed by Mcdonald, Hui, and Tierney (1992) and reanalyzed using principal stratification by Hirano *et al.* (2000). The study follows a randomized encouragement design: physicians were randomly selected to receive a letter to remind them to encourage their patients to take flu vaccine. The treatment Z is the encouragement, the intermediate variable D is the actual vaccination status of the patient, and the outcome Y is whether the patients had hospital visits for flu-related reasons. For each patient, several covariates are observed, including age and the rate of chronic obstructive pulmonary disease (COPD). The conventional Intention-to-Treat (ITT) approach estimates the causal effect of Z_i on Y_i, which is the effectiveness of the encouragement to vaccinate. Principal stratification can provide information about the efficacy of the vaccination.

Our illustration follows the analysis in Hirano *et al.* (2000). Mathematically, the randomized encouragement design can be framed exactly as a noncompliance problem. Specifically, we can define four principal strata: always-vaccinated ($S_i = (1, 1)$), never-vaccinated ($S_i = (0, 0)$), compliant-vaccinated ($S_i = (0,1)$), and defiers ($S_i = (1,0)$). Given that flu vaccination is not mandatory and can be taken irrespective of encouragement, noncompliance is two-sided here, that is, non-compliant patients can be either the always-vaccinated or never-vaccinated strata. We assume monotonicity to rule out defiers, which is deemed plausible in this context. Furthermore, we assume exclusion restriction for the always-vaccinated and never-vaccinated strata. The principal causal effect on the compliant-vaccinated patients can be interpreted as the efficacy of vaccination on hospitalization.

Define S-model and Y-model

Assuming monotonicity, there are three principal strata, and thus we imposed a multinomial logistic model with three categories as the S-model. The outcome is a binary variable and thus we imposed a logistic model as the Y-model. We include linear predictors age and COPD into both the S-model and Y-model. We assumed exclusion restriction for always-compliant and never-compliant strata. Further, we assume standard normal priors for both intercepts and coefficients, which are the default in **PStrata**. We sample from the posterior distribution of the parameters using 6 chains, each of which contains 2000 iterations including 1000 warm-up iterations. To speed out the computation, we distribute the sampling job to 6 cores.

```
R> fit <- PStrata(
+    S.formula = encouragement + vaccination ~ age + copd,
+    Y.formula = hospital ~ age + copd,
+    Y.family = binomial(link = "logit"),
+    data = flu,
+    strata = c(n = "00*", c = "01", a = "11*"),
+    prior_intercept = prior_normal(0, 1),
+    prior_coefficient = prior_normal(0, 1)
+    warmup = 1000, iter = 2000,
+    cores = 6, chains = 6
+  )
```

The fitted object `fit` is a `PStrata` object which contains essential information to calculate the causal estimands.

```
R> fit

PStrata Object with 3 strata.
The estimated proportion for each strata:
0.6940673 0.1137423 0.1921904
Use summary() to show confidence intervals.
Use PSOutcome() to show outcomes and PSContrast() to show contrasts.
```

Estimate stratum proportion

We can use the `summary()` function to obtain the quantiles of the proportion of each stratum over the 6000 posterior draws.

```
R> summary(fit)

       mean         sd      2.5%       25%    median       75%     97.5%
 n 0.6940673 0.01196718 0.6707998 0.6859948 0.6940854 0.7022795 0.7170083
 c 0.1137423 0.01621615 0.0803734 0.1034171 0.1138241 0.1246476 0.1455330
 a 0.1921904 0.01055071 0.1717857 0.1850637 0.1918401 0.1992098 0.2128891
```

From the fitted model, 69.4% (CI: 67.1% to 71.7%) of the patients are classified as never-vaccinated, 19.2% (CI: 17.2% to 21.3%) of the patients are classified as always-vaccinated, and the remaining 11.4% (CI: 8.0% to 14.6%) are compliant-vaccinated. These values are close to the non-parametric estimates 71.2%, 18.4% and 10.4% of the three strata, respectively.

Estimate mean effects

```
R> outcome <- PSOutcome(fit)
R> summary(outcome, "matrix")

        mean       sd     2.5%       25%    median      75%    97.5%
n:0 0.081546 0.007444 0.067090 0.076531 0.081497 0.086451 0.096345
n:1 0.081546 0.007444 0.067090 0.076531 0.081497 0.086451 0.096345
c:0 0.166652 0.063261 0.060931 0.122136 0.160119 0.204622 0.310435
c:1 0.068589 0.027496 0.024559 0.048624 0.065305 0.085192 0.129120
a:0 0.100001 0.014312 0.073536 0.090090 0.099535 0.109130 0.129424
a:1 0.100001 0.014312 0.073536 0.090090 0.099535 0.109130 0.129424
```

From the output, one can see that the mean outcome for both arms is 0.082 (95% CI: 0.067 to 0.096) for the never-vaccinated, and 0.100 (CI: 0.074 to 0.129) for the always-vaccinated. For these two strata, the mean outcomes for the treatment arm and the control arm are the same, which is a consequence of the exclusion restriction assumption. For the compliant-vaccinated, the mean outcome is 0.167 (CI: 0.061 to 0.310) for the control arm and 0.069 (CI: 0.026 to 0.129) for the treatment arm.

Estimate causal effects

The `PSContrast` function can be used to calculate the difference of mean outcomes under two treatment arms.

```
R> contrast <- PSContrast(outcome, Z = TRUE)
R> summary(contrast, "matrix")

                mean       sd      2.5%       25%    median       75%    97.5%
n:{1}-{0}  0.000000 0.000000  0.000000  0.000000  0.000000  0.000000 0.000000
c:{1}-{0} -0.098063 0.068129 -0.246702 -0.140799 -0.092666 -0.052169 0.022559
a:{1}-{0}  0.000000 0.000000  0.000000  0.000000  0.000000  0.000000 0.000000
```

In the above snippet, `Z = TRUE` specifies that we would like to find contrasts between all levels of treatment Z. The contrasts for never-vaccinated and always-vaccinated are identically zero due to the exclusion restriction assumption. The contrast for the compliers, i.e., the complier average causal effect, is -0.098 with 95% CI of (-0.247, 0.023).

Conclusion

Results from the principal stratification analysis suggest that, on average, encouragement for flu vaccination reduces the risk of hospital visits among the compliant-vaccinated patients by 9.8% with standard error 6.8%. The analysis also reveals heterogeneity between strata. For instance, the risk of hospitalization for the never-vaccinated and the always-vaccinated patients is 8.2% (standard error 0.7%) and 10.0% (standard error 1.4%), respectively. These results are similar to those obtained in Hirano *et al.* (2000). The slight difference in the standard error is mainly because of the different prior distribution specifications.

5. Case studies with simulations

This section use simulations to demonstrate **PStrata** in several more complex scenarios. For simplicity in illustration, we specify the correct models and do not include covariates in these examples.

5.1. Case 1: Two intermediate variables

This simulation study features a scenario with two intermediate variables D_1 and D_2. We define the principal strata as $S = (D_1(0), D_1(1), D_2(0), D_2(1))$. Here we include 5 out of 16 strata, namely $\mathcal{S} = \{0000, 0001, 0101, 0011, 1111\}$, and assume exclusion restriction for the "0000", "0011" and "1111" strata. There are no baseline covariates.

We simulate 10,000 sample units. We randomly assign principal stratum S and treatment status Z to each unit, with the stratum-assignment probability being $p = (0.15, 0.3, 0.2, 0.2, 0.15)$ and the treatment assignment probability $P(Z = 1) = 0.5$. The outcome variable Y is sampled from a Gaussian distribution given in Table 4.

	Probability	$Z = 0$	$Z = 1$
$S = 0000$	0.15	$\mathcal{N}(3, 1)$	
$S = 0001$	0.30	$\mathcal{N}(-1, 0.5)$	$\mathcal{N}(-3, 0.5)$
$S = 0011$	0.20	$\mathcal{N}(2, 0.5)$	$\mathcal{N}(5, 0.5)$
$S = 0101$	0.20	$\mathcal{N}(-1, 3)$	
$S = 1111$	0.15	$\mathcal{N}(1, 2)$	

Table 4: Outcome specification for Simulation 2

We run the sampler with 6 chains and 1000 warm-up iterations and 1000 sampling iterations for each chain. The true values of parameters and the respective posterior means are given in Table 5. The posterior mean estimates of the stratum probability are generally close to the true values, although larger discrepancy can be observed for the 0001 and 0011 group. Also, wide credible intervals are reported for the causal effect in the two strata where exclusion restriction is not assumed. These are as expected because the principal causal effects in these strata are weakly identified without additional information such as predictive covariates

(Imbens and Rubin 1997).

```
R> fit <- PStrata(
+    S.formula = Z + D1 + D2 ~ 1,
+    Y.formula = Y ~ 1,
+    Y.family = gaussian(),
+    data = data_sim1,
+    strata = c(
+      nn = "00|00*", nc = "00|01", cc = "01|01", na = "00|11*", aa = "11|11*"
+    ),
+    prior_intercept = prior_normal(0, 1),
+    warmup = 1000, iter = 2000,
+    cores = 6, chains = 6
+  )
```

	True value	Posterior mean	2.5% quantile	97.5% quantile
$\Pr(S = 0000)$	0.150	0.153	0.141	0.166
$\Pr(S = 0001)$	0.300	0.278	0.247	0.309
$\Pr(S = 0011)$	0.200	0.226	0.192	0.260
$\Pr(S = 0101)$	0.200	0.195	0.186	0.206
$\Pr(S = 1111)$	0.150	0.146	0.139	0.153
$\mathsf{E}[Y(1) - Y(0) \mid S = 0001]$	-2.000	-3.504	-5.072	-1.950
$\mathsf{E}[Y(1) - Y(0) \mid S = 0101]$	3.000	4.505	2.964	6.046

Table 5: The posterior summary of important quantities with true values in Simulation 2.

5.2. Case 2: Time-to-event outcome

We simulate a randomized experiment with $N = 10,000$ units under the non-compliance setting with binary Z and D and a time-to-event outcome. We assign $S \in \{(0,0),(0,1),(1,1)\}$ independently with probability 0.25, 0.60 and 0.15, respectively. The treatment assignment Z_i is independently drawn from a *Bernoulli*(0.5). The true uncensored failure time is generated from the Weibull-Cox model (2.8), with separate parameters for each of the six combination of stratum and the treatment assignment. For estimation, we specify the Weibull-Cox model as

$$h(t; S_i = s, Z_i = z) = t^{\exp(\theta_{s,z})-1} \exp(\alpha_{s,z}). \tag{5.1}$$

Table 6 presents the Y-model parameters for these two scenarios. We draw the censoring time C_i independently from an exponential distribution with rate 0.3, leading to an event rate being approximately 35%.

```
R> fit <- PStrata(
+    S.formula = Z + D ~ 1,
+    Y.formula = Y + delta ~ 1,
+    Y.family = survival(method = "Cox"),
```

		$\Pr(Y(0) \mid S=s)$		$\Pr(Y(1) \mid S=s)$	
s	$\Pr(S=s)$	$\theta_{s,0}$	$\alpha_{s,0}$	$\theta_{s,1}$	$\alpha_{s,1}$
n	0.25	log(2.0)	-4.0	log(1.5)	-3.0
c	0.60	log(1.5)	-2.5	log(1.0)	-1.5
a	0.15	log(1.0)	-1.0	log(0.6)	0.0

Table 6: Parameters of the true S-model and T-model in simulation 3.

```
+    data = data_sim2,
+    strata = c(n = "00", c = "01", a = "11"),
+    prior_intercept = prior_normal(0, 1),
+    warmup = 1000, iter = 2000,
+    cores = 6, chains = 6
+  )

R> outcome <- PSOutcome(fit, type = 'probability')
R> plot(outcome) + xlab("time") + ylab("survival probability")
```

The estimated probabilities for three strata are respectively 0.234 (CI: 0.222 to 0.245), 0.611 (CI: 0.596 to 0.627) and 0.155 (CI: 0.145 to 0.165), close to the true values 0.25, 0.60 and 0.15. With the `plot()` function, we can visualize the estimated survival probability curves and their 95% confidence regions (Figure 3). The true survival probability curves are added for reference. We can also calculate the restricted average causal effect (RACE) by using option `type = "RACE"`.

5.3. Case 3: Multilevel data

We simulate a randomized experiment with $N = 1,000$ units. We assign $S \in \{(0,0),(0,1),(1,1)\}$ independently with probability 0.25, 0.50 and 0.25, respectively. The treatment assignment Z_i is independently drawn from a *Bernoulli*(0.5). To simulate a clustering structure, we randomly assign these units to 10 clusters, denoted by $C_i = j$. We sample random effect ξ_j from the standard normal distribution for each cluster j. Conditional on the stratum S and treatment assignment Z, the outcome Y follows a Gaussian distribution where

$$\begin{aligned}
Y \mid S=(0,0), Z=z, X_1, X_2, C=j &\sim \mathcal{N}(X_1 - X_2 + \xi_j, 0.3)\\
Y \mid S=(0,1), Z=z, X_1, X_2, C=j &\sim \mathcal{N}(2X_1 - (1+z)X_2 + 2 + 6z + \xi_j, 0.2)\\
Y \mid S=(1,1), Z=z, X_1, X_2, C=j &\sim \mathcal{N}(X_1 + X_2 - 1 + \xi_j, 0.2).
\end{aligned}$$

We run the sampler with 6 chains and 1000 warm-up iterations and 1000 sampling iterations for each chain. The true values of stratum probability and the causal effects and the respective posterior distributions are given in Table 7. The posterior mean estimates of the stratum probability and the complier average causal effect are close to the corresponding true values.

```
R> fit <- PStrata(
+    S.formula = Z + D ~ 1,
```

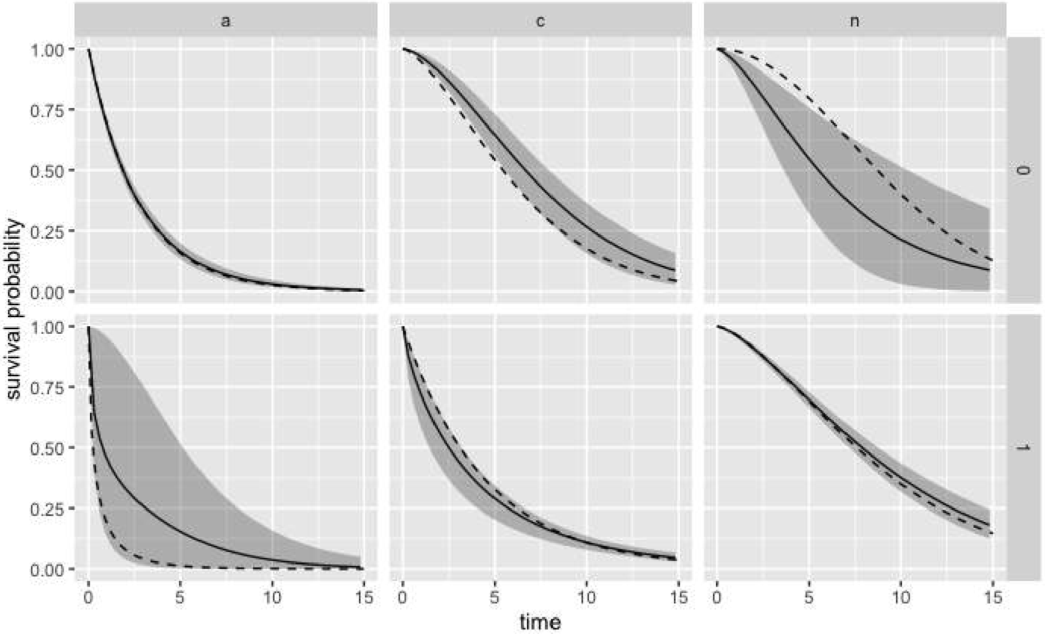

Figure 3: Plot of estimated survival probability curves (solid) with 95% confidence region (shaded). The true survival probability curves are also plotted (dashed).

```
+    Y.formula = Y ~ X1 + X2 + (1 | C),
+    Y.family = gaussian(),
+    data = data_sim3,
+    strata = c(n = "00*", c = "01", a = "11*"),
+    prior_intercept = prior_normal(0, 1),
+    warmup = 1000, iter = 2000,
+    cores = 6, chains = 6
+  )
```

	True value	Posterior mean	2.5% quantile	97.5% quantile
$\Pr(S = (0,0))$	0.300	0.281	0.252	0.308
$\Pr(S = (0,1))$	0.500	0.520	0.487	0.552
$\Pr(S = (1,1))$	0.200	0.199	0.176	0.222
$\mathsf{E}[Y(1) - Y(0) \mid S = (0,1)]$	6.000	5.978	5.943	6.012

Table 7: The posterior summary of important quantities with true values in Simulation 4.

6. Summary

Principal stratification is a general framework for handling post-treatment confounded variables in causal inference. This paper introduces the **PStrata** package, which targets at filling the gap between the theory and practice of principal stratification. **PStrata** adopts an innovative computing architecture that combines R and Stan to provide a unified computing platform for a wide range of settings of principal stratification. **PStrata** is under continuing development; future developments will extend to the settings with structural missing data such as truncation by death and recruitment bias, as well as more model diagnostics and visualization tools.

Computational details

PStrata 0.0.4 was built on R 4.0.3 and dependent on the **dplyr** 1.0.7 package, the **purrr** 0.3.4 package, the **rstan** 2.21.1 package, the **lme4** 1.1-27.1 package, the **ggplot2** 3.3.6 package, the **abind** 1.4-5 package and the **stringr** 1.5.0 package. All these packages used are available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/`.

Acknowledgement

This research is supported by the Patient-Centered Outcomes Research Institute (PCORI) contract ME-2019C1-16146. We thank Laine Thomas and Peng Ding for helpful comments.

References

Abrams K, Ashby D, Errington D (1996). "A Bayesian approach to Weibull survival models—Application to a cancer clinical trial." *Lifetime Data Analysis*, **2**(2), 159–174.

Angrist J, Imbens G, Rubin D (1996). "Identification of causal effects using instrumental variables." *Journal of the American Statistical Association*, **91**(434), 444–455.

Bates D, Mächler M, Bolker B, Walker S (2015). "Fitting Linear Mixed-Effects Models Using lme4." *Journal of Statistical Software*, **67**(1), 1–48. `doi:10.18637/jss.v067.i01`. URL `https://www.jstatsoft.org/index.php/jss/article/view/v067i01`.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017). "Stan: A Probabilistic Programming Language." *Journal of Statistical Software*, **76**(1), 1–32.

Cox DR (1972). "Regression Models and Life-Tables." *Journal of the Royal Statistical Society: Series B (Methodological)*, **34**(2), 187–202.

Frangakis C, Rubin D (2002). "Principal Stratification in Causal Inference." *Biometrics*, **58**(1), 21–29.

Frumento P, Mealli F, Pacini B, Rubin D (2013). "Evaluating the Effect of Training on Wages in the Presence of Noncompliance, Nonemployment, and Missing Outcome Data." *Journal of the American Statistical Association*, **107**, 450–466.

Geman S, Geman D (1984). "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **PAMI-6**(6), 721–741. `doi:10.1109/TPAMI.1984.4767596`.

Gilbert P, Hudgens M (2008). "Evaluating candidate principal surrogate endpoints." *Biometrics*, **64**(4), 1146–1154.

Hirano K, Imbens G, Rubin D, Zhou XH (2000). "Assessing the effect of an influenza vaccine in an encouragement design." *Biostatistics*, **1**, 69–88.

ICH E9 (R1) (2020). "Addendum on estimands and sensitivity analysis in clinical trials to the guideline on statistical principles for clinical trials." *Technical Report EMA/CHMP/ICH/436221/2017, Step 5 (17 February 2020).*

Imbens G, Rubin D (1997). "Bayesian inference for causal effects in randomized experiments with noncompliance." *The Annals of Statistics*, **25**(1), 305–327.

Li F, Ding P, Mealli F (2023). "Bayesian causal inference: a critical review." *Philosophical Transactions of the Royal Society A*, **381**(2247), 2022.0153.

Liu B, Wruck LM, Li F (2023). "Principal Stratification with Time-to-Event Outcomes." *Biometrics*, **in press**.

Long DM, Hudgens MG (2013). "Sharpening bounds on principal effects with covariates." *Biometrics*, **69**(4), 812–819.

Mcdonald C, Hui S, Tierney W (1992). "Effects of Computer Reminders of Influenza Vaccination on Morbidity during Influenza Epidemics." *M.D. computing : computers in medical practice*, **9**, 304–12.

Mealli F, Pacini B (2013). "Using secondary outcomes to sharpen inference in randomized experiments with noncompliance." *Journal of the American Statistical Association*, **108**, 1120–1131.

Neal RM (2011). *MCMC Using Hamiltonian Dynamics*, chapter 5. CRC Press.

Papadogeorgou G, Liu B, Li F, Li F (2023). "Addressing selection bias in cluster randomized experiments via weighting." *arXiv preprint arXiv:2309.07365.*

Rosenbaum P (1984). "The Consquences of Adjustment for a Concomitant Variable That Has Been Affected by the Treatment." *Journal of the Royal Statistical Society: Series B*, **147**(5), 656–666.

Rubin D (1984). "Bayesianly Justifiable and Relevant Frequency Calculations for the Applies Statistician." *The Annals of Statistics*, **12**(4), 1151–1172.

Rubin D (2006). "Causal inference through potential outcomes and principal stratification: application to studies with censoring due to death." *Statistical Science*, **91**, 299–321.

Stan Development Team (2022). "RStan: the R interface to Stan." R package version 2.21.7, URL https://mc-stan.org/.

Wei LJ (1992). "The accelerated failure time model: A useful alternative to the cox regression model in survival analysis." *Statistics in Medicine*, **11**(14-15), 1871–1879.

Zhang J, Rubin D, Mealli F (2009). "Likelihood-based analysis of the causal effects of job-training programs using principal stratification." *Journal of the American Statistical Association*, **104**, 166–176.

Affiliation:

Bo Liu, Fan Li
Department of Statistical Science
Duke University
Box 90251
Durham, NC 27705, United States of America
E-mail: bl226@duke.edu, fl35@duke.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd