Propensity Score Weighting in R: A Vignette

Tianhui Zhou* Duke University
Guangyu Tong* Yale University
Fan Li Duke University
Laine E. Thomas Duke University
Fan Li Yale University

Abstract

Propensity score weighting is an important tool for causal inference and comparative effectiveness research. Besides the inverse probability of treatment weights (IPW), recent development has introduced a general class of balancing weights, corresponding to alternative target populations and estimands. In particular, the overlap weights (OW) lead to optimal covariate balance and estimation efficiency, and a target population of scientific and policy interest. We develop the R package **PSweight** to provide a comprehensive design and analysis platform for causal inference based on propensity score weighting. **PSweight** supports (i) a variety of balancing weights, including OW, IPW, matching weights as well as optimal trimming, (ii) binary and multiple treatments, (iii) simple and augmented (doubly-robust) weighting estimators, (iv) nuisance-adjusted sandwich variances, and (v) ratio estimands for binary and count outcomes. **PSweight** also provides diagnostic tables and graphs for study design and covariate balance assessment. In addition, **PSweight** allows for propensity scores and outcome models to be estimated through machine learning methods including generalized boosted regression models and super learner, or other estimates obtained by users. We demonstrate the functionality of the package using a data example from the National Child Development Survey (NCDS), where we evaluate the causal effect of educational attainment on income.

Keywords: Causal inference, Propensity score, Weighting, Multiple treatments, Optimal trimming.

1. Introduction

Propensity score is one of the most widely used causal inference methods for observational studies (Rosenbaum and Rubin 1983). Propensity score methods include weighting, matching, stratification, regression, and mixed methods such as the augmented weighting estimators. The **PSweight** package provides an analysis pipeline for causal inference with propensity score weighting (Robins, Rotnitzky, and Zhao 1994; Robins, Hernán, and Brumback 2000; Hirano and Imbens 2001; Hirano, Imbens, and Ridder 2003; Lunceford and Davidian 2004; Li, Morgan, and Zaslavsky 2018). There are a number of existing R packages on propensity score

*T. Zhou and G. Tong contributed equally to this work.

weighting (see Table 1). Comparing to those, **PSweight** offers three major advantages: it incorporates (i) visualization and diagnostic tools of checking covariate overlap and balance, (ii) a general class of balancing weights, including overlap weights, inverse probability of treatment weights, and trimming, and (iii) multiple treatments. More importantly, **PSweight** comprises a wide range of functionalities, whereas each of the competing packages only supports a subset of these functionalities. As such, **PSweight** is currently the most comprehensive platform for causal inference with propensity score weighting, offering analysts a one-stop shop for the design and analysis. Table 1 summarizes the key functionalities of **PSweight** in comparison to related existing R packages. We elaborate the main features of **PSweight** below.

PSweight facilitates better practices in the design stage of observational studies, an aspect that has not been sufficiently emphasized in related packages. Specifically, we provide a design module that facilitates visualizing overlap (also known as the positivity assumption) and evaluating covariate balance without access to the final outcome (Austin and Stuart 2015). When there is limited overlap, **PSweight** allows for symmetric propensity score trimming (Crump, Hotz, Imbens, and Mitnik 2009; Yoshida, Solomon, Haneuse, Kim, Patorno, Tedeschi, Lyu, Franklin, Hernández-Díaz, and Glynn 2018) and optimal trimming (Crump *et al.* 2009; Yang, Imbens, Cui, Faries, and Kadziola 2016) to improve the internal validity. We extend the class of balance metrics suggested in Austin and Stuart (2015) and Li, Thomas, and Li (2019) for binary treatments, and those in McCaffrey, Griffin, Almirall, Slaughter, Ramchand, and Burgette (2013) and Li and Li (2019) for multiple treatments. In addition, the design module helps describe the weighted target population by providing the information required in the standard "Table 1" of a clinical article.

In addition to the standard inverse probability of treatment weights (IPW), **PSweight** implements the average treatment effect among the treated (Treated) weights, overlap weights (OW), matching weights (MW) and entropy weights (EW) for both binary (Li and Greene 2013; Mao, Li, and Greene 2018; Li *et al.* 2018; Zhou, Matsouaka, and Thomas 2020) and multiple treatments (Yoshida, Hernández-Díaz, Solomon, Jackson, Gagne, Glynn, and Franklin 2017; Li and Li 2019). All weights are members of the family of balancing weights (Li *et al.* 2018); the last three types of weights target at the subpopulation with improved overlap in the covariates between (or across) treatment groups, similar to the target population in randomized controlled trials (Thomas, Li, and Pencina 2020a,b). Among them, OW achieves optimal balance and estimation efficiency (Li *et al.* 2018, 2019). We also implement the augmented weighting estimators corresponding to each of the above weighting schemes (Mao *et al.* 2018). By default, **PSweight** employs parametric regression models to estimate propensity scores and potential outcomes. Nonetheless, it also allows for propensity scores to be estimated by external machine learning methods including generalized boosted regression models (McCaffrey *et al.* 2013) and super learner (Van der Laan, Polley, and Hubbard 2007), or importing any other propensity or outcome model estimates of interest, such as those via the covariate-balancing propensity score (Imai and Ratkovic 2014).

To our knowledge, **PSweight** is the first R package to accommodate a variety of balancing weighting schemes with multiple treatments. Existing R packages such as **twang** (Ridgeway *et al.* 2020), **CBPS** (Fong *et al.* 2019), **optweight** (Greifer 2019) have also implemented weighting-based estimation with multiple treatments, but focus on IPW. The **PSW** R package (Mao and Li 2018) implements both OW and MW and allows for nuisance-adjusted variance estimation, but it is only restricted to binary treatments.

To assist applied researchers to perform propensity score weighting analysis, this article pro-

Table 1: Comparisons of existing R packages that implement propensity score weighting with discrete treatments. Binary treatments and additive estimands are implemented in all packages, and therefore those two columns are omitted.

	Multiple treatments	Balance diagnostics	IPW/ATT weights	OW/other weights	Ratio estimands	Augmented weighting	Nuisance-adj variance	Optimal trimming
PSweight	✓	✓	✓	✓	✓	✓	✓	✓
twang	✓	✓	✓	×	×	×	×	×
CBPS	✓	✓	✓	×	×	✓	✓	×
PSW	×	✓	✓	✓	✓	✓	✓	×
optweight	✓	×	✓	×	×	×	×	×
ATE	✓	✓	✓	×	×	×	✓	×
WeightIt	✓	×	✓	✓	×	×	×	×
causalweight	✓	×	✓	×	×	✓	×	×
sbw	×	✓	✓	×	×	×	×	×

✓ indicates that the functionality is currently implemented in the package; × indicates otherwise.
References: **twang** (Version 1.6): Ridgeway, McCaffrey, Morral, Griffin, Burgette, and Cefalu (2020); **CBPS** (Version 0.21): Fong, Ratkovic, and Imai (2019); **PSW** (Version 1.1-3): Mao and Li (2018); **optweight** (Version 0.2.5): Greifer (2019); **ATE** (Version 0.2.0): Haris and Chan (2015); **WeightIt** (Version 0.10.2): Greifer (2020); **causalweight** (Version 0.2.1): Bodory and Huber (2020); **sbw** (Version 1.1.1): Zubizarreta and Li (2020).

vides a comprehensive illustration of the **PSweight** package. In Section 2, we explain the methodological foundation of **PSweight**. Section 3 outlines the main functions and their arguments. Section 4 illustrates the use of these functions with a data example that studies the causal effect of educational attainment on income. Section 5 concludes with a short discussion and outlines future development.

2. Overview of Propensity Score Weighting

Before diving into the implementation details of **PSweight**, we briefly introduce the basics of the propensity score weighting framework.

2.1. Binary Treatments

Additive Causal Estimands

Assume we have an observational study with N units. Each unit i ($i = 1, 2, \ldots, N$) has a binary treatment indicator Z_i ($Z_i = 0$ for control and $Z_i = 1$ for treated), a vector of p covariates $\boldsymbol{X}_i = (X_{1i}, \cdots, X_{pi})$. For each unit i, we assume a pair of potential outcomes $\{Y_i(1), Y_i(0)\}$ mapped to the treatment and control status, of which only the one corresponding to the observed treatment is observed, denoted by $Y_i = Z_i Y_i(1) + (1 - Z_i) Y_i(0)$; the other potential outcome is counterfactual.

Causal effects are contrasts of the potential outcomes of the same units in a *target population*, which usually is the population of a scientific interest (Thomas *et al.* 2020b). **PSweight** incorporates a general class of weighted average treatment effect (WATE) estimands. Specifically, assume the observed sample is drawn from a probability density $f(\boldsymbol{x})$, and let $g(\boldsymbol{x})$ denote the covariate density of the target population. The ratio $h(\boldsymbol{x}) \propto g(\boldsymbol{x})/f(\boldsymbol{x})$ is called the *tilting*

function, which adjusts the distribution of the observed sample to represent the target population. Denote the conditional expectation of the potential outcome by $m_z(\boldsymbol{x}) = \mathbb{E}[Y(z)|\boldsymbol{X} = \boldsymbol{x}]$ for $z = 0, 1$. Then, we can represent the average treatment effect over the target population by a WATE estimand:

$$\tau^h = \mathbb{E}_g[Y(1) - Y(0)] = \frac{\mathbb{E}\{h(\boldsymbol{x})(m_1(\boldsymbol{x}) - m_0(\boldsymbol{x}))\}}{\mathbb{E}\{h(\boldsymbol{x})\}}. \tag{2.1}$$

To estimate (2.1), **PSweight** maintains two standard assumptions: (1) *unconfoundedness*: $\{Y(1), Y(0)\} \perp Z \mid \boldsymbol{X}$; (2) *overlap*: $0 < P(Z = 1|\boldsymbol{X}) < 1$. The propensity score is the probability of a unit being assigned to the treatment group given the covariates (Rosenbaum and Rubin 1983): $e(\boldsymbol{x}) = P(Z = 1|\boldsymbol{X} = \boldsymbol{x})$. While assumption (1) is generally untestable and critically depends on substantive knowledge, assumption (2) can be checked visually from data by comparing the distribution of propensity scores between treatment and control groups.

Balancing Weights

For a given tilting function $h(\boldsymbol{x})$ (and correspondingly a WATE estimand τ^h), we can define the *balancing weights* (w_1, w_0) for the treated and control units: $w_1(\boldsymbol{x}) \propto h(\boldsymbol{x})/e(\boldsymbol{x})$ and $w_0(\boldsymbol{x}) \propto h(\boldsymbol{x})/\{1 - e(\boldsymbol{x})\}$. These weights balance the covariate distributions between the treated and control groups towards the target population (Li *et al.* 2018). **PSweight** implements the following Hájek estimator for WATE:

$$\hat{\tau}^h = \hat{\mu}_1^h - \hat{\mu}_0^h = \frac{\sum_{i=1}^N w_1(\boldsymbol{x}_i) Z_i Y_i}{\sum_{i=1}^N w_1(\boldsymbol{x}_i) Z_i} - \frac{\sum_{i=1}^N w_0(\boldsymbol{x}_i)(1 - Z_i) Y_i}{\sum_{i=1}^N w_0(\boldsymbol{x}_i)(1 - Z_i)}, \tag{2.2}$$

where the weights are calculated based on the propensity scores estimated from the data. Clearly, specification of $h(\boldsymbol{x})$ defines the target population and estimands. **PSweight** primarily implements the following three types of balancing weights (see Table 2 for a summary):

- *Inverse probability of treatment weights* (IPW) (Horvitz and Thompson 1952; Robins *et al.* 2000), whose target population is the combined treatment and control group represented by the observed sample, and the target estimand is the average treatment effect among the combined population (ATE).

- *Treated weights* (Hirano and Imbens 2001), whose target population is the treated group, and target estimand is the average treatment effect for the treated population (ATT). Treated weights can be viewed as a special case of IPW because it inversely weights the control group.

- *Overlap weights* (OW) (Li *et al.* 2018; Li and Li 2019), whose target population is the subpopulation with the most overlap in the observed covariates between treatment and group groups . In medicine this is known as the population in clinical equipoise and is the population eligible to be enrolled in randomized clinical trials. The target estimand of OW is the average treatment effect for the overlap population (ATO).

IPW has been the dominant weighting method in the literature, but has a well-known shortcoming of being sensitive to extreme propensity scores, which induces bias and large variance in estimating treatment effects. OW addresses the conceptual and operational problems of

Table 2: Target populations, tilting functions, estimands and the corresponding balancing weights for binary treatments in **PSweight**.

Target population	Tilting function $h(\boldsymbol{x})$	Estimand	Balancing weights (w_1, w_0)
Combined	1	ATE	$\left(\frac{1}{e(\boldsymbol{x})}, \frac{1}{1-e(\boldsymbol{x})}\right)$
Treated	$e(\boldsymbol{x})$	ATT	$\left(1, \frac{e(\boldsymbol{x})}{1-e(\boldsymbol{x})}\right)$
Overlap	$e(\boldsymbol{x})(1-e(\boldsymbol{x}))$	ATO	$(1-e(\boldsymbol{x}), e(\boldsymbol{x}))$
Matching	$\xi_1(\boldsymbol{x})$	ATM	$\left(\frac{\xi_1(\boldsymbol{x})}{e(\boldsymbol{x})}, \frac{\xi_1(\boldsymbol{x})}{1-e(\boldsymbol{x})}\right)$
Entropy	$\xi_2(\boldsymbol{x})$	ATEN	$\left(\frac{\xi_2(\boldsymbol{x})}{e(\boldsymbol{x})}, \frac{\xi_2(\boldsymbol{x})}{1-e(\boldsymbol{x})}\right)$

Notes: $\xi_1(\boldsymbol{x}) = \min\{e(\boldsymbol{x}), 1-e(\boldsymbol{x})\}$ and $\xi_2(\boldsymbol{x}) = -\{e(\boldsymbol{x})\log(e(\boldsymbol{x}))+(1-e(\boldsymbol{x}))\log(1-e(\boldsymbol{x}))\}$.

IPW. Among all balancing weights, OW leads to the smallest asymptotic (and often finite-sample) variance of the weighting estimator (2.2). (Li *et al.* 2018, 2019). Recent simulations also show that OW provides more stable causal estimates under limited overlap (Li *et al.* 2019; Mao *et al.* 2018; Yoshida *et al.* 2017, 2018), and is more robust to misspecification of the propensity score model (Zhou *et al.* 2020).

PSweight implements two additional types of balancing weights: matching weights (MW) (Li and Greene 2013), and entropy weights (EW) (Zhou *et al.* 2020). Similar to OW, MW and EW focus on target populations with substantial overlap between treatment groups. Though having similar operating characteristics, MW and EW do not possess the same theoretical optimality as OW, and are less used in practice. Therefore, we will not separately describe MW and EW hereafter.

Covariate Balance Check

In observational studies, propensity scores are generally unknown and need to be estimated. Therefore, propensity score analysis usually involves two steps: (1) estimating the propensity scores, and (2) estimating the causal effects based on the estimated propensity scores. In **PSweight**, the default model for estimating propensity scores with binary treatments is a logistic regression model. Spline or polynomial models can be easily incorporated by adding `bs()`, `ns()` or `poly()` terms into the model formula. **PSweight** also allows for importing propensity scores estimated from external routines, such as boosted models or super learner (Section 4.4).

Goodness-of-fit of the propensity score model is usually assessed based on the resulting covariate balance. In the context of propensity score weighting, this is measured based on either the absolute standardized difference (ASD):

$$\text{ASD} = \left| \frac{\sum_{i=1}^N w_1(\boldsymbol{x}_i) Z_i X_{pi}}{\sum_{i=1}^N w_1(\boldsymbol{x}_i) Z_i} - \frac{\sum_{i=1}^N w_0(\boldsymbol{x}_i)(1-Z_i) X_{pi}}{\sum_{i=1}^N w_0(\boldsymbol{x}_i)(1-Z_i)} \right| \Bigg/ \sqrt{\frac{s_1^2 + s_0^2}{2}}, \tag{2.3}$$

or the target population standardized difference (PSD), $\max\{\text{PSD}_0, \text{PSD}_1\}$, where

$$\text{PSD}_z = \left| \frac{\sum_{i=1}^N w_z(\boldsymbol{x}_i) \mathbb{1}\{Z_i = z\} X_{pi}}{\sum_{i=1}^N w_z(\boldsymbol{x}_i) \mathbb{1}\{Z_i = z\}} - \frac{\sum_{i=1}^N h(\boldsymbol{x}_i) X_{pi}}{\sum_{i=1}^N h(\boldsymbol{x}_i)} \right| \Bigg/ \sqrt{\frac{s_1^2 + s_0^2}{2}}. \tag{2.4}$$

In (2.3) and (2.4), s_z^2 is the variance (either unweighted or weighted, depending on user specification) of the pth covariate in group z, and (w_0, w_1) are the specified balancing weights. Setting $w_0 = w_1 = 1$ corresponds to the unweighted mean differences. ASD and PSD are often displayed as column in the baseline characteristics table (known as the "Table 1") and visualized via a Love plot (also known as a forest plot) (Greifer 2018). A rule of thumb for determining adequate balance is when ASD of all covariates is controlled within 0.1 (Austin and Stuart 2015).

2.2. Multiple Treatments

Li and Li (2019) extend the framework of balancing weights to multiple treatments. Assume that we have J ($J \geq 3$) treatment groups, and let Z_i stand for the treatment received by unit i, $Z_i \in \{1, \ldots, J\}$. We further define $D_{ij} = \mathbb{1}\{Z_i = j\}$ as a set of multinomial indicator, satisfying $\sum_{i=1}^{J} D_{ij} = 1$ for all j. Denote the potential outcome for unit i under treatment j as $Y_i(j)$, of which only the one corresponding to the received treatment, $Y_i = Y_i(Z_i)$, is observed. The generalized propensity score is the probability of receiving a potential treatment j given $\boldsymbol{X}$ (Imbens 2000): $e_j(\boldsymbol{x}) = P(Z = j|\boldsymbol{X} = \boldsymbol{x})$, with the constraint that $\sum_{j=1}^{J} e_j(\boldsymbol{x}) = 1$.

To define the target estimand, let $m_j(\boldsymbol{x}) = \mathbb{E}[Y(j)|\boldsymbol{X} = \boldsymbol{x}]$ be the conditional expectation of the potential outcome in group j. For specified tilting function $h(\boldsymbol{x})$ and target density $g(\boldsymbol{x}) \propto f(\boldsymbol{x})h(\boldsymbol{x})$, the jth average potential outcome among the target population is

$$\mu_j^h = \mathbb{E}_g[Y(j)] = \frac{\mathbb{E}\{h(\boldsymbol{x})m_j(\boldsymbol{x})\}}{\mathbb{E}\{h(\boldsymbol{x})\}}. \tag{2.5}$$

Causal estimands can then be constructed in a general manner as contrasts based on μ_j^h. For example, the most commonly seen estimands in multiple treatments are the pairwise average treatment effects between groups j and j': $\tau_{j,j'}^h = \mu_j^h - \mu_{j'}^h$. This definition can be generalized to arbitrary linear contrasts. Denote $\boldsymbol{a} = (a_i, \cdots, a_J)$ as a contrast vector of length J. A general class of additive estimands is

$$\tau^h(\boldsymbol{a}) = \sum_{j=1}^{J} a_j \mu_j^h. \tag{2.6}$$

Specific choices for $\boldsymbol{a}$ with nominal and ordinal treatments can be found in Li and Li (2019). Similar as before, propensity score weighting analysis with multiple treatments rests on two assumptions: (1) *weak unconfoundedness*: $Y(j) \perp \mathbb{1}\{Z = j\}|\boldsymbol{X}$, for all j, and (2) *Overlap*: the generalized propensity score is bounded away from 0 and 1: $0 < e_j(\boldsymbol{x}) < 1$, for all j.

With multiple treatments, the tilting function $h(\boldsymbol{x})$ specifies the target population, estimand, and balancing weights. For a given $h(\boldsymbol{x})$, the balancing weights for the jth treatment group $w_j(\boldsymbol{x}) \propto h(\boldsymbol{x})/e_j(\boldsymbol{x})$. Then the Hájek estimator for μ_j^h is

$$\hat{\mu}_j^h = \frac{\sum_{i=1}^{N} w_j(\boldsymbol{x}_i) D_{ij} Y_i}{\sum_{i=1}^{N} w_j(\boldsymbol{x}_i) D_{ij}}. \tag{2.7}$$

Contrasts based on $\hat{\mu}_j^h$ can be obtained for any $\boldsymbol{a}$ to estimate the additive causal estimand $\tau^h(\boldsymbol{a})$. Of note, we only consider types of estimands that are transitive, and therefore the ATT estimands introduced in Lechner (2001) is not implemented. In parallel to binary treatments

PSweight implements five types of balancing weights with multiple treatments: IPW, treated weights, OW, MW, and EW, and the corresponding target estimand of each weighting scheme is its pairwise (between each pair of treatments) counterpart in binary treatments. Among all the weights, OW minimizes the total asymptotic variances of all pairwise comparisons, and has been shown to have the best finite-sample efficiency in estimating pairwise WATEs (Li and Li 2019). Table 3 summarizes the target population, tilting function and balancing weight for multiple treatments that are available in **PSweight**.

Table 3: Target populations, tilting functions, and the corresponding balancing weights for multiple treatments in **PSweight**.

Target population	Tilting function $h(\boldsymbol{x})$	Balancing weights $\{w_j(\boldsymbol{x}),\ j = 1, \ldots, J\}$
Combined	1	$\{1/e_j(\boldsymbol{x})\}$
Treated (j'th group)	$e_{j'}(\boldsymbol{x})$	$\{e_{j'}(\boldsymbol{x})/e_j(\boldsymbol{x})\}$
Overlap	$\{\sum_{k=1}^{J} 1/e_k(\boldsymbol{x})\}^{-1}$	$\left\{\{\sum_{k=1}^{J} 1/e_k(\boldsymbol{x})\}^{-1}/e_j(\boldsymbol{x})\right\}$
Matching	$\min_k\{e_k(\boldsymbol{x})\}$	$\{\min_k\{e_k(\boldsymbol{x})\}/e_j(\boldsymbol{x})\}$
Entropy	$-\sum_{k=1}^{J} e_k(\boldsymbol{x}) \log\{e_k(\boldsymbol{x})\}$	$\left\{-\sum_{k=1}^{J} e_k(\boldsymbol{x}) \log\{e_k(\boldsymbol{x})\}/e_j(\boldsymbol{x})\right\}$

To estimate the generalized propensity scores for multiple treatments, the default model in **PSweight** is a multinomial logistic model. **PSweight** also allows for externally estimated generalized propensity scores. Goodness-of-fit of the generalized propensity score model is assessed by the resulting covariate balance, which is measured by the pairwise versions of the ASD and PSD. The detailed formula of these metrics can be found in Li and Li (2019). A common threshold for balance is that the maximum pairwise ASD or maximum PSD is below 0.1.

2.3. Propensity Score Trimming

Propensity score trimming excludes units with estimated (generalized) propensity scores close to zero (or one). It is a popular approach to address the extreme weights problem of IPW. **PSweight** implements the symmetric trimming rules in Crump *et al.* (2009) and Yoshida *et al.* (2018). Operationally, we allow users to specify a single cutoff δ on the estimated generalized propensity scores, and only includes units for analysis if $\min_j\{e_j(\boldsymbol{x})\} \in [\delta, 1]$. With binary treatments, the symmetric trimming rule reduces to $e(\boldsymbol{x}) \in [\delta, 1-\delta]$. The natural restriction $\delta < 1/J$ must be satisfied due to the constraint $\sum_{j=1}^{J} e_j(\boldsymbol{x}) = 1$. To avoid specifying an arbitrary trimming threshold δ, **PSweight** also implements the optimal trimming rules of Crump *et al.* (2009) and Yang *et al.* (2016), which minimizes the (total) asymptotic variance(s) for estimating the (pairwise) ATE among the class of all trimming rules. OW can be viewed as a continuous version of trimming because it smoothly down-weigh the units with propensity scores close to 0 or 1, and thus avoids specifying a threshold.

2.4. Augmented Weighting Estimators

PSweight also implements augmented weighting estimators, which augment a weighting estimator by an outcome regression and improves the efficiency. With IPW, the augmented weighting estimator is known as the doubly-robust estimator (Lunceford and Davidian 2004;

Bang and Robins 2005; Funk, Westreich, Wiesen, Stürmer, Brookhart, and Davidian 2011). With binary treatments, the augmented estimator with general balancing weights are discussed Hirano *et al.* (2003) and Mao *et al.* (2018). Below, we briefly outline the form of this estimator with multiple treatments. Recall the conditional mean of $Y_i(j)$ given $\boldsymbol{X}_i$ and treatment $Z_i = j$ as $m_j(\boldsymbol{x}_i) = \mathbb{E}[Y_i(j)|\boldsymbol{X}_i = \boldsymbol{x}_i] = \mathbb{E}[Y_i|\boldsymbol{X}_i = \boldsymbol{x}_i, Z_i = j]$. This conditional mean can be estimated by generalized linear models, kernel estimators, or machine learning models. **PSweight** by default employs the generalized linear models, but also allows estimated values from other routines. When $m_j(\boldsymbol{x}_i)$ is estimated by generalized linear models, **PSweight** currently accommodates three types of outcomes: continuous, binary and count outcomes (with or without an offset), using the canoncal link function.

With a pre-specified tilting function, the augmented weighting estimator for group j is

$$\hat{\mu}_j^{h,\text{aug}} = \frac{\sum_{i=1}^N w_j(\boldsymbol{x}_i) D_{ij}\{Y_i - m_j(\boldsymbol{x}_i)\}}{\sum_{i=1}^N w_j(\boldsymbol{x}_i) D_{ij}} + \frac{\sum_{i=1}^N h(\boldsymbol{x}_i) m_j(\boldsymbol{x}_i)}{\sum_{i=1}^N h(\boldsymbol{x}_i)}. \tag{2.8}$$

The first term of (2.8) is the Hájek estimator of the regression residuals, and the second term is the standardized average potential outcome (a *g*-formula estimator). With IPW, (2.8) is consistent to $\mathbb{E}[Y(j)]$ when either the propensity score model or the outcome model is correctly specified, but not necessarily both. For other balancing weights, (2.8) is consistent to the WATE when the propensity model is correctly specified, regardless of outcome model specification. When both models are correctly specified, (2.8) achieves the lower bound of the variance for regular and asymptotic linear estimators (Robins *et al.* 1994; Hirano *et al.* 2003; Mao *et al.* 2018).

2.5. Ratio Causal Estimands

With binary and count outcomes, ratio causal estimands are often of interest. Using notation from the multiple treatments as an example, once we use weighting to obtain estimates for the set of average potential outcomes $\{\mu_j^h, j = 1, \ldots, J\}$, we can directly estimate the causal relative risk (RR) and causal odds ratio (OR), defined as

$$\tau_{j,j'}^{h,\text{RR}} = \frac{\mu_j^h}{\mu_{j'}^h}, \qquad \tau_{j,j'}^{h,\text{OR}} = \frac{\mu_j^h/(1-\mu_j^h)}{\mu_{j'}^h/(1-\mu_{j'}^h)}. \tag{2.9}$$

Here the additive estimand $\tau_{j,j'}^{h,\text{RD}} = \mu_j^h - \mu_{j'}^h$ is the causal risk difference (RD). **PSweight** supports a class of ratio estimands for any given contrasts $\boldsymbol{a}$. Specifically, we define the log-RR type parameters by

$$\lambda^{h,\text{RR}}(\boldsymbol{a}) = \sum_{j=1}^J a_j \log\left(\mu_j^h\right), \tag{2.10}$$

and the log-OR type parameters by

$$\lambda^{h,\text{OR}}(\boldsymbol{a}) = \sum_{j=1}^J a_j \left\{\log\left(\mu_j^h\right) - \log\left(1-\mu_j^h\right)\right\}. \tag{2.11}$$

With nominal treatments, the contrast vector $\boldsymbol{a}$ can be specified to encode pairwise comparisons in the log scale (as in (2.10)) or in the log odds scale (as in (2.11)), in which

case $\exp\{\lambda^{h,\mathrm{RR}}(\boldsymbol{a})\}$ and $\exp\{\lambda^{h,\mathrm{OR}}(\boldsymbol{a})\}$ become the causal RR and causal OR in (2.9). User-specified contrasts $\boldsymbol{a}$ can provide a variety of nonlinear estimands. For example, when $J = 3$, with $\boldsymbol{a} = (1, -2, 1)^T$ one can use **PSweight** to assess the equality of two consecutive causal RR: $H_0 : \mu_3^h/\mu_2^h = \mu_2^h/\mu_1^h$.

2.6. Variance and Interval Estimation

Empirical Sandwich Variance

PSweight by default implements the empirical sandwich variance for propensity score weighting estimators (Lunceford and Davidian 2004; Li *et al.* 2019; Mao *et al.* 2018) based on the M-estimation theory (Stefanski and Boos 2002). The variance adjusted for the uncertainty in estimating the propensity score and outcome models, and are sometime referred to as the nuisance-adjusted sandwich variance. Below we illustrate the main steps with multiple treatments and general balancing weights. Write $\boldsymbol{\theta} = \left(\nu_1, \ldots, \nu_J, \eta_1, \ldots, \eta_J, \boldsymbol{\beta}^T, \boldsymbol{\alpha}^T\right)^T$ as the collection of parameters to be estimated. Then $\left\{\hat{\mu}_j^{h,\mathrm{aug}} = \hat{\nu}_j + \hat{\eta}_j : j = 1, \ldots, J\right\}$ jointly solve

$$\sum_{i=1}^{N} \Psi_i(\boldsymbol{\theta}) = \sum_{i=1}^{N} \begin{pmatrix} w_1(\boldsymbol{x}_i) D_{i1}\{Y_i - m_1(\boldsymbol{x}_i;\boldsymbol{\alpha}) - \nu_1\} \\ \vdots \\ w_J(\boldsymbol{x}_i) D_{iJ}\{Y_i - m_J(\boldsymbol{x}_i;\boldsymbol{\alpha}) - \nu_J\} \\ h(\boldsymbol{x}_i)\{m_1(\boldsymbol{x}_i;\boldsymbol{\alpha}) - \eta_1\} \\ \vdots \\ h(\boldsymbol{x}_i)\{m_J(\boldsymbol{x}_i;\boldsymbol{\alpha}) - \eta_J\} \\ S_{\boldsymbol{\beta}}(Z_i, \boldsymbol{x}_i; \boldsymbol{\beta}) \\ S_{\boldsymbol{\alpha}}(Y_i, Z_i, \boldsymbol{x}_i; \boldsymbol{\alpha}) \end{pmatrix} = \mathbf{0},$$

where $S_{\boldsymbol{\beta}}(Z_i, \boldsymbol{x}_i; \boldsymbol{\beta})$ and $S_{\boldsymbol{\alpha}}(Y_i, Z_i, \boldsymbol{x}_i; \boldsymbol{\alpha})$ are the score functions of the propensity score model and the outcome model. The empirical sandwich variance estimator is

$$\widehat{\mathbb{V}}(\hat{\boldsymbol{\theta}}) = \left\{\sum_{i=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}^T} \Psi_i(\hat{\boldsymbol{\theta}})\right\}^{-1} \left\{\sum_{i=1}^{N} \Psi_i(\hat{\boldsymbol{\theta}}) \Psi_i^T(\hat{\boldsymbol{\theta}})\right\} \left\{\sum_{i=1}^{N} \frac{\partial}{\partial \boldsymbol{\theta}} \Psi_i^T(\hat{\boldsymbol{\theta}})\right\}^{-1}.$$

Because $\hat{\mu}_j^{h,\mathrm{aug}} = \hat{\nu}_j + \hat{\eta}_j$, the variance of arbitrary linear contrasts based on the average potential outcomes can be easily computed by applying the Delta method to the joint variance $\widehat{\mathbb{V}}(\hat{\boldsymbol{\theta}})$. For the Hájek weighting estimators, variance is estimated by removing $S_{\boldsymbol{\alpha}}(Y_i, Z_i, \boldsymbol{x}_i; \boldsymbol{\alpha})$ as well as the components involving $m_j(\boldsymbol{x}_i; \boldsymbol{\alpha})$ in $\Psi_i(\boldsymbol{\theta})$. Finally, when propensity scores and potential outcomes are not estimated through the generalized linear model or are supplied externally, or MW are used (since the tilting function is not everywhere differentiable), **PSweight** ignores the uncertainty in estimating $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ and removes $S_{\boldsymbol{\beta}}(Z_i, \boldsymbol{x}_i; \boldsymbol{\beta})$ and $S_{\boldsymbol{\alpha}}(Y_i, Z_i, \boldsymbol{x}_i; \boldsymbol{\alpha})$ in $\Psi_i(\boldsymbol{\theta})$ in the calculation of the empirical sandwich variance. Based on the estimated variance, **PSweight** computes the associated symmetric confidence intervals and p-values via the normal approximation.

For ratio causal estimands, **PSweight** applies the logarithm transformation to improve the accuracy of the normal approximation (Agresti 2003). For estimating the variance of causal RR,

we first obtain the joint variance of $\left(\log\left(\hat{\mu}_1^{h,\text{aug}}\right), \ldots \log\left(\hat{\mu}_J^{h,\text{aug}}\right)\right)^T$ using the Delta method, and then estimate the variance of $\lambda^{h,\text{RR}}(\boldsymbol{a})$. Once the symmetric confidence intervals are obtained for $\lambda^{h,\text{RR}}(\boldsymbol{a})$ using the normal approximation, we can exponentiate the upper and lower confidence limits to derive the asymmetric confidence intervals for the causal RR. Confidence intervals for the causal OR are computed similarly.

Bootstrap Variance

PSweight also allows using bootstrap to estimate variances, which can be much more computationally intensive than the closed-form sandwich estimator but sometimes give better finite-sample performance in small samples. By default, **PSweight** resamples $R = 50$ bootstrap replicates with replacement. For each replicate, the weighting estimator (2.7) or the augmented weighting estimtor (2.8) is implemented, providing R estimates of the J average potential outcomes (an $R \times J$ matrix). Then for any contrast vector $\boldsymbol{a} = (a_1, \cdots, a_J)^T$, **PSweight** obtains R bootstrap estimates:

$$\hat{\mathbb{T}}^h(\boldsymbol{a})_{bootstrap} = \left\{ \hat{\tau}^h(\boldsymbol{a})_1 = \sum_{j=1}^{J} a_j \hat{\mu}_{j,1}^h, \ \ldots \ , \hat{\tau}^h(\boldsymbol{a})_R = \sum_{j=1}^{J} a_j \hat{\mu}_{j,R}^h \right\}.$$

The sample variance of $\hat{\mathbb{T}}^h(\boldsymbol{a})_{bootstrap}$ is reported by **PSweight** as the bootstrap variance; the lower and upper 2.5% quantiles of $\hat{\mathbb{T}}^h(\boldsymbol{a})_{bootstrap}$ form the 95% bootstrap interval estimate.

2.7. Covariate Adjustment in Randomized Trials

Although propensity score weighting has been largely developed in observational studies, it is also an important tool for covariate adjustment in randomized controlled trials (RCTs). Williamson, Forbes, and White (2014) showed that IPW can reduce the variance of the unadjusted difference-in-means treatment effect estimator in RCTs, and Shen, Li, and Li (2014) proved that the IPW estimator is semiparametric efficient and asymptotically equivalent to the analysis of covariance (ANCOVA) estimator (Tsiatis, Davidian, Zhang, and Lu 2008). Zeng, Li, Wang, and Li (2020) generalized these results of IPW to the family of balancing weights. Operationally, there is no difference in implementing propensity score weighting between RCTs and observational studies. Therefore, **PSweight** is directly applicable to perform covariate-adjusted analysis in RCTs.

3. Overview of Package

The **PSweight** package includes two modules tailored for design and analysis of observational studies. The design module provides diagnostics to assess the adequacy of the propensity score model and the weighted target population, prior to the use of outcome data. The analysis module provides functions to estimate the causal estimands discussed in Section 2. We briefly describe the two modules below.

3.1. Design Module

PSweight offers the `SumStat()` function to visualize the distribution of the estimated propensity scores, to assess the balance of covariates under different weighting schemes, and to

characterize the weighted target population. It uses the following code snippet:

```
SumStat(ps.formula, ps.estimate = NULL, trtgrp = NULL, Z = NULL, covM = NULL,
+     zname = NULL, xname = NULL, data = NULL, weight = "overlap", delta = 0,
+     method = "glm", ps.control = list())
```

By default, the (generalized) propensity scores are estimated by the (multinomial) logistic regression, through the argument `ps.formula`. Alternatively, `gbm()` functions in the **gbm** package (Greenwell, Boehmke, Cunningham, and Developers 2019) or the `SuperLearner()` function in the **SuperLearner** package (Polley, LeDell, Kennedy, and van der Laan 2019) can also be called by using `method = "gbm"` or `method = "SuperLearner"`. Additional parameters of those functions can be supplied through the `ps.control` argument. The argument `ps.estimate` supports estimated propensity scores from external routines. `SumStat()` produces a `SumStat` object, with estimated propensity scores, unweighted and weighted covariate means for each treatment group, balance diagnostics, and effective sample sizes (defined in Li and Li (2019)). We then provide a `summary.SumStat()` function, which takes the `SumStat` object and summarizes weighted covariate means by treatment groups and the between-group differences in either ASD or PSD. The default options in `weighted.var = TRUE` and `metric = "ASD"` yield ASD based on weighted standard deviations in Austin and Stuart (2015). The weighted covariate means can be used to build a baseline characteristics "Table 1" to illustrate the target population where trimming or balancing weights are applied.

```
summary(object, weighted.var = TRUE, metric = "ASD")
```

Table 4: Functions in the design module of **PSweight**.

Function	Description
`SumStat()`	Generate a `SumStat` object with information of propensity scores and weighted covariate balance
`summary.SumStat()`	Summarize the `SumStat` object and return weighted covariate means by treatment groups and weighted or unweighted between-group differences in ASD or PSD
`plot.SumStat()`	Plot the distribution of propensity scores or weighted covariate balance metrics from the `SumStat` object
`PStrim()`	Trim the data set based on estimated propensity scores

Diagnostics of propensity score models can be visualized with the `plot.SumStat()` function. It takes the `SumStat` object and produces a balance plot (`type = "balance"`) based on the ASD and PSD. A vertical dashed line can be set by the `threshold` argument, with a default value equal to 0.1. The `plot.SumStat()` function can also supply density plot (`type = "density"`), or histogram (`type = "hist"`) of the estimated propensity scores. The histogram, however, is only available for the binary treatment case. The plot function is implemented as follows:

```
plot(x, type = "balance", weighted.var = TRUE, threshold = 0.1,
+     metric = "ASD")
```

In the design stage, propensity score trimming can be carried out with the `PStrim()` function. The trimming threshold `delta` is set to 0 by default. `PStrim()` also enables optimal trimming rules (`optimal = TRUE`) that give the most statistically efficient (pairwise) subpopulation

ATE, among all possible trimming rules. A trimmed data set along with a summary of trimmed cases will be returned by `PStrim()`. This function is given below:

```
PStrim(data, ps.formula = NULL, zname = NULL, ps.estimate = NULL,
+     delta = 0, optimal = FALSE, method = "glm", ps.control = list())
```

Alternatively, trimming is also anchored in the `SumStat()` function with the `delta` argument. All functions in the design module are summarized in Table 4.

3.2. Analysis Module

The analysis module of **PSweight** includes two functions: `PSweight()` and `summary.PSweight()`. The `PSweight()` function estimates the average potential outcomes in the target population, $\{\mu_j^h, j = 1, \ldots, J\}$, and the associated variance-covariance matrix. By default, the empirical sandwich variance is implemented, but bootstrap variance can be obtained with the argument `bootstrap = TRUE)`. The `weight` argument can take `"IPW"`, `"treated"`, `"overlap"`, `"matching"` or `"entropy"`, corresponding to the weights introduced in Section 2. More detailed descriptions of each input argument in the `PSweight()` function can be found in Table 5. A typical `PSweight()` code snippet looks like

```
PSweight(ps.formula, ps.estimate, trtgrp, zname, yname, data,
+     weight = "overlap", delta = 0, augmentation = FALSE, bootstrap = FALSE,
+     R = 50, out.formula = NULL, out.estimate = NULL, family = "gaussian",
+     ps.method = "glm", ps.control = list(), out.method = "glm",
+     out.control = list())
```

Similar to the design module, the `summary.PSweight()` function synthesizes information from the `PSweight` object for statistical inference. A typical code snippet looks like

```
summary(object, contrast, type = "DIF", CI = TRUE)
```

In the `summary.PSweight()` function, the argument `type` corresponds to the three types estimands: `type = "DIF"` is the default argument that specifies the additive causal contrasts; `type = "RR"` specifies the contrast on the log scale as in equation (2.10); `type = "OR"` specifies the contrast on the log odds scale as in equation (2.11). Confidence intervals and p-values are obtained using normal approximation and reported by the `summary.PSweight()` function. The argument `contrast` represents a contrast vector $\boldsymbol{a}$ or matrix with multiple contrast row vectors. If `contrast` is not specified, `summary.PSweight()` provides all pairwise comparisons of the average potential outcomes. By default, confidence interval is printed (`CI = TRUE`); alternatively, one can print the test statistics and p-values by `CI = FALSE`.

Table 5: Arguments for function `PSweight()` in the analysis module of **PSweight**.

Argument	Description	Default
`ps.formula`	A symbolic description of the propensity score model.	–
`ps.estimate`	An optional matrix or data frame with externally estimated (generalized) propensity scores for each observation; can also be a vector with binary treatments.	`NULL`
`trtgrp`	An optional character defining the *treated* population for estimating (pairwise) ATT. It can also be used to specify the treatment level when only a vector of values are supplied for `ps.estimate` in the binary treatment setting.	Last value in alphabetic order
`zname`	An optional character specifying the name of the treatment variable when `ps.formula` is not provided.	`NULL`
`yname`	A character specifying name of the outcome variable in `data`.	
`weight`	A character specifying the type of weights to be used.	`"overlap"`
`delta`	Trimming threshold for (generalized) propensity scores.	0
`augmentation`	Logical value of whether augmented weighting estimators should be used.	`FALSE`
`bootstrap`	Logical value of whether bootstrap is used to estimate the standard error	`FALSE`
`R`	Number of bootstrap replicates if `bootstrap = TRUE`	50
`out.formula`	A symbolic description of the outcome model to be estimated when `augmentation = TRUE`	
`out.estimate`	An optional matrix or data frame containing externally estimated potential outcomes for each observation under each treatment level.	`NULL`
`family`	A description of the error distribution and canonical link function to be used in the outcome model if `out.formula` is provided	`"gaussian"`
`ps.method`	a character to specify the method for propensity model.	`"glm"`
`ps.control`	A list to specify additional options when `method` is set to `"gbm"` or `"SuperLearner"`.	`list()`
`out.method`	A character to specify the method for outcome model.	`"glm"`
`out.control`	A list to specify additional options when `methodout` is set to `"gbm"` or `"SuperLearner"`.	`list()`

4. Case Study with the NCDS Data

We demonstrate **PSweight** in a case study that estimates the causal effect of educational attainment on hourly wage, based on the National Child Development Survey (NCDS) data. Section 4.1 gives an overview of the study. Section 4.2 and 4.3 provides propensity score weighting analyses of the average treatment effect in the context of a binary treatment and a tri-valued treatment, respectively. Section 4.4 demonstrates how the machine learning method for propensity scores and potential outcomes can be implemented in **PSweight**.

4.1. NCDS Data Overview

The National Child Development Survey (NCDS) is a longitudinal study on children born in the United Kingdom (UK) in 1958 [1]. NCDS collected information such as educational attainment, familial backgrounds, and socioeconomic and health well being on 17,415 individuals. We followed Battistin and Sianesi (2011) to pre-process the data and obtain a subset of 3,642 males employed in 1991 with complete educational attainment and wage information for analysis. For illustration, we use the Multiple Imputation by Chained Equations (MICE, Buuren and Groothuis-Oudshoorn 2010) to impute missing covariates and obtain a single imputed data set for all subsequent analysis.[2] The outcome variable `wage` is log of the gross hourly wage in Pound. The treatment variable is educational attainment. For the binary treatment case, we created `Dany` to indicate whether one had attained any academic qualification. There are 2399 individuals that attained academic qualification, and 1,243 individuals without any. For the multiple treatment case, we created `Dmult` with three levels: `">=A/eq"`, `"O/eq"` and `"None"`, representing advanced qualification (1,806 individuals), intermediate qualification (941 individuals) and no qualification (895 individuals). We consider twelve pre-treatment covariates or potential confounders. The variable `white` indicates whether an individual identified himself as white race; `scht` indicates the school type they attended at age 16; `qmab` and `qmab2` are math test scores at age 7 and 11; `qvab` and `qvab2` are two reading test scores at age 7 and 11; `sib_u` stands for the number of siblings; `agepa` and `agema` are the ages of parents in year 1974; in the same year, the employment status of mother `maemp` was also collected; `paed_u` and `maed_u` are the years of education for parents. Information on the study variables can be summarized using the `str()` function as below:

```
R> str(NDCS)

'data.frame':        3642 obs. of  16 variables:
 $ white  : int  1 1 1 1 1 1 1 1 1 1 ...
 $ wage   : num  2.57 2.04 1.72 2.2 2.48 ...
 $ Dany   : int  1 1 0 1 1 0 0 1 1 1 ...
 $ Dmult  : chr  ">=A/eq" ">=A/eq" "None" "O/eq" ...
 $ maemp  : int  0 0 0 0 1 1 0 1 0 1 ...
 $ scht   : int  2 1 1 3 1 2 1 1 1 3 ...
 $ qmab   : int  2 5 4 5 3 1 4 5 5 2 ...
```

[1] https://cls.ucl.ac.uk/cls-studies/1958-national-child-development-study/

[2] Ten out of twelve pre-treatment covariates we considered have missingness. The smallest missingness proportion is 4.9% and the largest missingness proportion is 17.2%. We considered one imputed complete data set for illustrative purposes, but note that a more rigorous analysis could proceed by combining analyses from multiple imputed data sets via the Rubin's rule.

```
$ qmab2  : int  2 5 4 4 3 1 1 4 4 5 ...
$ qvab   : int  1 5 4 4 2 2 2 4 3 4 ...
$ qvab2  : int  2 5 5 5 3 2 1 3 1 5 ...
$ paed_u : int  9 0 0 10 9 10 0 11 10 10 ...
$ maed_u : int  9 0 0 10 9 9 0 11 9 10 ...
$ agepa  : int  60 56 57 40 57 43 43 46 43 47 ...
$ agema  : int  59 56 53 41 45 42 38 45 43 40 ...
$ sib_u  : int  3 0 0 1 1 1 1 1 0 3 ...
$ wagebin: num  1 0 0 1 1 0 1 1 1 1 ...
```

4.2. Propensity Score Weighting with Binary Treatments

Estimating Propensity Scores and Balance Check

Suppose we wish to estimate the causal effect of whether the attainment of any academic qualification leads to higher hourly wage. Because the attainment of any academic qualification is not randomized, and may be affected by potential confounders, we specify the following propensity score model and carry out weighting analysis.

```
R> ps.any <- Dany ~ white + maemp + as.factor(scht) + as.factor(qmab)
+     as.factor(qmab2) + as.factor(qvab) + as.factor(qvab2) + paed_u + maed_u
+     agepa + agema + sib_u + paed_u * agepa + maed_u * agema
```

In addition to the main effects of covariates, we considered adjusting for the interaction between the ages of parents and their education following Battistin and Sianesi (2011). We use the `Sumstat()` function to estimate the logistic propensity score model and obtain balance statistics under three types of weighting schemes, IPW, the treated weights and OW.

```
R> bal.any <- SumStat(ps.formula = ps.any, data = NCDS,
+     weight = c("IPW", "overlap", "treated"))
```

The output on screen from the `Sumstat()` function is the choice of weights and the treatment group selected (`trtgrp`) only if `"treated"` is included in the `weight` argument. In this example, as `trtgrp` is unspecified, `Sumstats()` automatically takes the last level in alphabetic order of the treatment variable as the treatment group: `Dany = 1`.

```
R> bal.any

trt group for PS model is:  1
weights estimated for:  IPW overlap treated
```

The full return of `SumStat` is a list including the treatment group level (for defining ATT) (`"trtgrp"`), estimated propensity scores (`"propensity"`), estimated weight under each weighting scheme (`"ps.weights"`), effective sample size (`"ess"`) and balance statistics under each weighting scheme (e.g., `"unweighted.sumstat"`, `"IPW.sumstat"`, `"overlap.sumstat"`, `"treated.sumstat"`) . Further, the balance statistics for each weighting scheme includes both ASD and PSD, with both the unweighted or weighted standard deviation of the covariates.

The `plot.SumStat()` function visualizes the distributions of estimated propensity scores and covariate balance statistics. Specifying argument `type = "hist"` generates the histogram of estimated propensity scores to receive the treatment (treatment as defined in `"trtgrp"`).

Alternatively, `type = "density"` provides the density of the estimated probability to receive each treatment level. Figure 1 presents the histogram and density plots of the estimated propensity scores in our analysis. The histogram suggests that there may be a slight lack of overlap due to minor separation of the two groups.

```
R> plot(bal.any, type = "density")
R> plot(bal.any, type = "hist")
```

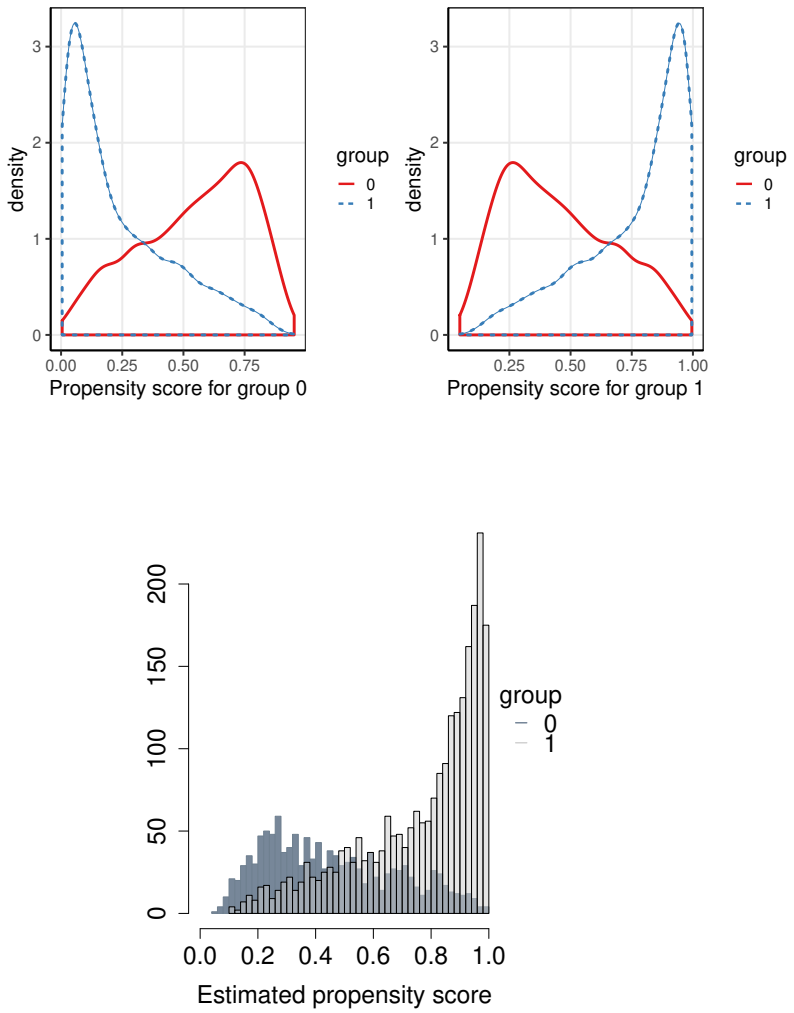

Figure 1: Histogram and density plots of estimated propensity scores with respect to the binary treatment variable `Dany` generated by `plot.SumStat()` function.

Finally, specifying argument `type = "balance"` in `plot.SumStat()` generates a love plot based on either the ASD metric (`metric = "ASD"`) or the maximum PSD metric (`metric = "PSD"`). Figure 2 presents the PSD-based love plot with the weighted standard deviation (by default `weighted.var = TRUE`). Clearly, the unweighted mean differences are substantially larger than the commonly used balance threshold 0.1, while propensity score weighting in general improves the covariate balance. Among the three weighting schemes, OW and IPW have controlled the maximum PSD for each covariate to be below 0.1, and OW provides the

best balance, with the maximum PSD for each covariate being close to zero.

```
R> plot(bal.any, type = "balance", metric = "PSD")
```

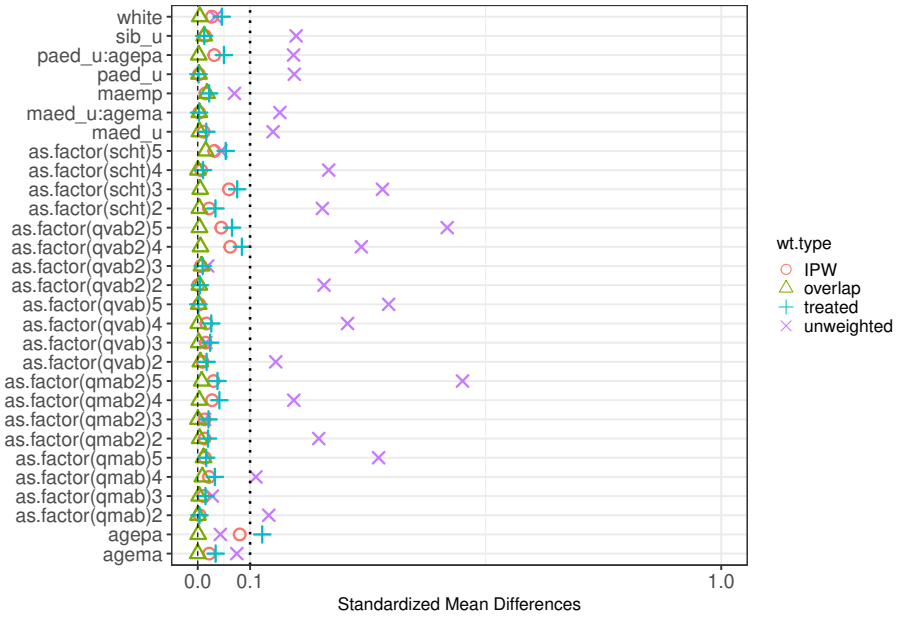

Figure 2: Love plot with the binary treatment variable `Dany` using the maximum PSD metric, generated by `plot.SumStat()` function in the **PSweight** package.

Estimation and Inference of (Weighted) Average Treatment Effects

Because the IPW, treated weights and OW achieve adequate balance according to Figure 2, we use these three weighting schemes to estimate the ATE, ATT and ATO. Based on the propensity score model `ps.any`, we first use the `PSweight()` function to obtain the average potential outcomes among the combined population using IPW.

```
R> ate.any <- PSweight(ps.formula = ps.any, yname = "wage", data = NCDS,
+     weight= "IPW")
R> ate.any

Original group value:  0, 1
Point estimate:
1.9002, 2.0927
```

The `ate.any` is an `PSweight` object returned by the `PSweight()` function. Printing `ate.any` will only provide the estimated average potential outcomes for each treatment level. In this case, 1.9002 and 2.0927 correspond to the average log hourly wages when the entire population attains no academic qualification (`Dany = 0`) and otherwise (`Dany = 1`). We observe that higher educational attainment leads to higher average hourly wage. Despite its simple on-screen output, `ate.any` contains a list of six elements: estimated propensity scores (`propensity`), estimated average potential outcomes (`muhat`), joint covariance matrix of the estimated average potential outcomes (`covmu`), estimates for each bootstrap sample

if `bootstrap = TRUE` (`muboot`), group label in alphabetic orders (`group`), and the indicated treatment group for defining ATT (`trtgrp`).

The average potential outcomes among the treated population and among the overlap population can be estimated in a similar fashion, by specifying the `weight` option in the `PSweight()` function. If `weight` is left unspecified, `PSweight()` function uses the OW by default and emphasizes the subpopulation with the optimal internal validity (Li *et al.* 2018). When `weight = "treated"`, we obtain the estimated average potential outcomes among the population with academic qualification. For estimating the ATT, if one leaves `trtgrp` unspecified, `PSweight()` function by default considers the last value (in alphabetic order) of the treatment variable to be the treatment group in defining ATT (`Dany = 1`). If the investigator is instead interested in estimating the causal effect among the population without academic qualification, the specification of `trtgrp = 0` should be used.

```
R> ato.any <- PSweight(ps.formula = ps.any, yname = "wage", data = NCDS)
R> att.any <- PSweight(ps.formula = ps.any, yname = "wage", data = NCDS,
+     weight= "treated")
R> ato.any

Original group value:  0, 1
Point estimate:
1.8617, 2.0408

R> att.any

Original group value:  0, 1
Treatment group value:  1
Point estimate:
1.9394, 2.1515
```

Compared to `ate.any` and `ato.any`, the on-screen output of `att.any` now includes an extra element,the treatment group that defines the ATT estimand. In the analysis of NCDS data, we only see minor differences between the estimated average potential outcomes across the three target populations. The average log hourly wage appear consistently higher if all individuals in either target population attained academic qualification, say, through some effective population-level educational intervention. Similar to the design module, we provide a `summary.PSweight()` function to estimate the (weighted) average treatment effects and their variances. By default, `summary.PSweight()` presents all pairwise contrasts of the estimated average potential outcomes `type = "DIF")`, and therefore targets on the additive causal esimands. For example, we can estimate the ATE and ATO along with their sandwich standard errors and 95% confidence intervals using the following code.

```
summary(ate.any, CI = FALSE)

Closed-form inference:

Original group value:  0, 1

Contrast:
```

```
            0 1
Contrast 1 -1 1

           Estimate Std.Error z value  Pr(>|z|)
Contrast 1 0.192543  0.021122  9.1158 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

R> summary(ato.any, CI = FALSE)

Closed-form inference:

Original group value:  0, 1

Contrast:
            0 1
Contrast 1 -1 1

           Estimate Std.Error z value  Pr(>|z|)
Contrast 1 0.179129  0.015609  11.476 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

The returns of the `summary.PSweight()` function indicate that the standard error for ATO is smaller and the associated confidence interval is tighter, matching the theoretical results of Li *et al.* (2018). The `summary.PSweight()` function also returns the p-value of (weak) causal null hypothesis that the specified contrast of the average potential outcomes is zero. In this case, the p-values correspond to $H_0 : \mu_1^h = \mu_0^h$ are all small and we reject the null.

In addition to specifying a propensity score model, we obtain an augmented estimator by specifying a model for log hourly wage as a function of potential confounders within each treatment group. The `PSweight()` function allows us to combine propensity score weighting and outcome modeling to achieve efficiency and/or increased robustness. We specify a regression formula through `out.formula` using the same set of confounders adjusted for in `ps.any`.

```
R> out.wage <- wage ~ white + maemp + as.factor(scht) + as.factor(qmab)
+     + as.factor(qmab2) + as.factor(qvab) + as.factor(qvab2) + paed_u + maed_u
+     + agepa + agema + sib_u + paed_u * agepa + maed_u * agema
```

The treatment variable is not included in `out.wage` as `PSweight` automatically fits a separate a potential outcome regression model within each treatment group, therefore allowing for full treatment-by-covariate interactions. For the continuous outcome `wage`, the `PSweight()` fits the linear model by default (`family = "gaussian"`). Loading the outcome regression formula and specifying `augmentation = TRUE`, we obtain the estimated average potential outcomes by the augmented weighting estimators introduced in Section 2.4

```
R> ate.any.aug <- PSweight(ps.formula = ps.any, yname = "wage", data = NCDS,
+    weight= "IPW", augmentation = TRUE, out.formula = out.wage)
```

```
R> ato.any.aug <- PSweight(ps.formula = ps.any, yname = "wage", data = NCDS,
+    augmentation = TRUE, out.formula = out.wage)
```

Similar to the simple weighting estimators, the output on screen includes the information of the treatment group levels as well as the estimated average potential outcomes in the respective target population (output omitted for brevity). In this analysis, we find that the point estimates do not differ substantially between the simple weighting and the augmented weighting estimators, for each weighting scheme under consideration. The fact that the augmented weighting estimates resemble the simple weighting estimates may serve as indirect evidence that the propensity score model is not grossly misspecified (Robins and Rotnitzky 2001; Mercatanti and Li 2014).

We then estimate the (weighted) average treatment effects using the `summary.PSweight()` function. In this example, while the point estimates do not change substantially between the augmented weighting estimators and simple weighting estimators, outcome augmentation reduces the standard errors for estimating ATE, but not so much for estimating ATO. Such comparison results match the simulation findings of Mao *et al.* (2018). Overall, we find that, regardless of the weighting scheme considered, attaining academic qualification on average leads to significantly higher hourly wage than not at the 0.05 level. We do acknowledge, however, that the interpretation of study results should not rely on a single dichotomy of a p-value that is great than or smaller than 0.05.

```
R> summary(ate.any.aug, CI=FALSE)

Closed-form inference:

Original group value:  0, 1

Contrast:
            0 1
Contrast 1 -1 1

           Estimate Std.Error z value  Pr(>|z|)
Contrast 1 0.186079  0.019842  9.3782 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R> summary(ato.any.aug, CI = FALSE)

Closed-form inference:

Original group value:  0, 1

Contrast:
            0 1
Contrast 1 -1 1

           Estimate Std.Error z value  Pr(>|z|)
Contrast 1 0.180004  0.015646  11.505 < 2.2e-16 ***
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Alternatively, the standard errors and confidence intervals can be estimated via nonparametric bootstrap. For example, we can specify `bootstrap = TRUE` in the `PSweight()` function and use `summary.PSweight()` to make bootstrap-based inference for any causal contrasts based on average potential outcomes. By default, the number of bootstrap replicates is set to 50, and other values can be specified using the `R` argument in `PSweight()` function. When `bootstrap = TRUE`, `PSweight()` prints a short message for completing every 50 runs for ease of monitoring.

```
R> ate.any.bs <- PSweight(ps.formula = ps.any, yname = "wage", data = NCDS,
+    weight= "IPW", bootstrap = TRUE)

bootstrap 50 samples
```

While the on screen output `ate.any.bs` is no different from `ate.any`, summarizing `ate.any.bs` now returns the bootstrap standard errors, (quantile-based) confidence intervals and associated p-values. We illustrate how to obtain these information using the following code.

```
R> summary(ate.any.bs, contrast = rbind(c(-1, 1),c(1, -1))

Use Bootstrap sample for inference:

Original group value:  0, 1

Contrast:
            0  1
Contrast 1 -1  1
Contrast 2  1 -1

            Estimate Std.Error       lwr      upr  Pr(>|z|)
Contrast 1  0.192543  0.024332  0.137837  0.23701 < 2.2e-16 ***
Contrast 2 -0.192543  0.024332 -0.237005 -0.13784 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

In the above example, we further illustrate how one can specify non-default contrasts through the `contrast` argument. By setting `contrast = rbind(c(-1, 1),c(1, -1))`, we can simultaneously report the causal comparison for `Dany = 1` versus `Dany = 0` and its reverse comparison. These two contrasts study the same causal effect from two opposite directions, therefore it is expected that the same numerical values are returned with a reverse sign. The bootstrap standard error is almost identical to the sandwich standard error, but the bootstrap confidence interval is no longer symmetric around the point estimate as it does not rely on normal approximation.

4.3. Propensity Score Weighting with Multiple Treatments

The syntax we provide in the binary treatment case in Section 4.2 can be generalized seamlessly to the multiple treatment case; therefore to avoid redundancy, the purpose of this subsection is not to repeat the same steps. Instead, we complement the last subsection by pointing out additional features of **PSweight** with multiple treatments. For simplicity, we will focus on IPW and the three types of weights that improve overlap: OW, MW and EW (Li and Li 2019).

Estimating Generalized Propensity Scores and Balance Assessment

We use `Dmult`, the three-level variable, as the treatment of interest. About one half of the population attained advanced academic qualification, the there are approximately equal number of individuals with intermediate academic qualification or no academic qualification. To illustrate the estimation and inference for ratio estimands, we also introduce a binary outcome of wage, `wagebin`. The dichotomized wage was obtained with the cutoff of the average hourly wage of actively employed British male aged 30-39 in 1991[3]. The averaged hourly wage is 8.23, and we take $\log(8.23) \approx 2.10$ as the cutoff. Among the study participants, we observe 1610 and 2032 individuals above and below the average, and we are interested in estimating the pairwise (weighted) average treatment effect of the academic qualification for obtaining above-average hourly wage.

We specify a multinominal regression model, `ps.mult`, to estimate the generalized propensity scores, with the same set of covariates used in the binary treatment case.

```
ps.mult <- Dmult ~ white + maemp + as.factor(scht) + as.factor(qmab) +
+     as.factor(qmab2) + as.factor(qvab) + as.factor(qvab2) + paed_u + maed_u +
+     agepa + agema + sib_u + paed_u * agepa + maed_u * agema
```

Then we obtain the propensity score estimates and assess weighted covariate balance with the `SumStat()` function. This component is similar to the binary treatment case, except that we only allow density plots for visualizing the generalized propensity scores (but not histograms). Specifically, the `plot.SumStat()` function returns a density plot even if one specifies `type = "hist"`. In this case, a warning message will be generated to indicate that `"Histogram only available for binary treatment"`.

```
R> bal.mult <- SumStat(ps.formula = ps.mult,
+     weight = c("IPW", "overlap", "matching", "entropy"), data = NCDS)
R> plot(bal.mult, type = "hist")

Warning message:
In plot.SumStat(bal.mult, type = "hist") :
  Histogram only available for binary treatment. Density plot provided instead.
```

The distributions of generalized propensity scores are given in Figure 3 (in alphabetic order of the names of treatment groups). For the generalized propensity score to receive the advanced qualification (`">=A/eq"`) or no qualification (`"None"`), there is a mild lack of overlap due to separation of the group-specific distribution. Since `bal.mult` includes four weighting schemes, we plot the maximum pairwise ASD and assess the (weighted) covariate balance in a single Love plot.

```
R> plot(bal.mult, metric = "ASD")
```

[3]https://www.ons.gov.uk/employmentandlabourmarket/peopleinwork/earningsandworkinghours/

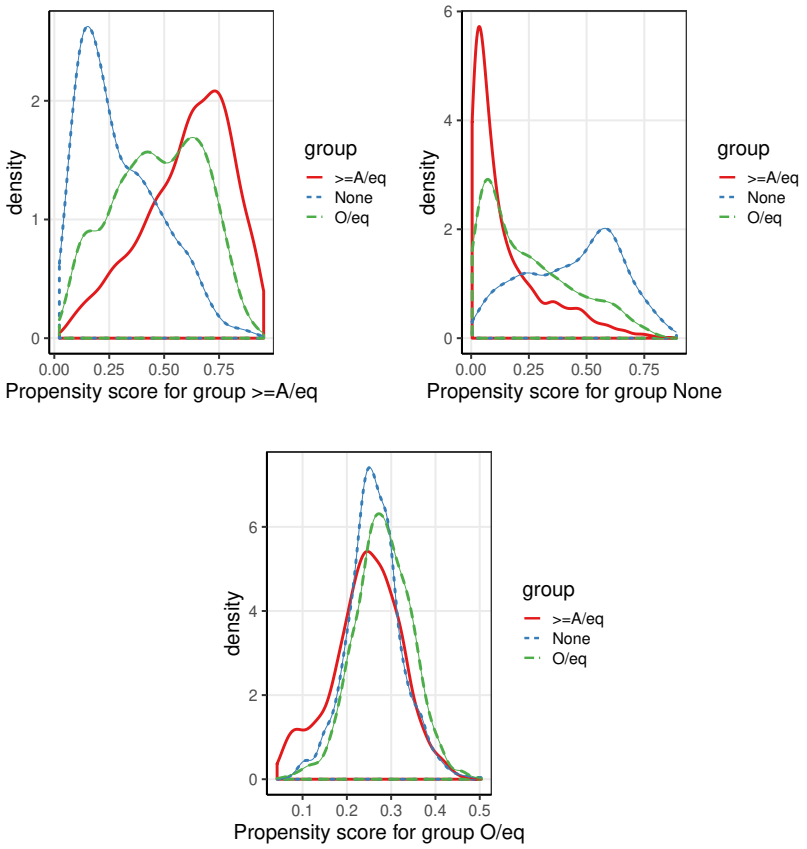

Figure 3: Density plots of estimated generalized propensity scores with respect to the three-level treatment variable `Dmult` generated by `plot.SumStat()` function in the **PSweight** package.

The covariates are imbalanced across the three groups prior to any weighting. Although IPW can generally improve covariate balance, the maximum pairwise ASD still ocassionally exceeds the threshold 0.1 due to lack of overlap. In contrast, OW, MW and EW all emphasize the subpopulation with improved overlap and provide better balance across all covariates.

Generalized Propensity Score Trimming

The **PSweight** package can perform trimming based on (generalized) propensity scores. As IPW does not adequately balance the covariates across the three groups in Figure 4, we explore trimming as a way to improve balance for IPW. There are two types of trimming performed by the **PSweight** package: (1) symmetric trimming that removes units with extreme (generalized propensity scores) (Crump *et al.* 2009; Yoshida *et al.* 2018) and (2) optimal trimming that provides the most efficient IPW estimator for estimating (pairwise) ATE (Crump *et al.* 2009; Yang *et al.* 2016). Specifically, the symmetric trimming is supported by both the `SumStat()` and `PSweight()` functions through the `delta` argument. Both functions refit the (generalized) propensity score model after trimming following the recommendations in Li *et al.* (2019). We also provide a stand-alone `PStrim` function that performs both symmetric trimming and optimal trimming. Following Yoshida *et al.* (2018), with three treatment groups, we exclude

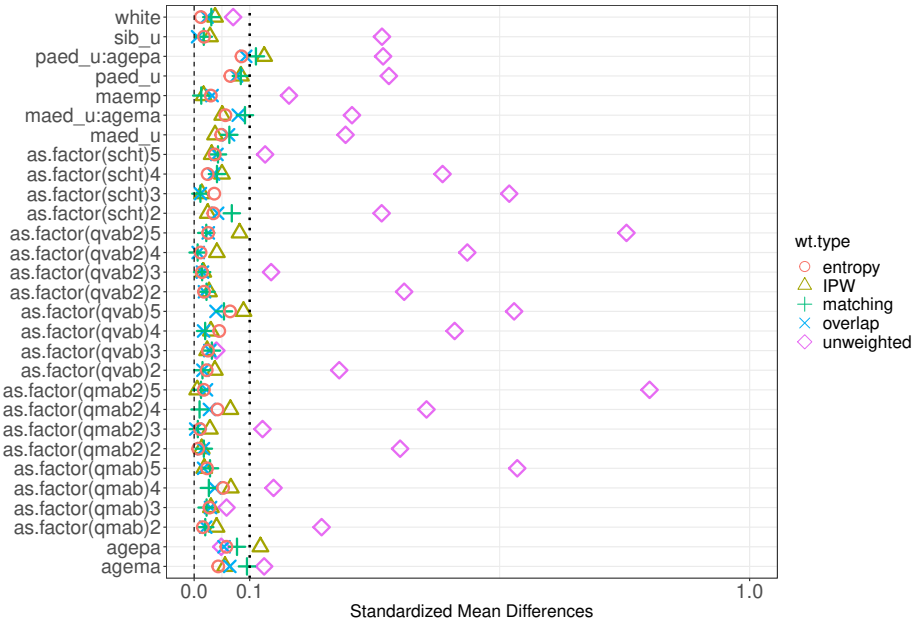

Figure 4: Love plot with the three-level treatment variable `Dmult` using the maximum pairwise ASD metric, generated by `plot.SumStat()` function in the **PSweight** package.

all individuals with the estimated generalized propensity scores less than $\delta = 0.067$. This threshold removes a substantial amount of individuals in the advanced qualification group (information can be pulled from the `trim` element in the `SumStat` object). As discussed in Yoshida *et al.* (2018), propensity trimming could improve the estimation of ATE and ATT, but barely have any effect for estimation of ATO and ATM. Evidently, Figure 5 indicates that IPW controls all pairwise ASD within 10% in the trimmed sample. Trimming had nearly no effect on the weighted balance for OW, MW and EW.

```
R> bal.mult.trim <- SumStat(ps.formula = ps.mult,
+    weight = c("IPW", "overlap", "matching", "entropy"),
+    data = NCDS, delta = 0.067)
R> bal.mult.trim

1050 cases trimmed,  2592 cases remained

trimmed result by trt group:
         >=A/eq None O/eq
trimmed     778   71  201
remained   1028  824  740

weights estimated for:  IPW overlap matching entropy

R> plot(bal.mult.trim,metric = "ASD")
```

Alternatively, if one does not specify the trimming threshold, the `PStrim` function supports the optimal trimming procedure that identifies the optimal threshold based on data. An

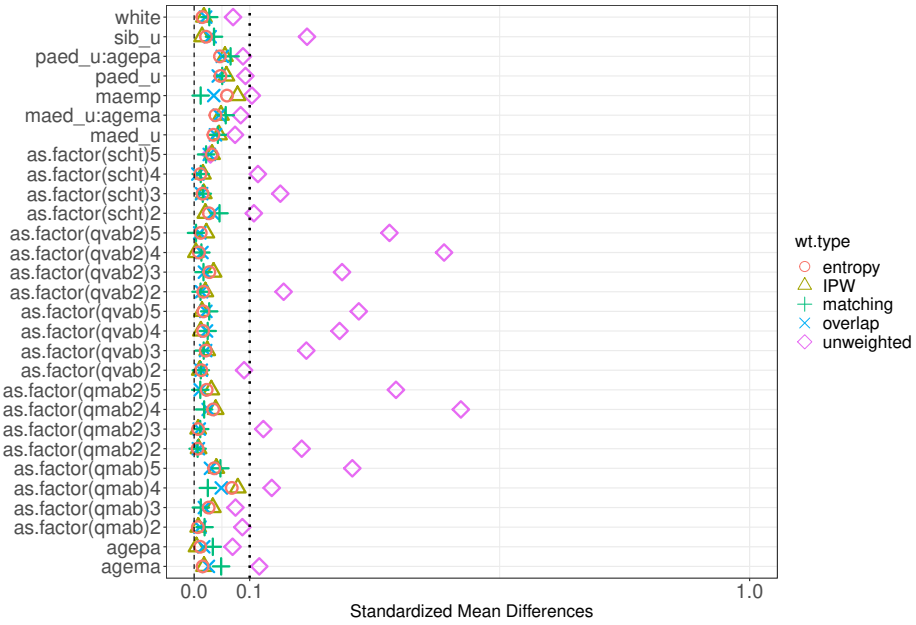

Figure 5: Love plot with the three-level treatment variable `Dmult` using the maximum pairwise ASD metric, after symmetric trimming with $\delta = 0.067$. This plot is generated by `plot.SumStat()` function in the **PSweight** package.

example syntax is given as follows. By pulling out the summary statistics for trimming, we can see that optimal trimming excludes 27%, 9% and 2% of the individuals among those with advanced qualification, intermediate qualification and no qualification, respectively. The exclusion is more conservative compared to symmetric trimming with $\delta = 0.067$. However, the resulting covariate balance after optimal trimming is similar to Figure 5 and omitted.

```
R> PStrim(ps.formula = ps.mult, data = NCDS, optimal = TRUE)

          >=A/eq None O/eq
trimmed      479   21   82
remained    1327  874  859
```

Estimation and Inference of Pairwise (Weighted) Average Treatment Effects

We estimate the ratio estimands introduced in Section 2.5 using the binary outcome `wagebin`. For illustration, we will only estimate the causal effects based on the data without trimming, and the analysis with the trimmed data follows the exact same steps. Based on the multinomial logistic propensity score model, we obtain the pairwise causal RR among the combined population via IPW.

```
R> ate.mult <- PSweight(ps.formula = ps.mult, yname = "wagebin", data = NCDS,
+     weight = "IPW")
R> contrasts.mult <- rbind(c(1,-1, 0), c(1, 0,-1), c(0, -1, 1))
R> sum.ate.mult.rr <- summary(ate.mult, type = "RR", contrast = contrasts.mult)
R> sum.ate.mult.rr
```

```
Closed-form inference:

Inference in log scale:
Original group value:  >=A/eq, None, O/eq

Contrast:
          >=A/eq None O/eq
Contrast 1      1   -1    0
Contrast 2      1    0   -1
Contrast 3      0   -1    1

            Estimate Std.Error       lwr     upr  Pr(>|z|)
Contrast 1  0.607027  0.115771  0.380120 0.83393 1.577e-07 ***
Contrast 2  0.459261  0.052294  0.356767 0.56176 < 2.2e-16 ***
Contrast 3  0.147766  0.121692 -0.090746 0.38628    0.2246
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

By providing the appropriate contrast matrix, we obtain all pairwise comparisons of the average potential outcomes on the log scale with the `summary.PSweight()` function, and estimate $\lambda^{h,\mathrm{RR}}(\boldsymbol{a})$ for contrast vector $\boldsymbol{a}$. The p-values provides statistical evidence against the weak causal null $H_0 : \lambda^{h,\mathrm{RR}}(\boldsymbol{a}) = 0$. It is found that, among the combined population, the proportion that receives above-average hourly wage when everyone attains advanced qualification is $\exp(0.607) = 1.83$ times that when everyone attains no academic qualification. Further, the proportion that receives above-average hourly wage when everyone attains advanced qualification is $\exp(0.459) = 1.58$ times that when everyone attains intermediate qualification. Both effects are significant at the 0.05 levels and provides strong evidence against the corresponding causal null (p-value < 0.001). However, if everyone attains intermediate qualification, the proportion that receives above-average hourly wage is only slightly higher compared to without qualification, with a p-value exceeding 0.05. To directly report the causal RR and its confidence intervals, we can simply exponentiate the point estimate and confidence limits provided by the `summary.PSweight()` function.

```
R> exp(sum.ate.mult.rr$estimates[,c(1,4,5)])

            Estimate       lwr      upr
Contrast 1 1.834968 1.4624601 2.302358
Contrast 2 1.582904 1.4287028 1.753749
Contrast 3 1.159241 0.9132496 1.471493
```

Focusing on the target population that has the most overlap in the observed covariates, we further use the OW to estimate the pairwise causal RR. OW theoretically provides the best internal validity for pairwise comparisons; Figure 5 also indicates that OW achieves better covariate balance among the overlap population. Exponentiating the results provided by the `summary.PSweight()` function, we observe each pairwise causal RR has a larger effect size among the overlap weighted population. Interestingly, among the overlap population, the proportion that receives above-average hourly wage when everyone attains intermediate

qualification becomes approximately 1.55 times that when everyone attains no academic qualification, and the associated 95% CI excludes the null. Moreover, the standard errors for the pairwise comparisons are smaller when using OW versus IPW, implying that OW analysis generally corresponds to increased power by focusing on a population with equipoise. We repeat the analysis using both MW and EW; the results are similar to OW for this analysis and therefore omitted for brevity.

```
R> ato.mult <- PSweight(ps.formula = ps.mult, yname = "wagebin", data = NCDS,
+     weight = "overlap")
R> sum.ato.mult.rr <- summary(ato.mult, type = "RR", contrast = contrasts.mult)
R> exp(sum.ato.mult.rr$estimates[,c(1,4,5)])

           Estimate      lwr      upr
Contrast 1 2.299609 1.947140 2.715882
Contrast 2 1.527931 1.363092 1.712705
Contrast 3 1.505048 1.257180 1.801785
```

The above output suggests that among the overlap population, the causal RR for comparing advanced qualification and intermediate qualification is similar in magnitude to that for comparing intermediate qualification and no qualification. We can formally test for the equality of two consecutive causal RR based on the null hypothesis $H_0 : \mu_3^h/\mu_2^h = \mu_2^h/\mu_1^h$ (also see Section 2.5). Operationally, we need to specify the corresponding contrast vector `contrast = c(1, 1, -2)`. The p-value for testing this null is 0.91 (output omitted for brevity), and suggests a lack of evidence against the equality of consecutive causal RR at the 0.05 level.

```
R> summary(ato.mult, type = "RR", contrast = c(1, 1, -2), CI = FALSE)
```

With the binary outcome `wagebin`, we can also estimate the pairwise causal OR among a specific target population. For example, using OW, the causal conclusions regarding the effectiveness due to attaining academic qualification do not change, because all three 95% confidence intervals exclude null. However, the pairwise causal OR appear larger than the pairwise causal RR. This is expected because our outcome of interest is not uncommon (Nurminen 1995). For rare outcomes, causal OR approximates causal RR.

```
R> sum.ato.mult.or <- summary(ato.mult, type = "OR", contrast = contrasts.mult)
R> exp(sum.ato.mult.or$estimates[,c(1,4,5)])

           Estimate      lwr      upr
Contrast 1 3.586050 2.841383 4.525879
Contrast 2 2.050513 1.696916 2.477791
Contrast 3 1.748855 1.375483 2.223578
```

As a final step, we illustrate how to combine OW with outcome regression and estimate the pairwise causal RR among the overlap population. Similar to Section 4.2, we use the same set of covariates in the binary outcome regression model.

```
R> out.wagebin <- wagebin ~ white + maemp + as.factor(scht) + as.factor(qmab) +
+     as.factor(qmab2) + as.factor(qvab) + as.factor(qvab2) + paed_u + maed_u +
+     agepa + agema + sib_u + paed_u * agepa + maed_u * agema
```

Loading this outcome regression formula into the `PSweight()` function, and specifying `family = "binomial"` to indicate the type of outcome, we obtain the augmented overlap weighting estimates on the log RR scale. Exponentiating the point estimates and confidence limits, one reports the pairwise causal RR. The pairwise causal RR reported by the augmented OW estimator is similar to that reported by the simple OW estimator; further, the width of the confidence interval is also comparable before and after outcome augmentation, and the causal conclusions based on pairwise RR remain the same. The similarity between simple and augmented OW estimators implies that OW itself may already be efficient.

```
R> ato.mult.aug <- PSweight(ps.formula = ps.mult, yname = "wagebin", data = NCDS,
+     augmentation = TRUE, out.formula = out.wagebin, family = "binomial")
R> sum.ato.mult.aug.rr <- summary(ato.mult.aug, type = "RR",
+     contrast = contrasts.mult)
R> exp(sum.ato.mult.aug.rr$estimates[,c(1,4,5)])

           Estimate      lwr      upr
Contrast 1 2.310628 1.957754 2.727105
Contrast 2 1.540176 1.375066 1.725111
Contrast 3 1.500237 1.253646 1.795331
```

4.4. Using Machine Learning to Estimate Propensity Scores and Potential Outcomes

As an alternative to the default generalized linear models, we can use more advanced machine learning models to estimate propensity scores and potential outcomes. Flexible propensity score and outcome estimation has been demonstrated to reduce bias due to model misspecification, and potentially improve covariate balance (Lee, Lessler, and Stuart 2010; Hill 2011; McCaffrey *et al.* 2013). This can be achieved in **PSweight** for both balance check and constructing weighted estimator by specifying the method as the generalized boosted model (GBM) or the super learner methods. Additional model specifications for these methods can be supplied through `ps.control` and `out.control`. Machine learning models that are included in neither `gbm` nor `SuperLearner` could be estimated externally and then imported through the `ps.estimate` and `out.estimate` arguments. These two arguments broaden the utility of **PSweight** where any externally generated estimates of propensity scores and potential outcomes models can be easily incorporated.

We now illustrate the use of GBM as an alternative of the default generalized linear models. The illustration is based on binary education: 'Dany'. GBM is a family of non-parametric tree-based regressions that allow for flexible non-linear relationships between predictors and outcomes (Friedman, Hastie, and Tibshirani 2000). The following propensity model formula is specified; the formula does not include interactions terms because boosted regression is already capable of capturing non-linear effects and interactions (McCaffrey, Ridgeway, and Morral 2004). In this illustration, we use the AdaBoost (Freund and Schapire 1997) algorithm to fit the propensity model through the control setting, `ps.control=list(distribution = "adaboost")`. We use the default values for other model parameters such as the number of trees (`n.trees = 100`), interaction depth (`interaction.depth = 1`), the minimum number of observations in the terminal nodes (`n.minobsinnode = 1`), shrinkage reduction

(shrinkage = 0.1), and bagging fraction (shrinkage = 0.5). Alternative values for these parameters could also be passed through ps.control.

```
R> ps.any.gbm <- Dany ~ white + maemp + as.factor(scht) + as.factor(qmab) +
+     as.factor(qmab2) + as.factor(qvab) + as.factor(qvab2) + paed_u +
+     maed_u+ agepa + agema + sib_u
R> bal.any.gbm <-SumStat(ps.formula = ps.any.gbm, data= NCDS, weight = "overlap",
+     method = "gbm", ps.control = list(distribution = "adaboost"))
```

The balance check through plot.SumStat() suggests substantial improvement in covariate balance with SMD of all covariates below 0.1 after weighting. After assessing balance and confirming the adequacy of the propensity score model, we further fit the outcome model using GBM with the default logistic regression and parameters. In the PSweight() function, we can specify both ps.method = "gbm" and out.method = "gbm" and leave the out.control argument as default. The detailed code and summary of the output is in below. Here we redefine the propensity score model without interaction terms because GBM considers interactions between covariates by default. The results using GBM are very similar to those using generalized linear models.

```
R> out.wage.gbm <- wage ~ white + maemp + as.factor(scht) + as.factor(qmab) +
+     as.factor(qmab2) + as.factor(qvab) + as.factor(qvab2) + paed_u +
+     maed_u + agepa + agema + sib_u
R> ato.any.aug.gbm <- PSweight(ps.formula = ps.any.gbm, yname = "wagebin",
+     data = NCDS, augmentation = TRUE, out.formula = out.wage.gbm,
+     ps.method = "gbm", ps.control = list(distribution = "adaboost"),
+     out.method = "gbm")
R> summary(ato.any.aug.gbm, CI = FALSE)

Closed-form inference:

Original group value:  0, 1

Contrast:
            0 1
Contrast 1 -1 1

           Estimate Std.Error z value  Pr(>|z|)
Contrast 1 0.186908  0.018609  10.044 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5. Summary

Propensity score weighting is an important tool for causal inference and comparative effectiveness research. This paper introduces the **PSweight** package and demonstrates its functionality with the NCDS data example in the context of binary and multiple treatment groups. In addition to providing easy-to-read balance statistics and plots to aid the design of observational studies, the **PSweight** offers point and variance estimation with a variety of weighting

schemes for the (weighted) average treatment effects on both the additive and ratio scales. These weighting schemes include the optimal overlap weights recently introduced in Li *et al.* (2018) and Li and Li (2019), and could help generate valid causal comparative effectiveness evidence among the population at equipoise.

The **PSweight** package is under continuing development to include other useful components for propensity score weighting analysis. Specifically, future versions of **PSweight** will include components to enable pre-specified subgroup analysis with balancing weights and flexible variable selection tools (Yang, Lorenzi, Papadogeorgou, Wojdyla, Li, and Thomas 2020). We are also studying overlap weighting estimators with time-to-event outcomes and complex survey designs. Those new features are being actively developed concurrently with our extensions to the methodology.

Computational details

PSweight 1.1.6 (license: GPL-2, GPL-3) was built on R 4.0.3 and dependent on the **MASS** 7.3.51-4 package, **ggplot2** 3.2.1 package, **nnet** 7.3-14, **gbm** 2.1.8, **SuperLearner** 2.0-26, and **numDeriv** 2016.8-11 package. Package **mice** is not a dependent package of **PSweight** but was used to impute the missing entries in our data example. All these packages used are available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/`.

Acknowledgement

The authors would like to acknowledge the NCDS replication data published on Harvard Dataverse (`https://dataverse.harvard.edu/`) (Battistin and Sianesi 2012), which provides a coded data set for our analysis in Section 4.

References

Agresti A (2003). *Categorical Data Analysis*, volume 482. John Wiley & Sons. `doi:10.1080/02664763.2013.854979`.

Austin PC, Stuart EA (2015). "Moving Towards Best Practice When Using Inverse Probability of Treatment Weighting (IPTW) Using the Propensity Score to Estimate Causal Treatment Effects in Observational Studies." *Statistics in Medicine*, **34**(28), 3661–3679. `doi:10.1002/sim.6607`.

Bang H, Robins JM (2005). "Doubly Robust Estimation in Missing Data and Causal Inference Models." *Biometrics*, **61**(4), 962–973. `doi:10.1111/j.1541-0420.2005.00377.x`.

Battistin E, Sianesi B (2011). "Misclassified Treatment Status and Treatment Effects: an Application to Returns to Education in the United Kingdom." *Review of Economics and Statistics*, **93**(2), 495–509. `doi:10.1162/REST_a_00175`.

Battistin E, Sianesi B (2012). "Replication data for: Misclassified Treatment Status and Treatment Effects: An Application to Returns to Education in the United Kingdom." `doi:10.7910/DVN/EPCYUL`. URL `https://doi.org/10.7910/DVN/EPCYUL`.

Bodory H, Huber M (2020). ***causalweight**: Causal Inference Based on Inverse Probability Weighting, Doubly Robust Estimation, and Double Machine Learning*. R package version 0.2.1, URL `https://CRAN.R-project.org/package=causalweight`.

Buuren Sv, Groothuis-Oudshoorn K (2010). "**mice**: Multivariate Imputation by Chained Equations in R." *Journal of Statistical Software*, pp. 1–68. `doi:10.18637/jss.v045.i03`.

Crump RK, Hotz VJ, Imbens GW, Mitnik OA (2009). "Dealing With Limited Overlap in Estimation of Average Treatment Effects." *Biometrika*, **96**(1), 187–199. `doi:10.1093/biomet/asn055`.

Fong C, Ratkovic M, Imai K (2019). ***CBPS**: Covariate Balancing Propensity Score*. R package version 0.21, URL `https://CRAN.R-project.org/package=CBPS`.

Freund Y, Schapire RE (1997). "A decision-theoretic generalization of on-line learning and an application to boosting." *Journal of computer and system sciences*, **55**(1), 119–139. `doi:10.1006/jcss.1997.1504`.

Friedman J, Hastie T, Tibshirani R (2000). "Additive Logistic Regression: A Statistical View of Boosting." *The Annals of statistics*, **28**(2), 337–407. `doi:10.1214/aos/1016218223`.

Funk MJ, Westreich D, Wiesen C, Stürmer T, Brookhart MA, Davidian M (2011). "Doubly Robust Estimation of Causal Effects." *American journal of epidemiology*, **173**(7), 761–767. `doi:10.1093/aje/kwq439`.

Greenwell B, Boehmke B, Cunningham J, Developers G (2019). ***gbm**: Generalized Boosted Regression Models*. R package version 2.1.5, URL `https://CRAN.R-project.org/package=gbm`.

Greifer N (2018). "Covariate balance tables and plots: A guide to the cobalt package." *Technical report*, Institute for Quantitative Social Science.

Greifer N (2019). *optweight: Targeted Stable Balancing Weights Using Optimization*. R package version 0.2.5, URL `https://CRAN.R-project.org/package=optweight`.

Greifer N (2020). ***WeightIt**: Weighting for Covariate Balance in Observational Studies*. R package version 0.10.2, URL `https://CRAN.R-project.org/package=WeightIt`.

Haris A, Chan G (2015). ***ATE**: Inference for Average Treatment Effects using Covariate Balancing*. R package version 0.2.0, URL `https://CRAN.R-project.org/package=ATE`.

Hill JL (2011). "Bayesian Nonparametric Modeling for Causal Inference." *Journal of Computational and Graphical Statistics*, **20**(1), 217–240. `doi:10.1198/jcgs.2010.08162`.

Hirano K, Imbens G, Ridder G (2003). "Efficient Estimation of Average Treatment Effects Using the Estimated Propensity Score." *Econometrica*, **71**, 1161–1189. `doi:10.3386/t0251`.

Hirano K, Imbens GW (2001). "Estimation of Causal Effects Using Propensity Score Weighting: An Application to Data on Right Heart Catheterization." *Health Services and Outcomes Research Methodology*, **2**, 259–278. `doi:10.1023/A:1020371312283`.

Horvitz DG, Thompson DJ (1952). "A Generalization of Sampling Without Replacement From a Finite Universe." *Journal of the American Statistical Association*, **47**, 663–685. doi:10.2307/2280784.

Imai K, Ratkovic M (2014). "Covariate Balancing Propensity Score." *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, **76**(1), 243–263. doi:10.1111/rssb.12027.

Imbens GW (2000). "The Role of the Propensity Score in Estimating Dose-Response Functions." *Biometrika*, **87**(3), 706–710. doi:10.1093/biomet/87.3.706.

Lechner M (2001). "Identification and Estimation of Causal Effects of Multiple Treatments Under the Conditional Independence Assumption." *Econometric Evaluations of Active Labor Market Policies in Europe*, pp. 43–58. doi:10.2139/ssrn.177089.

Lee BK, Lessler J, Stuart EA (2010). "Improving Propensity Score Weighting Using Machine Learning." *Statistics in medicine*, **29**(3), 337–346. doi:10.1002/sim.3782.

Li F, Li F (2019). "Propensity Score Weighting for Causal Inference With Multiple Treatments." *The Annals of Applied Statistics*, **13**(4), 2389–2415. doi:10.1214/19-AOAS1282.

Li F, Morgan KL, Zaslavsky AM (2018). "Balancing Covariates via Propensity Score Weighting." *Journal of the American Statistical Association*, **113**(521), 390–400. doi:10.1080/01621459.2016.1260466.

Li F, Thomas LE, Li F (2019). "Addressing Extreme Propensity Scores via the Overlap Weights." *American Journal of Epidemiology*, **1**(188), 250–257. doi:10.1093/aje/kwy201.

Li L, Greene T (2013). "A Weighting Analogue to Pair Matching in Propensity Score Analysis." *International Journal of Biostatistics*, **9**(2), 1–20. doi:10.1515/ijb-2012-0030.

Lunceford JK, Davidian M (2004). "Stratification and Weighting via the Propensity Score in Estimation of Causal Treatment Effects: A Comparative Study." *Statistics in Medicine*, **23**(19), 2937–2960. doi:10.1002/sim.1903.

Mao H, Li L (2018). *PSW: Propensity Score Weighting Methods for Dichotomous Treatments*. R package version 1.1-3, URL https://CRAN.R-project.org/package=PSW.

Mao H, Li L, Greene T (2018). "Propensity Score Weighting Analysis and Treatment Effect discovery." *Statistical Methods in Medical Research*, p. In press. doi:10.1177/0962280218781171.

McCaffrey DF, Griffin BA, Almirall D, Slaughter ME, Ramchand R, Burgette LF (2013). "A Tutorial on Propensity Score Estimation for Multiple Treatments Using Generalized Boosted Models." *Statistics in Medicine*, **32**(19), 3388–3414. doi:10.1002/sim.5753.

McCaffrey DF, Ridgeway G, Morral A (2004). "Propensity Score Estimation With Boosted Regression for Evaluating Causal Effects in Observational Studies." *Psychological Methods*, pp. 403–425. doi:10.1037/1082-989X.9.4.403.

Mercatanti A, Li F (2014). "Do Debit Cards Increase Household Spending? Evidence From a Semiparametric Causal Analysis of a Survey." *Annals of Applied Statistics*, **8**, 2405–2508. doi:10.1214/14-AOAS784.

Nurminen M (1995). "To Use or not to Use the Odds Ratio in Epidemiologic Analyses?" *European journal of epidemiology*, **11**(4), 365–371. doi:10.1007/BF01721219.

Polley E, LeDell E, Kennedy C, van der Laan M (2019). *SuperLearner: Super Learner Prediction*. R package version 2.0-26, URL https://CRAN.R-project.org/package=SuperLearner.

Ridgeway G, McCaffrey D, Morral A, Griffin BA, Burgette L, Cefalu M (2020). ***twang**: Toolkit for Weighting and Analysis of Nonequivalent Groups*. R package version 1.6, URL https://CRAN.R-project.org/package=twang.

Robins J, Hernán M, Brumback B (2000). "Marginal Structural Models and Causal Inference." *Epidemiology*, **11**, 550–560. doi:10.1093/ndt/gfw341.

Robins JM, Rotnitzky A (2001). "Comment on the Bickel and Kwon Article,"Inference for Semiparametric Models: Some Questions and an Answer"." *Statistica Sinica*, **11**(4), 920–936.

Robins JM, Rotnitzky A, Zhao LP (1994). "Estimation of Regression-coefficients When Some Regressors are not Always Observed." *Journal of the American Statistical Association*, **89**(427), 846–866. doi:10.2307/2290910.

Rosenbaum PR, Rubin DB (1983). "The Central Role of the Propensity Score in Observational Studies for Causal Effects." *Biometrika*, **70**(1), 41–55. doi:10.1093/biomet/70.1.41.

Shen C, Li X, Li L (2014). "Inverse Probability Weighting for Covariate Adjustment in Randomized Studies." *Statistics in medicine*, **33**(4), 555–568. doi:10.1186/s12874-020-00947-7.

Stefanski LA, Boos DD (2002). "The Calculus of M-estimation." *American Statistician*, **56**(1), 29–38. doi:10.1198/000313002753631330.

Thomas LE, Li F, Pencina MJ (2020a). "Overlap weighting: A Propensity Score Method that Mimics Attributes of a Randomized Clinical Trial." *Journal of the American Medical Association*, **323**(23), 2417–2418. doi:10.1001/jama.2020.7819.

Thomas LE, Li F, Pencina MJ (2020b). "Using Propensity Score Methods to Create Target Populations in Observational Clinical Research." *Journal of the American Medical Association*, **323**(5), 466–467. doi:10.1001/jama.2019.21558.

Tsiatis AA, Davidian M, Zhang M, Lu X (2008). "Covariate Adjustment for Two-sample Treatment Comparisons in Randomized Clinical Trials: A Principled Yet Flexible Approach." *Statistics in medicine*, **27**(23), 4658–4677. doi:10.1002/sim.3113.

Van der Laan MJ, Polley EC, Hubbard AE (2007). "Super Learner." *Statistical Applications in Genetics and Molecular Biology*, **6**(1). doi:10.2202/1544-6115.1309.

Williamson EJ, Forbes A, White IR (2014). "Variance Reduction in Randomised Trials by Inverse Probability Weighting Using the Propensity Score." *Statistics in Medicine*, **33**(5), 721–737. doi:10.1002/sim.5991.

Yang S, Imbens GW, Cui Z, Faries DE, Kadziola Z (2016). "Propensity Score Matching and Subclassification in Observational Studies With Multi-level Treatments." *Biometrics*, **72**(4), 1055–1065. doi:10.1111/biom.12505.

Yang S, Lorenzi E, Papadogeorgou G, Wojdyla DM, Li F, Thomas LE (2020). "Propensity Score Weighting for Causal Subgroup Analysis." *arXiv preprint arXiv:2010.02121*.

Yoshida K, Hernández-Díaz S, Solomon DH, Jackson JW, Gagne JJ, Glynn RJ, Franklin JM (2017). "Matching Weights to Simultaneously Compare Three Treatment Groups Comparison to Three-way Matching." *Epidemiology*, **28**(3), 387–395. doi:10.1097/EDE.0000000000000627.

Yoshida K, Solomon D, Haneuse S, Kim S, Patorno E, Tedeschi S, Lyu H, Franklin JMand Stürmer T, Hernández-Díaz S, Glynn R (2018). "Multinomial Extension of Propensity Score Trimming Methods: A Simulation Study." *American Journal of Epidemiology*, **183**(3), 609–616. doi:10.1093/aje/kwy263.

Zeng S, Li F, Wang R, Li F (2020). "Propensity Score Weighting for Covariate Adjustment in Randomized Clinical Trials." *arXiv preprint arXiv:2004.10075*.

Zhou Y, Matsouaka RA, Thomas L (2020). "Propensity Core Weighting Under Limited Overlap and Model Misspecification." *Statistical Methods in Medical Research*, **29**(12), 3721–3756. doi:10.1177/0962280220940334.

Zubizarreta JR, Li Y (2020). ***sbw**: Stable Balancing Weights for Causal Inference and Estimation with Incomplete Outcome Data*. R package version 1.1.1, URL https://CRAN.R-project.org/package=sbw.