Journal of Statistical Software

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

RCTrep: A R package for evaluation of methods for treatment effect estimation using real world data

Lingjie Shen
Tilburg University

Gijs Geleijnse
IKNL

Maurits Kaptein
JADS

Abstract

Evaluation of methods for treatment effect estimation is important because the performance of a given method depends on a dataset at hand and the comparative performance of different methods can vary widely. However, since we do not observe the true treatment effect for each individual - which is the fundamental problem of causal inference - evaluation using real-world data (RWD) is challenging. In this paper, we provide a R package **RCTrep** to evaluate methods for treatment effect estimation using RWD under minimal assumptions on the sampling mechanism. The assumptions assume RWD and experimental data are two random samples from a population, and hence allow for a fair evaluation of methods for treatment effect estimation. **RCTrep** proposes an evaluation metric in which the "truth" of treatment effect is replaced by an unbiased estimate obtained from experimental data. This article presents the theoretical background of our proposed evaluation approach, designs, and implementation details of the package. **RCTrep** also provides a solution to methods evaluation using aggregated data, and is heterogeneity-aware in error interpretation. **RCTrep** can help select the most reasonable model fitted to a RWD sample, opening up the door to leverage RWD and advanced modeling approaches to inform clinical decision-making.

Keywords: Causal inference, real world data (RWD), evaluation, confounding, privacy-preserved, heterogeneity-aware.

1. Introduction

There is a growing interest in estimating heterogeneous treatment effects using real world data (RWD) (Bica, Alaa, Lambert, and Van Der Schaar 2021; Colnet, Mayer, Chen, Dieng, Li, Varoquaux, Vert, Josse, and Yang 2020; Stuart 2010). Numerous methods tailored for estimating heterogeneous treatment effects using RWD were proposed, capitalizing on ideas of potential outcomes regression (Hill 2011; Hitsch and Misra 2018; Atan, Jordon, and Van der

Schaar 2018), potential outcome difference regression (Wager and Athey 2018), propensity score methods (Xie, Brand, and Jann 2012; Rosenbaum and Rubin 1983), doubly robust methods, and representation learning methods (Yao, Li, Li, Huai, Gao, and Zhang 2018; Johansson, Shalit, Kallus, and Sontag 2020), etc.[1] The increasing availability of RWD which contains large amounts of clinical information about heterogeneous patients and their responses to treatments has also encouraged the development of various flexible and well-performing models. Some notable recent advances include proposals based on lasso (Bloniarz, Liu, Zhang, Sekhon, and Yu 2016), Bayesian additive regression trees (Hill 2011), random forest (Wager and Athey 2018), boosting (Powers, Qian, Jung, Schuler, Shah, Hastie, and Tibshirani 2018), neural networks (Atan *et al.* 2018), and combinations (Künzel, Sekhon, Bickel, and Yu 2019), see (Dorie, Hill, Shalit, Scott, and Cervone 2019; Bica *et al.* 2021) for a recent survey and comparisons.

Different methods built on various promising modeling approaches produce infinite estimates of heterogeneous treatment effects. Choosing the best method from the infinite set of methods using RWD, however, is extremely challenging because the truth of the treatment effect is unobservable - which is the fundamental problem of causal inference (Imbens and Rubin 2015). While literature in the area of the methodology of causal inference has presented promising performance of a method on benchmark datasets, e.g., Twins (Almond, Chay, and Lee 2005), (Infant health development program) IHDP (Hill 2011), Atlantic causal inference conference benchmark (ACICB) (Hahn, Dorie, and Murray 2019), Jobs (LaLonde 1986), however, the performance of a given method depends on a dataset at hand, and the comparative performance of different methods can vary widely across datasets (Dorie *et al.* 2019). For instance, recent studies imply that G_computation method has the best performance among all methods when the outcome model is a good approximation of the "true" DGM, while "debiasing" the method by weighting an estimated inverse propensity score may inflate the variance hence reduce precision (Dorie *et al.* 2019; Loiseau, Trichelair, He, Andreux, Zaslavskiy, Wainrib, and Blum 2022; Le Borgne, Chatton, Léger, Lenain, and Foucher 2021; Chatton, Le Borgne, Leyrat, Gillaizeau, Rousseau, Barbin, Laplaud, Léger, Giraudeau, and Foucher 2020); for some theoretically appealing methods, a sample size of a given context may be smaller than that required by the theory or other regularity conditions of these methods, possibly leading to poor precision than a simple yet biased model (Dorie *et al.* 2019) in the context. Without methods evaluation for a given context, it is unclear which method should be used. Hence, given a large RWD, in the absence of "truth" of treatment effect, *how can we evaluate the performance of different methods in order to select the most reasonable method in that context?*

Despite the richness of literature of methods for heterogeneous treatment effect estimation, a principled approach for the methods evaluation is lacking. Absence of such evaluation approach can hinder the leverage of RWD and advanced methods into practice despite of their appealing advantages. Hence address the problem of the methods evaluation is urgent, and a good open-source implementation of the evaluation is needed. In this paper, we aim to provide an approach to evaluating methods for treatment effect estimation using RWD [2]. We

[1] These approaches are also called S/T-learner, F-leaner, doubly robust learner, domain adaptor learner. See (Jiang, Qi, Zhou, Zhou, and Rao 2021) for a short survey.

[2] Note that our approach is also applicable to evaluate methods for heterogeneous treatment effect obtained from experimental data where estimates of the "truth" of heterogeneous treatment effect defined by $\boldsymbol{x}$ is the simple difference in means of outcomes between treatment and control groups and $w(x) = 1$. We focus on RWD since it is more challenging and is urgently needed. For more elaboration of advantages of RWD over

consider a set of candidate treatment effect models $\mathcal{F} = \{f_1, ..., f_M\}$, where $f(\boldsymbol{x}) : \mathcal{X} \mapsto \tau(\boldsymbol{x})$, $\tau(\boldsymbol{x}) = \mathbb{E}[Y(1) - Y(0) \mid \boldsymbol{x}]$, hence $f(\boldsymbol{x})$ is an estimator of average treatment effect conditional on $\boldsymbol{x}$ [3]. We provide a software package **RCTrep** that makes it easy to try out various methods $f \in \mathcal{F}$ and select the best one using the following evaluation metric:

$$f^* = \arg\min_{f \in \mathcal{F}} \mathbb{L}(\hat{\tau}; f) = \arg\min_{f \in \mathcal{F}} \left(\hat{\tau} - \sum_{\boldsymbol{x}} w(\boldsymbol{x}) f(\boldsymbol{x}) \right)^2, \ s.t. \ \ p(\boldsymbol{x}) = q(\boldsymbol{x}) w(\boldsymbol{x}) \tag{1}$$

where $\hat{\tau}$ is an unbiased estimate of average treatment effect of a population that an experiment represents, $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ are the empirical density of $\boldsymbol{x}$ in experimental data and RWD, $w(\boldsymbol{x})$ is a weight for individuals in RWD with characteristics $\boldsymbol{X} = \boldsymbol{x}$ so that the weighted distribution of covariates in RWD and distribution of covaraites in experimental data is balanced [4]. Although our method for evaluation is not without assumptions, which we will introduce in the section 3, to the best of our knowledge, we think our method is the best the reason for which we will elaborate in section 1.1 and section 1.2.

This paper begins by formulating the problem of evaluation of methods for treatment effect estimation in the section 2; section 3 illustrates the assumptions that allow for the evaluation using RWD; section 4 provides an overview of the R package **RCTrep** and the first working example; section 5 introduces subclasses of core classes in **RCTrep**, and basic usage of the package in terms of three steps, namely, *identification*, *estimation*, and *evaluation*; section 6 elaborates core classes in **RCTrep**; four working examples are demonstrated in section 7; finally, the paper ends up with the discussion in section 8.

1.1. Related work

experimental data for treatment effect estimation, see our previous work (Shen, Visser, de Wilt, Verheul, van Erning, Geleijnse, and Kaptein 2020).

[3]Note that $f(\boldsymbol{x})$ can be G_computation, IPW, and doubly robust estimators for $\tau(\boldsymbol{x})$. Hence $f(\boldsymbol{x})$ is a function of an estimator of potential outcomes parameterized by $\boldsymbol{\beta}$, or a function of an estimator of treatment parameterized by $\boldsymbol{\alpha}$, or a function of an estimator of outcome and an estimator of treatment parameterized by $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$, where $\boldsymbol{X}$ is a vector of confounders, specified by `confounders_treatment_name` in the **RCTrep**. For instance, for G_computation, $f(\boldsymbol{x}; \boldsymbol{\beta}) = \hat{\mathbb{E}}[Y \mid \boldsymbol{x}, z = 1] - \hat{\mathbb{E}}[Y \mid \boldsymbol{x}, z = 0] = p(\boldsymbol{x}, 1; \hat{\boldsymbol{\beta}}) - p(\boldsymbol{x}, 0; \hat{\boldsymbol{\beta}})$; for IPW, $f(\boldsymbol{x}) = \hat{\mathbb{E}}\left[\frac{YZ}{\pi(\boldsymbol{x}; \hat{\boldsymbol{\alpha}})}\right] - \hat{\mathbb{E}}\left[\frac{Y(1-Z)}{1-\pi(\boldsymbol{X}; \hat{\boldsymbol{\alpha}})}\right] = \sum_{i:z_i=1} W_i Y_i Z_i - \sum_{i:z_i=0} W_i Y_i (1 - Z_i)$, where $\boldsymbol{x}_i = \boldsymbol{x}$,

$$W_i = \begin{cases} \frac{\frac{1}{\pi(\boldsymbol{x}_i; \hat{\boldsymbol{\alpha}})}}{\sum_{i:Z_i=1} \frac{1}{\pi(\boldsymbol{x}_i; \hat{\boldsymbol{\alpha}})}} & Z_i = 1 \\ \frac{\frac{1}{1-\pi(\boldsymbol{x}_i; \hat{\boldsymbol{\alpha}})}}{\sum_{i:Z_i=0} \frac{1}{1-\pi(\boldsymbol{x}_i; \hat{\boldsymbol{\alpha}})}} & Z_i = 0. \end{cases}$$

and $\hat{\boldsymbol{\beta}}$ is an estimator of parameters in a model for a conditional outcome, and $\hat{\boldsymbol{\alpha}}$ is an estimator of parameters in a model for propensity score respectively, and both of which can have infinite dimensions, depending on model constraints, regulation conditions and a sample. Different classes of estimators for average treatment effect combined with various modeling choices lead to an infinite number of methods, and hence an infinite number of estimates of $\mathbb{E}[Y(1) - Y(0) \mid \boldsymbol{X}]$.

[4]Note that in practice, weighting is to balance covariates between experimental data and RWD, the covariates can be smaller than confounders. Under the ignorability assumptions on the sampling mechanism, only predictive variable that vary between samples can lead to difference in estimates of average treatment effect between two samples. In our context, predictive variable - also called effect modifier - is a variable which has interaction effect with the treatment on the outcome. Treatment effect shows heterogeneity on levels of a predictive variable. On the other hand, if treatment effect is homogeneous, meaning there is no predictive variable, then treatment effect is a constant, and hence is readily comparable, $w(\boldsymbol{x}) = 1$.

Currently, there are a large body of softwares for heterogeneous treatment effect estimation, for instance, the Python library **CausalML** provides multiple methods for average and heterogeneous treatment effect estimation; the Python library **DoWhy** provides an end-to-end implementation for causal inference. However, studies on the evaluation of methods for treatment effect estimation using RWD are not adequately investigated. Powers *et al.* (2018) evaluates the performance of outcome regression methods based on the factual sample only, which is obviously not valid. Works by Wendling, Jung, Callahan, Schuler, Shah, and Gallego (2018), Alaa and Van Der Schaar (2019), Schuler, Jung, Tibshirani, Hastie, and Shah (2017), Franklin, Schneeweiss, Polinski, and Rassen (2014), the R package **MethodEvaluation**[5] (Schuemie, Cepeda, Suchard, Yang, Tian, Schuler, Ryan, Madigan, and Hripcsak 2020), the Python package **Causality-Benchmark**[6](Shimoni, Yanover, Karavani, and Goldschmnidt 2018), and Python package **JustCause**[7], use simulated "truth" of treatment effect to evaluate methods for treatment effect estimation using RWD. These approaches simulate the "truth" based on the observed data, implicitly assuming no unmeasured confounders. The simulated truth of the treatment effect can be problematic in case unmeasured confounders occurred. In the presence of an unobserved confounder, although these approaches can simulate the distribution of observed data as close as possible, a treatment group and a control group with the same characteristics still have a systematic difference in the distribution of an unobserved variable that is predictive to outcome, which is unknown and can not be simulated properly, and hence the simulated "truth" is still biased. Evaluating methods for treatment effect estimation using a possibly biased estimate of "truth" is not valid. We provide an overview of existing packages for evaluation of methods for treatment effect estimation in Table 1. For more elaborated overview on evaluation methods and measures for causal inference methods, see a recent survey by Cheng, Guo, Moraffah, Sheth, Candan, and Liu (2022).

There are some similar studies under the same assumptions in **RCTrep**, however, focus on generalization or transportation of treatment effect estimates. R packages **ExtendingInferences** (Dahabreh, Robertson, Steingrimsson, Stuart, and Hernan 2020), **generalize** (Ackerman, Lesko, Siddique, Susukida, and Stuart 2021), **genRCT** (Dong, Yang, Wang, Zeng, and Cai 2020), and **generalizing** (Cinelli and Pearl 2021) focus on generalizing treatment effect estimates of a randomized controlled trial (RCT) to a target population; R packages **transport** (Rudolph, Schmidt, Glymour, Crowder, Galin, Ahern, and Osypuk 2018) and **causaleffect** (Tikka and Karvanen 2018) aim to transport estimates of one population to another population. Approaches used in these softwares are similar to that of **RCTrep**, however, **RCTrep** is different from them in respective of the motivation. **RCTrep** aims to evaluate methods for treatment effect estimation using RWD, and choose the most reasonable one given the data at hand accordingly. The package provides a principled and viable approach to methods evaluation and selection using RWD, which to some extent may help address the fundamental problem of causal inference.

[5]https://github.com/OHDSI/MethodEvaluation

[6]https://github.com/IBM-HRL-MLHLS/IBM-Causal-Inference-Benchmarking-Framework. The package simulates treatment assignment, factual outcome, and counterfactual outcomes using a defined data generation process for a given dataset, and split the generated dataset to a training set for fitting models for treatment effect prediction, and a validation set for model evaluation respectively.

[7] https://github.com/inovex/justcause/blob/master/src/justcause

Task		Package			
		MethodEvaluation[5]	**CausalityBenchmark**[6]	**JustCause**[7]	**RCTrep**
Estimation Methods	propensity score	✓		✓	✓
	G_computation	✓			✓
	Doubly robust	✓		✓	✓
Evaluation level	population		✓	✓	✓
	sub-population	✓	✓	✓	✓
Metrics	(R)MSE	✓	✓	✓	✓[8]
	PEHE				
	Bias		✓		
	confidence interval		✓		✓
	coverage	✓	✓		
	AUC	✓			
	mean precision	✓			
	type 1 error	✓			
	type 2 error	✓			
	Regulatory agreement[9]				✓
	Estimate agreement[10]				✓
Truth	synthetic truth[11]	✓	✓	✓	
	unbiased estimate [12]				✓

Table 1: Comparisons of packages for evaluation of methods for treatment effect estimation with a focus on the methods for treatment effect estimation, level of estimand, evaluation metrics, and the generation of "truth" of individual treatment effect.

[8]In **RCTrep**, we compute the squared difference between an unbiased estimate from experimental data and an estimate from RWD as *pesudo MSE*, since we don't know the "truth", we compute pesudo MSE on both population and sub-population levels.

[9]Regulatory agreement is defined as the ability of a method using RWD to replicate the direction and statistical significance of the unbiased estimate using experimental data(Franklin, Pawar, Martin, Glynn, Levenson, Temple, and Schneeweiss 2020).

[10]Estimate agreement is defined as an estimate of treatment effect using RWD that lies within the 95% confidence interval for the unbiased estimate of the "truth" using experimental data(Franklin *et al.* 2020).

[11]Simulate the truth of individual treatment effect under the assumption of no unmeasured confounders

[12]Use unbiased estimate from a RCT as the "truth" for RWD under the assumption experimental data and RWD are two random samples from the same population

1.2. Strength of our work

There are several important areas where **RCTrep** makes contributions to:

1. We carefully identify conditions under which the expectation of estimates using RWD can be identical to the expectation of estimates using experimental data, and propose an approach to evaluating methods for treatment effect estimation, see appendix E for more elaboration of the conditions. Unlike previous research which simulates the "truth" of treatment effect implicitly under the assumption of no unmeasured confounders (Wendling *et al.* 2018; Alaa and Van Der Schaar 2019; Schuler *et al.* 2017; Franklin *et al.* 2014; Schuemie *et al.* 2020; Shimoni *et al.* 2018), we make assumptions on the sampling mechanism, i.e., we assume the weighted RWD and experimental data are two random samples from the same population. We do not make assumptions on ignorability of the treatment assignment mechanism as these are the assumptions to validate. Hence, **RCTrep** offers a more principled and fair approach to the evaluation of methods than existing evaluation approaches. See the section 3 for more elaboration of the theoretical background.

2. Different from the R package **MethodEvaluation** - which is the only R package for evaluating methods for treatment effect estimation using RWD by synthesizing the "truth" - we evaluate the performance of methods on both population and sub-population level, allowing a deeper understanding of how a method expresses error. For instance, a high bias method may have the lowest error at a population level but has no sufficient statistical power at sub-population levels.

3. **RCTrep** provides additional functions to support analysis. **RCTrep** provides functions to diagnose two overlap assumptions, i.e., a) overlap of covariates between treatment and control groups, and b) overlap of covariates between RWD and experimental data. **RCTrep** provides functions to diagnose model assumptions because the assumptions can imply plausibility of the assumption of no unmeasured confounders. For instance, we diagnose assumptions of outcome regression model to identify if there is omitted variable bias due to an unmeasured confounder; for the propensity-score method, we diagnose weighted covariates balance between treatment control groups.

4. **RCTrep** provides a solution to method evaluation in case that individual level data are not allowed to share. This solution can generate the same results in case full data sets are allowed to share while preserving privacy. See section 7.2 for a working example.

5. **RCTrep** is heterogeneity-aware. Instead of learning parameters of a model by minimizing a distance measure between the "truth" and estimates from the model, **RCTrep** allows a model to fit to RWD using factual loss defined as a distance measure between factual outcome and the estimate of factual outcome first, estimate treatment effect for each individual accordingly, and choose the best model according to the evaluation metric in equation 4 [13]. If all methods for treatment effect estimation show reasonably coherent results and users are willing to assume that all confounders are properly

[13] If we evaluate a method for treatment effect estimation using experimental data, then $w(\boldsymbol{x}) = 1$ because the sample used for modeling treatment effect and the sample used for generating the "truth" is the same sample. Learning a model for treatment effect using experimental data is the same as supervised learning by minimizing a loss $L(\hat{\tau}(\boldsymbol{x}_i), f(\boldsymbol{x}_i))$ where $\hat{\tau}(\boldsymbol{x}_i)$ is the difference in means of the outcome between treatment and control groups with the same characteristics $\boldsymbol{x}$.

adjusted, then the residual significant difference in estimates between RWD and experimental data can be attributed to an unobserved effect modifier that varies between RWD and experimental data. For more elaborations on conditions under which we can identify the existence of an possible effect modifier, see the section E.3 of the appendix E.

6. **RCTrep** uses object-oriented programming based on class R6 so that the package is maintainable, reusable, and easily extensible.

1.3. Limitation of our work

Despite appealing advantages of **RCTrep** over other packages, **RCTrep** is not without limitations. First, how confident we are about the performance of methods depends on the quality of experimental data. If an unbiased estimate of the "truth" using experimental data has a large uncertainty, even though most estimates using RWD can lie within the confidence interval of the estimate of the "truth", it is still hard to evaluate the comparative performance of different methods. However, on the other hand, if estimates from RWD can lie within the confidence interval of the "truth", then to some extent we ascertain that those estimates are not far from the "truth", and hence we can still leverage the RWD and those methods to provide evidence.

Second, we assume weighted RWD and experimental data are two random samples from a population. Only under this assumption, we can evaluate methods for treatment effect estimation using RWD, otherwise, we can not ascertain whether a difference in estimates using experimental data and RWD is because of unobserved confounders or unobserved effect modifiers, or both. Without the assumption, evaluation of methods for treatment effect estimation using RWD is not possible and evidence from RWD is not valid despite the appealing advantages of RWD and modeling approaches. Hence, readers should bear in mind that our evaluation approach and error interpretation depend on the assumption. Although the assumption is untestable, instead of making assumptions on the treatment assignment mechanism, we think our assumption is much weak, and **RCTrep** is the most viable and fair R package for evaluation of methods for treatment effect estimation using RWD.

1.4. Demonstration of usage

We can call the function `RCTREP()` in **RCTrep** to implement methods evaluation as follow:

```
output <- RCTREP(TEstimator="G_computation", SEstimator = "Exact",
                 source.data=source.data, target.data=target.data,
                 vars_name=list(confounders_treatment_name=c("x1","x2","x3","x4","x5","x6")
                                treatment_name=c("z"),
                                outcome_name=c("y")),
                 confounders_sampling_name = c("x2","x6"))

fusion <- Summary$new(output$target.obj,
                      output$source.obj,
                      output$source.rep.obj)
```

```
fusion$plot()
```

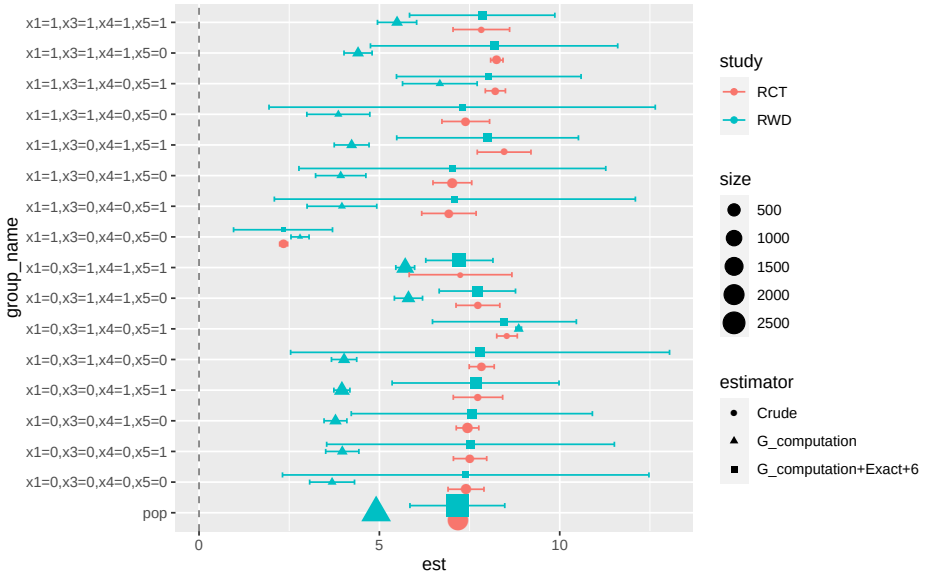


Figure 1: Treatment effect estimation using **RCTrep**. We use G_computation method with a default modeling approach to adjust the treatment assignment mechanism, i.e., generalized linear regression, and use the exact matching to adjust the sampling mechanism. We select `x1,x2,x3,x4,x5,x6` as confounders in treatment assignment mechanism, and `x2,x6` as confounders in sampling mechanism. By default, confounders in the sampling mechanism are the same as confounders in the treatment assignment mechanism. In this working example, since `x2,x6` are only effect modifiers, hence they are the minimal set of confounders in the sampling mechanism to allow for comparison of treatment effect estimation between RWD and experimental data.

where `TEstimator` specifies a method to correct for the treatment assignment mechanism; `SEstimator` specifies a method to correct for the sampling mechanism, so the experimental data and weighted RWD are assumed as random samples from the same population, and hence the unbiased estimate of the "truth" using the experimental data can be regarded as the unbiased estimate of the "truth" in the context of RWD setting; `target.obj` and `source.obj` specify an experimental dataset and a RWD dataset; `vars_name` specifies variable names of treatment, outcome, confounders due to treatment assignment mechanism; `confounder_sampling_mechanism` specifies confounders due to sampling mechanism; `...` are optional arguments to define, e.g., model choices, tuning parameters, and variables for subgroups selection for subgroup-level evaluation, etc. See section 7 for more working examples.

2. Problem setup

In this section, we demonstrate our framework for estimating treatment effects and the general approach to evaluating the methods performance.

2.1. Treatment effect estimation

We consider potential outcomes framework for modeling heterogeneity treatment effect using experimental data and RWD (Imbens and Rubin 2015). Let $\boldsymbol{X}$ denote d-dimensional vector of covariates, $z \in \mathcal{Z} = \{0, 1\}$ denote binary treatment indicator where 0 and 1 denote treatment and control, respectively; let $Y \in \{0, 1\}$ denote the binary outcome of interest, $Y(z)$ denote the potential outcome had the unit received treatment z. The observed outcome of unit i under the received treatment Z_i can be denoted as $Y_i = Z_i Y(1) + (1 - Z_i)Y_i(0)$. Then the individual-level treatment effect is denoted as simple difference $\tau_i = Y_i(1) - Y_i(0)$, heterogeneous treatment effect is defined as $\tau(\boldsymbol{X}) = \mathbb{E}[Y(1) - Y(0) \mid \boldsymbol{X}]$, and average treatment effect is defined as $\tau = \mathbb{E}[\tau(\boldsymbol{X})]$.

2.2. Method evaluation

We now consider a set of candidate treatment effect models $\mathcal{F} = \{f_1, ..., f_M\}$, where $f : \mathcal{X} \mapsto \tau(\boldsymbol{X})$. These may include, for example, different methods (G-computation, IPW, doubly robust) combined with different modeling choices (BART, gaussian process, causal forest), and different hyperparameter settings of one model, etc. The accuracy of a model f for average treatment effect estimation is characterized by a distance measure as follows:

$$\mathbb{L}(\tau; f) = \left(\tau - \sum_{\boldsymbol{x}} f(\boldsymbol{x}) p(\boldsymbol{x}) \right)^2 \tag{2}$$

The selected best model is derived based on

$$f^* = \arg\min_{f \in \mathcal{F}} \mathbb{L}(\tau; f) \tag{3}$$

Since τ is not observed, hence the evaluation metric in equation 2 is not observed, hindering evaluation of methods f for treatment effect estimation. In the following section, we provide our approach to evaluating methods for treatment effect estimation under minimal assumptions.

3. Methods evaluation via our proposed approach

In this section, we demonstrate our approach to evaluating models for treatment effect estimation. In section 3.1, we start from elaborating why an estimate of treatment effect using experimental data can be regarded as an unbiased estimate of the "truth" (i.e., τ) of treatment effect of a population that the experimental data represents; in section 3.2 we elaborate how to use the estimate from experimental data as the "truth" in the setting of RWD by introducing the ignorability assumption on sampling mechanism. Our approach indicates under which conditions the expectation of estimates of treatment effect from RWD can be identical to that from experimental data. In the following, we elaborate assumptions for treatment effect identification and methods for treatment effect estimation, illustrating why an estimate using experimental data is an unbiased estimate of the "truth" of the population that the experimental data represents.

3.1. Why an estimate of treatment effect using experimental data is an unbiased estimate of the "truth"?

By definition, treatment effect for each individual is not observed and can only be estimated. The following two assumptions allow for unbiased estimate of treatment effect:

Assumptions 1 Z-ignorability: $Y(1), Y(0) \perp\!\!\!\perp Z \mid \boldsymbol{X}$

Assumptions 2 Z-overlap: $0 < P(Z = 1 \mid \boldsymbol{X}) < 1$

where $\boldsymbol{X}$ is a set of confounders that isolate dependence between covariates and treatment. The assumption of $Z-$ignorability implies that conditional on $\boldsymbol{X}$, treatment is independent of potential outcomes, hence the change in observed outcomes between groups is only attributed to the treatment. The assumption of $Z-$overlap guarantees that there is a sufficient number of observations with characteristics $\boldsymbol{X} = \boldsymbol{x}$ in both groups. Given these two assumptions, we can obtain an unbiased estimate of treatment effect. In a RCT with experimental data, treatment is randomly assigned to each individual so that the treatment is independent of (both measured and unmeasured) covariates between groups, hence identification assumption holds given an empty set. In an observational study with RWD, estimate of treatment effect is unbiased only when there are no unmeasured confounders. The assumption of no unmeasured confounder(s) is however untestable, hence the estimate of average treatment effect using RWD is not valid. Although there is an debate on whether estimates of treatment effect using experimental data are likely to be closer to the "truth" than those estimated using RWD (Deaton and Cartwright 2018), we think a well-designed study with sufficient sample size can yield not only an unbiased but also precise estimate of treatment effect. Three classes of methods can be used to estimate treatment effect: G-computation, inverse propensity score method, doubly robust method (Chatton *et al.* 2020). We analyze variance of three estimators in the appendix B.

3.2. How to use the unbiased estimate of the "truth" from experimental data as an unbiased estimate of the "truth" of RWD?

Once we have an unbiased estimate of the "truth" using experimental data, how to use the estimate as a benchmark to evaluate methods for treatment effect estimation using RWD? In this section, we introduce assumptions and methods that allow for fair evaluation of methods for treatment effect estimation using RWD. We regard "selection into a RCT" as an intervention. Units selected to a RCT is regarded as "receiving an active treatment" and units selected to an observational study is regarded as "receiving a control treatment". Then analogy to assumptions and methods in section 3.1 to allow for fair comparison between treatment and control groups, we can use similar assumptions on sampling mechanism to allow for fair comparison between RWD and experimental data as follows:

Assumptions: S-ignorability: $Y(1), Y(0) \perp\!\!\!\perp S \mid \boldsymbol{X}$

Assumptions: S-overlap: $0 < P(S = 1 \mid \boldsymbol{X}) < 1$

The first assumption demonstrates conditioning on covariates $\boldsymbol{X}$, potential outcomes are exchangeable between samples $S = 1$ (i.e., experimental data) and $S = 0$ (i.e., RWD). Given the assumption, RWD and experimental data can be regarded as two random samples from the same population. The second assumption, i.e., S-overlap, similar to Z-overlap, guarantees that there is a sufficient number of observations with characteristics $\boldsymbol{X} = \boldsymbol{x}$ in both RWD and experimental data. Given these two assumptions, within subgroup with $\boldsymbol{X} = \boldsymbol{x}$, there is

no unobserved variable that varies between observations in RWD and observations in experimental data, and hence estimates of potential outcomes given $\boldsymbol{X}$ are directly comparable. See appendix G for more elaboration of these two assumptions.

Given these two assumptions, we can use multiple methods to adjust for the sampling mechanism [14] In **RCTrep**, we use weighting to adjust the sampling mechanism so that covariates between weighted RWD and experimental data are balanced. We introduce three methods for computing weight: 1) inverse sampling score weighting (ISW), analogy to IPW, which weights units according to their inverse (known or estimated) sampling scores, so that covariates are balanced between two samples, two samples are exchangeable and hence selection indicator S is independent of covariates and thereof independent of potential outcomes; 2) weights based on exact matching on $\boldsymbol{X}$; 3) weights based on subclassification according to either covariates $\boldsymbol{X}$ or a sampling score. More methods can be used, for instance, balancing-based weighting approach which implemented in (Chattopadhyay, Hase, and Zubizarreta 2020). The approach estimates weights for each individual by minimizing a distance measure between weighted distribution of $\boldsymbol{X}$(or $\phi(\boldsymbol{X})$) in RWD and the distribution of $\boldsymbol{X}$ (or $\phi(\boldsymbol{X})$) in experimental data, where $\phi(\boldsymbol{X})$ is a basis function of $\boldsymbol{X}$. In general, all weighting methods for adjusting the sampling mechanism need to estimate either a sampling score or density of $\boldsymbol{X}$ or $\phi(\boldsymbol{X})$. See C for elaboration of methods for adjusting the sampling mechanism used in **RCTrep** and a brief introduction of optimization-based method.

3.3. Putting all together

Given above four assumptions, we can replace $f(\boldsymbol{x})p(\boldsymbol{x})$ in the evaluation metric in the equation 2 by $w(\boldsymbol{x})f(\boldsymbol{x})$, and replace τ by $\hat{\tau}$ where $\hat{\tau}$ is the unbiased estimate of average treatment effect using experimental data, and $f(\boldsymbol{x})$ is estimated conditional treatment effect using RWD. The proposed evaluation metrics is as follows:

$$\mathbb{L}(\hat{\tau}; f) = \left(\hat{\tau} - \sum_{\boldsymbol{x}} w(\boldsymbol{x}) f(\boldsymbol{x})\right)^2, \ s.t. \ \ p(\boldsymbol{x}) = q(\boldsymbol{x}) w(\boldsymbol{x}) \tag{4}$$

where $w(\boldsymbol{x})$ is the weight for units in RWD with charactersitics $\boldsymbol{X}$ so that the weighted $\boldsymbol{x}$ in RWD and $\boldsymbol{x}$ in experimental data is the same distributed [15]. Then we select the best model $f^* = \arg\min_{f \in \mathcal{F}} \mathbb{L}(f; \tau)$. In practice, given minimum requirements to allow for treatment effect comparison and methods evaluation, we can compute weight for each individual with characteristics $\boldsymbol{X}$ according to effect modifiers (variables that are predictive to treatment effect and vary between samples) (Egami and Hartman 2018), which can not only allow for methods evaluation but also avoid inflating the variance of $w(\boldsymbol{X})f(\boldsymbol{X})$. Note that beyond evaluating f on population level, we also evaluate on sub-population level, which can quantify the ability of a model f to capture the heterogeneity of treatment effect, and can help us understand how f express error. In the following, we will move from math to code, we will first have an overview of the package **RCTrep**, and then demonstrate implementation of **RCTrep**.

[14] According to (Rosenbaum and Rubin 1983), ignorability assumptions hold conditioning on a balancing score; sampling score (anology to propensity score) is the coarsest balancing score, $\boldsymbol{x}$ is the finest balancing score, any score that is "finer" than sampling score is a balancing score based on which we can implement weighting/matching/subclassification methods to adjust the sampling mechanism.

[15] At least variables predictive to treatment effect, namely, effect modifiers, are the same distributed between RWD and experimental data. For binary variable X, the difference in weighted mean of X in RWD and the mean of X is small

4. Overview of software

The current section offers an overview of **RCTrep** implementation. We illustrate core classes and functions in the package, and demonstrate a first working example to implement evaluation of treatment effect estimation using **RCTrep**. In the next section, we introduce the implementation of **RCTrep** and predefined classses in the package.

4.1. Implementation

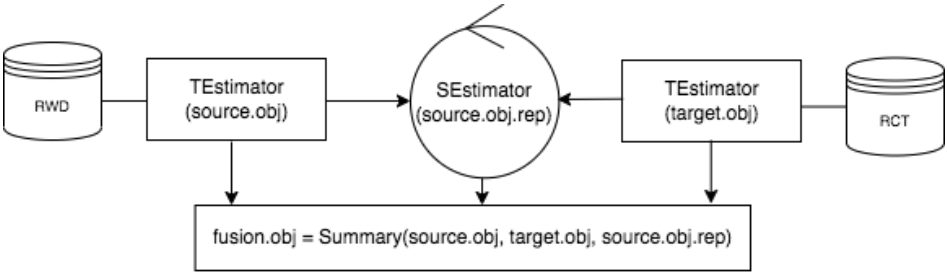


Figure 2: Diagram of **RCTrep** basic structure.

We provide an overview of implementaion of **RCTrep** in Figure 2. The following three core classes form the backbone of the package:

1. `TEstimator`: R6 class `TEstimator` is the parent class of all **RCTrep** `TEstimator` subclasses. It estimates average treatment effect of a sample, and diagnoses Z-overlap assumption, and diagnoses model fit for outcome regression or diagnoses covariate balance for a propensity score model.

2. `SEstimator`: R6 class `SEstimator` is the parent class of all **RCTrep** `SEstimator` subclasses. It is responsible for computing weights, so that the weighted covariates in `source.obj` and covariates in `target.obj` are balanced and hence estimates are comparable. It diagnoses the S-overlap assumption, presents weights distribution, and shows distance measure between weighted covariates in `source.obj` and the covariates in `target.obj`.

3. `Summary`: R6 class `Summary` combines estimates from an object of class `TEstimate` and/or an object of class `SEstimate`, and plots and evaluates estimates of average treatment effect and heterogeneous treatment effect. The number of objects of class `TEstimator` or `SEstimator` passed to its constructor is not limited.

4.2. A first example

In this section, we demonstrate a first simple example to evaluate methods for treatment effect estimation using **RCTrep**. We use one method to correct for the treatment assignment mechanism of RWD, namely, G-computation using linear regression, and one method to correct for the sampling mechanism of RWD, namely, exact matching. More methods can be found in section 5.

A first model evaluation

Building on the introduction of three core classes in **RCTrep** in the previous section, we can now put together the following lines to implement the method evaluation:

```
library(RCTrep)
source.data <- RCTrep::source.data
target.data <- RCTrep::target.data

vars_name <- list(confounders_treatment_name=c("x1","x2","x3","x4","x5","x6"),
                  treatment_name=c('z'),
                  outcome_name=c('y')
)
source.obj <- TEstimator_wrapper(
  Estimator = "G_computation",
  data = source.data,
  name = "RWD",
  vars_name = vars_name,
  outcome_method = "glm",
  outcome_formula = y ~ x1 + x2 + x3 + z + z:x1 + z:x2 +z:x3+ z:x6,
  data.public = TRUE
)

target.obj <- TEstimator_wrapper(
  Estimator = "Crude",
  data = target.data,
  name = "RCT",
  vars_name = vars_name,
  data.public = TRUE
)

confounders_sampling <- c("x1","x2","x3","x4","x5","x6")
source.obj.rep <- SEstimator_wrapper(
                    estimator="Exact",
                    target.obj=target.obj,
                    source.obj=source.obj,
                    confounders_sampling=confounders_sampling)
source.obj.rep$EstimateRep(stratification = c("x1","x3","x4","x5"),
                           stratification_joint = TRUE)
```

In these lines we start out by instantiating an object of class `TEstimator` for a study with RWD and a study with experimental data. We call `TEstimator_wrapper()` function to initialize the object `source.obj` by specifying the following arguments:

1. `Estimator`: specifying a method for treatment effect estimation. `TEstimator_wrapper()` will initialize the `TEstimator` subclass according to the specified method. For in-

stance, if `Estimator="G_computation"`, then the function initializes `TEstimator` subclass `G_computation` and returns the initialized object.

2. `data`: a `data.frame` with n rows and p columns, each row contains variables of characteristics, treatment, and outcome of each observation.

3. `name`: a character indicating object study name;

4. `vars_name`: a list containing three character vectors with the first element `confounders_treatment_nam` indicating confounding variable names, the second element `treatment_name` indicating a treatment variable name, and the third element `outcome_name` indicating an outcome variable name;

These arguments are mandatory to be specified, the remaining variables are optional:

1. `outcome_method/treatment_method`: a character indicating modeling approach for different subclasses of `TEstimator`, see **caret** for more options of models; if `Estimator` is `"IPW"`, then specify the argument `treatment_method`, if `Estimator` is `"DR"`, then specify both arguments `outcome_method` and `treatment_method`; the default method for `outcome_method` and `treatment_method` is `"glm"`;

2. `outcome_formula/treatment_formula:` a `formula` indicating a model specification for `outcome_method/treatment_method` which needs a formula. If `outcome_method/treatment_method` is `"glm"`, then the default formula for `outcome_formula` is main effects of all `confounders_treatment_n` and `treatment_name`, and the default formula for `treatment_formula` is main effects of all `confounders_treatment_name`;

3. `data.public`: a logical indicating whether publishing a full dataset; default is `TRUE`.

An argument list for `Estimator_wrapper` varies for different subclasses. Then we instantiate `TEstimator` subclass `Crude` for RCT dataset as `target.obj`.

Next, we instantiate a `SEstimator` subclass `SEexact` as `source.obj.rep` by calling the function `SEstimator_wrapper()`. The arguments list for the function is:

1. `estimator`: a character indicating a method for computing weights to balance distributions of `confounders_sampling_name` between `source.obj` and `target.obj`; then the wrapper function initializes a `SEstimator` subclass accordingly; each subclass implements an unique method for computing weight $w(\boldsymbol{X})$;

2. `target.obj` and `source.obj`: `target.obj` indicates an object of which the estimate of treatment effect is regarded as the unbiased estimates of the "truth"; `source.obj` indicates an object of which the method for treatment effect estimation is to be evaluated.

3. `confounders_sampling_name`: a character vector of variable names; the weighted distribution of `confounders_sampling_name` in `source.obj` and those in `target.obj` should be balanced using the weighting method specified in `estimator`.

Finally, we call `EstimateRep()` - the core function of the instantiated object `source.obj.rep` to compute the weighted estimates $\sum_{\boldsymbol{x}} w(\boldsymbol{x})f(\boldsymbol{x})$ on both population and subpopulation

level - which implements weight computation using `Exact` weighting approach. The resulting weighted distribution of `counfounders_sampling_name` in `source.obj` and those in `target.obj` are balanced, i.e., difference in means of each covariate is small (within 1.96 standard deviation). Two optional arguments for the function `EstimateRep()` are specified, which are used to select subpopulations of which the function computes the weighted treatment effect:

1. `stratification`: a character vector containing covariate names. `EstimateRep()` estimates the treatment effect for each level defined by variables in `stratification`; default value is `confounders_sampling_name`;

2. `stratification_joint`: a logical value, if `TRUE`, then subpopulation is defined according to joint combination of all variables in `stratification`; otherwise, then subpopulation is defined by levels of each variable in `stratification`.

Then in each subpopulation, the weighted distribution of `confounders_sampling_name` in `source.obj.rep` and the distribution of `confounders_sampling_name` in `target.obj` should be balanced.

A main loop that relates one to one to the five steps of evaluating methods for treatment effect estimation using RWD:

1) clients call `TEstimator_wrapper` to initialize a `TEstimator` subclass for RWD as `source.obj` and a `TEstimator` subclass for experimental data as `target.obj`. The objects fit model(s), estimate average treatment effect and heterogeneous treatment effect;

2) clients call `SEstimator_wrapper` to initialize a `SEstimator` subclass as `source.obj.rep` by assigning `source.obj` and `target.obj` to the function. `source.obj` and `target.obj` communicate in this step. `source.obj.rep` combines RWD from `source.obj` and experimental data from `target.obj`, and estimates weighted average treatment effect and weighted heterogeneous treatment effect to compare with `target.obj`.

3) clients can call `source.obj.rep$EstimateRep(...)`, specifying two arguments `stratification` and `stratification_joint` to the function. The function estimates weighted estimates of average treatment effect of subgroups stratified by these two arguments.

4) clients initialize a `Summary` class as `fusion` by assigning three objects, i.e., `source.obj`, `target.obj`, and `source.obj.rep` to the function. `fusion` aggregates average treatment effect and heterogeneous treatment effect, and plot and print estimates. The object evaluates methods for treatment effect estimation in `source.obj` and `source.obj.rep` and print evaluation metrics on both population and sub-population level. The number of objects of class `TEstimator` and `SEstimator` is not limited. Clients can diagnose possible reasons that cause differences in estimates between `source.obj.rep` and `target.obj` by calling `source.obj$diagnosis_t_overlap()`, `source.obj$summary()`, and `source.obj.rep$diagnosis_s_overlap()`. For instance, near violation of Z-overlap assumption can lead to high variance of estimates using RWD, near violation of S-overlap assumption can also lead to high variance of weighted estimates using RWD, small sample size of a subgroup as well as small number of outcome of interest can lead to underfit of parameters for that subgroup.

5) (Optional) Then repeat step 3) and step 4) to evaluate estimates on sub-population level defined by different `stratification`.

Results of methods evaluation

On completion of all class instantiation, before methods evaluation, we can first diagnose Z-overlap and S-overlap assumptions:

```
> source.obj$diagnosis_t_overlap()
> target.obj$diagnosis_t_overlap()
> source.obj.rep$diagnosis_s_overlap()
```

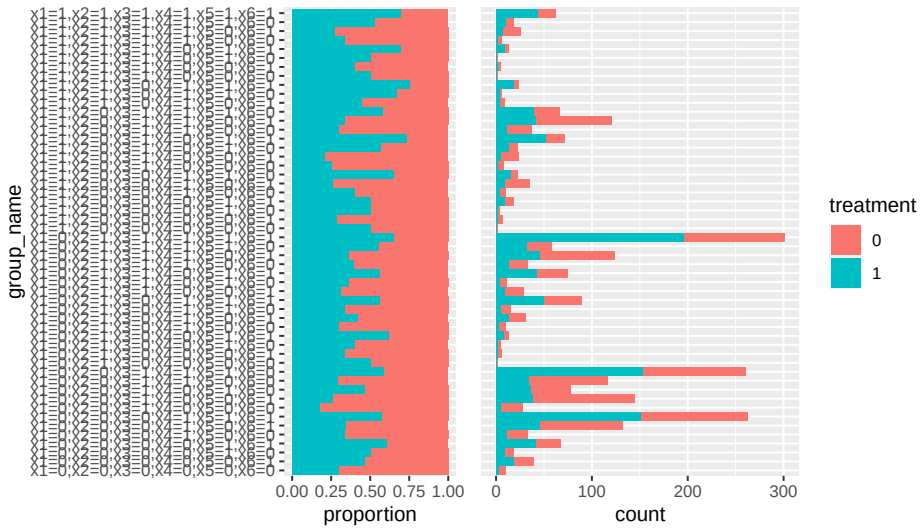


Figure 3: The distribution of observations receiving treatment and control in each level of `confounders_treatment_name` in `source.obj`

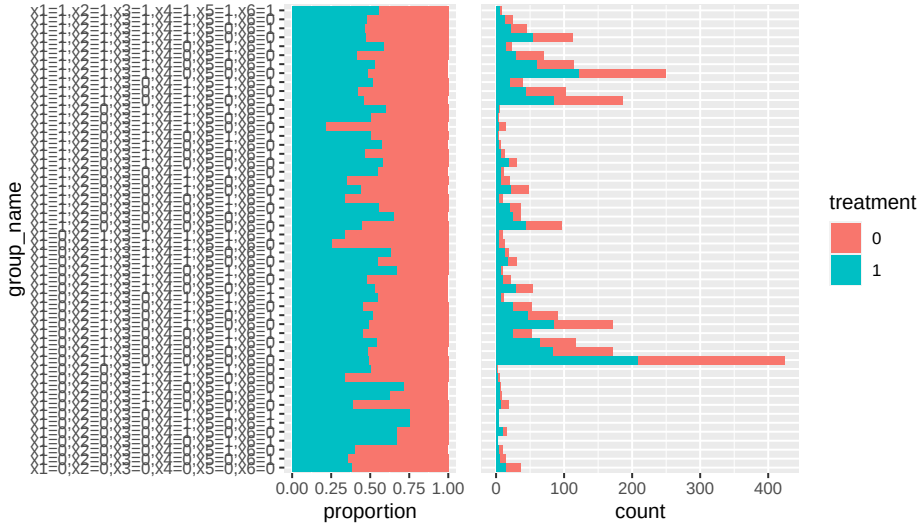


Figure 4: The distribution of observations receiving treatment and control in each level of `confounders_treatment_name` in `target.obj`

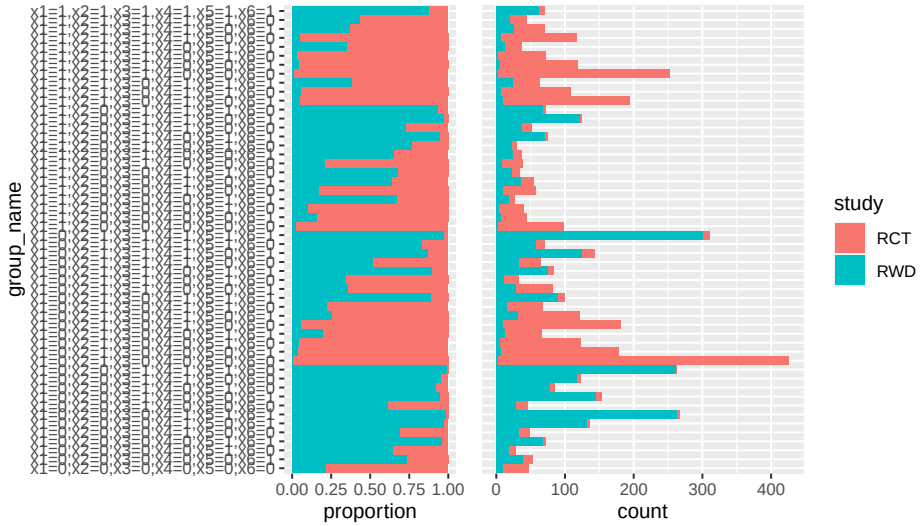


Figure 5: The distribution of observations selected to RWD and experimental data in each level of `confounders_sampling_name`

The class of `source.obj` is `TEstimator` subclass `G_computation`, hence the object fits an specified outcome regression model, i.e., generalized linear regression, and predicts potential outcomes for each observation. In outcome regression model, we implicitly assume $y_i \sim \mathcal{N}(f(\boldsymbol{x}_i), \sigma^2)$, $\mathbb{E}[\epsilon_i \mid \boldsymbol{x}_i] = 0$, and $\sigma_i^2 = \mathbb{V}(\epsilon_i)$ is a constant. These assumptions indicate no omitted variable bias and no heterogeneity in error (i.e., the regression estimator is the most efficient, no other variables exist that can explain away variation of the error term). No omitted variable bias guarantees the outcome model is unbiased, no variable that can cause the outcome and the treatment exist, and hence Z-ignorability assumption holds. Constant variance of residuals can imply that the model is the most efficient, no variables that can explain variation of residuals exists. However, variance of residuals is not necessary to be a constant in order to obtain unbiased estimates of treatment effect, because non-constant variance of residuals imply there is heterogeneity on error term which is independent of all variables in the model (there is non-confounding variable that is predictive to outcome and hence can explain variation of residuals), adjusting for the variable can improve the precision of estimate. Hence, for outcome regression model, it is necessary to diagnose the model assumptions as to inspect possible existence of omitted variable bias. We diagnose model assumptions and model fit using following metrics: 1) residual mean (1.98 standard error) of subgroups; 2) distribution of overall residuals; 3) mean squared error of subgroups. In **RCTrep**, we can call `source.obj$summary(...)` to have an overview of these metrics. Figure 6 shows output of the function:

```
> source.obj$summary()
>
```

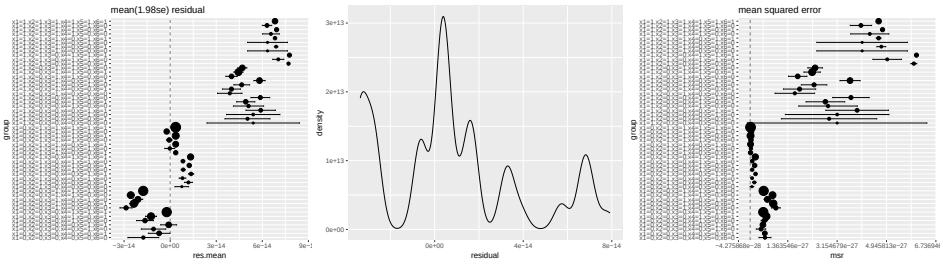


Figure 6: `source.obj` model assumption diagnosis and model fit evaluation: 1) mean of residuals of subgroups (default subgroups are levels of all variables in `treatment_confounders_name`), 2) distribution of overall residuals, and 3) mean square error of of subgroups

The figure 6 shows that means of residuals of subgroups are all very close to zero, although there are minor variations in means of residuals across subgroups probably due to sample sizes of subgroups. We also observe variation in the variance of residuals across subgroups, however, these variations are very minor and may be reduced as sample sizes of subgroups increase. The overall distribution of residual also centers around zero and spreads within acceptable dispersion. The mean squared error of each subgroup is close to zero although there are some variations across subgroups, which shows the same pattern as that of means of residual of subgroups. Overall, the summary in Figure 6 implies model assumptions should plausibly hold, and estimation of treatment effect in `source.obj` should be good.

Lastly, we initialize a class `Summary` as an object `fusion`, which compares both average treatment effect and heterogeneous treatment effect of subgroups defined by specified `stratification` and `stratification_joint` in `source.obj.rep$EstimateRep()`. We plot and print estimates from all objects, and evaluate methods for treatment effect estimation implemented in `source.obj` and `source.rep.obj` using five evaluation metrics, i.e., pseudo bias, pseudo mean squared error (mse), length of confidence interval, estimate agreement, and regulatory agreement Franklin *et al.* (2020):

```
> fusion <- Summary$new(target.obj,
                        source.obj,
                        source.obj.rep)
> fusion$evaluate()

# A tibble: 34 x 8
# Groups:   group_name [17]
   group_name          estimator              size    bias      mse len_ci agg.est agg.reg
   <chr>               <chr>                 <dbl>   <dbl>    <dbl>  <dbl> <lgl>   <lgl>
 1 pop                 G_computation+Exact+6  2618   0.073    0       4.52  TRUE    TRUE
 2 pop                 G_computation          2618 242.     588.      0.237 FALSE   TRUE
 3 x1=0,x3=0,x4=0,x5=0 G_computation+Exact+6    57  10.2      1.03  14.6   TRUE    TRUE
 4 x1=0,x3=0,x4=0,x5=0 G_computation            57 428.    1836.      1.11  FALSE   TRUE
 5 x1=0,x3=0,x4=0,x5=1 G_computation+Exact+6   104   1.32     0.017  9.02  TRUE    TRUE
 6 x1=0,x3=0,x4=0,x5=1 G_computation           104 408.    1662.      0.886 FALSE   TRUE
 7 x1=0,x3=0,x4=1,x5=0 G_computation+Exact+6   206  12.7      1.61   6.06  TRUE    TRUE
 8 x1=0,x3=0,x4=1,x5=0 G_computation           206 382.    1459.      0.666 FALSE   TRUE
 9 x1=0,x3=0,x4=1,x5=1 G_computation+Exact+6   367  14.0      1.97   6.20  TRUE    TRUE
10 x1=0,x3=0,x4=1,x5=1 G_computation           367 307.     941.      0.547 FALSE   TRUE
# ... with 24 more rows

> fusion$plot()
```

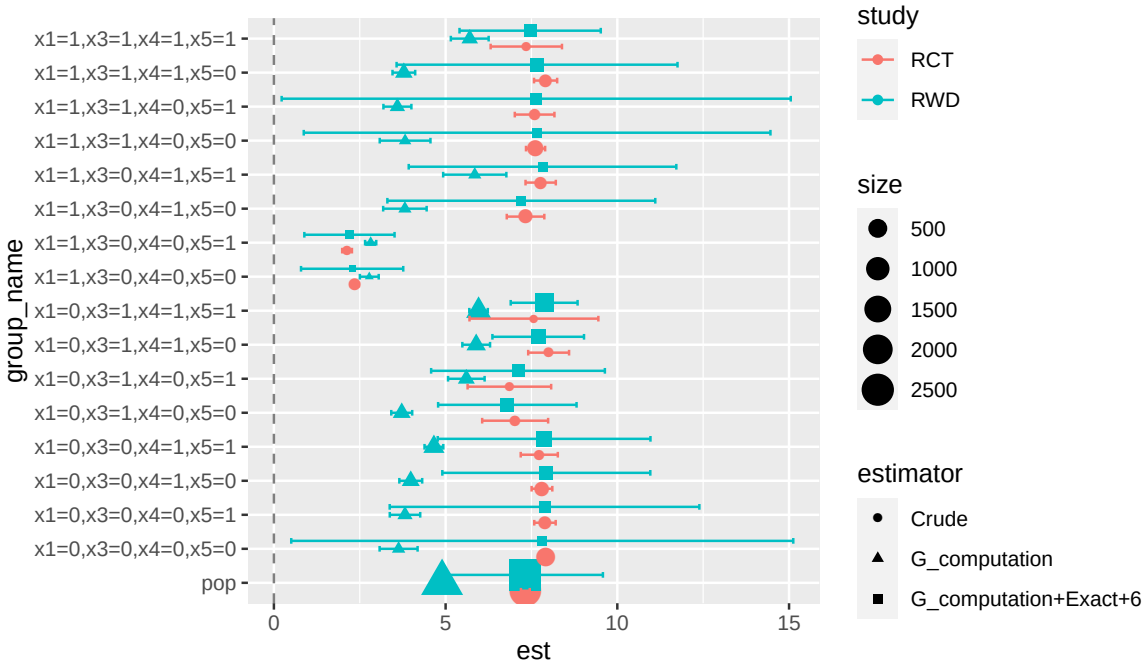


Figure 7: Evaluation of methods for treatment effect estimation (average treatment effect estimation is denoted by pop and average treatment effect of each subpopulation is denoted by group_name).

The result in Figure 7 shows that after correcting for the treatment assignment mechanism and the sampling mechanism, the estimates using RWD (indicated by `G_computation_Exact_6`) are very close to the estimates using experimental data (indicated by `Crude`), on both population and subpopulation levels. However, the estimates indicated by `G_computation` considerably differ from those indicated by `Crude`, implying that even though the treatment assignment mechanism of RWD can be properly corrected, there are still large differences in estimates between RWD and experimental data if the sampling mechanism of RWD is not properly corrected. The differences can be observed regardless of estimation methods tried out. Without considering the effect of the sampling mechanism on estimates comparison, people may easily attribute the spurious differences to unmeasured confounders of RWD, and hence question the validity of RWD to inform clinical decision making.

5. Basic usages

The current section offers an overview of **RCTrep**'s predefined subclasses of `TEstimator` each of which has an unique method for adjusting the treatment assignment mechanism, and subclasses of `SEstimator` each of which has an unique method for adjusting the sampling mechanism. Then we further demonstrate how to run and evaluate these methods.

5.1. Implemented methods

The package **RCTrep** provides three main methods for correcting the treatment assignment mechanism, hence provides three subclasses of parent class `TEstimator`; **RCTrep** provides

three methods for correcting the sampling mechanism, hence provides three subclasses of parent class `SEstimator`. See table 2 for details. Note that for each method, you can specify a model to estimate outcome, or propensity score, or sampling score. Note that we have a

`TEstiator`	`SEstimator`
Crude[1]	SEexact[6]
G_computation[2]	SEisw[7]
IPW[3]	SEsubclass[8]
DR[4]	SEexact_pp[5]
TEstimator_pp[5]	
Synthetic_TEstimator	

[1] difference in mean; [2] outcome regression;
[3] inverse propensity score weighting; [4] doubly robust estimator;
[5] pp means privacy-preserved;
[6] exact matching; [7] inverse sampling score weighting;
[8] subclassification

Table 2: Estimators in **RCTrep** for correcting for treatment assignment mechanism and sampling mechanism.

special subclass `Synthetic_TEstimator` to initialize an object using synthetic dataset from published trial meta-data. For instance, we can estimate the joint distribution of confounders by specifying pair-wise correlation given univariate distribution of each confounder [16]. See example 3 in the section 7 for utility of synthetic data for methods evaluation.

We also define a subclass `TEstimator_pp` and `SEexact_pp` to implement methods evaluation while preserving privacy. `TEstimator_pp` is a decorator of `TEstimator`. We assign an object of class `TEstimator` to initialize an object of class `TEstimator_pp`. The only difference between class `TEstimator` and `TEstimator_pp` is that the public field `data` of `TEstimator_pp` is the same as the public field `estimates$CATE` of `TEstimator`. `TEstimator_pp` has an unique implementation of `diagnosis_t_overlap`, `diagnosis_y_overlap`, and `est_ATE_SE()` and `est_weighted_ATE_SE()`. `SEexact_pp` has an unique implementation of public method `diagnosis_s_overlap`. We can only assign an object of class `TEstimator_pp` to the initialize function of the class `SEexact_pp`.

`SEsubclass` performs subclassification on the distance measure. Other than the default method (i.e., `glm`) to compute the distance measure (i.e., sampling score) for matching in `SEsubclass`, we can use other modeling approach to estimating the distance measure, for instance, `rpart`, by specifying the `distance` argument in the function `SEstimator_wrapper()`. RWD and experimental data are placed into subclasses based on quantiles of the sampling score of experimental data. Then weights for RWD are computed based on the proportion of experimental data in each subclass. See **MatchIt** for more subclass matching methods.

5.2. Running basic

In the current section we demonstrate how to implement methods evaluation using **RCTrep**. We break down the implementation into three steps: *Identification*, *Estimation*, and *Evaluation*. We regard the population that the experimental data represents as the target

[16] We provide a function `GenerateSyntheticData()` to generate synthetic dataset given marginal distribution

population, from which the experimental data is randomly sampled and RWD is sampled according to (unknown) sampling score. The estimate of treatment effect obtained from the experimental data is the unbiased estimates of "truth" which are used to evaluate $f(\boldsymbol{x})$ obtained from RWD. In the *Identification* step, we identify two covariates sets, 1) the covarate set `confounders_treatment_name`, which are confounders in RWD and need to adjust to correct the bias induced by non-randomization of treatment assignment mechanism; 2) the covariate set `confounders_sampling_name` which are variables that are imbalanced between RWD and experimental data due to the non-randomization of sampling mechanism of RWD, and needs to adjust in order to allow estimates from experimental data and RWD comparable [17]. In the *Estimation* step, we specify different estimation methods $f(\boldsymbol{x})$ for heterogeneous treatment effect. In *Evaluation* step, we compute evaluation metrics in equation 4 on population and subpopulation levels. We show the workflow in the figure 8. In the following, we first introduce the first step.

[17]By default, the two covariate sets are the same; under relaxed assumption of S-ignorability to allow for treatment effect exchangeable between RWD and experimental data, only effect modifiers which can cause the sampling mechanism need to adjust, and can avoid inflation of variance of the weighted estimate of treatment effect.

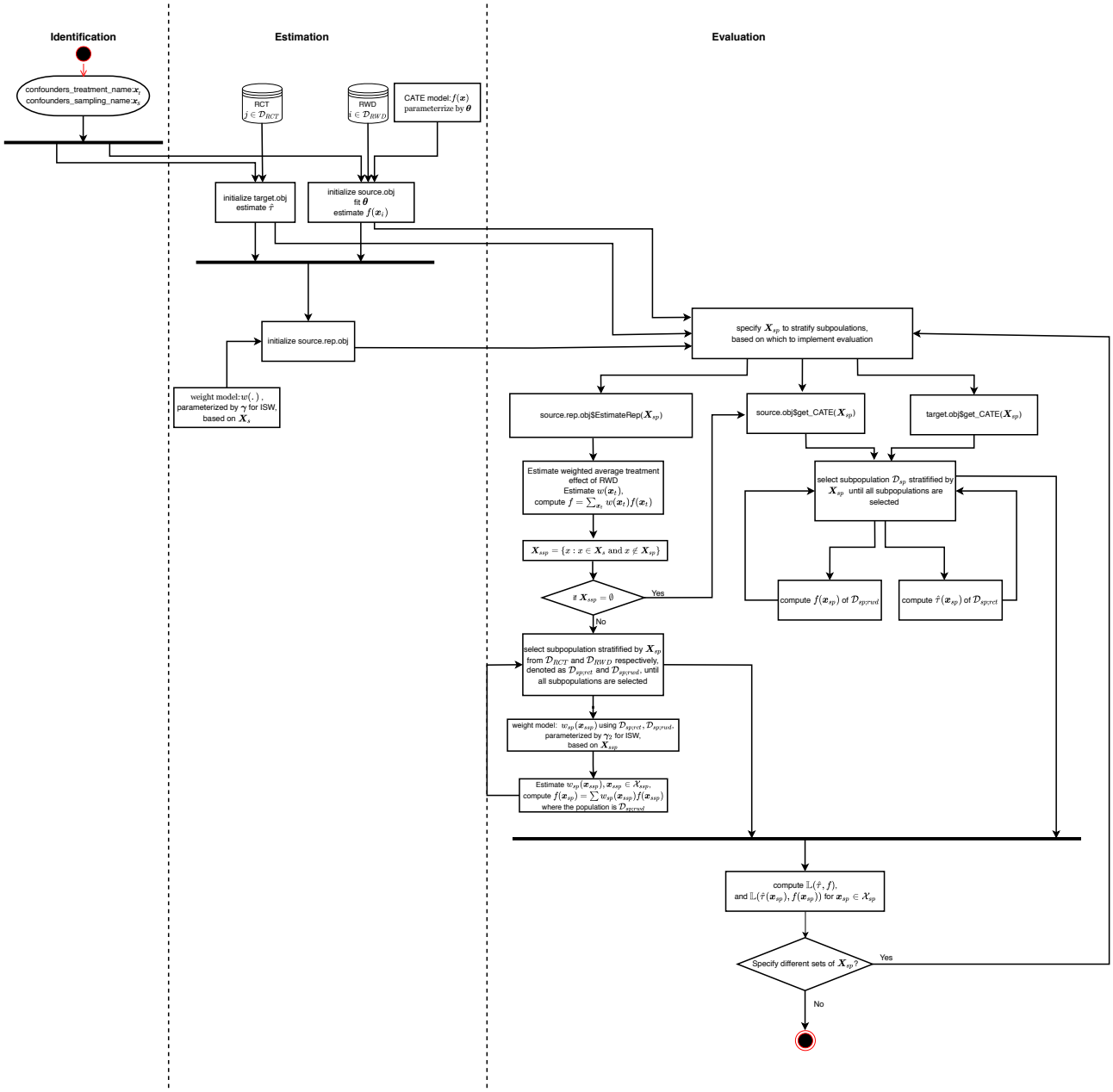


Figure 8: Set up of evaluation of methods $f(\boldsymbol{x}) \in \mathcal{F}$ for treatment effect estimation using RWD, in which the "truth" of average treatment effect of the target population (i.e., $\hat{\tau}$) and subpopulations (i.e., $\hat{\tau}(\boldsymbol{x})$ for subpopulations $i \in \mathcal{D}_{sp}$, $\boldsymbol{X}_i = \boldsymbol{x}$) are obtained from experimental data. The steps are detailed in the the sections 5.2.1, 5.2.2 and 5.2.3.

Identification

In this step, we should identify two sets of variables:

1) `confounders_treatment_name`: a set of variables to adjust the treatment assignment mechanism. `confouders_treatment_name` is passed to `TEstimator`.

2) `confounders_sampling_name`: a set of variables to adjust the sampling mechanism. `confounders_sampling_name` is passed to `SEstimator` class for to compute weight accordingly. By default, `confounders_treatment_name` and the `confounders_sampling_name` are the same. To avoid variance inflation, we can only assign effect modifiers that are associated with the sampling to `confounders_sampling_name`.

Identification of the two sets of variables needs expertise knowledge, previous evidence, preliminary analysis of feature analysis (which vairables are predictive to outcomes and predictive to treatment), as well as exploratory analysis of two datasets (which variables that are strongly predictive to outcomes are distributed differently). We can use relevant packages to identify sufficient, optimal, or minimal set of variables that allow Z-ignorability and S-ignorability to hold. The packages are but not limited to, e.g., R packages **dosearch** (Tikka, Hyttinen, and Karvanen 2019) and **causaleffect**(Tikka and Karvanen 2017), web-based software **causalfusion**[18] (Bareinboim and Pearl 2016). In figure 9, we demonstrate the two adjustment sets of simulated RWD used in examples throughout the paper via a structural causal diagram. In the figure, we present confounders, additional variables predictive to outcomes, and variables that are predictive to sample selection. In practice, although the true structural causal graph of a dataset is unknown and a causal diagram graph can only hold with assumptions, the graph can help us a) understand sources of bias that lead to difference in estimates between RWD and experimental data, and b) identify `confounders_treatment_name` and `confounders_sampling_name` easily.

[18] https://www.causalfusion.net/login

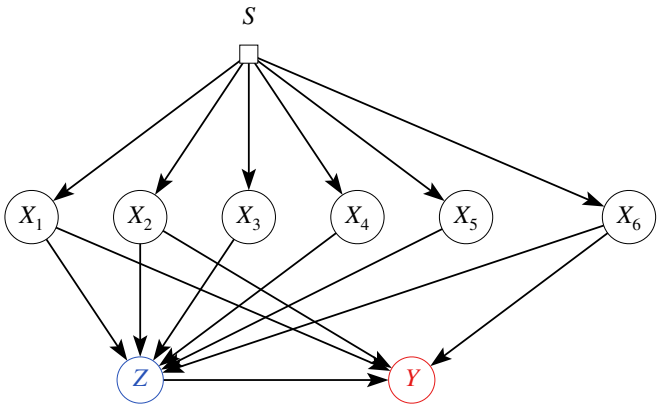


Figure 9: Structural causal diagram representing treatment Z, outcome Y, sample selection with S and other predictors of outcome. The diagram visualize the data generation mechanism of simulated data used in working examples in section 7. The figure is generated using the software **causalfusion**. Since `x3,x4,x5` are not predictive to outcome, and X_2 and X_6 are effect modifiers, according to back-door criteria, the minimal `confounders_treatment_name` and `confounders_sampling_name` that allow Z-ignorability and S-ignorability hold are `x1,x2,x6` and `x2,x6`. Adjusting X_3, X_4, X_5 can inflate the variance of estimates of average treatment effect and adjusting X_1, X_3, X_4, X_5 can inflate the variance of the weighted estimates.

.

Estimation

After identifying `confounders_treatment_name` to adjust in `TEstimator` and `confounders_sampling_name` to adjust in `SEstimator`, we can estimate treatment effects using methods implemented in `TEstimator` and estimate weighted treatment effects using methods implemented in `SEstimator`. In the following, we implement all combinations of classes of methods in `TEstimator` and `SEstimator` to compute weighted treatment effects of RWD:

```
source.obj.gc <- TEstimator_wrapper(
  Estimator = "G_computation",
  data = source.data,
  name = "RWD",
  vars_name = vars_name,
  outcome_method = "glm",
  outcome_formula = y ~ x1 + x2 + x3 + z + z:x1 + z:x2 +z:x3+ z:x6,
  data.public = TRUE
)

source.obj.ipw <- TEstimator_wrapper(
  Estimator = "IPW",
  data = source.data,
  name = "RWD",
  vars_name = vars_name,
  treatment_method = "glm",
  treatment_formula = z ~ x1 + x2 + x3 + x4 + x5 + x6 + x1:x2 + x3:x4,
  data.public = TRUE
)

source.obj.dr <- TEstimator_wrapper(
  Estimator = "DR",
  data = source.data,
  name = "RWD",
  vars_name = vars_name,
  outcome_method = "glm",
  outcome_formula = y ~ x1 + x2 + x3 + z + z:x1 + z:x2 +z:x3+ z:x6,
  treatment_method = "glm",
  treatment_formula = z ~ x1 + x2 + x3 + x4 + x5 + x6 + x1:x2 + x3:x4,
  data.public = TRUE
)

target.obj <- TEstimator_wrapper(
  Estimator = "Crude",
  data = target.data,
  name = "RCT",
  vars_name = vars_name,
  data.public = TRUE
)
```

```
strata <- c("x1","x4")
source.gc.exact <- SEstimator_wrapper(estimator="Exact",target.obj=target.obj, source.obj=
source.gc.exact$EstimateRep(stratification = strata, stratification_joint = TRUE)

source.gc.isw <- SEstimator_wrapper(estimator="ISW",target.obj=target.obj, source.obj=sour
source.gc.isw$EstimateRep(stratification = strata, stratification_joint = TRUE)

source.gc.subclass <- SEstimator_wrapper(estimator="Subclass",target.obj=target.obj, sourc
source.gc.subclass$EstimateRep(stratification = strata, stratification_joint = TRUE)

source.ipw.exact <- SEstimator_wrapper(estimator="Exact",target.obj=target.obj, source.obj
source.ipw.exact$EstimateRep(stratification = strata, stratification_joint = TRUE)

source.ipw.isw <- SEstimator_wrapper(estimator="ISW",target.obj=target.obj, source.obj=sou
source.ipw.isw$EstimateRep(stratification = strata, stratification_joint = TRUE)

source.ipw.subclass <- SEstimator_wrapper(estimator="Subclass",target.obj=target.obj, sour
source.ipw.subclass$EstimateRep(stratification = strata, stratification_joint = TRUE)

source.dr.exact <- SEstimator_wrapper(estimator="Exact",target.obj=target.obj, source.obj=
source.dr.exact$EstimateRep(stratification = strata, stratification_joint = TRUE)

source.dr.isw <- SEstimator_wrapper(estimator="ISW",target.obj=target.obj, source.obj=sour
source.dr.isw$EstimateRep(stratification = strata, stratification_joint = TRUE)

source.dr.subclass <- SEstimator_wrapper(estimator="Subclass",target.obj=target.obj, sourc
source.dr.subclass$EstimateRep(stratification = strata, stratification_joint = TRUE)

fusion <- Summary$new(target.obj,
                      source.gc.exact,
                      source.gc.isw,
                      source.gc.subclass,
                      source.ipw.exact,
                      source.ipw.isw,
                      source.obj.dr,
                      source.dr.exact,
                      source.dr.isw,
                      source.dr.subclass)
```

```
> fusion$plot()
```

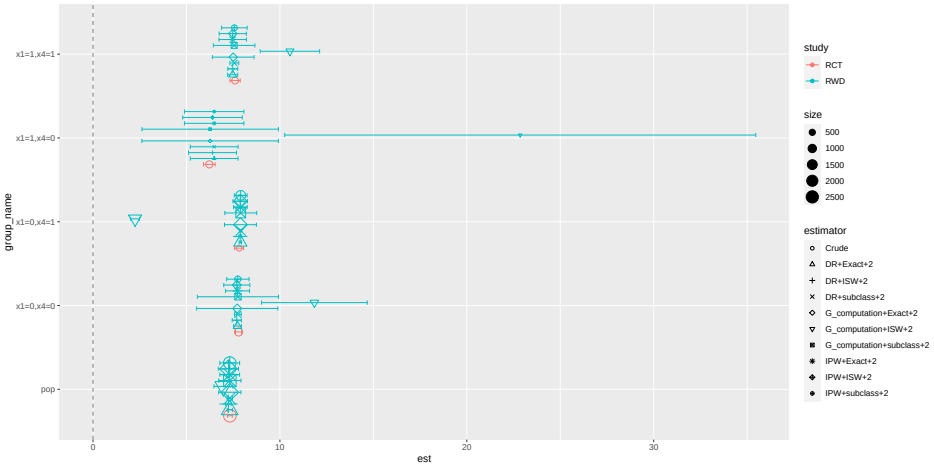


Figure 10: Comparisons of 9 (3*3) estimators combined

```
> fusion$print()
```

```
# A tibble: 50 x 9
# Groups:   group_name [5]
   group_name study estimator                 est  ci_l  ci_u  size y1.hat y0.hat
   <chr>      <chr> <chr>                   <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
   pop        RCT   Crude                    7.32  7.19  7.45  2686   7.03 -0.294
   pop        RWD   G_computation+Exact+2    7.32  6.72  7.92  2618   7.89  0.565
   pop        RWD   G_computation+ISW+2      6.97  6.47  7.48  2618   7.52  0.542
   pop        RWD   G_computation+subclass+2 7.32  6.72  7.92  2618   7.89  0.565
   pop        RWD   IPW+Exact+2              7.32  6.79  7.84  2618   7.88  0.566
   pop        RWD   IPW+ISW+2                7.25  6.70  7.79  2618   7.81  0.563
   pop        RWD   IPW+subclass+2           7.32  6.79  7.84  2618   7.88  0.566
   pop        RWD   DR+Exact+2               7.32  7.17  7.46  2618   7.88  0.566
   pop        RWD   DR+ISW+2                 7.25  7.10  7.40  2618   7.81  0.563
   pop        RWD   DR+subclass+2            7.32  7.17  7.46  2618   7.88  0.566
# ... with 40 more rows
```

Evaluation

Then we can evaluate all implemented methods and choose the most accurate one according to equation 4. Beyond evaluation metric in the equation 4 (i.e., pseudo MSE $\times 100$), length of confidence interval, estimate agreement, and regulatory agreement Franklin *et al.* (2020). We can evaluate performance of methods by calling:

```
> fusion$evaluate()

# A tibble: 45 x 7
# Groups:   group_name [5]
   group_name estimator                size    mse len_ci agg.est agg.reg
   <chr>      <chr>                   <dbl>  <dbl>  <dbl> <lgl>   <lgl>
   pop        G_computation+Exact+2    2618  0      1.19  TRUE    TRUE
   pop        G_computation+subclass+2 2618  0      1.19  TRUE    TRUE
   pop        IPW+Exact+2              2618  0.005  1.06  TRUE    TRUE
   pop        IPW+subclass+2           2618  0.005  1.06  TRUE    TRUE
   pop        DR+Exact+2               2618  0.005  0.283 TRUE    TRUE
   pop        DR+subclass+2            2618  0.005  0.283 TRUE    TRUE
   pop        IPW+ISW+2                2618  0.56   1.09  TRUE    TRUE
   pop        DR+ISW+2                 2618  0.56   0.298 TRUE    TRUE
   pop        G_computation+ISW+2      2618 12.1    1.01  FALSE   TRUE
   x1=0,x4=0  G_computation+subclass+2  527  0.101  4.34  TRUE    TRUE
# ... with 35 more rows
```

The results show that using G-computation and Exact weighting is the best combination in terms of pseudo mean squared error, which is inline with the existing literature in Chatton *et al.* (2020); Le Borgne *et al.* (2021); Loiseau *et al.* (2022). Different weighting approaches may have different impacts on the precision of estimates of treatment effect.

6. Core classes

The current section offers additional background information on **RCTrep**'s classes structures - both on R6 class system Chang and on each of the three previously introduced core **RCTrep** classes. Together with the information in the next section, on `TEstimator` and `SEstimator` implementation, this should be able to get you up and running with developing your own custom `TEstimator` and `SEstimator` subclasses.

6.1. Choice for the R6 class system

Though widely used as a procedural language, R offers several Object Oriented (OO) systems, which can significantly help in structuring the development of more complex packages. Out of the OO systems available (S3, S4, R5, and R6), we settled on R6, as it offers several advantages over other options. Firstly, it implements a mature object-oriented design compared to S3 and S4, hence is easier for developers with a background in programming languages such as C++ and Java to maintain. Secondly, when compared to the older R5 reference class systems, R6 classes are much lighter-weight, as they do not use S4 classes, do not require the **methods** package.

6.2. Core classes

In this section, we go over the three core classes on more detail - with an emphasis on the `TEstimator` and `SEstimator` classes. We provide an overview of the two classes in two

diagrams in appendix **??**. We illustrate the structure of classes, and enumerate core public functions of each classes.

TEstimator

The `TEstimator` class is responsible for fitting a model and estimating treatment effect. The following skeleton code gives an overview of how the above is implemented in **RCTrep**'s `TEstimator` class:

```
TEstimator <- R6::R6Class(
  "TEstimator",
  #------------------------public fields-----------------------------#
  public = list(
    id = NA,
    name = character(),
    statistics = list(n=numeric(),
                      density_confounders=data.frame()),
    data = NULL,
    estimates = list(ATE=data.frame(y1.hat=NA,
                                    y0.hat=NA,
                                    est=NA,
                                    se=NA),
                     CATE = data.frame()),
    model = list(),
    #------------------------constructor----------------------------
    initialize = function(df, vars_name, name) {
      self$name <- name
      self$data <- df
      self$data$id <- seq(dim(df)[1])
      private$confounders_treatment_name <- vars_name$confounders_treatment_name
      private$treatment_name <- vars_name$treatment_name
      private$outcome_name <- vars_name$outcome_name
      self$statistics <- list(n=dim(df)[1],
                              density_confounders=private$est_joint_denstiy())
    },
    get_CATE = function(stratification, stratification_joint=TRUE) {},
    plot_CATE = function(stratification, stratification_joint = TRUE) {},
    diagnosis_t_overlap = function(stratification, stratification_joint=TRUE){},
    diagnosis_y_overlap = function(stratification, stratification_joint=TRUE){},
    summary = function(){}

  ),
  #------------------------private fields and methods---------------------------#
  private = list(
    confounders_treatment_name = NA,
    treatment_name = NA,
    outcome_name = NA,
    var_method = "sandwitch",
```

```
    isTrial = FALSE,

    set_ATE = function(){},
    set_CATE = function(stratification, stratification_joint){},
    est_joint_denstiy = function(){},
    est_CATEestimation4JointStratification = function(stratification) {},
    est_CATEestimation4SeperateStratification = function(stratification) {},
    fit = function(){},
    est_ATE_SE = function(){},
    est_weighted_ATE_SE = function(){}
  )
)
```

Subclasses of `TEstimator` have their unique implementation of `summary()`, `fit()`, `est_ATE_SE()`, and `est_weighted_ATE_SE()`, and their unique private methods. The main `TEstimator` functions are:

1. `get_CATE(stratification, stratification_joint=TRUE)`

   (a) `stratification`: a character vector of length k specifies variables to select subgroups.

   (b) `stratification_joint`: logical to indicate if subgroups are selected based on levels of each variable in `stratification` or joint levels of all k variables in `stratification`.

   The function returns a `data.frame` containing treatment effects estimation of selected subgroups. If `stratification=TRUE`, then the function returns a `data.frame` with column names `c(stratification,"y1.hat","y0.hat","cate","se","size")`; if `stratification_join` then the function returns a `data.frame` with column names `c("name", "value","y1.hat","y0.hat",`

2. `diagnosis_t_overlap(stratification, stratification_joint)`: plot the percentage and number of individuals receiving treatment and control in each subgroup. Subgroups are defined by `stratification` and `stratification_joint`.

3. `diagnosis_y_overlap(stratification, stratification_joint)`: plot the distribution of outcomes in treatment and control groups in each subgroup defined by `stratification` and `stratification_joint`. For binary outcomes, the function plots the count of positive outcome and negative outcome; for continuous outcomes, the function plots the distribution of outcomes.

4. `summary()`: the function summarizes fit of a model implemented in a subclass of `TEstimator`. For instance, for subclass `G_computation`, the function summarizes outcome model fit using evaluation metrics, i.e., means of residuals of subgroups, distribution of overall residuals, mean squared errors of subgroups for continuous outcome, and mean of deviance of subgroups for binary outcome. For subclass `IPW`, the function summarizes the distribution of propensity score for treatment and control groups in each subgroup. For `DR`, the function summarizes both outcome model fit and propensity score distribution.

5. private method `set_ATE()`: the function implements the private method `est_ATE_SE(id)`, and gets the point estimate of average treatment effect, standard error of the estimate, mean of potential outcomes; the function assigns these estimates to the public fields `estimates$ATE$est`,`estimates$ATE$se`,`estimates$ATE$y1.hat`,`estimates$ATE$y0.hat` accordingly. The function is implemented in the initialize function of each `TEstimator` subclass.

6. private method `set_CATE(stratification, stratification_joint)`: the function implements the public method `get_CATE(stratification, stratification_joint)` which returns a `data.frame` (see below for details of returned object from the function `get_CATE()`); then the function `set_CATE()` assigns the returned estimates from `get_CATE()` to the public field `estimates$CATE`. The function is implemented in the initialize function of each subclass of `TEstimator` by calling `private$set_CATE(private$confounders_tr`

7. private method `est_CATEestimation4JointStratification(stratification)`: the function selects subgroups defined by joint levels of all variables specified in `stratification`, gets the index of selected data, and estimates the average treatment effect of each subgroup by calling the private method `est_ATE_SE(index)`. The function returns a `data.frame` with column name `c(stratification, "y1.hat", "y0.hat", "cate", "se", "size")`.

8. private method `est_CATEestimation4SeperateStratification(stratification)`: the function selects subgroups defined by levels of each variable specified in `stratification`, gets the index of selected data, and estimates the average treatment effect of each subgroup by calling the private method `est_ATE_SE(index)`. The function returns a `data.frame` with column name `c("name", "value", "y1.hat", "y0.hat", "cate", "se", "size")`.

9. private method `est_ATE_SE(index)`: the function estimates average treatment effect and its standard error. `index` indicates index of data. Different subclass has unique implementation of point estimation. We implement sandwich estimator to estimate standard error using package **geex** (Saul and Hudgens 2020). We need to specify an estimation function `estFUN`, and pass the function to `geex::m_estimate(data, estFUN, ...)`. `m_estimate` provides a consistent estimator for the asymptotic variance of estimate of average treatment effect. In **RCTrep**, we do not take uncertainty of estimation of parameters of models into account in order to speed up implementation, however, we can customize `estFUN` so the function can take account of uncertainty of estimation of parameters into estimation of variance of average treatment effect. For more details, see simulation codes in (Dahabreh *et al.* 2020) and tutorials by Saul and Hudgens (2020). `est_ATE_SE(index)` function returns a `list` with named elements `y1.hat`, `y0.hat`, `est`, and `se`. We provide an overview of estimators of variance of average treatment effect in appendix B.

10. private method `est_weighted_ATE_SE(index, weight)`: the function estimates weighted average treatment effect and its standard error. The function selects estimates of potential outcomes from `self$data[index,]$y1.hat` and `self$data[index,]$y0.hat`, and assigns weights for selected data. We implement sandwich estimator using **geex** to estimate standard error of weighted average treatment effect regarding weights as

constant values. The function returns a `list` with named elements `y1.hat`, `y0.hat`, `est`, and `se`.

11. private method `est_CATEestimation4JointStratification(stratification)`: the function estimates treatment effect of subgroups. The function selects a subgroup based on joint levels of variables in `stratification`, gets `id` of the selected subgroup, and computes the average treatment effect of the subgroup by calling `private_ATE_SE(id)`. Loop this procedure until all subgroups have been selected. The function returns a `data.frame` with column names `c(stratification, "y1.hat", "y0.hat", "cate", "se", "size")`.

12. private method `est_CATEestimation4SeperateStratification(stratification)`: the function estimates treatment effect of subgroups. The function selects a subgroup based on levels of each variables in `stratification`, gets `id` of the selected subgroup, and computes the average treatment effect of the subgroup by calling `private$est_ATE_SE(id)`. Loop this procedure until all subgroups have been selected. The function returns a `data.frame` with column names `c("name", "value", "y1.hat", "y0.hat", "cate", "se", "size")`.

SEstimator

The `SEstimator` class is responsible for balancing covariates in `confounders_sampling_name` between two objects of class `TEstimator`, and estimates the weighted average and heterogeneous treatment effect. The following skeleton code gives an overview of how weighted estimation is implemented in `RCTrep`'s `SEstimator` classes:

```
SEstimator <- R6::R6Class(
  "SEstimator",
  #-------------------------public fields-----------------------------#
  public = list(
    name = character(),
    id = character(),

    statistics = list(),
    estimates = list(ATE = data.frame(y1.hat=NA,
                                      y0.hat=NA,
                                      est=NA,
                                      se=NA),
                     CATE = data.frame()),
    model = NA,
    confounders_sampling_name = NA,
    weighting_method = character(),

    initialize = function(target.obj, source.obj, weighting_method=NULL,
                          confounders_sampling_name){
      private$target.obj <- target.obj
      private$source.obj <- source.obj
      self$weighting_method <- weighting_method
```

```
      self$confounders_sampling_name <- confounders_sampling_name
      private$ispublic <- !c("TEstimator_pp") %in% class(source.obj)
      self$name <- source.obj$name
      self$statistics <- source.obj$statistics
      self$id <- paste(private$source.obj$id, self$weighting_estimator, sep = '+')
    },

    EstimateRep = function(stratification=self$confounders_sampling_name,
                           stratification_joint=TRUE) {},
    diagnosis_s_overlap = function(stratification=NULL,
                                   stratification_joint=TRUE){}
  ),

  private = list(
    source.obj = NA,
    target.obj = NA,
    ispublic = NA,

    get_weight = function(){},
    set_weighted_ATE_SE = function() {},
    set_weighted_CATE_SE = function(stratification, stratification_joint) {},
    est_WeightedCATEestimation4JointStratification = function(stratification) {},
    est_WeightedCATEestimation4SeperateStratification = function(stratification) {}
  )
)
```

The following are public and private functions in `SEstimator`:

1. public function `EstimateRep(stratification, stratification_joint)`: the core function which estimates weighted average treatment effect and weighted heterogeneous treatment effect; `stratification` and `stratification_joint` specify a criteria to select subgroups, which is the same as `get_CATE()` in `TEstimator`.

2. `diagnosis_s_overlap(stratification, stratification_joint)`: the function selects subgroups according to `stratification, stratification_joint`; the function plots the percentage and numbers of observations from `source.obj` and `target.obj` for each subgroup.

3. private method `get_weight(source.data, target.data, vars_weighting)`: the function estimates weights for each individual in `source.obj`. The weights are computed based on specified variables `vars_weighting`. Each subclass of `SEstimator` has an unique implementation of the function.

4. private method `set_weighted_ATE_SE`: the function estimate the weighted average treatment effect of `source.obj`. The function calls `private$get_weight(source.data=private$sourc targe.data=private$target.obj$data, vars_weighting=self$confounders_sampling_name)` to compute weights, then calls the private method `est_weighted_ATE_SE()` of `source.obj` to estimate weighted average treatment effect and gets the weighted estimates of `y1.hat`,

y0.hat, est, and se accordingly, and finally assigns these estimates to self$estimates$ATE$y1.hat, self$estimatesATEy0.hat, self$estimates$ATE$est, self$estimatesATEse.

5. private method set_weighted_CATE_SE(stratification, stratification_joint): the function estimates weighted heterogeneous treatment effect; if stratification_joint=TRUE, then the function calls private$est_WeightedCATEestimation4JointStratification(stratificati if stratification_joint=FALSE, then the function calls private$est_WeightedCATEestimation4Sep Stratification is a character vector specifies variables for subgroup selection.

6. private method est_WeightedCATEestimation4JointStratification(stratification): the function estimates weighted heterogeneous treatment effect. The function selects subgroups from private$source.obj$data and private$target.obj$data, and calls private$get_weight() to compute weights of each individuals in source.obj so that variables in self$confounders_sampling_name are balanced between weighted source.obj and target.obj [19]. The function returns a data.frame in the same form as that returned from the private method est_CATEestimation4JointStratification(stratification) of the class TEstimator.

7. private method est_WeightedCATEestimation4SeperateStratification(stratification): the same as the est_WeightedCATEestimation4JointStratification(stratification) except for the criteria to select subgroups. The function returns a data.frame in the same form as that returned from the private method est_CATEestimation4SeperateStratification(s of the class TEstimator.

Summary

The Summary class is responsible for aggregating estimates from objects of classes TEstimator and SEstimator, evaluating methods for treatment effect estimation implemented in class TEstimator, plotting and printing results. The following skeleton code gives an overview of class Summary:

```
Summary <- R6::R6Class(
  "Summary",
  #------------------------public fields-----------------------------#
  public = list(
    objs.cate.data = data.frame(),
    objs.ate.data = data.frame(),
    stratification = NA,
    stratification_joint = NA,
    RCT.study.name = NA,
    RWD.study.name = NA,

    initialize = function(...){},
    plot = function(){},
```

[19] In case there are variables specified both in self$confounders_sampling_name and stratification, then we only need to select variables in self$confounders_sampling_name which are not in stratification, assign the selected variables to vars_weighting, and assign the vars_weighting to the function private$get_weight(source.data, target.data, vars_weighting) to compute weights.

```
    print = function(){},
    evaluate = function(){}
  ),

  private = list(
    aggregate_cate_estimates = function(...){},
    aggregate_ate_estimates = function(...){}
  )
)
```

The following are public and private methods in `Summary`:

1. constructor `initialize(...)` initializes an object of `Summary`; passes objects of class `TEstimator` and `SEstimator` to the argument `....` The number of objects passed to the function is not limited.

2. public function `plot()`, `print()` plots and prints average and heterogeneous treatment effect estimation using RWD.

3. public function `evaluate()`: the function computes pseudo mse, length of confidence interval, estimate agreement, regulatory agreement (Franklin *et al.* 2020). The function computes the evaluation metrics on population and sub-population levels. Sub-populations are selected according to `self$stratification` and `self$stratification_joint`, values of which are inherent from arguments passed to the function `EstimateRep()` of an object of class `SEstimator` that is passed to the initialize function of the class `Summary`.

4. private method `aggregate_ate_estimates` and private method `aggregate_cate_estimates`: the functions aggregate estimates of average treatment effect and heterogeneous treatment effect from all objects passed to `....`

6.3. Subclasses of TEstimator and SEstimator

Subclasses of `TEstimator` are mainly responsible for fitting models and estimating treatment effect using their unique methods `est_ATE_SE`. We can override `est_ATE_SE` for a new subclass of `TEstimator`. Subclasses of `SEstimator` are responsible for estimating weights $w(\boldsymbol{x})$ using their unique methods `get_weight`. We can override the function for a new subclass of `SEstimator`.

Since the aim of data sharing is to compute weights to balance covariates in two objects, hence it is not necessary to have full datasets. For instance, each object only needs to share density of $\boldsymbol{X}$, estimates $f(\boldsymbol{X})$, standard error of $f(\boldsymbol{X})$, and sample size for each subgroup stratified by $\boldsymbol{X}$, to estimate weighted treatment effect. Hence, in case that individual level data is not allowed to share, we define a subclass `TEstimator_pp` for `TEstimator` and `SEstimator_pp` for `SEstimator`. In `TEstimator_pp`, instead of assigning a full dataset to the public field `data`, we assign density of covariates in `confounders_treatment_name` and the estimates of treatment effect of subgroups stratified by `confounders_treatment_name` to the public field `data` of an object of class `TEstimator_pp`. Two objects are passed to an object of class `SEstimator`, and communicate `data` with each other within the object. The object computes weights for `source.obj` accordingly. For different weighting approach, we can share different aggregated

data. For instance, weighting using balanced-based methods only requires $p(B(\boldsymbol{x}))$ (Chatton *et al.* 2020), where $B(\boldsymbol{x})$ is the basis function of $\boldsymbol{x}$, e.g., interaction between two random variables. Hence, in this case, we only need to share the density of basis function $B(\boldsymbol{x})$, and override `data` in a new subclass of `TEstimator`, and override `get_weight` in a new subclass of `SEstimator` accordingly.

7. Running working examples

In this section, we demonstrate four examples using simulated data which is included in **RCTrep** package. The first example demonstrates the evaluation approach in case individual-level data are allowed to share. The second example demonstrates the evaluation approach in case only aggregated data of two data sets, i.e., $p(\boldsymbol{x})$, are allowed to share. The third example demonstrates the evaluation approach in case only marginal estimates of the "truth" of treatment effect using experimental data are available. The fourth example examines the effect of different adjustment sets on precision of estimation.

7.1. Example 1: Two datasets are allowed to share

If `source.data` and `target.data` are individual level data, we can call the wrapper function `RCTREP()` that implements evaluation directly. The function returns two objects of class `TEstimator`, namely, `source.obj` for RWD and `target.obj` for experimental data. The following codes implement methods evaluation:

```
strata <- c("x1","x4")
confounders_sampling_name <- c("x2","x6")
output <- RCTREP(TEstimator="G_computation", SEstimator = "Exact",
                 outcome_method = "glm",
                 source.name = "RWD", target.name = "RCT",
                 outcome_form=y ~ x1 + x2 + x3 + z + z:x1 + z:x2 +z:x3+ z:x6,
                 source.data=source.data, target.data=target.data,
                 vars_name=vars_name,
                 confounders_sampling_name = confounders_sampling_name,
                 stratification = strata, stratification_joint = TRUE)

fusion <- Summary$new(output$target.obj,
                      output$source.obj,
                      output$source.rep.obj)

> fusion$plot()
```

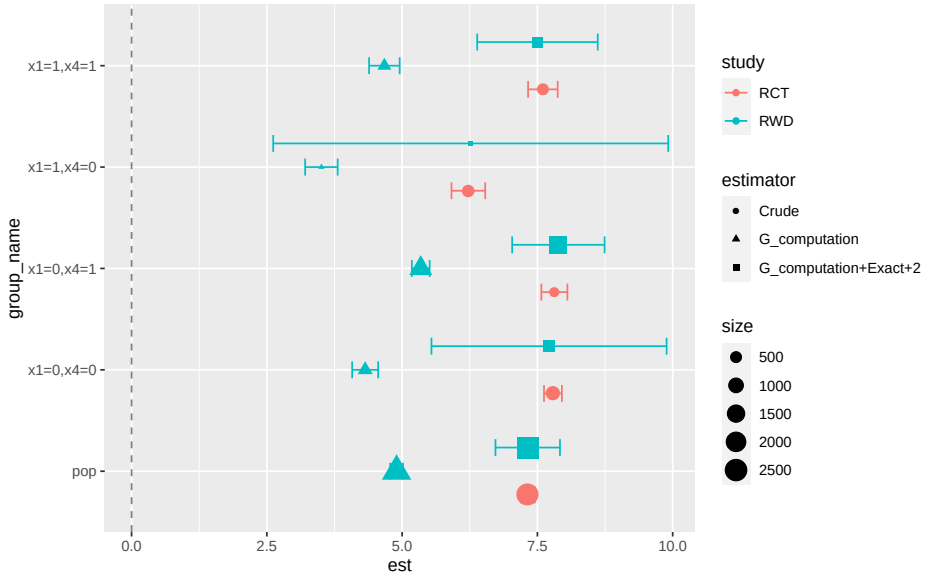


Figure 11: Caption

```
> fusion$print()
# A tibble: 15 x 9
# Groups:   group_name [5]
   group_name study estimator               est  ci_l  ci_u  size y1.hat y0.hat
   <chr>      <chr> <chr>                 <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
 1 pop        RCT   Crude                  7.32  7.19  7.45  2686   7.03 -0.294
 2 pop        RWD   G_computation          4.90  4.78  5.02  2618   5.50  0.599
 3 pop        RWD   G_computation+Exact+6  7.32  5.06  9.58  2618   8.17  0.848
 4 x1=0,x4=0  RCT   Crude                  7.79  7.62  7.95   947   7.00 -0.791
 5 x1=0,x4=1  RCT   Crude                  7.81  7.57  8.05   428   7.02 -0.791
 6 x1=1,x4=0  RCT   Crude                  6.22  5.91  6.54   691   6.43  0.209
 7 x1=1,x4=1  RCT   Crude                  7.60  7.33  7.88   620   7.81  0.209
 8 x1=0,x4=0  RWD   G_computation          4.32  4.08  4.56   527   4.68  0.36
 9 x1=0,x4=1  RWD   G_computation          5.34  5.18  5.51  1467   5.70  0.36
10 x1=1,x4=0  RWD   G_computation          3.51  3.21  3.81   179   4.87  1.36
11 x1=1,x4=1  RWD   G_computation          4.67  4.39  4.95   445   6.03  1.36
12 x1=0,x4=0  RWD   G_computation+Exact+6  7.72  2.48 12.9    527   8.07  0.36
13 x1=0,x4=1  RWD   G_computation+Exact+6  7.89  5.74 10.0   1467   8.25  0.36
14 x1=1,x4=0  RWD   G_computation+Exact+6  6.27  1.93 10.6    179   7.63  1.36
15 x1=1,x4=1  RWD   G_computation+Exact+6  7.50  5.19  9.81   445   8.86  1.36

> fusion$evaluate()
# A tibble: 10 x 8
# Groups:   group_name [5]
   group_name estimator              size    bias      mse len_ci agg.est agg.reg
   <chr>      <chr>                 <dbl>   <dbl>    <dbl>  <dbl> <lgl>   <lgl>
 1 pop        G_computation+Exact+6  2618   0.073    0       4.52  TRUE    TRUE
 2 pop        G_computation          2618 242.     588.      0.237 FALSE   TRUE
 3 x1=0,x4=0  G_computation+Exact+6   527   7.22     0.521 10.5   TRUE    TRUE
 4 x1=0,x4=0  G_computation           527 347.    1204.      0.481 FALSE   TRUE
 5 x1=0,x4=1  G_computation+Exact+6  1467   7.39     0.546  4.29  TRUE    TRUE
 6 x1=0,x4=1  G_computation          1467 247.     610.      0.329 FALSE   TRUE
 7 x1=1,x4=0  G_computation+Exact+6   179   4.39     0.193  8.67  TRUE    TRUE
 8 x1=1,x4=0  G_computation           179 272.     737.      0.601 FALSE   TRUE
 9 x1=1,x4=1  G_computation+Exact+6   445   9.98     0.996  4.62  TRUE    TRUE
10 x1=1,x4=1  G_computation           445 293.     858.      0.565 FALSE   TRUE
```

7.2. Example 2: Two data sets are not allowed to share

RCTrep provides a solution to methods evaluation using aggregated data while preserving privacy. In example 1, we assign full RWD and experimental data to `RCTREP()` to implement methods evaluation, in the example 2, we demonstrate a procedure for methods evaluation without sharing full datasets. We start out by instainating an object `source.obj` using RWD and an object `target.obj` using experimental data [20]:

[20]note that in the example 2, we have preprocessed two datasets: 1. we have filtered `source.data` and `target.data` so two data sets have overlap in covariates space defined by `confounders_sampling_name`; any co-

```
library(geex)
library(caret)
confounders_sampling_name <- c("x2","x6")

source.obj <- TEstimator_wrapper(
  Estimator = "G_computation",
  data = source.data,
  vars_name = vars_name,
  outcome_method = "glm",
  outcome_form=y ~ x1 + x2 + x3 + z + z:x1 + z:x2 +z:x3+ z:x6,
  name = "RWD",
  data.public = FALSE
)

target.obj <- TEstimator_wrapper(
  Estimator = "Crude",
  data = target.data,
  vars_name = vars_name,
  name = "RCT",
  data.public = FALSE,
  isTrial = TRUE
)
```

We specify `data.public=FALSE` to indicate that full dataset is not allowed to share. The default value of `data.public` is `TRUE`. `TEstimator_wrapper()` returns an object of class `TEstimator_pp` of which public field `data` is aggregated data on joint levels of all variables in `confounders_treatment_name`.

```
> class(source.obj)
[1] "TEstimator_pp" "TEstimator"    "R6"
> head(source.obj$data)
   x1 x2 x3 x4 x5 x6    y1.hat   y0.hat     cate           se size id
1   0  0  0  0  0  0  3.607016 1.607016 2.000000 3.598499e-08    5  1
2   0  0  0  0  0  1  4.607016 1.607016 3.000000 2.473959e-15   29  2
3   0  0  0  0  1  0  3.607016 1.607016 2.000000 4.586534e-16   15  3
4   0  0  0  0  1  1  4.607016 1.607016 3.000000 1.133129e-15   71  4
5   0  0  0  1  0  0  3.607016 1.607016 2.000000 9.071183e-16   29  5
6   0  0  0  1  0  1  4.607016 1.607016 3.000000 2.315886e-15  128  6
```

Then we instantiate an object `source.rep.obj` of class `SEstimator_pp` to compute weighted treatment effect using RWD:

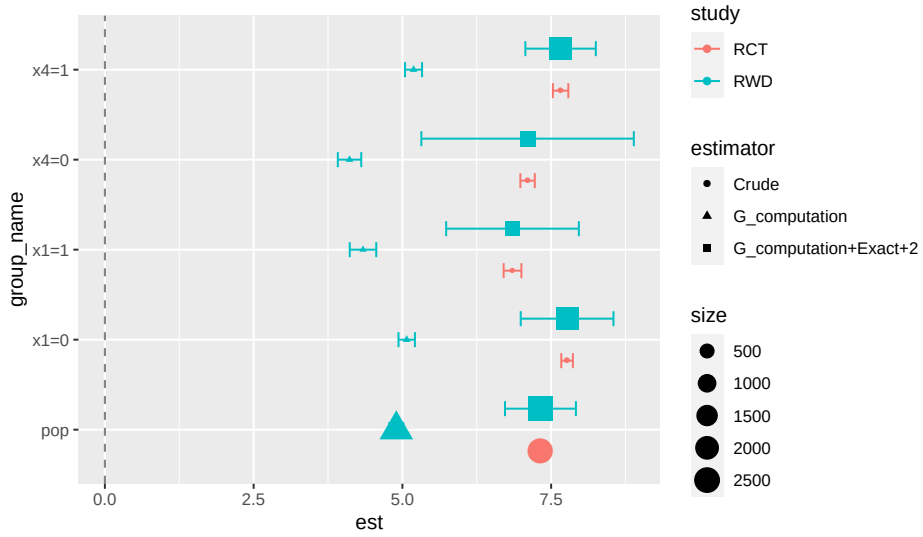
```
strata <- c("x1","x4")
source.rep.obj <- SEstimator_wrapper(estimator="SEexact_pp",
```

variates patterns defined by `confounders_sampling_name` should be findable in `source.data` and `target.data`. 2. the way to encode cateogorical data is the same. Researchers can transform their data into a common format using *OMOP common data model* (Hripcsak, Duke, Shah, Reich, Huser, Schuemie, Suchard, Park, Wong, Rijnbeek *et al.* 2015)

```
                                    target.obj=target.obj,
                                    source.obj=source.obj,
                                    confounders_sampling_name=confounders_sampling_name)
source.rep.obj$EstimateRep(stratification = strata, stratification_joint = TRUE)
fusion <- Summary$new(target.obj,
                      source.obj,
                      source.rep.obj)
```

Then we call `plot()`, `print()`, `evaluate()` to get results of methods:

```
> fusion$plot()
> fusion$print()
# A tibble: 15 x 9
# Groups:   group_name [5]
   group_name study estimator              est  ci_l  ci_u  size y1.hat y0.hat
   <chr>      <chr> <chr>                <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl>
 1 pop        RCT   Crude                 7.32  7.19  7.45  2686   7.03 -0.294
 2 pop        RWD   G_computation         4.90  4.78  5.02  2618   5.50  0.599
 3 pop        RWD   G_computation+Exact+2 7.32  6.72  7.92  2618   7.89  0.565
 4 x1=0       RCT   Crude                 7.77  7.67  7.87    27   6.98 -0.791
 5 x1=1       RCT   Crude                 6.85  6.70  7.00    25   7.06  0.209
 6 x4=0       RCT   Crude                 7.10  6.98  7.22    26   6.74 -0.369
 7 x4=1       RCT   Crude                 7.66  7.53  7.79    26   7.46 -0.2
 8 x1=0       RWD   G_computation         5.07  4.93  5.21    27   5.43  0.36
 9 x1=1       RWD   G_computation         4.34  4.12  4.56    25   5.70  1.36
10 x4=0       RWD   G_computation         4.11  3.92  4.31    26   4.73  0.614
11 x4=1       RWD   G_computation         5.19  5.04  5.33    26   5.78  0.593
12 x1=0       RWD   G_computation+Exact+2 7.77  6.99  8.55  1994   8.13  0.36
13 x1=1       RWD   G_computation+Exact+2 6.85  5.74  7.97   624   8.21  1.36
14 x4=0       RWD   G_computation+Exact+2 7.10  5.32  8.89   706   7.64  0.541
15 x4=1       RWD   G_computation+Exact+2 7.66  7.07  8.25  1912   8.22  0.566
> fusion$evaluate()
# A tibble: 10 x 7
# Groups:   group_name [5]
   group_name estimator              size   mse len_ci agg.est agg.reg
   <chr>      <chr>                 <dbl> <dbl>  <dbl> <lgl>   <lgl>
 1 pop        G_computation+Exact+2  2618    0   1.19  TRUE    TRUE
 2 pop        G_computation          2618  588.  0.237 FALSE   TRUE
 3 x1=0       G_computation+Exact+2  1994    0   1.56  TRUE    TRUE
 4 x1=0       G_computation            27  727.  0.276 FALSE   TRUE
 5 x1=1       G_computation+Exact+2   624    0   2.23  TRUE    TRUE
 6 x1=1       G_computation            25  632.  0.446 FALSE   TRUE
 7 x4=0       G_computation+Exact+2   706    0   3.57  TRUE    TRUE
 8 x4=0       G_computation            26  896.  0.394 FALSE   TRUE
 9 x4=1       G_computation+Exact+2  1912    0   1.18  TRUE    TRUE
10 x4=1       G_computation            26  611.  0.286 FALSE   TRUE
```



7.3. Example 3: Evaluation using synthetic experimental data

In example 2 we demonstrate the evaluation approach using aggregated data of $\boldsymbol{x}$. However, in practice, we don't even have access to the aggregated data of $\boldsymbol{x}$ of experimental data. Instead, we only have aggregated data of univariate covariates and average treatment effect of subgroups stratified by levels of univariate variable. In example 3 we demonstrate the evaluation approach based on univariate distribution of covariates of experimental data. First, we instantiate an object of class `Crude` using full experimental data which is only for demonstrative use:

```
target.obj <- TEstimator_wrapper(
  Estimator = "Crude",
  data = target.data,
  vars_name = vars_name,
  name = "RCT",
  data.public = FALSE,
  isTrial = TRUE
)
```

Then we generate a synthetic experimental dataset `synthetic.data` using R package **copula**. We specify marginal distribution of each variable from experimental data, specify pairwise correlations between these variables according to evidence from similar population or assumptions, and generate the synthetic experimental data accordingly:

```
t.d <- target.data[,c("x1","x2","x3","x4","x5","x6")]
pw.cor <- gdata::upperTriangle(cor(t.d), diag = FALSE, byrow = TRUE)
myCop <- copula::normalCopula(param=pw.cor,
                      dim = 6, dispstr = "un")
myMvd <- copula::mvdc(copula=myCop,
                      margins = c("binom","binom","binom","binom","binom","binom"),
                      paramMargins=list(list(1, mean(target.data$x1)),
                                        list(1, mean(target.data$x2)),
```

```
                                        list(1, mean(target.data$x3)),
                                        list(1, mean(target.data$x4)),
                                        list(1, mean(target.data$x5)),
                                        list(1, mean(target.data$x6))))
synthetic.data <- copula::rMvdc(dim(source.data)[1], myMvd)
synthetic.data <- data.frame(x1=synthetic.data[,1],
                             x2=synthetic.data[,2],
                             x3=synthetic.data[,3],
                             x4=synthetic.data[,4],
                             x5=synthetic.data[,5],
                             x6=synthetic.data[,6]) %>%
                             arrange(across(c("x1","x2","x3","x4","x5","x6")))
```

The generated synthetic experimental dataset `synthetic.data` has the same marginal distributions as that of the original experimental data `target.data`:

```
apply(synthetic.data, 2, mean)
```

```
>       x1        x2        x3        x4        x5        x6
0.4828113 0.8208556 0.3441558 0.3827349 0.2467532 0.3479756
```

```
c(mean(target.data$x1),mean(target.data$x2),mean(target.data$x3),
  mean(target.data$x4),mean(target.data$x5),mean(target.data$x6))
```

```
> [1] 0.4880864 0.8276247 0.3432614 0.3901713 0.2431124 0.3551750
```

Then we instantiate a target object of class `Synthetic_TEstimator`. We initialize the public field `data` by assigning `synthetic.data` to `df`; initialize the public field `estimates` by assigning `RCT.summary` to `estimates`; initialize the public field `confounders_treatment_name` by assigning `c("x1","x2","x3","x4","x5","x6")` to `vars_name`. Note that we match `source.data` and `synthetic.data` so that two datasets have overlap in variables space defined by `confounders_sampling_` which might slight shift the distribution of synthetic experimental data `synthetic.data` away from original experimental data `target.data`.

```
RCT.summary <- list(ATE_mean = target.obj$estimates$ATE$est,
                    ATE_se = target.obj$estimates$ATE$se,
                    CATE_mean_se=target.obj$get_CATE(
                                c("x1","x2","x3","x4","x5","x6"),FALSE))
synthetic.data <- semi_join(synthetic.data, source.data,
                            by = c("x1","x2","x3","x4","x5","x6"))
source.data <- semi_join(source.data, synthetic.data,
                         by = c("x1","x2","x3","x4","x5","x6"))
target.obj <- Synthetic_TEstimator$new(df = synthetic.data,
                                       estimates = RCT.summary,
                                       vars_name = c("x1","x2","x3","x4","x5","x6"),
                                       name = "RCT",
                                       isTrial = TRUE,
```

```
                                        data.public = FALSE)
target.obj$data
```

```
# A tibble: 52 x 7
# Groups:   x1, x2, x3, x4, x5 [30]
      x1    x2    x3    x4    x5    x6  size
   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int>
 1     0     0     0     0     0     0    60
 2     0     0     0     0     0     1    26
 3     0     0     0     0     1     0    19
 4     0     0     0     0     1     1     3
 5     0     0     0     1     0     0    15
 6     0     0     0     1     0     1    13
 7     0     0     0     1     1     1     3
 8     0     0     1     0     0     0    15
 9     0     0     1     0     0     1     7
10     0     0     1     0     1     0     2
# ... with 42 more rows
```

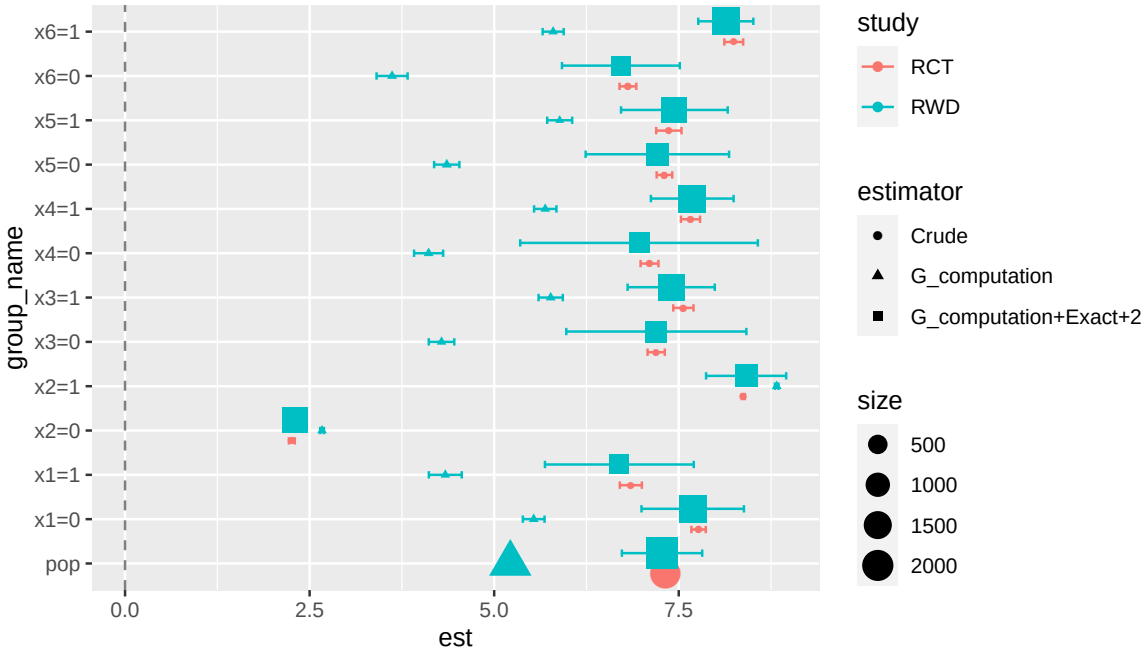
Then we instantiate an object `source.obj` of class `G_computation` with `data.public=FALSE`, and instantiate an object `source.rep.obj` of class `SEexact_pp`:

```
source.obj <- TEstimator_wrapper(
  Estimator = "G_Computation",
  data = source.data,
  vars_name = vars_name,
  outcome_method = "glm",
  outcome_form=y ~ x1 + x2 + x3 + z + z:x1 + z:x2 +z:x3+ z:x6,
  name = "RWD",
  data.public = FALSE
)

source.rep.obj <- SEstimator_wrapper(estimator="Exact_pp",
                                     target.obj=target.obj,
                                     source.obj=source.obj,
                                     confounders_sampling_name=c("x2","x6"))
source.rep.obj$EstimateRep(stratification = c("x1","x2","x3","x4","x5","x6"), stratificati
fusion <- Summary$new(target.obj,
                      source.obj,
                      source.rep.obj)
```

Finally, we can plot and evaluate the method in `source.rep.obj`:

```
> fusion$plot()
```


group_name
x6=1
x6=0
x5=1
x5=0
x4=1
x4=0
x3=1
x3=0
x2=1
x2=0
x1=1
x1=0
pop
0.0
2.5
5.0
7.5
est
study
RCT
RWD
estimator
Crude
G_computation
G_computation+Exact+2
size
500
1000
1500
2000

```
> fusion$evaluate()
```

	group_name	estimator	size	bias	mse	len_ci	agg.est	agg.reg
	<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<lgl>	<lgl>
1	pop	G_computation+Exact+6	2423	17.0	2.87	3.80	FALSE	TRUE
2	pop	G_computation	2423	219.	480.	0.247	FALSE	TRUE
3	x1=0	G_computation+Exact+6	1799	22.0	4.85	5.58	FALSE	TRUE
4	x1=0	G_computation	25	236.	558.	0.29	FALSE	TRUE
5	x1=1	G_computation+Exact+6	624	22.8	5.21	4.79	FALSE	TRUE
6	x1=1	G_computation	25	251.	632.	0.446	FALSE	TRUE
7	x2=0	G_computation+Exact+6	1447	3.79	0.143	0.812	TRUE	TRUE
8	x2=0	G_computation	24	37.9	14.3	0.05	FALSE	TRUE
9	x2=1	G_computation+Exact+6	976	3.49	0.122	4.71	FALSE	TRUE
10	x2=1	G_computation	26	45.2	20.4	0.048	FALSE	TRUE

Results in ?? show that even though we don't have full experimental dataset, given their aggregated data of each univaraite variable, the weighted estimates of treatment effect are close to the unbiased estimate using original experimental data, hence we can still evaluate methods for treatment effect estimation using RWD to some extent and obtain qualitative results based on evaluation metrics. Note that in this example, we know all the variables that are predictive to outcomes and simulate the synthetic dataset accordingly, in case without knowing all variables that are predictive to outcomes and how these variables affect outcomes, we are not able to properly recover the distribution of experimental data using only marginal distributions of the limited number of variables from a RCT. However, weighting variables that are highly predictive to outcomes and are highly imbalanced between RWD and experimental data will always obtain estimates that are much closer to those without weighting the variables.

7.4. Example 4: Effect of adjustment sets on precision

In example 4, we demonstrate how to use **RCTrep** to explore the effect of adjustment sets on estimation precisoin in an easy and fast manner. According to the DGM of `target.data` and `source.data` used throughout our examples, the treatment effect $\tau(\boldsymbol{x})$ is the function of X_2 and X_6, hence the variation of treatment effect only depends on the distribution of these two variables. We can relax the S-ignorability assumption - i.e., conditional on X_2 and X_6, estimation of treatment effect are comparable - and hence the minimal adjustment sets for `SEstimator` is `confounders_sampling_name=c("x2","x6")`. In the following, we compare the effect of different sets of `confounders_sampling_name` on the efficiency of estimatio. We instantiate two different objects of class `SEstimator` each with `confounders_sampling_name=c("x2","x6")` and `confounders_sampling_name=c("x1","x2","x3","x4",` respectively. First, we instantiate `source.obj` and `target.obj`, respectively:

```
source.obj <- TEstimator_wrapper(
  Estimator = "G_computation",
  data = source.data,
  vars_name = vars_name,
```

```
  outcome_method = "glm",
  outcome_form=y ~ x1 + x2 + x3 + z + z:x1 + z:x2 +z:x3+ z:x6,
  name = "RWD",
  data.public = TRUE
)

target.obj <- TEstimator_wrapper(
  Estimator = "Crude",
  data = target.data,
  vars_name = vars_name,
  name = "RCT",
  data.public = TRUE,
  isTrial = TRUE
)
```

Then we instantiate two objects `source.obj.1` and `source.obj.2` of class `SEstimator`. `source.obj.1` computes weight $w(\boldsymbol{x})$ for each individual according to distribution of `x2,x3`; `source.obj.2` computes weight according to distribution of `x1,x2,x3,x4,x5,x6`:

```
source.obj.1 <- SEstimator_wrapper(estimator="Exact",target.obj=target.obj,
                                   source.obj=source.obj,
                                   confounders_sampling_name=c("x2","x6"))
source.obj.1$EstimateRep(stratification = c("x1","x2","x3","x4","x5","x6"),
                         stratification_joint = FALSE)

source.obj.2 <- SEstimator_wrapper(estimator="Exact",target.obj=target.obj,
                                   source.obj=source.obj,
                                   confounders_sampling_name=
                                   c("x1","x2","x3","x4","x5","x6"))
source.obj.2$EstimateRep(stratification = c("x1","x2","x3","x4","x5","x6"),
                         stratification_joint = FALSE)

fusion <- Summary$new(target.obj,
                      source.obj,
                      source.obj.1,
                      source.obj.2)

fusion$plot()
fusion$evaluate()

# A tibble: 39 x 8
# Groups:   group_name [13]
   group_name estimator              size    bias     mse len_ci agg.est agg.reg
   <chr>      <chr>                 <dbl>   <dbl>   <dbl>  <dbl> <lgl>   <lgl>
 1 pop        G_computation+Exact+2  2618   0.073   0       1.19  TRUE    TRUE
 2 pop        G_computation+Exact+6  2618   0.073   0       4.52  TRUE    TRUE
 3 pop        G_computation          2618 242.    588.      0.237 FALSE   TRUE
```

```
 4 x1=0       G_computation+Exact+2  1994   2.69    0.072  1.56  TRUE    TRUE
 5 x1=0       G_computation+Exact+6  1994   2.69    0.072  7.36  TRUE    TRUE
 6 x1=0       G_computation          1994 272.     741.     0.276 FALSE   TRUE
 7 x1=1       G_computation+Exact+2   624   0.806   0.006  2.23  TRUE    TRUE
 8 x1=1       G_computation+Exact+6   624   0.806   0.006  5.09  TRUE    TRUE
 9 x1=1       G_computation           624 251.    628.     0.446 FALSE   TRUE
10 x2=0       G_computation+Exact+2  1642   2.45    0.06   0.097 TRUE    TRUE
# ... with 29 more rows
```

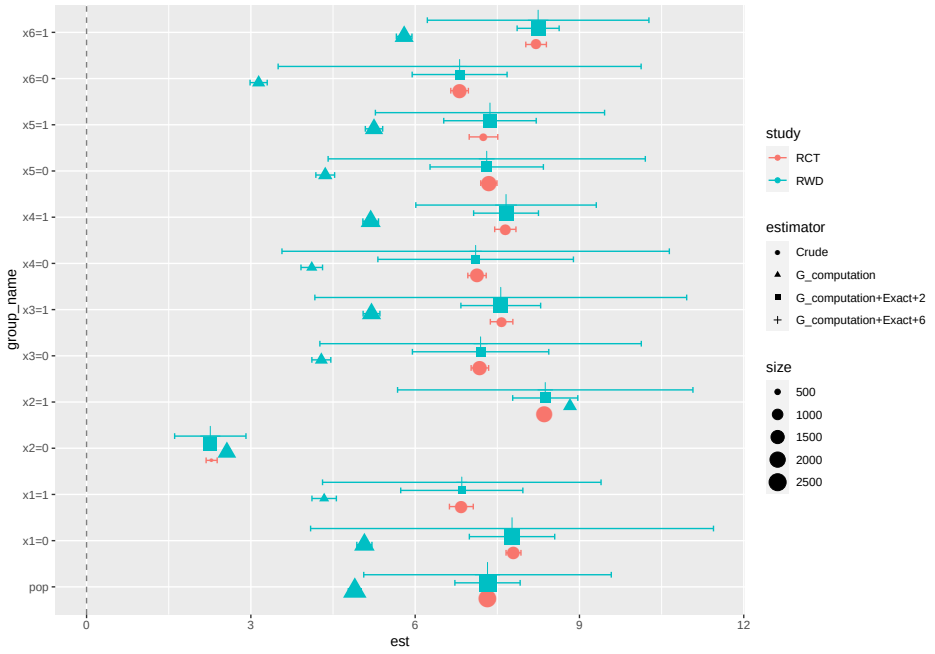


Figure 12: weighting according to heterogeneity set and weighting according to all pretreatment covariate sets

The results in the figure 12 show that standard error of estimates indicated by `G_computation+Exact+2` is smaller than that indicated by `G_computation+Exact+6`. The results imply that weighting variables which are not predictive to treatment effect can inflate variance of estimates, and hence reduce precision. Further investigation of effect of adjustment sets on bias and variance of estimates of average treatment effect and weighted estimates of average treatment effect can be quickly tested and validated using **RCTrep**.

8. Discussion

RCTrep provides a principled and fair approach for the evaluation of methods for treatment effect estimation using RWD. **RCTrep** evaluates methods on both population and subpopulation levels. **RCTrep** shows that using aggregated data, methods for treatment effect estimation can yield the same estimates as that using individual level data. Via proper method evaluation on both population and sub-population levels, the error in equation 4 can shed light on the possible existence of (unmeasured) confounder(s). The error on population

and sub-population levels can provide insight/evidence on the identification of new research questions and proposal of new assumptions to better understand the underlying mechanism. For instance, unmeasured confounders that are positively correlated with the treatment and the outcome can lead to overestimation of the treatment effect. Overestimation observed in a sub-population can guide us on where to identify possible important confounder(s) to improve the precision of the estimation [21]. However, in practice, beyond bias induced by confounders which can lead to a variation in estimates, a small sample size leading to underfit of model can affect the precision of estimates, and should be further investigated using more data to reach good statistical power.

Users should bear in mind that our approach is under the assumption of ignorability of the sampling mechanism - meaning after adjusting the sampling mechanism, experimental data and RWD are assumed as two random samples from the same population - otherwise, evaluation is meaningless since we are comparing two different populations [22] despite appealing potentials of RWD and advanced modeling choices. Hence, under the assumption of S-ignorability, there is no unobserved difference between two populations, and the difference is solely due to the performance of methods for treatment effect estimation. Besides, it is worth noting that for a binary outcome, G-computation using logistic regression can suffer from omitted variable bias even the variable is independent of treatment, hence can bias the association between a risk factor and an outcome towards a null hypothesis if some variables that cause outcome are not adjusted (Mood 2010). In addition, more samples are needed for binary outcome modeling. According to a guideline suggested in (Stoltzfus 2011), the incidence of outcome should be more than 50 for each subgroup.

Another limitation of the study is that to what extent we can have comparative performance of different methods depends on the quality of the selected RCT. If the confidence interval of the "truth" from the RCT is wide, then estimates from most methods may fall within the interval, leading to difficulty in performance evaluation. This may not matter if the regulatory agreement between "truth" and estimation using RWD is consistent; however, when the estimate and the "truth" have the same estimate agreement while inconsistent regulatory agreement, we should carefully investigate the uncertainty of weighted estimates using RWD, for instance, coverage and width of the confidence interval of weighted estimates, and more estimation methods for the variance of weighted estimates can be further evaluated, for instance, double bootstrap (Ackerman *et al.* 2021), a consistent sandwich-type variance estimator proposed by (Buchanan, Hudgens, Cole, Mollan, Sax, Daar, Adimora, Eron, and Mugavero 2018).

Our work provides the following insights for future research. Firstly, in case that the sampling mechanism of RWD is not properly corrected, can we still evaluate the performance of methods for treatment effect estimation using RWD and how? What is the effect of an omitted variable that is predictive to the heterogeneous treatment effect on the magnitude and direction of the bias and variance of weighted average treatment effect? Although we have provided preliminary discussion in appendix E, more studies need to be carried out in

[21] We provide an analysis of direction and magnitude of bias due to unmeasured confounder using G-computation method in appendix D.

[22] two populations are heterogeneous because there is unobserved variable(s) which is predictive to the outcome, however, is not balanced between two populations. Hence two populations have a systematic difference in the unobserved variable even though all observed variables are balanced. In case there is no overlap of the unobserved variable(s) between two populations, then two populations have heterogeneity on the level of the variable.

order to validate our findings. Secondly, studies regarding the optimal adjustment sets to allow for the evaluation of causal inference methods using RWD would be a fruitful area for further work. For instance, the precision of estimates using seven sets of covariates can be further evaluated (those causing the outcome, those causing the treatment assignment, those causing the sampling mechanism, those causing both outcome and treatment, those causing both the outcome and sampling, those causing both the treatment and sampling, and all). The impact of omitted variables in one of these seven covariate sets on the direction and magnitude of bias can be further assessed.

References

Aalen OO, Farewell VT, De Angelis D, Day NE, Nöel Gill O (1997). "A Markov model for HIV disease progression including the effect of HIV diagnosis and treatment: application to AIDS prediction in England and Wales." *Statistics in medicine*, **16**(19), 2191–2210.

Ackerman B, Lesko CR, Siddique J, Susukida R, Stuart EA (2021). "Generalizing randomized trial findings to a target population using complex survey population data." *Statistics in Medicine*, **40**(5), 1101–1120.

Alaa A, Van Der Schaar M (2019). "Validating causal inference models via influence functions." In *International Conference on Machine Learning*, pp. 191–201. PMLR.

Almond D, Chay KY, Lee DS (2005). "The costs of low birth weight." *The Quarterly Journal of Economics*, **120**(3), 1031–1083.

Atan O, Jordon J, Van der Schaar M (2018). "Deep-treat: Learning optimal personalized treatments from observational data using neural networks." In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Bareinboim E, Pearl J (2016). "Causal inference and the data-fusion problem." *Proceedings of the National Academy of Sciences*, **113**(27), 7345–7352.

Bica I, Alaa AM, Lambert C, Van Der Schaar M (2021). "From real-world patient data to individualized treatment effects using machine learning: current and future methods to address underlying challenges." *Clinical Pharmacology & Therapeutics*, **109**(1), 87–100.

Bloniarz A, Liu H, Zhang CH, Sekhon JS, Yu B (2016). "Lasso adjustments of treatment effect estimates in randomized experiments." *Proceedings of the National Academy of Sciences*, **113**(27), 7383–7390.

Buchanan AL, Hudgens MG, Cole SR, Mollan KR, Sax PE, Daar ES, Adimora AA, Eron JJ, Mugavero MJ (2018). "Generalizing evidence from randomized trials using inverse probability of sampling weights." *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, **181**(4), 1193–1209.

Chang W (????). "R6: Classes with Reference Semantics, 2017." *URL https://CRAN. R-project. org/package*, **6**, 90.

Chatton A, Le Borgne F, Leyrat C, Gillaizeau F, Rousseau C, Barbin L, Laplaud D, Léger M, Giraudeau B, Foucher Y (2020). "G-computation, propensity score-based methods, and targeted maximum likelihood estimator for causal inference with different covariates sets: a comparative simulation study." *Scientific reports*, **10**(1), 1–13.

Chattopadhyay A, Hase CH, Zubizarreta JR (2020). "Balancing vs modeling approaches to weighting in practice." *Statistics in Medicine*, **39**(24), 3227–3254.

Cheng L, Guo R, Moraffah R, Sheth P, Candan KS, Liu H (2022). "Evaluation Methods and Measures for Causal Learning Algorithms." *IEEE Transactions on Artificial Intelligence*.

Cinelli C, Pearl J (2021). "Generalizing experimental results by leveraging knowledge of mechanisms." *European Journal of Epidemiology*, **36**(2), 149–164.

Colnet B, Mayer I, Chen G, Dieng A, Li R, Varoquaux G, Vert JP, Josse J, Yang S (2020). "Causal inference methods for combining randomized trials and observational studies: a review." *arXiv preprint arXiv:2011.08047*.

Dahabreh IJ, Robertson SE, Steingrimsson JA, Stuart EA, Hernan MA (2020). "Extending inferences from a randomized trial to a new target population." *Statistics in medicine*, **39**(14), 1999–2014.

Deaton A, Cartwright N (2018). "Understanding and misunderstanding randomized controlled trials." *Social Science & Medicine*, **210**, 2–21.

Dong L, Yang S, Wang X, Zeng D, Cai J (2020). "Integrative analysis of randomized clinical trials with real world evidence studies." *arXiv preprint arXiv:2003.01242*.

Dorie V, Hill J, Shalit U, Scott M, Cervone D (2019). "Automated versus do-it-yourself methods for causal inference: Lessons learned from a data analysis competition." *Statistical Science*, **34**(1), 43–68.

Egami N, Hartman E (2018). "Covariate selection for generalizing experimental results." *Technical report*, Technical report Working Paper.

Franklin JM, Pawar A, Martin D, Glynn RJ, Levenson M, Temple R, Schneeweiss S (2020). "Nonrandomized real-world evidence to support regulatory decision making: process for a randomized trial replication project." *Clinical Pharmacology & Therapeutics*, **107**(4), 817–826.

Franklin JM, Schneeweiss S, Polinski JM, Rassen JA (2014). "Plasmode simulation for the evaluation of pharmacoepidemiologic methods in complex healthcare databases." *Computational statistics & data analysis*, **72**, 219–226.

Hahn PR, Dorie V, Murray JS (2019). "Atlantic causal inference conference (acic) data analysis challenge 2017." *arXiv preprint arXiv:1905.09515*.

Hill JL (2011). "Bayesian nonparametric modeling for causal inference." *Journal of Computational and Graphical Statistics*, **20**(1), 217–240.

Hitsch GJ, Misra S (2018). "Heterogeneous treatment effects and optimal targeting policy evaluation." *Available at SSRN 3111957*.

Hosmer Jr DW, Lemeshow S, Sturdivant RX (2013). *Applied logistic regression*, volume 398. John Wiley & Sons.

Hripcsak G, Duke JD, Shah NH, Reich CG, Huser V, Schuemie MJ, Suchard MA, Park RW, Wong ICK, Rijnbeek PR, *et al.* (2015). "Observational Health Data Sciences and Informatics (OHDSI): opportunities for observational researchers." *Studies in health technology and informatics*, **216**, 574.

Imbens GW, Rubin DB (2015). *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press.

Jiang H, Qi P, Zhou J, Zhou J, Rao S (2021). "A Short Survey on Forest Based Heterogeneous Treatment Effect Estimation Methods: Meta-learners and Specific Models." In *2021 IEEE International Conference on Big Data (Big Data)*, pp. 3006–3012. IEEE.

Johansson FD, Shalit U, Kallus N, Sontag D (2020). "Generalization bounds and representation learning for estimation of potential outcomes and causal effects." *arXiv preprint arXiv:2001.07426*.

Kang JD, Schafer JL (2007). "Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data." *Statistical science*, pp. 523–539.

Kouw WM, Loog M (2018). "An introduction to domain adaptation and transfer learning." *arXiv preprint arXiv:1812.11806*.

Künzel SR, Sekhon JS, Bickel PJ, Yu B (2019). "Metalearners for estimating heterogeneous treatment effects using machine learning." *Proceedings of the national academy of sciences*, **116**(10), 4156–4165.

LaLonde RJ (1986). "Evaluating the econometric evaluations of training programs with experimental data." *The American economic review*, pp. 604–620.

Le Borgne F, Chatton A, Léger M, Lenain R, Foucher Y (2021). "G-computation and machine learning for estimating the causal effects of binary exposure statuses on binary outcomes." *Scientific reports*, **11**(1), 1–12.

Little RJ, Vartivarian S (2005). "Does weighting for nonresponse increase the variance of survey means?" *Survey Methodology*, **31**(2), 161.

Loiseau N, Trichelair P, He M, Andreux M, Zaslavskiy M, Wainrib G, Blum MG (2022). "External control arm analysis: an evaluation of propensity score approaches, G-computation, and doubly debiased machine learning." *medRxiv*. doi:10.1101/2022.01.28.22269591. https://www.medrxiv.org/content/early/2022/01/30/2022.01.28.22269591.full.pdf, URL https://www.medrxiv.org/content/early/2022/01/30/2022.01.28.22269591.

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.

Mood C (2010). "Logistic regression: Why we cannot do what we think we can do, and what we can do about it." *European sociological review*, **26**(1), 67–82.

Powers S, Qian J, Jung K, Schuler A, Shah NH, Hastie T, Tibshirani R (2018). "Some methods for heterogeneous treatment effect estimation in high dimensions." *Statistics in medicine*, **37**(11), 1767–1787.

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.

Rudolph KE, Schmidt NM, Glymour MM, Crowder R, Galin J, Ahern J, Osypuk TL (2018). "Composition or context: using transportability to understand drivers of site differences in a large-scale housing experiment." *Epidemiology (Cambridge, Mass.)*, **29**(2), 199.

Saul BC, Hudgens MG (2020). "The Calculus of M-Estimation in R with geex." *Journal of statistical software*, **92**(2).

Schuemie MJ, Cepeda MS, Suchard MA, Yang J, Tian Y, Schuler A, Ryan PB, Madigan D, Hripcsak G (2020). "How confident are we about observational findings in healthcare: a benchmark study." *Harvard data science review*, **2**(1).

Schuler A, Jung K, Tibshirani R, Hastie T, Shah N (2017). "Synth-validation: Selecting the best causal inference method for a given dataset." *arXiv preprint arXiv:1711.00083*.

Shen L, Visser E, de Wilt H, Verheul H, van Erning F, Geleijnse G, Kaptein M (2020). "Estimating the effect of adjuvant chemo-therapy for colon-cancer using registry data: a method comparison and validation."

Shimoni Y, Yanover C, Karavani E, Goldschmnidt Y (2018). "Benchmarking framework for performance-evaluation of causal inference analysis." *arXiv preprint arXiv:1802.05046*.

Stoltzfus JC (2011). "Logistic regression: a brief primer." *Academic Emergency Medicine*, **18**(10), 1099–1104.

Stuart EA (2010). "Matching methods for causal inference: A review and a look forward." *Statistical science: a review journal of the Institute of Mathematical Statistics*, **25**(1), 1. doi:10.1214/09-STS313.

Sugiyama M, Suzuki T, Kanamori T (2012). *Density ratio estimation in machine learning*. Cambridge University Press.

Swaminathan A, Joachims T (2015). "The self-normalized estimator for counterfactual learning." *advances in neural information processing systems*, **28**.

Tikka S, Hyttinen A, Karvanen J (2019). "Causal effect identification from multiple incomplete data sources: A general search-based approach." *arXiv preprint arXiv:1902.01073*.

Tikka S, Karvanen J (2017). "Identifying Causal Effects with the R Package causaleffect." *Journal of Statistical Software*, **76**(12), 1–30. doi:10.18637/jss.v076.i12. URL https://www.jstatsoft.org/index.php/jss/article/view/v076i12.

Tikka S, Karvanen J (2018). "Identifying causal effects with the R package causaleffect." *arXiv preprint arXiv:1806.07161*.

Wager S, Athey S (2018). "Estimation and inference of heterogeneous treatment effects using random forests." *Journal of the American Statistical Association*, **113**(523), 1228–1242.

Wendling T, Jung K, Callahan A, Schuler A, Shah NH, Gallego B (2018). "Comparing methods for estimation of heterogeneous treatment effects using observational data from health care databases." *Statistics in medicine*, **37**(23), 3309–3324.

Xie Y, Brand JE, Jann B (2012). "Estimating heterogeneous treatment effects with observational data." *Sociological methodology*, **42**(1), 314–347.

Yao L, Li S, Li Y, Huai M, Gao J, Zhang A (2018). "Representation learning for treatment effect estimation from observational data." *Advances in Neural Information Processing Systems*, **31**.

Zeng S, Li F, Wang R, Li F (2021). "Propensity score weighting for covariate adjustment in randomized clinical trials." *Statistics in medicine*, **40**(4), 842–858.

A. Notations in the paper

In this section, we illustrate the notation we used throughout the paper. Let $\mathcal{D}_r = \{(\boldsymbol{x}_i, y_i, z_i); i \in \mathcal{R}\}$ denote an experimental dataset of a RCT , $\mathcal{D}_o = \{(X_i, Y_i, Z_i); i \in \mathcal{O}\}$ denote a RWD dataset of an observational study, $\tau(\boldsymbol{x})$ denote conditional average treatment effect conditional on $\boldsymbol{x}$, $f(\boldsymbol{x})$ denote estimate of treatment effect conditional on $\boldsymbol{x}$, τ denote the average treatment effect. See Table 3 for lists of all notations used throughout this paper.

Table 3: list of notations

notation	description
$\boldsymbol{X}$	random vector of length k of covariates
Z	treatment indicator ($Z = 1$ for treatment, $Z = 0$ for control)
Y	outcome of interest ($Y = 1$ for survival, $Z = 0$ for death)
S	selection indicator ($S = 1$ for a RCT, $S = 0$ for an observational study)
$\mathcal{R}$	$\mathcal{R} = \{i : S_i = 1\}$
$\mathcal{O}$	$\mathcal{O} = \{i : S_i = 0\}$
$\mathcal{D}_r$	a sample from a RCT, denoted as $\mathcal{D}_r = \{(x_i, z_i, y_i) : i \in \mathcal{R}\}$
$\mathcal{D}_o$	RWD from an observational study, denoted as $\mathcal{D}_o = \{(x_i, z_i, y_i) : i \in \mathcal{O}\}$
$\pi_z(\boldsymbol{x})$	propensity score of a unit with characteristics $\boldsymbol{X} = \boldsymbol{x}$ being selected to treatment $Z = 1$
$\pi_s(\boldsymbol{x})$	sampling score of a unit with characteristics being selected to an RCT $S = 1$
$\tau(\boldsymbol{x})$	the conditional average treatment effect, denoted as $\tau(\boldsymbol{x}) = \mathbb{E}[Y(1) - Y(0) \mid \boldsymbol{X} = \boldsymbol{x}]$
σ_1^2, σ_0^2	variance of potential outcomes $Y(1)$, $Y(0)$
$\sigma_z^2(\boldsymbol{x})$	conditional variance of $Y(z)$, denoted as $\mathbb{V}(Y(z) \mid \boldsymbol{x})$
$p(\boldsymbol{x}), q(\boldsymbol{x})$	density of covariates in an RCT, and observational study
$w(\boldsymbol{x})$	the density ratio of covariates $\boldsymbol{x}$ defined as $\frac{p(\boldsymbol{x})}{q(\boldsymbol{x})}$
$\pi_z(\boldsymbol{X}; \hat{\boldsymbol{\alpha}})$	an estimator for propensity score
$\pi_s(\boldsymbol{X}; \hat{\boldsymbol{\gamma}})$	an estimator for sampling score
$p(\boldsymbol{X}, z; \hat{\boldsymbol{\beta}})$	a G_computation estimator for the conditional expected potential outcome $\mathbb{E}[Y(z) \mid \boldsymbol{x}]$ parameterized by $\hat{\boldsymbol{\beta}}$
$f(\boldsymbol{X})$	an estimator for the conditional average treatment effect $\tau(\boldsymbol{x})$
$\hat{\sigma}_1, \hat{\sigma}_0$	estimator for variance of $Y(1), Y(0)$
$\hat{p}(\boldsymbol{X}), \hat{q}(\boldsymbol{X})$	estimator for density of $\boldsymbol{x}$ in RCT and observational data
ϵ_i^z	residual of estimator $p_z(\boldsymbol{X}, z_i; \hat{\boldsymbol{\beta}})$, defined as $\epsilon_i = Y_i - p(\boldsymbol{X}_i, z_i; \hat{\boldsymbol{\beta}})$
$\hat{\sigma}_z^2(\boldsymbol{x})$	an estimator of conditional variance of $Y(z)$, denoted as $\hat{\mathbb{V}}(Y(z) \mid \boldsymbol{x})$

B. Variance of estimators for average treatment effect

In **RCTrep**, we use three methods for treatment effect estimation, namely, G_computation, IPW, and doubly robust methods. G_computation method is unbiased and consistent as long as a model for outcome (i.e., $p(\boldsymbol{x}, z; \hat{\boldsymbol{\beta}})$) is correctly specified. IPW is unbiased and consistent as long as a model for treatment, i.e., propensity score $\pi_z(\boldsymbol{X}; \hat{\boldsymbol{\alpha}})$, is correctly specified. Doubly robust method is unbiased and consistent as long as either a model for outcome or a model for treatment is correctly specified, and is more efficient than IPW method. Note that we show variance of three methods for treatment effect estimation for illustrative purpose regarding the effect of weight on the variance estimation (i.e., IPW estimator), model fit on the variance estimation (i.e., G-computation), and sample size on the variance estimation. In the following, we analyze the variance of these methods.

B.1. Variance of G-computation

Assumptions of Z-ignorability implies that conditioning on confounders treatment assignment is regarded as random and hence the treatment effect can be identified as simple difference in means between two groups for each subgroup stratified by confounders. The average treatment effect using G-computation method is defined as:

$$\hat{\tau} = \mathbb{E}[f(\boldsymbol{X})] = \mathbb{E}[p(\boldsymbol{X}, 1; \hat{\boldsymbol{\beta}}) - p(\boldsymbol{X}, 0; \hat{\boldsymbol{\beta}})] \approx \frac{1}{n} \sum_i p(\boldsymbol{x}_i, 1; \hat{\boldsymbol{\beta}}) - p(\boldsymbol{x}_i, 0; \hat{\boldsymbol{\beta}}) \tag{5}$$

where $\boldsymbol{X}$ is a random vector of all confounders, $p(\boldsymbol{X}, Z; \hat{\boldsymbol{\beta}}) = \hat{\mathbb{E}}[Y \mid \boldsymbol{X}, Z]$. We can use both parametric and non-parametric models to estimate expected value of potential outcomes given the value of $\boldsymbol{x}$, in other words, $\hat{\boldsymbol{\beta}} \subset \mathbb{R}^{\mathbb{R}}$. Here we use $\boldsymbol{\beta}$ to denote a set of parameters that describe the distribution of conditional potential outcomes. We assume the conditional expectation is expressed as an equation linear in $\boldsymbol{x}$ and z, and hence can be described by a fixed length of parameters $\boldsymbol{\beta}$. We can also assume that conditional expectation can be described by a flexible function parameterized $\boldsymbol{\beta}$ of flexible length depending on model constraint, regularization and sample size. The simplest and unbiased estimator using experimental data is the difference in means between treatment and control groups of a subgroup stratified by $\boldsymbol{x}$, which is a non-parametric estimator describing the empirical distribution of potential outcomes within subgroups. $p(\boldsymbol{x}, 1; \hat{\boldsymbol{\beta}})$ is the estimate of $\mathbb{E}[Y(1) \mid \boldsymbol{x}, Z = 1]$ parameterized by $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\beta}}$ is estimated using RWD or experimental data. Then the variance of the estimator $f(\boldsymbol{X})$ is derived as:

$$\begin{aligned}
\mathbb{V}(f(\boldsymbol{X})) &= \mathbb{E}[\mathbb{V}(f(\boldsymbol{X}) \mid \boldsymbol{X})] + \mathbb{V}\left(\mathbb{E}[f(\boldsymbol{X}) \mid \boldsymbol{X}]\right) \quad \text{law of total variance} \\
&= \mathbb{E}\left[\mathbb{V}\left(p(\boldsymbol{X}, 1; \hat{\boldsymbol{\beta}}) - p(\boldsymbol{X}, 0; \hat{\boldsymbol{\beta}}) \mid \boldsymbol{X}\right)\right] + \mathbb{V}\left(\mathbb{E}[p(\boldsymbol{X}, 1; \hat{\boldsymbol{\beta}}) - p(\boldsymbol{X}, 0; \hat{\boldsymbol{\beta}}) \mid \boldsymbol{X}]\right) \\
&\approx \mathbb{E}\left[\mathbb{V}\left(p(\boldsymbol{X}, 1; \hat{\boldsymbol{\beta}}) \mid \boldsymbol{X}\right) + \mathbb{V}\left(p(\boldsymbol{X}, 0; \hat{\boldsymbol{\beta}}) \mid \boldsymbol{X}\right)\right] + \mathbb{V}\left(p(\boldsymbol{X}, 1; \bar{\hat{\boldsymbol{\beta}}}) - p(\boldsymbol{X}, 0; \bar{\hat{\boldsymbol{\beta}}})\right) \\
&\approx \frac{1}{n} \sum_i \left(\hat{\mathbb{V}}(p(\boldsymbol{x}_i, 1; \hat{\boldsymbol{\beta}})) + \hat{\mathbb{V}}(p(\boldsymbol{x}_i, 0; \hat{\boldsymbol{\beta}}))\right) + \hat{\mathbb{V}}(p(\boldsymbol{X}, 1; \bar{\hat{\boldsymbol{\beta}}}) - p(\boldsymbol{X}, 0; \bar{\hat{\boldsymbol{\beta}}}))
\end{aligned} \tag{6}$$

Note in the third line, the first term is the function of $\boldsymbol{X}$. When $\boldsymbol{X}$ is fixed, only $\hat{\boldsymbol{\beta}}$ is random and depends on the sample, hence the variance of this term depends on the variance of $\hat{\boldsymbol{\beta}}$

which depends on the sample. In logistic regression, variance of $\hat{\boldsymbol{\beta}}$ is well-developed and estimation is unbiased given model is correctly specified, and most of software can provide the estimate of variance of these parameters. In non-parametric methods, it is not trivial to write down the closed form of variance of parameters, alternative approaches to estimating $\mathbb{V}(p(\boldsymbol{x}_i, z_i; \hat{\boldsymbol{\beta}}))$ are delta method, bootstrap, etc. We introduce approaches for estimating the variance of $p(\boldsymbol{x}_i, z_i; \hat{\boldsymbol{\beta}})$ in the next section. The second term in the third line is the variance between groups $\mathbb{V}(f(\boldsymbol{X}; \bar{\hat{\boldsymbol{\beta}}}))$, and only $\boldsymbol{X}$ is random, hence the variance of the second term can be estimated using sample variance of estimated $f(\boldsymbol{X}; \bar{\hat{\beta}})$ where $\bar{\hat{\boldsymbol{\beta}}} = \mathbb{E}[\hat{\boldsymbol{\beta}}]$, which is the true value of $\boldsymbol{\beta}$ by OLS. We use an estimate of $\hat{\boldsymbol{\beta}}$ based on a sample as an estimate of $\bar{\hat{\boldsymbol{\beta}}}$, and estimate the sample variance of plugged in $f(\boldsymbol{X}; \bar{\hat{\boldsymbol{\beta}}})$.

Methods for estimating the variance of G-computation

In this section, we illustrate five methods for estimating the variance of G-Computation method for potential outcomes, i.e., $\mathbb{V}(p(\boldsymbol{x}, z; \hat{\boldsymbol{\beta}}))$. In the following, for demonstrative purpose, we use logistic regression to estimate probabilities of events if an individual receives treatment or control. $p(\boldsymbol{x}, z; \hat{\boldsymbol{\beta}}) = \sigma(\boldsymbol{x}, z; \hat{\boldsymbol{\beta}}) = \frac{1}{1+exp^{-(\boldsymbol{x},z)'\hat{\boldsymbol{\beta}}}}$, where $\mathbb{V}(p(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}}))$ can be estimated by the following five methods:

1. Model-based method, where $\mathbb{V}(\boldsymbol{\beta}) = \mathbf{I}^{-1}(\boldsymbol{\beta})$, $\mathbf{I}(\boldsymbol{\beta})$ is the observed information matrix. $\mathbb{V}(\boldsymbol{\beta})$ can be estimated at $\hat{\boldsymbol{\beta}}$, denoted as $\hat{\mathbb{V}}(\hat{\boldsymbol{\beta}}) = \left(\mathbf{X}'\hat{\mathbf{V}}\mathbf{X}\right)^{-1}$, where

$$\hat{\mathbf{V}} = \begin{bmatrix} \hat{p}_1(1-\hat{p}_1) & 0 & \cdots & 0 \\ 0 & \hat{p}_2(1-\hat{p}_2) & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & \hat{p}_n(1-\hat{p}_n) \end{bmatrix},$$

   $\hat{p}_i$ is the predicted observed outcome, then

$$\hat{\mathbb{V}}(p(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}})) = \mathbf{x}_i^{'}\hat{\mathbb{V}}(\hat{\boldsymbol{\beta}})\mathbf{x}_i = \sum_j x_{ij}^2 \hat{\mathbb{V}}(\hat{\beta}_j) + 2\sum_{j=0}^{p}\sum_{k=j+1}^{p} x_{ij}x_{ik}\widehat{\mathrm{Cov}}(\hat{\beta}_j, \hat{\beta}_k). \tag{7}$$

   where we regard $Z_i = z$ as an element in the vector $\mathbf{x}_i$, i.e., $\mathbf{x}_i = (\boldsymbol{x}_i, z)'$, $\hat{\mathbb{V}}(\hat{\boldsymbol{\beta}}_j)$ is the jth diagonal element of the matrix $\hat{\mathbb{V}}(\hat{\boldsymbol{\beta}})$, and $\widehat{\mathrm{Cov}}(\hat{\boldsymbol{\beta}}_j, \hat{\boldsymbol{\beta}}_k)$ is an off-diagonal element in the matrix. Then we can estimate $\hat{\mathbb{V}}(p(\boldsymbol{x}, 1; \hat{\beta}))$ and $\hat{\mathbb{V}}(p(\boldsymbol{x}, 0; \hat{\beta}))$ for each individual i. We estimate the sample average of $\hat{\mathbb{V}}(p(\boldsymbol{x}, z; \hat{\beta}))$ as the estimate of expectation of the variance within groups, i.e., the first term in the last line of variance decomposition in equation 6. For the variance between groups, i.e., the second term in the equation, we estimate the sample variance of $f(\boldsymbol{X}; \bar{\hat{\boldsymbol{\beta}}})$ at $\hat{\boldsymbol{\beta}}$. For more computation details, see chapter 2.5 in (Hosmer Jr, Lemeshow, and Sturdivant 2013). Note that for continous outcome, linear regression assumes that the variance of error does not depend on the conditional mean. We can use heteroskedasticity-consistent standard errors in case the assumption does not hold. However, in logistic regression we have binomial errors, and as a result, the error variance is a function of the conditional mean thereof is heterogeneous by nature (Hosmer Jr *et al.* 2013).

2. Simulation approach, where $\hat{\beta} \sim \mathcal{N}(\hat{\boldsymbol{\beta}}, \hat{\mathbb{V}}(\hat{\boldsymbol{\beta}}))$, the method used by (Chatton *et al.* 2020; Aalen, Farewell, De Angelis, Day, and Nöel Gill 1997) which shows similar results to bootstrap resampling but is much faster. We can simulate a set of parametric models from the distribution of $\hat{\boldsymbol{\beta}}$, in which expectation and variance of are estimated using OLS, then the sample variance of predicted potential outcomes for each $\boldsymbol{x}_i$ from a set of simulated models is the estimated variance for $\mathbb{V}(p(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}}))$.

3. Bayesian approach. We use bayesian logistic regression to estimate potential outcomes. Via bayesian approach each parameter in a model is regarded as a random variable and follows a distribution. Posterior distribution of model parameters are approximated using sampling approach, e.g., MCMC, and hence the resulting predicted value of potential outcomes for each individual also follows a similar distribution and the variance of the distribution can be estimated using sample variance, namely, $\mathbb{V}(p(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}})) \approx \mathbb{S}(p(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}}))$, where $\hat{\boldsymbol{\beta}} \sim p(\hat{\boldsymbol{\beta}}; \mathcal{D})$.

4. Bootstrap. Instead of resampling model parameters in the second method, we can bootstrap sample from a dataset, estimate $\hat{\boldsymbol{\beta}}$ based on the resampled data, repeat multiple times, and compute the sample variance of predicted potential outcomes for each individual as the estimation of variance of $p(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}})$. This method, however, is of computational burden.

5. Sandwitch style standard error of estimator using R package **geex**. The standard error of average treatment effect can be computed directly by calling the function `geex::m_estimate(data, estFUN, ...)`. See Saul and Hudgens (2020) for more theoretical proof and implementation details.

Variance of average treatment effect is composed of variance within groups (the first term in the fourth line of equation 6) and variance between groups (the second term in the fourth line of equation 6). Via simulation approach, bayesian approach, and bootstrap approach, then the variance of $p(\boldsymbol{x}, z; \hat{\boldsymbol{\beta}})$ within a group $\boldsymbol{X} = \boldsymbol{x}$ can be computed as follows:

$$\hat{\mathbb{V}}(p(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}})) = \frac{1}{D} \sum_{d=1}^{D} \left(p(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}}^d) - \bar{p}(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}}) \right)^2 \tag{8}$$

where D is the number of draws from the distribution of $\hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\beta}}^d \sim \hat{p}(\hat{\boldsymbol{\beta}})$, $\bar{p}(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}}) = \frac{1}{D} \sum_{d=1} p(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}}^d)$, where $\hat{p}(\hat{\boldsymbol{\beta}})$ is the approximated empirical sampling distribution of $\hat{\boldsymbol{\beta}}$ using simulation, bayesian, and bootstrap based variance estimation approaches. Then

$$\mathbb{E}[\mathbb{V}\left(f(\boldsymbol{X}) \mid \boldsymbol{X}\right)] \approx \frac{1}{n} \sum_{i} \hat{\mathbb{V}}(p(\boldsymbol{x}_i, 1; \hat{\boldsymbol{\beta}})) + \hat{\mathbb{V}}(p(\boldsymbol{x}_i, 0; \hat{\boldsymbol{\beta}})) \tag{9}$$

by assuming $p(\boldsymbol{x}, 1; \hat{\boldsymbol{\beta}})$ is independent of $p(\boldsymbol{x}, 0; \hat{\boldsymbol{\beta}})$. Then we estimate sample average of $\hat{\mathbb{V}}(f(\boldsymbol{x}_i))$ as the estimate of expectation of variance of estimates of treatment effect within groups. The variance of estimates of treatment effect between groups (the second term in the last line of equation 6) can be estimated as follows:

$$\mathbb{V}\left(\mathbb{E}[f(\boldsymbol{X}) \mid \boldsymbol{X}]\right) \approx \frac{1}{n} \sum_{i=1} \left(p(\boldsymbol{x}_i, 1; \bar{\hat{\boldsymbol{\beta}}}) - p(\boldsymbol{x}_i, 0; \bar{\hat{\boldsymbol{\beta}}}) - \bar{p}(1; \bar{\hat{\boldsymbol{\beta}}}) - \bar{p}(0; \bar{\hat{\boldsymbol{\beta}}}) \right)^2 \tag{10}$$

where $p(\boldsymbol{x}_i, z; \bar{\hat{\boldsymbol{\beta}}}) = \frac{1}{D}\sum_{d=1} p(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}}^d)$ for simulation, bayesian, and bootstrap method, and $p(\boldsymbol{x}_i, z; \bar{\hat{\boldsymbol{\beta}}}) = p(\boldsymbol{x}_i, z; \hat{\boldsymbol{\beta}})$ for model-based method; $\bar{p}(z; \bar{\hat{\boldsymbol{\beta}}}) = \frac{1}{n}\sum_{i=1} p(\boldsymbol{x}_i, z; \bar{\hat{\beta}})$. Then the variance of estimate of average treatment effect in equation 6 for G-computation is sum of estimated variance of estimate of treatment effects within groups in equation 9 and estimated variance of estimate of treatment effects between groups in equation 10. Note that using sandwich style standard error via **geex** can directly estimate variance of estimate of average treatment effect without manually computing the equation 9 and 10. Hence, for computational convenience, we use sandwich style standard error throughout **RCTrep**.

B.2. Variance of IPW

Propensity-score based method for treatment effect estimation has methodological advantage since it mimics a set-up of a RCT in which the treatment and control groups are balanced. Propensity score is defined as:

$$\pi_z(\boldsymbol{X}) = P(Z = 1 \mid \boldsymbol{X}) \tag{11}$$

IPW weighs each individual by inverse probability of receiving the observed treatment. In a RCT, propensity score is known; in observational study, propensity score is unknown but may be estimable. IPW method is defined as follows where we use self-normalized IPW estimator (i.e., Hajek estimator) since it has smaller variance (Swaminathan and Joachims 2015):

$$\hat{\tau} = \sum_{i:Z_i=1} \hat{w}(\boldsymbol{x}_i) Y_i - \sum_{i:Z_i=0} \hat{w}(\boldsymbol{x}_i) Y_i \tag{12}$$

where

$$\hat{w}(\boldsymbol{x}_i) = \begin{cases} \frac{\frac{1}{\pi_z(\boldsymbol{x}_i;\hat{\boldsymbol{\alpha}})}}{\sum_{i:Z_i=1}\frac{1}{\pi_z(\boldsymbol{x}_i;\hat{\alpha})}} & Z_i = 1 \\ \frac{\frac{1}{1-\pi_z(\boldsymbol{x}_i;\hat{\boldsymbol{\alpha}})}}{\sum_{i:Z_i=0}\frac{1}{1-\pi_z(\boldsymbol{x}_i;\hat{\alpha})}} & Z_i = 0. \end{cases}$$

Different modeling approaches can be used to model propensity score, for instance, logistic regression, random forest, etc. IPW method is unbiased and consistent as long as propensity score model is correctly specified. The variance of IPW method is approximated as:

$$\begin{aligned} \mathbb{V}(f(\boldsymbol{X})) =& \mathbb{V}\left(\frac{YZ}{\pi_z(\boldsymbol{X};\hat{\boldsymbol{\alpha}})} - \frac{Y(1-Z)}{1-\pi_z(\boldsymbol{X};\hat{\boldsymbol{\alpha}})}\right) \\ =& \mathbb{E}\left[\mathbb{V}\left(\frac{YZ}{\pi_z(\boldsymbol{X};\hat{\boldsymbol{\alpha}})} - \frac{Y(1-Z)}{1-\pi_z(\boldsymbol{X};\hat{\boldsymbol{\alpha}})} \mid \boldsymbol{X}\right)\right] + \\ & \mathbb{V}\left(\mathbb{E}\left[\frac{YZ}{\pi_z(\boldsymbol{X};\hat{\boldsymbol{\alpha}})} - \frac{Y(1-Z)}{1-\pi_z(\boldsymbol{X};\hat{\boldsymbol{\alpha}})} \mid \boldsymbol{X}\right]\right) \\ \approx& \sum_{i:z_i=1}^{n} w_i^2 \hat{\sigma}_1^2(\boldsymbol{x}_i) + \sum_{i:z_i=0}^{n} w_i^2 \hat{\sigma}_0^2(\boldsymbol{x}_i) + \hat{\mathbb{V}}\left(f(\boldsymbol{X};\bar{\hat{\boldsymbol{\alpha}}})\right) \end{aligned} \tag{13}$$

where $\sigma_1^2(\boldsymbol{x})$ and $\sigma_0^2(\boldsymbol{x})$ is conditional variance of $Y(1)$ and $Y(0)$ given $\boldsymbol{x}$, which is unknown and maybe estimable using exact matching, and regression adjustment, etc., see Imbens and Rubin (2015) chapter 19 for details. $f(\boldsymbol{X};\bar{\hat{\alpha}}) \approx f(\boldsymbol{X}_i;\hat{\alpha}) = \frac{Y_i Z_i}{\pi_z(\boldsymbol{X}_i;\hat{\alpha})} - \frac{Y_i(1-Z_i)}{1-\pi_z(\boldsymbol{X}_i;\hat{\alpha})}$, $\hat{\mathbb{V}}(f(\boldsymbol{X};\bar{\hat{\boldsymbol{\alpha}}}))$ is the sample variance of $f(\boldsymbol{X};\bar{\hat{\boldsymbol{\alpha}}})$.

RCTrep uses sandwitch style standard error via **geex** to estimate the variance of IPW for average treatment effect estimation. It is clearly to see that the variance of IPW depends on the variance of estimated weights, and can inflate the variance if there are extreme values of weights. Hence, IPW method can suffer from near violation of Z-overlap assumption. To have good estimation of the variance, we should try to keep the dependence of $w(\boldsymbol{x}_i)$ as mild as possible. On one hand, we can reduce the variability of weight using approaches in Dong *et al.* (2020); Chattopadhyay *et al.* (2020); Zeng, Li, Wang, and Li (2021) through optimization, which minimizes the variability of all weights while preserving balance in weighted covariates between groups; on the other hand, to reduce variability of weights, we can exclude covariates which are merely associated with treatment assignment from propensity score modeling, since balancing over these variables will decrease sample size (degree of freedom) in each subgroup hence can inflate the estimation of variance. All variables beyond confounders which can cause the outcome can be adjusted in propensity score models which can improve precision.

B.3. Variance of DR

DR method combines a propensity score model with an outcome model such that the method is unbiased and consistent if at least one of the two models is correctly specified, hence it offers protection against mismodeling. DR method gains in precision of estimation over IPW method, however, may not be as precise as G-computation method when outcome model is correctly specified (or has good approximation) (Lunceford and Davidian 2004). The study by Kang and Schafer (2007) indicates that when both models are incorrect but neither is grossly misspecified, many DR methods perform better than IPW, however, non of the DR methods tried in the study improved upon the performance of an outcome regression model. Although the study does not represent all scenarios of DGM, the study does demonstrate that, in at least some settings, two wrong models may not better than one. The DR method for ATE estimation is demonstrated as follows:

$$\mathbb{E}[f(\boldsymbol{X})] = \frac{1}{n_o}\sum_i \left(p(\boldsymbol{x}_i, 1; \hat{\boldsymbol{\beta}}) + \frac{Z_i}{\pi_z(\boldsymbol{x}_i; \hat{\alpha})}\epsilon_i^1 \right) - \frac{1}{n_o}\sum_i \left(p(\boldsymbol{x}_i, 0; \hat{\boldsymbol{\beta}}) + \frac{(1-Z_i)}{1-\pi_z(\boldsymbol{x}_i; \hat{\alpha})}\epsilon_i^0 \right) \quad (14)$$

where $\epsilon_i^1 = Y_i - p(\boldsymbol{x}_i, 1; \hat{\boldsymbol{\beta}})$ and $\epsilon_i^0 = Y_i - p(\boldsymbol{x}_i, 0; \hat{\boldsymbol{\beta}})$. Variance of DR method is derived as follows:

$$\begin{aligned}
\mathbb{V}(f(\boldsymbol{X})) = {} & \mathbb{E}\left[\mathbb{V}\left(p(\boldsymbol{X}, 1; \hat{\boldsymbol{\beta}}) + \frac{Z}{\hat{\pi}(\boldsymbol{X})}\epsilon_i^1 - p(\boldsymbol{X}, 0; \hat{\boldsymbol{\beta}}) - \frac{1-Z}{1-\hat{\pi}(\boldsymbol{X})}\epsilon_i^0 \mid \boldsymbol{X} \right) \right] + \\
& \mathbb{V}\left(\mathbb{E}\left[p(\boldsymbol{X}, 1; \hat{\boldsymbol{\beta}}) + \frac{Z}{\hat{\pi}(\boldsymbol{X})}\epsilon_i^1 - p(\boldsymbol{X}, 0; \hat{\boldsymbol{\beta}}) + \frac{1-Z}{1-\hat{\pi}(\boldsymbol{X})}\epsilon_i^0 \mid \boldsymbol{X} \right] \right) \\
\approx {} & \frac{1}{n}\sum_i^n \hat{\mathbb{V}}\left(p(\boldsymbol{x}_i, 1; \hat{\boldsymbol{\beta}}) \right) + \hat{\mathbb{V}}\left(p(\boldsymbol{x}_i, 0; \hat{\boldsymbol{\beta}}) \right) + \\
& \frac{1}{n_1}\sum_{i:Z_i=1} w_i^2 \hat{\sigma}_1^2(\boldsymbol{x}_i) + \frac{1}{n_0}\sum_{i:Z_i=0} w_i^2 \hat{\sigma}_0^2(\boldsymbol{x}_i) + \hat{\mathbb{V}}\left[f(\boldsymbol{X}; \bar{\hat{\boldsymbol{\beta}}}, \bar{\hat{\boldsymbol{\alpha}}}) \right]
\end{aligned} \quad (15)$$

Similar to variance of IPW method and variance of G-computation method, $\hat{\mathbb{V}}(p(\boldsymbol{x}, z; \hat{\boldsymbol{\beta}}))$, can be estimated using model-based, simulation, bayesian, bootstrap method, and $\hat{\sigma}_1^2(\boldsymbol{x}_i)$ and $\hat{\sigma}_0^2(\boldsymbol{x}_i)$ can be estimated using exact matching, regression adjustment approaches. In **RCTrep**, we use sandwitch style method in **geex** to estimate the variance of DR method. The standard error of mean of $f(\boldsymbol{X})$ is $\frac{\hat{\mathbb{V}}(f(\boldsymbol{X}))}{n}$.

B.4. Variance of difference in means of outcomes between groups

In this section, we demonstrate variance of difference in means of outcomes between treatment and control groups. The variance is derived as follows:

$$\begin{aligned}\mathbb{V}(\hat{\tau}) &= \mathbb{V}\left(\frac{1}{n_1}\sum_{i:Z_i=1} Y_i(1) - \frac{1}{n_0}\sum_{i:Z_i=0} Y_i(0)\right) \\ &= \frac{1}{n_1^2}\sum_{i:Z_i=1}\sigma_1^2(\boldsymbol{x}_i) + \frac{1}{n_0^2}\sum_{i:Z_i=1}\sigma_0^2(\boldsymbol{x}_i) \\ &\approx \frac{\hat{\sigma}_1^2}{n_1} + \frac{\hat{\sigma}_0^2}{n_0}\end{aligned} \tag{16}$$

Under simplifying assumption of homoscedasticity, i.e., $\sigma_1^2(\boldsymbol{x}) = \sigma_1^2$ and $\sigma_0^2(\boldsymbol{x}) = \sigma_0^2$ are constants across individuals, σ_1^2 and σ_0^2 can be estimated by sample variance of $Y(1)$ in the treatment group and sample variance of $Y(0)$ in the control group. We also assume observed outcomes Y_i are mutually independent, namely, observed outcome of each individual does not depend on the observed outcome of another individual. Since $\mathbb{V}(Y \mid \boldsymbol{x}) = \mathbb{V}(Y)(1-\rho)$ where ρ is the correlation between Y and $\boldsymbol{X}$, the estimated variance of average treatment effect in equation 16 is *conservative*, and can gain efficiency if conditioning on variables $\boldsymbol{X}$ that is predictive to outcomes. $\hat{\mathbb{V}}(\hat{\tau})$ is the standard error of the average treatment effect.

C. Methods for adjusting the sampling mechanism

In this section, we elaborate three approaches used in **RCTrep** to adjusting the sampling mechanism. 1) exact matching; 2) inverse sampling score weighting; 3) subclassification. We also give a brief introduction of optimization based method for weight estimation. Throughout **RCTrep**, we use **geex** to estimate the variance for weighed average treatment effect estimation.

C.1. Exact matching

In this section, we introduce weighting based on $\boldsymbol{X}$. This weighting approach is similar to importance sampling/transfer learning/domain adaption/covariate shift, which balances the distribution of $\boldsymbol{X}$ between two samples. We can also match units according to $\boldsymbol{X}$, for more details, see (Stuart 2010). Given assumptions on the sampling mechanism, RWD and experimental data can be regarded as two random samples from the same population, and hence the weighed average treatment effect of RWD can be similar to the treatment effect of experimental data. The weighted average treatment effect of RWD is defined as:

$$\hat{\tau} = \hat{\mathbb{E}}[w(\boldsymbol{X})f(\boldsymbol{X})] = \sum_{\boldsymbol{x}} w(\boldsymbol{x})f(\boldsymbol{x}), \quad s.t. \;\; p(\boldsymbol{x}) = w(\boldsymbol{x})q(\boldsymbol{x}) \tag{17}$$

where $w(\boldsymbol{X}) = \frac{p(\boldsymbol{X})}{q(\boldsymbol{X})}$, $p(\boldsymbol{x})$ and $q(\boldsymbol{x})$ are emprical density of $\boldsymbol{X}$ of experimental data and RWD, $f(\boldsymbol{X})$ heterogeneous treatment effect using G-computation, IPW, and doubly robust methods with additional variable $w(\boldsymbol{X})$. We can treat $w(\boldsymbol{X})$ as a fixed value for each individual, and use a standard Horvitz-Thompson-type sandwich variance estimator with the resulting weights via R packages **geex** or **survey**. However, it is important to consider that these weights are estimated and are unknown. Buchanan *et al.* (2018) derived a variance estimator that accounts for when the weights are unknown. We can also use double bootstrap to estimate variance used in (Ackerman *et al.* 2021), where both experimental data and RWD are resampled with replacement prior. This approach, however, is to computationally intensive, and results are very similar to standard sandwich variance estimator.

C.2. Inverse sampling score weighting

The sampling score is the conditional probability of being selected to the experimental data given a vector of observed covariates $\boldsymbol{X}$, which is defined as follows:

$$\pi_s(\boldsymbol{X}_i) = P(S = 1 \mid \boldsymbol{X}_i) \tag{18}$$

where $S = \{0, 1\}$, 1 indicates selection to experimental data and 0 indicates selection to RWD. In most of cases the sampling score is unknown, but maybe estimated from combined data. In **RCTrep**, we denote experimental data as the target population, we weight individuals in RWD according to odds of their sampling scores to resemble the target population. Hence the weights for each individual are:

$$w_i = \begin{cases} \frac{\pi_s(\boldsymbol{x}_i)}{1-\pi_s(\boldsymbol{x}_i)} & S_i = 0 \\ 1 & S_i = 1 \end{cases}$$

According to (Rosenbaum and Rubin 1983), ignorability assumption holds conditioning on a balance score. Sampling score is the "coarsest" balance score, $\boldsymbol{X}$ is the "finest" balance

score. Any balancing score finer than sampling score can allow assumptions ignorability hold. Sampling score is a propensity score when we aim to correct for confounding due to non-random sampling mechanism.

C.3. Subclassification

Individuals are assigned to a subclass according to a distance measure, for instance, sampling score $p(S = 1 \mid \boldsymbol{X})$. Many modeling approaches are provided in **RCTrep** for estimating sampling score, for instance, `glm`, `gbm`, `lasso`. RWD and experimental data are placed into subclasses based on quantiles of the sampling scores in experimental data. Weights are computed based on the proportion of experimental data in each subclass. For more details, see R package **MatchIt**.

C.4. Optimization approach

In this section, we give a brief introduction of optimization-based weight computation. Note that these approaches are not implemented in **RCTrep**, however, can be easily extended to the package. Beyond a sampling score as a balancing score, balancing score $\phi(\boldsymbol{X})$ can be a vector of basis functions of $\boldsymbol{X}$, e.g., the moments, the interaction, the non-linear transformations of components of $\boldsymbol{X}$, for instance, $\phi(\boldsymbol{X}) = \left(X_1^2, X_1X_2, \sqrt{X_3}\right)$. Optimization based methods design an objective function of weights (for instance, variability of weights) and balancing constraints (for instance, sum of weights equal to 1, weights should be non-negative, distance between two samples), then the weights are estimated as the solution to the objective function. For instance, entropy-based estimation (Dong *et al.* 2020), stable balancing weights (Chattopadhyay *et al.* 2020). These methods are mainly designed to reduce the variance of weights. For more methods, see (Kouw and Loog 2018; Sugiyama, Suzuki, and Kanamori 2012). In general, optimization-based approaches to weight computation can be formulated into the following optimization problem:

$$\begin{array}{ll} \underset{w}{\text{minimize}} & \text{measurement of variability of weights} \\ \text{subject to} & \text{balancing contraints,} \\ & \sum_{i:S_i=0} w_i = 1, w_i \geqslant 0, \quad D(w_i\phi(\boldsymbol{X}_i), \phi(\boldsymbol{X}_j)) < \delta \end{array} \tag{19}$$

where D is a distance measure between weighted distribution of $\phi(\boldsymbol{X})$ in RWD and distribution of $\phi(\boldsymbol{X})$ in experimental data, δ is a balance tolerance that defines a tolerance or maximum difference in means after weighting for covariates, i denotes an individual in RWD and j denotes an individual in experimental data. According to Stuart (2010), matching/weighting methods aim to balance covariates between samples/groups so that covariates are the same distributed. Distance measure for exact matching/weighting is

$$D_{ij} = \begin{cases} 0 & if \boldsymbol{X}_i = \boldsymbol{X}_j \\ \infty & if \boldsymbol{X}_i \neq \boldsymbol{X}_j \end{cases}$$

Multiple choices of distance measure can be used, see (Stuart 2010). Then the weight with minimal dispersion under balance constraints is the solution to this optimization problem. For instance, Chattopadhyay *et al.* (2020) computes the weight based on the following opti-

mization problem using R package **sbw**:

$$\begin{array}{ll} \underset{w}{\text{minimize}} & \sum_{i:S_i=0} \left(w_i - \bar{w}_0\right)^2 \\ \text{subject to} & \left| \sum_{i:S_i=0} w_i \phi_k \left(\boldsymbol{X}_i\right) - \frac{1}{n_{s1}} \sum_{i:S_i=1} \phi_k \left(\boldsymbol{X}_i\right) \right| \leqslant \delta_k, \quad k = 1, 2, \ldots, K-2 \\ & \sum_{i:Z_i=0} w_i = 1 \\ & w_i \geqslant 0, \quad i : S_i = 0 \end{array} \tag{20}$$

where $\boldsymbol{\phi}(\boldsymbol{X}) = (\phi_1(\boldsymbol{X}), ..., \phi_k(\boldsymbol{X}))$, $\phi_1(\boldsymbol{X}) = \boldsymbol{X}_{i1}$, $\phi_2(\boldsymbol{X}) = \boldsymbol{X}_{i2}$, $\phi_3(\boldsymbol{X}) = \boldsymbol{X}_{i3}$, $\phi_4(\boldsymbol{X}) = \boldsymbol{X}_{i1} \times \boldsymbol{X}_{i2}$, then $\frac{1}{n_{s1}} \sum_{i=1} \phi_1(\boldsymbol{X}_i)$ is the marginal distribution of $\boldsymbol{X}_1$ in $S = 1$ (experimental data). To preserve privacy, users can publish mean of basis functions of experimental data, and the weights of RWD can be computed via optimization and constraints by the published mean of basis functions of the experimental data. Weights can be computed using multiple solvers, for instance, Gurobi[23], CPLEX[24], MOSEK[25], POGS[26], QUADPROG[27]. CPLEX, Gurobi and MOSEK are commercial solvers, but free for academic users; POGS and QUADPROG are free for all. According to Chattopadhyay *et al.* (2020), POGS is the fastest solver option and able to handle larger datasets, but it can be difficult to install for non-Mac users. MOSEK is more stable than POGS.

[23]https://www.gurobi.com/documentation/9.5/quickstart$_w$*indows*/$r_i ns_t he_{rp}$*ackage.html*
[24]https://www.ibm.com/analytics/cplex-optimizer
[25]https://www.mosek.com/
[26]http://foges.github.io/pogs/stp/r
[27]https://cran.r-project.org/web/packages/quadprog/index.html

D. Analysis of omitted variable bias in G-computation

In this section, we elaborate the effect of omitted variable (namely, unmeasured confounder) on the strength and direction of the bias of estimate of treatment effect. Let β_z denote effect of treatment on outcome, β_{x_u} denote the effect of unobserved confounder on outcome, ρ_{zx_u} denote the pairwise correlation between treatment and unobserved confounder, y is the outcome of interest, z is the indicator of treatment, 1 denotes receiving treatment and 0 denotes receiving control only. Suppose that the "true" model is

$$y = \beta_0 + \beta_z z + \beta_{x_u} x_u + \epsilon \tag{21}$$

Suppose, however, due to our ignorance or data unavailability, we mistakenly model:

$$y = \tilde{\beta}_0 + \tilde{\beta}_z z + \tilde{\epsilon} \tag{22}$$

Note that

$$\begin{aligned}
\tilde{\beta}_1 &= \frac{cov(z, y)}{\sigma_z^2} \\
&= \frac{cov(z, \beta_0 + \beta_z x_1 + \beta_{x_u} x_u + \epsilon)}{\sigma_z^2} \\
&= \frac{cov(z, \beta_0) + cov(z, \beta_z z) + cov(z, \beta_{x_u} x_u) + cov(z, \epsilon)}{\sigma_z^2} \\
&= \frac{0 + \beta_z \sigma_z^2 + \beta_{x_u} cov(z, x_u) + 0}{\sigma_z^2} \\
&= \beta_z + \beta_{x_u} \frac{\rho_{zx_u} \sigma_z}{\sigma_z} \\
&= \beta_z + \beta_{x_u} \frac{\rho_{zx_u} \sigma_z}{\sigma_z}
\end{aligned} \tag{23}$$

Hence, omitted bias occurs in case:

1. Z and X_u is correlated
2. $\beta_{x_u} \neq 0$

In RCT, since Z is random assigned, so $\text{Corr}(X_u, Z)$ is 0 (Z is independent of all measured and unmeasured variables, $\rho_{zx_u} = 0$), hence OLS estimator of β_z is consistent and unbiased. The direction of bias depends on the sign of ρ_{zx_u} and the sign of β_u, which can provide insight on over/underestimation of average treatment effect using RWD compared to unbiased estimate of "truth" from experimental data. The effect of the sign of correlation $\text{Corr}(X_u, Z)$ and the sign of β_{x_u} on estimation can be formulated into four cases listed in table 4. The table 4 implies in which cases we can over/under estimate average treatment effect, and to what extent we may draw wrong conclusions of causal relation between treatment and outcome, for instance, if treatment assignment is positively correlated with an unmeasured confounder that has positive effect on outcomes, then omitting the variable can lead to overestimation since there are more individuals with the positive value of the confounder in treatment group than control group and treatment group shows better outcomes due to the confounder has positive effect on the outcome. In case there is no treatment effect of Z on Y, without adjusting the confounder, we can observe a spurious causal relation between Z and Y since we observe difference in mean of Y between groups, however, the difference is not because of the treatment.

Table 4: Direction of bias of estimation of average treatment effect β_z due to unmeasured confounder(s) X_u

	$\text{Corr}(X_u, Z) > 0$	$\text{Corr}(X_u, Z) < 0$
$\beta_{x_u} > 0$	positive bias (overestimate)	negative bias (underestimate)
$\beta_{x_u} > 0$	negative bias (underestimate)	positive bias(overestimate)

E. Conditions to allow for the fair comparison

In this section, we illustrate in which conditions the expectation of estimates of average treatment effect from RWD and the expectation of estimates of average treatment effect from experimental data are identical. We regard the population that experimental data represents for as the target population. The experimental data is randomly drawn from the target population and RWD is drawn according to a (known/unknown) sampling mechanism $\pi_s(\boldsymbol{X})$. The expectation of unbiased estimates of average treatment effect using experimental data is the "truth" of estimates using RWD if weighted RWD is a random sample from the target population. We estimate the average treatment effect using RWD under the condition that the sampling mechanism of RWD is correctly/incorrectly adjusted and the treatment assignment mechasnim of RWD is correctly/incorrectly adjusted; we numerate all combinations of outcomes of the sampling mechanism correction and outcomes of the treatment assignment mechanism correction, and elaborate if the expectation of estimates of treatment effect using RWD and experimental data are identical under each combination.

We use unbiased and consistent estimators for average treatment effect, and different estimators have different properties of efficiency. If the treatment assignment mechanism is not properly adjusted, then an unobserved confounder X_z between the treatment and the outcome occurs. If the sampling mechanism is not properly adjusted, an unobserved confounder X_s between the sampling and the outcome occurs. We further classify X_s into *prognostic*[28] and *predictive*[29] variable according to if X_s is an effect modifier, and discuss if the expectation of estimates of average treatment effect using RWD and the expectation of estimates using experimental data are still identical in the presence/absence of X_z and X_s respectively. We summarize the results in Table 5 for each case. In the following, we elaborate each condition (a)-(f) using a graph for demonstrative purpose.

E.1. Can we obtain the identical expectation of estimates using RWD under assumptions in (a)?

For the case **(a)** which is visualized in subfigure (a) in Fig 13, since RWD is selected from a well-defined target population via a known sampling score, by properly adjusting the sampling mechanism of RWD, the weighted RWD is representative to the target population and hence is comparable with experimental data. Since the treatment assignment mechanism of RWD is properly adjusted, there is no unmeasured confounders between treatment and outcome, and hence the estimates of treatment effect using RWD is unbiased. Given the sampling mechanism and the treatment assignment mechanism are all properly corrected, expectation of estimates of average treatment effect from RWD is identical to that from experimental data.

[28]Prognostic variable is a variable that is predictive to the outcome but has no interaction effect with the treatment. The variable informs about a likely outcome independent of treatment received. Prognostic variable is not in the function of $\tau(\boldsymbol{X})$

[29]Predictive variable is a variable that is predictive to the outcome and has interaction effect with the treatment, meaning it is also predictive to treatment effect. The variable is predictive if the treatment effect shows heterogeneity on the level of this variable. Predictive variable is in the function $\tau(\boldsymbol{X})$

[30]In this case, if we assume all confounders are properly controlled in RWD, although the unbiased estimate of "truth" of treatment effect is not exchangeable between RWD and experimental data hence the evaluation of methods for treatment effect is impossible, we can attribute the difference in estimates to unobserved heterogeneity on unobserved predictive variable which is not confounder between RWD and experimental data.

	Generation of RWD					Identical
	$\pi_s(\boldsymbol{X})$	$\pi_z(\boldsymbol{X})$	confounders			
			X_z	X_s		
				pg	pd	
(a)	✓	✓				✓
(b)	✓	✘	✓			✘
(c)	✘	✓		✓		✓
(d)	✘	✓			✓	✘ [30]
(e)	✘	✘	✓	✓		✘
(f)	✘	✘	✓		✓	?

Table 5: Overview of conditions under which the expectation of estimate of treatment effect from RWD and that from experimental data are identical. We regard the population that experimental data represents as the target population. RWD is sampled from the target population according to a (known/unknown) sampling score $\pi_s(\boldsymbol{x})$. $\pi_z(\boldsymbol{x})$ denotes the treatment assignment mechanism of RWD. ✓and ✘ for $\pi_s(\boldsymbol{x})$ and $\pi_z(\boldsymbol{x})$ denote whether $\pi_s(\boldsymbol{x})$ and $\pi_z(\boldsymbol{x})$ of RWD are properly corrected. $\boldsymbol{X}_z$ denotes a confounder between treatment and outcomes, $\boldsymbol{X}_s$ denotes a confounder between sampling and outcomes. ✓for $\boldsymbol{X}_s$ and $\boldsymbol{X}_z$ denotes presence of a confounder. "pg" is short for prognostic variable, "pd" is short for predictive variable. "Identical" denotes if the expectation of the estimate of treatment effect from RWD is identical to the expectation of the estimate from experimental data, ✓denotes identical, ✘ denotes not identical, "?" denotes unknown.

Since different methods for average treatment effect estimation show different efficiency, for instance, the effect of increasing the sample size and the effect of increasing the dimension of covariates on estimates precision varies across methods, hence under the condition (a), we can evaluate and select the best methods for treatment effect estimation using a given RWD.

E.2. Can we obtain the identical expectation of estimates using RWD under assumptions in (b)?

For the case **(b)** which is visualized in subfigure (b) in Fig 13, due to the sampling mechanism is properly adjusted, the expectation of the unbiased estimate of average treatment effect using RWD is identical to that from experimental data; due to the the treatment assignment mechanism of RWD is not properly adjusted due to unmeasured confounders X_z, the expectation of the estimate of average treatment effect is biased. Hence, in case (b), the expectation of estimates using RWD and that using experimental data is not identical, and the difference is due to an unmeasurd confounder X_z, methods for treatment effect estimation are not valid. Note that if the sampling mechanism of RWD is estimated and is assumed to be correct, then the confounder X_z should be balanced between samples.

E.3. Can we obtain the identical expectation of estimates using RWD under assumptions in (c) and (d)?

For the case **(c)** and **(d)** which is visualized in subfigure (c) in Fig 13, if the treatment assignment mechanism of RWD is properly corrected, even though the sampling mechanism of RWD is not properly corrected, the expectation of estimates of average treatment effect

using RWD and that using experimental data may still be identical, depending on whether X_s is predictive or prognostic variable:

1. if X_s is a prognostic variable which is not predictive to treatment effect, the expectation of estimates of average treatment effect from experimental data and that from RWD is identical;

2. if X_s is a predictive variable which is predictive to the treatment effect, without balancing the variable X_s, the expectation of estimates of average treatment effect from RWD and that from experimental data is not identical. The difference in estimates between RWD and experimental data is due to unobserved imbalanced on the X_s, and may provide insight on the possible existence of a predictive variable that can improve precision of estimates. However, the proposal of the assumption on the existence of a predictive variable depends on that the treatment assignment mechanism is properly corrected, and methods for treatment effect estimation have good statistical power. In table 6, we provide an overview of the effect of adjusting X_s on the bias and variance of estimates of average treatment effect of the target population that experimental data represents using RWD. The table implies that adjusting X_s which is highly associated with the sampling (meaning X_s is highly imbalanced between experimental data and RWD) and is highly predictive to the treatment effect $\tau(\boldsymbol{X})$ can significantly reduce the bias and variance, and adjusting X_s that is highly predictive to $\tau(\boldsymbol{X})$ can always reduce the variance regardless of its association with the sampling. In particular, in case weighting X_s can significantly reduce the bias (i.e., X_s is highly associated with $\tau(\boldsymbol{X})$ and X_s is highly associated with the sampling S), we further provide an overview of the direction of bias, and its association with β_{X_s} and $\text{Corr}(X_s, S)$ in table 7, where β_{X_s} is the effect of X_s on $\tau(X)$, $\text{Corr}(X_s, S)$ is the association between X_s and sampling indicator S, $S = 1$ indicates selection into the experimental data.

In particular, if X_s is predictive to treatment, then it must be balanced between treatment and control groups in RWD by either randomization or statistical methods adjustment.

Table 6: Effect of weighting adjustment on bias and variance of estimates of average treatment effect of the target population that experimental data represents using RWD, by strength of association of the adjustment variable X_s with selection indicator S and $\tau(\boldsymbol{X})$ (adapted from Table 1 in Little and Vartivarian (2005)).

	low association $(X_s, \tau(\boldsymbol{X}))$	high association $(X_s, \tau(\boldsymbol{X}))$
low association (X_s, S)	little effect on bias little effect on variance	little effect on bias variance reduction
high association (X_s, S)	little effect on bias variance inflation	bias reduction variance reduction

Table 7: Direction of bias of estimation of average treatment effect of the target population that experimental data represents using RWD. Bias is due to X_s, where β_{X_s} is the effect of X_s on $\tau(X)$, $\text{Corr}(X_s, S)$ is the association between X_s and selection indicator S, $S = 1$ indicates selection into experimental data

	$\text{Corr}(X_s, S)>0$	$\text{Corr}(X_s, S) < 0$
$\beta_{x_s} > 0$	overestimation	underestimation
$\beta_{x_s} < 0$	underestimation	overestimation

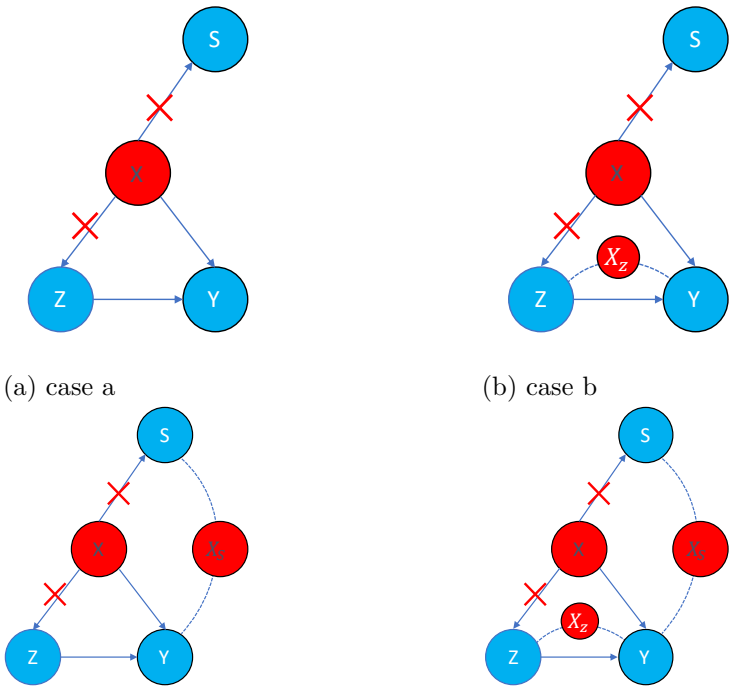
E.4. Can we obtain the identical expectation of estimates using RWD under assumptions in (e) and (f)?

For cases (e) and (f) which are visualized in subfigure (d) in Fig 13, we classify X_s into prognostic variable and predictive variable, and elaborate accordingly:

1. if X_s is prognostic variable as indicated in case (e), without balancing X_s, the expectation of unbiased estimates of average treatment effect using RWD and that using experimental data is still identical. Since the treatment assignment mechanism of RWD is not properly adjusted due to unmeasured confounder X_z, without adjustment X_z, the estimate using RWD is biased, and hence the expectation of estimates of average treatment effect using RWD and the expectation of estimates using experimental data is not identical. The difference in expectation of estimates of average treatment effect between RWD and experimental data (i.e., bias) is due to X_z in RWD, and its direction and magnitude depends on the effect of X_z on outcome and the strength of correlation between X_z and the treatment.

2. In case (f), there are two sources that lead to difference in expectation of estimates of average treatment effect between RWD and experimental data. On one hand, since the sampling mechanism is not properly corrected and the confounder X_s is a predictive variable implied in the case (f), without balancing X_s, the expectation of unbiased estimates of average treatment effect using RWD and that using experimental data is not identical. The difference in expectation of unbiased estimates beween RWD and experimental data depends on the strength of the correlation between the sampling S and X_s and the effect of X_s on $\tau(\boldsymbol{X})$. On the other hand, since the treatment assignment mechanism of RWD is not properly corrected due to an unmeasured confounder X_z, the

estimate using RWD is biased, and the direction and magnitude of bias depends on the correlation between X_z and the treatment and the effect of X_z on the outcome. When combining two sources of bias, the bias due to X_s and the bias due to X_z may happen to cancel out, and hence it is unknown if the expectation of estimate of average treatment effect using RWD and the estimate using experimental data is identical, and the direction and magnitude of combined bias depends on a) the correlation of X_s and S and the effect of X_s on $\tau(\boldsymbol{X})$; 2) the correlation between X_z and Z and the effect of X_z on the outcome. In particular, X_s and X_z may happen to be the same variable or two different variables respectively.



(a) case a (b) case b

(c) case c-d (d) case e-f

Figure 13: graphs to demonstrate confounders in between Z and Y, and S and Y

F. Overview of the package

The package's `TEstimator` and `SEstimator` are the core R6 classes that implements estimation of average treatment effect of a sample by correcting for treatment assignment mechanism and correcting for sampling mechanism. Figure 14 and figure 15 presents an overview of class `TEstimator` and `SEstimator`.

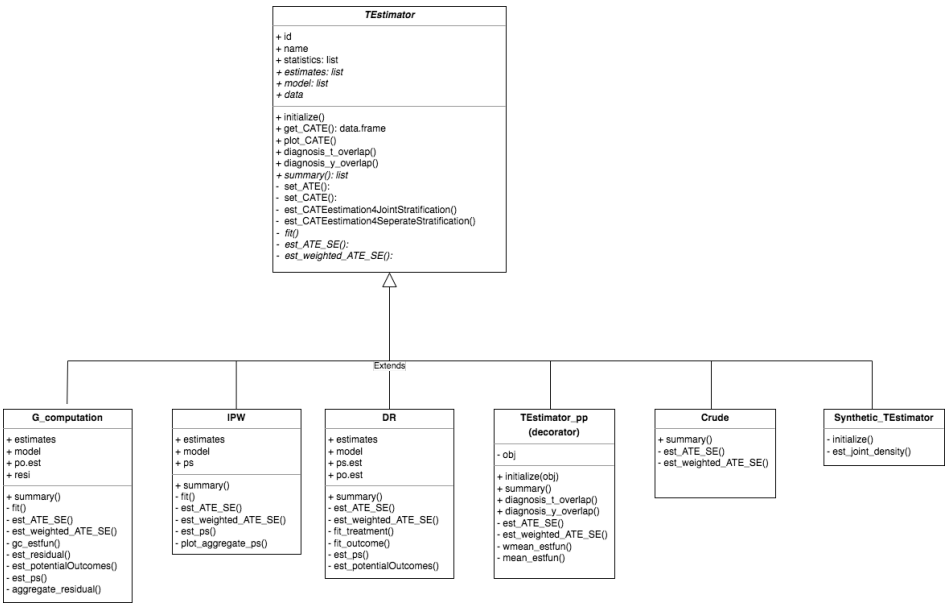


Figure 14: Overview of class TEstimator. The functions of the class `TEstimator` and fields form the main solutions for correcting for the treatment assignment mechanism.

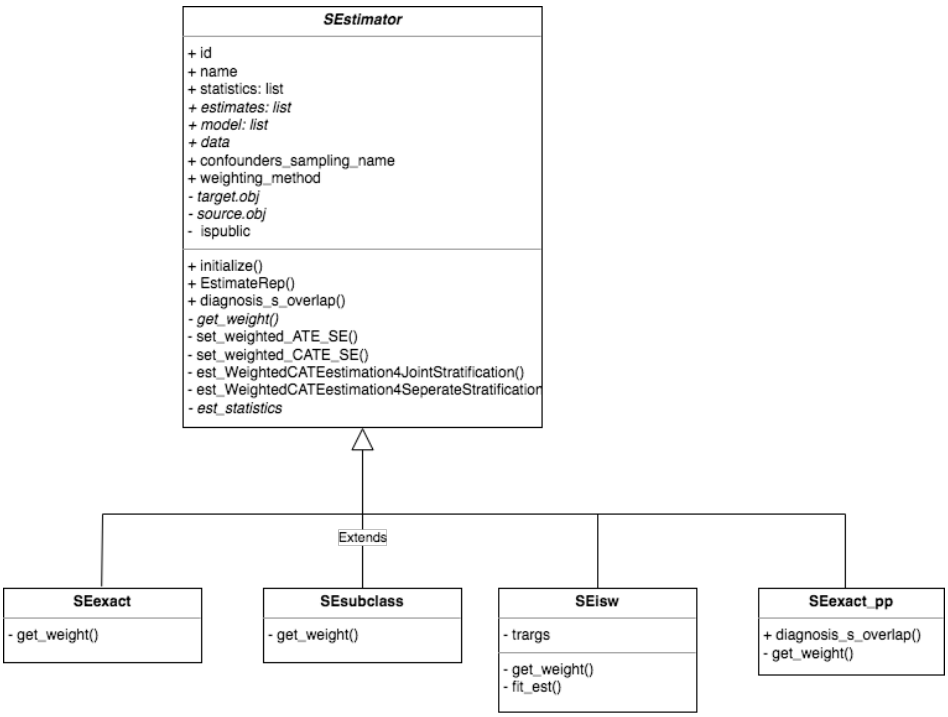


Figure 15: Overview of class SEstimator. The functions of the class `SEstimator` and fields form the main solutions for correcting for the sampling mechanism.

G. Some comment regarding assumptions of ignorability

Similar to definitions of confounders in identifying causal relation between treatment and outcomes, variables that can cause outcomes and sampling mechanism which can lead to imbalance of covariates between samples, are also called confounders between sampling and outcomes. The confounders can confound relation between sampling and outcomes, meaning there is systematic difference between two samples hence two samples are not comparable. Without properly controlling for the confounder(s), we may observe difference in estimates between outcomes. The magnitude of the difference depends on the magnitude of effect of the confounder(s) on outcomes and the strength of correlation between sampling and the confounder(s). For instance, a variable is highly predictive to outcomes and highly associated with the sampling mechanism (the distribution of the variable in one sample is very different from that in another sample). Without controlling the variable, we can observe huge difference in outcomes between two samples, leading to a spurious correlation between outcomes and sampling. Similar to random confounders induced by imbalance by chance in RCT, even in the case which we sample according to known sampling score, it is still likely to happen to have unobserved confounders that cause outcomes which however are not balanced between selected and non-selected samples, leading to systematic difference in outcomes between selected and non-selected samples (two samples have heterogeneity due to confounding). It is worth noting that if S-overlap assumptions are violated, two samples are heterogeneous on level, and they can be regarded as randomly drawn from two distinct populations with possibly different DGMs.

Unfortunately, the assumption of ignorability on sampling mechanism can rarely hold in non-random sampling since it requires us to identify all causes of outcomes, which is almost impossible. Therefore, with limited knowledge about the causal mechanism, (unobserved) heterogeneity of any two samples drawn by non-random sampling mechanism will always (at least almost) exist since we never (or at least rarely) be able to find the true model of outcome [31]. Regarding sets of covariates $\boldsymbol{X}$ to allow for treatment effect estimates comparable, some studies show that not all variables that cause sampling mechanism needs to be corrected, for instance, if the estimand of interest is difference in mean for continuous outcome, then adjusting for effect modifier is sufficient and efficient for the estimate comparison (Egami and Hartman 2018; Dahabreh *et al.* 2020). We have elaborate the conditions under which we can evaluate methods for treatment effect estimation in appendix E.

[31] Although we can't find the true model, however, we can find a good (unbiased and consistent) model by checking there is no omitted variable bias by looking at assumption of error term, i.e., $\mathbb{E}[\epsilon \mid \boldsymbol{x}_i] = 0$, and we can check the existence of other causes of outcomes independent of other observed variables by validating if $\sigma^2_\epsilon(\boldsymbol{x})$ is a constant or not. However, for binary outcome modeling by logistic regression, omitted non-confounding predictors can lead to heterogeneity in outcomes, and can bias the estimates of parameters

Affiliation:

Lingjie Shen

Department of Methodology and Statistics
Tilburg University
E-mail: L.Shen@uvt.nl

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