Comparing Classification Tree Algorithms When Modeling Survey Nonresponse in Clustered Survey Designs

Jennifer Kali[1], Tien-Huan Lin[1], William Cecere[1], Michael Jones[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, each performing well under different scenarios. Lin, Cecere, Jones, and Kali (2022) found that the amount of bias reduction in the weights varied by choice of algorithm. Nonresponse bias is a function of the predictor, the response mechanism, and the survey outcome (Little and Vartivarian 2005). In this paper, we investigate the sensitivity of select classification tree-based algorithms by conducting a simulation study of a clustered sample design under a variety of conditions, varying the correlation of the predictor with the response mechanism and survey outcome. We also consider the effectiveness of including design weights when modeling nonresponse. We compared survey estimates to evaluate the available algorithms on reducing nonresponse bias under various simulated conditions.

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

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 design weights to account for unit nonresponse (see Brick and Montaquila 2009). The goal of adjusting design weights for nonresponse is to produce estimates with reduced nonresponse bias while minimizing their variance. The weighting-class adjustment method is a popular method to adjust design weights for nonresponse (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, Hsu, and Montaquila 2015).

The focus of this paper is on formation of weighting classes based on the terminal nodes of classification trees fitted to the observed response status (i.e., respondent and nonrespondent). There are many classification tree software packages available, each with a different underlying model for tree fitting. Others have researched the use of classification trees for nonresponse adjustments. 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. 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. 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.

Kott (2012) argued that when estimating population means of survey variables that roughly behave as random variables with constant means within weighting classes, incorporating the design weights into the adjustment factors will usually be more efficient than not incorporating them. However, under their conditions, Lohr et al. (2015) found no benefit to using survey weights when building the classification trees (as opposed to creating weights within the weighting classes derived from the trees).

Cecere et al. (2020) and Jones et al. (2021) expanded the research of Lohr et al. (2015) and Lin and Flores Cervantes (2019) by comparing additional tree-building algorithms for stratified and clustered sample designs. Both Cecere et al. and Jones 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). For both stratified (Cecere et al.) and clustered (Jones et al.) designs, 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. Similar to Lohr et al. (2015), neither Cecere et al. (2020) nor Jones et al. (2021) found a benefit in including design weights in tree classification algorithms when adjusting for nonresponse.

The limitation of the simulation design utilized in Cecere et al. (2020) and Jones et al. (2021) was the reliance on the response indicator from the ACS. While defining the response indicator in this manner mirrored reality, there was little control in the simulation design to investigate under which conditions each tree classification software performed best. Nonresponse bias is a function of the predictor, the response mechanism, and the survey outcome (Little and Vartivarian 2005). Lin et al. (2022) expanded on the research of Cecere et al. (2020) and Jones et al. (2021) by recognizing this relationship. Lin et al. (2022) used a synthetic response mechanism to interrogate the effect of correlation of the predictor with the response mechanism and survey outcome. Additionally, we evaluate the use of design weights under conditions where the response propensity is correlated with the design weight. Similar to Cecere et al. (2020), Lin et al. (2022) simulated a stratified sample design using the same pseudo-population and Monte Carlo simulation design as Cecere et al. (2020) and Jones et al. (2021). Again, as with the other studies, the conditional inference tree method as implemented by the *ctree* package in R produced the most favorable results, and we found no benefit in including design weights in tree classification algorithms when adjusting for nonresponse.

This paper examines the simulation conditions from Lin et al. (2022) but with a cluster sample design. The performance of the tree classification software was 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, and Section 5 summarizes the simulation results. 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 two 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) compared 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 a measure of impurity used to split tree nodes using the entropy option 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 [PSUs]) in a survey, and some tree methods have a selection bias toward variables with many 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 stratified design. The PSUs, or clusters, were defined by public use microdata areas (PUMAs) or combined PUMAs containing at least 300 housing units and stratified by region. 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; an 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 design 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; however, results from Lin et al. (2022) give us some expectation that biasing the sample selection process will likely increase the nonresponse bias.

Table 2: Simulation design

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

The response mechanisms were generated using the basic model

$$\boldsymbol{r} = \frac{\boldsymbol{e}^{(\boldsymbol{\beta_0}+\boldsymbol{\beta_1}\boldsymbol{x}+\sum \boldsymbol{\beta_i}\boldsymbol{x_i})}}{\mathbf{1}+\boldsymbol{e}^{(\boldsymbol{\beta_0}+\boldsymbol{\beta_1}\boldsymbol{x}+\sum \boldsymbol{\beta_i}\boldsymbol{x_i})}}$$

Where $\boldsymbol{r}$ is the response mechanism, $\boldsymbol{x}$ is the household income from the previous year, $\boldsymbol{\beta_1}$ are different values to control for level of correlation with $\boldsymbol{x}$, and $\sum \boldsymbol{\beta_i}\boldsymbol{x_i}$, are modeled from frame data.

We selected three 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, proportions for the binary variable (at least one member of the household has a bachelor's degree), and a third synthetic variable. The three outcome variables were chosen based on their differing levels of correlation with household income.

Table 3: Outcome variable descriptions

Outcome variable	*Description*	*Type*	*Values*	*Correlation with household income*
Health Insurance	Percentage of households where all members have health insurance	Percentage	0-1	-0.14
Bachelor's degree	Percentage of households that have at least one member with a bachelor's degree	Percentage	0-1	0.40
Synthetic	Synthetically created variable	Continuous	0+	0.65

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

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

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

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 or entropy.

All other parameters were set to their default values.

4. Results

Our discussion of results relies on visual comparisons of bias and RMSE between the tree-building algorithms for various simulation settings. The comparisons of relative bias are found in Figures 1 to 4. Each figure is a trellis plot containing a 3 x 3 grid of bar charts with the rows showing the three outcomes that reflect differing levels of correlation (from top to bottom: health insurance [low], bachelor's degree [medium], and a synthetic outcome [high]) with the auxiliary variable (i.e., household income), and the columns showing the three levels of correlation between the auxiliary variable and response propensity. Each panel shows the unadjusted relative bias along with relative bias after adjusting for nonresponse using the different tree-building algorithms we considered. Results generated when not applying base weights to the algorithms and applying base weights are found within each panel.

Figure 1 shows the simulation results of the *high*-response setting under an SRS or uninformative design. The top row of panels (health insurance [low]) reflecting the scenario of low correlation between the auxiliary and outcome indicates minimal unadjusted nonresponse bias, showing that all the algorithms under consideration performed well under the applied conditions. The same is true for the panels in the middle row of the grid, indicating that when the response rate is high, changing only the correlation between the auxiliary variable and the outcome variable from low to a medium level did not affect the performance of the algorithms. These bar charts show that in the presence of a high response rate, when the relative bias is minimal the algorithms perform well independent of whether weights are used when forming nonresponse adjustment cells. This is not an unexpected result. However, the algorithms do not perform consistently as we move down and to the right of the grid. That is, under SRS and in the presence of high response, as the correlation between the auxiliary variable and outcome variable increases and the correlation between the auxiliary variable and response propensity also increases, it becomes clear that algorithm performance can be ranked. For example, consider the bottom row of panels (synthetic outcome [high]): in the bottom right panel, which represents a high correlation between the auxiliary variable and response propensity, while each algorithm reduced bias, the *ctree*, REEM and RPMS algorithms appear to have performed better than the CHAID and Entropy algorithms, regardless of whether weights were applied. Overall, the *ctree* algorithm was most successful and consistent in reducing bias associated with nonresponse compared to the other algorithms considered. The REEM algorithm also performed well under the conditions present in Figure 1. The RPMS algorithm also reduced bias, but not at the same levels as *ctree* and REEM, regardless of the use of weights.

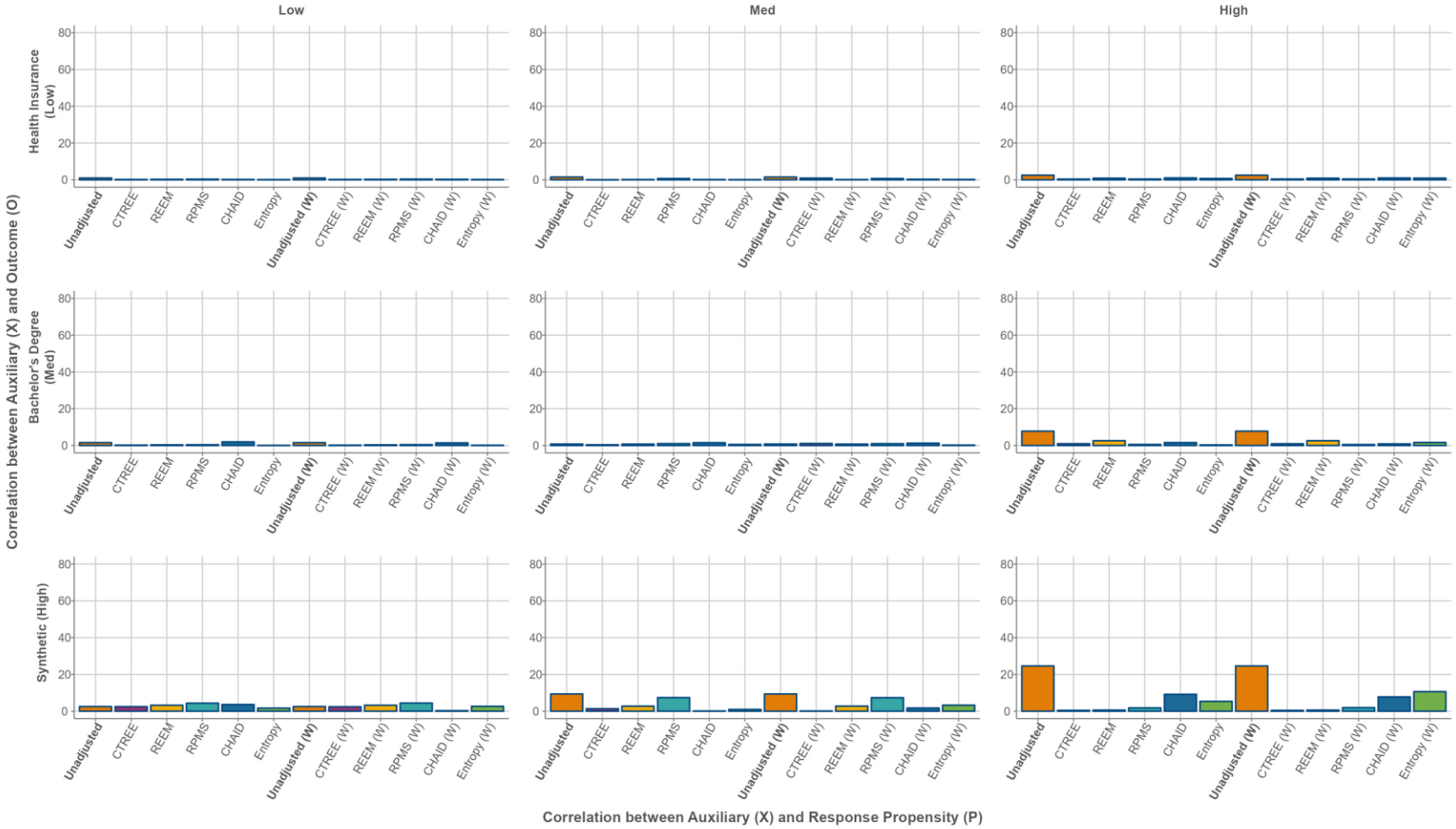

Figure 1: Estimates of RelBias under Uninformative sample design for high response

Figure 2 shows the simulation results of the *high*-response setting under PPS sampling, or an informative design. The top row of panels in the Figure 2 grid follows the same patterns as that in Figure 1 in that the simulation scenarios led to minimal unadjusted nonresponse bias and all the algorithms under consideration performed similarly well. Continuing with the panels in the bottom two rows of the grid, the patterns are again the same as those in the same positions of the Figure 1 grid. In particular, increasing the correlation between the auxiliary variable and outcome variable to a high level produced noticeable differences in algorithm performance, with the *ctree*, REEM, and RPMS algorithms performing best, and the CHAID and Entropy algorithms, regardless of whether weights were applied, performing worse.

Figure 3 shows the simulation results of the *low*-response setting under an SRS or uninformative design. The same patterns as those observed in Figure 1 can be found in Figure 3, at a much higher magnitude. For example, in the bottom row of the grid where the maximum RelBias in Figure 1 was just over 20 percent, it is now close to 80 percent for the equivalent panels in Figure 3. In particular, consider the lower right panel where the correlation between the auxiliary variable and outcome variable and the correlation between the auxiliary variable and response propensity are both high. The CHAID algorithm performed extremely poorly under those conditions, minimally reducing the high level of relative bias, and performed much worse than any other algorithm. A similar, yet not as extreme, result can be seen in the middle panel on the bottom row. In this scenario, instead of CHAID it is the RPMS algorithm that performed the worst at reducing the bias, with CHAID (when weights were applied) performing only slightly better than RPMS. Again, as with the conditions under Figure 1, under the Figure 3 conditions the CTREE algorithm was the most successful and consistent in reducing the bias associated with nonresponse compared to the other algorithms considered, followed by entropy and REEM.

Figure 4 shows the simulation results of the *low*-response setting under an informative design. The same patterns as those observed in Figure 3 can be found in Figure 4, with similar levels of RelBias, indicating the sample design has minimal effect on the performance of the algorithms.

The patterns exhibited in the bar charts of Figures 1 and 2 are very similar as are the patterns in Figures 3 and 4, lending credence to the conclusion that under our simulation conditions, the sample design did not affect the performance of the algorithms. Algorithm performance depended more on the level of nonresponse and the correlation between the auxiliary variable and both response propensity and outcome variable.

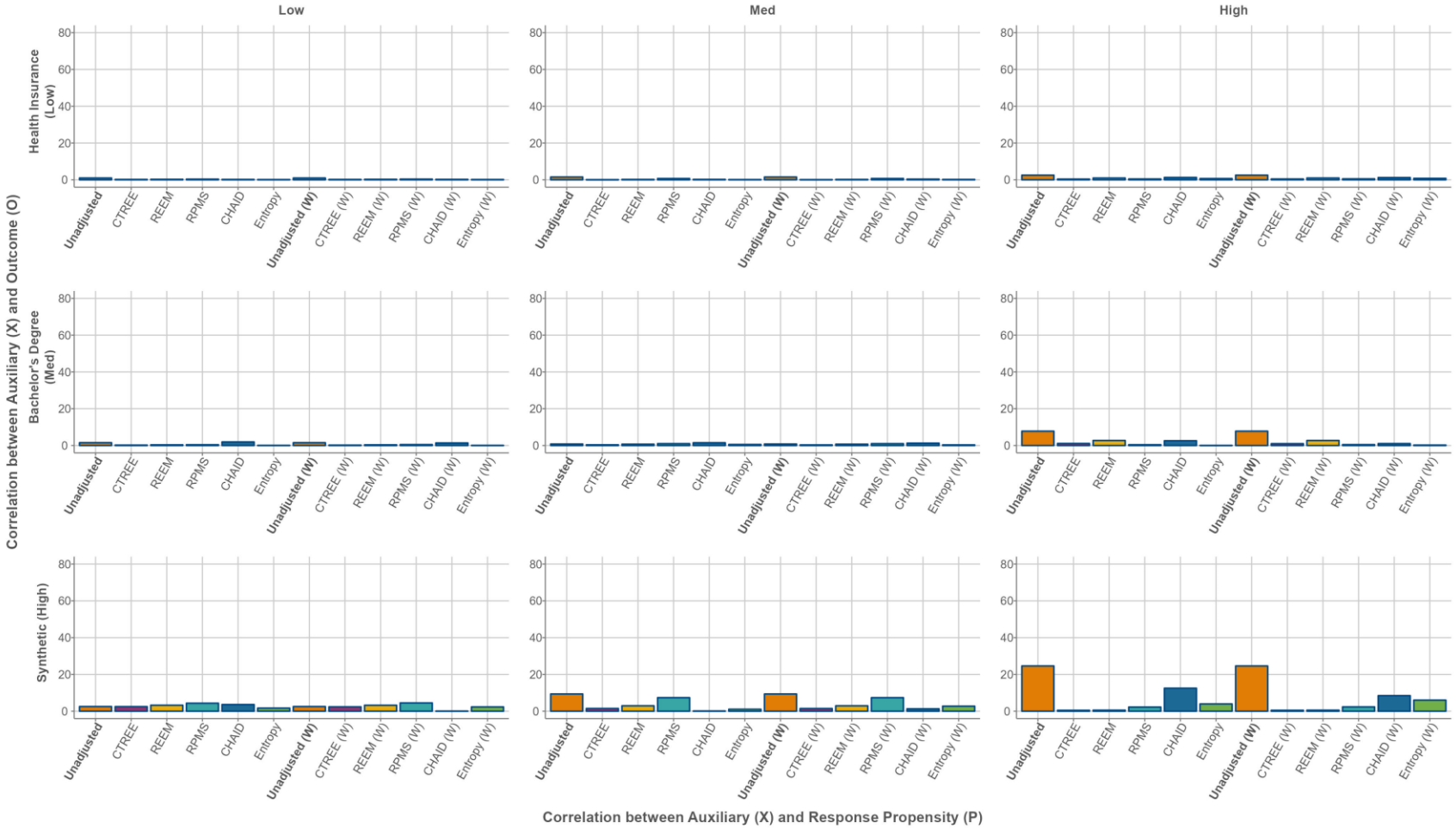

Figure 2: Estimates of RelBias under Informative sample design for high response

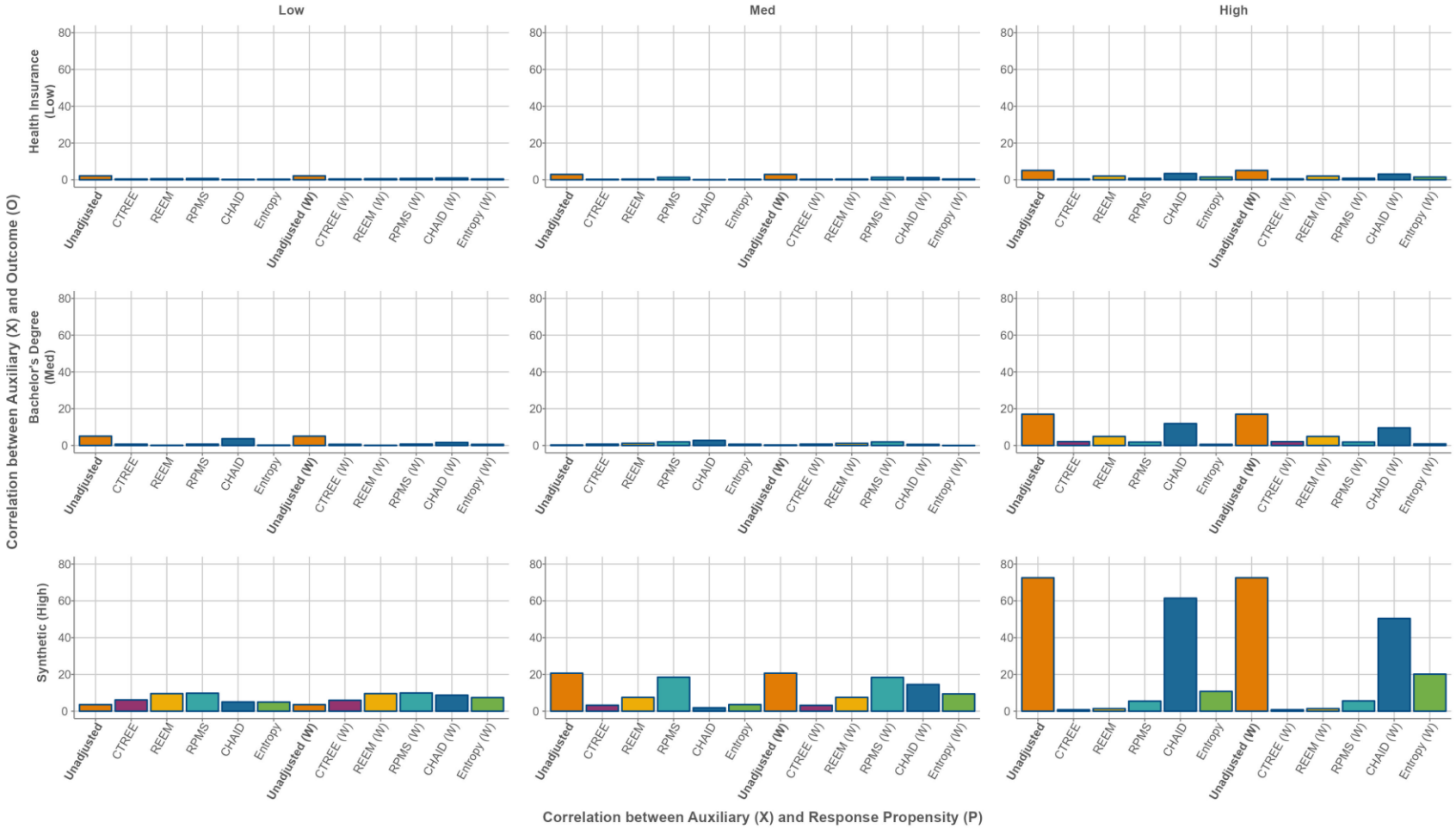

Figure 3: Estimates of RelBias under Uninformative sample design for low response

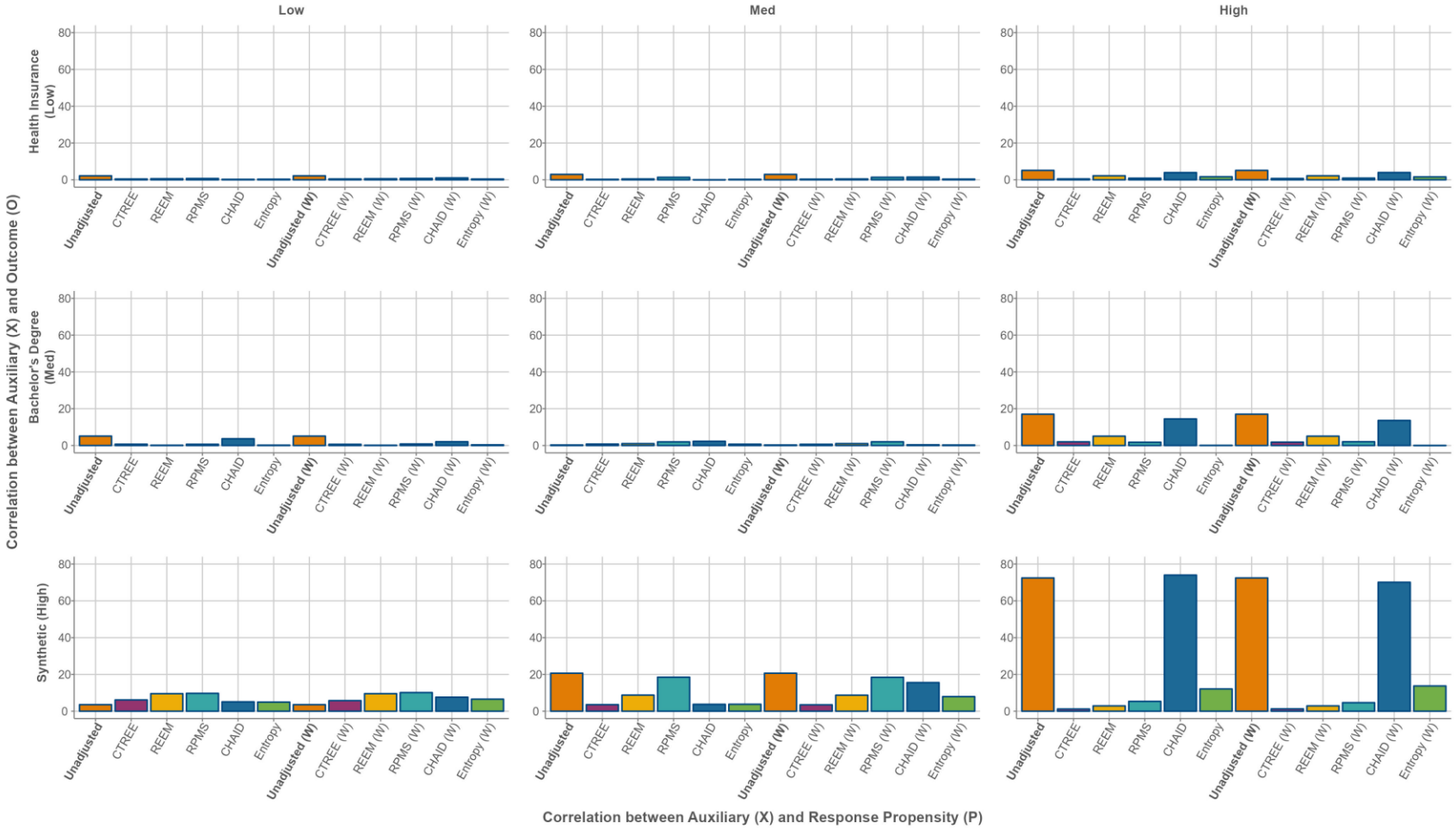

Figure 4: Estimates of RelBias under Informative sample design for low response

Next, we examine how the algorithms perform by considering the relative root mean squared error (RRMSE) under the applied simulation scenarios. As with the relative bias comparisons, our RRMSE comparisons will be visual. The figures showing results include two panels of RRMSEs; the one on the left for the uninformative design and the other for the informative design. Along the vertical axis of each panel in the figures are the resulting RMSEs for each algorithm tested and each level of correlation between the auxiliary variable and response propensity.

Figure 5 shows the RRMSE results under the high-response scenario. The majority of RRMSE values are below 10 percent for all the algorithms, with most of those over that threshold coming from the scenario of a high correlation between the auxiliary variable and response propensity for the synthetic outcome variable (i.e., a high correlation between the auxiliary variable and the outcome variable). The high values of RRMSE are also mostly associated with CHAID, when applying weights and not applying weights to the algorithm. The CTREE and REEM methods consistently produced the lowest RRMSEs except for one CTREE scenario.

Figure 6 includes RRMSE results under the low-response scenario. Most of the RRMSEs associated with low and medium correlation between the auxiliary variable and the outcome variable are below 10 percent. Almost all of the RRMSE values higher than 10 percent are associated with the synthetic outcome variable, i.e., the variable that is highly correlated with the auxiliary variable. Under the low-response condition, the CTREE algorithm consistently outperformed the other algorithms, especially with respect to the synthetic outcome variable. On the other hand, the CHAID algorithm consistently underperformed, with the RRMSE associated with the synthetic variable exceeding 70 percent.

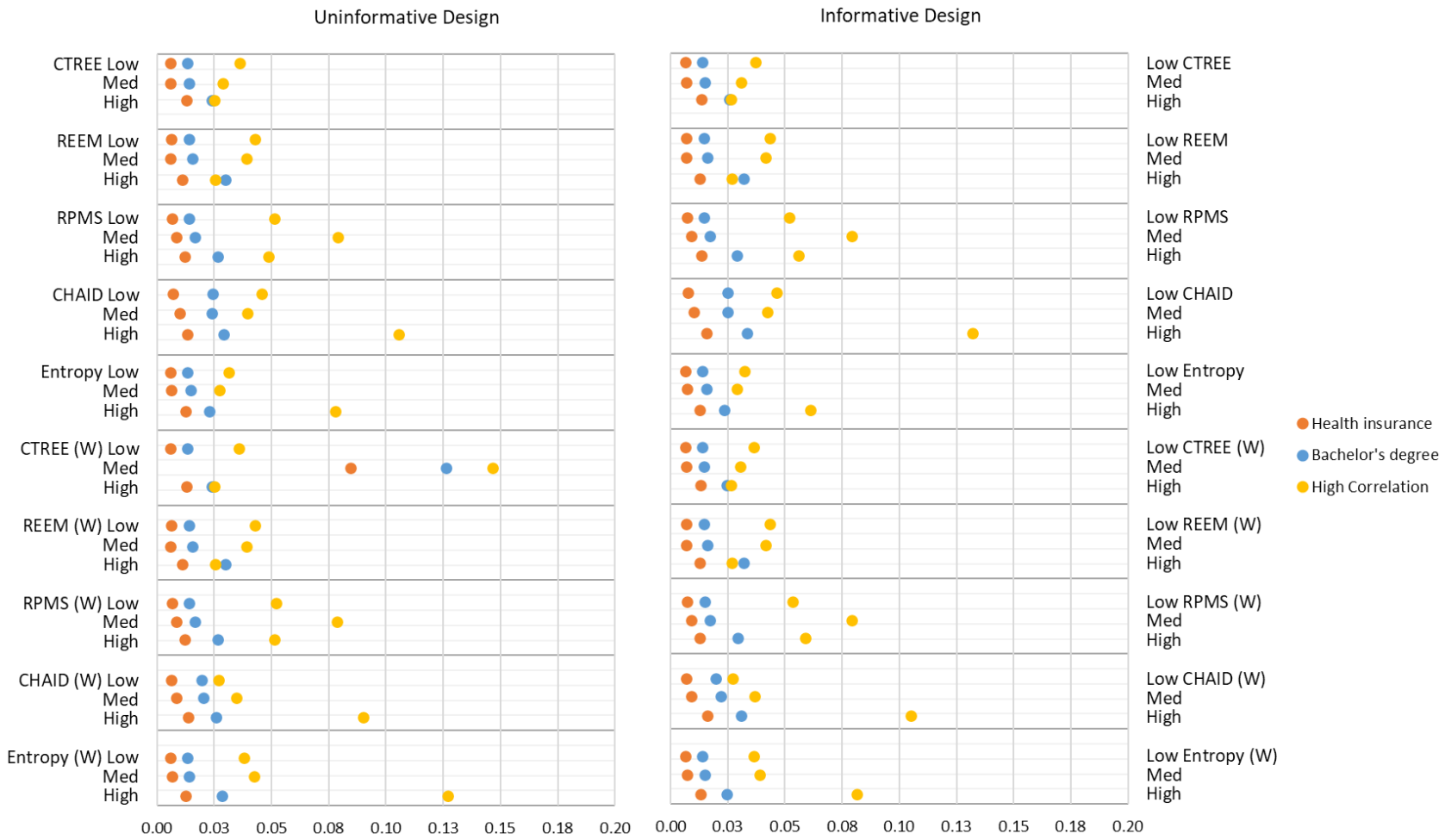

Figure 5: Estimates of RRMSE for high response

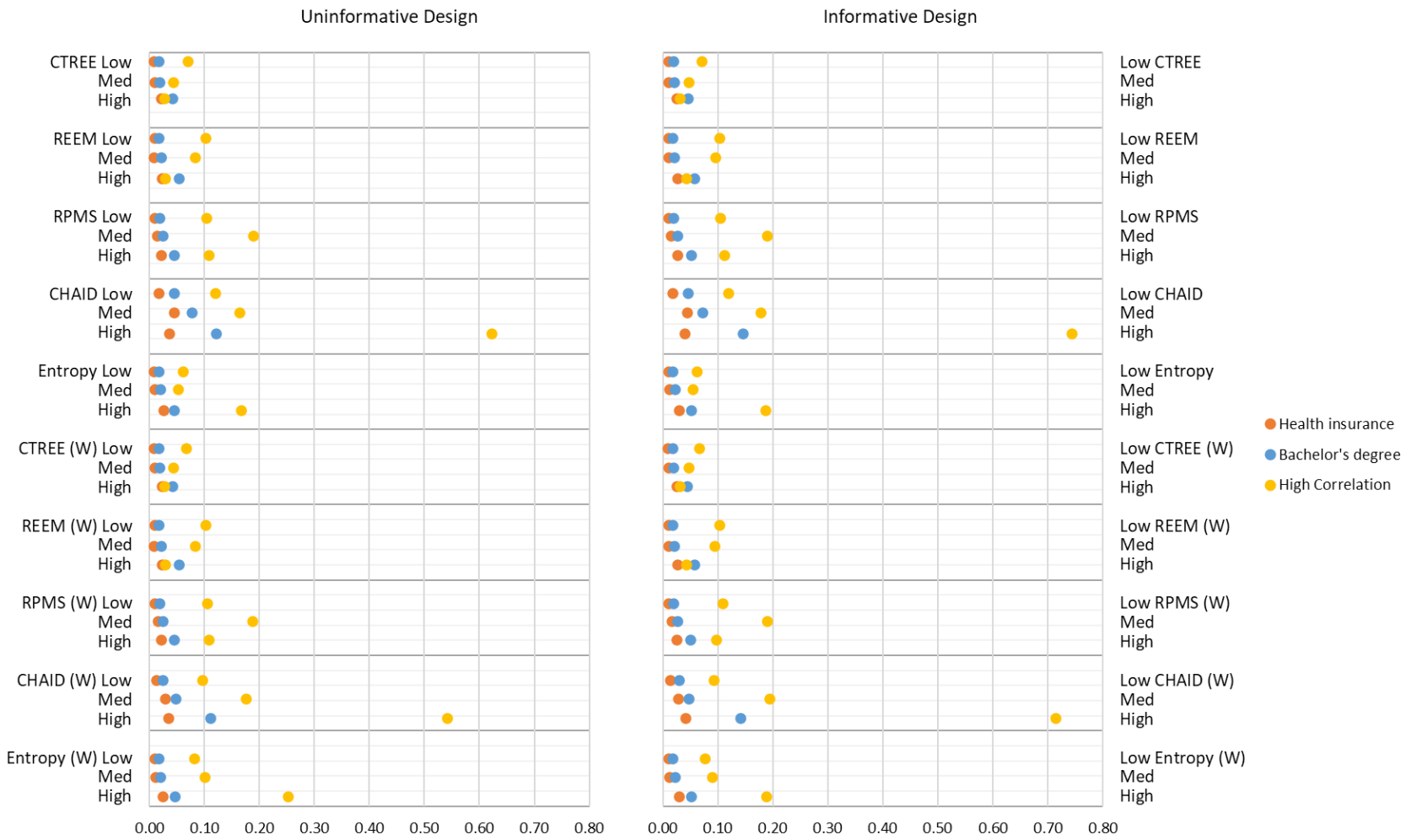

Figure 6: Estimates of RRMSE for low response

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 clustered 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." Under the stratified SRS in Lin et al. (2022), for the high-response scenario, when response propensity was not correlated to the auxiliary predictor, 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. When the outcome estimate was correlated to the auxiliary predictor, *ctree* and *rpms* still produced favorable relative root mean square error In the low-response scenario in our simulation—*ctree* and *rpms* continue to produce reasonable results. For all three of these studies, there were minimal differences between weighted and unweighted analyses for relative bias and RRMSE for both outcome variables.

For this study, under a clustered design, when considering the high response rate scenario, there is little difference among the algorithms except where either the response propensity or outcome variable is highly correlated with the auxiliary variables. Overall, the CTREE algorithm was most successful and consistent in reducing bias associated with nonresponse compared to the other algorithms considered, and the REEM algorithm also performed well. The use of design weights in the tree algorithm did not appear to have an impact.

When considering the low-response rate scenario, the choice of algorithm is more crucial. In this scenario there is more nonresponse bias in the design weights and thus more adjustment is required than with the low-response rate scenario. Again, in this setting, the CTREE algorithm was the most successful and consistent in reducing the bias associated with nonresponse compared to the other algorithms considered, followed by entropy and REEM. Again, the use of design weights in the tree algorithm did not appear to have an impact.

Our result—that the use of design weights in the tree algorithms when creating weighting class cells to adjust for nonresponse does not have an impact on the reduction of nonresponse bias—agrees with the recommendation of Lohr et al. (2015) in that weights do not provide a benefit when modeling response propensity. However, this could also be an indicator of the fact that except for RPMS, none of the tree classification models properly account for complex survey designs in their models. The S-CHAID R package is

in development. This is a recursive partitioning algorithm that aims to properly account for complex survey design. We will include this package in our future research.

Simulation results may be different for other sample designs or under different simulation conditions. We will continue our research examining additional simulation scenarios.

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 limits the ability to make precise inferences about the results.

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.

Lin, T. H., Cecere, W., Jones, M., Kali, J. (2022), "Evaluating the Use of Design Weights in Classification Trees for Modeling Survey Nonresponse," in American Statistical Association *Proceedings of the Survey Research Methods*, forthcoming.

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.

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

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.