

**Impact of the Treatment Assignment Model on Propensity Score-Based Methods**

by

**Avantika Srivastava**

BA, Philosophy, University of Maryland, 2015

Submitted to the Graduate Faculty of  
the Department of Biostatistics  
Graduate School of Public Health in partial fulfillment  
of the requirements for the degree of  
Master of Science

University of Pittsburgh

2019

UNIVERSITY OF PITTSBURGH  
GRADUATE SCHOOL OF PUBLIC HEALTH

This thesis was presented

by

**Avantika Srivastava**

and defended on

April 11, 2019

and approved by

Abdus Wahed, PhD, Professor, Biostatistics  
Graduate School of Public Health, University of Pittsburgh

Stewart Anderson, PhD, Professor, Biostatistics  
Graduate School of Public Health, University of Pittsburgh

**Thesis Advisor:** Douglas Landsittel, PhD, Professor, Biomedical Informatics  
School of Medicine, University of Pittsburgh

Copyright © by Avantika Srivastava

2019

# **Impact of the Treatment Assignment Model on Propensity Score-Based Methods**

Avantika Srivastava, MS

University of Pittsburgh, 2019

## **Abstract**

Standard statistical methods assess the association between treatments or exposures and the outcome. To better estimate the causal effect, defined as the expected difference in potential outcomes, several methods have been proposed, including propensity-score (PS) based methods. PS-based methods use the predicted probability of being assigned to a given treatment to produce a pseudo population which better resembles that of a randomized trial. Standard statistical methods can then be applied to that pseudo population to better estimate the causal effect.

The goal of this thesis is to describe how choice of treatment assignment model can impact the results of PS-based methods, using an empirical example of a binary outcome with a binary treatment variable and both continuous and categorical confounders. We hypothesize that using different treatment assignment mechanisms will produce differing degrees of overlap (between treatment groups), which will then lead to different pseudo populations and different estimates of treatment effect.

The treatment assignment mechanism was modeled using three approaches: logistic regression, classification trees, and a random forest model. Next, three pseudo populations were created from each of the resulting PS distributions: one using 1:1 propensity score matching, one using stratification into quintiles, and one using inverse probability of treatment weighting (IPTW). Covariate balance was assessed by calculating the standardized mean differences of

covariates in the entire sample and in each pseudo population. A Cox proportional hazard model was then fit for each pseudo population to estimate the treatment effect.

Results varied for different outcomes models. The forest model gave a significant estimate in matched pseudo populations, no model was significant in stratified pseudo populations, and both the unpruned tree and forest models gave significant estimates in IPTW pseudo populations. In conclusion, these mixed results indicated that the assignment mechanism model, the approach for forming the pseudo population, and the choice of outcomes model, can all significantly influence results.

This thesis is significant to public health because it illustrates a comparative effectiveness research analysis of the causal effect of two treatments using only observational data. The methods used are a frequent approach in public health studies involving nonrandomized data.

## Table of Contents

Preface.....	xii
<b>1.0 Introduction.....</b>	<b>1</b>
<b>1.1 Public Health Significance .....</b>	<b>1</b>
<b>1.2 Defining the Causal Effect .....</b>	<b>2</b>
<b>1.3 Propensity Score-Based Methods.....</b>	<b>3</b>
<b>1.4 Other Causal Inference Methods .....</b>	<b>6</b>
<b>1.5 Example Data Set .....</b>	<b>7</b>
<b>2.0 Methods.....</b>	<b>9</b>
<b>2.1 Treatment Assignment Mechanism .....</b>	<b>9</b>
<b>2.1.1 Logistic Regression .....</b>	<b>10</b>
<b>2.1.2 Classification Trees .....</b>	<b>10</b>
<b>2.1.3 Random Forest .....</b>	<b>13</b>
<b>2.2 Pseudo Population .....</b>	<b>13</b>
<b>2.2.1 Propensity Score Matching .....</b>	<b>13</b>
<b>2.2.2 Stratification .....</b>	<b>14</b>
<b>2.2.3 Inverse Probability of Treatment Weighting .....</b>	<b>15</b>
<b>2.3 Covariate Balance.....</b>	<b>16</b>
<b>2.4 Outcomes Model .....</b>	<b>16</b>
<b>3.0 Results .....</b>	<b>18</b>
<b>3.1 Summary statistics for variables at baseline.....</b>	<b>18</b>
<b>3.2 Propensity Score Distributions.....</b>	<b>20</b>

<b>3.3 Pseudo Populations.....</b>	<b>23</b>
<b>3.4 Covariate Balance.....</b>	<b>24</b>
<b>3.5 Outcomes Model.....</b>	<b>34</b>
<b>4.0 Discussion.....</b>	<b>50</b>
<b>Appendix A Additional Output for Classification Tree Models.....</b>	<b>54</b>
<b>Appendix B Post-Pseudo Randomization Propensities.....</b>	<b>58</b>
<b>Appendix C Code.....</b>	<b>65</b>
<b>Bibliography.....</b>	<b>98</b>

## List of Tables

Table 1 Summary statistics by transplant type .....	19
Table 2 Number of Deaths in 1:1 Matched Pseudo Populations .....	24
Table 3 Matched Observations by Treatment Assignment.....	24
Table 4 Survival Analysis – 1:1 Matching .....	35
Table 5 Survival Analysis – Stratification .....	36
Table 6 Survival Analysis – IPTW .....	36

## List of Figures

Figure 1 Overall Process of Propensity Score-Based Methods .....	5
Figure 2 Histogram of pre-pseudo randomized propensities (logistic regression).....	20
Figure 3 Histogram of pre-pseudo randomized propensities (classification tree, cp=0.0).....	21
Figure 4 Histogram of pre-pseudo randomized propensities (classification tree, cp=0.0157).....	21
Figure 5 Histogram of pre-pseudo randomized propensities (classification tree, cp=0.0).....	22
Figure 6 Histogram of pre-pseudo randomized propensities (classification tree, cp=0.01).....	22
Figure 7 Histogram of pre-pseudo randomized propensities (random forest).....	23
Figure 8 Balance Plot for 1:1 Matching, Logistic Regression.....	25
Figure 9 Balance Plot for 1:1 Matching, Classification Tree (cp=0.0).....	26
Figure 10 Balance Plot for 1:1 Matching, Classification Tree (cp=0.0157).....	26
Figure 11 Balance Plot for 1:1 Matching, Classification Tree (cp=0.0065).....	27
Figure 12 Balance Plot for 1:1 Matching, Classification Tree (cp=0.01).....	27
Figure 13 Balance Plot for 1:1 Matching, Random Forest .....	28
Figure 14 Balance Plot for Stratification (5 strata), Logistic Regression.....	28
Figure 15 Balance Plot for Stratification (5 strata), Classification Tree (cp=0.0).....	29
Figure 16 Balance Plot for Stratification (5 strata), Classification Tree (cp=0.0157).....	29
Figure 17 Balance Plot for Stratification (5 strata), Classification Tree (cp=0.0065).....	30
Figure 18 Balance Plot for Stratification (5 strata), Classification Tree (cp=0.01).....	30
Figure 19 Balance Plot for Stratification (5 strata), Random Forest .....	31
Figure 20 Balance Plot for IPTW, Logistic Regression .....	31
Figure 21 Balance Plot for IPTW, Classification Tree (cp=0.0) .....	32

Figure 22 Balance Plot for IPTW, Classification Tree (cp=0.0157) .....	32
Figure 23 Balance Plot for IPTW, Classification Tree (cp=0.0065) .....	33
Figure 24 Balance Plot for IPTW, Classification Tree (cp=0.01) .....	33
Figure 25 Balance Plot for IPTW, Random Forest.....	34
Figure 26 Kaplan-Meier curves – Logistic Regression (1:1 matching).....	37
Figure 27 Kaplan-Meier curves – Classification Tree, cp=0.0 (1:1 matching) .....	38
Figure 28 Kaplan-Meier curves – Classification Tree, cp=0.0157 (1:1 matching) .....	38
Figure 29 Kaplan-Meier curves – Classification Tree, cp=0.0065 (1:1 matching) .....	39
Figure 30 Kaplan-Meier curves – Classification Tree, cp=0.01 (1:1 matching) .....	39
Figure 31 Kaplan-Meier curves – Random Forest (1:1 matching).....	40
Figure 32 Kaplan-Meier curves – Logistic Regression (stratification) .....	41
Figure 33 Kaplan-Meier curves – Classification Tree, cp=0.0 (stratification).....	42
Figure 34 Kaplan-Meier curves – Classification Tree, cp=0.0157 (stratification).....	43
Figure 35 Kaplan-Meier curves – Classification Tree, cp=0.0065 (stratification).....	44
Figure 36 Kaplan-Meier curves – Classification Tree, cp=0.01 (stratification).....	45
Figure 37 Kaplan-Meier curves – Random Forest (stratification).....	46
Figure 38 Kaplan-Meier curves – Logistic Regression (IPTW).....	47
Figure 39 Kaplan-Meier curves – Classification Tree, cp=0.0 (IPTW) .....	47
Figure 40 Kaplan-Meier curves – Classification Tree, cp=0.0157 (IPTW) .....	48
Figure 41 Kaplan-Meier curves – Classification Tree, cp=0.0065 (IPTW) .....	48
Figure 42 Kaplan-Meier curves – Classification Tree, cp=0.01 (IPTW) .....	49
Figure 43 Kaplan-Meier curves – Random Forest (IPTW) .....	49
Figure 44 Complexity Parameter Selection .....	54

Figure 45 Classification Tree (cp=0.0) .....	55
Figure 46 Classification Tree (cp=0.0157) .....	56
Figure 47 Classification Tree (cp=0.0065) .....	56
Figure 48 Classification Tree (cp=0.01) .....	57
Figure 49 Post-pseudo randomized PS distributions, 1:1 Matching, Logistic Regression.....	58
Figure 50 Post-pseudo randomized PS distributions, 1:1 Matching, Tree (cp=0.0).....	59
Figure 51 Post-pseudo randomized PS distributions, 1:1 Matching, Tree (cp=0.0157).....	59
Figure 52 Post-pseudo randomized PS distributions, 1:1 Matching, Tree (cp=0.0065).....	60
Figure 53 Post-pseudo randomized PS distributions, 1:1 Matching, Tree (cp=0.01).....	60
Figure 54 Post-pseudo randomized PS distributions, 1:1 Matching, Random Forest .....	61
Figure 55 Post-pseudo randomized PS distributions, IPTW, Logistic Regression.....	61
Figure 56 Post-pseudo randomized PS distributions, IPTW, Tree (cp=0.0) .....	62
Figure 57 Post-pseudo randomized PS distributions, IPTW, Tree (cp=0.0157) .....	62
Figure 58 Post-pseudo randomized PS distributions, IPTW, Tree (cp=0.0065) .....	63
Figure 59 Post-pseudo randomized PS distributions, IPTW, Tree (cp=0.01) .....	63
Figure 60 Post-pseudo randomized PS distributions, IPTW, Random Forest.....	64

## Preface

I would first like to thank my thesis advisor, Dr. Douglas Landsittel, for all his support, guidance, and encouragement throughout this endeavor. His guidance over the past year has not just helped me understand and utilize the research tools I needed for this thesis, but also better prepared me for further studies, and helped me to develop research and organizational skills that will continue to benefit my future work.

I would like to thank Drs. Abdus Wahed and Stewart Anderson for serving on my thesis committee and for their feedback during this process. Beyond giving me further insights on my analysis, their comments and advice also helped me to better present my research. I also want to thank all the faculty in the Biostatistics department at the University of Pittsburgh, who have helped me continuously over the past two years. Their guidance has been invaluable during my time as a master's student and will continue to be so throughout my career.

I would also like to thank Carlos Fernández-Carrillo and Ramon Bataller Alberola from the Division of Gastroenterology, Hepatology, and Nutrition at the University of Pittsburgh Medical Center, for providing the data used in this analysis and for their guidance on clinical issues regarding the study.

Finally, I want to thank my family, particularly my parents, for their continuous love and support up to and including my time in this master's program. Their encouragement was a huge source of strength and motivation these past two years, and I could not have done it without them.

## **1.0 Introduction**

The goal of this thesis is to present a framework for implementing PS-based methods, and investigate how different variations to different components of that framework can affect final results, using an illustrative empirical example to test that hypothesis. Section 1.2 defines the concept of causal inference using potential outcomes, Section 1.3 gives a general overview of PS-based methods in comparative effectiveness research, briefly explains classification trees and random forests, and provides some background on the following methods used to generate pseudo populations: propensity score matching, stratification, and inverse probability of treatment weighting (IPTW). Section 1.4 introduces the dataset, describes the variables used in this analysis, and discusses the motivation for this particular analysis. Section 2 outlines the methodology for each step of the PS analysis; 2.1 describes the three treatment mechanisms used to generate propensity scores, 2.2 defines the three methods used to create pseudo populations, 2.3 defines the statistics used to measure covariate balance, and 2.4 defines the outcomes model. Section 3 presents the results of the PS analysis, and sections 4 and 5 contain the discussion and conclusion respectively.

### **1.1 Public Health Significance**

This paper considers propensity score-based methods, which are becoming an increasingly popular causal inference method used to analyze data from observational studies. While a considerable amount of the information used in studies comparing treatments can come

from randomized data, biostatisticians also work with observational data, which can have systematic bias in treatment selection for patients with certain covariates. Such bias needs to be addressed before the data can be used to accurately estimate the causal effect. This thesis illustrates one such instance of comparative effectiveness research and presents an analysis of the causal effect of two treatments using only observational data. The methods used to reduce the bias described above are a frequent research concern in public health studies involving nonrandomized data.

## **1.2 Defining the Causal Effect**

Causal inference methods aim to estimate the causal effect of a treatment on an outcome, which requires first selecting a framework for the causal effect being studied. One possible framework is that of potential outcomes, which are defined as the possible outcomes for a subject if they received a certain treatment. If a study is comparing two treatments A and B, then each subject in the sample would have two potential outcomes: the outcome if they receive treatment A, and the outcome if they receive treatment B.

The causal effect can then be defined as the difference in potential outcomes; a measure for this is the average treatment effect (ATE), which is the expected difference in potential outcomes. However, the problem with directly estimating the ATE is that each subject only experiences a single outcome, which means only the observed outcomes are available for analysis and it is not possible to compare both potential outcomes from the same subject at the same time point. With half the potential outcomes missing for each observation, causal inference then

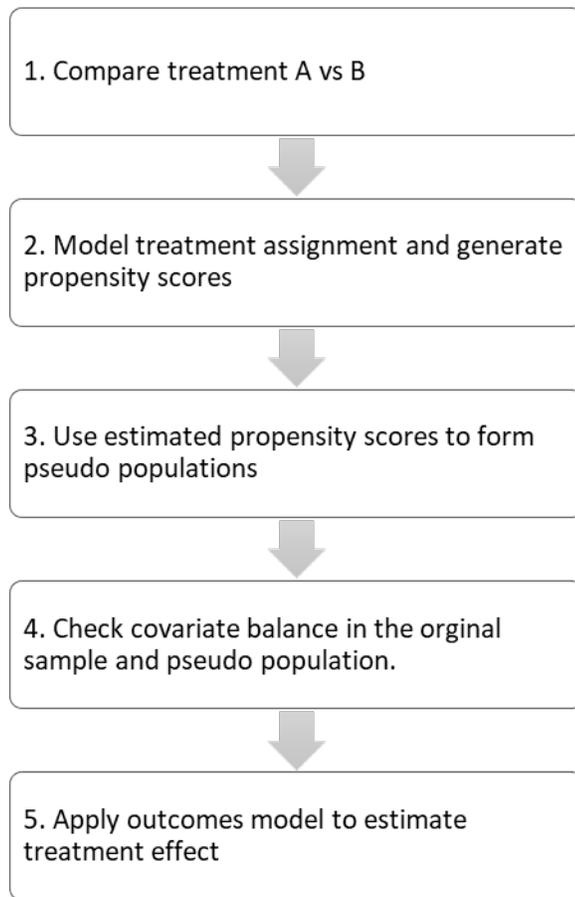
becomes a missing data problem, and requires methods that address it as such. (Rosenbaum and Rubin, 1983)

If we define the potential outcomes for two treatments  $T=0$  and  $T=1$  in subject  $i$  as  $Y_{i0}$  and  $Y_{i1}$  respectively, then the ATE would be given by  $E[Y_{i1} - Y_{i0}]$ . But rather than giving values for  $E[Y_0]$  and  $E[Y_1]$ , the data produces conditional expectations for each potential outcome given a certain treatment, which can be defined as  $E[Y_0|T = 0]$  and  $E[Y_1|T = 1]$ . Here, standard regression methods would take the difference in conditional expectations,  $E[Y_{i1}|T = 1] - E[Y_0|T = 0]$ , but these methods are not equivalent to estimating the ATE. They do not compare potential outcomes (for the same subjects at the same time), and therefore do not measure the causal relationship between treatment and outcome. (Rubin, 2005) Several methods exist to address this problem and accurately estimate the ATE, one of which are PS-based methods.

### **1.3 Propensity Score-Based Methods**

Comparative effectiveness research refers to questions regarding which treatment works best for a patient, given a certain set of characteristics. However, not all treatments can be studied using data from randomized trials; doing so is often either unfeasible or unethical, and in such cases, researchers must rely on data from observational studies instead. This data usually contains a systematic bias coming from certain kinds of patients being more likely to receive one treatment over the other. To accurately estimate the causal effect in this case, researchers need to first create a pseudo population with more balanced treatment arms that mimic those of a randomized trial. Standard statistical methods used for randomized data can then be applied on this pseudo population to estimate the treatment effect. (Austin and Stuart, 2015)

One possible way to create these pseudo populations is to use PS-based methods. Rather than being one fixed method, PS-based methods are a multi-step process, whose framework is outlined in five steps in Figure 1. The first step is to frame the research question as a hypothetical randomized trial comparing two treatments and to identify the treatment of interest. The propensity score is then calculated as the predicted probability of this treatment. The second step is to model the treatment assignment mechanism using either standard regression approaches or more complex methods (e.g. machine learning) to generate the estimated propensities; this paper explores three methods for treatment assignment, including logistic regression, classification trees, and random forests, which are described in Section 2.1. In step 3, the estimated propensity scores are used to create a pseudo population that better mimics data from a randomized sample. This thesis uses three methods to form the pseudo population for each set of propensity scores: propensity score matching, stratification, and inverse probability of treatment weighting, which are described in Section 2.2. In step 4, covariate balance is then assessed in the pseudo population and compared to the original sample. Then in step 5, an outcomes model is fit to the pseudo population to estimate the treatment effect.



**Figure 1 Overall Process of Propensity Score-Based Methods**

Two primary assumptions must hold in order to use PS-based methods to compare the effectiveness of two treatments. The first assumption is that of unconfoundedness, which says that treatment assignment is independent from the outcome conditional on covariates used in the model – that is, we have correctly and completely specified the model in order to prevent unmeasured confounding, and there are no unmeasured covariates. The second assumption is the overlap assumption, which says that given the set of measured confounders, observations must have a nonzero probability of being assignment to either treatment. That is, for a set of covariates  $X$  and binary treatment  $T$ ,  $0 < P(T = 1|X) < 1$ . (Caliendo and Kopeinig, 2008)

The hypothesis considered by this thesis is that using different treatment assignment mechanisms will produce differing degrees of overlap between treatment groups, which will then

lead to different pseudo populations and different estimates of treatment effect. This paper will also consider the effects of using different methods to create the pseudo population, which is hypothesized to also produce different estimates of the treatment effect.

#### **1.4 Other Causal Inference Methods**

PS-based methods are only one possible approach for dealing with the bias caused by lack of randomization in causal inference with observational studies; other commonly-used methods include instrumental variables, g-estimation, marginal structural models, and structural nested mean models. This section provides a brief overview of some of these additional methods.

Instrumental variable (IV) methods are a common alternate approach to comparative effectiveness research, and can control for unobserved confounding in a way PS-based methods cannot. In place of the usual randomization of a clinical trial, a variable is chosen that is strongly related to treatment but does not have any independent association with the outcome of interest. Consider a treatment variable for patients receiving catheterization upon hospital admission – in this case, an IV could be distance from a hospital that provides catheterization. Three conditions must now hold for this IV: the IV influences the treatment variable, the IV is not directly associated with the outcome variable (except through the treatment), and the IV does not share common causes with outcome to indirectly associate the two variables. Provided such a variable can be found and these conditions hold, it is possible to use IV methods and accurately estimate the treatment effect. (Rassen et al., 2009)

Marginal structural models and structural nested mean models are another set of alternative methods and can use time-varying covariates to estimate the causal effect of a binary treatment.

Unlike IVs, these covariates can be independently associated with both the treatment and outcome. Structural nested mean models use covariates up to time  $t$  to estimate the treatment effect at that time  $t$ , while marginal structural models use the baseline covariates. (Robins et al., 1999)

There are also tools available for causal inference involving time-varying and/or adaptive treatment strategies, which require additional assumptions regarding multiple time points to ensure unbiased estimates of treatment effect. Two such examples are marginal structural models with inverse probability of treatment weighting, and structural nested mean models with g-estimation, which can be used on longitudinal data to estimate a time-varying treatment's causal effect. (Young et al., 2010) However, PS-based methods – and this thesis – only work with a single time point intervention.

## 1.5 Example Data Set

The goal of this analysis was to conduct a propensity score-based analysis to compare the effectiveness of simultaneous kidney and liver transplants (SLKT) to liver-only transplants (LTA), in patients with either alcoholic liver disease (ALD) or nonalcoholic fatty liver disease (NAFLD), who were on the liver transplant wait list with dialysis between 2006-2016, with a minimum of one year of follow-up.

The motivation behind comparing these treatments was the question of assigning liver transplant patients to SLKT over LTA, as guidelines for choosing between the two treatments continue to change and remain unclear. Comparative research on SLKT and LTA patients has been done before and is still a concern, since SLKT rates are rising and there does not seem to be a set of standardized medical criteria for allocating these transplants. (Miles et al., 2018)

While there is not a strong consensus on whether there is significant benefit to one transplant type over the other, there have been notable differences in how the two treatments are assigned – for example, by transplant region (categorized by the Organ Procurement and Transplantation Network), which might be a seemingly non-clinical association. (Sharma et al., 2016) Studies have been done using various sets of covariates to compare the effectiveness of SLKT over LTA in different categories of liver transplant candidates, such as patients with renal dysfunction (Tanriover et al., 2016, Pham et al., 2016), with some studies using PS-based methods as well (Sharma et al., 2016, Tanriover et al., 2016).

This thesis performed a similar study with the subgroup of patients described earlier. SLKT was considered the treatment of interest, while LTA was considered the ‘control’ treatment. Covariates considered were gender (male/female), diabetes (none, type II, type I), region where listed/transplanted (regions 1-11), on ventilator at transplant registration (yes/no), risk factors for portal vein thrombosis at transplant registration (yes/no), etiology (ALD or NAFLD), race (Other, Hispanic/Latino, Black), BMI, and age. These variables were identified a priori by the investigators. Total survival time was used as the outcome measure, with death as the event of interest.

## **2.0 Methods**

The analysis was conducted in five steps. First, all variables were summarized at baseline. Second, the treatment assignment was modeled, and corresponding propensity scores were generated. Third, the estimated propensity scores were used to form a pseudo population through 1:1 matching, inverse probability of treatment weighting or stratification. Fourth, covariate balance was checked in the original sample and pseudo population. Fifth, an outcomes model for overall survival (consistent to how the pseudo population was created) was applied to the pseudo population to estimate treatment effect.

### **2.1 Treatment Assignment Mechanism**

Three methods were used to model treatment assignment and generate propensity score estimates: logistic regression, classification trees, and a random forest. Four classification trees were considered for this analysis, resulting in six different treatment assignment models and corresponding sets of propensity scores. The reason these logistic, tree and forest models were considered was because they represent a standard parametric approach, an easily interpretable nonparametric approach, and a more complex machine learning method respectively. (Austin, 2013)

### 2.1.1 Logistic Regression

The first treatment assignment mechanism model was a standard logistic regression where the logit of SLKT was modeled using a weighted linear sum of the given covariates. (Hosmer et al., 2013) The model equation is defined below in equation (1), where  $Y$  is the probability of being assigned to the treatment of interest (SLKT),  $X_i$  are the  $i$  measured confounders, and  $\beta_i$  are the estimated logistic regression parameters:

$$\text{logit}(Y) = \ln\left(\frac{Y}{1-Y}\right) = \beta_0 + \beta_1 X_1 + \dots + \beta_i X_i \quad (1)$$

Fitted values are found by using the MLE to find  $\hat{\beta}$  estimates that minimize the log likelihood, and estimated propensity scores  $\hat{Y}$  are calculated using equation (2) below:

$$\hat{Y} = \frac{1}{1 + \exp\{-(\beta_0 + \beta_1 X_1 + \dots + \beta_i X_i)\}} \quad (2)$$

Interaction and nonlinear terms were not considered for the logit model used in this analysis.

### 2.1.2 Classification Trees

The second treatment assignment model used in this analysis to generate a set of estimated propensity scores was a classification tree. Classification trees recursively partition the data, one variable at a time, into binary subgroups that are more alike with respect to treatment. (Austin, 2013) The metric for evaluating how well these ‘splits’ classify the data into the two treatment subgroups is decided by the Gini index defined in equation (3), where the variable used to create the split has a value of  $t = 0$  or  $t = 1$ , and  $P_t$  is the proportion of observations with the value  $t$ :

$$GI = 1 - \sum_{t=0}^1 P_t^2 \quad (3)$$

The steps for using classification tree model can be outlined as follows, using the methods of Atkinson and Therneau (2000):

1. Grow the classification tree. This is done by calculating the Gini index for each variable and selecting the variable with the lowest Gini index value to create the split. At each new node created by the split, the Gini index is again calculated for all the variables with the subset of data and used to create another split. This process is repeated to grow the tree until there are no possible splits to improve the partitioning of the data.
2. Prune the tree. The fully-grown tree has the disadvantage of potentially overfitting the data, so a pruning parameter, referred to as the complexity parameter and defined as the cost of adding more variables to the tree, is chosen that minimizes cross-validated error. The tree is then pruned from the bottom up to eliminate non-significant splits, using the chosen complexity parameter to determine how many nodes are removed. A complexity parameter of 0 will generate an unpruned tree, while a value of 1 will generate a tree with no splits.
3. Generate PS estimates for each observation. The predicted probability for an observation being assigned to a treatment group is calculated as shown in equation (4), where  $n_{SLKT}$  is the number of events of interest (SLKT) at the observation's terminal node, and  $n_{tot}$  is the total number of observations at that node:

$$\hat{Y} = \frac{n_{SLKT}}{n_{tot}} \quad (4)$$

Classification trees models were created using the 'rpart' package in R. Four trees were considered in this analysis: the unpruned tree, the optimally pruned tree – determined by the complexity parameter (CP) with the lowest value, the second-most optimally pruned tree, and a tree pruned with a CP of 0.1, which was a standard CP value in previous literature.

The 'rpart' package calculates a complexity parameter for every node – this value is defined as the amount by which splitting the tree at that node would improve the cross-validated error. When trees are pruned, nodes that do not improve cross-validated error by the chosen value are cut from the tree to create a 'sub tree'. Rather than being single values, complexity parameters are initially given as complexity intervals corresponding to each sub tree, where all complexity parameter values within that interval generate the same tree. The representative single complexity parameter value for each interval is calculated by taking the square root of the endpoints of that interval.

The optimal complexity parameter was chosen by selecting the sub tree with a value that minimized cross-validated error. The tree with complexity parameter (from the next interval) that had the second-lowest cross-validated error was chosen as the second-most optimally pruned tree. Cross-validated error was calculated with rpart's built-in cross-validation algorithm, which uses 10-fold cross-validation and is scaled so the first node (given in output as the tree with no splits) has an error of one. The steps for estimating cross-validation error for each complexity parameter (corresponding to a different pruned tree) are given below (Luellen et al., 2006):

1. The full dataset is used to fit the tree model.
2. The dataset is split into ten approximately equal-sized parts, or folds.
3. The tree models for each complexity parameter are fit using nine of those ten folds.
4. The prediction error is estimated for each tree using the remaining tenth fold. This error value is calculated using the misclassification risk, which is the proportion of observations whose predictions were incorrectly assigned.
5. Steps 3 and 4 are repeated ten times, setting aside a different fold each time to estimate prediction error.

6. Cross-validated error for every tree model (i.e., every complexity parameter) is then estimated by summing over the model's ten individual prediction errors. Once the error values are scaled as described earlier, the complexity parameters with the smallest cross-validated error values are used to fit pruned trees to the entire dataset.

### **2.1.3 Random Forest**

The third method used to estimate propensities for this analysis was a random forest. Random forests fit a chosen number of classification trees to different bootstrap samples of the data, with each split of each tree using a random selection of the x-variables. The model then averages predictions over all the trees, which leads to more stable estimates and accurate prediction than a single tree model. The random forest model in this analysis was fitted using the 'randomForest' package in R and used 10,000 trees to generate the propensity scores.

## **2.2 Pseudo Population**

### **2.2.1 Propensity Score Matching**

Once a set of estimated propensity scores has been generated, one method to create a corresponding pseudo population is to use propensity score matching. 1:1 PS matching, using nearest neighbor matching, finds the closest PS values between the two treatments, and matches those observations for the treatment of interest with one observation from the 'control' treatment.

For 1:m PS matching, treatment of interest observations are matched to m ‘control’ observations. The treatment effect can then be found by comparing outcomes between the two treatments within a matched set. (Burden et al., 2017).

Nearest neighbor matching creates the matched sets by finding sets of propensity scores within a certain caliper distance and leaves any observations with propensities greater than that distance unmatched. This distance measure can be defined by (5) below, where  $p_i$  and  $p_j$  are the PS estimates for two observations. Once a set of observations  $p_i$  and  $p_j$  with the lowest  $d_i$  is matched, the next set of observations from the remaining sample with the lowest  $d_i$  is matched, and so on until as many observations as possible have been matched.

$$d_i = |p_i - p_j| \quad (5)$$

Matched pseudo populations were generated with the ‘MatchIt’ package in R. 1:1 nearest neighbor matching was used, with caliper distance  $cd$  calculated as given in equation (6), where  $\sigma_p$  was the standard deviation of the propensity scores:

$$cd = 0.2 * \sigma_p \quad (6)$$

### 2.2.2 Stratification

Another method of generating a pseudo population is to use propensity score stratification, also referred to as subclassification. Stratification is an alternative to matching, since it may not always be possible to match all the observations in a sample to exactly or close to the same propensity score; instead, researchers might prefer to use all the observations in the sample. The data is stratified by sorting the PSs from lowest to highest, splitting the propensity scores into a

given number of mutually exclusive strata and then taking the mean for each treatment group across those strata, assuming homogeneity within strata. (Burden et al., 2017)

Stratified pseudo populations in this analysis were split into quintiles, since this is sufficient to remove at least 90% of the bias in treatment effect estimates. (Rosenbaum and Rubin, 1985)

### 2.2.3 Inverse Probability of Treatment Weighting

A third method of creating the pseudo population is to use inverse probability of treatment weighting (IPTW). IPTW uses the inverse of each observation's estimated propensity score to reweight that observation and create a weighted data set that resembles a hypothetical sample with equal propensity score distributions between the two treatments. This method is similar to the way survey weights are used in a sample to reflect a certain distribution in the population sampled from. (Burden et al., 2017) However, it should be noted that unlike survey weighting, IPTW does not use known probabilities to create the reweighted population. While the use of estimated propensity scores here instead of known probabilities is common, it is still a limitation of this approach.

Propensity scores were weighted using the 'survey' package in R. Observations with SLKT were weighted by the inverse of the probability of receiving their assigned treatment, which is given in (7), where observations  $i$  were weighted with weight  $W$ , using their estimated propensity score  $p$ , and treatment indicator  $T$  ( $T = 1$  for SLKT):

$$W_i = \frac{1}{p_i} * T + \frac{1}{1 - p_i} * (1 - T) \quad (7)$$

### 2.3 Covariate Balance

Covariate balance was assessed by calculating the standardized mean differences of covariates in the entire sample and in each pseudo population. The ‘cobalt’ package in R was used; standardized difference was defined as the difference in means between LTA and SLKT groups, divided by a measure of spread. The 1:1 matched and stratified pseudo populations used the standard deviation of the LTA group as the measure of spread, while the IPTW pseudo population used the pooled standard deviation, defined as the square root of the mean of the group variances. (Burden et al., 2017) That is, the standardized mean difference (SMD) was calculated as follows in (8), (9) and (10) for continuous variables, with mean  $\mu$  and standard deviation  $\sigma$ :

$$SMD_{matching/stratification} = \frac{|\mu_{SLKT} - \mu_{LTA}|}{\sigma_{LTA}} \quad (8)$$

$$SMD_{IPTW} = \frac{|\mu_{SLKT} - \mu_{LTA}|}{\sigma_{pooled}} \quad (9)$$

where

$$\sigma_{pooled} = \sqrt{\frac{\sigma_{SLKT}^2 + \sigma_{LTA}^2}{2}} \quad (10)$$

For binary variables, the standardized difference was the difference in proportion, given in (11):

$$SMD = |p_{SLKT} - p_{LTA}| \quad (11)$$

### 2.4 Outcomes Model

A Cox proportional hazard model was then fit for the pseudo population to estimate the treatment effect as defined in equation (12), where  $h(t)$  is defined as the hazard at time  $t$  for

treatment type  $X$ . The hazard ratio is used to estimate the relative risk of the event of interest occurring at time  $t$  for SLKT patients, compared to LTA patients, and is given in (13). (Kalbfleisch and Prentice, 2002)

$$h(t) = h_0(t) \exp(\beta X) \quad (12)$$

$$HR = \frac{\lambda_1(t)}{\lambda_0(t)} \quad (13)$$

Each outcome model was consistent with the method used to create the pseudo populations; matched pseudo populations used clustering on the matched set, stratified pseudo populations were divided into quintiles, and the IPTW pseudo-populations were analyzed using survey weights using the ‘survey’ package in R. The Cox models were fitted under the assumption that hazard ratios for treatment estimate were constant over time.

For the matched pseudo populations, the Cox proportional hazard was fit using clustering on the matching ID, which was the variable used to identify matched pairs. Clustering by the matching ID allowed the Cox model to treat matched pairs as potentially correlated while assuming each of these pairs were independent from one another. Variance was then calculated for the model using a robust sandwich estimator, which was aimed at correcting the bias caused by the clustered pairs. For the ‘cluster’ function with ‘coxph’ in R, this variance is defined as the “working independence” variance in a GEE model.

For the stratified pseudo populations, the Cox model was split into the same PS quintiles defined when creating the pseudo population, and a separate baseline hazard function was fit for each of these five strata. For the IPTW pseudo populations, a weighted Cox proportional hazards model was fitted using the same weights for each observation as the ones used to create the pseudo

population:  $W_{i,SLKT} = \frac{1}{p_i} * T$  and  $W_{i,LTA} = \frac{1}{1-p_i} * (1 - T)$ .

### **3.0 Results**

This section presents the results of the methods described above. Section 3.2 gives the pre-pseudo randomization propensity score distributions generated by each treatment assignment model in figures 2 through 7, 3.3 shows covariate balance in the pseudo populations, 3.4 presents the hazard ratios from the fitted outcomes models, and 3.5 gives the Kaplan Meier Curves for overall survival in the pseudo populations.

#### **3.1 Summary statistics for variables at baseline**

Table 1 provides baseline summary statistics for all covariates considered in the analysis, survival time, and the outcome of interest. The table reports frequencies for categorical variables, and the mean, standard deviation, median, minimum and maximum values for continuous variables. Summary statistics were calculated for both the entire sample and for subgroups of each transplant type.

As shown in the table, most covariates appeared to have different values for the SLKT and LTA groups before any analysis was performed. As previous literature suggested, there were some notable differences in SLKT and LTA assignment based on the transplant region – for example, about 12.6% of the transplants from LTA patients were done in region 7, while 22% of the transplants from SLKT patients were done in the same reason. Region 5 had similar dramatically different frequencies. Looking at the outcome (death), LTA patients had a lower survival time and higher frequency of deaths than SLKT patients.

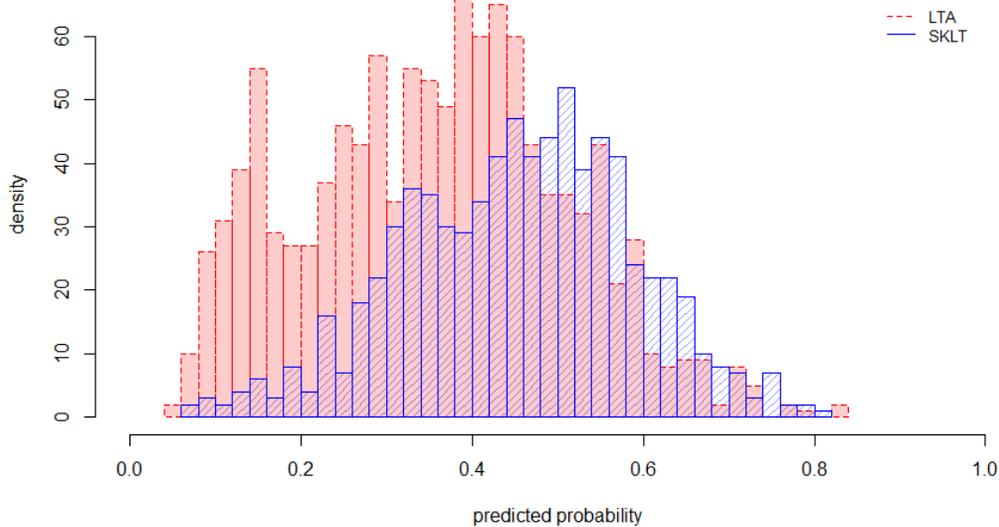
**Table 1 Summary statistics by transplant type**

		<b>Total (n=1930)</b>	<b>LTA (n=1165)</b>	<b>SLKT (n=765)</b>
<b>Diabetes – n (%)</b>	None	1276 (66.1)	824 (70.7)	452 (59.1)
	Type II	589 (30.5)	313 (26.9)	276 (36.1)
	Type I	65 (3.4)	28 (2.4)	37 (4.8)
<b>Region – n (%)</b>	1	85 (4.4)	52 (4.5)	33 (4.3)
	2	156 (8.1)	102 (8.8)	54 (7.1)
	3	234 (12.1)	138 (11.8)	96 (12.5)
	4	174 (9.0)	101 (8.7)	73 (9.5)
	5	494 (25.6)	353 (30.3)	141 (18.4)
	6	43 (2.2)	26 (2.2)	17 (2.2)
	7	315 (16.3)	147 (12.6)	168 (22.0)
	8	96 (5.0)	52 (4.5)	44 (5.8)
	9	78 (4.0)	54 (4.6)	24 (3.1)
	10	160 (8.3)	92 (7.9)	68 (8.9)
	11	95 (4.9)	48 (4.1)	47 (6.1)
<b>Race – n (%)</b>	Other	1446 (74.9)	871 (74.8)	575 (75.2)
	Hispanic/Latino	403 (20.9)	253 (21.7)	150 (19.6)
	Black	81 (4.2)	41 (3.5)	40 (5.2)
<b>Gender – n (%)</b>	Female	690 (35.8)	426 (36.6)	264 (34.5)
	Male	1240 (64.2)	739 (63.4)	501 (65.5)
<b>Etiology – n (%)</b>	ALD	1161 (60.2)	706 (60.6)	455 (59.5)
	NAFLD	769 (39.8)	459 (39.4)	310 (40.5)
<b>Portal vein thrombosis – n(%)</b>		190 (9.8)	117 (10.0)	73 (9.5)
<b>On ventilator – n (%)</b>		340 (17.6)	279 (23.9)	61 (8.0)
<b>BMI – mean (sd)</b>		30.6 (6.6)	31.0 (6.7)	29.9 (6.4)
<b>median (min, max)</b>		30.3(10.7, 63)	30.8 (10.7, 63)	29.5(14.9,53.5)
<b>Age – mean (sd)</b>		54.8 (9.5)	54.2 (9.5)	55.8 (9.4)
<b>median (min, max)</b>		56 (24, 81)	56 (24, 81)	57 (26, 78)
<b>Survival (years) – mean (sd)</b>		4.3 (2.9)	4.0 (2.8)	4.7 (3.0)
<b>median (min, max)</b>		4.0 (0.0, 11.4)	3.9 (0.0, 11.4)	4.1 (0.0, 11.2)
<b>Death – n (%)</b>		525 (27.2)	318 (27.3)	207 (27.1)

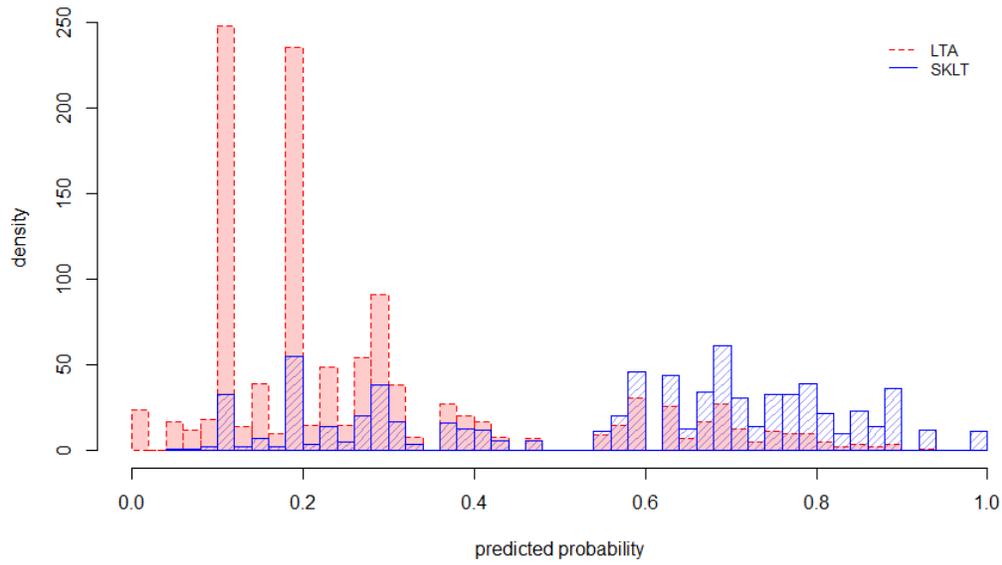
### 3.2 Propensity Score Distributions

Figures 2 through 7 show the overlap in pre-pseudo randomized estimated PS distributions for each of the six treatment assignment mechanisms. Enough overlap in the SLKT and LTA distributions was seen in all the treatment assignment models to be able to obtain precise estimates of the treatment effect and confirm balance in the covariates.

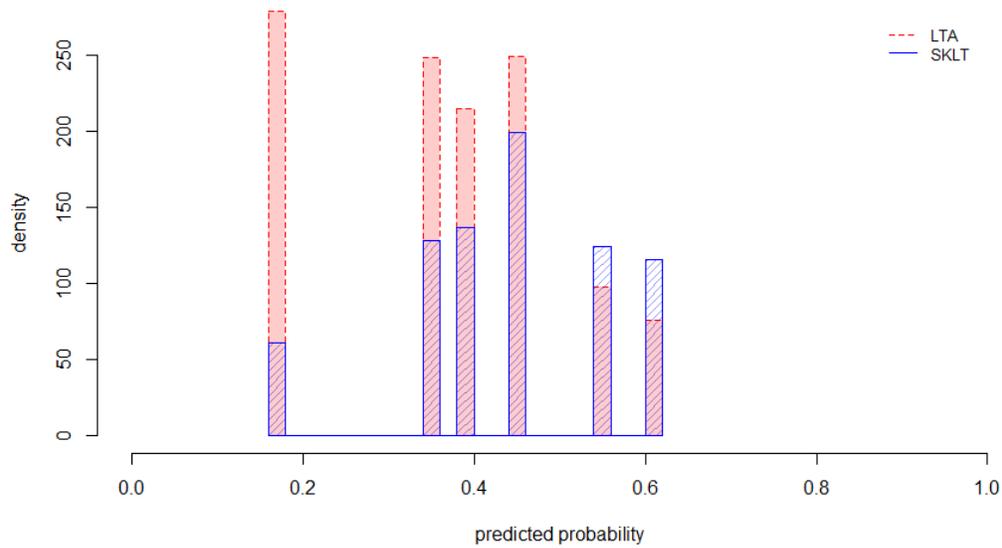
The logistic regression and random forest models appeared to have the strongest overlap in propensity scores, while the tree models – particularly the unpruned tree – seemed to have less overlap for the higher SLKT values. Since SLKT was the treatment of interest, it was more important in this analysis to have higher coverage for the SLKT propensities compared to LTA. Additionally, there was a larger proportion of LTA observations in the original sample, so the lesser overlap for the LTA propensities was expected.



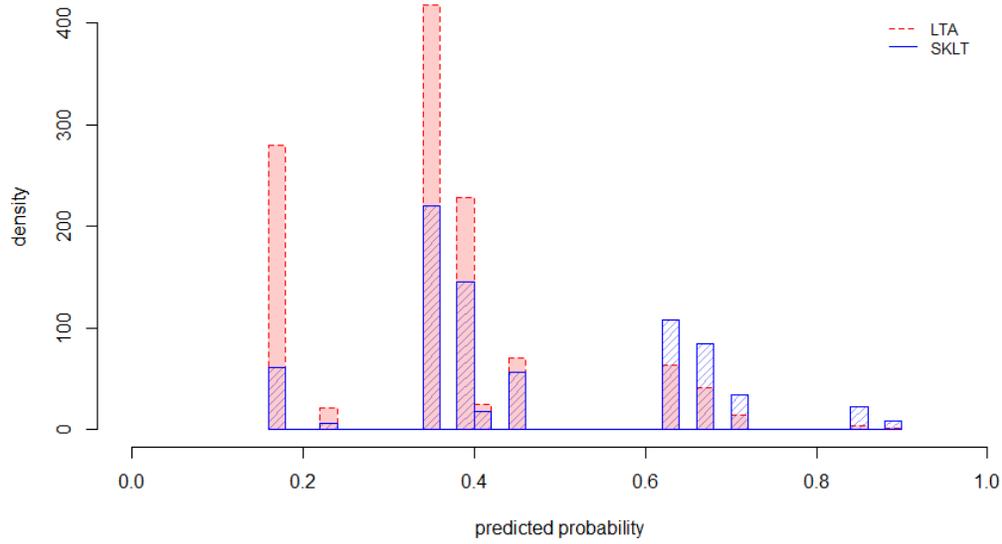
**Figure 2 Histogram of pre-pseudo randomized propensities (logistic regression)**



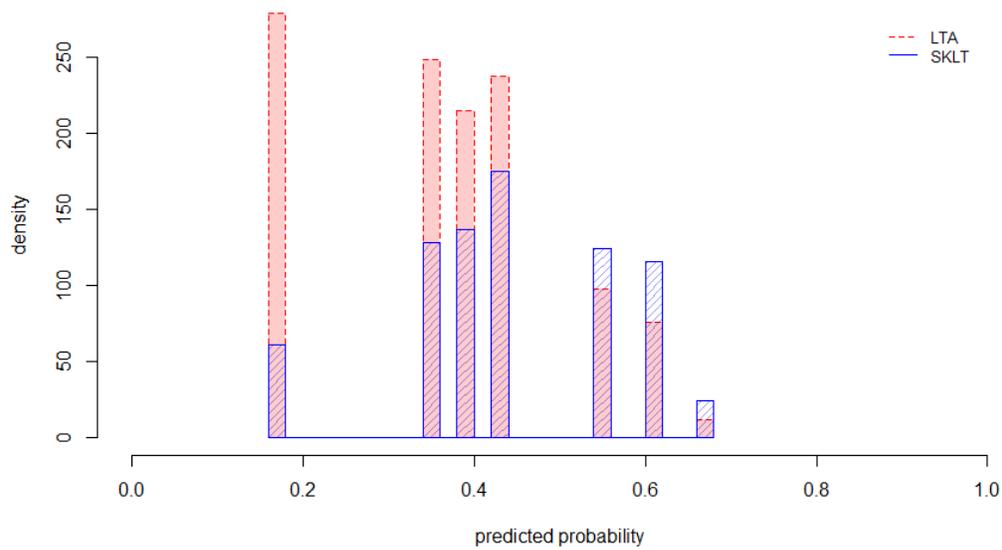
**Figure 3 Histogram of pre-pseudo randomized propensities (classification tree,  $cp=0.0$ )**



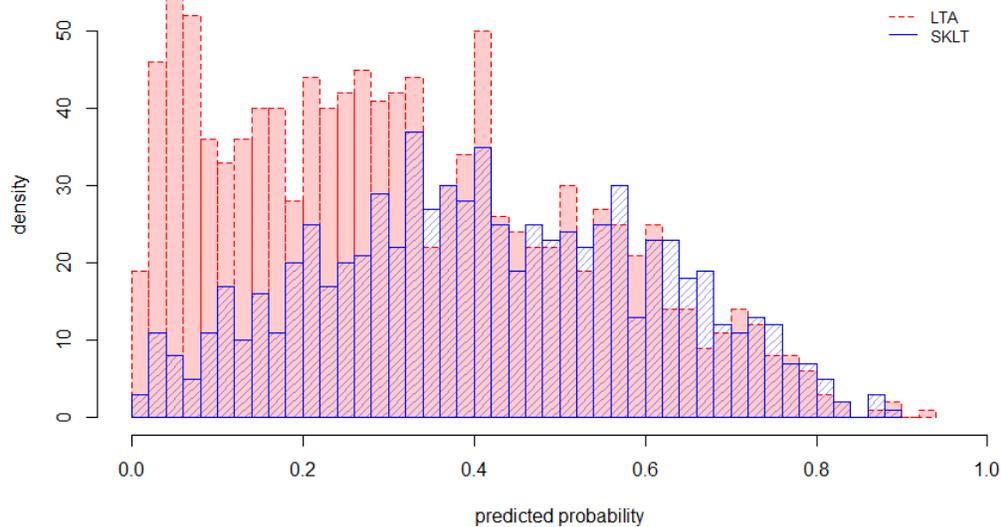
**Figure 4 Histogram of pre-pseudo randomized propensities (classification tree,  $cp=0.0157$ )**



**Figure 5 Histogram of pre-pseudo randomized propensities (classification tree, cp=0.0)**



**Figure 6 Histogram of pre-pseudo randomized propensities (classification tree, cp=0.01)**



**Figure 7 Histogram of pre-pseudo randomized propensities (random forest)**

### 3.3 Pseudo Populations

The matched pseudo populations only used the observations that were matched pairs. The number of events and matched sets for each treatment assignment model (out of 1165 LTA and 765 SLKT) are given below in Tables 2 and 3 respectively.

The random forest treatment assignment model used the most observations of all the matched pseudo populations, followed by the optimally matched classification tree ( $cp=0.0157$ ), then the tree pruned with a complexity parameter of 0.01. As such, the random forest pseudo population contained the most information of the six matching approaches. The stratified and IPTW pseudo populations used all 1930 observations from the original sample, and therefore contain more of the original sample’s information than any of the matched pseudo populations.

The number of events of interest for each matched pseudo population were tabulated to consider the validity of fitting outcomes models in section 3.5 to measure survival. Again, the random forest matched model contained the most information, with the highest number of events. The unpruned tree model had the second highest number of events, this time followed by the logistic regression matched model.

**Table 2 Number of Deaths in 1:1 Matched Pseudo Populations**

<b>Assignment</b>	<b>Total n=</b>	<b>No. of Events</b>
<b>Logistic Regression</b>	1314	370
<b>Tree (cp=0.0)</b>	912	246
<b>Tree (cp=0.0157)</b>	1398	375
<b>Tree (cp=0.0065)</b>	1258	334
<b>Tree (cp=0.01)</b>	1374	360
<b>Random Forest</b>	1454	412

**Table 3 Matched Observations by Treatment Assignment**

<b>Assignment</b>	<b>Total Obs.</b>	<b># Matched Sets</b>	<b>Unmatched LTA</b>	<b>Unmatched SLKT</b>
<b>Logistic Regression</b>	1930	657	508	108
<b>Tree (cp = 0.0)</b>	1930	456	709	309
<b>Tree (cp = 0.0157)</b>	1930	699	466	66
<b>Tree (cp = 0.0065)</b>	1930	629	536	136
<b>Tree (cp = 0.01)</b>	1930	687	478	78
<b>Random Forest</b>	1930	727	438	38

### 3.4 Covariate Balance

The covariate balance results for all pseudo population approaches, for all assignment mechanism models, are given in the forest plots (Figures 8 through 25) below. The open circle

represents the standardized mean difference for the original sample, and the black circle represents the standardized mean difference for the pseudo population. The broken vertical line at 0.1 (a 10% difference in covariate differences for the two treatment groups) on the x-axis represents the threshold for acceptable standardized differences; covariates with values to the left of this line and closer to zero can be considered well-balanced.

Pseudo populations created from the logistic regression models were notably better at balancing covariates, compared to the relatively poorly balanced random forest pseudo populations. The tree models appeared to perform the most poorly, with many of the covariates' standardized differences failing to even reach the 0.1 threshold. In terms of pseudo population approach, the 1:1 matched pseudo populations appeared to be the most successful, followed by the IPTW approach. The stratified pseudo populations (particularly the ones created from tree models) performed the poorest.

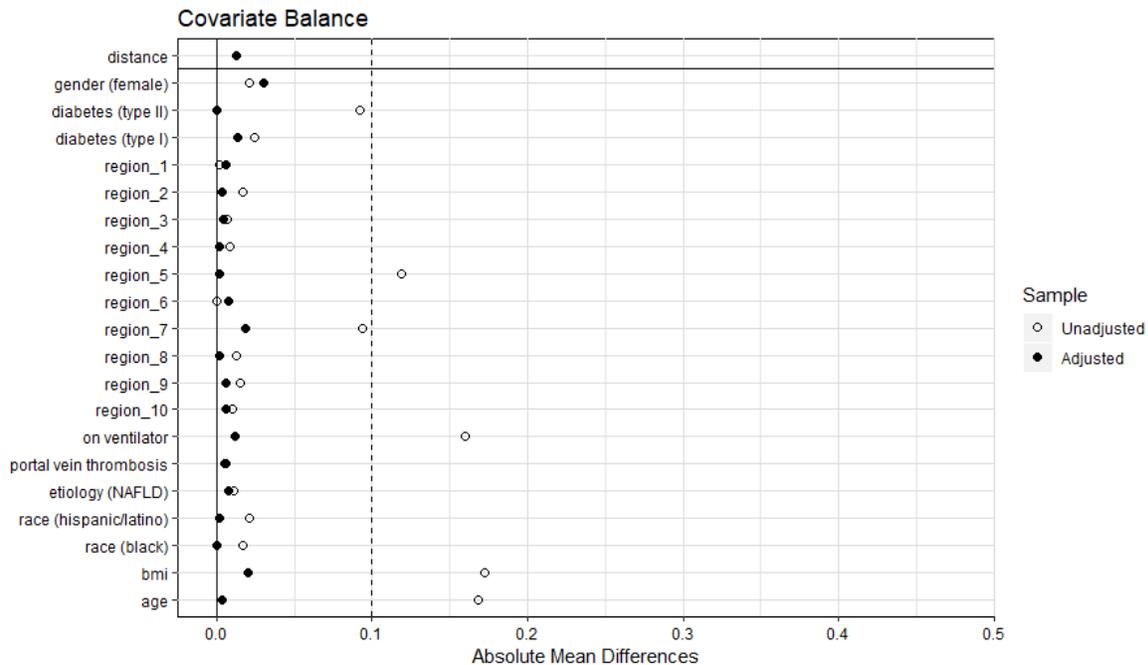
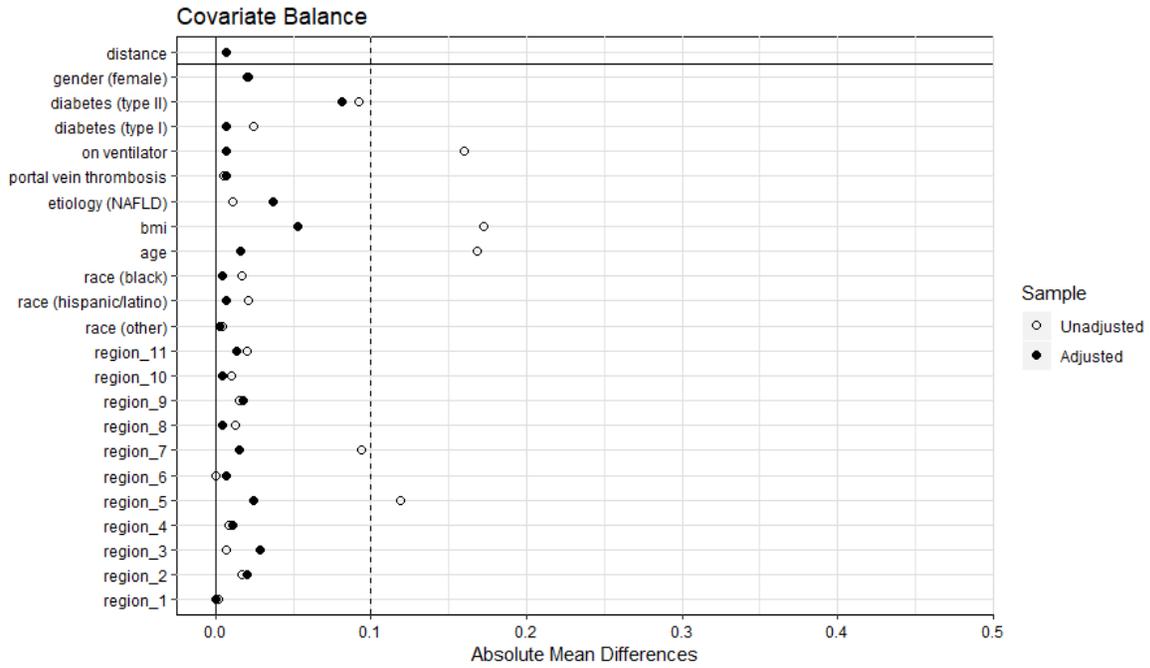
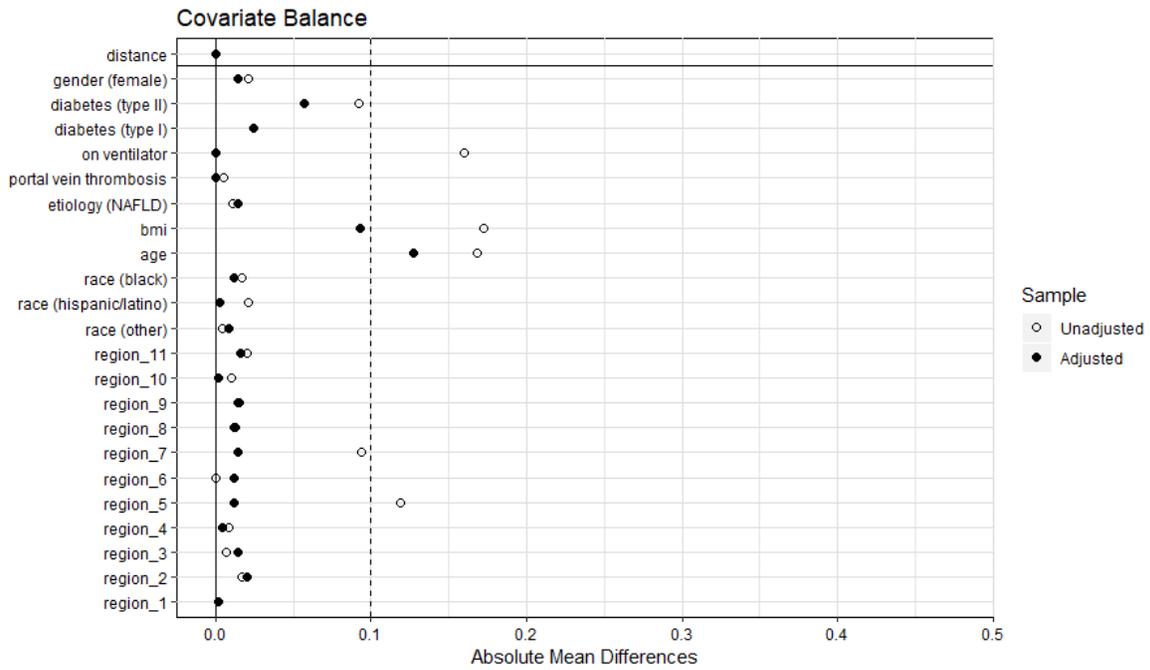


Figure 8 Balance Plot for 1:1 Matching, Logistic Regression



**Figure 9 Balance Plot for 1:1 Matching, Classification Tree (cp=0.0)**



**Figure 10 Balance Plot for 1:1 Matching, Classification Tree (cp=0.0157)**

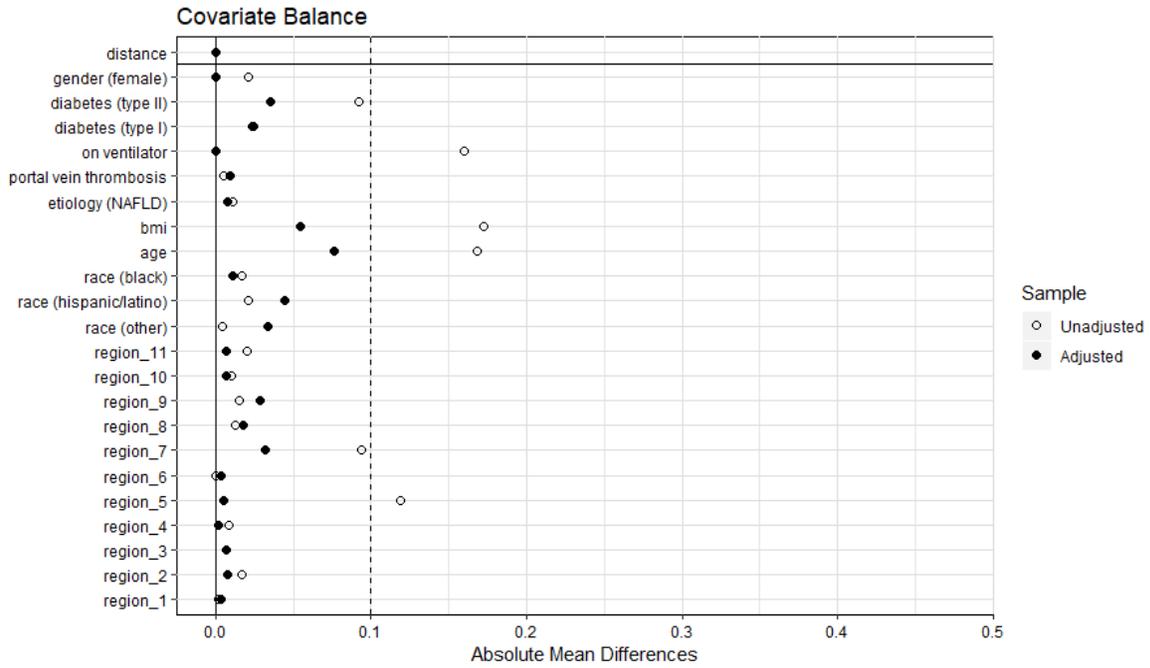


Figure 11 Balance Plot for 1:1 Matching, Classification Tree (cp=0.0065)

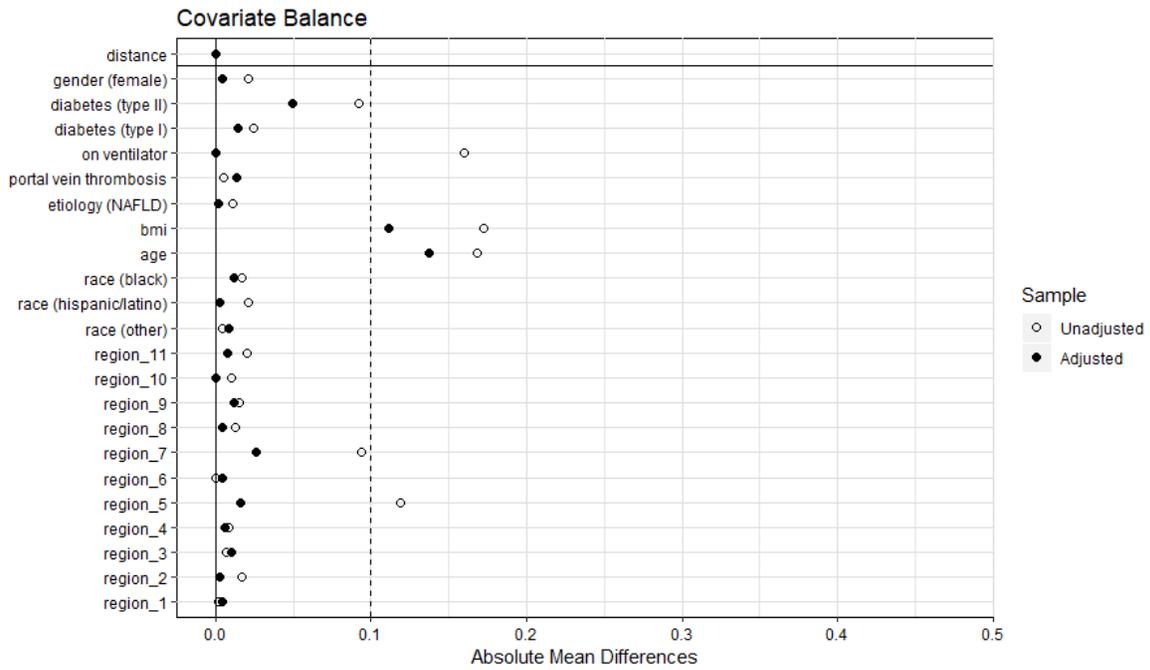
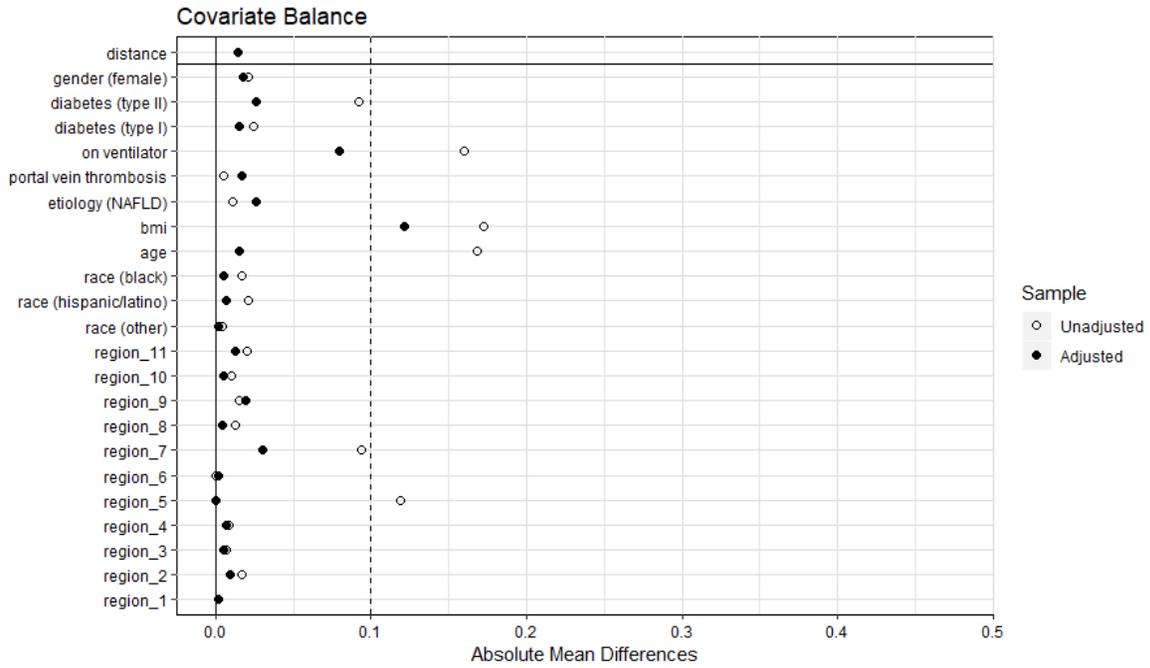
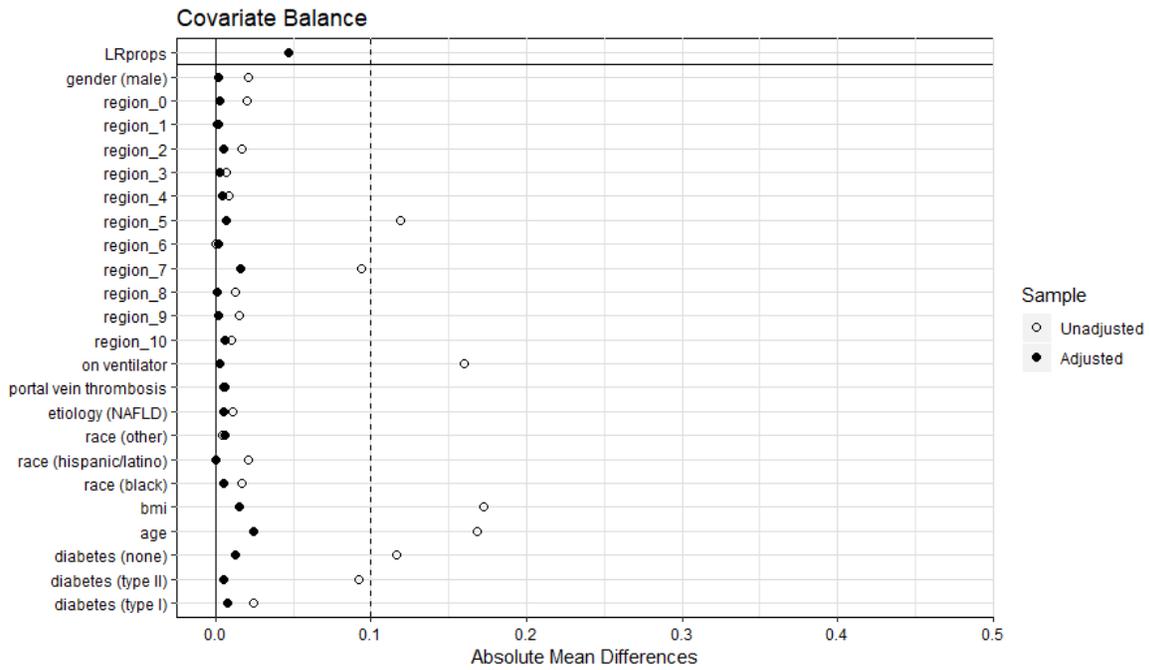


Figure 12 Balance Plot for 1:1 Matching, Classification Tree (cp=0.01)



**Figure 13 Balance Plot for 1:1 Matching, Random Forest**



**Figure 14 Balance Plot for Stratification (5 strata), Logistic Regression**

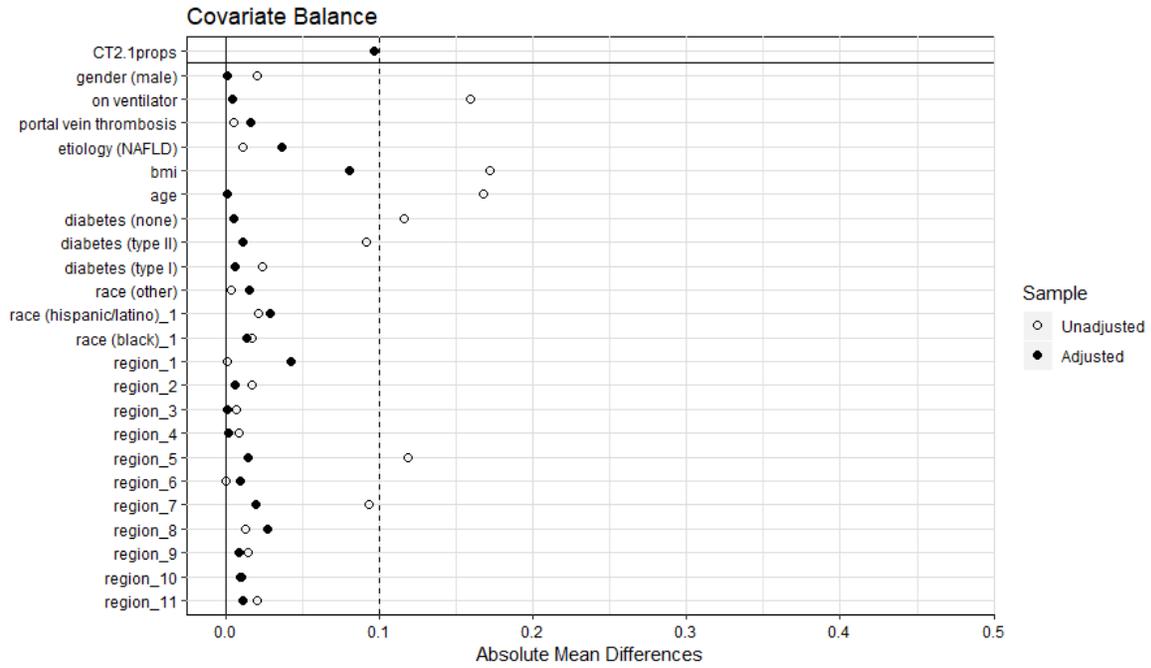


Figure 15 Balance Plot for Stratification (5 strata), Classification Tree (cp=0.0)

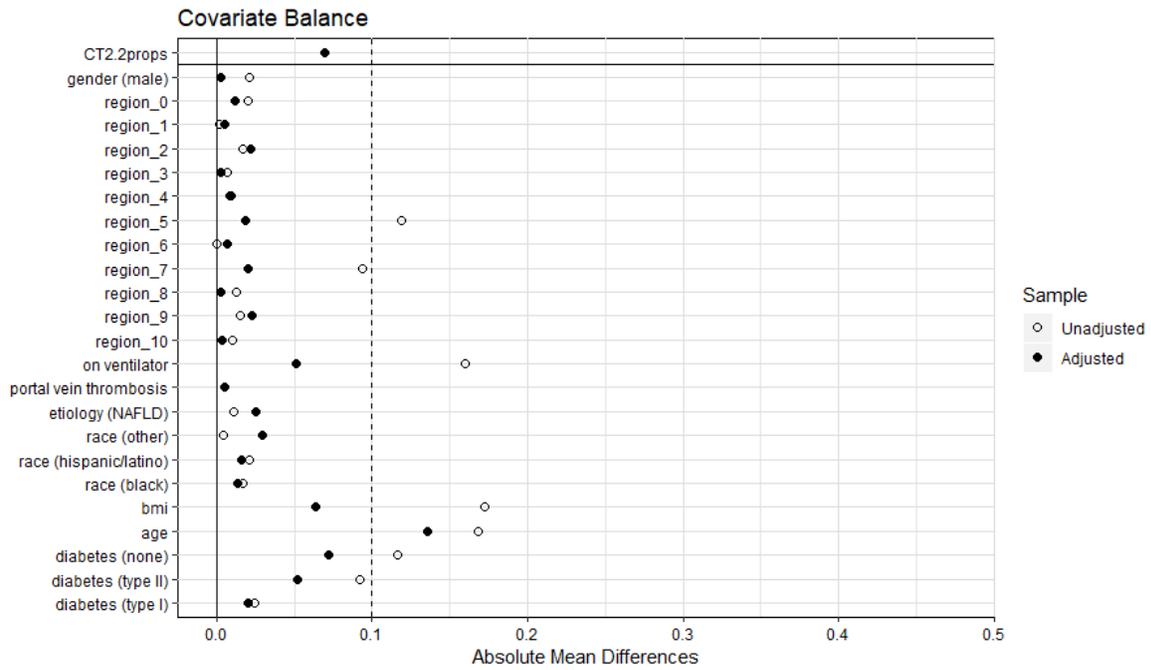
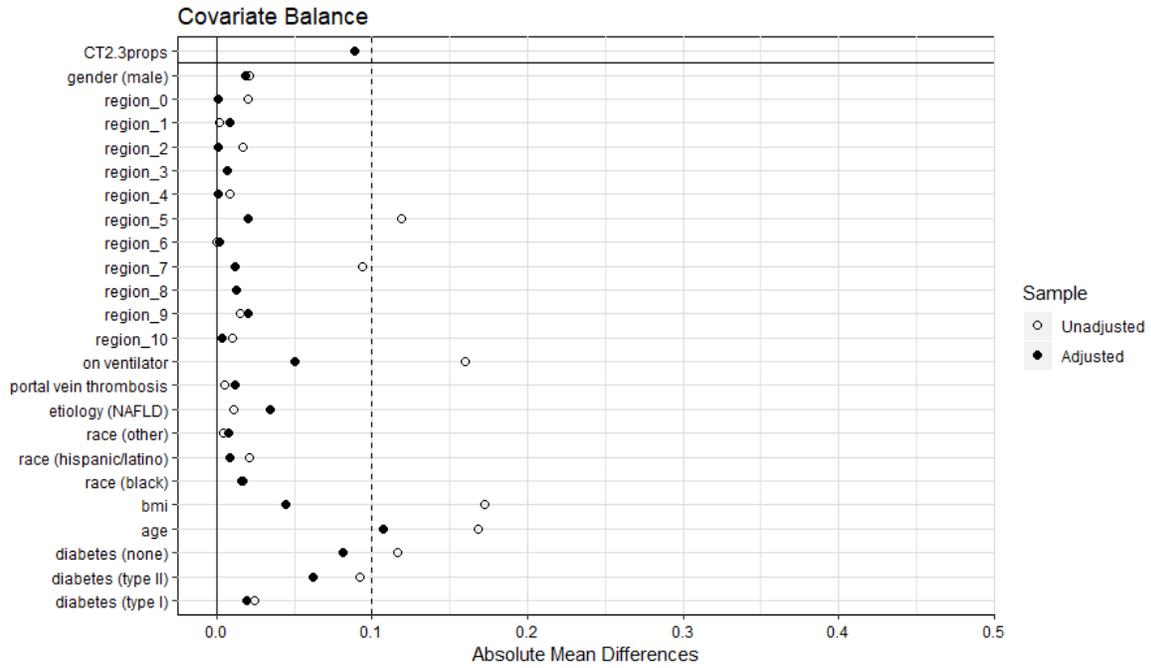
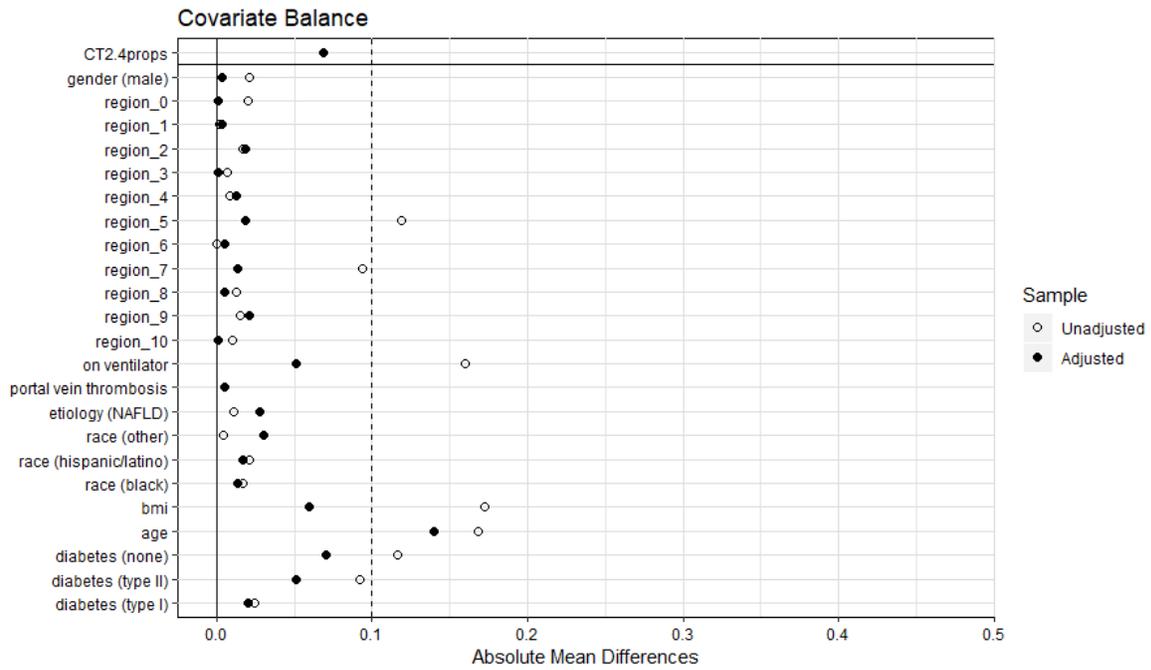


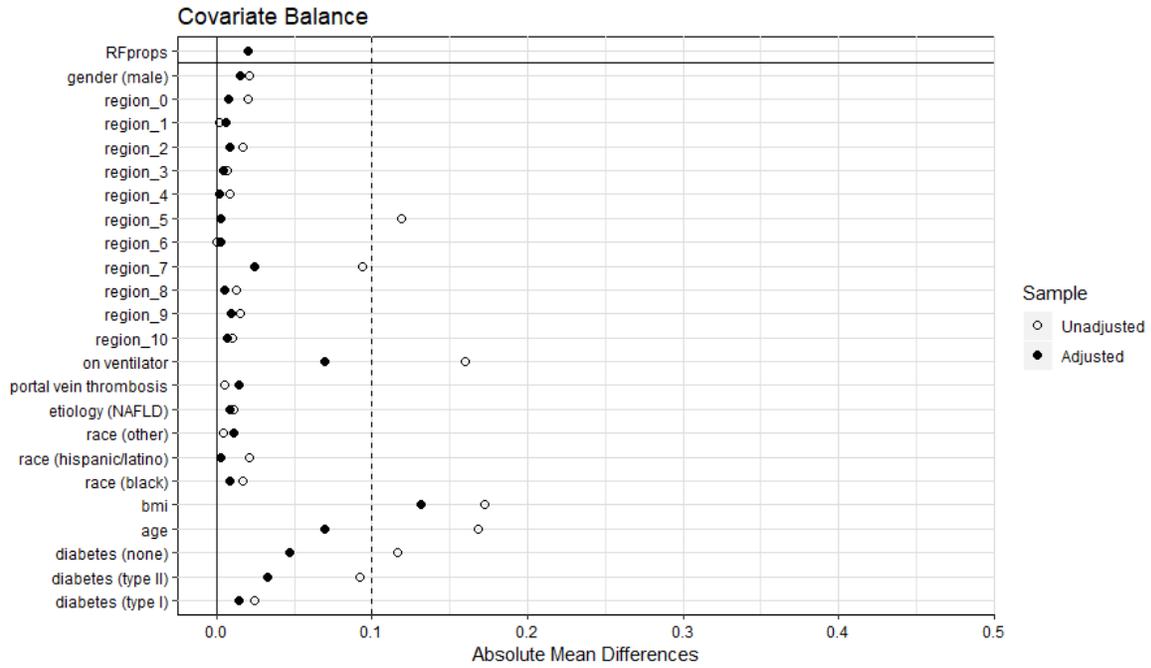
Figure 16 Balance Plot for Stratification (5 strata), Classification Tree (cp=0.0157)



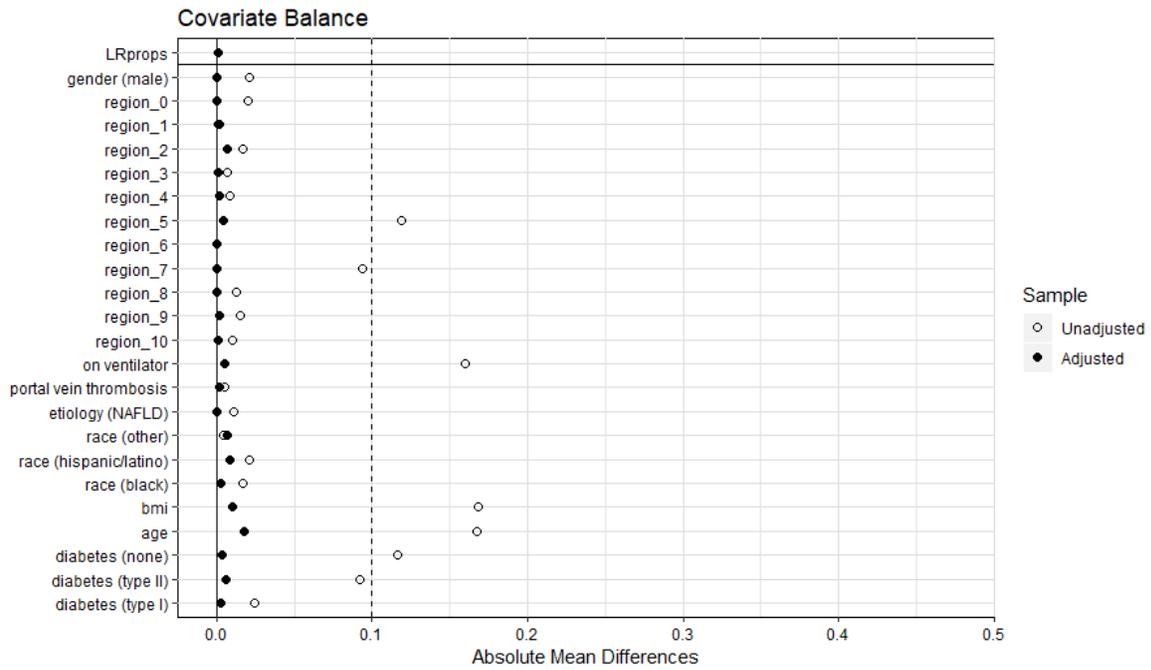
**Figure 17 Balance Plot for Stratification (5 strata), Classification Tree (cp=0.0065)**



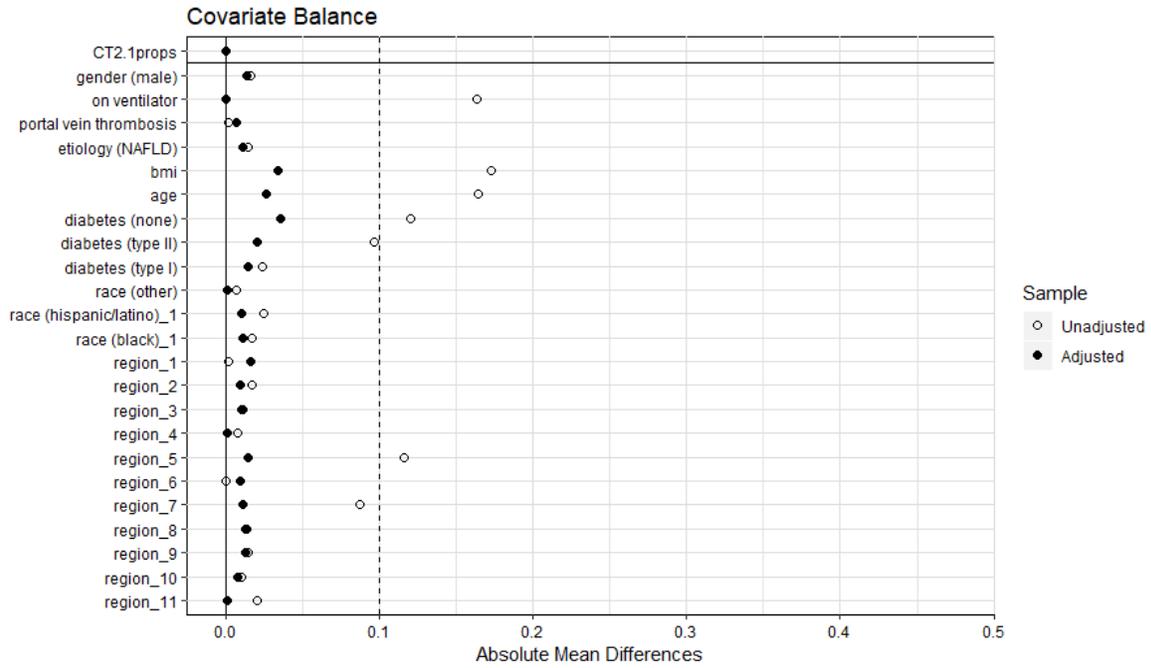
**Figure 18 Balance Plot for Stratification (5 strata), Classification Tree (cp=0.01)**



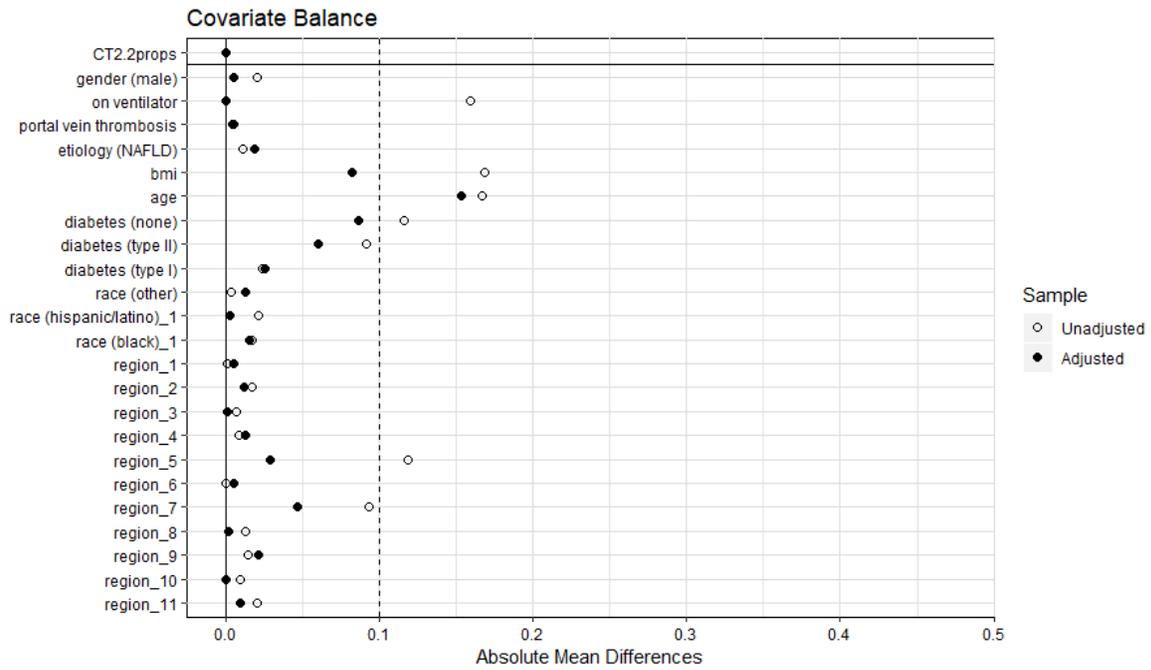
**Figure 19 Balance Plot for Stratification (5 strata), Random Forest**



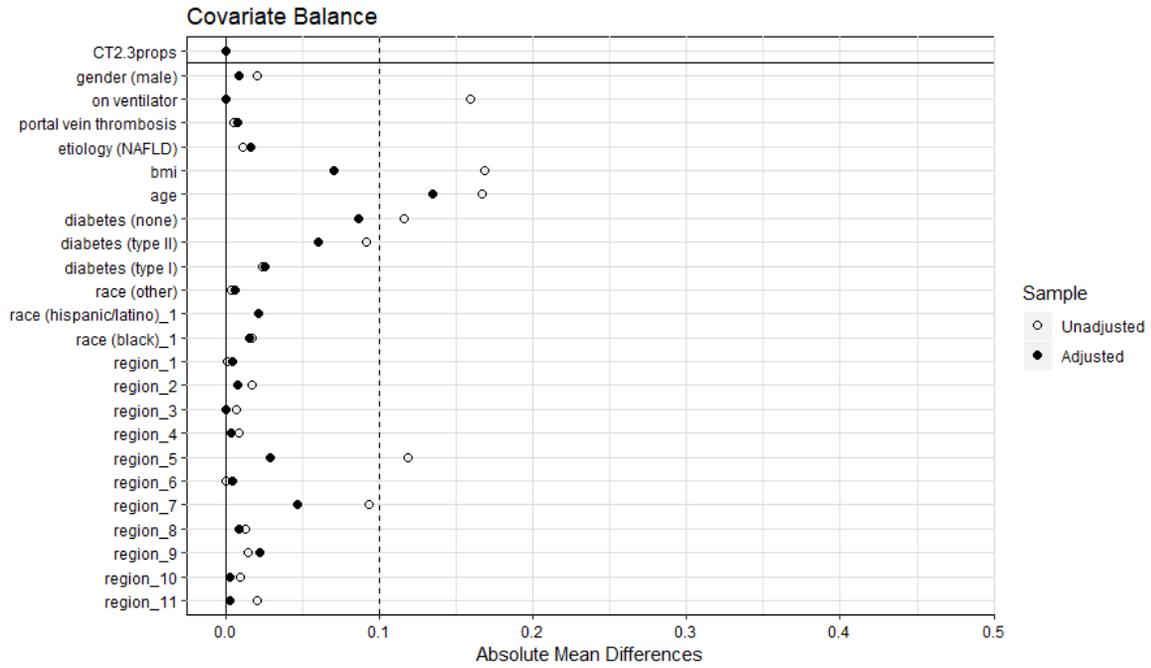
**Figure 20 Balance Plot for IPTW, Logistic Regression**



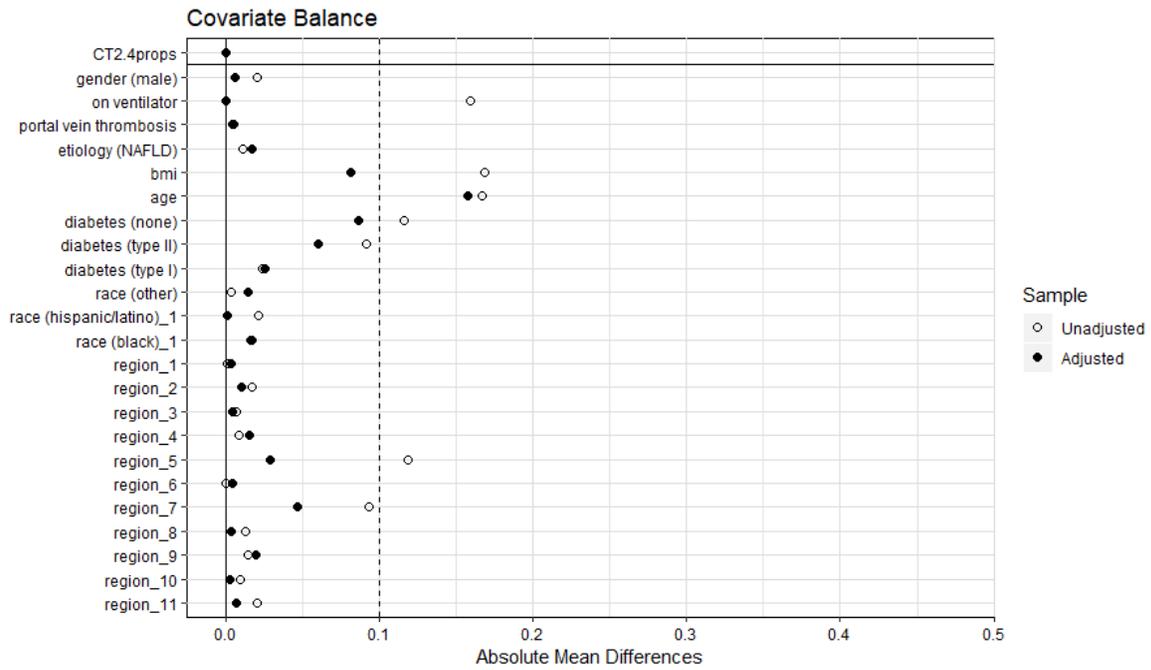
**Figure 21 Balance Plot for IPTW, Classification Tree (cp=0.0)**



**Figure 22 Balance Plot for IPTW, Classification Tree (cp=0.0157)**



**Figure 23 Balance Plot for IPTW, Classification Tree (cp=0.0065)**



**Figure 24 Balance Plot for IPTW, Classification Tree (cp=0.01)**

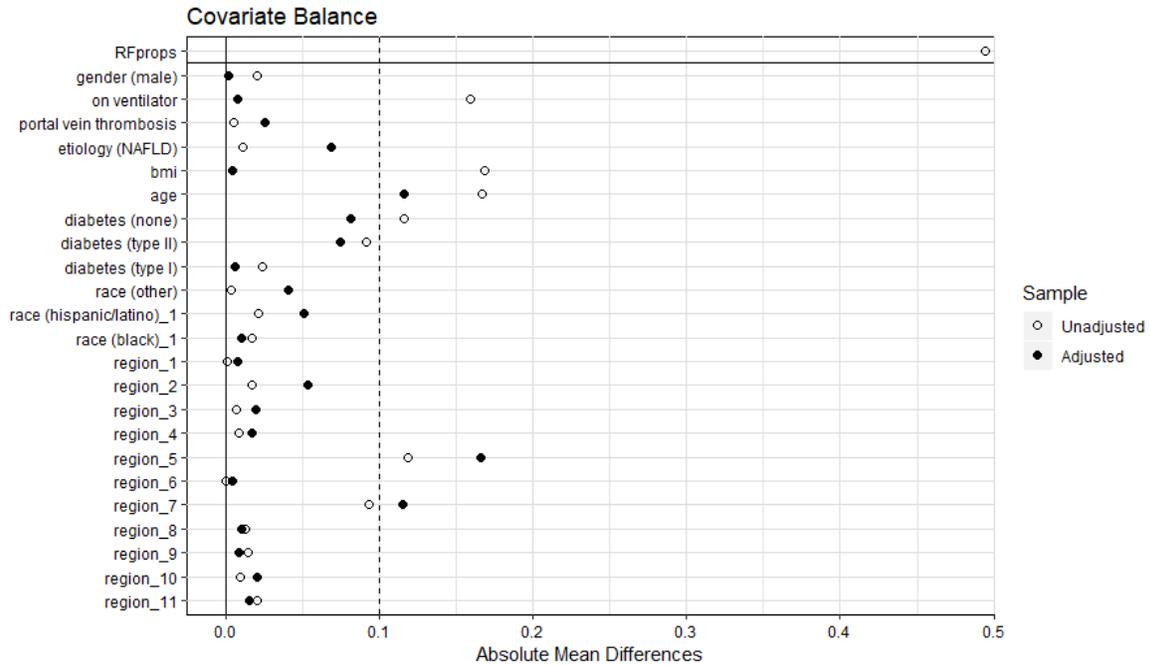


Figure 25 Balance Plot for IPTW, Random Forest

### 3.5 Outcomes Model

Tables 4 through 6 show the hazard ratio estimates from the outcomes models for twelve pseudo populations created from the three treatment assignment mechanism models. Survival curves, for each of these pseudo populations are shown with Kaplan-Meier curves in Figures 26 through 43.

While all the estimated hazard ratios were less than or at 1, different methods for treatment assignment and pseudo population generation gave different estimates. Notably, the matched pseudo populations gave a significant (p-value = 0.029) hazard ratio of 0.80 with a confidence interval of (0.655, 0.977), while the stratified pseudo populations had no significant results, even

for the logistic and/or forest models. Lastly, the IPTW pseudo population had significant estimates for both the unpruned tree and the random forest models, with a hazard ratio of 0.80 (p-value = 0.048) and confidence interval of (0.643, 0.999) for the unpruned tree, and a hazard ratio of 0.645 (p-value = 0.0011) and confidence interval of (0.495, 0.839) for the random forest. While the standard errors were relatively low for all outcomes models, the logistic regression appeared to have the highest error values.

The hazard ratio estimates fit the implications of the propensity score distributions given in Section 3.2 in terms of accurate estimates. The logistic regression and random forest treatment assignment models appeared to have the most overlap in propensities between the SLKT and LTA groups, and the random forest gave significant results in two of the three pseudo population approaches, while the logistic model gave significant results in one of the three. Meanwhile, the tree models only gave a borderline significant result using the IPTW approach.

**Table 4 Survival Analysis – 1:1 Matching**

Assignment		Events	HR	SE	p-value	95% CI
Logistic		370	0.9017	0.1042	0.313	(0.737, 1.103)
Classification Tree	CP=0	246	1.0200	0.1280	0.873	(0.801, 1.299)
	CP=0.0157	375	0.8964	0.1035	0.293	(0.731, 1.099)
	CP=0.0065	334	1.0650	0.1100	0.575	(0.854, 1.328)
	CP=0.01	360	0.9780	0.1058	0.827	(0.801, 1.195)
Random Forest		412	0.7999	0.0990	0.0285	(0.655, 0.977)

**Table 5 Survival Analysis – Stratification**

Assignment		Events	HR	SE	p-value	95% CI
Logistic		525	0.8747	0.0947	0.157	(0.727, 1.053)
Classification Tree	CP=0	525	0.8188	0.1056	0.0585	(0.666, 1.007)
	CP=0.0157	525	0.8828	0.0924	0.177	(0.737, 1.058)
	CP=0.0065	525	0.9116	0.0940	0.325	(0.758, 1.096)
	CP=0.01	525	0.8876	0.0926	0.198	(0.740, 1.064)
Random Forest		525	0.8639	0.0918	0.111	(0.722, 1.034)

**Table 6 Survival Analysis – IPTW**

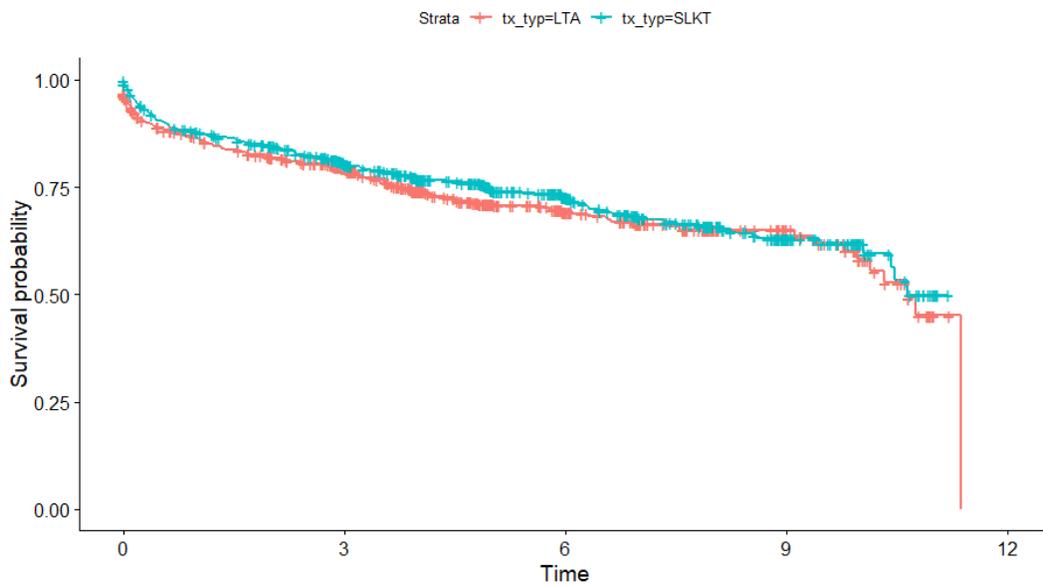
Assignment		Events	HR	SE	p-value	95% CI
Logistic		525	0.8572	0.0975	0.114	(0.708, 1.038)
Classification Tree	CP=0	525	0.8015	0.1121	0.0484	(0.643, 0.999)
	CP=0.0157	525	0.9050	0.0939	0.288	(0.753, 1.088)
	CP=0.0065	525	0.9327	0.0957	0.467	(0.773, 1.125)
	CP=0.01	525	0.9095	0.0942	0.314	(0.756, 1.094)
Random Forest		525	0.6447	0.1345	0.0011	(0.495, 0.839)

Figures 26 through 31 show the Kaplan-Meier (KM) survival curves for the 1:1 matched pseudo populations, separated by treatment group. The logistic regression treatment assignment model was the only significant matched pseudo population, and the KM curves imply that the proportional hazards assumption regarding the hazard ratios being constant over time was upheld.

Figures 32 through 37 show the KM curves for the stratified pseudo populations, with each figure separating the SLKT and LTA curves by strata. None of these pseudo populations had significant results for the Cox models, a fact that is reinforced by the KM graphs. Finally, Figures 38 through 43 present the KM curves for the IPTW pseudo populations, and show similar results

to the Cox models: the random forest and unpruned tree IPTW pseudo populations are significantly different with respect to treatment group.

All nonparametric results from the KM curves provided evidence for their corresponding hazard ratio estimates from the Cox models, and reinforced the idea that treatment assignment, pseudo population approach, and outcomes model selection made a significant difference.



**Figure 26 Kaplan-Meier curves – Logistic Regression (1:1 matching)**

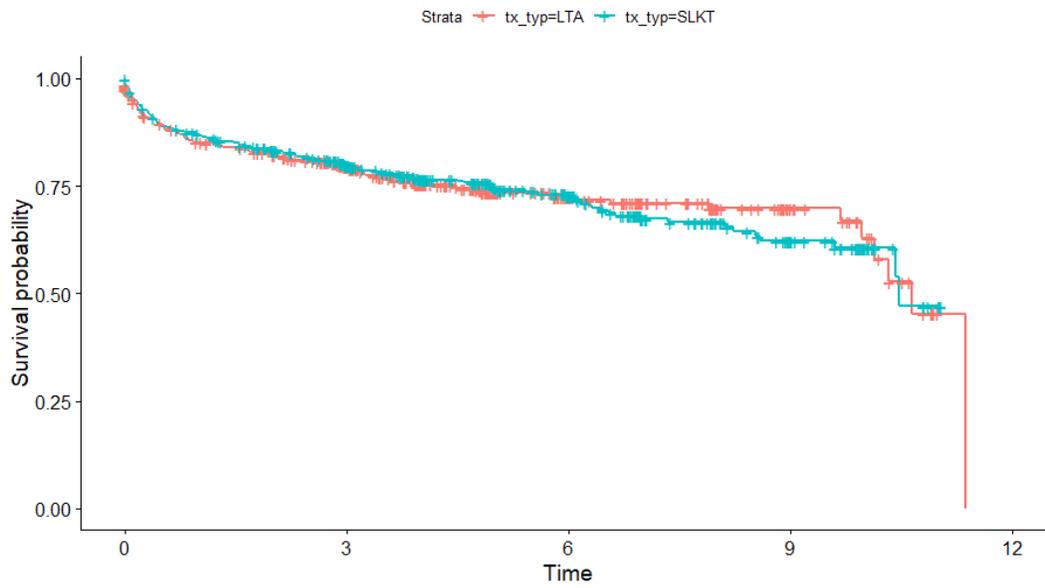


Figure 27 Kaplan-Meier curves – Classification Tree, cp=0.0 (1:1 matching)

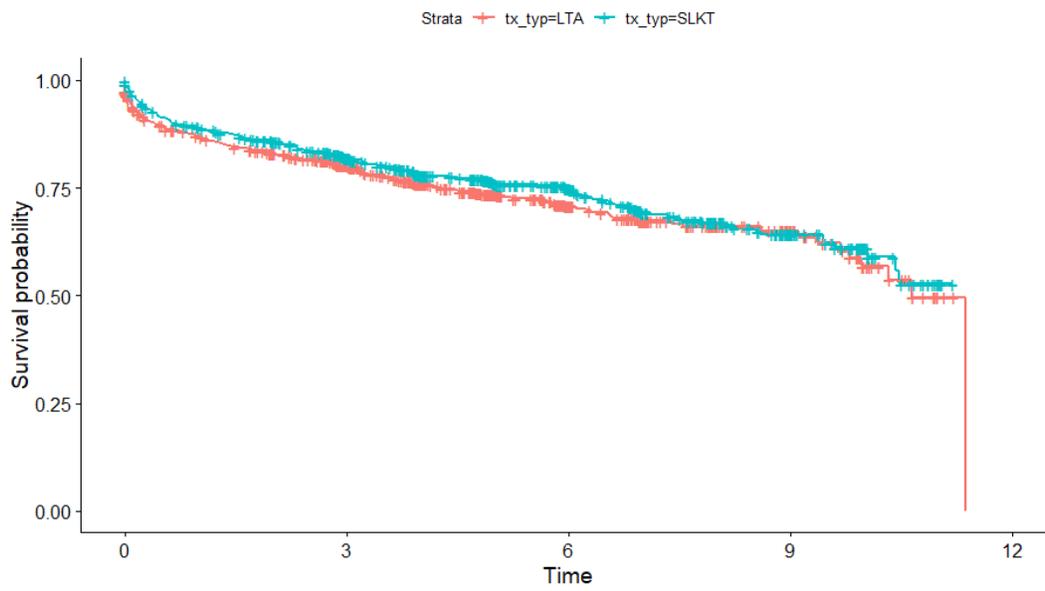


Figure 28 Kaplan-Meier curves – Classification Tree, cp=0.0157 (1:1 matching)

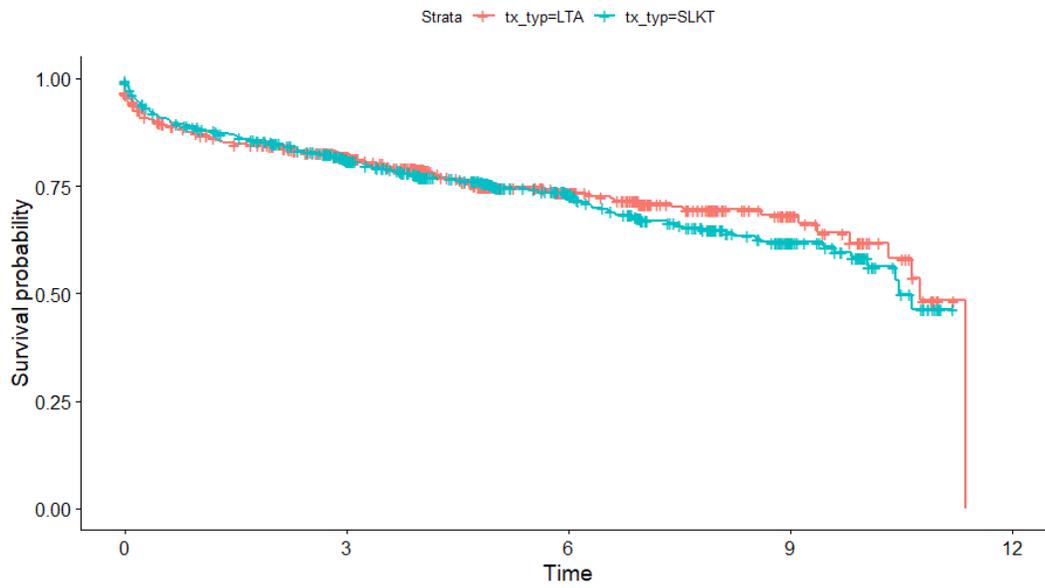


Figure 29 Kaplan-Meier curves – Classification Tree, cp=0.0065 (1:1 matching)

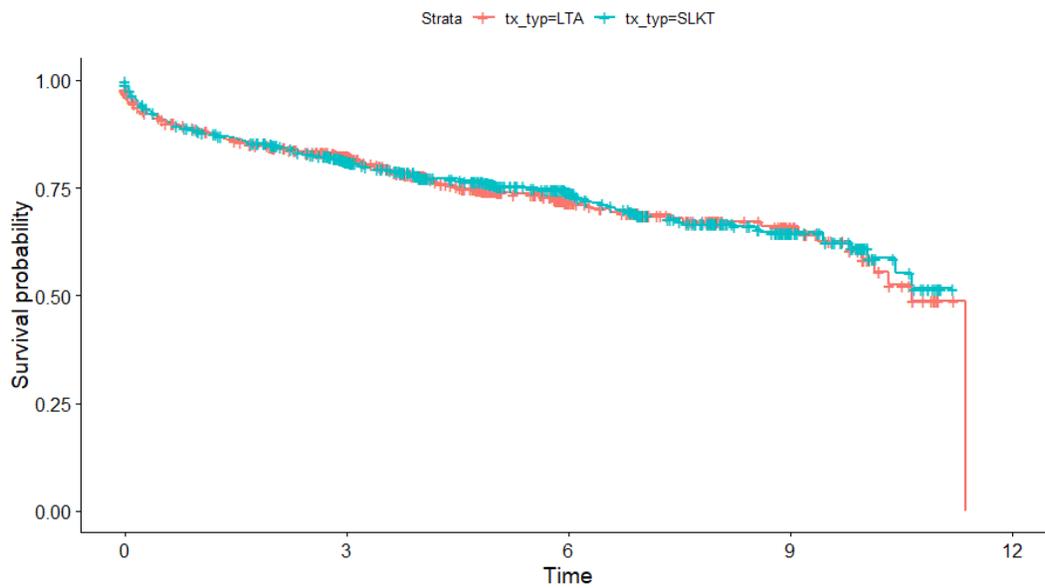


Figure 30 Kaplan-Meier curves – Classification Tree, cp=0.01 (1:1 matching)

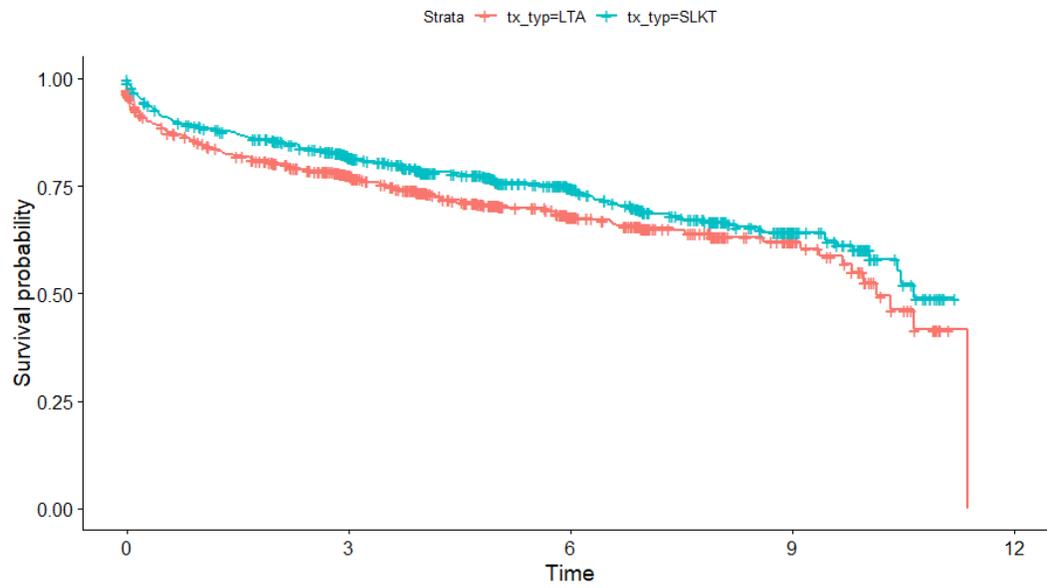


Figure 31 Kaplan-Meier curves – Random Forest (1:1 matching)

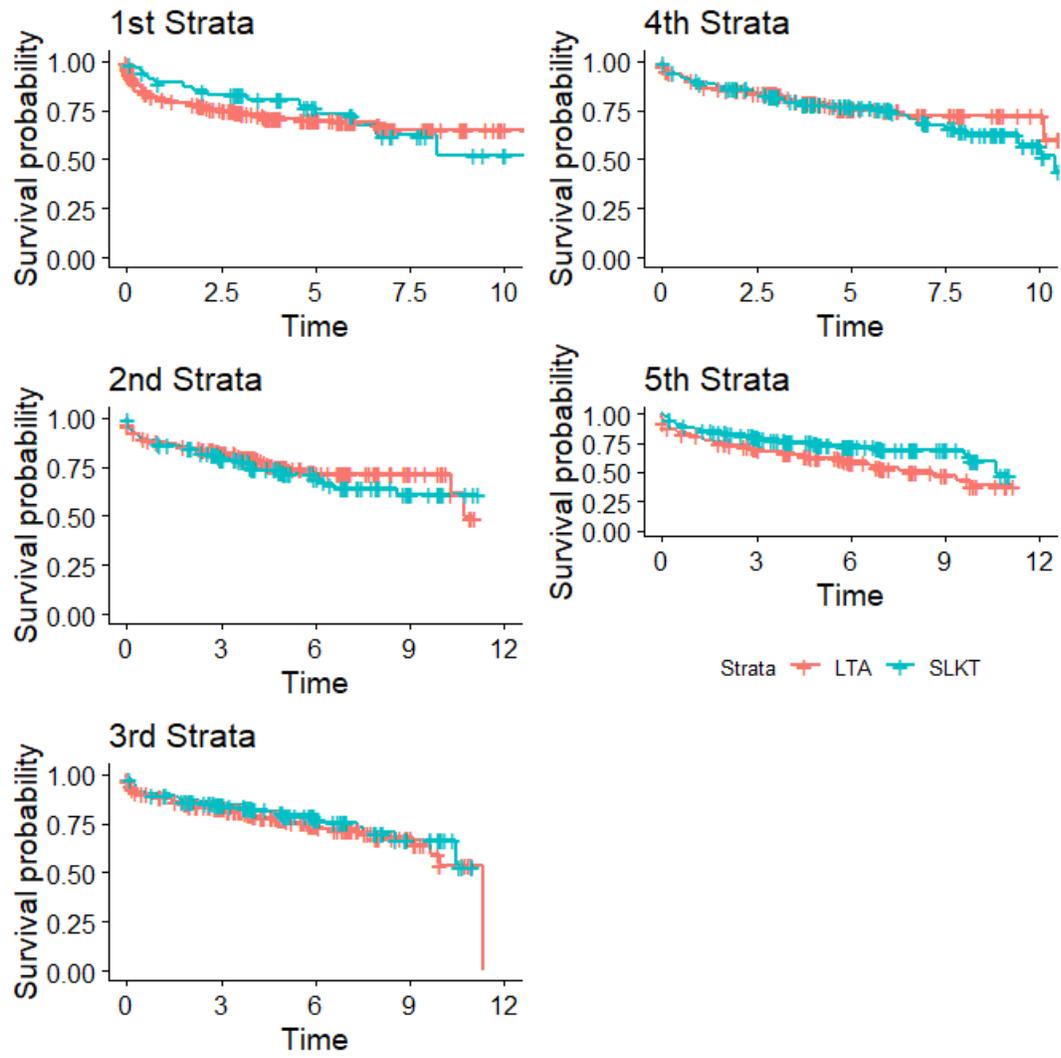


Figure 32 Kaplan-Meier curves – Logistic Regression (stratification)

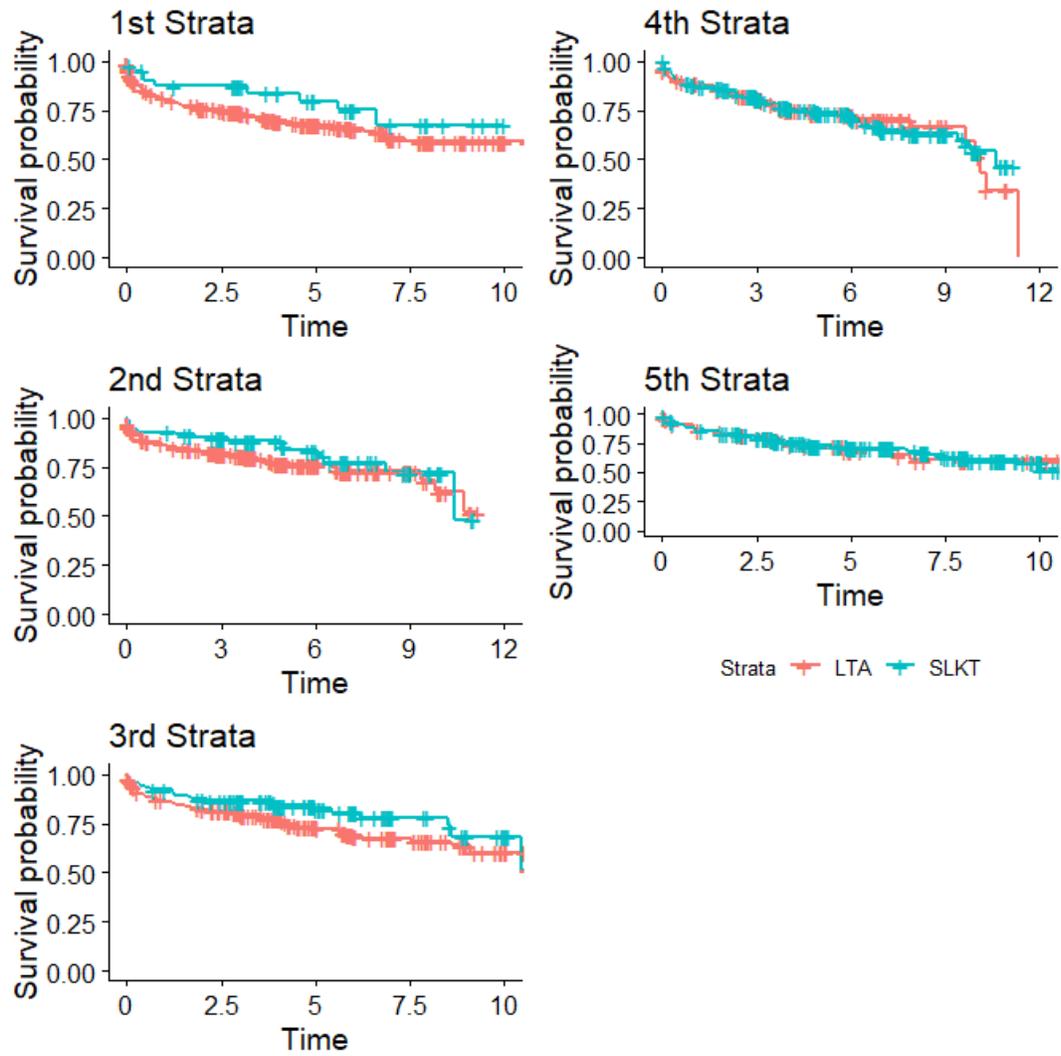


Figure 33 Kaplan-Meier curves – Classification Tree, cp=0.0 (stratification)

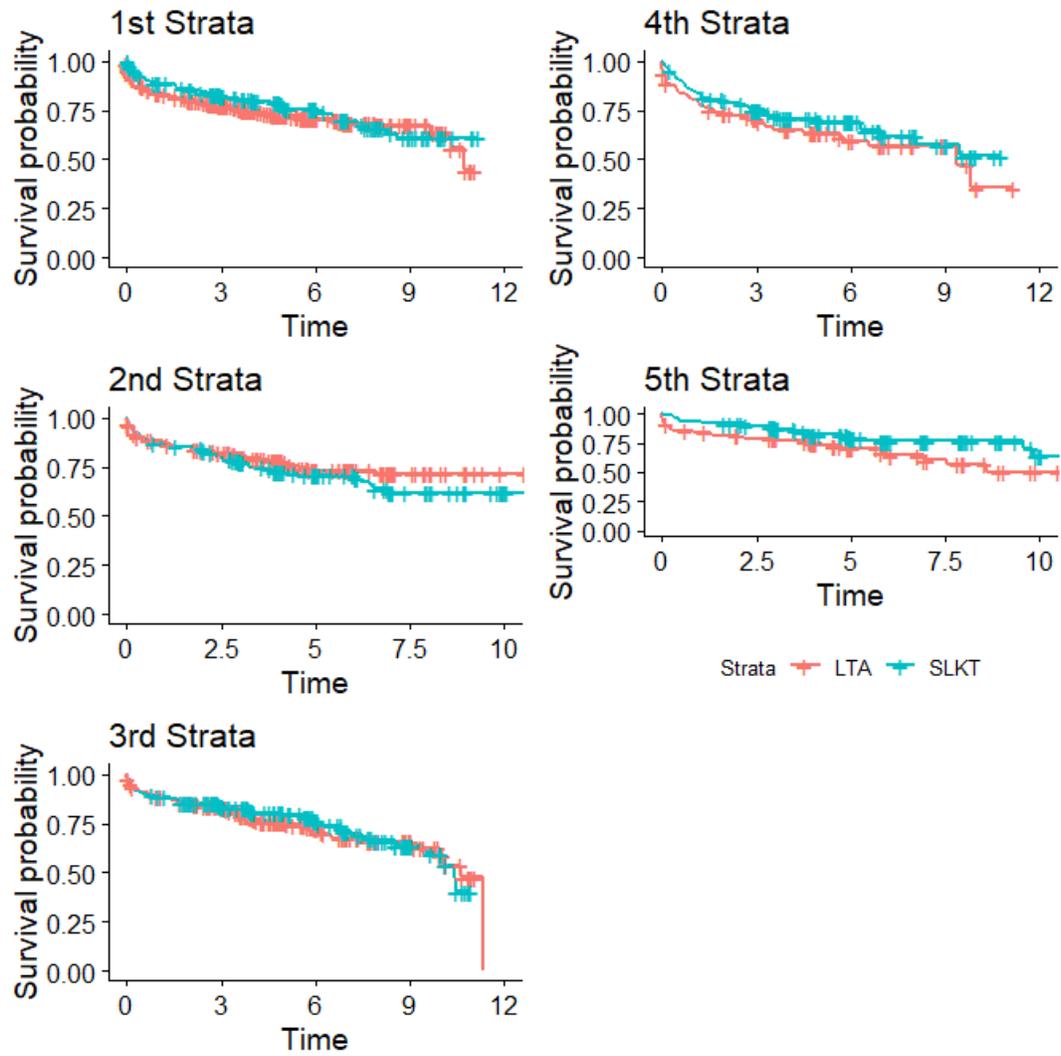


Figure 34 Kaplan-Meier curves – Classification Tree, cp=0.0157 (stratification)

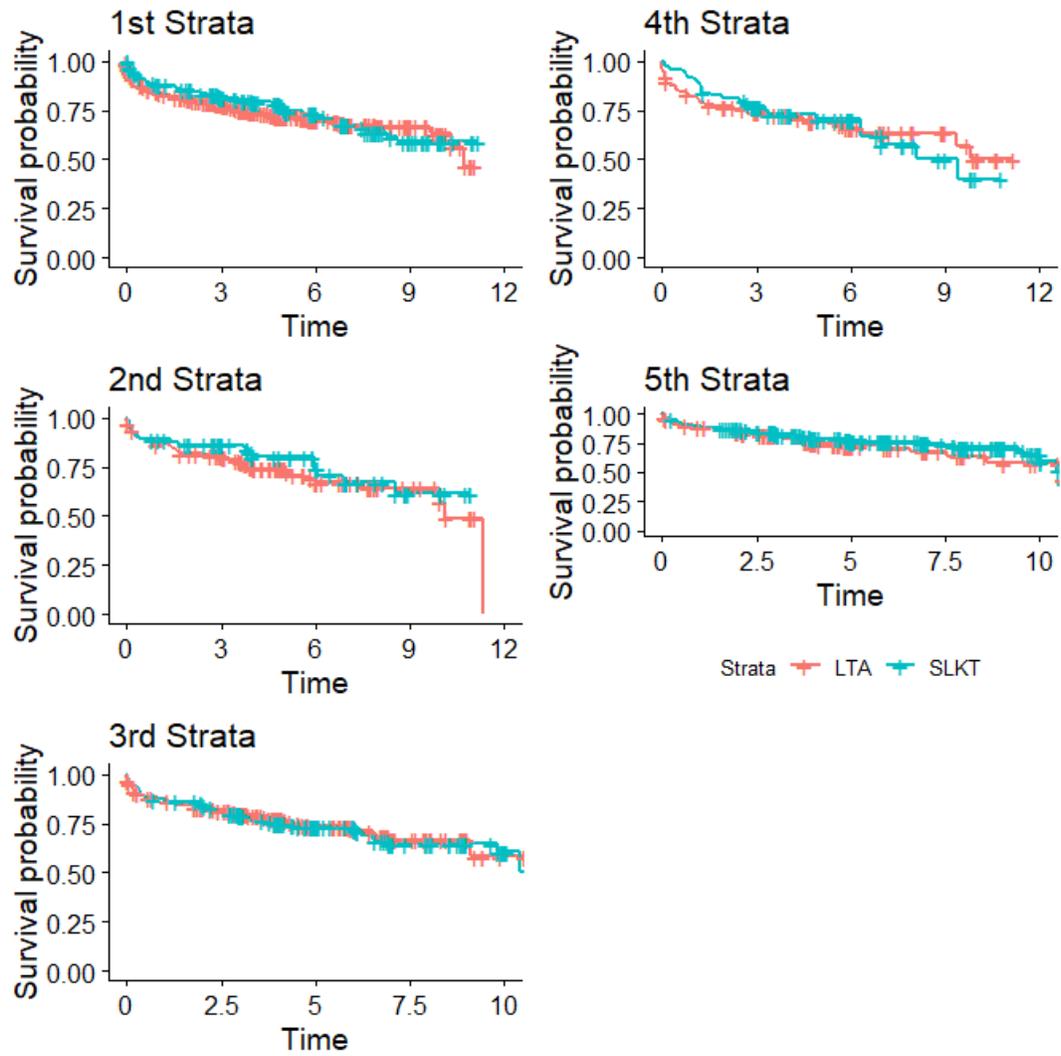


Figure 35 Kaplan-Meier curves – Classification Tree, cp=0.0065 (stratification)

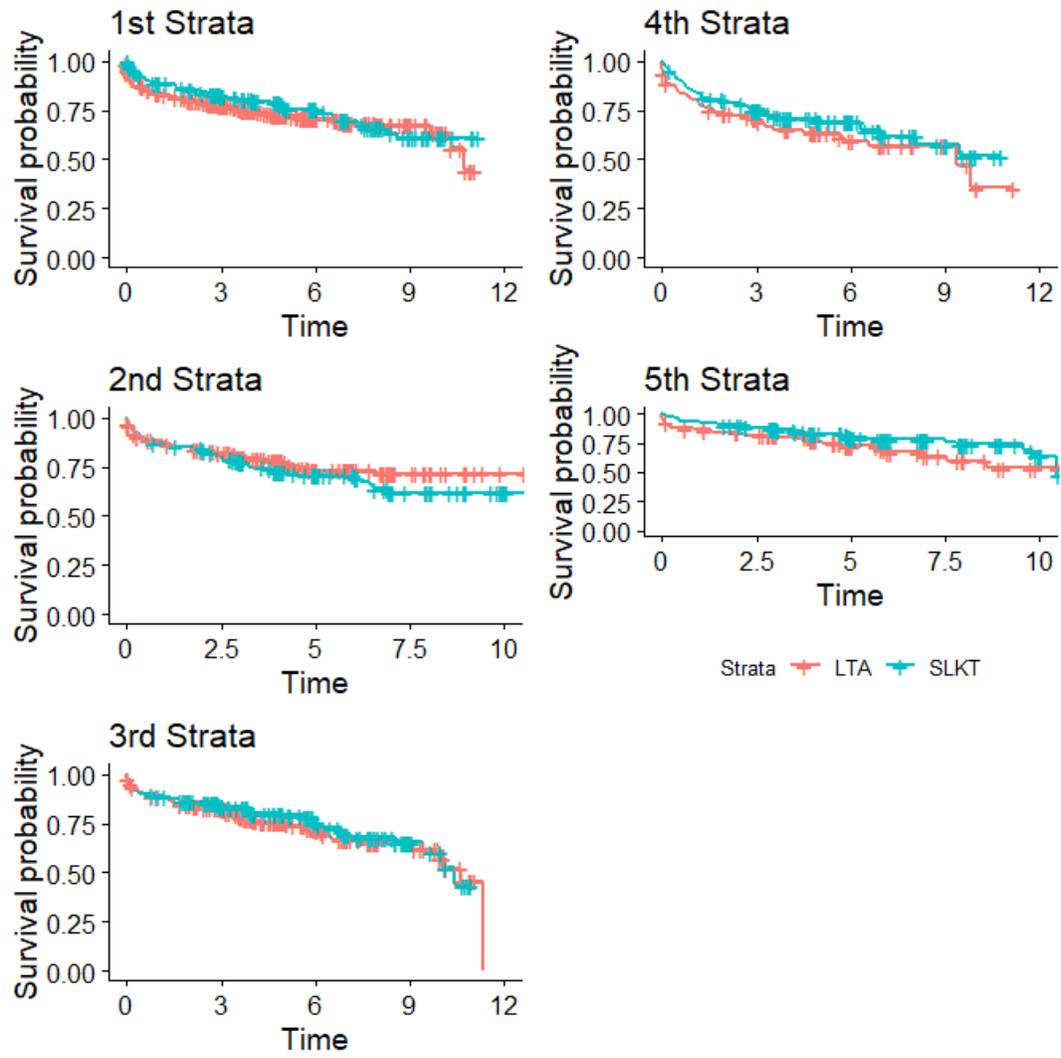


Figure 36 Kaplan-Meier curves – Classification Tree, cp=0.01 (stratification)

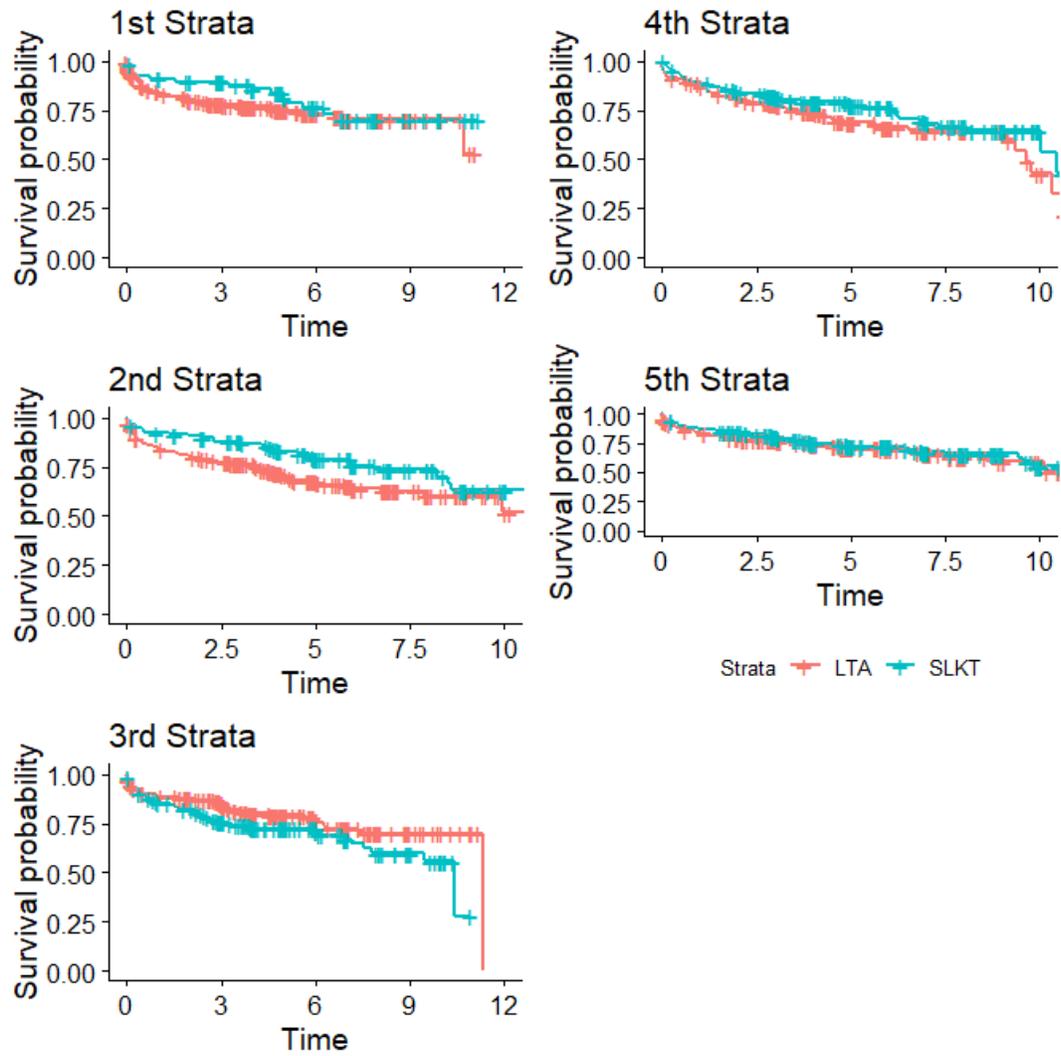
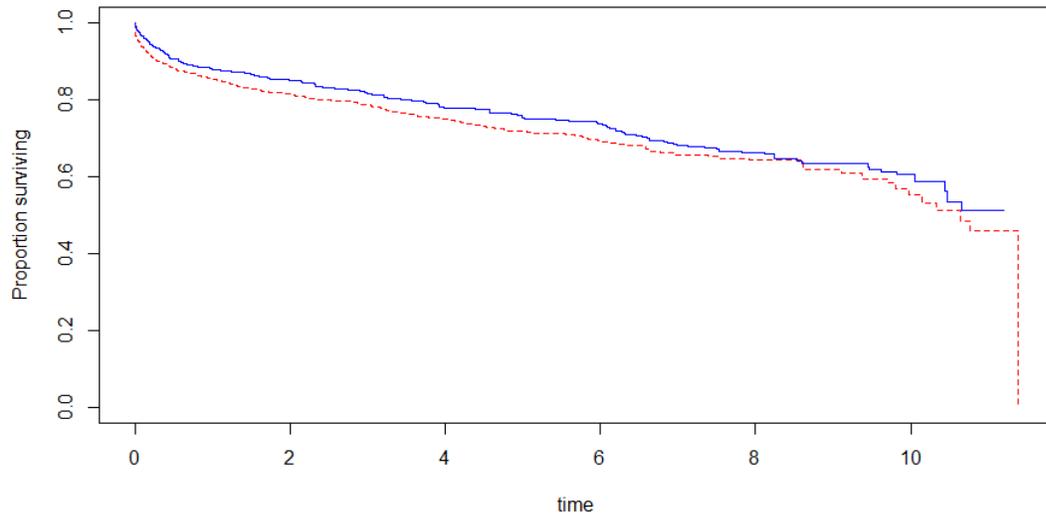
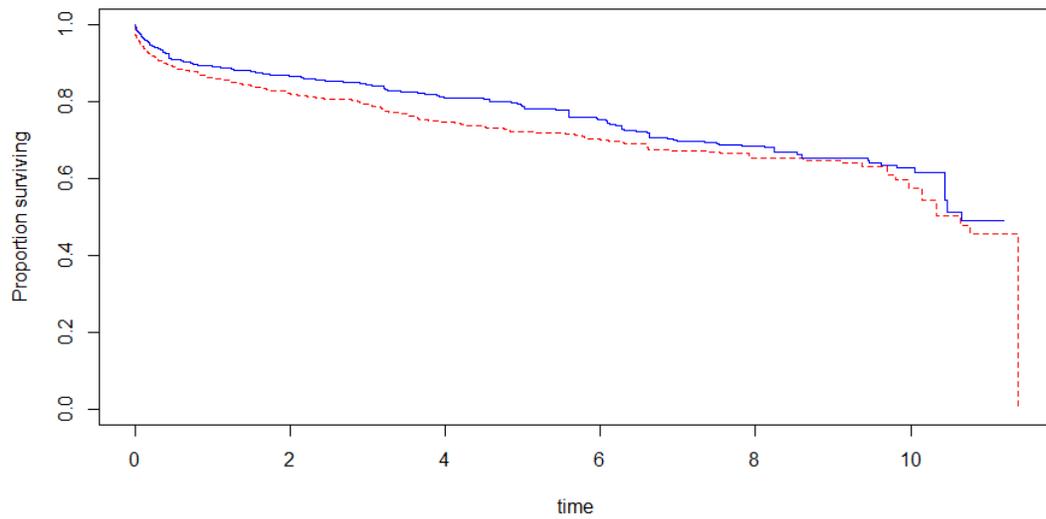


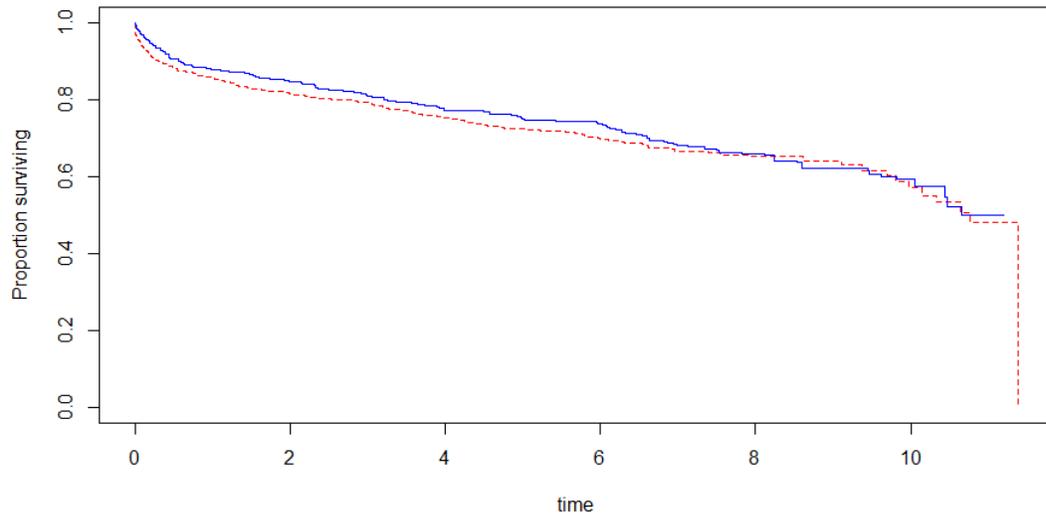
Figure 37 Kaplan-Meier curves – Random Forest (stratification)



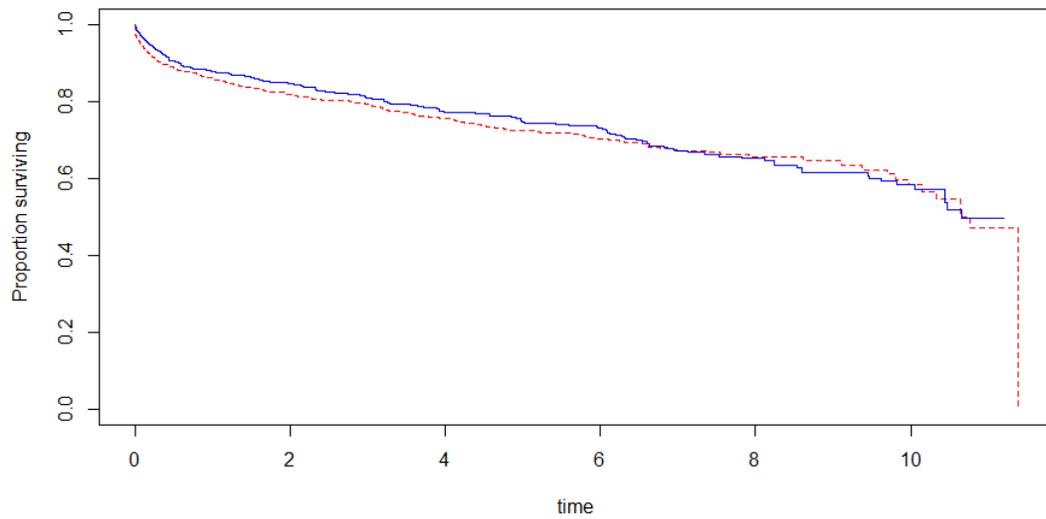
**Figure 38 Kaplan-Meier curves – Logistic Regression (IPTW)**



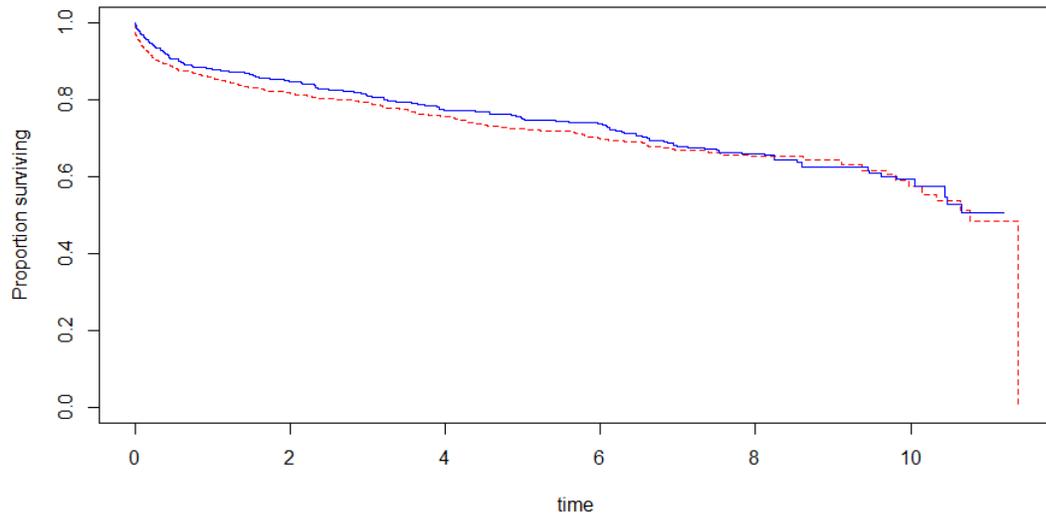
**Figure 39 Kaplan-Meier curves – Classification Tree, cp=0.0 (IPTW)**



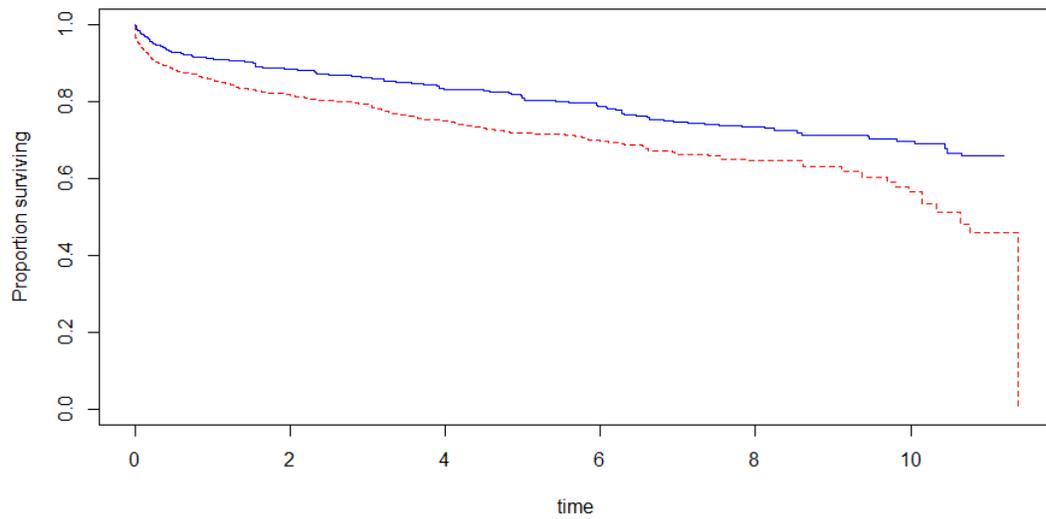
**Figure 40 Kaplan-Meier curves – Classification Tree, cp=0.0157 (IPTW)**



**Figure 41 Kaplan-Meier curves – Classification Tree, cp=0.0065 (IPTW)**



**Figure 42 Kaplan-Meier curves – Classification Tree, cp=0.01 (IPTW)**



**Figure 43 Kaplan-Meier curves – Random Forest (IPTW)**

## 4.0 Discussion

Overall results across most methods for the outcomes model indicated that the difference between SLKT and LTA treatments was insignificant for liver transplant patients on dialysis. However, the few results that were significant indicated that SLKT was, to some extent, a more effective treatment for dialysis patients than LTA. Considering choice of treatment assignment model and method for generating the pseudo population on estimates of causal effect, the outcome models showed that differences in methods used in steps 2 and 3 of the PS-based methods framework could in fact result in significantly different treatment effect estimates.

The patterns in treatment assignment model are unsurprising for the most part; the better performance of the logistic regression model when compared to the tree and forest models is reasonable and has also been noted in previous studies. (Austin, 2013) However, it is still interesting to note that while using the random forest model gave more significant results and showed more overlap in propensity score distributions when separated by treatment group, it was not as well-balanced as the logistic regression. The greater overlap in propensities, while helpful when considering the effectiveness of the matching approach, may not be indicative of a more successful treatment assignment mechanism; the true model is not known, so more separation in distributions may be closer to the true model.

The better balance in pseudo populations created from the logistic regression compared to those created from the random forest may be due to the difference inherent to the use of parametric (logistic regression) versus nonparametric (tree, forest) methods. If the model and covariates are specified correctly, then parametric methods will typically perform better than nonparametric ones. Furthermore, a random forest would introduce additional variations that a logistic regression model

would not. As such, a suggestion for future directions regarding this analysis would be to conduct a sensitivity analysis to assess the sensitivity of these PS-based methods to unmeasured confounding, which may currently be a major limitation to this analysis.

With respect to the classification tree treatment models, one concern might be the fact that the models do not use all the confounders fed into the initial model. The balance checks in section 3.4 considered balance for all possible covariates, not just the ones in each model, but a tree may not have been adept at balancing the covariates it did not include.

The PS distribution generated from a chosen treatment assignment model should be another consideration when selecting an approach to create the pseudo population with. This was shown by how the PS distribution of tree models, which had lower amounts of overlap than the logistic regression and random forest models, ultimately gave insignificant results, while the other treatment assignment models showed a significant difference in treatments.

In a similar way, the range of estimated propensities for both treatment groups generated from a treatment mechanism could also influence the suitability of the pseudo population and outcomes model. As the equations for IPTW in Section 2.2.3. indicate, propensities with more extreme values (close to 0 or 1) tend to be weighted higher. Therefore, a dataset with a large amount of 'borderline' values may find stronger inference from using a different pseudo population approach like matching, or – if the extreme values are outliers – it may be possible to truncate the treatment assignment model a small distance away from 0 and 1. Alternatively, if a treatment assignment model produces a set of PS estimates with too small a variance, particularly if they are clustered towards 0 or 1, it may be possible to consider a mechanism (or transformation) that spreads out the estimates before applying a pseudo population approach.

These results are comparable to similar studies on SLKT and LTA treatment in existing literature. Tanriover et al. (2016) conducted a PS matching analysis on liver transplant patients with moderate renal insufficiency and saw a decrease in mortality post-transplant for SLKT patients when compared to similar LTA patients. However, the results were not statistically significant until the analysis was done on patients who had not been exposed to dialysis. While this thesis looked only at patients on dialysis (with NAFLD or ALD) and saw significant results for one of the PS-matched pseudo populations, most results were similarly insignificant.

Sharma et al. (2016) used an IPTW analysis to consider liver transplant patients with and without dialysis, including kidney donor quality as a confounder, which this thesis did not do. Here, they saw a significant gain in 5-year mean post-transplant survival time (3.7 months) for SLKT patients, but a non-significant gain for patients not on dialysis. Like the analysis in this thesis, Sharma et al. observed a significant gain in survival time for SLKT patients on dialysis, although the advantage was small.

One characteristic of the sample used for this analysis was the notable difference in baseline proportions of SLKT observations in different regions. This inconsistency in regional allocation occurs because without standardized medical criteria for SLKT, allocation is often decided by geographic proximity between the SLKT donor and candidate. Since SLKT allocation is largely region-based, another possible method in future analyses to estimate the treatment effect may be to use the instrumental variables method mentioned in section 1.4, with transplant region being the instrumental variable.

In conclusion, this thesis, like the two papers discussed above, targeted a subset of patients on the liver transplant waiting list and attempted to add to the literature regarding SLKT transplants for such patients. It was shown that while it is not always clear how to optimally model the

treatment assignment mechanism or form the pseudo population, arbitrarily picking a model is not always a suitable approach, since choices regarding each step in the framework outlined in this thesis can make a difference in results. Propensity score-based approaches are a multi-step process, not a single method, and investigators need to be careful at each step of the approach.

## Appendix A Additional Output for Classification Tree Models

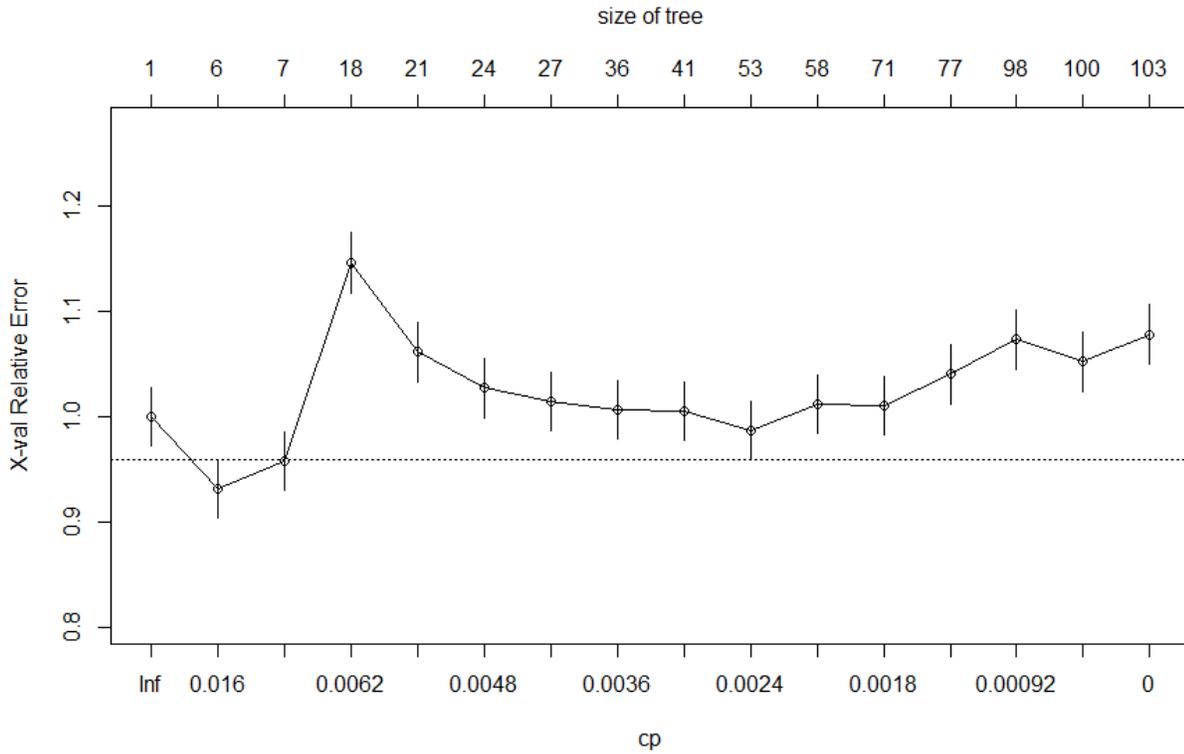


Figure 44 Complexity Parameter Selection

### CP selection with error values

Root node error:  $765/1930 = 0.39637$

n= 1930

	CP	nsplit	rel error	xerror	xstd	
1	0.01725490	0	1.00000	1.00000	0.028090	
2	0.01568627	5	0.91373	0.93072	0.027709	***
3	0.00653595	6	0.89804	0.95817	0.027871	**
4	0.00588235	17	0.82222	1.14641	0.028594	
5	0.00522876	20	0.80261	1.06144	0.028350	
6	0.00435730	23	0.78693	1.02745	0.028215	
7	0.00392157	26	0.77386	1.01438	0.028157	
8	0.00326797	35	0.73725	1.00654	0.028121	
9	0.00261438	40	0.72026	1.00523	0.028115	
10	0.00228758	52	0.68889	0.98693	0.028025	
11	0.00196078	57	0.67582	1.01176	0.028146	
12	0.00174292	70	0.64575	1.01046	0.028139	
13	0.00130719	76	0.62876	1.04052	0.028270	
14	0.00065359	97	0.60000	1.07320	0.028392	
15	0.00043573	99	0.59869	1.05229	0.028316	
16	0.00000000	102	0.59739	1.07843	0.028410	



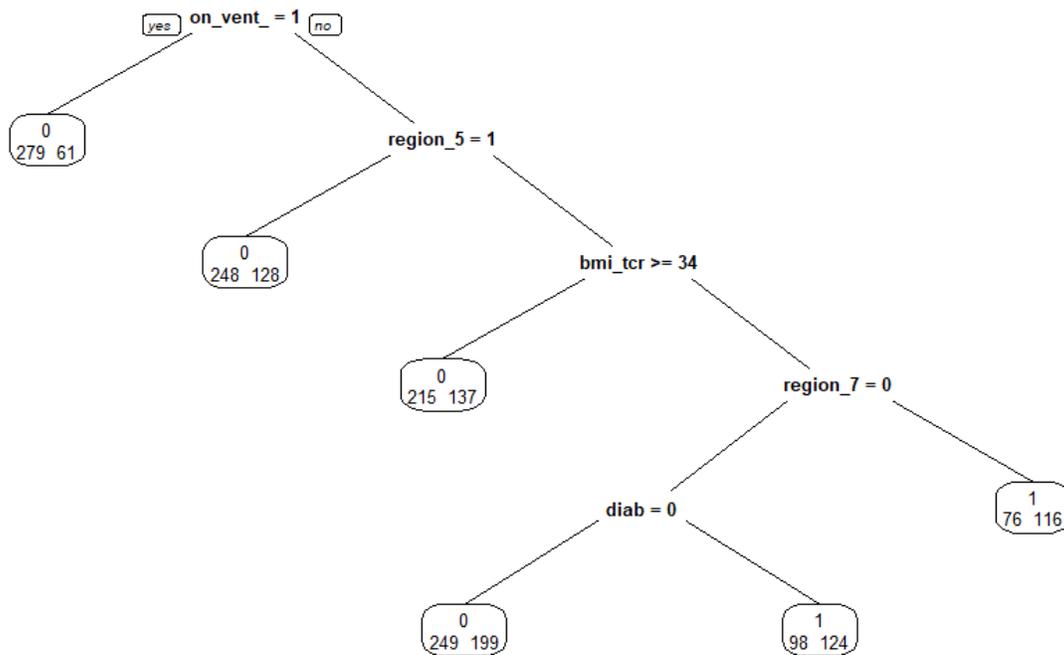


Figure 46 Classification Tree (cp=0.0157)

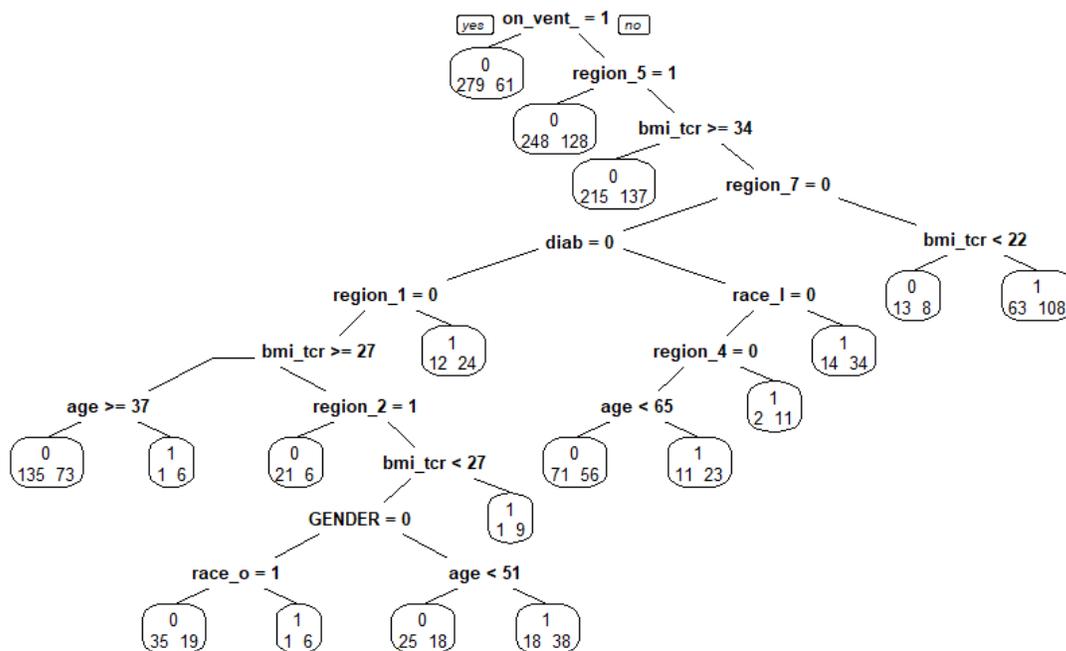


Figure 47 Classification Tree (cp=0.0065)

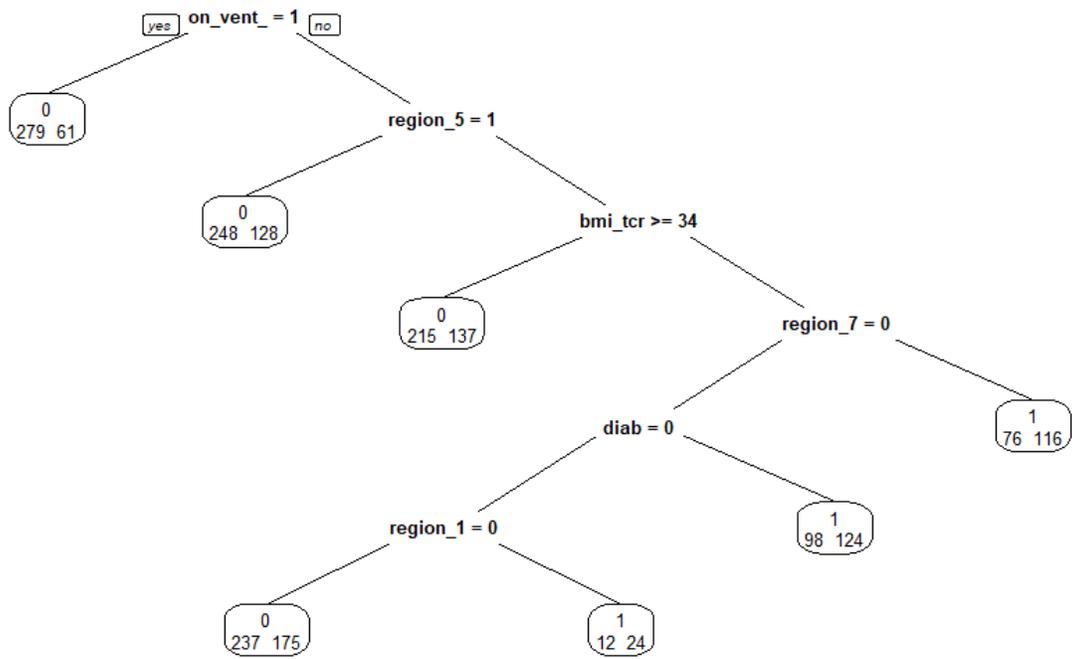


Figure 48 Classification Tree (cp=0.01)

## Appendix B Post-Pseudo Randomization Propensities

The figures below show the PS distributions, divided by treatment type, for the pseudo populations created using 1:1 matching and IPTW. Since stratified pseudo populations were simply created by dividing the pre-pseudo randomized PS distributions into quintiles, and can therefore be visualized in the pre-randomized PS distributions in section 3.2, they are not included in this appendix.

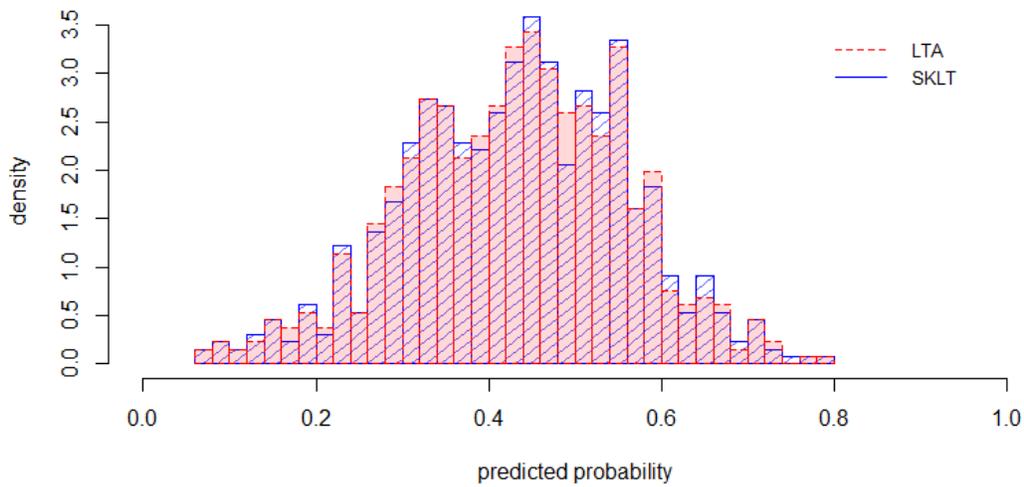


Figure 49 Post-pseudo randomized PS distributions, 1:1 Matching, Logistic Regression

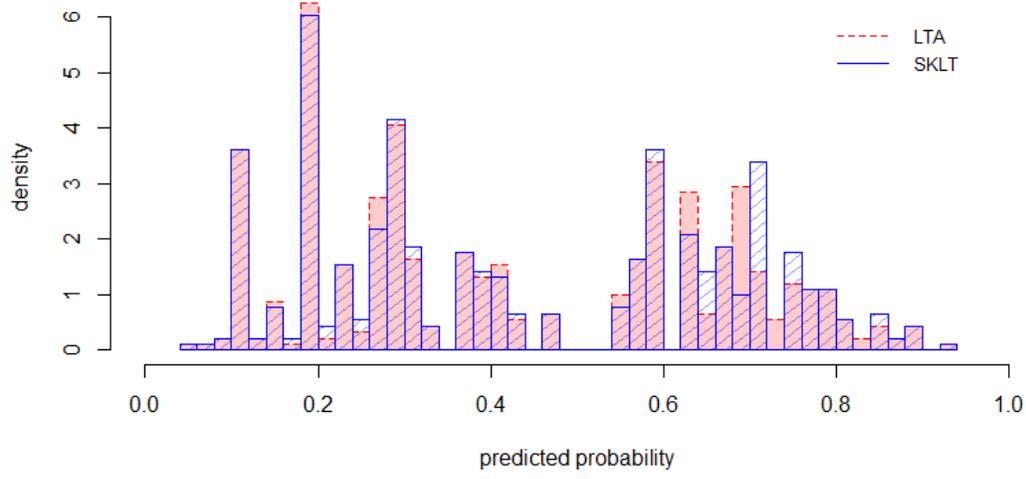


Figure 50 Post-pseudo randomized PS distributions, 1:1 Matching, Tree (cp=0.0)

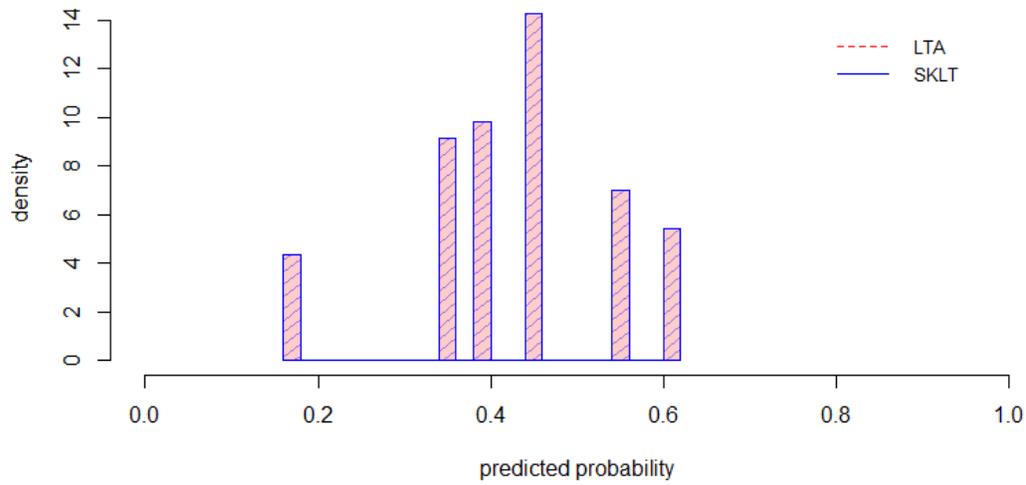
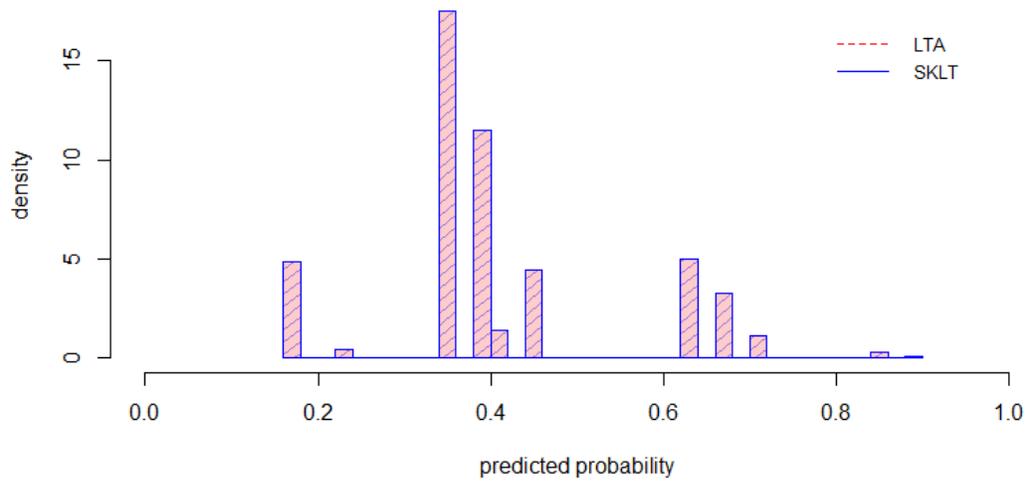
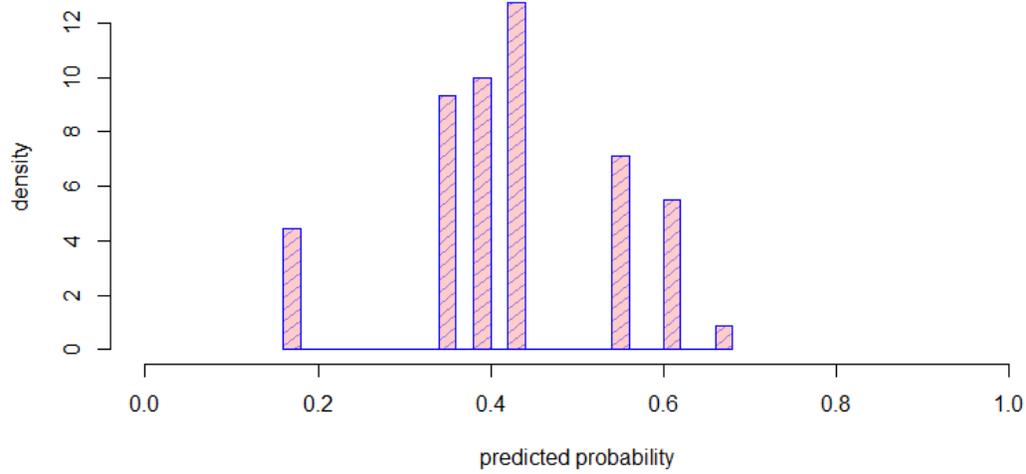


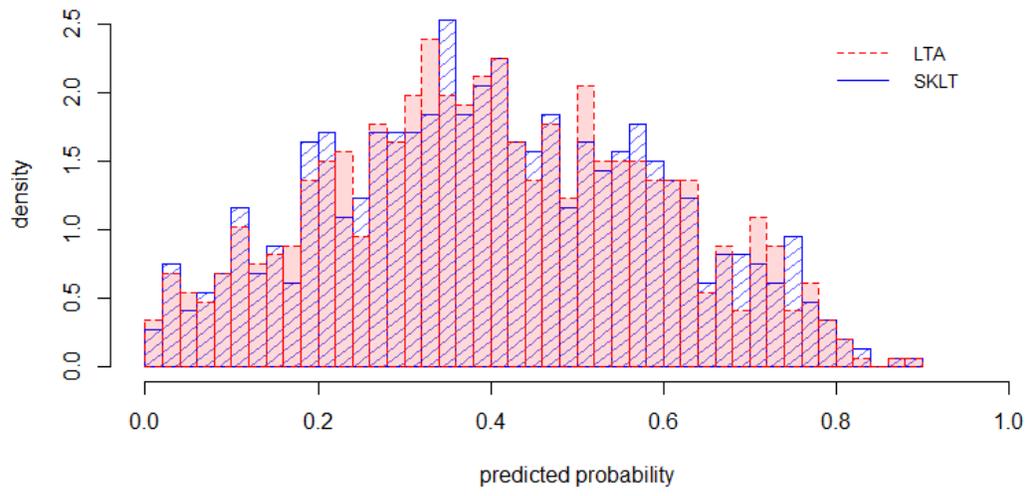
Figure 51 Post-pseudo randomized PS distributions, 1:1 Matching, Tree (cp=0.0157)



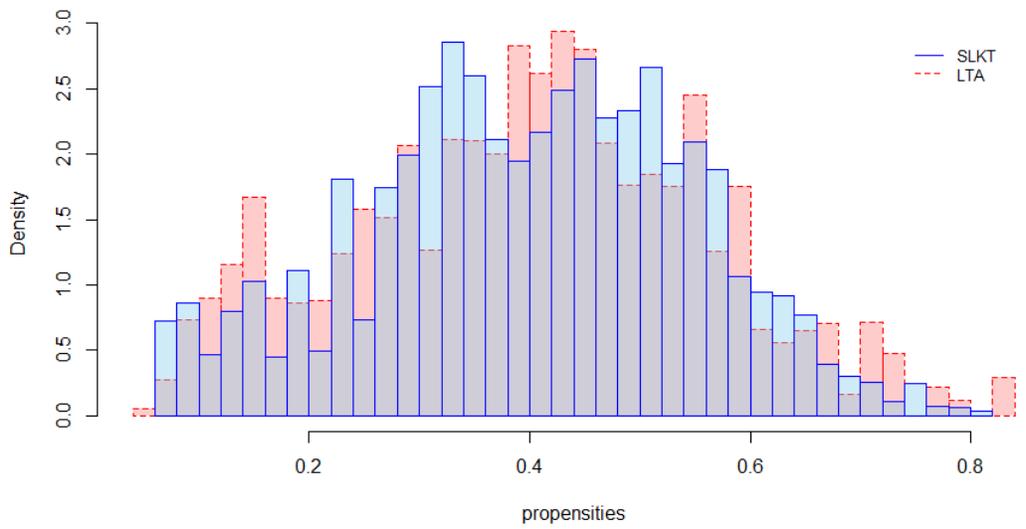
**Figure 52 Post-pseudo randomized PS distributions, 1:1 Matching, Tree (cp=0.0065)**



**Figure 53 Post-pseudo randomized PS distributions, 1:1 Matching, Tree (cp=0.01)**



**Figure 54 Post-pseudo randomized PS distributions, 1:1 Matching, Random Forest**



**Figure 55 Post-pseudo randomized PS distributions, IPTW, Logistic Regression**

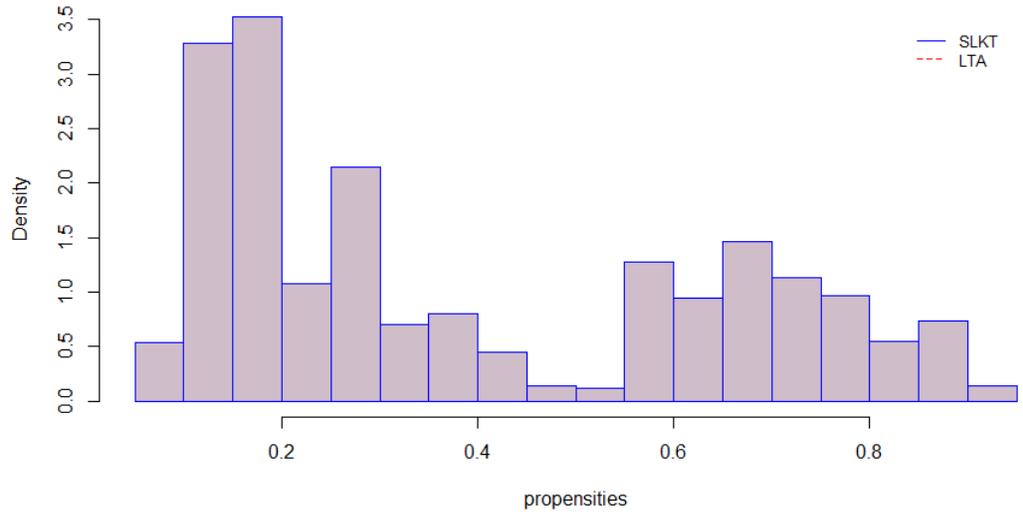


Figure 56 Post-pseudo randomized PS distributions, IPTW, Tree (cp=0.0)

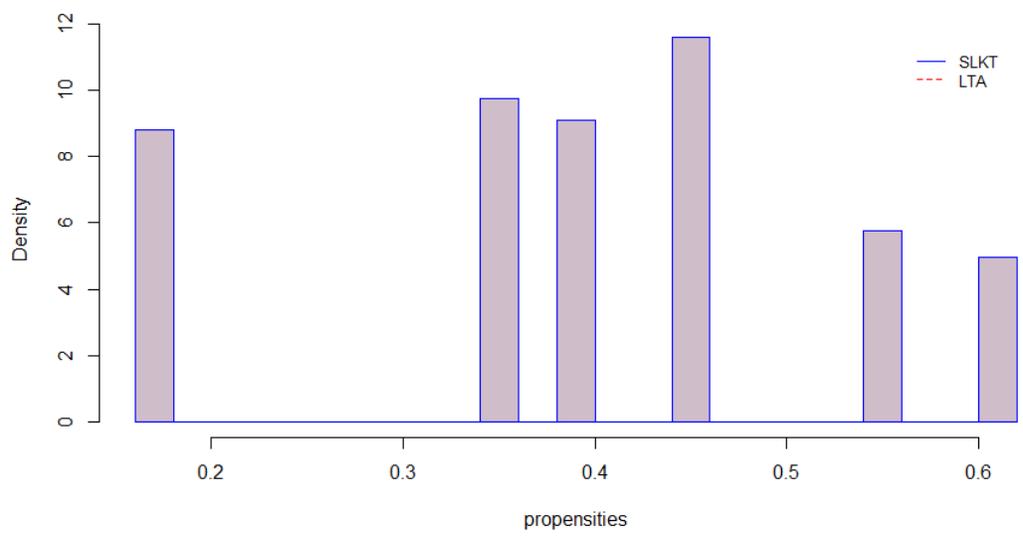
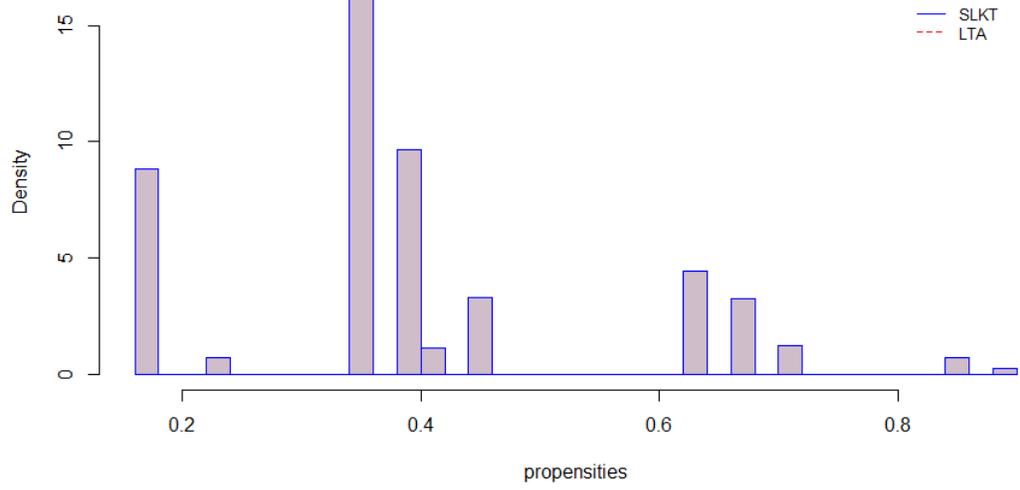
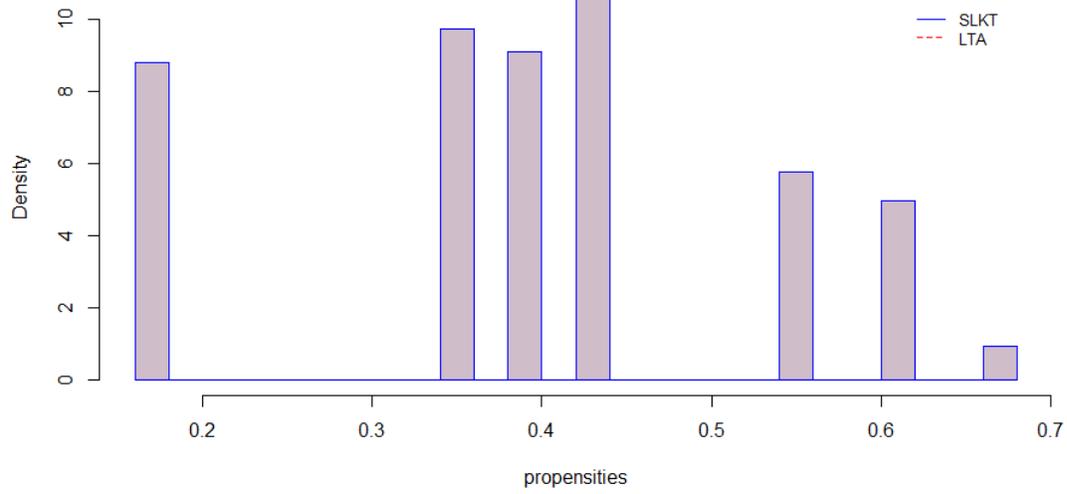


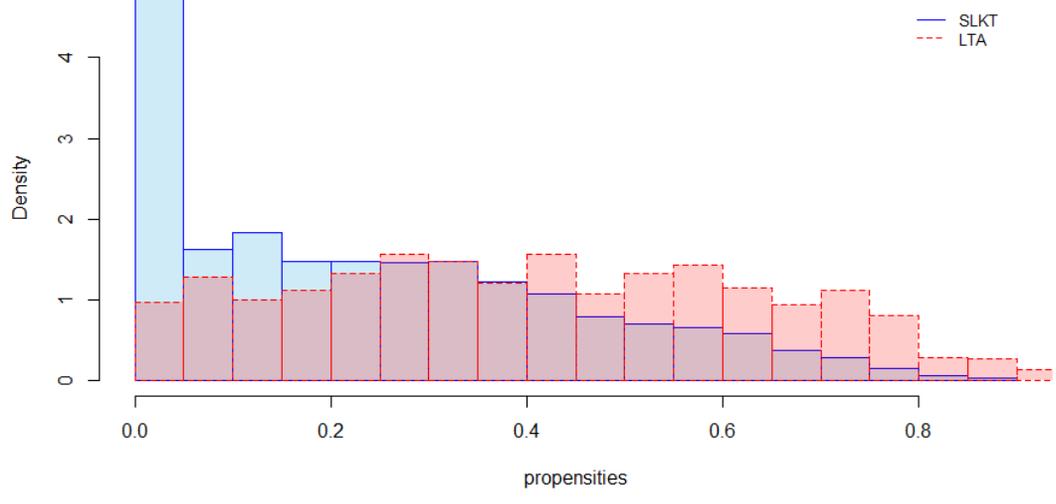
Figure 57 Post-pseudo randomized PS distributions, IPTW, Tree (cp=0.0157)



**Figure 58 Post-pseudo randomized PS distributions, IPTW, Tree (cp=0.0065)**



**Figure 59 Post-pseudo randomized PS distributions, IPTW, Tree (cp=0.01)**



**Figure 60** Post-pseudo randomized PS distributions, IPTW, Random Forest

## Appendix C Code

```
#install.packages("PSAgraphics")
#install.packages("rpart")
#install.packages("rpart.plot")
#install.packages("ggplot2")
#install.packages("survival")
#install.packages("survey")
#install.packages("HistogramTools")
#install.packages("randomForest")

#install.packages(c("e1071", "caret", "doSNOW", "ipred", "xgboost"))
#install.packages("bindrcpp")
#install.packages("ROSE")
#install.packages("pROC", dependencies=TRUE, repos='http://cran.rstudio.com/')

#install.packages("MatchIt")
#install.packages("Hmisc")
#install.packages("plyr")
#install.packages('gmodels')

#install.packages("survminer")
#install.packages("ggpubr")
#install.packages("magrittr")
#install.packages("stddiff")
#install.packages("nonrandom")
#install.packages("tableone")
#install.packages("cobalt")

library(PSAgraphics)
library(rpart)
library(rpart.plot)
library(survival)
library(HistogramTools)
library(randomForest)

#library(caret)
library(doSNOW)
library(ROSE)
library(pROC)

library(Hmisc)
library(ggplot2)
library(survey)
library(gmodels)

library(MatchIt)
library(survival)
library(survminer)
library(ggplot2)
```

```

library(ggpubr)
library(magrittr)
library(dplyr)

library(stddiff)
library(tableone)
library(cobalt)

setwd("Z:/PCORI/SLKT PS Example")
library(haven)
data <- read_dta("editedPSdata.dta")

colnames(data)
dim(data)
table(data$tx_typ)

# data pre-processing
data$region <- factor(data$region)
data$diab <- factor(data$diab)
data$PORTAL_VEIN_TRR <- factor(data$PORTAL_VEIN_TRR)
data$race <- factor(data$race)
data$tx_date<-as.Date(data$tx_date)

myvars<-c("tx_typ", "GENDER", "diab", "region", "on_vent_trr",
          "PORTAL_VEIN_TRR", "etiology", "race",
          "bmi_tcr", "age", "death", "tx_date", "ptime")
data_omit<-data[myvars]
data_omit<-na.omit(data_omit)

# SUBSETTING DATA
# 1 yr followup
d1<-subset(data_omit, tx_date < "2015-01-01")
d1_death<-subset(d1, death=="1")
d1_alive<-subset(d1, death=="0")
# working dataset is data_omit
data_omit<-d1

remove(d1)
remove(d1_alive)
remove(d1_death)

# NEW DEATH VARIABLE
#table for deaths and survival time <= 1 year
table(data_omit$death, data_omit$ptime<=365)

#adding indicator for death within 1 year
data_omit$death1yr<-data_omit$death
data_omit$death1yr<-ifelse(data_omit$ptime<=365,
                           ifelse(data_omit$death==1, 1, 0),0)

#checking count of subjects with death within 1 year
working<-data_omit$death1yr[data_omit$death1yr==1]
remove(working)

# SUMMARY STATISTICS

```

```

table(data_omit$death)

a<-data_omit$age
summary(a)
sd(a)

# RECODE VARIABLES

#recode diab to ordered 0=no 1=typeII 2=typeI
data_omit$diabNew<-data_omit$diab
data_omit$diabOld<-data_omit$diab

data_omit$diabNew[data_omit$diabNew==4]<-NA
data_omit$diabNew[data_omit$diabNew==5]<-NA
data_omit$diabNew[data_omit$diabNew==998]<-NA

#remove missing (1958 obs left)
data_omit<-na.omit(data_omit)
#drop empty levels
data_omit$diabNew<-droplevels(data_omit$diabNew)

data_omit$diabNew<-ifelse(data_omit$diabNew==3, 1,
                          ifelse(data_omit$diabNew==1, 0, 2))

data_omit$diabOld<-NULL
data_omit$diab<-NULL
data_omit$diab<-data_omit$diabNew
data_omit$diabNew<-NULL

#recode portal_vein_trr to treat U as missing
data_omit$PORTAL_VEIN_TRR[data_omit$PORTAL_VEIN_TRR==1]<-NA

#remove missing (1930 obs left)
data_omit<-na.omit(data_omit)

#recode portal_vein_trr y=2 to y=1
data_omit$PORTAL_VEIN_TRR<-ifelse(data_omit$PORTAL_VEIN_TRR==2, 1, 0)

#recode race as indicators (FOR TREE AND FOREST MODELS)
data_omit$race_o<-ifelse(data_omit$race==0, 1, 0)
data_omit$race_l<-ifelse(data_omit$race==1, 1, 0)
data_omit$race_b<-ifelse(data_omit$race==2, 1, 0)

#recode region 11 as region 0
data_omit$region<-ifelse(data_omit$region==11, 0, data_omit$region)

#recode region as indicators (FOR TREE AND FOREST MODELS)
data_omit$region_1<-ifelse(data_omit$region==1, 1, 0)
data_omit$region_2<-ifelse(data_omit$region==2, 1, 0)
data_omit$region_3<-ifelse(data_omit$region==3, 1, 0)
data_omit$region_4<-ifelse(data_omit$region==4, 1, 0)
data_omit$region_5<-ifelse(data_omit$region==5, 1, 0)
data_omit$region_6<-ifelse(data_omit$region==6, 1, 0)
data_omit$region_7<-ifelse(data_omit$region==7, 1, 0)
data_omit$region_8<-ifelse(data_omit$region==8, 1, 0)

```

```

data_omit$region_9<-ifelse(data_omit$region==9, 1, 0)
data_omit$region_10<-ifelse(data_omit$region==10, 1, 0)
data_omit$region_11<-ifelse(data_omit$region==0, 1, 0)

#check variable types
str(data_omit)

#recode as factor variables
data_omit$region<-factor(data_omit$region)
data_omit$region_1<-factor(data_omit$region_1)
data_omit$region_2<-factor(data_omit$region_2)
data_omit$region_3<-factor(data_omit$region_3)
data_omit$region_4<-factor(data_omit$region_4)
data_omit$region_5<-factor(data_omit$region_5)
data_omit$region_6<-factor(data_omit$region_6)
data_omit$region_7<-factor(data_omit$region_7)
data_omit$region_8<-factor(data_omit$region_8)
data_omit$region_9<-factor(data_omit$region_9)
data_omit$region_10<-factor(data_omit$region_10)
data_omit$region_11<-factor(data_omit$region_11)

data_omit$race <- factor(data_omit$race)
data_omit$race_o <- factor(data_omit$race_o)
data_omit$race_l <- factor(data_omit$race_l)
data_omit$race_b <- factor(data_omit$race_b)

data_omit$diab <- factor(data_omit$diab)
data_omit$PORTAL_VEIN_TRR <- factor(data_omit$PORTAL_VEIN_TRR)
data_omit$on_vent_trr <- factor(data_omit$on_vent_trr)
data_omit$etiology <- factor(data_omit$etiology)
data_omit$death <- factor(data_omit$death)
data_omit$death1yr <- factor(data_omit$death1yr)
data_omit$GENDER <- factor(data_omit$GENDER)
data_omit$tx_typ <- factor(data_omit$tx_typ)

# DESCRIPTIVES

table(data_omit$diab)

library(gmodels)
CrossTable(data_omit$diab)
CrossTable(data_omit$region)
CrossTable(data_omit$race)
CrossTable(data_omit$GENDER)
CrossTable(data_omit$etiology)
CrossTable(data_omit$PORTAL_VEIN_TRR)
CrossTable(data_omit$on_vent_trr)
CrossTable(data_omit$tx_typ)
CrossTable(data_omit$death1yr)

summary(data_omit$bmi_tcr)
sd(data_omit$bmi_tcr)
summary(data_omit$age)
sd(data_omit$age)

```

```

summary(data_omit$pyear)
sd(data_omit$pyear)

# SUMMARY STATISTICS BY LTA/SKLT GROUP BEFORE ANALYSIS
LTA<-subset(data_omit, tx_typ==0)
SKLT<-subset(data_omit, tx_typ==1)

#rerunning summary statistics for LTA group only
CrossTable(LTA$diab)
CrossTable(LTA$region)
CrossTable(LTA$race)
CrossTable(LTA$GENDER)
CrossTable(LTA$etiology)
CrossTable(LTA$PORTAL_VEIN_TRR)
CrossTable(LTA$on_vent_trr)
CrossTable(LTA$death1yr)
summary(LTA$bmi_tcr)
sd(LTA$bmi_tcr)
summary(LTA$age)
sd(LTA$age)
summary(LTA$pyear)
sd(LTA$pyear)

#rerunning summary statistics for SKLT group only
CrossTable(SKLT$diab)
CrossTable(SKLT$region)
CrossTable(SKLT$race)
CrossTable(SKLT$GENDER)
CrossTable(SKLT$etiology)
CrossTable(SKLT$PORTAL_VEIN_TRR)
CrossTable(SKLT$on_vent_trr)
CrossTable(SKLT$death1yr)
summary(SKLT$bmi_tcr)
sd(SKLT$bmi_tcr)
summary(SKLT$age)
sd(SKLT$age)
summary(SKLT$pyear)
sd(SKLT$pyear)

#####

# CALCULATING PROPENSITY SCORES

# ---Logistic Regression---
# propensity score model (from logistic regression)
ps_1=glm(tx_typ ~ GENDER + diab + region + on_vent_trr +
          PORTAL_VEIN_TRR + etiology + race + bmi_tcr + age,
          family=binomial(link='logit'), data=data_omit,
          control=list(maxit=50), na.action=na.omit)
summary(ps_1)

#predicted probs
preds1<-predict(ps_1,type="response")
#coding preds1 as a dataset
preds1.df <- as.data.frame(preds1)

```

```

#add PS column to dataset
data_omit <- cbind(data_omit, predict(ps_1,type="response"))
data_omit<-data_omit[c(1:29)]
names(data_omit)[29]<-"LRprops"

#graphing propensities by LTA and SKLT
hist(data_omit$LRprops[data_omit$tx_typ==0],
      main=" ",
      xlab="predicted probability", ylab="density",
      lty=2,
      border="red",
      xlim=c(0,1),
      col=scales::alpha('red',.2),
      breaks=35)
hist(data_omit$LRprops[data_omit$tx_typ==1], add=T,
      lty=1,
      border="blue",
      col=scales::alpha('blue',.4),
      angle=45, density=20, lwd=1,
      breaks=35)
legend("topright", inset=.05, legend=c("LTA", "SKLT"),
      col=c("red", "blue"), lty=2:1, cex=0.8, box.lty=0)

# ---Classification Trees---
# propensity score model (from classification tree)
# fit unpruned tree model
# note: xval = tot_Obs - 1 = 1930-1 = 1929
set.seed(1021)
fit.data <- rpart(formula = tx_typ ~ GENDER + diab + on_vent_trr +
                  PORTAL_VEIN_TRR + etiology + bmi_tcr + age +
                  race_b + race_l + race_o +
                  region_11 + region_10 + region_9 + region_8 +
                  region_7 + region_6 + region_5 + region_4 +
                  region_3 + region_2 + region_1,
                  data=data_omit, method = "class",
                  control = rpart.control (cp=0, xval=1929))
# print unpruned tree
prp(fit.data, faclen = 0, cex=0.8, extra = 1)
# print CP values and error
printcp(fit.data)
# plot CP values and error
plotcp(fit.data)
# lowest x-val rel. error is with CP=0.0157, 2nd lowest is 0.0065
#
#      CP nsplit rel error  xerror   xstd
#2  0.01568627    5  0.91373 0.93072 0.027709
#3  0.00653595    6  0.89804 0.95817 0.027871

#prune the tree - cp with lowest error was cp=.0157
fit.data.1<-prune(fit.data, cp=0.0157)
prp(fit.data.1, faclen = 0, cex=0.8, extra = 1)
printcp(fit.data.1)

#prune the tree - cp with 2nd lowest error was cp=.0065

```

```

fit.data.2<-prune(fit.data, cp=0.0065)
prp(fit.data.2, faclen = 0, cex=0.8, extra = 1)
printcp(fit.data.2)

#prune the tree - cp=.01
fit.data.3<-prune(fit.data, cp=0.01)
prp(fit.data.3, faclen = 0, cex=0.8, extra = 1)
printcp(fit.data.3)

#prune the tree - cp=.02 -NOTE: NO SPLIT
fit.data.4<-prune(fit.data, cp=0.02)
prp(fit.data.4, faclen = 0, cex=0.8, extra = 1)
printcp(fit.data.4)

#CT2.1 - cp=0
set.seed(1021)
ps_2.1=rpart(formula = tx_typ ~ GENDER + diab + on_vent_trr +
              PORTAL_VEIN_TRR + etiology + bmi_tcr + age +
              race_b + race_l + race_o +
              region_11 + region_10 + region_9 + region_8 +
              region_7 + region_6 + region_5 + region_4 +
              region_3 + region_2 + region_1,
              data=data_omit, method = "class",
              control = rpart.control (cp=0, xval=1929))

#add PS column to dataset
data_omit <- cbind(data_omit, predict(ps_2.1,type="prob"))
data_omit<-data_omit[c(1:29, 31)]
names(data_omit)[30]<-"CT2.1props"

#graphing propensities by LTA and SKLT
hist(data_omit$CT2.1props[data_omit$tx_typ==0],
      main=" ",
      xlab="predicted probability", ylab="density",
      lty=2,
      border="red",
      xlim=c(0,1),
      col=scales::alpha('red',.2),
      breaks=35)
hist(data_omit$CT2.1props[data_omit$tx_typ==1], add=T,
      lty=1,
      border="blue",
      col=scales::alpha('blue',.4),
      angle=45, density=20, lwd=1,
      breaks=35)
legend("topright", inset=.05, legend=c("LTA", "SKLT"),
      col=c("red", "blue"), lty=2:1, cex=0.8, box.lty=0)

#CT2.2 - cp=0.0157
set.seed(1021)
ps_2.2=rpart(formula = tx_typ ~ GENDER + diab + on_vent_trr +
              PORTAL_VEIN_TRR + etiology + bmi_tcr + age +
              race_b + race_l + race_o +
              region_11 + region_10 + region_9 + region_8 +
              region_7 + region_6 + region_5 + region_4 +

```

```

        region_3 + region_2 + region_1,
        data=data_omit, method = "class",
        control = rpart.control (cp=0.0157, xval=1929))

#add PS column to dataset
data_omit <- cbind(data_omit, predict(ps_2.2,type="prob"))
data_omit<-data_omit[c(1:30, 32)]
names(data_omit)[31]<-"CT2.2props"

#graphing propensities by LTA and SKLT
hist(data_omit$CT2.2props[data_omit$tx_typ==0],
      main=" ",
      xlab="predicted probability", ylab="density",
      lty=2,
      border="red",
      xlim=c(0,1),
      col=scales::alpha('red',.2),
      breaks=25)
hist(data_omit$CT2.2props[data_omit$tx_typ==1], add=T,
      lty=1,
      border="blue",
      col=scales::alpha('blue',.4),
      angle=45, density=20, lwd=1,
      breaks=25)
legend("topright", inset=.05, legend=c("LTA", "SKLT"),
      col=c("red", "blue"), lty=2:1, cex=0.8, box.lty=0)

#CT2.3 - cp=0.0065
set.seed(1021)
ps_2.3=rpart(formula = tx_typ ~ GENDER + diab + on_vent_trr +
  PORTAL_VEIN_TRR + etiology + bmi_tcr + age +
  race_b + race_l + race_o +
  region_11 + region_10 + region_9 + region_8 +
  region_7 + region_6 + region_5 + region_4 +
  region_3 + region_2 + region_1,
  data=data_omit, method = "class",
  control = rpart.control (cp=0.0065, xval=1929))

#add PS column to dataset
data_omit <- cbind(data_omit, predict(ps_2.3,type="prob"))
data_omit<-data_omit[c(1:31, 33)]
names(data_omit)[32]<-"CT2.3props"

CrossTable(data_omit$CT2.3props)
table(data_omit$CT2.3props)

#graphing propensities by LTA and SKLT
hist(data_omit$CT2.3props[data_omit$tx_typ==0],
      main=" ",
      xlab="predicted probability", ylab="density",
      lty=2,
      border="red",
      xlim=c(0,1),
      col=scales::alpha('red',.2),

```

```

    breaks=35)
hist(data_omit$CT2.3props[data_omit$tx_typ==1], add=T,
     lty=1,
     border="blue",
     col=scales::alpha('blue',.4),
     angle=45, density=20, lwd=1,
     breaks=35)
legend("topright", inset=.05, legend=c("LTA", "SKLT"),
      col=c("red", "blue"), lty=2:1, cex=0.8, box.lty=0)

#CT2.4 - cp=0.01
set.seed(1021)
ps_2.4=rpart(formula = tx_typ ~ GENDER + diab + on_vent_trr +
             PORTAL_VEIN_TRR + etiology + bmi_tcr + age +
             race_b + race_l + race_o +
             region_11 + region_10 + region_9 + region_8 +
             region_7 + region_6 + region_5 + region_4 +
             region_3 + region_2 + region_1,
             data=data_omit, method = "class",
             control = rpart.control (cp=0.01, xval=1929))

#add PS column to dataset
data_omit <- cbind(data_omit, predict(ps_2.4,type="prob"))
data_omit<-data_omit[c(1:32, 34)]
names(data_omit)[33]<-"CT2.4props"

CrossTable(data_omit$CT2.4props)
table(data_omit$CT2.4props)

#graphing propensities by LTA and SKLT
hist(data_omit$CT2.4props[data_omit$tx_typ==0],
     main=" ",
     xlab="predicted probability", ylab="density",
     lty=2,
     border="red",
     xlim=c(0,1),
     col=scales::alpha('red',.2),
     breaks=25)
hist(data_omit$CT2.4props[data_omit$tx_typ==1], add=T,
     lty=1,
     border="blue",
     col=scales::alpha('blue',.4),
     angle=45, density=20, lwd=1,
     breaks=25)
legend("topright", inset=.05, legend=c("LTA", "SKLT"),
      col=c("red", "blue"), lty=2:1, cex=0.8, box.lty=0)

# ---Random Forest---
library(randomForest)
# propensity score model (from random forest)
set.seed(1021)
ps_3<-randomForest(tx_typ ~ GENDER + diab + on_vent_trr +
                  PORTAL_VEIN_TRR + etiology + bmi_tcr + age +

```

```

        race_b + race_l + race_o +
        region_11 + region_10 + region_9 + region_8 +
        region_7 + region_6 + region_5 + region_4 +
        region_3 + region_2 + region_1,
        data=data_omit, na.action=na.omit, importance=T,
        ntree=10000)

#mtry - what will
#Number of variables randomly sampled as candidates at each split.
#Note that the default values are different for classification (sqrt(p)
#where p is number of variables in x) and regression (p/3)
# ---can know optimal value if u do crossvalidation

#adding PS column to dataset
data_omit <- cbind(data_omit, predict(ps_3,type="prob"))
data_omit<-data_omit[c(1:33,35)]
names(data_omit)[34]<-"RFprops"

#graphing propensities by LTA and SKLT
hist(data_omit$RFprops[data_omit$tx_typ==0],
      main=" ",
      xlab="predicted probability", ylab="density",
      lty=2,
      border="red",
      xlim=c(0,1),
      col=scales::alpha('red',.2),
      breaks=35)
hist(data_omit$RFprops[data_omit$tx_typ==1], add=T,
      lty=1,
      border="blue",
      col=scales::alpha('blue',.4),
      angle=45, density=20, lwd=1,
      breaks=35)
legend("topright", inset=.05, legend=c("LTA", "SKLT"),
      col=c("red", "blue"), lty=2:1, cex=0.8, box.lty=0)

#generate survival time variable in years
data_omit$pyear<-data_omit$ptime/365.25
#data_omit$new<-data_omit$ptime
#data_omit<-data_omit[c(1:35)]

#generate numeric treatment type variable
data_omit$tx_typ2<-as.numeric(data_omit$tx_typ)
data_omit$tx_typ2<-ifelse(data_omit$tx_typ2==1,0,1)

#generate numeric death1yr variable
data_omit$death1yr2<-as.numeric(data_omit$death1yr)
data_omit$death1yr2<-ifelse(data_omit$death1yr2==1,0,1)

#generate numeric death variable
data_omit$death2<-as.numeric(data_omit$death)
data_omit$death2<-ifelse(data_omit$death2==1,0,1)

```

```
#####

#--MATCHING--

library(MatchIt)
library(survival)
library(survminer)
library(ggplot2)
library(ggpubr)
library(magrittr)
library(dplyr)

#set sample size
n = 1930

#get matched dataset with match index (1:1 matching)
data_with_match_index_one <- function(match){
  m <- na.omit(match$match.matrix)
  m1 <- cbind(rownames(m),m)
  case <- as.numeric(m1)
  match_index <- rep(1:nrow(m),2)
  m2 <- as.data.frame(cbind(case,match_index))
  m3 <- m2[order(m2$case),]
  m4 <- cbind(match.data(match),m3$match_index)
  colnames(m4)[ncol(m4)] <- "match_index"
  return(m4)
}

#LOGISTIC REGRESSION
#1:1 nearest neighbor matching
ps_1.sd = sd(data_omit$LRprops)
set.seed(1021)
ps_1.one = matchit(tx_typ ~ GENDER + diab + region + on_vent_trr +
  PORTAL_VEIN_TRR + etiology + race + bmi_tcr + age,
  data=data_omit,
  method="nearest",
  distance=data_omit$LRprops,
  caliper = 0.2*ps_1.sd)
ps_1.one.data = data_with_match_index_one(ps_1.one)

#treatment effect - Logistic Regression (matching)
#matched pseudo-population using clustering on the matched set
cox1.3<-coxph(Surv(pyear,death2) ~ tx_typ + survival::cluster(match_index),
  data=ps_1.one.data)
summary(cox1.3)

#Kaplan-Meier curves and Log_rank test
ps_1.one.data$tx_typ<-factor(ps_1.one.data$tx_typ,
  levels=c("0", "1"),
  labels=c("LTA", "SLKT"))
surv_object<-Surv(time = ps_1.one.data$pyear,
  event=ps_1.one.data$death2)
fit1 <- survfit(surv_object ~ tx_typ, data=ps_1.one.data)
ggsurvplot(fit1,
```

```

        data=ps_1.one.data,
        pval=F)

#CLASSIFICATION TREE, CP=0
#1:1 nearest neighbor matching
ps_2.1.sd = sd(data_omit$CT2.1props)
set.seed(1021)
ps_2.1.one = matchit(tx_typ ~ GENDER + diab + on_vent_trr +
                    PORTAL_VEIN_TRR + etiology + bmi_tcr + age +
                    race_b + race_l + race_o +
                    region_11 + region_10 + region_9 + region_8 +
                    region_7 + region_6 + region_5 + region_4 +
                    region_3 + region_2 + region_1,
                    data=data_omit,
                    method="nearest",
                    distance=data_omit$CT2.1props,
                    caliper = 0.2*ps_2.1.sd)
ps_2.1.one.data = data_with_match_index_one(ps_2.1.one)

#treatment effect - Classification Tree, cp=0.0 (matching)
#matched pseudo-population using clustering on the matched set
cox1.3<-coxph(Surv(pyear,death2) ~ tx_typ + cluster(match_index),
              data=ps_2.1.one.data)
summary(cox1.3)

#Kaplan-Meier curves and Log_rank test
ps_2.1.one.data$tx_typ<-factor(ps_2.1.one.data$tx_typ,
                              levels=c("0", "1"),
                              labels=c("LTA", "SLKT"))
surv_object<-Surv(time = ps_2.1.one.data$year,
                  event=ps_2.1.one.data$death2)
fit1 <- survfit(surv_object ~ tx_typ, data=ps_2.1.one.data)
ggsurvplot(fit1,
           data=ps_2.1.one.data,
           pval=F)

#CLASSIFICATION TREE, CP=0.0157
#1:1 nearest neighbor matching
ps_2.2.sd = sd(data_omit$CT2.2props)
set.seed(1021)
ps_2.2.one = matchit(tx_typ ~ GENDER + diab + on_vent_trr +
                    PORTAL_VEIN_TRR + etiology + bmi_tcr + age +
                    race_b + race_l + race_o +
                    region_11 + region_10 + region_9 + region_8 +
                    region_7 + region_6 + region_5 + region_4 +
                    region_3 + region_2 + region_1,
                    data=data_omit,
                    method="nearest",
                    distance=data_omit$CT2.2props,
                    caliper = 0.2*ps_2.2.sd)
ps_2.2.one.data = data_with_match_index_one(ps_2.2.one)

#treatment effect - Classification Tree, cp=0.0157 (matching)
#matched pseudo-population using clustering on the matched set
cox1.3<-coxph(Surv(pyear,death2) ~ tx_typ + cluster(match_index),

```

```

        data=ps_2.2.one.data)
summary(cox1.3)

#Kaplan-Meier curves and Log_rank test
ps_2.2.one.data$tx_typ<-factor(ps_2.2.one.data$tx_typ,
                             levels=c("0","1"),
                             labels=c("LTA","SLKT"))
surv_object<-Surv(time = ps_2.2.one.data$pyear,
                  event=ps_2.2.one.data$death2)
fit1 <- survfit(surv_object ~ tx_typ, data=ps_2.2.one.data)
ggsurvplot(fit1,
           data=ps_2.2.one.data,
           pval=F)

#CLASSIFICATION TREE, CP=0.0065
#1:1 nearest neighbor matching
ps_2.3.sd = sd(data_omit$CT2.3props)
set.seed(1021)
ps_2.3.one = matchit(tx_typ ~ GENDER + diab + on_vent_trr +
                    PORTAL_VEIN_TRR + etiology + bmi_tcr + age +
                    race_b + race_l + race_o +
                    region_11 + region_10 + region_9 + region_8 +
                    region_7 + region_6 + region_5 + region_4 +
                    region_3 + region_2 + region_1,
                    data=data_omit,
                    method="nearest",
                    distance=data_omit$CT2.3props,
                    caliper = 0.2*ps_2.3.sd)
ps_2.3.one.data = data_with_match_index_one(ps_2.3.one)

#treatment effect - Classification Tree, cp=0.0065 (matching)
#matched pseudo-population using clustering on the matched set
cox1.3<-coxph(Surv(pyear,death2) ~ tx_typ + cluster(match_index),
              data=ps_2.3.one.data)
summary(cox1.3)

#Kaplan-Meier curves and Log_rank test
ps_2.3.one.data$tx_typ<-factor(ps_2.3.one.data$tx_typ,
                              levels=c("0","1"),
                              labels=c("LTA","SLKT"))
surv_object<-Surv(time = ps_2.3.one.data$pyear,
                  event=ps_2.3.one.data$death2)
fit1 <- survfit(surv_object ~ tx_typ, data=ps_2.3.one.data)
ggsurvplot(fit1,
           data=ps_2.3.one.data,
           pval=F)

#CLASSIFICATION TREE, CP=0.01
#1:1 nearest neighbor matching
ps_2.4.sd = sd(data_omit$CT2.4props)
set.seed(1021)
ps_2.4.one = matchit(tx_typ ~ GENDER + diab + on_vent_trr +
                    PORTAL_VEIN_TRR + etiology + bmi_tcr + age +
                    race_b + race_l + race_o +
                    region_11 + region_10 + region_9 + region_8 +

```

```

        region_7 + region_6 + region_5 + region_4 +
        region_3 + region_2 + region_1,
        data=data_omit,
        method="nearest",
        distance=data_omit$CT2.4props,
        caliper = 0.2*ps_2.4.sd)
ps_2.4.one.data = data_with_match_index_one(ps_2.4.one)

#treatment effect - Classification Tree, cp=0.01 (matching)
#matched pseudo-population using clustering on the matched set
cox1.3<-coxph(Surv(pyear,death2) ~ tx_typ + cluster(match_index),
              data=ps_2.4.one.data)
summary(cox1.3)

#Kaplan-Meier curves and Log_rank test
ps_2.4.one.data$tx_typ<-factor(ps_2.4.one.data$tx_typ,
                              levels=c("0","1"),
                              labels=c("LTA","SLKT"))
surv_object<-Surv(time = ps_2.4.one.data$pyear,
                  event=ps_2.4.one.data$death2)
fit1 <- survfit(surv_object ~ tx_typ, data=ps_2.4.one.data)
ggsurvplot(fit1,
           data=ps_2.4.one.data,
           pval=F)

#RANDOM FOREST
#1:1 nearest neighbor matching
ps_3.sd = sd(data_omit$RFprops)
set.seed(1021)
ps_3.one = matchit(tx_typ ~ GENDER + diab + on_vent_trr +
                  PORTAL_VEIN_TRR + etiology + bmi_tcr + age +
                  race_b + race_l + race_o +
                  region_11 + region_10 + region_9 + region_8 +
                  region_7 + region_6 + region_5 + region_4 +
                  region_3 + region_2 + region_1,
                  data=data_omit,
                  method="nearest",
                  distance=data_omit$RFprops,
                  caliper = 0.2*ps_3.sd)
ps_3.one.data = data_with_match_index_one(ps_3.one)

#treatment effect - Random forest (matching)
#matched pseudo-population using clustering on the matched set
cox1.3<-coxph(Surv(pyear,death2) ~ tx_typ + cluster(match_index),
              data=ps_3.one.data)
summary(cox1.3)

#Kaplan-Meier curves and Log_rank test
ps_3.one.data$tx_typ<-factor(ps_3.one.data$tx_typ,
                              levels=c("0","1"),
                              labels=c("LTA","SLKT"))
surv_object<-Surv(time = ps_3.one.data$pyear,
                  event=ps_3.one.data$death2)
fit1 <- survfit(surv_object ~ tx_typ, data=ps_3.one.data)
ggsurvplot(fit1,

```

```

    data=ps_3.one.data,
    pval=F)

#---STRATIFICATION---

#LOGISTIC REGRESSION
#stratification
data_omit<-cbind(data_omit, strat.LR = (cut(data_omit$LRprops,
                                           breaks=c(quantile(data_omit$LRprops,
                                                                probs = seq(0, 1, by =
0.20))),
                                           labels=c("1st", "2nd", "3rd", "4th", "5th"),
                                           include.lowest=TRUE)))

#treatment effect - Logistic Regression (stratification)
#stratified pseudo-population analyzed as stratified data
cox1.2<-coxph(Surv(pyear, death2)~tx_typ + strata(strat.LR),
              data=data_omit)
summary(cox1.2)

#Kaplan-Meier curves and Log_rank test
dat<-data_omit
dat$tx_typ<-factor(dat$tx_typ,
                  labels=c("LTA", "SLKT"))
surv_object<-Surv(time = dat$pyear,
                  event=dat$death2)
fit1 <- survfit(surv_object ~ tx_typ + strata(strat.LR),
               data=dat)
ggsurvplot(fit1,
           data=dat,
           risk.table=F,
           legend='bottom',
           pval=T) +
  guides(colour = guide_legend(nrow = 5))

g<-list()
g[[1]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.LR)=1st",
                               "tx_typ=SLKT, strata(strat.LR)=1st")],
                  group.by=fit1$tx_typ,
                  title="1st Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[2]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.LR)=2nd",
                               "tx_typ=SLKT, strata(strat.LR)=2nd")],
                  group.by=fit1$tx_typ,
                  title="2nd Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[3]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.LR)=3rd",
                               "tx_typ=SLKT, strata(strat.LR)=3rd")],
                  group.by=fit1$tx_typ,
                  title="3rd Strata",
                  data=dat, conf.int = F,

```

```

        risk.table=F,
        legend='none')
g[[4]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.LR)=4th",
                                "tx_typ=SLKT, strata(strat.LR)=4th")],
                  group.by=fit1$tx_typ,
                  title="4th Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[5]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.LR)=5th",
                                "tx_typ=SLKT, strata(strat.LR)=5th")],
                  group.by=fit1$tx_typ,
                  title="5th Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='bottom', legend.labs = c("LTA", "SLKT")) +
  guides(colour = guide_legend(nrow = 1))
arrange_ggsurvplots(g, ncol=2, nrow=3)
remove(g)

#CLASSIFICATION TREE, CP=0
data_omit<-cbind(data_omit, strat.CT2.1=(cut(data_omit$CT2.1props,
                                             breaks=c(quantile(data_omit$CT2.1props,
                                                                 probs=seq(0, 1, by =
0.20))),
                                             labels=c("1st", "2nd", "3rd", "4th", "5th"),
                                             include.lowest=TRUE)))

#Kaplan-Meier curves and Log_rank test
dat<-data_omit
dat$tx_typ<-factor(dat$tx_typ,
                  labels=c("LTA", "SLKT"))
surv_object<-Surv(time = dat$year,
                  event=dat$death2)
fit1 <- survfit(surv_object ~ tx_typ + strata(strat.CT2.1),
               data=dat)
ggsurvplot(fit1,
           data=dat,
           risk.table=F,
           legend='bottom',
           pval=T) +
  guides(colour = guide_legend(nrow = 5))

g<-list()
g[[1]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.1)=1st",
                                "tx_typ=SLKT, strata(strat.CT2.1)=1st")],
                  group.by=fit1$tx_typ,
                  title="1st Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[2]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.1)=2nd",
                                "tx_typ=SLKT, strata(strat.CT2.1)=2nd")],
                  group.by=fit1$tx_typ,

```

```

        title="2nd Strata",
        data=dat, conf.int = F,
        risk.table=F,
        legend='none')
g[[3]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.1)=3rd",
                                "tx_typ=SLKT, strata(strat.CT2.1)=3rd")],
                  group.by=fit1$tx_typ,
                  title="3rd Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[4]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.1)=4th",
                                "tx_typ=SLKT, strata(strat.CT2.1)=4th")],
                  group.by=fit1$tx_typ,
                  title="4th Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[5]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.1)=5th",
                                "tx_typ=SLKT, strata(strat.CT2.1)=5th")],
                  group.by=fit1$tx_typ,
                  title="5th Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='bottom', legend.labs = c("LTA", "SLKT")) +
  guides(colour = guide_legend(nrow = 1))
arrange_ggsurvplots(g, ncol=2, nrow=3)
remove(g)

#CLASSIFICATION TREE, CP=0.0157
data_omit<-cbind(data_omit, strat.CT2.2=(cut(data_omit$CT2.2props,
                                             breaks=c(quantile(data_omit$CT2.2props,
                                                                 probs=seq(0, 1, by =
0.20))),
                                             labels=c("1st", "2nd", "3rd", "4th", "5th"),
                                             include.lowest=TRUE)))

#treatment effect - Classification Tree, cp=0.0157 (stratification)
#stratified pseudo-population analyzed as stratified data
cox1.2<-coxph(Surv(pyear,death2) ~ tx_typ + strata(strat.CT2.2),
              data=data_omit)
summary(cox1.2)

#Kaplan-Meier curves and Log_rank test
dat<-data_omit
dat$tx_typ<-factor(dat$tx_typ,
                  labels=c("LTA", "SLKT"))
surv_object<-Surv(time = dat$pyear,
                  event=dat$death2)
fit1 <- survfit(surv_object ~ tx_typ + strata(strat.CT2.2),
               data=dat)
ggsurvplot(fit1,
           data=dat,
           risk.table=F,
           legend='bottom',

```

```

      pval=T) +
  guides(colour = guide_legend(nrow = 5))

g<-list()
g[[1]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.2)=1st",
                                "tx_typ=SLKT, strata(strat.CT2.2)=1st")],
                  group.by=fit1$tx_typ,
                  title="1st Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[2]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.2)=2nd",
                                "tx_typ=SLKT, strata(strat.CT2.2)=2nd")],
                  group.by=fit1$tx_typ,
                  title="2nd Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[3]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.2)=3rd",
                                "tx_typ=SLKT, strata(strat.CT2.2)=3rd")],
                  group.by=fit1$tx_typ,
                  title="3rd Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[4]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.2)=4th",
                                "tx_typ=SLKT, strata(strat.CT2.2)=4th")],
                  group.by=fit1$tx_typ,
                  title="4th Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[5]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.2)=5th",
                                "tx_typ=SLKT, strata(strat.CT2.2)=5th")],
                  group.by=fit1$tx_typ,
                  title="5th Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='bottom', legend.labs = c("LTA", "SLKT")) +
  guides(colour = guide_legend(nrow = 1))
arrange_ggsurvplots(g, ncol=2, nrow=3)
remove(g)

#CLASSIFICATION TREE, CP=0.0065
data_omit<-cbind(data_omit, strat.CT2.3=(cut(data_omit$CT2.3props,
                                             breaks=c(quantile(data_omit$CT2.3props,
                                                                 prob=seq(0, 1, by =
0.20))),
                                             labels=c("1st", "2nd", "3rd", "4th", "5th"),
                                             include.lowest=TRUE)))

#treatment effect - Classification Tree, cp=0.0065 (stratification)
#stratified pseudo-population analyzed as stratified data
cox1.2<-coxph(Surv(pyear,death2) ~ tx_typ + strata(strat.CT2.3),
              data=data_omit)

```

```

summary(cox1.2)

#Kaplan-Meier curves and Log_rank test
dat<-data_omit
dat$tx_typ<-factor(dat$tx_typ,
                  labels=c("LTA","SLKT"))
surv_object<-Surv(time = dat$year,
                 event=dat$death2)
fit1 <- survfit(surv_object ~ tx_typ + strata(strat.CT2.3),
              data=dat)
ggsurvplot(fit1,
          data=dat,
          risk.table=F,
          legend='bottom',
          pval=T) +
  guides(colour = guide_legend(nrow = 5))

g<-list()
g[[1]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.3)=1st",
                              "tx_typ=SLKT, strata(strat.CT2.3)=1st")],
                group.by=fit1$tx_typ,
                title="1st Strata",
                data=dat, conf.int = F,
                risk.table=F,
                legend='none')
g[[2]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.3)=2nd",
                              "tx_typ=SLKT, strata(strat.CT2.3)=2nd")],
                group.by=fit1$tx_typ,
                title="2nd Strata",
                data=dat, conf.int = F,
                risk.table=F,
                legend='none')
g[[3]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.3)=3rd",
                              "tx_typ=SLKT, strata(strat.CT2.3)=3rd")],
                group.by=fit1$tx_typ,
                title="3rd Strata",
                data=dat, conf.int = F,
                risk.table=F,
                legend='none')
g[[4]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.3)=4th",
                              "tx_typ=SLKT, strata(strat.CT2.3)=4th")],
                group.by=fit1$tx_typ,
                title="4th Strata",
                data=dat, conf.int = F,
                risk.table=F,
                legend='none')
g[[5]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.3)=5th",
                              "tx_typ=SLKT, strata(strat.CT2.3)=5th")],
                group.by=fit1$tx_typ,
                title="5th Strata",
                data=dat, conf.int = F,
                risk.table=F,
                legend='bottom', legend.labs = c("LTA", "SLKT")) +
  guides(colour = guide_legend(nrow = 1))
arrange_ggsurvplots(g, ncol=2, nrow=3)

```

```

remove(g)

#CLASSIFICATION TREE, CP=0.01
data_omit<-cbind(data_omit, strat.CT2.4=(cut(data_omit$CT2.4props,
                                           breaks=c(quantile(data_omit$CT2.4props,
                                                                probs=seq(0, 1, by =
0.20))),
                                           labels=c("1st", "2nd", "3rd", "4th", "5th"),
                                           include.lowest=TRUE)))

#treatment effect - Classification Tree, cp=0.01 (stratification)
#stratified pseudo-population analyzed as stratified data
cox1.2<-coxph(Surv(pyear, death2) ~ tx_typ + strata(strat.CT2.4),
              data=data_omit)
summary(cox1.2)

#Kaplan-Meier curves and Log_rank test
dat<-data_omit
dat$tx_typ<-factor(dat$tx_typ,
                  labels=c("LTA", "SLKT"))
surv_object<-Surv(time = dat$pyear,
                  event=dat$death2)
fit1 <- survfit(surv_object ~ tx_typ + strata(strat.CT2.4),
               data=dat)
ggsurvplot(fit1,
           data=dat,
           risk.table=F,
           legend='bottom',
           pval=T) +
  guides(colour = guide_legend(nrow = 5))

g<-list()
g[[1]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.4)=1st",
                                "tx_typ=SLKT, strata(strat.CT2.4)=1st")],
                  group.by=fit1$tx_typ,
                  title="1st Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[2]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.4)=2nd",
                                "tx_typ=SLKT, strata(strat.CT2.4)=2nd")],
                  group.by=fit1$tx_typ,
                  title="2nd Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[3]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.4)=3rd",
                                "tx_typ=SLKT, strata(strat.CT2.4)=3rd")],
                  group.by=fit1$tx_typ,
                  title="3rd Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[4]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.4)=4th",
                                "tx_typ=SLKT, strata(strat.CT2.4)=4th")],

```

```

        group.by=fit1$tx_typ,
        title="4th Strata",
        data=dat, conf.int = F,
        risk.table=F,
        legend='none')
g[[5]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.CT2.4)=5th",
                                "tx_typ=SLKT, strata(strat.CT2.4)=5th")],
        group.by=fit1$tx_typ,
        title="5th Strata",
        data=dat, conf.int = F,
        risk.table=F,
        legend='bottom', legend.labs = c("LTA", "SLKT")) +
  guides(colour = guide_legend(nrow = 1))
arrange_ggsurvplots(g, ncol=2, nrow=3)
remove(g)

#RANDOM FOREST
data_omit<-cbind(data_omit, strat.RF=(cut(data_omit$RFprops,
                                          breaks=c(quantile(data_omit$RFprops,
                                                                probs=seq(0, 1, by =
0.20))),
                                          labels=c("1st", "2nd", "3rd", "4th", "5th"),
                                          include.lowest=TRUE)))

#treatment effect - random forest (stratification)
#stratified pseudo-population analyzed as stratified data
cox1.2<-coxph(Surv(pyear,death2) ~ tx_typ + strata(strat.RF),
              data=data_omit)
summary(cox1.2)

#Kaplan-Meier curves and Log_rank test
dat<-data_omit
dat$tx_typ<-factor(dat$tx_typ,
                  labels=c("LTA", "SLKT"))
surv_object<-Surv(time = dat$pyear,
                  event=dat$death2)
fit1 <- survfit(surv_object ~ tx_typ + strata(strat.RF),
               data=dat)
ggsurvplot(fit1,
           data=dat,
           risk.table=F,
           legend='bottom',
           pval=T) +
  guides(colour = guide_legend(nrow = 5))

g<-list()
g[[1]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.RF)=1st",
                                "tx_typ=SLKT, strata(strat.RF)=1st")],
        group.by=fit1$tx_typ,
        title="1st Strata",
        data=dat, conf.int = F,
        risk.table=F,
        legend='none')
g[[2]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.RF)=2nd",
                                "tx_typ=SLKT, strata(strat.RF)=2nd")],

```

```

        group.by=fit1$tx_typ,
        title="2nd Strata",
        data=dat, conf.int = F,
        risk.table=F,
        legend='none')
g[[3]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.RF)=3rd",
                                "tx_typ=SLKT, strata(strat.RF)=3rd")],
                  group.by=fit1$tx_typ,
                  title="3rd Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[4]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.RF)=4th",
                                "tx_typ=SLKT, strata(strat.RF)=4th")],
                  group.by=fit1$tx_typ,
                  title="4th Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='none')
g[[5]]<-ggsurvplot(fit1[strata=c("tx_typ=LTA, strata(strat.RF)=5th",
                                "tx_typ=SLKT, strata(strat.RF)=5th")],
                  group.by=fit1$tx_typ,
                  title="5th Strata",
                  data=dat, conf.int = F,
                  risk.table=F,
                  legend='bottom', legend.labs = c("LTA", "SLKT")) +
  guides(colour = guide_legend(nrow = 1))
arrange_ggsurvplots(g, ncol=2, nrow=3)
remove(g)

#---IPTW---

library(survey)

#LOGISTIC REGRESSION
#iptw
dat<-data_omit
#dat<-dat[c(1:42)]
dat$tx_typ2<-as.numeric(dat$tx_typ)
dat$tx_typ2<-ifelse(dat$tx_typ2==1,0,1)

#create weights and add column to dataset
dat<-cbind(dat,weights=((dat$tx_typ2/dat$LRprops) + (1-dat$tx_typ2)/(1-dat$LRprops)))

#add to new dataset that includes all weights
dat.wts<-dat
#names(dat.wts)[45]<-"wts.LR"
dat.wts<-cbind(dat.wts,wts.LR=((dat.wts$tx_typ2/dat.wts$LRprops) +
                              (1-dat.wts$tx_typ2)/(1-dat.wts$LRprops)))

#iptw
des1 <- svydesign(id = ~1, weights = ~weights , data=dat)

#treatment effect - Logistic Regression (IPTW)
#IPTW pseudo-population analyzed with survey weights

```

```

cox1.4<- svycoxph(Surv(pyear,death2) ~ tx_typ, design=des1)
summary(cox1.4)

#Kaplan-Meier curves
fit2<-svykm(Surv(pyear,death2) ~ tx_typ,
            design=des1)
plot(fit2,
     pars=list(lty=c(2,1), col=c("red","blue")),
     xlab="time",
     ylab="Proportion surviving",
     ylim=c(0,1))

#CLASSIFICATION TREE, CP=0
#iptw
dat<-data_omit
#dat<-dat[c(1:43)]
dat$tx_typ2<-as.numeric(dat$tx_typ)
dat$tx_typ2<-ifelse(dat$tx_typ2==1,0,1)

#create weights and add column to dataset
dat<-cbind(dat,weights=((dat$tx_typ2/dat$CT2.1props) + (1-dat$tx_typ2)/(1-
dat$CT2.1props)))
#remove missing weights - now n=1895
dat<-na.omit(dat)

#add to new dataset that includes all weights
dat.wts<-cbind(dat.wts,wts.CT2.1=((dat.wts$tx_typ2/dat.wts$CT2.1props) +
(1-dat.wts$tx_typ2)/(1-dat.wts$CT2.1props)))

#iptw
des1 <- svydesign(id = ~1, weights = ~weights , data=dat)

#treatment effect - Classification Tree, cp=0.0 (IPTW)
#IPTW pseudo-population analyzed with survey weights
cox1.4<- svycoxph(Surv(pyear,death2) ~ tx_typ, design=des1)
summary(cox1.4)

#Kaplan-Meier curves
fit2<-svykm(Surv(pyear,death2) ~ tx_typ,
            design=des1)
plot(fit2,
     pars=list(lty=c(2,1), col=c("red","blue")),
     xlab="time",
     ylab="Proportion surviving",
     ylim=c(0,1))

#CLASSIFICATION TREE, CP=0.0157
#iptw
dat<-data_omit
#dat<-dat[c(1:43)]
dat$tx_typ2<-as.numeric(dat$tx_typ)
dat$tx_typ2<-ifelse(dat$tx_typ2==1,0,1)

```

```

#create weights and add column to dataset
dat<-cbind(dat,weights=((dat$tx_typ2/dat$CT2.2props) + (1-dat$tx_typ2)/(1-
dat$CT2.2props)))
#remove missing weights - now n=1895
dat<-na.omit(dat)

#add to new dataset that includes all weights
dat.wts<-cbind(dat.wts,wts.CT2.2=((dat.wts$tx_typ2/dat.wts$CT2.2props) +
(1-dat.wts$tx_typ2)/(1-dat.wts$CT2.2props)))

#iptw
des1 <- svydesign(id = ~1, weights = ~weights , data=dat)

#treatment effect - Classification Tree, cp=0.0 (IPTW)
#IPTW pseudo-population analyzed with survey weights
cox1.4<- svycoxph(Surv(pyear,death2) ~ tx_typ, design=des1)
summary(cox1.4)

#Kaplan-Meier curves
fit2<-svykm(Surv(pyear,death2) ~ tx_typ,
design=des1)
plot(fit2,
pars=list(lty=c(2,1), col=c("red","blue")),
xlab="time",
ylab="Proportion surviving",
ylim=c(0,1))

#CLASSIFICATION TREE, CP=0.0065
#iptw
dat<-data_omit
#dat<-dat[c(1:43)]
dat$tx_typ2<-as.numeric(dat$tx_typ)
dat$tx_typ2<-ifelse(dat$tx_typ2==1,0,1)

#create weights and add column to dataset
dat<-cbind(dat,weights=((dat$tx_typ2/dat$CT2.3props) + (1-dat$tx_typ2)/(1-
dat$CT2.3props)))
#remove missing weights - now n=1895
dat<-na.omit(dat)

#add to new dataset that includes all weights
dat.wts<-cbind(dat.wts,wts.CT2.3=((dat.wts$tx_typ2/dat.wts$CT2.3props) +
(1-dat.wts$tx_typ2)/(1-dat.wts$CT2.3props)))

#iptw
des1 <- svydesign(id = ~1, weights = ~weights , data=dat)

#treatment effect - Classification Tree, cp=0.0 (IPTW)
#IPTW pseudo-population analyzed with survey weights
cox1.4<- svycoxph(Surv(pyear,death2) ~ tx_typ, design=des1)
summary(cox1.4)

#Kaplan-Meier curves
fit2<-svykm(Surv(pyear,death2) ~ tx_typ,

```

```

        design=des1)
plot(fit2,
     pars=list(lty=c(2,1), col=c("red","blue")),
     xlab="time",
     ylab="Proportion surviving",
     ylim=c(0,1))

#CLASSIFICATION TREE, CP=0.01
#iptw
dat<-data_omit
#dat<-dat[c(1:43)]
dat$tx_typ2<-as.numeric(dat$tx_typ)
dat$tx_typ2<-ifelse(dat$tx_typ2==1,0,1)

#create weights and add column to dataset
dat<-cbind(dat,weights=((dat$tx_typ2/dat$CT2.4props) + (1-dat$tx_typ2)/(1-
dat$CT2.4props)))
#remove missing weights - now n=1895
dat<-na.omit(dat)

#add to new dataset that includes all weights
dat.wts<-cbind(dat.wts,wts.CT2.4=((dat.wts$tx_typ2/dat.wts$CT2.4props) +
(1-dat.wts$tx_typ2)/(1-dat.wts$CT2.4props)))

#iptw
des1 <- svydesign(id = ~1, weights = ~weights , data=dat)

#treatment effect - Classification Tree, cp=0.0 (IPTW)
#IPTW pseudo-population analyzed with survey weights
cox1.4<- svycoxph(Surv(pyear,death2) ~ tx_typ, design=des1)
summary(cox1.4)

#Kaplan-Meier curves
fit2<-svykm(Surv(pyear,death2) ~ tx_typ,
            design=des1)
plot(fit2,
     pars=list(lty=c(2,1), col=c("red","blue")),
     xlab="time",
     ylab="Proportion surviving",
     ylim=c(0,1))

#RANDOM FOREST
#iptw
dat<-data_omit
#dat<-dat[c(1:43)]
dat$tx_typ2<-as.numeric(dat$tx_typ)
dat$tx_typ2<-ifelse(dat$tx_typ2==1,0,1)

#create weights and add column to dataset
dat<-cbind(dat,weights=((dat$tx_typ2/dat$RFprops) + (1-dat$tx_typ2)/(1-dat$RFprops)))
#remove missing weights - now n=1895
dat<-na.omit(dat)

```

```

#add to new dataset that includes all weights
dat.wts<-cbind(dat.wts,wts.RF=((dat.wts$tx_typ2/dat.wts$RFprops) +
                        (1-dat.wts$tx_typ2)/(1-dat.wts$RFprops)))

#iptw
des1 <- svydesign(id = ~1, weights = ~weights , data=dat)

#treatment effect - Classification Tree, cp=0.0 (IPTW)
#IPTW pseudo-population analyzed with survey weights
cox1.4<- svycoxph(Surv(pyear,death2) ~ tx_typ, design=des1)
summary(cox1.4)

#Kaplan-Meier curves
fit2<-svykm(Surv(pyear,death2) ~ tx_typ,
            design=des1)
plot(fit2,
     pars=list(lty=c(2,1), col=c("red","blue")),
     xlab="time",
     ylab="Proportion surviving",
     ylim=c(0,1))

#####

#Cox PH with all of the predictors that went into the PS
# as adjustment variables (but not using the PS)
cox1.1<-coxph(Surv(pyear,death2) ~ tx_typ +
              GENDER + diab + region + on_vent_trr +
              PORTAL_VEIN_TRR + etiology + race + bmi_tcr + age,
              data=data_omit)
summary(cox1.1)

#Kaplan-Meier curves and Log_rank test
dat<-data_omit
dat$tx_typ<-factor(dat$tx_typ,
                  labels=c("LTA","SLKT"))

#Cox PH with only the PS as a covariate

#LOGISTIC REGRESSION
cox1.1<-coxph(Surv(pyear,death2) ~ tx_typ + LRprops,
              data=data_omit)
summary(cox1.1)

#CLASSIFICATION TREE, CP=0
cox1.1<-coxph(Surv(pyear,death2) ~ tx_typ + CT2.1props,
              data=data_omit)
summary(cox1.1)

#CLASSIFICATION TREE, CP=0.0157
cox1.1<-coxph(Surv(pyear,death2) ~ tx_typ + CT2.2props,
              data=data_omit)
summary(cox1.1)

```

```

#Kaplan-Meier curves and Log_rank test
dat<-data_omit
dat$tx_typ<-factor(dat$tx_typ,
                  labels=c("LTA","SLKT"))
surv_object<-Surv(time = dat$pyear,
                  event=dat$death2)
fit1 <- survfit(surv_object ~ tx_typ + CT2.2props,
               data=dat)
ggsurvplot(fit1,
           data=dat,
           risk.table=F,
           legend='bottom',
           pval=T) +
  guides(colour = guide_legend(nrow = 6))

#CLASSIFICATION TREE, CP=0.0065
cox1.1<-coxph(Surv(pyear,death2) ~ tx_typ + CT2.3props,
              data=data_omit)
summary(cox1.1)

#CLASSIFICATION TREE, CP=0.01
cox1.1<-coxph(Surv(pyear,death2) ~ tx_typ + CT2.4props,
              data=data_omit)
summary(cox1.1)

#RANDOM FOREST
cox1.1<-coxph(Surv(pyear,death2) ~ tx_typ + RFprops,
              data=data_omit)
summary(cox1.1)

#####

#COVARIATE BALANCE
library(cobalt)

bal.tab(ps_1.one)
bal.plot(ps_1.one,
        var.name = "age",
        which = "both",
        type="hist",
        mirror = T)
bal.plot(ps_3.one,
        var.name = "diab1",
        which = "both")

v2 <- data.frame(old = c("GENDER0", "on_vent_trr1", "PORTAL_VEIN_TRR1",
                        "etiology1",
                        "race_0", "race_1", "race_2",
                        "bmi_tcr",
                        "diab0", "diab1", "diab2",
                        "race_o1", "race_l1", "race_b1",
                        "region1", "region2", "region3", "region4",
                        "region5", "region6", "region7", "region8",
                        "region9", "region10", "region11",

```

```

        "race0", "race1", "race2",
        "region_11", "region_21", "region_31", "region_41",
        "region_51", "region_61", "region_71", "region_81",
        "region_91", "region_101", "region_111"),
new = c("gender (female)", "on ventilator", "portal vein
thrombosis",
        "etiology (NAFLD)",
        "race (other)", "race (hispanic/latino)", "race (black)",
        "bmi",
        "diabetes (none)", "diabetes (type II)", "diabetes (type
I)",
        "race (other)", "race (hispanic/latino)", "race (black)",
        "region_1", "region_2", "region_3", "region_4",
        "region_5", "region_6", "region_7", "region_8",
        "region_9", "region_10", "region_11",
        "race (other)", "race (hispanic/latino)", "race (black)",
        "region_1", "region_2", "region_3", "region_4",
        "region_5", "region_6", "region_7", "region_8",
        "region_9", "region_10", "region_11"))

#covariate balance summary love plots (1:1 matching)
bal.tab(ps_1.one)
love.plot(bal.tab(ps_1.one),
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v2)
print(bal.tab(ps_1.one),
      which="both",
      un=T)

bal.tab(ps_2.1.one)
love.plot(bal.tab(ps_2.1.one),
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v2)
print(bal.tab(ps_2.1.one),
      which="both",
      un=T)

bal.tab(ps_2.2.one)
love.plot(bal.tab(ps_2.2.one),
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v2)
print(bal.tab(ps_2.2.one),
      which="both",
      un=T)

bal.tab(ps_2.3.one)
love.plot(bal.tab(ps_2.3.one),
          threshold = .1,
          shapes=c(1,16),

```

```

        limits=c(0,.5),
        var.names=v2)
print(bal.tab(ps_2.3.one),
      which="both",
      un=T)

bal.tab(ps_2.4.one)
love.plot(bal.tab(ps_2.4.one),
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v2)
print(bal.tab(ps_2.4.one),
      which="both",
      un=T)

bal.tab(ps_3.one)
love.plot(bal.tab(ps_3.one),
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v2)
print(bal.tab(ps_3.one),
      which="both",
      un=T)

#covariate balance summary love plots (Stratification)
covs <- subset(data_omit, select=c(2:9,14))
covs_tree <- subset(data_omit, select=c(2,4:6,8,9,14:28))
v <- data.frame(old = c("GENDER_1", "on_vent_trr_1", "PORTAL_VEIN_TRR_1",
                       "etiology_1", "race_0", "race_1", "race_2", "bmi_tcr",
                       "diab_0", "diab_1", "diab_2",
                       "race_o_1", "race_l", "race_b",
                       "region_1_1", "region_2_1", "region_3_1", "region_4_1",
                       "region_5_1", "region_6_1", "region_7_1", "region_8_1",
                       "region_9_1", "region_10_1", "region_11_1"),
                 new = c("gender (male)", "on ventilator", "portal vein thrombosis",
                       "etiology (NAFLD)", "race (other)", "race (hispanic/latino)",
                       "race (black)", "bmi", "diabetes (none)",
                       "diabetes (type II)", "diabetes (type I)",
                       "race (other)", "race (hispanic/latino)", "race (black)",
                       "region_1", "region_2", "region_3", "region_4",
                       "region_5", "region_6", "region_7", "region_8",
                       "region_9", "region_10", "region_11"))

#logit
lr.s.bal<-bal.tab(covs=covs,
                 data=data_omit,
                 treat="tx_typ",
                 match.strata="strat.LR",
                 distance="LRprops")
love.plot(lr.s.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),

```

```

        var.names=v) #colors=c("#F8766D", "#00BFC4"))
print(lr.s.bal,
      which="both",
      disp.means=F,
      un=T)

#tree (cp=0.0)
ct1.s.bal<-bal.tab(covs=covs_tree,
                  data=data_omit,
                  treat="tx_typ",
                  match.strata="strat.CT2.1",
                  distance="CT2.1props")
love.plot(ct1.s.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v)
print(ct1.s.bal,
      which="both",
      un=T)

#tree (cp=0.0157)
ct2.s.bal<-bal.tab(covs=covs,
                  data=data_omit,
                  treat="tx_typ",
                  match.strata="strat.CT2.2",
                  distance="CT2.2props")
love.plot(ct2.s.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v)
print(ct2.s.bal,
      which="both",
      un=T)

#tree (cp=0.0065)
ct3.s.bal<-bal.tab(covs=covs,
                  data=data_omit,
                  treat="tx_typ",
                  match.strata="strat.CT2.3",
                  distance="CT2.3props")
love.plot(ct3.s.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v)
print(ct3.s.bal,
      which="both",
      un=T)

#tree (cp=0.01)
ct4.s.bal<-bal.tab(covs=covs,
                  data=data_omit,
                  treat="tx_typ",

```

```

                match.strata="strat.CT2.4",
                distance="CT2.4props")
love.plot(ct4.s.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v)
print(ct4.s.bal,
      which="both",
      un=T)

#forest
rf.s.bal<-bal.tab(covs=covs,
                 data=data_omit,
                 treat="tx_typ",
                 match.strata="strat.RF",
                 distance="RFprops")
love.plot(rf.s.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v)
print(rf.s.bal,
      which="both",
      un=T)

#tx_typ ~ "bmi_tcr" + "age" + "on_vent_trr" +
# "PORTAL_VEIN_TRR" + "race" + "GENDER" +
# "region" + "etiology" + "diab"

#covariate balance summary love plots (IPTW)
covs.wt <- subset(dat.wts, select=c(2:9,14))
covs.wt_tree <- subset(dat.wts, select=c(2,4:6,8,9,14:28))

#logit
lr.i.bal<-bal.tab(covs=covs.wt,
                 data=dat.wts,
                 treat="tx_typ",
                 weights="wts.LR",
                 distance="LRprops")
love.plot(lr.i.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v)
print(lr.i.bal,
      which="both",
      un=T)

#tree (cp=0.0)
tot<-dat.wts
tot<-tot[c(1:44,47)]
tot<-na.omit(tot)
covs.wt <- subset(tot, select=c(2:9,14))
covs.wt_tree <- subset(tot, select=c(2,4:6,8,9,14:28))

```

```

ct1.i.bal<-bal.tab(covs=covs.wt_tree,
                  data=tot,
                  treat="tx_typ",
                  weights="wts.CT2.1",
                  distance="CT2.1props")
love.plot(ct1.i.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v)
print(ct1.i.bal,
      which="both",
      un=T)

#tree (cp=0.0157)
tot<-dat.wts
tot<-tot[c(1:44,48)]
tot<-na.omit(tot)
covs.wt <- subset(tot, select=c(2:9,14))
covs.wt_tree <- subset(tot, select=c(2,4:6,8,9,14:28))

ct2.i.bal<-bal.tab(covs=covs.wt_tree,
                  data=tot,
                  treat="tx_typ",
                  weights="wts.CT2.2",
                  distance="CT2.2props")
love.plot(ct2.i.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v)
print(ct2.i.bal,
      which="both",
      un=T)

#tree (cp=0.0065)
tot<-dat.wts
tot<-tot[c(1:44,49)]
tot<-na.omit(tot)
covs.wt <- subset(tot, select=c(2:9,14))
covs.wt_tree <- subset(tot, select=c(2,4:6,8,9,14:28))

ct3.i.bal<-bal.tab(covs=covs.wt_tree,
                  data=tot,
                  treat="tx_typ",
                  weights="wts.CT2.3",
                  distance="CT2.3props")
love.plot(ct3.i.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v)
print(ct3.i.bal,
      which="both",

```

```

un=T)

#tree (cp=0.01)
tot<-dat.wts
tot<-tot[c(1:44,50)]
tot<-na.omit(tot)
covs.wt <- subset(tot, select=c(2:9,14))
covs.wt_tree <- subset(tot, select=c(2,4:6,8,9,14:28))

ct4.i.bal<-bal.tab(covs=covs.wt_tree,
                  data=tot,
                  treat="tx_typ",
                  weights="wts.CT2.4",
                  distance="CT2.4props")
love.plot(ct4.i.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v)
print(ct4.i.bal,
      which="both",
      un=T)

#forest
tot<-dat.wts
tot<-tot[c(1:44,51)]
tot<-na.omit(tot)
covs.wt <- subset(tot, select=c(2:9,14))
covs.wt_tree <- subset(tot, select=c(2,4:6,8,9,14:28))

rf.i.bal<-bal.tab(covs=covs.wt_tree,
                  data=tot,
                  treat="tx_typ",
                  weights="wts.RF",
                  distance="RFprops")
love.plot(rf.i.bal,
          threshold = .1,
          shapes=c(1,16),
          limits=c(0,.5),
          var.names=v)
print(rf.i.bal,
      which="both",
      un=T)

#####

```

## Bibliography

- Donald B Rubin (2005) Causal Inference Using Potential Outcomes, *Journal of the American Statistical Association*, 100:469, 322-331, DOI: 10.1198/016214504000001880
- Rosenbaum, P. R., & Rubin, D. B. (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, 41-55.
- Donald B Rubin (2005). Causal Inference Using Potential Outcomes, *Journal of the American Statistical Association*, 100:469, 322-331, DOI: 10.1198/016214504000001880
- Austin, P. C., & Stuart, E. A. (2015). Moving towards best practice when using inverse probability of treatment weighting (IPTW) using the propensity score to estimate causal treatment effects in observational studies. *Statistics in medicine*, 34(28), 3661-3679.
- Caliendo M, Kopeinig S. (2008). Some Practical Guidance for the Implementation of Propensity Score Matching. *Journal of Economic Surveys*. 22(1):31-72. DOI:10.1111/j.1467-6419.2007.00527.x.
- Rassen JA, Brookhart MA, Glynn RJ, Mittleman MA, Schneeweiss S. (2009). Instrumental variables I: instrumental variables exploit natural variation in nonexperimental data to estimate causal relationships. *J Clin Epidemiol*. 62(12):1226-32.
- Robins JM. (1999). Marginal structural models versus structural nested models as tools for causal inference. In: Halloran E, Berry D, eds. *Statistical Models in Epidemiology: The Environment and Clinical Trials*. New York: Springer-Verlag, 95-134.
- Young, J.G., Hernán, M.A., Picciotto, S. et al. (2010). Relation between three classes of structural models for the effect of a time-varying exposure on survival. *Lifetime Data Anal* 16: 71. <https://doi-org.pitt.idm.oclc.org/10.1007/s10985-009-9135-3>
- Miles CD, Westphal S, Liapakis A, Formica R. (2018). Simultaneous Liver-Kidney Transplantation: Impact on Liver Transplant Patients and the Kidney Transplant Waiting List. *Curr Transplant Rep*. 5(1):1-6.
- Sharma P, Shu X, Schaubel DE, Sung RS, Magee JC. (2016). Propensity score-based survival benefit of simultaneous liver-kidney transplant over liver transplant alone for recipients with pretransplant renal dysfunction. *Liver Transpl*. 22(1):71-9.
- Tanriover B, MacConmara MP, Parekh J, et al. (2016). Simultaneous liver kidney transplantation in liver transplant candidates with renal dysfunction: importance of creatinine levels, dialysis, and organ quality in survival. *Kidney Int Rep*. 1(4):221-229.

- Pham, P.T., Lunsford, K.E., Bunnapradist, S., Danovitch, G.M. (2016). Simultaneous liver-kidney transplantation or liver transplantation alone for patients in need of liver transplantation with renal dysfunction. *Curr Opin Organ Transplant*. 21(2): 194–200. doi: 10.1097/MOT.0000000000000299
- Peter C Austin. (2013). "Using methods from the data-mining and machine-learning literature for disease classification and prediction: a case study examining classification of heart failure subtypes" *Journal of Clinical Epidemiology* Vol. 66 Iss. 4
- Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*. Hoboken, NJ: Wiley.
- Atkinson, E. J., & Therneau, T. M. (2000). *An introduction to recursive partitioning using the RPART routines*. Rochester: Mayo Foundation.
- Burden A, Roche N, Miglio C, et al. (2017). An evaluation of exact matching and propensity score methods as applied in a comparative effectiveness study of inhaled corticosteroids in asthma. *Pragmat Obs Res*. 2017;8:15-30. doi:10.2147/POR.S122563
- Rosenbaum PR, Rubin DB. (1985). Constructing a control group using multivariate matched sampling methods that incorporate the propensity score. *Am Stat*. 39(1):33–38.
- Kalbfleisch, J. D., & Prentice, R. L. (2002). *The statistical analysis of failure time data*. Hoboken, NJ: Wiley-Interscience.
- K Luellen, Jason & R Shadish, William & Clark, M. (2006). *Propensity Scores An Introduction and Experimental Test*. Evaluation review. 29. 530-58. 10.1177/0193841X05275596.