Modeling Survey Nonresponse Under a ClusterSample Design: Classification and Regression Tree Methodologies Compared

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

Abstract
When computing survey weights for use during the analysis of complex sample survey data, an adjustment for nonresponse is often performed to reduce the bias of the estimates. Many algorithms and methodologies are available to analysts for modeling survey nonresponse for these adjustments. Lohr et al. (2015) discussed the benefits of using classification trees for estimating response propensities in surveys and how these methods could be used to reduce nonresponse bias. Cecere et al. (2020) extended their findings and recommendations using expanded simulations for more complex sample designs, such as a stratified design with equal sample size allocation. This paper will compare select algorithms when working with a complex cluster sample design. We evaluate the effect of the classification tree-based methods on the reduction of nonresponse bias and investigate the performance of the methods when they are used to adjust survey weights. We compare our results to those found in Cecere et al. (2020), and we discuss the benefits and limitations of using these methods for estimating response propensities in surveys that utilize a cluster sample.

Key Words: classification trees, clustering, nonresponse bias, response propensities, survey weights, weighting class adjustments

1. Introduction

Missing information that results 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 analysts adjusting weights to account for this type of nonresponse (i.e., unit nonresponse; Brick & Montaquila, 2009). When undertaking this task, researchers are faced with the choice of methods to use to best adjust 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 & Kalsbeek, 1992). The weighting classes are created either by fitting regression models to predict for 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 of

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 & 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.

Our research builds specifically upon the results of Cecere et al. (2020), which studied select tree-building algorithms under a stratified random sample design. We expand upon their findings by comparing the same tree algorithms but under a cluster sample design, and under a low response and a high response scenario. As with Cecere 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 American Community Survey (ACS) Public Use Microdata Sample (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

There is an extensive list of tree algorithms 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	Gini index - Node splits based on impurity
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 in Lin and Flores Cervantes (2019). We include the CHAID algorithm via the SAS HPSPLIT procedure because CHAID is a popular choice for creating nonresponse adjustment cells. Lin et al. (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. More details on each of these algorithms are presented in the following sections.

2.1 *ctree* Algorithm

In the R package partykit (Hothorn & Zeileis, 2015, Hothorn et al., 2017), the function *ctree* (Hothorn et al., 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 PSU-to-PSU variability in the models for nonresponse. One solution is to treat the PSU as a fixed effect covariate. However, often there are a large number of PSUs 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 et al., 2021) utilizes the package rpart (Therneau et al., 2019) 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, 2018). 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. Three commonly used options are included in this research. The first criterion maximizes reduction in node impurity as measured by the Gini index. The second uses entropy information for classification. The third 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 *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

The sampling frame for the simulation study was created using the household-level 2013-2017 American Community Survey (ACS) Public Use Microdata Sample File (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 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. We study two response levels: low and high. In many simulation studies a response model for the propensity to respond is posited using a set of predictors. Our simulation attempts to mirror a real-world scenario by using the ACS mode of response to derive the simulation response. As shown in Table 2, for our low response setting, only households that responded to the ACS by mail were assigned as respondents for the simulation. The response rate for the low response setting is 27.9 percent. For high response, households

that responded by internet or mail were considered simulation respondents. The response rate for the high response setting is 68.7 percent.

Table 2: Response Status Definitions for Low and High Response

Low response	*High response*	*Mode of response to the ACS*	*Population percentage*
Nonrespondent	Respondent	Internet	40.8
Respondent	Respondent	Mail	27.9
Nonrespondent	Nonrespondent	In person (CATI/CAPI)	31.3

Although our definitions of response allowed us to compute realistic estimates in the presence of nonresponse and also obtain population values, this may produce biased estimates if any of the predictors in the model do not explain the response mechanism.

We selected two outcome variables for the simulation study that are listed in Table 3. The empirical study compared estimates of means for the continuous variable (household income) and proportions for the binary variable (health insurance) of these outcome variables.

Table 3: Outcome Variable Descriptions

Outcome variable	*Description*	*Type*	*Values*
Household Income	Household income for the past 12 months	Continuous	Min: -$8,500 Max: $2,034,500
Health Insurance	Flag indicating all members of the household have health insurance coverage	Binary	1: yes 0: no

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 summarized 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 empirical relative bias, and empirical relative root mean squared error, defined as

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

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

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

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, Recursive Partitioning for Modeling Survey Data 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 et al., 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 low response setting for each of the algorithms studied under a stratified random sample design (see Cecere et al., 2020) and a cluster design, including both outcomes: the mean of past 12-month household income and the proportion of households in which all residents have health insurance. The table also shows the values of the empirical relative bias (RelBias) and empirical relative root mean squared error (RRMSE) defined in the previous section.

Table 4: Relative Bias and Relative Root Mean Square Error for Outcome Variables Under a Stratified Random Sample and Cluster Sample Designs for Low Response

	Outcome variable							
	Household Income				*Health Insurance*			
	Stratified random sample		*Cluster sample*		*Stratified random sample*		*Cluster sample*	
Algorithms	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*
ctree (unweighted)	-2.0	2.3	-3.0	4.5	1.2	1.5	1.2	1.7
REEMtree (unweighted)	-4.6	5.4	-5.2	6.7	2.2	2.9	1.8	2.6
rpms (unweighted)	-5.1	5.5	-5.1	6.0	1.3	1.9	1.1	1.6
rpms (weighted)	-5.1	5.4	-5.1	6.1	1.3	1.9	1.1	1.6
SAS CHAID (unweighted)	-9.6	11.2	-1.8	4.6	6.1	7.4	5.3	7.0
SAS CHAID (weighted)	-4.7	6.7	-0.7	4.7	2.7	4.3	5.7	6.8
SAS Gini (unweighted)	-9.3	10.8	-1.0	4.5	5.6	7.1	6.5	7.1
SAS Gini (weighted)	-4.1	5.5	-1.6	4.6	2.3	3.5	5.5	6.9
SAS Entropy (unweighted)	-9.6	11.2	-0.7	4.3	6.1	7.4	5.9	7.8
SAS Entropy (weighted)	-4.9	6.3	-0.8	4.4	2.2	3.7	6.7	7.7

Results for past 12-month household income show that estimated relative bias for all algorithms is negative for both sample designs. Under the stratified simple random sample (i.e., stratified SRS) design, the three estimators based on the SAS unweighted options had the largest relative bias, while the lone *ctree* algorithm had the smallest bias (-2.0). However, under the cluster design, the results were quite different. With the new sample design, the relative bias associated with the SAS options had the lowest relative bias, with all seeing a sizeable reduction of that under the stratified random sample design. The relative bias for the R package algorithms under the cluster sample design either did not change or increased slightly compared to that of the stratified SRS design. The variance as measured by the RRMSE exhibited a similar pattern, with SAS options displaying a sizeable reduction and R packages either unchanged or seeing a slight increase in variance. In addition, under the stratified random sample design in Cecere et al. (2020), it was observed that applying weights to the SAS options appeared to improve relative bias and variance. However, applying the weights to the new sample design appears to have had a nominal effect on bias and variance. For example, the relative bias for the unweighted and weighted estimates using entropy to split nodes is -0.7 and -0.8, respectively. The only case where use of weights reduced the relative bias was CHAID.

For the estimates of the proportion of households where all were covered by health insurance, the relative bias is positive for all estimates under both sample designs. Under the stratified SRS, the three SAS unweighted estimators had the largest RelBias, as with

household income outcome. While the *ctree* estimator had the smallest at 1.2, both *rpms* weighted and unweighted estimators were fairly close at 1.3. The relative bias under the cluster design and unweighted estimates were close to the relative bias under the stratified SRS with the exception of the weighted SAS options, in which the relative bias increased under the cluster design. In addition, applying weights to the algorithms did not reduce relative bias or variance under the cluster design as it did under the stratified SRS. Finally, the relative bias and RRMSE were smaller for the R package algorithms compared to the SAS algorithms, with *rpms* performing best with a weighted and unweighted relative bias and RRMSE of 1.1 and 1.6, respectively.

The results in Table 4 are for a low-response scenario. We present results for our high-response scenario in Figure 1. The figure displays the bias of each algorithm for the health insurance outcome under high response. The horizontal yellow line indicates the true population value. The left-most box plot shows the base-weighted estimate prior to implementing nonresponse. To the immediate right of that is the base-weighted results for respondents only. The area between the box plot and the population value gives an indication of the nonresponse bias. The remaining box plots illustrate the distribution of the bias of each method over the 5,000 simulation runs. The figure shows that for the high-response scenario, all the methods performed almost equally well in reducing nonresponse bias for the health insurance outcome. Another feature the figure shows is that the application of weights did not affect the amount of bias reduced. Similar results were found for the past 12-month household income outcome under the high-response scenario.

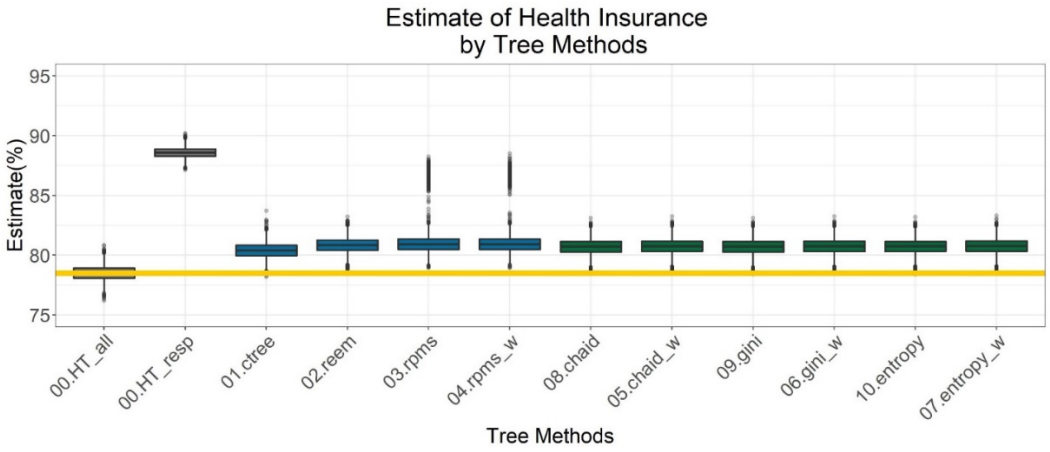

Figure 1: Box plot of bias by algorithm for health insurance outcome under high response

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 Public Use Microdata Areas serving as the primary sampling units. 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 six tree algorithms for producing nonresponse classification cells using a simulation study: *rpms*, *ctree*, *REEM*,

CHAID, Gini, and Entropy; the former three are R packages or part of R packages and the latter three are called by the HPSPLIT procedure in SAS.

Weighted and unweighted analyses were performed for the *rpms* and SAS algorithms, whereas *REEM* and *ctree* did not utilize weights. Simulation results from Cecere et al. (2020) that used a stratified simple random sample indicated that incorporating weights when applicable in the prediction of response propensity in classification trees outperformed unweighted classification trees for the three algorithms called using SAS's HPSPLIT procedure. However, when using a cluster sample design, our results showed minimal differences between weighted and unweighted analyses for empirical bias and RRMSE for both outcome variables. This observation surprisingly includes the *rpms* algorithm, in which improvement in results by the use of weights was expected since the algorithm was developed to account for complex sample design. Although this result agrees with the recommendation of Lohr et al. (2015) in that weights do not provide a benefit when modeling response propensity, it appears that whether or not one should use weights depends on the algorithm being used.

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 our simulation, the results were mixed. For the low-response scenario, the CHAID and entropy methods performed the best for household income, and for the health insurance outcome, *ctree* and *rpms* (unweighted and weighted) had the best results. For the high-response scenario, all the methods performed well at reducing nonresponse bias, with the *ctree* algorithm performing slightly better than the rest. However, there does not appear to be a practical difference between the performance of the algorithms under a high-response scenario. All the algorithms effectively reduced relative bias at about the same levels for both outcomes studied.

Simulation results may be different for other sample designs. For operational efficiency, national samples often incorporate a clustering stage, forming Primary Sampling Units (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 the target population. Our study relied on ACS PUMS data to mimic a national household-level sample when in reality, many surveys measure person-level characteristics. In addition to incorporating additional sample designs, the evaluation of the algorithms would also benefit from inclusion of person-level estimates in the simulation design.

Acknowledgments

The authors are grateful to Jean Opsomer, Bob Fay, and Ismael Flores Cervantes for the 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, pp. 163-185.

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." *Proceedings of the Survey Research Methods Section of the American Statistical Association*, 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), 651–674.

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

Hothorn, T., Hornik, K., Strobl, C., and Zeileis, A. (2017). "partykit: A Toolkit for Recursive Partytioning." Version 1.3-1. https://cran.r-project.org/package=partykit

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.

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 Methology*, 31, 161-168.

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

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

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." *Proceedings of the Survey Research Methods Section, American Statistical Association*, 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 for Longitudinal (Panel) Data." https://CRAN.R-project.org/package=REEMtree

Therneau, T., Atkinson, B., and Ripley, B. (2019). "rpart: Recursive Partitioning and Regression Trees." https://CRAN.R-project.org/package=rpart

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

Toth, D. (2018). "rpms: Recursive Partitioning for Modeling Survey Data." Version 0.3.0. https://CRAN.R-project.org/package=rpms