FINAL PROJECT FOR THE INTP ESTIMATION FOR THE STATE OF VIRGINIA

ANUJ AMIN

STAT 443

Contents

Abstract

The ACS data is a survey conducted by US Census Bureau. Each year, it gathers information such as ancestry, language, education, housing characteristics and income. This data is used by many lawmakers, non-profit organizations, private and public organizations to assess ways to efficiently allocate money, create policies, and track community changes (American Community Survey Information, 1). Given this information, the goal of this analysis was to design, create and compare the various statistical learning models to estimate the mean INTP (interest, dividends, and rental income) with sampling weights (PWGTP) based on over 120 variables. Because the true data set is around 3.5 million samples, we are using the Public Use Microdata sample for the personal records for the state of Virginia. This dataset is similar to the Census Bureau one, but it is a random sample of data for individuals in the state of Virginia. There are additional variables included to give us more information about the people of Virginia, and to provide confidentiality, some of the variables have been top coded. While most models were able to compute an estimate for INTP, they came with many issues. Some of these issues include hardware problems, poor interpretability and being poor at estimating INTP. It is clear that GUIDE was the easiest to model to compute the INTP with and interpret the results. The GUIDE algorithms created easy to interpret plots and had accurate results. To replicate these results and to see more details about the code, refer to Appendix S for more information.

Introduction

The US Census Bureau's American Community Survey (ACS) is a nationwide survey that collects and provides information on social, economic, housing and demographics on the US population every year. It is an important tool to track the changes in our communities. Each year, the Census Bureau contacts over 3.5 million households across the country to participate in the ACS. The ACS is an integral part of the US constitution because it helps legislators decide how and where resources can be allocated in an effective and efficient manner. The benefit of having a long form census is that lawmakers are constantly updated with a data on states and local areas to better help access community needs (American Community Survey Information, 1). It is important to participate in the ACS as it aids in accessing needs, planning, research, education, and advocacy work on a local and federal level (American Community Survey Information, 2). ACS data extends beyond the government as well. While it is true that federal, state, and local agencies use this data, the census data also has an effect on other nongovernmental organizations like businesses, emergency planners, nonprofit organizations, community groups and even the general public (American Community Survey Information, 4-5).

In order to collect the data in an unbiased manner, the Census Bureau selects a random sample of addresses to be included in the ACS, each having a 1 in 480 chance of being selected. Roughly 295,000 questionnaires are sent a month across the US to further randomize the sample. Additionally, no addressed is selected more than once every 5 years. In order to assure that each household is filling out the form, members of the census bureau also may also conduct visits to follow up on the nonresponse addresses and conduct telephone follow ups on for the incomplete questionnaires (American Community Survey Information, 6).

The application for this analysis is on the 2019 ACS PUMS (Public Use Microdata Sample) for the Person records on the state of Virginia. The goal of this file is to create estimates for user defined characteristics while protecting the confidentiality of the personnel. This file is very similar to the original ACS survey, but it includes additional variables to provide more information on estimating the user defined characteristics. Each record for the Personal Virginia dataset contains information about a single person. Individuals are organized into households in order to understand the relation between other family members. Additionally, the file includes people who reside in group quarters (i.e., nursing homes or college dormitories) (AMERICAN, 3). Because the PUMS data reflects only one percent of the US population, many of the variables such as birthplace and income have been top coded to maintain confidentiality. Although this dataset is a microsample, it contains 84939 observations. Additionally, many of response variables have missing data because it was not applicable to the person filling out the form. For example, DRATX has 61893 of the responses missing because a majority of the people have not served in the military. This occurs for a majority of the variables (refer to Appendix B, Clean Data Summary for more information). A majority of these variables area categorical while only 15 are numeric (see appendix A for more information).

The variable that we are interested in estimating is INTP. According to the ACS PUMS dictionary, INTP is the interest, dividends, and net rental income of an individual over the past 12 months. If the individual is less than 15 years old, INTP is encoded as NA. If there is no rental income, INTP is written as 0. In order to maintain confidentiality, INTP has been top and

bottom coded. If INTP is negative, the value ranges from -$10,000 to -$4 (rounded and bottom-coded). In contrast, if INTP is positive it ranges from $4 to $999999 (rounded and top coded) (2019 ACS, 36). For the classification algorithms, there is a variable FINTP that serves as an indicator if INTP was recorded or not. If INTP is NA, then FINTP is 0. If FINTP is 1, then it means than the INTP variable was imputed or corrected (2019 ACS, 123). Because of this issue, it was recommended by Professor Loh to undo those imputations those values as shown in appendix A. The total number of observations where INTP is unknown is now 11238 (appendix B, Clean Data Summary).

In this analysis, I use a myriad of models to estimate the mean of the INTP (interests, dividends, and rental income) with sampling weight w_i in variable PWGTP. The methods used are: GUIDE regression and classification trees, GUIDE regression and classification forests, RPART regression and classification trees, CTREE regression and classification trees, logistic regression, randomForest regression forest, and CForest regression forest. First these models are created and INTP is estimated, then their corresponding prediction capabilities are compared based on their prediction accuracy of the mean INTP.

Statistical Analysis

Prior to model building, data preprocessing occurred to reduce the total bloat of the data, and to impute missing data. Many of the variables that were in the initial dataset were removed because they were unnecessary to our goal of estimating mean INTP. Additionally, visualizations and data analysis were produced to understand the relationships between the various predictor variables and the response variable, INTP. After the visualizations, analysis was done to understand the relationships and the significance between INTP and the respectable predictor variables.

Data Preprocessing

Prior to conducting writing any code, I removed variables with no predictive power from the CSV file. The constants that I removed were ADJINC, DIVISION, REGION, RT, and ST. Because we were looking only at one state, many of these variables would be the same, so it was useless to keep these variables. I then removed SERIALNO and SPORDER because for this analysis, I did not need to uniquely identify any observations. I proceeded to remove the replicate weight variables PWGTP1-80 because no confidence intervals were created for this analysis. Finally, because we were only concerned with predicting INTP, the only flag variable that was kept was FINTP. For convenience, the removal of all variables was done in Excel.

I then proceeded to import the csv file into R and proceeded to export it into a data file called cleandata.txt. However, it was discovered later that the Census Bureau had imputed some of the data because INTP is blank if the respondent was less than 15 years old (appendix A). Hence, some of my code and all of my visualizations have been updated to reflect these changes. The biggest change was the reduction in size of the dataset. After retracting the imputations done by the US Census Bureau, the dataset was reduced to roughly 71,000 observations from about 84,000 observations.

I used GUIDE to create two subsets for the data. The first subset was created using the classification description file (see appendix I), and it was used for all the non-GUIDE algorithms that were used to estimate the mean using the IPW method (see appendix T, IPW Method). The second subset was created based off the regression description file (see appendix J), and it was used for all the non-GUIDE algorithms to estimate mean INTP via the imputation method (see appendix T, Imputation Method).

Because this dataset had many variables, I had to use feature selection to reduce the total number of variables in order to make the logistic regression model. In order to accomplish this, I used the GUIDE importance scores to reduce the model (see appendix G). Additionally, I used MICE to impute missing values (see appendix O, MICE Imputation Code). Imputations were done for step 4 of the analysis.

Because the randomForest does not allow any missing values, and many of the predictors had missing values, GUIDE importance scores for regression were used for feature selection in order

to reduce the number of predictors in the dataset. I then used Amelia for imputation of the missing values (see appendix O, Amelia Imputation). Finally, I dropped the variables MARKHYP and FOD1P because they had 80 and 173 levels respectively, from the imputed dataset as it caused performance issues. Removing these variables, caused the code to run without crashing (refer to appendix Q).

Importance Scoring

Because GUIDE important scoring is integral to many of the algorithms that require feature reduction, it is important to give a brief overview of how it was used. GUIDE important scoring is a very simple way to rank the most important variables to the least important ones. This is crucial where feature selection is needed. Feature selection is need for many of the algorithms such as logistic regression and randomForest. For added convenience, there is a cutoff for categorizing which variables are highly important, likely important and unimportant. In both the regression and classification important scores, all default settings were used to create the scores. Important scores were taken from unpruned trees. For the classification important scores, variables with a score above 1.2 were considered highly important, scores below 1.0 were unimportant and everything between 1.0 and 1.2 was considered likely important (appendix G, GUIDE Classification Scoring Output File). The regression scores were taken from a least square regression tree. Variables with scores above 1.15 were considered highly important, variables with scores between 1.0 and 1.15 were considered likely important and scores below 1.0 are considered unimportant (appendix H, GUIDE Regression Scoring Output File). Although scoring took a while to complete, it saved a lot of time compared to other methods such as trial and error.

Proposed Methods

For this analysis, a variety of different methods were used and compared to estimate INTP with weight variables: GUIDE classification tree and forest, GUIDE regression tree and forest, CTREE classification tree and regression tree, RPART classification tree and regression tree, Logistic Regression, randomForest and CForest. From these choices, all classification algorithms will compute an estimated mean INTP via the inverse probability weighted (IPW) method (appendix T, IPW Method). Any algorithm that uses regression will compute an estimated mean INTP via the imputation method (appendix T, Imputation Method).

The best methods for this analysis are the ones that have the best classification accuracy and actually produce a valid answer. The more accurate the model is on this data set, the better we are able to predict mean INTP and interpret its results. Additionally, it is important that we are able to compute the mean INTP and our results are interpretable. Below are the breakdowns of each method and their corresponding results.

GUIDE Classification Tree

The GUIDE classification tree is a powerful and easy to use way for obtaining predicts on of a discrete variable. In the case of classification, GUIDE predicts the variable FINTP, which is a flag variable indicating if somebody has responded to having INTP (1) or not (0). Compared to other methods learned in class, GUIDE provides unbiased, linear splits. GUIDE provides unbiased variable selection, even if there is missing data! Another important part of this algorithm is that it provides interaction tests, and it prunes the tree (User Manual, 2)! The benefit of pruning the tree is that our tree size is reduced, leading to less overfitting. Therefore, we can use our tree to predict on data that is not in the training data. While we do sacrifice some accuracy using a classification tree, it is the most interpretable tree. This means that we can learn from it the most.

A GUIDE classification tree was constructed for step 1. The classification description file was used for this dataset (see appendix A). For this tree, all the default options were used in order to get an estimate of π (probability of response). This meant that this tree was a simple model relying on one variable to split each node and that 10-fold cross validation was used to prune the tree (appendix C, GUIDE Classification Tree Input File). When we look at the results of the tree, we have a cross validation mean cost error rate of 9.089E-02 which is really low considering the amount of missing values in the data. While this error rate is impressive, we will need to do leave-one-out cross validation in order to properly access the error rate. Because the tree was pruned to avoid overfitting, a downside of using this model is the relative time it took to construct the tree which was about 454 seconds (appendix C, GUIDE Classification Tree Output File). That being said, I argue that the accuracy far outweighs the con of a relatively long time to construct a classification, especially when we factor in the size of the dataset.

Then, the IPW method was used to estimate mean INTP. Our result showed that the mean INTP for the valid responders (when FINTP = 1) was $2292.50. This number seems reasonable when we look at the INTP distribution (appendix B, INTP Distribution). A majority of people do not have any INTP, so it seems reasonable, that the average INTP is relatively closer to 0.

One of the benefits of the GUIDE classification tree is how easy it is to read and interpret the results. GUIDE creates a nice binary tree where we can learn how to classify if somebody has INTP or not (appendix C, GUIDE Classification Tree). If somebody had no idea how to read the tree, GUIDE generates a caption under the tree telling the new user how to interpret the results and gives an overview of the tree. The summary contains important information ranging from the number of observations in the dataset to the second-best split variable. When looking at the tree we can see what variable a split occurred. For example, the root split occurs if WRK is NA. If WRK is not NA, the tree predicts that the INTP was not reported. Otherwise, the algorithm checks other variables in a similar fashion in order to classify if INTP was reported or not. At each terminal node, the tree shows the probability of FINTP and the number of observations in each terminal node. Additionally, each terminal node is color coded in order to improve readability. This can be important for the user in order to better learn about the population. This tree can be used to classify on a data that was not in the training model. After all, the misclassification cost for this model on this dataset was 9.089E-02!

GUIDE Classification Forest

The GUIDE Classification Forest is another powerful and easy to use way to predict a discrete variable. Again, in our case this is the variable FINTP. The idea of a classification forest is similar to the single classification tree. However, an ensemble of trees is used. This algorithm fits multiple, in this case 500, unpruned trees and proceeds to randomly select a small subset of variables to search for splits at each node, The GUIDE forest algorithm uses an unbiased algorithm to split the nodes and is often times faster. Additionally, GUIDE forests do not need any imputation (User manual, 300)! While we do lose some interpretability, we do gain more accuracy. This is because we aren't given a tree diagram where we can interpret our results and test other observations with our model. Another downfall to consider the resources used. In the case of the GUIDE forest, the process took a little under an hour, while the single tree took a couple of minutes.

A GUIDE Classification Forest was also constructed for step 1. The classification description file was used for this dataset (see appendix A). For this forest, almost all of the default options were used to get an estimate of π. I chose to split on nonrandom variables for missing values, and in order to save on resources I skipped interaction tests (appendix D, GUIDE Forest Input File). The time it took to run the was about 3185 seconds. While that certainly uses a lot more resources and takes a bit longer to compute results it usually is more accurate than a single tree. Additionally, to save time, the trees are not pruned, while by default, for a single tree, GUIDE does 10-fold cross validation. When comparing the mean misclassification cost with a single tree, there is marginal performance improvement. The out of bag mean misclassification cost was .0873, while the misclassification cost for the tree was 0.0909. Therefore, the forest was about 3.6% more accurate for a single dataset (Appendix D, GUIDE Forest Output File). In order to better access accuracy, leave-one-out cross validation would need to be computed because we are only looking at a single dataset here. While we do increase accuracy here (for this dataset), it is only marginal. Therefore, one can argue that the slight increase in model performance may not be worth it when you factor in resources used.

Our result showed that the mean INTP for the valid responders (when FINTP = 1) was $2019.17. Prior to retracting the imputations as announced by Professor Loh, the mean INTP was similar. In fact, the code was identical and the mean INTP was $2118.6 (Appendix D). However, the updated number (2019.17) is more accurate because it removes any imputations done by the US Census Bureau. Additionally, because of the increased accuracy prediction accuracy of the classification forest, one can argue that this number obtained may be more accurate than the classification tree.

GUIDE Regression Tree

The GUIDE regression tree some of the same benefits as its classification tree algorithm: you don't need any imputations, you have unbiased splits, class priors, pruning etc. However, a regression tree focuses on predicting a continuous variable, in this case INTP. Because the terminal nodes aren't true or false, but instead are a number (proportion or total number of

predicted observations), we lose some interpretability and classification features. That being said, regression trees are still more interpretable than a forest and still have some improved accuracy over the alternatives.

A GUIDE Regression Tree was also created for step 2. The regression description file was used for this dataset (see appendix A). For this regression tree, the default GUIDE settings were used to estimate $\hat{y}$ (INTP). This means that tree uses constant linear regression with interaction tests. In order to prune the tree, 10-fold cross validation is used in order to avoid overfitting (appendix E, GUIDE Tree Input File). While accuracy is sacrificed in the regression tree model, we are saving on resources when compared to using a tree. In this case, the time it took to run the tree was about 333 seconds (appendix E, GUIDE Tree Output File). Additionally, we do retain some interpretability when we use a tree which is important when we want to make decisions by ourselves in order to predict someone's INTP.

When we look at the output of the regression tree, the mean squared error is rather high. It has a value of 2.989E+08. This can be an indicator of high variance or a high biased estimate. Part of the problem may be from having a dataset with a lot of missing values. Even when we look at the R^2 value it indicates that 46.6% of the variation in INTP can be explained by the regression tree model (appendix E, GUIDE Tree Output File). It is worth noting that R^2 is not the best indicator in model performance, because R^2 values are biased. A poor model can have a high R^2 value, and a good model can have a low R^2 value! In order to better access the model's performance, leave-one-out cross validation will have to be performed so that the error rates can be compared with the other methods.

Then the imputation method was used to estimate mean INTP (appendix T, Imputation Method). Our result showed that the mean INTP for the valid responders was $2324.75 (appendix E). This number seems reasonable when we look at the INTP distribution (appendix B, INTP Distribution). A majority of people do not have any INTP, so it seems reasonable that the average INTP is relatively closer to 0. However, when compared to the IPW method, this number is higher and may or may not be a bit less accurate, as more explanatory analysis will have to be done and the true mean INTP is unknown.

Although a regression tree is a bit more difficult to interpret, the GUIDE regression tree is still fairly easy to read. GUIDE creates a nice binary tree where we can learn how to predict someone's INTP (appendix E, GUIDE Regression Tree). If somebody had no idea how to read the tree, GUIDE generates a caption under the tree telling the new user how to interpret the results and gives an overview of the tree. The summary contains important information ranging from the type of regression tree to whether to go left or right. When looking at the tree we can see what variable a split occurred. For example, the root split occurs if PINCP $\leq$ 260550. If PINCP is greater than 260550, the algorithm goes right and checks the WAGP value and so on. The algorithm checks other variables in a similar fashion in order to predict INTP. Each terminal node of the tree gives the number of observations and the predicted mean value of INTP for each of those observations. To improve transparency, terminal nodes are colored based on a threshold. This can be important for the user in order to better learn about the population. Additionally, this tree can be used to predict INTP on data that was not in the training model.

GUIDE Regression Forest

The GUIDE Regression Forest yet another algorithm that is included in GUIDE. It is powerful and easy way to predict a continuous variable, in this case INTP. The algorithm is the nearly the same as the classification forest, except that now uses an ensemble of unpruned regression trees. The benefits and tradeoffs are both the same for a classification forest.

A GUIDE Regression Forest was also created for step 2. The regression description file was used for this dataset (see appendix A). For this forest, almost all of the default options were used to get an estimate of $\hat{y}$ (INTP). I chose to split on nonrandom variables for missing values, and in order to save on resources I skipped interaction tests (appendix F, GUIDE Forest Input File). The time it took to run the was 13867 seconds. While that certainly uses a lot more resources and takes a bit longer to compute results it usually is more accurate than a single tree. Additionally, to save time, the trees are not pruned, while by default, for a single tree, GUIDE does 10-fold cross validation. When comparing the mean misclassification cost with a single tree, there is some performance improvement. When we compare the R^2 values, the current model explains 53.61 percent of the variation in INTP. That is a 7% increase in compared to the regression tree. Again, the R^2 value may not be the best indicator in performance. Additionally, the current model has an out of bag mean squared error of 1.981E+08. In comparison, the regression tree had an MSE of 2.989E+08. That is a drastic improvement for the current dataset (Appendix F, GUIDE Forest Output File). The MSE value is about 35.5% lower than the regression tree counterpart! Further data exploration will need to be done with leave-one-out cross validation in order to properly access the performance of the model. While we are using a significant more amount of time and resources, it is likely that we have significantly improved performance. Therefore, one can argue that the increase in model performance is worth it even if time and resources used are taken into consideration.

Our result showed that the mean INTP for the valid responders was $2302.425. Prior to retracting the imputations as announced by Professor Loh, the mean INTP was drastically larger. The calculated value before retracting the US Census Bureau's imputations was $12,295.47. In fact, the code was even identical (Appendix F). It is clear that the updated number ($2302.425) is more accurate because it removes any imputations done by the US Census Bureau and it drastically reduced the INTP value to be more similar to the other methods. Additionally, because of the increased accuracy prediction accuracy of the classification forest on this dataset, one can argue that this number obtained may be more accurate than the regression tree.

RPART Classification and Regression Tree

RPART is the CART algorithm implementation. One can say that is a precursor to more advanced algorithms like GUIDE. RPART has less features than GUIDE. For example, there are no unbiased splits, no interactive tests nor are there any options to create forests. However, I argue that the biggest con with using R is based on how it finds the best split. The CART algorithm has selection bias, because it is more biased towards selecting an X with more splits, more missing values. Additionally, it is biased toward selecting surrogate variables with fewer

missing values. Another reason why you may want to choose another algorithm is that CART becomes really inefficient when working with a high number of ordinal and categorical splits. If you are using a CART to create a regression tree, it is limited to only piecewise constant regression trees (STAT 443, 188).

A RPART Classification Tree was constructed for step 3. For the classification tree, again the default settings were used. Because this was a classification tree, FINTP was predicted while variables INTP and PWGTP were omitted. The dataset used for this model was a classification subset that was created from the cleandata.txt file with GUIDE (appendix I). Compared to GUIDE, the tree runs much faster. While the GUIDE classification tree took about 454 seconds, the RPART classification tree took less than 20 seconds. While this is impressive, the accuracy of our model takes a bit of a hit. We have an error rate of about 12.13% for a single dataset which is higher than all of the GUIDE models (appendix K, RPART Confusion Matrix). While we are saving on time and resources, we have an error rate that is roughly 3-4% bigger. In order to better compare the error rates, leave-one-out cross validation will need to be computed in order to properly compare each of the methods.

Then the IPW method was used to estimate mean INTP. Our result showed that the mean INTP for the valid responders (when FINTP = 1) was $2364.89 (appendix K, RPART Classification Tree Code). This number seems reasonable when we look at the INTP distribution (appendix B, INTP distribution). A majority of people do not have any INTP, so it seems reasonable, that the average INTP is relatively closer to 0. That being said, compared to any of the GUIDE models, which have higher accuracy, our mean INTP value is slightly higher. This may indicate that this number is slightly inflated. Prior to professor Loh's announcement where the US Census Bureau's imputations were retracted, this number is even more inflated with an IPW estimate of $2540.95. The code is to reproduce this output was identical to what was previously shown.

Trying to learn from this model is extremely difficult. The tree is very basic. It's not color coated nor is there any sort of description that is given like GUIDE does. Even I am a bit confused on the interpretation of the model looking at this. For example, the split at the root of the tree is a bunch of Z's (appendix K, RPART Classification Tree). In order better interpret the results, you would have to look at the text format of the RPART tree which contains information on each split (refer to Appendix K, RPART Classification Text Format for more information). If someone is not trained in data analysis and using RPART they will have a difficult time trying to learn from this model and predicting other observations not in this dataset. Hence, GUIDE is superior in its classification and tree visualizations.

A RPART Regression Tree was also constructed for step 3. The default settings were used for the regression tree. Because this was a regression tree, INTP was predicted while the variable FINTP was omitted. The dataset used for this model was a regression subset that was created from the cleandata.txt file with GUIDE (appendix J). Compared to GUIDE, the tree runs much faster. While the GUIDE classification tree took about 333 seconds, the RPART classification tree took less than 20 seconds. While this is impressive, the accuracy of our model may take a bit of a hit on this dataset, but more exploratory data analysis (leave-one-out cross validation) would need to be conducted to compare each model. If this were the case, this would likely be due to the combination of RPART having biased splits and numerous variables having a high number of levels.

Then the imputation method was used to estimate mean INTP. Our result showed that the mean INTP for the valid responders was $2378.52 (appendix L, RPART Regression Tree Code). This number seems reasonable when we look at the INTP distribution (appendix B, INTP Distribution). A majority of people do not have any INTP, so it seems reasonable that the average INTP is relatively closer to 0. That being said, compared to any of the GUIDE models, which likely have a higher accuracy, our mean INTP value is slightly higher. This may indicate that this number is slightly inflated. Prior to Professor Loh's announcement where the US Census Bureau's imputations were retracted, this number is actually smaller with an imputed estimate of $2066.16. The code is to reproduce this output was identical to what was previously shown.

Trying to learn from this model is extremely difficult. Like the classification tree, it is very basic. It's not color coated nor is there any sort of description that is given like the GUIDE Regression Tree does. Even I am a bit confused on the interpretation of the model looking at this. For example, the third split of the tree is unreadable due to the amount of overlap (appendix L, RPART Regression Tree). In order better interpret the results, you would have to run the tree in base R to get each split and the results properly (Appendix L, RPART Regression Text Format). That can be very tedious. If someone is not trained in data analysis and using RPART they will have a difficult time trying to learn from this model and predicting other observations not in this dataset. Hence, GUIDE is superior in its regression prediction and tree visualizations.

CTREE Classification and Regression Tree

When comparing the features of CTREE, it can be easy to see its shortcomings when compared to GUIDE. Yes, it has unbiased splits like GUIDE, which is something that CART does not have. However, it does not include interaction tests (User Manual, 2-3). When using the party package, we class priors were not allowed, and weight variables must be integers (STAT 443, 87). However, when using CTREE with partykit, it is even more limited. For example, categorical variables must not exceed 31 levels and there cannot be any variables with NAs. The biggest problem I had with CTREE using the party package is that the results were almost unreadable when dealing with a large dataset. This will be shown later in the analysis when attempting to interpret the results.

A CTREE Classification tree was also constructed for step 3. For the classification tree, the default settings were used. Because this was a classification tree, FINTP was predicted while variables INTP and PWGTP were omitted. The dataset used for this model was a classification subset that was created from the cleandata.txt file with GUIDE (refer to appendix I for more information). Although CTREEs generally run faster, this was not the case for this dataset. I took a subset that was roughly 15% of the size of the dataset (11,000 observations). This was because of memory allocation issues that CTREE was causing. Even while taking a subset, the CTREE took roughly the same time as the GUIDE classification tree did.

Then the IPW method was used to estimate mean. Our result showed that the mean INTP for the valid responders (when FINTP = 1) was actually NaN (appendix M, CTREE Classification Code). This number means that a majority of the predictions were zero, so using a CTREE is not

a good option for predicting INTP via the IPW method. One work around for this is to impute the predicted values with a small number. This number could be an arbitrary small number, or one could impute the zeros with the minimum predicted value. That being said, I argue that this may severely change the results, and is not good practice to change the predictions that our model generated. The same result would appear when running the code prior to Professor Loh's announcement of retracting the US Census Bureau's imputations. Because of this issue with the predictions, I was not able to produce a confusion matrix to compare to the GUIDE model on the same dataset, so more analysis must be done to compare the classification tree model.

When looking at the CTREE, it is much more readable than the RPART counterpart. However, it is still inferior to GUIDE. One reason is that it is much more cluttered and verbose than the GUIDE counterpart. This is likely due to CTREE's being unpruned. Another issue is interpreting the terminal nodes. The results for them are all overlapping so we cannot see the results (appendix M, CTREE Classification Tree). If somebody were to look at this, they would have a hard time with trying to classify if someone responded to having INTP or not. Even for somebody who has experience with CTREEs, they would have to look at the text output of the tree which is very verbose and complex (Appendix M, CTREE Classification Tree Text Format). Hence, while this tree is a bit more readable than the RPART tree, it still falls flat when compared to GUIDE.

A CTREE Regression tree was also constructed for step 3. The default settings were used for the regression tree. Because this was a regression tree, INTP was predicted while the variable FINTP was omitted. The dataset used for this model was a regression subset that was created from the cleandata.txt file with GUIDE (appendix I). Compared to GUIDE, the tree took roughly the same time. The GUIDE regression tree took about 333 seconds, the CTREE regression tree about 5 and a half minutes. However, using the full dataset caused performance issues stemming from an excess number of predictor variables and variables with a high number of levels. So, in order for the algorithm to finish running, a sample of the data was used in order to create a tree (appendix N, CTREE Regression Code). Like RPART, CTREE does not give an R^2 value, or any misclassification cost so more exploratory data analysis would need to be conducted in order to access the accuracy of the regression tree. In order to do this, the leave-one-out cross validation should be used in order to access the accuracy over multiple datasets.

Then the imputation method was used to estimate mean INTP. Our result showed that the mean INTP for the valid responders was $2107.75 (appendix N, CTREE Regression Code). This number seems reasonable when we look at the INTP distribution (appendix B, INTP Distribution). A majority of people do not have any INTP, so it seems reasonable that the average INTP is relatively closer to 0. Because the number is closer to zero, this may mean the number is more accurate, but again, more data analysis would need to be conducted in order to determine this. Prior to Professor Loh's announcement where the US Census Bureau's imputations were retracted, this number is even more inflated with an imputed estimate of $2691.58. The code to reproduce this output was identical to what was previously shown.

When looking at the CTREE, it is actually unreadable, and nobody can learn from this diagram. It is even more unreadable than its RPART counterpart and it is definitely inferior to the GUIDE counterpart. The tree is way more cluttered and verbose than the GUIDE and RPART counterpart (appendix N, CTREE Regression Tree). This is likely due to CTREE's being unpruned. Another issue is interpreting the terminal nodes. You cannot even see them! Even

trying to get the text format of the tree was futile because I was only able to get a partial tree. This is due to the tree output being so verbose that it actually got cut off in the R console (appendix N, CTREE Regression Tree Text Format Preview) If somebody were to look at this, they would likely give up in trying to predict somebody's INTP. Hence, this tree falls flat compared to GUIDE and RPART.

Logistic Regression

Logistic Regression is a very simple method used to predict a binary outcome. It's easy to implement, you can get quick predictions, and it is easy to find the most valuable predictors for your model. However, there are numerous faults with this algorithm, especially when working with a dataset of such a high dimension. For starters, we have over a hundred variables being used, so a model with interactions is impossible. If there are no interactions, the model won't converge. Another fault is that the features must be selected before fitting the model. Hence GUIDE important scores were used to reduce the dimensionality of the data. The biggest fault occurs when it comes to learning from the results of the model. With a tree you are able to come to an absolute decision based on yes or no questions. However, with the logistic regression model, it is likely that the probability of a decision is never 1. When we are able to get a decision with yes or no questions leading to the leaf node, we can learn what the important features are and how the algorithm arrives at the decision. But when a logistic regression model is used, we're just plugging in a bunch of values to get the odds of an event happening. We cannot learn how to arrive at a decision. Additionally, in order to improve simplicity, we have to omit several variables in order to make the logistic model more interpretable, and it is still very difficult to interpret!

The logistic regression model was constructed for step 5 of the analysis. Because the high dimension of this dataset, many variables were dropped in order to increase the interpretability of the model. The GUIDE important scores for classification were created (appendix G) and used to reduce the dataset. After getting the reduced dataset, MICE was used to impute missing values (appendix O, MICE imputation Code). Using MICE was step 4 of the assignment. After the imputation, the model was created with all of the variables included. The time it took to create this model was negligible. It was done in an instant. Then, the IPW method was used to estimate mean INTP (appendix P, Logistic Regression Code). Our result showed that the mean INTP for the valid responders was $2670.09. This number seems fairly large when we compare this value to the other models created. Again, more data analysis would need to be conducted in order to determine how accurate our model really is. We would need to perform a leave-one-out cross validation in order to access the accuracy properly. This number is likely less accurate because compared to the other algorithms, logistic regression is a fairly simple one and tends to be used as a baseline to compare other models to. Prior to Professor Loh's announcement where the US Census Bureau's imputations were retracted, this number is even more inflated with an imputed estimate of $3497.53. The code to reproduce this output was the same to what was previously shown.

CForest

CForest is an extension of the CTREE algorithm. CForest is an ensemble of CTREEs built from bootstrap samples. In fact, they share many of the same pros and cons. One pitfall of a CForest is that prediction fails on categorical predictor values not in the training Additionally, class priors, unequal misclassification costs, and weight variables are not allowed in the data. When using the party package, class priors were not allowed, and weight variables must be integers. However, when using CForest with partykit, it is even more limited (STAT 443, 87). For example, categorical variables must not exceed 31 levels and there cannot be any variables with NAs. The final issue is performance. There is an increase in accuracy, but CForest is very slow. In fact, it was so slow that it had to be ran on a random sample of the population in order to get any results (appendix R, CForest Code).

A CForest was created using the party package (not partykit) in order to compute the mean INTP via the imputation method for task 6. The default settings were used for the CForest. INTP was predicted, while PWGTP and FINTP were excluded from the model. The dataset used for this model was a regression subset that was created from the cleandata.txt file with GUIDE (appendix J). Additionally, because of memory allocation issues, a random subset was taken in order to compute the CForest. On top of that, because NAs in the predictor are not allowed, any observations with NAs in the dependent variable were excluded. The forest took roughly an hour to compute, which is pretty fast compared to the time it took the GUIDE regression forest. This is in part due to the reduced subset size. All memory allocation issues likely stem from this dataset having numerous predictor variables and variables with high level of factors. Compared to GUIDE and randomForest, the CForest text output is very minimal. There is no R^2 value or mean of squared residuals given (Appendix R, CForest Summary) for the results of this dataset. In order to access accuracy, exploratory data analysis would need to be conducted with the leave one out cross validation method across multiple datasets. The mean INTP computed was $2284.18, which is right in line with the other observations (appendix R, CForest Code).

randomForest

Compared to CForest and GUIDE regression forest, the randomForest algorithm runs much faster. No random sample of the population was needed in order to compute a forest. The pros and cons of using a randomForest are similar to any other forest algorithm. Although we are gaining accuracy compared to a single tree, we lose interpretability. The biggest cons of the randomForest are that NAs and sampling weights are not allowed (STAT 443, 87). Because many of the columns contain missing data, the dimensions of the data were reduced and then Amelia was used to impute the missing data. The top 15 variables from the GUIDE importance scores were selected here (appendix O, Amelia Imputation Code).

A randomForest was also constructed for step 6 of the analysis. In order to create the model, after all imputations were done, the variables FOD1P and MARHYP were then dropped due to performance issues stemming from too many levels in the dataset. Both these variables had 80 and 173 levels and there wasn't enough memory and processor power to handle the amount of

data. Because randomForest cannot predict on a missing INTP value, all the missing INTP values were omitted from the imputation calculation. The estimated mean INTP via imputation was $2305.64 (appendix Q, Random Forest Code). Compared to CForest, one way we can assess how well our model did on the current dataset is to look at the randomForest summary. Compared to GUIDE, our model actually performed a bit better when we look at the R^2 value or mean of squared residuals. The randomForest model had an R^2 value of .9066 which is much higher than the GUIDE model and had a MSR value of 3.882E+07. That being said these metrics are fairly limited and can be biased, so leave-one-out cross validation will have to be conducted in order to compare the result with the other methods.

Conclusion

ACS data is an important part of our government and daily life. It is something that we take for granted. Having the ability to properly access our community needs helps us be better prepared for an uncertain future and helps give assistance to communities in need. ACS Data is especially useful for learning about the country's demographics. In this case, I explored the INTP (Interests, Dividends and Net Rental Income) for the state of Virginia.

For the most part, I was successfully able to estimate the mean INTP (Interests, Dividends and Net Rental Income) with the help of GUIDE and various other algorithms. However, algorithms other than GUIDE proved to be problematic in the estimation. This was due to the PUMS data for the state of Virginia having numerous amounts of missing data. To combat this issue, for non-GUIDE models, MICE and Amelia were used. While this is nice, MICE and Amelia clearly have its limitations. Because of the dimension size, feature selection had to be done in order for imputation to work. This meant that lots of data was removed, which changes the true result. Additionally, many of these algorithms like CForest require a lot of resources meaning that we can only run them on a sample of the dataset which further skews our results. Thankfully, GUIDE was able to create importance scores for regression and classification, so feature selection was very simple to do. This saved a lot of time and resources. This project showed that while there were many ways to predict the mean INTP, most algorithms had their fair share of issues. Yet, GUIDE was able stand out and be the most reliable. Without GUIDE, we wouldn't be able to run most of the other algorithms. Therefore, while more analysis would need to be done in order to determine the accuracy of the models, GUIDE is clearly the winner due to its ability to handle missing data, give importance scores, and legible tree plots for interpretation.

Appendices

Appendix A

R Code

```
# Anuj Amin
# Final Project STAT 443

rm(list = ls())

#Import dataset and remove INTP imputations done by census bureau
z = read.csv("psam_p51.csv",na.strings="")
z = z[!is.na(z$INTP),]
z$INTP[z$FINTP == 1] = NA
write.table(z,"cleandata.txt",row.names=FALSE,col.names=TRUE)

#Creating the description file (regression)
dat <- read.table("cleandata.txt",header=TRUE)
nvar <- ncol(dat)
varnames <- names(dat)
roles <- rep("c",nvar)

n.vars <- c("PWGTP","AGEP","JWMNP","OIP", "PAP", "RETP", "SEMP", "SSIP",
"SSP", "WAGP", "WKHP", "WKWN", "PERNP",
            "PINCP", "POVPIP")
roles[varnames %in% n.vars] <- "n"

d.var <- "INTP"
roles[varnames %in% d.var] <- "d"

x.var <- "FINTP"
roles[varnames %in% x.var] <- "x"

write("cleandata.txt",file="descreg.txt")
write("NA",file="descreg.txt",append=TRUE)
write("2",file="descreg.txt",append=TRUE)
write.table(cbind(1:nvar,varnames,roles),file="descreg.txt",
            row.names=FALSE,col.names=FALSE,quote=FALSE,append=TRUE)


#Creating the description file (classification)
dat <- read.table("cleandata.txt",header=TRUE)
nvar <- ncol(dat)
varnames <- names(dat)
roles <- rep("c",nvar)

n.vars <- c("PWGTP","AGEP","JWMNP","OIP", "PAP", "RETP", "SEMP", "SSIP",
"SSP", "WAGP", "WKHP", "WKWN", "PERNP",
            "PINCP", "POVPIP")
roles[varnames %in% n.vars] <- "n"

x.var <- "INTP"
roles[varnames %in% x.var] <- "x"

d.var <- "FINTP"
roles[varnames %in% d.var] <- "d"

write("cleandata.txt",file="descclass.txt")
write("NA",file="descclass.txt",append=TRUE)
write("2",file="descclass.txt",append=TRUE)
write.table(cbind(1:nvar,varnames,roles),file="descclass.txt",
            row.names=FALSE,col.names=FALSE,quote=FALSE,append=TRUE)
```

Appendix B

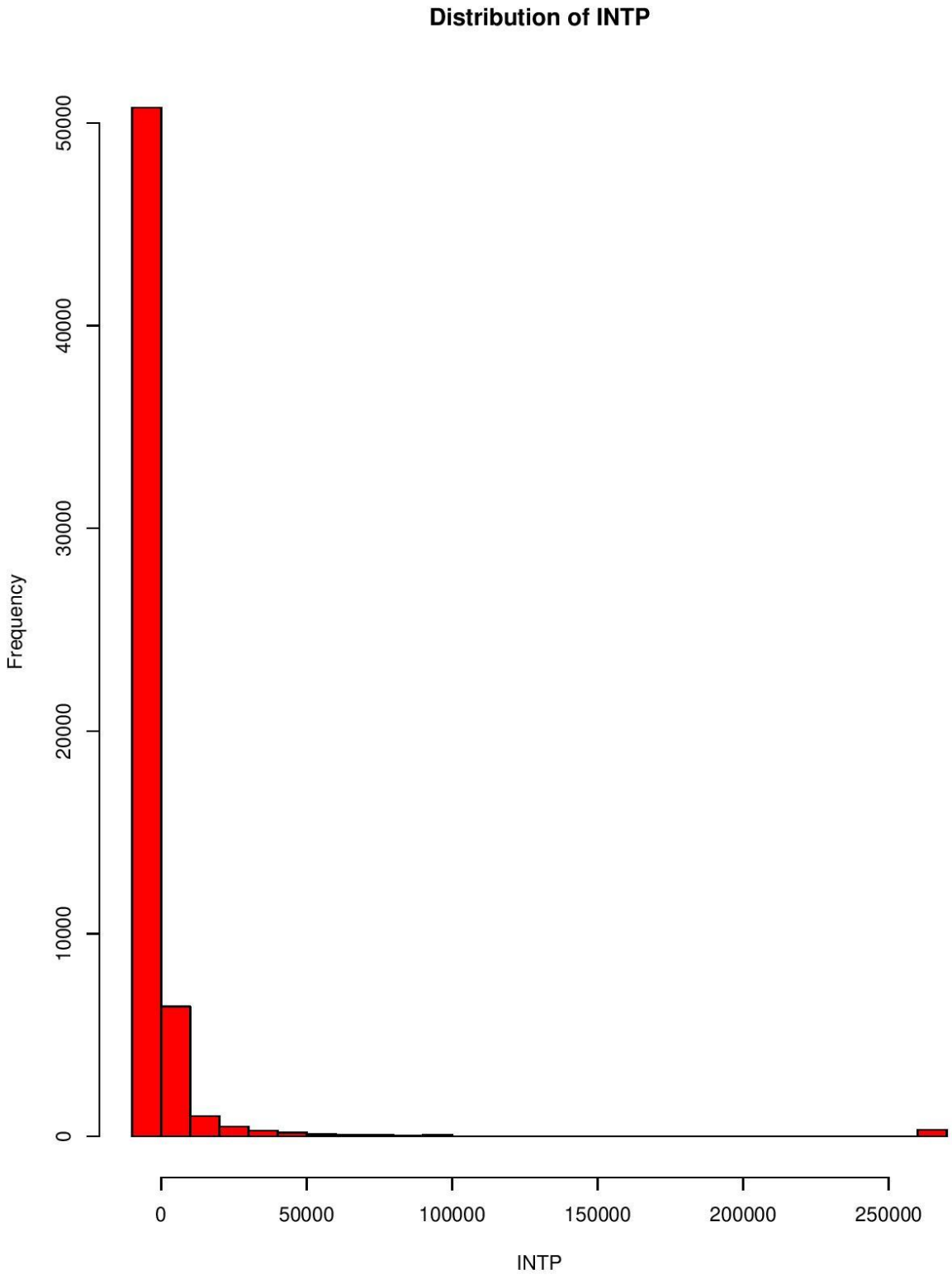
R Code

```
# summary of missing values and INTP infomation
z <- read.table("cleandata.txt",header=TRUE)
length(z$INTP) # total number of observations after unimputing dataset: 71066
table(z$FINTP) # number of missing INTP observations: 11238 (FINTP = 1)
apply(z, 2, FUN = function(x) sum(is.na(x)))

# Distribution of INTP
hist(z$INTP, breaks = 20, main = "Distribution of INTP", xlab = "INTP", col =
"red")
```

Clean Data Summary

```
> length(z$INTP) # total number of observations after unimputing dataset: 71066
[1] 71066
> table(z$FINTP) # number of missing INTP observations: 11238 (FINTP = 1)

    0     1
59828 11238
> apply(z, 2, FUN = function(x) sum(is.na(x)))
     PUMA     PWGTP      AGEP       CIT     CITWP       COW      DDRS      DEAR      DEYE      DOUT      DPHY      DRAT      DRATX
        0         0         0         0     65651     18549         0         0         0         0         0     68888     61893
     DREM       ENG       FER       GCL       GCM       GCR    HIMRKS     HINS1     HINS2     HINS3     HINS4     HINS5     HINS6
        0     61047     52504     15431     70465     69380         0         0         0         0         0         0         0
    HINS7      INTP     JWMNP     JWRIP     JWTRNS      LANX       MAR     MARHD     MARHM     MARHT     MARHW     MARHYP       MIG
        0     11238     32048     35587     29391         0         0     20644     20644     20644     20644     20644         0
      MIL      MLPA      MLPB     MLPCD      MLPE     MLPFG     MLPH      MLPI      MLPJ      MLPK      NWAB      NWAV      NWLA
     2002     62744     62744     62744     62744     62744     62744     62744     62744     62744       998       998       998
     NWLK      NWRE       OIP       PAP  RELSHIPP      RETP       SCH     SCHG      SCHL      SEMP       SEX      SSIP       SSP
      998       998         0         0         0         0         0     61166         0         0         0         0         0
     WAGP      WKHP       WKL      WKWN       WRK      YOEP       ANC     ANC1P     ANC2P    DECADE       DIS   DRIVESP       ESP
        0     24554       998     24554      8268     60944         0         0         0     60944         0     35587     68319
      ESR     FOD1P     FOD2P     HICOV      HISP      INDP      JWAP      JWDP      LANP   MIGPUMA    MIGSP       MSP    NAICSP
      998     44125     67753         0         0     18549     32048     32048     61047     61097     61097         0     18549
 NATIVITY       NOP        OC      OCCP      PAOC     PERNP     PINCP      POBP    POVPIP   POWPUMA    POWSP   PRIVCOV   PUBCOV
        0     68319      4249     18549     36561       998         0         0      3783     29391     29391         0         0
   QTRBIR     RAC1P     RAC2P     RAC3P    RACAIAN    RACASN    RACBLK    RACNH     RACNUM    RACPI     RACSOR    RACWHT       RC
        0         0         0         0         0         0         0         0         0         0         0         0      4249
  SCIENGP SCIENGRLP       SFN       SFR      SOCP       VPS      WAOB     FINTP
    44125     44125     69829     69829     18549     62744         0         0
>
```

INTP Distribution



Regression Description File

```
cleandata.txt
NA
2
1 PUMA c
2 PWGTP n
3 AGEP n
4 CIT c
5 CITWP c
6 COW c
7 DDRS c
8 DEAR c
9 DEYE c
10 DOUT c
11 DPHY c
12 DRAT c
13 DRATX c
14 DREM c
15 ENG c
16 FER c
17 GCL c
18 GCM c
19 GCR c
20 HIMRKS c
21 HINS1 c
22 HINS2 c
23 HINS3 c
24 HINS4 c
25 HINS5 c
26 HINS6 c
27 HINS7 c
28 INTP d
29 JWMNP n
30 JWRIP c
31 JWTRNS c
32 LANX c
33 MAR c
34 MARHD c
35 MARHM c
36 MARHT c
37 MARHW c
38 MARHYP c
39 MIG c
40 MIL c
41 MLPA c
42 MLPB c
43 MLPCD c
44 MLPE c
45 MLPFG c
46 MLPH c
47 MLPI c
48 MLPJ c
49 MLPK c
50 NWAB c
51 NWAV c
52 NWLA c
53 NWLK c
54 NWRE c
55 OIP n
56 PAP n
57 RELSHIPP c
58 RETP n
59 SCH c
60 SCHG c
61 SCHL c
62 SEMP n
63 SEX c
64 SSIP n
65 SSP n
66 WAGP n
67 WKHP n
68 WKL c
69 WKWN n
70 WRK c
71 YOEP c
72 ANC c
73 ANC1P c
74 ANC2P c
75 DECADE c
76 DIS c
77 DRIVESP c
78 ESP c
79 ESR c
80 FOD1P c
81 FOD2P c
82 HICOV c
83 HISP c
84 INDP c
85 JWAP c
86 JWDP c
87 LANP c
88 MIGPUMA c
89 MIGSP c
90 MSP c
91 NAICSP c
92 NATIVITY c
93 NOP c
94 OC c
95 OCCP c
96 PAOC c
97 PERNP n
98 PINCP n
99 POBP c
100 POVPIP n
101 POWPUMA c
102 POWSP c
103 PRIVCOV c
104 PUBCOV c
105 QTRBIR c
106 RAC1P c
107 RAC2P c
108 RAC3P c
109 RACAIAN c
110 RACASN c
111 RACBLK c
112 RACNH c
113 RACNUM c
114 RACPI c
115 RACSOR c
116 RACWHT c
117 RC c
118 SCIENGP c
119 SCIENGRLP c
120 SFN c
121 SFR c
122 SOCP c
123 VPS c
124 WAOB c
125 FINTP x
```

Classification Description File

```
cleandata.txt
NA
2
1 PUMA c
2 PWGTP n
3 AGEP n
4 CIT c
5 CITWP c
6 COW c
7 DDRS c
8 DEAR c
9 DEYE c
10 DOUT c
11 DPHY c
12 DRAT c
13 DRATX c
14 DREM c
15 ENG c
16 FER c
17 GCL c
18 GCM c
19 GCR c
20 HIMRKS c
21 HINS1 c
22 HINS2 c
23 HINS3 c
24 HINS4 c
25 HINS5 c
26 HINS6 c
27 HINS7 c
28 INTP x
29 JWMNP n
30 JWRIP c
31 JWTRNS c
32 LANX c
33 MAR c
34 MARHD c
35 MARHM c
36 MARHT c
37 MARHW c
38 MARHYP c
39 MIG c
40 MIL c
41 MLPA c
42 MLPB c
43 MLPCD c
44 MLPE c
45 MLPFG c
46 MLPH c
47 MLPI c
48 MLPJ c
49 MLPK c
50 NWAB c
51 NWAV c
52 NWLA c
53 NWLK c
54 NWRE c
55 OIP n
56 PAP n
57 RELSHIPP c
58 RETP n
59 SCH c
60 SCHG c
61 SCHL c
62 SEMP n
63 SEX c
64 SSIP n
65 SSP n
66 WAGP n
67 WKHP n
68 WKL c
69 WKWN n
70 WRK c
71 YOEP c
72 ANC c
73 ANC1P c
74 ANC2P c
75 DECADE c
76 DIS c
77 DRIVESP c
78 ESP c
79 ESR c
80 FOD1P c
81 FOD2P c
82 HICOV c
83 HISP c
84 INDP c
85 JWAP c
86 JWDP c
87 LANP c
88 MIGPUMA c
89 MIGSP c
90 MSP c
91 NAICSP c
92 NATIVITY c
93 NOP c
94 OC c
95 OCCP c
96 PAOC c
97 PERNP n
98 PINCP n
99 POBP c
100 POVPIP n
101 POWPUMA c
102 POWSP c
103 PRIVCOV c
104 PUBCOV c
105 QTRBIR c
106 RAC1P c
107 RAC2P c
108 RAC3P c
109 RACAIAN c
110 RACASN c
111 RACBLK c
112 RACNH c
113 RACNUM c
114 RACPI c
115 RACSOR c
116 RACWHT c
117 RC c
118 SCIENGP c
119 SCIENGRLP c
120 SFN c
121 SFR c
122 SOCP c
123 VPS c
124 WAOB c
125 FINTP d
```

Appendix C

```
# Using IPW to estimate mean (mu) via GUIDE classification tree (1)
z <- read.table("cleandata.txt",header=TRUE)
w <- z$PWGTP ### sampling weights
zclass <- read.table("classtree/gtree.txt",header=TRUE)
probmissing <- zclass[,5] ### estimated P(FINTP = 0)
p <- 1-probmissing ### estimated P(FINTP is nonmissing)
group <- !is.na(z$INTP) ### group of nonmissing INTP obs
ipw <- sum(w[group]*z$INTP[group]/p[group])/sum(w[group]/p[group])
print(ipw)
# 2292.499
```

GUIDE Classification Tree Input File

```
GUIDE       (do not edit this file unless you know what you are doing)
  36.2      (version of GUIDE that generated this file)
 1          (1=model fitting, 2=importance or DIF scoring, 3=data conversion)
"gtree.out"  (name of output file)
 1          (1=one tree, 2=ensemble)
 1          (1=classification, 2=regression, 3=propensity score grouping)
 1          (1=simple model, 2=nearest-neighbor, 3=kernel)
 1          (0=linear 1st, 1=univariate 1st, 2=skip linear, 3=skip linear and
interaction)
 1          (0=tree with fixed no. of nodes, 1=prune by CV, 2=by test sample,
3=no pruning)
"descclass.txt"  (name of data description file)
        10  (number of cross-validations)
 1          (1=mean-based CV tree, 2=median-based CV tree)
     0.500  (SE number for pruning)
 1          (1=estimated priors, 2=equal priors, 3=other priors)
 1          (1=unit misclassification costs, 2=other)
 2          (1=split point from quantiles, 2=use exhaustive search)
 1          (1=default max. number of split levels, 2=specify no. in next line)
 1          (1=default min. node size, 2=specify min. value in next line)
 2          (0=no LaTeX code, 1=tree without node numbers, 2=tree with node
numbers)
"gtree.tex" (latex file name)
 1          (1=color terminal nodes, 2=no colors)
 2          (0=#errors, 1=sample sizes, 2=sample proportions, 3=posterior probs,
4=nothing)
 1          (1=no storage, 2=store fit and split variables, 3=store split
variables and values)
 2          (1=do not save fitted values and node IDs, 2=save in a file)
"gtree.txt" (file name for fitted values and node IDs)
 2          (1=do not write R function, 2=write R function)
"gtree.R" (R code file)
 1          (rank of top variable to split root node)
```

GUIDE Classification Tree Output File

```
                GGG   U   U  I  DDDD   EEEE
               G   G  U   U  I  D   D  E
               G      U   U  I  D   D  E
               G  GG  U   U  I  D   D  EEE
               G   G  U   U  I  D   D  E
               G   G  U   U  I  D   D  E
                GGG    UUU   I  DDDD   EEEE

          GUIDE Classification and Regression Trees and Forests
          Version 36.2 (Build date: January 10, 2021)
          Compiled with Visual Fortran 64 18.0.1.156 on Windows 10
          Copyright (c) 1997-2020 Wei-Yin Loh. All rights reserved.
          This software is based upon work supported by the U.S. Army Research Office,
          the National Science Foundation and the National Institutes of Health.

          This job was started on 04/15/21 at 21:52

          Classification tree
          Pruning by cross-validation
          Data description file: descclass.txt
          Training sample file: cleandata.txt
          Missing value code: NA
          Records in data file start on line 2
          15 N variables changed to S
          D variable is FINTP
          Number of records in data file: 71066
          Length of longest entry in data file: 8
          Missing values found among categorical variables
          Separate categories will be created for missing categorical variables
          Missing values found among non-categorical variables
          Number of classes: 2
          Training sample class proportions of D variable FINTP:
          Class  #Cases     Proportion
          0       59828     0.84186531
          1       11238     0.15813469

          Summary information for training sample of size 71066
          d=dependent, b=split and fit cat variable using indicator variables,
          c=split-only categorical, i=fit-only categorical (via indicators),
          s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
          m=missing-value flag variable, p=periodic variable, w=weight
                                                      #Codes/
                                                      Levels/
           Column  Name            Minimum      Maximum    Periods   #Missing
                1  PUMA       c                                56
                2  PWGTP      s    1.000        2068.
                3  AGEP       s    15.00        94.00
                4  CIT        c                                 5
                5  CITWP      c                                73     65651
                6  COW        c                                 9     18549
                7  DDRS       c                                 2
                8  DEAR       c                                 2
                9  DEYE       c                                 2
               10  DOUT       c                                 2
               11  DPHY       c                                 2
               12  DRAT       c                                 6     68888
               13  DRATX      c                                 2     61893
               14  DREM       c                                 2
               15  ENG        c                                 4     61047
               16  FER        c                                 2     52504
               17  GCL        c                                 2     15431
               18  GCM        c                                 5     70465
               19  GCR        c                                 2     69380
               20  HIMRKS     c                                 3
```

21	HINS1	c			2	
22	HINS2	c			2	
23	HINS3	c			2	
24	HINS4	c			2	
25	HINS5	c			2	
26	HINS6	c			2	
27	HINS7	c			2	
29	JWMNP	s	1.000	149.0		32048
30	JWRIP	c			10	35587
31	JWTRNS	c			12	29391
32	LANX	c			2	
33	MAR	c			5	
34	MARHD	c			2	20644
35	MARHM	c			2	20644
36	MARHT	c			3	20644
37	MARHW	c			2	20644
38	MARHYP	c			80	20644
39	MIG	c			3	
40	MIL	c			4	2002
41	MLPA	c			2	62744
42	MLPB	c			2	62744
43	MLPCD	c			2	62744
44	MLPE	c			2	62744
45	MLPFG	c			2	62744
46	MLPH	c			2	62744
47	MLPI	c			2	62744
48	MLPJ	c			2	62744
49	MLPK	c			2	62744
50	NWAB	c			3	998
51	NWAV	c			4	998
52	NWLA	c			3	998
53	NWLK	c			3	998
54	NWRE	c			3	998
55	OIP	s	0.000	0.7500E+05		
56	PAP	s	0.000	0.1640E+05		
57	RELSHIPP	c			19	
58	RETP	s	0.000	0.1550E+06		
59	SCH	c			3	
60	SCHG	c			9	61166
61	SCHL	c			24	
62	SEMP	s	-6900.	0.4300E+06		
63	SEX	c			2	
64	SSIP	s	0.000	0.2500E+05		
65	SSP	s	0.000	0.3800E+05		
66	WAGP	s	0.000	0.5160E+06		
67	WKHP	s	1.000	99.00		24554
68	WKL	c			3	998
69	WKWN	s	1.000	52.00		24554
70	WRK	c			2	8268
71	YOEP	c			81	60944
72	ANC	c			4	
73	ANC1P	c			225	
74	ANC2P	c			191	
75	DECADE	c			8	60944
76	DIS	c			2	
77	DRIVESP	c			6	35587
78	ESP	c			8	68319
79	ESR	c			6	998
80	FOD1P	c			173	44125
81	FOD2P	c			158	67753
82	HICOV	c			2	
83	HISP	c			24	
84	INDP	c			270	18549
85	JWAP	c			285	32048

```
     86  JWDP       c                                    150     32048
     87  LANP       c                                    113     61047
     88  MIGPUMA    c                                    161     61097
     89  MIGSP      c                                    101     61097
     90  MSP        c                                      6
     91  NAICSP     c                                    270     18549
     92  NATIVITY   c                                      2
     93  NOP        c                                      8     68319
     94  OC         c                                      2      4249
     95  OCCP       c                                    527     18549
     96  PAOC       c                                      4     36561
     97  PERNP      s   -6900.       0.9460E+06                   998
     98  PINCP      s   -6900.       0.1137E+07
     99  POBP       c                                    218
    100  POVPIP     s    0.000        501.0                      3783
    101  POWPUMA    c                                    114     29391
    102  POWSP      c                                     45     29391
    103  PRIVCOV    c                                      2
    104  PUBCOV     c                                      2
    105  QTRBIR     c                                      4
    106  RAC1P      c                                      9
    107  RAC2P      c                                     53
    108  RAC3P      c                                     89
    109  RACAIAN    c                                      2
    110  RACASN     c                                      2
    111  RACBLK     c                                      2
    112  RACNH      c                                      2
    113  RACNUM     c                                      5
    114  RACPI      c                                      2
    115  RACSOR     c                                      2
    116  RACWHT     c                                      2
    117  RC         c                                      2      4249
    118  SCIENGP    c                                      2     44125
    119  SCIENGRLP  c                                      2     44125
    120  SFN        c                                      2     69829
    121  SFR        c                                      6     69829
    122  SOCP       c                                    527     18549
    123  VPS        c                                     15     62744
    124  WAOB       c                                      8
    125  FINTP      d                                      2

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
     71066          0      32819        1        0        0       15
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0      108        0
Number of cases used for training: 71066
Number of split variables: 123
Number of cases excluded due to 0 weight or missing D: 0

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: .5000

Simple node models
Estimated priors
Unit misclassification costs
Univariate split highest priority
Interaction and linear splits 2nd and 3rd priorities
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 30
Minimum node sample size: 710
Top-ranked variables and chi-squared values at root node
     1  0.1990E+05   WRK
```

2	0.1350E+05	ANC
3	0.1090E+05	ANC1P
4	0.1541E+04	NWAB
5	0.1433E+04	NWLA
6	0.1404E+04	NWLK
7	0.9267E+03	ANC2P
8	0.6020E+03	RELSHIPP
9	0.5322E+03	MSP
10	0.4586E+03	SCHL
11	0.3621E+03	POVPIP
12	0.3448E+03	NAICSP
13	0.3448E+03	INDP
14	0.3341E+03	SCIENGP
15	0.3305E+03	SCIENGRLP
16	0.3008E+03	RACBLK
17	0.2945E+03	RAC1P
18	0.2900E+03	RACWHT
19	0.2865E+03	FOD1P
20	0.2794E+03	MAR
21	0.2407E+03	WAGP
22	0.2382E+03	HINS3
23	0.2261E+03	PERNP
24	0.2225E+03	RAC3P
25	0.2188E+03	PUBCOV
26	0.2179E+03	RAC2P
27	0.2073E+03	SSP
28	0.1956E+03	AGEP
29	0.1943E+03	PINCP
30	0.1850E+03	MIGPUMA
31	0.1720E+03	OCCP
32	0.1720E+03	SOCP
33	0.1689E+03	MARHYP
34	0.1628E+03	PUMA
35	0.1423E+03	COW
36	0.1396E+03	POBP
37	0.1307E+03	WKL
38	0.1076E+03	SCHG
39	0.1060E+03	DPHY
40	0.1054E+03	WKHP
41	0.9951E+02	DRATX
42	0.9137E+02	HISP
43	0.9111E+02	HINS1
44	0.8957E+02	DOUT
45	0.8819E+02	PRIVCOV
46	0.8811E+02	WKWN
47	0.8591E+02	DDRS
48	0.8276E+02	MIGSP
49	0.8275E+02	MIG
50	0.8036E+02	DRAT
51	0.8018E+02	DIS
52	0.7920E+02	ESR
53	0.7221E+02	OC
54	0.6780E+02	VPS
55	0.6751E+02	HINS2
56	0.6682E+02	RC
57	0.6564E+02	MARHW
58	0.6364E+02	NWAV
59	0.6338E+02	LANP
60	0.6273E+02	JWTRNS
61	0.6153E+02	POWPUMA
62	0.5839E+02	HIMRKS
63	0.5222E+02	POWSP
64	0.5116E+02	SFR
65	0.5012E+02	MLPH

66	0.4122E+02	MARHM
67	0.4092E+02	MARHD
68	0.3836E+02	JWMNP
69	0.3797E+02	PAOC
70	0.3784E+02	MARHT
71	0.3683E+02	SCH
72	0.3220E+02	MIL
73	0.3144E+02	DEYE
74	0.2563E+02	DRIVESP
75	0.2543E+02	RACSOR
76	0.2510E+02	JWRIP
77	0.2482E+02	LANX
78	0.2187E+02	DREM
79	0.1989E+02	RETP
80	0.1949E+02	HINS4
81	0.1935E+02	SFN
82	0.1913E+02	ENG
83	0.1765E+02	MLPCD
84	0.1735E+02	NWRE
85	0.1630E+02	MLPFG
86	0.1471E+02	SSIP
87	0.1452E+02	PWGTP
88	0.1431E+02	SEMP
89	0.1350E+02	DEAR
90	0.1292E+02	GCL
91	0.1225E+02	FER
92	0.1081E+02	CIT
93	0.1076E+02	CITWP
94	0.1073E+02	JWDP
95	0.1069E+02	MLPI
96	0.1067E+02	ESP
97	0.8712E+01	HICOV
98	0.8281E+01	JWAP
99	0.8076E+01	NOP
100	0.7990E+01	MLPA
101	0.7645E+01	HINS5
102	0.7051E+01	MLPJ
103	0.5604E+01	WAOB
104	0.4436E+01	HINS6
105	0.4114E+01	MLPB
106	0.3866E+01	SEX
107	0.3690E+01	NATIVITY
108	0.2657E+01	GCR
109	0.1205E+01	DECADE
110	0.1194E+01	MLPE
111	0.1166E+01	FOD2P
112	0.4226E+00	OIP
113	0.4107E+00	YOEP
114	0.3632E+00	MLPK
115	0.2235E+00	RACASN
116	0.2066E+00	QTRBIR
117	0.3866E-01	RACNUM

Size and CV mean cost and SE of subtrees:

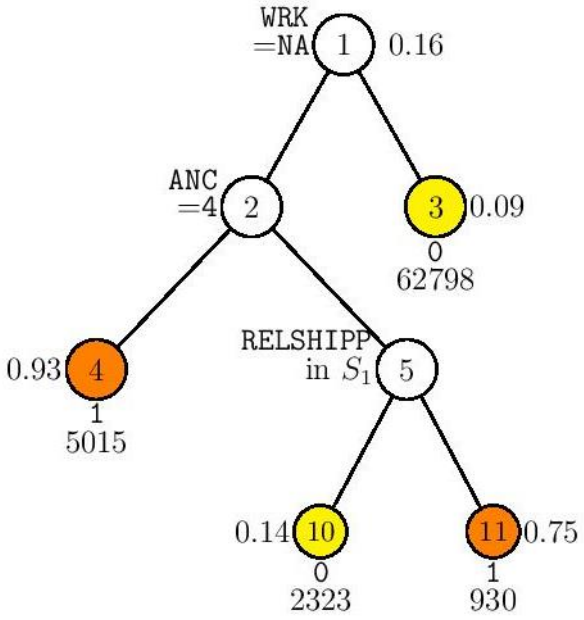
Tree	#Tnodes	Mean Cost	SE(Mean)	BSE(Mean)	Median Cost	BSE(Median)
1	80	9.089E-02	1.078E-03	6.778E-04	9.027E-02	6.580E-04
2	79	9.089E-02	1.078E-03	6.778E-04	9.027E-02	6.580E-04
3	78	9.089E-02	1.078E-03	6.778E-04	9.027E-02	6.580E-04
4	77	9.089E-02	1.078E-03	6.778E-04	9.027E-02	6.580E-04
5	76	9.089E-02	1.078E-03	6.778E-04	9.027E-02	6.580E-04
6	75	9.089E-02	1.078E-03	6.778E-04	9.027E-02	6.580E-04
7	74	9.089E-02	1.078E-03	6.778E-04	9.027E-02	6.580E-04
8	72	9.089E-02	1.078E-03	6.778E-04	9.027E-02	6.580E-04
9	70	9.089E-02	1.078E-03	6.778E-04	9.027E-02	6.580E-04

```
  10       69   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  11       68   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  12       67   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  13       64   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  14       63   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  15       61   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  16       60   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  17       59   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  18       58   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  19       57   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  20       56   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  21       55   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  22       53   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  23       47   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  24       46   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  25       45   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  26       44   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  27       43   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  28       42   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  29       41   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  30       39   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  31       38   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  32       36   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  33       35   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  34       34   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  35       27   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  36       26   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  37       25   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  38       24   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  39       23   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  40       21   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  41       17   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  42       14   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  43**      4   9.089E-02   1.078E-03   6.778E-04   9.027E-02   6.580E-04
  44        3   9.674E-02   1.109E-03   8.352E-04   9.638E-02   8.707E-04
  45        2   1.134E-01   1.190E-03   6.770E-04   1.133E-01   8.918E-04
  46        1   1.581E-01   1.369E-03   1.969E-05   1.582E-01   1.996E-05

 0-SE tree based on mean is marked with * and has 4 terminal nodes
 0-SE tree based on median is marked with + and has 4 terminal nodes
 Selected-SE tree based on mean using naive SE is marked with **
 Selected-SE tree based on mean using bootstrap SE is marked with --
 Selected-SE tree based on median and bootstrap SE is marked with ++
 * tree, ** tree, + tree, and ++ tree all the same

 Following tree is based on mean CV with naive SE estimate (**).

 Structure of final tree. Each terminal node is marked with a T.

 Node cost is node misclassification cost divided by number of training cases
       Node    Total    Train    Predicted        Node    Split
Interacting
      label    cases    cases    class            cost    variables
variable
          1    71066    71066    0             1.581E-01 WRK
          2     8268     8268    1             3.079E-01 ANC
         4T     5015     5015    1             6.500E-02 PERNP
          5     3253     3253    0             3.176E-01 RELSHIPP
        10T     2323     2323    0             1.442E-01 ESR
        11T      930      930    1             2.495E-01 -
         3T    62798    62798    0             8.784E-02 ANC

 Number of terminal nodes of final tree: 4
 Total number of nodes of final tree: 7
```

```
 Second best split variable (based on curvature test) at root node is ANC

 Classification tree:
 For categorical variable splits, values not in training data go to the right

 Node 1: WRK = "NA"
   Node 2: ANC = "4"
     Node 4: 1
   Node 2: ANC /= "4"
     Node 5: RELSHIPP = "25", "26", "27", "30", "35", "36", "37"
       Node 10: 0
     Node 5: RELSHIPP /= "25", "26", "27", "30", "35", "36", "37"
       Node 11: 1
 Node 1: WRK /= "NA"
   Node 3: 0

 ************************************************************

 Predictor means below are means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if WRK = "NA"
 WRK mode = "1"
 Class      Number   Posterior
 0           59828  0.8419E+00
 1           11238  0.1581E+00
 Number of training cases misclassified = 11238
 Predicted class is 0
 ---------------------------
Node 2: Intermediate node
 A case goes into Node 4 if ANC = "4"
 ANC mode = "4"
 Class      Number   Posterior
 0            2546  0.3079E+00
 1            5722  0.6921E+00
 Number of training cases misclassified = 2546
 Predicted class is 1
 ---------------------------
Node 4: Terminal node
 Class      Number   Posterior
 0             326  0.6500E-01
 1            4689  0.9350E+00
 Number of training cases misclassified = 326
 Predicted class is 1
 ---------------------------
Node 5: Intermediate node
 A case goes into Node 10 if RELSHIPP = "25", "26", "27", "30", "35", "36", "37"
 RELSHIPP mode = "37"
 Class      Number   Posterior
 0            2220  0.6824E+00
 1            1033  0.3176E+00
 Number of training cases misclassified = 1033
 Predicted class is 0
 ---------------------------
Node 10: Terminal node
 Class      Number   Posterior
 0            1988  0.8558E+00
 1             335  0.1442E+00
 Number of training cases misclassified = 335
 Predicted class is 0
 ---------------------------
Node 11: Terminal node
 Class      Number   Posterior
 0             232  0.2495E+00
```

```
 1             698  0.7505E+00
 Number of training cases misclassified = 232
 Predicted class is 1
 ---------------------------
Node 3: Terminal node
 Class      Number   Posterior
 0           57282  0.9122E+00
 1            5516  0.8784E-01
 Number of training cases misclassified = 5516
 Predicted class is 0
 ---------------------------

 Classification matrix for training sample:
 Predicted      True class
 class              0         1
 0              59270      5851
 1                558      5387
 Total          59828     11238

 Number of cases used for tree construction: 71066
 Number misclassified: 6409
 Resubstitution estimate of mean misclassification cost: .90183773E-01

 Observed and fitted values are stored in gtree.txt
 LaTeX code for tree is in gtree.tex
 R code is stored in gtree.R
 Elapsed time in seconds: 454.39
```

GUIDE Classification Tree



GUIDE v.36.2 0.50-SE classification tree for predicting FINTP using estimated priors and unit misclassification costs. Tree constructed with 71066 observations. Maximum number of split levels is 30 and minimum node sample size is 710. At each split, an observation goes to the left branch if and only if the condition is satisfied. Set $S_1 = \{25, 26, 27, 30, 35, 36, 37\}$. Predicted classes and sample sizes printed below terminal nodes; class sample proportion for FINTP = 1 beside nodes. Second best split variable at root node is ANC.

Appendix D

```
# Using IPW to estimate mean (mu) classification forest (1)
z <- read.table("cleandata.txt",header=TRUE)
w <- z$PWGTP ### sampling weights
zclass <- read.table("guide cf/gcfpred.txt",header=TRUE)
probmissing <- zclass[,2] ### estimated P(FINTP = 0)
p <- 1-probmissing ### estimated P(FINTP is nonmissing)
group <- !is.na(z$INTP) ### group of nonmissing INTP obs
ipw <- sum(w[group]*z$INTP[group]/p[group])/sum(w[group]/p[group])
print(ipw)
# 2019.165 (new IPW after loh's announcement)
# 2118.599 (old value before Professor Loh's announcement)
```

GUIDE Forest Input File

```
GUIDE       (do not edit this file unless you know what you are doing)
  36.2      (version of GUIDE that generated this file)
 1          (1=model fitting, 2=importance or DIF scoring, 3=data
conversion)
"gcf.out"  (name of output file)
 2          (1=one tree, 2=ensemble)
 2          (1=bagging, 2=rforest)
 2          (1=random splits of missing values, 2=nonrandom)
 1          (1=classification, 2=regression)
 2          (1=interaction tests, 2=skip them)
"descclass.txt"  (name of data description file)
 1          (1=accept default number of trees, 2=change)
 1          (1=accept default number of variables for splitting, 2=change
it)
 1          (1=estimated priors, 2=equal priors, 3=other priors)
 1          (1=unit misclassification costs, 2=other)
 1          (1=split point from quantiles, 2=use exhaustive search)
 1          (1=accept default splitting fraction, 2=change it)
 1          (1=default max. number of split levels, 2=specify no. in next
line)
 1          (1=default min. node size, 2=specify min. value in next line)
"gcfpred.txt" (file name for predicted class and probability estimates)
 1          (rank of top variable to split root node)
```

GUIDE Forest Output File

```
            GGG   U   U  I  DDDD   EEEE
           G   G  U   U  I  D   D  E
           G      U   U  I  D   D  E
           G  GG  U   U  I  D   D  EEE
           G   G  U   U  I  D   D  E
           G   G  U   U  I  D   D  E
            GGG    UUU   I  DDDD   EEEE

 GUIDE Classification and Regression Trees and Forests
 Version 36.2 (Build date: January 10, 2021)
 Compiled with Visual Fortran 64 18.0.1.156 on Windows 10
 Copyright (c) 1997-2020 Wei-Yin Loh. All rights reserved.
 This software is based upon work supported by the U.S. Army Research
Office,
 the National Science Foundation and the National Institutes of Health.

 This job was started on 04/15/21 at 21:59

 Random forest of classification trees
 No pruning
 Data description file: descclass.txt
 Training sample file: cleandata.txt
 Missing value code: NA
 Records in data file start on line 2
 15 N variables changed to S
 D variable is FINTP
 Number of records in data file: 71066
 Length of longest entry in data file: 8
 Missing values found among categorical variables
 Separate categories will be created for missing categorical variables
 Missing values found among non-categorical variables
 Number of classes: 2
 Training sample class proportions of D variable FINTP:
 Class  #Cases     Proportion
 0       59828     0.84186531
 1       11238     0.15813469

 Summary information for training sample of size 71066
 d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight
                                                  #Codes/
                                                  Levels/
  Column  Name            Minimum      Maximum    Periods   #Missing
       1  PUMA       c                              56
       2  PWGTP      s    1.000        2068.
       3  AGEP       s    15.00        94.00
       4  CIT        c                               5
       5  CITWP      c                              73      65651
       6  COW        c                               9      18549
       7  DDRS       c                               2
       8  DEAR       c                               2
       9  DEYE       c                               2
```

10	DOUT	c			2	
11	DPHY	c			2	
12	DRAT	c			6	68888
13	DRATX	c			2	61893
14	DREM	c			2	
15	ENG	c			4	61047
16	FER	c			2	52504
17	GCL	c			2	15431
18	GCM	c			5	70465
19	GCR	c			2	69380
20	HIMRKS	c			3	
21	HINS1	c			2	
22	HINS2	c			2	
23	HINS3	c			2	
24	HINS4	c			2	
25	HINS5	c			2	
26	HINS6	c			2	
27	HINS7	c			2	
29	JWMNP	s	1.000	149.0		32048
30	JWRIP	c			10	35587
31	JWTRNS	c			12	29391
32	LANX	c			2	
33	MAR	c			5	
34	MARHD	c			2	20644
35	MARHM	c			2	20644
36	MARHT	c			3	20644
37	MARHW	c			2	20644
38	MARHYP	c			80	20644
39	MIG	c			3	
40	MIL	c			4	2002
41	MLPA	c			2	62744
42	MLPB	c			2	62744
43	MLPCD	c			2	62744
44	MLPE	c			2	62744
45	MLPFG	c			2	62744
46	MLPH	c			2	62744
47	MLPI	c			2	62744
48	MLPJ	c			2	62744
49	MLPK	c			2	62744
50	NWAB	c			3	998
51	NWAV	c			4	998
52	NWLA	c			3	998
53	NWLK	c			3	998
54	NWRE	c			3	998
55	OIP	s	0.000	0.7500E+05		
56	PAP	s	0.000	0.1640E+05		
57	RELSHIPP	c			19	
58	RETP	s	0.000	0.1550E+06		
59	SCH	c			3	
60	SCHG	c			9	61166
61	SCHL	c			24	
62	SEMP	s	-6900.	0.4300E+06		
63	SEX	c			2	
64	SSIP	s	0.000	0.2500E+05		

65	SSP	s	0.000	0.3800E+05		
66	WAGP	s	0.000	0.5160E+06		
67	WKHP	s	1.000	99.00		24554
68	WKL	c			3	998
69	WKWN	s	1.000	52.00		24554
70	WRK	c			2	8268
71	YOEP	c			81	60944
72	ANC	c			4	
73	ANC1P	c			225	
74	ANC2P	c			191	
75	DECADE	c			8	60944
76	DIS	c			2	
77	DRIVESP	c			6	35587
78	ESP	c			8	68319
79	ESR	c			6	998
80	FOD1P	c			173	44125
81	FOD2P	c			158	67753
82	HICOV	c			2	
83	HISP	c			24	
84	INDP	c			270	18549
85	JWAP	c			285	32048
86	JWDP	c			150	32048
87	LANP	c			113	61047
88	MIGPUMA	c			161	61097
89	MIGSP	c			101	61097
90	MSP	c			6	
91	NAICSP	c			270	18549
92	NATIVITY	c			2	
93	NOP	c			8	68319
94	OC	c			2	4249
95	OCCP	c			527	18549
96	PAOC	c			4	36561
97	PERNP	s	-6900.	0.9460E+06		998
98	PINCP	s	-6900.	0.1137E+07		
99	POBP	c			218	
100	POVPIP	s	0.000	501.0		3783
101	POWPUMA	c			114	29391
102	POWSP	c			45	29391
103	PRIVCOV	c			2	
104	PUBCOV	c			2	
105	QTRBIR	c			4	
106	RAC1P	c			9	
107	RAC2P	c			53	
108	RAC3P	c			89	
109	RACAIAN	c			2	
110	RACASN	c			2	
111	RACBLK	c			2	
112	RACNH	c			2	
113	RACNUM	c			5	
114	RACPI	c			2	
115	RACSOR	c			2	
116	RACWHT	c			2	
117	RC	c			2	4249
118	SCIENGP	c			2	44125

```
    119  SCIENGRLP  c                                    2     44125
    120  SFN        c                                    2     69829
    121  SFR        c                                    6     69829
    122  SOCP       c                                  527     18549
    123  VPS        c                                   15     62744
    124  WAOB       c                                    8
    125  FINTP      d                                    2

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
     71066          0      32819        1        0        0       15
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0      108        0
Number of cases used for training: 71066
Number of split variables: 123
Number of cases excluded due to 0 weight or missing D: 0

Number of trees in ensemble: 500
Number of variables used for splitting: 42
Simple node models
Estimated priors
Unit misclassification costs
Univariate split highest priority
No interaction splits
Fraction of cases used for splitting each node: .0014
Maximum number of split levels: 30
Minimum node sample size: 355
Mean number of terminal nodes:    157.0

Classification matrix for training sample:
Predicted       True class
class              0         1
0              59337      5665
1                491      5573
Total          59828     11238

Number of cases used for tree construction: 71066
Number misclassified: 6156
Resubstitution estimate of mean misclassification cost: .0866

Number of OOB cases: 71066
Number OOB misclassified: 6202
OOB estimate of mean misclassification cost: .0873
Mean number of trees per OOB observation: 183.97

Predicted class probabilities are stored in gcfpred.txt
No. times splitting stopped because node numbers were too big: 62
Elapsed time in seconds: 3184.7
```

Appendix E

```
#Using imputation to estimate INTP via regression tree result (2)
zreg <- read.table("Reg tree/regtreepred.txt",header=TRUE)
yhat <- zreg$predicted
imputed <- (sum(w[group]*z$INTP[group])+sum(w[!group]*yhat[!group]))/sum(w)
print(imputed)
# 2324.75
simple <- sum(w[group]*z$INTP[group])/sum(w[group])
print(simple)
# 2284.181
```

GUIDE Tree Input File

```
GUIDE       (do not edit this file unless you know what you are doing)
  36.2      (version of GUIDE that generated this file)
 1          (1=model fitting, 2=importance or DIF scoring, 3=data
conversion)
"regout.txt"  (name of output file)
 1          (1=one tree, 2=ensemble)
 2          (1=classification, 2=regression, 3=propensity score grouping)
 1          (1=linear, 2=quantile, 3=Poisson, 4=censored response,
5=multiresponse or itemresponse, 6=longitudinal with T vars, 7=logistic)
 1          (1=least squares, 2=least median of squares)
 3          (0=stepwise, 1=multiple linear, 2=simple polynomial,
3=constant, 4=ANCOVA)
 1          (1=interaction tests, 2=skip them)
 1          (0=tree with fixed no. of nodes, 1=prune by CV, 2=no pruning)
"descreg.txt"  (name of data description file)
       10   (number of cross-validations)
 1          (1=mean-based CV tree, 2=median-based CV tree)
    0.500   (SE number for pruning)
 2          (1=split point from quantiles, 2=use exhaustive search)
 1          (1=default max. number of split levels, 2=specify no. in next
line)
 1          (1=default min. node size, 2=specify min. value in next line)
 2          (0=no LaTeX code, 1=tree without node numbers, 2=tree with
node numbers)
"regtree.tex" (latex file name)
 1          (0=all white,1=yellow-skyblue,2=yellow-purple,3=yellow-
orange,4=orange-skyblue,5=yellow-red,6=orange-purple,7=grayscale)
 1          (1=no storage, 2=store fit and split variables, 3=store split
variables and values)
 1          (1=do not save, 2=save regressor names in a file)
 2          (1=do not save fitted values and node IDs, 2=save in a file)
"regtree.fit" (file name for fitted values and node IDs)
 2          (1=do not write R function, 2=write R function)
"regtree.R" (R code file)
 1          (rank of top variable to split root node)
```

GUIDE Tree Output File

```
             GGG   U   U  I  DDDD   EEEE
            G   G  U   U  I  D   D  E
            G      U   U  I  D   D  E
            G  GG  U   U  I  D   D  EEE
            G   G  U   U  I  D   D  E
            G   G  U   U  I  D   D  E
             GGG    UUU   I  DDDD   EEEE

 GUIDE Classification and Regression Trees and Forests
 Version 36.2 (Build date: January 10, 2021)
 Compiled with Visual Fortran 64 18.0.1.156 on Windows 10
 Copyright (c) 1997-2020 Wei-Yin Loh. All rights reserved.
 This software is based upon work supported by the U.S. Army Research
Office,
 the National Science Foundation and the National Institutes of Health.

 This job was started on 04/22/21 at 22:26

 Least squares regression tree
 Pruning by cross-validation
 Data description file: descreg.txt
 Training sample file: cleandata.txt
 Missing value code: NA
 Records in data file start on line 2
 15 N variables changed to S
 D variable is INTP
 Piecewise constant model
 Number of records in data file: 71066
 Length of longest entry in data file: 8
 Missing values found in D variable
 Missing values found among categorical variables
 Separate categories will be created for missing categorical variables
 Missing values found among non-categorical variables

 Summary information for training sample of size 59828 (excluding
observations with
 non-positive weight or missing values in d, e, t, r or z variables)
 d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight
                                                  #Codes/
                                                  Levels/
  Column  Name            Minimum      Maximum    Periods   #Missing
       1  PUMA       c                              56
       2  PWGTP      s    1.000        2068.
       3  AGEP       s    15.00        94.00
       4  CIT        c                               5
       5  CITWP      c                              73     55371
       6  COW        c                               9     15123
       7  DDRS       c                               2
       8  DEAR       c                               2
       9  DEYE       c                               2
      10  DOUT       c                               2
```

11	DPHY	c			2	
12	DRAT	c			6	57834
13	DRATX	c			2	52086
14	DREM	c			2	
15	ENG	c			4	51562
16	FER	c			2	44034
17	GCL	c			2	12860
18	GCM	c			5	59337
19	GCR	c			2	58439
20	HIMRKS	c			3	
21	HINS1	c			2	
22	HINS2	c			2	
23	HINS3	c			2	
24	HINS4	c			2	
25	HINS5	c			2	
26	HINS6	c			2	
27	HINS7	c			2	
28	INTP	d	-2000.	0.2610E+06		
29	JWMNP	s	1.000	149.0		26658
30	JWRIP	c			10	29677
31	JWTRNS	c			12	24345
32	LANX	c			2	
33	MAR	c			5	
34	MARHD	c			2	17084
35	MARHM	c			2	17084
36	MARHT	c			3	17084
37	MARHW	c			2	17084
38	MARHYP	c			80	17084
39	MIG	c			3	
40	MIL	c			4	1717
41	MLPA	c			2	52800
42	MLPB	c			2	52800
43	MLPCD	c			2	52800
44	MLPE	c			2	52800
45	MLPFG	c			2	52800
46	MLPH	c			2	52800
47	MLPI	c			2	52800
48	MLPJ	c			2	52800
49	MLPK	c			2	52800
50	NWAB	c			3	857
51	NWAV	c			4	857
52	NWLA	c			3	857
53	NWLK	c			3	857
54	NWRE	c			3	857
55	OIP	s	0.000	0.7500E+05		
56	PAP	s	0.000	0.1640E+05		
57	RELSHIPP	c			19	
58	RETP	s	0.000	0.1550E+06		
59	SCH	c			3	
60	SCHG	c			9	51684
61	SCHL	c			24	
62	SEMP	s	-6900.	0.4300E+06		
63	SEX	c			2	
64	SSIP	s	0.000	0.2500E+05		

65	SSP	s	0.000	0.3800E+05		
66	WAGP	s	0.000	0.5160E+06		
67	WKHP	s	1.000	99.00		20237
68	WKL	c			3	857
69	WKWN	s	1.000	52.00		20237
70	WRK	c			2	2546
71	YOEP	c			81	51371
72	ANC	c			4	
73	ANC1P	c			225	
74	ANC2P	c			191	
75	DECADE	c			8	51371
76	DIS	c			2	
77	DRIVESP	c			6	29677
78	ESP	c			8	57445
79	ESR	c			6	857
80	FOD1P	c			173	36265
81	FOD2P	c			158	56904
82	HICOV	c			2	
83	HISP	c			24	
84	INDP	c			270	15123
85	JWAP	c			285	26658
86	JWDP	c			150	26658
87	LANP	c			113	51562
88	MIGPUMA	c			161	51753
89	MIGSP	c			101	51753
90	MSP	c			6	
91	NAICSP	c			270	15123
92	NATIVITY	c			2	
93	NOP	c			8	57445
94	OC	c			2	3388
95	OCCP	c			527	15123
96	PAOC	c			4	30694
97	PERNP	s	-6900.	0.9460E+06		857
98	PINCP	s	-6900.	0.1137E+07		
99	POBP	c			218	
100	POVPIP	s	0.000	501.0		2961
101	POWPUMA	c			114	24345
102	POWSP	c			45	24345
103	PRIVCOV	c			2	
104	PUBCOV	c			2	
105	QTRBIR	c			4	
106	RAC1P	c			9	
107	RAC2P	c			53	
108	RAC3P	c			89	
109	RACAIAN	c			2	
110	RACASN	c			2	
111	RACBLK	c			2	
112	RACNH	c			2	
113	RACNUM	c			5	
114	RACPI	c			2	
115	RACSOR	c			2	
116	RACWHT	c			2	
117	RC	c			2	3388
118	SCIENGP	c			2	36265

```
    119  SCIENGRLP  c                                      2     36265
    120  SFN        c                                      2     58835
    121  SFR        c                                      6     58835
    122  SOCP       c                                    527     15123
    123  VPS        c                                     15     52800
    124  WAOB       c                                      8

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
     71066      11238      38517        1        0        0       15
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0      108        0
No weight variable in data file
Number of cases used for training: 59828
Number of split variables: 123
Number of cases excluded due to 0 weight or missing D: 11238

Pruning by v-fold cross-validation, with v = 10
Selected tree is based on mean of CV estimates
Number of SE's for pruned tree: .5000

Nodewise interaction tests on all variables
Split values for N and S variables based on exhaustive search
Maximum number of split levels: 30
Minimum node sample size: 598
Top-ranked variables and chi-squared values at root node
     1  0.4035E+04   PINCP
     2  0.3090E+04   AGEP
     3  0.2435E+04   SCHL
     4  0.2432E+04   RETP
     5  0.2211E+04   SCIENGP
     6  0.2198E+04   SCIENGRLP
     7  0.2032E+04   HINS3
     8  0.1985E+04   FOD1P
     9  0.1813E+04   MARHYP
    10  0.1635E+04   SSP
    11  0.1610E+04   POVPIP
    12  0.1328E+04   RELSHIPP
    13  0.1216E+04   MSP
    14  0.1185E+04   MAR
    15  0.1156E+04   GCL
    16  0.1074E+04   WAGP
    17  0.9845E+03   MARHM
    18  0.9810E+03   PERNP
    19  0.9647E+03   MARHW
    20  0.9616E+03   PUBCOV
    21  0.9517E+03   ANC1P
    22  0.9483E+03   MARHD
    23  0.9187E+03   MARHT
    24  0.9010E+03   FER
    25  0.8054E+03   VPS
    26  0.7797E+03   SOCP
    27  0.7797E+03   OCCP
    28  0.7154E+03   COW
```

29	0.7033E+03	NWLK
30	0.6750E+03	PUMA
31	0.6288E+03	WRK
32	0.5834E+03	RAC1P
33	0.5657E+03	WKL
34	0.5577E+03	HIMRKS
35	0.5397E+03	RACWHT
36	0.5395E+03	MLPE
37	0.5393E+03	INDP
38	0.5393E+03	NAICSP
39	0.5384E+03	ESR
40	0.5356E+03	RACBLK
41	0.5214E+03	MIL
42	0.5181E+03	WKHP
43	0.5098E+03	MLPFG
44	0.4981E+03	HINS2
45	0.4928E+03	RC
46	0.4863E+03	SCH
47	0.4781E+03	OC
48	0.4773E+03	NWAB
49	0.4744E+03	NWLA
50	0.4655E+03	RAC2P
51	0.4561E+03	SCHG
52	0.4527E+03	WKWN
53	0.4336E+03	MLPA
54	0.4205E+03	POBP
55	0.3926E+03	MLPH
56	0.3923E+03	RAC3P
57	0.3752E+03	JWMNP
58	0.3722E+03	DRATX
59	0.3706E+03	MLPJ
60	0.3678E+03	MLPCD
61	0.3625E+03	PWGTP
62	0.3581E+03	HINS4
63	0.3484E+03	JWDP
64	0.3422E+03	JWTRNS
65	0.3381E+03	DECADE
66	0.3197E+03	MLPI
67	0.3155E+03	MLPB
68	0.3154E+03	MLPK
69	0.3150E+03	POWPUMA
70	0.3007E+03	DRIVESP
71	0.2902E+03	ANC2P
72	0.2891E+03	POWSP
73	0.2813E+03	JWRIP
74	0.2747E+03	PRIVCOV
75	0.2507E+03	ANC
76	0.2330E+03	HICOV
77	0.2223E+03	YOEP
78	0.1962E+03	PAOC
79	0.1778E+03	MIG
80	0.1777E+03	HINS5
81	0.1625E+03	JWAP
82	0.1611E+03	FOD2P

83	0.1472E+03	ESP
84	0.1469E+03	NOP
85	0.1463E+03	CITWP
86	0.1281E+03	SEMP
87	0.1241E+03	SEX
88	0.1133E+03	DEAR
89	0.1108E+03	NWRE
90	0.9517E+02	CIT
91	0.8773E+02	HISP
92	0.8115E+02	WAOB
93	0.8027E+02	ENG
94	0.7745E+02	MIGPUMA
95	0.7713E+02	DREM
96	0.7292E+02	MIGSP
97	0.7254E+02	LANX
98	0.6985E+02	SSIP
99	0.6737E+02	NWAV
100	0.6574E+02	OIP
101	0.5996E+02	DRAT
102	0.5927E+02	LANP
103	0.4874E+02	RACSOR
104	0.4339E+02	SFN
105	0.3593E+02	SFR
106	0.3072E+02	NATIVITY
107	0.2685E+02	RACNUM
108	0.1855E+02	HINS6
109	0.1749E+02	RACASN
110	0.1721E+02	GCR
111	0.1053E+02	DEYE
112	0.8994E+01	DOUT
113	0.7183E+01	DDRS
114	0.9397E+00	DPHY
115	0.4832E+00	HINS1
116	0.2925E+00	DIS
117	0.1946E+00	QTRBIR

Size and CV MSE and SE of subtrees:

Tree	#Tnodes	Mean MSE	SE(Mean)	BSE(Mean)	Median MSE	BSE(Median)
1+	75	2.939E+08	1.055E+07	9.968E+05	2.933E+08	1.524E+06
2	74	2.939E+08	1.055E+07	9.968E+05	2.933E+08	1.524E+06
3	73	2.939E+08	1.055E+07	9.968E+05	2.933E+08	1.524E+06
4	71	2.939E+08	1.055E+07	9.968E+05	2.933E+08	1.524E+06
5	70	2.939E+08	1.055E+07	9.968E+05	2.933E+08	1.524E+06
6*	69	2.939E+08	1.055E+07	9.968E+05	2.933E+08	1.524E+06
7	67	2.939E+08	1.055E+07	9.968E+05	2.933E+08	1.524E+06

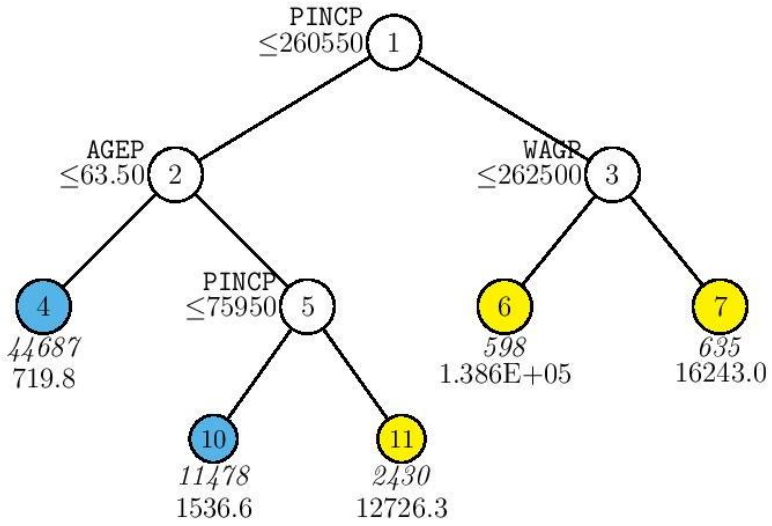
8	61	2.939E+08	1.055E+07	9.968E+05	2.933E+08
1.524E+06					
9	60	2.939E+08	1.055E+07	9.968E+05	2.933E+08
1.524E+06					
10	58	2.939E+08	1.055E+07	9.968E+05	2.933E+08
1.524E+06					
11	57	2.939E+08	1.055E+07	9.968E+05	2.933E+08
1.524E+06					
12	56	2.939E+08	1.055E+07	9.968E+05	2.933E+08
1.524E+06					
13	54	2.939E+08	1.055E+07	9.968E+05	2.933E+08
1.524E+06					
14	53	2.939E+08	1.055E+07	9.968E+05	2.933E+08
1.524E+06					
15	51	2.939E+08	1.055E+07	9.969E+05	2.933E+08
1.524E+06					
16	50	2.939E+08	1.055E+07	9.968E+05	2.933E+08
1.524E+06					
17	49	2.939E+08	1.055E+07	9.968E+05	2.933E+08
1.524E+06					
18	43	2.939E+08	1.055E+07	9.972E+05	2.933E+08
1.525E+06					
19	42	2.939E+08	1.055E+07	9.972E+05	2.933E+08
1.525E+06					
20	38	2.939E+08	1.055E+07	9.971E+05	2.933E+08
1.525E+06					
21	35	2.939E+08	1.055E+07	9.971E+05	2.933E+08
1.525E+06					
22	33	2.939E+08	1.055E+07	9.973E+05	2.933E+08
1.524E+06					
23	32	2.939E+08	1.055E+07	9.972E+05	2.933E+08
1.525E+06					
24	31	2.939E+08	1.055E+07	9.976E+05	2.933E+08
1.525E+06					
25	30	2.939E+08	1.055E+07	9.973E+05	2.933E+08
1.526E+06					
26	29	2.939E+08	1.055E+07	9.977E+05	2.933E+08
1.523E+06					
27	27	2.939E+08	1.055E+07	9.977E+05	2.933E+08
1.523E+06					
28	26	2.939E+08	1.055E+07	9.977E+05	2.933E+08
1.523E+06					
29	24	2.939E+08	1.055E+07	9.968E+05	2.933E+08
1.520E+06					
30	22	2.939E+08	1.055E+07	9.946E+05	2.933E+08
1.526E+06					
31	21	2.939E+08	1.055E+07	9.921E+05	2.933E+08
1.522E+06					
32	19	2.940E+08	1.055E+07	9.790E+05	2.934E+08
1.521E+06					
33	18	2.940E+08	1.055E+07	9.715E+05	2.935E+08
1.515E+06					
34	17	2.941E+08	1.055E+07	9.739E+05	2.935E+08
1.511E+06					

```
   35++     14   2.943E+08   1.055E+07   1.009E+06   2.937E+08
1.630E+06
   36       13   2.946E+08   1.055E+07   9.444E+05   2.941E+08
1.555E+06
   37       11   2.949E+08   1.056E+07   9.476E+05   2.954E+08
1.541E+06
   38        8   2.962E+08   1.056E+07   9.793E+05   2.954E+08
1.486E+06
   39        7   2.968E+08   1.056E+07   1.022E+06   2.958E+08
1.472E+06
   40**      5   2.989E+08   1.058E+07   1.024E+06   2.974E+08
1.674E+06
   41        3   3.045E+08   1.063E+07   1.147E+06   3.035E+08
1.263E+06
   42        2   3.045E+08   1.063E+07   1.147E+06   3.035E+08
1.263E+06
   43        1   4.156E+08   1.983E+07   1.580E+06   4.156E+08
2.022E+06

 0-SE tree based on mean is marked with * and has 69 terminal nodes
 0-SE tree based on median is marked with + and has 75 terminal nodes
 Selected-SE tree based on mean using naive SE is marked with **
 Selected-SE tree based on mean using bootstrap SE is marked with --
 Selected-SE tree based on median and bootstrap SE is marked with ++
 ++ tree same as -- tree

 Following tree is based on mean CV with naive SE estimate (**).

 Structure of final tree. Each terminal node is marked with a T.

 D-mean is mean of INTP in the node
 Cases fit give the number of cases used to fit node
 MSE is residual sum of squares divided by number of cases in node
        Node    Total    Cases Matrix    Node      Node      Split
Interacting
        label   cases      fit  rank    D-mean     MSE       variable
variable
           1    59828    59828     1  2.907E+03  4.156E+08   PINCP
           2    58595    58595     1  1.378E+03  5.275E+07   AGEP
          4T    44687    44687     1  7.198E+02  2.455E+07   PINCP
           5    13908    13908     1  3.492E+03  1.375E+08   PINCP
         10T    11478    11478     1  1.537E+03  3.208E+07   PINCP
         11T     2430     2430     1  1.273E+04  5.324E+08   PERNP
           3     1233     1233     1  7.557E+04  1.228E+10   WAGP
          6T      598      598     1  1.386E+05  1.500E+10   -
          7T      635      635     1  1.624E+04  2.470E+09   -

 Number of terminal nodes of final tree: 5
 Total number of nodes of final tree: 9
 Second best split variable (based on curvature test) at root node is
AGEP

 Regression tree:
```

```
 Node 1: PINCP <= 260550.00
   Node 2: AGEP <= 63.500000
     Node 4: INTP-mean = 719.76727
   Node 2: AGEP > 63.500000 or NA
     Node 5: PINCP <= 75950.000
       Node 10: INTP-mean = 1536.5944
     Node 5: PINCP > 75950.000 or NA
       Node 11: INTP-mean = 12726.346
 Node 1: PINCP > 260550.00 or NA
   Node 3: WAGP <= 262500.00
     Node 6: INTP-mean = 138570.35
   Node 3: WAGP > 262500.00 or NA
     Node 7: INTP-mean = 16243.008

 ************************************************************

 Predictor means below are means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap
solution see:

 1. Loh et al. (2016), "Identification of subgroups with differential
treatment effects
 for longitudinal and multiresponse variables", Statistics in Medicine,
v.35, 4837-4855.

 2. Loh et al. (2019), "Subgroups from regression trees with adjustment
for prognostic
 effects and post-selection inference", Statistics in Medicine, v.38,
545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical
Applications", Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if PINCP <= 260550.00
 PINCP mean = 54022.279
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant      2907.        34.87        0.000
 INTP mean = 2906.76
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if AGEP <= 63.500000
 AGEP mean = 47.752965
 ----------------------------
Node 4: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      719.8        30.71       0.2220E-14
 INTP mean = 719.767
 ----------------------------
Node 5: Intermediate node
```

```
 A case goes into Node 10 if PINCP <= 75950.000
 PINCP mean = 44774.967
 ---------------------------
Node 10: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      1537.        29.07        0.000
 INTP mean = 1536.59
 ---------------------------
Node 11: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1273E+05    27.19        0.000
 INTP mean = 12726.3
 ---------------------------
Node 3: Intermediate node
 A case goes into Node 6 if WAGP <= 262500.00
 WAGP mean = 271124.01
 ---------------------------
Node 6: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1386E+06    27.67        0.000
 INTP mean = 138570.
 ---------------------------
Node 7: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1624E+05    8.236        0.000
 INTP mean = 16243.0
 ---------------------------
 Proportion of variance (R-squared) explained by tree model: 0.4660

 Observed and fitted values are stored in regtree.fit
 LaTeX code for tree is in regtree.tex
 R code is stored in regtree.R
 Elapsed time in seconds: 332.89
```

GUIDE Regression Tree



GUIDE v.36.2 0.50-SE piecewise constant least-squares regression tree for predicting INTP. Tree constructed with 59828 observations (excluding observations with non-positive weight or missing values in D, T, R or Z variables). Maximum number of split levels is 30 and minimum node sample size is 598. At each split, an observation goes to the left branch if and only if the condition is satisfied. Sample size (*in italics*) and mean of INTP printed below nodes. Terminal nodes with means above and below value of 2906.8 at root node are colored yellow and skyblue, respectively. Second best split variable at root node is AGEP.

Appendix F

```
# Estimating mean via regression forest (2)

  zreg <- read.table("guide rf/grfpred.txt",header=TRUE)
  yhat <- zreg$predicted
  imputed <- (sum(w[group]*z$INTP[group])+sum(w[!group]*yhat[!group]))/sum(w)
  print(imputed)
  # 2302.425 (new INTP estimate)
  # 12295.47 (old value before Professor Loh's update)
  simple <- sum(w[group]*z$INTP[group])/sum(w[group])
  print(simple)
  # 2284.181 (new INTP estimate)
  # 2404.231 (old value before Professor Loh's update)
```

GUIDE Forest Input File

```
GUIDE       (do not edit this file unless you know what you are doing)
  36.2      (version of GUIDE that generated this file)
 1          (1=model fitting, 2=importance or DIF scoring, 3=data
conversion)
"grf.out"  (name of output file)
 2          (1=one tree, 2=ensemble)
 2          (1=bagging, 2=rforest)
 2          (1=random splits of missing values, 2=nonrandom)
 2          (1=classification, 2=regression)
 2          (1=interaction tests, 2=skip them)
"descreg.txt"  (name of data description file)
 1          (1=accept default number of trees, 2=change)
 1          (1=accept default number of variables for splitting, 2=change
it)
 1          (1=split point from quantiles, 2=use exhaustive search)
 1          (1=accept default splitting fraction, 2=change it)
 1          (1=default max. number of split levels, 2=specify no. in next
line)
 1          (1=default min. node size, 2=specify min. value in next line)
"grfpred.txt" (file name for predicted values)
 1          (rank of top variable to split root node)
```

GUIDE Forest Output File

```
             GGG   U   U  I  DDDD   EEEE
            G   G  U   U  I  D   D  E
            G      U   U  I  D   D  E
            G  GG  U   U  I  D   D  EEE
            G   G  U   U  I  D   D  E
            G   G  U   U  I  D   D  E
             GGG    UUU   I  DDDD   EEEE

 GUIDE Classification and Regression Trees and Forests
 Version 36.2 (Build date: January 10, 2021)
 Compiled with Visual Fortran 64 18.0.1.156 on Windows 10
 Copyright (c) 1997-2020 Wei-Yin Loh. All rights reserved.
 This software is based upon work supported by the U.S. Army Research
Office,
 the National Science Foundation and the National Institutes of Health.

 This job was started on 04/15/21 at 22:01

 Random forest of GUIDE least-squares regression trees
 No pruning
 Data description file: descreg.txt
 Training sample file: cleandata.txt
 Missing value code: NA
 Records in data file start on line 2
 15 N variables changed to S
 D variable is INTP
 Piecewise constant model
 Number of records in data file: 71066
 Length of longest entry in data file: 8
 Missing values found in D variable
 Missing values found among categorical variables
 Separate categories will be created for missing categorical variables
 Missing values found among non-categorical variables

 Summary information for training sample of size 59828 (excluding
observations with
 non-positive weight or missing values in d, e, t, r or z variables)
 d=dependent, b=split and fit cat variable using indicator variables,
 c=split-only categorical, i=fit-only categorical (via indicators),
 s=split-only numerical, n=split and fit numerical, f=fit-only numerical,
 m=missing-value flag variable, p=periodic variable, w=weight
                                                  #Codes/
                                                  Levels/
  Column  Name            Minimum      Maximum     Periods   #Missing
       1  PUMA       c                                 56
       2  PWGTP      s    1.000        2068.
       3  AGEP       s    15.00        94.00
       4  CIT        c                                  5
       5  CITWP      c                                 73      55371
       6  COW        c                                  9      15123
       7  DDRS       c                                  2
       8  DEAR       c                                  2
       9  DEYE       c                                  2
      10  DOUT       c                                  2
```

11	DPHY	c			2	
12	DRAT	c			6	57834
13	DRATX	c			2	52086
14	DREM	c			2	
15	ENG	c			4	51562
16	FER	c			2	44034
17	GCL	c			2	12860
18	GCM	c			5	59337
19	GCR	c			2	58439
20	HIMRKS	c			3	
21	HINS1	c			2	
22	HINS2	c			2	
23	HINS3	c			2	
24	HINS4	c			2	
25	HINS5	c			2	
26	HINS6	c			2	
27	HINS7	c			2	
28	INTP	d	-2000.	0.2610E+06		
29	JWMNP	s	1.000	149.0		26658
30	JWRIP	c			10	29677
31	JWTRNS	c			12	24345
32	LANX	c			2	
33	MAR	c			5	
34	MARHD	c			2	17084
35	MARHM	c			2	17084
36	MARHT	c			3	17084
37	MARHW	c			2	17084
38	MARHYP	c			80	17084
39	MIG	c			3	
40	MIL	c			4	1717
41	MLPA	c			2	52800
42	MLPB	c			2	52800
43	MLPCD	c			2	52800
44	MLPE	c			2	52800
45	MLPFG	c			2	52800
46	MLPH	c			2	52800
47	MLPI	c			2	52800
48	MLPJ	c			2	52800
49	MLPK	c			2	52800
50	NWAB	c			3	857
51	NWAV	c			4	857
52	NWLA	c			3	857
53	NWLK	c			3	857
54	NWRE	c			3	857
55	OIP	s	0.000	0.7500E+05		
56	PAP	s	0.000	0.1640E+05		
57	RELSHIPP	c			19	
58	RETP	s	0.000	0.1550E+06		
59	SCH	c			3	
60	SCHG	c			9	51684
61	SCHL	c			24	
62	SEMP	s	-6900.	0.4300E+06		
63	SEX	c			2	
64	SSIP	s	0.000	0.2500E+05		

65	SSP	s	0.000	0.3800E+05		
66	WAGP	s	0.000	0.5160E+06		
67	WKHP	s	1.000	99.00		20237
68	WKL	c			3	857
69	WKWN	s	1.000	52.00		20237
70	WRK	c			2	2546
71	YOEP	c			81	51371
72	ANC	c			4	
73	ANC1P	c			225	
74	ANC2P	c			191	
75	DECADE	c			8	51371
76	DIS	c			2	
77	DRIVESP	c			6	29677
78	ESP	c			8	57445
79	ESR	c			6	857
80	FOD1P	c			173	36265
81	FOD2P	c			158	56904
82	HICOV	c			2	
83	HISP	c			24	
84	INDP	c			270	15123
85	JWAP	c			285	26658
86	JWDP	c			150	26658
87	LANP	c			113	51562
88	MIGPUMA	c			161	51753
89	MIGSP	c			101	51753
90	MSP	c			6	
91	NAICSP	c			270	15123
92	NATIVITY	c			2	
93	NOP	c			8	57445
94	OC	c			2	3388
95	OCCP	c			527	15123
96	PAOC	c			4	30694
97	PERNP	s	-6900.	0.9460E+06		857
98	PINCP	s	-6900.	0.1137E+07		
99	POBP	c			218	
100	POVPIP	s	0.000	501.0		2961
101	POWPUMA	c			114	24345
102	POWSP	c			45	24345
103	PRIVCOV	c			2	
104	PUBCOV	c			2	
105	QTRBIR	c			4	
106	RAC1P	c			9	
107	RAC2P	c			53	
108	RAC3P	c			89	
109	RACAIAN	c			2	
110	RACASN	c			2	
111	RACBLK	c			2	
112	RACNH	c			2	
113	RACNUM	c			5	
114	RACPI	c			2	
115	RACSOR	c			2	
116	RACWHT	c			2	
117	RC	c			2	3388
118	SCIENGP	c			2	36265

```
    119  SCIENGRLP  c                                    2     36265
    120  SFN        c                                    2     58835
    121  SFR        c                                    6     58835
    122  SOCP       c                                  527     15123
    123  VPS        c                                   15     52800
    124  WAOB       c                                    8

     Total  #cases w/   #missing
    #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
     71066      11238      38517        1        0        0       15
    #P-var   #M-var   #B-var   #C-var   #I-var
         0        0        0      108        0
No weight variable in data file
Number of cases used for training: 59828
Number of split variables: 123
Number of cases excluded due to 0 weight or missing D: 11238

Number of trees in ensemble: 500
Number of variables used for splitting: 42
No nodewise interaction tests
Fraction of cases used for splitting each node: .0017
Maximum number of split levels: 30
Minimum node sample size: 299
Mean number of terminal nodes:    146.8
Resubstitution estimate of mean squared error: 192797506.7546
based on number of training cases: 59828
Proportion of variance (R-squared) explained by ensemble model: 0.5361

Number of OOB cases: 59828
OOB estimate of mean squared error: 198065009.3831
Mean number of trees per OOB observation: 183.93

Number of test cases with 0 weight and nonmissing responses = 0
Observed and fitted values are stored in grfpred.txt
No. times splitting stopped because node numbers were too big: 2
Elapsed time in seconds: 13867.
```

Appendix G

GUIDE Classification Scoring Input File

```
GUIDE        (do not edit this file unless you know what you are doing)
  36.2       (version of GUIDE that generated this file)
 2           (1=model fitting, 2=importance or DIF scoring, 3=data
conversion)
"classimp.out"  (name of output file)
 1           (1=classification, 2=regression, 3=propensity score grouping)
 1           (1=univariate and interaction splits, 2=skip interactions)
"descclass.txt"  (name of data description file)
 1           (1=estimated priors, 2=equal priors, 3=other priors)
 1           (1=unit misclassification costs, 2=other)
 2           (1=split point from quantiles, 2=use exhaustive search)
 1           (1=default max. number of split levels, 2=specify no. in next
line)
 1           (1=default min. node size, 2=specify min. value in next line)
 2           (0=no LaTeX code, 1=tree without node numbers, 2=tree with
node numbers)
"classimp.tex" (latex file name)
 1           (1=color terminal nodes, 2=no colors)
 2           (0=#errors, 1=sample sizes, 2=sample proportions, 3=posterior
probs, 4=nothing)
 1           (1=no storage, 2=store fit and split variables, 3=store split
variables and values)
 1           (1=do not create description file for selected variables,
2=create the file)
 1           (1=create file for importance scores, 2=do not create)
"classimp.scr" (file name for importance scores)
 1           (rank of top variable to split root node)
```

GUIDE Classification Scoring Output File

```
             GGG   U   U  I  DDDD   EEEE
            G   G  U   U  I  D   D  E
            G      U   U  I  D   D  E
            G  GG  U   U  I  D   D  EEE
            G   G  U   U  I  D   D  E
            G   G  U   U  I  D   D  E
             GGG    UUU   I  DDDD   EEEE

 GUIDE Classification and Regression Trees and Forests
 Version 36.2 (Build date: January 10, 2021)
 Compiled with Visual Fortran 64 18.0.1.156 on Windows 10
 Copyright (c) 1997-2020 Wei-Yin Loh. All rights reserved.
 This software is based upon work supported by the U.S. Army Research
Office,
 the National Science Foundation and the National Institutes of Health.

 This job was started on 04/15/21 at 22:06

 Classification tree
 No pruning
 Data description file: descclass.txt
 Training sample file: cleandata.txt
 Missing value code: NA
 Records in data file start on line 2
 15 N variables changed to S
 D variable is FINTP
 Number of records in data file: 71066
 Length of longest entry in data file: 8
 Missing values found among categorical variables
 Separate categories will be created for missing categorical variables
 Missing values found among non-categorical variables
 Number of classes: 2
 Training sample class proportions of D variable FINTP:
 Class  #Cases     Proportion
 0       59828     0.84186531
 1       11238     0.15813469

      Total  #cases w/   #missing
     #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      71066          0      32819        1        0        0       15
     #P-var   #M-var   #B-var   #C-var   #I-var
          0        0        0      108        0
 Number of cases used for training: 71066
 Number of split variables: 123
 Number of cases excluded due to 0 weight or missing D: 0

 Importance scoring of variables
 Simple node models
 Estimated priors
 Unit misclassification costs
 Univariate split highest priority
 Interaction splits 2nd priority; no linear splits
 Split values for N and S variables based on exhaustive search
 Maximum number of split levels: 4
```

```
 Minimum node sample size: 710
 Starting 300 permutations to standardize means of importance scores
 Finished permutations to standardize means of importance scores
 95 and 99% thresholds for unadjusted importance scores =   37.023
48.343

 Note: final tree is shorter due to pruning of sibling nodes with same
predicted values.
 Structure of final tree. Each terminal node is marked with a T.

 Node cost is node misclassification cost divided by number of training
cases
        Node    Total    Train    Predicted        Node    Split
Interacting
       label    cases    cases    class            cost    variables
variable
           1    71066    71066    0             1.581E-01 WRK
           2     8268     8268    1             3.079E-01 ANC
          4T     5015     5015    1             6.500E-02 PERNP
           5     3253     3253    0             3.176E-01 RELSHIPP
         10T     2323     2323    0             1.442E-01 ESR
         11T      930      930    1             2.495E-01 -
          3T    62798    62798    0             8.784E-02 ANC

 Number of terminal nodes of final tree: 4
 Total number of nodes of final tree: 7
 Second best split variable (based on curvature test) at root node is ANC

 Classification tree:
 For categorical variable splits, values not in training data go to the
right

 Node 1: WRK = "NA"
   Node 2: ANC = "4"
     Node 4: 1
   Node 2: ANC /= "4"
     Node 5: RELSHIPP = "25", "26", "27", "30", "35", "36", "37"
       Node 10: 0
     Node 5: RELSHIPP /= "25", "26", "27", "30", "35", "36", "37"
       Node 11: 1
 Node 1: WRK /= "NA"
   Node 3: 0

 ************************************************************

 Predictor means below are means of cases with no missing values.

Node 1: Intermediate node
 A case goes into Node 2 if WRK = "NA"
 WRK mode = "1"
 Class      Number   Posterior
 0           59828  0.8419E+00
 1           11238  0.1581E+00
 Number of training cases misclassified = 11238
```

```
 Predicted class is 0
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if ANC = "4"
 ANC mode = "4"
 Class      Number   Posterior
 0            2546  0.3079E+00
 1            5722  0.6921E+00
 Number of training cases misclassified = 2546
 Predicted class is 1
 ----------------------------
Node 4: Terminal node
 Class      Number   Posterior
 0             326  0.6500E-01
 1            4689  0.9350E+00
 Number of training cases misclassified = 326
 Predicted class is 1
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if RELSHIPP = "25", "26", "27", "30", "35",
"36", "37"
 RELSHIPP mode = "37"
 Class      Number   Posterior
 0            2220  0.6824E+00
 1            1033  0.3176E+00
 Number of training cases misclassified = 1033
 Predicted class is 0
 ----------------------------
Node 10: Terminal node
 Class      Number   Posterior
 0            1988  0.8558E+00
 1             335  0.1442E+00
 Number of training cases misclassified = 335
 Predicted class is 0
 ----------------------------
Node 11: Terminal node
 Class      Number   Posterior
 0             232  0.2495E+00
 1             698  0.7505E+00
 Number of training cases misclassified = 232
 Predicted class is 1
 ----------------------------
Node 3: Terminal node
 Class      Number   Posterior
 0           57282  0.9122E+00
 1            5516  0.8784E-01
 Number of training cases misclassified = 5516
 Predicted class is 0
 ----------------------------
 ****************************************************
 Variables used for splitting:
 ANC
 RELSHIPP
 WRK
```

Number of terminal nodes: 4

Scaled importance scores of predictor variables

Score	Rank	Variable
4.604E+02	1.00	WRK
4.380E+02	2.00	ANC
3.481E+02	3.00	ANC1P
5.447E+01	4.00	NWAB
5.184E+01	5.00	NWLK
5.022E+01	6.00	RELSHIPP
4.913E+01	7.00	NWLA
3.333E+01	8.00	RC
3.227E+01	9.00	OC
2.911E+01	10.00	ANC2P
2.894E+01	11.00	NWRE
2.876E+01	12.00	SCHL
2.825E+01	13.00	MSP
2.807E+01	14.00	POVPIP
2.464E+01	15.00	ESR
2.421E+01	16.00	PERNP
2.228E+01	17.00	PINCP
1.918E+01	18.00	NWAV
1.905E+01	19.00	SCHG
1.897E+01	20.00	AGEP
1.847E+01	21.00	MAR
1.802E+01	22.00	HINS3
1.746E+01	23.00	INDP
1.736E+01	24.00	NAICSP
1.721E+01	25.00	WKL
1.681E+01	26.00	WAGP
1.644E+01	27.00	MARHYP
1.623E+01	28.00	SSP
1.593E+01	29.00	PUBCOV
1.403E+01	30.00	MIL
1.364E+01	31.00	JWTRNS
1.327E+01	32.00	COW
1.307E+01	33.00	WKHP
1.282E+01	34.00	WKWN
1.258E+01	35.00	SCIENGP
1.191E+01	36.00	SCIENGRLP
1.183E+01	37.00	JWMNP
1.178E+01	38.00	POWSP
1.160E+01	39.00	RACBLK
1.145E+01	40.00	OCCP
1.116E+01	41.00	SOCP
1.112E+01	42.00	RAC1P
1.107E+01	43.00	RACWHT
1.098E+01	44.00	JWRIP
1.091E+01	45.00	POWPUMA
1.089E+01	46.00	PUMA
1.041E+01	47.00	DRIVESP
1.032E+01	48.00	PAOC
1.022E+01	49.00	FOD1P

```
 8.936E+00    50.00  RAC2P
 8.898E+00    51.00  RAC3P
 8.768E+00    52.00  PRIVCOV
 8.697E+00    53.00  NOP
 8.277E+00    54.00  ESP
 7.559E+00    55.00  JWDP
 7.144E+00    56.00  MARHW
 6.623E+00    57.00  HINS1
 6.482E+00    58.00  JWAP
 6.272E+00    59.00  HINS4
 6.248E+00    60.00  GCL
 6.150E+00    61.00  POBP
 6.143E+00    62.00  RETP
 5.811E+00    63.00  SCH
 5.807E+00    64.00  MIGPUMA
 5.760E+00    65.00  DPHY
 5.377E+00    66.00  MARHM
 5.288E+00    67.00  MARHD
 5.137E+00    68.00  HINS2
 5.039E+00    69.00  MARHT
 5.035E+00    70.00  DIS
 5.033E+00    71.00  HIMRKS
 4.270E+00    72.00  DDRS
 4.200E+00    73.00  DRATX
 4.097E+00    74.00  VPS
 3.879E+00    75.00  DOUT
 3.177E+00    76.00  MLPH
 3.136E+00    77.00  HISP
 2.915E+00    78.00  PWGTP
 2.842E+00    79.00  FER
 2.737E+00    80.00  LANP
 2.546E+00    81.00  DRAT
 2.379E+00    82.00  HINS7
 2.351E+00    83.00  DEYE
 2.252E+00    84.00  DREM
 2.146E+00    85.00  MIGSP
 1.857E+00    86.00  SFR
 1.817E+00    87.00  CIT
 1.768E+00    88.00  MIG
 1.757E+00    89.00  ENG
 1.753E+00    90.00  LANX
 1.661E+00    91.00  MLPA
 1.625E+00    92.00  WAOB
 1.596E+00    93.00  MLPFG
 1.565E+00    94.00  SSIP
 1.371E+00    95.00  MLPCD
 1.330E+00    96.00  YOEP
 1.312E+00    97.00  NATIVITY
 1.292E+00    98.00  DECADE
 1.279E+00    99.00  HICOV
 1.212E+00   100.00  MLPB
--------- variables above this line are highly important -----
 1.192E+00   101.00  MLPI
 1.178E+00   102.00  SEX
```

```
 1.130E+00   103.00  CITWP
 1.065E+00   104.00  MLPJ
 1.062E+00   105.00  RACSOR
 1.053E+00   106.00  GCR
 1.046E+00   107.00  DEAR
 1.043E+00   108.00  SEMP
 1.027E+00   109.00  MLPE
--------- variables below this line are unimportant ----------
 9.726E-01   110.00  RACASN
 8.751E-01   111.00  HINS6
 8.255E-01   112.00  MLPK
 8.215E-01   113.00  RACNH
 7.503E-01   114.00  HINS5
 7.302E-01   115.00  GCM
 7.302E-01   116.00  SFN
 6.585E-01   117.00  RACPI
 6.188E-01   118.00  OIP
 2.203E-01   119.00  PAP
 1.858E-01   120.00  FOD2P
 1.281E-01   121.00  RACNUM
 9.521E-02   122.00  RACAIAN
 5.704E-02   123.00  QTRBIR

 Variables with scores above 1.20 are highly important
 Variables with scores between 1.0 and 1.20 are likely important
 Variables with scores below 1.0 are unimportant
 No. highly important, likely important, and unimportant split variables:
100, 9, 14
 LaTeX code for tree is in classimp.tex
 Importance scores are stored in classimp.scr
 Elapsed time in seconds: 7949.3
```

Appendix H

GUIDE Regression Scoring Input File

```
GUIDE       (do not edit this file unless you know what you are doing)
  36.2      (version of GUIDE that generated this file)
 2          (1=model fitting, 2=importance or DIF scoring, 3=data conversion)
"regtimp.out"  (name of output file)
 2          (1=classification, 2=regression, 3=propensity score grouping)
 1          (1=linear, 2=quantile, 3=Poisson, 4=censored response,
5=multiresponse or itemresponse, 6=longitudinal with T vars, 7=logistic)
 1          (1=least squares, 2=least median of squares)
 1          (1=interaction tests, 2=skip them)
"descreg.txt"  (name of data description file)
 2          (1=split point from quantiles, 2=use exhaustive search)
 1          (1=default max. number of split levels, 2=specify no. in next line)
 1          (1=default min. node size, 2=specify min. value in next line)
 2          (0=no LaTeX code, 1=tree without node numbers, 2=tree with node
numbers)
"regtimp.tex" (latex file name)
 1          (0=all white,1=yellow-skyblue,2=yellow-purple,3=yellow-
orange,4=orange-skyblue,5=yellow-red,6=orange-purple,7=grayscale)
 1          (1=no storage, 2=store fit and split variables, 3=store split
variables and values)
 1          (1=do not save, 2=save regressor names in a file)
 1          (1=do not create description file for selected variables, 2=create
the file)
 1          (1=create file for importance scores, 2=do not create)
"regtimp.scr" (file name for importance scores)
 1          (rank of top variable to split root node)
```

GUIDE Regression Scoring Output File

```
             GGG   U   U  I  DDDD   EEEE
            G   G  U   U  I  D   D  E
            G      U   U  I  D   D  E
            G  GG  U   U  I  D   D  EEE
            G   G  U   U  I  D   D  E
            G   G  U   U  I  D   D  E
             GGG    UUU   I  DDDD   EEEE

 GUIDE Classification and Regression Trees and Forests
 Version 36.2 (Build date: January 10, 2021)
 Compiled with Visual Fortran 64 18.0.1.156 on Windows 10
 Copyright (c) 1997-2020 Wei-Yin Loh. All rights reserved.
 This software is based upon work supported by the U.S. Army Research Office,
 the National Science Foundation and the National Institutes of Health.

 This job was started on 04/15/21 at 22:08

 Least squares regression tree
 No pruning
 Data description file: descreg.txt
 Training sample file: cleandata.txt
 Missing value code: NA
 Records in data file start on line 2
 15 N variables changed to S
 D variable is INTP
 Piecewise constant model
 Number of records in data file: 71066
 Length of longest entry in data file: 8
 Missing values found in D variable
 Missing values found among categorical variables
 Separate categories will be created for missing categorical variables
 Missing values found among non-categorical variables

      Total  #cases w/   #missing
     #cases    miss. D  ord. vals   #X-var   #N-var   #F-var   #S-var
      71066      11238      38517        1        0        0       15
     #P-var   #M-var   #B-var   #C-var   #I-var
          0        0        0      108        0
 No weight variable in data file
 Number of cases used for training: 59828
 Number of split variables: 123
 Number of cases excluded due to 0 weight or missing D: 11238

 Importance scoring of variables
 Nodewise interaction tests on all variables
 Split values for N and S variables based on exhaustive search
 Maximum number of split levels: 4
 Minimum node sample size: 598
 Starting 300 permutations to standardize means of importance scores
 Finished permutations to standardize means of importance scores
 95 and 99% thresholds for unadjusted importance scores =   61.442  77.858

 Structure of final tree. Each terminal node is marked with a T.

 D-mean is mean of INTP in the node
 Cases fit give the number of cases used to fit node
 MSE is residual sum of squares divided by number of cases in node
        Node    Total    Cases Matrix    Node      Node      Split
Interacting
        label   cases      fit  rank    D-mean     MSE       variable
variable
           1    59828    59828     1  2.907E+03  4.156E+08   PINCP
           2    58595    58595     1  1.378E+03  5.275E+07   AGEP
           4    44687    44687     1  7.198E+02  2.455E+07   PINCP
```

```
          8    38725    38725     1  3.862E+02  1.083E+07   SCIENGRLP
         16T   26210    26210     1  1.835E+02  4.079E+06   AGEP
         17T   12515    12515     1  8.107E+02  2.470E+07   AGEP
          9     5962     5962     1  2.886E+03  1.083E+08   PINCP
         18T    4450     4450     1  2.130E+03  7.929E+07   WAGP
         19T    1512     1512     1  5.113E+03  1.871E+08   PERNP
          5    13908    13908     1  3.492E+03  1.375E+08   PINCP
         10    11478    11478     1  1.537E+03  3.208E+07   PINCP
         20T    8282     8282     1  6.764E+02  7.970E+06   PINCP
         21T    3196     3196     1  3.766E+03  8.769E+07   WAGP
         11     2430     2430     1  1.273E+04  5.324E+08   PERNP
         22T    1414     1414     1  1.853E+04  7.251E+08   RETP
         23T    1016     1016     1  4.647E+03  1.523E+08   -
          3     1233     1233     1  7.557E+04  1.228E+10   WAGP
          6T     598      598     1  1.386E+05  1.500E+10   -
          7T     635      635     1  1.624E+04  2.470E+09   -

 Number of terminal nodes of final tree: 10
 Total number of nodes of final tree: 19
 Second best split variable (based on curvature test) at root node is AGEP

 Regression tree:
 For categorical variable splits, values not in training data go to the right

 Node 1: PINCP <= 260550.00
   Node 2: AGEP <= 63.500000
     Node 4: PINCP <= 100650.00
       Node 8: SCIENGRLP = "NA"
         Node 16: INTP-mean = 183.48493
       Node 8: SCIENGRLP /= "NA"
         Node 17: INTP-mean = 810.70156
     Node 4: PINCP > 100650.00 or NA
       Node 9: PINCP <= 166360.00
         Node 18: INTP-mean = 2129.9348
       Node 9: PINCP > 166360.00 or NA
         Node 19: INTP-mean = 5113.0688
   Node 2: AGEP > 63.500000 or NA
     Node 5: PINCP <= 75950.000
       Node 10: PINCP <= 38980.000
         Node 20: INTP-mean = 676.41391
       Node 10: PINCP > 38980.000 or NA
         Node 21: INTP-mean = 3765.6352
     Node 5: PINCP > 75950.000 or NA
       Node 11: PERNP <= 33950.000
         Node 22: INTP-mean = 18531.775
       Node 11: PERNP > 33950.000 or NA
         Node 23: INTP-mean = 4646.7421
 Node 1: PINCP > 260550.00 or NA
   Node 3: WAGP <= 262500.00
     Node 6: INTP-mean = 138570.35
   Node 3: WAGP > 262500.00 or NA
     Node 7: INTP-mean = 16243.008

 ************************************************************

 Predictor means below are means of cases with no missing values.

 WARNING: p-values below not adjusted for split search. For a bootstrap solution
see:

 1. Loh et al. (2016), "Identification of subgroups with differential treatment
effects
 for longitudinal and multiresponse variables", Statistics in Medicine, v.35,
4837-4855.
```

```
 2. Loh et al. (2019), "Subgroups from regression trees with adjustment for
prognostic
 effects and post-selection inference", Statistics in Medicine, v.38, 545-557.

 3. Loh and Zhou (2020), "The GUIDE approach to subgroup identification",
 in "Design and Analysis of Subgroups with Biopharmaceutical Applications",
Springer, pp.147-165.

Node 1: Intermediate node
 A case goes into Node 2 if PINCP <= 260550.00
 PINCP mean = 54022.279
 Coefficients of least squares regression function:
 Regressor    Coefficient  t-stat      p-value
 Constant      2907.        34.87        0.000
 INTP mean = 2906.76
 ----------------------------
Node 2: Intermediate node
 A case goes into Node 4 if AGEP <= 63.500000
 AGEP mean = 47.752965
 ----------------------------
Node 4: Intermediate node
 A case goes into Node 8 if PINCP <= 100650.00
 PINCP mean = 46786.736
 ----------------------------
Node 8: Intermediate node
 A case goes into Node 16 if SCIENGRLP = "NA"
 SCIENGRLP mode = "NA"
 ----------------------------
Node 16: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      183.5        14.71        0.000
 INTP mean = 183.485
 ----------------------------
Node 17: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      810.7        18.25       0.1665E-14
 INTP mean = 810.702
 ----------------------------
Node 9: Intermediate node
 A case goes into Node 18 if PINCP <= 166360.00
 PINCP mean = 147277.26
 ----------------------------
Node 18: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      2130.        15.96        0.000
 INTP mean = 2129.93
 ----------------------------
Node 19: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      5113.        14.54        0.000
 INTP mean = 5113.07
 ----------------------------
Node 5: Intermediate node
 A case goes into Node 10 if PINCP <= 75950.000
 PINCP mean = 44774.967
 ----------------------------
Node 10: Intermediate node
 A case goes into Node 20 if PINCP <= 38980.000
 PINCP mean = 27872.132
```

```
 ---------------------------
Node 20: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      676.4        21.81        0.000
 INTP mean = 676.414
 ---------------------------
Node 21: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      3766.        22.73        0.000
 INTP mean = 3765.64
 ---------------------------
Node 11: Intermediate node
 A case goes into Node 22 if PERNP <= 33950.000
 PERNP mean = 45241.593
 ---------------------------
Node 22: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1853E+05    25.88        0.000
 INTP mean = 18531.8
 ---------------------------
Node 23: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant      4647.        12.00        0.000
 INTP mean = 4646.74
 ---------------------------
Node 3: Intermediate node
 A case goes into Node 6 if WAGP <= 262500.00
 WAGP mean = 271124.01
 ---------------------------
Node 6: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1386E+06    27.67        0.000
 INTP mean = 138570.
 ---------------------------
Node 7: Terminal node
 Coefficients of least squares regression functions:
 Regressor    Coefficient  t-stat      p-value
 Constant     0.1624E+05    8.236        0.000
 INTP mean = 16243.0
 ---------------------------
 ****************************************************
 Variables used for splitting or fitting:
 AGEP
 PERNP
 PINCP
 SCIENGRLP
 WAGP

 Number of terminal nodes: 10

 Scaled importance scores of predictor variables
      Score     Rank  Variable
  1.463E+02     1.00  PINCP
  1.013E+02     2.00  AGEP
  7.939E+01     3.00  RETP
  7.161E+01     4.00  SCIENGP
  7.048E+01     5.00  SCIENGRLP
  6.881E+01     6.00  SCHL
  5.739E+01     7.00  HINS3
```

5.343E+01	8.00	POVPIP
4.419E+01	9.00	SSP
4.227E+01	10.00	FOD1P
4.138E+01	11.00	MARHYP
3.957E+01	12.00	WAGP
3.690E+01	13.00	GCL
3.650E+01	14.00	RELSHIPP
3.454E+01	15.00	PERNP
3.427E+01	16.00	MSP
3.422E+01	17.00	MAR
3.368E+01	18.00	PUBCOV
3.005E+01	19.00	MARHM
2.769E+01	20.00	FER
2.622E+01	21.00	MARHT
2.559E+01	22.00	MARHW
2.453E+01	23.00	MARHD
2.415E+01	24.00	OCCP
2.247E+01	25.00	ANC1P
2.234E+01	26.00	VPS
2.149E+01	27.00	SOCP
2.047E+01	28.00	RACWHT
1.997E+01	29.00	RACBLK
1.986E+01	30.00	COW
1.952E+01	31.00	INDP
1.946E+01	32.00	NWLK
1.936E+01	33.00	WRK
1.901E+01	34.00	RC
1.877E+01	35.00	PUMA
1.725E+01	36.00	NAICSP
1.674E+01	37.00	OC
1.618E+01	38.00	WKWN
1.604E+01	39.00	HINS2
1.600E+01	40.00	WKHP
1.571E+01	41.00	MIL
1.555E+01	42.00	MLPE
1.538E+01	43.00	ESR
1.534E+01	44.00	NWLA
1.533E+01	45.00	RAC1P
1.511E+01	46.00	WKL
1.490E+01	47.00	HIMRKS
1.451E+01	48.00	MLPFG
1.421E+01	49.00	NWAB
1.384E+01	50.00	HINS4
1.337E+01	51.00	SCH
1.263E+01	52.00	SCHG
1.204E+01	53.00	DRATX
1.174E+01	54.00	PWGTP
1.140E+01	55.00	MLPA
1.133E+01	56.00	PRIVCOV
1.116E+01	57.00	RAC2P
1.075E+01	58.00	MLPH
1.062E+01	59.00	MLPJ
1.049E+01	60.00	MLPCD
1.005E+01	61.00	JWMNP
9.947E+00	62.00	RAC3P
9.793E+00	63.00	POBP
9.745E+00	64.00	MLPI
9.555E+00	65.00	POWPUMA
9.099E+00	66.00	JWTRNS
9.077E+00	67.00	MLPK
8.932E+00	68.00	MLPB
8.193E+00	69.00	POWSP
8.151E+00	70.00	ANC
8.075E+00	71.00	JWDP

```
 7.926E+00    72.00  DRIVESP
 7.888E+00    73.00  JWRIP
 7.796E+00    74.00  DECADE
 7.413E+00    75.00  NOP
 7.243E+00    76.00  HICOV
 6.819E+00    77.00  ESP
 5.976E+00    78.00  HINS5
 5.613E+00    79.00  ANC2P
 5.586E+00    80.00  PAOC
 4.674E+00    81.00  JWAP
 4.524E+00    82.00  FOD2P
 4.337E+00    83.00  SEMP
 4.049E+00    84.00  YOEP
 3.863E+00    85.00  MIG
 3.725E+00    86.00  SEX
 3.389E+00    87.00  DEAR
 3.250E+00    88.00  NWRE
 2.926E+00    89.00  SSIP
 2.832E+00    90.00  DREM
 2.804E+00    91.00  CITWP
 2.799E+00    92.00  CIT
 2.334E+00    93.00  WAOB
 2.212E+00    94.00  NWAV
 2.157E+00    95.00  HINS1
 2.077E+00    96.00  ENG
 2.069E+00    97.00  OIP
 1.909E+00    98.00  LANX
 1.775E+00    99.00  HISP
 1.603E+00   100.00  DRAT
 1.508E+00   101.00  PAP
 1.455E+00   102.00  RACSOR
 1.414E+00   103.00  MIGPUMA
 1.391E+00   104.00  SFN
 1.358E+00   105.00  LANP
 1.219E+00   106.00  MIGSP
--------- variables above this line are highly important -----
 1.087E+00   107.00  SFR
 1.018E+00   108.00  DIS
--------- variables below this line are unimportant ----------
 9.824E-01   109.00  NATIVITY
 9.596E-01   110.00  DOUT
 8.109E-01   111.00  DPHY
 7.712E-01   112.00  DEYE
 7.093E-01   113.00  GCR
 6.977E-01   114.00  RACNUM
 6.937E-01   115.00  DDRS
 6.135E-01   116.00  RACASN
 5.574E-01   117.00  HINS6
 4.966E-01   118.00  RACAIAN
 2.954E-01   119.00  HINS7
 1.745E-01   120.00  GCM
 1.502E-01   121.00  RACPI
 1.152E-01   122.00  RACNH
 3.184E-02   123.00  QTRBIR

 Variables with scores above 1.15 are highly important
 Variables with scores between 1.0 and 1.15 are likely important
 Variables with scores below 1.0 are unimportant
 No. highly important, likely important, and unimportant split variables: 106,
2, 15
 LaTeX code for tree is in regtimp.tex
 Importance scores are stored in regtimp.scr
 Elapsed time in seconds: 24625.
```

Appendix I

GUIDE Classification Subset Input File

```
GUIDE       (do not edit this file unless you know what you are doing)
  36.2      (version of GUIDE that generated this file)
 3          (1=model fitting, 2=importance or DIF scoring, 3=data conversion)
"classsubset.out"  (name of output file)
 1          (1=D is categorical, 2=D is real)
"descclass.txt"  (name of data description file)
  1         (choice of data format)
"classsubset.txt" (name of new data file)
```

GUIDE Classification Subset Output File

```
            GGG   U   U  I  DDDD   EEEE
           G   G  U   U  I  D   D  E
           G      U   U  I  D   D  E
           G  GG  U   U  I  D   D  EEE
           G   G  U   U  I  D   D  E
           G   G  U   U  I  D   D  E
            GGG    UUU   I  DDDD   EEEE

GUIDE Classification and Regression Trees and Forests
Version 36.2 (Build date: January 10, 2021)
Compiled with Visual Fortran 64 18.0.1.156 on Windows 10
Copyright (c) 1997-2020 Wei-Yin Loh. All rights reserved.
This software is based upon work supported by the U.S. Army Research Office,
the National Science Foundation and the National Institutes of Health.

This job was started on 04/16/21 at 13:01

No pruning
Data description file: descclass.txt
Training sample file: cleandata.txt
Missing value code: NA
Records in data file start on line 2
16 N variables changed to S
D variable is FINTP
Number of records in data file: 71066
Length of longest entry in data file: 8
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Number of classes: 2
Input your choice:            1
```

GUIDE Classification Subset Preview

```
# vartype <- rep("factor",125)
# vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
# z <- read.table("classsubset.txt",header=TRUE,colClasses=vartype)
#
PUMA PWGTP AGEP CIT CITWP COW DDRS DEAR DEYE DOUT DPHY DRAT DRATX DREM ENG FER GCL GCM GCR HIMRKS HINS1 HINS2 HINS
"51044" 52 19 "1" NA NA "2" "2" "2" "2" "2" NA NA "2" NA NA NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" NA NA NA NA "
"51245" 45 46 "4" "2006" "1" "2" "2" "2" "2" "2" NA NA "2" "2" NA "2" NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA N
"51255" 5 73 "1" NA NA "1" "2" "2" "2" "1" NA NA "2" NA NA "2" NA NA "2" "2" "1" "1" "2" "2" "2" "2" 0 NA NA NA "2
"51155" 49 21 "5" NA "1" "2" "2" "2" "2" "2" NA NA "2" "3" NA NA NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 NA NA NA
"51090" 40 19 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" "1" NA NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 NA NA NA
"51097" 33 34 "1" NA NA "2" "2" "2" "2" "2" NA NA "1" NA NA "2" NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA NA NA "
"51110" 109 19 "1" NA NA "2" "2" "2" "1" "2" NA NA "1" NA "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 NA NA NA
"51154" 115 22 "1" NA "5" "2" "2" "2" "2" "2" NA "2" "2" NA NA NA NA NA "0" "2" "1" "2" "2" "1" "2" "2" 0 5 "1" "1
"51010" 57 58 "1" NA NA "2" "2" "1" "2" "1" NA NA "1" NA NA "2" NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA NA NA "
"51215" 72 61 "1" NA NA "2" "2" "1" "2" "2" NA NA "1" NA NA "2" NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 NA NA NA "
"51105" 52 43 "4" "2012" NA "2" "2" "2" "2" "2" NA NA "2" "3" NA "2" NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA NA
"51045" 69 18 "1" NA "4" "2" "2" "2" "2" "2" NA NA "2" NA "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" NA 30 "1" "
"51135" 58 29 "1" NA NA "2" "2" "2" "2" "2" NA NA "2" NA NA NA NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 NA NA NA "2
"51110" 62 19 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" NA NA NA NA NA "2" "2" "1" "2" "2" "2" "2" "2" 0 NA NA NA "
"51135" 21 69 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" NA NA "2" NA NA "0" "2" "2" "1" "1" "2" "2" "2" 0 NA NA NA
"51155" 98 17 "1" NA NA "2" "2" "2" "2" "2" NA NA "1" NA NA NA NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 NA NA NA "2
"51045" 74 46 "1" NA NA "1" "2" "2" "1" "1" NA NA "2" NA NA "2" NA NA "0" "2" "2" "1" "1" "2" "2" "2" 0 NA NA NA "
"51186" 40 19 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" NA "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 10 NA "16
"51090" 68 21 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" "1" "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 25 NA "2
"51040" 88 21 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" NA NA NA NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 5 NA "10"
"51125" 16 83 "1" NA NA "1" "2" "2" "1" "1" NA NA "1" NA NA "2" NA NA "0" "2" "2" "1" "1" "2" "2" "2" NA NA NA NA
"51096" 10 20 "1" NA "2" "2" "2" "2" "2" "2" NA NA "2" NA "2" NA NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 10 NA "16
"51040" 25 17 "5" NA NA "2" "2" "2" "2" "2" NA NA "2" "1" NA NA NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA NA NA "
"51105" 29 50 "1" NA NA "2" "2" "2" "2" "2" NA NA "2" NA NA "2" NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 NA NA NA "
"51045" 8 21 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" NA "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" NA 30 "1" "1
"51154" 7 22 "1" NA "1" "2" "2" "2" "2" "2" NA NA "1" NA NA NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 NA NA NA "2
"51105" 30 31 "1" NA NA "2" "2" "2" "2" "1" NA NA "1" NA NA "2" NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA NA NA "
"51154" 88 21 "1" NA "5" "2" "2" "2" "2" "2" NA "2" "2" NA NA NA NA NA "0" "2" "2" "2" "2" "1" "2" "2" 0 8 "1" "1"
"51090" 55 18 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" "1" "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 NA NA NA
```

Appendix J

GUIDE Regression Subset Input File

```
GUIDE       (do not edit this file unless you know what you are doing)
  36.2      (version of GUIDE that generated this file)
 3          (1=model fitting, 2=importance or DIF scoring, 3=data conversion)
"regsubset.out"  (name of output file)
 2          (1=D is categorical, 2=D is real)
"descreg.txt"  (name of data description file)
  1         (choice of data format)
"regsubset.txt" (name of new data file)
```

GUIDE Regression Subset Output File

```
            GGG   U   U  I  DDDD   EEEE
           G   G  U   U  I  D   D  E
           G      U   U  I  D   D  E
           G  GG  U   U  I  D   D  EEE
           G   G  U   U  I  D   D  E
           G   G  U   U  I  D   D  E
            GGG    UUU   I  DDDD   EEEE

GUIDE Classification and Regression Trees and Forests
Version 36.2 (Build date: January 10, 2021)
Compiled with Visual Fortran 64 18.0.1.156 on Windows 10
Copyright (c) 1997-2020 Wei-Yin Loh. All rights reserved.
This software is based upon work supported by the U.S. Army Research Office,
the National Science Foundation and the National Institutes of Health.

This job was started on 04/16/21 at 13:02

No pruning
Data description file: descreg.txt
Training sample file: cleandata.txt
Missing value code: NA
Records in data file start on line 2
15 N variables changed to S
D variable is INTP
Number of records in data file: 71066
Length of longest entry in data file: 8
Missing values found in D variable
Missing values found among categorical variables
Separate categories will be created for missing categorical variables
Missing values found among non-categorical variables
Input your choice:           1
```

GUIDE Regression Subset Preview

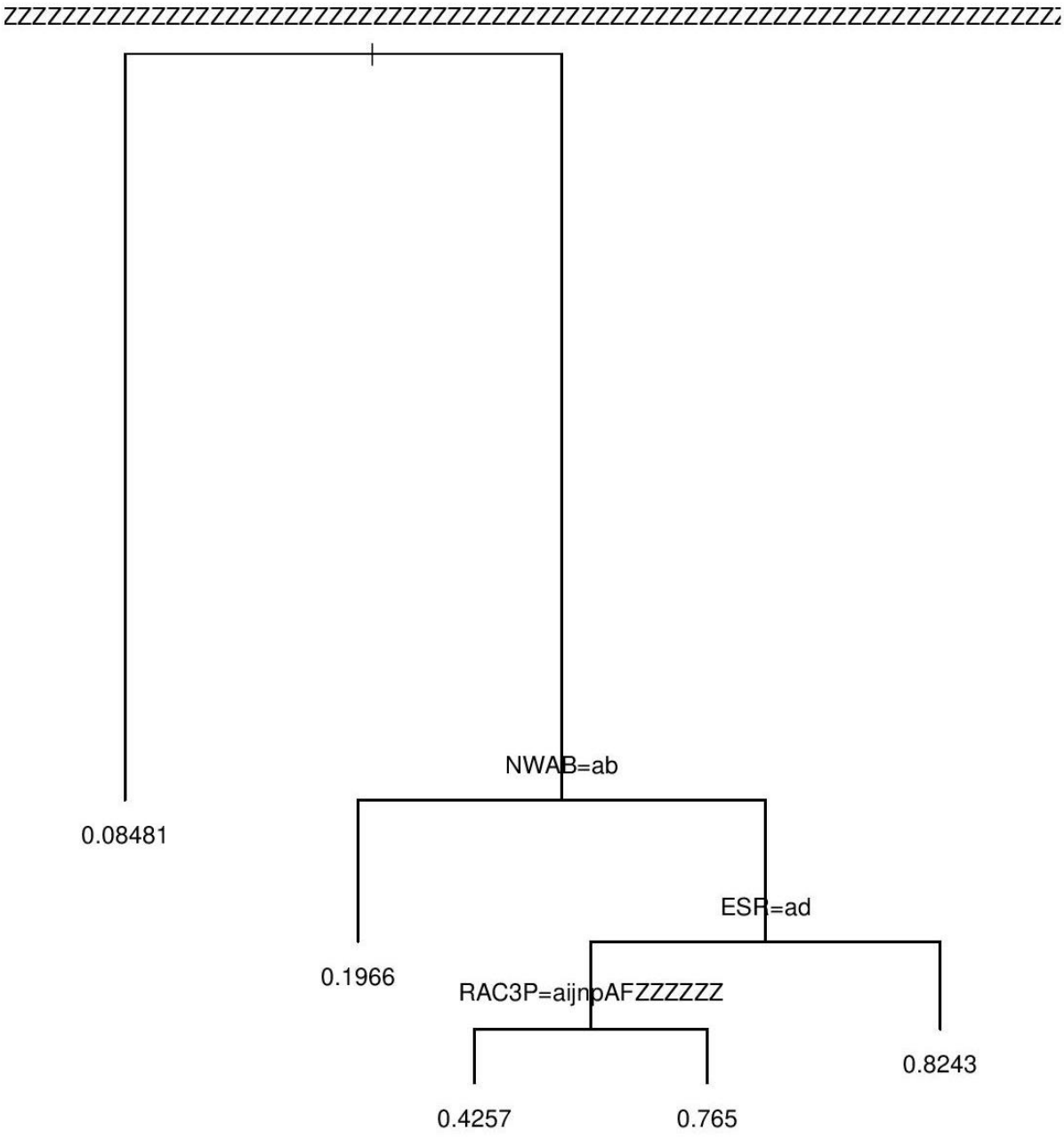
```
# vartype <- rep("factor",125)
# vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
# z <- read.table("classsubset.txt",header=TRUE,colClasses=vartype)
#
PUMA PWGTP AGEP CIT CITWP COW DDRS DEAR DEYE DOUT DPHY DRAT DRATX DREM ENG FER GCL GCM GCR HIMRKS HINS1 HINS2 HINS
"51044" 52 19 "1" NA NA "2" "2" "2" "2" "2" NA NA "2" NA NA NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" NA NA NA NA "
"51245" 45 46 "4" "2006" "1" "2" "2" "2" "2" "2" NA NA "2" "2" NA "2" NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA N
"51255" 5 73 "1" NA NA "1" "2" "2" "2" "1" NA NA "2" NA NA "2" NA NA "2" "2" "1" "1" "2" "2" "2" "2" 0 NA NA NA "2
"51155" 49 21 "5" NA "1" "2" "2" "2" "2" "2" NA NA "2" "3" NA NA NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 NA NA NA
"51090" 40 19 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" "1" NA NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 NA NA NA
"51097" 33 34 "1" NA NA "2" "2" "2" "2" "2" NA NA "1" NA NA "2" NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA NA NA "
"51110" 109 19 "1" NA NA "2" "2" "2" "1" "2" NA NA "1" NA "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 NA NA NA
"51154" 115 22 "1" NA "5" "2" "2" "2" "2" "2" NA "2" "2" NA NA NA NA NA "0" "2" "1" "2" "2" "1" "2" "2" 0 5 "1" "1
"51010" 57 58 "1" NA NA "2" "2" "1" "2" "1" NA NA "1" NA NA "2" NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA NA NA "
"51215" 72 61 "1" NA NA "2" "2" "1" "2" "2" NA NA "1" NA NA "2" NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 NA NA NA "
"51105" 52 43 "4" "2012" NA "2" "2" "2" "2" "2" NA NA "2" "3" NA "2" NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA NA
"51045" 69 18 "1" NA "4" "2" "2" "2" "2" "2" NA NA "2" NA "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" NA 30 "1" "
"51135" 58 29 "1" NA NA "2" "2" "2" "2" "2" NA NA "2" NA NA NA NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 NA NA NA "2
"51110" 62 19 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" NA NA NA NA NA "2" "2" "1" "2" "2" "2" "2" "2" 0 NA NA NA "
"51135" 21 69 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" NA NA "2" NA NA "0" "2" "2" "1" "1" "2" "2" "2" 0 NA NA NA
"51155" 98 17 "1" NA NA "2" "2" "2" "2" "2" NA NA "1" NA NA NA NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 NA NA NA "2
"51045" 74 46 "1" NA NA "1" "2" "2" "1" "1" NA NA "2" NA NA "2" NA NA "0" "2" "2" "1" "1" "2" "2" "2" 0 NA NA NA "
"51186" 40 19 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" NA "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 10 NA "16
"51090" 68 21 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" "1" "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 25 NA "2
"51040" 88 21 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" NA NA NA NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 5 NA "10"
"51125" 16 83 "1" NA NA "1" "2" "2" "1" "1" NA NA "1" NA NA "2" NA NA "0" "2" "2" "1" "1" "2" "2" "2" NA NA NA NA
"51096" 10 20 "1" NA "2" "2" "2" "2" "2" "2" "2" NA NA "2" NA "2" NA NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 10 NA "16
"51040" 25 17 "5" NA NA "2" "2" "2" "2" "2" NA NA "2" "1" NA NA NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA NA NA "
"51105" 29 50 "1" NA NA "2" "2" "2" "2" "2" NA NA "2" NA NA "2" NA NA "0" "2" "2" "2" "1" "2" "2" "2" 0 NA NA NA "
"51045" 8 21 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" NA "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" NA 30 "1" "1
"51154" 7 22 "1" NA "1" "2" "2" "2" "2" "2" NA NA "1" NA NA NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 NA NA NA "2
"51105" 30 31 "1" NA NA "2" "2" "2" "2" "1" NA NA "1" NA NA "2" NA NA "0" "2" "2" "2" "2" "2" "2" "2" 0 NA NA NA "
"51154" 88 21 "1" NA "5" "2" "2" "2" "2" "2" NA "2" "2" NA NA NA NA NA "0" "2" "2" "2" "2" "1" "2" "2" 0 8 "1" "1"
"51090" 55 18 "1" NA "1" "2" "2" "2" "2" "2" NA NA "2" "1" "2" NA NA NA "0" "1" "2" "2" "2" "2" "2" "2" 0 NA NA NA
```

Appendix K

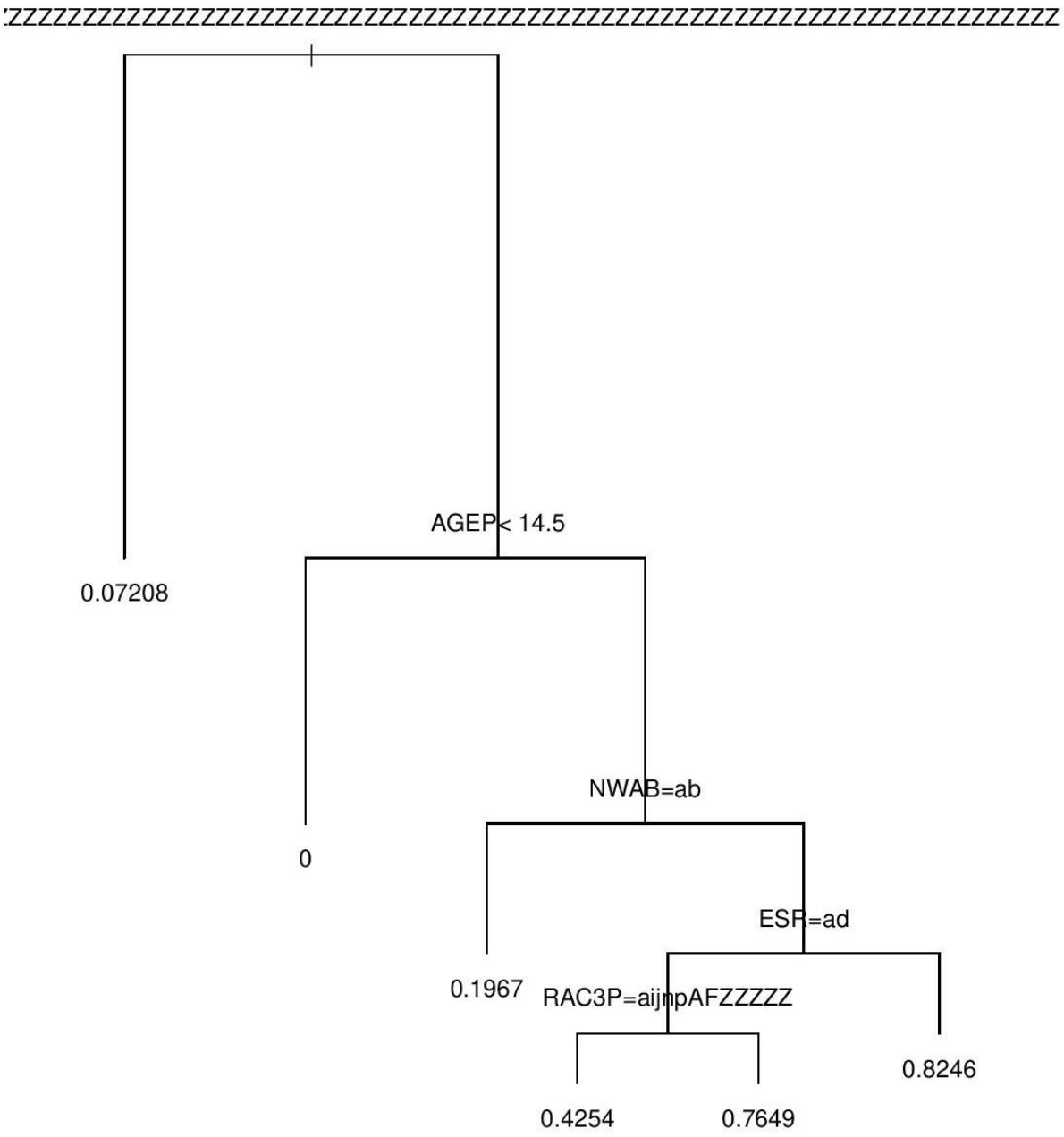
RPART Classification Tree Code

```
#3 Repeat steps 1 and 2 via RPART
library(rpart)

vartype <- rep("factor",125)
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("class subset/classsubset.txt",header=TRUE,colClasses=vartype)

# R code for IPW estimate via RPART
tmp <- rep(NA,nrow(z))
tmp[z$FINTP == 0] <- 0
tmp[z$FINTP == 1] <- 1
z$FINTP <- tmp ### convert FINTP to binary variable (which it already is)

### regression tree without INTP and PWGTP
rp <- rpart(FINTP ~ . - INTP - PWGTP, data=z, method="anova")
plot(rp,compress=TRUE,margin=0.1)
text(rp) ### plot is on next page
p <- predict(rp) ### predicted prob(FINTP = 1)
w <- z$PWGTP
y <- z$INTP
gp <- !is.na(y)
ipw <- sum(w[gp]*y[gp]/p[gp])/sum(w[gp]/p[gp])
print(ipw)
# IPW =  2364.894 (new INTP estimate)
# IPW = 2540.947 (old value before Professor Loh's update)

#creating a confusion matrix to access accuracy for RPART classification tree
p2 =(p>=0.5)+0
table(z$FINTP, p2)
print((841 + 7781)/71066) # error rate of 0.1213238 for this dataset, worse
than GUIDE!
```

RPART Classification Tree (After Professor Loh's Announcement)



RPART Classification Tree (Before Removing Imputations (Old))



RPART Confusion Matrix

	0	1
0	58987	841
1	7781	3457

RPART Classification Text Format

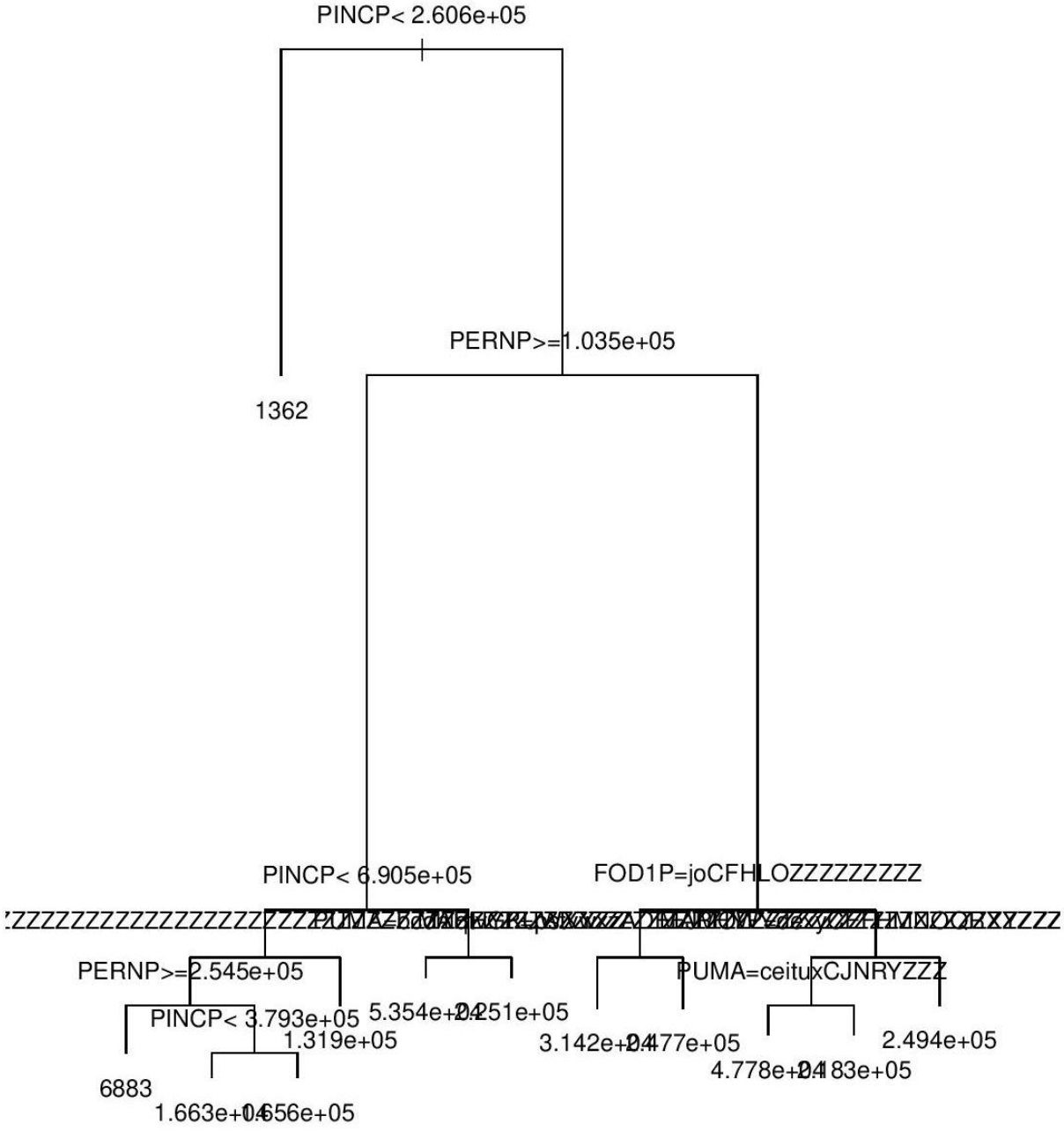
```
> rp
n= 71066

node), split, n, deviance, yval
      * denotes terminal node

 1) root 71066 9460.8820 0.15813470
   2) ANC1P=1,100,102,103,109,11,111,112,114,115,12,122,124,125,128,129,130,131,142,144,148,15
2,153,154,168,170,171,176,177,178,181,183,185,187,190,194,195,20,200,21,210,211,212,213,215,21
8,219,22,221,222,223,224,225,226,227,231,232,233,234,235,236,237,238,239,24,249,250,252,26,26
1,271,275,290,291,295,3,300,301,302,308,314,32,322,325,329,331,335,336,359,360,370,40,400,402,
406,411,416,417,419,421,425,427,429,431,434,435,442,46,465,483,49,490,495,496,499,5,50,508,51,
510,515,522,523,529,534,541,553,564,566,568,570,576,587,588,593,598,599,600,603,607,609,615,61
8,620,650,68,680,690,700,703,706,707,712,714,720,730,740,748,750,765,768,77,770,776,78,782,78
5,795,799,8,800,803,811,814,82,820,821,822,824,84,850,87,88,89,9,900,901,902,903,904,907,91,91
3,917,918,919,920,922,924,925,927,929,931,935,937,939,94,940,97,970,98,983,99,994,995,996,997,
998 58681 4554.8780 0.08481451 *
   3) ANC1P=146,169,251,330,530,586,999 12385 3095.8710 0.50553090
     6) NWAB=1,2 2788  440.2869 0.19655670 *
     7) NWAB=3 9597 2312.1070 0.59529020
      14) ESR=1,4 6746 1686.4850 0.49851760
        28) RAC3P=1,16,17,20,22,33,38,61,62,76,78,83,94 5299 1295.5290 0.42574070 *
        29) RAC3P=10,100,11,12,13,14,15,18,19,2,21,23,24,29,3,30,31,36,37,4,42,44,47,48,5,52,5
3,57,58,6,60,7,74,77,8,81,86,87,9,90,93,95 1447  260.1106 0.76503110 *
      15) ESR=2,3,6 2851  412.9604 0.82427220 *
```

Appendix L

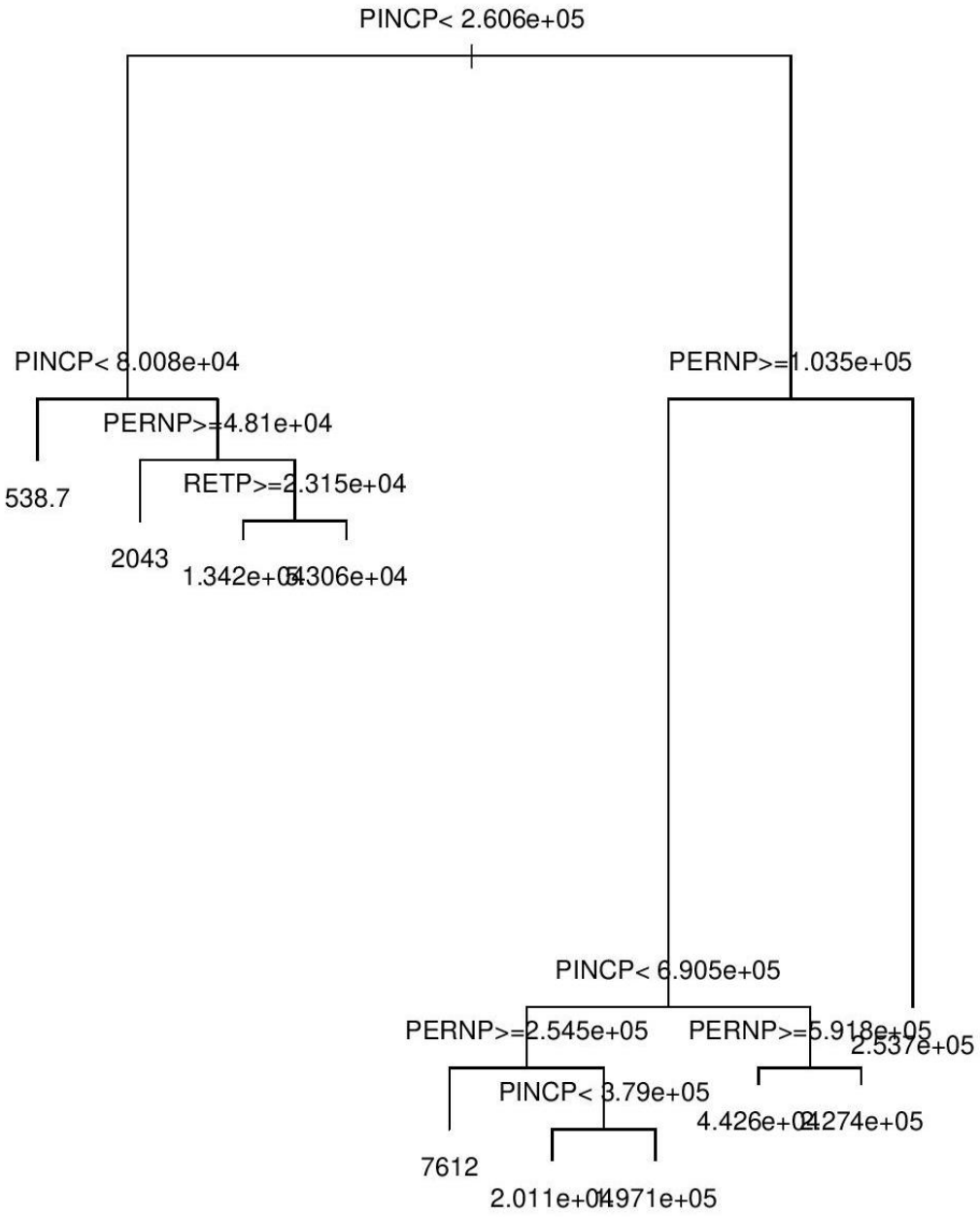
RPART Regression Tree Code

```
#R code for computing ^y with RPART
vartype <- rep("factor",125)
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("reg subset/regsubset.txt",header=TRUE,colClasses=vartype)

z$INTP[is.na(z$INTP)] = mean(z$INTP,na.rm = T)
z$FINTP = NULL
rp2 <- rpart(INTP ~ ., weight=PWGTP, data=z,
             method="anova")
plot(rp2,compress=TRUE,margin=0.1)
text(rp2)
y <- z$INTP
w <- z$PWGTP
miss <- is.na(y) ## obs with missing INTRDVX
yhat <- predict(rp2,newdata=z)
popmean <- (sum(w[!miss]*y[!miss])+sum(w[miss]*yhat[miss]))/sum(w)
# 2378.515 (new INTP estimate)
# y.hat = 2066.158 (old value before Professor Loh's update)
print(popmean)
```

RPART Regression Tree (After Professor Loh's Announcement)



RPART Regression Tree (Before Removing Imputations (Old))



RPART Regression Text Format

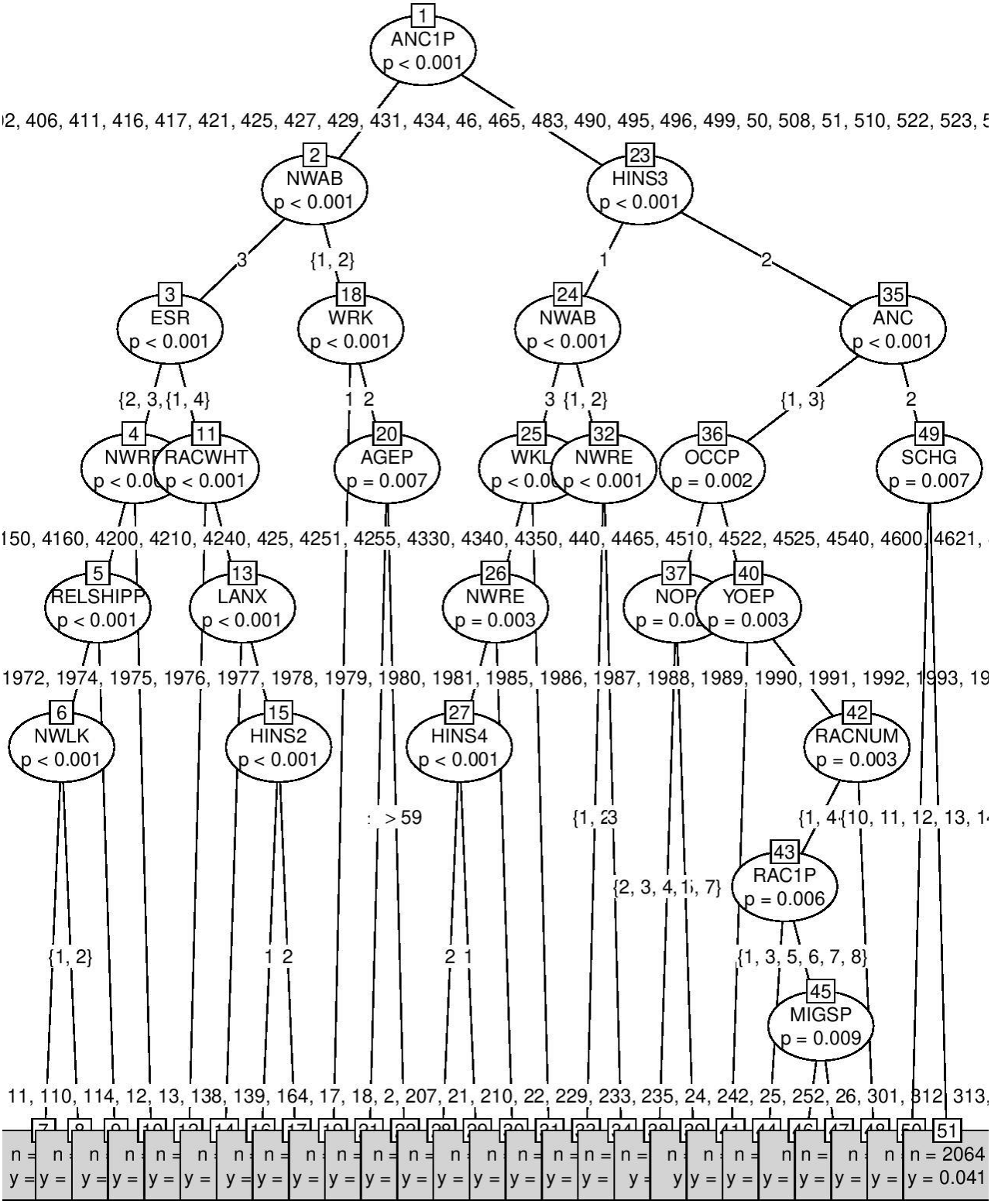
```
> rp2
n=59828 (11238 observations deleted due to missingness)

node), split, n, deviance, yval
      * denotes terminal node

 1) root 59828 1.941572e+15   2284.1810
   2) PINCP< 260550 58595 2.404462e+14   1085.6990
     4) PINCP< 80085 47520 5.677622e+13    496.8304 *
     5) PINCP>=80085 11075 1.736455e+14   4005.0760
      10) PERNP>=48100 9563 4.707209e+13   1890.1640 *
      11) PERNP< 48100 1512 9.271632e+13  20360.9700
        22) RETP>=23150 1265 4.122859e+13  13190.8200 *
        23) RETP< 23150 247 2.505741e+13  53252.5700 *
   3) PINCP>=260550 1233 1.190471e+15  74156.0300
     6) PERNP>=105000 951 2.256051e+14  20427.8500
      12) PINCP< 690500 903 1.308887e+14  15021.5300
        24) OCCP=10,1006,1007,1010,1021,1050,1065,110,1105,1106,1108,120,1305,136,1360,140,14
10,150,1530,1545,1551,1555,1650,1745,1800,1860,1910,1980,20,2002,205,2100,220,2205,230,2360,2
634,2640,2805,2810,2825,2840,2920,300,3010,3090,310,3100,3160,3250,3255,3256,3261,350,3550,36
03,3645,3710,3725,3910,3930,40,4000,410,4200,440,4510,4700,4710,4720,4750,4760,4800,4810,482
0,4830,4840,4850,4920,4965,5000,51,5165,52,5240,530,5510,5600,565,5860,5920,60,6005,6050,620
0,6260,630,640,6660,705,710,7200,735,750,7640,800,8030,845,850,8610,8630,8740,8800,8990,900,9
030,910,9130,940,9410,960 889 9.710069e+13  12841.8500
          48) PERNP>=254500 736 2.526734e+13   7495.8700 *
          49) PERNP< 254500 153 6.166852e+13  39947.3700
            98) PINCP< 375500 129 7.672236e+12  18354.2200 *
            99) PINCP>=375500 24 1.995099e+13 176384.9000 *
        25) OCCP=1220,1700,1760,2850,3040,4930,650,820 14 1.494439e+13 136082.8000 *
      13) PINCP>=690500 48 4.898331e+13 133479.3000
        26) PERNP>=591750 21 1.086916e+13  47001.5400 *
        27) PERNP< 591750 27 1.101871e+13 225228.2000 *
     7) PERNP< 105000 282 2.764549e+13 253552.0000 *
>
```

Appendix M

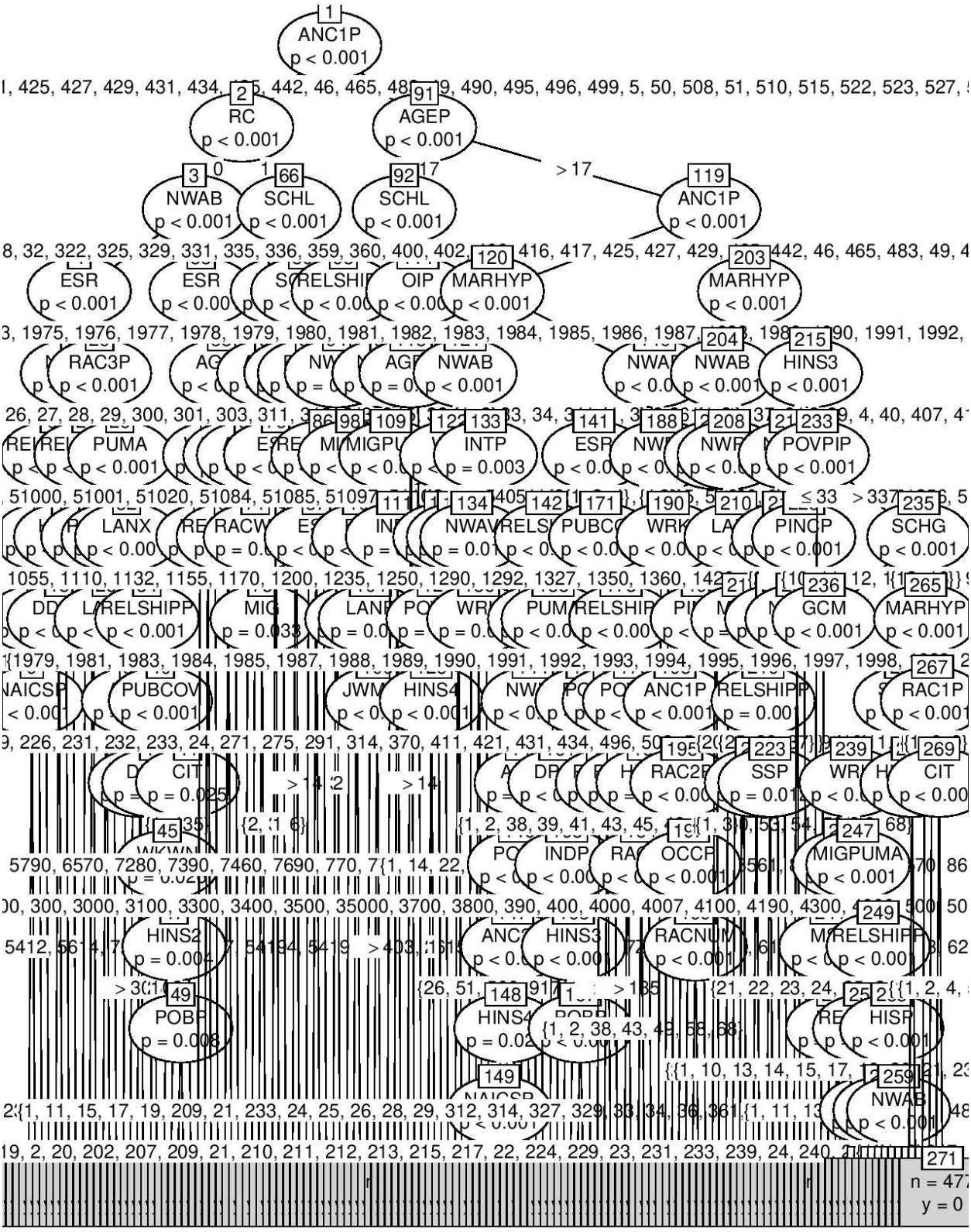
CTREE Classification Code

```
#3 R code for IPW Estimate via CTREE
library(party)

vartype <- rep("factor",125)
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("class subset/classsubset.txt",header=TRUE,colClasses=vartype)


tmp <- rep(NA,nrow(z))
tmp[z$FINTP == 0] <- 0
tmp[z$FINTP == 1] <- 1
z$FINTP <- tmp
### classification tree without INTP and PWGTP
fmla <- formula(FINTP ~ . - INTP - PWGTP)
z.small = z[sample(71066,11000),] # taking a random sample of the dataset

ct <- ctree(fmla, data=z.small)
plot(ct,type="simple",drop_terminal = TRUE)
y <- z$INTP
p <- predict(ct)
w <- z$PWGTP
gp <- !is.na(y)
ipw <-(sum(w[gp]*y[gp])/sum(p[gp]))/(sum(w[gp])/sum(p[gp]))
print(ipw) # Na (same with answer with older data prior to Professor Loh's
annoucemet)
# CTREE has many values where p = 0, so it isn't that good for predicting ipw
```

CTREE Classification Tree (After Professor Loh's Announcement)



CTREE Classification Tree (Before Removing Imputations (Old))



CTREE Classification Tree Text Format

```
> ct

       Conditional inference tree with 29 terminal nodes

Response:  FINTP
Inputs:  PUMA, PWGTP, AGEP, CIT, CITWP, COW, DDRS, DEAR, DEYE, DOUT, DPHY, DRAT,
DRATX, DREM, ENG, FER, GCL, GCM, GCR, HIMRKS, HINS1, HINS2, HINS3, HINS4, HINS5,
HINS6, HINS7, INTP, JWMNP, JWRIP, JWTRNS, LANX, MAR, MARHD, MARHM, MARHT, MARHW,
MARHYP, MIG, MIL, MLPA, MLPB, MLPCD, MLPE, MLPFG, MLPH, MLPI, MLPJ, MLPK, NWAB,
NWAV, NWLA, NWLK, NWRE, OIP, PAP, RELSHIPP, RETP, SCH, SCHG, SCHL, SEMP, SEX,
SSIP, SSP, WAGP, WKHP, WKL, WKWN, WRK, YOEP, ANC, ANC1P, ANC2P, DECADE, DIS,
DRIVESP, ESP, ESR, FOD1P, FOD2P, HICOV, HISP, INDP, JWAP, JWDP, LANP, MIGPUMA,
MIGSP, MSP, NAICSP, NATIVITY, NOP, OC, OCCP, PAOC, PERNP, PINCP, POBP, POVPIP,
POWPUMA, POWSP, PRIVCOV, PUBCOV, QTRBIR, RAC1P, RAC2P, RAC3P, RACAIAN, RACASN,
RACBLK, RACNH, RACNUM, RACPI, RACSOR, RACWHT, RC, SCIENGP, SCIENGRLP, SFN, SFR,
SOCP, VPS, WAOB
Number of observations:  11000

1) ANC1P == {146, 177, 219, 252, 335, 434, 496, 901, 994, 999}; criterion = 1,
statistic = 2326.89
  2) NWLK == {3}; criterion = 1, statistic = 341.694
    3) ESR == {3, 6}; criterion = 1, statistic = 348.014
      4) NWLA == {3}; criterion = 1, statistic = 97.683
        5) RELSHIPP == {20, 21, 22, 23, 25, 27, 28, 29, 30, 31, 32, 33, 34, 36,
38}; criterion = 1, statistic = 92.003
          6)*  weights = 306
        5) RELSHIPP == {37}
          7)*  weights = 42
      4) NWLA == {2}
        8)*  weights = 17
    3) ESR == {1, 2, 4}
      9) RACWHT == {0}; criterion = 1, statistic = 316.454
        10)*  weights = 244
      9) RACWHT == {1}
        11) LANX == {1}; criterion = 1, statistic = 278.275
          12)*  weights = 67
        11) LANX == {2}
          13)*  weights = 809
  2) NWLK == {1, 2}
    14) AGEP <= 42; criterion = 0.99, statistic = 129.795
      15)*  weights = 134
    14) AGEP > 42
      16) NWRE == {2}; criterion = 0.977, statistic = 88.525
        17)*  weights = 75
      16) NWRE == {3}
        18)*  weights = 234
1) ANC1P == {1, 100, 102, 103, 109, 11, 111, 114, 115, 12, 122, 125, 128, 129,
130, 142, 144, 148, 152, 153, 154, 168, 171, 178, 181, 183, 187, 190, 194, 195,
20, 200, 21, 210, 211, 212, 218, 22, 221, 222, 223, 224, 225, 226, 231, 232,
233, 234, 235, 237, 238, 239, 24, 26, 261, 271, 275, 290, 291, 3, 308, 314, 32,
336, 359, 360, 370, 402, 406, 411, 416, 417, 419, 421, 425, 427, 429, 431, 435,
442, 46, 465, 490, 495, 499, 5, 50, 508, 51, 515, 522, 523, 529, 530, 534, 541,
553, 564, 566, 568, 570, 576, 587, 588, 598, 599, 600, 603, 609, 615, 618, 620,
650, 68, 680, 703, 706, 712, 720, 730, 740, 750, 765, 770, 776, 78, 782, 785,
795, 799, 8, 800, 803, 811, 814, 82, 821, 822, 84, 850, 87, 88, 89, 9, 900, 902,
903, 904, 91, 917, 918, 919, 920, 924, 925, 929, 931, 935, 937, 939, 94, 940,
97, 970, 98, 99, 995, 996, 997, 998}
  19) HINS3 == {1}; criterion = 1, statistic = 532.198
    20) NWLK == {3}; criterion = 1, statistic = 237.172
      21) ESR == {6}; criterion = 1, statistic = 152.371
        22) HINS4 == {2}; criterion = 1, statistic = 68.041
          23)*  weights = 110
        22) HINS4 == {1}
          24)*  weights = 68
```

```
      21) ESR == {1, 2, 3}
        25)*  weights = 324
    20) NWLK == {1, 2}
      26) ESR == {1, 2}; criterion = 1, statistic = 169.822
        27) RACWHT == {0}; criterion = 0.953, statistic = 56.926
          28)*  weights = 20
        27) RACWHT == {1}
          29)*  weights = 49
      26) ESR == {3, 6}
        30) NWRE == {2}; criterion = 1, statistic = 177.695
          31)*  weights = 212
        30) NWRE == {1, 3}
          32)*  weights = 1476
  19) HINS3 == {2}
    33) RACBLK == {1}; criterion = 1, statistic = 556.074
      34)*  weights = 1180
    33) RACBLK == {0}
      35) NWAB == {1, 3}; criterion = 1, statistic = 542.495
        36) WKL == {3}; criterion = 1, statistic = 554.692
          37)*  weights = 70
        36) WKL == {1, 2}
          38) JWAP == {114, 117, 126, 139, 141, 149, 159, 161, 164, 167, 169,
172, 173, 181, 185, 187, 189, 192, 195, 198, 199, 205, 211, 214, 215, 216, 219,
231, 236, 25, 257, 264, 37, 47, 55, 6, 61, 63, 65, 68, 92, 95}; criterion = 1,
statistic = 532.213
            39)*  weights = 359
          38) JWAP == {100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110,
111, 112, 113, 115, 116, 118, 119, 120, 121, 122, 123, 124, 125, 127, 128, 129,
130, 131, 132, 133, 134, 135, 136, 140, 142, 143, 144, 145, 146, 147, 148, 150,
151, 152, 153, 154, 155, 156, 157, 158, 160, 162, 163, 165, 168, 170, 171, 174,
175, 176, 177, 178, 179, 180, 182, 183, 184, 186, 188, 190, 191, 193, 194, 196,
197, 2, 200, 201, 202, 203, 204, 206, 207, 208, 210, 212, 213, 217, 218, 22,
220, 221, 222, 223, 224, 225, 226, 227, 229, 230, 235, 237, 238, 239, 241, 242,
244, 247, 248, 250, 251, 252, 256, 259, 260, 261, 262, 266, 268, 269, 27, 270,
271, 272, 273, 274, 275, 278, 28, 29, 3, 30, 31, 32, 34, 35, 36, 38, 42, 43, 44,
45, 46, 48, 49, 50, 52, 53, 54, 56, 57, 58, 59, 60, 62, 64, 66, 67, 69, 70, 71,
72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91,
93, 94, 96, 97, 98, 99}
            40) ESR == {2, 3, 6}; criterion = 0.97, statistic = 512.965
              41)*  weights = 146
            40) ESR == {1, 4}
              42)*  weights = 3699
      35) NWAB == {2}
        43) INDP == {2190, 3291, 3990, 4270, 4590, 5070, 6170, 7490, 7570, 7680,
8670, 8770, 9190, 9370, 9490, 9670}; criterion = 0.997, statistic = 317.038
          44)*  weights = 47
        43) INDP == {1190, 1280, 1370, 1390, 1590, 1691, 170, 1870, 1990, 2170,
2270, 2290, 2370, 2470, 280, 2870, 2970, 2980, 3080, 3370, 3490, 3580, 3680,
3780, 3895, 3980, 4070, 4090, 4170, 4195, 4370, 4380, 4470, 4580, 4670, 4680,
4690, 4770, 4795, 4870, 4890, 4971, 4972, 4980, 5080, 5090, 5170, 5180, 5190,
5275, 5370, 5381, 5391, 5480, 5490, 5570, 5580, 5593, 5690, 570, 5790, 6190,
6290, 6370, 6380, 6390, 6470, 6480, 6672, 6680, 6695, 6770, 6870, 6890, 690,
6970, 6991, 6992, 7071, 7072, 7080, 7181, 7190, 7270, 7280, 7290, 7380, 7390,
7460, 7470, 7480, 7580, 7590, 7670, 7690, 770, 7770, 7780, 7860, 7870, 7890,
7970, 7980, 7990, 8080, 8090, 8170, 8180, 8191, 8192, 8270, 8290, 8370, 8470,
8561, 8562, 8563, 8564, 8570, 8590, 8660, 8680, 8980, 8990, 9070, 9090, 9160,
9170, 9290, 9390, 9470, 9480, 9570, 9590, 9690, 9770, 9920}
          45) JWTRNS == {10, 12, 3, 4, 5}; criterion = 0.994, statistic =
307.427
            46)*  weights = 7
          45) JWTRNS == {1, 11, 2}
            47) FOD1P == {1103, 2306, 5007, 6001, 6002}; criterion = 0.997,
statistic = 326.117
              48)*  weights = 10
```

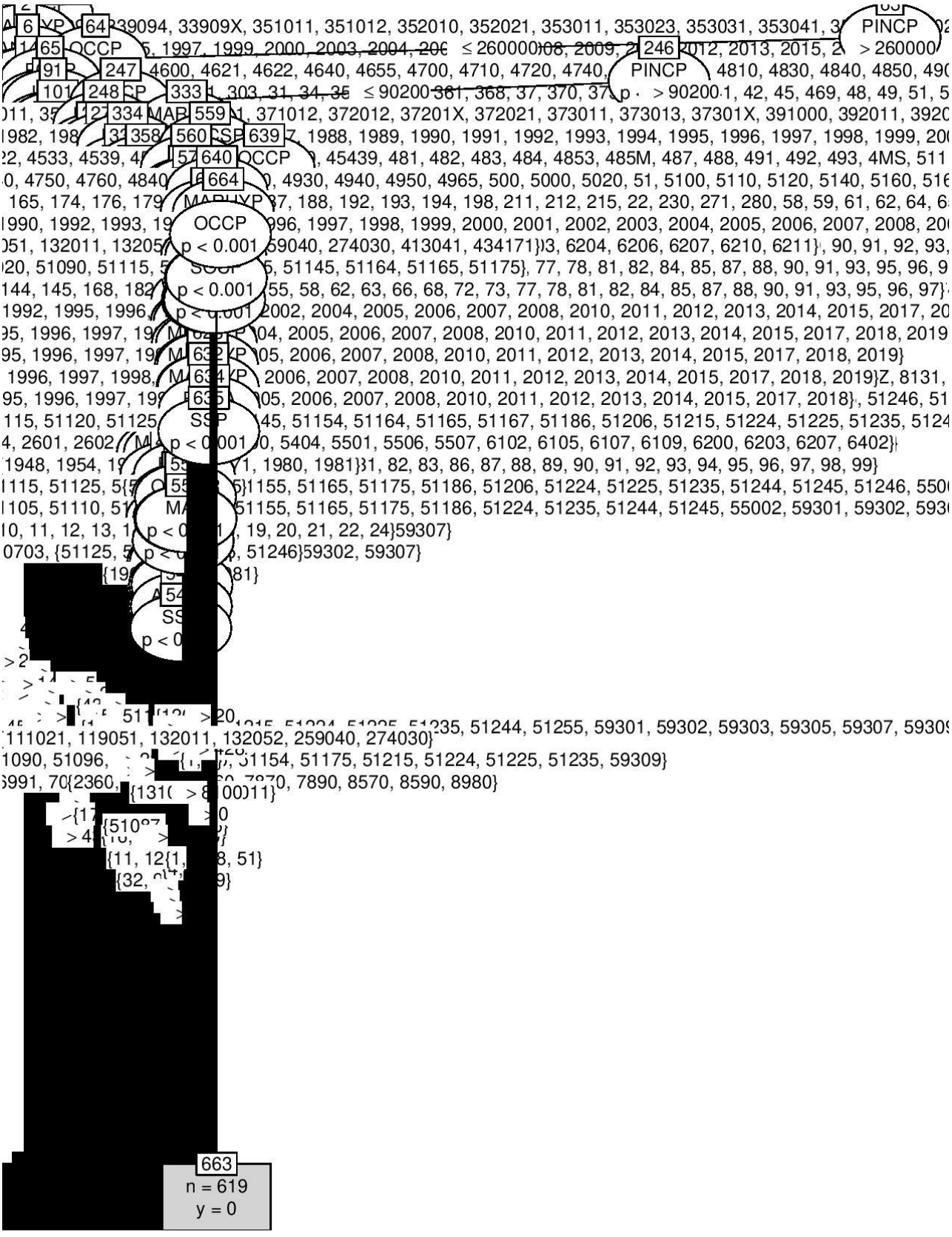
```
            47) FOD1P == {1101, 1301, 1401, 1501, 1901, 1902, 1903, 2100, 2102,
2105, 2300, 2304, 2307, 2309, 2310, 2311, 2313, 2314, 2400, 2401, 2404, 2405,
2406, 2407, 2408, 2409, 2412, 2414, 2417, 2418, 2499, 2502, 2503, 2601, 2602,
2603, 2901, 3301, 3401, 3600, 3601, 3603, 3608, 3609, 3611, 3699, 3700, 3702,
4001, 4002, 4101, 4801, 5001, 5098, 5200, 5203, 5301, 5401, 5403, 5404, 5500,
5501, 5502, 5504, 5505, 5506, 5507, 5601, 5901, 6000, 6003, 6004, 6005, 6007,
6100, 6102, 6103, 6105, 6106, 6107, 6109, 6110, 6200, 6201, 6203, 6205, 6206,
6207, 6209, 6210, 6211, 6212, 6402}
              49) RAC2P == {49, 58}; criterion = 0.999, statistic = 325.032
                50)*  weights = 18
              49) RAC2P == {1, 17, 28, 37, 38, 39, 43, 45, 48, 53, 54, 56, 57,
59, 67, 68}
                51) SOCP == {1520XX, 172011, 371012, 439XXX, 472040, 472080};
criterion = 0.988, statistic = 321.849
                  52)*  weights = 9
                51) SOCP == {111021, 1110XX, 112011, 112022, 112030, 113012,
113021, 113031, 113061, 113071, 113121, 119013, 119021, 119030, 119051, 119111,
119141, 119161, 1191XX, 131022, 131023, 131030, 131041, 131051, 131070, 131081,
131111, 131161, 132011, 132020, 132052, 132070, 132082, 151211, 151230, 151241,
151244, 15124X, 151251, 151252, 151253, 151254, 151299, 152031, 171011, 172070,
172110, 1721YY, 173023, 17302X, 192010, 193011, 1930XX, 194010, 1940YY, 195010,
211012, 211019, 211029, 211092, 211093, 21109X, 212099, 2310XX, 232011, 232093,
251000, 252010, 252020, 252030, 252050, 253041, 2530XX, 254010, 254031, 259040,
2590XX, 271010, 271023, 271024, 27102X, 272022, 272023, 272042, 272099, 273041,
273043, 273091, 291122, 291127, 29112X, 291141, 291181, 291210, 291292, 292010,
292053, 292056, 292061, 299000, 311122, 311131, 312020, 319091, 319092, 319094,
319096, 31909X, 332011, 333021, 333050, 339030, 339091, 339093, 33909X, 351011,
351012, 352010, 352021, 353011, 353023, 353031, 353041, 359011, 359021, 359031,
372012, 37201X, 373011, 392021, 395012, 395092, 397010, 399011, 399031, 399032,
399099, 411011, 412010, 412021, 412022, 412031, 413021, 413031, 413041, 413091,
414010, 419020, 419091, 419099, 431011, 433021, 433031, 433051, 433071, 434051,
434061, 434071, 434081, 434121, 434131, 434171, 435021, 435051, 435052, 435061,
435071, 436011, 436012, 436014, 439021, 439022, 439061, 4520XX, 453031, 471011,
472031, 472061, 472111, 472121, 472152, 473010, 474011, 474041, 475032, 491011,
492011, 492020, 492097, 492098, 493011, 493023, 493040, 493050, 499021, 499044,
499071, 499098, 5120XX, 513092, 514031, 514120, 514XXX, 515112, 516011, 51609X,
517042, 518010, 519191, 519195, 5191XX, 531000, 533030, 533053, 533054, 533099,
536030, 536061, 537021, 537051, 537061, 537062, 537064, 537065, 551010, 553010,
559830, 999920}
                  53) YOEP == {1962, 1966, 1968, 1974, 1976, 1981, 1983, 1985,
1987, 1992, 2010}; criterion = 0.999, statistic = 118.822
                    54)*  weights = 12
                  53) YOEP == {1957, 1972, 1975, 1978, 1979, 1980, 1984, 1986,
1988, 1989, 1990, 1991, 1994, 1995, 1996, 1997, 1998, 1999, 2000, 2001, 2002,
2003, 2004, 2005, 2006, 2007, 2008, 2009, 2011, 2012, 2013, 2014, 2015, 2016,
2017, 2018, 2019}
                    55) SCHL == {10, 14, 17, 18, 20}; criterion = 0.975,
statistic = 118.822
                      56)*  weights = 306
                    55) SCHL == {1, 11, 12, 13, 15, 16, 19, 21, 22, 23, 24, 3,
5, 6, 7, 9}
                      57)*  weights = 950
```

Appendix N

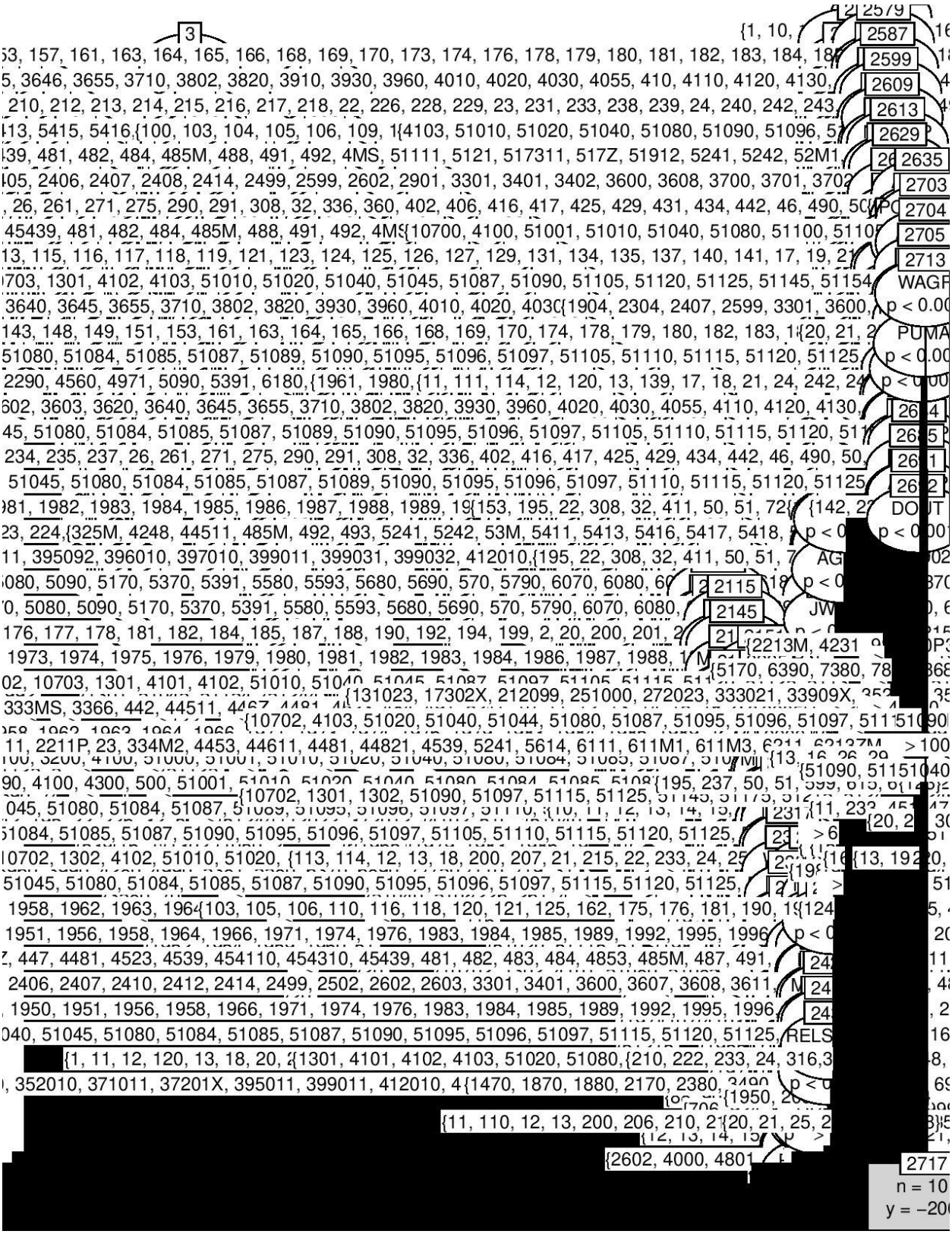
CTREE Regression Code

```
# CTREE Regression Tree
library(party)
vartype <- rep("factor",125)
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("reg subset/regsubset.txt",header=TRUE,colClasses=vartype)
z.samp = z[sample(71066,10000),]

# old code
#z.samp$INTP[is.na(z.samp$INTP)] = mean(z.samp$INTP,na.rm = T)

# regression without FINTP and PWGTP
z.samp = z.samp[-which(is.na(z.samp$INTP)), ] # omit missing values in
response
fmla <- formula(INTP ~ . - FINTP - PWGTP)
ct <- ctree(fmla, data=z.samp, weights = z.samp$PWGTP)
plot(ct,type="simple",drop_terminal = TRUE)
y <- z.samp$INTP
w <- z.samp$PWGTP
yhat <- predict(ct,newdata=z.samp)
miss <- is.na(y) ## obs with missing INTP
popmean <- (sum(w[!miss]*y[!miss])+sum(w[miss]*yhat[miss]))/sum(w)
print(popmean) #2691.584 (old value before Professor Loh's update)
# 2107.745 (new INTP estimate)
```

CTREE Regression Tree (After Professor Loh's Announcement)



CTREE Regression Tree (Before Removing Imputations (Old))



CTREE Regression Tree Text Format Preview

```
                                                                  2474)*  weights = 43
                                                               2473) PUMA == {10702, 51044, 51085, 51115, 51154, 51155, 51165, 5
1215, 51244}
                                                                  2475) SEMP <= 0; criterion = 1, statistic = 606
                                                                    2476)*  weights = 574
                                                                  2475) SEMP > 0
                                                                    2477)*  weights = 33
                                                            2472) INDP == {1990, 3580, 4770, 4870, 4880, 4971, 4972, 5070, 509
0, 5170, 5275, 5381, 5391, 5480, 5690, 6380, 6880, 6890, 7071, 7270, 7390, 7460, 7490, 7570, 7580, 7680, 7690, 770, 7780, 7860, 787
0, 7880, 7890, 7980, 8090, 8180, 8191, 8192, 8270, 8290, 8370, 8570, 8590, 8660, 8680, 8690, 9090, 9170, 9290, 9380, 9390, 9470, 95
70, 9670, 9770}
                                                               2478) CIT == {2, 3, 5}; criterion = 1, statistic = 12405.132
                                                                  2479) YOEP == {2007}; criterion = 1, statistic = 1376
                                                                    2480)*  weights = 56
                                                                  2479) YOEP == {1979, 1990, 1994, 2000, 2001, 2004, 2006, 2010,
 2011, 2014, 2016, 2018}
                                                                    2481)*  weights = 1321
                                                               2478) CIT == {1, 4}
                                                                  2482)*  weights = 19485
                                                         2427) OCCP == {1005, 1007, 102, 1050, 110, 1105, 1305, 136, 1460, 1545, 1555, 1
60, 1610, 1640, 1760, 1825, 1860, 1900, 1920, 1970, 20, 2002, 2004, 2006, 2012, 2014, 2015, 2016, 2040, 2060, 2170, 2180, 220, 231
0, 2330, 2350, 2440, 2545, 2633, 2635, 2640, 2722, 2723, 2830, 2865, 2910, 2920, 3010, 3030, 3040, 310, 3110, 3150, 3160, 3220, 323
0, 3245, 3258, 3300, 3310, 3321, 3322, 3324, 3330, 3401, 3402, 3421, 3423, 3424, 3430, 350, 3500, 3545, 3550, 3601, 3603, 3605, 361
0, 3620, 3630, 3640, 3645, 3646, 3647, 3648, 3649, 3655, 3700, 3710, 3720, 3725, 3740, 3750, 3802, 3820, 3870, 3900, 3910, 3960, 40
00, 4010, 4020, 4030, 4040, 4055, 4120, 4130, 4140, 4150, 420, 4200, 4210, 4220, 4240, 4251, 4255, 4330, 4350, 4420, 4435, 4465, 45
00, 4510, 4521, 4525, 4530, 4600, 4622, 4640, 4655, 4700, 4750, 4760, 4800, 4820, 4830, 4850, 4950, 4965, 5000, 5010, 5100, 5110, 5
150, 5165, 52, 520, 5220, 5240, 5260, 530, 5300, 5310, 5320, 5330, 5350, 5360, 540, 5522, 5540, 5550, 5560, 5600, 5610, 5630, 5720,
 5810, 5840, 5850, 5860, 5920, 5940, 60, 600, 6005, 6040, 6050, 6200, 6220, 6230, 6240, 6260, 630, 6305, 6330, 6360, 6410, 6442, 65
0, 6515, 6520, 6530, 6600, 6720, 6825, 6835, 700, 7020, 7030, 7100, 7120, 7140, 7150, 7200, 7210, 7220, 7240, 725, 726, 7260, 7315,
 7320, 7340, 7360, 7410, 7420, 750, 7610, 7640, 7750, 7810, 7830, 7855, 7905, 7950, 8030, 8140, 820, 8225, 8255, 830, 8300, 8320, 8
350, 8465, 8530, 8600, 8610, 8620, 8650, 8760, 8800, 8810, 8830, 8940, 8950, 8990, 9005, 9030, 9040, 9050, 910, 9121, 9122, 9130, 9
141, 9142, 9150, 9210, 9240, 9300, 9365, 9510, 960, 9600, 9610, 9620, 9640, 9645, 9760, 9800, 9825, 9920}
                                                      2483) MIGSP == {1, 114, 207, 233, 243, 26, 303, 5, 8}; criterion = 1, statist
ic = 13583.595
                                                        2484) SCHL == {14}; criterion = 1, statistic = 487
                                                          2485)*  weights = 55
                                                        2484) SCHL == {1, 13, 16, 21, 22, 23}
                                                          2486)*  weights = 433
                                                      2483) MIGSP == {10, 11, 110, 12, 13, 138, 15, 17, 18, 210, 22, 231, 24, 251,
 29, 30, 312, 313, 317, 34, 36, 37, 4, 42, 47, 48, 51, 54, 55, 6, 9}
```

Appendix O

MICE Imputation Code (Used with Logistic Regression)

```
# 4. Imputing missing values in the X variables using MICE
z = read.table("cleandata.txt",header=TRUE, stringsAsFactors = TRUE)
x = read.table("class imp/classimp.scr",header=TRUE)

#top 15 variables for selection

top = c("ANC","ANC1P","NWAB","NWLK","RELSHIPP","NWLA","RC","OC","ANC2P",
        "NWRE","SCHL","MSP","POVPIP","ESR", "PWGTP")
# Note I initially included the weight variable then took it out later

red = z[,top] # reduced data frame for feature selection
y = z$FINTP

library(mice)

out = mice(red)

c1 <- complete(out,1) ## 1st imputed data set
```

Amelia Imputation Code (Used with RandomForest)

```
vartype <- rep("factor",125)
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("reg subset/regsubset.txt",header=TRUE,colClasses=vartype)


x = read.table("reg imp/regtimp.scr",header=TRUE)
# top variables for feature selection

top = c("PINCP","AGEP","RETP","SCIENGP","SCIENGRLP",
        "SCHL","HINS3","POVPIP","SSP","FOD1P",
        "MARHYP","WAGP","GCL","RELSHIPP","PERNP")


# old code before Professor Loh's announcement is all the code that is
commented out

#top = c('PINCP', "AGEP", "RETP","SCHL","SCIENGRLP",
#         "SCIENGP","HINS3","POVPIP","SSP","FOD1P", "MARHYP","WAGP",
#         "PERNP", "GCL", "RELSHIPP")

y = z$INTP
red = z[,top] # reduced data frame for feature selection

sum(is.na(y)) # tons of missing data

library(Amelia) # running imputations on the reduced dataset

#out <- amelia(red,noms=c(5,6,7,14), ords = c(4,10,15,11))
out = amelia(red,noms=c(4,5,7,13), ords = c(6,10,11,14))

c1 = out$imputations$imp1
```

Appendix P

Logistic Regression Code

```
# 5. Use logistic regression to find ^pi and IPW to estimate μ
c1$FINTP = z$FINTP
c1$PWGTP = NULL # no weight variables for logistic regression
full = glm(FINTP~ . - FINTP, data = c1, family = "binomial")
summary(full)


# generating predictions from glm model
c1$pred = predict(full, type = "response")

c1$FINTPHAT=(c1$pred>=0.5)+0

z <- read.table("cleandata.txt",header=TRUE, stringsAsFactors = TRUE)
w <- z$PWGTP ### sampling weights
gp <- !is.na(z$INTP)
p <- c1$pred
ipw <- sum(w[gp]*z$INTP[gp]/p[gp])/sum(w[gp]/p[gp])
print(ipw) # 2670.091
# old code used before Professor Loh's announcement

#c1$prediction = c1$probability
#for(i in 1:nrow(c1)){
#  if(c1$probability[i] < .5 ){
#    c1$prediction[i] = 0
#  } else{
#    c1$prediction[i] = 1
#  }
#
#}
```

Logistic Regression Summary

```
Call:
glm(formula = FINTP ~ . - FINTP, family = "binomial", data = c1)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9941  -0.4713  -0.3786  -0.2774   3.2758

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -6.864e+00  1.824e-01 -37.629  < 2e-16 ***
ANC          6.400e-01  1.056e-02  60.605  < 2e-16 ***
ANC1P        4.232e-04  4.288e-05   9.869  < 2e-16 ***
NWAB         4.251e-01  6.716e-02   6.331 2.44e-10 ***
NWLK         9.015e-01  5.550e-02  16.243  < 2e-16 ***
RELSHIPP     3.995e-03  2.579e-03   1.549  0.12138
NWLA         3.162e-01  7.442e-02   4.249 2.14e-05 ***
RC          -2.795e-01  1.189e-01  -2.350  0.01876 *
OC          -3.411e-01  1.373e-01  -2.485  0.01294 *
ANC2P        9.600e-04  4.936e-05  19.447  < 2e-16 ***
NWRE        -7.288e-01  4.875e-02 -14.951  < 2e-16 ***
SCHL        -1.440e-02  3.526e-03  -4.084 4.43e-05 ***
MSP          1.767e-02  6.461e-03   2.735  0.00625 **
POVPIP      -1.509e-04  8.286e-05  -1.822  0.06853 .
ESR          2.195e-01  7.024e-03  31.256  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 62050  on 71065  degrees of freedom
Residual deviance: 48936  on 71051  degrees of freedom
AIC: 48966

Number of Fisher Scoring iterations: 5
```

Appendix Q

Random Forest Code

```
#6. randomForest and CForest (party) to obtain ^y and estimate μ with formula
(1)

# via randomForest

#install.packages("randomForest")
library(randomForest)


vartype <- rep("factor",125)
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("reg subset/regsubset.txt",header=TRUE,colClasses=vartype)


x = read.table("reg imp/regtimp.scr",header=TRUE)
# top variables for feature selection

top = c("PINCP","AGEP","RETP","SCIENGP","SCIENGRLP",
        "SCHL","HINS3","POVPIP","SSP","FOD1P",
        "MARHYP","WAGP","GCL","RELSHIPP","PERNP")


# old code before Professor Loh's announcement is all the code that is
commented out

#top = c('PINCP', "AGEP", "RETP","SCHL","SCIENGRLP",
#         "SCIENGP","HINS3","POVPIP","SSP","FOD1P", "MARHYP","WAGP",
#         "PERNP", "GCL", "RELSHIPP")

y = z$INTP
red = z[,top] # reduced data frame for feature selection

sum(is.na(y)) # tons of missing data

library(Amelia) # running imputations on the reduced dataset

#out <- amelia(red,noms=c(5,6,7,14), ords = c(4,10,15,11))
out = amelia(red,noms=c(4,5,7,13), ords = c(6,10,11,14))

c1 = out$imputations$imp1
c1$INTP = y

#c1$INTP[is.na(c1$INTP)] = 0 # changing all NAs to zero cannot have any INTP
if you're less than 15
#old code

sum(is.na(c1$INTP)) # tons of rows where INTP is missing
c1 = c1[!is.na(c1$INTP),] # therefore will omit them
summary(c1)
str(c1)
#dropping markhyp and fod1p because they have too many levels (80 and 173
respectively)
c1$FOD1P = NULL
c1$MARHYP = NULL
```

```
rf <- randomForest(x=c1[,-14],y = c1$INTP)
### ipw estimate


w = z$PWGTP[gp]
p = predict(rf,newdata = c1) # had to predict on the imputed dataset with no
missing values, due to randomForest
mean = sum(w[!is.na(c1$INTP)]*p)/sum(w[!is.na(c1$INTP)])
print(mean) #2305.564

#y = c1$INTP
#gp = !is.na(z$INTP)
#mu = sum(w*y)/sum(w)
#print(mu) 2284.181 (old value prior to professor Loh's announcement)
```

Random Forest Summary

```
> rf

Call:
 randomForest(x = c1[, -14], y = c1$INTP)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 4

          Mean of squared residuals: 38823974
                    % Var explained: 90.66
> |
```

Appendix R

CForest Code

```
# CForest
library(party)
z = z[!is.na(z$INTP),]
z.small = z[sample(nrow(z),10000),] #subset
fmla <- formula(INTP ~ . - FINTP - PWGTP)
cf <- cforest(fmla, data=z.small)
p <- predict(cf,newdata=z)
w <- z$PWGTP
y <- z$INTP
gp <- !is.na(y)
miss <- is.na(y) ## obs with missing INTP
popmean <- (sum(w[!miss]*y[!miss])+sum(w[miss]*p[miss]))/sum(w)
print(popmean) # 2284.181
```

CForest Summary

```
> cf

         Random Forest using Conditional Inference Trees

Number of trees:  500

Response:  INTP
Inputs:  PUMA, PWGTP, AGEP, CIT, CITWP, COW, DDRS, DEAR, DEYE, DOUT, DPHY, DRAT,
 DRATX, DREM, ENG, FER, GCL, GCM, GCR, HIMRKS, HINS1, HINS2, HINS3, HINS4, HINS
5, HINS6, HINS7, JWMNP, JWRIP, JWTRNS, LANX, MAR, MARHD, MARHM, MARHT, MARHW, MA
RHYP, MIG, MIL, MLPA, MLPB, MLPCD, MLPE, MLPFG, MLPH, MLPI, MLPJ, MLPK, NWAB, NW
AV, NWLA, NWLK, NWRE, OIP, PAP, RELSHIPP, RETP, SCH, SCHG, SCHL, SEMP, SEX, SSI
P, SSP, WAGP, WKHP, WKL, WKWN, WRK, YOEP, ANC, ANC1P, ANC2P, DECADE, DIS, DRIVES
P, ESP, ESR, FOD1P, FOD2P, HICOV, HISP, INDP, JWAP, JWDP, LANP, MIGPUMA, MIGSP,
 MSP, NAICSP, NATIVITY, NOP, OC, OCCP, PAOC, PERNP, PINCP, POBP, POVPIP, POWPUM
A, POWSP, PRIVCOV, PUBCOV, QTRBIR, RAC1P, RAC2P, RAC3P, RACAIAN, RACASN, RACBLK,
 RACNH, RACNUM, RACPI, RACSOR, RACWHT, RC, SCIENGP, SCIENGRLP, SFN, SFR, SOCP, V
PS, WAOB, FINTP
Number of observations:  10000
```

Appendix S

Note that for convenience, all of these files can be downloaded with this link:
https://drive.google.com/drive/folders/12Ue0TjzrznGztFrGRpQCy7yIIov6_U9O?usp=sharing

R Code for Reproducing Output (Final.R)

```
# Anuj Amin

# Final Project STAT 443


rm(list = ls())


#Import dataset and remove INTP imputations done by census bureau

z = read.csv("psam_p51.csv",na.strings="")

z = z[!is.na(z$INTP),]

z$INTP[z$FINTP == 1] = NA

write.table(z,"cleandata.txt",row.names=FALSE,col.names=TRUE)


#Creating the description file (regression)

dat <- read.table("cleandata.txt",header=TRUE)

nvar <- ncol(dat)

varnames <- names(dat)

roles <- rep("c",nvar)


n.vars <- c("PWGTP","AGEP","JWMNP","OIP", "PAP", "RETP", "SEMP", "SSIP", "SSP", "WAGP",
"WKHP", "WKWN", "PERNP",

        "PINCP", "POVPIP") # numeric variables

roles[varnames %in% n.vars] <- "n"


d.var <- "INTP"

roles[varnames %in% d.var] <- "d" # dependent variable
```

```
x.var <- "FINTP"
roles[varnames %in% x.var] <- "x" # excluded variable

write("cleandata.txt",file="descreg.txt") # writing clean data file
write("NA",file="descreg.txt",append=TRUE)
write("2",file="descreg.txt",append=TRUE)
write.table(cbind(1:nvar,varnames,roles),file="descreg.txt",
        row.names=FALSE,col.names=FALSE,quote=FALSE,append=TRUE)



#Creating the description file (classification)
dat <- read.table("cleandata.txt",header=TRUE)
nvar <- ncol(dat)
varnames <- names(dat)
roles <- rep("c",nvar)

n.vars <- c("PWGTP","AGEP","JWMNP","OIP", "PAP", "RETP", "SEMP", "SSIP", "SSP", "WAGP",
"WKHP", "WKWN", "PERNP",
        "PINCP", "POVPIP") # numeric variable
roles[varnames %in% n.vars] <- "n"

x.var <- "INTP"
roles[varnames %in% x.var] <- "x" # excluded variable

d.var <- "FINTP"
roles[varnames %in% d.var] <- "d" # dependent variable

write("cleandata.txt",file="descclass.txt") # writing to cleandata file
write("NA",file="descclass.txt",append=TRUE)
```

```
write("2",file="descclass.txt",append=TRUE)
write.table(cbind(1:nvar,varnames,roles),file="descclass.txt",
        row.names=FALSE,col.names=FALSE,quote=FALSE,append=TRUE)



# summary of missing values and INTP infomation
z <- read.table("cleandata.txt",header=TRUE)
length(z$INTP) # total number of observations after unimputing dataset: 71066
table(z$FINTP) # number of missing INTP observations: 11238 (FINTP = 1)
apply(z, 2, FUN = function(x) sum(is.na(x)))

# Distribution of INTP
hist(z$INTP, breaks = 20, main = "Distribution of INTP", xlab = "INTP", col = "red")
?hist



# Using IPW to estimate mean (mu) via GUIDE classification tree (1)
z <- read.table("cleandata.txt",header=TRUE)
w <- z$PWGTP ### sampling weights
zclass <- read.table("classtree/gtree.txt",header=TRUE)
probmissing <- zclass[,5] ### estimated P(FINTP = 0)
p <- 1-probmissing ### estimated P(FINTP is nonmissing)
group <- !is.na(z$INTP) ### group of nonmissing INTP obs
ipw <- sum(w[group]*z$INTP[group]/p[group])/sum(w[group]/p[group])
print(ipw)
# 2292.499

# Using IPW to estimate mean (mu) classification forest (1)
```

```
z <- read.table("cleandata.txt",header=TRUE)
w <- z$PWGTP ### sampling weights
zclass <- read.table("guide cf/gcfpred.txt",header=TRUE)
probmissing <- zclass[,2] ### estimated P(FINTP = 0)
p <- 1-probmissing ### estimated P(FINTP is nonmissing)
group <- !is.na(z$INTP) ### group of nonmissing INTP obs
ipw <- sum(w[group]*z$INTP[group]/p[group])/sum(w[group]/p[group])
print(ipw)
# 2019.165 (new IPW after loh's announcement)
# 2118.599 (old value before Professor Loh's announcement)



#Using imputation to estimate INTP via regression tree result (2)
zreg <- read.table("Reg tree/regtreepred.txt",header=TRUE)
yhat <- zreg$predicted
imputed <- (sum(w[group]*z$INTP[group])+sum(w[!group]*yhat[!group]))/sum(w)
print(imputed)
# 2324.75
simple <- sum(w[group]*z$INTP[group])/sum(w[group])
print(simple)
# 2284.181



# Estimating mean via regression forest (2)
zreg <- read.table("guide rf/grfpred.txt",header=TRUE)
yhat <- zreg$predicted
imputed <- (sum(w[group]*z$INTP[group])+sum(w[!group]*yhat[!group]))/sum(w)
print(imputed)
# 2302.425 (new INTP estimate)
```

```
# 12295.47 (old value before Professor Loh's update)
simple <- sum(w[group]*z$INTP[group])/sum(w[group])
print(simple)
# 2284.181 (new INTP estimate)
# 2404.231 (old value before Professor Loh's update)



#3 Repeat steps 1 and 2 via RPART
library(rpart)

vartype <- rep("factor",125)
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("class subset/classsubset.txt",header=TRUE,colClasses=vartype)

# R code for IPW estimate via RPART
tmp <- rep(NA,nrow(z))
tmp[z$FINTP == 0] <- 0
tmp[z$FINTP == 1] <- 1
z$FINTP <- tmp ### convert FINTP to binary variable (which it already is)
### regression tree without INTP and PWGTP
rp <- rpart(FINTP ~ . - INTP - PWGTP, data=z, method="anova")
plot(rp,compress=TRUE,margin=0.1)
text(rp) ### plot is on next page
p <- predict(rp) ### predicted prob(FINTP = 1)
w <- z$PWGTP
y <- z$INTP
gp <- !is.na(y)
ipw <- sum(w[gp]*y[gp]/p[gp])/sum(w[gp]/p[gp])
print(ipw)
```

```
# IPW =  2364.894 (new INTP estimate)
# IPW = 2540.947 (old value before Professor Loh's update)

#creating a confusion matrix to access accuracy for RPART classification tree
p2 =(p>=0.5)+0
table(z$FINTP, p2)
print((841 + 7781)/71066) # error rate of 0.1213238 for this dataset, worse than GUIDE!



#R code for computing ^y with RPART
vartype <- rep("factor",125)
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("reg subset/regsubset.txt",header=TRUE,colClasses=vartype)

z$INTP[is.na(z$INTP)] = mean(z$INTP,na.rm = T)
z$FINTP = NULL
rp2 <- rpart(INTP ~ ., weight=PWGTP, data=z,
             method="anova")
plot(rp2,compress=TRUE,margin=0.1)
text(rp2)
y <- z$INTP
w <- z$PWGTP
miss <- is.na(y) ## obs with missing INTRDVX
yhat <- predict(rp2,newdata=z)
popmean <- (sum(w[!miss]*y[!miss])+sum(w[miss]*yhat[miss]))/sum(w)
# 2378.515 (new INTP estimate)
# y.hat = 2066.158 (old value before Professor Loh's update)
print(popmean)
```

```
#3 R code for IPW Estimate via CTREE
library(party)

vartype <- rep("factor",125)
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("class subset/classsubset.txt",header=TRUE,colClasses=vartype)



tmp <- rep(NA,nrow(z))
tmp[z$FINTP == 0] <- 0
tmp[z$FINTP == 1] <- 1
z$FINTP <- tmp
### classification tree without INTP and PWGTP
fmla <- formula(FINTP ~ . - INTP - PWGTP)
z.small = z[sample(71066,11000),] # taking a random sample of the dataset

ct <- ctree(fmla, data=z.small)
plot(ct,type="simple",drop_terminal = TRUE)
y <- z$INTP
p <- predict(ct)
w <- z$PWGTP
gp <- !is.na(y)
ipw <-(sum(w[gp]*y[gp])/sum(p[gp]))/(sum(w[gp])/sum(p[gp]))
print(ipw) # Na (same with answer with older data prior to Professor Loh's annoucemet)
# CTREE has many values where p = 0, so it isn't that good for predicting ipw



# CTREE Regression Tree
library(party)
```

```
vartype <- rep("factor",125)
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("reg subset/regsubset.txt",header=TRUE,colClasses=vartype)
z.samp = z[sample(71066,10000),]

# old code
#z.samp$INTP[is.na(z.samp$INTP)] = mean(z.samp$INTP,na.rm = T)

# regression without FINTP and PWGTP
z.samp = z.samp[-which(is.na(z.samp$INTP)), ] # omit missing values in response
fmla <- formula(INTP ~ . - FINTP - PWGTP)
ct <- ctree(fmla, data=z.samp, weights = z.samp$PWGTP)
plot(ct,type="simple",drop_terminal = TRUE)
y <- z.samp$INTP
w <- z.samp$PWGTP
yhat <- predict(ct,newdata=z.samp)
miss <- is.na(y) ## obs with missing INTP
popmean <- (sum(w[!miss]*y[!miss])+sum(w[miss]*yhat[miss]))/sum(w)
print(popmean) #2691.584 (old value before Professor Loh's update)
# 2107.745 (new INTP estimate)


# 4. Imputing missing values in the X variables using MICE
z = read.table("cleandata.txt",header=TRUE, stringsAsFactors = TRUE)
x = read.table("class imp/classimp.scr",header=TRUE)

#top 15 variables for selection

top = c("ANC","ANC1P","NWAB","NWLK","RELSHIPP","NWLA","RC","OC","ANC2P",
```

```
      "NWRE","SCHL","MSP","POVPIP","ESR", "PWGTP")
# Note I initially included the weight variable then took it out later

red = z[,top] # reduced data frame for feature selection
y = z$FINTP

library(mice)

out = mice(red)

c1 <- complete(out,1) ## 1st imputed data set



# 5. Use logistic regression to find ˆpi and IPW to estimate μ
c1$FINTP = z$FINTP
c1$PWGTP = NULL # no weight variables for logistic regression
full = glm(FINTP~ . - FINTP, data = c1, family = "binomial")
summary(full)



# generating predictions from glm model
c1$pred = predict(full, type = "response")



c1$FINTPHAT=(c1$pred>=0.5)+0 # converting

z <- read.table("cleandata.txt",header=TRUE, stringsAsFactors = TRUE)
w <- z$PWGTP ### sampling weights
gp <- !is.na(z$INTP) # group of non missing Y values
```

```
p <- c1$pred # predicted p values
ipw <- sum(w[gp]*z$INTP[gp]/p[gp])/sum(w[gp]/p[gp])
print(ipw) # 2670.091
# old code used before Professor Loh's announcement

#c1$prediction = c1$probability
#for(i in 1:nrow(c1)){
#  if(c1$probability[i] < .5 ){
#    c1$prediction[i] = 0
#  } else{
#    c1$prediction[i] = 1
#  }
#
#}

#6. randomForest and CForest (party) to obtain ˆy and estimate μ with formula (1)

# via randomForest

#install.packages("randomForest")
library(randomForest)

# regression subset that is to be used
vartype <- rep("factor",125)
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("reg subset/regsubset.txt",header=TRUE,colClasses=vartype)



# easier for me to read the top 15 variables
```

```
x = read.table("reg imp/regtimp.scr",header=TRUE)
# top variables for feature selection
top = c("PINCP","AGEP","RETP","SCIENGP","SCIENGRLP",
        "SCHL","HINS3","POVPIP","SSP","FOD1P",
        "MARHYP","WAGP","GCL","RELSHIPP","PERNP")



# old code before Professor Loh's announcement is all the code that is commented out

#top = c('PINCP', "AGEP", "RETP","SCHL","SCIENGRLP",
#        "SCIENGP","HINS3","POVPIP","SSP","FOD1P", "MARHYP","WAGP",
#        "PERNP", "GCL", "RELSHIPP")

y = z$INTP
red = z[,top] # reduced data frame for feature selection

sum(is.na(y)) # tons of missing data

library(Amelia) # running imputations on the reduced dataset

#out <- amelia(red,noms=c(5,6,7,14), ords = c(4,10,15,11))
out = amelia(red,noms=c(4,5,7,13), ords = c(6,10,11,14))

c1 = out$imputations$imp1
c1$INTP = y

#c1$INTP[is.na(c1$INTP)] = 0 # changing all NAs to zero cannot have any INTP if you're less than 15
#old code
```

```
sum(is.na(c1$INTP)) # tons of rows where INTP is missing

c1 = c1[!is.na(c1$INTP),] # therefore will omit them

summary(c1)

str(c1)

#dropping markhyp and fod1p because they have too many levels (80 and 173 respectively)

c1$FOD1P = NULL

c1$MARHYP = NULL

rf <- randomForest(x=c1[,-14],y = c1$INTP)

### ipw estimate



w = z$PWGTP[gp] # weight variables

p = predict(rf,newdata = c1) # had to predict on the imputed dataset with no missing values, due to
randomForest

mean = sum(w[!is.na(c1$INTP)]*p)/sum(w[!is.na(c1$INTP)]) # note the simple imputation formula was
used because we

# cannot predict for missing values


print(mean) #2305.564


#y = c1$INTP

#gp = !is.na(z$INTP)

#mu = sum(w*y)/sum(w)

#print(mu) 2284.181 (old value prior to professor Loh's announcement)


# CForest


#Using the regression subset

vartype <- rep("factor",125)
```

```
vartype[c(2,3,28,29,55,56,58,62,64,65,66,67,69,97,98,100)] <- "numeric"
z <- read.table("reg subset/regsubset.txt",header=TRUE,colClasses=vartype)

library(party)
z = z[!is.na(z$INTP),] # all the INTP values that are non missing
z.small = z[sample(nrow(z),10000),] #subset of z to improve performance
fmla <- formula(INTP ~ . - FINTP - PWGTP)
cf <- cforest(fmla, data=z.small)
p <- predict(cf,newdata=z)
w <- z$PWGTP # weight variables
y <- z$INTP # INTP
miss <- is.na(y) ## obs with missing INTP
popmean <- (sum(w[!miss]*y[!miss])+sum(w[miss]*p[miss]))/sum(w)
print(popmean) # 2284.181
```

GUIDE Classification Tree Input Code (gtree.in)

```
GUIDE       (do not edit this file unless you know what you are doing)
 36.2       (version of GUIDE that generated this file)
 1          (1=model fitting, 2=importance or DIF scoring, 3=data conversion)
"gtree.out"  (name of output file)
 1          (1=one tree, 2=ensemble)
 1          (1=classification, 2=regression, 3=propensity score grouping)
 1          (1=simple model, 2=nearest-neighbor, 3=kernel)
 1          (0=linear 1st, 1=univariate 1st, 2=skip linear, 3=skip linear and interaction)
 1          (0=tree with fixed no. of nodes, 1=prune by CV, 2=by test sample, 3=no pruning)
"descclass.txt"  (name of data description file)
      10  (number of cross-validations)
 1          (1=mean-based CV tree, 2=median-based CV tree)
   0.500  (SE number for pruning)
 1          (1=estimated priors, 2=equal priors, 3=other priors)
 1          (1=unit misclassification costs, 2=other)
 2          (1=split point from quantiles, 2=use exhaustive search)
 1          (1=default max. number of split levels, 2=specify no. in next line)
 1          (1=default min. node size, 2=specify min. value in next line)
 2          (0=no LaTeX code, 1=tree without node numbers, 2=tree with node numbers)
"gtree.tex" (latex file name)
 1          (1=color terminal nodes, 2=no colors)
 2          (0=#errors, 1=sample sizes, 2=sample proportions, 3=posterior probs, 4=nothing)
 1          (1=no storage, 2=store fit and split variables, 3=store split variables and values)
 2          (1=do not save fitted values and node IDs, 2=save in a file)
"gtree.txt" (file name for fitted values and node IDs)
 2          (1=do not write R function, 2=write R function)
"gtree.R" (R code file)
```

1 (rank of top variable to split root node)

GUIDE Classification Forest Input Code (gcf.in)

GUIDE (do not edit this file unless you know what you are doing)

36.2 (version of GUIDE that generated this file)

1 (1=model fitting, 2=importance or DIF scoring, 3=data conversion)

"gcf.out" (name of output file)

2 (1=one tree, 2=ensemble)

2 (1=bagging, 2=rforest)

2 (1=random splits of missing values, 2=nonrandom)

1 (1=classification, 2=regression)

2 (1=interaction tests, 2=skip them)

"descclass.txt" (name of data description file)

1 (1=accept default number of trees, 2=change)

1 (1=accept default number of variables for splitting, 2=change it)

1 (1=estimated priors, 2=equal priors, 3=other priors)

1 (1=unit misclassification costs, 2=other)

1 (1=split point from quantiles, 2=use exhaustive search)

1 (1=accept default splitting fraction, 2=change it)

1 (1=default max. number of split levels, 2=specify no. in next line)

1 (1=default min. node size, 2=specify min. value in next line)

"gcfpred.txt" (file name for predicted class and probability estimates)

1 (rank of top variable to split root node)

GUIDE Regression Tree Input Code (regin.txt)

GUIDE (do not edit this file unless you know what you are doing)

36.2 (version of GUIDE that generated this file)

1 (1=model fitting, 2=importance or DIF scoring, 3=data conversion)

"regout.txt" (name of output file)

1 (1=one tree, 2=ensemble)

2 (1=classification, 2=regression, 3=propensity score grouping)

1 (1=linear, 2=quantile, 3=Poisson, 4=censored response, 5=multiresponse or itemresponse, 6=longitudinal with T vars, 7=logistic)

1 (1=least squares, 2=least median of squares)

3 (0=stepwise, 1=multiple linear, 2=simple polynomial, 3=constant, 4=ANCOVA)

1 (1=interaction tests, 2=skip them)

1 (0=tree with fixed no. of nodes, 1=prune by CV, 2=no pruning)

"descreg.txt" (name of data description file)

10 (number of cross-validations)

1 (1=mean-based CV tree, 2=median-based CV tree)

0.500 (SE number for pruning)

2 (1=split point from quantiles, 2=use exhaustive search)

1 (1=default max. number of split levels, 2=specify no. in next line)

1 (1=default min. node size, 2=specify min. value in next line)

2 (0=no LaTeX code, 1=tree without node numbers, 2=tree with node numbers)

"regtree.tex" (latex file name)

1 (0=all white,1=yellow-skyblue,2=yellow-purple,3=yellow-orange,4=orange-skyblue,5=yellow-red,6=orange-purple,7=grayscale)

1 (1=no storage, 2=store fit and split variables, 3=store split variables and values)

1 (1=do not save, 2=save regressor names in a file)

2 (1=do not save fitted values and node IDs, 2=save in a file)

"regtree.fit" (file name for fitted values and node IDs)

2 (1=do not write R function, 2=write R function)

"regtree.R" (R code file)

1 (rank of top variable to split root node)

GUIDE Regression Forest Input Code (grf.in)

GUIDE (do not edit this file unless you know what you are doing)

36.2 (version of GUIDE that generated this file)

1 (1=model fitting, 2=importance or DIF scoring, 3=data conversion)

"grf.out" (name of output file)

2 (1=one tree, 2=ensemble)

2 (1=bagging, 2=rforest)

2 (1=random splits of missing values, 2=nonrandom)

2 (1=classification, 2=regression)

2 (1=interaction tests, 2=skip them)

"descreg.txt" (name of data description file)

1 (1=accept default number of trees, 2=change)

1 (1=accept default number of variables for splitting, 2=change it)

1 (1=split point from quantiles, 2=use exhaustive search)

1 (1=accept default splitting fraction, 2=change it)

1 (1=default max. number of split levels, 2=specify no. in next line)

1 (1=default min. node size, 2=specify min. value in next line)

"grfpred.txt" (file name for predicted values)

1 (rank of top variable to split root node)

GUIDE Regression Importance Scoring Input Code (regtimp.in)

GUIDE (do not edit this file unless you know what you are doing)

36.2 (version of GUIDE that generated this file)

2 (1=model fitting, 2=importance or DIF scoring, 3=data conversion)

"regtimp.out" (name of output file)

2 (1=classification, 2=regression, 3=propensity score grouping)

1 (1=linear, 2=quantile, 3=Poisson, 4=censored response, 5=multiresponse or itemresponse, 6=longitudinal with T vars, 7=logistic)

1 (1=least squares, 2=least median of squares)

1 (1=interaction tests, 2=skip them)

"descreg.txt" (name of data description file)

2 (1=split point from quantiles, 2=use exhaustive search)

1 (1=default max. number of split levels, 2=specify no. in next line)

1 (1=default min. node size, 2=specify min. value in next line)

2 (0=no LaTeX code, 1=tree without node numbers, 2=tree with node numbers)

"regtimp.tex" (latex file name)

1 (0=all white,1=yellow-skyblue,2=yellow-purple,3=yellow-orange,4=orange-skyblue,5=yellow-red,6=orange-purple,7=grayscale)

1 (1=no storage, 2=store fit and split variables, 3=store split variables and values)

1 (1=do not save, 2=save regressor names in a file)

1 (1=do not create description file for selected variables, 2=create the file)

1 (1=create file for importance scores, 2=do not create)

"regtimp.scr" (file name for importance scores)

1 (rank of top variable to split root node)

GUIDE Classification Importance Scoring Input Code (classimp.in)

GUIDE (do not edit this file unless you know what you are doing)

36.2 (version of GUIDE that generated this file)

2 (1=model fitting, 2=importance or DIF scoring, 3=data conversion)

"classimp.out" (name of output file)

1 (1=classification, 2=regression, 3=propensity score grouping)

1 (1=univariate and interaction splits, 2=skip interactions)

"descclass.txt" (name of data description file)

1 (1=estimated priors, 2=equal priors, 3=other priors)

1 (1=unit misclassification costs, 2=other)

2 (1=split point from quantiles, 2=use exhaustive search)

1 (1=default max. number of split levels, 2=specify no. in next line)

1 (1=default min. node size, 2=specify min. value in next line)

2 (0=no LaTeX code, 1=tree without node numbers, 2=tree with node numbers)

"classimp.tex" (latex file name)

1 (1=color terminal nodes, 2=no colors)

2 (0=#errors, 1=sample sizes, 2=sample proportions, 3=posterior probs, 4=nothing)

1 (1=no storage, 2=store fit and split variables, 3=store split variables and values)

1 (1=do not create description file for selected variables, 2=create the file)

1 (1=create file for importance scores, 2=do not create)

"classimp.scr" (file name for importance scores)

1 (rank of top variable to split root node)

GUIDE Regression Subset Input Code (regsubset.in)

GUIDE (do not edit this file unless you know what you are doing)

36.2 (version of GUIDE that generated this file)

3 (1=model fitting, 2=importance or DIF scoring, 3=data conversion)

"regsubset.out" (name of output file)

2 (1=D is categorical, 2=D is real)

"descreg.txt" (name of data description file)

1 (choice of data format)

"regsubset.txt" (name of new data file)

GUIDE Classification Subset Input Code (classsubset.in)

GUIDE (do not edit this file unless you know what you are doing)

36.2 (version of GUIDE that generated this file)

3 (1=model fitting, 2=importance or DIF scoring, 3=data conversion)

"classsubset.out" (name of output file)

1 (1=D is categorical, 2=D is real)

"descclass.txt" (name of data description file)

1 (choice of data format)

"classsubset.txt" (name of new data file)

Appendix T

IPW Method

$$\left(\sum_{i\in S} w_i/\hat{\pi}_i\right)^{-1}\sum_{i\in S} w_i y_i/\hat{\pi}_i$$

Imputation Method

$$\left(\sum_{i\in S} w_i y_i + \sum_{j\in \bar{S}} w_j \hat{y}_j\right)/\sum_i w_{i\in S\cup\bar{S}}$$

Where S observations with non-missing INTP and $\bar{S}$ is its complement

Works Cited

"2019 ACS PUMS DATA DICTIONARY." US Census Bureau.

"AMERICAN COMMUNITY SURVEY 2019 ACS 1-YEAR PUMS FILES ReadMe." US Census Bureau, 15 Oct. 2020.

"American Community Survey Information Guide." US Census Bureau, Oct. 2017.

Loh, Wei-Yin. "STAT 443 Classification and Regression Trees." 19 Apr. 2021.

Loh, Wei-Yin. "User Manual for GUIDE Ver. 36.2∗." 10 Jan. 2021.