A Comparison of Classification and Regression Tree Methodologies When Modeling Survey Nonresponse

William Cecere[1], Amy Lin[1], Michael Jones [1], Jennifer Kali[1], Ismael Flores Cervantes[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. In this paper, we extend their findings and recommendations based on expanded simulations for more complex sample designs, such as a stratified design with equal sample size allocation. We evaluate the effect of some 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 discuss the benefits of using these methods for estimating response propensities in surveys.

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

1. Introduction

One of the major challenges during the creation of survey weights is to account for nonresponse. Missing information that results from sampled units who refuse to participate can negatively impact the quality of the estimates made from the survey data. When undertaking this task, researchers are faced with the choice of methods and options 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. Brick and Montaquila (2009) provide an overview of a wide range of nonresponse adjustment weighting methods. 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. 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).

Our research builds upon the results of Lohr et al. (2015), which 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. We also expanded the results in Lin and Flores Cervantes (2019), which found limited benefits of the gradient boosting (i.e., R package *xgboost*), over the weighting class approach with weighting classes created using a recursive partitioning algorithm for survey data (i.e., R package *rpms*). We expand upon these findings by comparing other tree algorithms in addition to those recommended by these papers, for the creation of nonresponse weighting classes. We compare 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). The performance of the method is evaluated using the empirical bias and variance of the estimators of four 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, and dependent variables, in addition to the sample design. Section 4 describes the simulation study, while Section 5 summarizes the simulation results. 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 four methods in this study, which were chosen based on recommendations from the literature. The methods are

- Conditional Inference Tree (*ctree*) algorithm from the PARTYKIT package in R.
- Random Effects Models (*REEM*) from the REEMtree package in R. The two later methods performed well in Lohr et al. (2015).
- CHAID algorithm from SAS, which is also a popular choice. The examined options of CHAID were the Gini and entropy options.
- Recursive Partitioning for Modeling Survey Data (*rpms*) algorithm from the RPMS package in R, which is a relatively new method designed specifically for surveys.

For each of these methods that we covered, we attempted both the classification and regression versions. The results were quite similar across the tree methods, except for some differences for SAS-CHAID. We will discuss only the results of the classification trees in this paper except for the results of *rpms*, which does not offer them. A more detailed discussion of the differences between the use of regression and classification trees for weighting can be found in Lohr et al. (2015).

2.1 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. Another uses entropy information for classification. The third type of criterion is based on a CHAID algorithm, which utilizes chi-square tests to partition the data into trees. The natural logarithm of the *p*-value from the selected statistical test determines the best split (Kass, 1980). The splitting measures available in the HPSPLIT procedure have the potential of overfitting the training data with the 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.

2.2 *ctree* Algorithm

In the R package PARTYKIT (Hothorn et al., 2017), the function *ctree* 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.

2.3 *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 et al. (2015) suggest. 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. REEMtree uses the R package RPART for tree building with the addition of a linear model for random effects. The algorithm in the 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.

2.4 *rpms* Algorithm

A relatively new classification algorithm revised 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.

This paper compares the empirical bias and variance of the estimates computed using the listed methods of four outcome variables and the strength of association between the modeled propensities and the outcome variables across methods.

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 mail 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 simple random samples of 2,500 households were drawn from each stratum from this fixed population for a total sample of 10,000 households in a simulation run. Table 1 shows the population size, sampling rate, and response rate for each stratum for the population. The sampling rates were constant across strata.

Table 1: Sampling frame, population size, sampling rate, and response rate by sampling stratum

Region		*Population size*	*Sample size*	*Sampling rate*	*Response rate*
1	Northeast	35,454	2,500	7.10%	28.40%
2	Midwest	44,284	2,500	5.60%	32.30%
3	South	76,633	2,500	3.30%	27.20%
4	West	43,629	2,500	5.70%	24.30%
	Total	200,000	10,000	5.00%	28.05%

Unlike other simulation studies where a response model for the propensity to respond, ϕ, is posited using a set of predictors, in our analysis, the response indicator r is derived by the ACS variable RESMODE, which indicated the data collection mode that was used to collect the household ACS data, as shown in Table 2:

Table 2: Response status definition

Response status r	*Description*	*Definition*	
0	Nonrespondent	RESMODE =1	Household in ACS responded by mail
1	Respondent	RESMODE =2	Household in ACS responded by CATO/CAPI
		RESMODE =3	Household in ACS responded by Internet

For our simulation, those who responded to the ACS by mail were treated as respondents, while CATI/CAPI respondents and internet respondents were treated as nonrespondents. Although this definition 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. On the other hand, one disadvantage is that since the response model is not known, we cannot evaluate if any of these methods can eliminate the response model when the predictors in the response model are available for the creation of the weighting classes.

The four variables selected as the outcome variables in the simulation study are listed in Table 3. The empirical study compared estimates of means and proportions of these outcome variables.

Table 3: Response status definition

Dependent variable	*Description*	*Type*	*Values*
HINC	Household income for the past 12 months	Continuous	
WIF	Number of workers in the family during the past 12 months	Count (truncated)	0 workers, 1 worker, 2 workers, 3 or more workers
HINS	Indicator flag for all members in the household to have health insurance coverage. The flag was created summarized from the person-level health coverage indicator from the ACS person-level file).	Binary	1: yes 0: no
FS	Indicator flag for any yearly food stamp/SNAP recipient in the household	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. Households with predictors with missing values were excluded from the population. Most tree algorithm packages used in the simulation do not handle predictors with missing values.

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

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

3.1 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 equal to their default values.

3.2 *ctree*

The following parameters were set equal 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 equal to their default values.

3.3 *REEM*

The following parameters were set equal 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 equal to their default values.

3.4 *rpms*

- *Bin_size*: the minimum number of observations in a terminal node was set equal 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.
- *P-val:* 0.05.

The following factors were varied:

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

All other parameters were set equal to their default values.

4. Results

Table 4 shows the simulation results for each of the algorithms and the four outcomes; the mean of HINC (household income, past 12 months), the mean of WIF (workers in the family, past 12 months), the proportion of HINS (all household has insurance), and the proportion of FS (yearly food stamp/SNAP recipient in the household). 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

	Outcome Variable							
	HINC		*WIF*		*HINS*		*FS*	
Estimates	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*	*RelBias (%)*	*RRMSE (%)*
rpms unwgt	-5.1	5.5	-2.4	3.2	1.3	1.9	13.9	15.6
rpms wgt	-5.1	5.4	-2.3	3.2	1.3	1.9	13.6	15.2
REEMtree	-4.6	5.4	1.6	3.6	2.2	2.9	11.3	13.5
ctree	-2.0	2.3	0.3	1.6	1.2	1.5	7.7	9.9
SAS CHAID (unweighted)	-9.6	11.2	8.9	11.1	6.1	7.4	17.6	20.1
SAS Entropy (unweighted)	-9.6	11.2	8.9	11.1	6.1	7.4	17.6	20.1
SAS Gini (unweighted)	-9.3	10.8	8.5	10.6	5.6	7.1	17.2	19.7
SAS CHAID (weighted)	-4.7	6.7	3.1	6.3	2.7	4.3	11.4	14.3
SAS Entropy (weighted)	-4.9	6.3	0.1	5.7	2.2	3.7	10.7	13.7
SAS Gini (weighted)	-4.1	5.5	2.3	4.9	2.3	3.5	10.7	13.5

Results for HINC show that estimated relative bias for all options is negative. The three estimators based on the SAS unweighted options had the largest RelBias, while the lone *ctree* option had the smallest bias (- 2.0). The estimators for SAS weighted options, *REEM*, and both *rpms* options have fairly similar performance for the empirical bias. The results of RRMSE were similar in rank to the bias amount with *ctree* performing the best with a RRMSE of 2.3 while the three SAS unweighted estimators had relative biases as high as 11.2 percentage points for CHAID and Entropy.

The estimates of the mean of WIF had a mix of positive and negative relative biases. As with HINC, the three SAS unweighted estimators had the largest RelBias (greater than 8.0). However, while *ctree* gave a small bias of 0.3, the SAS weighted entropy had the lowest bias of 0.1. The two *rpms* methods were the only ones yielding a negative bias. In terms of RRMSE, the results were similar to the HINC *ctree* estimator, with the lowest RRMSE of 1.6. In contrast, the unweighted CHAID-SAS estimators had a RRMSE as high as 11.1. The RRMSE of other estimators did fairly well, each having a value below 4.0.

For the estimates of the proportion HINS, the direction of the relative bias is positive for all estimates. The three SAS unweighted estimators had the largest RelBias as with the previous two outcomes. While the *ctree* estimator had the smallest at 1.2, both *rpms* weighted and unweighted estimators were fairly close at 1.3. The SAS weighted estimators and the *REEM* estimators had similar relative RelBias values. The performance of the empirical RRMSE of the estimators was similar to the performance of the relative empirical bias. The *ctree* and *rpms* estimators had the lowest RRMSE values of 1.5 and 1.9, respectively, and the SAS unweighted estimators had RMSE values as high as 7.4.

When creating estimates of FS, the RelBias was slightly larger in the positive direction. *ctree* had the lowest RelBias at 7.7 followed by SAS weighted options and *REEM*. The SAS unweighted options did slightly worse than the *rpms* options, with a relative bias as high as 17.6. The RRMSE values are largely proportional to that of the RelBias estimates, with *ctree* having the lowest value of 9.9.

It can be seen that over the four outcome variables, the SAS unweighted options consistently performed the worst while *ctree* was consistently the best. It should also be noted that in every case, the SAS Gini option yielded a slightly lower RRMSE than either the SAS CHAID or SAS Entropy options.

Figure 1 displays the bias of each algorithm for the outcome variable FS. The horizontal black line indicates the true population value, and the box plots illustrate the distribution of the bias of each method over the 5,000 simulation runs. Similar to what was discussed from Table 1, the SAS unweighted algorithms do poorly on bias and variance while the *ctree* algorithm does the best.

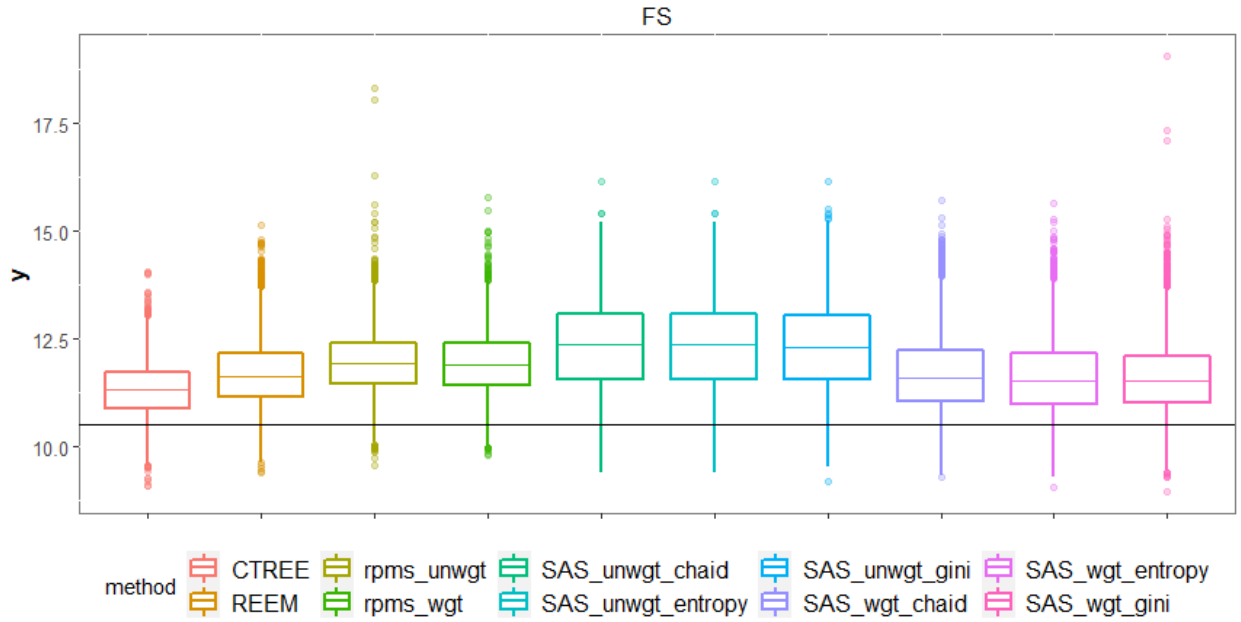

Figure 1: Box plot of bias by method for outcome variable FS

6. Conclusions

Using the 2013-2017 ACS PUMS data as a pseudo-population, we investigated the use of the following six tree algorithms for producing nonresponse classification cells using a simulation study: *rpms*, *ctree*, and *REEM* (each part of R packages), and CHAID, Gini, and Entropy (each of the latter three called by the HPSPLIT procedure in SAS). The *rpms* and SAS algorithms allowed for weighted and unweighted analyses whereas *REEM* and *ctree* did not utilize weights. Our simulation results indicate that incorporating weights in the prediction of response propensity in classification trees outperforms unweighted classification trees for the three algorithms called using SAS's HPSPLIT procedure. However, our results also showed minimal differences for bias and RMSE for all outcome variables between the weighted and unweighted *rpms*. This is likely due to the lack of a clustered design and low variation in design weights. 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. Additionally, the *rpms*, *REEM*, and weighted SAS HPSPLIT results were all fairly comparable. Finally, *ctree* stood out as the algorithm that produced the smallest RelBias and RRMSE for all outcomes compared to the other algorithms.

Our simulation selected repeated samples drawn from a fixed population with a one-stage stratified design with census region serving as the sampling strata. By using the ACS PUMS as our fixed population, we

were able to mimic a national household-level mail survey and introduce a nonresponse mechanism that allowed for comparisons between estimates and true population values.

Simulation results may be different for more complex 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. There is reason to believe that *rpms* and *REEMtree* may perform better under a clustered sample design due to the usage of area effects. A further step would be to test these algorithms under a clustered design framework.

Another limitation to 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 a cluster stage in the sample design, the evaluation of the algorithms would also benefit from inclusion of person-level estimates in the simulation design.

7. Acknowledgments

The authors are grateful to Tom Krenzke, Bob Fay, David Morganstein, Jeri Mulrow, Jill Dematties, Yuki Nakamoto, and John Riddles for the insightful suggestions.

8. References

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

Brick, J. M., and J. Montaquila. 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.

Hothorn, T., K. Hornik, C. Strobl, and A. Zeileis. 2017. "Party: A Laboratory for Recursive Partytioning." Version 1.3-1. http://cran.r-project.org/web/packages/party/

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 I. Flores Cervantes. 2019. A modeling approach to compensate for nonresponse and selection bias in surveys? In *JSM Proceedings.* Denver, CO: American Statistical Association. 827-834.

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

Lohr, S., V. Hsu, and J. Montaquila. 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* 1:1, 81-106.

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

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

Toth, D. and P. Phipps. 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/web/packages/rpms/index.html