Evaluating the Use of Design Weights in Classification Trees for Modeling Survey Nonresponse

Tien-Huan Lin[1], William Cecere[1], Michael Jones[1], Jennifer Kali[1]
[1]Westat, 1600 Research Blvd, Rockville, MD 20850

Abstract
Nonresponse adjustments are often performed on survey weights to reduce the bias of estimates when analyzing complex sample survey data. Several algorithms are available when modeling survey nonresponse for these adjustments, many of which include the option to incorporate design weights. The literature reports uncertain findings related to the benefits of weighting in these settings. Lohr, Hsu, and Montaquila (2015) found no benefits in using weights when modeling response propensity; Lin and Flores Cervantes (2019) showed minor improvements with weighted analysis; and Cecere, Lin, Jones, Kali, and Flores Cervantes (2020) and Jones, Cecere, Kali, and Lin (2021) reported mixed results. A shared limitation of these studies is that they were not specifically designed to assess the use of weights when using these methods. In this paper, we investigate the sensitivity of select classification tree-based algorithms when using weights by conducting a simulation study of a stratified sample design with design weights highly correlated to the outcome variable. We compare unweighted and weighted analysis and evaluate the effect of incorporating design weights on estimating response propensity and reducing nonresponse bias.

Key Words: classification trees, nonresponse bias, response propensities, design weights, survey weights

1. Introduction

Missing information resulting from sampled units who refuse to participate can negatively impact the quality of the estimates made from the survey data. Many methods are available to adjust weights to account for this type of nonresponse (i.e., unit nonresponse; Brick and Montaquila 2009). When undertaking this task, researchers are faced with the choice of methods to use to best adjust estimates for nonresponse; that is, to adjust the sampling weights that produce estimates with reduced nonresponse bias while minimizing their variance. A popular method among survey statisticians is the weighting-class adjustment method (Lessler and Kalsbeek 1992). The weighting classes are created either by fitting regression models to predict response propensity and making cutpoints of the estimated propensity or by utilizing terminal nodes of classification or regression trees (Lohr et al. 2015).

This paper focuses on nonresponse adjustments for weighting classes based on the terminal nodes of classification trees fitted to the observed response status (i.e., respondent and nonrespondent). Researchers have made progress in this area over the past few years. For example, Toth and Phipps (2014) explored the use of regression trees as a tool to study the characteristics of survey nonresponse, and Lohr et al. (2015) compared the estimates

obtained with nonresponse adjusted weights from various classification tree and random forest algorithms. Lohr et al. explored the choices of the parameters for these methods; for example, the inclusion or exclusion of survey weights and different pruning methods and loss functions. Their research favored the conditional inference tree method (i.e., R package *ctree*, explained in Section 2), advised against recursive partitioning (i.e., R package *rpart*), and found no benefit of using survey weights when modeling response propensity. More recently, Lin and Flores Cervantes (2019) compared nonresponse adjusted estimates based on weighting-class nonresponse adjustments to estimates with weights adjusted using a two-step modeling approach based on the gradient boosting algorithm. This method incorporated both the probability of response and estimated survey outcomes into the nonresponse adjustment to reduce bias while controlling for variance (Little and Vartivarian 2005). Lin and Flores Cervantes found benefits of the traditional weighting method combined with the recursive partitioning for modeling survey data (i.e., R package *rpms*) over the two-step modeling approach with gradient boosting. Cecere et al. (2020) expanded the research of Lohr et al. (2015) and Lin and Flores Cervantes (2019) by comparing additional tree-building algorithms as well as those they recommended for the creation of nonresponse weighting classes. Cecere et al. compared the algorithms empirically through a Monte Carlo simulation study using an artificial population and response based on the data from the American Community Survey (ACS) Public Use Microdata Sample (PUMS) and found that the conditional inference tree method as implemented by the *ctree* package in R produced the most favorable results as measured by empirical bias and variance. Jones et al. (2021) built upon the research of Cecere et al. (2020) by comparing select tree-building algorithms under a cluster sample design and under low- and high-response scenarios using the same artificial population and response based on data from the ACS PUMS.

Kott (2012) argues that when estimating population means of survey variables that roughly behave as random variables with constant means within weighting classes that incorporating the design weights into the adjustment factors will usually be more efficient than not incorporating them. Under their conditions, Lohr et al. (2015) found no benefit to using survey weights when building classification trees. Our research specifically evaluates the use of design weights under conditions where the response propensity is correlated with the design weight. As with Cecere et al. and Jones et al., we compare the algorithms empirically through a Monte Carlo simulation study using the same artificial population and response based on the data from the ACS PUMS. The performance of the method is evaluated using the empirical bias and variance of the estimators of two outcomes.

The rest of the paper is organized as follows. In Section 2, we describe the nonresponse adjustment algorithms included in the comparisons. Section 3 describes the details of the simulation, such as the source for the population frame, predictors, response definitions, and dependent variables, in addition to the sample design. Section 4 describes the simulation study, while Section 5 summarizes the simulation results, including direct comparisons to results found in Cecere et al. (2020). We finish in Section 6 with conclusions and recommendations for future research.

2. Nonresponse Weighting Candidate Models

A large number of tree-based algorithms have been described in the literature (see Loh 2014). We evaluated six tree-building algorithms in this study (see Table 1), which were chosen based on recommendations from the literature. Three of the methods are

implemented by packages in R, and three of the methods are options under the HPSPLIT procedure in SAS.

Table 1: Tree-building algorithms evaluated

Program	*Algorithm*
R Package	
partykit	Conditional Inference Tree (*ctree*)
REEMTree	Random Effects Models (*REEM*)
rpms	Recursive Partitioning for Modeling Survey Data (*rpms*)
SAS	
HPSLIT Procedure	CHAID - Node splits based on statistical tests
HPSLIT Procedure	Entropy - Node splits based on impurity

The *ctree* and *REEM* algorithms performed well in Lohr et al. (2015) and were therefore included in our research. The *rpms* algorithm is a relatively new method developed specifically to account for complex survey designs by treating weights appropriately. This algorithm was favored by Lin and Flores Cervantes (2019). We include the chi-square automatic interaction detection (CHAID) algorithm via the SAS HPSPLIT procedure because CHAID is a popular choice for creating nonresponse adjustment cells. Lin, Flores Cervantes, and Kwanisai (2021) compare two implementations of the CHAID method: SI-CHAID and the CHAID option under the HPSPLIT procedure in SAS. They found that under the conditions of their study, empirical bias and variance were not affected by differences between the two implementations. We round out our list of algorithms with two measures of impurity used to split tree nodes, the Gini index and entropy options under HPSPLIT. We present more details on each of these algorithms in the following sections.

2.1 *ctree* Algorithm

In the R package partykit (Hothorn and Zeileis 2015), the function *ctree* (Hothorn, Hornik, and Zeileis 2006) implements an algorithm that builds classification trees using the conditional distribution of the response variables given the covariates, assuming that the observations are independent. At each step, the method determines whether further partitioning is needed by testing the independence between the response variable and each covariate. If the null hypothesis is not rejected for each covariate, then it stops splitting. On the other hand, if the test is rejected for at least one covariate, it selects the covariate with the strongest association (i.e., the minimum p-value from the set of independence tests for all covariates) to be the basis of the split. The method then finds the split that results in the maximum difference of target between two nodes.

2.2 *REEM* Algorithm

It is often the case that practitioners want to account for cluster-to-cluster variability in the models for nonresponse. One solution is to treat the cluster as a fixed effect covariate. However, often there are a large number of clusters (or Primary Sampling Units) in a survey, and some tree methods have a selection bias toward variables with a large number of categories such as the PSUs, as Lohr (2015) suggests. As an alternative for accounting for area effects, Sela and Simonoff (2012) outlined an approach that uses the Expectation-Maximization (EM) algorithm for clustered data. The REEMtree package in R (Sela, Simonoff, and Jing 2021) utilizes the package rpart (Therneau, Atkinson, and Ripley 2022) for tree building with the addition of a linear model for random effects. The algorithm in the *REEM* function takes an iterative approach and alternates between fitting random effects through maximum likelihood estimation and fitting a tree after removing the

random effects. The resulting response propensities are a combination of estimates from leaves and estimated random effects.

2.3 *rpms* Algorithm

A relatively new classification algorithm reviewed in this paper is the recursive partitioning for modeling survey data algorithm implemented in the function *rpms* of the R package of the same name (Toth 2021). As implied by the name, the algorithm recursively classifies data using independent variables. This package is appropriate for survey data as it was developed explicitly to include parameters for sampling weights, clusters, and stratum definitions from complex survey designs into the trees. The *rpms* function fits a linear model to the data conditioning on the splits selected through a recursive partitioning algorithm. The models of the created classification trees are design-consistent and account for clustering, stratification, and unequal probabilities of selection at the first stage.

2.4 SAS HPSPLIT Algorithms

The HPSPLIT procedure in SAS/STAT® software (2015) builds classification and regression trees. The procedure offers several options for partitioning criteria. Two commonly used options are included in this research. The first criterion uses entropy information for classification. The second criterion used in our research is based on a CHAID algorithm, which utilizes chi-square tests to partition the data into trees. In CHAID, the natural logarithm of the *p*-value from the selected statistical test determines the best split (Kass 1980). The splitting algorithms in the HPSPLIT procedure that we studied have the potential of overfitting the training data with a full tree, resulting in a model that does not adequately generalize to new data. To prevent overfitting, HPSPLIT implements the method of *pruning*: the full tree is trimmed to a smaller subtree that balances the goals of fitting training data and predicting new data.

This paper compares the empirical bias and variance of the estimates computed using the listed methods of two outcome variables for a low-response and a high-response scenario.

3. Simulation

We created the sampling frame for the simulation study using the household-level 2013-2017 ACS PUMS. The sampling frame served as the population for a simulation study mirroring a national survey of households. The frame consisted of a one-time simple random sample (SRS) of 200,000 households (excluding group homes) of the ACS PUMS dataset. A total of 5,000 repeated samples were selected using a clustered design. The primary sampling units (PSUs), or clusters, were defined by Public Use Microdata Areas (PUMAs) or combined PUMAs containing at least 300 housing units. Sampling began by selecting 25 PSUs from each of the four Census regions for a total of 100 PSUs. One hundred housing units were then randomly selected from each of the sampled PSUs. Each simulation run consisted of 10,000 housing units.

Our simulation considers nonresponse for the final sampling unit and not the PSU level. Details of the simulation design can be viewed in Table 2. We study two response levels: low (30 percent) and high (70 percent). Previous year household income was one of the auxiliary predictors used in the modeling of response mechanisms. In one scenario, the mechanism produced response propensity that was correlated with household income, and in the other, it was not. We employed two types of sampling; a SRS of housing units as a baseline sample and a probability-proportional-to-size (PPS) sample (i.e., informative sampling) where high-income households had a greater chance of selection. In this paper

we use the terms PPS and informative sampling interchangeably. The PPS sample offered a setting where our auxiliary variable, previous year household income, was correlated with our outcome variables. We ran each setting with and without weights to assess the impact of using weights when creating nonresponse weighting cells.

We anticipated that the nonresponse bias in the SRS sampling would be the highest in the setting where correlation is high between our auxiliary variable and both response propensity and the outcome variables, and the lowest when both correlations are low. Literature on the effect of informative sampling is limited; therefore, we had no expectations of bias results under that sample design.

Table 2: Simulation design

Sampling	*Response rate*	*Auxiliary correlation*
PPS	High – 70%	High
PPS	High – 70%	Low
PPS	Low – 30%	High
PPS	Low – 30%	Low
SRS	High – 70%	High
SRS	High – 70%	Low
SRS	Low – 30%	High
SRS	Low – 30%	Low

The response mechanisms were generated using the basic model

$$\boldsymbol{r = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \beta_5 x_5}$$

Where $\boldsymbol{r}$ is the response mechanism, $\boldsymbol{\beta_0}$ is the control for levels of response in Table 2, $\boldsymbol{\beta_1}$ is the control for levels of correlation with our auxiliary predictor, and $\boldsymbol{\beta_2, \beta_3, \beta_4,}$ and $\boldsymbol{\beta_5}$ are the coefficients generated by a logistic model using the top predictors of ACS response. A logit transformation was then implemented on the various values of $\boldsymbol{r}$ to obtain response probabilities between 0 and 1.

We selected two outcome variables for the simulation study, listed in Table 3. The empirical study compared estimates of means for the continuous variable (percentage of households where all residents have health insurance) and proportions for the binary variable (at least one member of the household has a bachelor's degree) of these outcome variables.

Table 3: Outcome variable descriptions

Outcome variable	*Description*	*Type*	*Values*
Health Insurance	Percentage of households where all members have health insurance	Percentage	0-1
Bachelor's degree	Percentage of households that have at least one member with a bachelor's degree	Percentage	0-1

The population frame included 39 variables selected as predictors for nonresponse. Of those variables, 35 were household-level characteristics, while the remaining 4 were person-level characteristics derived by summarizing to the household level the corresponding person-level variables. The 39 predictors included 4 continuous variables and 35 categorical variables. The categorical variables were recoded such that the smallest

category contained at least 5 percent of the households in the population. Most tree algorithm packages used in the simulation do not handle predictors with missing values; therefore, missing values were assigned to a separate category.

The models predicting response propensities were fit using the methods in the statistical software packages discussed in Section 2. The fitted response propensity models were then used to compute weighting classes and nonresponse adjustment factors to adjust the design weights. Final weighted estimates of mean or proportions adjusted for unbalanced sample selection and nonresponse bias were computed for the outcome variables discussed above and compared against the true values from the population. The statistics examined for comparing the estimators $\hat{\bar{Y}}_E$ are the absolute empirical relative bias (RelBias), and empirical relative root mean squared error (RRMSE), defined as

$$Absolute\ Emperical\ Relative\ Bias: RelBias\left(\hat{\bar{Y}}_E\right)\% = |100 \times B^{-1} \sum_{b=1}^{B} \frac{\hat{\bar{Y}}_{E,b} - \bar{Y}}{\bar{Y}}|, \text{ as}$$

$$Relative\ Root\ Mean\ Squared\ Error: RRMSE = \sqrt{\frac{MSE(\hat{\bar{Y}}_E)}{\bar{Y}^2}},$$

where B is the number of simulations runs and $MSE\left(\hat{Y}_E\right)$ is the empirical mean squared error of $\hat{\bar{Y}}_E$ computed as $MSE\left(\hat{Y}_E\right) = \frac{\sum_{b=1}^{B}(\hat{\bar{Y}}_{E,b} - \bar{Y})^2}{B}$.

Each statistical software package contains unique sets of parameters to control tree fitting. Special effort was made to apply global settings among all packages to minimize subjective differences in bias and variance evaluation.

3.1 *ctree*

The following parameters were used for all trees:

- *Minbucket*: the minimum number of observations in a terminal node was set to 40.
- *Maxdepth*: NA.
- *Prune*: *ctree* avoids overfitting by using hypothesis tests to determine the splitting nodes stopping point, thus eliminating the need for pruning.
- *Weight:* in contrast to the other packages studied in this paper, *ctree* requires integer-valued weights and treats the weights as observation frequencies as opposed to survey weights. This parameter was not used for this reason.
- *Bonferroni*: use Bonferroni adjustment to compensate for multiple testing in the global null hypothesis, and therefore was set to Yes.
- *Alpha*: 0.05.
- *Mincriterion*: 0.95.

All other parameters were set to their default values.

3.2 *REEM*

The following parameters were used for all trees:

- *tree.control*: rpart.control.
- *Minbucket*: the minimum number of observations in a terminal node was set to 40.
- *Cp*: 0.01.
- *Random*: region was treated as the random effect in the mixed model.

All other parameters were set to their default values.

3.3 *rpms*

The following parameters were used for all trees:

- *Bin_size*: the minimum number of observations in a terminal node was set to 40.
- *Prune*: similar to the conditional inference tree, the *rpms* algorithm eliminates the step of pruning.
- *Strata*: Census region was specified as the sampling strata.
- *Cluster*: PSU was specified as the sampling clusters.
- *P-val:* 0.05.

The following factors were varied:

- *Weight:* weight = 1 for all observations or weight = design weight.

All other parameters were set to their default values.

3.4 SAS HPSPLIT Algorithms

The following parameters were set equal for all trees:

- *Minleafsize*: the minimum number of observations in a terminal node was set to 40.
- *Maxdepth*: the maximum level a tree could be grown was set to 5.
- *Prune:* to avoid overfitting, one procedure is to grow the tree out as far as possible and then prune back to a smaller subtree (Breiman, Friedman, Olshen, and Stone 1984). The pruning method specified for this package was reduced-error pruning (Quinlan 1986).

The following factors were varied:

- *Weight:* weight = 1 for all observations or weight = design weight.
- *Criterion:* CHAID, Gini, or entropy.

All other parameters were set to their default values.

4. Results

Table 4 shows the simulation results of the *high*-response setting without design weights, and when response propensity is *not* correlated to household income, our auxiliary predictor. The results are shown for each of the algorithms studied under a SRS or uninformative design, and a PPS or informative design. Results are shown for both outcomes: the proportion of household s in which all residents have health insurance and

the proportion of households in which at least one household members holds a bachelor's degree. We treat the results exhibited in Table 4 as baseline measures.

Table 4: Estimates of RelBias and RRMSE under SRS and PPS sample designs for high response when response propensity is not correlated to household income, unweighted

	Outcome variable							
	Health insurance				*Bachelor's degree*			
	Simple random sample		*Informative sampling*		*Simple random sample*		*Informative sampling*	
Algorithms	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*
ctree	0.2	0.6	0.2	0.6	0.2	0.9	0.3	1.0
REEMtree	0.2	0.6	0.3	0.7	0.4	1.0	0.3	1.0
rpms	0.3	0.6	0.3	0.7	0.5	1.0	0.6	1.2
SAS CHAID	1.0	1.4	0.1	0.7	1.7	2.8	1.4	1.9
SAS Entropy	0.7	1.2	0.1	0.6	0.3	2.1	0.0	1.0

The estimated absolute relative bias for the R package algorithms are comparable between the two sample designs and are close to zero across all simulation settings. This is observed in the results of both outcome variables, suggesting that these algorithms can largely reduce nonresponse bias when the probability of response is not correlated with an auxiliary predictor, regardless of the survey outcome or probability of selection. Compared to the R package algorithms, the SAS options generally produced estimates with elevated absolute relative biases, indicating less success in mitigating nonresponse bias. The exception to this was the result associated with health insurance when an informative sample was used, in which case the relative bias for the SAS options was lower than the R package options. The largest estimated absolute relative bias (1.7 percent) was associated with the bachelor's degree outcome measure calculated when CHAID was used and under an uninformed sample design. No apparent pattern is observed for SAS options between sample designs or outcome estimates, suggesting that the increase in bias is a result of the algorithms as opposed to the simulation settings.

The balance of bias and variance as measured by the RRMSE does not display much fluctuation among R package algorithms within a simulation setting. For the health insurance estimates, the RRMSE is consistently 0.6 percent under simple random sampling and ranges from 0.6 to 0.7 percent under informative sampling. For the bachelor's degree estimates, the range of RRMSE is 0.9 to 1.0 percent under simple random sampling and 1.0 to 1.2 percent under informative sampling. The RRMSE associated with the SAS options resulted in an overall increase, partially contributed by elevated levels of bias. A notable observation is that the SAS Entropy estimator of bachelor's degree under simple random sampling produced estimated absolute RelBias comparable to the R package algorithms (0.3 percent). However, the RRMSE is much higher (2.1 percent) indicating a higher variance associated with this outcome variable and algorithm.

Table 5 provides results under the same conditions as those in Table 4, but with design weights applied to the algorithms.

Table 5: Estimates of RelBias and RRMSE under SRS and PPS sample designs for high response when response propensity is not correlated to household income, weighted

	Outcome variable							
	Health insurance				*Bachelor's degree*			
	Simple random sample		*Informative sampling*		*Simple random sample*		*Informative sampling*	
Algorithms	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*
ctree	0.2	0.5	0.2	0.6	0.2	0.9	0.2	1.1
REEMtree	0.2	0.6	0.3	0.7	0.4	1.0	0.3	1.0
rpms	0.3	0.6	0.3	0.7	0.5	1.0	0.6	1.2
SAS CHAID	1.0	1.4	0.5	0.8	0.9	2.4	0.8	1.4
SAS Entropy	0.9	1.2	0.2	0.7	2.0	2.8	0.0	1.1

Applying design weights appears to have had a nominal effect on bias and variance. For example, the estimated absolute RelBias for the unweighted and weighted ctree estimates under most settings is 0.2 percent. The only case where use of design weights reduced the estimated absolute RelBias was SAS CHAID, but the effect was not consistent among all settings.

Table 6 provides results under conditions similar to those as Table 4, the only difference being the response propensity is *highly* correlated to the auxiliary predictor, household income.

Table 6: Estimates of RelBias and RRMSE under SRS and PPS sample designs for high response when response propensity is highly correlated to household income, unweighted

	Outcome variable							
	Health insurance				*Bachelor's degree*			
	Simple random sample		*Informative sampling*		*Simple random sample*		*Informative sampling*	
Algorithms	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*
ctree	0.4	1.3	0.4	1.4	0.9	2.2	1.5	2.6
REEMtree	0.9	1.1	1.1	1.5	2.6	2.8	2.9	3.3
rpms	0.4	1.0	0.3	1.3	0.4	1.8	0.3	2.1
SAS CHAID	1.0	1.4	1.3	1.7	1.7	2.8	3.5	4.1
SAS Entropy	0.7	1.2	0.7	1.6	0.3	2.1	1.2	2.9

Compared to Table 4, we observe an overall increase in the estimated absolute RelBias for the R package algorithms when response propensity is highly correlated to household income, the auxiliary predictor. Within an outcome estimate, the degree of increase is either similar between sample designs or sees a slightly higher degree of increase associated with informative sampling. A more prominent difference is observed between outcome estimates where the estimated absolute RelBias of the health insurance estimates are 2 to 4 times higher than those in Table 4, while the estimated absolute RelBias of the bachelor's degree estimates are 5 to 11 times higher than those in Table 4. This may suggest that the R package algorithms are better equipped to mitigate nonresponse bias induced by the probability of selection than that of survey outcome.

Among R package algorithms, the estimated absolute RelBias is comparable between the *ctree* and *rpms* estimators for the health insurance outcome, ranging from 0.3 to 0.4 percent under both sample designs, while the *REEMtree* estimators exhibit a slight increase with values between 0.9 to 1.1 percent. For the bachelor's degree outcome, the *rpms* estimators had the smallest absolute RelBias ranging from 0.3 to 0.4 percent (comparable to health insurance), followed by the *ctree* estimators ranging from 0.9 to 1.5 percent. The *REEMtree* estimators had the largest absolute RelBias ranging from 2.6 to 2.9 percent. In terms of the balance of bias and variance, the *rpms* estimators displayed the smallest RRMSE (1.0 to 2.1 percent) with the values increasing slowly from left to right of the table. The *ctree* estimators displayed slightly higher RRMSE especially in the bachelor's degree estimates (1.3 to 2.6 percent), following the same increase pattern as *rpms*. The *REEMtree* estimators displayed the highest RRMSE (1.1 to 3.3 percent), also with the values increasing from left to right of the table. The patterns of bias and variance suggest that *rpms* estimators are the most robust in handling interactions between sample design, response propensity, and survey outcome when the response rate is high and the response propensity is highly correlated to the auxiliary predictor, while *REEMtree* estimators are the least stable.

Among the SAS options, SAS Entropy produced results comparable or slightly preferable to *REEMtree* regardless of outcome estimate (RelBias: 0.3 to 1.2 percent; RRMSE: 1.2 to 2.9 percent). On the other hand, SAS CHAID produced results comparable or slightly less preferable to *REEMtree* in the estimates of health insurance but produced the least desired results among all algorithms for the bachelor's degree estimates. The largest estimated absolute RelBias and RRMSE are observed in the bachelor's degree estimate with the SAS CHAID estimator under informative sampling (3.5 percent and 4.1 percent, respectively).

Table 7 provides results under the same conditions as those in Table 6, but with design weights applied to the algorithms.

Table 7: Estimates of RelBias and RRMSE under SRS and PPS designs for high response when response propensity is highly correlated to household income, weighted

	Outcome variable							
	Health insurance				*Bachelor's degree*			
	Simple random sample		*Informative Sampling*		*Simple random sample*		*Informative sampling*	
Algorithms	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*
ctree	0.5	1.0	0.7	1.3	1.9	2.3	2.2	2.9
REEMtree	0.9	1.1	1.2	1.5	2.6	2.8	2.9	3.3
rpms	0.4	1.1	0.4	1.4	0.4	1.9	0.2	2.3
SAS CHAID	1.0	1.4	1.4	1.7	0.9	2.4	3.6	3.9
SAS Entropy	0.9	1.2	0.8	1.5	2.0	2.8	1.0	3.2

Applying design weights again appears to have been inconsequential. With a few exceptions, the RRMSE for the weighted algorithms are either unchanged or slightly higher than their unweighted counterpart. The exceptions occur in the *ctree* estimator of health insurance under simple random sampling (1.3 percent-unweighted vs. 1.0 percent-weighted), the SAS CHAID estimator of bachelor's degree under simple random sampling (2.8 percent-unweighted vs. 2.4 percent-weighted), and the SAS CHAID estimator of bachelor's degree under informative sampling (4.1 percent-unweighted vs. 3.9 percent-weighted). However, the difference in RRMSE exhibited in the exceptions are minimal and do not suggest a substantial improvement in weighted algorithms.

Table 8 provides results applying the same conditions as those from Table 4, but for the *low*-response setting.

Table 8: Estimates of RelBias and RRMSE under SRS and PPS sample designs for low response when response propensity is not correlated to household income, unweighted

	Outcome variable							
	Health insurance				*Household w/ bachelor's degree*			
	Simple random sample		*Informative Sampling*		*Simple random sample*		*Informative sampling*	
Algorithms	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*
ctree	0.4	1.0	0.5	1.1	0.7	1.6	1.0	1.9
REEMtree	0.6	1.0	0.5	1.1	0.2	1.5	0.2	1.7
rpms	0.7	1.1	0.7	1.2	0.9	1.7	1.2	2.1
SAS CHAID	3.5	3.7	0.4	1.7	12.0	12.4	1.9	3.2
SAS Entropy	1.5	2.6	0.4	1.1	0.4	4.5	0.4	1.8

The simulation results in Table 8 demonstrate that both bias and variance are amplified by low response rate when all other simulation settings are fixed. The R package algorithms are still largely successful in reducing nonresponse bias with the estimated absolute RelBias for health insurance ranging from 0.4 to 0.7 percent, and that of bachelor's degree ranging from 0.2 to 1.2 percent. In general, bachelor's degree estimates exhibit a slightly higher bias than health insurance estimates. The RRMSE for R package algorithms are twice as large as those displayed in Table 4 in almost all settings while still within the range of 2.1 percent. These patterns suggest that when the outcome estimate is highly correlated to the auxiliary predictor, R package algorithms may be slightly less effective in reducing nonresponse bias when the response rate is low.

Estimates from the SAS options continued to exhibit less desirable results than those of R package algorithms. In some settings, the SAS Entropy estimators showed comparable levels of bias reduction as R package algorithms but the pattern is not consistent. Moreover, the RRMSE values of SAS Entropy estimators are generally higher than those of R package algorithms. SAS CHAID estimators produce the least desired results. A notable observation is large measures of RelBias and RRMSE associated with the bachelor's degree estimate with SAS CHAID under a SRS (RelBias: 12.0 percent; RRMSE: 12.4 percent).

Table 9 provides results under the same conditions as Table 8, but with design weights applied to the algorithms. The results exhibit similar patterns shown in Table 5 and Table 7.

Table 9: Estimates of RelBias and RRMSE under SRS and PPS sample designs for low response when response propensity is not correlated to household income, weighted

	Outcome variable							
	Health insurance				*Household w/ bachelor's degree*			
	Simple random sample		*Informative sampling*		*Simple random Sample*		*Informative sampling*	
Algorithms	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*
ctree	0.4	1.0	0.3	1.1	0.8	1.7	0.7	1.9
REEMtree	0.6	1.0	0.5	1.1	0.2	1.4	0.2	1.7
rpms	0.7	1.1	0.7	1.2	1.0	1.8	1.4	2.2
SAS CHAID	3.1	3.7	1.2	1.6	10.0	11.4	2.9	3.5
SAS Entropy	1.5	2.5	0.5	1.3	1.3	4.8	0.7	2.0

Table 10 shows the simulation results of the *low*-response setting without design weights when response propensity is *highly* correlated to the auxiliary predictor for each of the algorithms studied under a SRS design and an informative design.

Table 10: Estimates of RelBias and RRMSE under SRS and PPS sample designs for low response when response propensity is highly correlated to household income, unweighted

	Outcome variable							
	Health insurance				*Household w/ bachelor's degree*			
	Simple random sample		*Informative sampling*		*Simple random Sample*		*Informative sampling*	
Algorithms	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*
ctree	0.5	2.3	0.4	2.7	2.0	4.2	1.7	4.4
REEMtree	2.0	2.4	2.2	2.9	4.8	5.3	5.2	6.0
rpms	1.5	2.4	1.5	3.0	1.7	4.2	1.4	4.9
SAS CHAID	3.5	3.7	4.1	4.3	12.0	12.4	15.9	16.2
SAS Entropy	1.5	2.6	1.7	3.5	0.4	4.5	0.8	6.1

With the underlying structure of a high correlation between response propensity and our auxiliary predictor, the *ctree*, *rpms*, *REEMtree*, and SAS Entropy algorithms continue to demonstrate effectiveness in reducing nonresponse bias while balancing variance when the outcome estimate is not correlated to the auxiliary predictor (i.e., health insurance). In both simple random sampling and informative sampling, the algorithm that best balances bias and variance (i.e, exhibits the least RRMSE) is *ctree* (2.3 percent and 2.7 percent, respectively). In the most extreme setting where the outcome estimate is also highly correlated to the auxiliary predictor (i.e., bachelor's degree), the algorithm that best balances bias and variance for both simple random sampling and informative sampling is again *ctree* (4.2 percent and 4.4 percent, respectively). SAS CHAID again produced the least desired results, with RRMSE values of 3.7 percent (SRS) and 4.3 percent (informative sampling) for the health insurance estimate, and enormous RRMSE values of 12.4 percent (SRS) and 16.2 percent (informative sampling) for the bachelor's degree estimate.

Table 11 provides results under the same conditions as those that produced results in Table 10, but applies weights to the algorithms. The observations are, again, similar to those displayed in Table 5, Table 7, and Table 9 where applying design weights appear to have a nominal effect on results.

Table 11: Estimates of RelBias and RRMSE under SRS and PPS sample designs for low response when response propensity is highly correlated to household income, weighted

	Outcome variable							
	Health insurance				*Household w/ bachelor's degree*			
	Simple random sample		*Informative sampling*		*Simple random sample*		*Informative sampling*	
Algorithms	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*
ctree	0.1	2.5	1.1	3.6	2.4	4.5	2.5	6.2
REEMtree	2.0	2.4	2.2	2.9	4.8	5.3	5.2	6.0
rpms	0.8	3.5	0.8	5.0	1.2	6.0	1.0	8.4
SAS CHAID	3.1	3.7	3.6	3.9	10.0	11.4	13.1	13.3
SAS Entropy	1.5	2.5	2.0	3.5	1.3	4.8	1.3	6.2

6. Conclusions

Using the 2013-2017 ACS PUMS data as a pseudo-population, and under a cluster sample design, we selected repeated samples drawn from a fixed population with PUMAs serving as the PSUs. By using the ACS PUMS as our fixed population, we were able to mimic a national household-level survey and introduce a nonresponse mechanism that allowed for comparisons between estimates and true population values. We investigated the use of the following five tree algorithms for producing nonresponse classification cells using a simulation study: *rpms*, *ctree*, *REEM*, CHAID, and Entropy; the former three are R packages or part of R packages and the latter two are called by the HPSPLIT procedure in SAS.

Under the stratified SRS in Cecere et al. (2020), *ctree* stood out as the algorithm that produced the smallest relative bias and RRMSE for all outcomes compared to the other algorithms. Under the cluster design in Jones et al. (2021), the results were mixed; for a high-response scenario, all the methods performed well at reducing nonresponse bias, with the *ctree* algorithm performing slightly better than the rest; for a low-response scenario there was no consistent "winner." For the high-response scenario in our simulation, when response propensity was not correlated to the auxiliary predictor (previous year household income), all R package algorithms effectively reduced nonresponse bias while SAS options were less successful. When response propensity was highly correlated to the auxiliary predictor, the three R package algorithms remained effective when the outcome estimate was not correlated to our auxiliary predictor. Results began to deteriorate when the outcome estimate was correlated to the auxiliary predictor, with *ctree* and *rpms* still producing favorable relative root mean square error and *REEMtree* seeing a higher increase in bias and/or variance. SAS Entropy results were comparable to *REEMtree*, while SAS CHAID produced the least desired relative root mean square error. Both bias and variance were amplified in the low-response scenario in our simulation – *ctree* and *rpms* continue to produce reasonable results, followed by *REEMtree* and SAS Entropy. Results from SAS CHAID continued to be the least desirable.

We performed weighted and unweighted analyses for all the algorithms. Our results showed minimal differences between weighted and unweighted analyses for relative bias and RRMSE for both outcome variables. Moreover, we observed no substantial improvement in the weighted analyses in the informative sampling scenario, which we specifically designed to evaluate the effectiveness of applying design weights in tree

algorithms. This observation surprisingly includes the *rpms* algorithm, in which we expected improvements from the use of weights since we developed the algorithm to account for complex sample design. This result agrees with the recommendation of Lohr et al. (2015) in that weights do not provide a benefit when modeling response propensity.

Simulation results may be different for other sample designs. For operational efficiency, national samples often incorporate a clustering stage, forming PSUs of smaller geographic areas and selecting households within PSUs. We anticipated that *rpms* and *REEMtree* would perform better under a clustered sample design due to the usage of area effects. However, this was not the case. These algorithms could also be tested under additional sample design frameworks.

A limitation of our simulation study is that statistical tests were not conducted comparing the results of the various software packages. Additionally, the number of simulations is only 5,000, which reduces the ability to make inferences about the results.

Our results showed minimal differences between weighted and unweighted analyses. There could potentially be two explanations: 1) weights have little effect in the tree algorithms tested, or 2) the outcome variables used in our analysis did not have a high enough correlation to the probability of selection. Therefore, we would like to test the effect of using weights during the nonresponse adjustments with an outcome variable that has a higher correlation with the probability of selection than what was used in this study. Additionally, this simulation study was designed to provide results for a stratified design. We would like to perform this analysis with a clustered design.

Acknowledgments

The authors are grateful to Jean Opsomer for his insightful suggestions.

References

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. (1984), *Classification and Regression Trees,* Belmont, CA: Wadsworth.

Brick, J. M., and Montaquila, J. (2009), "Nonresponse and Weighting," in D. Pfeffermann and C. R. Rao (eds.), *Handbook of Statistics, Vol. 29A. Sample Surveys: Design, Methods, and Applications*. Amsterdam: Elsevier, 163-185, DOI: 10.1016/S0169-7161(08)00008-4.

Cecere, W., Lin, T. H., Jones, M., Kali, J., and Flores Cervantes, I. (2020), "A Comparison of Classification and Regression Tree Methodologies When Modeling Survey Nonresponse," in American Statistical Association *Proceedings of the Survey Research Methods Section*, pp. 577-585.

Hothorn, T., Hornik, K., and Zeileis, A. (2006), "Unbiased Recursive Partitioning: A Conditional Inference Framework," *Journal of Computational and Graphical Statistics*, 15(3), pp. 651–674, DOI: 10.1198/106186006X133933.

Hothorn, T., and Zeileis, A. (2015), "partykit: A Modular Toolkit for Recursive Partytioning in R," *Journal of Machine Learning Research*, 16, 3905-3909. Available at https://jmlr.org/papers/v16/hothorn15a.html.

Jones, M., Cecere, W. E., Lin, T.-H., and Kali, J. (2021), "Modeling Survey Nonresponse Under a Cluster Sample Design: Classification and Regression Tree Methodologies Compared," in American Statistical Association *Proceedings of the Survey Research Methods Section*.

Kass, G. V. (1980), "An Exploratory Technique for Investigating Large Quantities of Categorical Data," *Journal of the Royal Statistical Society*, Series C 29, 119–127, DOI: 10.2307/2986296.

Kott, P. S. (2012), "Why One Should Incorporate the Design Weights When Adjusting for Unit Nonresponse Using Response Homogeneity Groups," *Survey Methodology*, 38, 95-99. Available at http://www.statcan.gc.ca/pub/12-001-x/2012001/article/11689-eng.pdf.

Lessler, J. T., and W. D. Kalsbeek. (1992), *Nonsampling Errors in Surveys* (1st Ed.), New York: John Wiley and Sons.

Little, R. J. A., and Vartivarian, S. (2005), "Does Weighting for Nonresponse Increase the Variance of Survey Means?" *Survey Methodology*, 31, 161-168.

Lin, T. H., and Flores Cervantes, I. (2019), "A Modeling Approach to Compensate for Nonresponse and Selection Bias in Surveys," in American Statistical Association *Proceedings of the Survey Research Methods Section*, pp. 827-834.

Lin, T. H., Flores Cervantes, I., and Kwanisai, M. (2021), "A Comparison of Two CHAID Packages for Modeling Survey Nonresponse," in American Statistical Association *Proceedings of the Survey Research Methods,* forthcoming.

Loh, W.-Y. (2014), "Fifty Years of Classification and Regression Trees," *International Statistical Review,* 82, 329-348.

Lohr, S., Hsu, V., and Montaquila, J. (2015), "Using Classification and Regression Trees to Model Survey Nonresponse," in American Statistical Association *Proceedings of the Survey Research Methods Section*, pp. 2071-2085.

Quinlan, J. R. (1986), "Induction of Decision Trees," *Machine Learning,* 1, 81-106.

SAS Institute, Inc. (2015), *SAS/STAT® 14.1 User's Guide,* Cary, NC: SAS Institute, Inc.

Sela, R. J., and Simonoff, J. S. (2012), "RE-EM Trees: A Data Mining Approach for Longitudinal and Clustered Data," *Machine Learning,* 86, 169-207.

Sela, R. J., Simonoff, J., and Jing, W. (2021), *REEMtree: Regression Trees with Random Effects*, R package version 0.90.4. Available at https://CRAN.R-project.org/package=REEMtree.

Therneau, T., Atkinson, B., and Ripley, B. (2022), *rpart: Recursive Partitioning and Regression Trees*, Version 4.1.16. Available at https://CRAN.R-project.org/package=rpart.

Toth, D., and Phipps, P. (2014), "Regression Tree Models for Analyzing Survey Response," in American Statistical Association *Proceedings of the Government Statistics Section,* pp. 339-351.

Toth, D. (2021). *rpms: Recursive Partitioning for Modeling Survey Data,* Version 0.5.1. Available at https://CRAN.R-project.org/package=rpms.