Hybrid_Est.R

Documentation for Running the R Program

January 2026

Peter Z. Schochet (pzschochet@gmail.com)

Under a "hybrid" multisite impact evaluation design, randomized controlled trials (RCTs) are conducted in some study sites, where feasible, and quasi-experimental designs (QEDs) are conducted in more sites, including the RCT sites (referred to as RCT-QED sites), to increase precision and generalizability. Estimation methods for hybrid designs are discussed in Schochet (*Reference*, 2026), which allows for hidden bias in the QED impact estimates after applying inverse probability weighting (IPW) to construct the comparison groups. A key feature of the hybrid design is that hidden QED biases can be corrected under the assumption that the QED biases estimated in the RCT-QED sites as a function of observed baseline covariates are transportable to the QED-only sites.

Hybrid_Est.R contains four sequential R functions for estimating impacts for hybrid designs:

1. **bias_tree()** estimates the bias functions—differences between the RCT and QED estimates in the RCT-QED sites—as a function of baseline covariates using a classification and regression tree (CART) supervised machine learning algorithm. The function returns a data frame that summarizes the initial (unpruned) tree and each nested subtree obtained using weakest link pruning (see Schochet, 2026). The data frame includes, for each tree node, the selected covariate and splitting value, the predicted (mean) bias, and sample sizes. A table is also produced that indicates (using a *) the subtree with the greatest predictive accuracy based on a holdout sample. The function can be run by specifying either the intention-to-treat (ITT) estimand or complier average causal effect (CACE) estimand, depending on the nature of the RCT and QED samples (see Schochet, 2026).
2. **view_selected_tree()** provides detailed tables and a tree plot for the CART subtree selected by users for subsequent analyses after reviewing the output from **bias_tree()**. The function requires inputs on the selected subtree number and the assigned data frame returned by **bias_tree()**.
3. **calc_pred_bias()** merges the predicted biases from the selected CART subtree onto the QED or RCT input data frame for each sample member. Each person in the input file is filtered through the selected subtree based on the person's covariate values until a terminal tree leaf is reached; the predicted bias is the mean bias in that leaf, as estimated in **bias_tree()**. These biases are needed to run the impact estimation function below to adjust for hidden QED biases in the QED-only sites.
4. **hybrid_impacts()** estimates pooled impacts by averaging the RCT estimates in the RCT-QED sites and the bias-corrected IPW estimates in the QED-only sites, with variances that incorporate estimation error in the bias corrections. The impacts are estimated for commonly used designs—including nonclustered and clustered designs—and the models allow for baseline covariates to improve precision. The function conducts hypothesis tests and produces output tables displaying detailed results for the pooled impacts and their RCT and QED components, overall and by site.

DATA REQUIREMENTS

Hybrid_Est.R requires, as inputs, two separate data frames with individual-level data containing:

i. RCT samples from the RCT-QED sites
ii. QED samples from *both* the RCT-QED and QED-only sites, with a 1/0 indicator signifying the type of site (see the next section).

The RCT and QED input data frames must both have the same rectangular format and contain the same column (variable) names for overlapping variables. ***Missing data are not allowed for all key variable inputs listed below, and error messages will result if missing data exist***. For instance, the package requires nonmissing values for the site IDs, treatment status indicator variable, outcome variable, IPW weights (for the QED samples), and covariates needed to estimate the bias functions using CART. Thus, users should impute missing data or remove them prior to running the functions.

Note that the package does not include a function that computes IPW weights for the QED analysis, which users will need to construct prior to running the package. Schochet (2026) provides existing R packages that can be used to estimate the IPW weights using CART or other methods.

FUNCTION INPUTS

Inputs for ***Hybrid_Est.R*** are entered directly into the function parentheses, separated by commas, when the functions are called. These inputs are discussed next for each of the four functions.

1. Inputs for the bias_tree() function

To run **bias_tree()**, users must ***assign the result of the function to a new variable,*** which will be a data frame containing summary CART tree information needed to run subsequent functions. For example, if the new data frame is to be named, "**best_trees_df**", it can be assigned as follows:

best_trees_df <- **bias_tree()**

The required and optional function inputs for **bias_tree()** are as follows:

Variable		**Example input**	**Description**
Required Inputs			
rct_dat_df	=	rct_df	RCT data frame containing nonmissing data for each individual in the RCT samples from the RCT-QED sites.
qed_dat_df	=	qed_df	QED data frame containing nonmissing data for each individual in the QED samples from *both* the RCT-QED and QED-only sites.
rct_qed_site	=	rq_site	Indicator variable: 1 = individual is in an RCT-QED site, 0 = individual is in a QED-only site. *Required only for the QED input data frame, but not for the RCT input data frame.*
yvar	=	y	Outcome variable. The function must be run separately for each analysis outcome variable.
xvars_cart	=	x1+x2+x3	Baseline covariates for CART estimation, separated by a "+" sign. *The function will run much faster if continuous covariates are coded into categorical variables.*
site_id	=	site	Unique site ID
t_c	=	trtv	Treatment status indicator: 1 = individual in treatment group, 0 = individual in RCT control or QED comparison group

Variable	Example input	Description
cace_itt_est =	1	1 = CACE analysis, 0 = ITT analysis (see Schochet, 2026)
got_treat =	d	Treatment participation indicator: 1 = RCT member received treatment services, 0 = RCT member did not receive treatment services. *Required only for the RCT input data frame for CACE analyses (i.e., when **cace_itt_est** = 1), but not otherwise*.
ipw_wgt =	ipw	Variable containing the IPW weights. *Required for the QED input data frame, but not for the RCT input data frame*.
out_cart =	cart_info.txt	Name of output text file with summary CART tree information

Optional Inputs

minsplit_site=	6	Minimum number of observations per site that must exist in a CART node for a split to be attempted. It is a control parameter used to prevent overfitting. For ITT analyses, the threshold applies separately to the RCT control and QED comparison groups, whereas for CACE analyses, the threshold applies separately to RCT control nonparticipants (those with **got_treat** = 0), RCT treatment nonparticipants, and QED comparisons. *Default = 6*.
minsplit_tot =	20	Same as **minsplit_site** except pertains to the minimum number of observations across all sites. *Default = 20*.
minbucket_site, minbucket_tot		Minimum number of observations required in any terminal leaf node of the CART tree. These are also control parameters to prevent overfitting. These inputs parallel the **min_split** inputs from above and are set to, **round(minsplit_site/3)** and **round(minsplit_tot/3)**. The current function does not allow alternative input values.
holdout =	1	1 = estimate the CART tree using random 30 percent holdout and 70 percent training samples in each site, 0 = estimate the tree without holdout samples. Future package versions will conduct cross-validation to select the complexity parameter and optimal subtrees if experience suggests that user study sample sizes are large enough to support 5 or 10 folds (see Schochet, 2026). *Default = 1*.
holdout_seed =	42	Seed used to select the holdout and training samples. The same seed will produce the same samples and CART tree. *Default = 42*.

2. Inputs for the view_selected_tree() function

Variable	Example input	Description

Required Inputs

subtree_num =	5	Subtree number of the selected subtree, which can be found using Table 4, Column 1 in the **out_cart** text file produced by bias_tree().
treeinfo_df =	best_trees_df	Name of data frame containing results of the bias_tree() function (see above section on inputs to bias_tree()).
out_view =	view_tree.txt	Name of output text file with details on the selected tree

3. Inputs for the calc_pred_bias() function

***The* calc_pred_bias() *function must be run separately using the same RCT and QED input data frames as specified for running* bias_tree()**.

Further, as with **bias_tree()**, to run **calc_pred_bias()**, users will need to assign the result of the function to a new data frame. This new data frame will be identical to the input data frame except that it will also contain two additional columns: (i) "**pred_bias**" containing the predicted bias for each individual in the input data set, and (ii) "**left_right**" that identifies the terminal leaf in which each individual lands based on the person's covariate values (it contains a string of "L" and "R" values denoting left or right movements along the CART tree branches). For example, if the new data frame for the QED input data is to be named, "**qed_pred_bias**", it can be assigned using the following code:

qed_pred_bias_df <- **calc_pred_bias()**

and similarly for the RCT input data:

rct_pred_bias_df <- **calc_pred_bias()**

The required and optional function inputs for **calc_pred_bias()** are as follows:

Variable	**Example input**	**Description**
Required Inputs		
pred_dat_df =	qed_df	Name of QED or RCT input data frame to obtain predicted values. These input files must be the same as for the **bias_tree()** function. The **calc_pred_bias()** function needs to be run twice, once using the QED input data frame and once using the RCT input data frame.
subtree_num =	5	Subtree number of the selected subtree for the impact analysis. It can be found in Table 4, Column 1 in the **out_cart** text file produced by **bias_tree()**.
treeinfo_df =	best_trees_df	Name of the assigned data frame when running **bias_tree()** (see above section on inputs to **bias_tree()**).
out_pred_bias =	qed_pred_bias_res.txt	Name of output text file containing summary information on the predicted values for those in the input data file

4. Inputs for the hybrid_impacts() function

***The* hybrid_impacts() *function must be run using, as inputs, the RCT and QED data frames assigned when running* calc_pred_bias()**. For instance, using the example in the previous section, the **rct_pred_bias_df** and **qed_pred_bias_df** data frames created by the separate runs of **calc_pred_bias()** using the RCT and QED samples could serve as inputs for **hybrid_impacts()**.

Below are the required and optional function inputs for **hybrid_impacts()**, many of which overlap with the inputs for the **bias_tree()** function:

Variable	**Example input**	**Description**
Required Inputs		
rct_dat_df =	rct_pred_bias_df	Name of assigned RCT data frame from running **calc_pred_bias()** using the RCT samples (see previous section)
qed_dat_df =	qed_pred_bias_df	Name of assigned QED data frame from running **calc_pred_bias()** using the QED samples (see previous section)

Variable	Example input	Description
rct_qed_site =	rq_site	Indicator variable: 1 = individual is in an RCT-QED site, 0 = individual is in a QED-only site. *Required only for the QED input data frame, but not for the RCT input data frame.*
yvar =	y	Outcome variable. The function must be run separately for each analysis outcome variable.
site_id =	site	Unique site ID
t_c =	trtv	Treatment status indicator: 1 = individual in treatment group, 0 = individual in RCT control or QED comparison group
cace_itt_est =	1	1 = CACE analysis, 0 = ITT analysis
got_treat =	d	Treatment participation indicator: 1 = RCT member received treatment services, 0 = RCT member did not receive treatment services. *Required only for the RCT input data frame for CACE analyses (i.e., when* ***cace_itt_est*** *= 1), but not otherwise*.
ipw_wgt =	ipw	Variable containing the IPW weights. *Required for the QED input data frame, but not for the RCT input data frame*.
out_impacts =	impact_results.txt	Name of output text file with impact results
Optional Inputs		
cluster_id =	0	0 = non-clustered design (individual is the unit of randomization) or: Name of cluster ID variable for clustered designs (groups, such as hospitals, communities, or schools, are the unit of randomization). *Default = 0*.
xvars_rct_adj =	x1+x2+x5	0 = no covariates or: Baseline covariates for regression adjustment in the RCT impact models, separated by a "+" sign. These covariates can differ from those included in the **xvars_cart** input for **bias_tree()**. *Default = 0*.
xvars_qed_adj =	0	0 = no covariates or: Baseline covariates for regression adjustment in the QED impact models, separated by a "+" sign. Inclusion of these covariates could cause overcorrection of the QED biases so should be used cautiously (see Schochet, 2026). *Default = 0*.
inv_var_agg_wgt =	1	1 = inverse probability weighting used to weight the RCT impacts and bias-adjusted QED impact estimates for pooling, 0 = RCT and QED sample sizes used for weighting. *Default = 1.*

RUNNING THE PROGRAM

The ***Hybrid_Est.R*** functions can be run sequentially either during the same R session or in separate sessions by reading in the stored data frames from earlier runs. Analysis results are displayed in the console as well as in more detail in the specified output text files. If the functions are rerun, make sure to provide new inputs for the output text files or the old output files will be overwritten.

R requirements and installing the required libraries

Hybrid_Est.R can be run in R using standard methods for running R programs: R Version 4.5.2 run using R Studio 2026.01.0+392 was used for testing. Before running the functions, users will need to download the following nine R packages from the official R repository (CRAN): stringr, listr, dplyr, data.tree, ivreg, survey, lmtest, sandwich, and clubSandwich. These packages can be installed, for example, using the install.packages("stringr") command, and similarly for the other eight R packages. If not installed, users may be asked if they want them installed the first time the program is run.

Steps for running the program

```
#  Set the working directory
setwd("C:/MyDirectory")

# Call the Hybrid_Est.R script that was saved to the working directory
source("Hybrid_Est.R")

# Read the input RCT and QED data frames stored in .rds format. For example, if the data frames,
# "rct_eval_dat.rds" and  "qed_eval_dat.rds", are stored in the specified working directory, R code
# for reading these files into data frames for the functions could be:
rct_df   <- readRDS("rct_eval_dat.rds")
qed_df <- readRDS("qed_eval_dat.rds")

# Call the bias_tree() function to estimate the CART bias trees, assigning the result of the function
# to a new data frame containing information on the constructed trees. An example input specification
# using default optional input values is:
best_trees_df <- bias_tree(rct_dat_df = rct_df,
                           qed_dat_df = qed_df,
                           rct_qed_site = rq_site,
                           yvar = y,
                           xvars_cart = x1+x2+x3,
                           site_id = site,
                           t_c = trtv,
                           cace_itt_est = 1,
                           got_treat = d,
                           ipw_wgt = ipw,
                           out_cart = cart_info.txt)

# Save the assigned data frame from bias_tree() (e.g., best_trees_df ) as an .rds file for future use:
saveRDS(best_trees_df, file = "best_trees_df.rds")

# Review the specified output text file from bias_tree() and select the desired subtree for subsequent
# analyses. In particular, examine Table 4, Column 1 to select the desired subtree number. Next, run
# the view_selected_tree() function to obtain more detailed information on the selected subtree, for
# example, using the following R code:
view_selected_tree(subtree_num = 5, treeinfo_df = best_trees_df,  out_view = view_tree.txt)
```

```
# Run calc_pred_bias() twice to calculate predicted values from the selected subtree using the QED and
# RCT input data frame files. Example R code, which also saves the assigned data frames for future use,
# is as follows:
qed_pred_bias_df <- calc_pred_bias(pred_dat_df = qed_df, subtree_num = 5,
                                   treeinfo_df = best_trees_df,
                                    out_pred_bias = qed_pred_bias_res.txt)

rct_pred_bias_df <- calc_pred_bias(pred_dat_df = rct_df, subtree_num = 5,
                                   treeinfo_df = best_trees_df,
                                   out_pred_bias = rct_pred_bias_res.txt)

saveRDS(qed_pred_bias_df, file = "qed_pred_bias_df.rds")

saveRDS(rct_pred_bias_df, file = "rct_pred_bias_df.rds")

# Run hybrid_impacts() to estimate the pooled RCT and bias-adjusted QED impact using the assigned
# data frames from the calc_pred_bias() runs, for example, using the following R code:
hybrid_impacts(rct_dat_df = rct_pred_bias_df,
               qed_dat_df = qed_pred_bias_df,
               rct_qed_site = rq_site,
               yvar = y,
               site_id = site,
               t_c = trtv,
               cace_itt_est = 1,
               got_treat = d,
               ipw_wgt = ipw,
               out_impacts = impact_results.txt,
               cluster_id = 0,
               xvars_rct_adj = x1+x2+x5)
```